Table of Contents

0 Overview **7**

Classical Results **13**

1 Graph Spectrum **15**
1.1 Adjacency Matrix . . . 15
1.2 Laplacian Matrix . . . 18
1.3 Normalized Adjacency and Laplacian Matrix . . . 20
1.4 Generalizations . . . 21
1.5 Problems . . . 22

2 Cheeger's Inequality **25**
2.1 Cheeger's Inequality for Graphs . . . 25
2.2 Easy Direction: Continuous Relaxation . . . 27
2.3 Hard Direction: Rounding Algorithm . . . 28
2.4 The Spectral Partitioning Algorithm . . . 32
2.5 Variants of Cheeger's Inequality . . . 34
2.6 Problems . . . 36

3 Random Walks on Graphs **39**
3.1 Markov Chains . . . 39
3.2 Random Walks on Undirected Graphs . . . 41
3.3 Spectral Analysis of Mixing Time for Undirected Graphs . . . 43
3.4 Applications of Random Walks . . . 47
3.5 Random Walks on Directed Graphs . . . 48
3.6 Problems . . . 50

4 Expander Graphs: Properties **53**
4.1 Expander Mixing Lemma . . . 54
4.2 Converse of Expander Mixing Lemma . . . 55
4.3 Graphs with Large Spectral Gap . . . 59
4.4 Small-Set Vertex Expansion . . . 62
4.5 Random Walks on Expander Graphs . . . 63
4.6 Problems . . . 64

5 Expander Graphs: Constructions **67**
5.1 Probabilistic Constructions . . . 67
5.2 Algebraic Constructions . . . 68
5.3 Combinatorial Constructions . . . 68
5.4 Problems . . . 73

6 Expander Graphs: Applications **77**
6.1 Pseudorandomness . . . 77
6.2 Constructing Efficient Networks and Algorithms . . . 78
6.3 Complexity Theory . . . 79
6.4 Error Correcting Codes . . . 80
6.5 Problems . . . 84

I Generalizations of Cheeger's Inequality **87**

7 Higher-Order Cheeger Inequality **89**
7.1 Cheeger Rounding and Spectral Embedding . . . 90
7.2 Isotropy Condition and Clustering by Directions . . . 91
7.3 Ideal Case: Far Apart Clusters . . . 92
7.4 General Case: Space Partitioning and Smooth Localization . . . 95
7.5 Alternative Randomized Rounding Algorithms . . . 98

8 Improved Cheeger Inequality **101**
8.1 Intuition and Proof Outline . . . 102
8.2 Constructing $2k$-Step Approximation . . . 102
8.3 Rounding $2k$-Step Function . . . 105
8.4 Problems . . . 107

9 Fastest Mixing Time, Vertex Expansion, and Reweighted Eigenvalues **109**
9.1 Definitions and Statements . . . 109
9.2 Semidefinite Program for Fastest Mixing Time . . . 111
9.3 Easy Direction by Reduction . . . 115
9.4 Hard Direction by Dimension Reduction and Rounding . . . 116
9.5 Reweighted Higher Eigenvalues . . . 119
9.6 Reweighting Conjectures . . . 121
9.7 Problems . . . 121

10 Reweighted Eigenvalues for Directed Graphs and Hypergraphs **123**
10.1 Cheeger Inequality for Directed Vertex Expansion . . . 123
10.2 Fastest Mixing Time on General Markov Chains . . . 125
10.3 Cheeger Inequality for Directed Edge Conductance . . . 126
10.4 Proof Overview . . . 127
10.5 Cheeger-Type Inequalities for Hypergraphs . . . 129
10.6 Discussions and Open Questions . . . 130
10.7 Problems . . . 131

II Random Walks and Graph Decompositions **133**

11 Small Sparse Cuts from Random Walks **135**
11.1 Approximating Small-Set Edge Conductance . . . 135
11.2 Spectral Approach . . . 137
11.3 Power Method . . . 138
11.4 Cheeger Inequality for Small-Set Edge Conductance . . . 139
11.5 Local Graph Partitioning by Random Walks . . . 140
11.6 Hard Example for Local Graph Partitioning . . . 143
11.7 Problems . . . 143

12 Expander Decomposition **147**
12.1 Expander Decomposition from Recursive Sparse Cuts . . . 147
12.2 Expander Decomposition from Most-Balanced Sparse Cuts . . . 149
12.3 Fast Algorithms Using Approximate Balanced Sparse Cuts . . . 150
12.4 Fast Algorithms Using Flows and Fair Cuts . . . 152
12.5 Balanced Sparse Cuts from Local Graph Partitioning . . . 154
12.6 Expander Hierarchy and Applications . . . 155
12.7 Problems . . . 156

13 Threshold Rank Decomposition **159**
13.1 Small-Set Expansion via Subspace Enumeration . . . 160
13.2 Subexponential Algorithm for Small-Set Expansion Conjecture . . . 162
13.3 Low Threshold Rank Graph Decomposition . . . 163
13.4 Subexponential Algorithm for Unique Games Conjecture . . . 164
13.5 Small-Set Expanders . . . 167
13.6 Problems . . . 168

IIISpectral Sparsification and Applications **169**

14 Spectral Sparsification, Matrix Concentration, and Effective Resistance **171**
14.1 Cut Sparsification . . . 171
14.2 Spectral Sparsification . . . 173
14.3 Random Sampling Algorithm . . . 175
14.4 Fast Algorithm and Applications . . . 178
14.5 Effective Resistance . . . 180
14.6 Problems . . . 184

15 Spectral Sparsification via Potential Function **187**
15.1 Linear-Sized Spectral Sparsification . . . 187
15.2 Deterministic Algorithm and Polynomial Perspective . . . 188
15.3 Potential Functions . . . 189
15.4 Changes of Potential Values . . . 191
15.5 Averaging Argument . . . 192
15.6 Further Developments . . . 196

16 Spectral Sparsification and Algorithmic Discrepancy Theory **199**
16.1 Discrepancy Theory and Matrix Sparsification . . . 199

16.2 From Matrix Partial Coloring to Matrix Sparsification . . . 201
16.3 Potential Function from Regularized Optimization . . . 202
16.4 Matrix Partial Coloring by Discrepancy Walk . . . 205
16.5 Fast Algorithms . . . 208
16.6 Problems . . . 209

IV Cut Matching Games and Matrix Multiplicative Weight Update 211

17 Cut-Matching Game 213
17.1 Sparsest Cut and Expander Flows . . . 213
17.2 Cut-Matching Game . . . 215
17.3 Algorithmic Applications . . . 219

18 Matrix Multiplicative Weight Update 225
18.1 Online Decision and Multiplicative Weight Update Method . . . 225
18.2 Matrix Generalization . . . 226
18.3 Cut Player Strategy from Matrix Multiplicative Update . . . 229
18.4 Problems . . . 232

19 Fast $\sqrt{\log n}$-Approximation Algorithm for Edge Expansion 235
19.1 Primal Program and Geometric Analysis . . . 235
19.2 Dual Semidefinite Program and Expander Flows . . . 239
19.3 Primal-Dual Approach to Semidefinite Programs . . . 241
19.4 Efficient Implementation of Oracle . . . 245
19.5 Problems . . . 250

V Semidefinite Programming and Approximation Algorithms 253

20 Local-to-Global Correlation Rounding on Expander Graphs 255
20.1 Unique Games and Semidefinite Programming Relaxation . . . 255
20.2 Global Correlation Rounding . . . 256
20.3 Local-to-Global Correlations on Expander Graphs . . . 258

21 Local-to-Global Correlation Rounding on Low Threshold Rank Graphs 263
21.1 Conditioning on Lasserre SDP Hierarchy . . . 263
21.2 Local Correlation and Independent Randomized Rounding . . . 266
21.3 Global Correlation and Conditioning . . . 267
21.4 Local-to-Global Correlation of Low Threshold Rank Graphs . . . 269

22 Local-to-Global Correlation Rounding on General Graphs? 273
22.1 Toward Subexponential Algorithm for Sparsest Cut . . . 273
22.2 Toward Subexponential Algorithm for Other Problems . . . 278

VI Matrix Concentration and Semi-Random Problems 281

23 Matrix Concentration Inequalities 283

23.1 Chernoff Bounds . . . 283
23.2 Matrix Exponential and Logarithm . . . 284
23.3 Matrix Concentration Inequalities via Matrix Exponentials . . . 285
23.4 Matrix Chernoff Bounds and Matrix Bernstein Inequality . . . 287
23.5 An Elementary Approach to Matrix Concentration . . . 290
23.6 Sharper Matrix Concentration Inequalities . . . 293
23.7 Problems . . . 295

24 Refutation Algorithms for Semi-Random Constraint Satisfaction Problems 297
24.1 Random 2-XOR: Quadratic Form . . . 297
24.2 Random k-XOR: Tensor Flattening . . . 300
24.3 Semi-Random CSP: Reweighting Trick . . . 301
24.4 Random 3-XOR: Cauchy-Schwarz Trick . . . 304

25 Kikuchi Matrices and Applications 307
25.1 Smooth Tradeoff for CSP Refutation . . . 307
25.2 Lower Bounds for Locally Decodable Codes . . . 311
25.3 Hypergraph Moore Bound . . . 315
25.4 Tensor Principal Component Analysis . . . 318
25.5 Problems . . . 322

VI High-Dimensional Expanders and Mixing Time 327

26 High-Dimensional Expanders 329
26.1 Simplicial Complexes . . . 329
26.2 Local Spectral Expanders . . . 331
26.3 Oppenheim's Trickling Down Theorem . . . 333
26.4 Local-to-Global Method . . . 335
26.5 Problems . . . 338

27 Higher Order Random Walks 341
27.1 Random Walks on Simplicial Complexes . . . 341
27.2 Kaufman-Oppenheim Theorem and Matroid Expansion . . . 343
27.3 Spectral Gap Bound in Product Form . . . 344

28 Spectral and Entropic Independence 349
28.1 Spectral Independence . . . 349
28.2 Simplicial Complex for Glauber Dynamics . . . 351
28.3 Applications of Spectral Independence . . . 353
28.4 Analyzing Mixing Time Using Entropy . . . 354
28.5 Log-Sobolev Constant for Strongly Log-Concave Distribution . . . 358
28.6 Problems . . . 360

Appendix 365

A Linear Algebra 365
A.1 Eigenvalues and Eigenvectors . . . 365

A.2 Formulas and Inequalities 371

B Notations **375**

Chapter 0

Overview

This book begins with a review of classical results in spectral graph theory, followed by an exploration of several recent major developments, with a focus on algorithmic results.

Classical Results

After introducing basic concepts and results from linear algebra, we study Cheeger's inequality, a foundational result in spectral graph theory. This theorem states that the spectral gap[1] is large if and only if the graph expansion[2] is large. This connection between an algebraic quantity and a combinatorial property has three major applications.

1. Random Walks on Graphs

Analyzing the mixing time of random walks[3] is an important topic with numerous applications in random sampling and approximate counting [LPW06]. A basic result in spectral graph theory is that the mixing time of random walks is roughly equal to the inverse of the spectral gap. Cheeger's inequality thus implies that a graph has small mixing time if and only if it has large expansion, providing a combinatorial characterization that is useful for analyzing mixing time.

2. Expander Graphs

Expander graphs, typically defined as sparse graphs with large expansion, have surprisingly many applications in theoretical computer science and mathematics [HLW06]. Cheeger's inequality provides an efficient method to certify that a graph has large expansion, which is crucial in the construction of expander graphs. We will study the basic properties of expander graphs and Ramanujan graphs[4], the combinatorial construction called the zig-zag product, and several interesting applications of expander graphs.

[1] Defined as the difference between the first and second eigenvalues.

[2] Which quantifies how well a graph is connected by comparing the number of edges leaving a subset of vertices to the size of the subset.

[3] Defined as the number of steps required for the probability distribution on vertices to converge to the limiting distribution.

[4] Expander graphs with the maximum possible spectral gap, roughly speaking.

3. Graph Partitioning

The proof of Cheeger's inequality provides a fast algorithm to output a subset of vertices of approximately minimum expansion. This is known as the spectral partitioning algorithm, a widely used heuristic in practical graph partitioning applications with good performance, e.g. [SM00].

Recent Developments

The recent developments still center around these three primary themes, but introduce significantly new ideas and techniques, extending the reach of spectral graph theory. These topics illustrate how spectral graph theory has developed from a beautiful mathematical subject into a broad algorithmic framework, leading to fast and simple algorithms for a wide range of problems.

1. Generalizations of Cheeger's Inequality and Graph Partitioning

Spectral graph theory has a long history, but only in the last decade have researchers begun to explore graph partitioning using higher eigenvalues, inspired by the influential work on small-set expansion [ABS10]. We study the so-called higher-order Cheeger inequality [LOT14, LRTV12] and the improved Cheeger inequality [KLL$^+$13], which use higher eigenvalues to design multi-way graph partitioning algorithms and to analyze the classical spectral partitioning algorithm. Until recently, all Cheeger-type inequalities applied only to the edge expansion of undirected graphs. We discuss how to use reweighted eigenvalues to extend these inequalities to vertex expansion [KLT22], as well as to directed graphs and hypergraphs [LTW23].

2. Random Walks and Graph Decompositions

Random walks can be used to design local graph partitioning algorithms with a running time depending only on the output size, providing a valuable algorithmic tool for processing massive graphs [ST13]. We present a unified spectral analysis for this result and the small-set expansion result in [ABS10]. We study how these results can be used to develop graph decomposition algorithms, including expander decompositions, which are crucial tools for designing fast graph algorithms. We also study threshold rank decompositions and the subspace enumeration method [Kol10] for designing subexponential-time approximation algorithm for Unique Games [ABS10].

3. Spectral Sparsification and Applications

Expander graphs can be seen as sparse approximations of complete graphs. Spectral sparsification involves constructing a sparse graph that approximates the spectral properties of a dense graph [ST11]. The study of this problem has been highly productive, leading to significant results and techniques. The analysis of a natural random sampling algorithm [SS11] brought the tools of matrix concentration inequalities to the field, with important applications in designing fast algorithms for solving Laplacian linear equations [ST14]. The design of optimal spectral sparsification algorithms led to the potential function developed in [BSS14, ALO15], with applications in designing approximation algorithms and far-reaching consequences in mathematics [MSS14]. We will study these results as well as a new discrepancy-theoretic approach for constructing spectral sparsifiers.

4. Cut-Matching Game and Matrix Multiplicative Update

The cut-matching game is an iterative framework for constructing expanders through flows and cuts, originally developed to design fast algorithms for approximating edge expansion [KRV09]. It has since found unexpected applications in designing both fast and approximation algorithms for other graph problems. We study the original proof in [KRV09] and a more systematic proof using the matrix multiplicative weight update method [AK16]. We further study an almost linear-time $O(\sqrt{\log n})$-approximation algorithm for edge expansion [She09], building on the expander flow framework in the seminal work [ARV09].

5. Semidefinite Programming and Approximation Algorithms

Expander graphs satisfy a local-to-global property that is useful for designing approximation algorithms [AKK+08]. We study the local-to-global correlation rounding method for semidefinite programming developed in [BRS11, GS11], which extends this approach to low-threshold-rank graphs[5]. It remains an important open question whether these methods can be extended to obtain subexponential-time approximation algorithms for general graphs beyond the best-known approximation guarantees for polynomial-time algorithms. We discuss an approach to this open question for the edge expansion problem based on conjectures about reweighting graphs to have small mixing time.

6. Matrix Concentration and Semi-Random Problems

The spectrum of random matrices is a well-studied topic with deep results [Tro15, BBvH23], which can be leveraged to design spectral algorithms for various semi-random graph problems, including the task of separating noise from signal. We study fundamental matrix concentration inequalities and see their applications in designing spectral algorithms for CSP refutation [GKM25]. We also study recent developments based on the Kikuchi matrix method [WEAM25], including proving lower bounds for locally decodable codes [AGKM23] and resolving Feige's conjecture on the hypergraph Moore bound [GKM22, HKM23].

7. High-Dimensional Expanders and Mixing Time

High-dimensional expanders generalize expander graphs to higher dimensions [Lub18], with recent breakthrough applications in error-correcting codes and the analysis of random walks. We study how this new concept provides a local-to-global way to bound the spectral gap of the random walk matrix [Opp18, KO20], leading to an elegant solution to the matroid expansion conjecture [ALOV19]. We will also see how this approach is sharpened to develop the spectral independence and entropic independence frameworks [AL20, ALO20, AJK+22], which provide powerful tools for analyzing the mixing time of random walks and have led to major advances in the field.

References

[ABS10] Sanjeev Arora, Boaz Barak, and David Steurer. Subexponential algorithms for unique games and related problems. In *51th Annual IEEE Symposium on Foundations of Computer Science, FOCS*, pages 563–572. IEEE Computer Society, 2010. 8, 89, 98

[5] Graphs with few eigenvalues close to the first eigenvalue.

[AGKM23] Omar Alrabiah, Venkatesan Guruswami, Pravesh K Kothari, and Peter Manohar. A near-cubic lower bound for 3-query locally decodable codes from semirandom csp refutation. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, pages 1438–1448, 2023. 9, 311, 312, 314

[AJK+22] Nima Anari, Vishesh Jain, Frederic Koehler, Huy Tuan Pham, and Thuy-Duong Vuong. Entropic independence: optimal mixing of down-up random walks. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1418–1430, 2022. 9, 360

[AK16] Sanjeev Arora and Satyen Kale. A combinatorial, primal-dual approach to semidefinite programs. *J. ACM*, 63(2):12:1–12:35, 2016. 9, 227, 228, 235, 240, 241, 242

[AKK+08] Sanjeev Arora, Subhash Khot, Alexandra Kolla, David Steurer, Madhur Tulsiani, and Nisheeth K. Vishnoi. Unique games on expanding constraint graphs are easy. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing*, pages 21–28. ACM, 2008. 9

[AL20] Vedat Levi Alev and Lap Chi Lau. Improved analysis of higher order random walks and applications. In *Proccedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, STOC 2020*, pages 1198–1211. ACM, 2020. 9, 344

[ALO15] Zeyuan Allen-Zhu, Zhenyu Liao, and Lorenzo Orecchia. Spectral sparsification and regret minimization beyond matrix multiplicative updates. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015*, pages 237–245. ACM, 2015. 8, 196, 203, 228

[ALO20] Nima Anari, Kuikui Liu, and Shayan Oveis Gharan. Spectral independence in high-dimensional expanders and applications to the hardcore model. In *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020*, pages 1319–1330. IEEE, 2020. 9, 349, 351, 353

[ALOV19] Nima Anari, Kuikui Liu, Shayan Oveis Gharan, and Cynthia Vinzant. Log-concave polynomials II: high-dimensional walks and an FPRAS for counting bases of a matroid. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, STOC 2019*, pages 1–12. ACM, 2019. 9, 329, 334, 344

[ARV09] Sanjeev Arora, Satish Rao, and Umesh V. Vazirani. Expander flows, geometric embeddings and graph partitioning. *J. ACM*, 56(2):5:1–5:37, 2009. 9

[BBvH23] Afonso S. Bandeira, March T. Boedihardjo, and Ramon van Hande. Matrix concentration inequalities and free probability. *Inventiones Mathematicae*, 2023. 9, 294, 295

[BRS11] Boaz Barak, Prasad Raghavendra, and David Steurer. Rounding semidefinite programming hierarchies via global correlation. In *IEEE 52nd Annual Symposium on Foundations of Computer Science, FOCS 2011*, pages 472–481. IEEE Computer Society, 2011. 9, 167

[BSS14] Joshua D. Batson, Daniel A. Spielman, and Nikhil Srivastava. Twice-ramanujan sparsifiers. *SIAM Rev.*, 56(2):315–334, 2014. 8

[GKM22] Venkatesan Guruswami, Pravesh K Kothari, and Peter Manohar. Algorithms and certificates for boolean csp refutation: smoothed is no harder than random. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, pages 678–689, 2022. 9, 315, 318

[GKM25] Venkatesan Guruswami, Pravesh K Kothari, and Peter Manohar. New spectral algorithms for refuting smoothed k-sat. *Communications of the ACM*, 68(3):83–91, 2025. 9, 315

[GS11] Venkatesan Guruswami and Ali Kemal Sinop. Lasserre hierarchy, higher eigenvalues, and approximation schemes for graph partitioning and quadratic integer programming with PSD objectives. In *IEEE 52nd Annual Symposium on Foundations of Computer Science, FOCS 2011*, pages 482–491. IEEE Computer Society, 2011. 9, 167

[HKM23] Jun-Ting Hsieh, Pravesh K Kothari, and Sidhanth Mohanty. A simple and sharper proof of the hypergraph moore bound. In *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2324–2344. SIAM, 2023. 9, 301, 302, 303, 304, 306, 311, 315, 318

[HLW06] Shlomo Hoory, Nathan Linial, and Avi Wigderson. Expander graphs and their applications. *Bull. Amer. Math. Soc.*, 43(04):439–562, 2006. 7, 25, 26, 60, 63, 67, 68, 73, 77, 78, 80

[KLL+13] Tsz Chiu Kwok, Lap Chi Lau, Yin Tat Lee, Shayan Oveis Gharan, and Luca Trevisan. Improved cheeger's inequality: analysis of spectral partitioning algorithms through higher order spectral gap. In *Symposium on Theory of Computing Conference, STOC'13*, pages 11–20. ACM, 2013. 8, 101

[KLT22] Tsz Chiu Kwok, Lap Chi Lau, and Kam Chuen Tung. Cheeger inequalities for vertex expansion and reweighted eigenvalues. In *63rd IEEE Annual Symposium on Foundations of Computer Science, FOCS*, pages 366–377. IEEE, 2022. 8, 111, 119, 120

[KO20] Tali Kaufman and Izhar Oppenheim. High order random walks: Beyond spectral gap. *Comb.*, 40(2):245–281, 2020. 9, 331, 333, 343

[Kol10] Alexandra Kolla. Spectral algorithms for unique games. In *Proceedings of the 25th Annual IEEE Conference on Computational Complexity, CCC 2010*, pages 122–130. IEEE Computer Society, 2010. 8

[KRV09] Rohit Khandekar, Satish Rao, and Umesh V. Vazirani. Graph partitioning using single commodity flows. *J. ACM*, 56(4):19:1–19:15, 2009. 9, 150, 153, 213, 215, 216, 218, 219, 220, 221

[LOT14] James R. Lee, Shayan Oveis Gharan, and Luca Trevisan. Multiway spectral partitioning and higher-order cheeger inequalities. *J. ACM*, 61(6):37:1–37:30, 2014. 8, 89, 90, 167

[LPW06] David A. Levin, Yuval Peres, and Elizabeth L. Wilmer. *Markov chains and mixing times*. American Mathematical Society, 2006. 7, 39, 40, 41, 48, 49, 50

[LRTV12] Anand Louis, Prasad Raghavendra, Prasad Tetali, and Santosh S. Vempala. Many sparse cuts via higher eigenvalues. In *Proceedings of the 44th Symposium on Theory of Computing Conference, STOC*, pages 1131–1140. ACM, 2012. 8, 89, 90, 98

[LTW23] Lap Chi Lau, Kam Chuen Tung, and Robert Wang. Cheeger inequalities for directed graphs and hypergraphs using reweighted eigenvalues. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing, STOC*, pages 1834–1847. ACM, 2023. 8, 123, 131

[Lub18] Alexander Lubotzky. High dimensional expanders. In *Proceedings of the International Congress of Mathematicians (ICM 2018)*, pages 705–730, 2018. 9, 331, 333

[MSS14] Adam W. Marcus, Daniel A. Spielman, and Nikhil Srivastava. Ramanujan graphs and the solution of the Kadison Singer problem. In *Proceedings of the international Congress of Mathematicians*, volume 3, pages 363–386. 2014. 8, 188, 196, 197

[Opp18] Izhar Oppenheim. Local spectral expansion approach to high dimensional expanders part I: descent of spectral gaps. *Discrete & Computational Geometry*, 59(2):293–330, 2018. 9, 329, 333, 334

[She09] Jonah Sherman. Breaking the multicommodity flow barrier for o(vlog n)-approximations to sparsest cut. In *50th Annual IEEE Symposium on Foundations of Computer Science, FOCS*, pages 363–372. IEEE Computer Society, 2009. 9, 235, 238, 240, 241, 242

[SM00] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(8):888–905, 2000. 8, 33

[SS11] Daniel A. Spielman and Nikhil Srivastava. Graph sparsification by effective resistances. *SIAM J. Comput.*, 40(6):1913–1926, 2011. 8

[ST11] Daniel A. Spielman and Shang-Hua Teng. Spectral sparsification of graphs. *SIAM J. Comput.*, 40(4):981–1025, 2011. 8, 155

[ST13] Daniel A. Spielman and Shang-Hua Teng. A local clustering algorithm for massive graphs and its application to nearly linear time graph partitioning. *SIAM J. Comput.*, 42(1):1–26, 2013. 8, 48, 135, 136, 140, 142, 143, 154

[ST14] Daniel A Spielman and Shang-Hua Teng. Nearly linear time algorithms for preconditioning and solving symmetric, diagonally dominant linear systems. *SIAM Journal on Matrix Analysis and Applications*, 35(3):835–885, 2014. 8

[Tro15] Joel A Tropp. An introduction to matrix concentration inequalities. *Foundations and Trends® in Machine Learning*, 8(1-2):1–230, 2015. 9, 283, 287, 290, 295

[WEAM25] Alexander Wein, Ahmed El Alaoui, and Cristopher Moore. The kikuchi hierarchy and tensor pca. *Journal of the ACM*, 72(5):1–40, 2025. 9, 307, 308, 310, 319, 320, 322

Part

Classical Results

We begin with the basic concepts and introduce Cheeger's inequality, along with its applications to random walks, expander graphs, and graph partitioning.

Chapter 1

Graph Spectrum

The linear algebraic approach to algorithmic graph theory views graphs as matrices and use concepts and tools in linear algebra to design and analyze algorithms for graph problems.

Spectral graph theory focuses on using eigenvalues and eigenvectors of matrices associated with the graph to study its combinatorial properties. While it may not be clear why eigenvalues provide useful information about the combinatorial properties of graphs, they do, and a surprising amount of information can be obtained from them.

In this chapter, we consider the adjacency matrix and the Laplacian matrix of an undirected graph, and study some basic results in spectral graph theory such as characterizations of bipartiteness and connectedness. General references for this chapter include [Spi19, Tre17].

1.1 Adjacency Matrix

We start with simple graphs for simplicity. The generalization to weighted graphs is straightforward.

Definition 1.1 (Adjacency Matrix of Simple Graphs)**.** *Given a simple graph $G = (V = [n], E)$, the adjacency matrix $A(G)$ is an $n \times n$ matrix where $A_{ij} = A_{ji} = 1$ if $ij \in E(G)$ and $A_{ij} = A_{ji} = 0$ otherwise.*

The adjacency matrix of an undirected graph is symmetric. Therefore, by the spectral theorem for real symmetric matrices in Theorem A.5, the adjacency matrix has an orthonormal basis of eigenvectors with real eigenvalues. We denote the eigenvalues of the adjacency matrix by

$$\alpha_1 \geq \alpha_2 \geq \cdots \geq \alpha_n.$$

Let us begin with some examples and compute their spectra.

Example 1.2 (Complete Graphs)**.** *If G is a complete graph, then $A(G) = J - I$, where J denotes the all-one matrix. Any vector is an eigenvector of I with eigenvalue 1. Hence, the eigenvalues of A are one less than those of J. Since J is rank 1, there are $n - 1$ eigenvalues of 0. The all-ones vector is an eigenvector of J with eigenvalue n. Thus, $n - 1$ is an eigenvalue of A with multiplicity 1, and -1 is an eigenvalue of A with multiplicity $n - 1$.*

This example exhibits the largest gap between the largest eigenvalue and the second largest eigenvalue.

Example 1.3 (Complete Bipartite Graphs). *Let $K_{p,q}$ be the complete bipartite graph with p vertices on one side and q vertices on the other side. Its adjacency matrix $A(K_{p,q})$ is rank 2, so 0 is an eigenvalue with multiplicity $p+q-2$, and there are two non-zero eigenvalues α and β. By Fact A.35, the sum of the eigenvalues is equal to the trace of A, which is 0 since there are no self-loops. Thus, $\alpha = -\beta$. To determine α, consider the characteristic polynomial $\det(xI - A) = (x-\alpha)(x+\alpha)x^{p+q-2} = x^{p+q} - \alpha^2 x^{p+q-2}$. Using the Leibniz formula for determinants in Fact A.26, any term contributing to x^{p+q-2} must have $p+q-2$ diagonal entries, with the remaining two entries must be $-A_{ij}$ and $-A_{ji}$ for some i, j. There are totally pq such terms (one for each edge), and the sign of the corresponding permutation is -1 because it has only one inversion pair. Therefore, $\alpha^2 = pq$, and thus $|\alpha| = \sqrt{pq}$. To conclude, the spectrum is $(\sqrt{pq}, 0, \ldots, 0, -\sqrt{pq})$, where 0 is an eigenvalue with multiplicity $p+q-2$.*

In Section 1.5, you are asked to compute the spectrum of the cycles and the hypercubes.

Bipartiteness

It turns out that bipartite graphs can be characterized by the spectrum of their adjacency matrix. The following lemma says that the spectrum of a bipartite graph is symmetric around the origin on the real line.

Lemma 1.4 (Spectrum of Bipartite Graph is Symmetric). *If G is a bipartite graph and α is an eigenvalue of $A(G)$ with multiplicity k, then $-\alpha$ is also an eigenvalue of $A(G)$ with multiplicity k.*

Proof. If G is a bipartite graph, we can permute its rows and columns of G to obtain the form

$$A(G) = \begin{pmatrix} 0 & B \\ B^\top & 0 \end{pmatrix}.$$

Suppose $u = \begin{pmatrix} x \\ y \end{pmatrix}$ is an eigenvector of $A(G)$ with eigenvalue α. Then

$$\begin{pmatrix} 0 & B \\ B^\top & 0 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \alpha \begin{pmatrix} x \\ y \end{pmatrix} \qquad \iff \qquad B^\top x = \alpha y \ \text{ and } \ By = \alpha x.$$

Now consider $\begin{pmatrix} x \\ -y \end{pmatrix}$. It satisfies:

$$\begin{pmatrix} 0 & B \\ B^\top & 0 \end{pmatrix} \begin{pmatrix} x \\ -y \end{pmatrix} = \begin{pmatrix} -By \\ B^\top x \end{pmatrix} = \begin{pmatrix} -\alpha x \\ \alpha y \end{pmatrix} = -\alpha \begin{pmatrix} x \\ -y \end{pmatrix}.$$

Thus, $\begin{pmatrix} x \\ -y \end{pmatrix}$ is an eigenvector of $A(G)$ with eigenvalue $-\alpha$. By construction, k linearly independent eigenvectors with eigenvalue α correspond to k linearly independent eigenvectors with eigenvalue $-\alpha$, so their multiplicity is the same. □

The next lemma shows that the converse is also true. The proof is by a trace argument that is commonly applied in bounding eigenvalues of random graphs.

Lemma 1.5 (Symmetric Spectrum Implies Bipartiteness). *Let G be an undirected graph and let $\alpha_1 \geq \ldots \geq \alpha_n$ be the eigenvalues of its adjacency matrix. If $\alpha_i = -\alpha_{n-i+1}$ for each $1 \leq i \leq n$, then G is a bipartite graph.*

Proof. Let k be any positive odd number. Then $\sum_{i=1}^{n} \alpha_i^k = 0$, by the symmetry of the spectrum. Note that $\alpha_1^k \geq \alpha_2^k \geq \ldots \geq \alpha_n^k$ are the eigenvalues of A^k, because if $Av = \alpha v$ then $A^k v = \alpha^k v$. By Fact A.35, it follows that $\mathrm{Tr}(A^k) = \sum_{i=1}^{n} \alpha_i^k = 0$. Observe that $(A^k)_{i,j}$ is the number of length-k walks from i to j in G, which can be proved by a simple induction. Now suppose G has an odd cycle of length k. Then $(A^k)_{i,i} > 0$ for each vertex i in the odd cycle, and this would imply that $\mathrm{Tr}(A^k) = \sum_{i=1}^{n} (A^k)_{i,i} > 0$, since each diagonal entry of A^k is non-negative. Therefore, since $\mathrm{Tr}(A^k) = 0$, G must have no odd cycles and is thus a bipartite graph. □

Combining Lemma 1.4 and Lemma 1.5, a graph is bipartite if and only if the spectrum of its adjacency matrix is symmetric around the origin.

Proposition 1.6 (Spectral Characterization of Bipartite Graphs)**.** *Let G be an undirected graph and let $\alpha_1 \geq \ldots \geq \alpha_n$ be the eigenvalues of its adjacency matrix. Then G is a bipartite graph if and only if $\alpha_i = -\alpha_{n-i+1}$ for each $1 \leq i \leq n$.*

When the graph is connected, the characterization is even simpler. In Section 1.5, you are asked to prove that a connected graph is bipartite if and only if $\alpha_1 = -\alpha_n$.

Largest Eigenvalue

Here we see some upper and lower bounds on the largest eigenvalue of the adjacency matrix.

Lemma 1.7 (Max Degree Upper Bound)**.** *Let $G = (V, E)$ be an undirected graph with maximum degree d, and let $\alpha_1 \geq \ldots \geq \alpha_n$ be the eigenvalues of its adjacency matrix. Then $\alpha_1 \leq d$.*

Proof. Let v be an eigenvector with eigenvalue α_1. Let j be a vertex with $v(j) \geq v(i)$ for all $i \in V(G)$. Then

$$\alpha_1 \cdot v(j) = (Av)(j) = \sum_{i:ij \in E(G)} v(i) \leq \sum_{i:ij \in E(G)} v(j) = \deg(j) \cdot v(j) \leq d \cdot v(j),$$

which implies that $\alpha_1 \leq d$. □

Following the proof more closely, we can characterize the connected graphs for which α_1 equals the maximum degree.

Exercise 1.8 (Tight Max Degree Upper Bound)**.** *Let G be a connected undirected graph with maximum degree d and the largest eigenvalue $\alpha_1 = d$. Then G is a d-regular graph.*

The maximum degree upper bound can be far from tight. In Section 1.5, you are asked to prove that the maximum eigenvalue of a tree of maximum degree d is at most $2\sqrt{d-1}$.

On the other hand, the average degree provides a lower bound on the largest eigenvalue. More generally, the largest eigenvalue is at least the average degree of the densest induced subgraph. One corollary of this is that the largest eigenvalue is at least the size of a maximum clique minus one.

Exercise 1.9 (Average Degree Lower Bound)**.** *Let $G = (V, E)$ be an undirected graph with largest eigenvalue α_1. For a subset $S \subseteq V$ and a vertex $v \in S$, let $\deg_S(v) := \big|\{u \mid uv \in E \text{ and } u \in S\}\big|$ be the degree of v induced in S. Then*

$$\alpha_1 \geq \max_{S:S \subseteq V} \frac{1}{|S|} \sum_{v \in S} \deg_S(v).$$

The largest eigenvalue of the adjacency matrix of a connected graph is always of multiplicity one, as guaranteed by the Perron-Frobenius Theorem A.18. The spectrum of the adjacency matrix satisfies

$$d \geq \alpha_1 > \alpha_2 \geq \ldots \geq \alpha_n \geq -d.$$

We will see a combinatorial characterization of the spectral radius of a matrix when we study the Expander Mixing Lemma later. However, there does not seem to be a simple graph-theoretic characterization of the largest eigenvalue of the adjacency matrix. See Section 1.5 for a question in this direction.

1.2 Laplacian Matrix

The Laplacian matrix plays a more important role in spectral graph theory than the adjacency matrix, as we will see some reasons shortly.

Definition 1.10 (Diagonal Degree Matrix)**.** *Let $G = (V, E)$ be an undirected graph with $V(G) = [n]$. The diagonal degree matrix $D(G)$ of G is the $n \times n$ diagonal matrix with $D_{i,i} = \deg(i)$ for each $1 \leq i \leq n$.*

Definition 1.11 (Laplacian Matrix)**.** *Let G be an undirected graph. The Laplacian matrix $L(G)$ of G is defined as $L(G) := D(G) - A(G)$, where $D(G)$ is the diagonal degree matrix in Definition 1.10 and $A(G)$ is the adjacency matrix in Definition 1.1.*

For d-regular graphs, the diagonal degree matrix $D(G)$ is simply $d \cdot I_n$, and so the spectrums of the adjacency matrix and the Laplacian matrix are essentially the same. That is, let $\alpha_1 \geq \alpha_2 \geq \ldots \geq \alpha_n$ be the eigenvalues of the adjacency matrix, and $\lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$ be the eigenvalues of the Laplacian matrix. For d-regular graphs, it holds that $\lambda_i = d - \alpha_i$ for $1 \leq i \leq n$, and thus the i-th largest eigenvalue of A corresponds to the i-th smallest eigenvalue of L.

Throughout this course, we use the convention that the eigenvalues of A are denoted by $\{\alpha_i\}_{i=1}^n$ and those of L are denoted by $\{\lambda_i\}_{i=1}^n$. The eigenvalues of A are ordered in non-increasing order while those of L are ordered in non-decreasing order. When we refer to the k-th eigenvalue of a graph, we mean either the k-th largest eigenvalue of the adjacency matrix or the k-th smallest eigenvalue of the Laplacian matrix.

For non-regular graphs, relating the eigenvalues of the adjacency matrix and the Laplacian matrix is more challenging. On the one hand, as discussed earlier, it is not clear what is a good characterization of α_1 for non-regular graphs. On the other hand, the smallest eigenvalue λ_1 of the Laplacian matrix is always equal to zero, as we will soon demonstrate.

We define a matrix for the proof, which will also be useful later.

Definition 1.12 (Edge Incidence Matrix)**.** *Let $G = (V, E)$ be an undirected graph with $V(G) = [n]$ and $m = |E|$. For each edge $e = ij \in E$, let b_e be the n-dimensional vector with the i-th position equal to $+1$, the j-th position equal to -1, and all other positions equal to 0. Let $B(G)$ be the $n \times m$ edge incidence matrix whose columns are $\{b_e \mid e \in E\}$.*

For an edge $e \in E$, let L_e be its Laplacian matrix, where $(L_e)_{i,i} = (L_e)_{j,j} = 1$ and $(L_e)_{i,j} = (L_e)_{j,i} = -1$. Note that the Laplacian L_e of an edge e can be written as $b_e b_e^\top$, and the Laplacian of the graph G can be written as

$$L(G) = \sum_{e \in E} L_e = \sum_{e \in E} b_e b_e^\top = B(G) \cdot B(G)^\top.$$

Using this definition, we see that zero is always the smallest eigenvalue of the Laplacian matrix.

Lemma 1.13 (Smallest Eigenvalue of Laplacian Matrix)**.** *The Laplacian matrix $L(G)$ of an undirected graph G is positive semidefinite, and its smallest eigenvalue is zero with the all-ones vector being a corresponding eigenvector.*

Proof. As L can be written as $BB^\top$, as shown in Definition 1.12, it follows that L is a positive semidefinite matrix by Fact A.9. Thus, all eigenvalues of L are non-negative. It is straightforward to verify that $L\vec{1} = 0$, so 0 is the smallest eigenvalue, and $\vec{1}$ is the corresponding eigenvector. □

Having a trivial smallest eigenvalue and a simple corresponding eigenvector is one reason that Laplacian matrix is easier to work with. Another reason is that the Laplacian matrix has a quadratic form which has a nice combinatorial interpretation.

Lemma 1.14 (Quadratic Form for Laplacian Matrix)**.** *Let L be the Laplacian matrix of an undirected graph $G = (V, E)$ with $V(G) = [n]$. For any vector $x \in \mathbb{R}^n$,*

$$x^\top L x = \sum_{ij \in E} \big(x(i) - x(j)\big)^2.$$

Proof. Using the decomposition of L in Definition 1.12,

$$x^\top L x = x^\top \Big(\sum_{ij \in E} L_{ij} \Big) x = x^\top \Big(\sum_{ij \in E} b_{ij} b_{ij}^\top \Big) x = \sum_{ij \in E} x^\top b_{ij} b_{ij}^\top x = \sum_{ij \in E} \big(x(i) - x(j)\big)^2.$$

□

When x is the characteristic vector χ_S for a subset $S \subset V$, note that $x^\top L x = |\delta(S)|$. Lemma 1.13 and Lemma 1.14 will be used to derive a useful formulation for the second smallest eigenvalue of the Laplacian matrix when we study Cheeger's inequality.

Connectedness

It turns out that the second smallest eigenvalue of the Laplacian matrix can be used to determine whether the graph is connected or not.

Proposition 1.15 (Spectral Characterization of Connected Graphs)**.** *Let G be an undirected graph and let $\lambda_1 \leq \ldots \leq \lambda_n$ be the eigenvalues of its Laplacian matrix L. Then G is a connected graph if and only if $\lambda_2 > 0$.*

Proof. Suppose G is disconnected. Then the vertex set can be partitioned into two sets S_1 and S_2 such that there are no edges between them. For a subset $S \subseteq V$, let $\chi_S \in \mathbb{R}^n$ be the characteristic vector of S. It is easy to verify that both χ_{S_1} and χ_{S_2} are eigenvectors of L with eigenvalue 0. Since χ_{S_1} and χ_{S_2} are linearly independent, it follows that 0 is an eigenvalue with multiplicity at least 2, and thus $\lambda_2 = 0$.

Conversely, suppose G is connected. Let x be an eigenvector with eigenvalue 0. Then its quadratic form $x^T L x = 0$, and so $\sum_{ij \in E} (x(i) - x(j))^2 = 0$ by Lemma 1.14, which implies that $x(i) = x(j)$ for every edge $ij \in E$. Since G is connected, it follows that $x = c \cdot \vec{1}$ for some c, and thus the eigenspace corresponding to eigenvalue 0 is one-dimensional. Therefore, the eigenvalue 0 has multiplicity 1, which implies $\lambda_2 > 0$. □

The proof of Proposition 1.15 can be extended to the following generalization.

Exercise 1.16 (Spectral Characterization of Number of Components)**.** *Prove that the Laplacian matrix $L(G)$ of an undirected graph G has 0 as its eigenvalue with multiplicity k if and only if G has k connected components.*

1.3 Normalized Adjacency and Laplacian Matrix

Recall that the spectrum of the adjacency matrix satisfies

$$d \geq \alpha_1 \geq \alpha_2 \geq \ldots \geq \alpha_n \geq -d,$$

where the upper and lower bounds depend on the maximum degree d of the graph. This often introduces a dependency on d when relating these eigenvalues to combinatorial parameters.

To remove this dependency and state the Cheeger's inequality more cleanly, we consider the normalized version of the adjacency matrix and the Laplacian matrix. These normalized matrices were popularized and systematically studied in Chung's book [Chu97] to generalize results for regular graphs to all graphs.

Definition 1.17 (Normalized Adjacency and Laplacian Matrix)**.** *Let G be an undirected graph with no isolated vertices. The normalized adjacency matrix $\mathcal{A}(G)$ of G is defined as*

$$\mathcal{A}(G) := D^{-\frac{1}{2}} A D^{-\frac{1}{2}},$$

where D is the diagonal degree matrix in Definition 1.10 and A is the adjacency matrix in Definition 1.1. The normalized Laplacian matrix $\mathcal{L}(G)$ of G is defined as

$$\mathcal{L}(G) := D^{-\frac{1}{2}} L D^{-\frac{1}{2}},$$

where L is the Laplacian matrix in Definition 1.11. Note that $\mathcal{L}(G) = I - \mathcal{A}(G)$.

We will use the same notation conventions as before. The eigenvalues of $\mathcal{A}(G)$ are denoted by $\alpha_1 \geq \alpha_2 \geq \ldots \geq \alpha_n$, and those of $\mathcal{L}(G)$ are denoted by $\lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$. Since $\mathcal{L}(G) = I - \mathcal{A}(G)$ as stated in Definition 1.17, the spectra of $\mathcal{L}(G)$ and $\mathcal{A}$ are essentially equivalent such that $\lambda_i = 1 - \alpha_i$ for $1 \leq i \leq n$. After normalization, the eigenvalues are bounded as follows.

Lemma 1.18 (Normalized Eigenvalues)**.** *Let G be an undirected graph with no isolated vertices. Let $\alpha_1 \geq \ldots \geq \alpha_n$ be the eigenvalues of its normalized adjacency matrix and $\lambda_1 \leq \ldots \leq \lambda_n$ be the eigenvalues of its normalized Laplacian matrix. Then $1 = \alpha_1 \geq \alpha_n \geq -1$ and $0 = \lambda_1 \leq \lambda_n \leq 2$.*

Proof. First, we show that $\lambda_1 = 0$. Note that 0 is an eigenvalue of $\mathcal{L}$, as

$$\mathcal{L}\big(D^{\frac{1}{2}}\vec{1}\big) = \big(D^{-\frac{1}{2}} L D^{-\frac{1}{2}}\big)\big(D^{\frac{1}{2}}\vec{1}\big) = \big(D^{-\frac{1}{2}} L \vec{1}\big) = 0.$$

Furthermore, note that

$$\mathcal{L} = D^{-\frac{1}{2}} L D^{-\frac{1}{2}} = D^{-\frac{1}{2}} B B^\top D^{-\frac{1}{2}} = \big(D^{-\frac{1}{2}} B\big)\big(D^{-\frac{1}{2}} B\big)^\top$$

where B is the edge incidence matrix defined in Definition 1.12. It follows that $\mathcal{L} = I - \mathcal{A}$ is a positive semidefinite matrix by Fact A.9, and thus 0 is the smallest eigenvalue of $\mathcal{L}$. This implies that $\alpha_1 = 1$ as $\lambda_1 = 1 - \alpha_1$.

Next, we prove that $\alpha_n \geq -1$. We will show that $D+A$ is also a positive semidefinite matrix. Then the same argument as above implies that $I + \mathcal{A} = D^{-\frac{1}{2}}(D+A)D^{-\frac{1}{2}}$ is also positive semidefinite, and thus $1 + \alpha_n \geq 0$. There are at least two ways to see that $D + A$ is positive semidefinite. One way is to define $\bar{B}$ as the "unsigned" matrix of B, where $\bar{B}_{ij} = |B_{ij}|$ for all $i, j \in V$. Using a similar argument as in Definition 1.12, we can verify that $D + A = \bar{B}\bar{B}^\top$. Another way is to use a similar decomposition as in Definition 1.12 and see that the quadratic form of $D + A$ can be written as

$$x^\top (D+A)x = \sum_{ij \in E} \big(x_i + x_j\big)^2,$$

which is a sum of squares and thus non-negative. This implies that $\lambda_n \leq 2$, as $\lambda_n = 1 - \alpha_n$. □

1.4 Generalizations

We discuss two natural directions to generalize these basic results, one direction with many interesting results, while the other direction not much is known.

Quantitative Generalizations for Undirected Graphs

So far, we have used the graph spectrum to deduce simple combinatorial properties of the graph, such as bipartiteness and connectedness. These properties are easy to deduce directly by simple combinatorial methods, such as breadth first search or depth first search. One might wonder why these spectral characterizations are useful.

The key feature of the spectral characterizations is that they can be generalized quantitatively to prove robust generalizations of the basic results. For example:

- λ_2 is close to zero if and only if the graph is close to being disconnected. This is the content of Cheeger's inequality.
- λ_n is close to 2 if and only if the graph has a structure close to a bipartite component. This is an analog of Cheeger's inequality for λ_n.
- λ_k is close to zero if and only if the graph is close to having k connected components. This is a generalization called the higher-order Cheeger's inequality.

These results form the basis of many spectral graph algorithms. We will see precise statements and proofs of these results in later chapters.

Directed Graphs and Hypergraphs

For directed graphs, we can consider its adjacency matrix A and Laplacian matrix $L = D - A$, where D is the diagonal out-degree matrix. We may ask whether the spectrum of these matrices can be related to the combinatorial properties of the directed graph. However, since these matrices are no longer symmetric, the eigenvalues can be complex numbers, and very little is known about the relationship between the spectrum and combinatorial properties of directed graphs.

For hypergraphs, it is not even clear what the natural associated matrices should be. It has been an open direction to develop a spectral theory for directed graphs and hypergraphs.

In this course, we will keep these directions in mind and mention reasonable questions and known results whenever possible. For directed graphs, we will discuss a recent generalization of Cheeger's inequality using reweighted eigenvalues, as well as how to define directed spectral sparsifiers and solve directed Laplacian equations. For hypergraphs, we will introduce the active research area of high-dimensional expanders, which provides a promising framework for developing an interesting spectral theory for hypergraphs.

1.5 Problems

Problem 1.19 (Cycles). *Compute the Laplacian spectrum of C_n, the cycle with n vertices.*

Hint: The eigenvectors of the Laplacian matrix of C_n involve the n-th roots of unity.

Problem 1.20 (Hypercubes). *A hypercube of n-dimension is an undirected graph with 2^n vertices. Each vertex corresponds to a string of n bits. Two vertices have an edge if and only if their corresponding strings differ by exactly one bit.*

1. *Given two undirected graphs $G = (V, E)$ and $H = (U, F)$, we define $G \times H$ as the undirected graph with vertex set $V \times U$, where two vertices (v_1, u_1) and (v_2, u_2) have an edge if and only if either (1) $v_1 = v_2$ and $u_1u_2 \in F$, or (2) $u_1 = u_2$ and $v_1v_2 \in E$. Let x be an eigenvector of the Laplacian of G with eigenvalue α, and let y be an eigenvector of the Laplacian of H with eigenvalue β. Show that we can use x and y to construct an eigenvector of the Laplacian of $G \times H$ with eigenvalue $\alpha + \beta$.*

2. *Use (1), or otherwise, to compute the Laplacian spectrum of the hypercube of n dimension.*

Problem 1.21 (Spectral Characterization of Connected Bipartite Graphs). *Let G be a connected undirected graph, and let $\alpha_1 \geq \ldots \geq \alpha_n$ be the eigenvalues of its adjacency matrix. Prove that G is bipartite if and only if $\alpha_1 = -\alpha_n$.*

You may need to use the Perron-Frobenius result in Theorem A.18 and also the optimization formulation of eigenvalues in Definition A.11 to solve this problem.

Problem 1.22 (Largest Eigenvalue of a Tree). *Prove that the maximum eigenvalue of the adjacency matrix of a tree with maximum degree d is at most $2\sqrt{d-1}$.*

(This bound is important in the study of Ramanujan graphs.)

Question 1.23 (Largest Eigenvalue of Graphs of Bounded Arborcity). *A graph $G = (V, E)$ is of arboricity k if k is the minimum number of edge-disjoint forests required to cover all the edges of the graph. A classic result in combinatorial optimization by Nash-Williams states that*

$$k = \max_{S \subseteq V} \left\lceil \frac{|E(S,S)|}{|S| - 1} \right\rceil,$$

which is closely related to the density of the densest subgraph.

What is the best upper bound on the largest eigenvalue of the adjacency matrix of a graph of arboricity k, expressed in terms of k and the maximum degree d?

Problem 1.24 (Number of Spanning Trees). *Let $G = (V, E)$ be an undirected graph with $V = [n]$.*

1. *Let B be the edge incidence matrix of G as defined in Definition 1.12. Prove that the determinant of any $(n-1)\times(n-1)$ submatrix of B is ± 1 if and only if the $n-1$ edges corresponding to the columns form a spanning tree of G.*

2. *Let L be the Laplacian matrix of G and let L' be the matrix obtained from L by deleting the last row and last column. Use (1), or otherwise, to prove that $\det(L')$ is equal to the number of spanning trees in G.*

   You may use the Cauchy-Binet formula in Fact A.30 to solve this problem.

Problem 1.25 (Wilf's Theorem)**.** *Let G be an undirected graph, and let α_1 be the largest eigenvalue of its adjacency matrix. Prove that $\chi(G) \leq \lfloor \alpha_1 \rfloor + 1$, where $\chi(G)$ is the chromatic number of G.*

You may find the Cauchy interlacing Theorem A.15 useful.

References

[Chu97] F. R. K. Chung. *Spectral Graph Theory*. American Mathematical Society, 1997. 20, 26

[Spi19] Daniel A. Spielman. *Spectral and Algebraic Graph Theory*. 2019. 15, 319, 365

[Tre17] Luca Trevisan. *Lecture Notes on Graph Partitioning, Expanders and Spectral Methods*. 2017. 15, 68, 366

Chapter 2

Cheeger's Inequality

Recall that a graph G is connected if and only if $\lambda_2 > 0$, where λ_2 is the second smallest eigenvalue of the normalized Laplacian matrix. Informally, Cheeger's inequality is a robust generalization that a graph is well-connected if and only if λ_2 is large. This connection between a combinatorial property and an algebraic quantity is important in the theory of random walks and the study of expander graphs, which we will explore in next chapters. Moreover, the proof of Cheeger's inequality provides an efficient algorithm for graph partitioning, which is useful in both theory and practice.

Cheeger's original inequality was proved in the setting of Riemannian manifolds [Che70]. The inequality in the graph setting was established in several works in the 1980s [Dod84, AM85, Alo86, SJ89]. In this chapter, we begin by motivating the formulation of Cheeger's inequality in the graph setting, using edge conductance as a measure of well-connectedness. We then interpret λ_2 as a continuous relaxation of edge conductance to establish the easy direction of Cheeger's inequality. Next, we follow the exposition of Trevisan [Tre08], which explains the hard direction through a rounding algorithm to relate the continuous relaxation to the discrete property. Finally, we present the spectral partitioning algorithm and discuss its strengths and limitations.

2.1 Cheeger's Inequality for Graphs

Cheeger's original inequality relates the isoperimetric constant of a manifold to the second smallest eigenvalue of its Laplace operator.

The discrete analog of the Laplace operator is the Laplacian matrix, as defined in Definition 1.11. For a more detailed explanation of why the Laplacian matrix is the discrete analog of the Laplace operator, please see [HLW06, page 472]. To provide a quick intuition: one application of the Laplace operator is in defining the heat equation $\partial u/\partial t = \Delta u$, where $u(x,t)$ represents the temperature at point x at time t and Δ is the Laplacian operator. The discrete analog of the heat equation is $du/dt = -Lu$, which implies that

$$\frac{du(i)}{dt} = -Lu(i) = -\sum_{j:ji\in E} (u(i) - u(j)).$$

This equation states that the rate of change of $u(i)$ is proportional to the net flow of heat from vertex i to its neighboring vertices.

The isoperimetric constant of a Riemannian manifold quantifies how well-connected the manifold is by measuring the ratio of the volume of the boundary of a subset to the volume of the subset itself.

More formally, the Cheeger constant is defined as

$$h(M) := \inf_S \frac{\mu_{n-1}(\partial S)}{\min\{\mu_n(S), \mu_n(M \setminus S)\}} = \inf_{S:\mu_n(S)\leq \frac{1}{2}\mu_n(M)} \frac{\mu_{n-1}(\partial S)}{\mu_n(S)},$$

where ∂A denotes the boundary of an open subset A, and μ_n and μ_{n-1} denotes the n-dimensional and $(n-1)$-dimensional measures respectively.

A natural way to define an isoperimetric constant of a graph is to measure the volume using the edges of the graph. (Other natural definitions will be discussed in Section 2.5.)

Definition 2.1 (Edge Conductance)**.** *Let $G = (V, E)$ be an undirected graph. The conductance of a subset $S \subseteq V$ and the conductance of the graph G are defined as*

$$\phi(S) := \frac{|\delta(S)|}{\mathrm{vol}(S)} \qquad \text{and} \qquad \phi(G) := \min_{S\subseteq V:\mathrm{vol}(S)\leq |E|} \phi(S),$$

where $\delta(S)$ denotes the set of edges with exactly one endpoint in S, and $\mathrm{vol}(S) := \sum_{v\in S} \deg(v)$ is the volume of the subset S. Note that the constraint $\mathrm{vol}(S) \leq |E|$ is equivalent to $\mathrm{vol}(S) \leq \frac{1}{2}\mathrm{vol}(V)$ as $\mathrm{vol}(V) = 2|E|$. Note also that for all $S \subseteq V$, it holds that $0 \leq \phi(S) \leq 1$, as $\phi(S)$ is the ratio of the number of edges cut by S to the total degree in S.

Cheeger's original inequality [Che70] states that for a compact Riemannian manifold M,

$$\lambda \geq \frac{h(M)^2}{4},$$

where λ is the smallest positive eigenvalue of the Laplace operator.

The corresponding inequality in the graph setting was established in several works during the 1980s [Dod84, AM85, Alo86, SJ89], with motivating applications in constructing expander graphs and analyzing random walks. We present the following version, which uses the second smallest eigenvalue of the normalized Laplacian matrix, as formulated by Chung [Chu97]. This provides the cleanest bound for non-regular graphs, without any dependency on the maximum degree.

Theorem 2.2 (Cheeger's Inequality for Graphs)**.** *Let $G = (V, E)$ be an undirected graph, and let λ_2 be the second smallest eigenvalue of its normalized Laplacian matrix $\mathcal{L}(G)$, as defined in Definition 1.17. Then*

$$\frac{\lambda_2}{2} \leq \phi(G) \leq \sqrt{2\lambda_2}.$$

The first inequality is called the easy direction, and the second inequality is called the hard direction which is the graph analog of Cheeger's original inequality for Riemannian manifolds. We will see that the easy direction corresponds to using the second eigenvalue as a "relaxation" for graph conductance, while the hard direction corresponds to "rounding" a fractional solution for graph conductance to an integral solution.

An important implication of Cheeger's inequality is that λ_2, the second smallest eigenvalue of the normalized Laplacian matrix, can be used to certify that a graph is an expander graph. We say that a graph G is an expander graph if $\phi(G) \geq c$ for some constant $0 < c < 1$. Sparse expander graphs are highly efficient combinatorial objects with numerous applications in theoretical computer science and mathematics [HLW06]. Cheeger's inequality implies that a graph is an expander if and only if λ_2 is a constant bounded away from zero. This provides an algebraic method for constructing expander graphs, bringing deep mathematical tools into their study, and leading to significant advances in the field.

2.2 Easy Direction: Continuous Relaxation

We prove the easy direction of Cheeger's inequality in this section. The key observation is that λ_2 and $\phi(G)$ can be written as optimization problems of the same form.

We start with the optimization formulation of the second eigenvalue of the normalized Laplacian matrix using the Rayleigh quotient in Definition A.11.

Lemma 2.3 (Optimization Formulation for λ_2)**.** *Let $G = (V = [n], E)$ be an undirected graph, and λ_2 be the second smallest eigenvalue of its normalized Laplacian matrix $\mathcal{L}(G)$. Then*

$$\lambda_2 = \min_{x \in \mathbb{R}^n} \frac{\sum_{ij \in E} \big(x(i) - x(j)\big)^2}{\sum_{i \in V} \deg(i) \cdot x(i)^2} \qquad \text{subject to} \qquad \sum_{i \in V} \deg(i) \cdot x(i) = 0.$$

Proof. By the Rayleigh quotient characterization in Lemma A.13,

$$\lambda_2 = \min_{x \in \mathbb{R}^n : x \perp v_1} R_{\mathcal{L}}(x) = \min_{x \in \mathbb{R}^n : x \perp D^{\frac{1}{2}}\vec{1}} \frac{x^\top \mathcal{L} x}{x^\top x} = \min_{x \in \mathbb{R}^n : x \perp D^{\frac{1}{2}}\vec{1}} \frac{x^\top D^{-\frac{1}{2}} L D^{-\frac{1}{2}} x}{x^\top x},$$

where v_1 is the first eigenvector for the normalized Laplacian matrix, which is parallel to $D^{\frac{1}{2}}\vec{1}$ from Lemma 1.18. By a change of variable $x = D^{\frac{1}{2}} y$ for $y \in \mathbb{R}^n$, this can be rewritten as

$$\lambda_2 = \min_{y \in \mathbb{R}^n : D^{\frac{1}{2}} y \perp D^{\frac{1}{2}}\vec{1}} \frac{y^\top L y}{y^\top D y} = \min_{y \in \mathbb{R}^n : \sum_{i \in V} \deg(i) \cdot y(i) = 0} \frac{\sum_{ij \in E} \big(y(i) - y(j)\big)^2}{\sum_{i \in V} \deg(i) \cdot y(i)^2},$$

where the last equality follows from the quadratic form of the Laplacian matrix in Lemma 1.14. □

Next, we observe that the graph conductance can also be written in the same form.

Lemma 2.4 (Optimization Formulation for Graph Conductance)**.** *Let $G = (V = [n], E)$ be an undirected graph. Then*

$$\phi(G) = \min_{x \in \{0,1\}^n} \frac{\sum_{ij \in E} \big(x(i) - x(j)\big)^2}{\sum_{i \in V} \deg(i) \cdot x(i)^2} \qquad \text{subject to} \qquad \sum_{i \in V} \deg(i) \cdot x(i)^2 \le |E|.$$

Proof. For a set $S \subseteq V$, let $\chi_S \in \{0,1\}^n$ be the characteristic vector of S. Note that

$$\phi(S) = \frac{|\delta(S)|}{\mathrm{vol}(S)} = \frac{\sum_{ij \in \delta(S)} 1}{\sum_{i \in S} \deg(i)} = \frac{\sum_{ij \in E} |\chi_S(i) - \chi_S(j)|}{\sum_{i \in V} \deg(i) \cdot \chi_S(i)} = \frac{\sum_{ij \in E} (\chi_S(i) - \chi_S(j))^2}{\sum_{i \in V} \deg(i) \cdot \chi_S(i)^2} = \frac{\chi_S^\top L \chi_S}{\chi_S^\top D \chi_S}.$$

Each vector x in $\{0,1\}^n$ corresponds to the characteristic vector of the subset $S := \{i \mid x(i) = 1\}$. The graph conductance $\phi(G)$ minimizes over subsets with volume at most $|E|$, which corresponds to the constraint that $\sum_{i \in V} \deg(i) \cdot x(i)^2 \le |E|$. □

Intuition: Continuous Relaxation

There are two differences between these two formulations in Lemma 2.3 and Lemma 2.4: one involves the domain, and the other involves the constraint.

The major difference is that the former optimizes over the continuous domain $x \in \mathbb{R}^n$, while the latter optimizes over the discrete domain $x \in \{0,1\}^n$. A good way to think of the relationship between the two optimization problems is that the former problem is a relaxation of the latter. This is a common idea in the design of approximation algorithms. The latter problem is actually an NP-hard combinatorial optimization problem. The relaxation idea is to optimize over a larger continuous domain, so that the problem becomes solvable in polynomial time. Since we optimize over a larger domain, the objective value of the former problem can only be smaller than that of the latter, and so we expect that $\lambda_2 \leq \phi(G)$. This is the main intuition behind the easy direction.

In this course, we will see that this is a common theme in spectral graph theory, where the spectral quantities involve the continuous domain $x \in \mathbb{R}^n$ while the combinatorial properties involve discrete domains such as $x \in \{0,1\}^n$ or $x \in \{-1,1\}^n$.

For these two formulations, however, the constraints are also different. And it turns out that the inequality $\lambda_2 \leq \phi(G)$ does not hold, but the slightly weaker inequality $\lambda_2 \leq 2\phi(G)$ does.

Proof of the Easy Direction

To upper bound λ_2, we just need to find a vector x satisfying the constraint $\sum_{i \in V} \deg(i) \cdot x(i) = 0$, and compute its Rayleigh quotient $R_{\mathcal{L}}(x)$. Let $S \subseteq V$ be an optimal solution to graph conductance, with $\phi(S) = \phi(G)$ and $\mathrm{vol}(S) \leq |E|$. Consider the following binary-valued solution $z \in \mathbb{R}^n$ with

$$z(i) = \begin{cases} \frac{1}{\mathrm{vol}(S)} & \text{if } i \in S \\ -\frac{1}{\mathrm{vol}(V-S)} & \text{if } i \notin S \end{cases}.$$

By construction, $\sum_{i \in V} \deg(i) \cdot z(i) = \sum_{i \in S} \deg(i)/\mathrm{vol}(S) - \sum_{i \in V-S} \deg(i)/\mathrm{vol}(V-S) = 0$. Thus, z is a feasible solution to the optimization problem for λ_2 in Lemma 2.3, and it follows that

$$\lambda_2 \leq \frac{\sum_{ij \in E} \big(z(i) - z(j)\big)^2}{\sum_{i \in V} \deg(i) \cdot z(i)^2} = \frac{|\delta(S)| \cdot \big(\frac{1}{\mathrm{vol}(S)} + \frac{1}{\mathrm{vol}(V-S)}\big)^2}{\frac{1}{\mathrm{vol}(S)} + \frac{1}{\mathrm{vol}(V-S)}} = \frac{|\delta(S)| \cdot 2|E|}{\mathrm{vol}(S) \cdot \mathrm{vol}(V-S)} \leq 2\phi(S),$$

where the last inequality uses the assumption that $\mathrm{vol}(S) \leq |E|$ which implies that $|E| \leq \mathrm{vol}(V-S)$. This completes the proof of the easy direction.

2.3 Hard Direction: Rounding Algorithm

By optimizing over a larger domain, the objective value of the continuous problem will typically be smaller than that of the discrete problem. The hard direction is to prove that λ_2 cannot be much smaller than $\phi(G)$, ensuring that λ_2 is a good approximation to $\phi(G)$.

For graph conductance, the objective is to find an integral solution $z \in \{0,1\}^n$ that minimizes the ratio in Lemma 2.4. However, once the problem is relaxed to the continuous domain, the optimal solution $x \in \mathbb{R}^n$ to λ_2 may be very smooth and continuous. The task in the hard direction is to prove that there always exists an integral solution z whose objective value is not much worse than that of x. A common approach in approximation algorithms is to design a procedure to "round" the

continuous solution x to an integral solution z, while bounding the objective value of z in terms of the objective value of x. This is the approach taken by Trevisan [Tre08], providing a more intuitive proof of Cheeger's inequality.

Ideas and Overview

We can think of the optimizer $x \in \mathbb{R}^n$ to the optimization problem in Lemma 2.3 as an embedding of the vertices of the graph into the real line, such that the total squared edge length $\sum_{ij\in E}(x(i)-x(j))^2$ is small. To produce an integral solution z, a natural idea is to do a "threshold rounding", where we pick a threshold t and set $z(i) = 0$ if $x(i) < t$ and $z(i) = 1$ if $x(i) \geq t$. The intuition is that if most edges are short in this embedding, then there must exist a threshold with not many edges crossing. A simple analogy is that if the average number of nonzeros in the rows of a matrix is small, then there must exist a column with few nonzeros. This intuition can be made precise by introducing the intermediate optimization problem of the following ℓ_1-form:

$$\min_{y\in\mathbb{R}^n_+} \frac{\sum_{ij\in E} |y(i) - y(j)|}{\sum_{i\in V} \deg(i) \cdot y(i)}.$$

The proof of the hard direction consists of the following three steps:

1. Truncate an optimal solution to λ_2 in Lemma 2.3 to obtain a solution $x \in \mathbb{R}^n_+$ that satisfies

$$\frac{\sum_{ij\in E} \big(x(i) - x(j)\big)^2}{\sum_{i\in V} \deg(i) \cdot x(i)^2} \leq \lambda_2 \qquad \text{and} \qquad \mathrm{vol}(\mathrm{supp}(x)) \leq |E|, \tag{2.1}$$

   where $\mathrm{supp}(x) := \{i \in V \mid x(i) \neq 0\}$ is the support set of the vector x. This step is to ensure that the output of the rounding algorithm has volume at most $|E|$. The assumption that the optimal solution to λ_2 satisfies the constraint $\sum_{i\in V} \deg(i) \cdot x(i) = 0$ is only used here. This step can be thought of as bridging the gap between the constraints in Lemma 2.3 and Lemma 2.4.

2. Use the solution x to construct a solution $y \in \mathbb{R}^n_+$ that satisfies

$$\frac{\sum_{ij\in E} |y(i) - y(j)|}{\sum_{i\in V} \deg(i) \cdot y(i)} \leq \sqrt{2\lambda_2} \qquad \text{and} \qquad \mathrm{vol}(\mathrm{supp}(y)) \leq |E|. \tag{2.2}$$

   This step can be interpreted as embedding from ℓ_2^2 to ℓ_1, and it incurs the square root loss in Cheeger's inequality.

3. Apply the threshold rounding procedure described above to y to obtain a set S satisfying

$$\phi(S) \leq \sqrt{2\lambda_2} \qquad \text{and} \qquad \mathrm{vol}(S) \leq |E|.$$

   This step is lossless and relies on a simple probabilistic analysis.

With this overview in mind, we proceed to present the details in reverse order, as the main ideas are in the last two steps.

Threshold Rounding

In the threshold rounding step, we take a vector $y \in \mathbb{R}^n_+$ from (2.2) and outputs a set $S \subseteq \operatorname{supp}(y)$ with the same objective value. Our analysis follows that of Trevisan [Tre08], whose idea is to choose a random $t > 0$ and consider the level set $S_t := \{i \in V \mid y(i) \geq t\}$. The conductance of S_t is then bounded by separately computing the expectation of the numerator $\mathbb{E}\left[|\delta(S_t)|\right]$ and the expectation of the denominator $\mathbb{E}\left[d|S_t|\right]$. The idea of choosing a random t is similar to randomized rounding in approximation algorithms, and Trevisan's approach of computing the expectations separately simplifies the analysis of the ratio.

Lemma 2.5 (Threshold Rounding)**.** *Let $G = (V = [n], E)$ be an undirected graph. Let $y \in \mathbb{R}^n_+$ be a non-zero vector with non-negative entries. There exists $t > 0$ such that the threshold set $S_t := \{i \in [n] \mid y(i) \geq t\}$ is nonempty and satisfies*

$$\phi(S_t) \leq \frac{\sum_{ij \in E} |y(i) - y(j)|}{\sum_{i \in V} \deg(i) \cdot y(i)}.$$

Proof. We scale y so that $\max_i y(i) = 1$. Let $t \in (0, 1]$ be chosen uniformly at random. Note that the threshold set $S_t := \{i \in V \mid y(i) \geq t\}$ is nonempty by construction. In the following, we compute separately the expected value of the numerator and of the denominator for S_t.

For an edge $ij \in E$, note that the probability that $ij \in \delta(S_t)$ is $|y(i) - y(j)|$, when the random threshold t falls between $y(i)$ and $y(j)$. By linearity of expectation,

$$\mathbb{E}_t\big[|\delta(S_t)|\big] = \sum_{ij \in E} \Pr_t(ij \in \delta(S_t)) = \sum_{ij \in E} \big|y(i) - y(j)\big|.$$

For a vertex $i \in V$, note that the probability that $i \in S_t$ is $y(i)$, when the random threshold t is at most $y(i)$. By linearity of expectation,

$$\mathbb{E}_t\big[\operatorname{vol}(S_t)\big] = \sum_{i \in V} \deg(i) \cdot \Pr_t(i \in S_t) = \sum_{i \in V} \deg(i) \cdot y(i).$$

It follows from Lemma 2.6 below that

$$\min_t \phi(S_t) = \min_t \frac{|\delta(S_t)|}{\operatorname{vol}(S_t)} \leq \frac{\mathbb{E}_t\big[|\delta(S_t)|\big]}{\mathbb{E}_t\big[\operatorname{vol}(S_t)\big]} \leq \frac{\sum_{ij \in E} |y(i) - y(j)|}{\sum_{i \in V} \deg(i) \cdot y(i)}.$$

□

Lemma 2.6 (Spielman's Favorite Inequality)**.** *Let $a_1, \ldots, a_n$ and $b_1, \ldots, b_n$ be positive numbers, and $p_1, \ldots, p_n$ be a probability distribution. Then*

$$\min_i \frac{a_i}{b_i} \leq \frac{\sum_{i=1}^n p_i a_i}{\sum_{i=1}^n p_i b_i} \leq \max_i \frac{a_i}{b_i}.$$

The proof of this inequality is left as an exercise to the reader.

Embedding from ℓ_2^2 to ℓ_1

In the embedding step, we construct an ℓ_1-solution y in (2.2) from an ℓ_2^2 solution x in (2.1). The most obvious mapping is $y(i) := x(i)^2$ so as to match the denominators, and it works.

Lemma 2.7 (Embedding Step). *Given an undirected graph $G = (V, E)$ and a vector $x \in \mathbb{R}^n_+$, there is a vector $y \in \mathbb{R}^n_+$ with $\operatorname{supp}(y) = \operatorname{supp}(x)$ such that*

$$\frac{\sum_{ij\in E} |y(i) - y(j)|}{\sum_{i\in V} \deg(i) \cdot y(i)} \le \sqrt{2 \cdot \frac{\sum_{ij\in E} \big(x(i) - x(j)\big)^2}{\sum_{i\in V} \deg(i) \cdot x(i)^2}}.$$

Proof. Set $y(i) = x(i)^2$ for all $i \in V$. By construction, the supports of x and y are the same. The main idea is to use the Cauchy-Schwarz inequality to bound the LHS by the RHS such that

$$\sum_{ij\in E} |y(i) - y(j)| = \sum_{ij\in E} |x(i) - x(j)| \cdot |x(i) + x(j)| \le \sqrt{\sum_{ij\in E} (x(i) - x(j))^2} \sqrt{\sum_{ij\in E} (x(i) + x(j))^2}.$$

Observe that

$$\sum_{ij\in E} (x(i) + x(j))^2 \le \sum_{ij\in E} 2\big(x(i)^2 + x(j)^2\big) = 2 \sum_{i\in V} \deg(i) \cdot x(i)^2.$$

Combining these inequalities, we conclude that

$$\frac{\sum_{ij\in E} |y(i) - y(j)|}{\sum_{i\in V} \deg(i) \cdot y(i)} \le \frac{\sqrt{\sum_{ij\in E} (x(i) - x(j))^2} \sqrt{2 \sum_{i\in V} \deg(i) \cdot x(i)^2}}{\sum_{i\in V} \deg(i) \cdot x(i)^2} = \sqrt{\frac{2 \sum_{ij\in E} \big(x(i) - x(j)\big)^2}{\sum_{i\in V} \deg(i) \cdot x(i)^2}}.$$

□

Truncation Step

Given an optimal solution x to λ_2, we first shift x to obtain $\bar{x}$ with at most the same objective value and the additional property that both the positive part of $\bar{x}$ and the negative part of $\bar{x}$ have volume at most $|E|$. The proof crucially relies on the assumption that $\sum_{i\in V} \deg(i) \cdot x(i) = 0$.

Lemma 2.8 (Shifting). *Let $x \in \mathbb{R}^n$ be an optimal solution to λ_2 in Lemma 2.3. There exists a vector $\bar{x} \in \mathbb{R}^n$ such that $\operatorname{vol}(\{i \mid \bar{x}(i) < 0\}) \le |E|$ and $\operatorname{vol}(\{i \mid \bar{x}(i) > 0\}) \le |E|$ and*

$$\frac{\sum_{ij\in E} \big(\bar{x}(i) - \bar{x}(j)\big)^2}{\sum_{i\in V} \deg(i) \cdot \bar{x}(i)^2} \le \frac{\sum_{ij\in E} \big(x(i) - x(j)\big)^2}{\sum_{i\in V} \deg(i) \cdot x(i)^2}.$$

Proof. Let $c \in \mathbb{R}$ be a median value such that $\operatorname{vol}(\{i \mid x(i) < c\}) \le |E|$ and $\operatorname{vol}(\{i \mid x(i) > c\}) \le |E|$. Set $\bar{x} := x - c\vec{1}$. By construction, $\operatorname{vol}(\{i \mid \bar{x}(i) < 0\}) \le |E|$ and $\operatorname{vol}(\{i \mid \bar{x}(i) > 0\}) \le |E|$.

For the ratio, observe that the numerator does not change by shifting, and the denominator cannot decrease because

$$\sum_{i\in V} \deg(i) \cdot \bar{x}(i)^2 = \sum_{i\in V} \deg(i) \cdot (x(i) - c)^2 = \sum_{i\in V} \deg(i) \cdot x(i)^2 + c^2 \sum_{i\in V} \deg(i) \ge \sum_{i\in V} \deg(i) \cdot x(i)^2,$$

where the last equality holds due to the assumption that $\sum_{i\in V} \deg(i) \cdot x(i) = 0$. This completes the proof. (Note that $\bar{x}$ may not satisfy the constraint $\sum_{i\in V} \deg(i) \cdot \bar{x}(i) = 0$, so it may not be a feasible solution to λ_2 in Lemma 2.3.) □

Next, we show that either the positive part of $\bar{x}$ or the negative part of $\bar{x}$ satisfies the requirements in (2.1).

Lemma 2.9 (Truncation). *Let $\bar{x} \in \mathbb{R}^n$ be a vector that satisfies the properties in Lemma 2.8. There exists a vector $\bar{x}_+ \in \mathbb{R}^n_+$ with $\mathrm{vol}(\mathrm{supp}(\bar{x})) \leq |E|$ and*

$$\frac{\sum_{ij \in E} \left(\bar{x}_+(i) - \bar{x}_+(j)\right)^2}{\sum_{i \in V} \deg(i) \cdot \bar{x}_+(i)^2} \leq \frac{\sum_{ij \in E} \left(\bar{x}(i) - \bar{x}(j)\right)^2}{\sum_{i \in V} \deg(i) \cdot \bar{x}(i)^2}$$

Proof. Let $\bar{x}_+ \in \mathbb{R}^n$ be the vector with $\bar{x}_+(i) := \max\{\bar{x}(i), 0\}$ for $1 \leq i \leq n$, and $\bar{x}_- \in \mathbb{R}^n$ be the vector with $\bar{x}_-(i) := \min\{\bar{x}(i), 0\}$ for $1 \leq i \leq n$. We argue that either $\bar{x}_+$ or $-\bar{x}_-$ would satisfy the requirements. By construction, both $\bar{x}_+$ and $-\bar{x}_-$ satisfy that the volume of the support is at most $|E|$. For the ratio, note that

$$\frac{\sum_{ij \in E} (\bar{x}_+(i) - \bar{x}_+(j))^2)^2 \ + \ \sum_{ij \in E} (\bar{x}_-(i) - \bar{x}_-(j))^2}{\sum_{i \in V} \deg(i) \cdot \bar{x}_+(i)^2 \ + \ \sum_{i \in V} \deg(i) \cdot \bar{x}_-(i)^2} \leq \frac{\sum_{ij \in E} \left(\bar{x}(i) - \bar{x}(j)\right)^2}{\sum_{i \in V} \deg(i) \cdot \bar{x}(i)^2},$$

where the denominators are equal, and the numerator on the LHS can only be smaller than that of the RHS by a simple case analysis. The conclusion then follows from Lemma 2.6. □

Proof of the Hard Direction

We summarize the proof of the hard direction. Let $v_2 \in \mathbb{R}^n$ be an eigenvector of $\mathcal{L}(G)$ with eigenvalue λ_2. First, we apply the transformation $u := D^{-\frac{1}{2}} v_2$ to obtain a vector u that satisfies the requirements in Lemma 2.3. Next, we apply the shifting and truncation steps in Lemma 2.8 and Lemma 2.9 on u to obtain a vector x that satisfies the requirements in (2.1). Then, we apply the embedding step in Lemma 2.7 on x to obtain a vector y that satisfies the requirements in (2.2). Finally, we apply the threshold rounding step in Lemma 2.5 on y to obtain a threshold set S_t with $\phi(S_t) \leq \sqrt{2\lambda_2}$ and $\mathrm{vol}(S_t) \leq |E|$. This completes the proof of the hard direction of Cheeger's inequality.

2.4 The Spectral Partitioning Algorithm

Finding a set of small conductance, also called a sparse cut, is an important algorithmic problem with numerous applications. It is useful in designing divide-and-conquer algorithms and has applications in image segmentation, data clustering, community detection, VLSI design, and more.

The problem of finding a sparsest cut is NP-hard. A popular heuristic for finding an approximate sparsest cut in practice is the following spectral partitioning algorithm.

Algorithm 1 The Spectral Partitioning Algorithm

Require: An undirected graph $G = (V, E)$ with $V = [n]$ and $m = |E|$.

1: Compute the second smallest eigenvalue λ_2 of $\mathcal{L}(G)$ and a corresponding eigenvector $x \in \mathbb{R}^n$.
2: Compute the vector $y := D^{-\frac{1}{2}} x$ and sort the vertices so that $y(1) \geq y(2) \geq \ldots \geq y(n)$.
3: For $1 \leq i \leq n-1$, let $S_i = [i]$ if $\mathrm{vol}_G\left([i]\right) \leq m$, and let $S_i = [n] \setminus [i]$ if $\mathrm{vol}_G\left([i]\right) > m$.
4: **return** $\min_{i:1 \leq i \leq n-1}\{\phi(S_i)\}$.

This algorithm is remarkably simple, with only a few lines of code when implemented in mathematical software such as MATLAB. This simplicity is one reason why this heuristic is popular.

There is a near-linear time randomized algorithm to compute an approximate eigenvector of the second eigenvalue, using the power method and a fast Laplacian solver. This makes the spectral partitioning algorithm both practically and theoretically efficient, which is another reason for its popularity.

The primary reason that it is popular is its excellent empirical performance in various applications, especially in image segmentation and clustering. The introduction of the concept of a normalized cut (closely related to a sparse cut) and the spectral partitioning algorithm by Shi and Malik [SM00] in the context of image segmentation were considered a breakthrough.

The proof of Cheeger's inequality provides a nontrivial performance guarantee of the spectral partitioning algorithm, that it will always output a set S with conductance $\phi(S) \leq \sqrt{2\lambda_2} \leq 2\sqrt{\phi(G)}$. This follows because the shifting, truncation, and embedding steps in the proof of the hard direction are only used for the analysis and do not change the ordering of the vertices. Therefore, the cuts considered in the threshold rounding step are also considered by the spectral partitioning algorithm.

The spectral partitioning algorithm is a constant factor approximation algorithm when $\phi(G)$ is a constant, providing an efficient way to certify that a graph is an expander as discussed earlier. However, its approximation ratio could be arbitrarily bad when $\phi(G)$ is small. For example, the approximation ratio is $\Theta(\sqrt{n})$ when $\phi(G) \leq 1/n$. Providing a theoretical explanation of the empirical success of the spectral partitioning algorithm remains an open problem. We will revisit this question when we study the improved Cheeger's inequality.

In the following, we discuss the theoretical performance of the spectral partitioning algorithm in more detail. We present examples where the easy direction is tight, where the hard direction is tight, and where the spectral partitioning algorithm is fooled.

Tight Example for the Hard Direction

Consider the cycle of $4n$ vertices. One can compute the second eigenvector of the cycle exactly (see Problem 1.19), but we do not need it here. Recall that $\lambda_2 = \min_{x \perp \vec{1}} x^T \mathcal{L} x / x^T x$, so to give an upper bound, we only need to demonstrate a solution with small objective value. Consider

$$x = \Big(1, 1 - \frac{1}{n}, 1 - \frac{2}{n}, \ldots, \frac{1}{n}, 0, -\frac{1}{n}, \ldots, -1 + \frac{1}{n}, -1, -1 + \frac{1}{n}, \ldots, -\frac{1}{n}, 0, \frac{1}{n}, \ldots, 1 - \frac{1}{n}\Big).$$

Then $x \perp \vec{1}$, and so

$$\lambda_2 \leq \frac{\sum_{ij \in E} \big(x(i) - x(j)\big)^2}{2 \sum_{i \in V} x(i)^2} = \Theta\Big(\frac{n\big(\frac{1}{n}\big)^2}{n}\Big) = \Theta\Big(\frac{1}{n^2}\Big).$$

On the other hand, the conductance of the cycle of $4n$ vertices is $\Theta(\frac{1}{n})$, by taking a balanced cut with 2 edges and $2n$ vertices. This is an example where the hard direction $\phi(G) \leq \sqrt{2\lambda_2}$ is tight up to a constant factor.

In this example, λ_2 is not a good estimate of $\phi(G)$, but the spectral partitioning algorithm works perfectly to output a set S with $\phi(S) \approx \phi(G)$, as it outputs a set S with $\phi(S) \approx \sqrt{\lambda_2} \approx \phi(G)$. It is actually a general phenomenon that rounding algorithms work perfectly for the worst integrality gap examples.

Tight Example for the Easy Direction

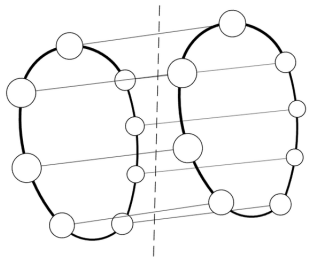
To find an example where the spectral partitioning algorithm performs poorly, we need to examine cases where the easy direction is tight, but the algorithm outputs a set S with $\phi(S) \approx \sqrt{\lambda_2} \approx \sqrt{\phi(G)}$.

For the easy direction, one can check that it is tight for the hypercubes; see Problem 1.20. However, there are vectors in the second eigenspace where the spectral partitioning algorithm performs poorly; see Problem 2.13. Since we do not have control over which eigenvector in the second eigenspace is returned, this provides an example where the spectral partitioning algorithm could perform poorly.

An Example Fooling the Spectral Partitioning Algorithm

The cycles and the hypercubes are the standard examples showing that both sides of Cheeger's inequality are tight. In the hypercube example, the spectral partitioning algorithm could only output a set S with $\phi(S) \approx \sqrt{\phi(G)}$. However, this example may not be fully satisfying, as the algorithm could still work perfectly. More importantly, we do not clearly see or gain intuition about how the spectral partitioning algorithm is fooled.

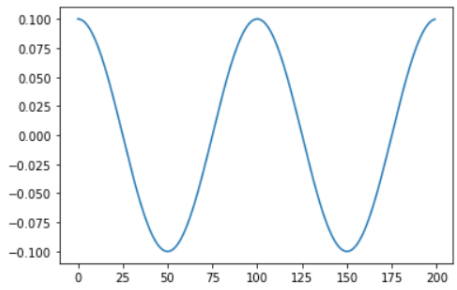
We construct such an example by tweaking the cycle example. Let G be the weighted graph with vertices $\{v_1, \ldots, v_n, v_{n+1}, \ldots, v_{2n}\}$, and two cycles $(v_1, v_2, \ldots, v_n)$ and $(v_{n+1}, v_{n+2}, \ldots, v_{2n})$ where every edge in these cycles is of weight one, and a "hidden" matching $\{v_1v_{n+1}, v_2v_{n+2}, \ldots, v_nv_{2n}\}$ where every edge in the matching has weight $100/n^2$. It is easy to see that the set of smallest



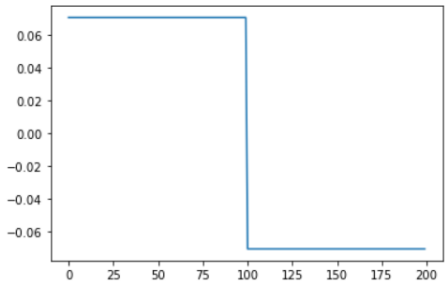
conductance is the set $S := \{v_1, \ldots, v_n\}$ with $\phi(S) = O(1/n^2)$. However, the edges in the hidden matching are just barely heavy enough that the spectral partitioning algorithm does not "feel" them, and still considers the smooth embedding of the cycle as the best way to map the vertices onto the real line. Indeed, one can verify that the second eigenvector x in this example is still the same as that in the cycle of n vertices, with $x(v_i) = x(v_{n+i})$ for $1 \leq i \leq n$. See Figure 2.1 for an illustration. Therefore, λ_2 is still $O(1/n^2)$ which is close to $\phi(G)$, but the cut of smallest conductance is completely lost in x and every threshold set has conductance $\Omega(1/n)$. This example provides a more insightful view into how the spectral partitioning algorithm is fooled. Althout this example is a weighted graph, one can modify it slightly to keep the same structure while making the graph unweighted.

2.5 Variants of Cheeger's Inequality

We discuss two variants of Cheeger's inequality, based on different definitions of the isoperimetric constant of a graph.



Matching edge weight 1/500



Matching edge weight 1/1000

Figure 2.1: The plots of the second eigenvector are generated for the graph with 200 vertices. Vertices 0 to 99 belong to the first cycle, and vertices 100 to 199 belong to the second cycle. A matching edge connects vertex i and vertex $i + 100$ for $0 \leq i \leq 99$. When the edge weight in the hidden matching is slightly heavier, the second eigenvector is the same as that of the cycle, with matched vertices having the same value. When the edge weight in the hidden matching is slightly lighter, the second eigenvalue becomes a binary vector, indicating the sparsest cut.

Edge Expansion

Cheeger's inequality is often stated using edge expansion rather than edge conductance.

Definition 2.10 (Edge Expansion)**.** *Let $G = (V, E)$ be an undirected graph. The edge expansion of a subset $S \subseteq V$ and the edge expansion of the graph G are defined as*

$$\Phi(S) := \frac{|\delta(S)|}{|S|} \qquad \text{and} \qquad \Phi(G) := \min_{S:|S|\leq|V|/2} \Phi(S).$$

Both edge expansion and edge conductance aim to identify the "bottleneck" in the graph. For d-regular graphs, the two definitions are essentially equivalent, with $\Phi(G) = d \cdot \phi(G)$. For non-regular graphs, the relationship between the edge conductance and the second smallest eigenvalue of the normalized Laplacian matrix is more elegant, without any dependency on the maximum degree of the graph.

Vertex Expansion

Another natural definition of an isopermetric constant of graphs is based on measuring the "volume" using the vertices of the graph.

Definition 2.11 (Vertex Expansion)**.** *Let $G = (V, E)$ be an undirected graph. The vertex expansion of a subset $S \subseteq V$ and the vertex expansion of the graph G are defined as*

$$\psi(S) := \frac{|\partial(S)|}{\min\{|S|, |V - S|\}} \qquad \text{and} \qquad \psi(G) := \min_{S \subseteq V} \psi(S),$$

where $\partial(S) := \{v \notin S \mid \exists u \in S \text{ with } uv \in E\}$ denotes the vertex boundary of S.

There is a Cheeger-type inequality relating vertex expansion and the second smallest eigenvalue of the Laplacian matrix.

Theorem 2.12 (Cheeger's Inequality for Vertex Expansion). *Let $G = (V, E)$ be an undirected graph with maximum degree d, and let λ_2' be the second smallest eigenvalue of its (unnormalized) Laplacian matrix $L(G)$, as defined in Definition 1.11. Then*

$$\psi(G) \geq \frac{2\lambda_2'(G)}{d + 2\lambda_2'(G)} \quad \text{and} \quad \lambda_2'(G) \geq \frac{\psi(G)^2}{4 + 2\psi(G)^2},$$

The first inequality is the easy direction proved by Tanner [Tan84] and Alon and Milman [AM85]. The second inequality is the hard direction proved by Alon [Alo86]. Note that these imply that $\lambda_2'(G)$ can be used to provide an $O(\sqrt{d \cdot \psi(G)})$-approximation algorithm to $\psi(G)$.

Comparing to Cheeger's inequality for edge conductance, there is an extra factor d loss between the upper and lower bounds. In Chapter 9, we will introduce a new Cheeger's inequality for vertex expansion using a concept called reweighted eigenvalues, which improves the dependency on the maximum degree from d to $\log d$.

2.6 Problems

Problem 2.13 (Spectral Partitioning for Hypercubes). *Let G be the hypercube of dimension d with 2^d vertices and $\mathcal{L}(G)$ be its normalized Laplacian matrix.*

(a) Show that there is an eigenvector $x \in \mathbb{R}^{2^d}$ of $\mathcal{L}(G)$ with eigenvalue λ_2 so that the spectral partitioning algorithm applied on x outputs a set S with $\phi(S) = R_{\mathcal{L}}(x) = \frac{1}{2}\lambda_2$.

(b) Show that there is an eigenvector $y \in \mathbb{R}^{2^d}$ of $\mathcal{L}(G)$ with eigenvalue λ_2 so that the spectral partitioning algorithm applied on y outputs a set S with $\phi(S) \approx \sqrt{R_{\mathcal{L}}(y)} = \sqrt{\lambda_2}$.

(Hint: Consider a convex combination of the good vectors in the previous part.)

Problem 2.14 (Houdré-Tetali Isoperimetric Constant). *Consider an isoperimetric constant of graphs introduced by Houdré and Tetali [HT04]. Assume the graph $G = (V = [n], E)$ is d-regular for simplicity. For a vertex i and a subset $S \subset V$, denote $d(i, \overline{S}) := |\{ij \in E \mid j \in V - S\}|$. For any $p \in [0, 1]$, the isoperimetric constant φ_p of a subset $S \subset V$ and of the graph are defined as*

$$\varphi_p(S) := \frac{1}{|S|} \sum_{i \in S} \left(\frac{d(i, \overline{S})}{d}\right)^p \quad \text{and} \quad \varphi_p(G) := \min_{S : |S| \leq n/2} \varphi_p(S).$$

(a) Verify that $\varphi_1(G)$ is equal to the edge conductance $\phi(G)$.

(b) Check that $\varphi_0(G)$ is equal to the inner vertex expansion $\psi_{in}(G)$ if we use the convention that $0^0 = 0$. Let $\partial_{in}(S) := |\{i \in S \mid d(i, \overline{S}) > 0\}|$ be the inner vertex boundary. Define $\psi_{in}(S) := |\partial_{in}(S)|/|S|$ and $\psi_{in}(G) := \min_{S : |S| \leq n/2} \psi_{in}(S)$.

(c) Show that $\varphi_{\frac{1}{2}}(G)^2 \leq \varphi_1(G) \cdot \varphi_0(G)$.

(d) Prove that $\varphi_{\frac{1}{2}}(G)^2 \lesssim \lambda_2 \cdot \log d$, where λ_2 is the second eigenvalue of the normalized Laplacian.

(e) Prove that $\varphi_p(G)^2 \lesssim \frac{1}{2p-1} \cdot \lambda_2$ for any $p \in (\frac{1}{2}, 1]$.

(Hint: A similar randomized rounding proof as in the hard direction of Cheeger's inequality would work to prove (d) and (e). See [LT24].)

Cheeger-Type Inequality for λ_n and Bipartiteness Ratio

Through his exposition of Cheeger's inequality using the intermediate ℓ_1-problem presented in this chapter, Trevisan [Tre09] discovered an analog of Cheeger's inequality for λ_n.

In this subsection, we follow his thought process to derive the result, which starts with a spectral characterization relating λ_n to the bipartiteness of the graph.

Exercise 2.15 (Spectral Characterization of Bipartiteness). *Let $G = (V, E)$ be an undirected graph and λ_n be the largest eigenvalue of its normalized Laplacian matrix $\mathcal{L}(G)$. Then $\lambda_n = 2$ if and only if G has a bipartite component, i.e., a connected component that is a bipartite graph.*

Trevisan [Tre09] proved a robust generalization that λ_n is close to 2 if and only if G is close to having a bipartite component, in the same style as in Cheeger's inequality in Theorem 2.2. To state his result, we write the optimization formulation for $2 - \lambda_n$ and then motivate the corresponding combinatorial property.

Exercise 2.16 (Optimization Formulation for $2 - \lambda_n$). *Let $G = (V, E)$ be an undirected graph and λ_n be the largest eigenvalue of $\mathcal{L}(G)$. Then*

$$2 - \lambda_n = \min_{x \in \mathbb{R}^n} \frac{\sum_{ij \in E} \big(x(i) + x(j)\big)^2}{\sum_{i \in V} \deg(i) \cdot x(i)^2}.$$

Trevisan *defined* the combinatorial property to measure the bipartiteness ratio of a subset of vertices using the ℓ_1-version of the optimization problem in Exercise 2.16.

Definition 2.17 (Bipartiteness Ratio). *Let $G = (V = [n], E)$ be an undirected graph. The bipartiteness ratio of a vector $x \in \{-1, 0, 1\}^n$ is defined as*

$$\beta(x) := \frac{\sum_{ij \in E} \big|x(i) + x(j)\big|}{\sum_{i \in V} \deg(i) \cdot |x(i)|}.$$

The bipartiteness ratio of a graph G is defined as

$$\beta(G) := \min_{x \in \{-1,0,1\}^n} \beta(x).$$

Given a subset S and a bipartition of S into (L, R), the corresponding vector $x \in \{-1, 0, 1\}$ is such that

$$x(i) = \begin{cases} +1 & \text{if } i \in L \\ -1 & \text{if } i \in R \\ 0 & \text{otherwise} \end{cases}.$$

Trevisan proved the following analog of Cheeger's inequality for $2 - \lambda_n$ and $\beta(G)$.

Problem 2.18 (Cheeger's Inequality for λ_n [Tre09]). *Let $G = (V, E)$ be an undirected graph and λ_n be the largest eigenvalue of $\mathcal{L}(G)$. Then*

$$\frac{1}{2}(2 - \lambda_n) \le \beta(G) \le \sqrt{2(2 - \lambda_n)}.$$

An interesting application of this inequality is the design of an approximation algorithm for the maximum cut problem using spectral techniques. An observation is that if λ_n is bounded away from 2, then the graph does not have a very large max-cut.

Problem 2.19. *Use the easy direction of Problem 2.18 to show that the trivial approximation algorithm of cutting* 50% *of edges is a* $1/\lambda_n$*-approximation algorithm for the maximum cut problem.*

On the other hand, if λ_n is close to 2, the hard direction of Problem 2.18 can be used to find a subset $S = (L, R)$ with small bipartiteness ratio. This ensures that more than 50% of the edges with an endpoint in S will be cut. We can then apply the same idea recursively on $V - S$ to obtain a better than 50% approximation algorithm for the maximum cut problem. See [Tre09] for details and [Sot15] for an improved analysis.

References

[Alo86] Noga Alon. Eigenvalues and expanders. *Comb.*, 6(2):83–96, 1986. 25, 26, 36

[AM85] Noga Alon and V. D. Milman. λ_1, isoperimetric inequalities for graphs, and superconcentrators. *J. Comb. Theory, Ser. B*, 38(1):73–88, 1985. 25, 26, 36

[Che70] J. Cheeger. A lower bound for the smallest eigenvalue of the laplacian. *Problems Anal.*, page 195, 1970. 25, 26

[Chu97] F. R. K. Chung. *Spectral Graph Theory*. American Mathematical Society, 1997. 20, 26

[Dod84] J. Dodziuk. Difference equations, isoperimetric inequality and transience of certain random walks. *Trans. Amer. Math. Soc.*, 284:787–794, 1984. 25, 26

[HLW06] Shlomo Hoory, Nathan Linial, and Avi Wigderson. Expander graphs and their applications. *Bull. Amer. Math. Soc.*, 43(04):439–562, 2006. 7, 25, 26, 60, 63, 67, 68, 73, 77, 78, 80

[HT04] Christian Houdré and Prasad Tetali. Isoperimetric invariants for product Markov chains and graph products. *Combinatorica*, 24:359–388, 2004. 36

[LT24] Lap Chi Lau and Dante Tjowasi. On the Houdré-Tetali conjecture about an isoperimetric constant of graphs. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM*, volume 317 of *LIPIcs*, pages 36:1–36:18, 2024. 36

[SJ89] Alistair Sinclair and Mark Jerrum. Approximate counting, uniform generation and rapidly mixing markov chains. *Inf. Comput.*, 82(1):93–133, 1989. 25, 26

[SM00] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(8):888–905, 2000. 8, 33

[Sot15] José A. Soto. Improved analysis of a max-cut algorithm based on spectral partitioning. *SIAM J. Discret. Math.*, 29(1):259–268, 2015. 38

[Tan84] Robert Michael Tanner. Explicit concentrators from generalized n-gons. *SIAM Journal on Algebraic and Discrete Methods*, 5:287–293, 1984. 36

[Tre08] Luca Trevisan. The spectral partitioning algorithm. *Blog post:* https://lucatrevisan.wordpress.com/2008/05/11/the-spectral-partitioning-algorithm/, 2008. 25, 29, 30

[Tre09] Luca Trevisan. Max cut and the smallest eigenvalue. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC*, pages 263–272. ACM, 2009. 37, 38

Chapter 3

Random Walks on Graphs

Given an undirected graph $G = (V, E)$, a random walk is a simple stochastic process where it starts from a vertex, and in each step the walk moves to a uniformly random neighbor of the current vertex. We are interested in understanding the long-term behavior of the random walk. Is there a limiting distribution on the vertices as the number of steps tends to infinity? If so, how many steps are needed to converge to the limiting distribution?

There are two main approaches to addressing these questions. One is probabilistic, based on the concept of "coupling" two random processes. The other is spectral, using the eigenvalues of the transition matrix. We study the spectral approach and refer the reader to [Häg02, LPW06] for expositions on the probabilistic approach.

In this chapter, we begin with the more general setting of a finite Markov chain and state the fundamental theorem. Next, we specialize the fundamental theorem to the case of random walks on undirected graphs and use spectral analysis to prove it. The spectral analysis builds naturally on the results in Chapter 1 and Chapter 2 and provides a useful upper bound on the mixing time. Finally, we discuss some interesting applications of random walks and mention some known results for random walks on directed graphs.

3.1 Markov Chains

A finite Markov chain is defined by a finite state space and a transition matrix.

Definition 3.1 (Transition Matrix)**.** *Let $[n]$ be the state space. A matrix $P \in \mathbb{R}^{n \times n}$ is a probability transition matrix if P is non-negative and $\sum_{j \in [n]} P_{i,j} = 1$ for each $i \in [n]$. For $1 \leq i, j \leq n$, the entry P_{ij} is the transition probability from state i to state j.*

Definition 3.2 (Markov Chain)**.** *A sequence of random variables $(X_0, X_1, \ldots)$ is a Markov chain with state space $[n]$ and transition matrix $P \in \mathbb{R}^{n \times n}$ if, for all $i, j \in [n]$ and $t \geq 1$,*

$$\Pr\left[X_{t+1} = j \mid X_t = i \cap X_{t-1} = i_{t-1} \cap \ldots \cap X_0 = i_0\right] = \Pr\left[X_{t+1} = j \mid X_t = i\right] = P_{i,j}.$$

This property, known as the Markov property, states that the transition probability from i to j depends only on the current states X_t, regardless of the states $X_0, \ldots, X_{t-1}$ that precedes it.

Let $\vec{p}_0 \in \mathbb{R}^n$ be an initial probability distribution over the states. Then, $\vec{p}_t := \vec{p}_0 P^t$ is the probability distribution on the states after t steps of random walks.

A Markov chain can be viewed as a random walk on a weighted directed graph $G = ([n], w)$, where the transition probability from state i to state j is proportional to the edge weight $w(i,j)$ such that $P_{i,j} = w(i,j)/\sum_{j\in[n]} w(i,j)$.

Irreducibility and Aperiodicity

Two key properties ensures the existence of a unique limiting distribution.

Definition 3.3 (Irreducibility)**.** *A Markov chain defined by transition matrix $P \in \mathbb{R}^{n\times n}$ is called irreducible if, for any two states i, j, there exists an integer t such that $\Pr[X_t = j \mid X_0 = i] > 0$.*

Equivalently, the underlying directed graph $G = ([n], E)$, where $E(G) := \{ij \mid P_{i,j} > 0\}$ of P, is strongly connected.

This property is called irreducibility because, if it is not satisfied, then the Markov chain can be reduced to a smaller one for studying the limiting distribution. Specifically, the limiting distribution, if exists, will only have support on a strongly connected component of the underlying directed graph.

Definition 3.4 (Aperiodicity)**.** *The period of a state i is defined as $\gcd\{t \mid \Pr[X_t = i \mid X_0 = i] > 0\}$, the greatest common divisor of the set of times when it is possible to return to the starting state i. A state i is aperiodic if its period is equal to 1. A Markov chain is aperiodic if all states are aperiodic; otherwise it is periodic.*

For example, random walks on an undirected bipartite graph are periodic, as every state has period 2. Similarly, random walks on a directed cycle of length $k > 1$ are periodic, with every state having a period of k. In general, a Markov chain does not have a limiting distribution if it is periodic.

Irreducibility and aperiodicity together imply the following property.

Proposition 3.5 (Reachability)**.** *For any finite, irreducible, and aperiodic Markov chain, there exists an integer $\tau < \infty$ such that $\Pr[X_t = j \mid X_0 = i] > 0$ for all i, j and all $t \geq \tau$.*

The proof of reachability uses aperiodicity and a simple number-theoretic argument to establish the statement for all $i = j$, and then uses irreducibility to extend the statement for all $i \neq j$. We do not need this result for the spectral analysis in this chapter. Interested readers are referred to [Häg02, LPW06] for a detailed proof.

Stationary Distribution and Convergence

Informally, when a limiting distribution exists, it is a stationary distribution, defined as follows.

Definition 3.6 (Stationary Distribution)**.** *For a Markov chain defined by a transition matrix $P \in \mathbb{R}^{n\times n}$, a probability distribution $\vec{\pi} \in \mathbb{R}^n$ is a stationary distribution if $\vec{\pi}P = \vec{\pi}$, where $\vec{\pi}$ is represented as a row vector.*

A stationary distribution $\vec{\pi}$ satisfies $\vec{\pi}P^t = \vec{\pi}$ for any $t \geq 1$. From a linear algebraic perspective, a stationary distribution is simply an eigenvector of $P^\top$ corresponding to the eigenvalue 1.

To define convergence, we need a measure of how close two probability distributions are. One commonly used measure is the total variation distance.

Definition 3.7 (Total Variation Distance)**.** *Given two probability distributions $\vec{p}, \vec{q} \in \mathbb{R}^n$, the total variation distance of $\vec{p}$ and $\vec{q}$ is defined as*

$$d_{\mathrm{TV}}(\vec{p}, \vec{q}) := \frac{1}{2} \sum_{i=1}^{n} |p(i) - q(i)| = \frac{1}{2} \|\vec{p} - \vec{q}\|_1.$$

We say that $\vec{p}_t$ converges to a probability distribution $\vec{q}$ as $t \to \infty$ if $\lim_{t\to\infty} d_{\mathrm{TV}}(\vec{p}_t, \vec{q}) = 0$.

Fundamental Theorem of Markov Chains

The fundamental theorem states that any finite, irreducible and aperiodic Markov chain has a unique limiting distribution. Moreover, this limiting distribution is independent of the initial distribution.

Theorem 3.8 (Fundamental Theorem of Markov Chains)**.** *Consider the Markov chain defined by a transition matrix $P \in \mathbb{R}^{n\times n}$. Let $\vec{p}_0 \in \mathbb{R}^n$ be an initial probability distribution on the states. Let $\vec{p}_t \in \mathbb{R}^n$ be the probability distribution after t steps, i.e., $\vec{p}_t := \vec{p}_0 P^t$ after t steps. If the Markov chain is finite, irreducible, and aperiodic, then the distribution $\vec{p}_t$ converges to a unique stationary distribution $\vec{\pi}$, regardless of the initial distribution $\vec{p}_0$.*

The intuition behind the probabilistic proof of the fundamental theorem is as follows: For any finite, irreducible, and aperiodic Markov chain defined by P, running the chain for a sufficiently long time ensures that it is possible to reach any state from any other state (Proposition 3.5). If two instances of the Markov chains, $(X_1, X_2, \ldots)$ and $(Y_1, Y_2, \ldots)$, meet at the same state at some time t (i.e., $X_t = Y_t$), then their future behavior becomes indistinguishable because Markov chains "forget" their history. By Proposition 3.5, any two instances of the Markov chains will eventually meet, and thus all distributions converge to the same limiting distribution as $t \to \infty$.

In the following sections, we specialize the fundamental theorem to the case of random walks on undirected graphs and use a spectral approach to prove it. The spectral analysis has the advantage that it also provides a useful upper bound on the mixing time.

For the general result, we refer the reader to [Häg02] for a probabilistic proof using coupling, [LPW06] for a probabilistic and algebraic proof, and [HJ13] for a purely algebraic proof related to the Perron-Frobenius Theorem A.18.

3.2 Random Walks on Undirected Graphs

We consider random walks on an unweighted undirected graph $G = (V, E)$, where in each step the walk moves to a uniformly random neighbor of the current vertex. The fundamental theorem becomes easier in this special case, as there are simple characterizations of irreducibility, aperiodicity, and the limiting distribution. We also consider lazy random walks at the end of this section.

Matrix Formulation: The transition probability P_{ij} from a vertex i to a vertex j is simply $1/\deg(i)$, and so the transition matrix is $P = D^{-1}A$ where D is the diagonal degree matrix in Definition 1.10 and A is the adjacency matrix in Definition 1.1. Let $\vec{p}_0 : V \to \mathbb{R}$ be an initial probability distribution, and $\vec{p}_t$ be the probability distribution after t steps of random walks. Then $\vec{p}_{t+1} = \vec{p}_t P^\top = \vec{p}_t D^{-1} A$, and by induction $\vec{p}_t = \vec{p}_0 (D^{-1}A)^t$.

Stationary Distribution: Recall that a probability distribution $\vec{\pi} : V \to \mathbb{R}$ is a stationary distribution of P if $\vec{\pi}P = \vec{\pi}$. It is equivalent to saying that $\vec{\pi}$ is a an eigenvector of $P^\top$ with eigenvalue 1. Given that $P = D^{-1}A$ for random walks on undirected graphs, it is not difficult to identify one such eigenvector with probabilities proportional to the degrees.

Lemma 3.9 (Stationary Distribution of Undirected Graphs)**.** *Let $G = (V, E)$ be an undirected graph and $P = D^{-1}A$ be its transition matrix. The distribution $\vec{\pi} : V \to \mathbb{R}$ with*

$$\vec{\pi}(i) = \frac{\deg(i)}{\sum_{j \in V} \deg(j)} = \frac{\deg(i)}{2|E|}$$

for all $i \in V$ is a stationary distribution of P.

Irreducibility: Is $\vec{\pi}$ in Lemma 3.9 the unique stationary distribution? Not necessarily. For example, if the graph is disconnected, the distribution after many steps depends on the initial distribution (e.g., which component does the starting vertex belongs to). This corresponds to the irreducibility condition in the fundamental theorem. For undirected graphs, the irreducibility condition is equivalent to the graph being connected.

Aperiodicity: Even if the graph is connected, a limiting distribution may not exist. For example, in a connected bipartite graph, if the initial distribution $\vec{p}_0$ is on a single vertex, then the distribution $\vec{p}_t$ depends on the parity of t, as the support of $\vec{p}_t$ oscillates between the two sides of the bipartite graph. This corresponds to the aperiodicity condition in the fundamental theorem. For connected undirected graphs, observe that the aperiodicity condition is equivalent to the condition that the graph is non-bipartite.

Fundamental Theorem: Given the simple characterizations of the conditions in the fundamental theorem, it reduces to the following statement for undirected graphs.

Theorem 3.10 (Fundamental Theorem for Undirected Graphs)**.** *Let G be a connected, non-bipartite undirected graph. Let $P = D^{-1}A$ be the transition matrix of random walks on G. The distribution $\vec{\pi}$ in Lemma 3.9 is the unique stationary distribution. Furthermore, $\vec{p}_t := \vec{p}_0 P^t$ converges to $\vec{\pi}$ as $t \to \infty$ regardless of the initial distribution $\vec{p}_0$.*

Lazy Random Walks

The non-bipartiteness condition ensures that the Markov chain is aperiodic. There is a simple modification of the random walks so that this assumption can be removed by adding self-loops in the graph.

Definition 3.11 (Lazy Random Walks)**.** *Let G be an undirected graph. The transition matrix W of the lazy random walks is defined as $W = \frac{1}{2}I + \frac{1}{2}D^{-1}A$. In other words, the lazy random walks stay at the current vertex with probability $\frac{1}{2}$ and moves to a uniform random neighbor of the current vertex with probability $\frac{1}{2}$.*

By performing lazy random walks, we make the graph non-bipartite and obtain the following corollary of Theorem 3.10.

Corollary 3.12 (Fundamental Theorem for Lazy Undirected Graphs)**.** *Let G be a connected undirected graph. Let $W = \frac{1}{2}I + \frac{1}{2}D^{-1}A$ be the transition matrix of lazy random walks on G. The distribution $\vec{\pi}$ in Lemma 3.9 is the unique stationary distribution. Furthermore, $\vec{p}_t := \vec{p}_0 W^t$ converges to $\vec{\pi}$ as $t \to \infty$ regardless of the initial distribution $\vec{p}_0$.*

It will be clear from the spectral analysis in the next section why the constant 1/2 is used.

3.3 Spectral Analysis of Mixing Time for Undirected Graphs

In this section, we will prove the fundamental theorem for undirected graphs in Theorem 3.10 using spectral analysis. Besides being elegant and insightful, spectral analysis can be used to analyze the mixing time, which is the rate of convergence to the unique stationary distribution.

We first assume that the undirected graph is d-regular. We then explain the modifications needed for non-regular undirected graphs.

Spectrum of the Transition Matrix for Regular Graphs

For a d-regular graph, the transition matrix P for random walks and the transition matrix W for lazy random walks are

$$P = D^{-1}A = \frac{1}{d}A = \mathcal{A} \quad \text{and} \quad W = \frac{1}{2}I + \frac{1}{2}\mathcal{A},$$

where $\mathcal{A}$ is the normalized adjacency matrix in Definition 1.17.

This is the main simplification from the d-regular assumption, as the matrices P and W are still real symmetric. Another simplification is that the stationary distribution $\vec{\pi}$ in Lemma 3.9 is simply the uniform distribution $\vec{1}/n$ for a d-regular graph.

Our goal is to prove that

$$\lim_{t\to\infty} P^t \vec{p}_0 = \frac{\vec{1}}{n} \quad \text{and} \quad \lim_{t\to\infty} W^t \vec{p}_0 = \frac{\vec{1}}{n},$$

regardless of the initial distribution $\vec{p}_0$, as long as the graph is connected and non-bipartite for random walks, and connected for lazy random walks.

For the spectral analysis, we write $\vec{p}_0$ and $\vec{p}_t$ as column vectors. To compute $P^t\vec{p}_0$ and $W^t\vec{p}_0$, a repeated application of the same operator, it is helpful to know the spectrum of the matrices P and W, as discussed in Appendix A.

Let $\alpha_1 \geq \alpha_2 \geq \ldots \geq \alpha_n$ be the eigenvalues of $\mathcal{A}$ and $v_1, \ldots, v_n$ be the corresponding orthonormal eigenvectors. Recall that

1. $\alpha_1 = 1$ and $v_1 = \vec{1}/\sqrt{n}$ from Lemma 1.7,
2. $\alpha_2 < 1$ if and only if G is connected from Proposition 1.15,
3. $\alpha_n > -1$ if and only if G is non-bipartite from Problem 1.21.

For the lazy random walk matrix W, the spectrum is $\frac{1}{2}(1+\alpha_1) \geq \frac{1}{2}(1+\alpha_2) \geq \ldots \geq \frac{1}{2}(1+\alpha_n)$, which implies that the smallest eigenvalue is always at least 0. This is why the non-bipartitness assumption can be removed when we consider lazy random walks.

Limiting Distribution

After translating the combinatorial conditions in the fundamental theorem into spectral conditions, we can restate the fundamental theroem for d-regular undirected graphs in Theorem 3.10 as follows, and the proof becomes transparent.

Proposition 3.13 (Limiting Distribution for Regular Graphs)**.** *Let $G = (V = [n], E)$ be a d-regular undirected graph. Let $P = \mathcal{A}$ be the transition matrix of random walks on G and $1 = \alpha_1 \geq \alpha_2 \geq \ldots \geq \alpha_n \geq -1$ be its eigenvalues. If $\alpha_2 < 1$ and $\alpha_n > -1$, then*

$$\lim_{t\to\infty} P^t \vec{p}_0 = \frac{\vec{1}}{n}.$$

Proof. Let $v_1, v_2, \ldots, v_n$ be the orthonormal eigenvectors. For any initial distribution $\vec{p}_0$, we write $\vec{p}_0 = c_1 v_1 + \ldots + c_n v_n$ where $c_i = \langle p_0, v_i \rangle$ for $1 \leq i \leq n$. Then,

$$P^t \vec{p}_0 = \mathcal{A}^t \Big(\sum_{i=1}^{n} c_i v_i \Big) = \sum_{i=1}^{n} c_i \mathcal{A}^t v_i = \sum_{i=1}^{n} c_i \alpha_i^t v_i.$$

The assumptions $\alpha_2 < 1$ and $\alpha_n > -1$ imply that $|\alpha_i| < 1$ for $2 \leq i \leq n$. Hence,

$$\lim_{t\to\infty} P^t \vec{p}_0 = \lim_{t\to\infty} \sum_{i=1}^{n} c_i \alpha_i^t v_i = c_1 v_1,$$

as all but the first term go to zero as $t \to \infty$. In the d-regular case, $v_1 = \vec{1}/\sqrt{n}$ and thus $c_1 = \langle \vec{p}_0, \vec{1}/\sqrt{n} \rangle = 1/\sqrt{n}$ as $\vec{p}_0$ is a probability distribution. Therefore,

$$\lim_{t\to\infty} P^t \vec{p}_0 = c_1 v_1 = \frac{1}{\sqrt{n}} \cdot \frac{\vec{1}}{\sqrt{n}} = \frac{\vec{1}}{n}.$$

□

The proof shows that under $|\alpha_i| < 1$ for $2 \leq i \leq n$, the distribution $P^t \vec{p}_0$ converges to the first eigenvector, which is proportional to the all-one vector. Check that the same proof works for lazy random walks (Corollary 3.12) on d-regular graphs.

Mixing Time

The mixing time quantifies how fast $\vec{p}_t := P^t \vec{p}_0$ converges to the limiting distribution.

The following definition applies to general Markov chains, where the transition matrix is not symmetric. When writing $\vec{p}_0$ and $\vec{p}_t$ as column vectors, $\vec{p}_t = (P^\top)^t \vec{p}_0$.

Definition 3.14 (Mixing Time)**.** *Consider the Markov chain defined by a transition matrix $P \in \mathbb{R}^{n\times n}$. Let $\vec{p}_0 \in \mathbb{R}^n$ be an initial probability distribution on the states, and let $\vec{p}_t \in \mathbb{R}^n$ be the probability distribution $\vec{p}_t := (P^\top)^t \vec{p}_0$ after t steps.*

Suppose the limiting distribution $\vec{\pi} = \lim_{t\to\infty} \vec{p}_t$ exists. For any $0 < \epsilon \leq 1$, the ϵ-mixing time $\tau_\epsilon(P)$ of P is defined as the smallest t such that $d_{\mathrm{TV}}(\vec{p}_t, \vec{\pi}) \leq \epsilon$ for any initial distribution $\vec{p}_0$, where d_{TV} is the total variation distance in Definition 3.7.

When ϵ is not specified, it is assumed to be a small constant such as $1/4$, and we simply say $\tau_{1/4}(P)$ is the mixing time of the Markov chain P.

To bound the mixing time, we use the same approach as in Proposition 3.13, but assume that α_2 and $|\alpha_n|$ are bounded away from one, ensuring α_i^t converges to zero quickly for $2 \le i \le n$.

Theorem 3.15 (Bounding Mixing Time by Spectral Gap)**.** *Let $G = (V, E)$ be a d-regular undirected graph with $V = [n]$. Let $P = \mathcal{A}$ be the transition matrix of random walks on G and $1 = \alpha_1 \ge \alpha_2 \ge \ldots \ge \alpha_n \ge -1$ be its eigenvalues. Let $g := \min\{1-\alpha_2, 1-|\alpha_n|\}$ be the two-sided spectral gap. Then the ϵ-mixing time of P satisfies*

$$\tau_\epsilon(P) \lesssim \frac{1}{g} \ln\left(\frac{n}{\epsilon}\right).$$

Proof. Continuing from Proposition 3.13,

$$P^t \vec{p}_0 = \frac{\vec{1}}{n} + \sum_{i=2}^{n} c_i \alpha_i^t v_i,$$

where $v_1, \ldots, v_n$ are the orthonormal eigenvectors, $c_i = \langle \vec{p}_0, v_i \rangle$ for $2 \le i \le n$, and $\vec{\pi} = \vec{1}/n$ is the limiting distribution. Then,

$$d_{\mathrm{TV}}(\vec{p}_t, \vec{\pi}) = d_{\mathrm{TV}}(P^t \vec{p}_0, \vec{\pi}) = \frac{1}{2}\left\| P^t \vec{p}_0 - \frac{\vec{1}}{n} \right\|_1 = \frac{1}{2}\left\| \sum_{i=2}^{n} c_i \alpha_i^t v_i \right\|_1 \lesssim \sqrt{n} \left\| \sum_{i=2}^{n} c_i \alpha_i^t v_i \right\|_2,$$

where the last inequality follows from $\|v\|_1 \le \sqrt{n}\|v\|_2$ for $v \in \mathbb{R}^n$ (which can be derived from Cauchy-Schwarz). Since $v_1, \ldots, v_n$ are orthonormal,

$$\left\| \sum_{i=2}^{n} c_i \alpha_i^t v_i \right\|_2^2 = \sum_{i=2}^{n} c_i^2 \alpha_i^{2t} \le (1-g)^{2t} \sum_{i=2}^{n} c_i^2.$$

Note that $\sum_{i=2}^n c_i^2 \le \sum_{i=1}^n c_i^2 = \|\vec{p}_0\|_2^2 \le \|\vec{p}_0\|_1^2 = 1$. Therefore,

$$d_{\mathrm{TV}}(\vec{p}_t, \vec{\pi}) \lesssim \sqrt{n} \left\| \sum_{i=2}^{n} c_i \alpha_i^t v_i \right\|_2 \le \sqrt{n(1-g)^{2t} \sum_{i=2}^{n} c_i^2} \le \sqrt{n}(1-g)^t \le \sqrt{n} e^{-gt}.$$

Setting $t \gtrsim \frac{1}{g} \ln(\frac{n}{\epsilon})$ ensures $d_{\mathrm{TV}}(p_t, \vec{\pi}) \le \epsilon$ for any initial distribution p_0. □

For the lazy random walk matrix W, the smallest eigenvalue is at least 0, so the spectral gap for W is simply $g = \frac{1}{2}(1-\alpha_2) = \frac{1}{2}\lambda_2$, where λ_2 is the second smallest eigenvalue of the normalized Laplacian matrix. Cheeger's inequality in Theorem 2.2 then implies the following important consequence.

Theorem 3.16 (Bounding Mixing Time by Edge Conductance)**.** *Let $G = (V, E)$ be a d-regular undirected graph with $V = [n]$. Let $W = \frac{1}{2}I + \frac{1}{2}\mathcal{A}$ be the transition matrix of lazy random walks on G. Then,*

$$\tau_\epsilon(W) \lesssim \frac{1}{\phi(G)^2} \ln\left(\frac{n}{\epsilon}\right).$$

This result provides a combinatorial condition for fast mixing. For an expander graph with $\phi(G) = \Omega(1)$, the mixing time of lazy random walks is $O(\ln n)$. Establishing a polylogarithmic mixing time is crucial for many applications, as we will discuss with examples.

Theorem 3.16 is useful in designing random sampling algorithms. For the purpose of uniform random sampling, the analysis for regular graphs is usually sufficient, as we can set up the Markov chain (e.g., by adding self-loops) so that the underlying graph is regular.

Spectrum of the Transition Matrix for General Graphs

The random walk matrix for general graphs is $P = D^{-1}A$, and the lazy random walk matrix is $W = \frac{1}{2}I + \frac{1}{2}P$. The main difference from the d-regular case is that these matrices are in general not symmetric, and so the spectral theorem in Theorem A.5 cannot be directly applied to reason about their eigenvalues and eigenvectors.

A simple but important observation is that P and W are similar to a real symmetric matrix (see Definition A.3), and so the eignevalues of P and W are still all real numbers.

Lemma 3.17 (Spectrum of Random Walk Matrices)**.** *Let $G = (V, E)$ be a connected undirected graph with $V = [n]$, and $\mathcal{A}$ be its normalized adjacency matrix. Let the eigenvalues of $\mathcal{A}$ be $\alpha_1 > \alpha_2 \geq \ldots \geq \alpha_n$ and let $v_1, v_2, \ldots, v_n$ be a corresponding orthonormal basis of eigenvectors.*

Then the eigenvalues of the random walk matrix $P = D^{-1}A$ are also $\alpha_1 > \alpha_2 \geq \ldots \geq \alpha_n$, and the corresponding eigenvectors of $P^\top = AD^{-1}$ are $D^{\frac{1}{2}}v_1, D^{\frac{1}{2}}v_2, \ldots, D^{\frac{1}{2}}v_n$.

The eigenvalues of the lazy random walk matrix $W = \frac{1}{2}I + \frac{1}{2}P$ are $\frac{1}{2}(1+\alpha_1) > \frac{1}{2}(1+\alpha_2) \geq \ldots \geq \frac{1}{2}(1+\alpha_n)$, and the corresponding eigenvectors of $W^\top$ are $D^{\frac{1}{2}}v_1, D^{\frac{1}{2}}v_2, \ldots, D^{\frac{1}{2}}v_n$.

Proof. Note that $P = D^{-1}A = D^{-\frac{1}{2}}(D^{-\frac{1}{2}}AD^{-\frac{1}{2}})D^{\frac{1}{2}} = D^{-\frac{1}{2}}\mathcal{A}D^{\frac{1}{2}}$, and so P is similar to $\mathcal{A}$ as D is non-singular when the graph is connected. By the same argument, W is similar to $\frac{1}{2}I + \frac{1}{2}\mathcal{A}$. By Fact A.4, P and $\mathcal{A}$ have the same spectrum, and W and $\frac{1}{2}I + \frac{1}{2}\mathcal{A}$ have the same spectrum.

Note that $D^{\frac{1}{2}}v_i$ is an eigenvector of $P^\top$ with eigenvalue α_i, as

$$P^\top\big(D^{\frac{1}{2}}v_i\big) = \big(D^{\frac{1}{2}}\mathcal{A}D^{-\frac{1}{2}}\big)\big(D^{\frac{1}{2}}v_i\big) = D^{\frac{1}{2}}\mathcal{A}v_i = \alpha_i\big(D^{\frac{1}{2}}v_i\big).$$

Similarly, $D^{\frac{1}{2}}v_i$ is an eigenvector of $W^\top$ with eigenvalue $\frac{1}{2}(1+\alpha_i)$. □

The vectors $D^{\frac{1}{2}}v_1, \ldots, D^{\frac{1}{2}}v_n$ are linearly independent because D is non-singular for a connected graph. Note that these vectors are in general not orthonormal with respect to the standard inner product, but they are orthonormal if we use the following weighted inner product:

$$\langle u, v\rangle_{D^{-1}} := u^\top D^{-1} v \quad \text{and} \quad \|v\|_{D^{-1}} := \sqrt{v^\top D^{-1} v}. \tag{3.1}$$

Spectral Analysis for General Undirected Graphs

Using this weighted inner product, we can generalize the spectral analysis in Proposition 3.13 and Theorem 3.15 to non-regular graphs. We describe the main modifications and leave the verification of the details to the reader.

To compute the limiting distribution $P^\top\vec{p}_0 = (D^{\frac{1}{2}}\mathcal{A}D^{-\frac{1}{2}})^t\vec{p}_0 = D^{\frac{1}{2}}\mathcal{A}^tD^{-\frac{1}{2}}\vec{p}_0$, we write the initial distribution $\vec{p}_0$ as $\sum_{i=1}^n c_i D^{\frac{1}{2}}v_i$ to take advantage of the orthonormality of $v_1, \ldots, v_n$, where $c_i = \langle\vec{p}_0, D^{\frac{1}{2}}v_i\rangle_{D^{-1}}$ for $1 \leq i \leq n$. We can then adapt the proof in Proposition 3.13 to prove the following equivalent form of the fundamental theorem for undirected graphs in Theorem 3.10.

Exercise 3.18 (Limiting Distribution for Undirected Graphs)**.** *Let $G = (V, E)$ be an undirected graph with $V = [n]$. Let $P = D^{-1}A$ be the transition matrix of random walks on G, and let $1 = \alpha_1 \geq \alpha_2 \geq \ldots \geq \alpha_n \geq -1$ be its eigenvalues. If $\alpha_2 < 1$ and $\alpha_n > -1$, then*

$$\lim_{t\to\infty}\big(P^\top\big)^t\vec{p}_0 = \frac{\vec{d}}{2|E|},$$

where $\vec{d}$ is the degree vector with $\vec{d}(i) = \deg(i)$ for $1 \le i \le n$.

To bound the mixing time, we adapt the proof in Theorem 3.15. The key steps are

$$\|\vec{p}_t - \vec{\pi}\|_1 \le \|\vec{1}\|_D \cdot \|\vec{p}_t - \vec{\pi}\|_{D^{-1}} \asymp \sqrt{|E|} \cdot \|\vec{p}_t - \vec{\pi}\|_{D^{-1}} \le (1-g)^t \sqrt{|E|} \cdot \|\vec{p}_0\|_{D^{-1}} \tag{3.2}$$

where the first inequality is by Cauchy-Schwarz and the second inequality is by an orthonormality argument as in Theorem 3.15. Then the same theorem as in the d-regular case can be proved.

Theorem 3.19 (Bounding Mixing Time by Spectral Gap and Edge Conductance)**.** *Let $G = (V, E)$ be an undirected graph with $V = [n]$. Let $P = D^{-1}A$ be the transition matrix of random walks on G and $1 = \alpha_1 \ge \alpha_2 \ge \ldots \ge \alpha_n \ge -1$ be its eigenvalues. Let $g := \min\{1 - \alpha_2, 1 - |\alpha_n|\}$ be the two-sided spectral gap. Then the ϵ-mixing time of P satisfies*

$$\tau_\epsilon(P) \lesssim \frac{1}{g} \ln\left(\frac{n}{\epsilon}\right).$$

Let $W = \frac{1}{2}I + \frac{1}{2}D^{-1}A$ be the transition matrix of lazy random walks on G. Then

$$\tau_\epsilon(W) \lesssim \frac{1}{\phi(G)^2} \ln\left(\frac{n}{\epsilon}\right).$$

For weighted undirected graphs, the same arguments can be used to prove that

$$\tau_\epsilon(P) \lesssim \frac{1}{g} \ln\left(\frac{1}{\epsilon \cdot \pi_{\min}}\right) \quad \text{and} \quad \tau_\epsilon(W) \lesssim \frac{1}{\phi(G)^2} \ln\left(\frac{1}{\epsilon \cdot \pi_{\min}}\right), \tag{3.3}$$

where $\pi_{\min} := \min_i \vec{\pi}(i)$ is the minimum stationary probability of a vertex. We leave the verification of these bound to Problem 3.22.

Remark 3.20. *This spectral approach can be further extended to prove the fundamental theorem for directed graphs, but it is considerably more involved and requires the Perron-Frobenius theorem and the Jordan normal form (see [HJ13] for proofs).*

3.4 Applications of Random Walks

We briefly discuss two applications of random walks that will be studied in later chapters. In both cases, random walks are used to solve the problem while exploring only a small portion of the graph.

Random Sampling

An important application of random walks is in random sampling. As an example, consider the following algorithm for generating a random spanning tree of an undirected graph.

To analyze this algorithm, we interpret it as performing random walks on a large "spanning tree exchange graph" H. In H, each vertex represents a spanning tree of the original graph, and two vertices are connected if their corresponding spanning trees T and T' can be obtained through one step of the algorithm (i.e. $T' = T + e - f$ for some edges e, f in the input graph).

Note that the exchange graph H could have as many as $\Omega(n^{n-2})$ vertices when the original graph has n vertices. Therefore, to show that $\tau \lesssim \text{poly}(n)$ is sufficient to return an almost uniform random

Algorithm 2 Random Exchange Algorithm for Sampling Random Spanning Trees

Require: An undirected graph $G = (V, E)$.

Compute an arbitrary spanning tree T_0 of the graph.
for $1 \leq t \leq \tau$ **do**
  Remove a uniform random edge $e \in T_{t-1}$.
  Let f be a uniform random edge in $E \setminus T_{t-1}$ that connects the two components of $T_{t-1} - e$.
  Set $T_t := T_{t-1} - e + f$.
end for
return T_τ.

spanning tree, we must prove that the random walks on H mix in polylogarithmic time relative to its size. From a combinatorial perspective, this requires proving that the spanning tree exchange graph H is an expander graph, a task that is generally quite challenging.

There are different approaches to proving fast mixing of Markov chains. One is the coupling method, the most common and versatile probabilistic technique in bounding mixing time (see [LPW06]). Another is the canonical path method, which uses multicommodity flow to lower bound the graph conductance, so that Theorem 3.16 can be used to upper bound the mixing time. A famous application of the canonical path method is in approximating the permanent of a non-negative matrix [JSV04], which is equivalent to counting the number of perfect matchings in a bipartite graph.

These methods are beyond the scope of this course. Instead, we will analyze the random exchange algorithm for sampling random spanning trees using the new techniques derived from high dimensional expanders in Chapter 26, Chapter 27, and Chapter 28.

Local Graph Partitioning

Another useful application of random walks is in graph partitioning. This idea, originally proposed by Spielman and Teng [ST13], is to use the random walk distribution $W^t\chi_i$ from some starting vertex i to identify a small sparse cut of the graph. They proved that the performance of the random walk algorithm for graph partitioning is comparable to that of the spectral partitioning algorithm in Chapter 2. Furthermore, the random walk algorithm has the significant advantage that it can be implemented locally, such that the running time depends only on output size but not on the original graph size. This provides a sublinear time algorithm for graph partitioning in some situations. Local graph partitioning is an active research topic on its own, and there are several other algorithms such as using PageRank vector [ACL06] and evolving sets [AOPT16]. We will discuss these results in Chapter 11.

3.5 Random Walks on Directed Graphs

For directed graphs, there are currently no direct relationships between the eigenvalues of their transition matrix and the mixing time of random walks.

In this section, we discuss some known results about the mixing time of random walks on directed graphs, using the second eigenvalue of symmetric matrices associated with directed graphs.

Stationary Flow Graph

Given a directed graph $G = (V, E)$ with an edge weight function $w : E \to \mathbb{R}_+$, let P be the transition matrix of the random walk, where $P_{i,j} = w(ij)/\sum_{j:ij\in E} w(ij)$. Assume P is irreducible and aperiodic. Let $\vec{\pi}$ be the unique stationary distribution of P, and define $\Pi := \mathrm{diag}(\vec{\pi})$.

To study the mixing time of the weighted directed graph $G = (V, E, w)$, it is helpful to consider the stationary flow graph $G_f = (V, E, f)$, where $f(i, j) = \vec{\pi}(i) \cdot P_{ij}$ is the probability flow on edge ij in the stationary distribution $\vec{\pi}$. Verify that the weighted directed graph $G_f = (V, E, f)$ is Eulerian, satisfying $\sum_{j:ji\in E} f(j, i) = \sum_{k:ik\in E} f(i, k)$ for all $i \in V$.

The adjacency matrix of G_f is denoted by $F := \Pi P$. The Eulerian property implies that the i-th row sum of F is equal to the i-th column sum of F for all i.

Symmetric Matrices for Directed Graphs

Fill [Fil91] defined the sum matrix as

$$\mathfrak{A} := \frac{1}{2}(P + \Pi^{-1}P^\top\Pi).$$

Chung [Chu05] defined the directed Laplacian matrix of G as

$$\mathfrak{L} = I - \frac{1}{2}\Big(\Pi^{\frac{1}{2}}P\Pi^{-\frac{1}{2}} + \Pi^{-\frac{1}{2}}P^\top\Pi^{\frac{1}{2}}\Big) = I - \Pi^{-\frac{1}{2}}\left(\frac{F + F^\top}{2}\right)\Pi^{-\frac{1}{2}}. \tag{3.4}$$

Observe that $\mathfrak{L}$ is the normalized Laplacian matrix of the symmetrized flow graph where the weight of edge ij is $\frac{1}{2}(f(i, j) + f(j, i))$, as the diagonal degree matrix of the symmetrized flow graph is still Π because of the Eulerian property.

Note that the spectra of $\mathfrak{A}$ and $\mathfrak{L}$ are essentially the same, as $\mathfrak{A}$ and $I - \mathfrak{L}$ are similar matrices.

Bounding Mixing Time by Spectral Gap of Symmetric Matrix

A main result from [Fil91, Chu05] uses the spectral gap of $\mathfrak{A}$ or $\mathfrak{L}$ to bound the mixing time of random walks on G.

Theorem 3.21 (Bounding Mixing Time by Second Eigenvalue of Directed Graphs [Fil91, Chu05])**.** *Let G be a strongly connected directed graph $G = (V, E)$ with a weight function $w : E \to \mathbb{R}_+$, and P be the transition matrix of the random walks on G with $P(i, j) = w(ij)/\sum_{i\in V} w(ij)$ for $ij \in E$. The ϵ-mixing time of the lazy random walks on G (with transition matrix $\frac{1}{2}(I+P)$) to the stationary distribution $\vec{\pi}$ satisfies*

$$\tau_\epsilon\Big(\frac{I + P}{2}\Big) \lesssim \frac{1}{\lambda_2(\mathfrak{L})} \cdot \log\Big(\frac{1}{\pi_{\min} \cdot \epsilon}\Big)$$

where $\lambda_2(\mathfrak{L})$ is the second smallest eigenvalue of $\mathfrak{L}$ in (3.4), and $\pi_{\min} = \min_{i\in V} \vec{\pi}(i)$.

Cheeger Constant of Directed Graphs

The Cheeger constant of a set and of a directed graph [Fil91, Chu05, LPW06] are defined as

$$h(S) := \frac{\sum_{i\in S, j\notin S} \pi(i)P(i, j)}{\pi(S)} = \frac{\sum_{i\in S, j\notin S} F(i, j)}{\pi(S)} \quad \text{and} \quad h(G) := \min_{S:\pi(S)\le\frac{1}{2}} h(S). \tag{3.5}$$

This is also known as the conductance or the bottleneck ratio in the literature [LPW06].

Since the flow graph F is Eulerian, the Cheeger constant of the flow graph is the same as that of the symmetrized flow graph $\frac{1}{2}(F + F^\top)$. So the following Cheeger's inequality by Chung [Chu05] for directed graphs is a direct consequence of the Cheeger's inequality for undirected graphs in Theorem 2.2:

$$\frac{1}{2}\lambda_2(\mathfrak{L}) \le h(G) \le \sqrt{2\lambda_2(\mathfrak{L})}.$$

A consequence of Theorem 3.21 is that

$$\tau_\epsilon\Big(\frac{I+P}{2}\Big) \lesssim \frac{1}{h(G)^2} \cdot \log\Big(\frac{1}{\pi_{\min}\cdot\epsilon}\Big).$$

It is also possible to directly prove this consequence using combinatorial methods [LS93, LPW06].

3.6 Problems

Problem 3.22 (Weighted Undirected Graphs)**.** *Extend the proof of Theorem 3.19 to establish (3.3). Fill in the proof details for Exercise 3.18 and Theorem 3.19.*

Problem 3.23 (Upper Bound on Mixing Time and Initial Distribution)**.** *Suppose the initial distribution $\vec{p}$ satisfies $\vec{p}(i) \le 2\pi(i)$ for all i where π is the unique stationary distribution. Prove that*

$$\tau_\epsilon(P) \lesssim \frac{1}{g}\ln\Big(\frac{1}{\epsilon}\Big) \quad \text{and} \quad \tau_\epsilon(W) \lesssim \frac{1}{\phi(G)^2}\ln\Big(\frac{1}{\epsilon}\Big).$$

In other words, the factor $\log(n)$ in Theorem 3.19 is only needed to get away from distributions concentrated on a small set.

Problem 3.24 (Lower Bound on Mixing Time)**.** *Let $G = (V, E)$ be an undirected graph with $V = [n]$. Let $W = \frac{1}{2}I + \frac{1}{2}D^{-1}A$ be the transition matrix of lazy random walks on G. Prove that the ϵ-mixing time of W is*

$$\tau_\epsilon(W) \gtrsim \frac{1}{1-\alpha_2}\ln\Big(\frac{1}{\epsilon}\Big),$$

where α_2 is the second largest eigenvalue of the normalized adjacency matrix $\mathcal{A}(G)$. A simpler problem is to prove that

$$\tau_\epsilon(W) \gtrsim \frac{1}{\phi(G)}\ln\Big(\frac{1}{\epsilon}\Big),$$

where $\phi(G)$ is the edge conductance of G. You may also consider the special case when G is d-regular.

Problem 3.25 (Page Ranking)**.** *Suppose someone searches a keyword (e.g., "car"), and we want to identify the webpages that are the most relevant for this keyword and those that are the most reliable sources (a page is considered reliable if it points to many highly relevant pages).*

First, we identify the pages with this keyword and ignore all others. Then we run the following ranking algorithm on the remaining pages. Each vertex corresponds to a remaining page, and there is a directed edge from page i to page j if there is a link from page i to page j. Call this directed graph $G = (V, E)$.

For each vertex i, we have two values, $s(i)$ and $r(i)$, where $r(i)$ represents the relevance of the page and $s(i)$ represents its reliability as a source (larger values are better). We start with arbitrary initial values, such as $s(i) = 1/|V|$ for all i, as we have no prior information.

At each step, we update s and r (where s and r are vectors of $s(i)$ and $r(i)$ values) as follows:

1. *Update* $r(i) = \sum_{j:ji \in E} s(j)$ *for all* i*, as a page is more relevant if it is linked by many reliable sources.*

2. *Update* $s(i) = \sum_{j:ij \in E} r(j)$ *for all* i *(using the just updated values* $r(j)$*), as a page is a more reliable source if it points to many relevant pages.*

To keep the values bounded, let $R = \sum_{i=1}^{|V|} r(i)$ *and* $S = \sum_{i=1}^{|V|} s(i)$*, and normalize by dividing each* $s(i)$ *by* S *and divide each* $r(i)$ *by* R*. We repeat these steps multiple times to refine the values.*

Let $s, r \in \mathbb{R}^{|V|}$ *be the vectors of the* s *and* r *values. Provide a matrix formulation for computing* s *and* r*.*

Suppose G *is weakly connected (the underlying undirected graph is connected when ignoring edge directions) and has a self-loop at each vertex. Prove that there is a unique limiting* s *and a unique limiting* r *for any initial* s*, provided* $s \geq 0$ *and* $s \neq 0$*. You may use the Perron-Frobenius Theorem A.18 to solve this problem.*

References

[ACL06] Reid Andersen, Fan R. K. Chung, and Kevin J. Lang. Local graph partitioning using pagerank vectors. In *FOCS*, pages 475–486. IEEE Computer Society, 2006. 48, 143, 155

[AOPT16] Reid Andersen, Shayan Oveis Gharan, Yuval Peres, and Luca Trevisan. Almost optimal local graph clustering using evolving sets. *J. ACM*, 63(2):15:1–15:31, 2016. 48, 143, 155

[Chu05] Fan Chung. Laplacians and cheeger inequality for directed graphs. *Annals of Combinatorics*, 9:1–19, 2005. 49, 50

[Fil91] James Allen Fill. Eigenvalue bounds on convergence to stationary for nonreversible markov chains, with an application to the exclusion process. *The Annals of Applied Probability*, 1:62–87, 1991. 49

[Häg02] Olle Häggström. *Finite Markov Chains and Algorithmic Applications*. London Mathematical Society Student Texts. Cambridge University Press, 2002. 39, 40, 41

[HJ13] Roger A. Horn and Charles R. Johnson. *Matrix Analysis*. Cambridge University Press, Cambridge; New York, 2nd edition, 2013. 41, 47, 370, 371

[JSV04] Mark Jerrum, Alistair Sinclair, and Eric Vigoda. A polynomial-time approximation algorithm for the permanent of a matrix with nonnegative entries. *J. ACM*, 51(4):671–697, 2004. 48

[LPW06] David A. Levin, Yuval Peres, and Elizabeth L. Wilmer. *Markov chains and mixing times*. American Mathematical Society, 2006. 7, 39, 40, 41, 48, 49, 50

[LS93] László Lovász and Miklós Simonovits. Random walks in a convex body and an improved volume algorithm. *Random structures & algorithms*, 4(4):359–412, 1993. 50, 136

[ST13] Daniel A. Spielman and Shang-Hua Teng. A local clustering algorithm for massive graphs and its application to nearly linear time graph partitioning. *SIAM J. Comput.*, 42(1):1–26, 2013. 8, 48, 135, 136, 140, 142, 143, 154

Chapter 4

Expander Graphs: Properties

There are several possible ways to define expander graphs:

1. Algebraically, expander graphs are graphs with a large spectral gap.
2. Combinatorically, expander graphs are graphs with very good connectivity properties.
3. Probabilistically, expander graphs are graphs in which random walks mix rapidly.

From what we have learnt in Chapter 2 and Chapter 3, these three perspectives are closely related:

- Cheeger's inequality in Theorem 2.2 states that $\phi(G) = \Omega(1)$ if and only if $\lambda_2 = \Omega(1)$.
- The spectral analysis in Theorem 3.16 and Problem 3.24 show that the mixing time τ of the lazy random walks on $G = (V, E)$ satisfies

$$\frac{1}{\lambda_2} \lesssim \tau \lesssim \frac{1}{\lambda_2} \log |V|.$$

Complete graphs are the best expander graphs under all three definitions, but we are interested in sparse expander graphs with a linear number of edges. In constructions of expander graphs, the spectral definition is the most convenient. We use the following stronger spectral definition that also bounds the last eigenvalue.

Definition 4.1 (Two-Sided Spectral Expanders)**.** *Let G be a d-regular graph and let the spectrum of its adjacency matrix be*

$$d = \alpha_1 \geq \alpha_2 \geq \ldots \geq \alpha_n \geq -d.$$

We say that G is an (n, d, α)-graph if G has n vertices, G is d-regular, and $\max\{\alpha_2, |\alpha_n|\} \leq \alpha$.

In this chapter, we first prove the expander mixing lemma and its converse, providing a combinatorial characterization of two-sided spectral expanders. Then we discuss the extremal question of how small can α be. Finally, we investigate the stronger combinatorial and probabilistic properties that an (n, d, α)-graph has when $\alpha = o(d)$, including small-set vertex expansion and constant mixing time.

4.1 Expander Mixing Lemma

A well-known and useful property of two-sided spectral expanders is that they behaves as random d-regular graphs. Consider the number of edges between two subsets S, T of vertices.

Definition 4.2 (Induced Edges)**.** *Given an undirected graph $G = (V, E)$ and $S, T \subseteq V$, define $\vec{E}(S,T) := \{(u,v) \mid u \in S, v \in T, uv \in E\}$ as the set of ordered pairs where $u \in S$ and $v \in T$. Note that an edge with $u \in S \cap T$ and $v \in S \cap T$ is counted twice, as both (u,v) and (v,u) are in $\vec{E}(S,T)$.*

In a random graph where every pair of vertices has an edge with probability d/n, the expected value of $|\vec{E}(S,T)|$ is $d|S||T|/n$. The expander mixing lemma by Alon and Chung [AC88] says that in a two-sided spectral expander, $|\vec{E}(S,T)|$ is close to this expected value for all $S, T \subseteq V$. This can be interpreted as a pseudorandom or discrepancy property of a two-sided spectral expander.

Theorem 4.3 (Expander Mixing Lemma [AC88])**.** *Let $G = (V, E)$ be an (n, d, α)-graph. Then, for every $S \subseteq V$ and $T \subseteq V$,*

$$\left| |\vec{E}(S,T)| - \frac{d|S||T|}{n} \right| \leq \alpha\sqrt{|S||T|}.$$

Proof. First, we write $|\vec{E}(S,T)|$ as an algebraic expression. Let χ_S and χ_T be the characteristic vectors of S and T. Notice that $|\vec{E}(S,T)| = \chi_S^\top A \chi_T$, where A is the adjacency matrix of G.

Next, we use eigen-decompositions of χ_S and χ_T to relate $|\vec{E}(S,T)|$ to the eigenvalues of A. Let $v_1, \ldots, v_n$ be an orthonormal basis of eigenvectors of A. Write $\chi_S = \sum_{i=1}^n a_i v_i$ and $\chi_T = \sum_{j=1}^n b_j v_j$, where $a_i = \langle \chi_S, v_i \rangle$ and $b_j = \langle \chi_T, v_j \rangle$. Recall that $\alpha_1 = d$ and $v_1 = \vec{1}/\sqrt{n}$, so $a_1 = |S|/\sqrt{n}$ and $b_1 = |T|/\sqrt{n}$. Then, by orthonormality of $v_1, \ldots, v_n$,

$$|\vec{E}(S,T)| = \chi_S^\top A \chi_T = \Big(\sum_{i=1}^n a_i v_i\Big)^\top A \Big(\sum_{j=1}^n b_j v_j\Big) = \sum_{i=1}^n \alpha_i a_i b_i = \frac{d|S||T|}{n} + \sum_{i=2}^n \alpha_i a_i b_i.$$

Therefore, by the definition of α and the Cauchy-Schwarz inequality,

$$\left| |\vec{E}(S,T)| - \frac{d|S||T|}{n} \right| = \left| \sum_{i=2}^n \alpha_i a_i b_i \right| \leq \alpha \sum_{i=2}^n |a_i||b_i| \leq \alpha \|\vec{a}\|_2 \|\vec{b}\|_2 = \alpha \|\chi_S\|_2 \|\chi_T\|_2 = \alpha\sqrt{|S||T|},$$

where $\vec{a} = (a_1, \ldots, a_n)$ and $\vec{b} = (b_1, \ldots, b_n)$ with $\|\vec{a}\|_2 = \|\chi_S\|_2$ and $\|\vec{b}\|_2 = \|\chi_T\|_2$. □

See Problem 4.16 for a slightly better upper bound for the expander mixing lemma. The same proof can be extended to non-regular graphs using the normalized adjacency matrix.

Exercise 4.4 (Expander Mixing Lemma for Non-Regular Graphs)**.** *Let $G = (V, E)$ be an undirected graph with $|V| = n$. Let $\mathcal{A}$ be its normalized adjacency matrix with eigenvalues $1 = \alpha_1 \geq \alpha_2 \geq \cdots \geq \alpha_n \geq -1$ and let $\alpha := \max\{\alpha_2, |\alpha_n|\}$. Prove that, for every $S \subseteq V$ and $T \subseteq V$,*

$$\left| |\vec{E}(S,T)| - \frac{\mathrm{vol}(S) \cdot \mathrm{vol}(T)}{\mathrm{vol}(V)} \right| \leq \alpha\sqrt{\mathrm{vol}(S) \cdot \mathrm{vol}(T)}.$$

The following is an application of the expander mixing lemma.

Exercise 4.5 (Maximum Independent Set and Chromatic Number of Two-Sided Spectral Expanders)**.** *Let $G = (V, E)$ be an (n, d, α) graph. Show that the size of a maximum independent set is at most $\alpha n/d$. Conclude that an (n, d, α)-graph has chromatic number at least d/α.*

4.2 Converse of Expander Mixing Lemma

Interestingly, Bilu and Linial [BL06] proved a converse of the expander mixing lemma, showing that it comes close to providing a combinatorial characterization of two-sided spectral expanders.

Theorem 4.6 (Converse of Expander Mixing Lemma [BL06]). *Let $G = (V = [n], E)$ be a d-regular graph. Suppose that*

$$\left| |\vec{E}(S,T)| - \frac{d|S||T|}{n} \right| \leq \alpha\sqrt{|S||T|} \quad \text{for all } S,T \subseteq V \text{ with } S \cap T = \emptyset. \tag{4.1}$$

Then all but the largest eigenvalue of $A(G)$ are bounded in absolute value by $O\big(\alpha(1+\log\frac{d}{\alpha})\big)$.

Compared to Cheeger's inequality in Theorem 2.2, this provides a tighter relationship between the spectral quantity and the combinatorial property, without a square root loss.

The proof of Theorem 4.6 is based on the following linear algebraic lemma, where the spectral radius of a matrix is defined as the maximum of the absolute values of its eigenvalues.

Lemma 4.7 (Bounding Spectral Radius [BL06]). *Let B be an $n \times n$ real symmetric matrix such that the ℓ_1-norm of each row of B is at most d, and all diagonal entries of B have absolute value $O\big(\alpha(1+\log\frac{d}{\alpha})\big)$ for some $1 \leq \alpha \leq d$. Suppose that*

$$\left|\chi_S^\top B \chi_T\right| \leq \alpha\sqrt{|S||T|} \quad \text{for all } S,T \subseteq [n] \text{ with } S \cap T = \emptyset. \tag{4.2}$$

Then the spectral radius of B is $O\big(\alpha(1+\log\frac{d}{\alpha})\big)$.

We first explain how the lemma implies the converse of the expander mixing lemma.

Proof of Theorem 4.6 Assuming Lemma 4.7. The idea is to apply Lemma 4.7 to the error matrix $B := A(G) - d \cdot v_1 v_1^\top$, where $v_1 = \vec{1}/\sqrt{n}$ is the first eigenvector of $A(G)$. Let the eigenvalues of A be $(d, \alpha_2, \ldots, \alpha_n)$. Then, the eigenvalues of B are $(0, \alpha_2, \ldots, \alpha_n)$, as A and B have the same eigenvectors. Therefore, bounding $\max_{2\leq i\leq n}|\alpha_i|$ is equivalent to bounding the spectral radius of B.

Note that the ℓ_1-norm of each row of B is at most $2d$, the diagonal entries have absolute value at most 1, and that the assumption in (4.1) and the assumption in (4.2) are equivalent. Therefore, we can apply Lemma 4.7 to obtain that the spectral radius of B is $O\big(\alpha(1+\log\frac{d}{\alpha})\big)$, and this implies that $\max_{2\leq i\leq n}|\alpha_i| \lesssim \alpha(1+\log\frac{d}{\alpha})$. □

The proof of Lemma 4.7 in [BL06] is by combining the constraints in (4.2) to establish that

$$\frac{|x^\top B x|}{x^\top x} \lesssim \alpha\left(1+\log\frac{d}{\alpha}\right) \quad \text{for all } x \in \mathbb{R}^n. \tag{4.3}$$

The combination of the constraints is guided by linear programming duality, and as a result the proof is not quite intuitive and it is not clear how the numbers were chosen.

We provide a different presentation of their proof following Trevisan's style. We prove the contrapositive that if (4.3) is violated then there must be a violating constraint in (4.2). To do so, we use a simple randomized rounding argument as in the proof of the hard direction of Cheeger's inequality in Section 2.3. This argument may clarify how the numbers in the proof are chosen.

Proof of a Weaker Version

To highlight the main idea, we first prove a weaker version of Lemma 4.7. We then explain the modifications needed to match Lemma 4.7 in the next subsection.

Contrapositive: In this subsection, our goal is to prove the weaker statement that

$$\exists x \in \mathbb{R}^n \text{ with } \frac{|x^\top B x|}{\|x\|_2^2} > 2(\log_2 n + 1)\alpha \quad \Longrightarrow \quad \exists y, z \in \{-1, 0, 1\}^n \text{ with } \frac{|y^\top B z|}{\|y\|_2 \|z\|_2} > \alpha.$$

Assumption: We assume that $\|x\|_2 = 1$ and each entry of x is a negative power of two. This is a natural first step to discretize x, and we will justify this assumption in the next subsection.

Notations: Let $S_k := \{i \in [n] \mid |x(i)| = 2^{-k}\}$ be the set of indices with absolute value 2^{-k}, and let $s_k := |S_k|$. Let χ_k be the signed pattern of x restricted to S_k, such that $\chi_k(i) = 1$ if $x(i) = 2^{-k}$, $\chi_k(i) = -1$ if $x(i) = -2^k$, and $\chi_k(i) = 0$ otherwise. Note that $x = \sum_k 2^{-k}\chi_k$.

Probability Distribution: We sample y and z independently from the same distribution, where

$$y, z := \chi_k \text{ with probability } 2^{-k}/c \text{ if } S_k \neq \emptyset,$$

and $c := \sum_{k:S_k \neq \emptyset} 2^{-k}$ is the normalizing constant to make this a probability distribution.

Probabilistic Arugment: As in the proof of Cheeger's inequality, we argue by Lemma 2.6 that there exist $y, z \in \{-1, 0, 1\}^n$ with

$$\frac{|y^\top B z|}{\|y\|_2 \cdot \|z\|_2} \geq \frac{\mathbb{E}\left[|y^\top B z|\right]}{\mathbb{E}\left[\|y\|_2 \cdot \|z\|_2\right]},$$

and so it remains to compute the expected values separately.

Expected Numerator: By the triangle inequality,

$$c^2 \cdot \mathbb{E}\left[|y^\top B z|\right] = \sum_k \sum_l 2^{-k} \cdot 2^{-l} \cdot |\chi_k^\top B \chi_l| \geq \left|\left(\sum_k 2^{-k}\chi_k\right)^\top B \left(\sum_l 2^{-l}\chi_l\right)\right| = |x^\top B x|. \quad (4.4)$$

This is the main motivation for defining the probability distribution in this way, so that the expected numerator can be easily compared to the numerator $|x^\top B x|$.

Expected Denominator: By independence,

$$c^2 \cdot \mathbb{E}\left[\|y\|_2 \cdot \|z\|_2\right] = c^2 \cdot \mathbb{E}\left[\|y\|_2\right]^2 = \left(\sum_k 2^{-k}\|\chi_k\|_2\right)^2 = \sum_k \sum_l 2^{-k} \cdot 2^{-l} \cdot \sqrt{s_k s_l}. \quad (4.5)$$

We would like to compare this to the denominator

$$x^\top x = \left(\sum_k 2^{-k}\chi_k\right)^\top \left(\sum_l 2^{-l}\chi_l\right) = \sum_k 2^{-2k} s_k. \quad (4.6)$$

To do so, we divide the right hand side of (4.5) into three terms:

$$\sum_k 2^{-2k} s_k \; + \sum_k \sum_{l:k<l\leq k+\log n} 2^{-k-l+1}\sqrt{s_k s_l} \; + \sum_k \sum_{l:l>k+\log n} 2^{-k-l+1}\sqrt{s_k s_l}.$$

Using the AM-GM inequality, the second term is bounded by

$$\sum_k \sum_{l:k<l\le k+\log n} 2^{-k-l+1}\sqrt{s_k s_l} \le \sum_k \sum_{l:k<l\le k+\log n} (2^{-2k}s_k + 2^{-2l}s_l) \le 2\log n \cdot \sum_k 2^{-2k}s_k. \tag{4.7}$$

The third term is bounded by

$$\sum_k \sum_{l:l>k+\log n} 2^{-2k-\log n}\sqrt{s_k s_l} = \frac{1}{n}\sum_k 2^{-2k}\sqrt{s_k}\sum_l \sqrt{s_l} \le \sum_k 2^{-2k}\sqrt{s_k} \le \sum_k 2^{-2k}s_k,$$

where the second last inequality uses $\sum_l \sqrt{s_l} \le \sqrt{n}\sqrt{\sum_l s_l} = n$ by Cauchy-Schwarz and $\sum_l s_l \le n$. Combining these inequalities,

$$c^2 \cdot \mathbb{E}\left[\|y\|_2 \cdot \|z\|_2\right] \le 2(\log n + 1)\sum_k 2^{-2k}s_k = 2(\log n+1)\cdot x^\top x.$$

Conclusion: Therefore, there exist $y, z \in \{-1,0,1\}^n$ such that

$$\frac{|y^\top Bz|}{\|y\|_2\|z\|_2} \ge \frac{\mathbb{E}\left[|y^\top Bz|\right]}{\mathbb{E}\left[\|y\|_2\cdot\|z\|_2\right]} \ge \frac{|x^\top Bx|}{2(\log n+1)\cdot\|x\|_2^2} > \alpha.$$

We would like to reduce the $\log n$ factor to $\alpha\log(d/\alpha)$. Observe that we did not use the assumption about the ℓ_1-norm of the rows in this proof. To exploit this assumption, we will modify the probability distribution to sample y and z.

Proof of Lemma 4.7

The proof has a similar structure to that in the previous subsection. We explain the modifications and the missing details here.

Zero Diagonal Entries: We assume that the diagonal entries of B are zero. See Exercise 4.17.

Contrapositive: To prove Lemma 4.7, we prove the contrapositive that

$$\exists x \in \mathbb{R}^n \text{ with } \frac{|x^\top Bx|}{\|x\|_2^2} > \alpha\Big(\log\frac{d}{\alpha}+1\Big) \implies \exists y,z\in\{0,1\}^n, \langle y,z\rangle = 0 \text{ with } \frac{|y^\top Bz|}{\|y\|_2\|z\|_2} \gtrsim \alpha.$$

Negative Powers of Two: We rescale x to satisfy $\|x\|_2 = 1$. Then, using a simple rounding argument in Problem 4.18, we construct a vector $\tilde{x} \in \mathbb{R}^n$ such that each entry of $\tilde{x}$ is a negative power of two and

$$\frac{|\tilde{x}^\top B\tilde{x}|}{\|\tilde{x}\|_2^2} \ge \frac{1}{4}\cdot\frac{|x^\top Bx|}{\|x\|_2^2}.$$

We let $x := \tilde{x}$ in the following. We use the same notations $S_k \subseteq [n]$ and $\chi_k \in \{-1,0,1\}^n$ as before.

Probability Distribution: We sample $y, z \in \{-1,0,1\}^n$ jointly (not independently) where

$$(y,z) = (\chi_k,\chi_l) \text{ with probability } 2^{-k-l}/c \text{ if } S_k \ne \emptyset, S_l \ne \emptyset, \text{ and } |k-l| \le \gamma,$$

where c is the normalization constant that makes this a probability distribution, and γ is a parameter that we will choose to be $\log(d/\alpha)$.

This is the main modification, where pairs (χ_k,χ_l) with $|k-l| > \gamma$ are not sampled. The motivation is to avoid the $\log n$ loss in the denominator in the previous analysis. However, this will make the numerator smaller, and the choice of γ is to balance the denominator and the numerator.

Expected Numerator:

$$c \cdot \mathbb{E}\left[\left|y^\top B z\right|\right] = \sum_{k,l:|k-l|\le\gamma} 2^{-k-l} \cdot |\chi_k^\top B \chi_l| \ge |x^\top B x| - \sum_{k,l:|k-l|>\gamma} 2^{-k-l} |\chi_k^\top B \chi_l|,$$

where the inequality is from the previous analysis (4.4) that $\sum_{k,l} 2^{-k-l}|\chi_k^\top B\chi_l| \ge |x^\top Bx|$.

The key observation is that the second term can be bounded using the ℓ_1-norm assumption:

$$\sum_{k,l:|k-l|>\gamma} 2^{-k-l}|\chi_k^\top B\chi_l| = \sum_{k,l:l>k+\gamma} 2^{-k-l+1}|\chi_k^\top B\chi_l| \le 2^{-\gamma}\sum_k 2^{-2k} \sum_{l>k+\gamma} |\chi_k^\top B\chi_l| \le 2^{-\gamma} d \sum_k 2^{-2k} s_k,$$

where the last inequality holds as $\sum_l |\chi_k^\top B\chi_l| \le d \cdot s_k$ which follows from the ℓ_1-norm assumption. Therefore, by (4.6),

$$c \cdot \mathbb{E}\left[\left|y^\top Bz\right|\right] \ge |x^\top Bx| - 2^{-\gamma} d \sum_k 2^{-2k} s_k = |x^\top Bx| - 2^{-\gamma} d \cdot x^\top x.$$

Expected Denominator: Recall that $\|\chi_k\|_2 = \sqrt{s_k}$, so

$$c \cdot \mathbb{E}\left[\|y\|_2 \cdot \|z\|_2\right] = \sum_{k,l:|k-l|\le\gamma} 2^{-k-l}\sqrt{s_k s_l} = \sum_k 2^{-2k} s_k + \sum_k \sum_{l:k<l\le k+\gamma} 2^{-k-l+1}\sqrt{s_k s_l}.$$

Using the same calculation in (4.7), we obtain that

$$c \cdot \mathbb{E}\left[\|y\|_2 \cdot \|z\|_2\right] \le \sum_k 2^{-2k} s_k + 2\gamma \sum_k 2^{-2k} s_k = (2\gamma+1) x^\top x.$$

Good $\{-1,0,1\}^n$ Vectors: By Lemma 2.6, there exist $y, z \in \{-1,0,1\}^n$ such that

$$\frac{|y^\top Bz|}{\|y\|_2\|z\|_2} \ge \frac{\mathbb{E}\left[\left|y^\top Bz\right|\right]}{\mathbb{E}\left[\|y\|_2 \cdot \|z\|_2\right]} \ge \frac{|x^\top Bx| - 2^{-\gamma} d \cdot x^\top x}{(2\gamma+1)x^\top x} = \underbrace{\frac{|x^\top Bx|}{(2\gamma+1)x^\top x}}_{(*)} - \underbrace{\frac{2^{-\gamma} d}{2\gamma+1}}_{(**)} \gtrsim \alpha,$$

where the last inequality holds by choosing $\gamma = \log(d/\alpha)$ and the assumption in the contrapositive.

To see how to choose γ and set the assumption in the contrapositive, the idea is to ensure the $(*)$ is $\Omega(\alpha)$ and $(**)$ is $o(\alpha)$. Setting $\gamma = \log_2(d/\alpha)$ ensures that the second term is $o(\alpha)$, and setting $|x^\top Bx|/x^\top x \gtrsim \gamma \cdot \alpha$ in the assumption of the contrapositive ensures that the first term is $\Omega(\alpha)$.

Good $\{0,1\}^n$ Vectors: Write $y = y^+ - y^-$ and $z = z^+ - z^-$, where $y^+, y^-, z^+, z^- \in \{0,1\}^n$. Let $(\bar{y}, \bar{z})$ be one of the four options of $(y^\pm, z^\pm)$ that maximizes the absolute value of the quadratic form. Check that

$$\frac{|\bar{y}^\top B\bar{z}|}{\|\bar{y}\|_2\|\bar{z}\|_2} \ge \frac{1}{4} \cdot \frac{|y^\top Bz|}{\|y\|_2\|z\|_2}.$$

Disjoint Supports: If $\mathrm{supp}(\bar{y}) \cap \mathrm{supp}(\bar{z}) = \emptyset$, we are done. Otherwise, by our construction, $\mathrm{supp}(\bar{y}) = \mathrm{supp}(\bar{z})$. Let $Y := \{i \in [n] \mid \bar{y}(i) = 1\}$. Consider a random partition (S,T) of Y where each $i \in Y$ is put in S with probability $1/2$ and in T with probability $1/2$ independently. Use the assumption that the diagonal entries of B are zero to argue that there exists a partition $S \cup T = Y$ with

$$\frac{|\chi_S^\top B\chi_T|}{\|\chi_S\|_2\|\chi_T\|_2} \ge \frac{1}{2} \cdot \frac{|\bar{y}^\top B\bar{y}|}{\|\bar{y}\|_2^2}.$$

Conclusion: The proof of Lemma 4.7 follows by chaining together the inequalities:

$$\frac{|\chi_S^\top B \chi_T|}{\|\chi_S\|_2 \|\chi_T\|_2} \gtrsim \frac{|\bar{y}^\top B \bar{y}|}{\|\bar{y}\|_2^2} \gtrsim \frac{|y^\top B z|}{\|y\|_2 \|z\|_2} \geq \frac{|\tilde{x}^\top B \tilde{x}| - 2^{-\gamma} d \cdot \tilde{x}^\top \tilde{x}}{(2\gamma+1)\tilde{x}^\top \tilde{x}} \gtrsim \alpha,$$

where the last inequality is by $|\tilde{x}^\top B \tilde{x}|/\|\tilde{x}\|_2^2 \gtrsim |x^\top B x|/\|x\|_2^2 > \alpha(\log(d/\alpha)+1)$.

Tightness: Bilu and Linial [BL06] proved that Theorem 4.6 is tight that there are graphs satisfying the conditions but have spectral radius $\Omega(\alpha(\log(d/\alpha)+1)$.

4.3 Graphs with Large Spectral Gap

How large can the spectral gap be? Or, equivalently, how small can α be in an (n,d,α)-graph?

In this section, we present matching lower and upper bounds to this question, and discuss some strong properties of graphs with large spectral gap.

Lower Bounds

We begin with a simple proof that $\alpha \gtrsim \sqrt{d}$ using a trace argument.

Claim 4.8 (Easy Lower Bound for α)**.** *Let G be an (n,d,α)-graph. Then*

$$\alpha \geq \sqrt{d} \cdot \sqrt{\frac{n-d}{n-1}}.$$

Proof. Let A be the adjacency matrix of G with eigenvalues $\alpha_1 \geq \ldots \geq \alpha_n$. By Fact A.35,

$$\mathrm{Tr}(A^2) = \sum_{i=1}^n \alpha_i^2 \leq d^2 + (n-1)\alpha^2.$$

On the other hand, $\mathrm{Tr}(A^2) \geq nd$, as each edge uv contributes a length-two walk from u to u and a length-two walk from v to v. Combining the two inequalities establishes the claim. □

A higher-order trace argument can be used to prove a lower bound close to $2\sqrt{d-1}$.

Theorem 4.9 (Trace Lower Bound for α)**.** *Let G be an (n,d,α)-graph. Then*

$$\alpha \geq 2\sqrt{d-1} - o_n(1).$$

Proof. Let A be the adjacency matrix of G with eigenvalues $\alpha_1 \geq \ldots \geq \alpha_n$. For any $k \in \mathbb{N}$,

$$\mathrm{Tr}(A^{2k}) = \sum_{i=1}^n \alpha_i^{2k} \leq d^{2k} + (n-1)\alpha^{2k}.$$

On the other hand, recall from Lemma 1.5 that $\mathrm{Tr}(A^{2k})$ is equal to the number of length-$2k$ walks in G. For each vertex v, the number of length-$2k$ walks from v to v is at least the number of such walks in an infinite d-regular tree. In an infinite d-regular tree, the number of such walks is at least $C_k \cdot (d-1)^k$, where C_k is the k-th Catalan number. This is because each such walk has k forward

steps and k backward steps, where every prefix of a walk has at least as many forward steps as backward steps, and there are at least $d-1$ options for each forward step. Thus,

$$\mathrm{Tr}(A^{2k}) \geq n \cdot C_k \cdot (d-1)^k = \frac{n}{k+1}\binom{2k}{k}(d-1)^k.$$

Combining the inequalities and using an estimate of Catalan numbers,

$$\alpha^{2k} \geq \frac{1}{k+1}\binom{2k}{k}(d-1)^k - \frac{d^{2k}}{n} \geq \frac{4^k(d-1)^k}{2(k+1)^{3/2}} - \frac{d^{2k}}{n}.$$

Therefore, by choosing $k \ll \log n / \log d$ so that $d^{2k}/n \ll 1$, but letting k goes to infinity as n grows,

$$\alpha \geq 2\sqrt{d-1}\Big(1 - \frac{O(\log k)}{k}\Big) - o_n(1) \geq 2\sqrt{d-1} - o_n(1).$$

□

The following well-known result by Alon and Boppana provides a tight lower bound on the second eigenvalue of the adjacency matrix of a d-regular graph.

Theorem 4.10 (Alon-Boppana Bound [Nil91])**.** *Let $G = (V, E)$ be a d-regular graph and α_2 be the second largest eigenvalue of its adjacency matrix. Then*

$$\alpha_2 \geq 2\sqrt{d-1} - \frac{2\sqrt{d-1}-1}{\lfloor \mathrm{diag}(G)/2 \rfloor},$$

where $\mathrm{diag}(G)$ *denotes the diameter of the graph G.*

The theorem implies that if we have an infinite family of d-regular graphs each has second eigenvalue at most α_2, then $\alpha_2 \geq 2\sqrt{d-1}$ as the diameter grows to infinity with the graph size.

The proof is by constructing a vector $x \perp \vec{1}$ with Rayleigh quotient $x^\top Ax / x^\top x \approx 2\sqrt{d-1}$. The vector x is similar to the first eigenvector of a d-regular tree (see Problem 1.22). We will not prove Theorem 4.10 and refer readers to [HLW06, Section 5.2] or Trevisan's exposition [Tre08].

Upper Bounds

A major discovery is that graphs with $\alpha \leq 2\sqrt{d-1}$ exist.

Theorem 4.11 (Lubotzky, Phillip, Sarnak [LPS88], Margulis [Mar88])**.** *For every prime p and every positive integer k, there exist infinitely many (n, d, α)-graphs with $\alpha \leq 2\sqrt{d-1}$ and $d = p^k + 1$.*

The graphs constructed in [LPS88] are the Cayley graph of certain groups. The proof relies on some deep results in number theory, specifically on proven cases of conjectures by Ramanujan, which is well beyond the scope of this course. That is the reason why they called graphs with $\alpha \leq 2\sqrt{d-1}$ "Ramanujan graphs".

This naturally leads to the question of whether there are combinatorial or probabilistic constructions of Ramanujan graphs. The simplest probabilistic construction is to generate a random d-regular graph. A famous result by Friedman shows that most d-regular graphs are nearly-Ramanujan.

Theorem 4.12 (Friedman [Fri08]). *Let G be a random d-regular graph on n vertices and let $\alpha := \max\{\alpha_2, |\alpha_n|\}$ where α_2 and α_n are the second and last eigenvalues of the adjacency matrix of G. Then, for every $\epsilon > 0$,*

$$\Pr\left(\alpha \leq 2\sqrt{d-1} + \epsilon\right) = 1 - o_n(1).$$

It has been a long standing open question whether most d-regular graphs are Ramanujan. Recent progress has significantly advanced our understanding of this problem, including simpler proofs, new approaches, and sharper bounds. A remarkable new paper by Huang, McKenzie and Yau [HMY24] proves that a random d-regular graph is Ramanujan with a probability of approximately 69%. The proofs of these results are also well beyond the scope of this course.

It is still an open problem to design deterministic combinatorial constructions of Ramanujan graphs, although there are breakthroughs in such constructions for bipartite Ramanujan graphs [MSS15, MSS18] using the method of interlacing polynomials.

Properties

What additional properties do Ramanujan graphs possess that typical expander graphs do not? Typical d-regular expander graphs $G = ([n], E)$ satisfy the following properties:

1. Algebraic: $\lambda_2(\mathcal{L}(G)) \geq \Omega(1)$ where $\mathcal{L}$ is the normalized Laplacian matrix;
2. Combinatorial: $\phi(G) \geq \Omega(1)$ where $\phi(G)$ is the edge conductance of G;
3. Probabilistic: $\tau(G) \leq O(\log n)$ where $\tau(G)$ is the mixing time of lazy random walks on G.

Note that the mixing time bound is optimal when d is a constant as the graph has diameter $\Omega(\log n)$, but it does not improve even if we assume d is large.

Definition 4.13 (Graphs with Large Spectral Gap). *We say a (n, d, α)-graph G has a large spectral gap if $\alpha = O(d^{1-c})$ for some constant $0 < c \leq 1/2$, and say G is near-Ramanujan if $c = 1/2$.*

For near-Ramanujan graphs, the lower bound on edge conductance is $\frac{1}{2}(1 - O(\frac{1}{\sqrt{d}}))$. This is only slightly stronger than that of typical expander graphs, and does not quantify the additional expansion properties that near-Ramanujan graph possess. The right combinatorial parameter to measure the additional expansion properties of near-Ramanujan graphs is the small-set vertex expansion.

A d-regular graph $G = ([n], E)$ with a large spectral gap satisfies the following properties:

1. Algebraic: $\lambda_2(\mathcal{L}(G)) \geq 1 - O(d^{-c})$ where $\mathcal{L}$ is the normalized Laplacian matrix;
2. Combinatorial: $\psi(S) \gtrsim d^{2c}$ if $|S| \lesssim n/d^{2c}$, where $\psi(S)$ is the vertex expansion of a set S;
3. Probabilistic: $\tau(G) \lesssim \log n / \log d^c$, where $\tau(G)$ is the mixing time of random walks on G.

These are significant upgrades of the combinatorial and probabilistic properties over that of typical expander graphs: The second property implies that a near-Ramanujan graph has near-perfect small-set vertex expansion. The third property implies that a graph with a large spectral gap has constant mixing time when $d = n^\epsilon$ for some constant $\epsilon > 0$. We will explore these in the next sections.

4.4 Small-Set Vertex Expansion

For an (n, d, α)-graph, Tanner's theorem shows that sets of size up to $\alpha^2 n/d^2$ have vertex expansion d^2/α^2. Note that the bound becomes interesting when $\alpha \ll d$. In the extreme case of a near-Ramanujan graph, sets of size up to $\Omega(n/d)$ have near-perfect vertex expansion of $\Omega(d)$.

Theorem 4.14 (Tanner's Theorem)**.** *Let $G = (V, E)$ be an (n, d, α)-graph. For any $0 < \delta \leq 1/2$ and any subset $S \subseteq V$ with $|S| = \delta n$,*

$$\psi(S) \geq \left(\delta\Big(1 - \frac{\alpha^2}{d^2}\Big) + \frac{\alpha^2}{d^2}\right)^{-1} - 1.$$

Proof. The idea is to consider the quantity $\|A\chi_S\|_2^2$, where A is the adjacency matrix and χ_S is the characteristic vector of $S \subseteq V$. This quantity is bounded in two ways. One way is to show that if the closed vertex boundary $|\partial[S]|$ is small, then $\|A\chi_S\|_2^2$ is large, where $\partial[S] := \partial(S) \cup S$. The other way is to upper bound $\|A\chi_S\|_2^2$ using the spectral property, as $\|Ax\|_2^2 \leq \alpha^2\|x\|_2^2$ for $x \perp \vec{1}$.

For a vertex $v \in V$, let $\deg_S(v) := |\{u \in S \mid uv \in E\}|$ be the number of neighbors of v in S. Then,

$$\|A\chi_S\|_2^2 = \sum_{v \in V} \deg_S(v)^2 = \sum_{v \in \partial[S]} \deg_S(v)^2 \geq \frac{\big(\sum_{v \in \partial[S]} \deg_S(v)\big)^2}{|\partial[S]|} = \frac{\big(d|S|\big)^2}{|\partial[S]|},$$

where the inequality follows from Cauchy-Schwarz. This proves the lower bound.

For the upper bound, we write $\chi_S = \sum_{i=1}^n c_i v_i$ as a linear combination of the orthonormal eigenvectors of A, with $v_1 = \vec{1}/\sqrt{n}$ and $c_1 = \langle \chi_S, v_1 \rangle = |S|/\sqrt{n}$. Then

$$\|A\chi_S\|_2^2 = \Big\|\sum_{i=1}^n c_i \alpha_i v_i\Big\|_2^2 = \sum_{i=1}^n c_i^2 \alpha_i^2 \leq \frac{d^2|S|^2}{n} + \alpha^2\big(\|\chi_S\|^2 - c_1^2\big) = d^2\delta|S| + \alpha^2|S| - \alpha^2\delta|S|,$$

Combining the inequalities yields

$$\psi(S) + 1 = \frac{|\partial[S]|}{|S|} \geq \frac{d^2}{\delta(d^2 - \alpha^2) + \alpha^2} = \left(\delta\Big(1 - \frac{\alpha^2}{d^2}\Big) + \frac{\alpha^2}{d^2}\right)^{-1}.$$

□

For Ramanujan graphs, Tanner's theorem show that sets of size up to $n/(Cd)$ for a large constant C have vertex expansion close to $d/4$.

Kahale [Kah95] improved Tanner's theorem and showed that small linear-sized subsets in Ramanujan graphs have vertex expansion close to $d/2$. The same paper provided an example where $\alpha \leq 2\sqrt{d-1} + o(1)$, but the graph has a small set with vertex expansion at most $d/2$, proving that the $d/2$ bound is tight.

For some applications in constructing error correcting codes, graphs with small-set vertex expansion strictly greater than $d/2$ are required. We will revisit this question in the following chapters.

4.5 Random Walks on Expander Graphs

Mixing Time

We usually consider random walks on (n, d, α)-graphs when $d = \Theta(1)$. In this regime, as shown in Theorem 3.15, random walks converge to the uniform distribution in $O(\log n)$ steps as long as $\alpha \leq (1-\epsilon)d$ for some constant $\epsilon > 0$. This bound is optimal because a d-regular graph with $d = \Theta(1)$ has diameter $\Omega(\log n)$, so even if the graph is Ramanujan the mixing time remains $\Omega(\log n)$.

Now, consider the regime when $d = n^{\epsilon}$ for some constant $\epsilon > 0$, so that the diameter of the graph could be a constant. For typical expander graphs with $\alpha = \Theta(d)$, there still exist (n, d, α)-graphs that have mixing times of $\Omega(\log n)$. In contrast, for graphs with a large spectral gap, where $\alpha = O(d^{1-c})$ for some constant $c > 0$, every (n, d, α)-graph has constant mixing time.

The verification of these claims is left as an exercise in Problem 4.20. This demonstrates that graphs with large spectral gaps, as defined in Definition 4.13, exhibit significantly stronger randomness properties compared to typical expander graphs.

Concentration Property

Interestingly, random walks on expander graphs not only provide strong randomness properties for the final vertex in the walk, but also for the sequence of vertices traversed during the walk. In some applications, the sequence of vertices in a walk can effectively replace a sequence of independent uniform random variables.

The following result is not presented in its most general form, but will suffice for the application of probability amplification that we will see in Chapter 6. For more general statements, the reader is referred to [HLW06, Vad12]. To develop intuition, it is useful to compare the probability bound stated below with the corresponding bound when each X_i is an independent uniform random sample.

Theorem 4.15 (Concentration Property of Random Walks on Two-Sided Spectral Expanders)**.** *Let $G = (V, E)$ be a (n, d, α)-graph with $\alpha \leq d/10$. Let $B \subseteq V$ with $|B| \leq \frac{1}{100}|V|$. Let X_0 be a uniform random vertex, and let $X_1, \ldots, X_t$ be the vertices produced by t steps of a random walk. Let $S = \{i \mid X_i \in B\}$ be the set of times when the random walk is in B. Then,*

$$\Pr\left(|S| > \frac{t}{2}\right) \leq \left(\frac{2}{\sqrt{5}}\right)^{t+1}.$$

Proof. We first set up the matrix formulation of the problem. The initial distribution of X_0 is $\vec{p}_0 = \vec{1}/n$. Let I_B be the diagonal matrix with a 1 in the i-th diagonal entry if $i \in B$ and zero otherwise, and similarly define $I_{\overline{B}}$ for $\overline{B} = V - B$. For a probability vector $\vec{p}$, $I_B \cdot \vec{p}$ restricts $\vec{p}$ to B. The probability that the random walk is in B precisely at the times in S is

$$p_S := \vec{1}^{\top}(I_{Z_t}\mathcal{A})(I_{Z_{t-1}}\mathcal{A})(I_{Z_{t-2}}\mathcal{A})\ldots(I_{Z_2}\mathcal{A})(I_{Z_1}\mathcal{A})\vec{p}_0,$$

where $Z_i = B$ if $i \in S$ and $Z_i = \overline{B}$ otherwise, and $\mathcal{A}$ is the normalized adjacency matrix (the transition matrix of the random walk).

We will prove that $p_S \leq (\frac{1}{5})^{|S|}$. The theorem will then follow by a union bound as

$$\Pr\left(|S| > \frac{t}{2}\right) \leq \sum_{S:|S|>t/2} p_S \leq \sum_{S:|S|>t/2} \left(\frac{1}{5}\right)^{|S|} \leq \sum_{S:|S|>t/2} \left(\frac{1}{5}\right)^{\frac{t+1}{2}} \leq 2^{t+1}\left(\frac{1}{5}\right)^{\frac{t+1}{2}} = \left(\frac{2}{\sqrt{5}}\right)^{t+1}.$$

To prove $p_S \le (\frac{1}{5})^{|S|}$, we use the operator norm $\|\cdot\|_{\text{op}}$; see Definition A.19. Note that $\|I_B\|_{\text{op}} = \|I_{\overline{B}}\|_{\text{op}} = \|\mathcal{A}\|_{\text{op}} = 1$. We will prove $\|I_B\mathcal{A}\|_{\text{op}} \le \frac{1}{5}$, which implies $p_S \le (\frac{1}{5})^{|S|}$ as follows:

$$\begin{aligned}
p_S &= \vec{1}^\top (I_{Z_t}\mathcal{A}) \dots (I_{Z_1}\mathcal{A})\vec{p}_0 \\
&\le \|\vec{1}\|_2 \cdot \big\|(I_{Z_t}\mathcal{A}) \dots (I_{Z_1}\mathcal{A})\vec{p}_0\big\|_2 && \text{(Cauchy-Schwarz)} \\
&\le \|\vec{1}\|_2 \cdot \big\|(I_{Z_t}\mathcal{A}) \dots (I_{Z_1}\mathcal{A})\big\|_{\text{op}} \cdot \|\vec{p}_0\|_2 && \text{(operator norm in Definition A.19)} \\
&\le \|\vec{1}\|_2 \cdot \Big(\prod_{i=1}^{t} \|I_{Z_i}\mathcal{A}\|_{\text{op}}\Big) \cdot \|\vec{p}_0\|_2 && \text{(multiplicative property in Fact A.21)} \\
&\le \|\vec{1}\|_2 \cdot \Big(\frac{1}{5}\Big)^{|S|} \cdot \|\vec{p}_0\|_2 && \text{(from } \|I_B\mathcal{A}\|_{\text{op}} \le \frac{1}{5} \text{ and } \|I_{\overline{B}}\mathcal{A}\|_{\text{op}} \le 1) \\
&= \Big(\frac{1}{5}\Big)^{|S|}. && \text{(since } \|\vec{1}\|_2 = \sqrt{n} \text{ and } \|\vec{p}_0\|_2 = \frac{1}{\sqrt{n}})
\end{aligned}$$

It remains to prove that $\|I_B\mathcal{A}\|_{\text{op}} \le \frac{1}{5}$, which is equivalent to proving that $\|I_B\mathcal{A}x\|_2^2 \le \|x\|_2^2/25$ for any nonzero vector x. Write $x = c_1v_1 + \ldots + c_nv_n$, where $v_1, \ldots, v_n$ are the orthonormal eigenvectors of $\mathcal{A}$ with eigenvalues $1 = \alpha_1 \ge \ldots \ge \alpha_n \ge -1$. Since G is an (n, d, α)-graph, we have $\max_{2\le i\le n}\{|\alpha_i|\} \le \alpha/d$. It is then natural to decompose $\|I_B\mathcal{A}x\|_2^2$ into two terms:

$$\|I_B\mathcal{A}x\|_2^2 = \|I_B\mathcal{A}(c_1v_1 + \ldots + c_nv_n)\|_2^2 = \Big\|I_B\sum_{i=1}^{n} c_i\alpha_iv_i\Big\|_2^2 \le 2\|I_Bc_1\alpha_1v_1\|_2^2 + 2\Big\|I_B\sum_{i=2}^{n} c_i\alpha_iv_i\Big\|_2^2.$$

Using $c_1 = \langle x, v_1\rangle = \langle x, \frac{\vec{1}}{\sqrt{n}}\rangle = \frac{1}{\sqrt{n}} \cdot \sum_{i=1}^{n} x(i)$ and $|B| \le \frac{n}{100}$, the first term is

$$2\big\|I_Bc_1\alpha_1v_1\big\|_2^2 = 2\Big\|\frac{1}{n}\Big(\sum_{i=1}^{n} x(i)\Big)I_B\vec{1}\Big\|_2^2 = 2|B|\Big(\frac{\sum_{i=1}^{n} x(i)}{n}\Big)^2 \le 2|B| \cdot \frac{\|x\|_2^2}{n} \le \frac{1}{50}\|x\|_2^2,$$

where the first inequality is by Cauchy-Schwarz. Using $\alpha/d \le 1/10$, the second term is

$$2\Big\|I_B\sum_{i=2}^{n} c_i\alpha_iv_i\Big\|_2^2 \le 2\|I_B\|_{\text{op}}^2 \cdot \Big\|\sum_{i=2}^{n} c_i\alpha_iv_i\Big\|_2^2 = 2\sum_{i=2}^{n} c_i^2\alpha_i^2 \le 2\Big(\frac{\alpha}{d}\Big)^2\sum_{i=2}^{n} c_i^2 \le 2\Big(\frac{\alpha}{d}\Big)^2\|x\|_2^2 \le \frac{1}{50}\|x\|_2^2,$$

Adding the two terms finishes the proof. □

4.6 Problems

Problem 4.16 (Tighter Expander Mixing Lemma). *Let $G = (V, E)$ be an (n, d, α)-graph. Prove that, for every $S \subseteq V$ and $T \subseteq V$,*

$$\Big||E(S,T)| - \frac{d|S||T|}{n}\Big| \le \alpha\sqrt{|S||T|\Big(1 - \frac{|S|}{n}\Big)\Big(1 - \frac{|T|}{n}\Big)}.$$

Obtain the corresponding improvement in the non-regular case.

Exercise 4.17 (Zero Diagonal Entries). *Argue that if Lemma 4.7 is true for matrices with zero diagonal entries, then Lemma 4.7 is true.*

Problem 4.18 (Powers of Two). *Show that given any $x \in \mathbb{R}^n$ with $\|x\|_2 = 1$, there is a vector $\tilde{x} \in \mathbb{R}^n$ such that each entry of $\tilde{x}$ is a negative power of two and*

$$\frac{|\tilde{x}^\top B \tilde{x}|}{\|\tilde{x}\|_2^2} \geq \frac{1}{4} \cdot \frac{|x^\top B x|}{\|x\|_2^2}.$$

Hint: Design a rounding such that $\mathbb{E}[\tilde{x}(i)] = x(i)$, and use the assumption that the diagonal entries of B are zero to argue that $\mathbb{E}\left[\tilde{x}^\top B \tilde{x}\right] = x^\top B x$.

Problem 4.19 (Small-Set Vertex Expansion from Expander Mixing Lemma). *Use the expander mixing lemma in Theorem 4.3 to prove that an (n, d, α)-graph with a large spectral gap (as defined in Definition 4.13) has good small-set vertex expansion. Compare your bound to Tanner's bound in Theorem 4.14.*

Problem 4.20 (Mixing Time of Graphs with Large Spectral Gap). *Let G be an (n, d, α)-graph with $d = n^\epsilon$ for some constant $\epsilon > 0$.*

(a) Give an example with $\alpha = c \cdot d$ for some constant $0 < c < 1$ and mixing time $\Omega(\log n)$.

(b) Show that the mixing time is $O(1)$ when $\alpha = d^{1-c}$ for a constant $0 < c \leq 1/2$.

Problem 4.21 (Two-Sided Probability Bound of Expander Walks [AB06, Exercise 22.2]). *Let $G = (V, E)$ be an (n, d, α)-graph. Let $B \subseteq V$ with $|B| = \beta n$. Let X_0 be a uniform random vertex, and let $X_1, \ldots, X_t$ be the vertices produced by t steps of a random walk on G. Prove that*

$$(\beta - 2\alpha)^{t+1} \leq \Pr[X_i \in B \text{ for all } 0 \leq i \leq t] \leq (\beta + 2\alpha)^{t+1}.$$

References

[AB06] S. Arora and B. Barak. *Computational Complexity: A Modern Approach*. Cambridge University Press, 2006. 65, 73, 80

[AC88] Noga Alon and Fan R. K. Chung. Explicit construction of linear sized tolerant networks. *Discret. Math.*, 72(1-3):15–19, 1988. 54

[BL06] Yonatan Bilu and Nathan Linial. Lifts, discrepancy and nearly optimal spectral gap. *Comb.*, 26(5):495–519, 2006. 55, 59, 73, 74

[Fri08] Joel Friedman. *A Proof of Alon's Second Eigenvalue Conjecture and Related Problems*. American Mathematical Society, 2008. 61, 67

[HLW06] Shlomo Hoory, Nathan Linial, and Avi Wigderson. Expander graphs and their applications. *Bull. Amer. Math. Soc.*, 43(04):439–562, 2006. 7, 25, 26, 60, 63, 67, 68, 73, 77, 78, 80

[HMY24] Jiaoyang Huang, Theo Mckenzie, and Horng-Tzer Yau. Ramanujan property and edge universality of random regular graphs. *arXiv preprint arXiv:2412.20263*, 2024. 61, 67

[Kah95] Nabil Kahalé. Eigenvalues and expansion of regular graphs. *J. ACM*, 42(5):1091–1106, 1995. 62

[LPS88] A. Lubotzky, R. Phillips, and P. Sarnak. Ramanujan graphs. *Combinatorica*, 8(3):261–277, 1988. 60, 68

[Mar88] Gregory A. Margulis. Explicit group-theoretic constructions of combinatorial schemes and their applications in the construction of expanders and concentrators. *Problemy Peredachi Informatsii*, 24(1):51–60, 1988. 60, 68

[MSS15] Adam W Marcus, Daniel A Spielman, and Nikhil Srivastava. Interlacing families i: Bipartite ramanujan graphs of all degrees. *Annals of Mathematics*, 182:307–325, 2015. 61, 73

[MSS18] Adam W. Marcus, Daniel A. Spielman, and Nikhil Srivastava. Interlacing families IV: bipartite ramanujan graphs of all sizes. *SIAM J. Comput.*, 47(6):2488–2509, 2018. 61

[Nil91] Alon Nilli. On the second eigenvalue of a graph. *Discret. Math.*, 91(2):207–210, 1991. 60

[Tre08] Luca Trevisan. The Alon-Boppana theorem. *Blog post: https://lucatrevisan.wordpress.com/2014/09/01/the-alon-boppana-theorem-2/*, 2008. 60

[Vad12] Salil P. Vadhan. Pseudorandomness. *Found. Trends Theor. Comput. Sci.*, 7(1-3):1–336, 2012. 63, 78, 80

Chapter 5

Expander Graphs: Constructions

In this chapter, we explore various constructions of expander graphs. We begin with a discussion of probabilistic and algebraic constructions, followed by an analysis of a combinatorial construction known as the zig-zag product, along with an overview of other combinatorial constructions. The content of this chapter is mostly based on [HLW06].

5.1 Probabilistic Constructions

Constructing expander graphs is generally considered a challenging task. Ironically, almost all graphs are expander graphs and even near-Ramanujan graphs!

For combinatorial properties, the probabilistic method can be used to prove that a random d-regular graph has constant edge conductance with probability tending to 1 as the graph size goes to infinity. Furthermore, it can be proved the vertex expansion of linear-sized subsets is close to $d-2$ [HLW06, Section 4.6], which goes beyond the $d/2$ bound achieved by Ramanujan graphs.

For spectral properties, as discussed in Section 4.3, Friedman [Fri08] proved that almost all d-regular graphs are near-Ramanujan. Moreover, the recent work by Huang, McKenzie, and Yau [HMY24] proved that a random d-regular graph is Ramanujan with a probability of approximately 69%.

For applications such as using expander graphs for designing randomized algorithms, these randomized constructions can be applied directly.

Generating Random d-Regular Graphs

A technical question is how to generate a random d-regular graph.

The configuration model is a commonly used method for constructing random graphs with a specified degree sequence. It works by creating "half-edges" for each vertex according to its degree and then randomly pairing these half-edges to form edges. The resulting graph may contain self-loops or multi-edges, but if we condition on simple graphs, the distribution becomes uniform.

In the permutation model, a $2d$-regular graph on n vertices is generated by independently choosing d random permutations $\pi_1, \ldots, \pi_d$ on $[n]$ and adding an edge $(v, \pi_i(v))$ for each $v \in [n]$ and $i \in [d]$. This does not yield the uniform distribution on $2d$-regular graphs, but it is known that a family of events has probability $1 - o(1)$ in the distribution induced by the permutation model if and only if it has probability $1 - o(1)$ in the uniform distribution [HLW06, JLR11].

For fast algorithms for generating random regular graphs, we refer the reader to [GW17].

5.2 Algebraic Constructions

For applications such as derandomization, the probabilistic constructions cannot be used. Moreover, for some applications, we cannot afford to generate the whole graph and need to explore the graph locally. Algebraic constructions are deterministic and are explicit, allowing efficient computation of neighbors for any vertex.

The first construction of expanders is due to Margulis, a family of 8-regular graphs G_m for every integer m. The vertex set is $V_m = \mathbb{Z}_m \times \mathbb{Z}_m$. The neighbors of the vertex (x, y) are $(x+y, y), (x-y, y), (x, y+x), (x, y-x), (x+y+1, y), (x-y+1, y), (x, y+x+1), (x, y-x+1)$, where all operations are modulo m. We recommend the proof by Lee and Trevisan [Tre17, Chapter 19].

Another interesting construction is a family of 3-regular p-vertex graphs for every prime p. The vertex set is $\mathbb{Z}_p$, and a vertex x is connected to $x+1$, $x-1$ and to its multiplicative inverse x^{-1} modulo p. The proof relies on a deep result in number theory.

The constructions of Ramanujan graphs by Lubotzky, Phillip, Sarnak [LPS88] and Margulis [Mar88] are Cayley graphs of specific groups. We refer the reader to [HLW06] for the construction and an exposition of Cayley expander graphs.

5.3 Combinatorial Constructions

Developing more elementary constructions and analysis of expander graphs remains an area of great interest. The general approach in combinatorial constructions is to build larger expander graphs from smaller ones. It turns out that this approach has significant applications in algorithms and complexity, as we will see in the next chapter.

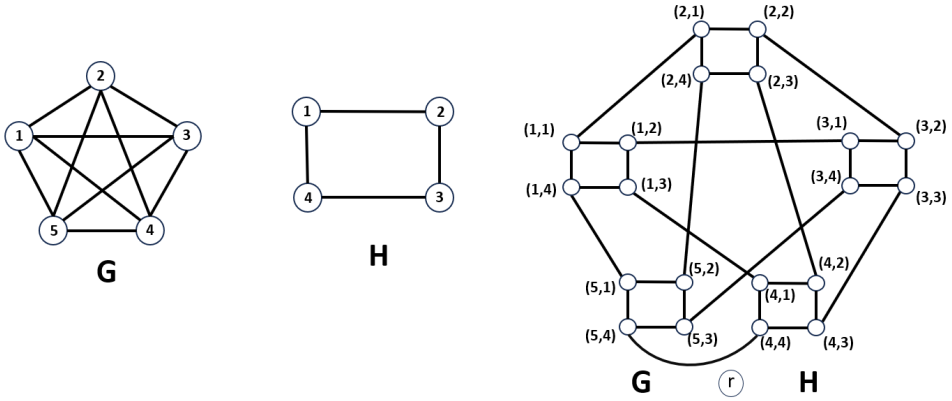
Replacement Products

The replacement product is perhaps the most natural product to begin with. The base case could simply be a constant size complete graph. Let G be an $(n, k, \epsilon_1 k)$-graph and H be an $(k, d, \epsilon_2 d)$-graph. A replacement product of G and H is to replace each vertex v in G with a copy of H, so that each edge incident on v connects to a different vertex of H.

Definition 5.1 (Replacement Product)**.** *Let G be a k-regular graph on n vertices and H be a d-regular graph on k vertices. The replacement product $G \textcircled{r} H$ is a graph where the vertex set is the Cartesian product $[n] \times [k]$ of the vertex set of G and H. Two vertices (u, i) and (v, j) have an edge if and only if:*

1. *$u = v$ and $ij \in E(H)$, or*
2. *$vu \in E(G)$, v is the i-th neighbor of u in G, and u is the j-th neighbor of v in G.*

Intuitively, $G \textcircled{r} H$ is a combinatorial expander if G and H are combinatorial expanders. Consider a set $S \subseteq V(G \textcircled{r} H)$. If S has either a large of small intersection with each "cloud" (copy of H), then S should have large expansion due to the large expansion of G, as S essentially corresponds to a subset of vertices in G. If S has medium intersections with many clouds, then S should have large expansion because H has large expansion, and there are many crossing edges within each cloud. See Problem 5.12 for a combinatorial analysis that makes this intuition precise.

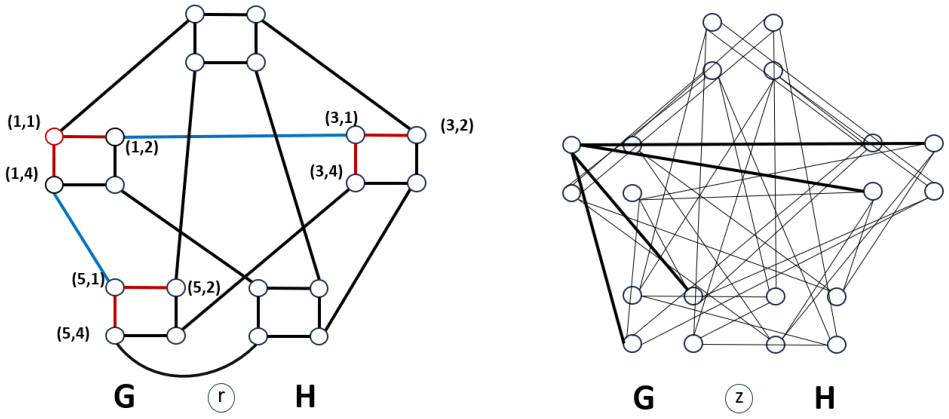


The spectral approach, which we will see shortly, can be thought of as a linear algebraic method to formalize this idea efficiently in a more general setting.

Zig-Zag Product

The actual construction by Reingold, Vadhan and Wigderson [RVW02] that we will analyze is slightly more complex.

Definition 5.2 (Zig-Zag Product). *Let G be a k-regular graph on n vertices and H be a d-regular graph on k vertices. The zig-zag product $G \textcircled{z} H$ is a graph with the same vertex set $[n] \times [k]$ as the replacement product. Two vertices (u, i) and (v, j) are connected by an edge if and only if $u \neq v$ and there exists $a \in [k]$ such that (u, i)–(u, a), (u, a)–(v, b), and (v, b)–(v, j) are all edges in the replacement product $G \textcircled{r} H$, where (u, a)–(v, b) is the unique edge incident on (u, a) with $v \neq u$ (i.e., the unique edge incident on (u, a) that leaves the cloud of u in the replacement product).*



In words, each edge in the zig-zag product $G \textcircled{z} H$ corresponds to a length three walk in the replacement product $G \textcircled{r} H$, where the first step is within a cloud, the second step is the unique way to leave a cloud, and the third step is within the other cloud.

The intuition that the zig-zag product is a spectral expander comes from random walks. Each edge in $G \textcircled{z} H$ corresponds to a random step in H, a deterministic step in G, and another random step

in H. We should think of the first two steps as going to a random neighboring cloud, and the third step corresponds to moving to a random neighbor within that cloud. Since both G and H are spectral expanders with fast mixing properties, after a few steps, the walk loses information about both the cloud and the location within the cloud. Thus, $G \textcircled{z} H$ inherits the fast mixing property and is a spectral expander.

Theorem 5.3 (Zig-Zag Theorem [RVW02])**.** *Let G be an $(n, k, \epsilon_1 k)$-graph and H be an $(k, d, \epsilon_2 d)$-graph. Then $G \textcircled{z} H$ is an $(nk, d^2, (\epsilon_1 + \epsilon_2 + \epsilon_2^2)d^2)$-graph.*

We will prove the theorem in the next subsection. First, let us see how the zig-zag product can be used to construct larger and larger constant degree expander graphs. The idea is to combine with the following standard operation that increases the spectral gap.

Definition 5.4 (Graph Power)**.** *Let G be a graph with adjacency matrix A. The k-th power G^k is the graph with the same vertex set as G and adjacency matrix A^k.*

In words, the number of parallel edges between u and v in G^k is equal to the number of length k walks between u and v in G. Note that while the spectral gap of G^k improves significantly, its degree also increases significantly.

Exercise 5.5 (Spectral Gap of Power)**.** *G^k is an $(n, d^k, \epsilon^k d^k)$-graph if G is an $(n, d, \epsilon d)$-graph.*

The idea of the combinatorial construction is to use graph power to increase the spectral gap, and then use the zig-zag product to reducing the degree without significantly reducing the spectral gap.

Theorem 5.6 (Expanders from Zig-Zag Product)**.** *For every sufficiently large constant d, there exists an infinite family of $(n, d^2, d^2/4)$-graphs.*

Proof. Let H be a $(d^4, d, d/16)$-graph. Its existence can be shown using a probabilistic argument when d is sufficiently, and it can be found via exhaustive search in constant time.

Using H as the building block, we define G_i inductively by $G_1 = H^2$ and $G_{i+1} = G_i^2 \textcircled{z} H$. We claim that G_i is a $(d^{4i}, d^2, d^2/4)$-graph for all $i \geq 1$. The base case follows from Exercise 5.5. Assuming G_i is a $(d^{4i}, d^2, d^2/4)$-graph, then G_i^2 is a $(d^{4i}, d^4, d^4/16)$-graph by Exercise 5.5, and $G_i^2 \textcircled{z} H$ is a $(d^{4(i+1)}, d^2, d^2/4)$-graph by Theorem 5.3. □

Proof of the Zig-Zag Theorem

Check that $G \textcircled{z} H$ has nk vertices and is d^2-regular. We bound the spectral gap of $G \textcircled{z} H$.

Matrix Formulation: The first step is to write down the walk matrix Z of the zig-zag product $G \textcircled{z} H$. Let $W(H)$ be the $k \times k$ walk matrix of H, which is simply $\frac{1}{d}A(H)$ where $A(H)$ is the adjacency matrix of H. Let W be the $nk \times nk$ matrix with n copies of W_H on the diagonal. This represents the transition matrix for one step of the random walk within the clouds in $G \textcircled{z} H$. The steps between clouds are deterministic: the walk moves from a vertex (u, i) to a unique vertex (v, j) with $v \neq u$. The transition matrix for this deterministic step is thus a permutation matrix P where $P_{(u,i),(v,j)} = 1$ for each inter-cloud edge and zero otherwise. It follows from the definition of the zig-zag product that

$$Z = WPW.$$

Thus, the random walk matrix of $G ⓩ H$ has a very clean form, which should be the reason for the definition of the zig-zag product in Definition 5.2.

The graph $G ⓩ H$ is regular, so $\vec{1}_{nk}$ is an eigenvector of Z with eigenvalue 1. To prove the zig-zag product theorem, we will prove that for all $f \perp \vec{1}_{nk}$, the Rayleigh quotient satisfies

$$R_Z(f) = \frac{|f^T Z f|}{\|f\|_2^2} \leq \epsilon_1 + \epsilon_2 + \epsilon_2^2.$$

This implies that all but the largest eigenvalue of Z have absolute value at most $\epsilon_1 + \epsilon_2 + \epsilon_2^2$, and hence all but the largest eigenvalue of $A(G ⓩ H)$ have absolute value at most $(\epsilon_1 + \epsilon_2 + \epsilon_2^2)d^2$.

Vector Decomposition: For any $f \perp \vec{1}_{nk}$, we decompose f into two vectors to apply the results in G and H. This step demonstrates the power of linear algebra, as in the larger domain $\mathbb{R}^{nk}$, there is a natural way to decompose a vector. In contrast, in the combinatorial setting, it is unclear how to decompose a set of vertices in $G ⓩ H$ into subsets of G and H to utilize their expansion properties as previously discussed.

Define f_G as the average of f on clouds, such that $f_G(u,i) = \frac{1}{k}\sum_{j=1}^{k} f(u,j)$ for all $(u,i) \in V(G ⓩ H)$, so that two vertices in the same cloud have the same value in f_G. Define $f_H = f - f_G$. Note that f_H sums to zero in each cloud, such that $\sum_{j=1}^{k} f_H(u,j) = 0$ for each $u \in G$. Using the triangle inequality,

$$|f^\top Z f| = |f^\top W P W f| = |(f_G + f_H)^\top W P W (f_G + f_H)| \leq |f_G^\top W P W f_G| + 2|f_G^\top W P W f_H| + |f_H^\top W P W f_H|.$$

Since $W(H) \cdot \vec{1}_k = \vec{1}_k$ as H is regular, it follows that $W f_G = f_G$, as vertices in the same cloud have the same value in f_G. Thus,

$$|f^\top Z f| \leq |f_G^\top P f_G| + 2|f_G^\top P W f_H| + |f_H^\top W P W f_H|.$$

We will use the spectral properties of G and H to establish:

- $|f_G^\top P f_G| \leq \epsilon_1 \|f_G\|_2^2$ (Claim 5.9),
- $|f_H^\top W P W f_H| \leq \epsilon_2^2 \|f_H\|_2^2$ (Claim 5.7),
- $2|f_G^\top P W f_H| \leq 2\epsilon_2 \|f_G\|_2 \|f_H\|_2$ (Claim 5.8).

Assuming these claims and using $\|f\|_2^2 = \|f_G\|_2^2 + \|f_H\|_2^2$ we conclude that

$$\begin{aligned} |f^\top Z f| &\leq \epsilon_1 \|f_G\|_2^2 + 2\epsilon_2 \|f_G\|_2 \|f_H\|_2 + \epsilon_2^2 \|f_H\|_2^2 \\ &\leq \epsilon_1 \|f_G\|_2^2 + \epsilon_2 \left(\|f_G\|_2^2 + \|f_H\|_2^2\right) + \epsilon_2^2 \|f_H\|_2^2 \\ &\leq (\epsilon_1 + \epsilon_2 + \epsilon_2^2)\|f\|_2^2. \end{aligned}$$

This completes the proof of Theorem 5.3, leaving the three claims to be proven.

Spectral Bounds: The following claim uses the spectral property of H and the fact that f_H sums to zero in each cloud.

Claim 5.7 (Quadratic Term of H). $|f_H^\top W P W f_H| \leq \epsilon_2^2 \|f_H\|_2^2$.

Proof. First, we claim that $\|W(H)\cdot x\|_2 \le \epsilon_2\|x\|_2$ for any $x \perp \vec{1}_k$. To see this, let $x = \sum_{i=1}^k c_i v_i$, where $v_1,\dots,v_n$ is an orthonormal basis of eigenvectors of $W(H)$ with eigenvalues $\alpha_1,\dots,\alpha_k$. Note that $c_1 = 0$, as $v_1 = \vec{1}/\sqrt{k}$ and $x \perp \vec{1}$. Since H is an $(k,d,\epsilon_2 d)$-graph, $\alpha_i^2 \le \epsilon_2^2$ for $2 \le i \le d$. Thus,

$$\|W(H)\cdot x\|_2^2 = \left\|W(H)\cdot\left(\sum_{i=2}^k c_i v_i\right)\right\|_2^2 = \left\|\sum_{i=2}^k c_i\alpha_i v_i\right\|_2^2 = \sum_{i=2}^k c_i^2\alpha_i^2 \le \epsilon_2^2\sum_{i=2}^k c_i^2 \le \epsilon_2^2\|x\|_2^2.$$

This implies that $\|Wf_H\|_2 \le \epsilon_2\|f_H\|$, as the sum of the entries in each cloud is zero in f_H. Therefore,

$$|f_H^\top WPWf_H| \le \|Wf_H\|_2\cdot\|PWf_H\|_2 = \|Wf_H\|_2^2 \le \epsilon_2^2\|f_H\|_2^2,$$

where the inequality is by Cauchy-Schwarz and the equality holds as P is a permutation matrix. □

The second claim is straightforward.

Claim 5.8 (Cross Term)**.** $|f_G^\top PWf_H| \le \epsilon_2\|f_G\|_2\|f_H\|_2$.

Proof. By Cauchy-Schwarz and $\|Wf_H\|_2 \le \epsilon_2\|f_H\|_2$ established in Claim 5.7,

$$|f_G^\top PWf_H| \le \|f_G\|_2\cdot\|PWf_H\|_2 = \|f_G\|_2\cdot\|Wf_H\|_2 \le \epsilon_2\|f_G\|_2\|f_H\|_2.$$

□

The final claim uses the spectral property of G and the fact that $f \perp \vec{1}_{nk}$.

Claim 5.9 (Quadratic Term of G)**.** $|f_G^\top Pf_G| \le \epsilon_1\|f_G\|_2^2$.

Proof. The main point is that $f_G^\top Pf_G$ is equal to a corresponding quadratic form of the walk matrix of G. To see this, we "contract" each cloud to a single vertex. Define $g : V(G) \to \mathbb{R}$ as $g(v) = \sqrt{k}\cdot f_G(v,1)$, so that $\|g\|_2^2 = \|f_G\|_2^2$. We claim that $f_G^\top Pf_G = g^\top W(G)g$, where $W(G)$ is the random walk matrix of G. This is because each edge (u,i)-(v,j) in $G\textcircled{z}H$ contributes $f_G(u,i)\cdot f_G(v,j)$ to $f_G^\top Pf_G$, while the corresponding edge $uv \in G$ contributes $g(u)\cdot W(G)_{u,v}\cdot g(v) = \left(\sqrt{k}f_G(u,1)\right)\left(\frac{1}{k}\right)\left(\sqrt{k}f_G(v,1)\right) = f_G(u,i)\cdot f_G(v,j)$ to $g^\top W(G)g$.

Since $f \perp \vec{1}$, it follows that $f_G \perp \vec{1}$ and thus $g \perp \vec{1}$. Therefore,

$$\frac{f_G^\top Pf_G}{\|f_G\|_2^2} = \frac{g^\top W(G)g}{\|g\|^2} \le \epsilon_1,$$

where the inequality is because G is an $(n,k,\epsilon_1 k)$-graph, □

This concludes the proof of Theorem 5.3. The idea of decomposing a vector into different components is useful in many proofs. We will use it again when we study high dimensional expanders.

Remark 5.10. *It has been an open question whether a variant of the zig-zag product can be used to construct (near) Ramanuajan graphs. See [CCM24] for a negative answer to this question for the zig-zag product in Definition 5.2.*

Lifts of Graphs

Another combinatorial approach to constructing expanders, first proposed by Friedman, is to "lift" a smaller Ramanujan graph into a larger one. A k-lift of an (n, d, α)-graph G is an (nk, d, α')-graph, where each vertex u of G is replaced by k vertices $u_1, \ldots, u_k$ in H, and each edge $uv \in G$ is replaced by a perfect matching between $u_1, \ldots, u_k$ and $v_1, \ldots, v_k$ in H.

There are several results on random k-lifts. The best known result is by Puder [Pud15] who proved that a random k-lift of a Ramanujan graph satisfies $\alpha' \leq 2\sqrt{d-1}+1$ with probability tending to 1 as $k \to \infty$.

Bilu and Linial [BL06] studied 2-lifts. They used the converse of expander mixing lemma in Theorem 4.6 and a derandomized probabilistic method to construct an infinite family of $\left(n, d, O(\sqrt{d \log^3 d})\right)$-graphs.

Marcus, Spielman, and Srivastava [MSS15] developed an original approach using interlacing polynomials to prove that every bipartite Ramanujan graph has a 2-lift which is bipartite Ramanujan (i.e., with $\max_{2 \leq i \leq n-1} |\alpha_i| \leq 2\sqrt{d-1}$).

Lossless Expanders

A graph is called a lossless expander if every small linear-sized subsets has vertex expansion close to d. These graphs have applications in constructing error correcting codes, as we will see in the next chapter.

As discussed previously, a random d-regular graph has small-set vertex expansion close to $d-2$, while there are Ramanujan graphs with small-set vertex expansion at most $d/2$. It remains a major open problem to design deterministic constructions of lossless expanders.

Capalbo, Reingold, Vadhan, and Wigerson [CRVW02] constructed one-sided bipartite lossless expanders using an intricate zig-zag product on "conductors" (a randomness-enchancing object); see also [HLW06, Chapter 10].

Recently, Golowich [Gol24] presented a simple and improved constructions of one-sided bipartite lossless expanders, by composing a small lossless expander with a large two-sided bipartite spectral expander. A new paper [HLM+24] demonstrated an explicit construction of two-sided vertex expanders beyond the spectral barrier $d/2$.

5.4 Problems

Problem 5.11 (Balanced Replacement Product [AB06, Lemma 21.18]). *The balanced replacement product is defined exactly the same as the replacement prodcut in Definition 5.1, except that we add d parallel edges for each edge of type (2) rather than a single edge as in Definition 5.1.*

Prove that if G is an $(n, k, 1-\epsilon)$-graph and H is an $(k, d, 1-\delta)$-graph, then the balanced replacement product of G and H is an $(nk, 2d, 1-\epsilon\delta^2/24)$-graph.

Problem 5.12 (Combinatorial Proof of Balanced Replacement Product [AB06, Exercise 21.24]). *Provide a combinatorial proof that if G is a k-regular graph of n vertices with edge conductance at least ϵ and H is a d-regular graph of k vertices with edge conductance at least δ, then the balanced replacement product of G and H has edge conductance at least $\epsilon^2\delta/1000$.*

Hint: Consider a subset S in the balanced replacement product of G and H and its intersection with each cloud. Treat differently the clouds that have a $(1-\epsilon/10)$-fraction of vertices within S and those that do not. For the former case, use the edge conductance of G, while for the latter, use the edge conductance of H to lower bound the edge conductance of S.

Problem 5.13 (Spectrum of 2-Lifts [BL06])**.** *Let $G = ([n], E)$ be an undirected graph. A signing of the edges of G is a function $s : E(G) \to \{-1, +1\}$. The 2-lift $\hat{G}_s = (\hat{V}, \hat{E})$ of G associated with a signing s is defined as follows. The vertex set $\hat{V}$ of $\hat{G}_s$ is $\hat{V} = \{1, \ldots, n, 1', \ldots, n'\}$, where each vertex $i \in V(G)$ has two copies i and i' in $\hat{G}_s$. For each edge $ij \in E(G)$, if $s(ij) = 1$, then the edges ij and $i'j'$ are in $\hat{E}$; otherwise, if $s(ij) = -1$, then the edges ij' and $i'j$ are in $\hat{E}$. The signed matrix $A_s \in \mathbb{R}^{n \times n}$ is defined as follows. If $ij \in E$, then $(A_s)_{ij} = (A_s)_{ji} = s(ij)$, otherwise $(A_s)_{ij} = 0$.*

Prove that the spectrum of the adjacency matrix $A(\hat{G}_s)$ of the 2-lift $\hat{G}_s$ is equal to the disjoint union of the spectrum of the adjacency matrix $A(G)$ of G (called the old eigenvalues) and the spectrum of the signed matrix A_s (called the new eigenvalues). That is, the multiplicity of an eigenvalue α of $A(\hat{G}_s)$ is equal to the sum of the multiplicity of α in $A(G)$ and the multiplicity of α in A_s.

References

[AB06] S. Arora and B. Barak. *Computational Complexity: A Modern Approach*. Cambridge University Press, 2006. 65, 73, 80

[BL06] Yonatan Bilu and Nathan Linial. Lifts, discrepancy and nearly optimal spectral gap. *Comb.*, 26(5):495–519, 2006. 55, 59, 73, 74

[CCM24] Gil Cohen, Itay Cohen, and Gal Maor. Tight bounds for the zig-zag product. In *65th IEEE Annual Symposium on Foundations of Computer Science, FOCS*, pages 1470–1499. IEEE, 2024. 72

[CRVW02] Michael Capalbo, Omer Reingold, Salil Vadhan, and Avi Wigderson. Randomness conductors and constant-degree lossless expanders. In *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, pages 659–668, 2002. 73, 82

[Fri08] Joel Friedman. *A Proof of Alon's Second Eigenvalue Conjecture and Related Problems*. American Mathematical Society, 2008. 61, 67

[Gol24] Louis Golowich. New explicit constant-degree lossless expanders. In *Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 4963–4971. SIAM, 2024. 73

[GW17] Pu Gao and Nicholas Wormald. Uniform generation of random regular graphs. *SIAM Journal on Computing*, 46(4):1395–1427, 2017. 68

[HLM+24] Jun-Ting Hsieh, Ting-Chun Lin, Sidhanth Mohanty, Ryan O'Donnell, and Rachel Yun Zhang. Explicit two-sided vertex expanders beyond the spectral barrier. *arXiv preprint arXiv:2411.11627*, 2024. 73

[HLW06] Shlomo Hoory, Nathan Linial, and Avi Wigderson. Expander graphs and their applications. *Bull. Amer. Math. Soc.*, 43(04):439–562, 2006. 7, 25, 26, 60, 63, 67, 68, 73, 77, 78, 80

[HMY24] Jiaoyang Huang, Theo Mckenzie, and Horng-Tzer Yau. Ramanujan property and edge universality of random regular graphs. *arXiv preprint arXiv:2412.20263*, 2024. 61, 67

[JLR11] Svante Janson, Tomasz Luczak, and Andrzej Rucinski. *Random graphs*. John Wiley & Sons, 2011. 67

[LPS88] A. Lubotzky, R. Phillips, and P. Sarnak. Ramanujan graphs. *Combinatorica*, 8(3):261–277, 1988. 60, 68

[Mar88] Gregory A. Margulis. Explicit group-theoretic constructions of combinatorial schemes and their applications in the construction of expanders and concentrators. *Problemy Peredachi Informatsii*, 24(1):51–60, 1988. 60, 68

[MSS15] Adam W Marcus, Daniel A Spielman, and Nikhil Srivastava. Interlacing families i: Bipartite ramanujan graphs of all degrees. *Annals of Mathematics*, 182:307–325, 2015. 61, 73

[Pud15] Doron Puder. Expansion of random graphs: New proofs, new results. *Inventiones mathematicae*, 201(3):845–908, 2015. 73

[RVW02] Omer Reingold, Salil Vadhan, and Avi Wigderson. *Annals of Mathematics*, 155(1):pp. 157–187, 2002. 69, 70

[Tre17] Luca Trevisan. *Lecture Notes on Graph Partitioning, Expanders and Spectral Methods*. 2017. 15, 68, 366

Chapter 6

Expander Graphs: Applications

This chapter highlights some key applications of expander graphs. More details are given on expander codes, which form the basis of recent breakthroughs in designing asymptotically good and locally testable codes [DEL+22, PK22]. The content of this chapter is mostly based on [HLW06].

6.1 Pseudorandomness

Expander graphs are pseudorandom objects, as suggested by the expander mixing lemma in Theorem 4.3. This pseudorandom property makes them useful for reducing or even eliminating randomness in certain settings.

Reducing Randomness in Probability Amplification

Suppose we have a randomized algorithm with error probability $1/100$ that requires n random bits. To decrease the failure probability, the standard approach is to run the randomized algorithm independently k times and take the majority answer as the output. By a standard Chernoff bound argument, this decreases the failure probability to δ^k for some small constant δ. However, this approach uses kn random bits.

We show how to achieve exponentially small error probability while using only $n + O(k)$ bits. To do so, let us reinterpret the above analysis from a random walk perspective.

Let V denote the set of all n-bit strings. The condition that the randomized algorithm has an error probability of at most $1/100$ is equivalent to saying that, among the 2^n n-bit strings, at most $2^n/100$ of them are "bad" strings. Let $B \subseteq V$ denote this set of bad strings. The majority-vote algorithm fails if more than $k/2$ strings sampled randomly are from B. Sampling k independent n-bit strings can be interpreted as performing a random walk of length k on the complete graph over V and use the corresponding bit strings of the vertices on this walk.

The key idea is to replace the random walk on the complete graph with a random walk on a constant degree expander graph over V. Let G be an $(2^n, d, d/10)$-graph, where d is a constant. Such graphs exist, for example, by taking a sufficiently large constant power of a Margulis expander as described in Section 5.2. In the first step of the random walk, we use an n-bit random string. In subsequent steps, instead of using n random bits to select the next n-bit string, we choose a random neighbor of the current string in G and use the corresponding string of this random neighbor. Since G is a d-regular graph, each subsequent step requires only $\log_2 d$ random bits to select a random neighbor.

Thus, the total number of bits used is $n+(k-1)\cdot\log_2 d = n+O(k)$. Importantly, the neighbors of a Margulis expander can be computed efficiently, allowing the corresponding strings to be computed quickly without constructing the entire graph G. This is where the explicitness of an algebraic construction becomes useful.

What about the error probability of this expander-walk algorithm? This is precisely what Theorem 4.15 is formulated for. It shows that the failure probability of taking the majority answer from a random walk of length k on a two-sided spectral expander with $\alpha \leq d/100$ is at most $(2/\sqrt{5})^k$.

Expander Random Walks

This is just one example of using expander graphs in derandomization; see [Vad12] for more. See [Gil98] for a well-known Chernoff bound for random walks on two-sided spectral expanders and [GLSS18] for a recent generalization to the matrix setting.

The above results can be interpreted as showing that the majority functions are fooled by expander random walks, meaning they cannot distinguish independent random samples from those produced by expander random walks. See [CPT21] for a recent paper demonstrating that many functions are similarly fooled by expander random walks.

6.2 Constructing Efficient Networks and Algorithms

A d-regular expander graph can be viewed as highly efficient, having only a linear number of edges while achieving very high connectivity. From this perspective, it is natural that expander graphs are used in constructing efficient networks.

Efficient Networks

One interesting example is the construction of superconcentrators, which are directed graphs with n input nodes and n output nodes (and possibly other notes) that satisfy the strong connectivity property: for any $k \leq n$, there exist k vertex disjoint paths between any k input nodes and any k output nodes. For example, the complete bipartite graph $K_{n,n}$ satisfies this property but requires $\Theta(n^2)$ edges. Valiant initially conjectured that no superconcentrator with $O(n)$ edges exists as part of an attempt to prove circuit lower bounds. Later, he developed a recursive construction of superconcentrator with $O(n)$ edges using expander graphs as building blocks; see [HLW06].

Another classical application of expander graphs is the construction of optimal sorting networks with $O(n \log n)$ edges and $O(\log n)$ depth [AKS83].

Efficient Algorithms

Superconcentrators and expander graphs can also be used to design efficient algorithms. A simple example is the design of fast algorithms for computing matrix rank [CKL13]. In this application, an expander graph or superconcentrator is used to "compress" a rectangular matrix $A \in \mathbb{F}^{m\times n}$ with $n \gg m$ into a square matrix $B \in \mathbb{F}^{m\times m}$ in linear time, such that $\text{rank}(A) = \text{rank}(B)$ with high probability. This leads to faster randomized algorithm for computing the rank of a rectangular matrix and finding linearly independent columns. For this application, probabilistic constructions of bipartite expander graphs are sufficient.

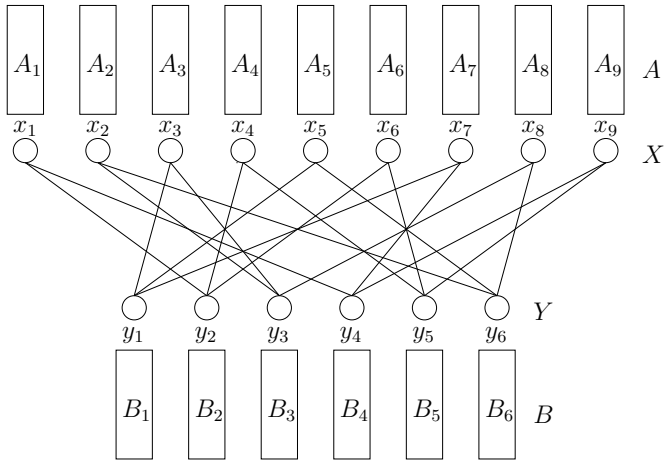


Figure 6.1: A bipartite expander graph $G = (X, Y; E)$ is used to compress the input matrix A into a smaller matrix B. Each column of B is a random linear combination of the columns of its neighbors; for example, B_3 is a random linear combination of A_2, A_3 and A_8.

Another example is to use superconcentrators to design a faster algorithm for computing edge connectivities in a directed acyclic graph [CLL13].

6.3 Complexity Theory

The construction of expanders using zig-zag products has inspired remarkable developments in complexity theory. In an interesting parallel, the construction of expanders using cut-matching games has been applied widely to designing efficient algorithms, as we will see later in this course. These results demonstrate the value of combinatorial constructions of expander graphs.

Graph Connectivity in Log-Space

A striking application of the zig-zag product in Definition 5.2 is solving the s-t connectivity problem in an undirected graph using logarithmic space.

If randomized algorithms are allowed, solving s-t connectivity in log-space is simple: just perform a random walk for $O(n^3)$ steps. This works because the expected cover time for any undirected graph is at most $O(n^3)$.

For deterministic algorithms, Savitch's algorithm solves the more general problem of s-t connectivity in directed graphs using $O(\log^2 n)$ space, by recursively guessing the midpoint of a directed s-t path. It remains an important open problem whether s-t connectivity in directed graphs can be solved in $O(\log n)$ space. Such an algorithm would imply $NL = L$, meaning the non-deterministic and deterministic log-space complexity classes are equal.

Reingold [Rei08] discovered a deterministic $O(\log n)$ space algorithm for s-t connectivity in undirected graphs using zig-zag products. If the input graph G is a d-regular expander graph with constant d, then G has a diameter of $O(\log n)$. Paths of length $O(\log n)$ in such a graph can be enumerated in $O(\log n)$ space, since each neighbor requires only $\log_2 d$ bits to describe. Thus, solving s-t connectivity in constant-degree expander graphs in $O(\log n)$ space is straightforward.

Reingold's approach is to transform any graph G into a regular expander graph G_k such that s, t are connected in G if and only if they are connected in G_k. First, G is converted into a D-regular graph G_1 with constant $D = d^{16}$ by replacing high degree vertices with constant-degree expander graphs and adding self-loops to low degree vertices, similar to the replacement product in Definition 5.1. The resulting G_1 is an $(n, d^{16}, \epsilon_1 d^{16})$-graph where $\epsilon_1 \leq 1 - 1/n^2$ since any connected undirected graph has a spectral gap of at least $1/n^2$.

To improve expansion, the idea is to construct $G_{i+1} := (G_i \textcircled{z} H)^8$, where H is a $(d^{16}, d, d/2)$-graph. Using a variant of the zig-zag theorem in Theorem 5.3, it can be shown that the spectral gap doubles with each iteration. More precisely, if G_i is an $(nd^{16i}, d^{16}, \epsilon_i d^{16})$-graph, then G_{i+1} is an $(nd^{16(i+1)}, d^{16}, \epsilon_{i+1} d^{16})$-graph where $\epsilon_{i+1} \leq \epsilon_i^2$. Therefore, repeating this construction for $k = O(\log n)$ times yields G_k, an $(nd^{16k}, d^{16}, 3/4)$-graph with constant spectral gap. Note that the size of G_k is at most a polynomial factor larger than G for $k = O(\log n)$, and s and t are connected in G if and only if they are connected in G_k.

A technical challenge in this approach is computing a neighbor of a vertex in G_k in log-space. The intuition is that there are only $O(\log n)$ recursion levels in the zig-zag construction, and each level requires only constant space since there are just three steps and the degree is constant. Reingold proved that this can indeed be achieved using a clever data structure; see [Rei08, Vad12] for details.

Hardness Amplification

Random walks on expander graphs can also be used for hardness amplifications, transforming instances that are hard to approximate into instances that are even harder to approximate. See [HLW06, Section 3.3] or [AB06, Chapter 22] for a simple application of expander random walks in proving the hardness of approximating maximum independent sets. This provides a nice example of using expander random walks to completely eliminate randomness from a probabilistic reduction.

Dinur [Din07] provided an elegant proof of the important PCP theorem using expander random walks. Her proof, inspired by Reingold's result, involves multiple iterations of "powering" and "degree reduction", which makes the underlying constraint satisfaction problem increasingly harder to approximate. See [AB06, Chapter 22] for a good exposition of the PCP theorem.

6.4 Error Correcting Codes

A major motivation behind the early development of expander graphs comes from coding theory, where small-set vertex expansion is the key combinatorial quantity.

A code $C \subseteq \{0,1\}^n$ of length n is a subset of n-bit strings, where each string in C is called a codeword. To design a good error correcting code, we aim to choose codewords that are far from each other so as to correct more errors, while also maximizing the number of codewords so as to achieve a high information rate. This can be viewed as a sphere packing problem, where the objective is to fit in as many disjoint spheres of a certain radius as possible within $\mathbb{F}_2^n$.

Definition 6.1 (Distance of Code)**.** *Given $C \subseteq \{0,1\}^n$, the distance of C is defined as* $\mathrm{dist}(C) := \min_{x \neq y \in C} d_H(x, y)$*, where $d_H(x, y)$ is the Hamming distance between two codewords x and y. The relative distance of C is defined as* $\mathrm{dist}(C)/n$.

Definition 6.2 (Rate of Code)**.** *Given $C \subseteq \{0,1\}^n$, the rate of C is defined as* $\log(|C|)/n$*, where $\log |C|$ can be thought of as the number of bits of information sent.*

Definition 6.3 (Asymptotically Good Code)**.** *A family $C_n \in \{0,1\}^n$ of codes is asymptotically good if there are constants $r > 0$ and $\delta > 0$ such that for all n, the relative distance of C_n is at least δ, and the rate of C_n is at least r.*

The existence of an asymptotically good code can be proved using a standard probabilistic method. However, for the codes to be practical, encoding and decoding should also be achievable in polynomial time in n (ideally linear time in n). This requirement makes designing good codes much more challenging.

A common class of codes is the class of linear codes, where C is a linear subspace of $\mathbb{F}_2^n$. Linear codes have the advantage that they can be described by a basis and so encoding can be done in $O(n^2)$ time. Additionally, a simple but useful property of linear codes is that the minimum distance of the code is equal to the minimum ℓ_1-norm of a non-zero codeword, because $d_H(x,y) = \|x-y\|_1$ and $x-y$ is a codeword. The natural decoding strategy is to find the nearest codeword of a received word, but this is an NP-complete problem even for linear codes.

Low Density Parity Check Codes

The idea of constructing codes from graphs was first suggested by Gallager, who uses sparse bipartite graphs to design low-density parity check codes (LDPC codes).

Let A be a parity check matrix for code C, such that $C = \{x \mid Ax = 0\}$ where $A \in \{0,1\}^{m\times n}$ with $m < n$. Each row i of A is a parity-check constraint, requiring $\sum_{j=1}^{n} A_{ij} \cdot x(j) = 0 \pmod 2$. Note that the rate of this code is $1 - m/n$. Thus, one objective is to minimize the number of constraints to ensure a lower bound on the rate.

The matrix A can be viewed as a bipartite graph $G = (L, R; E)$, where $L = [n]$ represents variables, $R = [m]$ represents constraints, and there is an edge between variable $i \in L$ and constraint $j \in R$ if $A_{ij} = 1$. The small-set vertex expansion of G is the key property in analyzing LPDC codes.

Definition 6.4 (One-Sided Small-Set Vertex Expansion)**.** *Let $G = (L, R; E)$ be a bipartite graph with $|L| = n$ and $|R| = m$ and $m < n$. For any $0 < \delta < 1$, define the left δ-small-set vertex expansion of G as*

$$\psi_\delta^L(G) := \min_{S\subseteq L : |S| \le \delta n} \frac{|\partial(S)|}{|S|},$$

where $\partial(S)$ is the vertex boundary as in Definition 2.11.

The following theorem relates the one-sided small-set vertex expansion of the graph to the minimum distance of the code. The proof uses the unique neighbor property guaranteed by strong enough one-sided small-set vertex expansion.

Theorem 6.5 (Distance of Expander Code [SS96])**.** *Let $G = (L, R; E)$ be a left d-regular bipartite graph with $\psi_\delta^L(G) > d/2$. Then the parity check code $C(G)$ defined by G has relative distance greater than δ.*

Proof. Let $S \subseteq L$ be a subset of left vertices with $|S| \le \delta n$. By the left small-set vertex expansion assumption of G, $|\partial(S)| > d|S|/2$. A simple counting argument shows that there exists a vertex $v \in \partial(S) \subseteq R$ with only one neighbor in S. Let us call such a vertex a unique neighbor of S.

To lower bound the minimum distance, recall that it is equivalent to lower bounding the ℓ_1-norm/support-size of a codeword $x \in \{0,1\}^n$. Let S be the support of x. If $|S| \le \delta n$, by the

above argument, there exists a unique neighbor $v \in R$ of S. This implies that the parity constraint on v is not satisfied by x, so x is not a codeword of the parity check code defined by G. Therefore, any codeword of this parity check code must have support size greater than δn, and thus the minimum distance of this code is greater than δn. □

For efficient decoding, a stronger requirement $\psi_\delta^L(G) \geq 3d/4$ is needed. As discussed in Section 5.3, this strong condition is satisfied with high probability in random d-left-regular bipartite graphs. For deterministic constructions, Capalbo, Reingold, Vadhan, Wigderson [CRVW02] used an intricate variant of the zig-zag product to construct one-sided small-set vertex expanders with $\psi_\delta^L(G) \geq 0.99d$ for some $\delta > 0$, while $m/n < 0.99$ so that the rate of the code is at least 0.01.

Fast Decoding Algorithm

A key feature of LPDC codes defined by expander graphs is that there exists a surprisingly simple and efficient decoding algorithm.

Algorithm 3 Flip Algorithm for Expander Code

Require: A parity check matrix $A \in \{0,1\}^{m \times n}$ and a bit string $x \in \{0,1\}^n$.

1: Let $x^{(0)} := x$ and $t = 0$.
2: **while** there is an unsatisfied parity check constraint **do**
3: Find a bit i such that flipping it decreases the number of unsatisfied parity constraints. That is, an $i \in [n]$ such that $\left\|A\left(x^{(t)} + \chi_i\right)\right\|_1 < \left\|Ax^{(t)}\right\|_1$, where χ_i is the characteristic vector of i and the addition is mod 2. Set $x^{(t+1)} := x^{(t)} + \chi_i$ and $t \leftarrow t + 1$.
4: **end while**
5: **return** $x^{(t)}$.

The analysis of the flip algorithm relies on a stronger assumption about the left small-set vertex expansion than that in Theorem 6.5.

Theorem 6.6 (Efficient Decoding of Expander Code [SS96])**.** *Let $G = (L, R; E)$ be a left d-regular bipartite graph with $L = [n]$ and $R = [m]$ and $\psi_\delta^L(G) > 3d/4$. Let x be an n-bit string whose distance from a codeword y is at most $\delta n/2$. Then Algorithm 3 will return y in at most m iterations.*

Proof. The plan is to argue that:

1. There exists a bit i such that flipping it decreases the number of unsatisfied constraints, as long as $\mathrm{dist}_H\left(x^{(t)}, y\right) \leq \delta n$;
2. $\mathrm{dist}_H\left(x^{(t)}, y\right) \leq \delta n$ for all t if $\mathrm{dist}_H\left(x^{(0)}, y\right) \leq \delta n/2$.

These imply that the number of unsatisfied constrained decreases with each iteration, so the algorithm must stop after at most $\tau \leq m$ iterations. At that point, $x^{(\tau)}$ is a codeword because all constraints are satisfied. Moreover, $x^{(\tau)}$ must be equal to y, as $\mathrm{dist}_H(x^{(\tau)}, y) \leq \delta n$, while the distance between y and other codewords is strictly bigger than δn by Theorem 6.5.

Let $\Delta := \{i \in [n] \mid x^{(t)}(i) \neq y(i)\}$ be the set of errors at the t-th iteration. Assume $0 < |\Delta| \leq \delta n$, we argue that a bit i exists such that flipping it decreases the number of unsatisfied constraints.

Partition $\partial(\Delta)$ into satisfied neighbors $\partial_+(\Delta)$ and unsatisfied neighbors $\partial_-(\Delta)$. Since $|\Delta| \leq \delta n$, by the left small-set vertex expansion assumption,

$$|\partial_+(\Delta)| + |\partial_-(\Delta)| = |\partial(\Delta)| > 3d|\Delta|/4.$$

Consider the $d|\Delta|$ number of edges between Δ and $\partial(\Delta)$. Each vertex in $\partial_+(\Delta)$ contributes at least two such edges while each vertex in $\partial_-(\Delta)$ contributes at least one such edge, and so

$$2|\partial_+(\Delta)| + |\partial_-(\Delta)| \leq d|\Delta|.$$

Combining these two inequalities shows $|\partial_-(\Delta)| > d|\Delta|/2$. Thus, there must exist a vertex $i \in \Delta$ with strictly more than $d/2$ unsatisfied neighbors, so flipping i decreases the number of unsatisfied constraints.

To complete the proof, we argue that $|\Delta| \leq \delta n$ in any iteration. Suppose this is not true. Since $|\Delta|$ changes by one in each iteration, there must be an (earliest) iteration such that $|\Delta| = \delta n$. At this point, using the argument in the previous paragraph, there are strictly more than $d|\Delta|/2 = d\delta n/2$ unsatisfied constraints. However, since the initial error is at most $\delta n/2$, the number of unsatisfied constraints in the beginning is at most $d\delta n/2$. This contradicts with the previous paragraph that the number of unsatisfied constraints is decreasing when $|\Delta| \leq \delta n$. □

Spielman [Spi96] further designed "superconcentrator codes" to construct asymptotically good codes that are linear-time encodable and decodable.

Tanner Codes

Tanner codes generalize LDPC codes by allowing the "base code" to be more general than a simple parity check. Let $C_0 \subseteq \{0,1\}^d$ be the base code, and let $G = (V, E)$ be a d-regular graph with $V = [n]$ and $E = [m]$. The Tanner code is defined as

$$C(G) := \{y \in \{0,1\}^m \mid y_{|\delta(i)} \in C_0 \ \forall i \in [n]\},$$

where $y_{|\delta(i)}$ is the vector y restricted to the d edges in $\delta(i)$ for a vertex $i \in V$. Each bit $y(j)$ of a codeword corresponds to an edge $j \in E$ of G, and a binary string y is a codeword if $y_{|\delta(i)}$ is a codeword of the base code C_0 for every vertex $i \in V$ of G.

The advantage of Tanner codes is that we could use a stronger base code with larger minimum distance, instead of a parity check code with minimum distance two. For a base code C_0 with minimum distance d_0, the vertex expansion requirement for G can be relaxed to d/d_0 to achieve the same distance as the corresponding LDPC code. In particular, by Tanner's theorem in Theorem 4.14, a spectral expander can be used as G to design asymptotically good codes that are linear time encodable and decodable, without requiring lossless expanders.

The decoding algorithm for Tanner codes is still an iterative "fixing" algorithm, where invalid codewords on a vertex are replaced by their nearest valid codewords in the base code. The analysis is similar to LDPC codes: if the decoding algorithm fails, one can argue that there must be a "denser" subgraph exists than what is allowed by the expander mixing lemma.

Recent breakthroughs [DEL+22, PK22] in designing asymptotically good codes that are also locally testable generalize Tanner codes to 2-dimensional expanders (where graphs are considered 1-dimensional expanders).

6.5 Problems

Problem 6.7 (Upper Bound on α). *Prove that an (n, d, α)-graph satisfies $\alpha \leq d\big(1 - \Omega(\frac{1}{n^2})\big)$ as long as it is connected and has a self-loop at each vertex.*

Hint: You may use the fact that the diameter of a d-regular graph is $O(n/d)$.

Problem 6.8 (Cover Time). *Let G be an (n, d, α)-graph with a self-loop at each vertex. Let s be a vertex in G, and let $(X_0, \ldots, X_l)$ be a random walk on G where $X_0 := s$. Show that, for every vertex t connected to s,*

$$\Pr[X_l = t] \gtrsim \frac{1}{n} \quad \textit{when} \quad l \gtrsim n^2 \log n.$$

Conclude that a random walk of $O(n^3 \log^2 n)$ steps will reach every vertex t connected to s with high probability.

References

[AB06] S. Arora and B. Barak. *Computational Complexity: A Modern Approach*. Cambridge University Press, 2006. 65, 73, 80

[AKS83] Miklós Ajtai, János Komlós, and Endre Szemerédi. An o(n log n) sorting network. In *Proceedings of the 15th Annual ACM Symposium on Theory of Computing, 25-27 April, 1983, Boston, Massachusetts, USA*, pages 1–9. ACM, 1983. 78

[CKL13] Ho Yee Cheung, Tsz Chiu Kwok, and Lap Chi Lau. Fast matrix rank algorithms and applications. *J. ACM*, 60(5):31:1–31:25, 2013. 78

[CLL13] Ho Yee Cheung, Lap Chi Lau, and Kai Man Leung. Graph connectivities, network coding, and expander graphs. *SIAM Journal on Computing*, 42(3):733–751, 2013. 79

[CPT21] Gil Cohen, Noam Peri, and Amnon Ta-Shma. Expander random walks: a Fourier-analytic approach. In *53rd Annual ACM SIGACT Symposium on Theory of Computing,*, pages 1643–1655. ACM, 2021. 78

[CRVW02] Michael Capalbo, Omer Reingold, Salil Vadhan, and Avi Wigderson. Randomness conductors and constant-degree lossless expanders. In *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, pages 659–668, 2002. 73, 82

[DEL+22] Irit Dinur, Shai Evra, Ron Livne, Alexander Lubotzky, and Shahar Mozes. Locally testable codes with constant rate, distance, and locality. In Stefano Leonardi and Anupam Gupta, editors, *54th Annual ACM SIGACT Symposium on Theory of Computing*, pages 357–374. ACM, 2022. 77, 83, 329

[Din07] Irit Dinur. The PCP theorem by gap amplification. *J. ACM*, 54(3):12, 2007. 80

[Gil98] David Gillman. A Chernoff bound for random walks on expander graphs. *SIAM J. Comput.*, 27(4):1203–1220, 1998. 78

[GLSS18] Ankit Garg, Yin Tat Lee, Zhao Song, and Nikhil Srivastava. A matrix expander Chernoff bound. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, (STOC)*, pages 1102–1114. ACM, 2018. 78

[HLW06] Shlomo Hoory, Nathan Linial, and Avi Wigderson. Expander graphs and their applications. *Bull. Amer. Math. Soc.*, 43(04):439–562, 2006. 7, 25, 26, 60, 63, 67, 68, 73, 77, 78, 80

[PK22] Pavel Panteleev and Gleb Kalachev. Asymptotically good quantum and locally testable classical LDPC codes. In *54th Annual ACM SIGACT Symposium on Theory of Computing*, pages 375–388. ACM, 2022. 77, 83

[Rei08] Omer Reingold. Undirected connectivity in log-space. *J. ACM*, 55(4):17:1–17:24, 2008. 79, 80

[Spi96] Daniel A. Spielman. Linear-time encodable and decodable error-correcting codes. *IEEE Trans. Inf. Theory*, 42(6):1723–1731, 1996. 83

[SS96] Michael Sipser and Daniel A. Spielman. Expander codes. *IEEE Trans. Inf. Theory*, 42(6):1710–1722, 1996. 81, 82

[Vad12] Salil P. Vadhan. Pseudorandomness. *Found. Trends Theor. Comput. Sci.*, 7(1-3):1–336, 2012. 63, 78, 80

Part I

Generalizations of Cheeger's Inequality

We study generalizations of Cheeger's inequality through higher eigenvalues and explore recent extensions for vertex expansion, directed graphs, and hypergraphs using reweighted eigenvalues.

Chapter 7

Higher-Order Cheeger Inequality

Cheeger's inequality in Chapter 2 is a robust generalization of the basic fact that $\lambda_2(G) = 0$ if and only if G is disconnected. The higher-order Cheeger inequality in this chapter is a robust generalization of the basic fact that $\lambda_k(G) = 0$ if and only if G has k connected components.

Arora, Barak and Steurer [ABS10] were the first to establish such a generalization. Informally, they proved that if λ_k is small then there exists a set S with small edge conductance and $|S| \lesssim |V|/k$, generalizing the fact that if $\lambda_k = 0$ then there exists a set S with edge conductance 0 and $|S| \leq |V|/k$. This result was used in [ABS10] to design a subexponential-time algorithm for Unique Games. This work has inspired many subsequent studies that use higher eigenvalues to design and analyze approximation algorithms. We will present their result later in Chapter 11 when we study random-walk based techniques for graph partitioning.

The higher-order Cheeger inequality provides a conceptually stronger generalization, asserting that λ_k is small if and only if G has at least k disjoint subsets $S_1, \ldots, S_k$ each with small edge conductance.

Definition 7.1 (k-Way Edge Conductance)**.** *Let $G = (V, E)$ be a graph. The k-way edge conductance is defined as*

$$\phi_k(G) = \min_{S_1, S_2, \ldots, S_k \subseteq V} \max_{1 \leq i \leq k} \phi(S_i),$$

where the minimization is over pairwise disjoint subsets $S_1, \ldots, S_k$ of V.

The following results were obtained independently by two research groups.

Theorem 7.2 (Higher-Order Cheeger Inequalities [LOT14, LRTV12])**.** *Let $G = (V, E)$ be a graph and let λ_k denote the k-th smallest eigenvalue of its normalized Laplacian matrix. Then*

$$\lambda_k \lesssim \phi_k(G) \lesssim k^2 \cdot \sqrt{\lambda_k}.$$

Moreover, for fewer disjoint subsets, there is an improved dependency on k:

$$\phi_{k/2}(G) \lesssim \sqrt{\log k \cdot \lambda_k}.$$

The direction $\lambda_k \lesssim \phi_k(G)$ is called the easy direction, while the direction $\phi_k(G) \lesssim k^2\sqrt{\lambda_k}$ is called the hard direction. As in Cheeger's inequality, the easy direction shows that λ_k is a relaxation of the k-way edge-conductance problem, while the hard direction is proved using a rounding algorithm for this relaxation. We will prove the hard direction and leave the easy direction in Problem 7.3. We assume the graph is d-regular throughout this chapter.

Problem 7.3. *Prove the easy direction. Hint: Use the Courant-Fischer theorem in Theorem A.14.*

7.1 Cheeger Rounding and Spectral Embedding

We first revisit the Cheeger rounding algorithm from Chapter 2 and then motivate the use of the spectral embedding for the k-way edge-conductance problem.

Cheeger Rounding

The following rounding algorithm is a consequence of the threshold rounding step in Lemma 2.5 and the embedding step in Lemma 2.7. Note that it applies to all vectors, not just eigenvectors.

Lemma 7.4 (Cheeger Rounding)**.** *Given graph $G = (V = [n], E)$ and a vector $x \in \mathbb{R}^n$, there exists an efficient algorithm to find a subset $S \subseteq \mathrm{supp}(x)$ such that*

$$\phi(S) \leq \sqrt{2R(x)} \quad \text{where} \quad R(x) = \frac{x^\top \mathcal{L} x}{x^\top x}$$

is the Rayleigh quotient of x, and $\mathrm{supp}(x) := \{i \in [n] \mid x(i) \neq 0\}$ is the support of vector x.

When λ_k is small, there are k orthogonal eigenvectors $v_1, \ldots, v_k$, each with a small Rayleigh quotient. Applying Lemma 7.4 to each v_i produces subsets $S_1, \ldots, S_k$, each with small edge conductance. Since the eigenvectors $v_1, \ldots, v_k$ are orthogonal, it is natural to expect that the sets $S_1, \ldots, S_k$ differ significantly. However, dealing with each vector separately does not provide a clear way to combine the resulting subsets $S_1, \ldots, S_k$ into k disjoint subsets with small edge conductance.

This motivates a more global view that considers all the k vectors simultaneously.

Spectral Embedding

An interesting idea in [LOT14, LRTV12] is to use the spectral embedding defined by the first k eigenvectors to find k disjoint subsets of small edge conductance (a set of small edge conductance is also called a sparse cut).

Definition 7.5 (Spectral Embedding)**.** *Let $G = (V = [n], E)$ be a graph. Let $\lambda_1 \leq \ldots \leq \lambda_k$ be the k smallest eigenvalues of $\mathcal{L}(G)$, and $v_1, \ldots, v_k \in \mathbb{R}^n$ be the corresponding orthonormal eigenvectors. Let $U \in \mathbb{R}^{n \times k}$ be the $n \times k$ matrix where the j-th column is v_j. The spectral embedding $u_i \in \mathbb{R}^k$ of a vertex i is defined as the i-th row of U.*

$$U = \begin{bmatrix} | & & | \\ v_1 & \ldots & v_k \\ | & & | \end{bmatrix} = \begin{bmatrix} — & u_1 & — \\ — & u_2 & — \\ & \vdots & \\ — & u_n & — \end{bmatrix}$$

In the spectral embedding, each vertex i is mapped to a k-dimensional point $u_i \in \mathbb{R}^k$. This embedding is commonly used in practice to find disjoint sparse cuts. Heuristics often apply geometric clustering algorithms, such as the k-means algorithm, to partition the points into k clusters, which are then used to partition the graph.

These heuristics are reported to work well in applications such as image segmentation and data clustering, but no theoretical guarantees were known. The proof of the higher-order Cheeger inequality provides a rigorous analysis of certain variants of these methods [NJW01], justifying the use of spectral embedding for graph partitioning. Analyzing the k-means heuristics remains an open problem. See [PSZ17] for a rigorous analysis of the k-means heuristic for "well-clustered" graphs.

7.2 Isotropy Condition and Clustering by Directions

For the spectral embedding to provide useful information for finding k disjoint sparse cuts, it is necessary that the points are reasonably "well spread out". For example, if all vertices are mapped to only two points in $\mathbb{R}^k$, then there is no well-defined way to partition the points into k clusters.

As expected, such bad cases do not occur because the k eigenvectors are orthogonal. The question is how to use the orthogonality to derive conditions that facilitate clustering.

Isotropy Condition

As $v_1, \ldots, v_k$ are orthonormal vectors, the matrix U in Definition 7.5 satisfies $U^\top U = I_k$, which can be rewritten as $\sum_{i=1}^n u_i u_i^\top = I_k$. This implies that the spectral embedding satisfies the following isotropy condition.

Lemma 7.6 (Isotropy Condition)**.** *Let $u_1, \ldots, u_n \in \mathbb{R}^k$ be the spectral embedding of the vertices in Definition 7.5. For any $x \in \mathbb{R}^k$ with $\|x\|_2 = 1$,*

$$\sum_{i=1}^n \langle x, u_i \rangle^2 = 1.$$

Proof. The condition $U^\top U = I_k$ implies that $x^\top U^\top U x = x^\top x = 1$ for any $x \in \mathbb{R}^k$ with $\|x\|_2 = 1$. Writing $U^\top U = \sum_{i=1}^n u_i u_i^\top$ shows that

$$1 = x^\top \Big(\sum_{i=1}^n u_i u_i^\top \Big) x = \sum_{i=1}^n x^\top u_i u_i^\top x = \sum_{i=1}^n \langle x, u_i \rangle^2.$$

□

This lemma says that for any direction $x \in \mathbb{R}^k$, the sum of the square of the projection of u_i on x is equal to 1. To develop some intuition, suppose $u_1 = u_2 = \cdots = u_l = y \in \mathbb{R}^k$ (that the first l vertices all mapped to the same point y), then the lemma implies that

$$1 = \sum_{i=1}^n \Big\langle \frac{y}{\|y\|_2}, u_i \Big\rangle^2 \geq \sum_{i=1}^l \Big\langle \frac{u_i}{\|u_i\|_2}, u_i \Big\rangle^2 = \sum_{i=1}^l \|u_i\|^2.$$

On the other hand, as each eigenvector satisfies $\|v_i\|_2 = 1$,

$$\sum_{i=1}^n \|u_i\|_2^2 = \|U\|_F^2 = \sum_{i=1}^k \|v_i\|_2^2 = k. \tag{7.1}$$

Therefore, if we consider the "mass" of a point i as $\|u_i\|_2^2$, then the above calculation shows that at most $1/k$ fraction of the total mass can be mapped to the same point. This ensures that the spectral embedding maps to at least k distinct points, ruling out the bad cases mentioned earlier.

Clustering by Directions

The same reasoning shows that points with similar directions carry at most $\approx 1/k$ fraction of the total mass. This motivates clustering points by their directions, ensuring they are reasonably well spread out. The following distance measure and spreading property formalize this idea.

Definition 7.7 (Radial Projection Distance)**.** *Let $u_1, \ldots, u_n \in \mathbb{R}^k$ be the spectral embedding of the vertices in Definition 7.5. The radial projection distance between two vertices i and j is defined as*

$$d(i,j) = \left\| \frac{u_i}{\|u_i\|_2} - \frac{u_j}{\|u_j\|_2} \right\|_2$$

if $\|u_i\| > 0$ and $\|u_j\| > 0$. Otherwise, if $u_i = u_j = 0$ then $d(i,j) := 0$, else $d(i,j) := \infty$.

Lemma 7.8 (Spreading Property)**.** *Let $G = (V = [n], E)$ be a graph. Let $u_1, \ldots, u_n \in \mathbb{R}^k$ be the spectral embedding of the vertices in Definition 7.5. If $S \subseteq V$ satisfies $d(i,j) \leq \Delta$ for all $i, j \in S$, then*

$$\sum_{i \in S} \|u_i\|^2 \leq \frac{1}{1-\Delta^2}.$$

Proof. Choose an arbitrary vertex $j \in S$ and consider the unit vector $u_j/\|u_j\|_2$. By the isotropy condition in Lemma 7.6,

$$1 = \sum_{i=1}^{n} \left\langle \frac{u_j}{\|u_j\|_2}, u_i \right\rangle^2 \geq \sum_{i \in S} \|u_i\|_2^2 \cdot \left\langle \frac{u_j}{\|u_j\|_2}, \frac{u_i}{\|u_i\|_2} \right\rangle^2 = \sum_{i \in S} \|u_i\|_2^2 \cdot \left(1 - \frac{d^2(i,j)}{2}\right)^2,$$

where the last equality holds because $\langle u, v \rangle = 1 - \|u - v\|_2^2/2$ for any two unit vectors u, v. Using the assumption $d(i,j) \leq \Delta$, it follows that

$$1 \geq \sum_{i \in S} \|u_i\|_2^2 \cdot \left(1 - \frac{\Delta^2}{2}\right)^2 \geq \sum_{i \in S} \|u_i\|_2^2 \cdot (1 - \Delta^2).$$

Rearranging gives the lemma. □

To ensure mass balance, we will choose Δ such that $\frac{1}{1-\Delta^2} \leq 1 + \frac{1}{2k}$ (e.g., $\Delta = \frac{1}{2\sqrt{k}}$). By forming subsets with diameter Δ, the lemma ensures that each subset has mass at most $1 + \frac{1}{2k}$. Thus, after selecting $k-1$ subsets, the remaining mass is still at least $1/2$. This ensures that k groups, each containing $\Omega(1/k)$ fraction of the total mass, can be formed by clustering based on directions.

7.3 Ideal Case: Far Apart Clusters

The previous section rules out the bad cases for clustering based on directions. In this section, we analyze the ideal scenario when the spectral embedding provides exactly what we want: k clusters that are far apart from each other.

Suppose there exist k disjoint subsets $S_1, S_2, \ldots, S_k$ such that (see Figure 7.1):.

- Each subset S_i has mass 1, i.e., $\sum_{j \in S_i} \|u_j\|_2^2 = 1$;
- The clusters are well-separated, i.e., $d(S_i, S_j) \geq \delta$ for all $i \neq j$ where $d(S_i, S_j) := \min\{d(a,b) \mid a \in S_i, b \in S_j\}$.

Can we conclude that these k subsets correspond to k disjoint sparse cuts in the graph?

Rayleigh Quotient of the Spectral Embedding and Cheeger Rounding

To analyze the spectral embedding, we extend the Rayleigh quotient definition and the Cheeger rounding result in Lemma 7.4 to work with k-dimensional embeddings.

Recall that the Rayleigh quotient of a vector $x : V \to \mathbb{R}$ is defined as

$$R(x) = \frac{x^\top \mathcal{L} x}{x^\top x} = \frac{x^\top L x}{d x^\top x} = \frac{\sum_{ij\in E}(x(i) - x(j))^2}{d\sum_{i\in V} x(i)^2}.$$

For k-dimensional embeddings, the Rayleigh quotient is defined as follows.

Definition 7.9 (Rayleigh Quotient of a k-Dimensional Embedding)**.** *Let $\psi_1, \ldots, \psi_n \in \mathbb{R}^k$ and $\Psi = (\psi_1, \ldots, \psi_n)$ be a k-dimensional embedding of the vertices. The Rayleigh quotient of Ψ is defined as*

$$R(\Psi) := \frac{\sum_{ij\in E}\|\psi_i - \psi_j\|_2^2}{d\sum_{i\in V}\|\psi_i\|_2^2}.$$

From a k-dimensional embedding with a small Rayleigh quotient, we can apply Cheeger rounding to obtain a sparse cut within its support.

Lemma 7.10 (Cheeger Rounding for k-Dimensional Embedding)**.** *Given graph $G = (V = [n], E)$ and a k-dimensional embedding $\Psi = (\psi_1, \ldots, \psi_n)$ of the vertices, there exists an efficient algorithm to find a subset $S \subseteq \mathrm{supp}(\Psi)$ such that*

$$\phi(S) \le \sqrt{2R(\Psi)},$$

where $\mathrm{supp}(\Psi) := \{i \in [n] \mid \psi_i \neq \vec{0}\}$ is the support of Ψ.

Proof. Expanding the k-dimensional embedding coordinate-wise and applying Lemma 2.6,

$$R(\Psi) = \frac{\sum_{ij\in E}\sum_{l=1}^k(\psi_i(l) - \psi_j(l))^2}{d\sum_{i\in V}\sum_{l=1}^k \psi_i(l)^2} = \frac{\sum_{l=1}^k\sum_{ij\in E}(\psi_i(l) - \psi_j(l))^2}{d\sum_{l=1}^k\sum_{i\in V}\psi_i(l)^2} \ge \min_l \frac{\sum_{ij\in E}(\psi_i(l) - \psi_j(l))^2}{d\sum_{i\in V}\psi_i(l)^2}.$$

Let $x_l(i) := \psi_i(l)$ for $i \in [n]$ and $1 \le l \le k$. The above implies that there is a coordinate l such that $R(x_l) \le R(\Psi)$. Applying Cheeger rounding in Lemma 7.4 on x_l gives the lemma. □

The spectral embedding provides an initial k-dimensional embedding with a small Rayleigh quotient.

Lemma 7.11 (Rayleigh Quotient of the Spectral Embedding)**.** *Let $u_1, \ldots, u_n \in \mathbb{R}^k$ be the spectral embedding of the vertices in Definition 7.5. The Rayleigh quotient of the spectral embedding is*

$$R(U) := \frac{\sum_{ij\in E}\|u_i - u_j\|_2^2}{d\sum_{i\in V}\|u_i\|_2^2} = \frac{1}{k}\sum_{l=1}^k \lambda_l \le \lambda_k.$$

Proof. Since $u_i(l) = v_l(i)$ where v_l is the l-th eigenvector of $\mathcal{L}$,

$$R(U) = \frac{\sum_{ij\in E}\sum_{l=1}^k(u_i(l) - u_j(l))^2}{d\sum_{i\in V}\sum_{l=1}^k u_i(l)^2} = \frac{\sum_{l=1}^k\sum_{ij\in E}(v_l(i) - v_l(j))^2}{d\sum_{l=1}^k\sum_{i\in V}v_l(i)^2} = \frac{\sum_{l=1}^k \lambda_l d}{dk} = \frac{1}{k}\sum_{l=1}^k \lambda_l,$$

where we used $\sum_{i\in V} v_l(i)^2 = \|v_l\|_2^2 = 1$ and $\sum_{ij\in E}(v_l(i) - v_l(j))^2 = d\cdot R(v_l)\cdot\sum_{i\in V} v_l(i)^2 = d\cdot\lambda_l$. □

The Basic Plan

We are now ready to analyze the ideal scenario. Starting from the spectral embedding U with a Rayleigh quotient at most λ_k, the plan is to construct k embeddings $\Psi_1, \ldots, \Psi_k$ such that $\text{supp}(\Psi_l) \subseteq S_l$. If this can be done in a way that the Rayleigh quotient of Ψ_l is small for all $1 \leq l \leq k$, then Cheeger rounding in Lemma 7.10 can be applied to each Ψ_l to obtain a sparse cut supported on S_l.

The question is how to construct Ψ_l and upper bound its Rayleigh quotient. The most natural way to define Ψ_l is to zero out everything outside S_l:

$$\Psi_l = (\psi_{l,1}, \ldots, \psi_{l,n}) \quad \text{where} \quad \psi_{l,i} = \begin{cases} u_i & \text{if } i \in S_l \\ \vec{0} & \text{otherwise.} \end{cases}$$

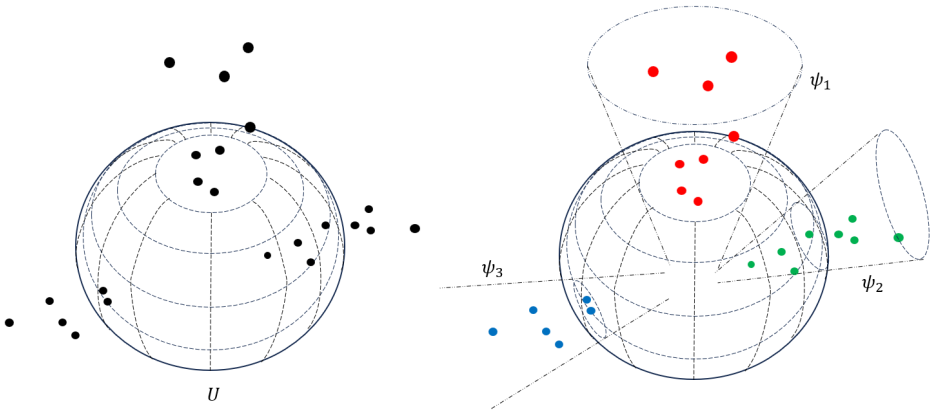


Figure 7.1: The spectral embedding U in the ideal scenario when $k = 3$, and the three disjoint supported embeddings constructed from U.

The Rayleigh quotient of Ψ_l is

$$R(\Psi_l) = \frac{\sum_{ij \in E} \|\psi_{l,i} - \psi_{l,j}\|_2^2}{d \sum_{i \in V} \|\psi_{l,i}\|_2^2} = \frac{\sum_{ij \in E, i \in S_l, j \in S_l} \|u_i - u_j\|^2 + \sum_{ij \in E, i \in S_l, j \notin S_l} \|u_i\|^2}{d \sum_{i \in S_l} \|u_i\|_2^2}$$

To bound $R(\Psi_l)$, we compare its numerator and denominator with $R(U)$ term-by-term.

For the denominator, $\sum_{i \in S_l} \|u_i\|^2 = 1$ by the assumption of the ideal scenario, and $\sum_{i \in V} \|u_i\|^2 = k$ by (7.1). Thus, the denominator of $R(\Psi_l)$ is $1/k$ times of that of $R(U)$.

For the numerator, edges $ij \in E$ with $i \in S_l$ and $j \in S_l$ contribute equally to $R(\Psi_l)$ and $R(U)$. For edges with $i \in S$ and $j \notin S$, the contribution to $R(\Psi_l)$ is $\|u_i\|^2$, while the contribution to $R(U)$ is $\|u_i - u_j\|^2$. By Claim 7.12, $\|u_i\|_2 \leq 2\|u_i - u_j\|_2 / d(i,j) \leq 2\|u_i - u_j\|_2 / \delta$.

Combining these bounds,

$$R(\Psi_l) = \frac{\sum_{ij \in E, i \in S_l, j \in S_l} \|u_i - u_j\|_2^2 + \sum_{ij \in E, i \in S_l, j \notin S_l} \|u_i\|_2^2}{d \sum_{i \in S_l} \|u_i\|_2^2} \lesssim \frac{k}{\delta^2} \cdot \frac{\sum_{ij \in E} \|u_i - u_j\|_2^2}{d \sum_{i \in V} \|u_i\|_2^2} = \frac{k}{\delta^2} \cdot R(U).$$

Applying Cheeger rounding in Lemma 7.10 on each Ψ_l gives a set $S_l' \subseteq S_l$ with $\phi(S_l') \lesssim \sqrt{k\lambda_k}/\delta$. Assuming δ is a constant, the sets $S_1', \ldots, S_l'$ are disjoint sparse cuts with edge conductance $O(\sqrt{k\lambda_k})$. This is the basic idea behind the hard direction of Theorem 7.2.

Claim 7.12. *For two vectors u_i, u_j, $\|u_i\|_2 \le 2\|u_i - u_j\|_2/d(i,j)$ where $d(i,j)$ is the radial projection distance in Definition 7.7.*

Proof. By the triangle inequality,

$$d(i,j) = \left\| \frac{u_i}{\|u_i\|} - \frac{u_j}{\|u_j\|} \right\| \le \left\| \frac{u_i}{\|u_i\|} - \frac{u_j}{\|u_i\|} \right\| + \left\| \frac{u_j}{\|u_i\|} - \frac{u_j}{\|u_j\|} \right\|.$$

For the second term, again by the triangle inequality,

$$\left\| \frac{u_j}{\|u_i\|} - \frac{u_j}{\|u_j\|} \right\| = \frac{\big|\|u_j\| - \|u_i\|\big|}{\|u_i\|} \le \frac{\|u_i - u_j\|}{\|u_i\|}.$$

Therefore, $d(i,j) \le 2\|u_i - u_j\|/\|u_i\|$. Rearranging gives the claim. □

Improved Dependency on k for Fewer Disjoint Sparse Cuts: If we focus on finding only $k/2$ disjoint sparse cuts, we can achieve a better dependency on k. Sort the sets $S_1, \dots, S_k$ so that

$$\sum_{ij \in E, i \in S_1} \|u_i - u_j\|_2^2 \le \cdots \le \sum_{ij \in E, i \in S_k} \|u_i - u_j\|_2^2.$$

Since each edge can contribute to at most two groups, an averaging argument implies that

$$\sum_{ij \in E, i \in S_{k/2}} \|u_i - u_j\|_2^2 \lesssim \frac{1}{k} \sum_{ij \in E} \|u_i - u_j\|_2^2.$$

This reduces the numerator by a factor of k and gives $R(\Psi_{k/2}) \lesssim R(U)/\delta^2$. Assuming δ is a constant, the sets $S_1', \dots, S_{k/2}'$ are disjoint sparse cuts with edge conductance $O(\sqrt{\lambda_k})$.

7.4 General Case: Space Partitioning and Smooth Localization

We follow a similar strategy in the general case. We aim to find k disjoint subsets $S_1, \dots, S_k \subseteq V$ such that:

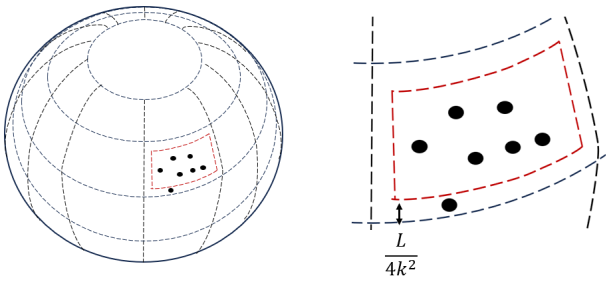
- Each subset S_i has mass $\Omega(1)$, or equivalently $\sum_{j \in S_i} \|u_j\|_2^2 \gtrsim \frac{1}{k} \sum_{i=1}^n \|u_i\|_2^2$;
- The clusters are well-separated, i.e., $d(S_i, S_j) \ge 2\delta$ for all $i \ne j$.

We describe below how to partition the space to construct these disjoint subsets, and then how to use smooth localization to obtain disjoint sparse cuts from these subsets.

Space Partitioning

The difficult case is when the points are evenly distributed across the space, in which it is not clear how to identify the disjoint subsets $S_1, \dots, S_k$ with the required properties.

A simple approach, presented in [Lee13], is to partition the directions in $\mathbb{S}^{k-1}$ (the k-dimensional sphere) into cubes with side length $L = \frac{1}{2k}$. All points $u_i \in \mathbb{R}^k$ with directions $u_i/\|u_i\|_2$ in the same



cube Q are grouped into a block B. The diameter of each cube Q is $L\sqrt{k} = \frac{1}{2\sqrt{k}}$. By the spreading property in Lemma 7.8, the points in each block B has mass at most $1 + \frac{1}{2k}$.

To ensure that points in different cubes are sufficiently far apart, we define $\tilde{Q} \subseteq Q$ as the set of points on $\mathbb{S}^{k-1}$ that are at least $\frac{L}{4k^2}$ away (in Euclidean distance) from every side of Q. Similarly, $\tilde{B} \subseteq B$ denotes the points $u_i \in \mathbb{R}^k$ with directions $u_i/\|u_i\|_2$ in $\tilde{Q}$. By this construction,

$$\mathrm{vol}(\tilde{Q}) = \left(1 - \frac{1}{4k^2}\right)^k \cdot \mathrm{vol}(Q) \geq \left(1 - \frac{1}{4k}\right) \cdot \mathrm{vol}(Q).$$

Thus, if we choose a uniformly random axis-parallel translation of the cube partition, the expected total mass of points in the shrunk blocks is at least $k - \frac{1}{4}$.

To construct disjoint $S_1, \ldots, S_k \subseteq V$ where each S_i has mass at least $\frac{1}{4}$, we sort the shrunk blocks by non-increasing mass and greedily form $S_1, \ldots, S_k$ so that each S_i has mass at least $\frac{1}{4}$. This is always possible since no block has mass greater than $1 + \frac{1}{2k}$ (in the worst case, the last subset S_k has mass $(k - \frac{1}{4}) - (k-1)(1 + \frac{1}{2k}) \geq \frac{1}{4}$).

To summarize, the result achieved in this subsection is as follows.

Lemma 7.13 (Disjoint Subsets)**.** *There are disjoint subsets $S_1, \ldots, S_k$ such that*

- *Each subset S_i has mass at least $\frac{1}{4}$, i.e., $\sum_{j \in S_i} \|u_j\|_2^2 \geq \frac{1}{4}$;*
- *The clusters are well-separated such that for all $i \neq j$, $d(S_i, S_j) \geq \frac{L}{2k^2} = \frac{1}{4k^3}$.*

Smooth Localization

Given disjoint subsets $S_1, \ldots, S_k$ in Lemma 7.13, the goal is to construct k embeddings $\Psi_1, \ldots, \Psi_k : V \to \mathbb{R}^k$ with small Rayleigh quotients and $\mathrm{supp}(\Psi_i) \subseteq S_i$, as in the ideal scenario.

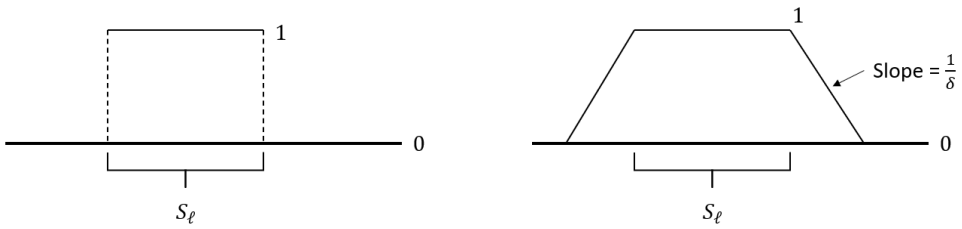
The difference from the ideal scenario is that there are points not in $S_1 \cup \cdots \cup S_k$. Suppose there is a point $j \notin S_1 \cdots \cup S_k$ but very close to some point $i \in S_l$. In this case, if Ψ_l is defined by zeroing out all points outside S_l, the length of the edge ij in Ψ_l would be $\|u_i\|_2$, which could be much larger than $\|u_i - u_j\|_2$. The ratio could be unbounded, and the term-by-term analysis would fail.

To handle this issue, we use the condition $d(S_i, S_j) \geq 2\delta$ to provide some room to "smoothly" decrease the length of the points close to S_l to zero.

Definition 7.14 (Smooth Localization)**.** *Let δ be a parameter and $S_1, \ldots, S_k$ be disjoint subsets. For each $1 \leq l \leq k$ and each point j, let $d(j, S_l) = \min_{i \in S_l} d(i, j)$ and define*

$$c_j := \max\left\{1 - \frac{d(j, S_l)}{\delta}, 0\right\} \quad \textit{and} \quad \psi_{l,j} := c_j u_j,$$

where $\Psi_l = (\psi_{l,1}, \ldots, \psi_{l,k})$ is the k-dimensional embedding that we construct and $U = (u_1, \ldots, u_n)$ is the spectral embedding in Definition 7.5.



Note that if $d(j, S_l) \geq \delta$, then $\psi_{l,j} = 0$, so the embeddings $\Psi_1, \ldots, \Psi_k$ are disjointly supported if $d(S_i, S_j) \geq 2\delta$ for all $i \neq j$.

On the other hand, if $d(j, S_l) \leq \delta$ then c_j decreases linearly with distance, with a slope of $1/\delta$. This smooth localization is designed to ensure that the term-by-term analysis works.

Lemma 7.15 (Distortion from Smooth Localization)**.** *Using the same notations in Definition 7.14, for all $ij \in E$,*

$$\|\psi_{l,i} - \psi_{l,j}\|_2 \leq \left(1 + \frac{2}{\delta}\right) \cdot \|u_i - u_j\|_2.$$

Proof. Following the definitions in Definition 7.14,

$$\|\psi_{l,i} - \psi_{l,j}\| = \|c_i u_i - c_j u_j\| = \|c_i u_i - c_j u_i + c_j u_i - c_j u_j\| \leq |c_i - c_j| \cdot \|u_i\| + |c_j| \cdot \|u_i - u_j\|.$$

Note that $|c_i - c_j| \leq d(i,j)/\delta$, as $d(j, S) - d(i, S) \leq d(i, j)$ by the triangle inequality. Combining this with Claim 7.12, the first term is

$$|c_i - c_j| \cdot \|u_i\| \leq \frac{d(i,j)}{\delta} \cdot \frac{2\|u_i - u_j\|}{d(i,j)} = \frac{2}{\delta}\|u_i - u_j\|.$$

Since the second term is at most $\|u_i - u_j\|$, the proof is complete. □

Putting Together

We put the pieces together to prove a weaker version of the hard direction of Theorem 7.2.

First, compute the spectral embedding $U = (u_1, \ldots, u_n)$ as in Definition 7.5.

Next, apply the space partitioning scheme in Lemma 7.13 to obtain the disjoint subsets $S_1, \ldots, S_k$, each with mass at least $1/4$, and pairwise well-separated such that $d(S_i, S_j) \geq 1/4k^3$. Note that this step relies on the spreading property in Lemma 7.8.

Then, apply smooth localization from Definition 7.14 with $\delta := 1/8k^3$ to obtain k-dimensional embeddings $\Psi_1, \ldots, \Psi_k$. By the choice of δ, the embeddings $\Psi_1, \ldots, \Psi_k$ are disjointly supported. By construction, for each $1 \leq l \leq k$, $\psi_{l,i} = u_i$ for all $i \in S_l$, and so

$$\sum_{i \in V} \|\psi_{l,i}\|_2^2 \geq \sum_{i \in S_l} \|u_i\|_2^2 \geq \frac{1}{4} \quad \Longrightarrow \quad \sum_{i \in V} \|\psi_{l,i}\|_2^2 \gtrsim \frac{1}{k} \sum_{i \in V} \|u_i\|_2^2.$$

Combining with Lemma 7.15 and Lemma 7.11, for each $1 \le l \le k$,

$$R(\Psi_l) = \frac{\sum_{ij\in E}\|\psi_{l,i} - \psi_{l,j}\|_2^2}{d\sum_{i\in V}\|\psi_{l,i}\|_2^2} \lesssim \frac{k}{\delta^2} \cdot \frac{\sum_{ij\in E}\|u_i - u_j\|_2^2}{d\sum_{i\in V}\|u_i\|_2^2} \le \frac{k}{\delta^2} \cdot \lambda_k \asymp k^7\lambda_k.$$

Finally, applying Cheeger rounding from Lemma 7.10 to each Ψ_l yields disjoint subsets $S_1', \ldots, S_l'$, each with conductance at most $O(\sqrt{k^7\lambda_k})$. This completes the proof.

Discussions

The tighter bound $O(k^2\sqrt{\lambda_k})$ is obtained using a more sophisticated random partitioning technique developed in metric embedding, namely the padded decomposition.

The bound $\phi_{k/2}(G) \lesssim \mathrm{polylog}(k) \cdot \sqrt{\lambda_k}$ is achieved by applying a dimension reduction technique to reduce the spectral embedding from k-dimensional space to $O(\log k)$-dimensional space.

The noisy hypercube provides an example where $\phi_k(G) \gtrsim \sqrt{\log k \cdot \lambda_k}$. It remains an open question whether the $\mathrm{poly}(k)$ factor in Theorem 7.2 can be replaced by a $\mathrm{polylog}(k)$ factor.

Question 7.16. *Is it true that* $\phi_k(G) \lesssim \mathrm{polylog}(k) \cdot \sqrt{\lambda_k}$*?*

Although Theorem 7.2 provides a conceptually stronger generalization, its quantitative bounds are weaker than that in [ABS10]. Finding a common generalization of these two results is an interesting open problem for future research.

7.5 Alternative Randomized Rounding Algorithms

The algorithm in [LRTV12] is elegant and simple to describe.

Algorithm 4 Randomized Rounding on Spectral Embedding [LRTV12]

Require: An undirected graph $G = (V, E)$ with $V = [n]$ and $m = |E|$, and a parameter k.

1: Compute the spectral embedding $u_1, \ldots, u_n \in \mathbb{R}^k$ in Definition 7.5.

2: Pick k independent Gaussian vectors $g_1, \ldots, g_k \in N(0,1)^n$. Construct disjointly supported vectors $h_1, \ldots, h_k \in \mathbb{R}^n$ as follows:

$$h_i(j) = \begin{cases} \langle u_j, g_i\rangle & \text{if } i = \mathrm{argmax}_{i\in[k]}\{\langle u_j, g_i\rangle\} \\ 0 & \text{otherwise.} \end{cases}$$

3: Apply Cheeger rounding from Lemma 2.5 to each h_i to obtain a set $S_i \subseteq \mathrm{supp}(h_i)$ and $\phi(S_i) \le \sqrt{2R_{\mathcal{L}}(h_i)}$.

4: **return** all S_i with $\phi(S_i) \lesssim \sqrt{\log k \cdot \lambda_k}$.

It is proved in [LRTV12] that this algorithm will return $\Omega(k)$ subsets with constant probability. The proof is by computing the expectation and the variance of the numerator and the denominator, using properties of Gaussian random variables.

See also [MMSV24] for a recent generalization of the higher-order Cheeger inequality, obtained using the orthogonal separator technique.

References

[ABS10] Sanjeev Arora, Boaz Barak, and David Steurer. Subexponential algorithms for unique games and related problems. In *51th Annual IEEE Symposium on Foundations of Computer Science, FOCS*, pages 563–572. IEEE Computer Society, 2010. 8, 89, 98

[Lee13] James R. Lee. No frills proof of higher-order cheeger inequality. 2013. 95

[LOT14] James R. Lee, Shayan Oveis Gharan, and Luca Trevisan. Multiway spectral partitioning and higher-order cheeger inequalities. *J. ACM*, 61(6):37:1–37:30, 2014. 8, 89, 90, 167

[LRTV12] Anand Louis, Prasad Raghavendra, Prasad Tetali, and Santosh S. Vempala. Many sparse cuts via higher eigenvalues. In *Proceedings of the 44th Symposium on Theory of Computing Conference, STOC*, pages 1131–1140. ACM, 2012. 8, 89, 90, 98

[MMSV24] Konstantin Makarychev, Yury Makarychev, Liren Shan, and Aravindan Vijayaraghavan. Higher-order cheeger inequality for partitioning with buffers. In *Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2236–2274. SIAM, 2024. 98

[NJW01] Andrew Ng, Michael Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. *Advances in neural information processing systems*, 14, 2001. 90

[PSZ17] Richard Peng, He Sun, and Luca Zanetti. Partitioning well-clustered graphs: Spectral clustering works! *SIAM J. Comput.*, 46(2):710–743, 2017. 90

Chapter 8

Improved Cheeger Inequality

Given the practical success of the spectral partitioning algorithm, an open question has been to provide a theoretical justification of this phenomenon.

Several research directions are relevant to this question. One approach is to analyze the average-case performance of spectral partitioning algorithm under planted random graph models [Bop87, McS01]. Another is to study spectral partitioning for specific graph classes, such as planar graphs [ST07] and minor-free graphs [BLR10]. These are significant mathematical results, but they do not necessarily capture practical instances.

An interesting recent approach is to analyze the algorithm performance on "well-clusterable" instances [BL12, DLS12], as these are particularly relevant in applications such as image segmentation and data clustering. While multiple combinatorial formulations of well-clusterable instances exist, they are not naturally amenable to analysis of spectral partitioning.

The higher-order Cheeger inequality in Theorem 7.2 provides an algebraic formulation of this notion. If λ_2 is small but λ_3 is large, then Theorem 7.2 implies that $\phi_2(G)$ is small but $\phi_3(G)$ is large. This means that there is a good way to cut the graph into two pieces but not into three. Informally, this suggests that the graph resembles two expanders connected by a sparse cut – an instance that is well-clusterable.

More generally, a graph can be considered well-clusterable if it has a large "higher-order spectral gap", meaning that λ_k is large for a small k. In such graphs, there are at most a few outstanding sparse cuts as $\phi_k(G)$ is large. The following theorem proves that the spectral partitioning algorithm performs much better on these instances.

Theorem 8.1 (Improved Cheeger's Inequality [KLL+13]). *Let $G = (V, E)$ be a graph and λ_k be the k-th smallest eigenvalue of its normalized Laplacian matrix $\mathcal{L}$. For any $k \geq 2$,*

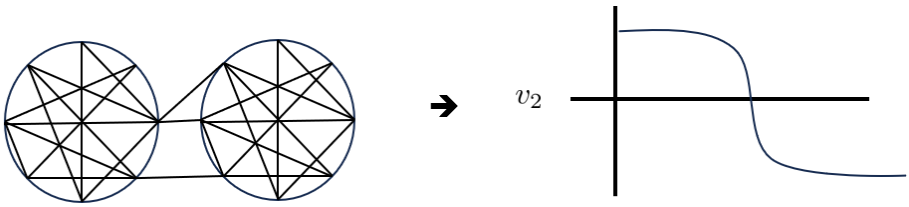
$$\frac{\lambda_2}{2} \leq \phi(G) \lesssim \frac{k\lambda_2}{\sqrt{\lambda_k}}.$$

Moreover, the spectral partitioning algorithm in Chapter 2 returns a set S with $\phi(S) \lesssim k\lambda_2/\sqrt{\lambda_k}$.

For a graph with $\lambda_k = \Omega(1)$ for a small constant k, Theorem 8.1 implies that the spectral partitioning algorithm is a constant factor approximation algorithm for edge conductance, avoiding the square-root loss in Cheeger's inequality (Theorem 2.2). This assumption is usually satisfied in practical instances from image segmentation and data clustering [VL07]. Thus, Theorem 8.1 provides a rigorous justification of the empirical success of the spectral partitioning algorithm.

8.1 Intuition and Proof Outline

Consider the case when λ_2 is small but λ_3 is large. Since λ_2 is small, Cheeger's inequality guarantees the existence of a sparse cut $(S, V-S)$. Since λ_3 is large, the higher-order Cheeger inequality implies that $\phi_3(G)$ is large. This means that neither $G[S]$ nor $G[V-S]$ can contain a sparse cut, as this would imply the existence of three disjoint sparse cuts, contradicting the assumption that $\phi_3(G)$ is large. Therefore, the induced graphs on both S and $V-S$ must be expander-like. Since $(S, V-S)$ is a sparse cut and the subgraphs are expanders, the minimizer for the Rayleigh quotient in Lemma 2.3 is expected to resemble a binary solution, which would imply $\lambda_2 \approx \phi(G)$.



The proof of Theorem 8.1 has two main steps. For simplicity, we assume the graph is d-regular.

The first step is to show that if λ_k is large, then any vector x with a small Rayleigh quotient $R(x) = x^\top \mathcal{L} x / x^\top x$ is close to a k-step function.

Definition 8.2 (k-Step Function)**.** *Let $G = (V = [n], E)$ be a graph. Given $y \in \mathbb{R}^n$ and $1 \le k \le n$, we say y is a k-step function if the number of distinct values in $\{y(i)\}_{i \in V}$ is at most k.*

Lemma 8.3 (Constructing $2k$-Step Approximation)**.** *Let $G = (V = [n], E)$ be a d-regular graph. For any $x \in \mathbb{R}^n_+$, there exists a $2k$-step function $y \in \mathbb{R}^n_+$ such that*

$$\frac{\|x - y\|_2}{\|x\|_2} \lesssim \sqrt{\frac{R(x)}{\lambda_k}} \quad \text{and} \quad \operatorname{supp}(y) \subseteq \operatorname{supp}(x).$$

The second step is to show that if a vector with a small Rayleigh quotient is close to a k-step function, then the spectral partitioning algorithm performs better.

Lemma 8.4 (Rounding $2k$-Step Approximation)**.** *Let $G = (V = [n], E)$ be a d-regular graph. Let $x \in \mathbb{R}^n_+$ and let $y \in \mathbb{R}^n_+$ be a $2k$-step function. The spectral partitioning algorithm applied to x outputs a set $S \subseteq \operatorname{supp}(x)$ with*

$$\phi(S) \lesssim k \cdot R(x) + k \cdot \frac{\|x - y\|_2}{\|x\|_2} \cdot \sqrt{R(x)}.$$

Given a second eigenvector $v \in \mathbb{R}^n$ with $R(v) = \lambda_2$, we apply the shifting and truncation steps in Lemma 2.8 and Lemma 2.9 to obtain a vector x with $R(x) \le R(v) \le \lambda_2$ and $|\operatorname{supp}(S)| \le n/2$. Then Theorem 8.1 follows immediately from Lemma 8.3 and Lemma 8.4.

8.2 Constructing $2k$-Step Approximation

In this section, we prove Lemma 8.3. We need to show that if λ_k is large, then a vector $x \in \mathbb{R}^n$ with a small Rayleigh quotient must be close to a $2k$-step function. We prove the contrapositive: if a vector x with a small Rayleigh quotient is far from any $2k$-step function, then λ_k is small.

The idea is that if x is far from any $2k$-step function, then x must be "smooth". We then use this smooth function to construct k disjointly supported functions $\psi_1, \ldots, \psi_k \in \mathbb{R}^n$ such that each $R(\psi_i)$ is not much larger than $R(x)$. The following lemma then implies that λ_k is small.

Lemma 8.5 (Generalization of Easy Direction of Higher-Order Cheeger Inequality)**.** *If $\psi_1, \ldots, \psi_k \in \mathbb{R}^n$ are vectors with disjoint supports, then*

$$\lambda_k \leq 2 \max_{1 \leq i \leq k} R(\psi_i).$$

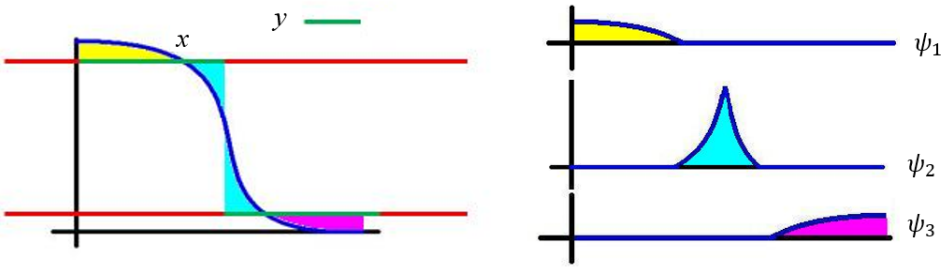
The Case When $k = 3$

We illustrate the main idea in the case when $k = 3$. To construct the disjointly supported functions ψ_1, ψ_2, ψ_3, we choose two threshold values $t_1 > t_2$ and partition the vertex set into three groups:

$$S_1 := \{i \mid x(i) \geq t_1\}, \quad S_2 := \{i \mid t_1 > x(i) \geq t_2\}, \quad S_3 := \{i \mid t_2 > x(i) \geq 0\}.$$

Each ψ_l is a vector with $\text{supp}(\psi_l) \subseteq S_l$, defined as

$$\psi_l(i) = \begin{cases} \min\{|x(i) - t_1|, |x(i) - t_2|\} & \text{if } i \in S_l \\ 0 & \text{otherwise.} \end{cases}$$



The values $t_1 > t_2$ are chosen such that

$$\|\psi_1\|_2^2 = \|\psi_2\|_2^2 = \|\psi_3\|_2^2.$$

These values exist because $\|\psi_1\|_2, \|\psi_2\|_2, \|\psi_3\|_2$ vary continuously with $t_1, t_2 \in \mathbb{R}_+$. We note that the values t_1 and t_2 are used for analysis only and do not need to be computed in the algorithm. Define the 2-step approximation $y \in \mathbb{R}_+^n$ as

$$y(i) = \begin{cases} t_1 & \text{if } |x(i) - t_1| \leq |x(i) - t_2| \\ t_2 & \text{otherwise.} \end{cases}$$

In other words, y is the closest 2-step approximation of x using values t_1 and t_2. Note that $\psi_l(i) = |x(i) - y(i)|$ for $i \in S_l$, and thus

$$\|x - y\|_2^2 = \sum_{l=1}^{3} \|\psi_l\|^2. \tag{8.1}$$

To bound $R(\psi_l)$, we compare $R(\psi_l)$ and $R(x)$. For the numerator, observe that $|\psi_l(i) - \psi_l(j)| \le |x(i) - x(j)|$ for all i, j by construction. For the denominator, $\|\psi_l\|_2^2 = \frac{1}{3}\|x - y\|_2^2$ by (8.1). Thus,

$$R(\psi_l) = \frac{\sum_{ij \in E} |\psi_l(i) - \psi_l(j)|^2}{d \cdot \|\psi_l\|_2^2} \le \frac{3\sum_{ij \in E} |x(i) - x(j)|^2}{d \cdot \|x - y\|_2^2} = \frac{3R(x) \cdot d \cdot \|x\|_2^2}{d \cdot \|x - y\|_2^2} = \frac{3R(x) \cdot \|x\|_2^2}{\|x - y\|^2}.$$

It follows from Lemma 8.5 that

$$\frac{\lambda_3}{2} \le \max_{1 \le l \le 3} R(\psi_l) \le \frac{3R(x) \cdot \|x\|_2^2}{\|x - y\|_2^2} \quad \Longrightarrow \quad \frac{\|x - y\|_2}{\|x\|_2} \le \sqrt{\frac{6R(x)}{\lambda_3}}.$$

To summarize, the idea is that if x is far from any 2-step function, then $\|x - y\|_2^2$ is large and thus $R(\psi_l)$ is small. Hence, Lemma 8.5 implies that λ_3 is small, proving Lemma 8.3 when $k = 3$.

The General Case

For general $k \ge 3$, we follow the same argument by introducing $k-1$ threshold values $t_1, \ldots, t_{k-1}$ to partition the vertices into k disjoint subsets $S_1, \ldots, S_k$. Define $\psi_l \in \mathbb{R}_+^n$ by $\psi_l(i) = \min_j\{|x(i) - t_j|\}$ if $i \in S_l$ and $\psi_l(i) = 0$ otherwise. Choose $t_1, \ldots, t_{k-1}$ such that $\|\psi_1\|_2 = \cdots = \|\psi_k\|_2$. Define the $(k-1)$-step approximation y as $y(i) = t_j$ if $|x(i) - t_j| = \min_{1 \le j \le k-1}\{|x(i) - t_j|\}$. Following the same calculation, we obtain

$$\lambda_k \le \frac{2k \cdot R(x) \cdot \|x\|_2^2}{\|x - y\|_2^2},$$

but this bound has an extra factor of k compared to Lemma 8.3.

To eliminate this extra factor, we apply a simple trick also used in the higher-order Cheeger inequality in Chapter 7. Instead of using only $k-1$ thresholds, we introduce $2k-1$ thresholds $t_1 > t_2 > \cdots > t_{2k-1}$ to define $2k$ disjointly supported functions $\psi_1, \ldots, \psi_{2k}$. We then sort ψ_l by their numerators so that

$$\sum_{ij \in E} \|\psi_1(i) - \psi_1(j)\|_2^2 \le \sum_{ij \in E} \|\psi_2(i) - \psi_2(j)\|_2^2 \le \cdots \le \sum_{ij \in E} \|\psi_{2k}(i) - \psi_{2k}(j)\|_2^2.$$

By counting the contribution of each edge, we see that

$$\sum_{l=1}^{2k} \sum_{ij \in E} \|\psi_l(i) - \psi_l(j)\|_2^2 \le \sum_{ij \in E} \|x(i) - x(j)\|_2^2.$$

An averaging argument shows that, for $1 \le l \le k$,

$$\sum_{ij \in E} \|\psi_l(i) - \psi_l(j)\|_2^2 \le \frac{1}{k} \sum_{ij \in E} \|x(i) - x(j)\|_2^2.$$

Thus, for the first k functions $\psi_1, \ldots, \psi_k$,

$$R(\psi_l) = \frac{\sum_{ij \in E} |\psi_l(i) - \psi_l(j)|^2}{d \cdot \|\psi_l\|_2^2} \le \frac{\frac{1}{k}\sum_{ij \in E} |x(i) - x(j)|^2}{d \cdot \frac{1}{2k}\|x - y\|_2^2} = \frac{2R(x) \cdot d \cdot \|x\|_2^2}{d \cdot \|x - y\|_2^2} = \frac{2R(x) \cdot \|x\|_2^2}{\|x - y\|^2}.$$

It follows from Lemma 8.5 that

$$\frac{\lambda_k}{2} \le \max_{1 \le l \le k} R(\psi_l) \le \frac{2R(x) \cdot \|x\|_2^2}{\|x - y\|_2^2} \quad \Longrightarrow \quad \frac{\|x - y\|_2}{\|x\|_2} \le 2\sqrt{\frac{R(x)}{\lambda_k}}.$$

We conclude that there exists a $2k$-step function y with the approximation guarantee in Lemma 8.3.

8.3 Rounding $2k$-Step Function

For Lemma 8.4, we follow the same randomized rounding approach as in the proofs of Cheeger's inequality in Theorem 2.2 and the converse of expander mixing lemma in Theorem 4.6.

We first analyze the ideal scenario when x is exactly a $2k$-step function, then extend the idea to the general case where x is close to a $2k$-step function y.

The Ideal Case

Let $y_1 > y_2 > \cdots > y_{2k} > 0$ be the $2k$ distinct values in x, and set $y_{2k+1} = 0$. For a vertex $i \in V$, let l_i be the index such that $x(i) = y_{l_i}$. For an edge $ij \in E$, we use the convention that $l_i \le l_j$.

Intuitively, we aim to return the threshold sets $S_l := \{i \mid x(i) \ge y_l\}$ where $y_l - y_{l+1}$ is large, as edges leaving S_l must have long lengths and so there cannot be too many such edges.

Concretely, we output S_l with probability $(y_l - y_{l+1})^2$, assuming $\sum_{l=1}^{2k}(y_l - y_{l+1})^2 = 1$ by scaling x.

The expected numerator of $\phi(S_l)$ is

$$\mathbb{E}\left[|\delta(S_l)|\right] = \sum_{ij\in E} \Pr(ij \text{ is cut}) = \sum_{ij\in E}\sum_{l=l_i}^{l_j}(y_l - y_{l+1})^2 \le \sum_{ij\in E}(y_{l_i} - y_{l_j})^2 = \sum_{ij\in E}(x(i) - x(j))^2,$$

where we used the inequality $\sum_i a_i^2 \le (\sum_i a_i)^2$ for non-negative a_i's.

The expected denominator of $\phi(S_l)$ is

$$\mathbb{E}\left[d|S_l|\right] = \sum_{i\in V} d \cdot \Pr(i \in S_l) = \sum_{i\in V} d \cdot \sum_{l=l_i}^{2k}(y_l - y_{l+1})^2 \ge \sum_{i\in V} d \cdot \frac{1}{2k}(y_{l_i} - y_{2k+1})^2 = \frac{d}{2k} \cdot \sum_{i\in V} x(i)^2,$$

where the inequality is by Cauchy-Schwarz that $\sum_{i=1}^{j} a_i^2 \ge (\sum_{i=1}^{j} a_i)^2/j$.

Applying Lemma 2.6,

$$\min_l \phi(S_l) = \min_l \frac{|\delta(S_l)|}{d|S_l|} \le \frac{\mathbb{E}\left[|\delta(S_l)|\right]}{\mathbb{E}\left[d|S_l|\right]} \le \frac{\sum_{ij\in E}(x(i) - x(j))^2}{\frac{d}{2k} \cdot \sum_{i\in V} x(i)^2} = 2k \cdot R(x).$$

This proves Lemma 8.4 when x is exactly a $2k$-step function.

The General Case

In the general case, we are only given a vector x that is close to a $2k$-step function y.

There is an interesting way to generalize the above argument. Let $t_1 > t_2 > \cdots > t_{2k} > 0$ be the $2k$ distinct values of y, and set $t_{2k+1} = 0$. We choose $t \in \mathbb{R}_+$ with probability proportional to $\min_{1\le l\le 2k}\{|t - t_l|\}$ (the distance to the closest threshold) and output $S_t = \{i \mid x(i) \ge t\}$. See Figure 8.1. We assume these probabilities sum to one by appropriately scaling x.

Expected Denominator:

$$\mathbb{E}\left[d|S_t|\right] = \sum_{i\in V} d \cdot \Pr(i \in S_t) = \sum_{i\in V} d \cdot [\text{area below } x(i)].$$

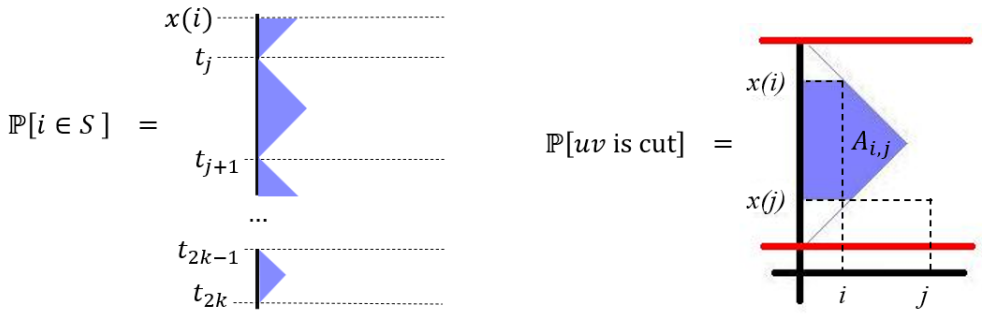


Figure 8.1: The picture on the left is for the analysis of the denominator, where the shaded area represents the probability density function of the random threshold t. The picture on the right is for the analysis of the numerator where A_{ij} is the shaded area.

Let t_j be the largest threshold value smaller than $x(i)$. Then the area below $x(i)$ is the sum of the areas of the triangles below $x(i)$ as shown in Figure 8.1, so

$$\mathbb{E}\left[d|S_t|\right] \geq d \cdot \sum_{i \in V} \left(\frac{1}{4}(x(i) - t_j)^2 + \sum_{l=j}^{2k} \frac{1}{4}(t_l - t_{l+1})^2 \right) \geq d \cdot \sum_{i \in V} \frac{1}{8k}(x(i) - t_{2k+1})^2 = \frac{d}{8k}\|x\|_2^2,$$

where we used Cauchy-Schwarz that $\sum_{i=1}^{j} a_i^2 \geq (\sum_{i=1}^{j} a_i)^2/j$.

Expected Numerator:

$$\mathbb{E}\left[|\delta(S_t)|\right] = \sum_{ij \in E} \Pr(ij \text{ is cut}) = \sum_{ij \in E} [\text{area between } x(i) \text{ and } x(j)] \leq \sum_{ij \in E} \text{area}(A_{ij}),$$

where $\text{area}(A_{ij})$ is as shown in Figure 8.1. This is the largest possible, as having a threshold in between $x(i)$ and $x(j)$ can only make the area smaller. This area is equal to the area of the big triangle minus the two small triangles on the sides. Thus,

$$\begin{aligned}
\mathbb{E}\left[|\delta(S_t)|\right] &\leq \sum_{ij \in E} \Bigg(\underbrace{\frac{1}{4}\big(|y(i) - x(i)| + |x(i) - x(j)| + |x(j) - y(j)|\big)^2}_{\text{big triangle}} - \underbrace{\frac{1}{2}(x(i) - y(i))^2}_{\text{small triangle of } i} - \underbrace{\frac{1}{2}(x(j) - y(j))^2}_{\text{small triangle of } j} \Bigg) \\
&\leq \sum_{ij \in E} \frac{1}{4}\Big[(x(i) - x(j))^2 + 2|x(i) - x(j)| \cdot \big(|x(i) - y(i)| + |x(j) - y(j)|\big)\Big] \\
&= \frac{1}{4}R(x) \cdot d\|x\|_2^2 + \frac{1}{2}\sum_{ij \in E} |x(i) - x(j)| \cdot \big(|x(i) - y(i)| + |x(j) - y(j)|\big) \\
&\leq \frac{1}{4}R(x) \cdot d\|x\|_2^2 + \frac{1}{2}\sqrt{\sum_{ij \in E} \big(x(i) - x(j)\big)^2} \cdot \sqrt{\sum_{ij \in E} \big(|x(i) - y(i)| + |x(j) - y(j)|\big)^2} \\
&\leq \frac{1}{4}R(x) \cdot d\|x\|_2^2 + \frac{1}{2}\sqrt{R(x) \cdot d\|x\|_2^2} \cdot \sqrt{\sum_{ij \in E} \big(2(x(i) - y(i))^2 + 2(x(j) - y(j))^2\big)} \\
&= \frac{1}{4}R(x) \cdot d\|x\|_2^2 + \frac{1}{2}\sqrt{R(x) \cdot d\|x\|_2^2} \cdot \sqrt{\sum_{i \in V} 2d \cdot (x(i) - y(i))^2} \\
&= \frac{d}{4}R(x) \cdot \|x\|_2^2 + \frac{d}{\sqrt{2}}\sqrt{R(x)} \cdot \|x\|_2 \cdot \|x - y\|_2,
\end{aligned}$$

where we used Cauchy-Schwarz once and the inequality $(a+b)^2 \le 2a^2 + 2b^2$ twice.

Combining these inequalities and applying Lemma 2.6,

$$\min_l \phi(S_l) = \min_l \frac{|\delta(S_l)|}{d|S_l|} \le \frac{\mathbb{E}\left[|\delta(S_l)|\right]}{\mathbb{E}\left[d|S_l|\right]} \le 2k \cdot R(x) + 4\sqrt{2k} \cdot \sqrt{R(x)} \cdot \frac{\|x-y\|_2}{\|x\|_2}.$$

This completes the proof of Lemma 8.4.

8.4 Problems

Problem 8.6 (Tight Example for Theorem 8.1)**.** *Provide an example where the improved Cheeger's inequality is tight up to a constant factor for any k.*

Problem 8.7 (Improved Cheeger Inequality Using k-Way Edge-Conductance [KLL17])**.** *Using the results in this chapter, prove that the spectral partitioning algorithm outputs a set S satisfying*

$$\phi(S) \lesssim \frac{k\lambda_2}{\phi_k},$$

where ϕ_k is the k-way edge-conductance in Definition 7.1.

References

[BL12] Yonatan Bilu and Nathan Linial. Are stable instances easy? *Combinatorics, Probability and Computing*, 21(5):643–660, 2012. 101

[BLR10] Punyashloka Biswal, James R Lee, and Satish Rao. Eigenvalue bounds, spectral partitioning, and metrical deformations via flows. *Journal of the ACM (JACM)*, 57(3):1–23, 2010. 101

[Bop87] Ravi B Boppana. Eigenvalues and graph bisection: An average-case analysis. In *28th Annual Symposium on Foundations of Computer Science*, pages 280–285. IEEE, 1987. 101

[DLS12] Amit Daniely, Nati Linial, and Michael Saks. Clustering is difficult only when it does not matter. *arXiv preprint arXiv:1205.4891*, 2012. 101

[KLL+13] Tsz Chiu Kwok, Lap Chi Lau, Yin Tat Lee, Shayan Oveis Gharan, and Luca Trevisan. Improved cheeger's inequality: analysis of spectral partitioning algorithms through higher order spectral gap. In *Symposium on Theory of Computing Conference, STOC*, pages 11–20. ACM, 2013. 8, 101

[KLL17] Tsz Chiu Kwok, Lap Chi Lau, and Yin Tat Lee. Improved cheeger's inequality and analysis of local graph partitioning using vertex expansion and expansion profile. *SIAM J. Comput.*, 46(3):890–910, 2017. 107, 135, 142

[McS01] Frank McSherry. Spectral partitioning of random graphs. In *Proceedings 42nd IEEE Symposium on Foundations of Computer Science*, pages 529–537. IEEE, 2001. 101

[ST07] Daniel A Spielman and Shang-Hua Teng. Spectral partitioning works: Planar graphs and finite element meshes. *Linear Algebra and its Applications*, 421(2-3):284–305, 2007. 101

[VL07] Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17:395–416, 2007. 101

Chapter 9

Fastest Mixing Time, Vertex Expansion, and Reweighted Eigenvalues

Cheeger's inequality connects (i) the edge conductance of a graph, (ii) the second eigenvalue, and (iii) the mixing time of random walks.

In this chapter, we introduce a new Cheeger-type inequality that connects (i) the vertex expansion of a graph, (ii) the second *reweighted* eigenvalue, and (iii) the *fastest* mixing time of random walks.

Towards the end, we discuss how to define the k-th reweighted eigenvalue and use it to establish analogs of higher-order Cheeger inequality and improved Cheeger inequality for vertex expansion.

Finally, we discuss conjectures on reweighting that could have significant implications for approximation algorithms.

9.1 Definitions and Statements

Fastest Mixing Time

The fastest mixing time problem was proposed in [BDX04]. In this problem, we are given an undirected graph $G = (V = [n], E)$ and a target probability distribution $\vec{\pi} : V \to \mathbb{R}$. The task is to assign a transition probability $P(i, j)$ on each edge $ij \in E(G)$ such that the stationary distribution of random walks with transition matrix P is $\vec{\pi}$. The objective is to find a transition matrix P that minimizes the mixing time to $\vec{\pi}$, among all transition matrices with stationary distribution $\vec{\pi}$.

Reweighted Eigenvalue

From Chapter 3, the mixing time to the stationary distribution is approximately inversely proportional to the spectral gap $1 - \alpha_2(P)$ of the transition matrix P, where $1 = \alpha_1(P) \geq \alpha_2(P) \geq \cdots \geq \alpha_n(P) \geq -1$ are the eigenvalues of P. The fastest mixing time problem was thus formulated by the maximum spectral gap achievable through a "reweighting" P of the adjacency matrix of G.

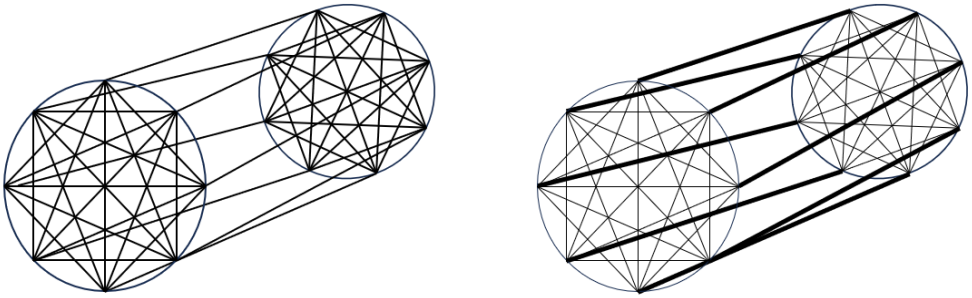


Figure 9.1: An n-regular graph with two cliques connected by a perfect matching. On the left, where each edge has probability $\frac{1}{n}$, the mixing time is $\Theta(n)$. On the right, where each matching edge has probability $\frac{1}{2}$ and each clique edge has probability $\frac{1}{2(n-1)}$, the mixing time is $\Theta(1)$.

Definition 9.1 (Maximum Reweighted Second Eigenvalue [BDX04])**.** *Given an undirected graph $G = (V, E)$ and a probability distribution $\vec{\pi}$ on V, the maximum reweighted second eigenvalue is*

$$\begin{aligned} \lambda_2^*(G) \;:=\; \max_{P \geq 0} \quad & 1 - \alpha_2(P) & \\ \text{subject to} \quad & P(i,j) = 0 & \forall ij \notin E \\ & \sum_{j \in V} P(i,j) = 1 & \forall i \in V \\ & \vec{\pi}(i) \cdot P(i,j) = \vec{\pi}(j) \cdot P(j,i) & \forall ij \in E. \end{aligned}$$

The last constraint is called the time reversible condition, ensuring that the stationary distribution of P is $\vec{\pi}$ and that the transition matrix corresponds to random walks on an undirected graph.

Note that $\lambda_2^*(G) = \max_{P \geq 0}(1 - \alpha_2(P)) = \max_{P \geq 0} \lambda_2(I - P)$, which is the maximum reweighted second smallest eigenvalue of the normalized Laplacian matrix of G, subject to the constraints above.

We assume that the graph has a self-loop at each vertex, so that the problem in Definition 9.1 is always feasible. In the context of Markov chains, this corresponds to allowing a non-negative holding probability at each vertex.

This optimization problem can be expressed as a semidefinite program and thus $\lambda_2^*(G)$ can be computed in polynomial time [BDX04].

Cheeger Inequality for Vertex Expansion

It was observed in [Roc05] that the $\vec{\pi}$-weighted vertex expansion is an upper bound on $\lambda_2^*(G)$.

Definition 9.2 (Weighted Vertex Expansion)**.** *Let $G = (V, E)$ be an undirected graph and $\vec{\pi}$ be a probability distribution on V. The weighted vertex expansion of a set $S \subseteq V$ and of a graph G are defined as*

$$\psi(S) := \frac{\vec{\pi}(\partial S)}{\vec{\pi}(S)} \quad \text{and} \quad \psi(G) := \min\left\{1, \min_{S \subseteq V : 0 < \vec{\pi}(S) \leq 1/2} \psi(S)\right\},$$

where $\vec{\pi}(S) := \sum_{i \in S} \vec{\pi}(i)$. When $\vec{\pi}$ is the uniform distribution, $\psi(S)$ is the usual vertex expansion $|\partial S|/|S|$ in Definition 2.11.

Recently, the following Cheeger-type inequality established that small vertex expansion is qualitatively the only obstruction to large $\lambda_2^*(G)$.

Theorem 9.3 (Cheeger Inequality for Vertex Expansion [OTZ24, KLT22, JPV23])**.** *For any undirected graph $G = (V, E)$ with maximum degree d and any probability distribution $\vec{\pi}$,*

$$\lambda_2^*(G) \lesssim \psi(G) \lesssim \sqrt{\lambda_2^*(G) \cdot \log d}.$$

Comparing to Theorem 2.12, the reweighted second eigenvalue provides a much tighter relationship to the vertex expansion than the ordinary second eigenvalue. The bound is optimal up to a constant factor, as demonstrated by the discretized sphere example in [KLT22] and also the conditional hardness result in [LRV13].

By applying the mixing time bound from (3.3), a direct consequence of Theorem 9.3 is a combinatorial characterization of the fastest mixing time of a graph.

Corollary 9.4 (Fastest Mixing Time and Vertex Expansion)**.** *For any undirected graph $G = (V, E)$ with maximum degree d and any distribution $\vec{\pi}$,*

$$\frac{1}{\psi(G)} \lesssim \tau^*(G) \lesssim \frac{\log d \cdot \log(1/\pi_{\min})}{\psi^2(G)},$$

where $\tau^*(G)$ denotes the fastest mixing time of random walks and $\pi_{\min} = \min_{i \in V} \vec{\pi}(i)$.

The main goal of this chapter is to prove Theorem 9.3 in the case when $\vec{\pi}$ is the uniform distribution. In this special case, the transition matrix P is symmetric by the time reversible condition in Definition 9.1.

9.2 Semidefinite Program for Fastest Mixing Time

In this section, we derive the mathematical programs needed for the proof of Theorem 9.3. We begin with a semidefinite programming (SDP) relaxation of the ordinary second eigenvalue. Using this relaxation and Von Neumann's minimax theorem, we construct the dual SDP corresponding to the formulation in Definition 9.1. Finally, we reformulate the dual program into a directed graph formulation, which will be used in the proof of Theorem 9.3.

Semidefinite Program for Second Eigenvalue

To formulate the second eigenvalue using an SDP, we express it as the Rayleigh quotient of an n-dimensional embedding and show that this provides an exact relaxation, as stated in the following lemma. This approach is analogous to the one introduced in Chapter 7, where we analyzed the Rayleigh quotient of a k-dimensional embedding and related it to the Rayleigh quotient of a 1-dimensional embedding.

Lemma 9.5 (Semidefinite Program for the Second Eigenvalue)**.** *Let $P \in \mathbb{R}^{n \times n}$ be a transition matrix of a graph $G = (V = [n], E)$ satisfying the constraints in Definition 9.1. Then*

$$1 - \alpha_2(P) = \min_{f: V \to \mathbb{R}^n,\ \sum_{i \in V} f(i) = 0} \frac{\sum_{ij \in E} \|f(i) - f(j)\|_2^2 \cdot P(i,j)}{\sum_{i \in V} \|f(i)\|_2^2}.$$

Proof. In the special case where the stationary distribution $\vec{\pi}$ is the uniform distribution $\vec{1}/n$, the transition matrix P symmetric, corresponding to the normalized adjacency matrix of a weighted graph with weighted degree one for each vertex. From Section 1.3, we have $1-\alpha_2(P) = \lambda_2(I-P)$, where $I-P$ is the normalized Laplacian matrix of the weighted graph. By Lemma 2.3,

$$1-\alpha_2(P) = \lambda_2(I-P) = \min_{f:V\to\mathbb{R},\ \sum_{i\in V} f(i)=0} \frac{\sum_{ij\in E} |f(i)-f(j)|^2 \cdot P(i,j)}{\sum_{i\in V} f(i)^2}.$$

This is nearly identical to the statement of the lemma, except that $f: V \to \mathbb{R}$ instead of $f: V \to \mathbb{R}^n$.

Clearly, allowing functions $f: V \to \mathbb{R}^n$ expands the feasible set, which could only result in a smaller optimal value. Conversely, given any solution $f: V \to \mathbb{R}^n$, we can apply Spielman's favorite inequality (Lemma 2.6) as in Lemma 7.10 to show that at least one coordinate of f yields a one-dimensional function $f: V \to \mathbb{R}$ with an objective value no larger than that of the n-dimensional solution. Therefore, the relaxation from $f: V \to \mathbb{R}$ to $f: V \to \mathbb{R}^n$ is exact. □

To see that the formulation in Lemma 9.5 is an SDP, recall that a positive semidefinite matrix Y can be written as $F^\top F$ where $F \in \mathbb{R}^{n\times n}$, by Fact A.9. We associate the i-th column of F with $f(i)$, so that $Y_{i,j} = \langle f(i), f(j)\rangle$ for all $i,j \in V$. Then, the formulation in Lemma 9.5 can be rewritten as

$$\begin{aligned}
\min \quad & \sum_{ij\in E} (Y_{i,i} - 2Y_{i,j} + Y_{j,j}) \cdot P(i,j) && \text{(equivalent to } \sum_{ij\in E} \|f(i)-f(j)\|_2^2 \cdot P(i,j)\text{)} \\
\text{s.t.} \quad & \sum_{i\in V} Y_{i,i} = 1 && \text{(equivalent to } \sum_{i\in V} \|f(i)\|_2^2 = 1\text{)} \\
& \sum_{i,j\in V} Y_{i,j} = 0 && \text{(equivalent to } \sum_{i\in V} f(i) = 0\text{)} \\
& Y \succcurlyeq 0, && \text{(ensuring } Y_{i,j} = \langle f(i), f(j)\rangle \text{ for all } i,j \in V\text{)}
\end{aligned}$$

Note that the objective function is equal to the numerator of the objective function in Lemma 9.5, as we normalize the denominator of the objective function in Lemma 9.5 to one. The second constraint is equivalent to the constraint $\|\sum_{i\in V} f(i)\|_2^2 = 0$, which is equivalent to $\sum_{i\in V} f(i) = 0$. Thus, the program in Lemma 9.5 can be written as optimizing a linear function with linear constraints over the entries of a positive semidefinite matrix Y. This confirms that it is a semidefinite program and can be solved in polynomial time.

The Dual Semidefinite Program of Definition 9.1

The reason that we use the above semidefinite program for the second eigenvalue instead of the one-dimensional spectral program is that the set of feasible solutions is a convex set, which is not the case for the spectral program. The convexity allows us to apply Von-Neumann's minimax theorem to derive the dual program.

Theorem 9.6 (Von Neumann's Minimax Theorem)**.** *Let X, Y be compact convex sets. If h is a real-valued continuous function on $X \times Y$ such that $h(x,\cdot)$ concave on Y for all $x \in X$ and $h(\cdot, y)$ convex on X for all $y \in Y$, then*

$$\min_{x\in X}\max_{y\in Y} h(x,y) = \max_{y\in Y}\min_{x\in X} h(x,y).$$

The following dual program was derived in [Roc05] (without using the minimax theorem).

Proposition 9.7 (Dual Program for Fastest Mixing [Roc05]). *Given an undirected graph $G = (V = [n], E)$ with a self-loop on each vertex and the uniform distribution $\pi = \vec{1}/n$ on V, the following semidefinite program $\gamma(G)$ is dual to the primal program $\lambda_2^*(G)$ in Definition 9.1 where*

$$\begin{aligned}
\gamma(G) := \min_{f:V\to\mathbb{R}^n,\ g:V\to\mathbb{R}_+} \quad & \sum_{i\in V} g(i) \\
\text{subject to} \quad & \sum_{i\in V} \|f(i)\|_2^2 = 1 \\
& \sum_{i\in V} f(i) = \vec{0} \\
& g(i) + g(j) \geq \|f(i) - f(j)\|_2^2 \qquad \forall ij \in E.
\end{aligned}$$

Strong duality $\lambda_2^*(G) = \gamma(G)$ holds.

Proof. For a fixed transition matrix P, by Lemma 9.5,

$$1 - \alpha_2(P) = \min_{f:V\to\mathbb{R}^n,\ \sum_{i\in V} f(i)=\vec{0}} \frac{\sum_{ij\in E}\|f(i) - f(j)\|_2^2 \cdot P(i,j)}{\sum_{i\in V}\|f(i)\|_2^2}.$$

The maximum reweighted second eigenvalue in Definition 9.1 can thus be formulated as

$$\begin{aligned}
\lambda_2^*(G) = \max_{P\geq 0}(1 - \alpha_2(P)) \quad = \quad & \max_{P\geq 0} \min_{f:V\to\mathbb{R}^n,\ \sum_{i\in V} f(i)=\vec{0}} \frac{\sum_{ij\in E}\|f(i) - f(j)\|_2^2 \cdot P(i,j)}{\sum_{i\in V}\|f(i)\|_2^2} \\
\text{subject to} \quad & P(i,j) = 0 \quad \forall ij \notin E \\
& \sum_{j\in V} P(i,j) = 1 \quad \forall i \in V \\
& P = P^\top.
\end{aligned}$$

Observe that the assumptions in the Von Neumann minimax Theorem 9.6 are satisfied, and so we can switch the order of the max and the min to obtain the dual program:

$$\gamma(G) \quad := \quad \min_{f:V\to\mathbb{R}^n,\ \sum_{i\in V} f(i)=\vec{0}} \max_{P\geq 0} \frac{\sum_{ij\in E}\|f(i) - f(j)\|_2^2 \cdot P(i,j)}{\sum_{i\in V}\|f(i)\|_2^2},$$

subjected to the same constraints on P as above.

For a fixed embedding $f : V \to \mathbb{R}^n$, note that the inner maximization problem is a linear program over the entries of P, so we can reformulate it using linear programming duality to obtain

$$\begin{aligned}
\gamma(G) \quad = \quad & \min_{f:V\to\mathbb{R}^n,\ \sum_{i\in V} f(i)=\vec{0}} \min_{g\geq 0} \sum_{i\in V} g(i) \\
\text{subject to} \quad & g(i) + g(j) \geq \frac{\|f(i) - f(j)\|^2}{\sum_{i\in V}\|f(i)\|^2} \quad \forall ij \in E,
\end{aligned}$$

where $g(i)$ is a dual variable for the constraint $\sum_{j\in V} P(i,j) = 1$. Note that the constraint $g \geq 0$ comes from the assumption that there is a self-loop at each vertex. Normalizing so that $\sum_{i\in V}\|f(i)\|^2 = 1$ gives the formulation in the lemma. □

We remark that the self-loop assumption ensures that the dual program has the inequality $g \geq 0$. This is a crucial but subtle condition that will be used only once, and we will explicitly indicate when it is needed.

Directed Graph Formulation

We further simplify the dual program in Proposition 9.7 by eliminating the variables $g : V \to \mathbb{R}_+$. The observation is that $g(i)+g(j) \geq \|f(i)-f(j)\|_2^2$ implies that $\max\{g(i), g(j)\} \geq \frac{1}{2}\|f(i)-f(j)\|_2^2$. Thus, we assign a direction to each edge ij so that only the tail is "responsible" for the edge, allowing us to eliminate the variables $g(i)$ from the program.

Lemma 9.8 (Directed Dual Program for Fastest Mixing)**.** *Let $G = (V = [n], E)$ be an undirected graph and let $\vec{E}$ be an orientation of the edges in E. Consider the directed dual program*

$$\begin{aligned} \vec{\gamma}(G) = \min_{f:V\to\mathbb{R}^n} \min_{\vec{E}} \quad & \sum_{i\in V} \max_{j:ij\in\vec{E}} \|f(i)-f(j)\|_2^2 \\ \text{subject to} \quad & \sum_{i\in V} \|f(i)\|_2^2 = 1 \\ & \sum_{i\in V} f(i) = \vec{0}. \end{aligned}$$

The inequality $\vec{\gamma}(G) \leq 2\gamma(G)$ holds.

Proof. Given an optimal solution $f : V \to \mathbb{R}^n$ and $g : V \to R_+$ to $\gamma(G)$, then f and $g' := 2g$ from a feasible solution to the following program:

$$\begin{aligned} \gamma'(G) := \min_{f:V\to\mathbb{R}^n,\ g':V\to\mathbb{R}_+} \quad & \sum_{i\in V} g'(i) \\ \text{subject to} \quad & \sum_{i\in V} \|f(i)\|_2^2 = 1 \\ & \sum_{i\in V} f(i) = \vec{0} \\ & \max\{g'(i), g'(j)\} \geq \|f(i)-f(j)\|_2^2 \qquad \forall ij \in E. \end{aligned}$$

It follows that $\gamma'(G) \leq 2\gamma(G)$.

Define an orientation $\vec{E}$ of E such that each directed edge $ij \in \vec{E}$ satisfies $g'(i) \geq \|f(i)-f(j)\|_2^2$. Then, $g'(i) \geq \max_{j:ij\in\vec{E}} \|f(i)-f(j)\|_2^2$ for every vertex i, where we used the assumption that $g(i) \geq 0$ for those vertices with outdegree zero. This gives us a solution to

$$\begin{aligned} \min_{f:V\to\mathbb{R}^n,\ g':V\to\mathbb{R}_+} \quad & \sum_{i\in V} g'(i) \\ \text{subject to} \quad & \sum_{i\in V} \|f(i)\|_2^2 = 1 \\ & \sum_{i\in V} f(i) = \vec{0} \\ & g'(i) \geq \max_{j:ij\in\vec{E}} \|f(i)-f(j)\|_2^2 \qquad \forall i \in V. \end{aligned}$$

Setting $g'(i) := \max_{j:ij\in\vec{E}} \|f(i)-f(j)\|_2^2$ satisfies all the constraints and does not increase the objective value of $\gamma'(G)$. This eliminates the variables $g'(i)$ from the program, giving us a solution $f, \vec{E}$ to $\vec{\gamma}(G)$ with objective value at most $\gamma'(G) \leq 2\gamma(G)$. □

The assumption $g \geq 0$ is subtly used in the proof above. It is instructive to construct a counterexample to $\vec{\gamma}(G) \leq 2\gamma(G)$ when $g : V \to \mathbb{R}$ is allowed to have negative entries.

9.3 Easy Direction by Reduction

There are two ways to prove the easy direction of Theorem 9.3.

One way is to show that $\gamma(G)$ is a relaxation of $\psi(G)$, by plugging in a binary solution defined by a set S minimizing $\psi(S)$ to upper bound $\gamma(G)$. We leave it as an exercise to the reader.

Exercise 9.9 (Easy Direction for Cheeger Inequality for Vertex Expansion)**.** *For any undirected graph $G = (V = [n], E)$ and the uniform distribution $\vec{\pi} = \vec{1}/n$,*

$$\lambda_2^*(G) = \gamma(G) \leq 2\psi(G).$$

Another way is to understand the easy direction directly, as using the edge conductance of a 1-regular reweighted graph H of G to certify the vertex expansion of G.

Proposition 9.10 (Vertex Expansion through Edge Conductance)**.** *Let H be a reweighted graph of $G = (V, E)$ with weighted adjacency matrix P satisfying the constraints in the primal program in Definition 9.1 when $\vec{\pi}$ is the uniform distribution. Then $\phi(H) \leq \psi(G)$, where $\phi(H)$ is the weighted edge conductance of H.*

Proof. As the matrix P satisfies the constraints in Definition 9.1 and $\vec{\pi}$ is the uniform distribution, the graph H is a weighted 1-regular graph, so its weighted edge conductance is given by

$$\phi(H) = \min_{S:0<\mathrm{vol}_w(S)\leq\frac{1}{2}\mathrm{vol}_w(V)} \frac{w(\delta(S))}{\mathrm{vol}_w(S)} = \min_{S:0<|S|\leq\frac{1}{2}|V|} \frac{w(\delta(S))}{|S|},$$

where we denote $w(u, v) = P(u, v)$ as the weight of an edge and $w(\delta(S)) := \sum_{e\in\delta(S)} w(e)$.

An important observation is that $w(\delta(S)) \leq \sum_{v\in\partial(S)} \deg_w(v) = |\partial(S)|$, because each edge in $\delta(S)$ has an endpoint in $\partial(S)$ and each vertex in $\partial(S)$ has weighted degree one. Therefore,

$$\phi(H) = \min_{S:0<|S|\leq\frac{1}{2}|V|} \frac{w(\delta(S))}{|S|} \leq \min_{S:0<|S|\leq\frac{1}{2}|V|} \frac{|\partial(S)|}{|S|} = \psi(G).$$

□

Proposition 9.10 shows that the edge conductance of any 1-regular reweighted graph H of G is a lower bound on the vertex expansion of G. By the easy direction of Cheeger's inequality in Theorem 2.2, the edge conductance of the reweighted graph H is lower bounded by the second eigenvalue of the normalized Laplacian matrix of H. To prove the best lower bound on the vertex expansion of G, we thus maximize the second eigenvalue of the edge reweighted graph H. Therefore,

$$\lambda_2^*(G) = \max_{H:H \text{ is a reweighting of } G} \lambda_2(H) \leq \max_{H:H \text{ is a reweighting of } G} 2\phi(H) \leq 2\psi(G).$$

To summarize, a more direct and intuitive way to understand the easy direction of the Cheeger inequality for vertex expansion in Theorem 9.3 is as a method to certify the vertex expansion of a graph through a reduction to the edge conductance and spectral gap of a 1-regular reweighted graph. This perspective plays an important role in the next chapter.

Interestingly, the hard direction shows that there is always a reweighted graph so that this reduction works well to certify the vertex expansion.

9.4 Hard Direction by Dimension Reduction and Rounding

There are two main steps in the hard direction. The first step is to project the n-dimensional SDP solution to a 1-dimensional SDP solution.

Definition 9.11 (One-Dimensional Directed Dual Program for Fastest Mixing)**.** *Let $G = (V, E)$ be an undirected graph and let $\vec{E}$ be an orientation of the edges in E. The 1-dimensional directed dual program is*

$$\begin{aligned} \vec{\gamma}^{(1)}(G) := \min_{f:V\to\mathbb{R}} \min_{\vec{E}} \quad & \sum_{i\in V} \max_{j:ij\in\vec{E}} (f(i)-f(j))^2 \\ \text{subject to} \quad & \sum_{i\in V} f(i)^2 = 1 \\ & \sum_{i\in V} f(i) = 0. \end{aligned}$$

Using the Johnson-Lindenstrauss theorem for dimension reduction, one can prove that $\vec{\gamma}^{(1)}(G) \lesssim \log n \cdot \vec{\gamma}(G)$. The following improvement is obtained by using a Gaussian projection technique developed in [LRV13]. The assumption about the maximum degree is only used in this step.

Proposition 9.12 (Dimension Reduction for Directed Dual Program)**.** *For any undirected graph $G = (V, E)$ with maximum degree d,*

$$\vec{\gamma}^{(1)}(G) \lesssim \log d \cdot \vec{\gamma}(G).$$

The second step is to round the fractional solution to $\vec{\gamma}^{(1)}(G)$ to a subset $S \subseteq V$ with $\psi(S) \lesssim \sqrt{\vec{\gamma}^{(1)}(G)}$, using a relatively standard threshold rounding argument as in Chapter 2.

Proposition 9.13 (Threshold Rounding for Vertex Expansion)**.** *For any undirected graph $G = (V, E)$, there is a subset $S \subseteq V$ satisfying*

$$\psi(S) \lesssim \sqrt{\vec{\gamma}^{(1)}(G)}.$$

Assuming these two propositions, the hard direction of Theorem 9.3 is established as follows:

$$\psi(S) \lesssim \sqrt{\vec{\gamma}^{(1)}(G)} \lesssim \sqrt{\log d \cdot \vec{\gamma}(G)} \lesssim \sqrt{\log d \cdot \gamma(G)} = \sqrt{\log d \cdot \lambda_2^*(G)},$$

where the inequality $\vec{\gamma}(G) \le 2\gamma(G)$ is from Lemma 9.8 and the equality $\gamma(G) = \lambda_2^*(G)$ is by strong duality in Proposition 9.7. This completes the proof of the hard direction of Theorem 9.3.

It remains to prove Proposition 9.12 and Proposition 9.13 in the next two subsections.

Dimension Reduction

We prove Proposition 9.12 in this section. We first obtain a $O(\log n)$ bound using the Johnson-Lindenstrauss theorem, and then prove the $O(\log d)$ bound using a Gaussian projection method.

$O(\log n)$ **Approximation:** An important result in metric embedding is the dimension reduction theorem by Johnson and Lindenstrauss, which states that any set of n high-dimensional vectors can be projected into $O(\log n)$-dimensional space while approximately preserving their pairwise Euclidean distances.

Theorem 9.14 (Johnson-Lindenstrauss Theorem)**.** *Given $0 < \epsilon < 1$, a set X of n points in $\mathbb{R}^m$, there exists a linear map $A : \mathbb{R}^m \to \mathbb{R}^k$ for $k \lesssim \ln(n)/\epsilon^2$ such that for all $u, v \in X$,*

$$(1-\epsilon) \cdot \|u - v\|_2^2 \leq \|Au - Av\|_2^2 \leq (1+\epsilon) \cdot \|u - v\|_2^2.$$

In particular, if each entry of Q is defined as $Q(i,j) := g_{i,j}/\sqrt{k}$ where $g_{i,j}$ are independent random variables drawn from the normal distribution $N(0,1)$, then the random matrix $Q \in \mathbb{R}^{k \times m}$ satisfies the above property with probability at least $1 - 1/n$.

First, we apply the Johnson-Lindenstrauss lemma to the n-dimensional solution f to $\vec{\gamma}(G)$ in Lemma 9.8 to obtain a $O(\log n)$-dimensional solution f', which preserve distances and lengths up to a constant factor. Then, by selecting the "best" coordinate in f' as a solution to $\vec{\gamma}^{(1)}(G)$ in Definition 9.11, we can establish the bound $\vec{\gamma}^{(1)}(G) \lesssim \log n \cdot \vec{\gamma}(G)$. See Problem 9.30.

$O(\log d)$ **Approximation:** The Gaussian projection method projects the SDP solution into a 1-dimensional solution by setting $y(i) = \langle f(i), g \rangle$, where $g \sim N(0,1)^n$ is a random Gaussian vector with independent entries. This is a common technique in designing rounding algorithms for semidefinite programming relaxations.

For the analysis, we used the following basic properties of Gaussian random variables.

Fact 9.15 ([LRV13], Fact B.3)**.** *Let $Y_1, Y_2, \ldots, Y_d$ be d Gaussian random variables with mean 0 and variance at most σ^2. Let Y be the random variable defined as $Y := \max\{Y_i \mid i \in [d]\}$. Then*

$$\mathbb{E}\left[Y^2\right] \leq 4\sigma^2 \log d.$$

Fact 9.16 ([LRV13], Lemma 9.8)**.** *Suppose $Y_1, \ldots, Y_d$ are Gaussian random variables such that $\mathbb{E}\left[\sum_{i=1}^d Y_i^2\right] = 1$. Then*

$$\Pr\left[\sum_{i=1}^d Y_i^2 \geq \frac{1}{2}\right] \geq \frac{1}{12}.$$

We are ready to prove Proposition 9.12.

Proof of Proposition 9.12. Let $f : V \to \mathbb{R}^n$ and $\vec{E}$ be a solution to $\vec{\gamma}(G)$ as stated in Lemma 9.8. We construct a 1-dimensional solution $y : V \to \mathbb{R}$ to $\vec{\gamma}^{(1)}(G)$ by setting $y(i) = \langle f(i), g \rangle$, where $g \sim N(0,1)^n$ is a Gaussian random vector with independent entries.

First, consider the expected objective value of y to $\vec{\gamma}^{(1)}(G)$. For each max term in the summand,

$$\mathbb{E}\left[\max_{j:ij \in \vec{E}} \big(y(i) - y(j)\big)^2\right] = \mathbb{E}\left[\max_{j:ij \in \vec{E}} \big\langle f(i) - f(j), g \big\rangle^2\right] \leq 4 \max_{j:ij \in \vec{E}} \|f(i) - f(j)\|_2^2 \cdot \log d,$$

where the last inequality follows from Fact 9.15 on the centered Gaussian random variables $\langle f(i) - f(j), g \rangle$ with variance $\|f(i) - f(j)\|^2$. By linearity of expectation, the expected objective value of

$\vec{\gamma}^{(1)}(G)$ is

$$\mathbb{E}\left[\sum_{i\in V}\max_{j:ij\in\vec{E}}\big(y(i)-y(j)\big)^2\right] \le 4\log d\cdot\sum_{i\in V}\max_{j:ij\in\vec{E}}\|f(i)-f(j)\|^2 = 4\log d\cdot\vec{\gamma}(G).$$

Therefore, by Markov's inequality,

$$\Pr\left[\sum_{i\in V}\max_{j:ij\in\vec{E}}\big(y(i)-y(j)\big)^2 \ge 96\log d\cdot\vec{\gamma}(G)\right] \le \frac{1}{24}.$$

Next, by applying Fact 9.16 with $Y_i = y(i)$, it follows that

$$\mathbb{E}\left[\sum_{i\in V} y(i)^2\right] = \sum_{i\in V}\|f(i)\|^2 = 1 \quad\Longrightarrow\quad \Pr\left[\sum_{i\in V} y(i)^2 \ge \frac{1}{2}\right] \ge \frac{1}{12}.$$

Finally, since $\sum_{i\in V} f(i) = \vec{0}$, it also holds that

$$\sum_{i\in V} y(i) = \sum_{i\in V}\langle f(i), g\rangle = \left\langle\sum_{i\in V} f(i), g\right\rangle = 0.$$

Therefore, with probability at least $1/24$, all of these events hold simultaneously.

Since the second event $\sum_{i\in V} y(i)^2 \ge 1/2$ holds, we can rescale y by a factor of at most $\sqrt{2}$ to satisfy the constraint $\sum_{i\in V} y(i)^2 = 1$, while ensuring the objective value is at most $192\log d\cdot\vec{\gamma}(G)$. Hence, we conclude that $\vec{\gamma}^{(1)}(G) \lesssim \vec{\gamma}(G)\cdot\log d$. □

Threshold Rounding

The proof of Proposition 9.13 is similar to that of the hard direction of Cheeger's inequality in Chapter 2. So we just describe the main steps and leave the details to the exercises.

The first step is a truncation step to to ensure that the output set S satisfies $|S| \le |V|/2$.

Exercise 9.17 (Truncation). *Let $G = (V = [n], E)$ be an undirected graph and $\vec{\pi} = \vec{1}/n$ be the uniform distribution on V. Given a solution to $\vec{\gamma}^{(1)}(G)$, there is a solution $x \in \mathbb{R}^n_+$ and $\vec{E}$ with $|\operatorname{supp}(x)| \le n/2$ and*

$$\frac{\sum_{i\in V}\max_{j:ij\in\vec{E}}(x(i)-x(j))^2}{\sum_{i\in V} x(i)^2} \lesssim \vec{\gamma}^{(1)}(G).$$

For the threshold rounding step, we define the appropriate vertex boundary $\overset{\leftrightarrow}{\partial} S$ for the analysis of the directed program $\vec{\gamma}^{(1)}(G)$. Unlike ∂S, $\overset{\leftrightarrow}{\partial} S$ may contain vertices in S. A good interpretation is that $\overset{\leftrightarrow}{\partial} S$ serves as a vertex cover of the edge boundary $\delta(S)$ in the undirected sense.

Definition 9.18 (Vertex Cover Boundary and Expansion). *Let $G = (V, \vec{E})$ be a directed graph. For $S \subseteq V$, define the vertex-cover boundary and the vertex-cover expansion as*

$$\overset{\leftrightarrow}{\partial} S := \big\{i \in S \mid \exists j \notin S \text{ with } ij \in \vec{E}\big\} \cup \big\{i \notin S \mid \exists j \in S \text{ with } ij \in \vec{E}\big\} \quad \text{and} \quad \overset{\leftrightarrow}{\psi}(S) := |\overset{\leftrightarrow}{\partial} S|/|S|.$$

The main step is to prove that the standard threshold rounding will find a set S with small vertex-cover expansion $\overset{\leftrightarrow}{\psi}(S)$. See Problem 9.31.

Proposition 9.19 (Threshold Rounding for $\gamma(G)$)**.** *Let $G = (V = [n], E)$ be an undirected graph and $\vec{\pi} = \vec{1}/n$ be the uniform distribution on V. Given a solution $x \in \mathbb{R}^n_+$ and $\vec{E}$ with*

$$\frac{\sum_{v \in V} \pi(v) \max_{u:u\to v}(x(u) - x(v))^2}{\sum_{v\in V} \pi(v) x(v)^2} \lesssim \vec{\gamma}^{(1)}(G),$$

there exists a set $S \subseteq \operatorname{supp}(x)$ with $\overset{\leftrightarrow}{\psi}(S) \lesssim \sqrt{\vec{\gamma}^{(1)}(G)}$.

Finally, given a set S with small vertex-cover expansion $\overset{\leftrightarrow}{\psi}(S)$, we find a set $S' \subseteq S$ with small vertex expansion $\psi(S')$. See Problem 9.32.

Lemma 9.20 (Postprocessing)**.** *Let $G = (V, \vec{E})$ be a directed graph. Given a set S with $\vec{\psi}(S) < 1/2$, there is a set $S' \subseteq S$ with $\psi(S') \le 2\overset{\leftrightarrow}{\psi}(S)$ in the underlying undirected graph of G.*

Combining these steps proves Proposition 9.13.

9.5 Reweighted Higher Eigenvalues

It is natural to extend the idea of reweighted eigenvalues to higher eigenvalues. In this section, we specialize in the case where $\vec{\pi}$ is the uniform distribution and highlight some known results.

Definition 9.21 (Maximum Reweighted k-th Smallest Eigenvalue)**.** *Given an undirected graph $G = (V = [n], E)$, the maximum reweighted k-th smallest eigenvalue of the normalized Laplacian matrix of G is defined as*

$$\lambda_k^*(G) := \max_{P \ge 0} \lambda_k(I - P),$$

where P satisfies the same constraints as in Definition 9.1.

Using reweighted eigenvalues, the higher-order Cheeger inequality in Chapter 7 and the improved Cheeger inequality in Chapter 8 have natural analogs for vertex expansion.

Theorem 9.22 (Higher-Order Cheeger Inequality for Vertex Expansion [KLT22])**.** *Given an undirected graph $G = (V, E)$, the k-way vertex expansion of G is defined as*

$$\psi_k(G) := \min\left\{1, \min_{S_1,\dots,S_k \subseteq V} \max_{1\le i\le k} \psi(S_i)\right\},$$

where the minimum is taken over pairwise disjoint subsets $S_1, \dots, S_k$ of V.

If $G = (V, E)$ is of maximum degree d, then

$$\lambda_k^*(G) \lesssim \psi_k(G) \lesssim k^{9/2} \log k \sqrt{\log d \cdot \lambda_k^*(G)} \quad \text{and} \quad \psi_{k/2}(G) \lesssim \sqrt{k} \log k \sqrt{\log d \cdot \lambda_k^*(G)}.$$

The dependency on k in the above theorem is likely not tight, but the following improved Cheeger inequality for vertex expansion is tight up to a constant factor.

Theorem 9.23 (Improved Cheeger Inequality for Vertex Expansion [KLT22])**.** *For any undirected graph $G = (V, E)$ with maximum degree d and any $k \geq 2$,*

$$\lambda_2^*(G) \lesssim \psi(G) \lesssim \frac{k \cdot \lambda_2^*(G) \cdot \log d}{\sqrt{\lambda_k^*(G)}}.$$

The proofs of these theorems follow as natural extensions of Theorem 7.2 and Theorem 8.1, demonstrating a systematic way to extend results from edge conductance to vertex expansion.

Semidefinite Programs for Reweighted Sums of Eigenvalues

A technical issue is that maximizing the k-th eigenvalue for $k > 2$ is not a convex optimization problem. Instead, the sum of the k smallest eigenvalues can be formulated as a semidefinite program.

Proposition 9.24 (Sum of k Smallest Eigenvalues)**.** *Let $X \in \mathbb{R}^{n \times n}$ be a symmetric matrix and let $1 \leq k \leq n$. Suppose the eigenvalues of X are $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$. Then, $\lambda_1 + \lambda_2 + \cdots + \lambda_k$ is the value of the following semidefinite program:*

$$\begin{aligned} \min_{Y \in \mathbb{R}^{n \times n}} \quad & \operatorname{Tr}(XY) \\ \text{subject to} \quad & 0 \preccurlyeq Y \preccurlyeq I_n \\ & \operatorname{Tr}(Y) = k. \end{aligned}$$

In the proofs, the reweighted sum of eigenvalues is used instead of the reweighted k-th eigenvalue.

Definition 9.25 (Maximum Reweighted Sum of k Smallest Eigenvalues)**.** *Given an undirected graph $G = (V, E)$ and the uniform distribution $\vec{\pi}$ on V, the maximum reweighted sum of the k smallest eigenvalues of the normalized Laplacian matrix of G is defined as*

$$\sigma_k^*(G) := \max_{P \geq 0} \sum_{i=1}^{k} \lambda_k(I - P),$$

where P satisfies the same constraints as in Definition 9.1. Note that

$$\lambda_k^*(G) \leq \sigma_k^*(G) \leq k \cdot \lambda_k^*(G).$$

The dual program has a nice formulation that is similar to the isotropy condition in Lemma 7.6.

Proposition 9.26 (Dual Program for $\sigma_k^*(G)$)**.** *For any undirected graph $G = (V, E)$ with a self loop at each vertex and the uniform distribution $\vec{\pi}$, the following SDP is dual to the primal program in Definition 9.25 where*

$$\begin{aligned} \kappa(G) := \min_{f: V \to \mathbb{R}^n,\ g: V \to \mathbb{R}_+} \quad & \sum_{i \in V} g(i) \\ \text{subject to} \quad & g(i) + g(j) \geq \|f(i) - f(j)\|^2 \quad \forall ij \in E \\ & \sum_{i \in V} f(i) f(i)^\top \preccurlyeq I_n \\ & \sum_{i \in V} \|f(i)\|^2 = k. \end{aligned}$$

Strong duality $\sigma_k^*(G) = \kappa(G)$ holds.

9.6 Reweighting Conjectures

The result in this chapter (Corollary 9.4) implies that for any graph with constant vertex expansion, there exists a reweighting such that the mixing time to the uniform distribution is $O(\log d \cdot \log n)$.

Conjecture 9.27 (Fastest Mixing Time of Vertex Expander)**.** *Any graph with constant vertex expansion admits a reweighting such that the mixing time to the uniform distribution is* $O(\log n)$.

If true, this conjecture would imply a positive resolution to a conjecture by Steurer [Ste10, Conjecture 9.2], which in turn would lead to an improved subexponential-time approximation algorithm for the sparsest cut problem.

Arora and Ge also proposed a related reweighting conjecture for small-set vertex expanders.

Conjecture 9.28 (Reweighting for Small-Set Vertex Expanders [AG11])**.** *There exists an algorithm such that for any graph* G*, it either:*

- *Find a subset* S *of vertices with* $\psi(S) \le (n/|S|)^{1/c}$ *for some constant* $c \ge 1$*, or*
- *Find a reweighting of* G *such that at most* n^{δ} *eigenvalues of its normalized adjacency matrix are* $-\frac{1}{16}$*, for some constant* δ.

If true, this conjecture would yield an improved subexponential-time algorithm for coloring 3-colorable graph. See [AG11, Appendix D] for details on this implication.

9.7 Problems

Problem 9.29 (λ_∞ and Symmetric Vertex Expansion [BHT00])**.** *Bobkov, Houdré and Tetali defined an interesting quantity*

$$\lambda_\infty(G) := \min_{x:V\to\mathbb{R},\ x\perp\vec{1}} \frac{\sum_{u\in V} \max_{v:(v,u)\in E} (x(u)-x(v))^2}{\sum_{u\in V} x(u)^2}$$

and prove an analog of Cheeger's inequality that

$$\lambda_\infty(G) \lesssim \Phi^V(G) \lesssim \sqrt{\lambda_\infty(G)},$$

where

$$\Phi^V(S) := |V| \cdot \frac{|\partial(S) \cup \partial(V-S)|}{|S| \cdot |V-S|} \quad \text{and} \quad \Phi^V(G) := \min_{S\subset V} \Phi^V(S)$$

is called the symmetric vertex expansion of the graph. Provide a proof of their theorem.

Problem 9.30 (Dimension Reduction)**.** *Given a* k*-dimensional solution to* $\vec{\gamma}(G)$ *with objective value* γ*, show that there is a* 1*-dimensional solution to* $\vec{\gamma}^{(1)}(G)$ *with objective value at most* $O(k \cdot \gamma)$.

Problem 9.31 (Threshold Rounding)**.** *Prove Proposition 9.13.*

Hint: The inequality $|x(i)^2 - x(j)^2| \le |x(i)-x(j)| \cdot \big(|x(i)-x(j)| + 2|x(i)|\big)$ *may be useful.*

Problem 9.32 (Postprocessing)**.** *Prove Lemma 9.20. Hint: Consider* $S' := S - \vec{\partial} S$.

References

[AG11] Sanjeev Arora and Rong Ge. New tools for graph coloring. In *International Workshop on Approximation Algorithms for Combinatorial Optimization*, pages 1–12. Springer, 2011. 121

[BDX04] Stephen P. Boyd, Persi Diaconis, and Lin Xiao. Fastest mixing markov chain on a graph. *SIAM Rev.*, 46(4):667–689, 2004. 109, 110

[BHT00] Sergey G. Bobkov, Christian Houdré, and Prasad Tetali. λ_∞ vertex isoperimetry and concentration. *Comb.*, 20(2):153–172, 2000. 121

[JPV23] Vishesh Jain, Huy Pham, and Thuy-Duong Vuong. Dimension reduction for maximum matchings and the fastest mixing markov chain. *Comptes Rendus. Mathématique*, 361(G5):869–876, 2023. 111, 129

[KLT22] Tsz Chiu Kwok, Lap Chi Lau, and Kam Chuen Tung. Cheeger inequalities for vertex expansion and reweighted eigenvalues. In *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 366–377. IEEE, 2022. 8, 111, 119, 120

[LRV13] Anand Louis, Prasad Raghavendra, and Santosh S. Vempala. The complexity of approximating vertex expansion. In *54th Annual IEEE Symposium on Foundations of Computer Science, FOCS*, pages 360–369. IEEE Computer Society, 2013. 111, 116, 117

[OTZ24] Sam Olesker-Taylor and Luca Zanetti. Geometric bounds on the fastest mixing markov chain. *Probability Theory and Related Fields*, 188(3):1017–1062, 2024. 111

[Roc05] Sébastien Roch. Bounding Fastest Mixing. *Electronic Communications in Probability*, 10(none):282 – 296, 2005. 110, 112, 113

[Ste10] David Steurer. *On the Complexity of Unique Games and Graph Expansion*. PhD thesis, Princeton University, 2010. 121

Chapter 10

Reweighted Eigenvalues for Directed Graphs and Hypergraphs

It has been an open direction to develop a spectral theory for directed graphs and hypergraphs. For directed graphs, the natural associated matrices are not symmetric, and little is known about the relationship between their complex eigenvalues and combinatorial properties. For hypergraphs, there is no universally accepted choice for the associated matrices.

In this chapter, we define reweighted eigenvalues for directed graphs and hypergraphs [LTW23], and use these formulations to derive new Cheeger-type inequalities related to their expansion properties. This approach provides a general method for reducing expansion problems in more general settings to the basic setting of edge conductance of undirected graphs. As a consequence, it yields a combinatorial characterization of the fastest mixing time problem for general Markov chains, and an efficient algorithm to certify that a directed graph is an expander.

10.1 Cheeger Inequality for Directed Vertex Expansion

We begin with directed vertex expansion, as it is the closest analogue to the undirected vertex expansion studied in the previous chapter.

Definition 10.1 (Directed Vertex Expansion)**.** *Let $G = (V, E)$ be a directed graph. For a subset $S \subseteq V$, define the set of out-neighbors of S be*

$$\partial^+(S) := \{j \notin S \mid \exists i \in S \text{ with } ij \in E\}.$$

The directed vertex expansion of a set $S \subseteq V$ and of the graph G are defined as

$$\vec{\psi}(S) := \frac{\min\left\{\left|\partial^+(S)\right|, \left|\partial^+(\overline{S})\right|\right\}}{\min\left\{|S|, |\overline{S}|\right\}} \quad \text{and} \quad \vec{\psi}(G) := \min_{\emptyset \neq S \subset V} \vec{\psi}(S).$$

Note that $\vec{\psi}(S) \leq 1$ for all $S \subseteq V$ as $\partial^+(\overline{S}) \subseteq S$.

To define the reweighted eigenvalue for directed vertex expansion, we follow the approach in Section 9.3 to reduce directed vertex expansion to undirected edge conductance.

To certify that a directed graph $G = (V, E)$ has large vertex expansion, the idea is to find the best reweighted *Eulerian* subgraph $H = (V, E, w)$ of G, where each edge $ij \in E$ has weight $w(ij)$, and the weighted degrees satisfy

$$\sum_{i \in V} w(ij) = \sum_{i \in V} w(ji) = 1, \quad \forall j \in V.$$

The directed edge conductance of H (see Definition 10.6) then provides a lower bound on the vertex expansion of G. This step follows the reduction in Section 9.3.

Since H is Eulerian, the edge conductance of H is equal to that of its underlying undirected graph $\bar{H}$, where each edge weight is given by

$$\bar{w}(ij) = \frac{1}{2}\big(w(ij) + w(ji)\big).$$

Since $\bar{H}$ is undirected, we can apply Cheeger's inequality to lower bound its edge conductance using the second smallest eigenvalue of its normalized Laplacian matrix.

This leads to the following formulation of the reweighted second eigenvalue for directed vertex expansion (see Proposition 10.11 for a proof about this reduction).

Definition 10.2 (Maximum Reweighted Second Eigenvalue with Vertex Capacity Constraints)**.** *Given a directed graph $G = (V, E)$, the maximum reweighted second eigenvalue with vertex capacity constraints is defined as*

$$\begin{aligned}
\vec{\lambda}_2^{v*}(G) \;:=\; \max_{A \geq 0} \quad & \lambda_2\Big(I - \frac{A + A^\top}{2}\Big) & \\
\text{subject to} \quad & A(i,j) = 0 & \forall ij \notin E \\
& \sum_{i \in V} A(i,j) = \sum_{i \in V} A(j,i) & \forall j \in V \\
& \sum_{i \in V} A(i,j) = 1 & \forall j \in V
\end{aligned}$$

where A is the adjacency matrix of the reweighted Eulerian subgraph. Then $\frac{1}{2}(A + A^\top)$ is the adjacency matrix of the underlying undirected graph, $\mathcal{L} := I - \frac{1}{2}(A + A^\top)$ is its normalized Laplacian matrix, and $\lambda_2(\mathcal{L})$ is the second smallest eigenvalue of $\mathcal{L}$.

To ensure that the optimization problem for $\vec{\lambda}_2^{v*}(G)$ is always feasible, we assume that each vertex has a self-loop as in Chapter 9.

The following Cheeger-type inequality relates $\vec{\lambda}_2^{v*}(G)$ and $\vec{\psi}(G)$, showing that the directed vertex expansion is large if and only if the reweighted second eigenvalue is large.

Theorem 10.3 (Cheeger Inequality for Directed Vertex Expansion)**.** *For any directed graph G,*

$$\vec{\lambda}_2^{v*}(G) \lesssim \vec{\psi}(G) \lesssim \sqrt{\vec{\lambda}_2^{v*}(G) \cdot \log \frac{\Delta}{\vec{\lambda}_2^{v*}(G)}},$$

where Δ is the maximum total degree of a vertex of G, defined as $\max_{i \in V}\{\deg^+(i) + \deg^-(i)\}$.

10.2 Fastest Mixing Time on General Markov Chains

The reweighted eigenvalue for undirected vertex expansion in Definition 9.1 was used to bound the fastest mixing time on *reversible* Markov chains in Chapter 9.

The reweighted eigenvalue for directed vertex expansion $\vec{\lambda}_2^{v*}(G)$ in Definition 10.2 can be used to bound the fastest mixing time on *general* Markov chains.

Definition 10.4 (Fastest Mixing Time on General Markov Chain)**.** *Given a directed graph $G = (V, E)$ and a probability distribution $\vec{\pi}$ on V, the fastest mixing time problem is defined as*

$$\begin{aligned} \tau^*(G) \;:=\; \min_{P \geq 0} \quad & \tau(P) & \\ \text{subject to} \quad & P(i,j) = 0 & \forall ij \notin E \\ & \sum_{j \in V} P(i,j) = 1 & \forall i \in V \\ & \sum_{i \in V} \pi(i) \cdot P(i,j) = \pi(j) & \forall j \in V \end{aligned}$$

where P is the transition matrix of the Markov chain. The constraints are to ensure that P has nonzero entries only on the edges of G, that P is a row stochastic matrix, and that the stationary distribution of P is $\vec{\pi}$. The objective is to minimize the mixing time $\tau(P)$ to the stationary distribution $\vec{\pi}$.

A consequence of Theorem 10.3 is a combinatorial characterization of the fastest mixing time of general Markov chains, showing that small directed vertex expansion is the only obstruction to small fastest mixing time.

Theorem 10.5 (Fastest Mixing Time and Directed Vertex Expansion)**.** *For any strongly connected directed graph $G = (V = [n], E)$ with maximum total degree Δ and $\vec{\pi} = \vec{1}/n$,*

$$\frac{1}{\vec{\psi}(G)} \lesssim \tau^*(G) \lesssim \frac{1}{\vec{\psi}(G)^2} \cdot \log \frac{\Delta}{\vec{\psi}(G)} \cdot \log n.$$

Proof. Since the graph is strongly connected, $\vec{\lambda}_2^{v*}(G) > 0$. To prove the upper bound, we prove that

$$\tau^*(G) \lesssim \frac{1}{\vec{\lambda}_2^{v*}(G)} \cdot \log n,$$

and then the result will follow from Theorem 10.3.

Let A be an optimal reweighted Eulerian subgraph in Definition 10.2. Observe that A is a transition matrix satisfying the constraints in Definition 10.4 when $\pi = \vec{1}/n$, and so is $(I + A)/2$. Therefore, by the result by Chung and Fill in Theorem 3.21,

$$\tau^*(G) \leq \tau\Big(\frac{I + A}{2}\Big) \lesssim \frac{1}{\lambda_2(\mathfrak{L})} \cdot \log n = \frac{1}{\vec{\lambda}_2^{v*}(G)} \cdot \log n,$$

where $\mathfrak{L}$ as defined in (3.4) is equal to $I - (A + A^\top)/2$ when $\vec{\pi} = \vec{1}/n$. Thus, $\lambda_2(\mathfrak{L}) = \vec{\lambda}_2^{v*}(G)$.

The proof of the lower bound is left to Problem 10.16. □

In Theorem 10.5, we only state the result when $\vec{\pi}$ is the uniform distribution for simplicity. It can be extended to general stationary distribution $\vec{\pi}$ if we define weighted directed version expansion as in Definition 9.2.

Together, Theorem 10.3 and Theorem 10.5 connect the reweighted second eigenvalue, directed vertex expansion, and fastest mixing time on directed graphs, similar to how the classical results connect the second eigenvalue, undirected edge conductance, and mixing time on undirected graphs.

10.3 Cheeger Inequality for Directed Edge Conductance

Cheeger's inequality has two primary applications: certifying whether an undirected graph is an expander graph and providing a spectral algorithm for graph partitioning. This motivates the development of a Cheeger-type inequality for directed edge conductance.

Definition 10.6 (Directed Edge Conductance)**.** *Let $G = (V, E)$ be a directed graph. For a subset $S \subseteq V$, define the set of outgoing edges of S as*

$$\delta^+(S) := \{ij \in E \mid i \in S \text{ and } j \notin S\}.$$

The directed edge conductance of a set $S \subseteq V$ and of the graph G are defined as

$$\vec{\phi}(S) := \frac{\min\left\{\left|\delta^+(S)\right|, \left|\delta^+(\overline{S})\right|\right\}}{\min\left\{\vec{\mathrm{vol}}(S), \vec{\mathrm{vol}}(\overline{S})\right\}} \quad \text{and} \quad \vec{\phi}(G) := \min_{\emptyset \neq S \subset V} \vec{\phi}(S),$$

where the volume is given by $\vec{\mathrm{vol}}(S) := \sum_{i \in S}(\deg^+(i) + \deg^-(i))$, the sum of total degrees in S.

To certify that a directed graph $G = (V, E)$ has large edge conductance, we find the best reweighted Eulerian subgraph H where each edge has weight $w(ij) \leq 1$, and the weighted total degrees match those in G (by possibly adding self-loops). The edge conductance of H provides a lower bound on that of G. Since H is Eulerian, its edge conductance is equal to the edge conductance of the underlying undirected graph $\bar{H}$ with edge weight $\bar{w}(ij) = \frac{1}{2}(w(ij) + w(ji))$. Applying Cheeger's inequality to $\bar{H}$ provides a lower bound on its edge conductance in terms of the second smallest eigenvalue of its normalized Laplacian matrix.

Definition 10.7 (Maximum Reweighted Second Eigenvalue with Edge Capacity Constraints)**.** *Given a directed graph $G = (V, E)$, the maximum reweighted second eigenvalue under edge capacity constraints is defined as*

$$\begin{aligned}
\vec{\lambda}_2^{c*}(G) \;:=\; \max_{A \geq 0} \quad & \lambda_2\left(I - D^{-\frac{1}{2}}\left(\frac{A + A^\top}{2}\right)D^{-\frac{1}{2}}\right) & \\
\text{subject to} \quad & A(i,j) = 0 & \forall ij \notin E \\
& \sum_{i \in V} A(i,j) = \sum_{i \in V} A(j,i) & \forall j \in V \\
& \sum_{i \in V} A(i,j) = \frac{1}{2}\left(\deg_G^+(j) + \deg_G^-(j)\right) & \forall j \in V \\
& A(i,j) \leq 1 & \forall ij \in E \text{ and } i \neq j,
\end{aligned}$$

where A is the adjacency matrix of the reweighted Eulerian subgraph, and D is the diagonal total-degree matrix of G with $D(j,j) = \deg_G^+(j) + \deg_G^-(j)$. The matrix $\frac{1}{2}(A + A^\top)$ is the adjacency matrix

of the underlying undirected graph, and its normalized Laplacian matrix is $\mathcal{L} := I - \frac{1}{2}D^{-1/2}(A + A^\top)D^{-1/2}$. *The objective function is to maximize the second eigenvalue of* $\mathcal{L}$.

To ensure that the optimization problem for $\vec{\lambda}_2^{e*}(G)$ *is always feasible, we assume that each vertex has a self-loop with no capacity constraints.*

The following result establishes a tighter Cheeger-type inequality relating $\vec{\lambda}_2^{e*}(G)$ and $\vec{\phi}(G)$.

Theorem 10.8 (Cheeger Inequality for Directed Edge Conductance)**.** *For any directed graph* G,

$$\vec{\lambda}_2^{e*}(G) \lesssim \vec{\phi}(G) \lesssim \sqrt{\vec{\lambda}_2^{e*}(G) \cdot \log \frac{1}{\vec{\lambda}_2^{e*}(G)}}.$$

The important point about Theorem 10.8 is that there is no dependence on the maximum degree of G as in Theorem 10.3. As a consequence, $\vec{\lambda}_2^{e*}(G)$ is a polynomial-time computable quantity that can certify whether a directed graph has constant edge conductance, analogous to the role of the second eigenvalue in Cheeger's inequality for undirected graphs.

Furthermore, as in the proof of Cheeger's inequality, the proof of Theorem 10.8 provides a polynomial time algorithm to find a set S with the stated guarantee. This introduces a new spectral approach for clustering and partitioning directed graphs.

See [Yos16, Yos19] for a related Cheeger-type inequality and its applications.

10.4 Proof Overview

The proofs of Theorem 10.3 and Theorem 10.8 have a similar structure as that of Theorem 9.3. We just outline the proofs and highlight the role of the asymmetric ratio of a directed graph.

Semidefinite Programs for Reweighted Eigenvalues

We can follow the same approach as in Section 9.2 to construct the dual semidefinite programs.

Lemma 10.9 (Dual Program of $\vec{\lambda}_2^{v*}(G)$)**.** *Given a directed graph* $G = (V = [n], E)$, *the dual semidefinite program of* $\vec{\lambda}_2^{v*}(G)$ *is*

$$\begin{aligned}
\vec{\gamma}^v(G) := \min_{f:V\to\mathbb{R}^n} \min_{q:V\to\mathbb{R}_+,\ r:V\to\mathbb{R}} \quad & \frac{1}{2}\sum_{i\in V} q(i) \\
\text{subject to} \quad & q(j) \geq \|f(i) - f(j)\|_2^2 - r(i) + r(j) \qquad \forall ij \in E \\
& \sum_{i\in V} f(i) = \vec{0} \\
& \sum_{i\in V} \|f(v)\|_2^2 = 1.
\end{aligned}$$

Strong duality $\vec{\gamma}^v(G) = \vec{\lambda}_2^{v*}(G)$ *holds.*

Lemma 10.10 (Dual Program of $\vec{\lambda}_2^{e*}(G)$). *Given a directed graph $G = (V = [n], E)$, the dual semidefinite program of $\vec{\lambda}_2^{e*}(G)$ is*

$$\begin{aligned}\vec{\gamma}^e(G) := \min_{f:V\to\mathbb{R}^n} \min_{q:E\to\mathbb{R}_+,\ r:V\to\mathbb{R}} \quad & \frac{1}{2}\sum_{ij\in E} q(ij) \\ \text{subject to} \quad & q(ij) \geq \|f(i)-f(j)\|_2^2 - r(i) + r(j) \qquad \forall ij \in E \\ & \sum_{i\in V}(\deg^+(i)+\deg^-(i))\cdot f(i) = \vec{0} \\ & \sum_{i\in V}(\deg^+(i)+\deg^-(i))\cdot \|f(i)\|_2^2 = 1.\end{aligned}$$

Strong duality $\vec{\gamma}^e(G) = \vec{\lambda}_2^{v*}(G)$ holds.

Easy Directions by Reductions

There are two ways to prove the easy directions in Theorem 10.3 and Theorem 10.8. A standard approach is to construct a solution to the dual programs $\vec{\gamma}^v(G)$ or $\vec{\gamma}^e(G)$ with small objective value when the directed vertex expansion or the directed edge conductance is small. Instead, we use the reductions discussed earlier to prove the easy directions in a more direct and intuitive way.

Proposition 10.11 (Easy Direction for Directed Vertex Expansion). *For any directed graph G,*

$$\vec{\lambda}_2^{v*}(G) \leq 2\vec{\psi}(G).$$

Proof. Let $w(ij) := A(i,j)$ be the edge weight in the Eulerian reweighted subgraph for $ij \in E$. For any nonempty $S \subset V$, by Definition 10.1 of directed vertex expansion and Definition 10.6 of directed edge conductance,

$$\vec{\psi}(S) = \frac{\min\{|\partial^+(S)|, |\partial^+(\overline{S})|\}}{\min\{|S|,|\overline{S}|\}} \geq \frac{2\cdot\min\{w(\delta^+(S)), w(\delta^-(S))\}}{\min\{\vec{\mathrm{vol}}_w(S), \vec{\mathrm{vol}}_w(\overline{S})\}} = 2\vec{\phi}(S),$$

where we use the degree constraints in Definition 10.2 to establish that $w(\delta^+(S)) \leq |\partial^+(S)|$ and $w(\delta^-(S)) \leq |\partial^+(\overline{S})|$, and $\mathrm{vol}_w(S) = 2|S|$ for every nonempty $S \subset V$ as the total degree of each vertex is 2.

Since the edge-weighted directed graph $H = (V, E, w)$ is Eulerian, it follows that $w(\delta^+(S)) = w(\delta^-(S))$ for every nonempty $S \subset V$. Thus, the directed edge conductance of H is half the edge conductance of the underlying undirected graph $\bar{H}$ with edge weight $\bar{w}(ij) = \frac{1}{2}\big(w(ij) + w(ji)\big)$, because

$$2\vec{\phi}(S) = \frac{\min\{w(\delta^+(S)), w(\delta^-(S))\}}{\frac{1}{2}\cdot\min\{\vec{\mathrm{vol}}_w(S), \vec{\mathrm{vol}}_w(\overline{S})\}} = \frac{\bar{w}(\delta(S))}{\min\{\mathrm{vol}_{\bar{w}}(S), \mathrm{vol}_{\bar{w}}(\overline{S})\}} = \phi(S).$$

Since $\bar{H}$ is undirected, we can apply Cheeger's inequality in (2.2) to lower bound its edge conductance using the second smallest eigenvalue of its normalized Laplacian matrix $\mathcal{L}(A) := I - \frac{1}{2}(A + A^\top)$. Therefore, for any nonempty $S \subset V$,

$$\vec{\psi}(S) \geq 2\vec{\phi}(S) = \phi(S) \geq \frac{1}{2}\lambda_2(\mathcal{L}(A)).$$

Since this holds for any nonempty $S \subset V$ and any weighted Eulerian subgraph defined by A satisfying the constraints in Definition 10.2, we conclude that $2\psi(G) \geq \max_A \lambda_2(\mathcal{L}(A)) = \vec{\lambda}_2^{v*}(G)$. □

The proof of the easy direction for directed edge conductance is similar and is left to Exercise 10.17.

Hard Directions via Asymmetric Ratio

As in Section 9.4, the proof of the hard direction has two steps: dimension reduction and threshold rounding. Both steps are more involved but follow a similar structure, so we omit the details and only highlight a key parameter that plays an important role in the proofs.

Definition 10.12 (Asymmetric Ratio of Directed Graphs)**.** *Given a directed graph $G = (V, E)$, the asymmetric ratio of a set $S \subseteq V$ and of the graph G are defined as*

$$a(S) := \frac{|\delta^+(S)|}{|\delta^+(\overline{S})|} \quad \text{and} \quad a(G) := \max_{\emptyset \neq S \subset V} a(S).$$

The asymmetric ratio measures how close a directed graph is to an undirected graph. When $a(G) = 1$, the directed graph is Eulerian, with its edge conductance equal to that of the underlying undirected graph. This parameter satisfies two useful properties.

First, it can be used to prove more refined Cheeger inequalities that

$$\vec{\phi}(G) \leq \sqrt{\vec{\lambda}_2^{e*}(G) \cdot \log a(G)} \quad \text{and} \quad \vec{\psi}(G) \leq \sqrt{\vec{\lambda}_2^{v*}(G) \cdot \log\big(\Delta \cdot a(G)\big)}. \tag{10.1}$$

The improvement comes from a better dimension reduction analysis using the techniques in [JPV23], and Hoffman's result on bounded-weighted circulations. The threshold rounding step has a twist that considers the two orderings defined by $f \pm r$.

Second, it relates to the directed edge conductance and directed vertex expansion as follows:

$$a(G) \leq \frac{1}{\vec{\phi}(G)} \quad \text{and} \quad a(G) \leq \frac{\Delta}{\vec{\psi}(G)}. \tag{10.2}$$

The proofs of these inequalities follow from elementary combinatorial arguments.

Combining these inequalities yields Theorem 10.8 and Theorem 10.3.

10.5 Cheeger-Type Inequalities for Hypergraphs

A spectral theory for hypergraphs based on a continuous time diffusion process and a nonlinear Laplacian operator has been developed [Lou15, CLTZ18].

In this section, we define reweighted eigenvalues for hypergraphs and use them to provide an elementary approach to derive their Cheeger-type inequalities for hypergraphs.

Definition 10.13 (Hypergraph Edge Conductance)**.** *Let $H = (V, E)$ be a hypergraph. For a subset $S \subseteq V$, define the edge boundary and volume of S as*

$$\delta(S) := \{e \in E \mid e \cap S \neq \emptyset \text{ and } e \cap \overline{S} \neq \emptyset\} \quad \text{and} \quad \mathrm{vol}(S) := \sum_{i \in S} \deg(i) := \sum_{i \in S} \big|\{e : i \in e\}\big|.$$

The hypergraph edge conductance of a set $S \subseteq V$ and of the graph G are defined as

$$\phi(S) := \frac{\big|\delta(S)\big|}{\min\big\{\mathrm{vol}(S), \mathrm{vol}(\overline{S})\big\}} \quad \text{and} \quad \phi(H) := \min_{\emptyset \neq S \subset V} \phi(S).$$

To define reweighted eigenvalues for hypergraphs, the idea is simply to consider the "clique-graph" of the hypergraph H. We find the best reweighted subgraph G of the clique-graph and use its edge conductance to provide a lower bound on the edge conductance of H, subject to the constraint that the total weight of the "clique-edges" of a hyperedge e is bounded by 1.

Definition 10.14 (Maximum Reweighted Second Eigenvalue for Hypergraph Edge Conductance)**.** *Given a hypergraph $H = (V, E)$, the maximum reweighted second eigenvalue for H is defined as*

$$\begin{aligned} \lambda_2^*(H) \;:=\; \max_{A \geq 0} \quad & \lambda_2\left(I - D^{-\frac{1}{2}} A D^{-\frac{1}{2}}\right) \\ \text{subject to} \quad & \sum_{i,j \in e} c(i,j,e) \leq 1 && \forall e \in E \\ & \sum_{e \in E : i,j \in e} c(i,j,e) = A(i,j) && \forall i,j \in V \\ & \sum_{j \in V} A(i,j) = \deg_H(i) && \forall i \in V. \end{aligned}$$

In this formulation, there is a clique-edge variable $c(i, j, e)$ for each pair of vertices i, j in a hyperedge e, with the constraints that the total weight of the clique-edges in e is bounded by 1. The matrix A is the adjacency matrix of the reweighted subgraph of the clique-graph, with edge weight $A(i, j)$ equal to the sum of the weight of the clique-edges involving i and j. The diagonal degree matrix D is defined as $D(i, i) = \deg_H(i)$.

To ensure that the optimization problem for $\lambda_2^*(H)$ is always feasible, we assume that each vertex has a self-loop without a capacity constraint.

Theorem 10.15 (Cheeger Inequality for Hypergraph Edge Conductance)**.** *For any hypergraph $H = (V, E)$ with hyperedges of size at most r,*

$$\lambda_2^*(H) \lesssim \phi(H) \lesssim \sqrt{\lambda_2^*(H) \cdot \log(r)}.$$

The following dual program can be derived using the same approach in Section 9.2.

$$\begin{aligned} \gamma_2^*(H) = \min_{f: V \to \mathbb{R}^n} \min_{g: E \to \mathbb{R}_+} \quad & \sum_{e \in E} g(e) \\ \text{subject to} \quad & g(e) \geq \|f(i) - f(j)\|_2^2 && \forall \{i,j\} \subseteq e, \forall e \in E \\ & \sum_{i \in V} \deg_H(i) \cdot f(i) = \vec{0} \\ & \sum_{i \in V} \deg_H(i) \cdot \|f(i)\|_2^2 = 1. \end{aligned}$$

We note that $\gamma_2^*(H)$ provides the same semidefinite program as in [CLTZ18], and thus Theorem 10.15 follows from their results. Alternatively, we can follow the same approach in Chapter 9 to prove Theorem 10.15, by doing dimension reduction and then threshold rounding.

10.6 Discussions and Open Questions

Reductions: The reweighted eigenvalue approach provides a general method for reducing expansion problem in more general settings to the basic problem of edge conductance in undirected graphs.

The effectiveness of this approach depends on how closely the given problem resembles the edge conductance problem in an undirected graph. The maximum degree d for undirected vertex expansion, the asymmetric ratio $a(G)$ for directed edge conductance, and the maximum hyperedge size r serve as measures of this closeness. Trivial reductions to undirected edge conductance incur a factor loss of d for undirected vertex expansion, $a(G)$ for directed edge-conductance (by ignoring directions), and r for hypergraph edge conductance (by considering the clique graph). In contrast, reductions via the reweighted eigenvalue approach lose only a factor of $\log d$ in Theorem 9.3, $\log a(G)$ in (10.1), and $\log r$ in Theorem 10.15, respectively.

Generalizations: As in Section 9.5, reweighted higher eigenvalues can be defined for directed graphs and hypergraphs. Using these reweighted higher eigenvalues, one can establish an analog of higher-order Cheeger inequality for hypergraphs, and an analog of improved Cheeger inequality for directed graphs and hypergraphs [LTW23]. However, the natural analog of higher-order Cheeger inequality does not hold for directed graphs. It remains an open problem to determine whether an alternative analog can be formulated in this setting.

Optimal Bound: For Theorem 10.3 and Theorem 10.8, it is not yet known whether the $\log(1/\lambda_2^*)$ term is necessary. In particular, if this logarithmic term could be completely removed from Theorem 10.8, it would yield a perfect analog of Cheeger's inequality in Theorem 2.2.

Algorithms: Current algorithms for computing an optimal reweighted subgraph rely on semidefinite programing, which is both computationally expensive and not intuitive. It would be useful to have faster and more insightful algorithms for computing an optimal reweighting. Some progress in this direction has been made in [LTW24], where the matrix multiplicative weight algorithm is used to develop a combinatorial approach based on flows for computing an approximately optimal reweighting.

Unified Solution: Different reweighted eigenvalues may have different optimal reweighted subgraphs. A basic open question is whether there exists a single reweighted subgraph that is approximately optimal for λ_k^* for all $k \geq 2$.

10.7 Problems

Problem 10.16 (Lower Bound on Fastest Mixing Time)**.** *Prove the lower bound in Theorem 10.5.*

Exercise 10.17 (Easy Direction for Directed Edge Conductance)**.** *Prove the easy direction in Theorem 10.8.*

Exercise 10.18 (Easy Direction for Hypergraph Edge Conductance)**.** *Prove the easy direction in Theorem 10.15.*

Problem 10.19 (Hard Direction for Hypergraph Edge Conductance)**.** *Prove the hard direction in Theorem 10.15. Hint: The techniques in Chapter 9 suffice.*

Problem 10.20 (Asymmetric Ratio)**.** *Prove* (10.2).

References

[CLTZ18] T.-H. Hubert Chan, Anand Louis, Zhihao Gavin Tang, and Chenzi Zhang. Spectral properties of hypergraph laplacian and approximation algorithms. *J. ACM*, 65(3):15:1–15:48, 2018. 129, 130

[JPV23] Vishesh Jain, Huy Pham, and Thuy-Duong Vuong. Dimension reduction for maximum matchings and the fastest mixing markov chain. *Comptes Rendus. Mathématique*, 361(G5):869–876, 2023. 111, 129

[Lou15] Anand Louis. Hypergraph markov operators, eigenvalues and approximation algorithms. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC*, pages 713–722. ACM, 2015. 129

[LTW23] Lap Chi Lau, Kam Chuen Tung, and Robert Wang. Cheeger inequalities for directed graphs and hypergraphs using reweighted eigenvalues. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing, STOC*, pages 1834–1847. ACM, 2023. 8, 123, 131

[LTW24] Lap Chi Lau, Kam Chuen Tung, and Robert Wang. Fast algorithms for directed graph partitioning using flows and reweighted eigenvalues. In *Proceedings of the 2024 ACM-SIAM Symposium on Discrete Algorithms, SODA*, pages 591–624. SIAM, 2024. 131, 230, 232, 241, 246, 250

[Yos16] Yuichi Yoshida. Nonlinear laplacian for digraphs and its applications to network analysis. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*, pages 483–492, 2016. 127

[Yos19] Yuichi Yoshida. Cheeger inequalities for submodular transformations. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2582–2601. SIAM, 2019. 127

Part II

Random Walks and Graph Decompositions

In this part, we see that ideas from random walks can be used to find small sparse cuts, which serve as the building blocks for constructing expander decompositions and low threshold rank decompositions, providing a new paradigm for designing fast algorithms and approximation algorithms.

Chapter 11

Small Sparse Cuts from Random Walks

The problem of finding a small sparse cut is of both theoretical and practical interest.

On the theoretical side, the small-set expansion conjecture is closely related to the unique games conjecture. Arora, Barak, and Steurer [ABS15] proved a Cheeger-type inequality for small-set expansion and used it to design a subexponential time algorithm for small-set expansion and unique games. This influential result was the first to explore graph partitioning using higher eigenvalues and inspired further developments in Cheeger inequalities and approximation algorithms.

On the practical side, finding a small sparse cut has important applications in processing massive graphs. Spielman and Teng [ST13] designed the first local graph partitioning algorithm, which identifies a small sparse cut while only exploring a limited portion of the graph. This work opened up a new research direction and inspired further developments, including local graph partitioning algorithms using pagerank vector and evolving sets.

In this chapter, we present a unified spectral approach [KLL17] to establish these results.

11.1 Approximating Small-Set Edge Conductance

Informally, a small sparse cut is a set $S \subseteq V$ with small edge conductance $\phi(S)$ and small size $|S|$.

Definition 11.1 (Conductance Profile)**.** *Given an undirected graph $G = (V, E)$ and a parameter $\delta \leq 1/2$, the δ-small-set edge conductance of G is defined as*

$$\phi_\delta(G) := \min_{S:|S|\leq\delta|V|} \phi(S).$$

The small-set expansion conjecture formulated by Raghavendra and Steurer [RS10] postulates that the problem becomes harder as δ becomes smaller.

Conjecture 11.2 (Small-Set Expansion Conjecture [RS10])**.** *For any constant $\phi > 0$, there exists a constant $\delta > 0$ such that it is NP-hard to distinguish whether a d-regular graph $G = (V, E)$ is in one of the following two cases:*

1. *$\phi_\delta(G) \leq \phi$, or*
2. *$\phi_\delta(G) \geq 1 - \phi$.*

A Cheeger-type inequality for small-set edge conductance would disprove this conjecture. Specifically, an algorithm that always returns a set S with $\phi(S) \leq \sqrt{\phi_\delta(G)}$ and $|S| \leq \delta|V|$ would distinguish the two cases. However, the spectral partitioning algorithm based on the second eigenvalue and eigenvector in Chapter 2 does not have control over the size of the output set.

The best known approximation algorithm for small-set expansion is an SDP-based approximation algorithm by Raghavendra, Steurer and Tetali [RST10], which returns a set S of size $O(\delta|V|)$ with edge conductance $\phi(S) \leq \sqrt{\phi_\delta(G) \cdot \log(1/\delta)}$.

Arora, Barak, and Steurer [ABS15] proved a Cheeger-type inequality for small-set edge conductance using (much) higher eigenvalues. The key point is that, when k is sufficiently large, the edge conductance has no dependency on k, unlike in the higher-order Cheeger inequality in Theorem 7.2.

Theorem 11.3 (Cheeger Inequality for Small-Set Edge Conductance [ABS15])**.** *For any d-regular graph $G = (V = [n], E)$ and any $0 < \epsilon \leq 1$, there exists a set $S \subseteq V$ with*

$$\phi(S) \lesssim \sqrt{\frac{\lambda_k \ln n}{\epsilon \ln k}} \quad \text{and} \quad |S| \lesssim \frac{n}{k^{1-\epsilon}},$$

where λ_k is the k-th smallest eigenvalue of the normalized Laplacian matrix. For $k \geq n^\beta$,

$$\phi(S) \lesssim \sqrt{\frac{\lambda_k}{\epsilon \cdot \beta}} \quad \text{and} \quad |S| \lesssim n^{1-\beta(1-\epsilon)},$$

We will prove this result in this chapter, and explain how it can be used to design a subexponential-time algorithm to distinguish between the two cases of the small-set expansion conjecture in Chapter 13. The proof is based on analyzing the collision probability of random walks using a trace argument.

Random walks can also be applied directly to design approximation algorithms for the small-set edge conductance problem. The intuition for distinguishing the two cases in Conjecture 11.2 is as follows. In the first case, if we start a random walk from a random vertex i in S, then the random walk is expected to remain in S with high probability. In the second case, the random walk is expected to mix quickly for sets of size up to roughly $2\delta|V|$. Therefore, by computing $W^t\chi_i$ for an appropriately chosen t (where W is the lazy random walk matrix), we can sum the largest $\delta|V|$ entries in $W^t\chi_i$ to distinguish between the two cases.

This idea underlies Spielman and Teng's [ST13] local graph partitioning algorithm, which can return a set S with an approximation guarantee close to that of Cheeger's inequality while ensuring that the output set size satisfies $|S| \lesssim \delta|V|$.

Theorem 11.4 (Small-Set Edge Conductance from Random Walks [ST13])**.** *Let $G = (V, E)$ be a d-regular graph. For any $\delta > 0$, there exists a polynomial time algorithm using random walks that returns a set $S \subseteq V$ with*

$$\phi(S) \lesssim \sqrt{\phi_\delta(G) \cdot \log(\delta|V|)} \quad \text{and} \quad |S| \lesssim \delta|V|.$$

Their original proof is based on a combinatorial approach by Lovász and Simonovits [LS93].

In this chapter, we will present a unified spectral analysis of random walks that proves both Theorem 11.3 and Theorem 11.4. We will also explain how the random walk algorithm in Theorem 11.4 can be implemented locally (allowing it to run in sublinear time when δ is small enough) with a local approximation guarantee. See Theorem 11.13 for the precise statement.

11.2 Spectral Approach

The spectral approach seeks to find a vector $x \in \mathbb{R}^n_+$ with a small Rayleigh quotient and small support size. Applying the Cheeger rounding from Chapter 2 (see Lemma 7.4) to x would then yield a subset S with

$$\phi(S) \lesssim \sqrt{R(x)} \quad \text{and} \quad S \subseteq \operatorname{supp}(x),$$

where the Rayleigh quotient of x is

$$R(x) = \frac{x^\top \mathcal{L} x}{x^\top x} = \frac{\sum_{ij \in E}(x(i) - x(j))^2}{d\|x\|_2^2}. \tag{11.1}$$

Combinatorially and Analytically Sparse Vectors

The combinatorial requirement that x has small support size is not particularly suitable for spectral analysis. The idea in [ABS15] is to relax this condition to make it amendable to spectral analysis while maintaining essentially the same effect. By Cauchy-Schwarz, a vector x with support size at most δn satisfies

$$\|x\|_1 = \sum_{i=1}^n |x(i)| = \sum_{i \in \operatorname{supp}(x)} |x(i)| \le \sqrt{|\operatorname{supp}(x)|} \cdot \sqrt{\sum_{i \in \operatorname{supp}(x)} x(i)^2} \le \sqrt{\delta n} \cdot \|x\|_2,$$

which motivates the following definition.

Definition 11.5 (Combinatorially and Analytically Sparse Vectors)**.** *For $\delta \in [0,1]$, a vector $x \in \mathbb{R}^n_+$ is called δ-combinatorially sparse if $|\operatorname{supp}(x)| \le \delta n$, and δ-analytically sparse if $\|x\|_1 \le \sqrt{\delta n} \cdot \|x\|_2$.*

The following lemma shows that a δ-combinatorially sparse vector with a small Rayleigh quotient can be extracted from a δ-analytically sparse vector with a small Rayleigh quotient.

Lemma 11.6 (Combinatorially Sparse Vector from Analytically Sparse Vector)**.** *Given a δ-analytically sparse vector $x \in \mathbb{R}^n_+$, there exists a 4δ-combinatorially sparse vector $y \in \mathbb{R}^n_+$ with $R(y) \le 2R(x)$.*

Proof. The proof follows from a simple truncation argument. By scaling, we assume that $\|x\|_2^2 = \delta n$ and $\|x\|_1 \le \delta n$. Define $y \in \mathbb{R}^n_+$ as

$$y(i) := \max\left\{x(i) - \frac{1}{4}, 0\right\}.$$

It is clear that $|\operatorname{supp}(y)| \le 4\delta n$, as otherwise $\|x\|_1 > \delta n$, so y is 4δ-combinatorially sparse.

We compare $R(y)$ with $R(x)$. For the numerator, as truncation does not increase the edge length,

$$(y(i) - y(j))^2 \le (x(i) - x(j))^2.$$

For the denominator, note that $y(i)^2 \ge (x(i) - \frac{1}{4})^2 \ge x(i)^2 - \frac{1}{2}x(i)$, which implies that

$$\|y\|_2^2 = \sum_{i \in V} y(i)^2 \ge \sum_{i \in V} x(i)^2 - \frac{1}{2}\sum_{i \in V} x(i) \ge \delta n - \frac{1}{2}\delta n = \frac{1}{2}\delta n = \frac{1}{2}\|x\|_2^2.$$

Therefore,

$$R(y) = \frac{\sum_{ij \in E}(y(i) - y(j))^2}{d\sum_{i \in V} y(i)^2} \le \frac{\sum_{ij \in E}(x(i) - x(j))^2}{\frac{d}{2}\sum_{i \in V} x(i)^2} = 2R(x).$$

□

Algorithm and Analysis Outline

The algorithm is quite simple. We first describe it informally without specifying the parameters. Let $W = \frac{1}{2}I + \frac{1}{2}\mathcal{A}$ be the lazy random walk matrix.

1. For each vertex $i \in V$, compute $W^t\chi_i$ for some appropriately chosen t.
2. Truncate $W^t\chi_i$ to a vector y with small support using Lemma 11.6.
3. Apply Cheeger rounding from Lemma 7.4 to y to obtain a small sparse cut.

The following is an outline of the analysis.

1. For step (1), we will prove that the vectors $W^t\chi_i$ have small Rayleigh quotient for all $i \in V$ when t is large enough. The analysis here is similar to that of the power method used for computing an eigenvector corresponding to the largest eigenvalue.
2. For step (2), there are two different arguments.

   To prove Theorem 11.3, we argue that if there are many small eigenvalues in the normalized Laplacian matrix, then there exists some vertex $i \in V$ such that $W^t\chi_i$ is analytically sparse when t is small enough. This follows from a trace argument.

   To prove Theorem 11.4, we argue that if there is a small sparse cut S, then there exists some vertex $i \in S$ such that $W^t\chi_i$ is analytically sparse when t is small enough. This follows from a simple combinatorial argument based on staying probability.
3. Choose t appropriately to balance the Rayleigh quotient and the analytic sparsity of $W^t\chi_i$.

In the following sections, we will analyze step (1) of the algorithm in Section 11.3, prove Arora-Barak-Steurer's Theorem 11.3 in Section 11.4, and prove Spielman-Teng's Theorem 11.4 in Section 11.5, where we also explain how to implement the algorithm locally.

11.3 Power Method

In this section, we show that the Rayleigh quotient of $W^t\chi_i$ is small when t is large enough. This is not surprising, as $W^t\chi_i \to \vec{1}/n$ when G is a regular graph, so the Rayleigh quotient tends to zero when $t \to \infty$. What matters is the precise convergence rate, since for analytic sparsity, t cannot be set too large, creating a tension in its choice.

The following power mean inequality, which states that $\mathbb{E}[X^p]^{\frac{1}{p}} \geq \mathbb{E}[X^q]^{\frac{1}{q}}$ for a non-negative random variable X and $p > q > 0$, will be used in the analysis.

Lemma 11.7 (Power Mean Inequality)**.** *For non-negative real numbers $a_1, \ldots, a_n$ and $w_1, \ldots, w_n$ with $\sum_{j=1}^n w_j = 1$, for any $p > q > 0$,*

$$\Big(\sum_{j=1}^n w_j a_j^p\Big)^{\frac{1}{p}} \geq \Big(\sum_{j=1}^n w_j a_j^q\Big)^{\frac{1}{q}}.$$

The analysis is similar to that of the power method, an algorithm for computing an eigenvector corresponding to the largest eigenvalue of a matrix.

Lemma 11.8 (Power Method). *Let $G = (V, E)$ be a d-regular graph. Let $W = \frac{1}{2}(I + \mathcal{A})$ be the lazy random walk matrix of G. For any integer $t \geq 0$,*

$$R(W^t\chi_i) \leq 2 - 2\|W^t\chi_i\|_2^{\frac{1}{t}},$$

where $R(x)$ is the Rayleigh quotient as defined in (11.1).

Proof. Let the eigenvalues of W be $1 = \alpha_1 \geq \alpha_2 \geq \ldots \geq \alpha_n \geq 0$ with corresponding orthonormal eigenvectors $v_1, v_2, \ldots, v_n$. Write $\chi_i = \sum_{j=1}^n c_j v_j$. Then

$$W^t\chi_i = \sum_{j=1}^n c_j\alpha_j^t v_j \quad \text{and} \quad \|W^t\chi_i\|_2^2 = \sum_{j=1}^n c_j^2\alpha_j^{2t}.$$

Since $\mathcal{L} = I - \mathcal{A} = 2(I - W)$,

$$R(W^t\chi_i) = \frac{(\chi_i^\top W^t)\mathcal{L}(W^t\chi_i)}{\|W^t\chi_i\|_2^2} = 2 - 2\frac{(\chi_i^\top W^t)W(W^t\chi_i)}{\|W^t\chi_i\|_2^2} = 2 - 2\frac{\sum_{j=1}^n c_j^2\alpha_j^{2t+1}}{\sum_{j=1}^n c_j^2\alpha_j^{2t}}.$$

Since $\sum_{j=1}^n c_j^2 = \|\chi_i\|_2^2 = 1$ and $\alpha_j \geq 0$ for all j, we can apply the power mean inequality to obtain

$$\Big(\sum_{j=1}^n c_j^2\alpha_j^{2t+1}\Big)^{\frac{1}{2t+1}} \geq \Big(\sum_{j=1}^n c_j^2\alpha_j^{2t}\Big)^{\frac{1}{2t}}.$$

This implies that

$$\frac{\sum_{j=1}^n c_j^2\alpha_j^{2t+1}}{\sum_{j=1}^n c_j^2\alpha_j^{2t}} \geq \Big(\sum_{j=1}^n c_j^2\alpha_j^{2t}\Big)^{\frac{1}{2t}} = \Big(\|W^t\chi_i\|_2^2\Big)^{\frac{1}{2t}} = \|W^t\chi_i\|_2^{\frac{1}{t}}.$$

Thus, the lemma follows. □

To understand what this result implies, first note that $\|W^t\chi_i\|_2 \geq 1/\sqrt{n}$, which is minimized when $W^t\chi_i = \vec{1}/n$. Thus, using $1 - x \leq e^{-x}$,

$$R(W^t\chi_i) \leq 2\Big(1 - \Big(\frac{1}{\sqrt{n}}\Big)^{\frac{1}{t}}\Big) = 2\Big(1 - e^{-\frac{\ln n}{2t}}\Big) \leq \frac{\ln n}{t}.$$

Therefore, setting $t \gtrsim \ln n/\lambda$ ensures $R(W^t\chi_i) \leq \lambda$, while setting $t \gtrsim 1/\lambda$ ensures $R(W^t\chi_i) \leq \lambda \ln n$.

11.4 Cheeger Inequality for Small-Set Edge Conductance

The idea in [ABS15] is to prove a lower bound on $\|W^t\chi_i\|_2^2$ using higher eigenvalues.

Lemma 11.9 (Analytically Sparse Vector from Higher Eigenvalues). *Let $G = (V = [n], E)$ be a graph, and λ_k be the k-th smallest eigenvalue of its normalized Laplacian matrix. For any integers $k \geq 1$ and $t \geq 1$, there exists a vertex $i \in V$ with*

$$\|W^t\chi_i\|_2^2 \geq \frac{k}{n}\Big(1 - \frac{\lambda_k}{2}\Big)^{2t}.$$

Proof. The proof follows from a simple trace argument. Let $\alpha_1 \geq \cdots \geq \alpha_n$ be the eigenvalues of W. Since $W = I - \frac{1}{2}\mathcal{L}$, $\alpha_k = 1 - \frac{1}{2}\lambda_k$. Using Fact A.35 that the trace is equal to the sum of eigenvalues,

$$\sum_{j=1}^{n} \|W^t \chi_j\|_2^2 = \sum_{j=1}^{n} \chi_j^\top W^{2t} \chi_j = \mathrm{Tr}(W^{2t}) = \sum_{j=1}^{n} \alpha_j^{2t} \geq k\Big(1 - \frac{\lambda_k}{2}\Big)^{2t}.$$

The lemma follows by averaging. □

If $\|W^t \chi_i\|_2^2 \geq k/n = k\|W^t \chi_i\|_1^2/n$ (since $\|W^t \chi_1\|_1 = 1$), then $W^t \chi_i$ is an $1/k$-analytically sparse vector by Definition 11.5. It would then follow from Lemma 11.6 that there exists a $1/k$-combinatorially sparse vector with Rayleigh quotient at most $2R(W^t \chi_i)$ which would be ideal. We cannot achieve this exactly, but we obtain something close that the set size will be $O(n/k^{1-\epsilon})$ for some $0 < \epsilon < 1$.

Proof of Theorem 11.3. Set $t = \epsilon \ln k / 2\lambda_k$. By Lemma 11.9, there exists a vertex $i \in V$ such that

$$\|W^t \chi_i\|_2^2 \geq \frac{k}{n}\Big(1 - \frac{\lambda_k}{2}\Big)^{2t} \geq \frac{k}{n} \cdot e^{-2t\lambda_k} = \frac{k}{n} \cdot e^{-\epsilon \ln k} = \frac{k^{1-\epsilon}}{n},$$

where we used $1 - \frac{x}{2} \geq e^{-x}$ for $x \leq 1$. This implies that $W^t \chi_i$ is $k^{-(1-\epsilon)}$-analytically sparse.

For the Rayleigh quotient, using Lemma 11.8 and the bound $1 - x \leq e^{-x}$,

$$R(W^t \chi_i) \leq 2 - 2\|W^t \chi_i\|_2^{\frac{1}{t}} \leq 2 - 2\Big(\frac{k^{1-\epsilon}}{n}\Big)^{\frac{1}{t}} \leq 2 - 2\Big(\frac{1}{n}\Big)^{\frac{1}{t}} = 2 - 2e^{-\frac{2\lambda_k \ln n}{\epsilon \ln k}} \leq \frac{4\lambda_k \ln n}{\epsilon \ln k}.$$

Applying the truncation argument in Lemma 11.6 to $W^t \chi_i$, we obtain a vector $y \in R_+^n$ satisfying

$$R(y) \leq \frac{8\lambda_k \ln n}{\epsilon \ln k} \quad \text{and} \quad |\mathrm{supp}(y)| \leq \frac{4n}{k^{1-\epsilon}}.$$

Applying Cheeger rounding in Lemma 7.4 to y yields a set $S \subseteq V$ with

$$\phi(S) \lesssim \sqrt{\frac{\lambda_k \ln n}{\epsilon \ln k}} \quad \text{and} \quad |S| \lesssim \frac{n}{k^{1-\epsilon}}.$$

□

11.5 Local Graph Partitioning by Random Walks

One idea in [ST13] is to lower bound the probability of a random walk staying within a subset S.

Lemma 11.10 (Staying Probability)**.** *Let $G = (V, E)$ be a d-regular graph, and let W be its lazy random walk matrix. If the initial distribution is $\vec{p}_0 := \chi_S/|S|$ for a subset $S \subseteq V$, then $\vec{p}_t := W^t \vec{p}_0$ satisfies*

$$\sum_{j \in S} \vec{p}_t(j) \geq 1 - \frac{1}{2} \cdot t \cdot \phi(S).$$

Proof. The proof follows from a simple inductive argument. We lower bound $\sum_{j \in S} \vec{p}_t(j)$ by the probability that the random walk stays within S in all t steps. Equivalently, we upper bound the probability that the random walk leaves S in any of these t steps.

The initial distribution $\vec{p}_0$ assigns each vertex in S a probability of $1/|S|$. Since the graph is d-regular and the random walk is lazy, each edge in $\delta(S)$ carries a probability of $1/2d|S|$ leaving S. Thus, the total probability of leaving S in the first step is $|\delta(S)|/2d|S| = \phi(S)/2$.

We will argue that the probability of leaving S at each time step remains at most $\phi(S)/2$. This implies that the total probability of leaving S in any of the t steps is at most $t \cdot \phi(S)/2$, so

$$\sum_{j \in S} \vec{p}_t(j) \geq \Pr\left[\text{the random walk stays within } S \text{ for all } t \text{ steps}\right] \geq 1 - \frac{1}{2} \cdot t \cdot \phi(S).$$

To complete the proof, we just need to observe the invariant that the probability at each vertex at any time step is at most $1/|S|$. This ensures that the above calculation of the probability leaving S holds for all time step. The invariant follows from induction and the equation

$$\vec{p}_{t+1}(i) = \frac{1}{2} \cdot \vec{p}_t(i) + \frac{1}{2d} \sum_{j : ij \in E} \vec{p}_t(j) \leq \frac{1}{|S|}.$$

□

Since the random walk matrix is a linear transformation, an averaging argument shows that there exist good starting vertices within S.

Lemma 11.11 (Good Starting Vertices)**.** *Under the assumptions in Lemma 11.10, there exists $S' \subseteq S$ with $|S'| \geq |S|/2$ such that if $\vec{p}_0 = \chi_i$ for $i \in S'$, then $\sum_{j \in S} \vec{p}_t(j) \geq 1 - t \cdot \phi(S)$.*

Proof. From Lemma 11.10, if the initial distribution is $\vec{p}_0 = \chi_S/|S|$, then the total probability of leaving S in any of the t steps is at most $t \cdot \phi(S)/2$. As $\chi_S/|S|$ is a convex combination of $\{\chi_i \mid i \in S\}$,

$$\vec{p}_t = W^t \vec{p}_0 = W^t\left(\frac{\chi_S}{|S|}\right) = \frac{1}{|S|} \sum_{j \in S} W^t \chi_j.$$

By Markov's inequality, there exists $S' \subseteq S$ with $|S'| \geq |S|/2$ such that if $\vec{p}_0 = \chi_i$ for $i \in S'$, then the total probability of leaving S in any of the t steps is at most $t \cdot \phi(S)$. Then the lemma follows as in Lemma 11.10. □

This bound implies analytic sparsity by Cauchy-Schwarz.

Lemma 11.12 (Analytically Sparse Vectors from Staying Probability)**.** *Under the assumptions in Lemma 11.10 and Lemma 11.11, there exists $S' \subseteq S$ with $|S'| \geq |S|/2$ such that for $i \in S'$,*

$$\|W^t \chi_i\|_2^2 \geq \frac{1}{|S|}\big(1 - t \cdot \phi(S)\big)^2.$$

Proof. For any vertex $i \in S'$ from Lemma 11.11, by Cauchy-Schwarz

$$\|W^t \chi_i\|_2^2 \geq \sum_{j \in S} (W^t \chi_i)(j)^2 \geq \frac{1}{|S|}\Big(\sum_{j \in S} (W^t \chi_i)(j)\Big)^2 \geq \frac{1}{|S|}\big(1 - t \cdot \phi(S)\big)^2$$

□

Approximation Algorithm for Small-Set Edge Conductance

The Rayleigh quotient upper bound in Lemma 11.8 and the analytic sparsity lower bound in Lemma 11.12 can be combined to obtain an algorithm for small-set edge conductance with an approximation guarantee close to that of Cheeger's inequality.

Proof of Theorem 11.4. Let S be a subset with $\phi(S) = \phi_\delta(G)$ and $|S| \le \delta|V|$. Set

$$t = \frac{1}{2\phi(S)}.$$

From Lemma 11.12, there exists a vertex $i \in S$ with

$$\|W^t \chi_i\|_2^2 \ge \frac{1}{4|S|}.$$

From Lemma 11.8, the Rayleigh quotient of $W^t\chi_i$ satisfies

$$R(W^t\chi_i) \le 2\Big(1 - \|W^t\chi_i\|_2^{\frac{1}{t}}\Big) \le 2\Big(1 - \Big(\frac{1}{4|S|}\Big)^{\phi(S)}\Big) = 2\Big(1 - e^{-\ln(4|S|)\cdot\phi(S)}\Big) \lesssim \phi(S)\cdot\ln|S|.$$

Therefore, by Lemma 11.6, the truncated vector $y \in \mathbb{R}_+^n$ satisfies

$$R(y) \lesssim \phi(S)\cdot\ln|S| \quad \text{and} \quad |\operatorname{supp}(y)| \lesssim |S|.$$

Applying Cheeger rounding in Lemma 7.4 yields a subset $S' \subseteq V$ such that

$$\phi(S') \lesssim \sqrt{\phi(S)\cdot\ln|S|} = \sqrt{\phi_\delta(G)\cdot\ln|S|} \quad \text{and} \quad |S'| \lesssim |S| \le \delta|V|.$$

□

Local Graph Partitioning

The random walk algorithm has the significant advantage that it can be implemented locally, such that the running time depends only on the output size but not on the original graph size.

The idea is simply to truncate the random walk vector at each step. Let $\vec{q}_0 = \chi_i$ be the initial distribution, and let ϵ be a parameter. For each $t \ge 0$, define

$$\tilde{p}_t(i) = \begin{cases} 0 & \text{if } \vec{q}_t(i) < \epsilon d \\ \vec{q}_t(i) & \text{otherwise,} \end{cases} \quad \text{and} \quad \vec{q}_{t+1} = W\tilde{p}_t.$$

The truncated vector $\tilde{p}_t$ satisfies $\tilde{p}_t \le \vec{q}_t \le \tilde{p}_t + \epsilon t d \cdot \vec{1}$ and can be computed in $O(t/\epsilon)$ time. By setting

$$\epsilon \asymp \frac{\phi(S)}{td|S|} \quad \text{and} \quad t \asymp \frac{1}{\phi(S)},$$

it can be proved that the analytic sparsity and the Rayleigh quotient remain within a constant factor of that of p_t (see [ST13, KL12, KLL17] for details). This leads to the following local algorithm for computing a small sparse cut.

Theorem 11.13 (Local Algorithm for Small Sparse Cut [ST13]). *Let $G = (V, E)$ be a d-regular graph. For any unknown target set $S \subseteq V$, at least half the vertices in S have the property that if we run the truncated random walk algorithm starting from one of these vertices, then the algorithm returns a set S' satisfying*

$$\phi(S') \lesssim \sqrt{\phi(S) \cdot \ln |S|} \quad \text{and} \quad |S'| \lesssim |S|.$$

The running time of the algorithm is $O(d|S|/\phi(S)^3)$.

This provides a sublinear time algorithm for graph partitioning when $|S|$ is sufficiently small. Note that the approximation guarantee is also local in the sense that it compares to the unknown target set S rather than to the set of smallest conductance of a given size.

It is possible to remove the factor of $\ln |S|$ in the approximation guarantee, but at the cost of a worse output size bound, $|S'| \lesssim |S|^{1+\epsilon}$; see Problem 11.15. Additionally, improved approximation guarantee can be obtained when $\lambda_k(G)$ or $\phi_k(G)$ is large for a small k; see Problem 11.16.

Local graph partitioning has become an active research topic, with several alternative approaches using PageRank vectors [ACL06] and evolving sets [AOPT16]. We will briefly discuss these in the next section.

11.6 Hard Example for Local Graph Partitioning

It was conjectured in [AOPT16] that a local graph partitioning algorithm using evolving sets can be used to distinguish between the two cases of the small-set expansion Conjecture 11.2. However, a hard example was presented in [CKL17] that fooled all the existing local graph partitioning algorithms in distinguishing between the two cases.

The example is a noisy hypercube H over a large alphabet size $k = 1/\delta$. Formally, H is a graph on $n := k^l$ vertices, representing all strings of length l over alphabet $[k]$. For two vertices x, y, the edge weight $w(x, y)$ is set according to the probability of transitioning from x to y in one step of a random walk, where each symbol of x is independently re-randomized with probability ϵ, i.e., for each $i \in [l]$, with probability $1 - \epsilon$, set $y(i) = x(i)$; otherwise, $y(i)$ is sampled uniformly at random from $[k]$. Note that H is 1-regular.

It is easy to see that H has a small sparse cut. Indeed, the coordinate cut $S = \{x \in [k]^l \mid x(1) = 0\}$ has size δn and edge conductance at most ϵ.

On the other hand, all existing local graph partitioning will only explore the Hamming balls of the noisy hypercube, but the edge conductance of all Hamming balls of size $O(\delta n)$ is close to one. See the details in [CKL17].

11.7 Problems

Problem 11.14 (Local Cheeger's Inequality). *In this question, we study the relation between "local eigenvalues" and "local edge conductance". Let $G = (V, E)$ be an undirected d-regular graph, and let $\mathcal{L}$ be its normalized Laplacian matrix. Let $S \subseteq V$ be a subset of vertices with $|S| \leq |V|/2$.*

First, we define local eigenvalues. Let $\mathcal{L}_S$ be the $|S| \times |S|$ submatrix of $\mathcal{L}$ with rows and columns restricted to those indexed by vertices in S. Let λ_S be the smallest eigenvalue of $\mathcal{L}_S$. We say λ_S is the smallest local eigenvalue of S.

Next, we define local edge conductance. Let $\phi(S)$ be the edge conductance of S in G, and let $\phi^*(S) = \min_{S' \subseteq S} \phi(S')$. We say $\phi^*(S)$ is the local edge conductance of S.

Prove that

$$\phi^*(S) \geq \lambda_S \geq \frac{(\phi^*(S))^2}{2}.$$

Problem 11.15 (Improved Approximation Guarantee)**.** *Let $G = (V, E)$ be an undirected d-regular graph, and let $S \subseteq V$. Let $\vec{p}_t = W^t \vec{p}_0$ where W is the lazy random walk matrix.*

(a) Prove that there exists a probability distribution $\vec{q}$ on V such that if $\vec{p}_0 = \vec{q}$, then

$$\sum_{i \in S} \vec{p}_t(i) \geq (1 - \phi(S))^t.$$

(b) Use part (a) to prove that the random walk algorithm in this chapter has the following performance guarantee: For $S \subseteq V$ and $\epsilon > 0$, there exists a starting vertex $i \in S$ such that the random walk algorithm returns a set S' satisfying

$$\phi(S') \lesssim \sqrt{\frac{\phi(S)}{\epsilon}} \quad \text{and} \quad |S'| \lesssim |S|^{1+\epsilon}.$$

Problem 11.16 (Improved Cheeger Guarantee)**.** *Let $G = (V, E)$ be a d-regular graph. Prove that, for any unknown target set $S \subseteq V$, the random walk algorithm in this chapter returns a set S' with $|S'| \lesssim |S|$ satisfying*

$$\phi(S') \lesssim \frac{k\phi(S)}{\phi_k(G)} \quad \text{and} \quad \phi(S') \lesssim \frac{k\phi(S)}{\sqrt{\lambda_k}},$$

where $\phi_k(G)$ is the k-way edge conductance in Definition 7.1 and λ_k is the k-th smallest eigenvalue of the normalized Laplacian matrix for any $k \geq 2$.

References

[ABS15] Sanjeev Arora, Boaz Barak, and David Steurer. Subexponential algorithms for unique games and related problems. *Journal of the ACM (JACM)*, 62(5):1–25, 2015. 135, 136, 137, 139, 159, 160, 162, 166

[ACL06] Reid Andersen, Fan R. K. Chung, and Kevin J. Lang. Local graph partitioning using pagerank vectors. In *FOCS*, pages 475–486. IEEE Computer Society, 2006. 48, 143, 155

[AOPT16] Reid Andersen, Shayan Oveis Gharan, Yuval Peres, and Luca Trevisan. Almost optimal local graph clustering using evolving sets. *J. ACM*, 63(2):15:1–15:31, 2016. 48, 143, 155

[CKL17] Siu On Chan, Tsz Chiu Kwok, and Lap Chi Lau. Random walks and evolving sets: Faster convergences and limitations. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1849–1865. SIAM, 2017. 143

[KL12] Tsz Chiu Kwok and Lap Chi Lau. Finding small sparse cuts by random walk. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques - 15th International Workshop, APPROX 2012*, volume 7408 of *Lecture Notes in Computer Science*, pages 615–626. Springer, 2012. 142

[KLL17] Tsz Chiu Kwok, Lap Chi Lau, and Yin Tat Lee. Improved cheeger's inequality and analysis of local graph partitioning using vertex expansion and expansion profile. *SIAM Journal on Computing*, 46(3):890–910, 2017. 107, 135, 142

[LS93] László Lovász and Miklós Simonovits. Random walks in a convex body and an improved volume algorithm. *Random structures & algorithms*, 4(4):359–412, 1993. 50, 136

[RS10] Prasad Raghavendra and David Steurer. Graph expansion and the unique games conjecture. In *Proceedings of the forty-second ACM symposium on Theory of computing*, pages 755–764, 2010. 135, 166

[RST10] Prasad Raghavendra, David Steurer, and Prasad Tetali. Approximations for the isoperimetric and spectral profile of graphs and related parameters. In *Proceedings of the forty-second ACM symposium on Theory of computing*, pages 631–640, 2010. 136

[ST13] Daniel A. Spielman and Shang-Hua Teng. A local clustering algorithm for massive graphs and its application to nearly linear time graph partitioning. *SIAM J. Comput.*, 42(1):1–26, 2013. 8, 48, 135, 136, 140, 142, 143, 154

Chapter 12

Expander Decomposition

Graph decomposition provides an important paradigm for designing fast algorithms and approximation algorithms. The results in Chapter 11 are useful in their construction.

In this chapter, we introduce expander decomposition, study efficient construction algorithms, and mention various applications in designing fast algorithms. In the next chapter, we will introduce low threshold rank decomposition and explore its applications in designing approximation algorithms.

12.1 Expander Decomposition from Recursive Sparse Cuts

A general paradigm in algorithm design is the divide-and-conquer method. Given a graph, a natural way to divide it is through a sparse cut, and doing so recursively leads to the following approach.

Algorithm 5 DECOMPOSE(G, ϕ)

Require: An undirected graph $G = (V, E)$ and a parameter $\phi \in [0, 1]$.

1: If $\phi(G) \geq \phi$, **return** V.

2: Let $S \subseteq V$ be a set satisfying

$$\phi_G(S) = \frac{|\delta_G(S)|}{\mathrm{vol}_G(S)} < \phi \quad \text{and} \quad \mathrm{vol}_G(S) \leq \mathrm{vol}_G(V \setminus S).$$

3: **return** DECOMPOSE$(G[S], \phi) \cup$ DECOMPOSE$(G[V \setminus S], \phi)$.

This simple approach leads to an optimal existential theorem for expander decomposition.

Theorem 12.1 (Expander Decomposition)**.** *For any undirected graph $G = (V, E)$ and any $\phi \in [0, 1]$, there exists a partition of vertices $(V_1, \ldots, V_k)$ that satisfies the following two properties.*

1. *High edge conductance: Each induced subgraph $G[V_i]$ has $\phi(G[V_i]) \geq \phi$.*
2. *Few crossing edges: $|\delta(V_1, \ldots, V_k)| \lesssim \phi |E| \log |E|$, where $\delta(V_1, \ldots, V_k) := \{e \in E \mid e \in \delta(V_i)$ for some $i\}$ denotes the set of crossing edges.*

Proof. The first property is satisfied by construction.

For the second property, let G_0 be the input graph and let G be an intermediate graph during the execution of the recursive algorithm. Then,

$$|\delta(V_1,\ldots,V_k)| = \sum_S |\delta_G(S)| < \sum_S \phi \cdot \mathrm{vol}_G(S) \le \sum_S \phi \cdot \mathrm{vol}_{G_0}(S) = \phi \cdot \sum_S \sum_{i\in S} \deg_{G_0}(i),$$

where S runs over the sparse cuts found by the recursive algorithm in step (2). Note that each vertex belongs to the smaller side of a sparse cut at most $\log_2(2|E|)$ times, since the volume decreases by a factor of two at each step, and the initial volume is $2|E|$. Therefore, by a change of summation,

$$|\delta(V_1,\ldots,V_k)| = \phi \cdot \sum_{i\in V} \sum_{S:S\ni i} \deg_{G_0}(i) \le \phi \cdot \sum_{i\in V} \log_2(2|E|) \cdot \deg_{G_0}(i) = \phi \cdot 2|E| \log_2(2|E|).$$

□

Corollary: This theorem implies that for any simple graph and any $\epsilon > 0$, there exists a partition $(V_1,\ldots,V_k)$ such that

$$\phi(G[V_i]) \gtrsim \frac{\epsilon}{\log|V|} \text{ for each } 1 \le i \le k \quad \text{and} \quad \left|\delta(V_1,\ldots,V_k)\right| \lesssim \epsilon|E|. \tag{12.1}$$

Optimality: There are graphs where Theorem 12.1 is tight up to a constant factor. For hypercubes, removing a small constant fraction of edges leaves a component with edge conductance at most $O(1/\log|V|)$. See Problem 12.15.

Complexity: The only drawback of this theorem is that it does not lead to an efficient algorithm for computing the expander decomposition, as finding a set of minimum edge conductance is NP-hard. Most of the remainder of this chapter is to study how to design an efficient algorithm for finding an expander decomposition – possibly with worse parameters.

Quadratic Algorithms

Naturally, we replace an exact algorithm for minimizing edge conductance with an approximation algorithm, and the same recursive approach has the following guarantee.

Theorem 12.2 (Expander Decomposition Using Approximate Sparse Cuts [KVV04])**.** *Suppose the graph $G = (V = [n], E)$ has a partition $(V_1,\ldots,V_k)$ where*

$$\phi(G[V_i]) \ge \phi \quad \text{and} \quad |\delta(V_1,\ldots,V_k)| \le \epsilon \cdot |E|.$$

Let A be an approximation algorithm that returns a set of edge conductance at most $f(n)\cdot x^c$ for a fixed constant $c \in [0,1]$ if the minimum edge conductance is x. Then, running Algorithm 5 with step (2) replaced by using A to find an approximately minimum sparse cut returns a partition $(U_1,\ldots,U_l)$ of V such that

$$\phi(G[U_i]) \gtrsim \left(\frac{\phi}{f(n)\log\frac{n}{\epsilon}}\right)^{\frac{1}{c}} \quad \text{and} \quad |\delta(V_1,\ldots,V_k)| \lesssim f(n)\cdot\epsilon^c \cdot \log\frac{n}{\epsilon}\cdot|E|.$$

This result holds for any approximation algorithm for the minimum edge conductance problem. In particular, if we use Spectral Partitioning in Algorithm 1 (with $f(n) = \sqrt{2}$ and $c = 1/2$ from Theorem 2.2), then it outputs an expander decomposition with the following guarantee.

Corollary 12.3 (Expander Decomposition Using Spectral Partitioning [KVV04]). *Suppose the graph $G = (V = [n], E)$ has a partition $(V_1, \ldots, V_k)$ where*

$$\phi(G[V_i]) \geq \phi \quad \text{and} \quad |\delta(V_1, \ldots, V_k)| \leq \epsilon \cdot |E|.$$

Then, running Algorithm 5 with Spectral Partitioning in Algorithm 1 returns a partition $(U_1, \ldots, U_l)$ of V such that

$$\phi(G[U_i]) \gtrsim \frac{\phi^2}{\log^2 \frac{n}{\epsilon}} \quad \text{and} \quad |\delta(V_1, \ldots, V_k)| \lesssim \sqrt{\epsilon} \cdot \log \frac{n}{\epsilon} \cdot |E|.$$

Since the spectral partitioning algorithm can be implemented in nearly linear time, the recursive algorithm provides an $\tilde{O}(|E|^2)$-time algorithm for computing an expander decomposition.

Alternatively, one could use a near-linear time $O(\log n)$-approximation algorithm for the edge conductance problem (with $f(n) = \log n$ and $c = 1$), yielding another quadratic-time algorithm for expander decomposition with different parameters.

12.2 Expander Decomposition from Most-Balanced Sparse Cuts

The bottleneck of the approach in the previous section is that the recursion depth can be as large as $\Omega(n)$, which arises from spending near-linear time identifying an unbalanced sparse cut.

In this section, we explore the idea of finding a most-balanced sparse cut, so as to reduce the recursion depth to $O(\log n)$. Define

$$\mathrm{vol}_{\max}(\phi) := \max_S \{\mathrm{vol}(S) \mid \phi(S) \leq \phi \text{ and } \mathrm{vol}(S) \leq \mathrm{vol}(V)/2\} \tag{12.2}$$

as the maximum volume of a set S with edge conductance at most ϕ and volume at most half of the total volume. Assume that we have an exact algorithm for the most-balanced sparse cut problem to find a set S with $\phi(S) \leq \phi$ and $\mathrm{vol}(S) = \mathrm{vol}_{\max}(\phi)$ in step (2) of Algorithm 5.

An important observation is that if $\mathrm{vol}(S) \leq \mathrm{vol}(V)/4$, then we do not need to call the recursion on $V \setminus S$, as it already has sufficiently large edge conductance.

Lemma 12.4 (Union of Sparse Cuts is Sparse). *Let $G = (V, E)$ be a graph, and let S be a set attaining the maximum in $\mathrm{vol}_{\max}(\phi)$. If $\mathrm{vol}_G(S) \leq \mathrm{vol}_G(V)/4$, then $\phi(G[V \setminus S]) \geq \phi/3$.*

Proof. Denote $H := G[V \setminus S]$ for ease of notation. Suppose that $\phi(H) < \phi/3$. We will show that there exists a set S' with $\phi_G(S') \leq \phi$ and $\mathrm{vol}_G(S) < \mathrm{vol}_G(S') \leq \mathrm{vol}_G(V)/2$, contradicting the maximality of S in $\mathrm{vol}_{\max}(\phi)$.

Let R be a non-empty set of $V \setminus S$ with

$$\phi_H(R) = \frac{|\delta_H(R)|}{\mathrm{vol}_H(R)} < \frac{\phi}{3} \quad \text{and} \quad \phi_H(\overline{R}) = \frac{|\delta_H(\overline{R})|}{\mathrm{vol}_H(\overline{R})} < \frac{\phi}{3}.$$

Since $\mathrm{vol}_G(R) + \mathrm{vol}_G(\overline{R}) = \mathrm{vol}_G(V \setminus S)$, we assume without loss of generality that

$$\mathrm{vol}_G(R) \leq \frac{1}{2}\mathrm{vol}_G(V \setminus S) = \frac{1}{2}(\mathrm{vol}_G(V) - \mathrm{vol}_G(S)).$$

We consider two cases. The first case is that $\mathrm{vol}_G(S \cup R) \leq \mathrm{vol}_G(V)/2$. Then,

$$\phi_G(S \cup R) = \frac{|\delta_G(S \cup R)|}{\mathrm{vol}_G(S \cup R)} \leq \frac{|\delta_G(S)| + |\delta_H(R)|}{\mathrm{vol}_G(S) + \mathrm{vol}_G(R)} \leq \max\left\{ \frac{\delta_G(S)}{\mathrm{vol}_G(S)}, \frac{\delta_H(R)}{\mathrm{vol}_G(R)} \right\} \leq \phi.$$

Thus, setting $S' := S \cup R$ contradicts the maximality of S in $\mathrm{vol}_{\max}(\phi)$.

The second case is that $\mathrm{vol}_G(S \cup R) > \mathrm{vol}_G(V)/2$. Since $\mathrm{vol}_G(R) \leq \frac{1}{2}(\mathrm{vol}_G(V) - \mathrm{vol}_G(S))$,

$$\mathrm{vol}_G(S \cup R) = \mathrm{vol}_G(S) + \mathrm{vol}_G(R) \leq \frac{1}{2}(\mathrm{vol}_G(V) + \mathrm{vol}_G(S)) \leq \frac{5}{8}\mathrm{vol}_G(V).$$

This implies that $\mathrm{vol}_G(\overline{S \cup R}) \geq \frac{3}{8}\mathrm{vol}_G(V) > \mathrm{vol}_G(S)$. The numerator $|\delta_G(\overline{S \cup R})|$ is equal to

$$|\delta_G(S \cup R)| \leq |\delta_G(S)| + |\delta_H(R)| \leq \phi\,\mathrm{vol}_G(S) + \frac{\phi}{3}\mathrm{vol}_G(R) \leq \frac{5\phi}{6}\mathrm{vol}_G(S) + \frac{\phi}{6}\mathrm{vol}_G(V) \leq \frac{3\phi}{8}\mathrm{vol}_G(V).$$

Setting $S' = \overline{S \cup R}$, we obtain

$$\phi_G(S') = \frac{|\delta_G(\overline{S \cup R})|}{\mathrm{vol}_G(\overline{S \cup R})} \leq \frac{3\phi\,\mathrm{vol}_G(V)/8}{3\,\mathrm{vol}_G(V)/8} = \phi,$$

again contradicting the maximality of S in $\mathrm{vol}_{\max}(\phi)$. □

Using this lemma, we only need to recurse on both sides when $\mathrm{vol}(S) \geq \mathrm{vol}(V)/4$ and $\mathrm{vol}(\overline{S}) \geq \mathrm{vol}(V)/4$. This implies that the recursion depth of Algorithm 5 is $O(\log n)$ when step (2) is implemented using an exact most-balanced sparse cut algorithm.

Note that this provides an alternative proof of Theorem 12.1.

12.3 Fast Algorithms Using Approximate Balanced Sparse Cuts

Of course, we do not know of a fast exact algorithm for the most-balanced sparse cut problem.

Bicriteria Approximation Algorithm: Instead, we assume the existence of a fast (a, b)-bicriteria approximation algorithm for the most-balanced sparse cut problem with the following guarantee: If $\phi(G) \leq \phi$, it returns a set S with

$$\phi(S) \leq a \cdot \phi \quad \text{and} \quad \mathrm{vol}(S) \geq \mathrm{vol}_{\max}(\phi)/b, \tag{12.3}$$

for some factors $a, b \geq 1$ that could depend on the graph size (a for approximation, b for balance).

Using the cut-matching game [KRV09] that we will study in Chapter Chapter 17, one can design a near-linear time $\big(O(\log^3 n), O(\log^2 n)\big)$-bicriteria approximation algorithm for the most-balanced cut problem. See Section 17.3.

Issue: In the previous section, if the exact algorithm for the most-balanced cut problem returns a set S of small volume, then $\phi(G[V \setminus S]) \geq \phi/3$, and there is no need to recurse on $G[V \setminus S]$.

However, with only an (a, b)-bicriteria approximation algorithm for the most-balanced cut problem, we can no longer conclude that $\phi(G[V \setminus S])$ is sufficiently large. Suppose the bicriteria approximation algorithm returns a set S with $\phi(S) \leq a \cdot \phi$ and $\mathrm{vol}(S)$ small. Suppose further that $\phi(G[V \setminus S]) \ll \phi$.

Then, there exists a set R with $\phi_{G[V\setminus S]}(R) \ll \phi$. By the argument that the union of sparse cuts is sparse in Lemma 12.4, we can only conclude that $\phi(S \cup R) \leq a \cdot \phi$. This contradicts the assumption that S is the most-balanced cut with conductance at most ϕ only when $a = 1$. In general, it only establishes that S is not the most-balanced cut with conductance at most $a \cdot \phi$, but the (a, b)-bicriteria approximation algorithm does not provide such a guarantee.

Idea: Still, if the (a, b)-bicriteria approximation algorithm returns a set S with small volume, this implies that G is a "large-set expander", meaning that all sets with volume at least $b \cdot \mathrm{vol}(S)$ have conductance at least ϕ, since $\mathrm{vol}_{\max}(\phi) \leq b \cdot \mathrm{vol}(S)$. Thus, progress has been made, and it remains to examine the small-set conductance of G.

The approach in [NS17] is to reduce the conductance parameter by a factor of a, setting $\phi' := \phi/a$. By doing so, the bicriteria approximation algorithm either certifies that G is a ϕ'-expander (a weaker expansion guarantee) or returns a set S' with conductance at most $a \cdot \phi' = \phi$, for which we still have the upper bound $\mathrm{vol}(S') \leq \mathrm{vol}_{\max}(\phi) \leq b \cdot \mathrm{vol}(S)$.

Now, suppose the recursive algorithm keeps returning sets $S_1, S_2, \ldots, S_t$ with roughly the same volume as $\mathrm{vol}(S)$ (i.e., the volume remains at the same "level"). Then, this process can repeat for at most $t \approx b$ iterations, as otherwise, the union $S_1 \cup S_2 \cup \cdots \cup S_t$ would form a set with conductance at most ϕ and volume exceeding $b \cdot \mathrm{vol}(S)$, contradicting the bound $\mathrm{vol}_{\max}(\phi) \leq b \cdot \mathrm{vol}(S)$. This helps control the recursion depth on the larger side of the graph.

If the recursive algorithm returns a set S' with volume much smaller than $\mathrm{vol}(S)$, then we cannot bound the recursive depth in the same way. However, we gain the new information that all sets with volume at least $b \cdot \mathrm{vol}(S')$ must have conductance at least ϕ'. This returns us to the same situation as before, allowing us to apply the same idea recursively, further reducing the conductance parameter by another factor of a to control the recursion depth.

Algorithm: In the following formal algorithm description, ϵ is a small number, and we define $L = 1/\epsilon$. The sequence $\phi_1 \geq \phi_2 \geq \cdots \geq \phi_L = \phi$ is geometric, with $\phi_{l+1} = \phi_l/a$.

Algorithm 6 DECOMPOSE(G, l) Using Approximate Most-Balanced Sparse Cuts

Require: An undirected graph $G = (V, E)$ and a parameter $1 \leq l \leq L$.

1: Compute an approximate most-balanced cut S with $\phi(S) \leq a \cdot \phi_l$ and $\mathrm{vol}(S) \geq \mathrm{vol}_{\max}(\phi_l)/b$.
2: If the algorithm fails to find such a set S, then $\phi(G) \geq \phi_l$ and **return** V.
3: If $\mathrm{vol}(S) < |E|^{1-l\epsilon}$ (implying $\mathrm{vol}_{\max}(\phi_l) < b \cdot |E|^{1-l\epsilon}$),
return DECOMPOSE$(G, l+1)$.
4: Otherwise, if $\mathrm{vol}(S) \geq |E|^{1-l\epsilon}$ (i.e., the volume remains at the same "level"),
return DECOMPOSE$(G[S], 1) \cup$ DECOMPOSE$(G[V \setminus S], l)$.

Analysis: By using the idea of reducing the conductance parameter to control the recursion depth, the algorithm achieves the following tradeoff between its runtime and the number of edges cut.

Theorem 12.5 (Expander Decomposition from Approximate Balanced Cuts [NS17])**.** *Given a graph $G = (V, E)$, a parameter $\phi \in [0, 1]$, and an (a, b)-bicriteria approximation algorithm A for the most-balanced sparse cut problem, Algorithm 6 computes a partition $(V_1, \ldots, V_k)$ satisfying:*

1. *Each induced subgraph $G[V_i]$ has $\phi(G[V_i]) \geq \phi$.*

2. $|\delta(V_1, \ldots, V_k)| \leq \phi \cdot a^{O(1/\epsilon)} \cdot |E| \log |E|$.

Algorithm 6 calls the bicriteria approximation algorithm A *on graphs with a total of* $O(\frac{b}{\epsilon} \cdot |E|^{1+\epsilon} \log |V|)$ *edges (i.e., if* $G_1, \ldots, G_t$ *are the intermediate graphs in the recursive algorithm, the total number of edges is defined as* $\sum_i |E(G_i)|$*).*

Proof Sketch. The algorithm has at most $L = 1/\epsilon$ levels, since the condition in step (3) cannot be satisfied when $l = L$.

Step (4) can execute at most $b \cdot |E|^{\epsilon}$ consecutive times. Otherwise, the union $S_1 \cup \cdots \cup S_t$ from these consecutive iterations would form a set with conductance $a \cdot \phi_l = \phi_{l-1}$ (by the union of sparse cuts is sparse argument in Lemma 12.4) and volume exceeding $\left(b \cdot |E|^{\epsilon}\right) \cdot |E|^{1-l\epsilon} = b \cdot |E|^{1-(l-1)\epsilon}$, contradicting $\mathrm{vol}_{\max}(\phi_{l-1}) < b \cdot E^{1-(l-1)\epsilon}$.

Thus, the recursive depth is at most $\log |V| \cdot L \cdot b \cdot |E|^{\epsilon}$, where the $\log |V|$ term accounts for recursion on the smaller side of the graph. Since the graphs at each layer of the recursion tree are edge-disjoint and thus have total volume $2|E|$, the claim about the runtime follows.

As an approximate most-balanced cut S may have conductance as high as $a \cdot \phi_1 = \phi \cdot a^{L+1}$, the number of edges cut in the algorithm can be at most $\phi \cdot a^{L+1} \cdot |E| \log |E|$, as in Theorem 12.1. □

Consequence: One advantage of Theorem 12.5 is that it works with any bicriteria approximation algorithm for the most-balanced sparse cut problem. The following corollary is obtained by instantiating Theorem 12.5 with the $O(|E|^{1+o(1)})$-time $\left(O(\log^2 |V|), O(\log^2 |V|)\right)$-bicriteria approximation algorithm based on the cut-matching game, using the choice $\epsilon \asymp 1/\sqrt{\log |E|}$.

Corollary 12.6 (Fast Expander Decomposition from Approximate Balanced Cuts [NS17])**.** *Given a graph* $G = (V, E)$ *and a parameter* $\phi \in [0, 1]$*, there exists an* $\tilde{O}(|E|^{1+o(1)})$ *time algorithm that computes a partition* $(V_1, \ldots, V_k)$ *satisfying:*

1. *Each induced subgraph* $G[V_i]$ *has* $\phi(G[V_i]) \geq \phi$.
2. $|\delta(V_1, \ldots, V_k)| \leq \tilde{O}(\phi \cdot |E|^{1+o(1)})$.

Remark 12.7 (Extensions)**.** *This approach is quite general and extends to other notions of expansion, such as vertex expansion, as well as more general models, including hypergraphs and directed graphs. The cut-matching game framework is also flexible in designing bicriteria approximation algorithm for the most-balanced sparse cut problem in these different settings.*

12.4 Fast Algorithms Using Flows and Fair Cuts

The idea of using a flow algorithm to improve the solution of an approximate sparse cut algorithm was first introduced in [AL08] and later developed into local algorithms for graph partitioning [OZ14, HRW20]. This approach was adapted in [SW19] to obtain an improved expander decomposition algorithm, using the recent breakthrough in almost linear-time maximum flow algorithms [CKL+22] as a black box.

Theorem 12.8 (Fast Expander Decomposition from Flows [SW19])**.** *Given a graph* $G = (V, E)$ *and a parameter* $\phi \in [0, 1]$*, there exists an* $\tilde{O}(|E|^{1+o(1)})$ *time algorithm that computes a partition* $(V_1, \ldots, V_k)$ *of vertices satisfying:*

1. *Each induced subgraph $G[V_i]$ has $\phi(G[V_i]) \geq \phi$.*
2. $|\delta(V_1, \ldots, V_k)| \lesssim \phi \cdot |E| \cdot \log^3 |E|$.

The approach follows the framework of finding approximate balanced sparse cuts, as in the previous section. In step (3) of Algorithm 6, when the volume of the returned set S (obtained via the cut-matching game algorithm [KRV09]) is smaller than $O(|E|/\log^2 |E|)$, their key observation is that the larger side $G[\overline{S}]$ is a "near-expander".

Definition 12.9 (Near Expander [SW19])**.** *Given a graph $G = (V, E)$ and a set of vertices $R \subseteq V$, we say R is a nearly ϕ-expander in G if*

$$|E(T, V \setminus T)| \geq \phi \cdot \mathrm{vol}(T) \quad \text{for all } T \subseteq R \text{ with } \mathrm{vol}(T) \leq \mathrm{vol}(R)/2.$$

Note that if $|E(T, R \setminus T)| \geq \phi \,\mathrm{vol}(T)$ for all $T \subseteq R$ with $\mathrm{vol}(T) \leq \mathrm{vol}(R)/2$, then $\phi(G[R]) \geq \phi$.

Their idea then is to formulate a max-flow min-cut problem to "trim" $\overline{S}$, obtaining a large subset $S' \subseteq \overline{S}$ such that $\phi(G[S']) \gtrsim \phi$. The expander decomposition algorithm then just needs to recurse on $V - S'$, returning $S' \cup \textsc{Decompose}(G[V - S'], \phi)$ as the solution. This eliminates the need for parameter adjustment used in the previous section.

Consider the graph where the subset S is contracted to a single vertex s. The max-flow problem is set up as follows:

- The source s has a supply of $2|E(S, \overline{S})|/\phi$ units.
- Each vertex i in $\overline{S}$ is a sink with a demand of $\deg(i)$ units.
- Every edge in the contracted graph has a capacity $2/\phi$.

Now, suppose $\overline{S}$ is a nearly ϕ-expander but $\phi(G[\overline{S}]) \leq \phi/6$. Then, there exists a set $T \subseteq \overline{S}$ such that $|E(S, T)| \geq 5|E(T, \overline{S} \setminus T)|$. It follows from a straightforward calculation that the flow problem is infeasible [SW19, Proposition 3.2]; see Problem 12.17.

Thus, if the flow problem is feasible, it certifies that $\phi(G[\overline{S}]) \geq \phi/6$, and we are done. If the flow problem is infeasible, let T be a minimal min-cut in the contracted graph that contains s. Using that T is an *exact* minimum cut, it is proved in [SW19, Section 3.2] that $S' := \overline{S} \setminus T$ satisfies $G[S'] \geq \phi/6$, and again we are done. See Problem 12.17.

Recently, the notion of a "fair cut" was introduced [LNPS23] to enable the use of approximate max-flow algorithms in place of exact max-flow algorithms for designing fast algorithms for various cut problems. By employing fair cuts and a near linear-time approximate max-flow algorithm [Pen16] for the trimming step, this approach leads to a nearly linear-time algorithm for computing an expander decomposition that comes close to matching the optimal guarantee in Theorem 12.1.

Theorem 12.10 (Fast Expander Decomposition from Fair Cuts [LNPS23])**.** *Given a graph $G = (V, E)$ and a parameter $\phi \in [0, 1]$, there exists an $\tilde{O}(|E|)$ time algorithm that computes a partition $(V_1, \ldots, V_k)$ of vertices satisfying:*

1. *Each induced subgraph $G[V_i]$ has $\phi(G[V_i]) \geq \phi$.*
2. $|\delta(V_1, \ldots, V_k)| \lesssim \phi \cdot |E| \cdot \log^3 |E|$.

Question 12.11 (Deterministic Algorithms)**.** *It remains an open question whether a near linear-time deterministic algorithm can be designed for expander decomposition, as in Theorem 12.10. The best known deterministic algorithm runs in almost linear-time algorithm [CGL+20].*

12.5 Balanced Sparse Cuts from Local Graph Partitioning

Local graph partitioning algorithms can also be used to find an approximate balanced sparse cut in nearly linear time [ST13]. The intuition is that these algorithms output a sparse cut S in time proportional to $\mathrm{vol}(S)$. Thus, if $\mathrm{vol}(S)$ is small, the runtime remains small, bypassing the issue of spending nearly linear time to find an unbalanced sparse cut, as encountered in the previous sections. In this section, we provide a rough sketch of the algorithm and its analysis.

Recall the local graph partitioning algorithm by truncated random walks in Section 11.5. The algorithm has the property that for any sparse cut S, at least half of the vertices inside it are good, meaning that if we start a truncated random walk with parameter ϵ from a good vertex, the returned set C satisfies:

1. $\phi(C) \lesssim \sqrt{\phi(S) \cdot \ln |S|}$,
2. $\mathrm{vol}(C) \lesssim \mathrm{vol}(S)$, and
3. $\mathrm{vol}(C \cap S) \gtrsim 1/\epsilon$.

The first two properties follow from the analysis in Section 11.5, while the third property requires a more involved argument [ST13, Theorem 2.1].

To find a balanced sparse cut, their approach repeatedly identifies a random sparse cut as follows:

1. Choose a random vertex i with probability proportional to $\deg(i)$.
2. Choose a random ϵ such that $\Pr(\epsilon \approx 2^{-k}) \propto 2^{-k}$ for $1 \le k \le \log |E|$.
3. Run a truncated random walk starting from vertex i with truncation parameter ϵ.

The choice of the probability distribution on ϵ is to ensure that the expected running time of the algorithm is $O(\mathrm{polylog}\,|E|/\operatorname{poly}\phi)$. Using the third property above, the returned set C satisfies $\mathbb{E}\left[\mathrm{vol}(C \cap S)\right] \gtrsim \mathrm{vol}(S)/\mathrm{vol}(V)$ [ST13, Lemma 3.1].

They then proved that the union of these random sparse cuts, $C' := C_1 \cup \cdots \cup C_t$, either satisfies $\mathrm{vol}(C') \ge \mathrm{vol}(V)/4$, in which case a balanced sparse cut is found, or $\mathrm{vol}(C' \cap S) \ge \mathrm{vol}(S)/2$ for every sparse cut S, certifying that no balanced sparse cut exists in the graph [ST13, Theorem 3.2].

Theorem 12.12 (Balanced Sparse Cut from Random Walks [ST13])**.** *Given a graph $G = (V, E)$ and a parameter ϕ, there exists a random-walk-based algorithm with expected running time $\tilde{O}(|E|/\phi^4)$ and the following approximation guarantee. If there exists a set S with $\mathrm{vol}(S) \le \mathrm{vol}(V)/2$ and $\phi(S) \lesssim \phi^3/\log^2 |E|$, then the algorithm returns a set C satisfying $\phi(C) \le \phi$ and $\frac{1}{2}\mathrm{vol}(S) \le \mathrm{vol}(C) \le \frac{5}{6}\mathrm{vol}(V)$ with high probability.*

Expander Decomposition from Local Graph Partitioning

Spielman and Teng used this balanced sparse cut algorithm from local graph partitioning to develop a nearly linear-time algorithm for expander decomposition, which was then applied to design a nearly linear time algorithm for the spectral sparsification problem [ST11].

To achieve nearly linear runtime, they proved that if this balanced sparse cut algorithm returns a set S of small volume, then the complement $\overline{S}$ is contained in a set R such that $G[R]$ is an expander. Consequently, they do not recurse on $G[\overline{S}]$.

Theorem 12.13 (Expander Decomposition from Random Walks [ST11])**.** *Given a graph $G = (V, E)$ and a parameter $\phi \in [0, 1]$, there exists an $\tilde{O}(|E|/\phi^4)$ time algorithm that computes a partition $(V_1, \ldots, V_k)$ of vertices satisfying:*

1. *For each V_i, there exists $W_i \supseteq V_i$ such that $\phi(G[W_i]) \geq \phi$.*
2. *$|\delta(V_1, \ldots, V_k)| \lesssim \phi^{1/c_1} \cdot |E| \cdot \log^{c_2} |E|$ for some constants $c_1, c_2 \geq 1$.*

This expander decomposition suffices for their spectral sparsification purpose but is weaker than the requirement in previous sections, where $G[\overline{S}]$ itself needed to be an expander.

Remark 12.14 (Trimming)**.** *One can prove that when $\overline{S}$ satisfies the property of being contained in an expander, then $G[\overline{S}]$ is a nearly expander, as defined in Definition 12.9. Hence, the flow technique from Section 12.4 can be applied to trim $\overline{S}$, obtaining an expander decomposition where each induced subgraph is an expander.*

Expander Decomposition from Other Graph Partitioning Algorithms

Both the PageRank algorithm [ACL06] and the evolving set algorithm [AOPT16] provide better approximation guarantee and faster running times than the truncated random walks algorithm for approximating sparse cuts, approximating balanced sparse cuts, and computing expander decompositions. But they are not as competitive as the algorithms using cut-matching games and flows.

There is also an SDP-based algorithm [OV11] for finding balanced sparse cuts and computing expander decomposition. This algorithm achieves both faster running time (with no dependence on ϕ) and improved approximation guarantee comparing to the local graph partitioning algorithms.

12.6 Expander Hierarchy and Applications

Fast expander decomposition has found many applications in designing graph algorithms, including spectral sparsification and Laplacian solvers [ST11, CKP+17], as well as graph connectivity problems [Chu12, Sar21].

By applying expander decomposition recursively to the contracted graph – where each component V_i is contracted to a single vertex v_i – one can build an *expander hierarchy* on the graph. See, e.g., [Räc08, GRST21] for the formal definition.

This concept has led to even more powerful applications, such as fast algorithms for oblivious routing [Räc08], exact maximum flow [VDBCK+24, BBST24], and numerous dynamic graph algorithms [NS17, CGL+20, GRST21].

12.7 Problems

Problem 12.15 (Expander Decomposition for Hypercubes)**.** *Prove that, for hypercubes $G = (V, E)$, removing a small constant fraction of edges leaves a component with edge conductance at most $O(1/\log |V|)$.*

Exercise 12.16 (Near Expander)**.** *Show that if $V \subseteq W$ and $G[W]$ is an expander, then V is a nearly expander.*

Problem 12.17 (Trimming)**.** *Show that the flow problem in Section 12.4 is infeasible if $\phi(G[\overline{S}]) \leq \phi/6$. Prove also that if the flow problem is infeasible, then $\phi(G[\overline{S} \setminus T]) \geq \phi/6$, where T is a minimal min-cut in the contracted graph.*

References

[ACL06] Reid Andersen, Fan R. K. Chung, and Kevin J. Lang. Local graph partitioning using pagerank vectors. In *FOCS*, pages 475–486. IEEE Computer Society, 2006. 48, 143, 155

[AL08] Reid Andersen and Kevin J Lang. An algorithm for improving graph partitions. In *SODA*, volume 8, pages 651–660, 2008. 152

[AOPT16] Reid Andersen, Shayan Oveis Gharan, Yuval Peres, and Luca Trevisan. Almost optimal local graph clustering using evolving sets. *J. ACM*, 63(2):15:1–15:31, 2016. 48, 143, 155

[BBST24] Aaron Bernstein, Joakim Blikstad, Thatchaphol Saranurak, and Ta-Wei Tu. Maximum flow by augmenting paths in $n^{2+o(1)}$ time. In *2024 IEEE 65th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 2056–2077. IEEE, 2024. 155

[CGL+20] Julia Chuzhoy, Yu Gao, Jason Li, Danupon Nanongkai, Richard Peng, and Thatchaphol Saranurak. A deterministic algorithm for balanced cut with applications to dynamic connectivity, flows, and beyond. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1158–1167. IEEE, 2020. 154, 155

[Chu12] Julia Chuzhoy. On vertex sparsifiers with steiner nodes. In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, pages 673–688, 2012. 155

[CKL+22] Li Chen, Rasmus Kyng, Yang P Liu, Richard Peng, Maximilian Probst Gutenberg, and Sushant Sachdeva. Maximum flow and minimum-cost flow in almost-linear time. In *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 612–623. IEEE, 2022. 152, 220, 221

[CKP+17] Michael B Cohen, Jonathan Kelner, John Peebles, Richard Peng, Anup B Rao, Aaron Sidford, and Adrian Vladu. Almost-linear-time algorithms for markov chains and new spectral primitives for directed graphs. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 410–419, 2017. 155

[GRST21] Gramoz Goranci, Harald Räcke, Thatchaphol Saranurak, and Zihan Tan. The expander hierarchy and its applications to dynamic graph algorithms. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2212–2228. SIAM, 2021. 155

[HRW20] Monika Henzinger, Satish Rao, and Di Wang. Local flow partitioning for faster edge connectivity. *SIAM Journal on Computing*, 49(1):1–36, 2020. 152

[KRV09] Rohit Khandekar, Satish Rao, and Umesh Vazirani. Graph partitioning using single commodity flows. *Journal of the ACM (JACM)*, 56(4):1–15, 2009. 9, 150, 153, 213, 215, 216, 218, 219, 220, 221

[KVV04] Ravi Kannan, Santosh Vempala, and Adrian Vetta. On clusterings: Good, bad and spectral. *Journal of the ACM (JACM)*, 51(3):497–515, 2004. 148, 149

[LNPS23] Jason Li, Danupon Nanongkai, Debmalya Panigrahi, and Thatchaphol Saranurak. Near-linear time approximations for cut problems via fair cuts. In *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 240–275. SIAM, 2023. 153

[NS17] Danupon Nanongkai and Thatchaphol Saranurak. Dynamic spanning forest with worst-case update time: adaptive, las vegas, and $o(n^{1/2-\varepsilon})$-time. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1122–1129, 2017. 151, 152, 155, 221, 222

[OV11] Lorenzo Orecchia and Nisheeth K Vishnoi. Towards an sdp-based approach to spectral methods: A nearly-linear-time algorithm for graph partitioning and decomposition. In *Proceedings of the twenty-second annual ACM-SIAM symposium on Discrete Algorithms*, pages 532–545. SIAM, 2011. 155

[OZ14] Lorenzo Orecchia and Zeyuan Allen Zhu. Flow-based algorithms for local graph clustering. In *Proceedings of the twenty-fifth annual ACM-SIAM symposium on Discrete algorithms*, pages 1267–1286. SIAM, 2014. 152

[Pen16] Richard Peng. Approximate undirected maximum flows in $o(m\,\mathrm{polylog}(n))$ time. In *Proceedings of the twenty-seventh annual ACM-SIAM symposium on Discrete algorithms*, pages 1862–1867. SIAM, 2016. 153, 221, 222

[Räc08] Harald Räcke. Optimal hierarchical decompositions for congestion minimization in networks. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 255–264, 2008. 155

[Sar21] Thatchaphol Saranurak. A simple deterministic algorithm for edge connectivity. In *Symposium on Simplicity in Algorithms (SOSA)*, pages 80–85. SIAM, 2021. 155

[ST11] Daniel A Spielman and Shang-Hua Teng. Spectral sparsification of graphs. *SIAM Journal on Computing*, 40(4):981–1025, 2011. 8, 155

[ST13] Daniel A. Spielman and Shang-Hua Teng. A local clustering algorithm for massive graphs and its application to nearly linear time graph partitioning. *SIAM J. Comput.*, 42(1):1–26, 2013. 8, 48, 135, 136, 140, 142, 143, 154

[SW19] Thatchaphol Saranurak and Di Wang. Expander decomposition and pruning: Faster, stronger, and simpler. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2616–2635. SIAM, 2019. 152, 153

[VDBCK+24] Jan Van Den Brand, Li Chen, Rasmus Kyng, Yang P Liu, Simon Meierhans, Maximilian Probst Gutenberg, and Sushant Sachdeva. Almost-linear time algorithms for decremental graphs: Min-cost flow and more via duality. In *2024 IEEE 65th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 2010–2032. IEEE, 2024. 155

Chapter 13

Threshold Rank Decomposition

In this chapter, we consider a spectral approach to graph decomposition that is analogous to expander decomposition and study its applications in designing approximation algorithms.

First, when the graph has few small Laplacian eigenvalues, we show how to find a small sparse cut using the subspace enumeration method introduced by Kolla [Kol11]. Combined with the Cheeger inequality for small-set edge conductance in Theorem 11.3, this provides a subexponential time algorithm for the small-set expansion conjecture in Conjecture 11.2.

Next, we show how to apply Theorem 11.3 to decompose the graph into components with few small Laplacian eigenvalues, and use this decomposition to obtain a subexponential time algorithm for the Unique Games Conjecture [ABS15]. We conclude with some interesting examples and discuss the limitations of this spectral approach [BGH+15].

Threshold Rank and Graph Decomposition

The definition of threshold rank is to formalize the notion of graphs with few small eigenvalues.

Definition 13.1 (Threshold Rank)**.** *Let G be a graph and $\lambda \in [0,1]$ be a parameter. The λ-threshold rank of G, denoted by $\mathrm{rank}_\lambda(G)$, is defined as the number of eigenvalues of the normalized Laplacian matrix $\mathcal{L}(G)$ that are smaller than λ.*[1]

An expander graph G has $\mathrm{rank}_{\lambda_2}(G) = 1$, where λ_2 is the second smallest eigenvalue of the normalized Laplacian matrix. More generally, if a graph $G = (V, E)$ has low threshold rank, meaning $\mathrm{rank}_\lambda(G) \le r$ for a large λ and a small r, then the graph has large multi-way edge conductance $\phi_{r+1}(G) > \lambda$, as shown by the higher-order Cheeger inequality in Theorem 7.2.

Thus, such a graph G is expected to admit an expander decomposition into at most r components, where each component is a λ-expander, and the number of crossing edges between components is at most $\tilde{O}(\sqrt{\lambda} \cdot |E|)$. This intuition underlies the idea that a low threshold rank graph is a graph with low complexity. Morally, one may also think of a low threshold rank graph as a small-set expander.

[1] This definition differs from that in [ABS15], as it is based on the normalized Laplacian matrix rather than the normalized adjacency matrix, and thus loses the direct connection to the traditional rank.

13.1 Small-Set Expansion via Subspace Enumeration

The main idea of the subspace enumeration method [Kol11, ABS15] is that if a graph $G = (V, E)$ has low threshold rank, then the characteristic vector χ_S of a sparse cut $S \subseteq V$ must have a large projection onto the eigenspace corresponding to small Laplacian eigenvalues. Since this eigenspace has low dimension, it is feasible to discretize it using an epsilon-net and enumerate over all vectors in the epsilon-net to identify one with a high correlation to χ_S, thereby recovering a sparse cut S' with a small symmetric difference to S.

Sparse Cut from Close Vector in Top Eigenspace

First, we show that the characteristic vector of a sparse cut must be close to the eigenspace of small eigenvalues. In the following, let $\bar{\chi}_S := \chi_S / \|\chi_S\|_2$ denote the normalized characteristic vector of S with $\|\bar{\chi}_S\|_2 = 1$.

Lemma 13.2 (Distance Between Sparse Cut and Top Eigenspace)**.** *Let $G = (V = [n], E)$ be a graph. Let $S \subseteq V$ be a set with $\phi(S) \leq \phi$. Suppose $\mathrm{rank}_\lambda(G) \leq r$. Then, there exists a vector $y \in \mathrm{span}\{v_1, \ldots, v_r\}$ with $\|y\|_2 \leq 1$ such that*

$$\left\|\bar{\chi}_S - y\right\|_2^2 \leq \frac{\phi}{\lambda},$$

where $v_1, \ldots, v_r$ are the eigenvectors corresponding to the r smallest eigenvalues of $\mathcal{L}(G)$.

Proof. The idea is to simply project $\bar{\chi}_S$ onto the top-r eigenspace. Write $\bar{\chi}_S = \sum_{i=1}^n c_i v_i$, where $\{v_1, \ldots, v_n\}$ are the orthonormal eigenvectors of $\mathcal{L}(G)$ with eigenvalues $\lambda_1, \ldots, \lambda_n$. Since $\phi(S) \leq \phi$, it follows from Lemma 2.4 that $R(\bar{\chi}_S) \leq \phi$. Since $\mathrm{rank}_\lambda(G) \leq r$,

$$\phi \geq R(\bar{\chi}_S) = \frac{\bar{\chi}_S^\top \mathcal{L} \bar{\chi}_S}{\|\bar{\chi}_S\|_2^2} = \Big(\sum_{i=1}^n c_i v_i\Big)^\top \mathcal{L}\Big(\sum_{i=1}^n c_i v_i\Big) = \sum_{i=1}^n c_i^2 \lambda_i \geq \sum_{i=r+1}^n c_i^2 \lambda.$$

This implies that $\sum_{i=r+1}^n c_i^2 \leq \phi/\lambda$. Let $y := \sum_{i=1}^r c_i v_i$. Then,

$$\|\bar{\chi}_S - y\|_2^2 = \Big\|\sum_{i=r+1}^n c_i v_i\Big\|_2^2 = \sum_{i=r+1}^n c_i^2 \leq \frac{\phi}{\lambda}.$$

Clearly, $y \in \mathrm{span}\{v_1, \ldots, v_r\}$ and $\|y\|_2 \leq 1$, so the lemma follows. □

Next, given a vector y that is close to some (unknown) sparse cut S, we show how to extract a sparse cut S' from y that has a small symmetric difference to S.

Lemma 13.3 (Sparse Cut from y)**.** *Let $G = (V = [n], E)$ be a d-regular graph. Given a vector $y \in \mathbb{R}^n$ that satisfies $\|y\|_2 \leq 1$ and $\|y - \bar{\chi}_S\|_2^2 \leq \epsilon \leq \frac{1}{8}$ for some (unknown) set $S \subseteq V$, the set*

$$S' := \left\{ i \;\middle|\; y(i) \geq \frac{1}{2\sqrt{|S|}} \right\} \quad \textit{satisfies} \quad |S'| \lesssim |S| \quad \textit{and} \quad \phi(S') \lesssim \phi(S) + \epsilon.$$

Proof. Since $\|y\|_2^2 \le 1$ and each vertex in S' contributes at least $1/(4|S|)$ to $\|y\|_2^2$, it follows that $|S'| \le 4|S|$.

Let $S \Delta S'$ denote the symmetric difference between S and S'. Since $\|y - \bar{\chi}_S\|_2^2 \le \epsilon$ and each vertex in $S\Delta S'$ contributes at least $1/(4|S|)$ to $\|y - \bar{\chi}_S\|_2^2$, it follows that $|S\Delta S'| \le 4\epsilon|S| \le |S|/2$. Then,

$$\phi(S') = \frac{|\delta(S')|}{d|S'|} \le \frac{|\delta(S)| + |\delta(S\Delta S')|}{d|S| - d|S\Delta S'|} \le \frac{|\delta(S)| + d|S\Delta S'|}{\frac{1}{2}d|S|} \le 2\phi(S) + 8\epsilon.$$

□

To summarize, Lemma 13.2 shows that if there is a small sparse cut S in the graph, then there exists a vector y in the top eigenspace such that y is close to $\bar{\chi}_S$. From such a vector y, Lemma 13.3 shows that a sparse cut S' can be extracted with size and conductance close to that of S. Note that S' can be found without prior knowledge of S by simply testing all threshold sets of the vector y.

Epsilon-Net and Subspace Enumeration

Following the discussion in the previous subsection, our task is to find $y \in \text{span}\{v_1, \ldots, v_r\}$ with $\|y\|_2 \le 1$ such that y is close to a small sparse cut S.

The idea of subspace enumeration is to simply discretize the top eigenspace and search for such a vector y exhaustively.

Definition 13.4 (ϵ-Net)**.** *Let U be a set of points. A subset $U' \subseteq U$ is called an ϵ-net of U if for every point $u \in U$ there exists a point $u' \in U'$ such that $\|u - u'\|_2 \le \epsilon$.*

We use the following upper bound on the size of an ϵ-net of the unit ball, which follows from a simple volume argument.

Lemma 13.5 (ϵ-Net of Unit Ball)**.** *Let B^r be the unit ball in $\mathbb{R}^r$. For $\epsilon \le 1$, there exists an ϵ-net of B^r with at most $\left(\frac{3}{\epsilon}\right)^r$ points.*

Proof. Starting from $B' := \emptyset$, we add a point $p \in B^r$ to B' if there is no point in B' with Euclidean distance at most ϵ to p, and repeat until B' forms an ϵ-net. Let $p_1, \ldots, p_M$ be the points in B'. Observe that their pairwise distance is at least ϵ, as otherwise they would not be added to B'. Thus, the $\frac{\epsilon}{2}$-balls centered at the p_i's are disjoint.

Each $\frac{\epsilon}{2}$-ball has volume $c_r \cdot \left(\frac{\epsilon}{2}\right)^r$, where c_r is a constant depending on r only, so the total volume of these $\frac{\epsilon}{2}$-balls is $M \cdot c_r \cdot \left(\frac{\epsilon}{2}\right)^r$. On the other hand, all these balls are contained within the $(1+\frac{\epsilon}{2})$-ball in $\mathbb{R}^r$, which has volume $c_r \cdot \left(1+\frac{\epsilon}{2}\right)^r \le c_r \cdot (\frac{3}{2})^r$. Therefore, we obtain the inequality $M \cdot c_r \cdot \left(\frac{\epsilon}{2}\right)^r \le c_r \cdot (\frac{3}{2})^r$, which implies that $M \le \left(\frac{3}{\epsilon}\right)^r$. □

Algorithm 7 The subspace enumeration algorithm

Require: A d-regular graph $G = (V = [n], E)$ with $\text{rank}_\lambda(G) \le r$, and a parameter $\phi > 0$.

1: Compute the subspace U spanned by the first r eigenvectors $v_1, \ldots, v_r$ of $\mathcal{L}(G)$.
2: Let U' be an $\sqrt{\phi/\lambda}$-net for the vectors $u \in U$ with $\|u\|_2 \le 1$.
3: For each $y' \in U'$, for each $1 \le s \le n$, output the set $S = \{i \mid y'(i) \ge 1/(2\sqrt{s})\}$.

Theorem 13.6 (Subspace Enumeration). *Let $G = (V, E)$ be a d-regular graph with $\operatorname{rank}_\lambda(G) \le r$. For any set $S \subseteq V$ with $\phi(S) \le \phi$, the subspace enumeration algorithm outputs a set S' with $|S'| \lesssim |S|$ and $\phi(S') \lesssim \phi/\lambda$ with a running time $\tilde{O}\big(|V| \cdot \exp(O(r \log \frac{\lambda}{\phi}))\big)$.*

Proof. Fix a set $S \subseteq V$ with $\phi(S) \le \phi$. By Lemma 13.2, there exists a vector $y \in U$ with $\|y\|_2 \le 1$ and $\|\bar{\chi}_S - y\|_2 \le \sqrt{\phi/\lambda}$. By Definition 13.4, there exists a vector $y' \in U'$ with $\|y - y'\|_2 \le \sqrt{\phi/\lambda}$, and thus $\|y' - \bar{\chi}_S\|_2 \le 2\sqrt{\phi/\lambda}$. Since step (3) of the subspace enumeration algorithm tries every $1 \le s \le n$, it follows from Lemma 13.3 that the algorithm will output a set S' satisfying

$$|S'| \lesssim |S| \quad \text{and} \quad \phi(S') \lesssim \phi(S) + \frac{\phi}{\lambda} \lesssim \frac{\phi}{\lambda}.$$

The runtime of the algorithm is $\tilde{O}(|U'| \cdot |V|) = \tilde{O}\big(|V| \cdot (3\sqrt{\lambda/\phi})^r\big)$ by Lemma 13.5. □

13.2 Subexponential Algorithm for Small-Set Expansion Conjecture

To obtain a subexponential-time algorithm for distinguishing the two cases of the small-set expansion conjecture in Conjecture 11.2, we apply the subspace enumeration algorithm in Algorithm 7 when the threshold rank is low, and use the Cheeger inequality for small-set edge conductance in Theorem 11.3 when the threshold rank is high.

Theorem 13.7 (Subexponential Algorithm for Small-Set Expansion [ABS15]). *Let $G = (V = [n], E)$ be a d-regular graph. Suppose there exists a set S with $|S| \le \delta n$ and $\phi(S) \le \phi$ for constants δ and ϕ. There exists an algorithm with running time $\exp(n^{O(\phi^{1/4})})$ that returns a set S' with*

$$|S'| \lesssim \delta n \quad \textit{and} \quad \phi(S') \lesssim \phi^{1/4}.$$

Proof. Let λ be a parameter (to be chosen as $\phi^{3/4}$), and define $r := \operatorname{rank}_\lambda(G)$.

If $r \ge n^{2\beta}$ for some constant β (to be chosen as $\phi^{1/4}$), then Theorem 11.3 provides a polynomial time algorithm that returns a set S' with

$$\phi(S') \lesssim \sqrt{\frac{\lambda}{\beta}} \quad \text{and} \quad |S'| \lesssim n^{1-\Omega(\beta)} \ll \delta n,$$

thus the high threshold rank case is actually easy by the Cheeger-type inequality in Theorem 11.3.

For the low threshold rank case, when $r < n^{2\beta}$, Theorem 13.6 provides an algorithm that finds a set S' with

$$\phi(S') \lesssim \frac{\phi}{\lambda} \quad \text{and} \quad |S'| \lesssim |S| \le \delta n \quad \text{in time} \quad \tilde{O}\left(n \cdot \exp\left(O\left(n^{2\beta} \log \frac{\lambda}{\phi}\right)\right)\right).$$

Choosing $\lambda = \phi^{3/4}$ and $\beta = \phi^{1/4}$ balances and approximately minimizes the two conductance bounds while keeping the running time subexponential, ensuring that

$$\phi(S') \lesssim \phi^{1/4} \quad \text{and} \quad |S'| \lesssim \delta n \quad \text{in time} \quad \tilde{O}\left(\exp\left(n^{O(\phi^{1/4})}\right)\right).$$

□

In the first case of Conjecture 11.2 when $\phi_\delta(G) \le \phi$, Theorem 13.7 provides a subexponential time algorithm that outputs a set S' with $|S'| \le 4\delta n$ and $\phi(S') \lesssim \phi^{1/4}$. This shows that $\phi_{4\delta}(G) \lesssim \phi^{1/4} < 1 - \phi$, bringing it close to distinguish it from the second case.

One way to complete the proof is to show that there exists a subset $S \subseteq S'$ with $|S| = \delta n$ and $\phi(S) < 1 - \phi$ using a simple probabilistic argument; see Problem 13.15.

Another approach is to show that $|S'|$ can be made arbitrarily close to δn by allowing $\phi(S')$ to be larger (modifying Lemma 13.3). Then, removing extra vertices until the size reach δn would do.

13.3 Low Threshold Rank Graph Decomposition

The Cheeger-type inequality for small-set edge conductance in Theorem 11.3 provides a spectral approach to graph decomposition.

Theorem 13.8 (Low Threshold Rank Graph Decomposition)**.** *For any d-regular graph $G = (V, E)$ and any $\phi \in [0, 1]$, there exists a partition of vertices $(V_1, \dots, V_k)$ satisfying:*

1. *Low threshold rank: Each induced subgraph $G[V_i]$ has $\operatorname{rank}_{\phi^5}(G[V_i]) \le |V|^{2\phi}$.*
2. *Few crossing edges: $|\delta(V_1, \dots, V_k)| \lesssim \phi \cdot \log \frac{1}{\phi} \cdot |E|$.*

Comparing with the expander decomposition in Theorem 12.1, if the number of crossing edges is a small constant fraction of the number of edges, this theorem guarantees that each induced subgraph has threshold rank at most $n^{O(\phi)}$, whereas Theorem 12.1 guarantees that each induced subgraph has edge conductance $\Omega(1/\log n)$. These guarantees are incomparable, even though a low threshold rank graph can be qualitatively understood as a small-set expander using Theorem 7.2.

The proof of Theorem 13.8 follows from a simple recursive algorithm, similar to the one for expander decomposition in Algorithm 5. In the following, let N be the number of vertices in the input graph, and let $\mathring{G}[S]$ denote the induced subgraph on S with additional self-loops added to ensure that the induced subgraph remains d-regular.

Algorithm 8 RANK-DECOMPOSE(G, ϕ)

Require: A d-regular graph $G = (V, E)$ and a parameter $\phi \in [0, 1]$.

1: If $\operatorname{rank}_{\phi^5}(G) \le N^{2\phi}$, **return** V.

2: Apply Theorem 11.3 with $r = \operatorname{rank}_{\phi^5}(G)$, $\beta = 2\phi$, and $\epsilon = \frac{1}{2}$ to obtain $S \subseteq V$ with

$$|S| \lesssim |V|^{1-\beta(1-\epsilon)} = |V|^{1-\phi} \quad \text{and} \quad \phi_G(S) \lesssim \sqrt{\frac{\lambda_r}{\epsilon\beta}} \le \sqrt{\frac{\phi^5}{\phi}} = \phi^2.$$

3: **return** RANK-DECOMPOSE$(\mathring{G}[S], \phi) \cup$ RANK-DECOMPOSE$(\mathring{G}[V \setminus S], \phi)$.

Proof. The proof is similar to that of Theorem 12.1. The first property is satisfied by construction.

For the second property, let G_0 be the input graph and let G be an intermediate graph during the execution of the recursive algorithm. Since $\operatorname{vol}_G(S) = \operatorname{vol}_{G_0}(S)$ due to the self-loops,

$$|\delta(V_1, \dots, V_k)| = \sum_S |\delta_G(S)| \lesssim \sum_S \phi^2 \cdot \operatorname{vol}_G(S) = \sum_S \phi^2 \cdot \operatorname{vol}_{G_0}(S) = \phi^2 \cdot \sum_S \sum_{i \in S} \deg_{G_0}(i),$$

where S runs over the sparse cuts found by the recursive algorithm in step (2).

We claim that each vertex belongs to the smaller side of a sparse cut at most $O\big(\frac{1}{\phi}\log\frac{1}{\phi}\big)$ times. Assuming the claim, by a change of summation, the second property follows as

$$|\delta(V_1,\ldots,V_k)| = \phi^2 \cdot \sum_{i\in V}\sum_{S:S\ni i} \deg_{G_0}(i) \lesssim \phi^2 \cdot \sum_{i\in V} \frac{1}{\phi}\log\frac{1}{\phi}\cdot\deg_{G_0}(i) = \phi\cdot\log\frac{1}{\phi}\cdot 2|E|.$$

It remains to prove the claim. Suppose vertex i belongs to $V(G_0)\supset S_1\supset S_2\supset\cdots\supset S_{t+1}$, where each S_i is the smaller side of a sparse cut found in step (2) of the algorithm. Since the threshold rank is at most the number of vertices, $|S_t|\geq N^{2\phi}$ as otherwise the recursion would stop. Since $|S_{i+1}|\leq c\cdot|S_i|^{1-\phi}$ for some constant c by step (2) of the algorithm, it follows that

$$N^{2\phi}\leq|S_t|\leq c\cdot|S_{t-1}|^{(1-\phi)}\leq\cdots\leq c^t\cdot N^{(1-\phi)^t}\leq c^t\cdot N^{e^{-\phi\cdot t}}.$$

This implies that $t\lesssim\frac{1}{\phi}\log\frac{1}{\phi}$ and completes the proof. □

13.4 Subexponential Algorithm for Unique Games Conjecture

Using the low threshold rank decomposition in Theorem 13.8, the subspace enumeration method can be extended to provide a subexponential time algorithm for the Unique Games Conjecture.

Definition 13.9 (Max2Lin(k)). *Given n variables $x_1,\ldots,x_n\in\mathbb{Z}_k$ and m linear equations of the form*

$$x_i\equiv x_j+c_{ij}\mod k,$$

where c_{ij} is a different constant for each equation, the Max2Lin(k) problem is to find an assignment to the variables to satisfy the maximum number of constraints.

This is called a Unique Games problem because, for each equation, any value assigned to one variable uniquely determines the value of the other variable that satisfies the equation.

For any Unique Games problem, it is easy to determine whether there exists an assignment that satisfies all the equations. However, the Unique Games Conjecture by Khot states that it is difficult to distinguish between instances that are almost satisfiable and those that are far from satisfiable. The conjecture in the special case Max2Lin(k) is equivalent to the conjecture for a general Unique Games problem.

Conjecture 13.10 (Unique Games Conjecture [Kho02]). *For any constant $\epsilon>0$, there exists a constant k such that it is NP-hard to distinguish between the following two cases:*

1. *The Max2Lin(k) problem has an assignment that satisfies at least $(1-\epsilon)$-fraction of equations.*
2. *The Max2Lin(k) problem has no assignment that satisfies at least ϵ-fraction of equations.*

Informally, the conjecture states that the Max2Lin(k) problem becomes harder as the alphabet size increases. If true, the conjecture would imply optimal inapproximability results for various combinatorial optimization problems, including Minimum Vertex Cover and Maximum Cut.

Unique Games and Small-Set Expansion

The connection between the Unique Games problem and the Small-Set Expansion problem can be seen by considering the label-extended graph of a unique games instance.

Definition 13.11 (Graph and Label-Extended Graph)**.** *Given an instance of* Max2Lin(k) *with n variables and m equations, its graph $G = (V, E)$ is defined as follows.*

1. *Each variable x_i corresponds to a vertex i in V.*
2. *For each equation involving two variables x_i and x_j, there is an edge between i and j in E.*

The label-extended graph $\hat{G} = (\hat{V}, \hat{E})$ is defined as follows:

1. *Each variable x_i has a cloud of k vertices $(i, 0), \ldots, (i, k-1)$ in $\hat{V}$.*
2. *For each equation involving two variables x_i and x_j, there is a perfect matching between $(i, 0), \ldots, (i, k-1)$ and $(j, 0), \ldots, (j, k-1)$ in $\hat{E}$, where an edge exists between (i, p) and (j, q) if and only if assigning p to x_i and q to x_j satisfies the equation.*

Thus, G has n vertices and m edges, and the label-extended graph $\hat{G}$ has nk vertices and mk edges.

Each good assignment to a Unique Games instance corresponds to a small sparse cut in its label-extended graph.

Lemma 13.12 (Small Sparse Cut from Good Assignment)**.** *Given an instance of* Max2Lin(k) *with n variables and m equations, if there exists an assignment that satisfies at least a $(1-\epsilon)$-fraction of equations for $\epsilon \leq 1/2$, then there exists a set S in the label-extended graph $\hat{G} = (\hat{V}, \hat{E})$ such that*

$$\phi(S) \leq 2\epsilon \quad \text{and} \quad |S| = |\hat{V}|/k$$

and it intersects each cloud of $\hat{G}$ in exactly one vertex.

Proof. Let $x_1 = a_1, x_2 = a_2, \ldots, x_n = a_n$ be an assignment that satisfies at least a $(1-\epsilon)$-fraction of equations. Consider the set $S = \{(1, a_1), (2, a_2), \ldots, (n, a_n)\}$ in the label-extended graph $\hat{G} = (\hat{V}, \hat{E})$. Clearly, $|S| = |\hat{V}|/k$ and S intersects each cloud of $\hat{G}$ in exactly one vertex.

For each satisfied equation involving x_i and x_j, it contributes one edge (i, a_i)-(j, a_j) to $\hat{E}[S]$ and zero edges to $\delta(S)$. For each unsatisfied equation involving x_i and x_j, it contributes two edges (i, a_i)-(j, b_j) and (j, a_j)-(i, b_i) to $\delta(S)$ (where b_j is the unique value assigned to variable x_j to satisfy the equation if variable x_i is assigned value a_i, and b_i is defined similarly) and zero edges to $\hat{E}[S]$. Since at least $(1-\epsilon)m$ equations are satisfied and at most ϵm equations are unsatisfied,

$$\phi(S) = \frac{|\delta(S)|}{\mathrm{vol}(S)} \leq \frac{|\delta(S)|}{2|\hat{E}[S]|} \leq \frac{2\epsilon m}{2(1-\epsilon)m} \leq 2\epsilon$$

when $\epsilon \leq 1/2$. This completes the proof. □

The other direction is, in general, not true, as a small sparse cut in the label-extended graph may contain zero or more than one vertex in the cloud $\{(i,0),\dots,(i,k-1)\}$ of a variable x_i, and thus does not necessarily correspond to an assignment to the variables.

Still, the Unique Games Conjecture and the Small-Set Expansion Conjecture are intimately connected. Raghavendra and Steurer [RS10] proved that the Small-Set Expansion Conjecture implies the Unique Games Conjecture. On the other hand, we will see that the subspace enumeration method developed for the Small-Set Expansion conjecture can be extended to the Unique Games Conjecture.

Intuitively, the set size parameter δ in the Small-Set Expansion Conjecture corresponds to the inverse of the alphabet size parameter k from Lemma 13.12, and the problems become harder as δ decreases and k increases.

Subspace Enumeration for Unique Games Conjecture

Given a Unique Games instance Max2Lin(k), if the instance has an almost satisfiable assignment, then its label-extended graph $\hat{G}$ has a small sparse cut S that intersects each cloud in exactly one vertex, as shown in Lemma 13.12.

Suppose the label-extended graph has low threshold rank. Then, we can apply the subspace enumeration method from Theorem 13.6 to find a small sparse cut S' with a small symmetric difference to S (Lemma 13.3). Since S' still intersects most clouds in exactly one vertex, we can extract an assignment from S' that is almost satisfiable. (If S' intersects a cloud $\{(i,0),\dots,(i,k-1)\}$ in exactly one vertex (i,a), then we assign value a to x_i).

For a general graph $G=(V,E)$ from a Unique Games instance, we first apply the low threshold rank decomposition in Theorem 13.8 to find a partition $(V_1,\dots,V_l)$ such that each induced subgraph $G[V_i]$ has low threshold rank, with only a small fraction of crossing edges. The idea is simply to ignore or remove the equations corresponding to these crossing edges (since they form only a small fraction), and then apply the subspace enumeration method to the label-extended graph $\hat{G}[V_i]$ of each component V_i. This allows us to find an almost-satisfiable partial assignment in most of the components, which can then be combined to form an almost-satisfiable global assignment.

Theorem 13.13 (Subexponential Algorithm for Unique Games Conjecture [ABS15])**.** *Given a Unique Games instance Max2Lin(k) on n variables that has an assignment satisfying a $(1-\epsilon^7)$-fraction of equations, there exists an $\exp(kn^{O(\epsilon)})\cdot\mathrm{poly}(n)$-time algorithm to output an assignment satisfying a $\left(1-O(\epsilon\log\frac{1}{\epsilon})\right)$-fraction of equations.*

The proof follows by applying the low threshold rank decomposition in Theorem 13.8 with $\phi:=\epsilon$ to obtain a partition $(V_1,\dots,V_l)$ such that each $G[V_i]$ satisfies

$$\mathrm{rank}_{\epsilon^5}(G[V_i])\le n^{O(\epsilon)}$$

with at most a fraction of $O(\epsilon\log 1/\epsilon)$ of crossing edges we ignore or remove. Then, we argue below that the label-extended graph $\hat{G}[V_i]$ still has low threshold rank, such that

$$\mathrm{rank}_{\epsilon^6}(\hat{G}[V_i])\le k\cdot n^{O(\epsilon)}.$$

Hence, we apply the subspace enumeration method in Theorem 13.6 with $\phi:=\epsilon^7$, $\lambda:=\epsilon^6$, and threshold rank $r:=k\cdot n^{O(\epsilon)}$ in each $\hat{G}[V_i]$ to complete the proof.

Most details are straightforward and similar to those in Theorem 13.7. We highlight the connection between the threshold rank of $G[V_i]$ and that of its label-extended graph $\hat{G}[V_i]$, which follows from a trace argument.

Lemma 13.14 (Threshold Rank of Label-Extended Graph)**.** *Let G and $\hat{G}$ be the graph and the label-extended graph of a Unique Games instance* Max2Lin(k)*. Then*

$$\mathrm{rank}_{\epsilon^6}(\hat{G}) \lesssim k \cdot \mathrm{rank}_{\epsilon^5}(G) \cdot n^{\epsilon}.$$

Proof. From the definition of threshold rank in Definition 13.1, for any $t \geq 1$,

$$\mathrm{rank}_{2\lambda}(G) \cdot (1-\lambda)^{2t} \leq \sum_{i=1}^{n} (1-\lambda_i)^{2t} = \mathrm{Tr}(W^{2t}) \leq n \cdot (1-\lambda)^{2t} + \mathrm{rank}_{2\lambda}(G),$$

where W is the lazy random walk matrix of G, and λ_i is the i-th smallest eigenvalue of the normalized Laplacian matrix. We now show that

$$\mathrm{Tr}(\hat{W}^{2t}) \leq k \cdot \mathrm{Tr}(W^{2t}),$$

where $\hat{W}$ is the lazy random walk matrix of $\hat{G}$. This follows because each walk of length $2t$ that starts and returns to the vertex (i, a) in $\hat{G}$ corresponds to a length $2t$ walk that starts and returns to the vertex i in G, and at most k such walks in $\hat{G}$ corresponds to one such walk in G. Therefore,

$$\mathrm{rank}_{2\epsilon^6}(\hat{G}) \leq \frac{\mathrm{Tr}(\hat{W}^{2t})}{(1-\epsilon^6)^{2t}} \leq \frac{k \cdot \mathrm{Tr}(W^{2t})}{(1-\epsilon^6)^{2t}} \leq \frac{k \cdot \left(n \cdot (1-\epsilon^5)^{2t} + \mathrm{rank}_{2\epsilon^5}(G)\right)}{(1-\epsilon^6)^{2t}} \lesssim k \cdot \mathrm{rank}_{2\epsilon^5}(G) \cdot n^{\epsilon},$$

where the last inequality follows by setting $t = \frac{1}{2\epsilon^5} \ln n$. □

Theorem 13.13 inspired subsequent works [BRS11, GS11] that developed SDP-based approximation algorithms for various combinatorial optimization problems using spectral information of the graph. We will study some of these in Chapter **??**.

It remains an important open question whether these techniques can be pushed further to design subexponential algorithms with better approximation ratios for various combinatorial optimization problems, most notably the maximum cut problem and the sparsest cut problem.

13.5 Small-Set Expanders

In this section, we discuss some constructions of small-set expanders and their implications.

Noisy Hypercubes

The noisy hypercube H is a graph on $n := 2^l$ vertices, representing all bit strings of length l. For two vertices x, y, the edge weight $w(x, y)$ is the probability of transitioning from x to y if each bit of x is flipped independently with probability ϵ.

It can be shown that, with an appropriate choice of ϵ, the normalized Laplacian eigenvalue $\lambda_k \lesssim 1/\log k$, while all sets of size $\frac{2}{k}2^l$ have edge conductance at least $1/2$. This implies that the bound $\phi_{k/2} \lesssim \sqrt{\lambda_k \log k}$ in higher-order Cheeger inequality in Theorem 7.2 is tight [LOT14].

Small-Set Expanders from Locally Testable Codes

The key of the subexponential time algorithm for the Small-Set Expansion conjecture is the Cheeger-type inequality in Theorem 11.3, which states that for $k \geq n^{\Omega(1)}$, there exists a set with conductance $O(\sqrt{\lambda_k})$ and of size roughly n/k. If this bound could be improved to $n^{o(1)}$, then the Small-Set Expansion conjecture would be essentially disproved.

An interesting construction from locally testable codes [BGH+15] exhibits a small-set expander with $\exp(\log(n)^{\Omega(1/\log(1/\epsilon))})$ normalized Laplacian eigenvalues smaller than ϵ. This implies that the subspace enumeration method in Algorithm 7 require $\exp(\exp(\log(n)^{\Omega(1/\log(1/\epsilon))}))$ time for some graphs.

13.6 Problems

Problem 13.15 (Reducing the Set Size)**.** *Let $G = (V, E)$ be a d-regular graph. Given a set S with $\phi(S) \leq f(\phi)$ and $|S| = c \cdot \delta n$ for a universal constant $c > 1$ independent of δ and ϕ, show that a uniformly random subset $S' \subseteq S$ of size $|S'| = \delta n$ satisfies*

$$\mathbb{E}\left[\phi(S')\right] \leq c \cdot f(\phi) + 1 - \frac{1}{c}.$$

Conclude that there exists a subset $S' \subseteq S$ satisfying $\phi(S') < 1 - \phi$ for a sufficiently small constant ϕ (depending only on c).

References

[ABS15] Sanjeev Arora, Boaz Barak, and David Steurer. Subexponential algorithms for unique games and related problems. *Journal of the ACM (JACM)*, 62(5):1–25, 2015. 135, 136, 137, 139, 159, 160, 162, 166

[BGH+15] Boaz Barak, Parikshit Gopalan, Johan Håstad, Raghu Meka, Prasad Raghavendra, and David Steurer. Making the long code shorter. *SIAM Journal on Computing*, 44(5):1287–1324, 2015. 159, 168

[BRS11] Boaz Barak, Prasad Raghavendra, and David Steurer. Rounding semidefinite programming hierarchies via global correlation. In *2011 ieee 52nd annual symposium on foundations of computer science*, pages 472–481. IEEE, 2011. 9, 167

[GS11] Venkatesan Guruswami and Ali Kemal Sinop. Lasserre hierarchy, higher eigenvalues, and approximation schemes for graph partitioning and quadratic integer programming with psd objectives. In *2011 IEEE 52nd Annual Symposium on Foundations of Computer Science*, pages 482–491. IEEE, 2011. 9, 167

[Kho02] Subhash Khot. On the power of unique 2-prover 1-round games. In *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, pages 767–775, 2002. 164

[Kol11] Alexandra Kolla. Spectral algorithms for unique games. *computational complexity*, 20:177–206, 2011. 159, 160

[LOT14] James R. Lee, Shayan Oveis Gharan, and Luca Trevisan. Multiway spectral partitioning and higher-order cheeger inequalities. *J. ACM*, 61(6):37:1–37:30, 2014. 8, 89, 90, 167

[RS10] Prasad Raghavendra and David Steurer. Graph expansion and the unique games conjecture. In *Proceedings of the forty-second ACM symposium on Theory of computing*, pages 755–764, 2010. 135, 166

Part III

Spectral Sparsification and Applications

The study of spectral sparsification has led to major breakthroughs in algorithm design and mathematics. It is a striking example of how linear algebraic techniques can be used to solve combinatorial problems more effectively.

Chapter 14

Spectral Sparsification, Matrix Concentration, and Effective Resistance

In this chapter, we introduce the spectral sparsification problem, a generalization of the cut sparsification problem. We will see that this more general problem leads to a natural random sampling algorithm and an elegant analysis using matrix concentration inequalities. We will also explore the concept of effective resistance in electrical networks and its connection to the sampling probability.

14.1 Cut Sparsification

In this section, we briefly review results on the cut sparsification problem to provide context and perspective for the spectral sparsification problem.

The cut sparsification problem, formulated by Karger [Kar99], seeks to find a sparse graph that approximates all cut values of a given graph.

Definition 14.1 (Cut Approximator)**.** *Let $G = (V, E)$ be an undirected graph with edge weights $w_G(e)$ for each $e \in E$, and let $H = (V, F)$ be another undirected graph on the same vertex set with edge weights $w_H(e)$ for each $e \in F$. For $0 \leq \epsilon < 1$, we say H is a $(1 \pm \epsilon)$-cut approximator of G if*

$$(1 - \epsilon) \cdot w_G\big(\delta_G(S)\big) \leq w_H\big(\delta_H(S)\big) \leq (1 + \epsilon) \cdot w_G\big(\delta_G(S)\big) \quad \text{for all } S \subseteq V,$$

where $w(\delta(S)) := \sum_{e \in \delta(S)} w(e)$ denotes the total weight of edges in $\delta(S)$.

Note that this definition does not require that H is a subgraph of G (that is, $F \subseteq E$). However, all constructions that we will see satisfy this property, which is useful in certain applications.

Uniform Sampling

A natural example to consider is when G is a complete graph. From Chapter 4, we know that a random sparse graph H is typically an expander graph, which intuitively is a good approximation of the complete graph. This suggests a natural strategy for constructing a sparsifier H by sampling a uniform random subgraph of G.

The following simple uniform random sampling algorithm was proposed in [Kar99], based on the idea that the expected weight of each edge e in H matches its weight in G.

Algorithm 9 Uniform Sampling Algorithm for Cut Sparsification

Require: An unweighted undirected graph $G = (V, E)$.
1: Set a sampling probability p.
2: **for** each $e \in E$ **do**
3: Add e in F with weight $\frac{1}{p}$ with probability p.
4: **end for**
5: **return** $H = (V, F)$.

Karger proved that the uniform sampling algorithm effectively sparsifies G when the minimum cut value of G is $\Omega(\log n)$. The well-known Chernoff bound is used to analyze the success probability of the algorithm.

Theorem 14.2 (Chernoff Bound for Heterogeneous Coin Flips)**.** *Let $X_1, X_2, \ldots, X_n$ be independent random variables, where each $X_i = 1$ with probability p_i and $X_i = 0$ otherwise. Define $X = \sum_{i=1}^{n} X_i$ and let $\mu = \mathbb{E}[X] = \sum_{i=1}^{n} \mathbb{E}[X_i] = \sum_{i=1}^{n} p_i$ be the expected value of X. Then, for any $0 < \delta < 1$,*

$$\Pr\left(|X - \mu| \geq \delta\mu\right) \leq 2e^{-\delta^2\mu/3}.$$

The analysis goes roughly as follows. Given the assumption that the minimum cut value is $\Omega(\log n)$, the Chernoff bound can be applied to show that the probability of $w_H(\delta_H(S))$ deviating from a $(1 \pm \epsilon)$-approximation of $w_G(\delta_G(S))$ for a particular subset $S \subseteq V$ is at most $1/\operatorname{poly}(n)$. While this probability is small, it is not sufficiently small to apply a union bound over the exponential number of possible subsets. The key observation is that the number of approximately minimum cuts is only polynomial, allowing for a more refined union bound based on the cut value to establish the result.

Non-Uniform Sampling

It is easy to see that the uniform sampling algorithm may fail without the minimum cut assumption. For example, consider a dumbbell graph, where two complete graphs are connected by a single edge.

Benczur and Karger [BK15] designed a clever non-uniform sampling algorithm, where the sampling probability p_e for each edge $e = ij$ is proportional to the "connectivity" between i and j. The idea is that edges with low connectivity are included in H with higher probability p_e and assigned a smaller weight $1/p_e$, as they are crucial and should essentially be preserved. In contrast, edges with high connectivity are sampled with lower probability and assigned a larger weight to effectively sparsify the graph.

To formalize this approach, they introduced a notion called "strong connectivity" for the non-uniform sampling algorithm and proved that every graph has a cut approximator with only $O(n \log n)$ edges.

Theorem 14.3 (Near-Linear Sized Cut Sparsification [BK15])**.** *For any edge-weighted undirected graph $G = (V, E)$ and any $0 < \epsilon < 1$, there exists a reweighted subgraph $H = (V, F)$ on the same vertex set with at most $O((|V| \log |V|)/\epsilon^2)$ edges such that H is a $(1 \pm \epsilon)$-cut approximator of G. Furthermore, H can be computed in nearly linear time $\tilde{O}(|E|)$.*

This result was not only a surprising mathematical theorem, but also introduced a novel approach to designing fast graph algorithms, which has since become highly influential. We will discuss some algorithmic applications in Section 14.4.

Benczur and Karger conjectured that strong connectivity could be replaced by the more traditional edge-connectivity between u and v, a conjecture that was later confirmed [FHHP19].

14.2 Spectral Sparsification

In their work on designing a near-linear time algorithm for solving Laplacian systems of linear equations, Spielman and Teng [ST11] introduced the following stronger notion of spectral sparsification for Laplacian matrices.

Definition 14.4 (Spectral Approximator). *Let $G = (V, E)$ be an edge-weighted undirected graph, and let $H = (V, F)$ be another edge-weighted undirected graph on the same vertex set. For $0 \leq \epsilon \leq 1$, we say that H is a $(1 \pm \epsilon)$-spectral approximator of G if*

$$(1-\epsilon) \cdot x^\top L_G x \leq x^\top L_H x \leq (1+\epsilon) \cdot x^\top L_G x \quad \text{for all } x \in \mathbb{R}^{|V|},$$

where L_G and L_H are the weighted Laplacian matrices of G and H respectively. Equivalently, H is a $(1 \pm \epsilon)$-spectral approximator of G if

$$(1-\epsilon) L_G \preccurlyeq L_H \preccurlyeq (1+\epsilon) L_G.$$

It is easy to check that if H is a $(1 \pm \epsilon)$-spectral approximator of G, then all the eigenvalues of H and G are within a factor of $1 \pm \epsilon$ of each other; see Exercise 14.20.

Spectral Sparsification and Cut Sparsification

The spectral sparsification problem seeks to find a sparse graph H that is a good spectral approximator of the input graph G. The original motivation in [ST14] is to use L_H as a "preconditioner" for solving the linear system $L_G \cdot z = b$. Their definition is inspired by the result on cut sparsification in Theorem 14.3 and is indeed a more demanding one.

Lemma 14.5 (Spectral Approximator is Cut Approximator). *If H is a $(1 \pm \epsilon)$-spectral approximator of G, then H is also a $(1 \pm \epsilon)$-cut approximator of G.*

Proof. Let S be a subset of vertices, and let χ_S be its characteristic vector. By Lemma 1.14,

$$\chi_S^\top L_G \chi_S = \sum_{ij \in E(G)} w(i,j)\big(\chi_S(i) - \chi_S(j)\big)^2 = w_G\big(\delta_G(S)\big) \quad \text{and} \quad \chi_S^\top L_H \chi_S = w_H\big(\delta_H(S)\big).$$

Since H is a $(1 \pm \epsilon)$-spectral approximator of G,

$$(1-\epsilon) \cdot \chi_S^\top L_G \chi_S \leq \chi_S^\top L_H \chi_S \leq (1+\epsilon) \cdot \chi_S^\top L_G \chi_S,$$

which implies that

$$(1-\epsilon) \cdot w_G\big(\delta_G(S)\big) \leq w_H\big(\delta_H(S)\big) \leq (1+\epsilon) \cdot w_G\big(\delta_G(S)\big).$$

Since this holds for any subset $S \subseteq V$, we conclude that H is a $(1 \pm \epsilon)$-cut approximator of G. □

Since spectral sparsification is a strictly stronger requirement than cut sparsification, one might expect it to be a strictly harder problem to solve. Initially, Spielman and Teng [ST11] proved that a $(1 \pm \epsilon)$-spectral sparsifier with $O(n \log^c(n)/\epsilon^2)$ edges always exists for some large constant c. They also provided a fast algorithm for constructing such sparsifiers, using ideas from local graph partitioning and expander decomposition as discussed in Section 12.5. This result was sufficient for their goal of designing a nearly-linear time algorithm for solving Laplacian equations, which has since become the foundation for a new generation of fast algorithms for various graph problems.

Reduction to Isotropy Condition

Spielman and Srivastava [SS11] revisited the spectral sparsification problem and proved that there is always a $(1\pm\epsilon)$-spectral approximator with $O((n \log n)/\epsilon^2)$ edges, thereby generalizing Theorem 14.3 for cut sparsification.

They reduced the spectral sparsification problem to the following simpler form where the vectors are in isotropy condition (see Lemma 7.6). In this form, the graph structure is removed, and the objective is reduced to bounding only the maximum and minimum eigenvalues.

Theorem 14.6 (Sparse Spectral Approximator of Identity Matrix [SS11]). *For any m vectors $v_1, \ldots, v_m \in \mathbb{R}^n$ satisfying*

$$\sum_{i=1}^{m} v_i v_i^\top = I_n,$$

there always exist scalars $s_1, \ldots, s_m$ with at most $O((n \log n)/\epsilon^2)$ nonzeros such that

$$(1-\epsilon) I_n \preccurlyeq \sum_{i=1}^{m} s_i v_i v_i^\top \preccurlyeq (1+\epsilon) I_n.$$

For the reduction, the concept of the pseudoinverse of a matrix (Definition A.7) is required.

Definition 14.7 (Pseudoinverse of the Laplacian Matrix). *Let G be a connected graph. Let $0 = \lambda_1 < \lambda_2 \leq \ldots \leq \lambda_n$ be the eigenvalues of $L(G)$, and let $u_1, \ldots, u_n$ be the corresponding orthonormal eigenvectors. Then the pseudoinverse of $L(G)$ and its square root are given by*

$$L_G^\dagger = \sum_{i=2}^{n} \frac{1}{\lambda_i} u_i u_i^\top \quad \text{and} \quad L_G^{\dagger/2} = \sum_{i=2}^{n} \frac{1}{\sqrt{\lambda_i}} u_i u_i^\top.$$

The reduction follows from a simple linear transformation by the pseudoinverse.

Theorem 14.8 (Near-Linear Sized Spectral Sparisification [SS11]). *For any edge-weighted undirected graph $G = (V, E)$ and any $0 < \epsilon < 1$, there exists a reweighted subgraph $H = (V, F)$ on the same vertex set with at most $O((|V| \log |V|)/\epsilon^2)$ edges such that H is a $(1\pm\epsilon)$-spectral approximator of G. Furthermore, H can be computed in nearly linear time $\tilde{O}(|E|)$.*

Proof. We assume without loss of generality that G is a connected graph. Let $L_G = \sum_{e \in E} b_e b_e^\top$ be the Laplacian matrix of G, as written in Definition 1.12. Define

$$v_e := U^\top L_G^{\dagger/2} b_e, \tag{14.1}$$

where $L_G^{\dagger/2}$ is from Definition 14.7 and U is the $n \times (n-1)$ matrix where the i-th column is the $(i+1)$-th eigenvector u_{i+1} of $L(G)$ for $1 \le i \le n-1$. Then, we compute

$$\sum_{e \in E} v_e v_e^\top = \sum_{e \in E} U^\top L_G^{\dagger/2} b_e b_e^\top L_G^{\dagger/2} U = U^\top L_G^{\dagger/2} L_G L_G^{\dagger/2} U = U^\top \left(\sum_{i=2}^{n} u_i u_i^\top \right) U = I_{n-1}.$$

By Theorem 14.6, there exist scalars $s_1, \ldots, s_m$ with at most $O((n \log n)/\epsilon^2)$ nonzeros such that

$$(1-\epsilon) I_{n-1} \preccurlyeq \sum_{e \in E} s_e v_e v_e^\top \preccurlyeq (1+\epsilon) I_{n-1}.$$

Multiplying $L_G^{1/2} U$ on the left and $U^\top L_G^{1/2}$ on the right of these inequalities, we obtain

$$(1-\epsilon) L_G \preccurlyeq \sum_{e \in E} s_e b_e b_e^\top \preccurlyeq (1+\epsilon) L_G.$$

Let H be the graph with weight s_e on edge e. Since $L_H = \sum_{e \in E} s_e b_e b_e^\top$, it follows that H is a $(1 \pm \epsilon)$-spectral approximator of G with $O((n \log n)/\epsilon^2)$ edges. □

Henceforth, we focus on proving Theorem 14.6.

14.3 Random Sampling Algorithm

The formulation in Theorem 14.6 leads to a natural random sampling algorithm and an elegant analysis using matrix concentration inequalities.

Intuition and Idea

We discussed the isotropy condition $\sum_{i=1}^m v_i v_i^\top = I_n$ when studying the higher-order Cheeger inequality. As stated in Lemma 7.6, for any unit vector $y \in \mathbb{R}^n$, it holds that $\sum_{i=1}^m \langle y, v_i \rangle^2 = 1$. Informally, the vectors are "evenly spread out", meaning their projections onto any direction y are the same. Given $\sum_{i=1} v_i v_i^\top = I_n$, our goal is to find a small subset of vectors $S \subseteq \{1, \ldots, m\}$ and appropriate scaling factors such that $\sum_{i \in S} s_i v_i v_i^\top \approx I_n$. Thus, the subset should remain "evenly spread out", preserving a balanced contribution across all directions.

As in the cut sparsification case, uniform sampling may fail. For instance, if a vector v_j has $\|v_j\| = 1$, it must be included in the solution, as otherwise the corresponding direction will not be covered in the solution, preventing it from being a spectral sparsifier. The analogy in cut sparsification is that a cut edge must be included in any cut sparsifier. Thus, as in the cut sparsification case, non-uniform sampling is necessary.

The non-uniform sampling idea is similar, but now the choice of sampling probability becomes clear. For longer vectors, we should set the sampling probability p_e higher because they are crucial and should essentially be preserved. For shorter vectors, we can afford to set the sampling probability p_e lower and the weight $1/p_e$ higher in order to reduce the number of vectors. Concretely, we sample each vector v_i with probability $\|v_i\|_2^2$, and if it is chosen, we set the scalar $s_i = \frac{1}{\|v_i\|_2^2}$, so that the expected contribution is

$$\mathbb{E}\left[s_i v_i v_i^T\right] = \frac{v_i v_i^T}{\|v_i\|^2} \cdot \Pr[v_i \text{ is chosen}] = \frac{v_i v_i^T}{\|v_i\|^2} \cdot \|v_i\|^2 = v_i v_i^T.$$

Algorithm

The actual algorithm is the same as discussed above, except that the sampling probability is multiplied by a scaling factor τ to ensure concentration.

Algorithm 10 Random Sampling Algorithm for Spectral Sparsification

Require: Vectors $v_1, \ldots, v_m \in \mathbb{R}^n$ satisfying $\sum_{i=1}^m v_i v_i^\top = I_n$.
1: Initialization: $\vec{s} \leftarrow \vec{0}$ and $\tau = (6 \ln n)/\epsilon^2$.
2: **for** $1 \leq i \leq m$ **do**
3: Set $s_i \leftarrow \frac{1}{p_i}$ with probability $p_i = \tau \cdot \|v_i\|_2^2$.
4: **end for**
5: **return** $\sum_{i=1}^m s_i v_i v_i^\top$.

The analysis consists of two steps. One is to show that there are $O(n \log n/\epsilon^2)$ non-zero scalars. The other is to prove that the solution is a $(1 \pm \epsilon)$-spectral approximator of the identity matrix.

Expected Number of Vectors

We bound the number of nonzero scalars by computing its expected value and applying Markov's inequality. The choice of sampling probability proportional to the squared length is used to bound the expected value.

Lemma 14.9 (Number of Nonzeros)**.** *Let $\vec{s}$ be the output of Algorithm 10, and let $S = \mathrm{supp}(\vec{s})$ be the set of vectors with nonzero scalars. Then $|S| \lesssim (n \log n)/\epsilon^2$ with probability at least 0.9.*

Proof. The expected number of nonzeros is

$$\mathbb{E}\left[|S|\right] = \sum_{i=1}^m \Pr[i \in S] = \tau \sum_{i=1}^m p_i,$$

where $\tau = (6 \ln n)/\epsilon^2$ is the scaling factor defined in Algorithm 10. Observe that

$$\sum_{i=1}^m p_i = \sum_{i=1}^m \|v_i\|_2^2 = \sum_{i=1}^m v_i^\top v_i = \sum_{i=1}^m \mathrm{Tr}(v_i^\top v_i) = \sum_{i=1}^m \mathrm{Tr}(v_i v_i^\top) = \mathrm{Tr}\left(\sum_{i=1}^m v_i v_i^\top\right) = \mathrm{Tr}(I_n) = n,$$

where we used $\mathrm{Tr}(AB) = \mathrm{Tr}(BA)$ in Fact A.34. Therefore,

$$\mathbb{E}\left[|S|\right] = \tau \sum_{i=1}^m p_i = \tau n = \frac{6n \ln n}{\epsilon^2},$$

and the result follows from Markov's inequality that $\Pr\left[|S| \geq 10 \cdot \mathbb{E}\left[|S|\right]\right] \leq 1/10$. □

Matrix Chernoff Bound

There is an elegant generalization of the Chernoff-Hoeffding bound to the matrix setting.

Theorem 14.10 (Matrix Chernoff Bound [Tro12]). *Let $X_1, \ldots, X_m$ be independent, $n \times n$ real symmetric matrices with $0 \preccurlyeq X_i \preccurlyeq rI$ for $1 \le i \le m$. Suppose*

$$\mu_{\min} \cdot I_n \preccurlyeq \sum_{i=1}^{m} \mathbb{E}[X_i] \preccurlyeq \mu_{\max} \cdot I_n.$$

Then, for any $0 \le \epsilon \le 1$,

$$\Pr\left[\lambda_{\max}\left(\sum_{i=1}^{m} X_i\right) \ge (1+\epsilon)\mu_{\max}\right] \le n e^{-\frac{\epsilon^2 \mu_{\max}}{3r}} \text{ and } \Pr\left[\lambda_{\min}\left(\sum_{i=1}^{m} X_i\right) \le (1-\epsilon)\mu_{\min}\right] \le n e^{-\frac{\epsilon^2 \mu_{\min}}{2r}}.$$

This bound is almost an exact analog of the Chernoff bound in Theorem 14.2, using the maximum and minimum eigenvalues to measure the "size" of a matrix. Informally, it states that if the sum of independent random matrices consists of matrices that are not too "large" or "influential", then its eigenvalues remain concentrated around their expectation. The proof will be discussed in Chapter 23.

Spectral Approximation

The algorithm is designed so that the proof of correctness follows directly from the matrix Chernoff bound. The reweighting by the sampling probability and the choice of iterations τ ensure that no random variable is too influential.

Lemma 14.11 (Success Probability of Spectral Approximation). *The output of Algorithm 10 satisfies $(1-\epsilon)I_n \preccurlyeq \sum_{i=1}^{m} s_i v_i v_i^\top \preccurlyeq (1+\epsilon)I_n$ with probability at least $1 - 2/n$.*

Proof. We apply Theorem 14.10 to prove the lemma. The random variables are

$$X_i = \begin{cases} v_i v_i^\top / p_i & \text{with probability } p_i \\ 0 & \text{otherwise,} \end{cases}$$

for each vector i. The output of the algorithm is $Y := \sum_{i=1}^{m} X_i$. The expected output is

$$\mathbb{E}[Y] = \sum_{i=1}^{m} \mathbb{E}[X_i] = \sum_{i=1}^{m} p_i \cdot \frac{v_i v_i^\top}{p_i} = \sum_{i=1}^{m} v_i v_i^\top = I_n.$$

Thus, the expected output is exactly the identity matrix, with $\mu_{\max} = \mu_{\min} = 1$. To apply the matrix Chernoff bound in Theorem 14.10, we need to find a bound r such that $X_i \preccurlyeq rI$. Observing that

$$X_{i,t} = \frac{v_i v_i^\top}{p_i} = \frac{v_i v_i^\top}{\tau \|v_i\|^2} = \frac{1}{\tau}\left(\frac{v_i}{\|v_i\|}\right)\left(\frac{v_i}{\|v_i\|}\right)^\top,$$

which is a rank-one matrix of a unit vector, we conclude that the maximum eigenvalue is $1/\tau$, so we set $r = 1/\tau$. By Theorem 14.10, using $\tau = (6 \ln n)/\epsilon^2$, we obtain that

$$\Pr[\lambda_{\max}(Y) \ge 1+\epsilon] \le n \cdot e^{-\epsilon^2 \tau/3} = n \cdot e^{-2\ln n} = \frac{1}{n}.$$

The lower tail follows similarly. Thus, with probability at least $1 - \frac{2}{n}$, $\lambda_{\max}(Y) \le 1+\epsilon$ and $\lambda_{\min}(Y) \ge 1-\epsilon$. This implies $(1-\epsilon)I_n \preccurlyeq Y \preccurlyeq (1+\epsilon)I$, proving that the solution is a $(1 \pm \epsilon)$-spectral approximator of the identity matrix. □

By combining Lemma 14.9 and Lemma 14.11, we conclude that a $(1 \pm \epsilon)$-spectral approximator of the identity matrix with $O(n \log n/\epsilon^2)$ vectors exists. Moreover, the random sampling algorithm succeeds with constant probability. This proves Theorem 14.6 and thus Theorem 14.8, showing that every graph has a $(1 \pm \epsilon)$-spectral approximator with $O(n \log n/\epsilon^2)$ edges.

The algorithmic aspect will be explained in the next section.

Discussions

The analysis of the random sampling algorithm is tight. Sparsifying the complete graph on n vertices using independent random sampling requires $\Omega(n \log n)$ edges. See Problem 14.22.

By considering spectral sparsification, we obtain an elegant and arguably simpler proof of Theorem 14.3 for cut sparsification. In the cut sparsification problem, the appropriate sampling probability was not immediately clear. In the more general spectral sparsification problem, after removing the graph structure, there appears to be only one natural choice for the sampling probability, and the analysis follows directly from the matrix Chernoff bound. The sampling probability turns out to be the effective resistance of an edge, which will be discussed in Section 14.5.

This is a great example where a more general problem can be easier to solve than its special case. In the special case, multiple reasonable approaches seem possible, whereas in the generalization, the correct approach naturally emerges.

14.4 Fast Algorithm and Applications

In this section, we first describe the near-linear time algorithm for spectral sparsification, and then discuss some algorithmic applications of cut sparsification and spectral sparsification.

Near-Linear Time Algorithm for Spectral Sparsification

To implement Algorithm 10, we need a fast algorithm to compute the sampling probabilities.

Fast Laplacian Solver: For this, we assume access to a near-linear time algorithm for solving Laplacian equations [ST14]. That is, given the weighted Laplacian matrix L_G of an edge weighted graph $G = (V, E)$ and a vector $b \in \mathbb{R}^{|V|}$, we assume there exists a $\tilde{O}(|E|)$-time algorithm to compute an approximate solution $x \in \mathbb{R}^{|V|}$ such that

$$L_G \cdot x \approx b, \quad \text{or equivalently,} \quad x \approx L_G^{\dagger} \cdot b.$$

We will study a near-linear time Laplacian solver in **??**. For now, we ignore numerical accuracy in the following discussion.

Sampling Probabilities: By (14.1), the sampling probability is

$$\|v_e\|_2^2 = \|U^\top L_G^{\dagger/2} b_e\|_2^2 = b_e^\top L_G^{\dagger/2} U U^\top L_G^{\dagger/2} b_e = b_e^\top L_G^{\dagger} b_e,$$

where we used the facts that $UU^\top = I - u_1 u_1^\top$ and $L_G^{\dagger/2} u_1 = 0$, which follow from the definitions of U in Theorem 14.8 and $L_G^{\dagger}$ in Definition 14.7.

To leverage the fast Laplacian solver, we rewrite the sampling probabilities as

$$\|v_e\|_2^2 = b_e^\top L_G^\dagger b_e = b_e^\top L_G^\dagger L_G L_G^\dagger b_e = b_e^\top L_G^\dagger B B^\top L_G^\dagger b_e = \|B^\top L_G^\dagger b_e\|_2^2 = \|B^\top L_G^\dagger(\chi_i - \chi_j)\|_2^2,$$

where $B \in \mathbb{R}^{n\times m}$ is the edge incidence matrix from Definition 1.12, with $n = |V|$ and $m = |E|$. Thus, we only need to compute the lengths of at most $m \le n^2$ vectors in an n-dimensional space.

Dimension Reduction: It is sufficient to compute the sampling probabilities approximately [SS11]. The idea then is to use the Johnson-Lindenstrauss theorem from Theorem 9.14 to reduce the dimension of the vectors in order to speedup the computations. Let $Q \in \mathbb{R}^{k\times m}$ be a random Gaussian matrix, where each entry is defined as $Q(i,j) := g_{i,j}/\sqrt{k}$, with $k \asymp \log(n)/\epsilon^2$ and $g_{i,j}$ being independent random variables drawn from the normal distribution $N(0,1)$. By Theorem 9.14, with high probability,

$$(1-\epsilon)\|B^\top L_G^\dagger(\chi_i - \chi_j)\|_2^2 \le \|QB^\top L_G^\dagger(\chi_i - \chi_j)\|_2^2 \le (1+\epsilon)\|B^\top L_G^\dagger(\chi_i - \chi_j)\|_2^2 \quad \text{for all } i,j \in V.$$

Computations: To obtain a good approximation of the sampling probability $\|v_e\|_2^2$, it suffices to compute $\|QB^\top L_G^\dagger(\chi_i - \chi_j)\|_2^2$ efficiently. To achieve this, we first compute the matrix $Z = QB^\top L_G^\dagger$ efficiently and store this $O(\log n) \times m$ matrix. Then, whenever we need to compute $\|QB^\top L^\dagger(\chi_i - \chi_j)\|_2^2 = \|Z(\chi_i - \chi_j)\|_2^2$, we simply take the difference of two columns of Z, which can be done in $O(\log n)$ time. Thus, the total time to compute the approximate sampling probabilities after computing Z is $\tilde{O}(m)$.

It remains to show how to compute Z in $\tilde{O}(m)$ time using a fast Laplacian solver. First, we compute $QB^\top$, which can be done in $O(km) = \tilde{O}(m)$ time since $B^\top$ has only two nonzeros per row. Then, the i-th row of Z is equal to the i-th row of $QB^\top$ multiplied by $L_G^\dagger$. Thus, it is of the form $L_G^\dagger \cdot y$ for some y, which can be computed by solving $L_G \cdot x = y$ using a fast Laplacian solver in $\tilde{O}(m)$ time. Therefore, the total time to compute Z is $\tilde{O}(m)$, and this completes the description of the algorithm.

Applications of Cut Sparsification

Cut sparsification has become an essential tool in designing fast graph algorithms.

As an example, consider the problem of solving the minimum s-t cut problem in a graph G. Standard algorithms have a runtime that depends on the number of edges in G, which makes them inefficient when G is dense with $\Omega(n^2)$ edges. To design a fast approximation algorithm, we can first use Theorem 14.3 to compute a $(1\pm\epsilon)$-cut approximator H of G with only $O((n\log n)/\epsilon^2)$ edges in near-linear time. We then run a standard minimum s-t cut algorithm on H, yielding a solution that can be shown to be a $(1+3\epsilon)$-approximate minimum s-t cut in G.

More generally, these sparsification algorithms have led to significant speedups in solving many graph problems related to cuts (e.g., graph conductance). They allow us to trade a small loss in solution optimality for a substantial reduction in time complexity – usually improving runtime by at least one order of n. For fast approximation algorithms, this small loss in solution optimality is negligible.

Remarkably, cut sparsification can also be applied to design fast *exact* algorithm. Karger [Kar00] used the uniform sampling algorithm to develop a near-linear time algorithm that solves the minimum cut problem optimally. For this application, it is crucial that the sparsifier remains unweighted, so for example the stronger Theorem 14.3 cannot be used. It took more than twenty

years for researchers to finally discover a deterministic near-linear time algorithm for the minimum cut problem [KT19].

Applications of Spectral Sparsification

A major application of spectral sparsification is in the design of near-linear time Laplacian solvers [ST14], which have numerous applications in graph algorithms and numerical linear algebra. We will study Laplacian solvers in **??**.

We illustrate the techniques of spectral sparsification using a fundamental problem in numerical linear algebra, the least squares problem. In the least squares problem, we are given an $n \times d$ matrix A and $b \in \mathbb{R}^n$, and the objective is to find an $x \in \mathbb{R}^d$ that minimizes $\|Ax - b\|_2$. We are typically interested in the case when $n \gg d$, when the problem is over-constrained. Exact algorithms require $\Omega(n \operatorname{poly}(d))$ time, which is too slow for large n.

The goal is to design an approximation algorithm that returns a solution x' satisfying $\|Ax' - b\| \leq (1+\epsilon) \min_x \|Ax - b\|_2$ in $\tilde{O}(nd + \operatorname{poly}(\frac{d}{\epsilon}))$ time, which is near linear when $n \gg d$. The key idea is to compress the matrix A into a smaller $k \times d$ matrix $B = SA$, where $k = \operatorname{poly}(\frac{d}{\epsilon})$, and then solve the least square problem $\min_x \|S(Ax - b)\|_2$ exactly as an approximate solution to the original problem.

Given $A \in \mathbb{R}^{n \times d}$ and $b \in \mathbb{R}^n$, we first reduce the problem to the case where the columns of A are orthonormal. This is similar to the reduction to the identity matrix in Theorem 14.8, so that $A^\top A = I_d$, or equivalently, $\sum_{i=1}^n a_i a_i^\top = I_d$ where a_i is the i-th row of A. We then construct a matrix B by sampling and rescaling each row proportional to its squared length, so that $B = \sum_{i=1}^n s_i a_i a_i^\top \approx I_d$ with only $O(d \log d/\epsilon^2)$ nonzero scalars. Thus, B has $O(d \log d/\epsilon^2)$ rows, where each row is $\sqrt{s_i} a_i$ such that $(1-\epsilon) A^\top A \preccurlyeq B^\top B \preccurlyeq (1+\epsilon) A^\top A$.

This property, known as "subspace embedding", ensure that $\|Ax\|_2^2 \approx \|Bx\|_2^2$ since $x^\top A^\top A x \approx x^\top B^\top B x$. The technique details are closely related to those in spectral sparsification, such as the use of the matrix Chernoff bound. The sampling probability in this context is known as the leverage score of a row, which is a generalization of effective resistance. See [Woo14] for more details and applications in numerical linear algebra.

14.5 Effective Resistance

The sampling probability of an edge $e = ij$ is proportional to

$$\|v_e\|_2^2 = b_e^\top L_G^\dagger b_e = (\chi_i - \chi_j)^\top L_G^\dagger (\chi_i - \chi_j). \tag{14.2}$$

In this section, we briefly review some background on electrical networks and effective resistance, explaining why the sampling probability is proportional to the effective resistence of the edge ij and providing some intuition about this quantity. Since effective resistance is a useful concept in spectral graph theory, we also discuss some additional properties at the end of this section.

Electrical Networks and Electrical Flows

Given an undirected graph $G = (V, E)$ with edge weights $w : E \to \mathbb{R}_+$, we model G as an electrical network, where each edge $e \in E$ represents a resistor with resistance $r(e) = 1/w(e)$.

Definition 14.12 (Electrical Flows)**.** *Let $G = (V, E)$ be a graph with edge weights $w : E \to \mathbb{R}_+$. Let $s, t \in V$ be two distinct vertices (a source and a sink), and let $b : V \to \mathbb{R}$ be a demand function such that*

$$b(s) = 1, \quad b(t) = -1, \quad \text{and} \quad b(i) = 0 \text{ for all } i \neq s, t.$$

An electrical flow $f : E \to \mathbb{R}$ satisfies:

1. *Kirchhoff's Law:*
$$\sum_{j : ij \in E} f(i, j) = b(i) \quad \text{for all } i \in V.$$
This ensures that the total outgoing flow at every vertex satisfies the demand function.

2. *Ohm's Law: For each edge $ij \in E$, the flow satisfies*
$$f(i, j) = w(ij) \cdot (x(i) - x(j)) = \frac{x(i) - x(j)}{r(ij)},$$
where $x : V \to \mathbb{R}$ is a potential function assigning a voltage to each vertex.

The flow is directed such that $f(i, j)$ is positive if the flow moves from i to j and is negative if the flow moves from j to i.

Combining Kirchhoff's Law and Ohm's Law, these equations can be written compactly as

$$L_G \cdot x = b, \quad \text{or equivalently,} \quad x = L_G^\dagger \cdot b \tag{14.3}$$

where L_G is the weighted Laplacian matrix, with $L_G(i, j) = w(i, j)$ for $i \neq j$, and $L_G^\dagger$ is its pseudoinverse. When the graph G is connected (so that the kernel of L_G is $\text{span}\{\vec{1}\}$), the set of solutions to the voltage function is given by

$$\{L_G^\dagger \cdot b + c\vec{1} \mid c \in \mathbb{R}\}.$$

Since shifting all voltages by a constant does not affect the potential differences between vertices, the electrical flow f is uniquely defined.

Conversely, any Laplacian system of linear equations corresponds to computing the voltage vector of an electrical flow problem (with a general demand vector).

Effective Resistances and Energy

We now define effective resistance and show that it is equal to the sampling probability.

Definition 14.13 (Effective Resistance)**.** *Let $G = (V, E)$ be a graph with edge weights $w : E \to \mathbb{R}_+$. The effective resistance $\text{Reff}(s, t)$ between vertices s and t is defined as the voltage difference $x(s) - x(t)$ when one unit of electrical flow is sent from s to t, as in Definition 14.12.*

From this definition and (14.3), we see that

$$\text{Reff}(s, t) = x(s) - x(t) = (\chi_s - \chi_t)^\top x = (\chi_s - \chi_t)^\top L_G^\dagger \cdot b = (\chi_s - \chi_t)^\top L_G^\dagger \cdot (\chi_s - \chi_t),$$

which is equal to the sampling probability of an edge st stated in (14.2).

To gain intuition about effective resistance, it is helpful to consider an equivalent characterization using energy.

Definition 14.14 (Energy of Flow). *Let $G = (V, E)$ be a graph with edge weights $w : E \to \mathbb{R}_+$. The energy of a flow function $f : E \to \mathbb{R}$ is defined as*

$$\mathcal{E}(f) := \sum_{e \in E} \frac{f(e)^2}{w(e)} = \sum_{e \in E} r(e) \cdot f(e)^2,$$

where $r(e)$ is the resistance, defined as $1/w(e)$.

The electrical flow minimizes the energy among all flows of one unit from s to t, and the effective resistance between s and t is equal to this minimum energy.

Theorem 14.15 (Electrical Flow Minimizes Energy). *The electrical flow f from s to t, defined in Definition 14.12, satisfies*

$$\mathcal{E}(f) \leq \mathcal{E}(f') \quad \text{for all flows } f' \text{ that satisfy Kirchoff's Law.}$$

Moreover, the energy of the electrical flow f from s to t is equal to the effective resistance between s and t, such that

$$\mathrm{Reff}(s, t) = \mathcal{E}(f).$$

The proof of the first statement follows from the optimality conditions of the convex minimization problem, and the second statement follows from an algebraic manipulation. See Problem 14.24.

Intuition about Sampling Probability: Using Theorem 14.15, we can interpret effective resistance between s and t as an interpolation between the shortest path distance and the edge-connectivity between s and t. Consider an unweighted graph $G = (V, E)$ where each edge has weight one. Let

$$\mathcal{E}_p(s, t) := \min_f \left\{ \left(\sum_e |f(e)|^p \right)^{1/p} \;\middle|\; f \text{ is a unit } s\text{-}t \text{ flow} \right\}$$

be the minimum ℓ_p-energy of a unit s-t flow that the graph can support. Then,

- $\mathcal{E}_1(s, t)$ is equal to the shortest path distance between s and t, and
- $\mathcal{E}_\infty(s, t)$ is equal to the reciprocal of the edge connectivity between s and t.

Since ℓ_2 is between ℓ_1 and ℓ_∞, the effective resistance $\mathrm{Reff}(s, t) = \mathcal{E}_2^2(s, t)$ accounts for both the s-t shortest path distance and the s-t edge connectivity. Informally, the effective resistance between two vertices is small when there are many short paths connecting them.

Since the sampling probability for the spectral sparsification algorithm is proportional to effective resistance, this provides an intuitive explanation of the sparsification process:

- If the effective resistance of an edge ij is small, then i and j are well-connected, so we sample the edge with lower probability in order to reduce the number of edges.
- If the effective resistance of an edge ij is large, then this edge is crucial for connectivity, so we sample it with higher probability to preserve it.

Another characterization is that the effective resistance of an edge e is equal to the probability that edge e appears in a uniformly random spanning tree of the graph; see Problem 14.25. This property reinforces the intuition that effective resistance measures the well-connectedness of an edge.

More Properties of Effective Resistance

Since effective resistance is a useful concept in spectral graph theory, we collect additional properties for reference.

Proposition 14.16 (Metric Property)**.** *Effective resistance satisfies the triangle inequality, meaning that for any vertices a, b, c,*

$$\operatorname{Reff}(a, b) + \operatorname{Reff}(b, c) \geq \operatorname{Reff}(a, c).$$

Moreover, this metric satisfies the additional property that it can be isometrically embedded into some ℓ_1 space [LP17, Exercise 4.34].

Effective resistances are also closely related to random walks (see [DS84]). The following connection arises because hitting times can also be computed by solving Laplacian equations.

Theorem 14.17 (Effective Resistance and Commute Time)**.** *Let $G = (V, E)$ be an unweighted graph. For two vertices $s, t \in V$, let $H(s, t)$ be the hitting time, which is the expected number of steps of a random walk starting at s to reach t. Define the commute time as $C(s, t) := H(s, t) + H(t, s)$. Then,*

$$C(s, t) = 2|E| \cdot \operatorname{Reff}(s, t).$$

Using this theorem, one can show that the resistance diameter provides a logarithmic approximation to the cover time of random walks.

Theorem 14.18 (Resistance Diameter and Cover Time)**.** *Let $G = (V, E)$ be an unweighted graph. The cover time from s, denoted by $C_s(G)$, is the expected number of steps required for a random walk starting from s to visit every vertex in G at least once. The cover time of the graph is defined as $C(G) := \max_s C_s(G)$. Let $R(G) := \max_{a,b \in V} \operatorname{Reff}(a, b)$ be the resistance diameter of G. Then,*

$$|E| \cdot R(G) \leq C(G) \lesssim |E| \cdot R(G) \cdot \log |V|.$$

A d-regular expander graph with constant edge conductance has the smallest possible resistance diameter $O(1/d)$. The following result shows that one can remove a small constant fraction of edges so that each component achieves this property.

Theorem 14.19 (Low Effective Resistance Diameter Decomposition [AALO18])**.** *For any d-regular unweighted graph $G = (V, E)$ and any sufficiently large constant $c > 1$, there is a polynomial time algorithm that finds a partition of vertices $(V_1, \ldots, V_k)$ that satisfies:*

1. *Low Effective Resistance Diameter: Each induced subgraph $G[V_i]$ satisfies $R(G[V_i]) \lesssim c^3/d$.*
2. *Few crossing edges: $|\delta(V_1, \ldots, V_k)| \lesssim |E|/c$.*

This result can be compared to Theorem 12.1. Although it is not always possible to decompose a d-regular graphs into expanders with constant edge conductance while removing only a constant fraction of edges, it is always possible to find a decomposition achieving the electrical properties that expanders of constant edge conductance enjoy.

14.6 Problems

Exercise 14.20 (Spectrum of Spectral Approximator). *Let G and H be weighted undirected graphs with Laplacian spectrums $\lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$ and $\gamma_1 \leq \gamma_2 \leq \ldots \leq \gamma_n$ respectively. Prove that if H is a $(1 \pm \epsilon)$-spectral approximator of G, then $(1-\epsilon)\lambda_i \leq \gamma_i \leq (1+\epsilon)\lambda_i$ for every $1 \leq i \leq n$.*

Exercise 14.21 (Cut Sparsifier vs. Spectral Sparsifier). *Provide graphs G and H such that H is a $(1 \pm \epsilon)$-cut sparsifier of G but not a $(1 \pm \epsilon)$-spectral sparsifer of G.*

Problem 14.22 (Tight Example of Random Sampling). *The analysis of the random sampling algorithm is tight. In a complete graph, the sampling probability of every edge is the same, so Algorithm 10 reduces to the uniform sampling algorithm. Prove that random sampling cannot find a cut sparsifier with $o(n \log n)$ edges with high probability.*

Exercise 14.23 (Effective Resistance and Minimum Cut). *Show that $\mathrm{Reff}(s,t) \geq 1/\mathrm{mincut}(s,t)$, where $\mathrm{mincut}(s,t)$ denotes the minimum number of edges whose removal disconnects s and t.*

Problem 14.24 (Electrical Flow Minimizes Energy). *Prove Theorem 14.15*

Problem 14.25 (Effective Resistances and Spanning Trees). *Let $G = (V, E)$ be an undirected graph where each edge e has an integral weight $w(e)$. The weight of a spanning tree T is defined as $w_T := \prod_{e \in T} w(e)$. Let $p_T = w_T / \sum_{T'} w_{T'}$, where the sum is over all the spanning trees T' of G. Let T^* be a random spanning tree sampled from the distribution p.*

1. *Prove that $\Pr(e \in T^*) = w(e) \cdot \mathrm{Reff}(e)$ for any $e \in E$. (Hint: You may use Problem 1.24 and Fact A.29.)*

2. *Prove that*

$$\Pr(e \in T^* \mid f \in T^*) \leq \Pr(e \in T^*),$$

for any two edges $e, f \in E$. In words, conditioned on the event $f \in T^*$, the probability of the event $e \in T^*$ does not increase, meaning that the events $e \in T^*$ and $f \in T^*$ are negatively correlated.

References

[AALO18] Vedat Levi Alev, Nima Anari, Lap Chi Lau, and Shayan Oveis Gharan. Graph clustering using effective resistance. In *9th Innovations in Theoretical Computer Science Conference (ITCS 2018)*, pages 41–1. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2018. 183

[BK15] András A. Benczúr and David R. Karger. Randomized approximation schemes for cuts and flows in capacitated graphs. *SIAM J. Comput.*, 44(2):290–319, 2015. 172

[DS84] Peter G Doyle and J Laurie Snell. *Random walks and electric networks*, volume 22. American Mathematical Soc., 1984. 183

[FHHP19] Wai Shing Fung, Ramesh Hariharan, Nicholas J. A. Harvey, and Debmalya Panigrahi. A general framework for graph sparsification. *SIAM J. Comput.*, 48(4):1196–1223, 2019. 173

[Kar99] David R. Karger. Random sampling in cut, flow, and network design problems. *Math. Oper. Res.*, 24(2):383–413, 1999. 171

[Kar00] David R. Karger. Minimum cuts in near-linear time. *J. ACM*, 47(1):46–76, 2000. 179

[KT19] Ken-ichi Kawarabayashi and Mikkel Thorup. Deterministic edge connectivity in near-linear time. *J. ACM*, 66(1):4:1–4:50, 2019. 180

[LP17] Russell Lyons and Yuval Peres. *Probability on trees and networks*, volume 42. Cambridge University Press, 2017. 183

[SS11] Daniel A. Spielman and Nikhil Srivastava. Graph sparsification by effective resistances. *SIAM J. Comput.*, 40(6):1913–1926, 2011. 174, 179

[ST11] Daniel A. Spielman and Shang-Hua Teng. Spectral sparsification of graphs. *SIAM J. Comput.*, 40(4):981–1025, 2011. 173, 174

[ST14] Daniel A Spielman and Shang-Hua Teng. Nearly linear time algorithms for preconditioning and solving symmetric, diagonally dominant linear systems. *SIAM Journal on Matrix Analysis and Applications*, 35(3):835–885, 2014. 173, 178, 180

[Tro12] Joel A Tropp. User-friendly tail bounds for sums of random matrices. *Foundations of computational mathematics*, 12(4):389–434, 2012. 177

[Woo14] David P Woodruff. Sketching as a tool for numerical linear algebra. *Foundations and Trends® in Theoretical Computer Science*, 10(1–2):1–157, 2014. 180

Chapter 15

Spectral Sparsification via Potential Function

In Chapter 14, we have seen that spectral sparsification provides a stronger linear algberaic formulation of the cut sparsfication problem. This formulation connects the problem to matrix concentration inequalities, leading to an elegant solution that matches the best-known result for cut sparsification.

In this chapter, we will see that this formulation leads to a surprising solution that goes beyond what was previously known for cut sparsification. The key ingredient is a potential function that is inspired by the Stieltjes transform of the eigenvalue distribution, which leads to a deterministic algorithm to construct a linear-sized spectral sparsifier for any graph. We also discuss further developments inspired by this result.

15.1 Linear-Sized Spectral Sparsification

The main theorem that we will study is due to Batson, Spielman and Srivastava.

Theorem 15.1 (Linear-Sized Spectral Approximator of Identity Matrix [BSS14]). *For any m vectors $v_1, \ldots, v_m \in \mathbb{R}^n$ satisfying $\sum_{i=1}^m v_i v_i^\top = I_n$, there always exist scalars $s_1, \ldots, s_m$ with at most dn nonzeros such that*

$$\left(1 - \frac{1}{\sqrt{d}}\right)^2 \cdot I_n \preccurlyeq \sum_{i=1}^m s_i v_i v_i^\top \preccurlyeq \left(1 + \frac{1}{\sqrt{d}}\right)^2 \cdot I_n.$$

The ratio of the maximum eigenvalue and the minimum eigenvalue is within a factor of two of that of a Ramanujan graph, hence the name twice Ramanujan sparsifiers. From the reduction in Theorem 14.8, it follows that every graph has a linear-sized spectral sparsifier.

Theorem 15.2 (Linear-Sized Spectral Sparsifier [BSS14]). *For any edge-weighted undirected graph $G = (V = [n], E)$ and any $0 < \epsilon \leq 1$, there exists a reweighted subgraph $H = (V, F)$ on the same vertex set with at most $O(n/\epsilon^2)$ edges such that H is a $(1 \pm \epsilon)$-spectral approximator of G.*

By Lemma 14.5, a direct corollary is that every graph has a $(1 \pm \epsilon)$-cut sparsifier with at most $O(n/\epsilon^2)$ edges, improving upon Theorem 14.3 for cut sparsification. It is remarkable that solving a harder problem leads to a stronger solution in this well-studied special case. To date, no alternative approach is known for obtaining linear-sized cut sparsifiers without relying on the concept of spectral

sparsification. This remains an intriguing challenge, particularly for those who prefer combinatorial algorithms for combinatorial problems.

15.2 Deterministic Algorithm and Polynomial Perspective

The approach taken to prove Theorem 15.1 is different from the random sampling approach in Theorem 14.3 and Theorem 14.6. It follows a deterministic "greedy" strategy, where a potential function guides the algorithm in adding one vector at a time.

Intuition from Characteristic Polynomials

As discussed in [BSS14], the intuition behind their approach comes from a polynomial perspective. Let $A \in \mathbb{R}^{n\times n}$ be the current partial solution, initialized as $A = 0$. They considered the characteristic polynomial $p_A(x) = \det(xI - A) = \prod_{j=1}^{n}(x - \lambda_j)$, whose roots correspond to the eigenvalues of A, and analyzed how it evolves when a vector is added. Using the matrix determinant formula in Fact A.29,

$$p_{A+vv^\top}(x) = \det(xI - A - vv^\top) = \det(xI - A)\cdot\left(1 - v^\top(xI - A)^{-1}v\right) = p_A(x)\cdot\left(1 - \sum_{j=1}^{n}\frac{\langle v, u_j\rangle^2}{x - \lambda_j}\right).$$

where $\{\lambda_j\}_{j=1}^n$ are the eigenvalues of A and $\{u_j\}_{j=1}^n$ are the corresponding orthonormal eigenvectors. Suppose we add a uniformly random vector v from $v_1, \ldots, v_m$ to A. By the isotropy assumption,

$$\mathbb{E}\left[\langle v, u_j\rangle^2\right] = \frac{1}{m}\sum_{i=1}^{m}\langle v_i, u_j\rangle^2 = \frac{1}{m}\cdot u_j^\top\left(\sum_{i=1}^{m} v_i v_i^\top\right)u_j = \frac{\|u_j\|^2}{m} = \frac{1}{m}.$$

This implies that the expected characteristic polynomial is

$$\mathbb{E}\left[p_{A+vv^\top}(x)\right] = p_A(x)\left(1 - \frac{1}{m}\sum_{j=1}^{n}\frac{1}{x - \lambda_j}\right) = p_A(x) - \frac{1}{m}\partial_x p_A(x),$$

as $\partial_x p_A(x)/p_A(x) = \sum_{j=1}^{n} 1/(x - \lambda_j)$. Starting from $A = 0$, the initial polynomial is $p_A(x) = x^n$. After t iterations, the expected characteristic polynomial becomes

$$p_t(x) = \left(1 - \frac{1}{m}\partial_x\right)^t x^n.$$

This generates a standard family of orthogonal polynomials, known as the associated Laguerre polynomials, whose roots are well-studied. After $t = dn$ iterations, the ratio of the largest root to the smallest root of $p_{dn}(x)$ is known to be

$$\frac{d + 1 + 2\sqrt{d}}{d + 1 - 2\sqrt{d}},$$

which corresponds to the ratio of the maximum and minimum eigenvalues in Theorem 15.1.

This is only a heuristic argument, as there may not exist a vector v whose characteristic polynomial $p_{A+vv^\top}(x)$ matches the expected characteristic polynomial exactly. The proof of Theorem 15.1 in [BSS14] does not follow this approach directly, but the potential function used is inspired by the calculation here. This perspective also foreshadows the polynomial method developed by Marcus, Spielman, and Srivastava to resolve the Kadison-Singer problem [MSS14].

Algorithm Structure

As discussed in Chapter 14, one advantage of the algebraic formulation for spectral sparsification in Theorem 14.6 is that we only need to keep track of the maximum and minimum eigenvalues of the current partial solution, rather than the exponentially many cut values required in the cut sparsification problem.

The general approach is to maintain an upper bound on the maximum eigenvalue and a lower bound on the minimum eigenvalue, controlling their evolution over time. This will be achieved using two potential functions Φ^u and Φ_l, which we define and analyze in the next section.

Assuming the existence of these potential functions, we describe the structure of the deterministic greedy algorithm. Initially, we start with the empty solution $A_0 = 0$, an upper bound u_0 for the maximum eigenvalue, and a lower bound l_0 for the minimum eigenvalue, such that the initial potential values satisfy $\Phi^{u_0}(A_0) \leq \phi_u$ and $\Phi_{l_0}(A_0) \leq \phi_l$ for some fixed values of ϕ_u and ϕ_l.

In each iteration t, we select a vector v_i and a scalar s, adding $s \cdot v_i v_i^\top$ to the current solution so that $A_{t+1} \leftarrow A_t + s v_i v_i^\top$. We shift the upper and lower bounds as $u_{t+1} \leftarrow u_t + \delta_u$ and $l_{t+1} \leftarrow l_t + \delta_l$, where δ_u and δ_l are fixed increments. We maintain the invariants $\Phi^{u_{t+1}}(A_{t+1}) \leq \phi_u$ and $\Phi^{l_{t+1}}(A_{t+1}) \leq \phi_l$ and ensure that u_{t+1} and l_{t+1} remain valid bounds for the maximum and minimum eigenvalues of A_{t+1}.

Algorithm 11 Deterministic Greedy Algorithm for Spectral Sparsification

Require: Vectors $v_1, \ldots, v_m \in \mathbb{R}^n$ satisfying $\sum_{i=1}^m v_i v_i^\top = I_n$.

1: Initialization: $A_0 = 0$ and $\tau = dn$.
2: Choose u_0, l_0, ϕ_u, ϕ_l so that $\Phi^{u_0}(A_0) \leq \phi_u$ and $\Phi_{l_0}(A_0) \leq \phi_l$.
3: Choose two parameters δ_u and δ_l and set $u_t = u_0 + t \cdot \delta_u$ and $l_t = l_0 + t \cdot \delta_l$ for any $t \geq 1$.
4: **for** $1 \leq t \leq \tau$ **do**
5: Find vector $v \in \{v_1, \ldots, v_m\}$ and scalar s, and set $A_t = A_{t-1} + s \cdot vv^\top$ to maintain the invariants $\Phi^{u_t}(A_t) \leq \phi_u$, $\Phi_{l_t}(A_t) \leq \phi_l$, $\lambda_{\max}(A_t) \leq u_t$, and $\lambda_{\min}(A_t) \geq l_t$.
6: **end for**
7: **return** A_τ.

There are many parameters $u_0, l_0, \phi_u, \phi_l, \delta_u, \delta_l$ to be chosen, which we will determine at the end. For intuition, $l_0 \approx -\sqrt{d}n$, $u_0 \approx \sqrt{d}n$, $\delta_u \approx 1$, $\delta_l \approx 1$, $\phi_u \approx 1/\sqrt{d}$, and $\phi_l \approx 1/\sqrt{d}$. The logic of choosing these parameters will be clear.

15.3 Potential Functions

The key element in this algorithm is the definition of the potential functions, that will make everything work beautifully. Before stating the potential functions used in [BSS14], we first explore some natural attempts and examine what properties of the potential functions are needed.

Norm of Eigenvalues

A natural first attempt is to use the maximum and minimum eigenvalues as potential functions, i.e., $\Phi^u(A_t) = \lambda_{\max}(A_t)$ and $\Phi_l(A_t) = \lambda_{\min}(A_t)$, and then inductively prove that $\lambda_{\max}(A_t) \leq \lambda_{\max}(A_{t-1}) + \delta_u$ and $\lambda_{\min}(A_t) \leq \lambda_{\min}(A_{t-1}) + \delta_l$. However, this approach does not work well for this problem. Since A_t is an n-dimensional matrix, tracking only the maximum eigenvalue fails

to distinguish between the case where all directions grow uniformly and the case where a single direction dominates while others remain small. Ideally, after n iterations, every direction should increase by one unit. To prove this inductively, we need a potential function that allows us to argue that, on average, the largest direction is increased by $1/n$ unit per step. However, the maximum eigenvalue is not smooth or robust enough for such an argument.

From the above discussion, we need a more global measure that accounts for all directions. One possible candidate is $\mathrm{Tr}(A)/n$, the average eigenvalue of the current solution. While this quantity increases smoothly, it does not help us bound the maximum eigenvalue when the average eigenvalue is small.

Thus, we seek a potential function that is both smooth enough to track incremental progress and ensures that the maximum eigenvalue remains small when the function's value is small. Let $\vec{\lambda} = \big(\lambda_1(A), \lambda_2(A), \ldots, \lambda_n(A)\big)$ be the spectrum of the current solution. Note that $\lambda_{\max}(A) := \|\vec{\lambda}\|_\infty$ is the infinity norm of the spectrum, while $\mathrm{Tr}(A) := \|\vec{\lambda}\|_1$ is the 1-norm of the spectrum. Interpolating between these extremes, we may consider the quantity $\left(\frac{1}{n}\sum_{i=1}^n \lambda_i^p\right)^{1/p} = n^{-1/p} \cdot \|\lambda\|_p$. Setting $p \approx \log n$ provides a good approximation of $\|\vec{\lambda}\|_\infty$, but p-norms are not convenient for calculations.

A well-known function in convex optimization is the softmax function, defined as $\log \sum_{i=1}^n e^{\lambda_i}$, which is convex, differentiable, and a good approximation of the maximum eigenvalue. In matrix form, the softmax function is naturally expressed as $\log \mathrm{Tr}(e^A)$, where $e^A := \sum_{k=0}^\infty X^k/k!$ is the matrix exponential. This is a good potential function for spectral sparsification and is also used in the proof of the matrix concentration inequalities; see Chapter 23. Indeed, this function could be used to yield a deterministic algorithm with guarantees matching those of the random sampling algorithm in Theorem 14.6. However, it is not known how to use this function to achieve linear-sized spectral sparsification.

Barrier Functions

The Stieltjes transform of the eigenvalue distribution is defined as

$$S(z) = \frac{1}{n}\sum_{i=1}^n \frac{1}{z - \lambda_i} \quad \text{for } z \in \mathbb{C}^+.$$

The spectrum of a matrix corresponds to the set of poles of this rational function. In random matrix theory, eigenvalue perturbations are studied using this function [EY17].

Batson, Spielman, and Srivastava used evaluations of this function at specific points as potential functions. They also noted in [BSS14] that their potential functions were inspired by the expected characteristic polynomial calculation presented in Section 15.2.

Definition 15.3 (Barrier Functions)**.** *Given $u, l \in \mathbb{R}$ and a real symmetric matrix $A \in \mathbb{R}^{n\times n}$ with eigenvalues $\lambda_1, \ldots, \lambda_n$, the upper and lower barrier functions are defined as*

$$\Phi^A(u) := \Phi^u(A) := \mathrm{Tr}(uI_n - A)^{-1} = \sum_{i=1}^n \frac{1}{u - \lambda_i} \quad \text{and} \quad \Phi_A(l) := \Phi_l(A) := \mathrm{Tr}(A - lI_n)^{-1} = \sum_{i=1}^n \frac{1}{\lambda_i - l}.$$

We use the notations $\Phi^u(A)$ and $\Phi_l(A)$ when treating the barrier functions as functions of the matrix A for fixed u and l. Conversely, we use $\Phi^A(u)$ and $\Phi_A(l)$ when viewing them as functions of u or l for a fixed A.

The strategy in Algorithm 11 is to ensure that u_t increases slowly while maintaining the invariant that the potential value $\Phi^{u_t}(A_t)$ remains small. When $u > \lambda_{\max}(A)$ and $l < \lambda_{\min}(A)$, these functions measure how far the eigenvalues of A are from the barriers u and l, and they blow up as any eigenvalue approaches a barrier. Suppose we could maintain the invariant $\Phi^{u_t}(A) \le 1$ for all t. This ensures that u_t is a "comfortable" upper bound of the maximum eigenvalue, as there could be at most one eigenvalue at least $u_t - 1$, at most two eigenvalues at least $u_t - 2$, and so on. This is a global quantity that accounts for all eigenvalues while changing smoothly to measure the progress in each iteration.

The following properties of the barrier functions are simple but useful.

Exercise 15.4 (Monotonicity and Convexity). *Let $A \in \mathbb{R}^{n\times n}$ be a real symmetric matrix. For any $u > \lambda_{\max}(A)$ and any $\delta > 0$, the upper barrier function satisfies*

$$\Phi^A(u) \ge \Phi^A(u+\delta) \quad \text{and} \quad \Phi^A(u) + \delta \cdot \big(\Phi^A(u+\delta)\big)' \ge \Phi^A(u+\delta).$$

For any $l + \delta < \lambda_{\min}(A)$ and any $\delta > 0$, the lower barrier function satisfies

$$\Phi_A(l) \le \Phi_A(l+\delta) \quad \text{and} \quad \Phi_A(l) + \delta \cdot \big(\Phi_A(l+\delta)\big)' \ge \Phi_A(l+\delta).$$

We call them barrier functions as they resemble the log-barrier functions used in the interior point method for convex optimization.

Remark 15.5 (Log-Barrier Functions). *Let $p_A(x) = \det(xI - A)$ be the characteristic polynomial of A. Note that*

$$\Phi^x(A) = \frac{\partial_x p_A(x)}{p_A(x)} = \partial_x \log\big(p_A(x)\big) \quad \text{and} \quad \Phi_x(A) = -\partial_x \log\big(p_A(x)\big).$$

These functions blow up as x approaches a root.

15.4 Changes of Potential Values

There are elegant formulas to analyze how the barrier functions change when we add a vector and perform a rank-one update.

Upper Barrier Function

For the upper barrier function $\Phi^u(A)$, adding a vector $s \cdot vv^\top$ increase the potential value, while increasing the upper bound u compensates for this change to maintain the invariants $\Phi^{u+\delta_u}(A + s \cdot vv^\top) \le \Phi^u(A)$ and $u + \delta_u$ remains an upper bound on the maximum eigenvalue.

Lemma 15.6 (Upper Barrier Change). *Suppose $u > \lambda_{\max}(A)$. For any vector v, if*

$$\frac{1}{s} \ge \frac{v^\top\big((u+\delta_u)I - A\big)^{-2}v}{\Phi^u(A) - \Phi^{u+\delta_u}(A)} + v^\top\big((u+\delta_u)I - A\big)^{-1}v =: U_A(v),$$

then

$$\Phi^{u+\delta_u}\big(A + s\cdot vv^\top\big) \le \Phi^u(A) \quad \text{and} \quad \lambda_{\max}\big(A + s\cdot vv^\top\big) < u + \delta_u.$$

Proof. Let $u' := u + \delta_u$. By the Sherman-Morrison rank-one update formula in Fact A.22,

$$\begin{aligned}
\Phi^{u+\delta_u}(A + s\cdot vv^\top) &= \operatorname{Tr}\left(\left(u'I - A - s\cdot vv^\top\right)^{-1}\right) \\
&= \operatorname{Tr}\left(\left(u'I - A\right)^{-1} + \frac{s(u'I-A)^{-1}vv^\top(u'I-A)^{-1}}{1 - s\cdot v^\top(u'I-A)^{-1}v}\right) \\
&= \Phi^{u+\delta_u}(A) + \frac{s\cdot v^\top(u'I-A)^{-2}v}{1 - s\cdot v^\top(u'I-A)^{-1}v} \\
&= \Phi^{u}(A) - \underbrace{\left(\Phi^{u}(A) - \Phi^{u+\delta_u}(A)\right)}_{\text{gain}} + \underbrace{\frac{v^\top(u'I-A)^{-2}v}{\frac{1}{s} - v^\top(u'I-A)^{-1}v}}_{\text{loss}}.
\end{aligned}$$

Rearranging shows that $\Phi^{u+\delta_u}\left(A + s\cdot vv^\top\right) \le \Phi^u(A)$ when $\frac{1}{s} \ge U_A(v)$.

This also implies that $\lambda_{\max}(A + s\cdot vv^\top) \le u + \delta_u$, since otherwise, there exists some $s' \le s$ such that $\lambda_{\max}(A + s'\cdot vv^\top) = u + \delta_u$. In that case, we would have $\Phi^{u+\delta_u}\left(A + s'\cdot vv^\top\right) = \infty$, contradicting the bounded condition $\Phi^{u+\delta_u}\left(A + s'\cdot vv^\top\right) \le \Phi^u(A)$. □

Lower Barrier Function

For the lower barrier function $\Phi_l(A)$, adding a vector $s\cdot vv^\top$ decreases the potential value, while increasing the lower bound l increases the potential value. Note that an additional condition on the barrier value is required to ensure that we still maintain a lower bound on the minimum eigenvalue.

Lemma 15.7 (Lower Barrier Change)**.** *Suppose $\lambda_{\min}(A) > l$ and $\Phi_l(A) \le 1/\delta_l$. For any vector v, if*

$$0 < \frac{1}{s} \le \frac{v^\top\left(A - (l+\delta_l)I\right)^{-2}v}{\Phi_{l+\delta_l}(A) - \Phi_l(A)} - v^\top\left(A - (l+\delta_l)I\right)^{-1}v =: L_A(v),$$

then

$$\Phi_{l+\delta_l}\left(A + s\cdot vv^\top\right) \le \Phi_l(A) \quad \text{and} \quad \lambda_{\min}\left(A + s\cdot vv^\top\right) > l + \delta_l.$$

Proof. Note that $\lambda_{\min}(A) > l$ and $\Phi_l(A) = \sum_{i=1}^n 1/(\lambda_i - l) \le 1/\delta_l$ imply that $1/(\lambda_{\min} - l) < 1/\delta_l$ and thus $\lambda_{\min} > l + \delta_l$. Thus, $\lambda_{\min}(A + s\cdot vv^\top) \ge \lambda_{\min}(A) > l + \delta_l$. Using a similar calculation with the Sherman-Morrison formula as in Lemma 15.6,

$$\Phi_{l+\delta_l}(A + s\cdot vv^\top) = \Phi_l(A) + \underbrace{\left(\Phi_{l+\delta_l}(A) - \Phi_l(A)\right)}_{\text{loss}} - \underbrace{\frac{v^\top(A - l'I)^{-2}v}{\frac{1}{s} + v^\top(A-l'I)^{-1}v}}_{\text{gain}}.$$

Rearranging shows that $\Phi_{l+\delta_l}\left(A + s\cdot vv^\top\right) \le \Phi_l(A)$ when $\frac{1}{s} \le L_A(v)$. □

15.5 Averaging Argument

We need to prove that there exists a vector v and a scalar s such that both the assumptions in Lemma 15.6 and Lemma 15.7 hold. This will ensure that the invariants $\Phi^{u+\delta_u}\left(A + s\cdot vv^\top\right) \le \Phi^u(A)$, $\lambda_{\max}\left(A + s\cdot vv^\top\right) < u + \delta_u$, $\Phi_{l+\delta_l}\left(A + s\cdot vv^\top\right) \le \Phi_l(A)$, and $\lambda_{\min}\left(A + s\cdot vv^\top\right) > l + \delta_l$ in Algorithm 11 hold simultaneously.

The idea in [BSS14] is to prove that $\sum_{i=1}^m L_A(v_i) \geq \sum_{i=1}^m U_A(v_i)$, which guarantees the existence of a vector v_i such that $L_A(v_i) \geq U_A(v_i)$. By choosing s such that $L_A(v_i) \geq \frac{1}{s} \geq U_A(v_i)$, both assumptions in Lemma 15.6 and Lemma 15.7 hold, ensuring that all invariants hold simultaneously for $A + s \cdot v_i v_i^\top$.

Upper Barrier Function

Lemma 15.8 (Total Upper Barrier Shift). *Given $v_1, \ldots, v_m \in \mathbb{R}^n$ such that $\sum_{i=1}^m v_i v_i^\top = I_n$,*

$$\sum_{i=1}^m U_A(v_i) \leq \frac{1}{\delta_u} + \Phi^u(A).$$

Proof. Using the isotropy assumption $\sum_{i=1}^m v_i v_i^\top = I_n$, it follows that

$$\sum_{i=1}^m v_i^\top \big((u+\delta_u)I - A\big)^{-2} v_i = \sum_{i=1}^m \mathrm{Tr}\left(\big((u+\delta_u)I - A\big)^{-2} v_i v_i^\top\right) = \mathrm{Tr}\left(\big((u+\delta_u)I - A\big)^{-2}\right),$$

and similarly

$$\sum_{i=1}^m v_i^\top \big((u+\delta_u)I - A\big)^{-1} v_i = \mathrm{Tr}\left(\big((u+\delta_u)I - A\big)^{-1}\right) = \Phi^{u+\delta_u}(A).$$

By the convexity of the barrier function $\Phi^u(A) = \Phi^A(u)$ in terms of u in Exercise 15.4, the "gain" is

$$\Phi^u(A) - \Phi^{u+\delta_u}(A) = \Phi^A(u) - \Phi^A(u+\delta_u) \geq -\delta_u \cdot \Big(\Phi^A(u+\delta_u)\Big)' = \delta_u \cdot \mathrm{Tr}\left(\big((u+\delta_u)I - A\big)^{-2}\right),$$

where the last equality uses Fact A.42. Therefore,

$$\begin{aligned}
\sum_{i=1}^m U_A(v_i) &:= \sum_{i=1}^m \left(\frac{v_i^\top \big((u+\delta_u)I - A\big)^{-2} v_i}{\Phi^u(A) - \Phi^{u+\delta_u}(A)} + v_i^\top \big((u+\delta_u)I - A\big)^{-1} v_i\right) \\
&= \frac{\mathrm{Tr}\left(\big((u+\delta_u)I - A\big)^{-2}\right)}{\Phi^u(A) - \Phi^{u+\delta_u}(A)} + \Phi^{u+\delta_u}(A) \\
&\leq \frac{1}{\delta_u} + \Phi^u(A),
\end{aligned}$$

where the last inequality uses the monotonicity in Exercise 15.4. □

Lower Barrier Function

The calculations for the total lower barrier shift is similar but slightly more intricate. Note that the following lemma also requires the assumption that $\Phi_l(A) \leq 1/\delta_l$ as in Lemma 15.7.

Lemma 15.9 (Total Lower Barrier Shift). *Given $v_1, \ldots, v_m \in \mathbb{R}^n$ such that $\sum_{i=1}^m v_i v_i^\top = I_n$, if $\Phi_l(A) \leq 1/\delta_l$, then*

$$\sum_{i=1}^m L_A(v_i) \geq \frac{1}{\delta_l} - \Phi_l(A).$$

Proof. As in the proof of Lemma 15.8, using the isotropy assumption $\sum_{i=1}^m v_i v_i^\top = I_n$,

$$\sum_{i=1}^m v_i^\top \big(A-(l+\delta_l)I\big)^{-2} v_i = \mathrm{Tr}\left(\big(A-(l+\delta_l)I\big)^{-2}\right) \quad \text{and} \quad \sum_{i=1}^m v_i^\top \big(A-(l+\delta_l)I\big)^{-1} v_i = \Phi_{l+\delta_l}(A).$$

Therefore,

$$\begin{aligned}\sum_{i=1}^m L_A(v_i) &:= \sum_{i=1}^m \left(\frac{v_i^\top \big(A-(l+\delta_l)I\big)^{-2} v_i}{\Phi_{l+\delta_l}(A)-\Phi_l(A)} - v_i^\top \big(A-(l+\delta_l)I\big)^{-1} v_i\right)\\ &= \frac{\mathrm{Tr}\left(\big(A-(l+\delta_l)I\big)^{-2}\right)}{\Phi_{l+\delta_l}(A)-\Phi_l(A)} - \Phi_{l+\delta_l}(A).\end{aligned}$$

Using convexity in Exercise 15.4 as in the proof of Lemma 15.8,

$$\Phi_{l+\delta_l}(A)-\Phi_l(A) = \Phi_A(l+\delta_l)-\Phi_A(l) \le \delta_l \cdot \big(\Phi_A(l+\delta_l)\big)' = \delta_l \cdot \mathrm{Tr}\,\big(A-(l+\delta_l)I\big)^{-2}.$$

This gives $\sum_{i=1}^m L_A(v_i) \ge \frac{1}{\delta_l} - \Phi_{l+\delta_l}(A)$, which is slightly weaker than the desired bound and insufficient for maintaining invariants throughout the algorithm. To prove the stated bound, we need to work harder and show that

$$\frac{\mathrm{Tr}\left(\big(A-(l+\delta_l)I\big)^{-2}\right)}{\Phi_{l+\delta_l}(A)-\Phi_l(A)} - \Phi_{l+\delta_l}(A) \ge \frac{1}{\delta_l} - \Phi_l(A),$$

which, upon rearrangement, is equivalent to the following claim.

Claim 15.10 (Lemma 4.3 of [MSS21]). *If $\Phi_l(A) \le 1/\delta_l$, then*

$$\big(\Phi_{l+\delta_l}(A)-\Phi_l(A)\big)^2 \le \mathrm{Tr}\left(\big(A-(l+\delta_l)I\big)^{-2}\right) - \frac{1}{\delta_l}\big(\Phi_{l+\delta_l}(A)-\Phi_l(A)\big).$$

Proof. By definition of the lower barrier function in Definition 15.3

$$\big(\Phi_{l+\delta_l}(A)-\Phi_l(A)\big)^2 = \left(\sum_{i=1}^n \left(\frac{1}{\lambda_i-(l+\delta_l)} - \frac{1}{\lambda_i - l}\right)\right)^2 = \left(\sum_{i=1}^n \frac{\delta_l}{\big(\lambda_i-(l+\delta_l)\big)\cdot(\lambda_i-l)}\right)^2$$

Applying Cauchy-Schwarz inequality and the assumption $\delta_l \cdot \Phi_l(A) \le 1$, the right hand side satisfies

$$\le \left(\sum_{i=1}^n \frac{\delta_l}{\lambda_i - l}\right)\left(\sum_{i=1}^n \frac{\delta_l}{\big(\lambda_i-(l+\delta_l)\big)^2\cdot(\lambda_i-l)}\right) \le \sum_{i=1}^n \frac{\delta_l}{\big(\lambda_i-(l+\delta_l)\big)^2\cdot(\lambda_i-l)}.$$

Check that this is equal to the right hand side of the claim. □

The claim completes the proof of this lemma. □

Both Barrier Functions

Combining Lemma 15.8 and Lemma 15.9 with the averaging argument in this section, we arrive at the following conditions for the invariants in Algorithm 11 to hold throughout.

Lemma 15.11 (Invariants). *Let $A_0 = 0$. If we choose $u_0 > 0, l_0 < 0, \phi_u, \phi_l, \delta_u, \delta_l$ so that*

$$\Phi^{u_0}(A_0) \le \phi_u \quad \text{and} \quad \Phi_{l_0}(A_0) \le \phi_l \quad \text{and} \quad \phi_l \le \frac{1}{\delta_l} \quad \text{and} \quad \frac{1}{\delta_l} - \phi_l \ge \frac{1}{\delta_u} + \phi_u,$$

then Algorithm 11 can always find a vector v and a scalar s in each iteration t to maintain the invariants $\Phi^{u_t}(A_t) \le \phi_u$ and $\Phi_{l_t}(A_t) \le \phi_l$ and $\lambda_{\max}(A_t) \le u_t$ and $\lambda_{\min}(A_t) \ge l_t$, where $u_t = u_0 + t\delta_u$ and $l_t = l_0 + t\delta_l$ as defined in Algorithm 11.

Proof. The proof is by a simple induction. The induction hypothesis is that $\Phi^{u_t}(A_t) \le \phi_u$ and $\Phi_{l_t}(A_t) \le \phi_l$ and $\lambda_{\max}(A_t) \le u_t$ and $\lambda_{\min}(A_t) \ge l_t$. This holds at $t = 0$ by our assumptions. For the induction step, by Lemma 15.8 and Lemma 15.9 and our assumption,

$$\sum_{i=1}^{m} U_{A_t}(v_i) \le \frac{1}{\delta_u} + \Phi^{u_t}(A_t) \le \frac{1}{\delta_u} + \phi_u \le \frac{1}{\delta_l} - \phi_l \le \frac{1}{\delta_l} - \Phi_{l_t}(A_t) \le \sum_{i=1}^{m} L_{A_t}(v_i).$$

So, there exists some $v \in \{v_1, \ldots, v_m\}$ such that $U_{A_t}(v) \le L_{A_t}(v)$. Let s be a scalar such that $U_{A_t}(v) \le \frac{1}{s} \le L_{A_t}(v)$. Then, by Lemma 15.6 and Lemma 15.7, the invariants hold for $t+1$ with $A_{t+1} = A_t + s \cdot vv^\top$ and $u_{t+1} = u_t + \delta_u$ and $l_{t+1} = l_t + \delta_l$. □

Wrapping Up

With Lemma 15.11, it remains to choose $u_0, l_0, \phi_u, \phi_l, \delta_u, \delta_l$ to prove Theorem 15.1. Choose $\phi_u, \phi_l, \delta_u, \delta_l$ such that the last inequality in Lemma 15.11 holds as an equality. The choice of ϕ_u, ϕ_l determines the initial eigenvalue lower bound l_0 and upper bound u_0. The objective is to minimize the ratio of $u_0 + dn \cdot \delta_u$ and $l_0 + dn \cdot \delta_l$, which forms a multi-parameter optimization problem.

Batson, Spielman and Srivastava set

$$l_0 := -\sqrt{d}n, \quad u_0 := \Big(\frac{d+\sqrt{d}}{\sqrt{d}-1}\Big)n, \quad \phi_l := \Phi_{l_0}(A_0) = -\frac{n}{l_0} = \frac{1}{\sqrt{d}}, \quad \phi_u := \Phi^{u_0}(A_0) = \frac{n}{u_0} = \frac{\sqrt{d}-1}{d+\sqrt{d}},$$

so that the first two conditions in Lemma 15.11 are satisfied. Then, set

$$\delta_l := 1 \quad \text{and} \quad \delta_u := \frac{\sqrt{d}+1}{\sqrt{d}-1} \quad \Longrightarrow \quad \frac{1}{\delta_l} - \phi_l = \frac{1}{\delta_u} + \phi_u,$$

ensuring that the last two conditions in Lemma 15.11 are also satisfied. Therefore, after dn iterations of Algorithm 11,

$$\frac{\lambda_{\max}(A_{dn})}{\lambda_{\min}(A_{dn})} \le \frac{u_{dn}}{l_{dn}} = \frac{u_0 + dn \cdot \delta_u}{l_0 + dn \cdot \delta_l} = \frac{\frac{d+\sqrt{d}}{\sqrt{d}-1} + d \cdot \frac{\sqrt{d}+1}{\sqrt{d}-1}}{-\sqrt{d}+d} = \left(\frac{\sqrt{d}+1}{\sqrt{d}-1}\right)^2,$$

completing the proof of Theorem 15.1.

15.6 Further Developments

There has been extensive subsequent work on spectral sparsification, and we discuss some of these developments here. A major breakthrough is the resolution of the Kadison-Singer problem [MSS14].

Fast Algorithms

The deterministic greedy algorithm in [BSS14] provides a polynomial time algorithm to construct a linear-sized spectral sparsifier. It is thus natural to ask whether a near-linear time algorithm exists.

Regret Minimization: Allen-Zhu, Liao, and Orecchia [ALO15] constructed linear-sized spectral sparsifiers using the regret minimization framework in convex optimization. This provides a more systematic way to derive the result in [BSS14] and offers a different interpretation of the result as employing a new regularizer within the regret minimization framework. This framework yields a quadratic time algorithm for linear-sized spectral sparsification. The tools developed are more convenient for certain applications. We discuss the regret minimization approach in the next chapter for spectral sparsification and in Chapter 18 for the cut-matching game.

Near-Linear Time Algorithms: Lee and Sun gave an almost-linear time algorithm [LS18] and later a nearly-linear time algorithm [LS17] to construct linear-sized spectral sparsifiers.

Their first algorithm [LS18] is an adaptive sampling algorithm that is interesting and easy to describe. In the first iteration, the algorithm samples say $n^{0.99}$ vectors using effective resistance, as in Theorem 14.6. Then, it updates the sampling probability using barrier functions and repeat this process for $n^{0.01}$ iterations. Their intuition comes from the balls-and-bins model, where the maximum load remains constant after throwing $n^{0.99}$ balls into n bins.

Their second algorithm [LS17] uses the new potential functions $\mathrm{Tr}\left(\exp\left((uI-A)^{-1}\right)\right)$ and $\mathrm{Tr}\left(\exp\left((A-lI)^{-1}\right)\right)$, and reduces the problem to a "packing" semidefinite program.

Alternative Approaches

Besides independent random sampling in the previous chapter and the potential function based approach in this chapter, there are other interesting methods for constructing spectral sparsifiers.

Random Spanning Trees: An interesting approach analyzed by Kyng and Song [KS18] shows that the union of $O(\log(n)/\epsilon^2)$ random spanning trees forms a $(1 \pm \epsilon)$-spectral approximator, and this bound is tight. The underlying reason that this random sampling algorithm works is that the effective resistance of an edge e is equal to the probability that e belongs to a random spanning tree; see Problem 14.25. To establish this result, they developed a new matrix concentration inequality for negatively associated random variables.

Cycle Decomposition: Another approach by Cohen et al. [CGP+20] is based on decomposing the edge set of the graph into even cycles (plus a linear number of edges) and applying a dependent rounding method that samples alternating edges in the even cycles to sparsify the graph. The main advantage of this approach is that the sparsifier is degree-preserving, which is crucial in constructing

more refined notions of spectral sparsifiers. We will discuss degree-preserving spectral sparsification in the next chapter.

Related Notions

Spectral sparsification has inspired new notions with applications in approximation algorithms.

Thin Tree: A spanning tree T is an α-thin tree of a graph $G = (V, E)$ if $|\delta_T(S)| \leq \alpha \cdot |\delta_G(S)|$ for every $S \subseteq V$. It is conjectured that every k-edge-connected graph has an $O(1/k)$-thin tree.

Anari and Oveis Gharan [AO15] defined a stronger notion called a spectrally thin tree, which requires that $L_T \preccurlyeq \alpha L_G$. Using this stronger notion, along with various techniques from spectral graph theory (including the solution to the Kadison-Singer problem, which we will discuss soon), they made significant progress on the thin tree conjecture and used it to design improved approximation algorithm for the asymmetric traveling salesman problem.

Spectral Rounding: In the spectral rounding problem, we are given a graph $G = (V, E)$ and a fraction solution $x : E \to \mathbb{R}_+$, and the goal is to find an integral solution $z : E \to \mathbb{Z}_+$ such that $\sum_{e \in E} x_e L_e \approx \sum_{e \in E} z_e L_e$ along with some linear constraints $\sum_{e \in E} c_e \cdot x_e \approx \sum_{e \in E} c_e \cdot z_e$. The techniques for linear-sized spectral sparsification can be extended to solve the one-sided spectral rounding problem, where

$$\sum_{e \in E} z_e L_e \succcurlyeq \sum_{e \in E} x_e L_e \quad \text{and} \quad \sum_{e \in E} c_e \cdot z_e \approx \sum_{e \in E} c_e \cdot x_e,$$

leading to optimal approximation algorithms for experimental design problems [ALSW21, LZ22a] and a spectral approach for network design problems [LZ22b].

Kadison-Singer Problem

Last but not least, Marcus, Spielman, Srivastava [MSS14] have extended the techniques from linear-sized spectral sparsification to construct bipartite Ramanujan graphs and resolve the Kadison-Singer problem, stated in the following form.

Theorem 15.12 (Solution to Weaver's Conjecture [MSS14]). *Let $v_1, \ldots, v_m \in \mathbb{R}^n$ be a set of vectors such that*

$$\sum_{i=1}^{m} v_i v_i^\top = I_n \quad \textit{and} \quad \|v_i\|_2^2 \leq \epsilon \ \ \textit{for} \ \ 1 \leq i \leq n,$$

there exists a partition of $\{1, \ldots, m\}$ into two sets S_1 and S_2 so that

$$\lambda_{\max}\left(\sum_{i \in S_j} v_i v_i^\top\right) \leq \frac{1}{2}\left(1 + \sqrt{2\epsilon}\right)^2 \ \ \textit{for} \ \ 1 \leq j \leq 2.$$

A consequence for spectral sparsification is that if every edge has small effective resistance, then there exists an unweighted spectral sparsifier. This can be seen as an analog of Karger's result, which states that if the minimum cut is large, then there exists an unweighted cut sparsifier.

The proof introduces a novel probabilistic method, incorporating many interesting new ideas, including the concept of interlacing polynomials and a multivariate version of the barrier function.

Question 15.13 (Efficient Algorithm for the Kadison-Singer Problem)**.** *A major open question is whether an efficient algorithm can be designed to compute the partition in Theorem 15.12.*

References

[ALO15] Zeyuan Allen-Zhu, Zhenyu Liao, and Lorenzo Orecchia. Spectral sparsification and regret minimization beyond matrix multiplicative updates. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC*, pages 237–245. ACM, 2015. 8, 196, 203, 228

[ALSW21] Zeyuan Allen-Zhu, Yuanzhi Li, Aarti Singh, and Yining Wang. Near-optimal discrete optimization for experimental design: A regret minimization approach. *Mathematical Programming*, 186:439–478, 2021. 197

[AO15] Nima Anari and Shayan Oveis Gharan. Effective-resistance-reducing flows, spectrally thin trees, and asymmetric tsp. In *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*, pages 20–39. IEEE, 2015. 197

[BSS14] Joshua D. Batson, Daniel A. Spielman, and Nikhil Srivastava. Twice-ramanujan sparsifiers. *SIAM Rev.*, 56(2):315–334, 2014. 187, 188, 189, 190, 193, 196

[CGP+20] Timothy Chu, Yu Gao, Richard Peng, Sushant Sachdeva, Saurabh Sawlani, and Junxing Wang. Graph sparsification, spectral sketches, and faster resistance computation via short cycle decompositions. *SIAM Journal on Computing*, 52(6):FOCS18–85, 2020. 196

[EY17] László Erdős and Horng-Tzer Yau. *A dynamical approach to random matrix theory*, volume 28. American Mathematical Soc., 2017. 190

[KS18] Rasmus Kyng and Zhao Song. A matrix chernoff bound for strongly rayleigh distributions and spectral sparsifiers from a few random spanning trees. In *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS*, pages 373–384. IEEE Computer Society, 2018. 196

[LS17] Yin Tat Lee and He Sun. An sdp-based algorithm for linear-sized spectral sparsification. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC*, pages 678–687. ACM, 2017. 196

[LS18] Yin Tat Lee and He Sun. Constructing linear-sized spectral sparsification in almost-linear time. *SIAM J. Comput.*, 47(6):2315–2336, 2018. 196

[LZ22a] Lap Chi Lau and Hong Zhou. A local search framework for experimental design. *SIAM Journal on Computing*, 51(4):900–951, 2022. 197

[LZ22b] Lap Chi Lau and Hong Zhou. A spectral approach to network design. *SIAM Journal on Computing*, 51(4):1018–1064, 2022. 197

[MSS14] Adam W. Marcus, Daniel A. Spielman, and Nikhil Srivastava. Ramanujan graphs and the solution of the Kadison Singer problem. In *Proceedings of the international Congress of Mathematicians*, volume 3, pages 363–386. 2014. 8, 188, 196, 197

[MSS21] Adam W Marcus, Daniel A Spielman, and Nikhil Srivastava. Interlacing families iii: Sharper restricted invertibility estimates. *Israel Journal of Mathematics*, pages 1–28, 2021. 194

Chapter 16

Spectral Sparsification and Algorithmic Discrepancy Theory

In this chapter, we study a recent approach to construct spectral sparsifiers using techniques from algorithmic discrepancy theory. This approach provides additional flexibility in incorporating constraints, such as preserving degrees exactly, that are crucial in certain recent applications. Moreover, it provides principled proofs and fast algorithms using the regularized optimization framework and convex geometric techniques.

16.1 Discrepancy Theory and Matrix Sparsification

In this section, we briefly review the settings in discrepancy theory, and then state the main technical theorems about matrix partial coloring and matrix sparsification in this chapter.

Vector Discrepancy

In the vector discrepancy problem, we are given vectors $v_1, \ldots, v_n \in \mathbb{R}^m$, and the goal is to find a coloring $s \in \{\pm 1\}^n$ to minimize $\left\|\sum_i s(i) \cdot v_i\right\|_\infty$. This captures several well-studied combinatorial discrepancy problems. For example, a fundamental result in combinatorial discrepancy theory, proved by Spencer [Spe85], states that for binary vectors $v_i \in \{0,1\}^m$, there exists a coloring with discrepancy $O\big(\sqrt{n \log(m/n)}\big)$. This improves upon the bound obtained by a random coloring.

Many classical results in combinatorial discrepancy theory were established using non-constructive probabilistic methods [Spe85] and convex geometric techniques [Ban98]. Over the past fifteen years, there has been significant progress in developing efficient algorithms to match these non-constructive bounds using various ideas from random walks and optimization [Ban10, LM15, Rot17, BDGL18].

Matrix Discrepancy

A natural generalization of the vector setting is the matrix setting. In the matrix discrepancy problem, we are given matrices $A_1, \ldots, A_n \in \mathbb{R}^{m \times m}$, and the goal is to find a coloring $s \in \{\pm 1\}^n$ to minimize $\left\|\sum_i s(i) \cdot A_i\right\|_{\mathrm{op}}$. The matrix Spencer conjecture [Zou12] states that if $\|A_i\|_{\mathrm{op}} \leq 1$ for $1 \leq i \leq n$, then there is a coloring with discrepancy $O\big(\sqrt{n \log(m/n)}\big)$. There has been significant recent progress towards this conjecture [BJM23], relying on a new matrix concentration inequality using free probability [BBVH23].

Matrix Partial Coloring

A common technique in discrepancy theory is to find a partial (fractional) coloring, where a constant fraction of the entries are set to ± 1, which can then be applied recursively to obtain a full coloring.

Motivated by the matrix Spencer problem, Reis and Rothvoss [RR20] proved the following matrix partial coloring theorem, which was later refined to incorporate linear constraints in [JRT24].

Theorem 16.1 (Matrix Partial Coloring with Linear Constraints [RR20, JRT24]). *Let $A_1, \dots, A_m \in \mathbb{R}^{n \times n}$ be real symmetric matrices such that $\sum_{i=1}^m |A_i| \preccurlyeq I_n$ (where for a real symmetric matrix A with eigendecomposition $A = \sum_j \alpha_j u_j u_j^\top$, the matrix $|A|$ is defined as $|A| := \sum_j |\alpha_j| u_j u_j^\top$). Let $\mathcal{C} \subseteq \mathbb{R}^m$ be the set of good partial fractional colorings, defined as*

$$\mathcal{C} := \left\{ x \in \mathbb{R}^m : \left\| \sum_{i=1}^m x(i) \cdot A_i \right\|_{\mathrm{op}} \leq 16\sqrt{\frac{n}{m}} \right\}.$$

In addition, let $\mathcal{H} \subseteq \mathbb{R}^m$ be a linear subspace of dimension at least $\frac{4}{5}m$. There exists a polynomial time algorithm that returns a partial fractional coloring $x \in [-1,1]^m$ such that

$$x \in \mathcal{C} \cap \mathcal{H} \quad \text{and} \quad |\{i \in [m] \mid x(i) = \pm 1\}| \geq c \cdot m \quad \text{for some constant } c.$$

Matrix Sparsification

Reis and Rothvoss [RR20] showed that their partial coloring theorem can be applied to obtain the following matrix sparsification result, which was refined to incorporate linear constraints in [JRT24].

Theorem 16.2 (Matrix Sparsification with Linear Constraints [RR20, JRT24]). *Given positive semidefinite matrices $A_1, A_2, \dots, A_m \in \mathbb{R}^{n \times n}$ such that $\sum_{i=1}^m A_i \preccurlyeq I_n$ and a linear subspace $\mathcal{H} \subseteq \mathbb{R}^m$ of dimension $m - O(n)$, there is a polynomial time algorithm to construct a reweighting $s \in \mathbb{R}^m_+$ such that*

$$\|A(s) - A(\vec{1})\|_{\mathrm{op}} \lesssim \epsilon, \quad \text{and} \quad s - \vec{1}_m \in \mathcal{H}, \quad \text{and} \quad |\operatorname{supp}(s)| \lesssim \frac{n}{\epsilon^2},$$

where, for $x \in \mathbb{R}^m$, $A(x)$ denotes the shorthand for $\sum_{i=1}^m x(i) \cdot A_i$.

This is a more general sparsification result than Theorem 15.1, as it holds for arbitrary positive semidefinite matrices (see also [dCSHS16]), not just for matrices of the form $v_i v_i^\top$.

Linear-Sized Degree-Preserving Spectral Sparsification

The linear constraints allow one to construct spectral sparsifiers satisfying additional constraints, such as preserving the weighted degrees of vertices exactly. The existence of linear-sized degree-preserving spectral sparsifiers was not known before.

Theorem 16.3 (Linear-Sized Degree-Preserving Spectral Sparsification [JRT24]). *For any edge-weighted undirected graph $G = (V = [n], E)$ and any $0 < \epsilon < 1$, there exists a reweighted subgraph $H = (V, F)$ on the same vertex set with at most $O(n/\epsilon^2)$ edges such that H is a $(1 \pm \epsilon)$-spectral approximator of G, also satisfying*

$$w_G(\delta_G(i)) = w_H(\delta_H(i)) \quad \text{for all } i \in V.$$

Proof. Given an edge-weighted undirected graph $G = (V, E)$, for each edge $e \in E$, define a positive semidefinite matrix $A_e := w_G(e) \cdot L_G^{\dagger/2} b_e b_e^\top L_G^{\dagger/2}$, so that $\sum_{e \in E} A_e \preccurlyeq I_n$, as in Theorem 14.8. For the degree constraints, define the $(m-n)$-dimensional subspace

$$\mathcal{H} := \left\{ x \;\middle|\; \sum_{j:ij \in E} x(ij) \cdot w_G(ij) = 0 \;\; \forall i \in V \right\}.$$

Applying Theorem 16.2 with $A_1, \ldots, A_m$ and $\mathcal{H}$, we obtain a reweighted subgraph $H = (V, E)$ with at most $O(n/\epsilon^2)$ nonzero edges with edge weights $w_H(ij) = s(ij) \cdot w_G(ij)$. This ensures that

$$(1-\epsilon) L_G \preccurlyeq L_H \preccurlyeq (1+\epsilon) L_G \quad \text{and} \quad \sum_{j:ij \in E} w_H(ij) = \sum_{j:ij \in E} s(ij) \cdot w_G(ij) = \sum_{j:ij \in E} w_G(ij) \;\; \text{for all } i \in V,$$

where the first condition is by applying the linear transformation $L_G^{1/2}$ on A_e as in Theorem 14.8, and the second condition follows from the linear subspace so that $\sum_{j:ij \in E} (s(ij) - 1) \cdot w_G(ij) = 0$ for all $i \in V$. □

Degree-preserving spectral sparsifiers are crucial in constructing spectral sparsifiers for directed graphs, with applications in solving Laplacian equations and estimating random walk probabilities in directed graphs (see [JSS+25, LWZ25] and the references therein). Previously, degree-preserving spectral sparsifiers were constructed using the cycle decomposition method discussed in Section 15.6.

16.2 From Matrix Partial Coloring to Matrix Sparsification

In this section, we use the matrix partial coloring result in Theorem 16.1 to derive the matrix sparsification result in Theorem 16.2. The reduction is due to Reis and Rothvoss [RR20]. The idea is to find a partial coloring x with small discrepancy, ensuring that a constant fraction of entries of x being -1, and use it to zero out a constant fraction of the entries in s in each iteration. In the ideal case when $x \in \{\pm 1\}^m$ is a full coloring with small discrepancy, we either double an entry of s or set it to zero in an iteration.

Proof of Theorem 16.2 assuming Theorem 16.1. We first assume that Theorem 16.1 can be successfully applied to obtain the desired partial coloring in Step (3) of the algorithm in all iterations, and verify that the output $\vec{s}$ of the algorithm satisfies the three properties stated in the theorem.

First, we check that the algorithm terminates. Note that Step (3) and (4) imply that $\vec{x}_t$ has at least $\frac{1}{2} c \cdot m_t$ entries with value -1 in the support of $\vec{s}_t$, which ensures that the support size satisfies $m_{t+1} \le (1 - \frac{c}{2}) \cdot m_t$ by Step (5) of the algorithm. Therefore, the support size decreases by a constant factor in each iteration, so the while loop terminates within $O(\log(\epsilon^2 m/n))$ iterations. When the algorithm terminates, the support size of the output is $O(n/\epsilon^2)$.

Next, we verify that the linear subspace constraint is satisfied. From Step (5) of the algorithm,

$$\vec{s}_1 = \vec{1} \quad \textit{and} \quad \vec{s}_{t+1} = \vec{s}_t + \operatorname{diag}(\vec{s}_t) \cdot \vec{x}_t \;\; \forall t \quad \Longrightarrow \quad \vec{s} = \vec{1} + \sum_{t=1}^{\tau-1} \operatorname{diag}(\vec{s}_t) \cdot \vec{x}_t.$$

Since $\vec{x}_t \in \mathcal{H}_t$ for all t, we have $\operatorname{diag}(\vec{s}_t) \cdot \vec{x}_t \in \mathcal{H}$ for all t. This implies that $\vec{s} - \vec{1} \in \mathcal{H}$.

Finally, we check the matrix approximation guarantee. By telescoping and the triangle inequality,

$$\|A(\vec{s}) - A(\vec{1}_m)\| \le \sum_{t=1}^{\tau-1} \left\| \sum_{i=1}^{m} \left(\vec{s}_{t+1}(i) - \vec{s}_t(i)\right) \cdot A_i \right\| = \sum_{t=1}^{\tau-1} \left\| \sum_{i=1}^{m} \vec{x}_t(i) \cdot \vec{s}_t(i) \cdot A_i \right\| \lesssim \sum_{t=1}^{\tau-1} \sqrt{\frac{n}{m_t}} \lesssim \epsilon, \quad (16.1)$$

Algorithm 12 Matrix Sparsification with Linear Subspace Constraints

Require: Positive semidefinite matrices $A_1, A_2, ..., A_m \in \mathbb{R}^{n\times n}$ such that $\sum_{i=1}^m A_i \preccurlyeq I_n$, a linear subspace $\mathcal{H} \subseteq \mathbb{R}^m$ of dimension $m - \beta n$ for a constant β, and a target accuracy parameter ϵ.

1: Initialization: Set $\vec{s}_1 = \vec{1}$ and $t = 1$.

2: **while** $m_t := |\operatorname{supp}(\vec{s}_t)| > 5\beta n/\epsilon^2$ for some fixed constant c **do**

3: Apply Theorem 16.1 to obtain a partial coloring $\vec{x}_t \in [-1, 1]^m$ in the subspace

$$\mathcal{H}_t := \{\vec{x} \in \mathbb{R}^m \mid \operatorname{diag}(\vec{s}_t) \cdot \vec{x} \in \mathcal{H}\} \cap \{\vec{x} \in \mathbb{R}^m \mid \vec{x}(i) = 0 \ \forall i \notin \operatorname{supp}(\vec{s}_t)\}$$

such that

$$\left\|\sum_{i=1}^m \vec{x}_t(i) \cdot \vec{s}_t(i) \cdot A_i\right\|_{\text{op}} \leq 64\sqrt{\frac{n}{m_t}} \quad \text{and} \quad \left|\{i \mid \vec{x}_t(i) = \pm 1\}\right| \geq c \cdot m_t.$$

4: If there are more entries with $\vec{x}_t(i) = 1$ than with $\vec{x}_t(i) = -1$, then update $\vec{x}_t \leftarrow -\vec{x}_t$.

5: Update $\vec{s}_{t+1}(i) \leftarrow \vec{s}_t(i) \cdot (1 + \vec{x}_t(i))$ for $1 \leq i \leq m$, and set $t \leftarrow t + 1$.

6: **end while**

7: **return** $\vec{s} = \vec{s}_\tau$ where τ is the last iteration.

where the last inequality follows as m_t is a geometric sequence dominated by $m_{\tau-1} \geq 5\beta n/\epsilon^2$.

It remains to show that we can indeed find the desired partial coloring $\vec{x}_t$ in each iteration. We maintain that $\sum_{i=1}^m \vec{s}_t(i) \cdot A_i \preccurlyeq 2I_n$ at each iteration t, which clearly holds in the first iteration as $s_1 = \vec{1}$. This implies that $\sum_{i=1}^m \frac{1}{2}\vec{s}_{t-1}(i) \cdot A_i \preccurlyeq I_n$ satisfies the assumption of Theorem 16.1. Since the input subspace $\mathcal{H}_t$ has dimension $m_t - \beta n \geq \frac{4}{5}m_t$ for $m_t = 5\beta n/\epsilon^2$, we can apply Theorem 16.1 with the nonzero matrices in $\left\{\frac{1}{2}\vec{s}_t(i)A_i\right\}_{i=1}^m$ as the input matrices and $\mathcal{H}_t$ as the input subspace to find a partial coloring x_t that satisfies the guarantees in Step (3) of the algorithm in polynomial time. The property $\sum_{i=1}^m \vec{s}_t(i) \cdot A_i \preccurlyeq 2I_n$ is maintained by the same argument in (16.1) as long as ϵ is a sufficiently small constant. This completes the proof of Theorem 16.2. □

16.3 Potential Function from Regularized Optimization

To find a good partial coloring, a common approach in algorithmic discrepancy theory is to perform some variants of continuous random walk [Ban10, LM15, RR20], starting from the origin $x = \vec{0}$.

The approach that we will take in the next section is to perform a deterministic discrepancy walk guided by a potential function. In this section, we first introduce the potential function and derive a bound on how it changes under small perturbation.

Reducing Operator Norm to Maximum Eigenvalue

The objective in Theorem 16.1 is to bound the operator norm $\|A(x)\|_{\text{op}}$ of the partial coloring $x \in [-1, 1]^m$. A standard trick in discrepancy theory for handling the operator norm is the reduction

$$\|A(x)\|_{\text{op}} = \left\|\sum_{i=1}^m x(i) \cdot A_i\right\|_{\text{op}} = \lambda_{\max}\left(\sum_{i=1}^m x(i) \cdot \begin{pmatrix} A_i & 0 \\ 0 & -A_i \end{pmatrix}\right),$$

which only increases the dimension of the matrices by a factor of two. For the remainder of this chapter, we will abuse notation by using A_i to denote the blow-up matrix so that we only need to bound the maximum eigenvalue $\lambda_{\max}(A(x))$.

This is an advantage of the more general formulation in Theorem 16.1, which allows the input matrices $A_1, \ldots, A_m$ to be arbitrary symmetric matrices (not just rank one symmetric matrices), so that the above reduction applies and simplifies the analysis. In contrast, in Chapter 15, two potential functions were used to track the maximum and minimum eigenvalues separately, making the analysis more involved.

Regularized Optimization

Allen-Zhu, Liao, and Orecchia [ALO15] developed a regularized optimization framework to derive the spectral sparsification result in Theorem 14.6 in a more principled way.

Note that the maximum eigenvalue $\lambda_{\max}(A)$ of a matrix A can be formulated as

$$\lambda_{\max}(A) := \max_{M \in \Delta_n} \langle A, M \rangle \quad \text{where} \quad \Delta_n := \{M \succcurlyeq 0 \mid \operatorname{Tr}(M) = 1\}$$

is the set of density matrices. In the regularized optimization framework, the potential function is a regularized version of the maximum eigenvalue

$$\Phi(x) := \max_{M \in \Delta_n} \langle A(x), M \rangle - \frac{\phi(M)}{\eta},$$

where the regularizer $\phi(M)$ is a non-positive strongly convex function, and η is a parameter controlling the contribution of the regularization term.

They showed that the negative entropy regularizer $\phi(M) := \langle M, \log M \rangle$ can be used to obtain a deterministic algorithm to recover the $O(n \log(n)/\epsilon^2)$ spectral sparsification result in Theorem 14.8, and the $\ell_{1/2}$-regularizer $\phi(M) = -2 \operatorname{Tr}(M^{\frac{1}{2}})$ can be used to recover the $O(n/\epsilon^2)$ spectral sparsification result in Theorem 15.2.

Potential Function

It is then natural to set the potential function as

$$\Phi(x) := \max_{M \in \Delta_n} \langle A(x), M \rangle + \frac{2 \operatorname{Tr}(M^{\frac{1}{2}})}{\eta}. \tag{16.2}$$

A simple application of Cauchy-Schwarz gives $\operatorname{Tr}(M^{\frac{1}{2}}) \leq \sqrt{n}$ for $M \in \Delta_n$, implying

$$\lambda_{\max}(A(x)) \leq \Phi(x) \leq \lambda_{\max}(A(x)) + \frac{2\sqrt{n}}{\eta}. \tag{16.3}$$

The following is the closed-form solution for the optimizer and the potential function, which is quite similar to the potential function in Definition 15.3.

Lemma 16.4 (Closed-Form Solution of Potential Function)**.** *Given symmetric matrices $A_1, \cdots, A_m \in \mathbb{R}^{n \times n}$ and $x \in \mathbb{R}^m$, the unique optimizer in (16.2) is*

$$M = (u_x I_n - \eta A(x))^{-2}$$

where $u_x \in \mathbb{R}$ is the unique value such that $M \in \Delta_n$. Moreover, the potential function is

$$\Phi(x) = \frac{1}{\eta} \operatorname{Tr}\left((u_x I_n - \eta A(x))^{-1}\right) + \frac{u_x}{\eta}.$$

The proof uses the KKT optimality conditions of (16.2) to compute the optimizer M, and then the eigendecomposition of M to compute the closed-form solution of the potential function; see Exercise 16.8.

Informally, the value u_x in the potential function $\Phi(x)$ corresponds to the upper bound u in the barrier function $\Phi^u(A)$ in Definition 15.3. In the sparsification algorithm (Algorithm 11) in the previous chapter, the value of u is manually adjusted in each iteration. One can view the potential function in Lemma 16.4 as an automatic version of the barrier function in Definition 15.3, where the value u is adjusted automatically so that $\operatorname{Tr}(M) = 1$.

Change of Potential Function

Once the potential function is chosen, the next task is to bound its increase under perturbations.

Lemma 16.5 (Potential Increase [LWZ25])**.** *For any $y \in \mathbb{R}^m$, the change in the potential function is bounded by*

$$\Phi(x+y) - \Phi(x) \le \operatorname{Tr}(MA(y)) + c\eta \operatorname{Tr}\left(M^{\frac{1}{2}} A(y) M^{\frac{1}{2}} A(y) M^{\frac{1}{2}}\right)$$

where $M = (u_x I_n - \eta A(x))^{-2}$, and $|c| \le 2$ as long as $\|M^{\frac{1}{2}} \cdot \eta A(y)\|_{\mathrm{op}} \le \frac{1}{2}$.

Proof Sketch. Using Lemma 16.4, the potential increase is

$$\Phi(x+y) - \Phi(x) = \frac{1}{\eta}\Big(\operatorname{Tr}\left((u_{x+y} I_n - \eta A(x+y))^{-1}\right) - \operatorname{Tr}\left((u_x I_n - \eta A(x))^{-1}\right)\Big) + \frac{u_{x+y} - u_x}{\eta}.$$

Consider the univariate function $g(u) = \operatorname{Tr}\left((uI_n - \eta A(x+y))^{-1}\right)$. Check that $g(u)$ is convex in the interval between u_x and u_{x+y} for sufficiently small perturbation y. It follows that $g(u_{x+y}) \le g(u_x) + (u_{x+y} - u_x) \cdot g'(u_{x+y})$. By Lemma 16.4, $(u_{x+y} I_n - \eta A(x+y))^{-2}$ is a density matrix, so the derivative of $g(u)$ at u_{x+y} is

$$g'(u_{x+y}) = -\operatorname{Tr}\left((u_{x+y} I_n - \eta A(x+y))^{-2}\right) = -1.$$

Thus, $g(u_{x+y}) \le g(u_x) - (u_{x+y} - u_x)$, which implies

$$\Phi(x+y) - \Phi(x) \le \frac{1}{\eta}\left(\operatorname{Tr}\left((u_x I_n - \eta A(x+y))^{-1}\right) - \operatorname{Tr}\left((u_x I_n - \eta A(x))^{-1}\right)\right).$$

Using a Taylor approximation for the trace of the inverse of a matrix from [RR20, Lemma 11],

$$\operatorname{Tr}((X - \eta Y)^{-1}) = \operatorname{Tr}(X^{-1}) + \eta \operatorname{Tr}(X^{-1} Y X^{-1}) + c\eta^2 \operatorname{Tr}(X^{-1} Y X^{-1} Y X^{-1}),$$

for $|c| \le 2$ when $X \succ 0$ and Y is sufficiently small, setting $X := u_x I_n - \eta A(x)$ and $Y := A(y)$ gives

$$\operatorname{Tr}((u_x I_n - \eta A(x+y))^{-1}) - \operatorname{Tr}((u_x I_n - \eta A(x))^{-1}) = \eta \operatorname{Tr}(MA(y)) + c\eta^2 \operatorname{Tr}(M^{\frac{1}{2}} A(y) M^{\frac{1}{2}} A(y) M^{\frac{1}{2}}).$$

Substituting this back completes the proof sketch of Lemma 16.5. □

16.4 Matrix Partial Coloring by Discrepancy Walk

The goal in this section is to prove the matrix partial coloring result in Theorem 16.1.

A typical analysis in algorithmic discrepancy theory establishes that as long as the degree of freedom is large, a random perturbation of the current partial solution has small discrepancy. We present a deterministic version, showing that as long as the dimension of the restricted subspace is large, there exists a good direction in which the potential function does not increase much.

The Deterministic Discrepancy Walk Framework

We use the deterministic discrepancy walk framework in [PV23, LWZ25] to compute a partial coloring with small discrepancy, with the potential function Φ from Lemma 16.4.

Algorithm 13 Deterministic Discrepancy Walk for Matrix Partial Coloring

Require: Real symmetric matrices $A_1, A_2, ..., A_m \in \mathbb{R}^{n\times n}$ such that $\sum_{i=1}^m |A_i| \preccurlyeq I_n$.

1: Initialization: Set $x_0 = \vec{0}_m$ as the initial point. Set $\alpha = 1/\operatorname{poly}(m)$ as the maximal step size. Set $t = 1$ and $H_1 = [m]$ as the initial set of active coordinates.
2: **while** $m_t := |H_t| > \frac{3}{4}m$ **do**
3: Pick y_t as a unit vector from an appropriate subspace $\mathcal{U}_t \subseteq \mathbb{R}^m$ satisfying $y_t \perp x_{t-1}$, $\operatorname{supp}(y_t) \subseteq H_t$, and $\Phi(x_{t-1} + y_t) - \Phi(x_{t-1})$ is bounded.
4: Let δ_t be the largest step size such that $\delta_t \le \alpha$ and $x_{t-1} + \delta_t y_t \in [-1, 1]^m$.
5: Update $x_t \leftarrow x_{t-1} + \delta_t y_t$. Update $t \leftarrow t + 1$.
6: Update $H_t := \{i \in [n] \mid |x_{t-1}(i)| < 1\}$ as the set of fractional coordinates.
7: **end while**
8: **return** $x := x_\tau$, where τ is the last iteration.

We set the maximal step size of δ_t for all t to be $\alpha := \frac{1}{2\eta}$ to ensure that

$$\left\|M^{\frac{1}{2}} \cdot \eta A(\delta_t y_t)\right\|_{\mathrm{op}} \le \eta \left\|\sum_{i=1}^m \delta_t \cdot y_t(i) \cdot A_i\right\|_{\mathrm{op}} \le \eta \cdot \delta_t \cdot \|y_t\|_\infty \cdot \left\|\sum_{i=1}^m |A_i|\right\|_{\mathrm{op}} \le \eta \cdot \delta_t \le \frac{1}{2}, \tag{16.4}$$

where we used that M is a density matrix, y_t is a unit vector, and the assumption $\sum_{i=1}^m |A_i| \preccurlyeq I_n$. Therefore, we can apply (16.3) and Lemma 16.4 to bound

$$\begin{aligned}\lambda_{\max}(A(x_\tau)) &\le \frac{2\sqrt{n}}{\eta} + \Phi(x_\tau) = \frac{2\sqrt{n}}{\eta} + \sum_{t=1}^{\tau} \left(\Phi(x_t) - \Phi(x_{t-1})\right) \\ &\le \frac{2\sqrt{n}}{\eta} + \sum_{t=1}^{\tau} \left(\operatorname{Tr}\left(M_{t-1} A(\delta_t y_t)\right) + c_t \eta \operatorname{Tr}\left(M_{t-1}^{\frac{1}{2}} A(\delta_t y_t) M_{t-1}^{\frac{1}{2}} A(\delta_t y_t) M_{t-1}^{\frac{1}{2}}\right)\right), \end{aligned} \tag{16.5}$$

where $|c_t| \le 2$ and $M_t = (u_{x_t} I - \eta A(x_t))^{-2}$ for all t.

Restricted Subspaces

The key task is to find an appropriate unit vector y_t so that the potential increase in (16.5) is bounded. The nontrivial part is bounding the second-order term.

The idea is to express the second-order term as the quadratic form of a matrix N_t and restrict y_t to lie in the low eigenspace of N_t. Formally, let H_t be the active coordinates in the t-th iteration and $m_t = |H_t|$. The second-order term in (16.5) can be written as

$$\mathrm{Tr}\left(M_{t-1}^{\frac{1}{2}} A(y_t) M_{t-1}^{\frac{1}{2}} A(y_t) M_{t-1}^{\frac{1}{2}}\right) = \sum_{i,j \in H_t} y_t(i) \cdot y_t(j) \cdot \mathrm{Tr}\left(M_{t-1}^{\frac{1}{2}} A_i M_{t-1}^{\frac{1}{2}} A_j M_{t-1}^{\frac{1}{2}}\right) = (y_t|_{H_t})^\top N_t (y_t|_{H_t}),$$

where $y_t|_{H_t} \in \mathbb{R}^{m_t}$ is obtained by restricting y_t to the coordinates in H_t, and N_t is the $(m_t \times m_t)$-dimensional matrix defined as

$$N_t := \left\{ \mathrm{Tr}\left(M_{t-1}^{\frac{1}{2}} A_i M_{t-1}^{\frac{1}{2}} A_j M_{t-1}^{\frac{1}{2}}\right) \right\}_{i,j \in H_t}.$$

Observe that N_t is a positive semidefinite matrix. Let $N_t = \sum_{i=1}^{m_t} \lambda_i u_i u_i^\top$ be the eigenvalue decomposition of N_t with $0 \le \lambda_1 \le \lambda_2 \le \cdots \le \lambda_{m_t}$. To bound the second order term, we restrict $y_t|_{H_t}$ to lie in the low eigenspace spanned by $\{u_1, \ldots, u_{m_t/3}\}$. With these notations, we formally define the good subspace for the unit update vector y_t as $\mathcal{U}^t := U^0 \cap U^1 \cap U^2 \cap U^3$ where

$$\begin{aligned}
U^0 &= \{y \in \mathbb{R}^m \mid y_i = 0 \text{ for all } i \notin H_t\}, \\
U^1 &= \{y \in \mathbb{R}^m \mid y \perp x_{t-1}\}, \\
U^2 &= \left\{ y \in \mathbb{R}^m \mid \mathrm{Tr}(M_{t-1} A(y)) = \sum_{i=1}^{m} y(i) \cdot \mathrm{Tr}(M_{t-1} A_i) = 0 \right\}, \\
U^3 &= \left\{ y \in \mathbb{R}^m \mid y|_{H_t} \in \mathrm{span}\{u_1, u_2, \cdots, u_{m_t/3}\} \right\}.
\end{aligned}$$

The subspaces are chosen as follows:

- U^0 ensures that only active coordinates are updated.
- U^1 ensures that $\|x_t\|_2^2$ is monotone increasing, which bounds the number of iterations.
- U^2 ensures that the linear term in (16.5) is zero.
- U^3 ensures that the second order term in (16.5) is small.

Since $m_t \ge \frac{3}{4}m$ while the algorithm is running, it follows that

$$\dim(\mathcal{U}^t) \ge m_t - 2 - \frac{2m_t}{3} = \frac{m_t}{3} - 2 \ge \frac{m}{4} - 2. \tag{16.6}$$

Proof of Theorem 16.1

The following lemma provides an upper bound on the eigenvalue of the low eigenspace, which is used to upper bound the second-order term.

Lemma 16.6 (Potential Increase in Low Eigenspace)**.** *Given real symmetric matrices $A_1, \cdots, A_m \in \mathbb{R}^{n \times n}$ such that $\sum_{i=1}^m |A_i| \preccurlyeq I_n$, any unit vector $y \in U^0 \cap U^3$ satisfies*

$$\mathrm{Tr}\left(M_{t-1}^{\frac{1}{2}} A(y) M_{t-1}^{\frac{1}{2}} A(y) M_{t-1}^{\frac{1}{2}}\right) \le \frac{9\sqrt{n}}{m_t^2}.$$

We will prove Lemma 16.6 using a spectral argument in the next subsection. In this subsection, we first assume Lemma 16.6 to complete the proof of Theorem 16.1.

Proof of Theorem 16.1. Given the input matrices $A_1, \cdots, A_m$ such that $\sum_{i=1}^m |A_i| \preccurlyeq I_n$ and the linear subspace $\mathcal{H}$, we apply the deterministic discrepancy walk Algorithm 13 for matrix partial coloring with the subspace $\mathcal{U}_t = \mathcal{U}^t \cap \mathcal{H}$. By (16.6),

$$\dim(\mathcal{U}_t) \geq \dim(\mathcal{U}^t) - \dim(\mathcal{H}^\perp) \geq \frac{m}{4} - 2 - \frac{m}{5} > 0$$

as long as $m \gtrsim 1$. Thus, in each iteration, there is always a unit vector $y_t \in \mathcal{U}_t$.

As shown in (16.4), by taking the maximal step size to be $\alpha = \frac{1}{2\eta}$, we can assume that $|c_t| \leq 2$ in (16.5). Therefore,

$$\lambda_{\max}\left(A(x_\tau)\right) \leq \frac{2\sqrt{n}}{\eta} + \sum_{t=1}^{\tau} \left(\delta_t \operatorname{Tr}(M_t A(y_t)) + 2\eta\delta_t^2 \operatorname{Tr}\left(M_t^{\frac{1}{2}} A(y_t) M_t^{\frac{1}{2}} A(y_t) M_t^{\frac{1}{2}} \right) \right).$$

Since $y_t \in U^2$, the linear term is $\operatorname{Tr}(M_t A(y_t)) = 0$. As $y_t \in U^0 \cap U^3$, by Lemma 16.6, the second order term satisfies $\operatorname{Tr}(M_t^{\frac{1}{2}} A(y_t) M_t^{\frac{1}{2}} A(y_t) M_t^{\frac{1}{2}}) \leq 9\sqrt{n}/m_t^2$. Therefore,

$$\lambda_{\max}\left(A(x_\tau)\right) \leq \frac{2\sqrt{n}}{\eta} + \frac{18\eta\sqrt{n}}{m_t^2} \sum_{t=1}^{\tau} \delta_t^2 \leq \frac{2\sqrt{n}}{\eta} + \frac{32\eta\sqrt{n}}{m^2} \sum_{t=1}^{\tau} \delta_t^2,$$

where the last inequality holds as $m_t \geq 3m/4$ before the while loop terminates.

Since $y_t \in U^1$ (such that $y_t \perp x_{t-1}$ for all $t \in [\tau]$) and $\|x_\tau\| \in [-1, 1]^m$, it follows that

$$m \geq \|x_\tau\|_2^2 = \left\| \sum_{t=1}^{\tau} (x_t - x_{t-1}) \right\|_2^2 = \left\| \sum_{t=1}^{\tau} \delta_t y_t \right\|_2^2 = \sum_{t=1}^{\tau} \|\delta_t y_t\|_2^2 = \sum_{t=1}^{\tau} \delta_t^2.$$

Therefore, by setting $\eta = \frac{1}{4}\sqrt{m}$, we conclude that

$$\lambda_{\max}\left(A(x_\tau)\right) \leq \frac{2\sqrt{n}}{\eta} + \frac{32\eta\sqrt{n}}{m} \leq 16\sqrt{\frac{n}{m}}.$$

Finally, to show the polynomial runtime, we bound the number of iterations of Algorithm 13. Note that in step 5, the update either (i) freezes a new coordinate, or (ii) increases the squared length of the solution $\|x_t\|^2 = \|x_{t-1}\|^2 + \alpha^2$ by α^2 since $y_t \perp x_{t-1}$. Clearly, the number of the first type of iterations is at most m. The number of the second type of iterations is at most m/α^2 as $\|x_\tau\|^2 \leq m$. Therefore, by our choice of $\alpha = \frac{1}{2\eta} = \frac{2}{\sqrt{m}}$, the total number of iterations is at most $m/\alpha^2 + m = O(m^2)$. □

Spectral Argument

Proof of Lemma 16.6. For a unit vector $y \in U^0$, the restricted vector $y|_{H_t}$ is a unit vector in $\mathbb{R}^{H_t}$. Since $y \in U^3$, the second-order term is bounded by the eigenvalue of N_t in the low eigenspace, so that

$$\operatorname{Tr}\left(M_t^{\frac{1}{2}} A(y) M_t^{\frac{1}{2}} A(y) M_t^{\frac{1}{2}} \right) = (y|_{H_t})^\top N_t (y|_{H_t}) \leq \lambda_{m_t/3}(N_t).$$

To bound $\lambda_{m_t/3}(N_t)$, the idea is to upper bound the trace of a large principal submatrix $\tilde{N}_t$ of N_t, and to use Cauchy interlacing theorem (Theorem A.15) to bound $\lambda_{m_t/3}$. Let

$$S = \left\{ i \in H_t \;\middle|\; \operatorname{Tr}\left(M_t^{\frac{1}{2}} |A_i| \right) \geq \frac{3 \operatorname{Tr}\left(M_t^{\frac{1}{2}}\right)}{m_t} \right\}$$

be the set of "large" active coordinates. Note that

$$\sum_{i\in H_t} \mathrm{Tr}\left(M_t^{\frac{1}{2}}|A_i|\right) \le \mathrm{Tr}\left(M_t^{\frac{1}{2}}\right)\cdot\left\|\sum_{i=1}^m |A_i|\right\|_{\mathrm{op}} \le \mathrm{Tr}\left(M_t^{\frac{1}{2}}\right),$$

where the first inequality is follows from Fact A.37 and the second inequality from our assumption. Since each $\mathrm{Tr}\left(M_t^{\frac{1}{2}}|A_i|\right) \ge 0$ by Fact A.36, it follows from Markov's inequality that $|S| \le \frac{1}{3}m_t$. Let $\tilde{N}_t$ be the principal submatrix of N restricted to the indices in $H_t - S$. Then $\dim(\tilde{N}_t) \ge m_t - |S| \ge \frac{2}{3}m_t$. By Cauchy interlacing in Theorem A.15, $\lambda_{m_t/3}(N_t) \le \lambda_{m_t/3}(\tilde{N}_t)$. To bound $\lambda_{m_t/3}(\tilde{N}_t)$, we simply compute the trace of $\tilde{N}_t$ and use an averaging argument. Applying Lemma A.38 (from [RR20, Lemma 10]) to each diagonal entry of $\tilde{N}_t$,

$$\mathrm{Tr}(\tilde{N}_t) = \sum_{i\in H_t-S} \mathrm{Tr}\left(M_t A_i M_t^{\frac{1}{2}} A_i\right) \le \sum_{i\in H_t-S} \mathrm{Tr}(M_t|A_i|)\cdot \mathrm{Tr}\left(M_t^{\frac{1}{2}}|A_i|\right).$$

Now, by the definition of S, the assumption that $\sum_{i=1}^m |A_i| \preccurlyeq I_n$, and (16.3), we obtain

$$\mathrm{Tr}(\tilde{N}_t) \le \frac{3\,\mathrm{Tr}\left(M_t^{\frac{1}{2}}\right)}{m_t}\cdot \sum_{i\in H_t-S} \mathrm{Tr}(M_t|A_i|) \le \frac{3\,\mathrm{Tr}\left(M_t^{\frac{1}{2}}\right)}{m_t}\cdot \mathrm{Tr}(M_t) \le \frac{3\sqrt{n}}{m_t}.$$

Since $\dim(\tilde{N}_t) \ge \frac{2}{3}m_t$, the average value of the eigenvalues of $\tilde{N}_t$ is $\mathrm{Tr}(\tilde{N}_t)/\dim(\tilde{N}_t) \le \frac{9}{2}\sqrt{n}/m_t^2$. As $\tilde{N}_t$ is positive semidefinite, by Markov's inequality, at most half of the eigenvalues of $\tilde{N}_t$ exceed $9\sqrt{n}/m_t^2$. Combining the inequalities, we conclude that, for any unit vector $y \in U^0 \cap U^3$,

$$\mathrm{Tr}\left(M_t^{\frac{1}{2}} A(y) M_t^{\frac{1}{2}} A(y) M_t^{\frac{1}{2}}\right) \le \lambda_{m_t/3}(N_t) \le \lambda_{m_t/3}(\tilde{N}_t) \le \lambda_{\dim(\tilde{N}_t)/2}(\tilde{N}_t) \le \frac{9\sqrt{n}}{m_t^2}.$$

□

16.5 Fast Algorithms

Algorithmic discrepancy theory also leads to fast algorithms for spectral sparsification [JRT24]. The approach is based on an elegant theorem by Rothvoss.

Theorem 16.7 (Partial Coloring by Gaussian Projection [Rot17]). *For any symmetric convex set $K \subseteq \mathbb{R}^n$ with Gaussian measure at least $e^{-O(n)}$, there exists a point $x \in K \cap [-1,1]^n$ with $\Omega(n)$ many coordinates in $\{-1,1\}$. Moreover, let $g \in N^n(0,1)$ be a random Gaussian vector, then*

$$x^* = \mathrm{argmin}\{\|g - x\|_2 \mid x \in K \cap [-1,1]^n\}$$

is such a point with probability $1 - 2^{-\Omega(n)}$.

Reis and Rothvoss [RR20] and Jambulapati, Reis, and Tian [JRT24] developed convex geometric techniques to prove that the convex bodies associated with spectral sparsification have Gaussian measure at least $e^{-O(n)}$. Fast algorithms were developed in [JRT24] to solve the Gaussian projection problem approximately to find a partial coloring. This leads to an interesting new approach for designing a near-linear time algorithm to compute a linear-sized spectral sparsifier, and an

almost-linear time algorithm for computing a linear-sized degree-preserving spectral sparsifier as in Theorem 16.3.

The techniques used to prove Gaussian measure lower bound in [RR20, JRT24] and those used in analyzing the deterministic discrepancy walk in this chapter [PV23, LWZ25] are similar. An interesting direction for future work is to unify these techniques into a broader framework to obtain faster algorithms and tighter bounds for more general spectral sparsification problems.

16.6 Problems

Exercise 16.8 (Closed-Form Solution of Potential Function). *Prove Lemma 16.4. Hint: A simple perturbation argument shows that the optimizer M is non-singular, which implies that the optimal dual variable associated with the constraint $M \succcurlyeq 0$ must be zero. Then use the Lagrangian optimality condition to compute the optimizer M, and use the eigendecomposition of M to derive the closed-form solution of the potential function.*

Exercise 16.9 (Unit-Circle Approximation for Undirected Graphs [LWZ25]). *Let $G = (V, E)$ be an edge-weighted graph with diagonal degree matrix D_G and adjacency matrix A_G. An edge-weighted graph $H = (V, F)$ is called a $(1 \pm \epsilon)$-unit-circle spectral approximator of G if*

1. *$D_H = D_G$,*
2. *$D_H - A_H$ is a $(1 \pm \epsilon)$-spectral approximation of $D_G - A_G$, and*
3. *$D_H + A_H$ is a $(1 \pm \epsilon)$-spectral approximation of $D_G + A_G$.*

Use Theorem 16.1 to prove that any graph G has a $(1 \pm \epsilon)$-unit-circle spectral approximation H with $O(|V|/\epsilon^2)$ edges.

References

[ALO15] Zeyuan Allen-Zhu, Zhenyu Liao, and Lorenzo Orecchia. Spectral sparsification and regret minimization beyond matrix multiplicative updates. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC*, pages 237–245. ACM, 2015. 8, 196, 203, 228

[Ban98] Wojciech Banaszczyk. Balancing vectors and gaussian measures of n-dimensional convex bodies. *Random Structures & Algorithms*, 12(4):351–360, 1998. 199

[Ban10] Nikhil Bansal. Constructive algorithms for discrepancy minimization. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pages 3–10. IEEE, 2010. 199, 202

[BBVH23] Afonso S Bandeira, March T Boedihardjo, and Ramon Van Handel. Matrix concentration inequalities and free probability. *Inventiones mathematicae*, 234(1):419–487, 2023. 199

[BDGL18] Nikhil Bansal, Daniel Dadush, Shashwat Garg, and Shachar Lovett. The gram-schmidt walk: a cure for the banaszczyk blues. In *Proceedings of the 50th annual acm sigact symposium on theory of computing*, pages 587–597, 2018. 199

[BJM23] Nikhil Bansal, Haotian Jiang, and Raghu Meka. Resolving matrix spencer conjecture up to polylogarithmic rank. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, pages 1814–1819, 2023. 199

[dCSHS16] Marcel Kenji de Carli Silva, Nicholas J. A. Harvey, and Cristiane M. Sato. Sparse sums of positive semidefinite matrices. *ACM Trans. Algorithms*, 12(1):9:1–9:17, 2016. 200

[JRT24] Arun Jambulapati, Victor Reis, and Kevin Tian. Linear-sized sparsifiers via near-linear time discrepancy theory. In *Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 5169–5208. SIAM, 2024. 200, 208, 209

[JSS+25] Arun Jambulapati, Sushant Sachdeva, Aaron Sidford, Kevin Tian, and Yibin Zhao. Eulerian graph sparsification by effective resistance decomposition. In *Proceedings of the 2025 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1607–1650. SIAM, 2025. 201

[LM15] Shachar Lovett and Raghu Meka. Constructive discrepancy minimization by walking on the edges. *SIAM Journal on Computing*, 44(5):1573–1582, 2015. 199, 202

[LWZ25] Lap Chi Lau, Robert Wang, and Hong Zhou. Spectral sparsification by deterministic discrepancy walk. In *2025 Symposium on Simplicity in Algorithms (SOSA)*, pages 315–340. SIAM, 2025. 201, 204, 205, 209

[PV23] Lucas Pesenti and Adrian Vladu. Discrepancy minimization via regularization. In *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1734–1758. SIAM, 2023. 205, 209

[Rot17] Thomas Rothvoss. Constructive discrepancy minimization for convex sets. *SIAM Journal on Computing*, 46(1):224–234, 2017. 199, 208

[RR20] Victor Reis and Thomas Rothvoss. Linear size sparsifier and the geometry of the operator norm ball. In *Proceedings of the Thirty-First Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2337–2348, 2020. 200, 201, 202, 204, 208, 209

[Spe85] Joel Spencer. Six standard deviations suffice. *Transactions of the American mathematical society*, 289(2):679–706, 1985. 199

[Zou12] Anastasios Zouzias. A matrix hyperbolic cosine algorithm and applications. In *International Colloquium on Automata, Languages, and Programming*, pages 846–858. Springer, 2012. 199

Part IV

Cut Matching Games and Matrix Multiplicative Weight Update

The concept of expander flows has been influential in both the design of approximation algorithms and fast graph algorithms. We study an interesting cut-matching game in constructing expander graphs, which has found various applications in algorithm design. We then introduce the general matrix multiplicative weight method and use it to derive cut-matching games and to design an almost linear-time $O(\sqrt{\log n})$-approximation algorithm for the sparsest cut problem.

Chapter 17

Cut-Matching Game

The cut-matching game was introduced to design a fast approximation algorithm for the sparsest cut problem through the expander flow framework. In this chapter, we begin by reviewing previous work on approximating sparsest cuts, and then study the cut-matching game by Khandekar, Rao, and Vazirani [KRV09] and discuss its various applications.

17.1 Sparsest Cut and Expander Flows

The sparsest cut problem is one of the most studied problems in approximation algorithms, leading to important techniques for designing approximation algorithms and beyond.

We consider a special case called the uniform sparsest cut problem, which minimizes the ratio between the number of edges cut and the number of pairs disconnected.

Definition 17.1 (Uniform Sparsest Cut and Edge Expansion)**.** *Given a graph $G = (V, E)$, the uniform sparsest cut problem is to find a subset $S \subseteq V$ that minimizes the ratio*

$$\frac{|\delta(S)|}{|S| \cdot |V \setminus S|}.$$

The edge expansion of a set $S \subseteq V$ and the graph G are defined as

$$\varphi(S) = \frac{|\delta(S)|}{|S|} \quad \text{and} \quad \varphi(G) = \min_{S:|S| \leq |V|/2} \varphi(S).$$

Note that the objective value of the sparsest cut problem is between $\varphi(G)/|V|$ and $\varphi(G)/(2|V|)$, so the two problems are essentially equivalent. Henceforth, we focus on the edge expansion problem.

One approach to approximating edge expansion is the spectral partitioning algorithm discussed in Chapter 2. It runs in near-linear time and yields a constant factor approximation if the edge expansion is large, but the approximation ratio is poor if the edge expansion is small.

Linear Programming Relaxation and Multicommodity Flows

An important result in approximating sparsest cuts is the linear programming-based approximation algorithm by Leighton and Rao [LR99], which establishes an approximate min-max relation between sparsest cut and maximum multicommodity flow.

In the maximum multicommodity flow problem, the objective is to maximize ν so that every pair of vertices $i, j \in V$ can simultaneously route ν units of (fractional) flow in the graph, while ensuring that every edge has congestion at most one. This problem can be formulated as a linear program.

Let ν^* be the optimal value of the maximum multicommodity flow problem, and let φ^* be the optimal value of the uniform sparsest cut problem. Leighton and Rao proved that

$$\nu^* \leq \varphi^* \lesssim \nu^* \cdot \log |V|. \tag{17.1}$$

The first inequality follows directly from a simple combinatorial argument. The second inequality is established using the region growing technique introduced in [LR99], which has numerous applications in designing approximation algorithms. This result provides a polynomial time $O(\log |V|)$-approximation algorithm for the uniform sparsest cut problem.

The approximate min-max relation was later generalized to the non-uniform sparsest cut problem in [LLR95], introducing the influential technique of metric embeddings into the study of approximation algorithms. We refer the reader to [Tre17] for an exposition of these results.

Semidefinite Programming Relaxation and Expander Flows

A major breakthrough in approximating the uniform sparsest cut problem is the $O(\sqrt{\log |V|})$-approximation algorithm by Arora, Rao, and Vazirani [ARV09]. They developed powerful geometric techniques to analyze the following semidefinite programming relaxation of the problem.

$$\begin{aligned} \min_{v_1,\ldots,v_n \in \mathbb{R}^n} \quad & \sum_{ij \in E} \|v_i - v_j\|_2^2 \\ \text{subject to} \quad & \sum_{i<j} \|v_i - v_j\|_2^2 = 1 \\ & \|v_i - v_k\|_2^2 + \|v_k - v_j\|_2^2 \geq \|v_i - v_k\|_2^2 \qquad \forall i, j, k \in V. \end{aligned} \tag{17.2}$$

If the triangle inequalities are removed, this semidefinite program is equivalent to computing the second eigenvalue of the normalized Laplacian matrix when the graph is regular (up to a scaling factor, see Lemma 9.5 and Lemma 26.20). The triangle inequalities are satisfied by the integral solutions but violated by the fractional solutions of the integrality gap examples for spectral partitioning (see Section 2.4), providing a tighter relaxation.

The dual of this semidefinite program involves embedding an expander graph into the input graph to certify its edge expansion (see Section 19.2 for a formal proof of this statement). The following claim generalizes the first inequality in (17.1).

Claim 17.2 (Graph Embedding)**.** *An edge-weighted graph $H = (V, F)$ can be embedded into an edge-weighted graph $G = (V, E)$ with congestion c if, for all edges $ij \in F$, $w_H(ij)$ units of flow can be simultaneously routed between i and j in G, while ensuring that every edge in G has congestion at most $c \cdot w_G(ij)$.*

If H can be embedded in G with congestion c, then $\varphi(G) \geq \varphi(H)/c$, where the edge expansion of a set S in an edge-weighted graph is defined as $\varphi(S) = w(\delta(S))/|S|$.

The result by Leighton and Rao can be interpreted as embedding a complete graph into the input graph G to certify its edge expansion. The approach by Arora, Rao, and Vazirani improves upon this by embedding a sparse expander graph H instead of a dense complete graph. At a high level,

their technique combines spectral methods (used to certify the edge expansion of H) with linear programming methods (used to embed H into G), leading to an improved approximation ratio. Again, we refer the reader to [Tre17] for an exposition of these results.

We will study an almost linear time $O(\sqrt{\log |V|})$-approximation algorithm for the uniform sparsest cut problem in Chapter 19, where the geometric techniques and the expander flow framework will be discussed in more detail.

17.2 Cut-Matching Game

The approximation algorithms by linear programming and semidefinite programming both involve solving multicommodity flow problems, which are not known to be implementable in nearly linear time. To design a nearly linear-time algorithm for approximating edge expansion, Khandekar, Rao and Vazirani [KRV09] introduced the cut-matching game, a novel method for constructing expander graphs. As we will see in the next section, this approach enables the use of single-commodity flows to embed an expander graph into the input graph, providing a fast and combinatorial algorithm for approximating edge expansion through spectral analysis.

Definition 17.3 (Cut-Matching Game)**.** *The cut-matching game involves two players, a cut player and a matching player, who attempts to build an expander graph starting from an empty graph $G = (V, \emptyset)$. The game proceeds in iterations indexed by $t \geq 1$, and each iteration consists of the following steps:*

- *The cut player chooses a bisection $(S_t, \overline{S_t})$ of V.*
- *The matching player chooses a perfect matching M_t between S_t and $\overline{S_t}$.*

The goal of the cut player is to minimize the number of rounds needed so that the union of the matchings $\cup_t M_t$ forms an expander graph, while the goal of the matching player is to maximize the number of rounds required to achieve this.

The key question is whether there exists a cut player strategy that ensures the game finishes within a small number of rounds. The matching player is treated as an adversary to model the scenario where there is no control over the perfect matching returned in each round (e.g., while the sources and sinks of the flow problem can be specified, there is no control over which flow is returned).

The main result in this chapter is that there exists an effective and efficient cut player strategy.

Theorem 17.4 (Cut Player Strategy [KRV09])**.** *There exists a cut player strategy such that the cut-matching game always builds a graph G on n vertices with edge expansion $\varphi(G) = \Omega(1)$ in $O(\log^2 n)$ iterations. Furthermore, the cut player strategy can be implemented in $\widetilde{O}(n)$ time.*

Cut Player Strategy

A straightforward cut player strategy is to choose a minimum bisection of the current graph, i.e., a set S with $|S| = n/2$ that minimizes $|\delta(S)|$. This approach would indeed work, ensuring that the cut-matching game builds a graph with constant edge expansion in $O(\log n)$ iterations [KKOV07]. However, the drawback of this strategy is that there is no known polynomial-time implementation.

To address this issue, a natural idea is to use a fast approximation algorithm for the minimum balanced cut problem. Since linear programming-based and semidefinite programming-based approximation algorithms rely on multicommodity flows, which are not efficient enough, we are left with spectral methods. As discussed in Section 12.5, local graph partitioning algorithms can be used to design a near-linear-time approximation algorithm for the balanced cut problem. Although this approach was mentioned in [KRV09], they proposed an alternative spectral approach that is simpler, faster, and provides a better approximation algorithm for the balanced cut problem than the one in Theorem 12.12.

Their algorithm for the cut player is remarkably simple to describe and implement. Let $\mathcal{M}_t$ be the lazy random walk matrix of the perfect matching M_t in the t-th iteration, i.e., $\mathcal{M}_t(i,i) = 1/2$ for all $i \in V$ and $\mathcal{M}_t(i,j) = \mathcal{M}_t(j,i) = 1/2$ for each $ij \in M_t$.

Algorithm 14 Cut Player Strategy in the $(t+1)$-th Iteration in [KRV09]

1: Choose a random unit vector $r_{t+1} \in \mathbb{R}^n$ such that $r_{t+1} \perp \vec{1}$ and let

$$u_{t+1} = \mathcal{M}_t \mathcal{M}_{t-1} \cdots \mathcal{M}_1 r_{t+1}.$$

2: **return** $(S_{t+1}, \overline{S}_{t+1})$, where S_{t+1} is the set of $n/2$ vertices i with the smallest values of $u_{t+1}(i)$.

An alternative strategy is to compute a second eigenvector $v \in \mathbb{R}^n$ of the normalized Laplacian matrix of the current graph, and return S_{t+1} as the set of $n/2$ vertices i with the smallest values of $v(i)$. However, it is not known whether this approach works or not.

The remainder of this section is to explain and analyze Algorithm 14 and to prove Theorem 17.4.

Random Walk Embedding and Potential Function

A key question in the analysis is how to measure the progress of the cut-matching game. A natural candidate for this measure is the second smallest eigenvalue of the normalized Laplacian matrix of the current graph. However, as discussed in Section 15.3 for spectral sparsification, using an eigenvalue as the potential function is not smooth enough and does not adequately capture the global structure of the current solution. In the next chapter, we will see that a regularized version of the second smallest eigenvalue provides a more suitable potential function, reminiscent of the analysis in Section 16.3 for spectral sparsification using the discrepancy theoretic approach.

The approach in [KRV09] is quite different and interesting. They used the mixing property of the current graph $H_t := \sum_{l=1}^{t} M_l$ as a progress measure. However, instead of considering the lazy random walk matrix $W_t := \frac{1}{t}\sum_{l=1}^{t} \mathcal{M}_l$ of the current graph, they consider the "product" matrix.

Definition 17.5 (Round-Robin Walk Matrix)**.** *In the l-th step of the round-robin walk, the walk stays at the current vertex with probability $1/2$ and traverses the incident matching edge in M_l with probability $1/2$. Let t be the current iteration. The round-robin walk matrix at time t is defined as*

$$P_t = \mathcal{M}_t \mathcal{M}_{t-1} \cdots \mathcal{M}_1.$$

Let $\vec{p}_{t,i} := \{P_t(i,1), \ldots, P_t(i,n)\}$ be the i-th row of P_t, where $P_t(i,j)$ is the probability that the t-step round-robin walk started at vertex j ends up in vertex i (since we do right-multiplication).

Verify that P_t is doubly stochastic, meaning that each row sum and each column sum is equal to one.

Note that when $P_t = J/n$, where J is the all-ones matrix, the round-robin walk is completely mixed. To measure how close the current graph $H_t := \sum_{l=1}^{t} M_l$ is to an expander graph, a potential function is defined to measure how close the round-robin walk matrix P_t is to the matrix J/n.

Definition 17.6 (Potential Function)**.** *The potential function is defined as*

$$\psi_t := \psi_t(P_t) := \left\| P_t - \frac{J}{n} \right\|_F^2 = \sum_{i=1}^{n} \left\| \vec{p}_{t,i} - \frac{\vec{1}}{n} \right\|_2^2 = \sum_{i=1}^{n} \sum_{j=1}^{n} \left(P_t(i,j) - \frac{1}{n} \right)^2.$$

They showed that if the potential function ψ_t is sufficiently small, then the current graph H_t has constant edge expansion. The proof involves showing that P_t is an expander graph that can be embedded into H_t with congestion one.

Lemma 17.7 (Terminating Condition)**.** *If $\psi_t \leq 1/(4n^2)$, then the expansion of H_t is at least $1/2$.*

Proof. The assumption $\psi_t \leq 1/(4n^2)$ implies that $P_t(i,j) \geq 1/(2n)$ for every pair of vertices i and j. It follows that the weighted graph P_t, where each edge ij has a weight of $P_t(i,j) + P_t(j,i) \geq 1/n$, has edge expansion at least $1/2$.

We argue by induction that P_t can be embedded into H_t with congestion at most one. That is, for each pair $a, b \in V$, there are $P_t(a,b)$ units of flows from b to a in H_t such that each edge in H_t has congestion at most one. When a new matching M_{t+1} is added, for each edge $ij \in M_{t+1}$ and any vertex k, $P_{t+1}(i,k) = \frac{1}{2}(P_t(i,k) + P_t(j,k))$. For half of the units of flow paths connecting k to j in H_t, we extend them using the edge $ij \in M_{t+1}$ to connect k to i in H_{t+1}, Similarly, for half of the units of flow paths connecting k to i in H_t, we extend them using the edge $ij \in M_{t+1}$ to connect k to j in H_{t+1}. This construction provides an embedding of P_{t+1} into H_{t+1}.

Now, we check the congestion of the edges. The congestion of the edges in H_t remains unchanged. Since each row sum is at most one by Definition 17.5, the congestion of the edge ij in M_{t+1} is

$$\sum_k \frac{1}{2}(P_t(i,k) + P_t(j,k)) = 1.$$

Since P_{t+1} has edge expansion at least $1/2$ and can be embedded into H_{t+1} with congestion at most one, we conclude that H_{t+1} has edge expansion at least $1/2$ by Claim 17.2. □

One advantage of the potential function is that it is easy to track how it changes when a new matching is added.

Lemma 17.8 (Potential Decrease)**.** *After a new matching M_{t+1} is added, the potential function decreases by*

$$\frac{1}{2} \sum_{ij \in M_{t+1}} \|\vec{p}_{t,i} - \vec{p}_{t,j}\|_2^2.$$

Proof. After matching M_{t+1} is added, for each edge $ij \in M_{t+1}$,

$$\vec{p}_{t+1,i} = \vec{p}_{t+1,j} = \frac{1}{2}(\vec{p}_{t,i} + \vec{p}_{t,j}).$$

Using the identity $\|u\|_2^2 + \|v\|_2^2 - 2\|\frac{u+v}{2}\|_2^2 = \frac{1}{2}\|u - v\|_2^2$ with $u = \vec{p}_{t,i} - \frac{\vec{1}}{n}$ and $v = \vec{p}_{t,i} - \frac{\vec{1}}{n}$, the potential decrease contributed by the edge ij is

$$\frac{1}{2}\|\vec{p}_{t,i} - \vec{p}_{t,j}\|_2^2.$$

Summing over all edges in M_{t+1} gives the lemma. □

Random Projection

Given the random walk embedding $\vec{p}_{t,1}, \ldots, \vec{p}_{t,n}$ and the potential function ψ_t, a natural approach of the cut player is to find a bisection $(S, \overline{S})$ of the vectors, so that any perfect matching between S and $\overline{S}$ induces a large potential decrease, as described in Lemma 17.8.

The issue with this approach is that computing the vectors $\vec{p}_{t,1}, \ldots, \vec{p}_{t,n}$ takes $\Omega(n^2)$ time, making the algorithm inefficient. Even if the computational issue is ignored, it is not clear how to directly work with these high-dimensional vectors.

A common technique for handling high-dimensional vectors is to use random projections (e.g., the random hyperplane method in rounding semidefinite programming solutions). This is the approach taken in [KRV09]. The idea is to choose a random unit vector $r \in \mathbb{R}^n$ and use $u(i) = \langle \vec{p}_{t,i}, r \rangle$ to find the bisection. Note that $u = P_t r$ can be computed directly in $O(tn)$ time. This is precisely the algorithm described in Algorithm 14, with the additional requirement that $r \perp \vec{1}$.

If $|u(i) - u(j)|^2 \approx \|\vec{p}_i - \vec{p}_j\|_2^2$ for all i, j, then the cut player can focus on returning a bisection such that any perfect matching M between the two parts has large $\sum_{ij \in M} |u(i) - u(j)|^2 \approx \sum_{ij \in M} \|\vec{p}_i - \vec{p}_j\|^2$, ensuring a large potential decrease. The following lemma analyzes the projection.

Lemma 17.9 (Gaussian Behavior of Projections)**.** *Let v be an arbitrary vector in $\mathbb{R}^d$ and r a random unit vector in $\mathbb{R}^d$. Then*

- $\mathbb{E}[\langle v, r \rangle^2] = \frac{1}{d}\|v\|_2^2$.
- $\Pr[\langle v, r \rangle^2 \geq \frac{x}{d}\|v\|_2^2] \leq e^{-\frac{x}{4}}$ *for* $x \leq \frac{d}{16}$.

Since r in Algorithm 14 is a random unit vector in $\mathbb{R}^n$ perpendicular to $\vec{1}$, we apply the lemma on the space orthogonal to $\vec{1}$ to obtain that

$$\mathbb{E}\big[|u(i) - u(j)|^2\big] = \frac{1}{n-1}\|\vec{p}_i - \vec{p}_j\|_2^2 \quad \text{and} \quad \mathbb{E}\big[u(i)^2\big] = \frac{1}{n-1}\Big\|\vec{p}_i - \frac{\vec{1}}{n}\Big\|_2^2. \tag{17.3}$$

Setting $x \gtrsim \log n$, with probability at least $1 - 1/\operatorname{poly}(n)$,

$$|u(i) - u(j)|^2 \lesssim \frac{\log n}{n-1} \cdot \|\vec{p}_i - \vec{p}_j\|_2^2. \tag{17.4}$$

We now analyze the potential decrease in Lemma 17.8 for the projected random variables.

Lemma 17.10 (Projected Potential Decrease)**.** *For any perfect matching M between the bisection $(S_{t+1}, \overline{S}_{t+1})$ given in Algorithm 14,*

$$\frac{1}{2}\sum_{ij \in M} |u(i) - u(j)|^2 \geq \frac{1}{2}\sum_i u(i)^2.$$

Proof. Recall that S_{t+1} consists of the set of $n/2$ vertices with the smallest values of $u(i)$. Let η be a real number so that $u(i) \le \eta \le u(j)$ for $i \in S_{t+1}$ and $j \in \overline{S}_{t+1}$. Then,

$$\sum_{ij \in M} |u(i) - u(j)|^2 \ge \sum_{ij \in M} (|u(i) - \eta|^2 + |u(j) - \eta|^2) = \sum_{i \in V} (u(i) - \eta)^2 \ge \sum_{i \in V} u(i)^2,$$

where the last inequality uses that $\sum_i u(i) = \sum_i \langle \vec{p}_i, r \rangle = \langle \vec{1}, r \rangle = 0$, because P_t is doubly stochastic by Definition 17.5 and $r \perp \vec{1}$. □

We are ready to analyze the expected decrease of the potential function.

Lemma 17.11 (Expected Potential Decrease)**.** *Assuming* (17.4) *holds,*

$$\mathbb{E}[\psi_t - \psi_{t+1}] \gtrsim \frac{\psi_t}{\log n}.$$

Proof. By Lemma 17.10 and (17.3), for any perfect matching M between the bisection $(S_{t+1}, \overline{S}_{t+1})$,

$$\mathbb{E}\left[\sum_{ij \in M} |u_{t+1}(i) - u_{t+1}(j)|^2\right] \ge \mathbb{E}\left[\sum_{i \in V} u_{t+1}(i)^2\right] = \frac{1}{n-1} \sum_{i \in V} \left\| \vec{p}_{t,i} - \frac{\vec{1}}{n} \right\|_2^2 = \frac{\psi_t}{n-1}.$$

Assuming (17.4) holds, it follows from Lemma 17.8 that

$$2\mathbb{E}[\psi_t - \psi_{t+1}] = \mathbb{E}\left[\sum_{ij \in M_{t+1}} \|\vec{p}_{t,i} - \vec{p}_{t,j}\|_2^2\right] \gtrsim \frac{n-1}{\log n} \cdot \mathbb{E}\left[\sum_{ij \in M_{t+1}} |u_{t+1}(i) - u_{t+1}(j)|^2\right] = \frac{\psi_t}{\log n}.$$

□

Therefore, assuming (17.4) holds, the potential function decreases by a $1/\log(n)$ fraction for each new matching added. Since the initial potential value is at most $n+1$, the cut-matching game will terminate in $O(\log^2 n)$ iterations in expectation by Lemma 17.7.

A more rigorous probabilistic analysis shows that the cut-matching game terminates in $O(\log^2 n)$ iterations with high probability, establishing Theorem 17.4; see [KRV09] for details.

17.3 Algorithmic Applications

In this section, we show how to apply the cut-matching game result in Theorem 17.4 to design a fast approximation algorithm for the uniform sparsest cut problem. We also demonstrate how this result can be used to provide a bicriteria approximation algorithm for the most balanced sparse cut problem, which plays a key role in the fast expander decomposition described in Section 12.3. Finally, we discuss how the cut-matching game can be used to design approximation algorithms for the edge-disjoint paths problem.

Approximating Uniform Sparsest Cut

The cut-matching game can be used to find a primal-dual pair consisting of a sparse cut S and an expander flow, which certifies that $\varphi(S) \lesssim \log^2 n \cdot \varphi(G)$.

Theorem 17.12 (Almost-Linear Time Approximate Sparse Cut [KRV09]). *Given an undirected graph $G = (V, E)$ with n vertices and m edges, there is a randomized algorithm that, with probability at least $1 - 1/\operatorname{poly}(n)$, outputs:*

- *a cut $S \subseteq V$, and*
- *an expander H on V that can be embedded into G with congestion at most $O(\log^2 n/\varphi(S))$.*

The algorithm can be implemented in almost linear-time $\widetilde{O}(m + n^{1+o(1)})$ using $O(\log^2 n)$ max-flow computations in [CKL+22].

Given $G = (V, E)$, we run the cut-matching game in Algorithm 14 to build an expander graph H on n vertices. For each bisection $(S_t, \overline{S}_t)$ returned by the cut player in each iteration, we set up the following flow problem with a congestion parameter c.

- Add a source vertex s to G with an edge of capacity one to each vertex in S_t.
- Add a sink vertex t to G with an edge of capacity one to each vertex in $\overline{S}_t$.
- Send a flow of value $n/2$ from s to t, where the capacity of each edge $e \in E$ is set to c.

We perform a binary search on c to find a value such that the flow problem with parameter c is feasible, but with parameter $c/2$ is infeasible.

The main observation is that if the flow problem is infeasible, then a corresponding minimum cut in the constructed graph provides a cut with small edge expansion in G.

Lemma 17.13 (Sparse Cut from Minimum Cut). *Suppose the flow problem is infeasible with congestion parameter c. Then there exists a set S with $\varphi(S) \leq 1/c$.*

Moreover, if the minimum cut value is at most $n/2 - k$, then $\min\{|S|, |\overline{S}|\} \geq k$

Proof. By the max-flow min-cut theorem, there exists a cut S in the constructed graph such that $s \in S$, $t \in \overline{S}$ and $|\delta(S)| < n/2$. Let n_+ be the number of edges in $\delta(S)$ incident to s, and let n_- be the number of edges in $\delta(S)$ incident to t. Since each edge in G has capacity c, the number of edges in G in $\delta(S)$ is at most $(n/2 - n_+ - n_-)/c$. Observe that these edges in G with capacity c separate at least $n/2 - n_+$ vertices in S_t (those v in S_t where $sv \notin \delta(S)$) from at least $n/2 - n_-$ vertices in $\overline{S}_t$ (those v in $\overline{S}_t$ where $vt \notin \delta(S)$) in G. Therefore, the expansion of this cut $(S, \overline{S})$ in G satisfies

$$\min\{\varphi(S), \varphi(\overline{S})\} = \frac{|\delta(S)|}{\min\{|S|, |\overline{S}|\}} \leq \frac{(\frac{n}{2} - n_+ - n_-)/c}{\min\{\frac{n}{2} - n_+, \frac{n}{2} - n_-\}} \leq \frac{1}{c}.$$

The second statement follows from $n_+, n_- \leq n/2 - k$, and thus each side has at least k vertices. □

If the flow problem is feasible, we decompose the solution into integral flow paths. This gives us a perfect matching between S_t and $\overline{S_t}$, where there is a matching edge for the two endpoints of each path. We then provide this perfect matching to the cut-matching game to proceed to the next iteration.

We run the cut-matching game for $O(\log^2 n)$ iterations until the graph H has edge expansion at least $1/2$ by Theorem 17.4. Let c_l be the parameter in the l-th iteration such that the flow problem is feasible with parameter c_l but not with parameter $c_l/2$. Define $c_{\max} = \max_{1 \le l \le O(\log^2 n)} c_l$. By Lemma 17.13, there exists a set S with

$$\varphi(S) \le \frac{2}{c_{\max}}.$$

On the other hand, the graph H, which has edge expansion at least $1/2$, can be embedded into G with congestion at most

$$O(\log^2 n \cdot c_{\max}) = O\Big(\frac{\log^2 n}{\varphi(S)}\Big),$$

using the flow paths computed in each iteration. It follows from Claim 17.2 that

$$\varphi(G) \gtrsim \frac{\varphi(S)}{\log^2 n}.$$

This proves the approximation guarantee in Theorem 17.12.

For the running time, we first sparsify the graph to $O(n \log n)$ edges using Theorem 14.8 in $\tilde{O}(m)$ time. Then, we use the almost linear-time algorithm in [CKL+22] to solve each flow problem in $O(n^{1+o(1)})$ time. After computing each flow, we apply the random walk algorithm in [LRS13, Theorem 5] to decompose the flow into integral flow paths in $O(m \log m)$ time, providing a perfect matching for the cut-matching game to proceed. The total runtime is $\tilde{O}(m + n^{1+o(1)})$.

Approximating Most Balanced Sparse Cut

It was shown in [KRV09] how to extend Theorem 17.12 to obtain an $O(\log^2 n)$-approximation for the balanced separator problem. In this subsection, we describe the modifications in [NS17], which allow the use of an approximate max-flow min-cut algorithm (which has a near-linear-time algorithm [Pen16]) to further speedup the computation, and also to solve the approximate most balanced sparse cut problem in Section 12.3.

Approximate Max-Flow Min-Cut: Suppose we have a constant factor γ-approximation algorithm for computing max-flow in near-linear time. In each iteration of the cut-matching game, instead of computing one exact max-flow as in the previous subsection, we solve $O(\log n)$ approximate max-flow problems until we obtain a perfect matching to continue the cut-matching game.

The flow problem is similar to the one described above, except that the source vertex s is only connected to a subset of vertices $A_t \subseteq S_t$, and the sink vertex is only connected to a subset of vertices $B_t \subseteq \overline{S}_t$ with $|A_t| = |B_t|$. Initially, $A_t = S_t$ and $B_t = \overline{S}_t$.

We use the approximate max-flow algorithm to find an s-t flow in this constructed graph with congestion parameter c. If the flow value is less than $|A_t|/\gamma$, then there exists an s-t cut S with value less than $|A_t|$. The argument in Lemma 17.13 can be adapted to show that $\varphi(S) \le 1/c$. We can stop the algorithm and proceed with the binary search by increasing the congestion parameter.

If the flow value is at least $|A_t|/\gamma$, we round the flow solution into an integral flow using the random walk algorithm. This provides a matching of size at least $|A_t|/\gamma$ between A_t and B_t, where a matching edge corresponds to the two endpoints of each path. We then remove the matched vertices from A_t and B_t, and repeat the procedure. The process is repeated at most $O(\log n)$ times, as the size of A_t and B_t decreases by a constant factor each time. After $O(\log n)$ times, if the algorithm has not stopped, we obtain a perfect matching between S_t and $\overline{S}_t$, which can be embedded into G with congestion $O(c \log n)$. We then proceed with the cut-matching game as before.

To summarize, this approach provides a near-linear-time algorithm for approximating edge expansion, but with a slightly worse approximation guarantee of $O(\log^3 n)$.

Approximate Most-Balanced Sparse Cuts: To find a sparse cut of size at least k, we extend the observation in Lemma 17.13 that if the constructed graph with congestion parameter c has an s-t cut S with value at most $|A_t| - k$, then $\varphi(S) \leq 1/c$ and $\min\{|S|, |\overline{S}|\} \geq k$.

We proceed as before. If the flow value in this constructed graph is less than $(|A_t| - k)/\gamma$, then we find a cut S with $\varphi(S) \leq 1/c$ and $|S| \geq k$. We can stop the algorithm and proceed with the binary search by increasing the congestion parameter.

If the flow value is at least $(|A_t| - k)/\gamma$, then we construct a partial matching of size $(|A_t| - k)/\gamma$ as before. If the algorithm has not stopped after $O(\log n)$ iterations, then $|A_t|, |B_t| \leq k$. In this case, we just complete the current partial matching with dummy edges to a perfect matching arbitrarily, and proceed with the cut-matching game.

When the cut-matching game is completed, we obtain a graph H with edge expansion at least $1/2$ that can be embedded into G with congestion at most $O(c \cdot r \cdot \log n)$, where $r = O(\log^2 n)$ is the number of iterations in the cut-matching game. The observation is that after removing the dummy edges from H, every set S of size at least $4rk$ still has edge expansion at least $1/4$. This is simply because the total number of dummy edges is at most rk. We conclude with the following theorem.

Theorem 17.14 (Near-Linear Time Approximate Sparse Cut with Size Constraints [NS17])**.** *Given an undirected graph $G = (V, E)$ with n vertices and m edges, and a size parameter $k \lesssim n/\log^2 n$, there is a randomized algorithm that, with probability at least $1 - 1/\operatorname{poly}(n)$, outputs:*

- *a cut $S \subseteq V$ with $k \leq |S| \leq n/2$, and*
- *a graph H on V that can be embedded into G with congestion at most $O(\log^3 n/\varphi(S))$, where every set in H of size at least $\Omega(k \cdot \log^2 n)$ has edge expansion at least $1/4$.*

The algorithm can be implemented in near linear-time $\tilde{O}(m)$ using $O(\log^3 n)$ approximate max-flow computations in [Pen16].

To design a bicriteria approximation algorithm for the most balanced sparse cut problem in Section 12.3, we first run the near-linear time approximation algorithm (with no size constraint) to obtain a set S with edge expansion $\varphi \lesssim \log^3 n \cdot \varphi(G)$. Then we perform a binary search on s using Theorem 17.14 to find a value s^* so that the cut S^* returned when $s \leq s^*$ has edge expansion at most φ, while the cut returned when $s > s^*$ has edge expansion greater than φ. Then, S^* is an approximate most balanced sparse cut with $a = O(\log^3 n)$ and $b = O(\log^2 n)$ as stated in Section 12.3.

Approximating Edge-Disjoint Paths

The cut-matching game has found surprising applications in problems that are not directly related to graph expansion.

In the edge-disjoint paths problem, we are given an undirected graph $G = (V, E)$ and vertex pairs $(s_1, t_1), \ldots, (s_k, t_k)$, and the objective is to connect the maximum number of source-sink pairs using edge-disjoint paths.

On expander graphs, there are various algorithms to find many edge-disjoint paths, using random walks and probabilistic methods [BFU92, Fri01] or greedy algorithms [KR96, AC07].

On general graphs, the cut-matching game was used to embed an expander graph into the input graph, so that the existing algorithms for expanders can be applied. This approach led to breakthroughs in designing approximation algorithms for the edge-disjoint paths problem [RZ10, And10, CL16].

Deterministic Cut-Matching Game

An open question in this area is whether there is a deterministic almost linear-time cut player strategy with similar expansion guarantee as in Theorem 17.4. This would imply faster deterministic algorithms for various problems; see [CGL+20] for the applications as well as the current best known deterministic algorithm.

References

[AC07] Noga Alon and Michael Capalbo. Finding disjoint paths in expanders deterministically and online. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07)*, pages 518–524. IEEE, 2007. 223

[And10] Matthew Andrews. Approximation algorithms for the edge-disjoint paths problem via raecke decompositions. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pages 277–286. IEEE, 2010. 223

[ARV09] Sanjeev Arora, Satish Rao, and Umesh Vazirani. Expander flows, geometric embeddings and graph partitioning. *Journal of the ACM (JACM)*, 56(2):1–37, 2009. 214, 235, 236, 237, 238, 240

[BFU92] Andrei Z Broder, Alan M Frieze, and Eli Upfal. Existence and construction of edge disjoint paths on expander graphs. In *Proceedings of the twenty-fourth annual ACM symposium on Theory of Computing*, pages 140–149, 1992. 223

[CGL+20] Julia Chuzhoy, Yu Gao, Jason Li, Danupon Nanongkai, Richard Peng, and Thatchaphol Saranurak. A deterministic algorithm for balanced cut with applications to dynamic connectivity, flows, and beyond. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1158–1167. IEEE, 2020. 223

[CKL+22] Li Chen, Rasmus Kyng, Yang P Liu, Richard Peng, Maximilian Probst Gutenberg, and Sushant Sachdeva. Maximum flow and minimum-cost flow in almost-linear time. In *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 612–623. IEEE, 2022. 152, 220, 221

[CL16] Julia Chuzhoy and Shi Li. A polylogarithmic approximation algorithm for edge-disjoint paths with congestion 2. *Journal of the ACM (JACM)*, 63(5):1–51, 2016. 223

[Fri01] Alan M Frieze. Edge-disjoint paths in expander graphs. *SIAM Journal on Computing*, 30(6):1790–1801, 2001. 223

[KKOV07] Rohit Khandekar, Subhash Khot, Lorenzo Orecchia, and Nisheeth K Vishnoi. On a cut-matching game for the sparsest cut problem. *Univ. California, Berkeley, CA, USA, Tech. Rep. UCB/EECS-2007-177*, 6(7):12, 2007. 215

[KR96] Jon Kleinberg and Ronitt Rubinfeld. Short paths in expander graphs. In *Proceedings of 37th Conference on Foundations of Computer Science*, pages 86–95. IEEE, 1996. 223

[KRV09] Rohit Khandekar, Satish Rao, and Umesh Vazirani. Graph partitioning using single commodity flows. *Journal of the ACM (JACM)*, 56(4):1–15, 2009. 9, 150, 153, 213, 215, 216, 218, 219, 220, 221

[LLR95] Nathan Linial, Eran London, and Yuri Rabinovich. The geometry of graphs and some of its algorithmic applications. *Combinatorica*, 15:215–245, 1995. 214

[LR99] Tom Leighton and Satish Rao. Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms. *Journal of the ACM (JACM)*, 46(6):787–832, 1999. 213, 214

[LRS13] Yin Tat Lee, Satish Rao, and Nikhil Srivastava. A new approach to computing maximum flows using electrical flows. In *Symposium on Theory of Computing Conference, STOC'13*, pages 755–764. ACM, 2013. 221, 246, 247

[NS17] Danupon Nanongkai and Thatchaphol Saranurak. Dynamic spanning forest with worst-case update time: adaptive, las vegas, and $o(n^{1/2-\varepsilon})$-time. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1122–1129, 2017. 151, 152, 155, 221, 222

[Pen16] Richard Peng. Approximate undirected maximum flows in $o(m\,\mathrm{polylog}(n))$ time. In *Proceedings of the twenty-seventh annual ACM-SIAM symposium on Discrete algorithms*, pages 1862–1867. SIAM, 2016. 153, 221, 222

[RZ10] Satish Rao and Shuheng Zhou. Edge disjoint paths in moderately connected graphs. *SIAM Journal on Computing*, 39(5):1856–1887, 2010. 223

[Tre17] Luca Trevisan. Lecture notes on graph partitioning, expanders and spectral methods. 2017. https://people.eecs.berkeley.edu/luca/books/expanders-2016.pdf. 214, 215, 235, 236

Chapter 18

Matrix Multiplicative Weight Update

In this chapter, we introduce the matrix multiplicative weight update method, and use it to design an improved strategy for the cut-matching game.

18.1 Online Decision and Multiplicative Weight Update Method

Consider a scenario that there are n experts providing advice (e.g., in the stock market), and we need to make a series of online decisions based on their advice. Initially, we have no idea how good the experts are, but as time goes on, we observe the outcomes of their advice and learn how they have performed so far. The question is whether there exists a strategy to make these online decisions such that, in the end, we perform nearly as well as the best expert in hindsight. We formalize the model and the objective as follows.

Definition 18.1 (Online Decision Vector Model and Regret Minimization [AHK12])**.** *In each iteration $t \geq 1$, we decide on a probability distribution $p_t \in \mathbb{R}^n_+$, and then we observe the outcome vector $m_t \in \mathbb{R}^n$. The objective is to minimize the regret, defined as*

$$\sum_t \langle p_t, m_t \rangle - \min_{1 \leq i \leq n} \sum_t m_t(i).$$

In the online expert scenario, the first term $\sum_t \langle p_t, m_t \rangle$ is the cost of our decisions, while the second term $\min_i \sum_t m_t(i)$ is the cost of the best expert in hindsight. The goal is to devise an adaptive strategy $\{p_t\}_{t \geq 1}$ so that our cost is nearly as low as that of the best expert.

Multiplicative Weight Update Algorithm

The following is a simple and intuitive way to update the probability distribution. Illustrated by the online expert scenario, we maintain weight $w_t(i)$ for each expert i, representing our trust on expert i at time t. The probability distribution p_t is proportional to these weights. Initially, we have no information about which expert is good, so we set the same weight $w_t(i) = 1$ for all experts. Then, we observe the outcome $m_t \in \mathbb{R}^n$ and update the weights multiplicatively based on the outcome: the higher the cost $m_t(i)$, the larger the decrease in the weight $w_{t+1}(i)$, and vice versa. The following is the formal algorithm.

Algorithm 15 Multiplicative Weight Update Algorithm

Require: A width upper bound ρ on the maximum outcome entries, and a parameter $\epsilon \leq 1/2$.

1: Set $t = 1$ and $w_t(i) = 1$ for $1 \leq i \leq n$.
2: **for** $t \geq 1$ **do**
3: Set $p_t(i) = w_t(i) / \sum_i w_t(i)$ for $1 \leq i \leq n$.
4: Observe the outcome $m_t \in \mathbb{R}^n$ which satisfies $|m_t(i)| \leq \rho$ for $1 \leq i \leq n$.
5: Update weights:

$$w_{t+1}(i) := w_t(i) \cdot \left(1 - \epsilon \cdot \frac{m_t(i)}{\rho}\right)$$

for $1 \leq i \leq n$, and set $t \leftarrow t + 1$.
6: **end for**

The width upper bound ρ is crucial for the analysis. Informally, if some entry $m_t(i)$ is large, then it takes many iterations to recover from a mistake made.

Regret Minimization Bounds

The analysis is also natural. If our cost is high in a round, then the total weight will decrease much. So, if there is a good expert, their weight will stand out, and the algorithm will follow that expert before long. The total weight serves as a potential function in the analysis.

Interestingly, one can perform nearly as well as the best expert in hindsight.

Theorem 18.2 (Regret Minimization Bound for the Vector Setting [AHK12])**.** *Suppose $|m_t(i)| \leq \rho$ for all $1 \leq i \leq n$ and all $t \geq 1$. For $\epsilon \leq 1/2$, after T rounds,*

$$\sum_{t=1}^{T} \langle p_t, m_t \rangle \leq \sum_{t=1}^{T} m_t(i) + \epsilon \cdot \sum_{t=1}^{T} |m_t(i)| + \frac{\rho \ln n}{\epsilon} \quad \text{for all } 1 \leq i \leq n.$$

In particular, if $m_t(i) \in [0, \rho]$ for all $1 \leq i \leq n$ and all $t \geq 1$, then

$$\sum_{t=1}^{T} \langle p_t, m_t \rangle \leq (1 + \epsilon) \sum_{t=1}^{T} m_t(i) + \frac{\rho \ln n}{\epsilon} \quad \text{for all } 1 \leq i \leq n.$$

Remarkably, this theorem has numerous applications in diverse areas, including approximation algorithms, game theory, solving linear programs, learning theory, and complexity theory; see [AHK12].

We will not prove this theorem, as we will prove a matrix generalization later.

18.2 Matrix Generalization

Arora and Kale generalized the online decision model in Definition 18.1 to the matrix setting, where there is an expert for each unit vector in $\mathbb{R}^n$. The outcome in each iteration t is a matrix $M_t \in \mathbb{R}^{n \times n}$, and the cost of a vector v is $\langle vv^\top, M_t \rangle = v^\top M_t v$.

The corresponding notion of a probability distribution in the matrix setting is the set of density matrices $\Delta_n := \{X \in \mathbb{R}^{n \times n} \mid X \succcurlyeq 0 \text{ and } \operatorname{Tr}(X) = 1\}$ as discussed in Section 16.3. Any probability distribution over unit vectors of the form $\sum_i p_i v_i v_i^\top$ is a density matrix. Conversely, any density matrix can be written as $\sum_i \lambda_i u_i u_i^\top$ using an eigen-decomposition, where each u_i is a unit vector and $(\lambda_1, \ldots, \lambda_n)$ forms a probability distribution.

Definition 18.3 (Online Decision Matrix Model and Regret Minimization [AK16, Kal07]). *In each iteration $t \geq 1$, we decide on a density matrix $P_t \in \Delta^n$, and then we observe the outcome real symmetric matrix $M_t \in \mathbb{R}^n$. The objective is to minimize the regret, defined as*

$$\sum_t \langle P_t, M_t \rangle - \min_{v \in \mathbb{R}^n : \|v\|_2 = 1} \sum_t \langle vv^\top, M_t \rangle = \sum_t \langle P_t, M_t \rangle - \lambda_{\min}\Big(\sum_t M_t\Big),$$

where $\lambda_{\min}(M)$ denotes the minimum eigenvalue of the matrix M and the identity is by Lemma A.12.

Matrix Multiplicative Weight Update Algorithm

The intuition behind the algorithm for the matrix setting [AK16, Kal07] is the same as in the vector setting: the higher the cost $\langle vv^\top, M_t \rangle = v^\top M_t v$, the larger the decrease in the relative weight on v.

The update rule in Algorithm 15 can be slightly modified as $w_{t+1}(i) := w_t(i) \cdot e^{-\epsilon \cdot m_t(i)/\rho}$. This update rule can be naturally generalized using the matrix exponential defined in Section 23.2, to which we refer the reader for the basic properties of matrix exponentials.

Algorithm 16 Matrix Multiplicative Weight Update Algorithm

Require: A width upper bound ρ on the maximum operator norm, and a parameter $\epsilon \leq 1/2$.

1: Set $W_1 = I_n$ and $t = 1$.
2: **for** $t \geq 1$ **do**
3: Set $P_t = \frac{W_t}{\mathrm{Tr}(W_t)}$.
4: Observe the outcome real symmetric matrix $M_t \in \mathbb{R}^{n \times n}$ which satisfies $\|M_t\|_{\mathrm{op}} \leq \rho$.
5: Update weight matrix:

$$W_{t+1} := \exp\Big(-\frac{\epsilon}{\rho} \sum_{l=1}^{t} M_l\Big).$$

Set $t \leftarrow t + 1$.
6: **end for**

Note that in general $e^{A+B} \neq e^A \cdot e^B$, so the above update rule is different from $W_{t+1} = W_t \cdot e^{-\epsilon \cdot M_t/\rho}$, which is another natural way to define the update rule.

It may be harder to visualize, but a direction v with large $v^\top(\sum_t M_t)v$ (corresponding to an expert having high cost) has lower weight in W_{t+1}, while a direction v with small $v^\top(\sum_t M_t)v$ (corresponding to an expert having low cost) has higher weight in W_{t+1}. The density matrix P_t should be interpreted as a smart probability distribution on the unit sphere that captures the global structure of the current outcome $\sum_t M_t$.

Regret Minimization Bound

The analysis of the regret bound is syntactically the same as that of Theorem 18.2 in [AHK12]. The potential function becomes $\mathrm{Tr}(W_t)$, an analog of the total weight in the vector setting. If our cost $\langle P_t, M_t \rangle$ is high in a round, then the proof shows that the potential function $\mathrm{Tr}(W_t)$ will decrease significantly, using the Golden-Thompson inequality. On the other hand, the potential function is lower bounded by the minimum eigenvalue, such that $\mathrm{Tr}(W_t) \geq \exp(-\frac{\epsilon}{\rho}\lambda_{\min})$. Combining the upper bound and the lower bound gives the following result.

Theorem 18.4 (Regret Minimization Bound for the Matrix Setting [AK16, Kal07]). *Suppose the outcome matrices satisfy $0 \preccurlyeq M_t \preccurlyeq \rho I$ or $-\rho I \preccurlyeq M_t \preccurlyeq 0$ for all $t \geq 1$. For $\epsilon \leq 1/2$, after T rounds,*

$$(1-\epsilon)\sum_{t:M_t \succcurlyeq 0}\langle P_t, M_t\rangle + (1+\epsilon)\sum_{t:M_t \preccurlyeq 0}\langle P_t, M_t\rangle \leq \sum_{t=1}^{T}\langle vv^\top, M_t\rangle + \frac{\rho \ln n}{\epsilon} \quad \text{for all } v \in \mathbb{R}^n, \|v\|_2 = 1.$$

In particular, if $0 \preccurlyeq M_t \preccurlyeq \rho I$ for all $t \geq 1$, then

$$\sum_{t=1}^{T}\langle P_t, M_t\rangle \leq \frac{1}{1-\epsilon}\cdot \lambda_{\min}\left(\sum_{t=1}^{T} M_t\right) + \frac{\rho \ln n}{\epsilon(1-\epsilon)}.$$

This generalization also has diverse and significant applications, including spectral sparsification [ALO15], solving semidefinite programs [AK16], and proving QIP=PSPACE.

Proof

We prove the following extension of Theorem 18.4, which can be applied directly to lower bound the second smallest eigenvalue for the cut-matching game in the next section. We also relax the assumption that M_t is either positive semidefinite or negative semidefinite.

Theorem 18.5 (Regret Bound for Smallest Nonzero Eigenvalue). *Suppose $-\rho I \preccurlyeq M_t \preccurlyeq \rho I$ for all $t \geq 1$. Moreover, suppose all outcome matrices $\{M_t\}_{t\geq 1}$ share a common nullspace of dimension k, and line (3) in Algorithm 16 is modified to*

$$P_t = \frac{W_t}{\mathrm{Tr}(W_t) - k}.$$

Then, for $\epsilon \leq 1$, after T rounds of the modified algorithm,

$$\lambda_{k+1}\left(\sum_{t=1}^{T} M_t\right) \geq \sum_{t=1}^{T}\langle P_t, M_t\rangle - \epsilon\cdot\sum_{t=1}^{T}\langle P_t, |M_t|\rangle - \frac{\rho \ln n}{\epsilon},$$

where λ_{k+1} denotes the $(k+1)$-th smallest eigenvalue, and $|M|$ is as defined in Theorem 16.1.

In particular, if $0 \preccurlyeq M_t \preccurlyeq \rho I$ for all $t \geq 1$, then

$$\lambda_{k+1}\left(\sum_{t=1}^{T} M_t\right) \geq (1-\epsilon)\cdot\sum_{t=1}^{T}\langle P_t, M_t\rangle - \frac{\rho \ln n}{\epsilon},$$

Proof. Define the potential function as

$$\Phi_t := \mathrm{Tr}(W_t) - k.$$

Let $\{\lambda_i\}_{i=1}^n$ be the eigenvalues of $\sum_{t=1}^{T} M_t$. Then the eigenvalues of $W_T = \exp(-\frac{\epsilon}{\rho}\sum_{t=1}^{T} M_t)$ are $\{\exp(-\epsilon\lambda_i/\rho)\}_{i=1}^n$. Since all outcome matrices $\{M_t\}_{t\geq 1}$ share a common nullspace of dimension k, it follows that $\lambda_1 = \cdots = \lambda_k = 0$ and $\exp(-\epsilon\lambda_i/\rho) = 1$ for $1 \leq i \leq k$. Therefore,

$$\Phi_T = \mathrm{Tr}(W_T) - k = \sum_{i=k+1}^{n}\exp\left(-\frac{\epsilon}{\rho}\cdot\lambda_i\right) \geq \exp\left(-\frac{\epsilon}{\rho}\cdot\lambda_{k+1}\right). \tag{18.1}$$

For the upper bound, using the Golden-Thompson inequality in Theorem A.40,

$$\begin{aligned}
\mathrm{Tr}(W_{t+1}) &= \mathrm{Tr}\left(\exp\left(-\frac{\epsilon}{\rho}\sum_{l=1}^{t} M_l - \frac{\epsilon}{\rho}M_{t+1}\right)\right) \\
&\leq \mathrm{Tr}\left(\exp\left(-\frac{\epsilon}{\rho}\sum_{l=1}^{t} M_l\right)\cdot\exp\left(-\frac{\epsilon}{\rho}M_{t+1}\right)\right) \\
&= \mathrm{Tr}\left(W_t\cdot\exp\left(-\frac{\epsilon}{\rho}M_{t+1}\right)\right) \\
&\leq \mathrm{Tr}\left(W_t\cdot\left(I-\frac{\epsilon}{\rho}\cdot M_{t+1}+\frac{\epsilon^2}{\rho^2}\cdot M_{t+1}^2\right)\right) \\
&\leq \mathrm{Tr}\left(W_t\cdot\left(I-\frac{\epsilon}{\rho}\cdot M_{t+1}+\frac{\epsilon^2}{\rho}\cdot |M_{t+1}|\right)\right) \\
&= \mathrm{Tr}(W_t)-\frac{\epsilon}{\rho}\cdot\Big(\mathrm{Tr}(W_t\cdot M_{t+1})-\epsilon\cdot\mathrm{Tr}(W_t\cdot|M_{t+1}|)\Big),
\end{aligned}$$

where the fourth line uses $e^{-A} \preccurlyeq I - A + A^2$ for $-I \preccurlyeq A \preccurlyeq I$ (which follows from $e^{-x} \leq 1 - x + x^2$ for all $-1 \leq x \leq 1$; see Claim 23.3) and $\mathrm{Tr}(WX) \leq \mathrm{Tr}(WY)$ for $W \succcurlyeq 0$ and $Y \succcurlyeq X \succcurlyeq 0$, and the fifth line follows similarly as $M_{t+1}^2 \preccurlyeq \rho \cdot |M_{t+1}|$ (by our assumption $-\rho I \preccurlyeq M_{t+1} \preccurlyeq \rho I$).

Using $\Phi_t = \mathrm{Tr}(W_t) - k$ and $P_t = W_t/(\mathrm{Tr}(W_t) - k)$, the potential decrease satisfies

$$\begin{aligned}
\Phi_{t+1} &\leq \Phi_t\cdot\left(1-\frac{\epsilon}{\rho}\cdot\left(\frac{\mathrm{Tr}(W_t\cdot M_{t+1})}{\Phi_t}-\epsilon\cdot\frac{\mathrm{Tr}(W_t\cdot|M_{t+1}|)}{\Phi_t}\right)\right) \\
&= \Phi_t\cdot\Big(1-\frac{\epsilon}{\rho}\cdot\Big(\mathrm{Tr}(P_t\cdot M_{t+1})-\epsilon\cdot\mathrm{Tr}(P_t\cdot|M_{t+1}|)\Big)\Big).
\end{aligned}$$

Since the initial potential function is $\Phi_1 = \mathrm{Tr}(I) - k = n - k$, it follows by induction that

$$\begin{aligned}
\Phi_T &\leq (n-k)\cdot\prod_{t=1}^{T}\Big(1-\frac{\epsilon}{\rho}\cdot\Big(\langle P_t, M_{t+1}\rangle-\epsilon\cdot\langle P_t,|M_{t+1}|\rangle\Big)\Big) \\
&\leq n\cdot\exp\left(-\frac{\epsilon}{\rho}\cdot\sum_{t=1}^{T}\Big(\langle P_t, M_{t+1}\rangle-\epsilon\cdot\langle P_t,|M_{t+1}|\rangle\Big)\right).
\end{aligned}$$

Combining with the lower bound in (18.1), taking logarithms, and rearranging completes the proof. □

18.3 Cut Player Strategy from Matrix Multiplicative Update

Using the regret minimization framework, we can devise an improved cut player strategy for the cut-matching game in Section 17.2 in a rather straightforward manner.

Plan

Recall that in the cut-matching game, the goal of the cut player is to produce bisection $(S_t, \overline{S_t})$ in each iteration t, such that for any perfect matching M_t between S_t and $\overline{S_t}$, the graph $H_T = \cup_{t=1}^{T} M_t$ has large edge expansion.

Our plan is to lower bound the edge expansion by the second eigenvalue of the Laplacian matrix, and then use the regret minimization result in Theorem 18.5 to lower bound the second eigenvalue as follows:

$$2\varphi(H_T) \geq \lambda_2(L(H_T)) = \lambda_2\left(\sum_{t=1}^{T} L(M_t)\right) \geq (1-\epsilon)\cdot\sum_{t=1}^{T}\langle P_t, L(M_t)\rangle - \frac{\rho \ln n}{\epsilon}, \tag{18.2}$$

where the first inequality is from Exercise 18.8.

The regret minimization approach reduces the task of the cut player to finding a bisection $(S_t, \overline{S_t})$ such that any perfect matching M_t connecting S_t to $\overline{S_t}$ has large $\langle P_t, L(M_t)\rangle$. Since $P_t \succcurlyeq 0$ as it is a matrix exponential, we can write $P_t = V^\top V$ so that $P_t(i,j) = \langle v_i, v_j\rangle$ for all i, j (see Fact A.9). Now, observe that

$$\langle P_t, L(M_t)\rangle = \left\langle V^\top V, \sum_{ij\in M_t} L_{ij}\right\rangle = \sum_{ij\in M_t}\langle V^\top V, L_{ij}\rangle = \sum_{ij\in M_t}\|v_i - v_j\|_2^2, \tag{18.3}$$

where L_{ij} denotes the Laplacian matrix of a single edge ij.

Therefore, the task of the cut player is to find a bisection such that any perfect matching M_t between the bisection has large $\sum_{ij\in M_t}\|v_i - v_j\|_2^2$. This is exactly what was done in Section 17.2; see the potential decrease in Lemma 17.8. Thus, we can follow the same approach in Section 17.2 to find a good bisection.

Cut Player Strategy

The algorithm is quite similar to that in Algorithm 14, but we replace the round-robin walk matrix with the matrix exponential $P_t^{\frac{1}{2}}$ from Algorithm 16. This specific choice of decomposing $P_t = P_t^{\frac{1}{2}}\cdot P_t^{\frac{1}{2}}$ is crucial for the analysis, as it is symmetric, and for efficient computation as well.

Algorithm 17 Cut Player Strategy from Regret Minimization

1: Let $M_1, \ldots, M_t$ be the perfect matchings returned by the matching player so far, and let $L(M_l)$ be the Laplacian matrix of the matching M_l. Define

$$W_{t+1} := \exp\left(-\frac{\epsilon}{\rho}\sum_{l=1}^{t} L(M_l)\right) \quad \text{and} \quad P_{t+1} := \frac{W_{t+1}}{\mathrm{Tr}(W_{t+1}) - 1},$$

2: Choose a random unit vector $r_{t+1} \in \mathbb{R}^n$ such that $r_{t+1} \perp \vec{1}$ and compute

$$u_{t+1} = P_{t+1}^{\frac{1}{2}}\cdot r_{t+1}.$$

3: **return** $(S_{t+1}, \overline{S}_{t+1})$, where S_{t+1} is the set of $n/2$ vertices i with the smallest values of $u_{t+1}(i)$.

This strategy improves the expansion guarantee of Theorem 17.4 since the potential function provided by the regret minimization approach can establish a stronger lower bound.

Theorem 18.6 (Improved Cut Player Strategy [OSVV08, LTW24]). *The cut player strategy in Algorithm 17 always constructs a graph G on n vertices with edge expansion $\varphi(G) = \Omega(\log n)$ in $O(\log^2 n)$ iterations. Furthermore, the cut player strategy can be implemented in $\tilde{O}(n)$ time.*

Applying Theorem 18.6 as in Section 17.3 gives an almost linear-time $O(\log n)$-approximation algorithm and a near linear-time $O(\log^2 n)$-approximation algorithm for the uniform sparsest cut problem, providing improved algorithms for balanced separators and expander decomposition.

Analysis

We can reuse some calculations in the analysis in Section 17.2, but with differences in the properties of the round-robin walk matrix and the matrix exponential $P_t^{\frac{1}{2}}$ from Algorithm 17.

Since $P_t^{\frac{1}{2}}$ is symmetric, let $\vec{p}_i$ be the i-th row and the i-th column of $P_t^{\frac{1}{2}}$. As stated in (18.3), our goal is to lower bound

$$\langle P_t, L(M_t)\rangle = \sum_{ij\in M_t} \|\vec{p}_i - \vec{p}_j\|_2^2.$$

As in Section 17.2, we relate $\|\vec{p}_i - \vec{p}_j\|_2^2$ to the distance in the one-dimensional projection $|u(i)-u(j)|^2$, where $u(i) = \langle \vec{p}_i, r\rangle$ by Algorithm 17. Since all Laplacian matrices share a common nullspace $\vec{1}$, the vector $\vec{1}$ is an eigenvector with eigenvalue $e^0 = 1$ of the matrix exponentials, such that

$$W_t\vec{1} = \vec{1}, \quad \text{and} \quad W_t^{\frac{1}{2}}\vec{1} = \vec{1}. \tag{18.4}$$

Since $P_t^{\frac{1}{2}}$ is a scaled version of $W_t^{\frac{1}{2}}$, it follows that each row $\vec{p}_i$ has the same sum, and thus $\vec{p}_i - \vec{p}_j$ is perpendicular to $\vec{1}$ for all i, j. Since r is a random unit vector in $\mathbb{R}^n$ perpendicular to $\vec{1}$, we use Lemma 17.9 as in (17.3) to obtain that

$$\mathbb{E}\big[|u(i) - u(j)|^2\big] = \frac{1}{n-1}\|\vec{p}_i - \vec{p}_j\|_2^2. \tag{18.5}$$

Similarly, applying Lemma 17.9 as in (17.4), with probability at least $1 - 1/\operatorname{poly}(n)$,

$$\|\vec{p}_i - \vec{p}_j\|_2^2 \gtrsim \frac{n-1}{\log n} \cdot |u(i) - u(j)|^2. \tag{18.6}$$

Using the property that $\vec{p}_i$ is also the i-th column of $P_t^{\frac{1}{2}}$ and (18.4) that $\vec{1}$ is an eigenvector of $P_t^{\frac{1}{2}}$,

$$\sum_i u(i) = \sum_i \langle \vec{p}_i, r\rangle = \Big\langle \sum_i \vec{p}_i, r\Big\rangle = \Big\langle P_t^{\frac{1}{2}} \cdot \vec{1}, r\Big\rangle = 0.$$

Thus, the same proof in Lemma 17.10 gives the same bound

$$\sum_{ij\in M_t} |u(i) - u(j)|^2 \geq \sum_i u(i)^2. \tag{18.7}$$

Putting together, assuming (18.6) holds and using (18.7) and (18.5),

$$\sum_{ij\in M_t} \mathbb{E}\big[\|\vec{p}_i - \vec{p}_j\|_2^2\big] \gtrsim \frac{n-1}{\log n} \cdot \sum_{ij\in M_t} \mathbb{E}\big[|u(i) - u(j)|^2\big] \geq \frac{n-1}{\log n} \cdot \sum_{i\in V} \mathbb{E}\big[u(i)^2\big] = \frac{1}{\log n}, \tag{18.8}$$

where the equality is by Exercise 18.9 that uses the property of $P_t^{\frac{1}{2}}$. Therefore, by (18.2) and (18.3),

$$2\mathbb{E}\left[\varphi(H_T)\right] \geq \mathbb{E}\left[\lambda_2\Big(\sum_{t=1}^T L(M_t)\Big)\right] \geq (1-\epsilon) \cdot \sum_{t=1}^T \sum_{ij\in M_t} \mathbb{E}\big[\|\vec{p}_i - \vec{p}_j\|_2^2\big] - \frac{\rho \ln n}{\epsilon} \gtrsim \frac{(1-\epsilon)T}{\log n} - \frac{\rho \ln n}{\epsilon}.$$

Noting that $\rho = \lambda_{\max}(L(M_t)) \leq 2$ as M_t is a matching and choosing $\epsilon = 1/2$, it follows that the expected edge expansion of H_T is $\Theta(\log n)$ when $T = \Theta(\log^2 n)$.

A standard concentration argument can then be used to establish the expansion guarantee in Theorem 18.6; see [LTW24].

Fast Implementation

For the cut player strategy, in each iteration, we need to compute the vector

$$P_t^{\frac{1}{2}} \cdot r_t = \exp\left(-\frac{\epsilon}{2\rho} \sum_{l=1}^{t} L(M_l) \right) \cdot r_t.$$

To achieve a fast implementation, we approximate the matrix exponential using the first few terms of its Taylor expansion. The error can be bounded using the following lemma.

Lemma 18.7 (Approximating Matrix Exponential by Taylor Expansion [Kal07, Lemma 23]). *Given a symmetric matrix L and a unit vector r, for $k \geq \max\{e^2 \cdot \|L\|_{\text{op}}, \ln \frac{1}{\epsilon}\}$,*

$$\left\| e^L \cdot r - \sum_{j=1}^{k} \frac{L^j}{j!} \cdot r \right\|_2 \leq \left\| e^L \right\|_{\text{op}} \cdot \epsilon$$

For our application, taking $k = O(\log^2 n)$, the vector u_t returned by the Taylor approximation satisfies

$$\left\| P_t^{\frac{1}{2}} \cdot r_t - u_t \right\| \leq \frac{1}{\text{poly}(n)}.$$

This approximation can be computed in $O(km) = \tilde{O}(m)$ time, where m is the number of edges in the current graph H_t. This completes the proof of Theorem 18.6.

18.4 Problems

Exercise 18.8 (Edge Expansion and Second Eigenvalue). *Show that $\varphi(G) \geq \frac{1}{2}\lambda_2(L(G))$.*

Hint: Use the same argument as in the easy direction of Cheeger's inequality in Section 2.2.

Exercise 18.9 (Expected Norm). *Prove that $\sum_{i \in V} \mathbb{E}\big[u(i)^2\big] = \frac{1}{n-1}$.*

Idea: If $r \in \mathbb{R}^n$ is a random unit vector (without the requirement that $r \perp 1$) and $P_t := W_t / \operatorname{Tr}(W_t)$ (without the nullspace modification), then $\mathbb{E}\big[u(i)^2\big] = \frac{1}{n}\|\vec{p}_i\|_2^2$ and $\sum_{i \in V} \mathbb{E}\big[u(i)^2\big] = \frac{1}{n}\operatorname{Tr}(P_t) = \frac{1}{n}$.

Problem 18.10 (Cut-Matching Game for Directed Graphs [LTW24]). *Generalize Theorem 18.6 to obtain a similar result for directed graphs.*

Problem 18.11 (Reweighted Eigenvalues by Matrix Multiplicative Weight Update [LTW24]). *Use Theorem 18.4 to design a $(1-\epsilon)$-approximation algorithm for computing the reweighted eigenvalue λ_2^* in Definition 10.7 in $O(\log n/(\epsilon^2 \lambda_2^*))$ iterations.*

Combine this with an almost linear time $O(\log n)$-approximation for directed edge conductance to compute a set S with $\vec{\phi}(S) \lesssim \sqrt{\vec{\phi}(G) \cdot \log \frac{1}{\vec{\phi}(G)}}$ as stated in Theorem 10.8 in almost linear time.

References

[AHK12] Sanjeev Arora, Elad Hazan, and Satyen Kale. The multiplicative weights update method: a meta-algorithm and applications. *Theory of computing*, 8(1):121–164, 2012. 225, 226, 227

[AK16] Sanjeev Arora and Satyen Kale. A combinatorial, primal-dual approach to semidefinite programs. *Journal of the ACM (JACM)*, 63(2):1–35, 2016. 9, 227, 228, 235, 240, 241, 242

[ALO15] Zeyuan Allen-Zhu, Zhenyu Liao, and Lorenzo Orecchia. Spectral sparsification and regret minimization beyond matrix multiplicative updates. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC*, pages 237–245. ACM, 2015. 8, 196, 203, 228

[Kal07] Satyen Kale. *Efficient algorithms using the multiplicative weights update method*. Princeton University, 2007. 227, 228, 232, 236, 237, 241, 247

[LTW24] Lap Chi Lau, Kam Chuen Tung, and Robert Wang. Fast algorithms for directed graph partitioning using flows and reweighted eigenvalues. In *Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 591–624. SIAM, 2024. 131, 230, 232, 241, 246, 250

[OSVV08] Lorenzo Orecchia, Leonard J Schulman, Umesh V Vazirani, and Nisheeth K Vishnoi. On partitioning graphs via single commodity flows. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 461–470, 2008. 230

Chapter 19

Fast $\sqrt{\log n}$-Approximation Algorithm for Edge Expansion

In previous chapters, we have studied two fast algorithms achieving $O(\log^2 n)$-approximation and $O(\log n)$-approximation for edge expansion, where n is the number of vertices in the input graph. In this chapter, we outline Sherman's [She09] almost linear-time $O(\sqrt{\log n})$-approximation algorithm for edge expansion, building on the max-flow min-cut iterations in Chapter 17, the matrix multiplicative weight update method in Chapter 18, and the deep geometric results developed by Arora–Rao–Vazirani [ARV09].

In Section 19.1, we first review the semidefinite program and the geometric results behind the proofs of [ARV09, She09]. We then derive the dual semidefinite program and explain its connection to expander flows in Section 19.2. Next, in Section 19.3, we present the primal-dual approach based on the matrix multiplicative weight update algorithm of Arora and Kale [AK16]. Finally, we describe the detailed implementation of the Oracle for the matrix multiplicative weight update method in Section 19.4, based on Sherman's chaining argument.

19.1 Primal Program and Geometric Analysis

To understand Sherman's result, we first review the semidefinite-programming-based $O(\sqrt{\log n})$-approximation algorithm for edge expansion in [ARV09]. We do not provide proofs of this result, as they are quite technically demanding and not directly related to the main theme of this book, and because there are already multiple excellent expositions [Tre17, Rot16, BS16] of this important work in the literature.

Primal Semidefinite Program with Triangle Inequalities

Given an undirected graph $G = (V, E)$, consider the following semidefinite programming relaxation for the edge expansion problem.

$$\begin{aligned} \min_{v_1,\ldots,v_n \in \mathbb{R}^n} \quad & \sum_{ij \in E} \|v_i - v_j\|_2^2 \\ \text{subject to} \quad & \Big\|\sum_{i \in V} v_i\Big\|_2^2 = 0 \\ & \sum_{i \in V} \|v_i\|_2^2 = 1 \\ & \|v_i - v_j\|_2^2 + \|v_j - v_k\|_2^2 \geq \|v_i - v_k\|_2^2 \qquad \forall i, j, k \in V. \end{aligned} \tag{19.1}$$

Using the same argument as in the easy direction of Cheeger's inequality, one can verify that the objective value of this program is at most $2\varphi(G)$ (see Exercise 18.8), and hence this program is indeed a relaxation.

Note that when G is regular, this program is a strengthening of the semidefinite program for the second eigenvalue in Lemma 9.5 obtained by adding the triangle inequalities.

Geometric Structural Theorem

A major contribution in [ARV09] is a structural theorem on vectors satisfying the ℓ_2^2 triangle inequalities in (19.1). It asserts that, given a well-spread set of vectors satisfying the ℓ_2^2 triangle inequalities, there exist two large subsets of vectors L and R such that every vector in L is far away from every vector in R.

Definition 19.1 (Well-Spread Case and Large-Core Case). *Let $u_1, \ldots, u_n$ be a set of vectors that satisfy $\sum_{i,j \in [n]} \|u_i - u_j\|_2^2 = n^2$. We say these vectors have a large core if there exists $i \in [n]$ such that*

$$\left|\left\{ j \in [n] \;\middle|\; \|u_i - u_j\|_2^2 \leq \frac{1}{40} \right\}\right| > \frac{n}{4}.$$

Otherwise, we say these vectors are well-spread.

Theorem 19.2 (ℓ_2^2 Structural Theorem [ARV09]). [1] *Let $u_1, \ldots, u_n$ be a set of vectors that satisfy $\sum_{i,j \in [n]} \|u_i - u_j\|_2^2 = n^2$. If these vectors are well-spread and satisfy the ℓ_2^2 triangle inequalities in (19.1), then there is a randomized polynomial-time algorithm that, with constant probability, finds two sets $L, R \subseteq [n]$ such that*

$$|L|, |R| \gtrsim n \quad \text{and} \quad \min_{i \in L, j \in R} \|u_i - u_j\|_2^2 \gtrsim \frac{1}{\sqrt{\log n}}.$$

In the well-spread case, using this structural theorem, one can apply a threshold rounding algorithm (similar to that for the hard direction of Cheeger's inequality in Lemma 2.5) to obtain a set S with $|S| \leq n/2$ and $\varphi(S)$ at most $O(\sqrt{\log n})$ times the objective value of the semidefinite program; see Problem 19.17.

[1] We remark that the definition of well-spread vectors and the formulation of the structural theorem are slightly different across the literature, but they are all essentially equivalent; our formulation follows from [Kal07, Lemma 5] and [Tre17, Lemma 13.1].

In the large-core case, one can directly apply a threshold rounding algorithm to obtain a set S with $\varphi(S)$ at most $O(1)$ times the objective value of the semidefinite program; see Problem 19.18.

Henceforth, we focus on finding the two large and well-separated sets L and R in Theorem 19.2. Note that this is a purely geometric problem, with no graph structure involved anymore.

Randomized Algorithm for Finding Well-Separated Sets

The random projection algorithm used in the cut-player strategies in Algorithm 14 and Algorithm 17 was originally introduced in [ARV09] to find well-separated sets. For simplicity, we assume that each $\|u_i\|_2^2$ is both upper and lower bounded by a constant, which can be achieved by selecting a constant fraction of vectors (e.g., using the arguments in [Kal07, Lemma 5] and [Rot16]).

Algorithm 18 Random Projection Algorithm for Well-Separated Sets

Require: Vectors $u_1, \dots, u_n$ satisfying (i) $\sum_{i,j\in[n]} \|u_i - u_j\|_2^2 = n^2$, (ii) $\|u_i\|_2^2 \asymp 1$ for each $i \in [n]$, (iii) the well-spread property in Definition 19.1, and (iv) the triangle inequalities in (19.1).

1: Choose a random Gaussian vector $g \sim N(0, I_n)$.
2: Set $L := \{i \in [n] \mid \langle u_i, g\rangle \le -1\}$ and $R := \{j \in [n] \mid \langle u_j, g\rangle \ge 1\}$.
3: **while** there exist $i \in L$ and $j \in R$ such that $\|u_i - u_j\|_2^2 \le \frac{c}{\sqrt{\log n}}$ for some small constant c **do**
4: Remove i from L and j from R.
5: **end while**
6: **return** L and R.

The algorithm is simple and is successful if it returns $|L|, |R| \gtrsim n$. The difficulty lies in lower bounding the success probability of the algorithm.

Geometric Analysis via Chaining

There are two steps in which the algorithm can fail. One is in Step (2), when $|L|$ or $|R|$ may already be too small. The well-spread property, along with some standard Gaussian analysis, is used to ensure that $|L|, |R| \gtrsim n$ with constant probability (see, e.g., [Rot16, Lemma 5]). We note that this is the only place in the proof where the well-spread property is used.

The tricky step is in the while loop, when we may remove too many vectors from L and R. Standard Gaussian analysis (see Lemma 17.9) implies that it is very unlikely that there exist $i \in L$ and $j \in R$ with $\|u_i - u_j\|_2^2 \lesssim 1/\log n$. Hence, if we only aim for an $O(\log n)$-approximation, then with high probability no pairs need to be removed. However, it is likely that there exist many $i \in L$ and $j \in R$ with $\|u_i - u_j\|_2^2 \lesssim 1/\sqrt{\log n}$, so this simple analysis does not suffice. To rule out that too many pairs $i \in L$ and $j \in R$ are removed in the while loop, with high probability over the random direction g, we will need to carefully exploit the property that the vectors $u_1, \dots, u_n$ satisfy the triangle inequalities.

First, we formulate the structure that arises from failure of the while loop with high probability, which will also be used later in this chapter. For a fixed failed direction g, this implies that there is a large directed matching $M(g)$ of vectors, where each ordered pair $(i, j) \in M(g)$ has small distance such that $\|u_i - u_j\|_2^2 \lesssim 1/\sqrt{\log n}$ but large projection on g such that $\langle u_i - u_j, g\rangle \ge 2$. If a constant fraction of random directions are failed directions, then we obtain an $(\Omega(1), \Omega(1))$-matching-cover, which is defined as follows.

Definition 19.3 (Matching Cover). *A (σ, δ)-matching-cover for vectors $u_1, \ldots, u_n$ is a function M assigning a directed matching $M(g)$ to each direction g satisfying the following conditions:*

- Stretch*: for all $(i,j) \in M(g)$, $\langle u_i - u_j, g\rangle \geq \sigma$;*
- Skew-symmetry*: $(i,j) \in M(g)$ if and only if $(j,i) \in M(-g)$;*
- Largeness*: $\mathbb{E}_g[|M(g)|] \geq \delta n$ when $g \sim N(0, I)$.*

Furthermore, each edge (i,j) in these matchings satisfies $\|u_i - u_j\|_2^2 \leq c/\sqrt{\log n}$, as described in Step (3) of the random projection algorithm. This property will be used later in deriving a contradiction to the triangle inequalities, but it is not needed for the following key "chaining" step introduced in [ARV09], which is why it does not appear in the definition of a matching-cover above.

Definition 19.4 (Chained Matching). *Let M be a matching-cover for vectors $u_1, \ldots, u_n$. For directions $g_1, \ldots, g_k$, let $M(g_1, \ldots, g_k)$ be the graph that contains a directed edge (i,j) if and only if there exist $i_0, \ldots, i_k$ with $i_0 = i$, $i_k = j$, and $(i_{l-1}, i_l) \in M(g_l)$ for all $l \leq k$. Note that $M(g_1, \ldots, g_k)$ is not necessarily a matching, but rather a graph with maximum indegree and outdegree one.*

The following main theorem for the geometric analysis is proved by an intricate inductive argument using the measure concentration phenomenon in high-dimensional geometry.

Theorem 19.5 (Chaining Far-Away Pairs [ARV09, Lee05]). *For any $(\Omega(1), \Omega(1))$-matching-cover M for vectors $u_1, \ldots, u_n$, there exist $i, j \in [n]$ and $g_1, \ldots, g_k$ with $k \lesssim \sqrt{\log n}$ such that $(i,j) \in M(g_1, \ldots, g_k)$ and $\|u_i - u_j\|_2^2 \gtrsim 1$.*

This leads to the desired contradiction if we choose c to be a sufficiently small constant, as the ℓ_2^2 triangle inequalities imply that the pair (i,j) connected by the chained matching in Theorem 19.5 should have length

$$\|u_i - u_j\|_2^2 \leq \sum_{l=1}^{k} \|u_{i_{l-1}} - u_{i_l}\|_2^2 \lesssim \frac{k \cdot c}{\sqrt{\log n}} \lesssim c.$$

Hence, we conclude that the random projection algorithm must succeed with constant probability.

Sherman's Algorithmic Chaining Proof

The proof of Theorem 19.5 does not provide an algorithm for finding the directions $g_1, \ldots, g_k$, nor does it provide a lower bound on the number of edges $(i,j) \in M(g_1, \ldots, g_k)$ with $\|u_i - u_j\|_2^2 \gtrsim 1$. Sherman proved the following stronger, algorithmic version of Theorem 19.5.

Theorem 19.6 (Chaining Far-Away Pairs, Algorithmic Version [She09]). *For any $(\Omega(1), \Omega(1))$-matching-cover M for vectors $u_1, \ldots, u_n$ and for any sufficiently small constant ℓ and $k \lesssim \sqrt{\ell \log n}$, there is an efficiently samplable distribution[2] $\mathcal{D}$ over vectors $(g_1, \ldots, g_k)$ such that the expected number of edges (i,j) with $(i,j) \in M(g_1, \ldots, g_k)$ and $\|u_i - u_j\|_2^2 \geq \ell$ is at least $e^{-O(k^2)} \cdot n$ when $(g_1, \ldots, g_k)$ is sampled from $\mathcal{D}$.*

[2] The explicit distribution is over a shuffling of independent Gaussian vectors and correlated Gaussian vectors: Let N_ρ^k be a distribution over vectors $g_1, \ldots, g_k$ such that each g_i has distribution $N(0, I_n)$, and g_{l+1} is ρ-correlated with g_l such that $g_{l+1}(j)$ and $g_l(j)$ have covariance ρ for each entry j. The distribution $\mathcal{D}$ in [She09] is roughly a shuffling of $N_{1-\frac{1}{k}}^k$ and N_0^k.

We will use Theorem 19.6 as a black box and derive an almost-linear-time $O(\sqrt{\log n})$-approximation algorithm for the edge expansion problem.

19.2 Dual Semidefinite Program and Expander Flows

In this section, we construct the dual semidefinite program of (19.1) and explain its connection to the expander flow formulation that we discussed in Section 17.1.

Primal Semidefinite Program with Path Constraints

To see this connection, we consider the following primal semidefinite program, which is equivalent to (19.1) and has one constraint for each path.

$$\begin{aligned} \min_{v_1,\dots,v_n\in\mathbb{R}^n} \quad & \sum_{ij\in E} \|v_i - v_j\|_2^2 \\ \text{subject to} \quad & \Big\|\sum_{i\in V} v_i\Big\|_2^2 = 0 \\ & \sum_{i\in V} \|v_i\|_2^2 = 1 \\ & \sum_{j=1}^{k-1} \|v_{i_j} - v_{i_{j+1}}\|_2^2 \geq \|v_{i_1} - v_{i_k}\|_2^2 \qquad \forall p = (i_1,\dots,i_k) \in \mathcal{P}(K_n), \end{aligned} \tag{19.2}$$

where $\mathcal{P}(K_n)$ denotes the set of paths in the complete graph K_n on the vertex set $[n]$. Note that the set of paths is not restricted to the set of paths in the input graph G.

We now rewrite this program into standard form to construct the dual program. Let V be the $n \times n$ matrix with the i-th column being v_i for $i \in [n]$, and let $X = V^\top V$. Let $L_{i,j}$ be the Laplacian of the edge ij, where ij need not be an edge in G. For a path $p = (i_1,\dots,i_k) \in \mathcal{P}(K_n)$, let

$$T_p := \sum_{j=1}^{k-1} L_{i_j,i_{j+1}} - L_{i_1,i_k}.$$

Note that $\langle L_{i,j}, X\rangle = X_{ii} + X_{jj} - X_{ij} - X_{ji} = \|v_i\|_2^2 + \|v_j\|_2^2 - \langle v_i, v_j\rangle - \langle v_j, v_i\rangle = \|v_i - v_j\|_2^2$. Therefore, the primal semidefinite program in (19.2) can be written as

$$\begin{aligned} \min_{X\succcurlyeq 0} \quad & \langle L(G), X\rangle \\ \text{subject to} \quad & \langle 1_n 1_n^\top, X\rangle = 0 \\ & \langle I, X\rangle = 1 \\ & \langle T_p, X\rangle \geq 0 \quad \forall p \in \mathcal{P}(K_n). \end{aligned}$$

Dual Semidefinite Program

To construct the dual program, associate a dual variable f_p for each path constraint, a dual variable λ for $\langle I, X\rangle = 1$, and a dual variable x for $\langle 1_n 1_n^\top, X\rangle = 0$. One can check that strong duality holds,

and the dual program is

$$\max_{\lambda,\ x,\ f_p \geq 0\ :\ p \in \mathcal{P}(K_n)} \quad \lambda$$
$$\text{subject to} \quad \lambda I + x 1_n 1_n^\top \preccurlyeq L(G) - \sum_p f_p T_p.$$

Note that the all-one vector 1_n is in the null space of the right-hand side. Therefore, for the dual constraint to hold, an optimal dual solution must set $x = -\lambda/n$, so that the left-hand side vanishes in the direction of 1_n as well. Thus, the maximum feasible λ is exactly the second eigenvalue of $L(G) - \sum_p f_p T_p$, and the dual program of (19.2) can be written succinctly as

$$\max_{f_p \geq 0\ :\ p \in \mathcal{P}(K_n)} \lambda_2\Big(L(G) - \sum_p f_p T_p\Big). \tag{19.3}$$

Expander Flows as Dual Solutions

To understand the dual program, we now explain how expander flows provide feasible dual solutions to (19.3). The following is a more formal definition of the concept of graph embedding in Claim 17.2.

Definition 19.7 (Flow Embedding)**.** *Let G and H be two edge-weighted graphs with the same vertex set V. We say that H has a flow embedding into G if there exists a collection of flow paths $\mathcal{P}$ and nonnegative scalars f_p for each path $p \in \mathcal{P}$ such that for each pair of vertices (i, j):*

1. *$\sum_{p \in \mathcal{P}_{ij}} f_p = w_H(ij)$ where $\mathcal{P}_{ij}$ consists of paths in $\mathcal{P}$ with endpoints i, j, and*
2. *$\sum_{p : ij \in p} f_p \leq w_G(ij)$.*

Given a flow embedding of H into G, we use $(f_p)_{p \in \mathcal{P}}$ as a dual solution to (19.3). Let F be the flow graph with the same vertex set as G and H, with edge weight $w_F(ij) = \sum_{p : ij \in p} f_p$ for each pair of vertices (i, j). For a path $p \in \mathcal{P}_{ij}$, note that $T_p = L_p - L_{i,j}$, where L_p denotes the Laplacian of the path p. Using property (1) in Definition 19.7, one can verify that

$$\lambda_2\Big(L(G) - \sum_{p \in \mathcal{P}} f_p T_p\Big) = \lambda_2\Big(L(G) - \sum_{i,j} \sum_{p \in \mathcal{P}_{ij}} f_p L_p + \sum_{i,j} w_H(ij) \cdot L_{i,j}\Big) = \lambda_2\big(L(G) - L(F) + L(H)\big),$$

where $L(F)$ and $L(H)$ denote the Laplacians of the edge-weighted graphs F and H respectively. Since $w_F(ij) \leq w_G(ij)$ by property (2) in Definition 19.7, it follows that $L(G) \succcurlyeq L(F)$. Therefore, the following formulation provides a lower bound on (19.2):

$$\max_H \ \lambda_2\big(L(H)\big) \tag{19.4}$$
$$\text{subject to } H \text{ has a flow embedding into } G.$$

This is the expander flow formulation used in [ARV09, AK16, She09], where it is proved that the objective values of (19.4) and (19.2) are within a factor of $O(\sqrt{\log n})$ of each other.

General Dual Program

The above dual formulation is a restricted one, where only the dual variables corresponding to paths arising from a flow embedding are used. The more general formulation is as follows:

$$\max_{H,\ y_p \geq 0 : p \in \mathcal{P}(K_n)} \lambda_2\Big(L(H) - \sum_p y_p T_p\Big) \tag{19.5}$$
$$\text{subject to } H \text{ has a flow embedding into } G.$$

This formulation can be derived using the same argument as above, and we use the variables y_p to distinguish them from the flow-paths weights f_p of H. We will use this more general dual program to derive a fast approximation algorithm for edge expansion, as it leads to a simpler algorithm in [LTW24].

Intuition: We provide intuition about the term $-\sum_p y_p T_p$ in the objective function and how it will be used to simplify Sherman's algorithm. We may interpret each $-T_p$ as a "shortcut cycle" C_p, where the edges in p have weight -1 and the edge connecting the two endpoints has weight 1. Since a shortcut cycle has only one positive edge, any cut across the cycle has non-positive weight. Thus, adding a shortcut cycle to a graph does not increase edge expansion.

A more combinatorial way to understand why the dual program is a lower bound on the edge expansion is as follows:

$$\varphi(G) \geq \varphi(H) \geq \varphi\Big(H + \sum_p y_p C_p\Big) \gtrsim \lambda_2\Big(L(H) - \sum_p y_p T_p\Big), \tag{19.6}$$

where the first inequality is by Claim 17.2 because H has an embedding into G, the second inequality is by the discussion above, which shows that adding shortcut cycles does not increase edge expansion, and the third inequality is by Exercise 18.8.

Why would adding shortcut cycles help in obtaining a stronger lower bound? There are graphs for which the easy direction of Cheeger's inequality is not tight, so that $\varphi \approx \lambda_2^2$ rather than $\varphi \approx \lambda_2$. A prototypical example is a long path, where every edge in the path is short in its spectral embedding, which leads to a severe violation of the ℓ_2^2 triangle inequality.

Intuitively, given an embedding of the vertices, we would like to add shortcut cycles along the paths that heavily violate ℓ_2^2 triangle inequalities, in order to increase the objective value of this embedding and thereby improve the lower bound provided by the second eigenvalue, while not decreasing the edge expansion of H by much. Thus, the dual program (19.6) can be understood as finding the best way to add these shortcut cycles to obtain the strongest possible spectral lower bound.

This interpretation is also consistent with the primal program (19.1), in which triangle inequalities are added to the spectral program. In the primal-dual algorithm that we present next, we will indeed add shortcut cycles for paths that strongly violate the ℓ_2^2 triangle inequalities in the embedding.

19.3 Primal-Dual Approach to Semidefinite Programs

Arora and Kale [AK16, Kal07] developed the matrix multiplicative weight update method to design a primal-dual $O(\sqrt{\log n})$-approximation algorithm for edge expansion in quadratic time, and Sherman [She09] extended this approach to obtain an $O(\sqrt{\log n})$-approximation algorithm running in almost-linear time. In this section, we follow the presentation of Sherman's result in [LTW24], which slightly simplifies the algorithm by using the more general dual program in (19.5).

Dual Solution from Regret Minimization Framework

To construct a solution to the dual program in (19.5), the idea is to use the regret minimization framework from Theorem 18.5 to build the solution incrementally.

In the regret minimization framework, the matrix multiplicative weight method in Algorithm 16 produces a density matrix X_t at each iteration. Suppose that in each iteration the algorithm can respond with a matrix M_t such that $\|M_t\| \leq \rho$ is small and $\langle X_t, M_t \rangle \geq 1$ is large. Then Theorem 18.5 (with $\epsilon = 1/(4\rho)$) implies that the average response $\frac{1}{T}\sum_{t=1}^{T} M_t$ has a large second eigenvalue, namely,

$$\lambda_2\Big(\frac{1}{T}\sum_{t=1}^{T} M_t\Big) \geq \frac{1}{T}\sum_{t=1}^{T}\langle X_t, M_t\rangle - \frac{1}{4} - \frac{\rho \ln n}{T\epsilon} = \frac{1}{T}\sum_{t=1}^{T}\langle X_t, M_t\rangle - \frac{1}{2} \geq \frac{1}{2},$$

where the second inequality follows by setting the number of iterations to $T \geq 16\rho^2 \ln n$.

For our dual program in (19.5), this framework applies if, given any density matrix X_t, we can either

- return a graph H_t that has a flow embedding into G with $\langle X_t, L(H_t)\rangle \geq 1$ and $\|L(H_t)\| \leq \rho$,
- or return a collection of paths $\mathcal{P}_t$ in K_n with weights $y_p \geq 0$ for each $p \in \mathcal{P}_t$ such that $\langle X_t, \sum_p y_p T_p\rangle \leq -1$ and $\|\sum_p y_p T_p\| \leq \rho$.

In this case, we can apply Theorem 18.5 to construct a dual solution to (19.5) by taking the average solution $\overline{H} = \frac{1}{T}\sum_{t=1}^{T} H_t$ (where $H_t = \emptyset$ if we return paths $\mathcal{P}_t$), such that

$$\lambda_2\Big(L(\overline{H}) - \frac{1}{T}\sum_{t=1}^{T}\sum_{p\in\mathcal{P}_t} y_p T_p\Big) \geq \frac{1}{2} \quad \text{and} \quad \overline{H} \text{ has a flow embedding into } G.$$

This certifies that the primal program in (19.2) has objective value at least $1/2$, and hence the edge expansion of G is at least $1/4$ by Exercise 18.8.

Primal-Dual Approach

The primal dual approach in combinatorial optimization proceeds by iteratively constructing either a primal solution or a dual solution, where failure to find a good primal solution leads to an update of the dual solution, and vice versa.

Geometric Embedding and Combinatorial Cut: In the setting of approximating edge expansion using the matrix multiplicative weight update method, we can interpret the density matrix $X_t \in \mathbb{R}^{n\times n}$ as a geometric embedding of the vertices into $\mathbb{R}^n$, serving as a potential primal solution to (19.2). Since $X_t \succcurlyeq 0$, we can take a Gram decomposition $X_t = V^\top V$ and let $v_i \in \mathbb{R}^n$ be the i-th column of V, so that $X_t(i,j) = \langle v_i, v_j\rangle$ for $i, j \in [n]$. Note that the vectors $\{v_1, \ldots, v_n\}$ do not necessarily satisfy the triangle inequalities; nevertheless, we can still attempt to use this geometric embedding to find a set with small edge expansion.

To do so, the algorithms in [AK16, She09] apply a random projection step to find two disjoint sets L and R, as in the cut-player strategy in Section 17.2 and Algorithm 17. They then set up a flow problem between L and R to identify a set of small edge expansion, as in the flow-player implementation in Section 17.3. The precise setting will be described in Section 19.4.

Primal-Dual Algorithm: We now outline the primal-dual algorithm.

1. In each iteration $1 \leq t \lesssim \rho^2 \log n$, we are given a density matrix X_t, and we compute a geometric embedding $v_1, \ldots, v_n \in \mathbb{R}^n$ via a Gram decomposition of X_t.

2. Perform a random projection on $\{v_1, \ldots, v_n\}$ to produce two disjoint sets L and R and setup a flow problem between L and R. This random-projection-and-flow step is repeated a number of times.

   a) If any of the flow problems is infeasible, then by the max-flow min-cut theorem (as in Lemma 17.13), we find a set S with $|S| \leq n/2$ and $\varphi(S) \lesssim \beta/c$ for some congestion parameter c and approximation ratio β. In this case, we obtain an integral primal solution to the edge expansion problem, and the primal-dual algorithm terminates.

   b) Suppose that all flows problems are feasible. Then each feasible flow yields an edge-weighted graph H_t that has a flow embedding into G with congestion c. There are two cases to consider, depending on the value (see (18.3)) of

$$\langle L(H_t), X_t \rangle = \sum_{ij \in E(H_t)} w_{H_t}(ij) \cdot \|v_i - v_j\|_2^2.$$

      i. If $\langle L(H_t), X_t \rangle \geq 1$ with constant probability, we return the matrix $L(H_t)$ as the response to the matrix multiplicative weight update algorithm. Combinatorially, this corresponds to routing flow between vertices that are far apart in the geometric embedding $v_1, \ldots, v_n$.

      ii. If $\langle L(H_t), X_t \rangle < 1$ with constant probability, this implies the existence of a matching-cover for $v_1, \ldots, v_n$ (see Definition 19.3). We then use Sherman's chaining result in Theorem 19.6 to construct a collection of paths $\mathcal{P}_t$ and weights $y_p \geq 0$ for $p \in \mathcal{P}_t$ such that

$$-1 \geq \left\langle \sum_{p \in \mathcal{P}_t} y_p T_p, X_t \right\rangle = \sum_{p=(i_1,\ldots,i_k) \in \mathcal{P}_t} \left(\sum_{j=1}^{k-1} \|v_{i_j} - v_{i_{j+1}}\|_2^2 - \|v_{i_1} - v_{i_k}\|_2^2 \right),$$

      and return the matrix $-\sum_{p \in \mathcal{P}_t} y_p T_p$ as the response to the matrix multiplicative weight update algorithm. Combinatorially, this corresponds to identifying paths that significantly violate the triangle inequalities.

   In either case, the matrix multiplicative weight update algorithm uses the returned matrix to update P_{t+1} matrix multiplicatively, producing a new embedding for the next iteration. For the algorithm to be efficient, the response matrix needs to have small norm, so that the width of the multiplicative update method is small. To achieve this, we will ensure that the graph H_t or the union of the paths in $\mathcal{P}_t$ has small maximum degree.

To summarize, the algorithm either finds a set S with $|S| \leq n/2$ and $\varphi(S) \lesssim \beta/c$ if some flow problem is infeasible in an iteration, or else, in every iteration, returns either a graph H_t that has a flow embedding into G or a collection of paths $\mathcal{P}_t$ that heavily violate the triangle inequalities as a response to the matrix multiplicative update algorithm, thereby updating the geometric embedding. In the latter case, the regret minimization framework allows us to conclude that the average of the responses forms a feasible dual solution certifying that $\varphi(G) \gtrsim 1/c$. Hence, performing a binary search over c yields an $O(\beta)$-approximation algorithm for the edge expansion problem.

Main Algorithm via Matrix Multiplicative Weight Update

Given the above ideas and intuition, we are now ready to formally state the main algorithm in Algorithm 19. We reduce the edge expansion problem to designing an efficient "oracle" for the matrix multiplicative weight update algorithm; the implementation of this oracle will be described in detail in Section 19.4.

Algorithm 19 Approximating Edge Expansion via Matrix Multiplicative Weights Update

Input: An undirected graph G on n vertices, a step size $\eta \in (0,1)$, a width bound $\rho > 0$, a congestion parameter $c > 0$, and an approximation factor $\beta \geq 1$.

Output: Either a set S with $|S| \leq n/2$ and $\varphi(S) \lesssim \beta/c$, or a dual solution to (19.5) with objective value $\gtrsim 1/c$.

Initialization: $X_1 = \frac{1}{n-1}(I_n - \frac{1}{n}1_n 1_n^\top)$.

For $t = 1$ to $T \asymp \rho^2 \ln n$:

1. Given $X_t \succcurlyeq 0$ such that $\mathrm{Tr}(X_t) = 1$ and $X_t \perp 1_n$, compute a Gram decomposition $X_t = V^\top V$ and let $v_i \in \mathbb{R}^n$ be the i-th column of V for $i \in [n]$.
2. **(Oracle)** Given the geometric embedding $v_1, \ldots, v_n$, return one of the following:
   a) A set $S \subseteq V$ with $|S| \leq n/2$ and $\varphi(S) \lesssim \beta/c$. If this succeeds, terminate the algorithm.
   b) A graph H_t that has a flow embedding into G with congestion c such that
   $$\langle L(H_t), X_t \rangle \geq 1 \quad \text{and} \quad \|L(H_t)\| \leq \rho.$$
   If this succeeds, then set $M_t := L(H_t)$.
   c) A collection of paths $\mathcal{P}_t$ in K_n and weights $y_p \geq 0$ for $p \in \mathcal{P}_t$ such that
   $$\Big\langle \sum_{p \in \mathcal{P}_t} y_p T_p, X_t \Big\rangle \leq -1 \quad \text{and} \quad \Big\| \sum_{p \in \mathcal{P}_t} y_p T_p \Big\| \leq \rho.$$
   If this succeeds, then set $M_t := -\sum_{p \in \mathcal{P}_t} y_p T_p$.
3. If Oracle succeeds, update
   $$X'_{t+1} := \exp\Big(-\frac{\eta}{\rho} \sum_{i=1}^{t} M_i \Big) \quad \text{and} \quad X_{t+1} := \frac{X'_{t+1} - \frac{1}{n}1_n 1_n^\top}{\mathrm{Tr}(X'_{t+1}) - 1}.$$

Let $\overline{M} := \frac{1}{T}\sum_{t=1}^{T} M_t$ be the average feedback matrix. Return $\overline{M}/c$ as the dual solution to (19.5).

We analyze Algorithm 19 under the assumption that there is a black-box implementation of the Oracle. The following lemma is essentially the same as the discussion earlier in this section; the only difference is the presence of the congestion parameter c.

Lemma 19.8 (Correctness of Main Algorithm)**.** *Suppose there is a black-box algorithm for the Oracle that always returns one of Cases (a), (b), or (c) in Step (2) of Algorithm 19. Then, by setting $\eta \lesssim 1/\rho$, Algorithm 19 either finds a cut $S \subseteq V$ with $|S| \leq |V|/2$ and $\varphi(S) \lesssim \beta/c$, or certifies that $\varphi(G) \gtrsim 1/c$.*

Proof. If the Oracle ever finds a cut S with $|S| \le n/2$ and $\varphi(S) \lesssim \beta/c$, then we are done. Otherwise, by the regret bound in Theorem 18.5,

$$\lambda_2(\overline{M}) \ge \frac{1}{T}\sum_{t=1}^{T}\langle M_t, X_t\rangle - 2\eta\rho - \frac{\rho \log n}{\eta T} \ge 1 - 2\eta\rho - \frac{\rho \log n}{\eta T} \ge 1 - \frac{1}{4} - \frac{1}{4} = \frac{1}{2},$$

where the first inequality uses the fact that $P_t \succcurlyeq X_t$ (where P_t is defined in Theorem 18.5) and that $\langle |M_t|, P_t\rangle \le \rho \cdot \mathrm{Tr}(P_t) \le 2\rho$, since $\|M_t\| \le \rho$ and $\mathrm{Tr}(P_t) \le 2$. The second inequality follows from the Oracle guarantee $\langle M_t, X_t\rangle \ge 1$, and the third inequality follows by setting $\eta \le 1/(8\rho)$ and $T \ge 32\rho^2 \log n$.

Note that the average feedback matrix $\overline{M}$ is of the form $L(\overline{H}) - \sum_p y_p T_p$, where $\overline{H}$ is the average of the graphs H_t, each of which has a flow embedding into G with congestion at most c. Therefore, by scaling down $\overline{H}$ by a factor of c, we obtain a solution to the dual program in (19.5) with objective value $1/c$, which certifies that $\varphi(G) \ge 1/c$. □

19.4 Efficient Implementation of Oracle

In this section, we describe the detailed implementation of the Oracle[3].

Given the geometric embedding $v_1, \ldots v_n$ in Step (1) of Algorithm 19, note that these vectors satisfy $\sum_{i=1}^n \|v_i\|_2^2 = 1$ and $\sum_{i=1}^n v_i = 0$, which together imply that $\sum_{i,j\in[n]} \|v_i - v_j\|_2^2 = 2n$. Let

$$u_i := \sqrt{\frac{n}{2}} \cdot v_i \text{ for } i \in [n], \qquad \text{so that } \sum_{i,j\in[n]} \|u_i - u_j\|_2^2 = n^2.$$

As in the geometric analysis of the structural theorem in Theorem 19.2, we distinguish between the well-spread case and the large-core case, as described in Definition 19.1. Once again, the large-core case is the easier case for implementing the Oracle; see Problem 19.19. Henceforth, we assume that the vectors $u_1, \ldots, u_n$ are well-spread.

Random Projection and Max-Flow-Min-Cut

Given well-spread vectors $u_1, \ldots, u_n$, we use the following "project-and-flow-cut" algorithm to either identify a set with small edge expansion or obtain a graph H_t that has a flow embedding into G. As mentioned earlier, this algorithm is closely related to the cut-player strategy and the flow-player implementation in Chapter 17.

[3] We remark that Sherman's Oracle only includes cases (a) and (b). To find a graph H_t that has a flow embedding into G satisfying the requirements in case (b), Sherman used a multi-commodity flow computation, which is in turn implemented via multiple single-commodity flow computations using another (classical) multiplicative weights update method. By using the general dual program in (19.5) and introducing case (c), the multi-commodity flow computation is bypassed, resulting in a simpler algorithm.

Algorithm 20 Project and Flow-Cut

Input: An undirected graph $G = (V, E)$, well-spread vectors $u_1, \ldots, u_n \in \mathbb{R}^n$, a direction $g \in \mathbb{R}^n$, a size parameter $0 < \alpha \leq 1/2$, a congestion parameter $c > 0$, and an approximation ratio β.

1. Order the vertices $i \in V$ by the values of $\langle g, u_i \rangle$. Let L be the set of the αn smallest vertices in this ordering, and let R be the set of the αn largest vertices.
2. Construct a flow network G' from G as follows.
   - Add a source vertex s to G with an edge of capacity β to each vertex in L.
   - Add a sink vertex t to G with an edge of capacity β to each vertex in R.
   - Set the capacity of each edge in $E(G)$ to be c.
3. Compute an s-t flow of value $\alpha\beta n$ in G'.
   a) If the flow problem is infeasible, output a set S with $|S| \leq n/2$ and $\varphi(S) \lesssim \beta/c$ obtained from a minimum s-t cut in G'.
   b) If the flow value is feasible, output a graph H with maximum weighted degree β that has a flow embedding into G with congestion c

Step 3(a) of Algorithm 20 can be implemented in the same way as described in the proof of Lemma 17.13, and we leave it as an exercise to the reader (see [LTW24, Lemma 3.3]). The following is an immediate consequence.

Claim 19.9 (Oracle (a))**.** *If the flow problem in Algorithm 20 is infeasible, then the Oracle succeeds in Case (a) in Algorithm 19.*

Step 3(b) can be implemented by computing a flow-path decomposition $\{f_p\}$ of the flow solution, which can be done in nearly linear time using the random-walk algorithm of [LRS13, Theorem 5]. Each flow path p corresponds to a path in G with one endpoint $i \in L$ and the other endpoint $j \in R$, and we add weight f_p to the edge ij in H. The weighted degree of each $i \in L$ is β, since the edge si has capacity β, and the same holds for each $j \in R$. Moreover, the flow has congestion at most c in G, as each edge in G has capacity c. Note that $\|L(H_t)\| \leq \beta$, since the maximum eigenvalue of a Laplacian matrix is bounded by its maximum degree. The following is an immediate consequence.

Claim 19.10 (Oracle (b))**.** *If the flow problem in Algorithm 20 is feasible and the graph H constructed in Step 3(b) of Algorithm 20 satisfies $\langle L(H), X_t \rangle \geq 1$, then the Oracle succeeds in Case (b) of Algorithm 19 with $\|L(H_t)\| \leq \beta$.*

Henceforth, we focus on the remaining case where the flow problem in Algorithm 20 is feasible but $\langle L(H), X_t \rangle < 1$ with high probability over the random direction g.

In the next subsection, we show that this implies the existence of an $(\Omega(1), \Omega(1))$-matching-cover for the vectors $u_1, \ldots, u_n$. Then, in the subsequent subsection, we explain how to apply Sherman's chaining result in Theorem 19.6 to this matching-cover in order to construct many paths that significantly violate the triangle inequalities, thereby ensuring that the Oracle succeeds in Case (c) of Algorithm 19. We note that the flow-paths f_p will not be used from this point on.

Existence of Matching-Cover

A standard Gaussian analysis shows that, with constant probability over the random direction g, the sets L and R in Algorithm 20 are well-separated along the direction g. In the following lemma, we assume that each $\|u_i\|_2^2$ is upper bounded by a constant, which can be achieved by selecting a constant fraction of vectors (see [Kal07, Lemma 5]). We remark that this lemma is the only place in the Oracle where the well-spread property of the vectors is used.

Lemma 19.11 (Good Direction [Kal07, Lemma 14]). *Let $u_1, \ldots, u_n$ be well-spread vectors satisfying $\sum_{i,j\in[n]} \|u_i - u_j\|_2^2 = n^2$ and $\|u_i\| \le 1$ for each $i \in [n]$. Then there exist positive constants α, σ, γ such that if we sample $g \sim N(0, I_n)$, then with probability at least γ, the sets L, R of size αn in Step (1) of Algorithm 20 satisfy*

$$\langle g, u_i - u_j \rangle \ge \sigma \quad \text{for all } i \in L \text{ and } j \in R.$$

We refer to such vectors g as good directions.

Suppose that g is a good direction but the graph H constructed in Algorithm 20 does not satisfy $\langle L(H), X_t \rangle \ge 1$. We show that, in this case, there exists a matching $M(g)$ containing many pairs of vectors u_i, u_j that are close in ℓ_2^2 distance but have a large projection along g.

Lemma 19.12 (Matching $M(g)$). *Given a good direction g as input to Algorithm 20, suppose the flow problem is feasible but the graph H constructed in Step 3(b) does not satisfy $\langle L(H), X_t \rangle \ge 1$ in Case (b) of Algorithm 19. Then there exists a matching $M(g)$ with at least $\alpha n/2$ edges, such that each edge $ij \in M(g)$ satisfies*

$$\langle g, u_i - u_j \rangle \ge \sigma \quad \text{and} \quad \|u_i - u_j\|_2^2 \le \frac{1}{\alpha\beta}.$$

Proof. We construct H as described in the paragraph preceding Claim 19.10. Since the flow problem is feasible, the total weight of edges in H is $\sum_{ij\in E(H)} w_H(i,j) = \alpha\beta n$. Because g is a good direction, every edge $ij \in E(H)$ satisfies $\langle g, u_i - u_j \rangle \ge \sigma$. To construct the matching, we discard any edge $ij \in E(H)$ with $\|u_i - u_j\|_2^2 > 1/(\alpha\beta)$. By assumption,

$$1 > \langle L(H), X_t \rangle = \sum_{i\in L, j\in R} w_H(i,j) \cdot \|v_i - v_j\|^2 = \frac{2}{n} \sum_{i\in L, j\in R} w_H(i,j) \cdot \|u_i - u_j\|^2,$$

and hence at most half of the total edge weight of H is discarded. Let H' denote the resulting graph, which has total edge weight at least $\alpha\beta n/2$. Since each vertex in H' has weighted degree at most β, the scaled graph H'/β is a fractional matching with total edge weight at least $\alpha n/2$. We can round H'/β into an integral matching $M(g)$ with at least $\alpha n/2$ edges in nearly linear time using the random walk algorithm of [LRS13, Theorem 5]. □

Now, suppose the graph H constructed in Algorithm 20 does not satisfy $\langle L(H), X_t \rangle \ge 1$ with high probability over a random choice of $g \sim N(0, I_n)$. Then we obtain a matching-cover as defined in Definition 19.3.

Lemma 19.13 (Matching Cover). *Suppose that, conditioned on g being a good direction, the probability that the assumption in Lemma 19.12 holds is at least $1/2$. Then there exists an $(\Omega(1), \Omega(1))$-matching-cover for the vectors $u_1, \ldots, u_n$, where every non-empty matching $M(g)$ satisfies the properties stated in Lemma 19.12.*

Proof. For each good direction g that satisfies the assumption in Lemma 19.12, we define $M(g)$ as in the conclusion of Lemma 19.12; for all other directions g, we set $M(g)$ to be the empty matching. The stretch and skew-symmetry conditions in Definition 19.3 are then clearly satisfied, with $\sigma = \Omega(1)$ as specified in Lemma 19.11. Furthermore,

$$\mathbb{E}_g[|M(g)|] \geq \frac{\gamma}{2} \cdot \frac{\alpha n}{2} = \Omega(n),$$

since g is a good direction satisfying the assumption in Lemma 19.12 with probability at least $\gamma/2 = \Omega(1)$ by Lemma 19.11, and $|M(g)| \geq \alpha n/2 = \Omega(n)$ by Lemma 19.12. □

Violating Paths via Chaining

We are now ready to apply Sherman's chaining result in Theorem 19.6 to the matching-cover, in order to construct many paths that significantly violate the triangle inequalities. This will show that the Oracle succeeds in Case (c) of Algorithm 19.

Lemma 19.14 (Violating Paths via Chaining). *Given the $(\Omega(1), \Omega(1))$-matching cover for vectors $u_1, \ldots, u_n$ from Lemma 19.13, by setting $\beta \asymp \sqrt{(\log n)/\epsilon}$, there exists a randomized algorithm that finds a collection of paths $\mathcal{P}$ and weights $y_p \geq 0$ for $p \in \mathcal{P}$ such that the feedback matrix*

$$M_t := -\sum_{p \in \mathcal{P}} y_p T_p \quad \textit{satisfies} \quad \langle M_t, X_t \rangle \geq 1 \textit{ and } \|M_t\| \leq n^{2\epsilon},$$

with success probability $\Omega(n^{-\epsilon})$.

Proof. We apply Sherman's Theorem 19.6 to the $(\Omega(1), \Omega(1))$-matching-cover from Lemma 19.13 with $\ell = \epsilon$ and $k = \sqrt{\epsilon \log n}$. This yields an efficiently samplable distribution $\mathcal{D}$ over $(g_1, \ldots, g_k)$ such that the expected number of edges (i, j) in $M(g_1, \ldots, g_k)$ with $\|u_i - u_j\|^2 \geq \ell = \epsilon$ is at least

$$e^{-k^2} \cdot n = n^{1-\epsilon}.$$

Since the total number of edges in $M(g_1, \ldots, g_k)$ is at most $2n$ (see Definition 19.4), a reverse application of Markov's inequality implies that, with probability at least $\Omega(n^{-\epsilon})$ over $(g_1, \ldots, g_k) \sim \mathcal{D}$, the number of edges in $M(g_1, \ldots, g_k)$ satisfying $\|u_i - u_j\|^2 \geq \ell = \epsilon$ is at least $\frac{1}{2} n^{1-\epsilon}$.

Fix such a choice of $(g_1, \ldots, g_k)$. Each edge ij in $M(g_1, \ldots, g_k)$ corresponds to a path $p_{ij} = (i_0, i_1, \ldots, i_k)$ with $i_0 = i$ and $i_k = j$, where $(i_{r-1}, i_r) \in M(g_r)$ for all $r \leq k$. Since each edge $(i_{r-1}, i_r) \in M(g_r)$ satisfies $\|u_{i_{r-1}} - u_{i_r}\|_2^2 \leq \frac{1}{\alpha\beta}$ by Lemma 19.12, setting $\beta = \frac{2}{\alpha}\sqrt{\frac{\log n}{\epsilon}}$ gives

$$\sum_{r=0}^{k-1} \|u_{i_r} - u_{i_{r+1}}\|_2^2 \leq \frac{k}{\alpha\beta} = \frac{\sqrt{\epsilon \log n}}{\alpha\beta} \leq \frac{\epsilon}{2}.$$

Thus, each such path p_{ij} with $\|u_i - u_j\|_2^2 \geq \epsilon$ violates the triangle inequality. Let

$$\mathcal{P} := \{p_{ij} \mid ij \in M(g_1, \ldots, g_k) \text{ and } \|u_i - u_j\|_2^2 \geq \epsilon\}$$

denote the collection of such paths, so that $|\mathcal{P}| \geq \frac{1}{2} n^{1-\epsilon}$. Since $u_i = \sqrt{\frac{n}{2}} \cdot v_i$, it follows that

$$\langle T_p, X_t \rangle = \sum_{r=0}^{k-1} \|v_{i_r} - v_{i_{r+1}}\|_2^2 - \|v_i - v_j\|_2^2 \leq -\frac{\epsilon}{n} \quad \Longrightarrow \quad \left\langle \sum_{p \in \mathcal{P}} T_p, X_t \right\rangle \leq -\frac{\epsilon}{2} \cdot n^{-\epsilon}.$$

Setting $y_p = \frac{2}{\epsilon} \cdot n^{\epsilon}$ for each $p \in \mathcal{P}$, the feedback matrix $M_t := -\sum_{p \in \mathcal{P}} y_p T_p$ satisfies $\langle M_t, X_t \rangle \geq 1$.

It remains to bound $\|M_t\|$. The edges appearing in the paths in $\mathcal{P}$ form a subgraph of $M(g_1) \cup \cdots \cup M(g_k)$. Since each $M(g)$ is a matching, the maximum degree in this subgraph is at most k. Therefore, $\|M_t\| \leq y_p \cdot k \leq n^{2\epsilon}$. □

Oracle Structure

We summarize the main result on the implementation of the Oracle.

Proposition 19.15 (Oracle in Well-Spread Case)**.** *Let $\epsilon > 0$ be a sufficiently small constant. Given well-spread vectors $u_1, \ldots, u_n \in \mathbb{R}^n$, there is a randomized implementation of the Oracle in Algorithm 19 that, with high probability and using at most $n^{2\epsilon}$ max-flow computations, either returns a feedback matrix M_t satisfying*

$$\langle M_t, X_t \rangle \geq 1 \quad \textit{and} \quad \|M_t\| \leq n^{2\epsilon},$$

or outputs a set $S \subseteq V$ with

$$|S| \leq \frac{n}{2} \quad \textit{and} \quad \varphi(S) \lesssim \frac{1}{c}\sqrt{\frac{\log n}{\epsilon}}.$$

Proof. We run the project-and-flow-cut Algorithm 20 for $O(\log n)$ iterations, each with an independent Gaussian vector $g \in N(0, I_n)$, with α as specified in Lemma 19.11 and $\beta \asymp \sqrt{(\log n)/\epsilon}$ as specified in Lemma 19.14.

1. If any flow problem is infeasible, then the Oracle succeeds in Case (a) of Algorithm 19, returning a set $S \subseteq V$ with
$$|S| \leq \frac{n}{2} \quad \text{and} \quad \varphi(S) \lesssim \frac{\beta}{c} = \frac{1}{c}\sqrt{\frac{\log n}{\epsilon}}.$$

2. Suppose all flow problems are feasible. By Claim 19.10, if the graph H constructed in Step 3(b) in Algorithm 20 satisfies $\langle L(H), X_t \rangle \geq 1$, then the Oracle succeeds in Case (b) in Algorithm 19 with $\|L(H_t)\| \leq \beta$.

   a) Suppose $\langle L(H), X_t \rangle \geq 1$ with probability at least $1/2$, conditioned on g being a good direction. By Lemma 19.11, this event occurs in one of the $O(\log n)$ iterations with high probability. Therefore, the Oracle succeeds in Case (b) with high probability, returning a feedback matrix M_t satisfying
   $$\langle M_t, X_t \rangle \geq 1 \quad \text{and} \quad \|M_t\| \leq \beta \lesssim \sqrt{\frac{\log n}{\epsilon}} \leq n^{2\epsilon}.$$

   b) The remaining case is when $\langle L(H), X_t \rangle \geq 1$ holds with probability less than $1/2$, conditioned on g being a good direction. By Lemma 19.13, there exists an $(\Omega(1), \Omega(1))$-matching-cover for the vectors $u_1, \ldots, u_n$. Applying Lemma 19.14 to construct violating paths, the Oracle succeeds in Case (c) of Algorithm 19, returning a feedback matrix M_t with
   $$\langle M_t, X_t \rangle \geq 1 \quad \text{and} \quad \|M_t\| \leq n^{2\epsilon},$$
   with probability $\Omega(n^{-\epsilon})$. Repeating this procedure $O(n^{\epsilon})$ times reduces the failure probability to an arbitrarily small constant.

The runtime is dominated by Case (c), which requires $O(n^\epsilon)$ repetitions of the procedure. Each repetition performs $O(\sqrt{\epsilon \log n})$ max-flow computations to construct and chain together the matchings as described in Lemma 19.12 and Lemma 19.14. The width of the oracle is also dominated by Case (c), with $\|M_t\| \le n^{2\epsilon}$. □

Sherman's Theorem

Finally, we arrive at Sherman's almost linear-time $O(\sqrt{\log n})$-approximation algorithm for the edge expansion problem.

Theorem 19.16 (Almost Linear Time $O(\sqrt{\log n})$-Approximation for Edge Expansion)**.** *For sufficiently small $\epsilon > 0$, there exists a randomized algorithm that, given any undirected graph $G = (V, E)$, uses at most $n^{O(\epsilon)}$ max-flow computations to compute a cut $S \subseteq V$ such that $\varphi(S)$ is at most $O(\sqrt{\log n/\epsilon})$ times the optimal value of (19.1), with constant probability.*

This follows by using the Oracle in Proposition 19.15 within the matrix multiplicative weights update Algorithm 19, which runs for at most $\rho^2 \log n \le n^{O(\epsilon)}$ iterations. In each iteration, an approximate Gram decomposition of X_t can be computed in almost linear time using Taylor expansion and dimension reduction; see [LTW24, Appendix B]. Correctness follows from Lemma 19.8, together with a binary search over the congestion parameter c.

It remains open to further improve Theorem 19.16 and obtain an $O(\sqrt{\log n})$-approximation algorithm for edge expansion running in nearly linear time. Such a result would match the runtime guarantees of cut-matching games while achieving the best known approximation ratio for the problem.

19.5 Problems

Problem 19.17 (Rounding in Well-Spread Case)**.** *Let $v_1, \dots, v_n \in \mathbb{R}^n$ be an optimal solution to the semidefinite programming relaxation for edge expansion in (19.1).*

1. *Let $u_i := \sqrt{\frac{n}{2}} \cdot v_i$ for $i \in [n]$. Show that $u_1, \dots, u_n \in \mathbb{R}^n$ satisfy $\sum_{i,j \in [n]} \|u_i - u_j\|^2 = n^2$.*
2. *Assume $u_1, \dots, u_n$ are well-spread (see Definition 19.1). Use Theorem 19.2 to return a set S with $\varphi(S)$ at most $O(\sqrt{\log n})$ times the objective value of (19.1).*

   Hint: Let L and R be the sets returned by Theorem 19.2. Define $d(i, L) := \min_{j \in L} \|u_i - u_j\|_2^2$. For each $0 \le r \le \Omega(1/\sqrt{\log n})$, consider the threshold set $S_r := \{i \in [n] \mid d(i, L) \le r\}$. You may use the analysis in Lemma 2.5 to choose a uniformly random r and bound the expected numerator and denominator separately.

Problem 19.18 (Rounding in Large-Core Case)**.** *In the same setting as in Problem 19.17, assume there is a large core around u_i (see Definition 19.1) and let $L := \{j \in [n] \mid \|u_i - u_j\|_2^2 \le \frac{1}{40}\}$.*

1. *Show that $\sum_{i \notin L} d(i, L) \gtrsim n$, where $d(i, L)$ is as defined in Problem 19.17.*

   Hint: You may use $d(i, j) \le d(i, L) + \mathrm{diam}(L) + d(j, L)$ where $\mathrm{diam}(L) := \max_{i,j \in L} \|u_i - u_j\|_2^2$.
2. *Prove that there exists $r \ge 0$ such that $\min\{\varphi(S_r), \varphi(\overline{S_r})\}$ is at most $O(1)$ times the objective value of (19.1), where S_r is defined as in Problem 19.17.*

   Hint: You may use part (1) to analyze the expected denominator.

Problem 19.19 (Oracle in Large-Core Case)**.** *In the large core case, design an algorithm that implements the Oracle and either outputs a set S with $|S| \leq n/2$ and $\varphi(S) \lesssim 1/c$, or returns a graph H that has a flow embedding into G with congestion c such that $\langle L(H), X_t \rangle \geq 1$ and $\|L(H)\| \lesssim 1$.*

References

[AK16] Sanjeev Arora and Satyen Kale. A combinatorial, primal-dual approach to semidefinite programs. *Journal of the ACM (JACM)*, 63(2):1–35, 2016. 9, 227, 228, 235, 240, 241, 242

[ARV09] Sanjeev Arora, Satish Rao, and Umesh Vazirani. Expander flows, geometric embeddings and graph partitioning. *Journal of the ACM (JACM)*, 56(2):1–37, 2009. 214, 235, 236, 237, 238, 240

[BS16] Boaz Barak and David Steurer. Proofs, beliefs, and algorithms through the lens of sum-of-squares. *Course notes: http://www. sumofsquares. org/public/index. html*, 1:3, 2016. 235

[Kal07] Satyen Kale. *Efficient algorithms using the multiplicative weights update method*. Princeton University, 2007. 227, 228, 232, 236, 237, 241, 247

[Lee05] James R Lee. On distance scales, embeddings, and efficient relaxations of the cut cone. In *Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 92–101, 2005. 238

[LRS13] Yin Tat Lee, Satish Rao, and Nikhil Srivastava. A new approach to computing maximum flows using electrical flows. In *Symposium on Theory of Computing Conference, STOC'13*, pages 755–764. ACM, 2013. 221, 246, 247

[LTW24] Lap Chi Lau, Kam Chuen Tung, and Robert Wang. Fast algorithms for directed graph partitioning using flows and reweighted eigenvalues. In *Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 591–624. SIAM, 2024. 131, 230, 232, 241, 246, 250

[Rot16] Thomas Rothvoss. Lecture notes on the arv algorithm for sparsest cut. *arXiv preprint arXiv:1607.00854*, 2016. 235, 237

[She09] Jonah Sherman. Breaking the multicommodity flow barrier for o (v log n)-approximations to sparsest cut. In *2009 50th Annual IEEE Symposium on Foundations of Computer Science*, pages 363–372. IEEE, 2009. 9, 235, 238, 240, 241, 242

[Tre17] Luca Trevisan. Lecture notes on graph partitioning, expanders and spectral methods. 2017. https://lucatrevisan.github.io/books/expanders-2016.pdf. 214, 215, 235, 236

Part V

Semidefinite Programming and Approximation Algorithms

We study how the local-to-global characterization of eigenvalues can be used to design approximation algorithms for various problems on low threshold rank graphs. The goal of this line of work is to design better approximation algorithms for general graphs, and we end with some conjectures towards this goal.

Chapter 20

Local-to-Global Correlation Rounding on Expander Graphs

In this chapter, we study the global correlation rounding technique introduced in [AKK+08] for designing improved approximation algorithms for Unique Games on expander graphs, based on the local-to-global characterization of the second eigenvalue.

20.1 Unique Games and Semidefinite Programming Relaxation

Unique Games is a constraint satisfaction problem in which one is given a constraint graph $G = (V, E)$, a label set $[k]$, and for each edge $uv \in E(G)$ a bijective mapping $\pi_{uv} : [k] \to [k]$. The goal is to assign to each vertex of G a label from $[k]$ so as to maximize the number of satisfied constraints. An edge uv is said to be satisfied by an assignment if u is assigned a label i and v is assigned a label j such that $\pi_{uv}(i) = j$.

An important special case of Unique Games is the Max2Lin problem (see Definition 13.9), where the constraint for an edge uv is of the form

$$x_u \equiv x_v + c_{u,v} \mod k,$$

for constant $c_{u,v}$ associated with the edge uv. Notice that this captures the classical Maximum Cut problem, which can be written as a constraint satisfaction problem using the constraints $x_u \equiv x_v + 1 \mod 2$ for every $uv \in E(G)$.

Semidefinite Programming Relaxation

The following is the standard semidefinite programming relaxation for Unique Games, where for each vertex $u \in V$ and label $i \in [k]$ there is a vector u_i indicating whether u is assigned label i.

$$\begin{aligned} \mathsf{sdp}(G) \;:=\; \max \quad & \frac{1}{|E|} \sum_{uv \in E} \sum_{i \in [k]} \langle \mathsf{u}_i, \mathsf{v}_{\pi_{uv}(i)} \rangle & & (20.1) \\ \text{subject to} \quad & \sum_{i \in [k]} \|\mathsf{u}_i\|_2^2 = 1 & \forall u \in V & \\ & \langle \mathsf{u}_i, \mathsf{u}_j \rangle = 0 & \forall u \in V, \;\; \forall i \neq j. & \end{aligned}$$

The intended integral solution is $\mathsf{u}_i = e_1$ if vertex u is assigned label i, and $\mathsf{u}_i = 0$ otherwise. Under such an assignment, the objective function equals the fraction of satisfied constraints. Note that the objective value can be rewritten as

$$\frac{1}{|E|}\sum_{uv\in E}\sum_{i\in[k]}\langle \mathsf{u}_i, \mathsf{v}_{\pi_{uv}(i)}\rangle = 1 - \frac{1}{2|E|}\sum_{uv\in E}\sum_{i\in[k]}\big\|\mathsf{u}_i - \mathsf{v}_{\pi_{uv}(i)}\big\|_2^2. \tag{20.2}$$

Main Theorem

We illustrate the main result of [AKK+08] in the special case of Max2Lin, which admits a better approximation ratio and also a simpler proof.

Theorem 20.1 (Max2Lin on Expander Graphs)**.** *Given a Max2Lin instance on a regular graph $G = (V, E)$, if there exists an assignment satisfying at least a $1-\epsilon$ fraction of constraints, then there is a polynomial-time algorithm that returns an assignment satisfying at least a*

$$1 - O\Big(\frac{\epsilon}{\lambda_2(\mathcal{L}(G))}\Big)$$

fraction of constraints, where $\lambda_2(\mathcal{L}(G))$ denotes the second smallest eigenvalue of the normalized Laplacian matrix of G.

For the Maximum Cut problem on general graphs, the celebrated SDP-based algorithm due to Goemans and Williamson [GW95] returns a cut with at least a $1 - O(\sqrt{\epsilon})$ fraction of edges when $\mathsf{sdp}(G) \geq 1-\epsilon$. The above result improves this guarantee when G is a spectral expander, i.e., when $\lambda_2(\mathcal{L}(G)) = \Omega(1)$.

20.2 Global Correlation Rounding

Suppose $\mathsf{sdp}(G) \geq 1-\epsilon$ for some small constant $\epsilon > 0$. Then for most edges uv, the vectors $\{\mathsf{u}_i\}_{i\in[k]}$ and $\{\mathsf{v}_i\}_{i\in[k]}$ are highly correlated in the sense that $\sum_{i\in[k]}\langle \mathsf{u}_i, \mathsf{v}_{\pi_{uv}(i)}\rangle \geq 1 - O(\epsilon)$. A natural idea is to fix the label of a vertex u to some value i, and propagate this assignment by assiging each neighbor v a label j that maximizes $\langle \mathsf{u}_i, \mathsf{v}_j\rangle$, and then continue propagating to their neighbors, and so on. In an ideal scenario, this would define a locally consistent assignment that satisfies most constraints in a small constant-hop neighborhood of u. However, it need not define a globally consistent assignment that satisfies most constraints over the entire graph.

Propagation Rounding Algorithm

To define a globally consistent assignment, for every pair of vertices u, v, let $\sigma_{uv} : [k] \to [k]$ be a bijection that maximizes

$$\sum_{i\in[k]}\langle \mathsf{u}_i, \mathsf{v}_{\sigma_{uv}(i)}\rangle, \tag{20.3}$$

which can be computed using a maximum-weight bipartite matching algorithm. Note that σ_{uv} is not necessarily equal to π_{uv}, the bijection defining the constraint on the edge uv.

The following is a simple randomized algorithm that propagates the label of a vertex u to all vertices in the graph via the bijections $\{\sigma_{uv}\}_{v\in V}$.

Algorithm 21 Propagation Rounding Algorithm

Require: An SDP solution $\{\mathsf{u}_i\}_{u\in V, i\in[k]}$ to (20.1).

1: Compute the bijections $\{\sigma_{uv}\}_{u,v\in V}$ defined in (20.3).
2: Choose a uniformly random vertex $u \in V$.
3: Sample a label $i \in [k]$ with probability $\|\mathsf{u}_i\|_2^2$, and set $x_u = i$.
4: For every vertex $v \neq u$, set $x_v = \sigma_{uv}(i)$.
5: **return** The assignment $x : V \to [k]$.

Clearly, this randomized algorithm can be easily derandomized by trying all possible choices of the vertex u and all possible labels i. We state it as a randomized algorithm because this formulation is more convenient for the analysis.

Analysis via Global Correlation

For the analysis, define the distance between a pair of vertices by

$$d(u,v) := \frac{1}{2}\sum_{i\in[k]} \|\mathsf{u}_i - \mathsf{v}_{\sigma_{uv}(i)}\|_2^2 = 1 - \sum_{i\in[k]} \langle \mathsf{u}_i, \mathsf{v}_{\sigma_{uv}(i)}\rangle. \tag{20.4}$$

Note that two vertices u and v are highly correlated if and only if the distance $d(u,v)$ is small. Let

$$\mathbb{E}_{u,v}[d(u,v)] := \frac{1}{|V|^2}\sum_{u,v\in V} d(u,v)$$

denote the average distance between a pair of vertices. The following lemma shows that if the average distance is small, then the propagation rounding algorithm performs well.

Lemma 20.2 (Global Correlation Implies High Value)**.** *Suppose $\mathsf{sdp}(G) \geq 1-\epsilon$ for a regular graph G. For an assignment $x : V \to [k]$, let $\mathsf{val}(x)$ denote the fraction of constraints satisfied by x. Then the expected value of the assignment produced by the propagation rounding algorithm satisfies*

$$\mathbb{E}_x[\mathsf{val}(x)] \geq 1 - 3\epsilon - 6\cdot\mathbb{E}_{u,v}[d(u,v)].$$

For intuition, consider the extreme case of an edge vw where $d(u,v) = d(u,w) = 0$ with $\mathsf{u}_i = \mathsf{v}_i = \mathsf{w}_i$ for $i \in [k]$, but $\pi_{vw}(i) \neq i$ for all $i \in [k]$. In this case, the constraint on vw is violated with probability one when we perform propagation rounding from u. Since the vectors $\{\mathsf{v}_i\}_{i\in[k]}$ and $\{\mathsf{w}_i\}_{i\in[k]}$ satisfy the orthogonality constraints in (20.1), it follows that $\sum_{i\in[k]}\langle \mathsf{v}_i, \mathsf{w}_{\pi_{vw}(i)}\rangle = 0$, and thus this edge contributes zero to the objective value of (20.1). Since the objective value of (20.1) is close to one, this situation cannot occur too often. Therefore, if the average distance is close to zero, most constraints are satisfied by the propagation rounding algorithm. The following lemma formalizes this intuition.

Lemma 20.3 (Bounding Violating Probability by Distances and Objective Value)**.** *For every vertex $u \in V$ and every edge $vw \in E$,*

$$\Pr\left[x_w \neq \pi_{vw}(x_v) \mid u \text{ is chosen}\right] \leq 3\cdot\left(d(u,v) + d(u,w) + \frac{1}{2}\sum_{i\in[k]}\left\|\mathsf{v}_i - \mathsf{w}_{\pi_{vw}(i)}\right\|_2^2\right).$$

Proof. By relabeling the coordinates, we may assume that σ_{uv} and σ_{uw} are the identity permutations. Hence $x_v = x_w = x_u$ whenever u is chosen. Conditioned on u being the chosen vertex,

$$\begin{aligned}\Pr[x_w \neq \pi_{vw}(x_v) \mid u \text{ is chosen}] &= \sum_{i\in[k]:\pi_{vw}(i)\neq i} \|\mathsf{u}_i\|_2^2 \\ &= \frac{1}{2}\sum_{i\in[k]:\pi_{vw}(i)\neq i} \left(\|\mathsf{u}_i\|_2^2 + \left\|\mathsf{u}_{\pi_{vw}(i)}\right\|_2^2\right) \\ &= \frac{1}{2}\sum_{i\in[k]:\pi_{vw}(i)\neq i} \left(\left\|\mathsf{u}_i - \mathsf{u}_{\pi_{vw}(i)}\right\|_2^2\right),\end{aligned}$$

where the last equality uses the orthogonality constraints in (20.1). By the triangle inequality,

$$\left\|\mathsf{u}_i - \mathsf{u}_{\pi_{vw}(i)}\right\|_2 \leq \left\|\mathsf{u}_i - \mathsf{v}_i\right\|_2 + \left\|\mathsf{v}_i - \mathsf{w}_{\pi_{vw}(i)}\right\|_2 + \left\|\mathsf{w}_{\pi_{vw}(i)} - \mathsf{u}_{\pi_{vw}(i)}\right\|_2.$$

Using the inequality $(a+b+c)^2 \leq 3(a^2+b^2+c^2)$,

$$\left\|\mathsf{u}_i - \mathsf{u}_{\pi_{vw}(i)}\right\|_2^2 \leq 3\left(\left\|\mathsf{u}_i - \mathsf{v}_i\right\|_2^2 + \left\|\mathsf{v}_i - \mathsf{w}_{\pi_{vw}(i)}\right\|_2^2 + \left\|\mathsf{w}_{\pi_{vw}(i)} - \mathsf{u}_{\pi_{vw}(i)}\right\|_2^2\right).$$

Summing over $i \in [k]$, it follows that

$$\sum_{i\in[k]} \left\|\mathsf{u}_i - \mathsf{u}_{\pi_{vw}(i)}\right\|_2^2 \leq 6\Big(d(u,v) + d(u,w) + \frac{1}{2}\sum_{i\in[k]} \left\|\mathsf{v}_i - \mathsf{w}_{\pi_{vw}(i)}\right\|_2^2\Big).$$

Substituting back into the probability bound proves the lemma. □

Summing the inequality in Lemma 20.3 over $u \in V$ and over $vw \in E$ gives Lemma 20.2.

Proof of Lemma 20.2. By the definition of the propagation rounding algorithm,

$$\begin{aligned}\mathbb{E}_x[\mathsf{val}(x)] &= \frac{1}{|V|}\sum_{u\in V}\frac{1}{|E|}\sum_{vw\in E} \Pr[x_w = \pi_{vw}(x_v) \mid u \text{ is chosen}] \\ &\geq \frac{1}{|V|}\sum_{u\in V}\frac{1}{|E|}\sum_{vw\in E}\Big(1 - 3\cdot\Big(d(u,v) + d(u,w) + \frac{1}{2}\sum_{i\in[k]}\left\|\mathsf{v}_i - \mathsf{w}_{\pi_{vw}(i)}\right\|_2^2\Big)\Big) \\ &= -2 + 3\mathsf{sdp}(G) - \frac{3}{|V||E|}\sum_{u\in V}\sum_{vw\in E}(d(u,v)+d(u,w)) \\ &= -2 + 3\mathsf{sdp}(G) - \frac{6}{|V|^2}\sum_{u,v\in V} d(u,v) \\ &\geq 1 - 3\epsilon - 6\cdot\mathbb{E}_{u,v}[d(u,v)],\end{aligned}$$

where the second line uses Lemma 20.3, the third line uses the definition of the objective value in (20.2) the fourth line uses that the graph is regular so each $d(u,v)$ is counted the same number of times in the inner summation, and the last line uses the assumption that $\mathsf{sdp}(G) \geq 1-\epsilon$. □

20.3 Local-to-Global Correlations on Expander Graphs

Global Correlation: From Lemma 20.2, if the average distance over pairs satisfies $\mathbb{E}_{u,v\in V}[d(u,v)] \lesssim \epsilon$, then the propagation rounding algorithm outputs an assignment satisfying a $1 - O(\epsilon)$ fraction of constraints.

Local Correlation: On the other hand, by (20.4), the average distance over edges is

$$\mathbb{E}_{uv\in E}[d(u,v)] := \frac{1}{|E|}\sum_{uv\in E} d(u,v) = \frac{1}{2}\frac{1}{|E|}\sum_{uv\in E}\sum_{i\in[k]}\|\mathsf{u}_i - \mathsf{v}_{\sigma_{uv}(i)}\|_2^2 = 1 - \frac{1}{|E|}\sum_{uv\in E}\sum_{i\in[k]}\langle \mathsf{u}_i, \mathsf{v}_{\sigma_{uv}(i)}\rangle.$$

Given a solution with $\mathsf{sdp}(G) \geq 1-\epsilon$, observe that the average edge distance is small, since

$$\mathbb{E}_{uv\in E}[d(u,v)] = 1 - \frac{1}{|E|}\sum_{uv\in E}\sum_{i\in[k]}\langle \mathsf{u}_i, \mathsf{v}_{\sigma_{uv}(i)}\rangle \leq 1 - \frac{1}{|E|}\sum_{uv\in E}\sum_{i\in[k]}\langle \mathsf{u}_i, \mathsf{v}_{\pi_{uv}(i)}\rangle = 1 - \mathsf{sdp}(G) \leq \epsilon. \quad (20.5)$$

where the inequality follows from the definition of σ_{uv} in (20.3).

Local-to-Global Characterization of Second Eigenvalue

Following the above observation, given a graph $G = (V, E)$, if we can show that

$$\mathbb{E}_{uv\in E}[d(u,v)] \leq \epsilon \quad \Longrightarrow \quad \mathbb{E}_{u,v\in V}[d(u,v)] \lesssim \epsilon,$$

then we prove that the propagation rounding algorithm works well for G.

Given a regular graph $G = (V, E)$, by Lemma 26.20, the second eigenvalue of the normalized Laplacian can be formulated[1] as

$$\lambda_2(\mathcal{L}(G)) = \min_{f:V\to\mathbb{R}^n} \frac{\mathbb{E}_{uv\in E}[\|f(u) - f(v)\|_2^2]}{\mathbb{E}_{u,v\in V}[\|f(u) - f(v)\|_2^2]}.$$

Therefore, for any $f : V \to \mathbb{R}^n$,

$$\mathbb{E}_{u,v\in V}[\|f(u) - f(v)\|_2^2] \leq \frac{\mathbb{E}_{uv\in E}[\|f(u) - f(v)\|_2^2]}{\lambda_2(\mathcal{L}(G))}. \quad (20.6)$$

This provides a way to upper bound global correlation by local correlation, and is most effective when G is a spectral expander with $\lambda_2(\mathcal{L}(G)) \gtrsim 1$.

Low Distortion Embedding

To apply the spectral formulation to bound global correlation by local correlation, a natural strategy is to find a geometric embedding $f : V \to \mathbb{R}^n$ such that

$$\|f(u) - f(v)\|_2^2 \approx d(u,v) \quad \forall u, v \in V.$$

A main technical lemma in [AKK+08] is the following embedding result.

Lemma 20.4 (Low Distortion Embedding of Distances [AKK+08, Lemma 2.2]). *For every positive even integer t and every SDP solution $\{\mathsf{u}_i\}_{u\in V, i\in[k]}$, there exists a set of vectors $\{f(u)\}_{u\in V}$ such that for every pair of vertices u, v,*

$$\frac{1}{2t}\|f(u) - f(v)\|_2^2 \;\leq\; d(u,v) \;\leq\; \|f(u) - f(v)\|_2^2 + O\big(2^{-\frac{t}{2}}\big).$$

[1] In a regular graph, $\pi = \vec{1}/|V|$ and $\tilde{\mathcal{L}} = \mathcal{L}$. The formulation in Lemma 26.20 is over $f : V \to \mathbb{R}$, but it can be extended to $f : V \to \mathbb{R}^n$ for any $n \in \mathbb{N}$ using Spielman's favorite inequality (Lemma 2.6), as in the proof of Lemma 9.5.

The above result is for general Unique Games. For Max2Lin, there is a simpler embedding that implies a stronger approximation result. Note that the following embedding does not preserve the distances term-by-term, but only preserves the average distance, which suffices for the purpose of local-to-global correlation.

Lemma 20.5 (Low Distortion Embedding of Distances for Max2Lin [AKK+08, Lemma 3.1]). *Let $\{\mathsf{u}_i\}_{u\in V, i\in[k]}$ be a solution to (20.1) with objective value $1-\epsilon$. Let $y : V \to [k]$ be a labeling that satisfies at least a $1-\delta$ fraction of constraints. There exists a set of vectors $\{f(u)\}_{u\in V}$ such that*

1. *$d(u,v) \le \frac{1}{2}\|f(u)-f(v)\|_2^2$ for every pair of vertices $u, v \in V$.*
2. *$\frac{1}{2}\mathbb{E}_{uv\in E}[\|f(u)-f(v)\|_2^2] \le \epsilon + 2\delta$.*

We remark that the embedding result is used only in the analysis, but not in the algorithm, so there is no need to compute a good assignment y in the assumption of Lemma 20.5.

Proof of Main Theorem

We first prove Theorem 20.1 assuming Lemma 20.5, whose proof is presented in the next subsection.

Proof of Theorem 20.1. Given an SDP solution $\{\mathsf{u}_i\}_{u\in V, i\in[k]}$ with objective value at least $1-\epsilon$, it follows from (20.5) that

$$\mathbb{E}_{uv\in E}[d(u,v)] \le \epsilon.$$

Using the embedding $f : V \to \mathbb{R}^n$ in Lemma 20.5 and the spectral characterization in (20.6),

$$2\mathbb{E}_{u,v\in V} d(u,v) \le \mathbb{E}_{u,v\in V}[\|f(u)-f(v)\|_2^2] \le \frac{\mathbb{E}_{uv\in E}[\|f(u)-f(v)\|_2^2]}{\lambda_2(\mathcal{L}(G))} \le \frac{2\epsilon+4\delta}{\lambda_2(\mathcal{L}(G))} \le \frac{6\delta}{\lambda_2(\mathcal{L}(G))}.$$

Note that $\epsilon \le \delta$ since the SDP is a relaxation of the Unique Games problem. Therefore, we conclude from Lemma 20.2 that the derandomized propagation rounding algorithm succeeds in outputting an assignment with value at least

$$\mathbb{E}_x[\mathsf{val}(x)] \ge 1 - 3\epsilon - 6\cdot\mathbb{E}_{u,v}[d(u,v)] \ge 1 - 3\epsilon - \frac{36\delta}{\lambda_2(\mathcal{L}(G))} \ge 1 - O\Big(\frac{\delta}{\lambda_2(\mathcal{L}(G))}\Big).$$

□

Proof of Low Distortion Embedding

It remains to prove Lemma 20.5. The definition of $d(u,v)$ in (20.4) is consistent with an ℓ_2^2-distance, but the difficulty is that every edge uv has a different permutation σ_{uv}. Thus, it is unclear what geometric embedding $f : V \to \mathbb{R}^n$ of the *vertices* would preserve these edge distances.

Shift Invariance: The special property of Max2Lin is that the constraints π_{uv} satisfy

$$\pi_{uv}(i+s) = \pi_{uv}(i) + s \quad \text{for all } i, s \in [k],$$

where the additions are modulo k. This property implies that we may assume the SDP solution $\{\bar{\mathsf{u}}_i\}_{u\in V, i\in[k]}$ is *shift-invariant*, namely,

$$\langle \bar{\mathsf{u}}_{i+s}, \bar{\mathsf{v}}_{j+s}\rangle = \langle \bar{\mathsf{u}}_i, \bar{\mathsf{v}}_j\rangle \quad \text{for all } i, j, s \in [k],$$

where the additions are modulo k. For example, given an arbitrary SDP solution $\{\mathsf{u}_i\}_{u\in V, i\in[k]}$, one can verify that

$$\left\{\bar{\mathsf{u}}_i := \frac{1}{\sqrt{k}}(\mathsf{u}_i, \mathsf{u}_{i+1}, \ldots, \mathsf{u}_{i+k-1})\right\}_{u\in V, i\in[k]}$$

is also an SDP solution with the same objective value and is shift-invariant, where $(\mathsf{u}_i, \mathsf{u}_{i+1}, \ldots, \mathsf{u}_{i+k-1})$ denotes the concatenation of the k vectors, and the additions are again modulo k.

The following simplifications arise from a shift-invariant solution $\{\bar{\mathsf{u}}_i\}_{u\in V, i\in[k]}$ for both σ_{uv} and π_{uv}.

1. Note that $\|\bar{\mathsf{u}}_i\|_2^2 = 1/k$ for all $u \in V$ and $i \in [k]$.

2. For an edge uv, let $a, b \in [k]$ be a pair that maximizes $\langle \bar{\mathsf{u}}_a, \bar{\mathsf{v}}_b\rangle$. Since $\langle \bar{\mathsf{u}}_{a+s}, \bar{\mathsf{v}}_{b+s}\rangle = \langle \bar{\mathsf{u}}_a, \bar{\mathsf{v}}_b\rangle$ for $s \in [k]$ by shift invariance, the bijection $((a \to b), (a+1 \to b+1), \ldots, (a+k-1 \to b+k-1))$ is a maximizer for σ_{uv} in (20.3). This implies that

$$d(u,v) = \frac{1}{2}\sum_{i\in[k]} \left\|\bar{\mathsf{u}}_i - \bar{\mathsf{v}}_{\sigma_{uv}(i)}\right\|_2^2 = \frac{k}{2}\min_{i,j\in[k]} \left\|\bar{\mathsf{u}}_i - \bar{\mathsf{v}}_j\right\|_2^2.$$

3. For an edge uv, by shift invariance and the identity $\pi_{uv}(i+s) = \pi_{uv}(i) + s$ for the Max2Lin constraints, its contribution to the objective value in (20.2) is

$$\frac{1}{2}\sum_{i\in[k]} \left\|\bar{\mathsf{u}}_i - \bar{\mathsf{v}}_{\pi_{uv}(i)}\right\|_2^2 = \frac{1}{2}\sum_{s\in[k]} \left\|\bar{\mathsf{u}}_{j+s} - \bar{\mathsf{v}}_{\pi_{uv}(j+s)}\right\|_2^2 = \frac{1}{2}\sum_{s\in[k]} \left\|\bar{\mathsf{u}}_{j+s} - \bar{\mathsf{v}}_{\pi_{uv}(j)+s}\right\|_2^2 = \frac{k}{2}\left\|\bar{\mathsf{u}}_j - \bar{\mathsf{v}}_{\pi_{uv}(j)}\right\|_2^2$$

for any $j \in [k]$, where we used $\|\bar{\mathsf{u}}_i\|_2^2 = 1/k$ for all $u \in V$ and $i \in [k]$.

The simplifications suggest a good geometric embedding from a good assignment.

Proof of Lemma 20.5. Given a shift-invariant SDP solution $\{\bar{\mathsf{u}}_i\}_{u\in V, i\in[k]}$ and an integral solution $y : V \to [k]$, define

$$f(u) = \sqrt{k}\cdot \bar{\mathsf{u}}_{y_u}.$$

Then the first statement in Lemma 20.5 holds, since

$$d(u,v) = \frac{k}{2}\min_{i,j\in[k]}\|\bar{\mathsf{u}}_i - \bar{\mathsf{v}}_j\|_2^2 \le \frac{k}{2}\|\bar{\mathsf{u}}_{y_u} - \bar{\mathsf{v}}_{y_v}\|_2^2 = \frac{1}{2}\|f(u) - f(v)\|_2^2.$$

For the second statement,

$$\frac{1}{2}\mathbb{E}_{uv\in E}[\|f(u) - f(v)\|_2^2] = \frac{1}{2|E|}\sum_{uv\in E}[\|f(u) - f(v)\|_2^2] = \frac{k}{2|E|}\sum_{uv\in E}\|\bar{\mathsf{u}}_{y_u} - \bar{\mathsf{v}}_{y_v}\|_2^2.$$

If (y_u, y_v) satisfies the constraint uv, then

$$\frac{k}{2}\|\bar{\mathsf{u}}_{y_u} - \bar{\mathsf{v}}_{y_v}\|_2^2 = \frac{1}{2}\sum_{i\in[k]}\|\bar{\mathsf{u}}_i - \bar{\mathsf{v}}_{\pi_{uv}(i)}\|_2^2$$

by the third item above. Otherwise,

$$\frac{k}{2}\|\bar{\mathsf{u}}_{y_u} - \bar{\mathsf{v}}_{y_v}\|_2^2 \le 2$$

as $\|\bar{\mathsf{u}}_i\|_2^2 = \frac{1}{k}$ for all $u \in V$ and $i \in [k]$. Since the integral solution y satisfies at least a $(1-\delta)$ fraction of constraints,

$$\frac{1}{2}\mathbb{E}_{uv\in E}[\|f(u) - f(v)\|_2^2] \le 2\delta + \frac{1}{2|E|}\sum_{uv\in E}\sum_{i\in[k]}\|\bar{\mathsf{u}}_i - \bar{\mathsf{v}}_{\pi_{uv}(i)}\|_2^2 \le 2\delta + \epsilon,$$

where the last inequality follows from (20.2) and the assumption that the objective value of the SDP solution $\{\bar{\mathsf{u}}_i\}_{u\in V, i\in[k]}$ is at least $1-\epsilon$. □

The low distortion embedding for general Unique Games in Lemma 20.4 is based on a tensoring operation

$$f(u) = \frac{1}{\sqrt{k}}\sum_{i\in[k]}\|\mathsf{u}_i\|\Big(\frac{\mathsf{u}_i}{\|\mathsf{u}_i\|}\Big)^{\otimes t}$$

and the proof is more involved. Using Lemma 20.4, [AKK+08] showed that the propagation rounding algorithm returns an assignment satisfying at least a

$$1 - O\Big(\frac{\epsilon}{\lambda_2(\mathcal{L}(G))}\log\Big(\frac{\lambda_2(\mathcal{L}(G))}{\epsilon}\Big)\Big)$$

fraction of constraints, which has an additional log factor compared to Theorem 20.1.

We refer the reader to Makarychev–Makarychev [MM10] for an optimal analysis that removes the logarithmic factor for general Unique Games.

References

[AKK+08] Sanjeev Arora, Subhash A Khot, Alexandra Kolla, David Steurer, Madhur Tulsiani, and Nisheeth K Vishnoi. Unique games on expanding constraint graphs are easy. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 21–28, 2008. 255, 256, 259, 260, 262, 263

[GW95] Michel X Goemans and David P Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM (JACM)*, 42(6):1115–1145, 1995. 256, 273

[MM10] Konstantin Makarychev and Yury Makarychev. How to play unique games on expanders. In *International Workshop on Approximation and Online Algorithms*, pages 190–200. Springer, 2010. 262

Chapter 21

Local-to-Global Correlation Rounding on Low Threshold Rank Graphs

In this chapter, we study how to generalize the propagation rounding algorithm in Chapter 20 to low threshold rank graphs, in which there are few small Laplacian eigenvalues (see Definition 13.1).

There are two independent works that generalize the propagation rounding algorithm for expander graphs [AKK+08]. Both use the same algorithm based on the Lasserre SDP hierarchy and propagation rounding, but their analyses differ. The analysis of Guruswami and Sinop [GS11] reveals an interesting connection to the column selection problem for low-rank approximation of matrices. The analysis of Barak, Raghavendra, and Steurer [BRS11] emphasizes the local-to-global correlation phenomenon in low threshold rank graphs, which is the perspective we adopt.

21.1 Conditioning on Lasserre SDP Hierarchy

Low Threshold Rank Graphs: From the discussions in Chapter 7 and Chapter 8, a graph with $\lambda_k(\mathcal{L}(G)) = \Omega(1)$ can be loosely interpreted as admitting a partition of vertices $V_1, \ldots, V_\ell$ with $\ell \leq k$ such that each $G[V_i]$ is an expander graph and there are few edges crossing the partition. Given such a partition and the technique in the previous chapter, one can choose a vertex $u_i \in V_i$ so that the propagation rounding algorithm on $G[V_i]$ from u_i satisfies most of the constraints. This approach, however, incurs a quantitative loss due to the expander decomposition step (see Chapter 12).

Conditioning: The algorithms in [GS11, BRS11] avoid this decomposition step by directly conditioning on the values of a set of $O(k)$ vertices and propagating them to the rest of the graph. To define a consistent assignment to all the variables, they require "higher-order SDP solutions" provided by the Lasserre SDP hierarchy. This hierarchy is closely related to the Sum-of-Squares hierarchy, which has a large and well-developed literature.

Lasserre Hierarchy: For our purposes, we only give a brief description of the Lasserre SDP hierarchy for constraint satisfaction problems and state the properties needed for the analysis without proofs. We refer the reader to [Rot13, BS16] for further background on the Lasserre and Sum-of-Squares hierarchies, with an emphasis on designing approximation algorithms.

Local Distributions and Abstract Relaxations

A systematic way to define increasingly stronger relaxations is to enforce consistency on progressively larger sets of variables.

Local Distributions: For constraint satisfaction problems, let $V = [n]$ be the set of vertices and let $[k]$ be the set of labels. An ℓ-local distribution is a distribution μ_T over assignments $[k]^T$ on a subset $T \subseteq V$ with $|T| \leq \ell + 2$. A collection of ℓ-local distributions $\{\mu_T\}_{T \subseteq V, |T| \leq \ell+2}$ is consistent if for all $T, T' \subseteq V$ with $|T|, |T'| \leq \ell + 2$, the distributions μ_T and $\mu_{T'}$ agree on their intersection $T \cap T'$, that is, all marginal probabilities on $T \cap T'$ coincide.

Local Random Variables: We view these consistent distributions as defining a collection $X_1, \ldots, X_n$ of ℓ-local random variables, so that for any subset $T \subseteq [n]$ with $|T| \leq \ell + 2$, there is a sample space (defined by μ_T) of integral assignments to the variables $\{X_i\}_{i \in T}$, and the marginal probabilities of at most $\ell + 2$ variables, such as $\tilde{\Pr}[X_i = a]$ and $\tilde{\Pr}[X_i = a, X_j = b]$, are consistently defined. We use the notations $\tilde{\Pr}$ and $\tilde{\mathbb{E}}$ to emphasize that these statistics arise from the local distributions, to distinguish them from Pr and $\mathbb{E}$, which refer to statistics under a global distribution.

Abstract Relaxations: Given a Unique Games instance with constraint graph $G = (V, E)$, label set $[k]$, and bijections $\{\pi_{uv}\}_{uv \in E}$, the following is an abstract ℓ-level hierarchy relaxation:

$$\begin{aligned} \max \quad & \frac{1}{|E|} \sum_{ij \in E} \tilde{\Pr}[X_j = \pi_{ij}(X_i)] \\ \text{subject to} \quad & X_1, \ldots, X_n \text{ is a collection of } \ell\text{-local random variables.} \end{aligned} \tag{21.1}$$

There are different ways to find ℓ-local random variables optimizing (21.1), including the Sherali–Adams LP hierarchy and the Lasserre SDP hierarchy. An important algorithmic result that we use without proof is that an ℓ-level hierarchy solution can be computed in $n^{O(\ell+1)}$ time [Rot13, BS16].

Propagation Sampling Algorithm

Given a collection $X_1, \ldots, X_n$ of ℓ-local random variables, the following is a natural algorithm to define an assignment to all the variables, by conditioning on a subset of variables S being fixed to some values $x_S \in [k]^S$.

Algorithm 22 Propagation Sampling Algorithm

Require: A collection of ℓ-local random variables $X_1, \ldots, X_n$ for (21.1).

1: Choose $r \in \{1, \ldots, \ell\}$ uniformly at random.
2: Sample $S \subseteq V$ uniformly at random with $|S| = r$.
3: Sample an assignment $x_S \in [k]^S$ for S according to the local distribution μ_S.
4: For every vertex $i \in V \setminus S$, sample a label $x_i \in [k]$ according to the local distribution $\mu_{S \cup \{i\}}$ conditioned on the assignment x_S for vertices in S.
5: **return** The assignment $x : V \to [k]$.

Both works in [GS11, BRS11] show that this algorithm works for low threshold rank graphs when the collection of ℓ-local random variables is given by the Lasserre SDP hierarchy.

Lasserre Hierarchy

The Lasserre hierarchy for constraint satisfaction problems is a family of increasingly larger SDP relaxations, where the level-ℓ program finds ℓ-local random variables optimizing (21.1).

Geometric Representation: Because of its positive semidefinite constraints, solutions to the level-ℓ program satisfy a stronger structural property: there exists a geometric representation $\{v_{S,\alpha}\}_{S\subseteq V,\,|S|\leq \ell+2,\,\alpha\in[k]^S}$ of the consistent ℓ-local distributions such that for every $S,T\subseteq V$ with $|S\cup T|\leq \ell+2$ and any $\alpha\in[k]^S$ and $\beta\in[k]^T$,

$$\langle v_{S,\alpha}, v_{T,\beta}\rangle = \tilde{\Pr}[X_S=\alpha,\, X_T=\beta].$$

Thus the joint probabilities of all local events are realized as inner products of vectors. In particular, one can verify that the semidefinite program for Unique Games in (20.1) is precisely the level-0 program of the Lasserre hierarchy in (21.1).

Conditioning: Given the ℓ-local random variables, one can use the laws of conditional probability to define the $(\ell-r)$-local random variables after conditioning on the values of a set of r variables. Similarly, given the geometric representation of the ℓ-local random variables, one obtains a corresponding geometric representation of the $(\ell-r)$-local random variables after conditioning. For example, if we condition on a binary variable $X_i=0$, then we update

$$\tilde{\Pr}[X_j=1] \leftarrow \frac{\tilde{\Pr}[X_j=1]-\tilde{\Pr}[X_j=1,\,X_i=1]}{1-\tilde{\Pr}[X_i=1]} \quad\text{and}\quad v_{j,1}\leftarrow \frac{v_{j,1}-v_{\{i,j\},\{1,1\}}}{\sqrt{1-\tilde{\Pr}[X_i=1]}}.$$

One can use the inclusion-exclusion formula to derive the updates after conditioning on a set of variables. The order of conditioning is irrelevant. We refer the reader to [Rot13] for more details.

Covariance Matrix: An important consequence for our analysis is that for every set $S\subseteq V$ with $|S|\leq\ell$ and every local assignment $x_S\in[k]^S$, the covariance matrix after conditioning is positive semidefinite, that is,

$$\big\{\tilde{\mathrm{Cov}}(X_{ia}, X_{jb}\mid X_S=x_S)\big\}_{i,j\in V,a,b\in[k]}\succcurlyeq 0 \tag{21.2}$$

where $\tilde{\mathrm{Cov}}(X,Y)=\tilde{\mathbb{E}}[(X-\tilde{\mathbb{E}}[X])(Y-\tilde{\mathbb{E}}[Y])]$ denotes the covariance of two random variables. In particular, after conditioning on $X_S=x_S$, there exist vectors $\{u_{ia}\}_{i\in V,\,a\in[k]}$ such that

$$\tilde{\mathrm{Cov}}(X_{ia},X_{jb})=\langle u_{ia},u_{jb}\rangle. \tag{21.3}$$

These vectors will be crucial in extending the local-to-global correlation analysis for expander graphs in Section 20.3 to low threshold rank graphs.

Main Theorem

We illustrate the main theorem of [BRS11] in the special case of the Maximum Cut problem.

Theorem 21.1 (Approximating Maximum Cut on Low Threshold Rank Graphs)**.** *Given a graph $G = (V, E)$ and $\epsilon > 0$, there exists an $(1 - O(\epsilon))$-approximation algorithm for the Maximum Cut problem with running time*

$$|V|^{O\left(\frac{1}{\epsilon^{12}} \operatorname{rank}_{1-\epsilon^6}(G)\right)},$$

where $\operatorname{rank}_\delta(G)$ denotes the number of eigenvalues of the normalized Laplacian matrix of G with eigenvalue at most δ (see Definition 13.1). In particular, the propagation sampling algorithm applied to a level-ℓ Lasserre hierarchy solution to (21.1) for $\ell \lesssim \frac{1}{\epsilon^{12}} \operatorname{rank}_{1-\epsilon^6}(G)$ achieves this guarantee.

The structure of the proof is as follows. In Section 21.2, we show that if the local correlation over edges of the SDP solution is small, then independent randomized rounding yields a good approximate solution. In Section 21.3, we show that if the global correlation over pairs of the SDP solution is large, then conditioning reduces the global correlation significantly. Finally, in Section 21.4, we show that large local correlation implies large global correlation in a low threshold rank graph. Combining these statements, we conclude that for a low threshold rank graph, after conditioning on a small number of variables in the SDP solution, the local correlation becomes small, and hence independent randomized rounding succeeds.

21.2 Local Correlation and Independent Randomized Rounding

For the Maximum Cut problem, the objective function in (21.1) becomes

$$\mathsf{sdp}(G) = \frac{1}{|E|} \sum_{ij \in E} \tilde{\Pr}[X_i \neq X_j],$$

where each X_i is a $0/1$ variable indicating whether vertex i is included in the output set $S \subseteq V$. For any vertex $i \in V$ and any pair of vertices $i \neq j \in V$, we denote

$$y_i = \tilde{\Pr}[X_i = 1] = \tilde{\mathbb{E}}[X_i] \quad \text{and} \quad y_{ij} = \tilde{\Pr}[X_i = 1, X_j = 1] = \tilde{\mathbb{E}}[X_i X_j].$$

It follows that

$$\tilde{\Pr}[X_i \neq X_j] = y_i + y_j - 2y_{ij}.$$

Independent Randomized Rounding: Given this distributional view of the SDP solution to (21.1), a simple strategy to define a global distribution over V is to set $X_i = 1$ independently for each vertex $i \in V$ with probability y_i. Under this global distribution, the probability that an edge $ij \in E$ is cut is

$$\Pr[X_i \neq X_j] = y_i(1 - y_j) + y_j(1 - y_i) = y_i + y_j - 2y_i y_j.$$

If $\Pr[X_i \neq X_j] \approx \tilde{\Pr}[X_i \neq X_j]$ for most edges $ij \in E$, then independent randomized rounding would yield an integral solution whose value is close to the SDP objective value in (21.1).

Local Correlation: It is therefore natural to consider the covariance of X_i and X_j under the local distribution,

$$\tilde{\operatorname{Cov}}(X_i, X_j) = \tilde{\mathbb{E}}\big[\big(X_i - \tilde{\mathbb{E}}[X_i]\big)\big(X_j - \tilde{\mathbb{E}}[X_j]\big)\big] = y_{ij} - y_i y_j = \frac{1}{2}\big(\Pr[X_i \neq X_j] - \tilde{\Pr}[X_i \neq X_j]\big),$$

which measures the discrepancy between the independent rounding distribution and the local SDP distribution. We denote the local correlation over edges by

$$\mathbb{E}_{ij\in E}\left[\left|\tilde{\mathrm{Cov}}(X_i, X_j)\right|\right] := \frac{1}{|E|}\sum_{ij\in E}|y_{ij} - y_i y_j|. \tag{21.4}$$

The following lemma formalizes the idea that if the local correlation over edges is small, then independent randomized rounding yields a good approximate solution.

Lemma 21.2 (Local Correlation and Independent Randomized Rounding)**.** *Suppose that for a level-ℓ Lasserre solution to (21.1) for the Maximum Cut problem, the local correlation over edges satisfies*

$$\mathbb{E}_{ij\in E}[|\tilde{\mathrm{Cov}}(X_i, X_j)|] \le 2\epsilon^3.$$

Then independent randomized rounding outputs a set $S \subseteq V$ with

$$\mathbb{E}\left[|\delta(S)|\right] \ge (1 - O(\epsilon)) \cdot \mathit{sdp}(G) \cdot |E|.$$

Proof. Call an edge $ij \in E$ *good* if

$$|\tilde{\mathrm{Cov}}(X_i, X_j)| \le \epsilon^2 \quad \text{and} \quad \tilde{\Pr}[X_i \neq X_j] \ge \epsilon,$$

and *bad* otherwise. Since the local correlation over edges is at most $2\epsilon^3$, Markov's inequality implies that there are at most $2\epsilon|E|$ edges with $|\tilde{\mathrm{Cov}}(X_i, X_j)| > \epsilon^2$. Hence the total contribution of the bad edges to the SDP objective is

$$\sum_{ij\in E:ij \text{ bad}} \tilde{\Pr}[X_i \neq X_j] \le (2\epsilon|E|) \cdot 1 + |E| \cdot \epsilon = 3\epsilon|E| \le 6\epsilon \cdot \mathsf{sdp}(G) \cdot |E|,$$

where the last inequality uses that $\mathsf{sdp}(G) \ge |E|/2$ for the Maximum Cut problem. On the other hand, the contribution of a good edge $ij \in E$ to the objective value is

$$\Pr[X_i \neq X_j] = \tilde{\Pr}[X_i \neq X_j] + 2\,\tilde{\mathrm{Cov}}(X_i, X_j) \ge \tilde{\Pr}[X_i \neq X_j] - 2\epsilon^2 \ge (1 - 2\epsilon) \cdot \tilde{\Pr}[X_i \neq X_j],$$

where the inequalities follow from ij being good. Therefore, by linearity of expectation,

$$\mathbb{E}\left[|\delta(S)|\right] = \sum_{ij\in E}\Pr[X_i \neq X_j] \ge (1 - 2\epsilon) \cdot \sum_{ij\in E:ij \text{ good}} \tilde{\Pr}[X_i \neq X_j] \ge (1 - 2\epsilon)(1 - 6\epsilon) \cdot \mathsf{sdp}(G) \cdot |E|.$$

□

21.3 Global Correlation and Conditioning

The failure of independent randomized rounding implies that there is non-trivial local correlation over edges. The key observation is that the presence of high correlations can also make the sampling of $X_1, \ldots, X_n$ easier. When two random variables are highly correlated (for example, $y_i = y_j = 1/2$ and $y_{ij} = 1/2$ in the highly positively correlated case, or $y_i = y_j = 1/2$ and $y_{ij} = 0$ in the highly negatively correlated case), conditioning on the value of one variable essentially determines the value of the other variable. Thus, a natural strategy is to show that if there is large global correlation over pairs in the SDP solution to (21.1), then there exists a conditioning on a single variable that significantly reduces the uncertainty of many other variables.

Potential Function: To quantify the uncertainty of a random variable, a natural choice[1] is to use the variance

$$\tilde{\mathrm{Var}}(X_i) = \tilde{\Pr}[X_i = 1] \cdot (1 - \tilde{\Pr}[X_i = 1]) = y_i(1 - y_i),$$

which is small if and only if X_i is close to being a 0/1-valued variable. To measure progress, consider the potential function

$$\mathbb{E}_{i \in V}\left[\tilde{\mathrm{Var}}(X_i)\right] = \frac{1}{|V|}\sum_{i \in V} \tilde{\mathrm{Var}}(X_i).$$

Conditioning on Lasserre Hierarchy: To condition on a variable X_j, we set $X_j = 1$ with probability y_j and $X_j = 0$ with probability $1 - y_j$. Based on the outcome of X_j, we update the consistent local distributions of the ℓ-local random variables $X_1, \ldots, X_n$ to consistent local distributions of $(\ell - 1)$-local random variables $X'_1, \ldots, X'_n$ using the laws of conditional probability. The geometric representation $\{v'_{S,\alpha}\}_{S \subseteq V, |S| \leq \ell+1, \alpha \in [k]^S}$ is updated accordingly so that

$$\left\langle v'_{S,\alpha}, v'_{T,\beta} \right\rangle = \tilde{\Pr}\left[X'_S = \alpha, X'_T = \beta\right].$$

A direct calculation shows that the decrease of the potential can be quantified in terms of the covariance.

Claim 21.3 (Potential Decrease)**.** *After conditioning on a variable X_j,*

$$\tilde{\mathbb{E}}_{X_j}\left[\tilde{\mathrm{Var}}(X_i \mid X_j)\right] \leq \tilde{\mathrm{Var}}(X_i) - \frac{1}{4}\tilde{\mathrm{Cov}}(X_i, X_j)^2.$$

Proof. The law of total variance states that

$$\tilde{\mathbb{E}}_{X_j}\left[\tilde{\mathrm{Var}}(X_i \mid X_j)\right] = \tilde{\mathrm{Var}}(X_i) - \tilde{\mathrm{Var}}_{X_j}\left(\tilde{\mathbb{E}}[X_i \mid X_j]\right).$$

Since

$$\tilde{\mathbb{E}}[X_i \mid X_j = 1] = \frac{y_{ij}}{y_j}, \quad \tilde{\mathbb{E}}[X_i \mid X_j = 0] = \frac{y_i - y_{ij}}{1 - y_j}, \quad \tilde{\mathbb{E}}_{X_j}\left[\tilde{\mathbb{E}}[X_i \mid X_j]\right] = \tilde{\mathbb{E}}[X_i] = y_i,$$

it follows from $\mathrm{Var}(Y) = \mathbb{E}\left[Y^2\right] - \mathbb{E}\left[Y\right]^2$ that

$$\tilde{\mathrm{Var}}_{X_j}\left(\tilde{\mathbb{E}}[X_i \mid X_j]\right) = y_j \cdot \left(\frac{y_{ij}}{y_j}\right)^2 + (1 - y_j) \cdot \left(\frac{y_i - y_{ij}}{1 - y_j}\right)^2 - y_i^2 = \frac{(y_i y_j - y_{ij})^2}{y_j(1 - y_j)} = \frac{\tilde{\mathrm{Cov}}(X_i, X_j)^2}{\tilde{\mathrm{Var}}(X_j)}.$$

Therefore, using $\tilde{\mathrm{Var}}(X_j) \leq 1/4$, we obtain

$$\tilde{\mathbb{E}}_{X_j}\left[\tilde{\mathrm{Var}}(X_i \mid X_j)\right] = \tilde{\mathrm{Var}}(X_i) - \frac{\tilde{\mathrm{Cov}}(X_i, X_j)^2}{\tilde{\mathrm{Var}}(X_j)} \leq \tilde{\mathrm{Var}}(X_i) - \frac{1}{4}\tilde{\mathrm{Cov}}(X_i, X_j)^2.$$

□

[1] The mutual information of random variables is used in [RT12] to quantify uncertainty and correlation.

Global Correlation: Given the inequality for potential decrease in Claim 21.3, we denote the global correlation over pairs by

$$\mathbb{E}_{i,j\in V}\left[|\tilde{\mathrm{Cov}}(X_i,X_j)|^2\right] := \frac{1}{|V|^2}\sum_{i,j\in V}|y_{ij}-y_iy_j|^2. \tag{21.5}$$

The following lemma shows that by conditioning on a small number of variables, one can obtain a Lasserre SDP solution with small global correlation.

Lemma 21.4 (Global Correlation and Conditioning)**.** *Given a level-ℓ Lasserre solution to (21.1) for the Maximum Cut problem, one can condition on at most ℓ variables so that the global correlation over pairs satisfies*

$$\mathbb{E}_{i,j\in V}\left[|\tilde{\mathrm{Cov}}(X_i,X_j)|^2\right] \le \frac{1}{\ell}.$$

Proof. Choose a uniformly random vertex j to condition on. By Claim 21.3, the average value of the potential function after conditioning is

$$\begin{aligned}\frac{1}{|V|}\sum_{j\in V}\frac{1}{|V|}\sum_{i\in V}\tilde{\mathbb{E}}_{X_j}\left[\tilde{\mathrm{Var}}(X_i\mid X_j)\right] &\le \frac{1}{|V|^2}\sum_{i,j\in V}\left(\tilde{\mathrm{Var}}(X_i)-\frac{1}{4}\tilde{\mathrm{Cov}}(X_i,X_j)^2\right)\\ &= \mathbb{E}_{i\in V}[\tilde{\mathrm{Var}}(X_i)]-\frac{1}{4}\mathbb{E}_{i,j\in V}\left[|\tilde{\mathrm{Cov}}(X_i,X_j)|^2\right].\end{aligned}$$

Therefore, there exists a vertex j such that the potential function value after conditioning on X_j decreases by at least $\mathbb{E}_{i,j\in V}\left[|\tilde{\mathrm{Cov}}(X_i,X_j)|^2\right]/4$.

Repeat this procedure for ℓ iterations. Since the initial potential function value is at most $1/4$, there must exist at least one iteration at which $\mathbb{E}_{i,j\in V}\left[|\tilde{\mathrm{Cov}}(X_i,X_j)|^2\right]\le 1/\ell$, for otherwise the potential function would become negative, a contradiction. The intermediate SDP solution at that iteration is a level-r Lasserre solution to (21.1) (for some $r\ge 0$) satisfying the global correlation over pairs is at most $1/\ell$. □

21.4 Local-to-Global Correlation of Low Threshold Rank Graphs

On one hand, Lemma 21.4 shows that after conditioning on a small number of variables, the Lasserre SDP solution has small *global* correlation over pairs. On the other hand, Lemma 21.2 implies that if the Lasserre SDP solution has small *local* correlation over edges, then independent randomized rounding succeeds. In Section 20.3, we proved that small global correlation implies small local correlation on an expander graph. In this section, we extend this implication to low threshold rank graphs, thereby completing the proof of the main theorem.

Geometric Representation and Spectral Analysis: To bound the local correlation over edges $\mathbb{E}_{ij\in E}[|\tilde{\mathrm{Cov}}(X_i,X_j)|]$ which involves an absolute value, we apply the Cauchy–Schwarz inequality and instead bound $\mathbb{E}_{ij\in E}[|\tilde{\mathrm{Cov}}(X_i,X_j)|^2]$. Recall from (21.3) that an important property of the Lasserre hierarchy is the existence of a geometric representation $\{u_i\}_{i\in V}$ such that

$$\langle u_i,u_j\rangle = \tilde{\mathrm{Cov}}(X_i,X_j) \quad \text{for } i,j\in V.$$

To bound $\mathbb{E}_{ij\in E}[|\tilde{\mathrm{Cov}}(X_i, X_j)|^2]$, we consider the vectors $v_i := u_i \otimes u_i$ for $i \in V$, so that

$$\langle v_i, v_j\rangle = \left|\tilde{\mathrm{Cov}}(X_i, X_j)\right|^2 \quad \text{for } i, j \in V.$$

Let $C \in \mathbb{R}^{n\times n}$ be the positive semidefinite matrix with $C_{ij} = \langle v_i, v_j\rangle$. For a regular graph G with normalized adjacency matrix $\mathcal{A}$, the local correlation can be bounded as

$$\left(\mathbb{E}_{ij\in E}\left[|\tilde{\mathrm{Cov}}(X_i, X_j)|\right]\right)^2 \le \mathbb{E}_{ij\in E}[|\tilde{\mathrm{Cov}}(X_i, X_j)|^2] = \mathbb{E}_{ij\in E}[C_{ij}] = \frac{1}{|V|}\langle C, \mathcal{A}\rangle,$$

so that it is amenable to spectral analysis. The following lemma thus relates the local and global correlation in a low threshold rank graph. The presentation follows [Hsi26, Lemma 2.4].

Lemma 21.5 (Local-to-Global Correlation of Low Threshold Rank Graphs)**.** *Let $C \in \mathbb{R}^{n\times n}$ be a positive semidefinite matrix with $\mathrm{Tr}(C) \le n$. Let $G = (V, E)$ be a regular graph on n vertices. For any $\delta \in (0, 1)$,*

$$\mathbb{E}_{ij\in E}[C_{ij}] \le \delta\sqrt{\mathrm{rank}_\delta(G)}\cdot\sqrt{\mathbb{E}_{i,j\in V}[C_{ij}^2]} + (1-\delta).$$

Proof. Let $\mathcal{A}$ be the normalized adjacency matrix of G with eigenvalues $1 = \alpha_1 \ge \alpha_2 \ge \cdots \ge \alpha_n$, and let $\mathcal{A} = \sum_{i=1}^n \alpha_i u_i u_i^\top$ be an eigen-decomposition. Let $r := \mathrm{rank}_\delta(G)$. By assumption, the normalized Laplacian matrix of G has at most r eigenvalues at most δ, which is equivalent to $\alpha_{r+1} \le 1-\delta$. Using $\sum_{i=1}^n u_i u_i^\top = I_n$, we bound

$$\mathcal{A} = \sum_{i=1}^n \alpha_i u_i u_i^\top \preccurlyeq \sum_{i=1}^r u_i u_i^\top + \sum_{i=r+1}^n (1-\delta) u_i u_i^\top = \delta\sum_{i=1}^r u_i u_i^\top + (1-\delta) I_n.$$

Since C is positive semidefinite, it follows that

$$\mathbb{E}_{ij\in E}[C_{ij}] = \frac{1}{n}\langle C, \mathcal{A}\rangle \le \frac{\delta}{n}\left\langle C, \sum_{i=1}^r u_i u_i^\top\right\rangle + \frac{1-\delta}{n}\mathrm{Tr}(C) \le \frac{\delta}{n}\|C\|_F\cdot\sqrt{r} + \frac{1-\delta}{n}\mathrm{Tr}(C),$$

where the last inequality uses the Cauchy-Schwarz inequality and $\|\sum_{i=1}^r u_i u_i^\top\|_F = \sqrt{r}$. The statement follows since $\frac{1}{n}\|C\|_F = \sqrt{\frac{1}{n^2}\sum_{i,j\in V} C_{ij}^2} = \sqrt{\mathbb{E}_{i,j\in V}[C_{ij}^2]}$ and $\mathrm{Tr}(C) \le n$ by assumption. □

One consequence of Lemma 21.5 is the following: If each vertex i of a regular graph $G = (V, E)$ is associated with a vector v_i in the unit ball, then

$$\mathbb{E}_{ij\in E}\left[\langle v_i, v_j\rangle\right] \ge \rho \quad \Longrightarrow \quad \mathbb{E}_{i,j\in V}\left[|\langle v_i, v_j\rangle|\right] \ge \mathbb{E}_{i,j\in V}\left[\langle v_i, v_j\rangle^2\right] \gtrsim \rho^2/\mathrm{rank}_{1-\Omega(\rho)}(G).$$

This is quantitatively weaker than the result in [BRS11], which proves $\mathbb{E}_{i,j\in V}\left[\langle v_i, v_j\rangle^2\right] \gtrsim \rho/\mathrm{rank}_{1-\Omega(\rho)}(G)$. However, Lemma 21.5 has a simpler proof and is sufficient for establishing the main theorem.

Proof of Main Theorem

We now combine the pieces to establish Theorem 21.1. By Lemma 21.4, given a level-ℓ Lasserre solution to (21.1) for the Maximum Cut problem, one can condition on at most ℓ variables so that the global correlation over pairs satisfies

$$\mathbb{E}_{i,j\in V}\left[|\tilde{\mathrm{Cov}}(X_i, X_j)|^2\right] \le \frac{1}{\ell}.$$

Let $C \in \mathbb{R}^{n\times n}$ be the positive semidefinite matrix with $C_{ij} = \left|\tilde{\mathrm{Cov}}(X_i, X_j)\right|^2$ and $\mathrm{Tr}(C) = \sum_{i=1}^{n} \mathrm{Var}(X_i) \le n$. Applying the local-to-global analysis in Lemma 21.5 to C yields

$$\mathbb{E}_{ij\in E}[|\tilde{\mathrm{Cov}}(X_i, X_j)|^2] \le \delta\sqrt{\mathrm{rank}_\delta(G)}\cdot\sqrt{\mathbb{E}_{i,j\in V}[|\tilde{\mathrm{Cov}}(X_i, X_j)|^4]} + (1-\delta) \le \frac{\delta}{\sqrt{\ell}}\sqrt{\mathrm{rank}_\delta(G)} + (1-\delta),$$

where we use $|\tilde{\mathrm{Cov}}(X_i, X_j)|^4 \le |\tilde{\mathrm{Cov}}(X_i, X_j)|^2$. Setting

$$\delta = 1-\epsilon^6 \quad \text{and} \quad \ell = \frac{\mathrm{rank}_\delta(G)}{\epsilon^{12}},$$

it follows that

$$\mathbb{E}_{ij\in E}\left[|\tilde{\mathrm{Cov}}(X_i, X_j)|\right] \le \sqrt{\mathbb{E}_{ij\in E}\left[|\tilde{\mathrm{Cov}}(X_i, X_j)|^2\right]} \le 2\epsilon^3.$$

Hence, applying the independent randomized rounding algorithm in Lemma 21.2 gives an $(1-O(\epsilon))$-approximation algorithm for the Maximum Cut problem.

References

[AKK+08] Sanjeev Arora, Subhash A Khot, Alexandra Kolla, David Steurer, Madhur Tulsiani, and Nisheeth K Vishnoi. Unique games on expanding constraint graphs are easy. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 21–28, 2008. 255, 256, 259, 260, 262, 263

[BRS11] Boaz Barak, Prasad Raghavendra, and David Steurer. Rounding semidefinite programming hierarchies via global correlation. In *2011 IEEE 52nd Annual Symposium on Foundations of Computer Science*, pages 472–481. IEEE, 2011. 263, 264, 265, 270, 273

[BS16] Boaz Barak and David Steurer. Proofs, beliefs, and algorithms through the lens of sum-of-squares. *Lecture Notes*, 2016. 263, 264, 279

[GS11] Venkatesan Guruswami and Ali Kemal Sinop. Lasserre hierarchy, higher eigenvalues, and approximation schemes for graph partitioning and quadratic integer programming with psd objectives. In *2011 IEEE 52nd Annual Symposium on Foundations of Computer Science*, pages 482–491. IEEE, 2011. 263, 264, 273

[Hsi26] Jun-Ting Hsieh. Coloring 3-colorable graphs with low threshold rank. In *Proceedings of the 2026 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 718–726. SIAM, 2026. 270, 273

[Rot13] Thomas Rothvoß. The lasserre hierarchy in approximation algorithms. *Lecture Notes for the MAPSP*, pages 1–25, 2013. 263, 264, 265

[RT12] Prasad Raghavendra and Ning Tan. Approximating csps with global cardinality constraints using sdp hierarchies. In *Proceedings of the twenty-third annual ACM-SIAM symposium on Discrete Algorithms*, pages 373–387. SIAM, 2012. 268, 273

Chapter 22

Local-to-Global Correlation Rounding on General Graphs?

The rounding techniques developed in [BRS11, GS11] can be used to obtain sub-exponential time algorithms for distinguishing the two cases in the Unique Games conjecture (together with Theorem 11.3), and to design approximation algorithms for constraint satisfaction and graph partitioning problems in low threshold rank graphs. Subsequent works sharpened this approach on small-set expanders and low threshold rank graphs for Unique Games [BBK+21], constraint satisfaction problems [RT12, AJT19], graph partitioning problems [GS13, AGS13], and graph coloring problems [AG11, BHK25, BHSVK25, Hsi26].

It has been an important open direction to understand whether this approach can be extended to general graphs to obtain subexponential time algorithms that outperform the best known polynomial time approximation algorithms. One outstanding question along this direction is whether there are subexponential time algorithms for Maximum Cut that achieve a better approximation guarantee than the Goemans–Williamson algorithm [GW95].

In this chapter, we discuss a potential approach based on conjectures about reweighting graphs to have fast mixing time or low threshold rank, similar to the settings in Chapter 9 and Chapter 10. If true, these conjectures would extend the reach of the global correlation rounding framework to general graphs and lead to improved subexponential time algorithms for Sparsest Cut and Graph Coloring.

22.1 Toward Subexponential Algorithm for Sparsest Cut

The approach for Sparsest Cut is based on the following geometric conjecture proposed in Steurer's thesis [Ste10, Conjecture 9.2], which states that if the global correlation of a collection of unit vectors is polynomially small, then there exist two sets of linear size that are $\Omega(1)$-separated.

Conjecture 22.1 (Steurer's Conjecture)**.** *For every $\epsilon > 0$, there exist positive constants $\eta = \eta(\epsilon)$ and $\delta = \delta(\epsilon)$ such that the following holds. For every collection of unit vectors $v_1, \dots, v_n \in \mathbb{R}^n$ with $\mathbb{E}_{i,j\in[n]}|\langle v_i, v_j\rangle| \le n^{-\epsilon}$, there are two sets $L, R \subseteq [n]$ with $|L|, |R| \ge \delta n$ such that $\|v_i - v_j\|_2^2 \ge 2\eta$ for all $i \in L$ and $j \in R$.*

One can interpret Conjecture 22.1 as a strengthening of Theorem 19.2 by Arora–Rao–Vazirani: Conjecture 22.1 guarantees two linear-sized subsets with distance $\Omega(1)$ under the strong assumption

that almost every pair of vectors is nearly orthogonal; Theorem 19.2 guarantees two linear-sized subsets with distance $\Omega(1/\sqrt{\log n})$ under the assumption that the vectors satisfy the ℓ_2^2 triangle inequalities. The significance of the conjecture is the following consequence.

Claim 22.2 (Subexponential Algorithm for Sparsest Cut)**.** *An algorithmic proof of Steurer's conjecture implies a subexponential-time constant-factor approximation algorithm for Sparsest Cut.*

Proof Sketch. The vectors in the assumption of Conjecture 22.1 can be obtained by solving a level-$n^{O(\epsilon)}$ Lasserre relaxation and then conditioning on $n^{O(\epsilon)}$ variables to decrease the global correlation $\mathbb{E}_{i,j\in[n]}|\langle v_i, v_j\rangle| \leq n^{-\epsilon}$ as in Lemma 21.4. Once we find the two large and well-separated sets L and R guaranteed by Conjecture 22.1, we can apply a threshold rounding algorithm as in Problem 19.17 to return a constant-factor approximation to Sparsest Cut. □

In the remainder of this section, we reduce Steurer's conjecture to conjectures about reweighting graphs with good vertex expansion so that they have low threshold rank or small mixing time. This reduction is based on unpublished work of Steurer.

Geometric Proximity Graph and Vertex Expander

Given the unit vectors $v_1, \ldots, v_n$ in Conjecture 22.1, we construct a geometric proximity graph G with vertex set $[n]$, where there is an edge ij if and only if

$$\|v_i - v_j\|_2^2 \leq 2\eta \quad \iff \quad \langle v_i, v_j\rangle \geq 1 - \eta.$$

By construction, if there are two disjoint subsets of vertices $L, R \subseteq V(G)$ such that $|L|, |R| \geq \delta n$ and there are no edges between L and R in G, then the vectors corresponding to L and R satisfy the conclusion of Conjecture 22.1.

The following combinatorial argument shows that if such sets L and R do not exist, then there exists a large induced vertex expander in G. Recall the definition of vertex expansion ψ from Definition 2.11.

Lemma 22.3 (Large Induced Vertex Expander or Large Separated Sets)**.** *Let $G = (V, E)$ be an undirected graph. At least one of the following holds:*

1. *There exists a subset of vertices $U \subseteq V$ such that $|U| \geq \frac{3}{4}|V|$ and $\psi(G[U]) \geq \frac{1}{4}$.*
2. *There are disjoint sets $L, R \subseteq V$ with $|L|, |R| \geq \frac{1}{5}|V|$ such that $E(L, R) = \emptyset$.*

Proof. Consider the following procedure.

Algorithm 23 Removing Small Non-Expanding Sets

```
Set V_0 := V.
while |V_i| ≥ (3/4)|V| and there exists a set S_i ⊆ V_i such that |S_i| ≤ (1/2)|V_i| and ψ_{G[V_i]}(S_i) ≤ 1/4 do
    Set V_{i+1} := V_i − S_i and i ← i + 1.
end while
```

There are two cases in which this procedure stops. The first case is when $|V_i| \geq \frac{3}{4}|V|$ but there is no subset $S_i \subseteq V_i$ with $|S_i| \leq \frac{1}{2}|V_i|$ and $\psi_{G[V_i]}(S_i) \leq \frac{1}{4}$. In this case, we let $U := V_i$ and $G[U]$ clearly satisfies the guarantees in case (1) of the lemma.

The second case is when $|V_i| < \frac{3}{4}|V|$. In this case, we let $L := S_0 \cup \cdots \cup S_{i-1}$ and $R := V - L - N_G(L)$. By construction, there are no edges between L and R. Since $|V_i| < \frac{3}{4}|V|$ and $|V_{i-1}| \geq \frac{3}{4}|V|$,

$$|L| = |V| - |V_i| > \frac{1}{4}|V| \quad \text{and} \quad |L| = |V| - |V_{i-1}| + |S_{i-1}| \leq |V| - \frac{1}{2}|V_{i-1}| \leq \frac{5}{8}|V|.$$

Observe that $|N_G(L)| \leq \frac{1}{4}|L|$ by the vertex expansion of S_j for each $0 \leq j \leq i-1$. Therefore,

$$|R| = |V| - |L| - |N_G(L)| \geq |V| - \frac{5}{4}|L| \geq |V| - \frac{5}{4} \cdot \frac{5}{8}|V| = \frac{7}{32}|V|.$$

This shows that L and R satisfy the guarantees in case (2) of the lemma. □

Note that this lemma holds for all graphs, not just for geometric proximity graphs. We also remark that the lemma can be slightly strengthened to show that the induced vertex expander has larger small-set vertex expansion. Since we do not know whether this strengthened version is needed, we state the simpler version and focus on illustrating its connection to the reweighting conjecture in the next subsection.

Reweighting Conjectures

Given the geometric proximity graph $G = (V, E)$, if there are two large well-separated sets L and R as in Case (2) of Lemma 22.3, then Conjecture 22.1 holds. Henceforth, we assume that there is a large induced vertex expander $G[U]$ in G as in Case (1) of Lemma 22.3, and we aim to derive a contradiction to the global correlation assumption in Conjecture 22.1.

The observation is that large local correlation $\langle v_i, v_j \rangle \geq 1 - \epsilon$ holds for every edge $ij \in E$ in the geometric proximity graph G, and thus it also holds for every edge reweighting of G (as defined in Chapter 9 and Chapter 10). The key idea is that if there exists an edge reweighting $G'[U]$ of $G[U]$ such that $G'[U]$ has low threshold rank or fast mixing time, then a local-to-global analysis (similar to that in Lemma 21.5) applied to the reweighted graph $G'[U]$ would imply that the global correlation $\mathbb{E}_{i,j \in [n]} |\langle v_i, v_j \rangle|$ is large, contradicting the assumption in Conjecture 22.1.

The following reweighting conjectures were made with this motivation.

Fast Mixing Time: This formulation is very similar to the fastest mixing time problem in Chapter 9, but the key point is the *logarithmic* mixing time bound.

Conjecture 22.4 (Reweighting of Vertex Expander to Graph with Logarithmic Mixing Time)**.** *Let $H = (V, E)$ be a graph with self-loops and vertex expansion $\psi(H) \geq \frac{1}{4}$. There exists a reweighted matrix $P \in \mathbb{R}^{V \times V}$ of H with the following properties:*

1. *(Reweighting:) $P(i, j) = 0$ for all $ij \notin E$;*
2. *(Symmetric:) $P(i, j) = P(j, i)$ for all $ij \in E$;*
3. *(Row Stochastic:) $\sum_{j \in V} P(i, j) = 1$ for all $i \in V$;*
4. *(Positive Semidefinite:) $P \succcurlyeq 0$;*
5. *(Logarithmic Mixing Time:) $\mathrm{Tr}(P^{2t}) - 1 \leq |V|^{o(1)}$ for $t \gtrsim \log |V|$.*

The first three properties are the same as in Definition 9.1 for the maximum reweighted second eigenvalue problem, together with the assumption that the graph H has a self-loop at each vertex so that the first three properties are always satisfiable. If P is a random walk matrix of H, then the positive semidefinite property can be satisfied by using the lazy random walk matrix $\frac{1}{2}(I+P)$.

To see that the last property is related to fast mixing time, note that

$$\sum_{i,j\in V}\left(P^t(i,j)-\frac{1}{n}\right)^2=\sum_{i,j\in V}\left(P^t(i,j)^2-\frac{2}{n}P^t(i,j)+\frac{1}{n^2}\right)=\|P^t\|_F^2-2+1=\mathrm{Tr}(P^{2t})-1. \quad (22.1)$$

The quantity on the left-hand side measures how close the distributions $P(i,\cdot)$ are to the uniform distribution $\frac{\vec{1}}{n}$. For example, as shown in Theorem 3.15, for a graph with spectral gap $g \gtrsim 1$, its lazy random walk matrix satisfies $\sum_{j\in V}(P^t(i,j)-\frac{1}{n})^2 \le (1-g)^t \lesssim \frac{1}{n}$ for all $i \in V$ when $t \gtrsim \log n$, and therefore $\mathrm{Tr}(P^{2t})-1 \lesssim 1$ when $t \gtrsim \log n$.

We remark that the Cheeger inequality for vertex expansion in Theorem 9.3 implies that a graph H with vertex expansion $\psi(H) \ge \frac{1}{4}$ can be reweighted to have mixing time $O(\log^2 n)$, but this has no implications for Steurer's conjecture. The stronger $O(\log n)$ mixing time bound is crucial here, as we will see at the end of this section.

Low Threshold Rank: Another formulation is to require that the reweighting to have low threshold rank (see Definition 13.1).

Conjecture 22.5 (Reweighting of Vertex Expander to Graph with Low Threshold Rank)**.** *Let $H=(V,E)$ be a graph with self-loops and vertex expansion $\psi(H) \ge \frac{1}{4}$. There exists a reweighting matrix $P \in \mathbb{R}^{V\times V}$ of H satisfying the first four properties in Conjecture 22.4 and*

5. *(Low Threshold Rank:) P has at most $k \le n^{o(1)}$ eigenvalues larger than $\frac{1}{4}$.*

Note that the conclusion in Conjecture 22.5 implies that $\mathrm{Tr}(P^{2t})-1 \le |V|^{o(1)}$ when $t \gtrsim \log|V|$, and thus Conjecture 22.5 implies Conjecture 22.4.

A Local-to-Global Analysis Using Reweighting

Now we relate the local correlation to the global correlation using the reweighted matrix P. In the following lemma, think of $v_1,\dots,v_n$ as the unit vectors in Conjecture 22.1, and P as the reweighted matrix of its geometric proximity graph. The local correlation with respect to P is defined as

$$\mathbb{E}_{ij\sim P}[\langle v_i,v_j\rangle] := \frac{1}{n}\sum_{i,j\in[n]} P(i,j)\langle v_i,v_j\rangle,$$

which corresponds to picking a random edge ij by first choosing a vertex i uniformly at random and then choosing a random neighbor j of i according to the distribution $P(i,\cdot)$. The global correlation is defined as

$$\mathbb{E}_{i,j\in[n]}[\langle v_i,v_j\rangle] := \frac{1}{n^2}\sum_{i,j\in[n]}\langle v_i,v_j\rangle,$$

which corresponds to picking a pair of vertices i,j uniformly at random.

Lemma 22.6 (Local-to-Global Correlation Using Reweighting). *Let $v_1, \ldots, v_n$ be unit vectors and $P \in \mathbb{R}^{n \times n}$ be a doubly stochastic positive semidefinite matrix. For any integer $t \geq 1$,*

$$\left(\mathbb{E}_{ij \sim P}[\langle v_i, v_j \rangle]\right)^t \leq \mathbb{E}_{i,j \in [n]}[\langle v_i, v_j \rangle] + \sqrt{\mathbb{E}_{i,j \in [n]}[\langle v_i, v_j \rangle^2]} \cdot \sqrt{\mathrm{Tr}(P^{2t}) - 1}.$$

Proof. The idea is to compare P^t with the uniform distribution. Consider

$$\begin{aligned}
\left|\mathbb{E}_{i,j \in [n]}[\langle v_i, v_j \rangle] - \mathbb{E}_{ij \sim P^t}[\langle v_i, v_j \rangle]\right| &= \left|\frac{1}{n} \sum_{i,j \in [n]} \langle v_i, v_j \rangle \left(P^t(i,j) - \frac{1}{n}\right)\right| \\
&\leq \sqrt{\frac{1}{n^2} \sum_{i,j \in [n]} \langle v_i, v_j \rangle^2} \cdot \sqrt{\sum_{i,j \in [n]} \left(P^t(i,j) - \frac{1}{n}\right)^2} \\
&= \sqrt{\mathbb{E}_{i,j \in [n]}[\langle v_i, v_j \rangle^2]} \cdot \sqrt{\mathrm{Tr}(P^{2t}) - 1},
\end{aligned}$$

where the inequality is by Cauchy-Schwarz and the last equality follows from (22.1). The lemma then follows from Claim 22.7, as

$$\begin{aligned}
\left(\mathbb{E}_{ij \sim P}[\langle v_i, v_j \rangle]\right)^t - \mathbb{E}_{i,j \in [n]}[\langle v_i, v_j \rangle] &\leq \mathbb{E}_{ij \sim P^t}[\langle v_i, v_j \rangle] - \mathbb{E}_{i,j \in [n]}[\langle v_i, v_j \rangle] \\
&\leq \sqrt{\mathbb{E}_{i,j \in [n]}[\langle v_i, v_j \rangle^2]} \cdot \sqrt{\mathrm{Tr}(P^{2t}) - 1}.
\end{aligned}$$

Rearranging proves the lemma. It remains to prove the following claim.

Claim 22.7 (Power Inequality). *Under the assumptions of the lemma, for any integer $t \geq 1$,*

$$\mathbb{E}_{ij \sim P^t}[\langle v_i, v_j \rangle] \geq \left(\mathbb{E}_{ij \sim P}[\langle v_i, v_j \rangle]\right)^t.$$

Proof. Let $V \in \mathbb{R}^{n \times n}$ be the matrix whose columns are $v_1, \ldots, v_n$. Denote the rows of V by $w_1^\top, \ldots, w_n^\top$. Then

$$\mathbb{E}_{ij \sim P}[\langle v_i, v_j \rangle] = \frac{1}{n} \langle P, V^\top V \rangle = \frac{1}{n} \mathrm{Tr}\left(PV^\top V\right) = \frac{1}{n} \mathrm{Tr}\left(VPV^\top\right) = \frac{1}{n} \sum_{i=1}^n w_i^\top P w_i.$$

Let $P = \sum_{i=1}^n \lambda_i u_i u_i^\top$ be an eigen-decomposition. Then

$$\frac{1}{n} \sum_{i=1}^n w_i^\top P w_i = \frac{1}{n} \sum_{i=1}^n \sum_{j=1}^n \lambda_j \langle w_i, u_j \rangle^2 = \sum_{j=1}^n \lambda_j \left(\frac{1}{n} \sum_{i=1}^n \langle w_i, u_j \rangle^2\right) = \sum_{j=1}^n \lambda_j q_j,$$

where $q_j := \frac{1}{n} \sum_{i=1}^n \langle w_i, u_j \rangle^2$. As $u_1, \ldots, u_n$ form an orthonormal basis, it follows that

$$\sum_{j=1}^n q_j = \frac{1}{n} \sum_{i=1}^n \sum_{j=1}^n \langle w_i, u_j \rangle^2 = \frac{1}{n} \sum_{i=1}^n \| w_i \|_2^2 = \frac{1}{n} \sum_{i=1}^n \| v_i \|_2^2 = 1,$$

and thus $q_1, \ldots, q_n$ form a probability distribution. Therefore, we can apply Jensen's inequality to conclude that

$$\left(\mathbb{E}_{ij \sim P}[\langle v_i, v_j \rangle]\right)^t = \left(\sum_{j=1}^n \lambda_j q_j\right)^t \leq \sum_{j=1}^n \lambda_j^t q_j = \mathbb{E}_{ij \sim P^t}[\langle v_i, v_j \rangle],$$

where the last equality follows from the same argument used to derive $\mathbb{E}_{ij \sim P}[\langle v_i, v_j \rangle] = \sum_{j=1}^n \lambda_j q_j$. □

□

Reduction from Steurer's Conjecture to Reweighting Conjecture

We now put the pieces together to show that the reweighting conjectures imply Steurer's conjecture.

Conjecture 22.4 implies Conjecture 22.1: The plan is to consider a counterexample to Conjecture 22.1 and derive a contradiction to Conjecture 22.4. Suppose $v_1, \dots, v_n \in \mathbb{R}^n$ are unit vectors satisfying $\mathbb{E}_{i,j\in[n]}|\langle v_i, v_j\rangle| \le n^{-\epsilon}$, but for any constants δ and η there are no two subsets $L, R \subseteq [n]$ with $|L|, |R| \ge \delta n$ such that $\|v_i - v_j\|_2^2 \ge 2\eta$ for every $i \in L$ and $j \in R$.

Let G be the geometric proximity graph of $v_1, \dots, v_n$, where there is an edge ij in G if and only if

$$\|v_i - v_j\|_2^2 < 2\eta \quad \iff \quad \langle v_i, v_j\rangle > 1 - \eta,$$

for some sufficiently small constant η to be specified later. By construction, there are no disjoint subsets $L, R \subseteq V(G)$ with $|L|, |R| \ge \frac{1}{5}n$ such that there are no edges between L and R. Therefore, by Lemma 22.3, there is an induced subgraph $H = (V, E)$ of G, with $|V| \ge \frac{3}{4}n$ and $\psi(H) \ge \frac{1}{4}$.

Hence, by Conjecture 22.4, there is a reweighted matrix P of H such that $\mathrm{Tr}(P^{2t}) - 1 \le n^{o(1)}$ for $t \ge c\log n$, where c is a constant. Since $\langle v_i, v_j\rangle > 1 - \eta$ for every edge $ij \in H$, the local correlation with respect to P satisfies

$$\mathbb{E}_{ij\sim P}[\langle v_i, v_j\rangle] > 1 - \eta.$$

The local-to-global analysis in Lemma 22.6 therefore implies that

$$(1-\eta)^t < \left(\mathbb{E}_{ij\sim P}[\langle v_i, v_j\rangle]\right)^t \le \mathbb{E}_{i,j\in V(H)}[\langle v_i, v_j\rangle] + \sqrt{\mathbb{E}_{i,j\in V(H)}[\langle v_i, v_j\rangle^2]} \cdot \sqrt{\mathrm{Tr}(P^{2t}) - 1}.$$

Since $\mathbb{E}_{i,j\in[n]}|\langle v_i, v_j\rangle|^2 \le \mathbb{E}_{i,j\in[n]}|\langle v_i, v_j\rangle| \le n^{-\epsilon}$ and $|V(H)| \ge \frac{3}{4}n$, by Markov's inequality the global correlation in H satisfies

$$\mathbb{E}_{i,j\in V(H)}|\langle v_i, v_j\rangle|^2 \le \mathbb{E}_{i,j\in V(H)}|\langle v_i, v_j\rangle| \le \frac{16}{9}n^{-\epsilon}.$$

Therefore, by setting $t = c\log n$, the local-to-global analysis implies that

$$n^{-2\eta c} = e^{-2\eta t} \le (1-\eta)^t \lesssim n^{-\epsilon} + n^{-\frac{\epsilon}{2}} \cdot n^{o(1)},$$

which is a contradiction if $\eta \le \frac{\epsilon}{8c}$ for sufficiently large n. To summarize, if Conjecture 22.4 is true, then there exist two subsets L, R with $|L|, |R| \ge \frac{1}{5}n$ and $\|v_i - v_j\|_2^2 \ge \frac{\epsilon}{4c}$ for every $i \in L$ and $j \in R$, where c is a constant, establishing Conjecture 22.1.

22.2 Toward Subexponential Algorithm for Other Problems

Graph Coloring: Following a similar approach as in Section 22.1, Arora and Ge [AG11] formulated a reweighting conjecture about graphs with near-perfect vertex expansion.

Conjecture 22.8 (Reweighting Conjecture for Graph Coloring)**.** *Let $H = (V, E)$ be a graph such that*

$$\psi(S) \ge \left(\frac{|V|}{|S|}\right)^{\frac{1}{c}}$$

for every subset $S \subseteq V$ for some constant $c > 1$. There exists a reweighted matrix $P \in \mathbb{R}^{V\times V}$ of H with the following properties:

1. *(Reweighting:)* $P(i,j) = 0$ *for all* $ij \notin E$;
2. *(Symmetric:)* $P(i,j) = P(j,i)$ *for all* $ij \in E$;
3. *(Row Stochastic:)* $\sum_{j \in V} P(i,j) = 1$ *for all* $i \in V$;
4. *(Low Threshold Rank:)* P *has at most* $k \leq n^{o(1)}$ *eigenvalues smaller than* $-\frac{1}{16}$.

Combining with Blum's combinatorial techniques, they showed that if there is an algorithmic proof of the conjecture, then there is an algorithm that finds a $O(n^{\frac{1}{c}} \log n)$ coloring of a 3-colorable graph in $n^{O(k)} \leq n^{n^{o(1)}}$ time. This would improve on the best known guarantees, which find a $O(n^{0.19539})$-coloring [KTY24, BHL26] of a 3-colorable graph in polynomial time.

We remark that the 3-coloring problem can be reduced to the case where every edge has large local correlation, and thus the reweighting idea can be applied to establish global correlation, as in the case of the geometric proximity graph for Steurer's conjecture.

Semi-Random Problems: It has been an important open direction to understand whether the global correlation rounding approach can be extended to general graphs to obtain subexponential time algorithms that outperform the best known polynomial time approximation algorithms, but to date no such algorithms are known for worst-case approximation guarantees.

Interestingly, it turns out that the Lasserre hierarchy and Sum-of-Squares algorithms are proven to be powerful for "semi-random" instances, yielding subexponential time algorithms that outperform the best known polynomial time approximation algorithms and provide a nice time/approximation tradeoff [BKS14, BS16].

In the next topic, we study spectral theory for random matrices and some novel spectral algorithms that match the performance guarantees of these SDP-based algorithms.

References

[AG11] Sanjeev Arora and Rong Ge. New tools for graph coloring. In *International Workshop on Approximation Algorithms for Combinatorial Optimization*, pages 1–12. Springer, 2011. 273, 278

[AGS13] Sanjeev Arora, Rong Ge, and Ali Kemal Sinop. Towards a better approximation for sparsest cut? In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pages 270–279. IEEE, 2013. 273

[AHRS25] Prashanti Anderson, Samuel B Hopkins, Amit Rajaraman, and David Steurer. Faster max-cut on bounded threshold rank graphs. *arXiv preprint arXiv:2511.11499*, 2025.

[AJT19] Vedat Levi Alev, Fernando Granha Jeronimo, and Madhur Tulsiani. Approximating constraint satisfaction problems on high-dimensional expanders. In *2019 IEEE 60th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 180–201. IEEE, 2019. 273

[BBK+21] Mitali Bafna, Boaz Barak, Pravesh K Kothari, Tselil Schramm, and David Steurer. Playing unique games on certified small-set expanders. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 1629–1642, 2021. 273

[BHK25] Mitali Bafna, Jun-Ting Hsieh, and Pravesh K Kothari. Rounding large independent sets on expanders. In *Proceedings of the 57th Annual ACM Symposium on Theory of Computing*, pages 631–642, 2025. 273

[BHL26] Nikhil Bansal, Neng Huang, and Euiwoong Lee. Improved sdp-based algorithm for coloring 3-colorable graphs. *arXiv preprint arXiv:2602.05904*, 2026. 279

[BHSVK25] Rares-Darius Buhai, Yiding Hua, David Steurer, and Andor Vári-Kakas. Finding colorings in one-sided expanders. In *FOCS 2025*, 2025. 273

[BKS14] Boaz Barak, Jonathan A Kelner, and David Steurer. Rounding sum-of-squares relaxations. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, pages 31–40, 2014. 279

[BRS11] Boaz Barak, Prasad Raghavendra, and David Steurer. Rounding semidefinite programming hierarchies via global correlation. In *2011 IEEE 52nd Annual Symposium on Foundations of Computer Science*, pages 472–481. IEEE, 2011. 263, 264, 265, 270, 273

[BS16] Boaz Barak and David Steurer. Proofs, beliefs, and algorithms through the lens of sum-of-squares. *Lecture Notes*, 2016. 263, 264, 279

[GS11] Venkatesan Guruswami and Ali Kemal Sinop. Lasserre hierarchy, higher eigenvalues, and approximation schemes for graph partitioning and quadratic integer programming with psd objectives. In *2011 IEEE 52nd Annual Symposium on Foundations of Computer Science*, pages 482–491. IEEE, 2011. 263, 264, 273

[GS13] Venkatesan Guruswami and Ali Kemal Sinop. Approximating non-uniform sparsest cut via generalized spectra. In *Proceedings of the twenty-fourth annual ACM-SIAM symposium on Discrete algorithms*, pages 295–305. SIAM, 2013. 273

[GW95] Michel X Goemans and David P Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM (JACM)*, 42(6):1115–1145, 1995. 256, 273

[Hsi26] Jun-Ting Hsieh. Coloring 3-colorable graphs with low threshold rank. In *Proceedings of the 2026 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 718–726. SIAM, 2026. 270, 273

[KTY24] Ken-ichi Kawarabayashi, Mikkel Thorup, and Hirotaka Yoneda. Better coloring of 3-colorable graphs. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing*, pages 331–339, 2024. 279

[OGT13] Shayan Oveis Gharan and Luca Trevisan. A new regularity lemma and faster approximation algorithms for low threshold rank graphs. In *International Workshop on Approximation Algorithms for Combinatorial Optimization*, pages 303–316. Springer, 2013.

[Rot13] Thomas Rothvoß. The lasserre hierarchy in approximation algorithms. *Lecture Notes for the MAPSP*, pages 1–25, 2013. 263, 264, 265

[RT12] Prasad Raghavendra and Ning Tan. Approximating csps with global cardinality constraints using sdp hierarchies. In *Proceedings of the twenty-third annual ACM-SIAM symposium on Discrete Algorithms*, pages 373–387. SIAM, 2012. 268, 273

[Ste10] David Steurer. *On the complexity of unique games and graph expansion*. Princeton University, 2010. 273

Part VI

Matrix Concentration and Semi-Random Problems

The study of the spectrum of random matrices is a classical and beautiful area in mathematics. In this part, we introduce fundamental matrix concentration inequalities, study their applications to CSP refutation, and present recent developments based on Kikuchi matrices for designing spectral algorithms for semi-random problems and beyond.

Chapter 23

Matrix Concentration Inequalities

In this chapter, we begin with a proof of the Chernoff–Hoeffding bound in the scalar setting using the moment generating function, and introduce the matrix exponential and logarithm to generalize it to the matrix setting. We then study the approaches of Ahlswede and Winter [AW02], using the Golden–Thompson inequality, and of Tropp [Tro15], using Lieb's concavity theorem, to prove matrix concentration inequalities. We present the proofs of the matrix Chernoff bounds and the matrix Bernstein inequality as instantiations of their framework. Finally, we present the matrix Khintchine inequality and discuss an alternative, more elementary, approach to deriving matrix concentration inequalities. We conclude with recent progress on obtaining sharper matrix concentration inequalities.

This chapter can be treated as a reference chapter and is more technical than the first chapter in other topics. The reader may wish to first explore the applications of matrix concentration inequalities in the subsequent chapters and return later to the proofs.

23.1 Chernoff Bounds

The classical Chernoff–Hoeffding bound states that if $X = \sum_{i=1}^{m} X_i$ is a sum of independent "small" random variables, then X is sharply concentrated around its mean, with exponentially decaying tail probability.

Theorem 23.1 (Chernoff–Hoeffding Bound)**.** *Let $X_1, X_2, \ldots, X_m$ be independent random variables such that $X_i \in [0,1]$ for each i. Let $X = \sum_{i=1}^{m} X_i$ and $\mu_{\min} \le \mathbb{E}[X] \le \mu_{\max}$. For any $\delta > 0$,*

$$\Pr[X \ge (1+\delta)\mu_{\max}] \le \left(\frac{e^{\delta}}{(1+\delta)^{1+\delta}}\right)^{\mu_{\max}} \quad \text{and} \quad \Pr[X \le (1-\delta)\mu_{\min}] \le \left(\frac{e^{-\delta}}{(1-\delta)^{1-\delta}}\right)^{\mu_{\min}}.$$

We do not provide a full proof of this result, as it can be found in many textbooks. Instead, we highlight the three main steps of the proof, as these will be extended to the matrix setting.

The key idea is to consider the exponential of the random variable. For any $\theta \ge 0$,

$$\Pr(X \ge t) = \Pr\left(e^{\theta X} \ge e^{\theta t}\right) \le \frac{\mathbb{E}\left[e^{\theta X}\right]}{e^{\theta t}}, \tag{23.1}$$

where the last inequality follows from Markov's inequality. This transformation allows us to establish exponentially small tail probabilities, provided we obtain a good upper bound on $\mathbb{E}\left[e^{\theta X}\right]$.

The function $M_X(\theta) = \mathbb{E}\left[e^{\theta X}\right]$ is called the moment generating function of the random variable X, since the moments $\mathbb{E}\left[X^k\right]$ can be read off from the coefficients of the Taylor expansion of $M_X(\theta)$. An important advantage of this function is that it can be computed easily when X is a sum of independent random variables:

$$\mathbb{E}\left[e^{\theta X}\right] = \mathbb{E}\left[e^{\theta(X_1+\ldots+X_m)}\right] = \mathbb{E}\left[e^{\theta X_1}\cdots e^{\theta X_m}\right] = \prod_{i=1}^{m}\mathbb{E}\left[e^{\theta X_i}\right]. \tag{23.2}$$

To upper bound $\mathbb{E}\left[e^{\theta X_i}\right]$, we use the convexity of $e^{\theta x}$ and upper bound it by the chord connecting the two endpoints $\{0,1\}$, which yields

$$\mathbb{E}\left[e^{\theta X_i}\right] \leq \mathbb{E}\left[1 + (e^{\theta}-1)\cdot X_i\right] = 1 + (e^{\theta}-1)\cdot\mathbb{E}\left[X_i\right]. \tag{23.3}$$

With some standard calculations, we can choose the optimal value of θ to establish the upper tail bound in Theorem 23.1. We will explain these details in the proof of the matrix Chernoff bound.

Note that this approach applies more generally to other types of random variables, not just when $X_i \in [0,1]$, as long as we can obtain a suitable upper bound on $\mathbb{E}\left[e^{\theta X_i}\right]$ (for example, when the X_i are Gaussian random variables; we will see a matrix concentration inequality involving Gaussian random variables).

23.2 Matrix Exponential and Logarithm

To extend the above approach to the matrix setting, we need to define the exponential of a matrix, and more generally, the notion of a function of a matrix.

Function of a Real Symmetric Matrix: This definition is natural when the matrix is diagonal. Let f be a real-valued function and $D \in \mathbb{R}^{n\times n}$ be a diagonal matrix. Then $f(D) \in \mathbb{R}^{n\times n}$ is simply the diagonal matrix

$$f(D) := \begin{bmatrix} f(D_{1,1}) & & \\ & \ddots & \\ & & f(D_{n,n}) \end{bmatrix}.$$

This definition can be extended to real symmetric matrices as follows. Given a real symmetric matrix A, write $A = VDV^\top$ using an eigenvalue decomposition as described in (A.1). Define

$$f(A) := V \cdot f(D) \cdot V^\top.$$

Note that the definition of $f(A)$ does not depend on the particular eigenvalue decomposition $A = VDV^\top$ chosen. The following property is immediate from the definition.

Claim 23.2 (Spectral Mapping)**.** *If $(\lambda_1,\ldots,\lambda_n)$ are the eigenvalues of a real symmetric matrix $A \in \mathbb{R}^{n\times n}$, then $(f(\lambda_1),\ldots,f(\lambda_n))$ are the eigenvalues of $f(A) \in \mathbb{R}^{n\times n}$.*

A useful property of this definition of matrix functions is that an inequality in the scalar setting can be transferred to an inequality in the matrix setting. The following claim can be shown easily for diagonal matrices, and then extended to real symmetric matrices by conjugation.

Claim 23.3 (Transfer Rule)**.** *Let f and g be real-valued functions defined on an interval $I \subseteq \mathbb{R}$. Let A be a real symmetric matrix whose eigenvalues are contained in I. Then*

$$f(a) \leq g(a) \quad \textit{for each } a \in I \quad \Longrightarrow \quad f(A) \preccurlyeq g(A).$$

A simple example of such a function is $f(x) = x^k$ for a natural number k, for which $f(A) = A^k$.

Matrix Exponential: The matrix exponential e^A of a real symmetric matrix A is defined using the principle above, with the exponential function $f(x) = e^x$. Using the Taylor expansion of e^x, we can equivalently write

$$e^A = I + \sum_{k=1}^{\infty} \frac{A^k}{k!}.$$

Since $e^x > 0$ for all $x \in \mathbb{R}$, it follows from Claim 23.2 that

$$e^A \succ 0 \quad \text{for any real symmetric matrix } A. \tag{23.4}$$

Alternatively, this can be shown by observing that $e^A = e^{\frac{A}{2}} \cdot e^{\frac{A}{2}}$.

Given an inequality such as $1 + x \le e^x$ for $x \in \mathbb{R}$, we can apply the transfer rule in Claim 23.3 to establish that $I + A \preccurlyeq e^A$ for any real symmetric matrix. This approach will allow us to extend the inequality in (23.3) for the Chernoff–Hoeffding bound to establish the matrix Chernoff bound.

Matrix Logarithm: The matrix logarithm $\log A$ of a positive definite matrix A is defined using the same principle as above, with the logarithm function $f(x) = \log x$, which is only defined for positive real numbers. Note that the matrix logarithm is the functional inverse of the matrix exponential:

$$\log(e^A) = A \quad \text{for any real symmetric matrix } A.$$

The matrix logarithm also plays an important role in Tropp's proof of the matrix Chernoff bound.

23.3 Matrix Concentration Inequalities via Matrix Exponentials

Ahlswede and Winter [AW02] were the first to lift the ideas used in proving Chernoff bounds to the matrix setting. The following lemma generalizes the first step in the proof of the Chernoff bound in (23.1) using matrix exponentials.

Proposition 23.4 (Tail Bound for Eigenvalues)**.** *Let $X \in \mathbb{R}^{n \times n}$ be a random real symmetric matrix. For any $t \in \mathbb{R}$,*

$$\Pr[\lambda_{\max}(X) \ge t] \le \inf_{\theta > 0} e^{-\theta t} \cdot \mathbb{E}\big[\operatorname{Tr} e^{\theta X}\big] \quad \textit{and} \quad \Pr[\lambda_{\min}(X) \le t] \le \inf_{\theta < 0} e^{-\theta t} \cdot \mathbb{E}\big[\operatorname{Tr} e^{\theta X}\big].$$

Proof. Exponentiating and applying Markov's inequality as in (23.1),

$$\Pr[\lambda_{\max}(X) \ge t] = \Pr\left[e^{\theta \lambda_{\max}(X)} \ge e^{\theta t}\right] \le e^{-\theta t} \cdot \mathbb{E}\big[e^{\theta \lambda_{\max}(X)}\big] = e^{-\theta t} \cdot \mathbb{E}\big[\lambda_{\max}\big(e^{\theta X}\big)\big],$$

where the last equality follows from the spectral mapping property in Claim 23.2. Since the matrix exponential is positive definite (23.4), it follows from Fact A.35 that

$$\lambda_{\max}\big(e^{\theta X}\big) \le \sum_{i=1}^{n} \lambda_i\big(e^{\theta X}\big) = \operatorname{Tr}\big(e^{\theta X}\big).$$

This proves the upper tail bound, and the lower tail bound follows by an analogous argument. □

One point to notice is that the maximum eigenvalue $\lambda_{\max}\big(e^{\theta X}\big)$ is replaced by the larger quantity $\operatorname{Tr}\big(e^{\theta X}\big)$. The reason is that there are no nice formulas to bound $\lambda_{\max}\big(e^{\theta X}\big)$ in terms of quantities involving $X_1, \ldots, X_n$, whereas there are tools to do so for the trace, as we will see next.

Bounding Trace Exponential Using Golden–Thompson Inequality

The main difficulty in the matrix setting is non-commutativity. In the scalar setting, $e^{a+b} = e^a e^b$, but in the matrix setting, $e^{A+B} \neq e^A e^B$ unless A and B commute. As a result, it is unclear how to generalize the second step in the proof of the Chernoff bound in (23.2) to the matrix setting.

The key tool used by Ahlswede and Winter to get around this difficulty is the Golden–Thompson inequality, which states that

$$\operatorname{Tr}\left(e^{A+B}\right) \leq \operatorname{Tr}\left(e^A e^B\right) \quad \text{for real symmetric matrices } A, B. \tag{23.5}$$

This inequality has an interesting history with various connections and proofs; we refer the reader to [FT14] for a nice exposition. We will not provide a proof here, as this does not directly lead to a proof of the matrix Chernoff bound.

By considering the trace and using the Golden–Thompson inequality, Ahlswede and Winter proved the following generalization of the second step in the proof of the Chernoff bound in (23.1).

Proposition 23.5 (Bounding Trace Exponential Using Golden–Thompson)**.** *For independent random real symmetric matrices $X_1, \ldots, X_m \in \mathbb{R}^{n \times n}$,*

$$\mathbb{E}\left[\operatorname{Tr}\left(e^{\theta(X_1+\cdots+X_m)}\right)\right] \leq n \prod_{i=1}^{m} \left\|\mathbb{E}\left[e^{\theta X_i}\right]\right\|_{\mathrm{op}}.$$

Proof. By the Golden–Thompson inequality and the independence of the random variables,

$$\begin{aligned}
\mathbb{E}\left[\operatorname{Tr}\left(e^{\theta(X_1+\cdots+X_m)}\right)\right] &\leq \mathbb{E}\left[\operatorname{Tr}\left(e^{\theta(X_1+\cdots+X_{m-1})} \cdot e^{\theta X_m}\right)\right] \\
&= \mathbb{E}_{X_1,\ldots,X_{m-1}}\left[\mathbb{E}_{X_m}\left[\operatorname{Tr}\left(e^{\theta(X_1+\cdots+X_{m-1})} \cdot e^{\theta X_m}\right)\right]\right] \\
&= \mathbb{E}_{X_1,\ldots,X_{m-1}}\left[\operatorname{Tr}\left(e^{\theta(X_1+\cdots+X_{m-1})} \cdot \mathbb{E}_{X_m}\left[e^{\theta X_m}\right]\right)\right] \\
&\leq \left\|\mathbb{E}\left[e^{\theta X_m}\right]\right\|_{\mathrm{op}} \cdot \mathbb{E}\left[\operatorname{Tr}\left(e^{\theta(X_1+\cdots+X_{m-1})}\right)\right],
\end{aligned}$$

where the last inequality is by Fact A.37. Repeating this argument recursively proves the proposition, where the final step uses $\operatorname{Tr}(e^{\theta \cdot 0}) = \operatorname{Tr}(I_n) = n$. □

Using Proposition 23.4 and Proposition 23.5, together with a simple bound on $\left\|\mathbb{E}[e^{\theta Z}]\right\|_{\mathrm{op}}$, Ahlswede and Winter proved the following matrix concentration inequality for independent, identically distributed random matrices.

Theorem 23.6 (Ahlswede–Winter [AW02])**.** *Let Z be a random $n \times n$ positive semidefinite matrix, with $Z \preccurlyeq r \cdot \mathbb{E}[Z]$ for some $r \geq 1$. Let $Z_1, Z_2, \ldots, Z_m$ be independent copies of Z. For any $\epsilon \in (0, 1)$,*

$$\Pr\left[(1-\epsilon)\mathbb{E}[Z] \preccurlyeq \frac{1}{m}\sum_{i=1}^{m} Z_i \preccurlyeq (1+\epsilon)\mathbb{E}[Z]\right] \geq 1 - 2n \exp\left(\frac{-\epsilon^2 m}{4r}\right).$$

In Problem 23.22, we outline a proof of Theorem 23.6. In Problem 23.23, we show that Theorem 23.6 can also be used to obtain the spectral sparsification result in Theorem 14.8.

The reason that this approach does not lead to a proof of the matrix Chernoff bound is that Proposition 23.5 can be quite loose when the random variables are not identically distributed.

Bounding Trace Exponential Using Lieb's Concavity Theorem

Tropp discovered that a deep result of Lieb can be applied effectively to prove sharp matrix concentration inequalities in much broader settings. The proof of the following theorem is far beyond the scope of this book. We refer the reader to [Tro15, Chapter 8] for a self-contained exposition.

Theorem 23.7 (Lieb's Concavity Theorem)**.** *For any real symmetric matrix H, the function*

$$A \mapsto \operatorname{Tr}(e^{H+\log A})$$

is a concave function on the convex cone of positive definite matrices.

Using Lieb's concavity theorem, Tropp derived a sharper generalization of the second step in the proof of the Chernoff bound in (23.1), using both matrix exponentials and matrix logarithms.

Proposition 23.8 (Bounding Trace Exponential Using Lieb's Concavity)**.** *Let $X_1, \dots, X_m$ be independent random real symmetric matrices. Then, for any $\theta \in \mathbb{R}$,*

$$\mathbb{E}\big[\operatorname{Tr}\big(e^{\theta(X_1+\dots+X_m)}\big)\big] \le \operatorname{Tr}\big(e^{\sum_{i=1}^m \log \mathbb{E}[e^{\theta X_i}]}\big).$$

Proof. Using Jensen's inequality that $\mathbb{E}[f(X)] \le f(\mathbb{E}[X])$ for a concave function f, together with Lieb's concavity theorem with $H = \sum_{i=1}^{m-1} \theta X_i$ and $A = e^{\theta X_m}$, we obtain

$$\begin{aligned}\mathbb{E}\big[\operatorname{Tr}\big(e^{\theta(X_1+\dots+X_m)}\big)\big] &= \mathbb{E}_{X_1,\dots,X_{m-1}}\big[\mathbb{E}_{X_m}\big[\operatorname{Tr}\big(e^{\sum_{i=1}^{m-1}\theta X_i + \log e^{\theta X_m}}\big)\big]\big] \\ &\le \mathbb{E}_{X_1,\dots,X_{m-1}}\big[\operatorname{Tr}\big(e^{\sum_{i=1}^{m-1}\theta X_i + \log \mathbb{E}[e^{\theta X_m}]}\big)\big]\end{aligned}$$

The proposition follows by repeatedly applying the same argument, for $k = m-1, \dots, 1$, with $H = \sum_{i=1}^{k-1} \theta X_i + \sum_{j=k+1}^{m} \log \mathbb{E}[e^{\theta X_j}]$ and $A = e^{\theta X_k}$. □

As we will see, this bound allows us to retain the additive structure of the random variables and therefore applies naturally to heterogeneous settings.

23.4 Matrix Chernoff Bounds and Matrix Bernstein Inequality

In this section, we derive matrix Chernoff bounds in two settings and a matrix Bernstein inequality, following closely the presentation in [Tro15]. All proofs use the same framework, while the details differ in how the moment generating functions are bounded.

Matrix Chernoff Bound for Positive Semidefinite Matrices

The following lemma uses the transfer rule in Claim 23.3 to generalize the third step in the proof of the Chernoff bound in (23.3).

Lemma 23.9 (Bounding Moment Generating Function for Positive Semidefinite Matrices)**.** *Let X be a random real symmetric matrix that satisfies $X \succcurlyeq 0$ and $\|X\| \le r$. For $\theta \in \mathbb{R}$,*

$$\mathbb{E}[e^{\theta X}] \preccurlyeq \exp\Big(\frac{e^{\theta r}-1}{r} \cdot \mathbb{E}[X]\Big) \quad \text{and} \quad \log \mathbb{E}[e^{\theta X}] \preccurlyeq \frac{e^{\theta r}-1}{r} \cdot \mathbb{E}[X]$$

Proof. The function $f(x) = e^{\theta x}$ is convex, so its graph lies below the chord connecting the two points $\{0, r\}$. This yields

$$e^{\theta x} \le 1 + \frac{e^{\theta r} - 1}{r} \cdot x \quad \text{for } x \in [0, r] \qquad \Longrightarrow \qquad e^{\theta X} \preccurlyeq I + \frac{e^{\theta r} - 1}{r} \cdot X,$$

where the implication follows from the transfer rule in Claim 23.3 and the assumption that the eigenvalues of X lie in the interval $[0, r]$.

Taking expectations preserves the semidefinite ordering, which proves the first inequality. The second inequality follows from the nontrivial fact that the matrix logarithm also preserves the semidefinite ordering; see [Lee21] for a nice exposition of a proof of this fact. □

The matrix Chernoff bound in Theorem 14.10 follows readily.

Proof of Theorem 14.10. Applying Proposition 23.4, Proposition 23.8, and Lemma 23.9,

$$\Pr[\lambda_{\max}(X) \ge t] \le \inf_{\theta > 0} e^{-\theta t} \cdot \mathbb{E}\big[\operatorname{Tr} e^{\theta X}\big] \le \inf_{\theta > 0} e^{-\theta t} \cdot \operatorname{Tr}\left(e^{\sum_{i=1}^m \log \mathbb{E}[e^{\theta X_i}]}\right) \le \inf_{\theta > 0} e^{-\theta t} \cdot \operatorname{Tr}\left(e^{\sum_{i=1}^m g(\theta) \cdot \mathbb{E}[X_i]}\right),$$

where in the last inequality we use the fact that $A \preccurlyeq B$ implies $\operatorname{Tr}(e^A) \le \operatorname{Tr}(e^B)$, together with the shorthand $g(\theta) := (e^{\theta r} - 1)/r$. It follows from the assumption $\mathbb{E}[X] \preccurlyeq \mu_{\max} \cdot I_n$ that

$$\Pr[\lambda_{\max}(X) \ge t] \le \inf_{\theta > 0} e^{-\theta t} \cdot \operatorname{Tr}\left(e^{g(\theta) \cdot \mathbb{E}[X]}\right) \le \inf_{\theta > 0} e^{-\theta t} \cdot n \cdot e^{g(\theta) \cdot \mu_{\max}}.$$

Setting $t = (1 + \epsilon)\mu_{\max}$ and $\theta = r^{-1} \log(1 + \epsilon)$, we obtain

$$\Pr\left[\lambda_{\max}(X) \ge (1 + \epsilon)\mu_{\max}\right] \le n\left(\frac{e^{\epsilon}}{(1 + \epsilon)^{1 + \epsilon}}\right)^{\frac{\mu_{\max}}{r}} \le n e^{-\frac{\epsilon^2 \mu_{\max}}{3r}},$$

where the last inequality holds for $0 \le \epsilon \le 1$ by standard calculations for classical Chernoff bounds. This proves the upper tail bound, and the lower tail bound follows by an analogous argument. □

Matrix Chernoff Bound for Gaussian and Rademacher Series

We will use the following version of the matrix Chernoff bound for Gaussian series for the tensor principle component analysis problem in Section 25.4. A new feature is the use of the matrix variance parameter σ^2 to control the moment generating function.

Theorem 23.10 (Matrix Gaussian and Rademacher Series)**.** *Let $A_1, \dots, A_m$ be $n \times n$ real symmetric matrices, and let $g_1, \dots, g_m$ be independent $N(0, 1)$ random variables. Let*

$$Z = \sum_{i=1}^m g_i A_i \quad \text{and} \quad \sigma^2(Z) = \left\| \mathbb{E}\left[Z^2\right] \right\|_{\text{op}} = \left\| \sum_{i=1}^m A_i^2 \right\|_{\text{op}}.$$

Then, for any $t \ge 0$,

$$\Pr(\|Z\|_{\text{op}} \ge t) \le 2n \cdot \exp\left(-\frac{t^2}{2\sigma^2(Z)}\right).$$

The same bound holds when we replace $g_1, \dots, g_m$ by independent Rademacher random variables $s_1, \dots, s_m$, i.e., ± 1-valued with equal probabilities.

The following lemma is based on some standard results about Gaussian random variables.

Lemma 23.11 (Bounding Moment Generating Function for Gaussian Series)**.** *Let A be a real symmetric matrix and g be a standard normal random variable. For $\theta \in \mathbb{R}$,*

$$\mathbb{E}[e^{g\theta A}] = e^{\theta^2 A^2/2} \quad \textit{and} \quad \log \mathbb{E}[e^{g\theta A}] = \frac{1}{2}\theta^2 A^2.$$

Proof. We may assume $\theta = 1$ by absorbing θ into the matrix A. For $q \in \mathbb{N}$, it is well known that the moments of a standard normal random variable satisfy

$$\mathbb{E}\left[g^{2q+1}\right] = 0 \quad \text{and} \quad \mathbb{E}\left[g^{2q}\right] = \prod_{i=1}^{q}(2i-1)! = \frac{(2q)!}{2^q q!},$$

which can be derived from the Gaussian integration-by-parts formula in Fact 23.14. It follows that

$$\mathbb{E}\left[e^{gA}\right] = I + \sum_{q=1}^{\infty} \frac{\mathbb{E}\left[g^{2q}\right]}{(2q)!} A^{2q} = I + \sum_{q=1}^{\infty} \frac{1}{q!}\left(\frac{A^2}{2}\right)^q = e^{\frac{A^2}{2}}.$$

□

The proof of Theorem 23.10 follows the same steps as in the previous matrix Chernoff bound.

Proof of Theorem 23.10. Applying Proposition 23.4, Proposition 23.8, and Lemma 23.11,

$$\Pr[\lambda_{\max}(Z) \geq t] \leq \inf_{\theta>0} e^{-\theta t} \cdot \mathbb{E}\left[\operatorname{Tr} e^{\theta Z}\right] \leq \inf_{\theta>0} e^{-\theta t} \cdot \operatorname{Tr}\left(e^{\sum_{i=1}^m \log \mathbb{E}[e^{\theta A_i}]}\right) = \inf_{\theta>0} e^{-\theta t} \cdot \operatorname{Tr}\left(e^{\frac{\theta^2}{2}\sum_{i=1}^m A_i^2}\right).$$

Since the trace is at most n times the maximum eigenvalue, it follows from the spectral mapping property in Claim 23.2 that

$$\Pr[\lambda_{\max}(Z) \geq t] \leq \inf_{\theta>0} e^{-\theta t} \cdot n \cdot \lambda_{\max}\left(e^{\frac{\theta^2}{2}\sum_{i=1}^m A_i^2}\right) = n \cdot \inf_{\theta>0} e^{-\theta t + \frac{1}{2}\theta^2\sigma^2(Z)}.$$

Choosing $\theta = t/\sigma^2(Z)$ yields the theorem. The proof for the minimum eigenvalue is analogous, and the theorem follows by a union bound. □

The proof for the matrix Rademacher series is left to Problem 23.25.

Matrix Bernstein Inequality

We will also use the following matrix Bernstein inequality for analyzing refutation algorithms for random constraint satisfaction problems in Chapter 24.

Theorem 23.12 (Matrix Bernstein Inequality)**.** *Let $X_1, \ldots, X_m$ be independent random $n \times n$ real symmetric matrices such that $\mathbb{E}[X_i] = 0$ and $\|X_i\|_{\text{op}} \leq L$ for $1 \leq i \leq m$. Let*

$$Z = \sum_{i=1}^m X_i \quad \textit{and} \quad \sigma^2(Z) = \left\|\mathbb{E}\left[Z^2\right]\right\|_{\text{op}} = \left\|\sum_{i=1}^m \mathbb{E}\left[X_i^2\right]\right\|_{\text{op}}.$$

Then, for any $t \geq 0$,

$$\Pr(\|Z\|_{\text{op}} \geq t) \leq 2n \cdot \exp\left(\frac{-\frac{1}{2}t^2}{\sigma^2(Z) + \frac{1}{3}Lt}\right).$$

The moment generating function is bit trickier to bound in this setting.

Lemma 23.13 (Bounding Moment Generating Function for Matrix Bernstein)**.** *Let X be a random real symmetric matrix that satisfies $\mathbb{E}[X] = 0$ and $\lambda_{\max}(X) \le L$. Then, for $0 < \theta < 3/L$,*

$$\mathbb{E}[e^{\theta X}] \preccurlyeq \exp\left(\frac{\frac{1}{2}\theta^2}{1 - \frac{1}{3}\theta L} \cdot \mathbb{E}\left[X^2\right]\right) \quad \text{and} \quad \log \mathbb{E}[e^{\theta X}] \preccurlyeq \frac{\frac{1}{2}\theta^2}{1 - \frac{1}{3}\theta L} \cdot \mathbb{E}\left[X^2\right]$$

Proof. To express the moment generating function using the mean and variance, we write

$$e^{\theta X} = I + \theta X + (e^{\theta X} - \theta X - 1) = I + \theta X + X \cdot f(X) \cdot X,$$

where $f(x) = \frac{1}{x^2}(e^{\theta x} - \theta x - 1)$ for $x \neq 0$ and $f(0) = \frac{1}{2}\theta^2$. Since the function f is increasing and the eigenvalues of X is upper bounded by L, it follows from the transfer rule in Claim 23.3 that

$$f(X) \preccurlyeq f(L) \cdot I \quad \Longrightarrow \quad e^{\theta X} \preccurlyeq I + \theta X + f(L) \cdot X^2.$$

Observe that

$$f(L) = \frac{e^{\theta L} - \theta L - 1}{L^2} = \frac{1}{L^2} \sum_{q=2}^{\infty} \frac{(\theta L)^q}{q!} \le \frac{\theta^2}{2} \sum_{q=2}^{\infty} \frac{(\theta L)^{q-2}}{3^{q-2}} = \frac{\frac{1}{2}\theta^2}{1 - \frac{1}{3}\theta L}.$$

Using this bound and taking expectation yields

$$\mathbb{E}[e^{\theta X}] \preccurlyeq I + \theta \cdot \mathbb{E}[X] + f(L) \cdot \mathbb{E}\left[X^2\right] \preccurlyeq I + \frac{\frac{1}{2}\theta^2}{1 - \frac{1}{3}\theta L} \cdot \mathbb{E}\left[X^2\right] \preccurlyeq \exp\left(\frac{\frac{1}{2}\theta^2}{1 - \frac{1}{3}\theta L} \cdot \mathbb{E}\left[X^2\right]\right),$$

where the last inequality follows from applying the transfer rule to the inequality $1 + x \le e^x$. □

The proof of the matrix Bernstein inequality follows the same steps as in matrix Chernoff bounds.

Proof of Theorem 23.12. Let $g(\theta) = \frac{\theta^2/2}{1 - \theta L/3}$. Applying Proposition 23.4, Proposition 23.8, and Lemma 23.11,

$$\Pr[\lambda_{\max}(Z) \ge t] \le \inf_{\theta > 0} e^{-\theta t} \cdot \mathbb{E}\left[\operatorname{Tr} e^{\theta Z}\right] \le \inf_{\theta > 0} e^{-\theta t} \cdot \operatorname{Tr}\left(e^{\sum_{i=1}^m \log \mathbb{E}[e^{\theta X_i}]}\right) = \inf_{\theta > 0} e^{-\theta t} \cdot \operatorname{Tr}\left(e^{g(\theta) \sum_{i=1}^m \mathbb{E}[X_i^2]}\right).$$

As in the proof of Theorem 23.10,

$$\Pr[\lambda_{\max}(Z) \ge t] \le \inf_{\theta > 0} e^{-\theta t} \cdot n \cdot \lambda_{\max}\left(e^{g(\theta) \sum_{i=1}^m X_i^2}\right) \le n \cdot \inf_{\theta > 0} e^{-\theta t + g(\theta) \cdot \sigma^2(Z)}.$$

Choosing $\theta = t/(\sigma^2(Z) + Lt/3)$ yields the theorem. The proof for the minimum eigenvalue is analogous, and the theorem follows by a union bound. □

We highly recommend [Tro15] for more details and historical background on these matrix concentration inequalities.

23.5 An Elementary Approach to Matrix Concentration

In this section, we discuss an alternative, more elementary, approach to deriving matrix concentration inequalities. The previous approach is based on bounding the moment generating function, which requires the use of advanced results such as the Golden–Thompson inequality and Lieb's concavity theorem. The current approach is based on bounding the *expected value* of the p-th moment of the eigenvalues using the matrix Khintchine inequality, and then applying classical (scalar) concentration inequalities to obtain tail bounds. Our presentation is based on the survey by Van Handel [VH17] and the course notes on matrix concentration by Kothari.

Moment Method

For a vector $x \in \mathbb{R}^n$ and any $p \in \mathbb{N}$, it is a simple bound that

$$\|x\|_\infty \le \|x\|_p \le n^{\frac{1}{p}} \cdot \|x\|_\infty \qquad \Longrightarrow \qquad \|x\|_\infty \asymp \|x\|_p \ \text{ for } p \asymp \log n.$$

Let $\lambda = (\lambda_1, \ldots, \lambda_n)$ be the eigenvalues of a real symmetric matrix X. Then $\|\lambda\|_\infty = \|X\|_{\mathrm{op}}$ and $\|\lambda\|_{2p} = \mathrm{Tr}[X^{2p}]^{\frac{1}{2p}}$. This implies that

$$\|X\|_{\mathrm{op}} \asymp \mathrm{Tr}\left[X^{2p}\right]^{\frac{1}{2p}} \quad \text{for } p \asymp \log n. \tag{23.6}$$

This provides a way to obtain a constant-factor approximation to the operator norm by computing the trace of the $\Theta(\log n)$-th power of the matrix.

Gaussian Model

Let $A_1, \ldots, A_m$ be $n \times n$ real symmetric matrices. Consider the random matrix

$$Z = \sum_{i=1}^m g_i A_i, \tag{23.7}$$

where $g_i \sim N(0,1)$ are independent Gaussian random variables. Note that the variance is

$$\sigma^2(Z) = \left\|\mathbb{E}\left[Z^2\right]\right\|_{\mathrm{op}} = \left\|\sum_{i=1}^m A_i^2\right\|_{\mathrm{op}}. \tag{23.8}$$

One advantage of this Gaussian model is that it allows for easier calculations, and the resulting bound can be adapted to other models such as the random Rademacher model $Z = \sum_{i=1}^m s_i A_i$ where $s_i \in \{\pm 1\}$ with equal probability. Using $\mathbb{E}\left[|g_i|\right] = \sqrt{2/\pi}$ and Jensen's inequality, we see that

$$\mathbb{E}_g\left\|\sum_{i=1}^m g_i A_i\right\|_{\mathrm{op}} = \mathbb{E}_{g,s}\left\|\sum_{i=1}^m |g_i| s_i A_i\right\|_{\mathrm{op}} \ge \mathbb{E}_s\left\|\sum_{i=1}^m s_i \mathbb{E}_g|g_i| A_i\right\|_{\mathrm{op}} = \sqrt{\frac{2}{\pi}}\mathbb{E}_s\left\|\sum_{i=1}^m s_i A_i\right\|_{\mathrm{op}}. \tag{23.9}$$

A useful formula for calculations in the Gaussian model is the following.

Fact 23.14 (Gaussian Integration by Parts). *Let $g \sim N(0,1)$. For smooth function f with compact support,*

$$\mathbb{E}\left[g \cdot f(g)\right] = \mathbb{E}\left[f'(g)\right].$$

Using this formula, one can derive, for instance, that the expected maximum of n i.i.d. Gaussian random variables is of order $O(\log n)$.

Matrix Khintchine Inequality

Lust-Piquard and Pisier proved the following non-commutative Khintchine inequality.

Theorem 23.15 (Non-Commutative Khintchine Inequality). *In the Gaussian model $Z = \sum_{i=1}^m g_i A_i$ in (23.7), for any $p \ge 1$,*

$$\mathbb{E}\left[\mathrm{Tr}[Z^{2p}]\right]^{\frac{1}{2p}} \lesssim \sqrt{2p-1} \cdot \mathrm{Tr}\left[\mathbb{E}\left[Z^2\right]^p\right]^{\frac{1}{2p}} = \sqrt{2p-1} \cdot \mathrm{Tr}\left[\left(\sum_{i=1}^m A_i^2\right)^p\right]^{\frac{1}{2p}}.$$

The same holds for the Rademacher model $Z = \sum_{i=1}^m s_i A_i$, where $s_i = \pm 1$ with equal probability.

Proof. The idea is to use Gaussian integration by parts with $f(g_i) = \operatorname{Tr}[A_i Z^{2p-1}]$, and the product rule to compute $f'(g_i)$ so that

$$\mathbb{E}\left[\operatorname{Tr}[Z^{2p}]\right] = \sum_{i=1}^{m} \mathbb{E}\left[g_i \operatorname{Tr}[A_i Z^{2p-1}]\right] = \sum_{i=1}^{m} \sum_{l=0}^{2p-2} \mathbb{E}\left[\operatorname{Tr}\left[A_i Z^l A_i Z^{2p-2-l}\right]\right].$$

The generalized Lieb–Thirring inequality in Lemma 23.17 states that for $1 \le i \le m$,

$$\operatorname{Tr}[A_i Z^l A_i Z^{2p-2-l}] \le \operatorname{Tr}[A_i^2 Z^{2p-2}].$$

Using the matrix Hölder's inequality in Lemma A.39, which states that for p, q with $\frac{1}{p} + \frac{1}{q} = 1$,

$$\operatorname{Tr}(AB) \le \|A\|_p \cdot \|B\|_q = \operatorname{Tr}\left(|A|^p\right)^{\frac{1}{p}} \cdot \operatorname{Tr}\left(|B|^{\frac{p}{p-1}}\right)^{1-\frac{1}{p}},$$

it follows that

$$\mathbb{E}\left[\operatorname{Tr}[Z^{2p}]\right] \le (2p-1) \sum_{i=1}^{m} \mathbb{E}\left[\operatorname{Tr}\left[A_i^2 Z^{2p-2}\right]\right] \le (2p-1) \cdot \operatorname{Tr}\left[\left(\sum_{i=1}^{m} A_i^2\right)^p\right]^{\frac{1}{p}} \cdot \mathbb{E}\left[\operatorname{Tr}[Z^{2p}]^{1-\frac{1}{p}}\right].$$

Using (23.8) and Jensen's inequality that $\mathbb{E}\left[\operatorname{Tr}[Z^{2p}]^{1-\frac{1}{p}}\right] \le \mathbb{E}\left[\operatorname{Tr}[Z^{2p}]\right]^{1-\frac{1}{p}}$, rearranging and taking the square root gives the conclusion. □

Together with the moment method, the following is an elegant corollary that is widely applicable.

Theorem 23.16 (Matrix Khintchine Inequality)**.** *In the Gaussian model $Z = \sum_{i=1}^{m} g_i A_i$ in (23.7), where $Z \in \mathbb{R}^{n \times n}$,*

$$\mathbb{E}\left[\|Z\|_{\mathrm{op}}\right] \lesssim \sqrt{\log n} \cdot \left\|\mathbb{E}\left[Z^2\right]\right\|_{\mathrm{op}}^{\frac{1}{2}}$$

The same holds for the Rademacher model $Z = \sum_{i=1}^{m} s_i A_i$, where $s_i = \pm 1$ with equal probability.

Proof. Combining Jensen's inequality, the moment method in (23.6) with $p \asymp \log n$, and the non-commutative Khintchine inequality in Theorem 23.15 gives

$$\mathbb{E}\left[\|Z\|_{\mathrm{op}}\right] \le \mathbb{E}\left[\|Z\|_{\mathrm{op}}^{2p}\right]^{\frac{1}{2p}} \asymp \mathbb{E}\left[\operatorname{Tr}[Z^{2p}]\right]^{\frac{1}{2p}} \lesssim \sqrt{\log n} \cdot \operatorname{Tr}\left[\mathbb{E}\left[Z^2\right]^{\log n}\right]^{\frac{1}{2\log n}} \asymp \sqrt{\log n} \cdot \left\|\mathbb{E}\left[Z^2\right]\right\|_{\mathrm{op}}^{\frac{1}{2}}.$$

The conclusion for the Rademacher model follows from (23.9). □

It remains to prove the generalized Lieb–Thirring inequality.

Lemma 23.17 (Generalized Lieb–Thirring Inequality)**.** *Let X be a real symmetric matrix. For integers $p \ge l \ge 0$,*

$$\operatorname{Tr}\left[AX^l AX^{p-l}\right] \le \operatorname{Tr}\left[A^2 X^p\right].$$

Proof. Let $X = \sum_{i=1}^{n} \lambda_i v_i v_i^\top$. Then the left-hand side is

$$\operatorname{Tr}\left[A\left(\sum_{i=1}^{n} \lambda_i^l v_i v_i^\top\right) A\left(\sum_{j=1}^{n} \lambda_j^{p-l} v_j v_j^\top\right)\right] = \sum_{i=1}^{n}\sum_{j=1}^{n} \lambda_i^l \lambda_j^{p-l} \cdot \left|\langle v_i, Av_j\rangle\right|^2 = \sum_{i=1}^{n}\sum_{j=1}^{n} \left(\frac{\lambda_i}{\lambda_j}\right)^l \lambda_j^p \cdot \left|\langle v_i, Av_j\rangle\right|^2.$$

Note that the last expression is a convex function of l, and so its maximum is attained at either $l = 0$ or $l = p$, and in both cases it is equal to

$$\sum_{i=1}^{n}\sum_{j=1}^{n} \lambda_i^p \cdot \left|\langle v_i, Av_j\rangle\right|^2 = \operatorname{Tr}\left[A\left(\sum_{i=1}^{n} v_i v_i^\top\right) A\left(\sum_{j=1}^{n} \lambda_j^p v_j v_j^\top\right)\right] = \operatorname{Tr}\left[A^2 X^p\right].$$

□

Deriving Matrix Concentration Inequalities

Using the matrix Khintchine inequality in Theorem 23.16 with some additional (but elementary) calculations, one can derive upper bounds on the expected value of the operator norm in different settings. For example, in the setting of the matrix Chernoff bound for positive semidefinite matrices in Theorem 14.10, one can show that

$$\mathbb{E}\left\|\sum_{i=1}^{m} X_i\right\|_{\text{op}} \leq (1+\epsilon)\left\|\sum_{i=1}^{m} \mathbb{E}\left[X_i\right]\right\|_{\text{op}} + O\left(1+\frac{1}{\epsilon}\right) r\sqrt{\log n}.$$

In the setting of the matrix Bernstein inequality in Theorem 23.12, one can show that

$$\mathbb{E}\left\|\sum_{i=1}^{m} X_i\right\|_{\text{op}} \lesssim \sqrt{\log n}\left\|\sum_{i=1}^{m} \mathbb{E}\left[X_i^2\right]\right\|_{\text{op}}^{\frac{1}{2}} + L\log n.$$

We refer the reader to the course notes on matrix concentration by Kothari for proofs.

To prove that $\|\sum_{i=1}^{m} X_i\|_{\text{op}}$ is close to its expected value with high probability, one can then apply classical (scalar) concentration inequalities for Lipschitz functions (note that any norm is 1-Lipschitz). For example, the following theorem applies to the Gaussian model.

Theorem 23.18 (Gaussian Concentration)**.** *Let $g = (g_1, \ldots, g_m)$ be a vector of m independent standard normal random variables. Let $f : \mathbb{R}^m \to \mathbb{R}$ denote an ℓ-Lipschitz function. For any $t \geq 0$,*

$$\Pr\left[f(g) - \mathbb{E}\left[f(g)\right] \geq t\right] \leq e^{-\frac{t^2}{2\ell^2}}.$$

The following theorem applies to the settings in the matrix Chernoff bound and the matrix Bernstein inequality where the random variables are bounded.

Theorem 23.19 (Concentration of Convex Lipschitz Functions)**.** *Let $X_1, \ldots, X_m$ be independent random variables such that $|X_i| \leq k$. If $f : \mathbb{R}^m \to \mathbb{R}$ is an ℓ-Lipschitz function and is separately convex (i.e., f is a convex function in the i-th variable when the other variables are fixed), then for any $t \geq 0$,*

$$\Pr\left[f(X_1, \ldots, X_m) - \mathbb{E}\left[f(X_1, \ldots, X_m)\right] \geq t\right] \leq e^{-\frac{t^2}{2\ell^2 k^2}}.$$

This completes our discussion on the alternative approach for deriving matrix concentration inequalities. We refer the reader to the excellent book [BLM13] for proofs of these concentration theorems.

23.6 Sharper Matrix Concentration Inequalities

The matrix Khintchine inequality shows that

$$\left\|\mathbb{E}\left[Z^2\right]\right\|_{\text{op}}^{\frac{1}{2}} \lesssim \mathbb{E}\left[\|Z\|_{\text{op}}\right] \lesssim \sqrt{\log n} \cdot \left\|\mathbb{E}\left[Z^2\right]\right\|_{\text{op}}^{\frac{1}{2}},$$

where the proof of the lower bound can be found in [VH17].

The dimension-dependent logarithmic factor between the lower and upper bounds is suboptimal and undesirable in some applications. This motivates the question of whether there is a more refined quantity, based on the structure of the A_i, that provides a tighter approximation to $\mathbb{E}\left[\|Z\|_{\text{op}}\right]$.

Tropp's Second-Order Khintchine Inequality

The progress on this question was initiated by Tropp [Tro18]. His intuition is that the generalized Lieb-Thirring inequality is only tight when the matrices commute. To prove a tighter bound for non-commutative matrices, his idea is to apply Gaussian integration by parts once again to each term $\mathbb{E}\left[\mathrm{Tr}[A_i X A_i X^{2p-l}]\right]$ in the proof of the non-commutative Khintchine inequality, in the hope that they are much smaller than the upper bound $\mathbb{E}\left[\mathrm{Tr}[A_i^2 X^{2p}]\right]$. Using techniques from complex analysis, Tropp introduced convexity to the problem by relaxing to a larger space and proved the following theorem.

Theorem 23.20 (Tropp's Second-Order Khintchine Inequality [Tro18])**.** *In the Gaussian model $Z = \sum_{i=1}^m g_i A_i$ in (23.7) where $Z \in \mathbb{R}^{n\times n}$, define*

$$\sigma(Z) := \left\|\sum_i A_i^2\right\|_{\mathrm{op}}^{\frac{1}{2}} \quad \text{and} \quad \tilde{\sigma}(Z) := \sup_{U_1,U_2,U_3} \left\|\sum_{i,j} A_i U_1 A_j U_2 A_i U_3 A_j\right\|_{\mathrm{op}}^{\frac{1}{4}},$$

where the supremum is taken over all triples U_1, U_2, U_3 of commuting unitary matrices. Then

$$\mathbb{E}\left[\|Z\|_{\mathrm{op}}\right] \le \sigma(Z)\cdot \log^{\frac{1}{4}} n + \tilde{\sigma}(Z)\cdot \log^{\frac{1}{2}} n.$$

Informally, the matrix alignment parameter $\tilde{\sigma}(Z)$ measures the non-commutativeness of the matrices $A_1, \ldots, A_m$. This parameter provides improved bounds on the operator norm in some situations, but its scope is limited as it is often difficult to estimate.

Sharp Matrix Concentration Inequality from Free Probability

Recently, this approach has been significantly advanced by Bandeira, Boedihardjo, and van Handel [BBvH23]. They defined the $n^2 \times n^2$ covariance matrix $\mathrm{Cov}(Z)$ where

$$\mathrm{Cov}(Z)_{ij,kl} = \mathbb{E}\left[Z_{ij}\overline{Z_{kl}}\right] \quad \text{and} \quad \nu(Z)^2 := \|\mathrm{Cov}(Z)\|_{\mathrm{op}} = \left\|\sum_{i=1}^m \mathrm{vec}(A_i)\cdot \mathrm{vec}(A_i)^\top\right\|_{\mathrm{op}},$$

where $\mathrm{vec}(A_i)$ denotes the n^2-dimensional vector representation of the $n\times n$ matrix A_i. They proved a friendly bound on the matrix alignment parameter:

$$\tilde{\sigma}(Z)^2 \le \nu(Z)\cdot\sigma(Z).$$

More significantly, they developed a novel interpolation approach to compare the spectrum of the Gaussian model Z with the spectrum of its "limiting object" Z_{free} in the theory of free probability, which satisfies $\|Z_{\mathrm{free}}\|_{\mathrm{op}} \lesssim \sigma(Z)$. Their main theorem, in a simple form, is as follows.

Theorem 23.21 (Sharp Matrix Concentration Inequality from Free Probability [BBvH23])**.** *In the Gaussian model $Z = \sum_{i=1}^m g_i A_i$ in (23.7) where $Z \in \mathbb{R}^{n\times n}$,*

$$\mathbb{E}\left[\|Z\|_{\mathrm{op}}\right] \le 2\sigma(Z) + \sigma(Z)^{\frac{1}{2}}\cdot\nu(Z)^{\frac{1}{2}}\cdot(\log n)^{\frac{3}{4}}.$$

They showed that the parameter $\nu(Z)$ is much smaller than $\sigma(Z)$ in many interesting settings, for which the new inequality provides an asymptotically sharp upper bound $\mathbb{E}\left[\|Z\|_{\mathrm{op}}\right] \lesssim \sigma(Z)$. Notably, this new inequality was at the heart of recent major progress towards proving the matrix Spencer conjecture, which we discussed in Chapter 16.

Universality Principle

In a subsequent work, Brailovskaya and van Handel [BvH24] extended these matrix concentration inequalities to a much larger class of random matrices. The random matrix model is $Z = A_0 + \sum_{i=1}^n Z_i$, where $Z_1, \dots, Z_n$ are independent Hermitian random matrices with $\mathbb{E}[Z_i] = 0$ and each $\|Z_i\|$ bounded. This captures many commonly studied random graph models, with arbitrary dependency patterns in their entries. They proved a general non-commutative universality principle, showing that the spectral statistics of Z are closely approximated by those of a Gaussian model. They also showed that these results for the general model can be applied to give new probabilistic constructions of near-Ramanujan graphs.

Derandomization

Some of the results in [BBvH23, BvH24] are derandomized in [WLZ26], providing deterministic algorithms for the matrix Spencer problem and constructions of near-Ramanujan graphs. Their proofs show that the concepts and techniques in free probability are useful not only for mathematical analyses but also for efficient computations.

23.7 Problems

Problem 23.22 (Proof of Ahlswede–Winter Inequality). *We outline a proof of Theorem 23.6.*

1. *Reduce to the case that $\mathbb{E}[Z] = I$ and $\|Z\|_{\mathrm{op}} \le r$. (Hint: see Theorem 14.8.)*
2. *Consider the random variable $X := (Z - \mathbb{E}[Z])/r$. Prove that*

$$\mathbb{E}[e^{\theta X}] \preccurlyeq e^{\theta^2 \mathbb{E}[X^2]} \quad \text{and} \quad \left\|\mathbb{E}[e^{\theta X}]\right\|_{\mathrm{op}} \le e^{\theta^2/r}.$$

   (Hint: Use the inequalities $1 + x \le e^x \le 1 + x + x^2$ for $x \in [-1, 1]$ and the transfer rule.)

3. *Conclude the proof of Theorem 23.6.*

Problem 23.23 (Application of Ahlswede–Winter Inequality to Spectral Sparsification). *The goal of this problem is to use Theorem 23.6 to prove Theorem 14.8.*

Let $G = (V, E)$ be an undirected graph. Let p be the probability distribution over the edges so that p_e is proportional to the effective resistance of the edge e in G.

Consider the following random sampling algorithm with $k \asymp |V| \log |V| / \epsilon^2$ iterations: in each iteration, we sample a random edge e according to the distribution p, and add $(kp_e)^{-1}$ to the weight of edge e (where initially all edges have weight zero).

Use Theorem 23.6 to analyze this random sampling algorithm and prove Theorem 14.8.

Hint: You may need to prove that $L_e \preccurlyeq \mathrm{Reff}(e) \cdot L_G$ and $\sum_{e \in E} \mathrm{Reff}(e) = |V| - 1$.

Problem 23.24 (Connectivity of Random Graphs [Tro15, Section 5.3.3]). *Prove that an Erdös-Rényi graph $G(n, p)$, where there are n vertices and each pair has an edge independently with probability p, is connected with high probability as $n \to \infty$ when $p > \frac{2 \log(n-1)}{n}$.*

Hint: Use the matrix Chernoff bound to lower bound the minimum eigenvalue of its Laplacian matrix.

Problem 23.25 (Matrix Rademacher Series)**.** *Prove Theorem 23.10 for* $Z = \sum_{i=1}^{m} s_i A_i$*, where* $s_i = \pm 1$ *with equal probabilities for* $1 \leq i \leq m$*.*

Hint: Prove $\mathbb{E}\left[e^{s\theta A}\right] \preccurlyeq e^{\theta^2 A^2/2}$ *by establishing the inequality* $\frac{1}{2}e^{a} + \frac{1}{2}e^{-a} \leq e^{a^2/2}$*.*

Problem 23.26 (Matrix Bernstein Inequality)**.** *Prove Theorem 23.12 for random* $n \times n$*, not necessarily symmetric, matrices* $X_1, \ldots, X_m$*.*

Hint: Consider the matrices $X_i = \begin{bmatrix} 0 & X_i \\ X_i^\top & 0 \end{bmatrix}$*.*

References

[AW02] Rudolf Ahlswede and Andreas Winter. Strong converse for identification via quantum channels. *IEEE Transactions on Information Theory*, 48(3):569–579, 2002. 283, 285, 286

[BBvH23] Afonso S. Bandeira, March T. Boedihardjo, and Ramon van Hande. Matrix concentration inequalities and free probability. *Inventiones Mathematicae*, 2023. 9, 294, 295

[BLM13] Stéphane Boucheron, Gábor Lugosi, and Pascal Massart. *Concentration Inequalities: A Nonasymptotic Theory of Independence*. Oxford University Press, 02 2013. 293

[BvH24] Tatiana Brailovskaya and Ramon van Handel. Universality and sharp matrix concentration inequalities. *Geometric and Functional Analysis*, 34(6):1734–1838, 2024. 295

[FT14] Peter J Forrester and Colin J Thompson. The golden-thompson inequality: Historical aspects and random matrix applications. *Journal of Mathematical Physics*, 55(2), 2014. 286

[Lee21] James R. Lee. Lecture 5: Monotonicity and convexity. *The art and science of positive definite matrices*, 2021. 288

[Tro15] Joel A Tropp. An introduction to matrix concentration inequalities. *Foundations and Trends® in Machine Learning*, 8(1-2):1–230, 2015. 9, 283, 287, 290, 295

[Tro18] Joel A Tropp. Second-order matrix concentration inequalities. *Applied and Computational Harmonic Analysis*, 44(3):700–736, 2018. 294

[VH17] Ramon Van Handel. Structured random matrices. *Convexity and concentration*, pages 107–156, 2017. 290, 293

[WLZ26] Robert Wang, Lap Chi Lau, and Hong Zhou. Derandomizing matrix concentration inequalities from free probability. *arXiv preprint arXiv:2601.08111*, 2026. 295

Chapter 24

Refutation Algorithms for Semi-Random Constraint Satisfaction Problems

Constraint satisfaction problems (CSPs), such as k-SAT, are central problems in theoretical computer science. They are canonical NP-complete problems that are used both to prove the hardness of other problems and to design efficient algorithms through modern SAT-solvers. It is thus of fundamental interest to generate hard instances of CSPs as benchmarks for algorithms and complexity, across a range of difficulty levels.

A simple and effective way to generate hard instances is to construct a random CSP instance with m constraints and n variables. By probabilistic methods, there is a sharp threshold c^* for the constraint-to-variable ratio such that when $\frac{m}{n} < c^*$ the random instance is almost always satisfiable, while when $\frac{m}{n} > c^*$ it is almost always unsatisfiable. It is generally believed that a random instance becomes harder to solve as the constraint-to-variable ratio approaches this threshold. This provides a natural testbed for studying the power and limitations of CSP algorithms.

In this chapter, we study algorithms for certifying that a random instance is unsatisfiable when the constraint-to-variable ratio is above the threshold. In this regime, spectral techniques are surprisingly powerful, providing some of the simplest and fastest algorithms for CSP refutation. Furthermore, these techniques extend to semi-random CSP instances, with strong applications that we will see in the next chapter.

24.1 Random 2-XOR: Quadratic Form

In this section, we start with the simplest setting of refuting random 2-XOR formulas and see how spectral techniques apply naturally. Before we do so, we first explain the problem formulations.

k-XOR, Random k-XOR, and Strong Refutation

k-XOR: In a Boolean CSP instance, there are n Boolean variables $x_1, \ldots, x_n \in \{0,1\}$ and m constraints on the variables. In a k-XOR formula, each constraint involves k variables and is of the form

$$x_1 + x_2 + \cdots + x_k = b \mod 2, \quad \text{where} \quad b \in \{0,1\}.$$

XOR Principle: The "XOR principle" [Fei02, FO07, AOW15, AGK21] states that, in order to show that an arbitrary Boolean CSP instance of a given constraint-to-variables ratio is unsatisfiable, it suffices to strongly refute k-XOR formulas for various $k \geq 1$ with the same ratio (by considering the discrete Fourier transform of the constraints).

Henceforth, in the rest of this chapter, we focus on strongly refuting random and semi-random k-XOR formulas.

Strong Refutation: Given an arbitrary k-XOR formula with m constraints, refuting the formula means proving that there does not exist an assignment $x \in \{0,1\}^n$ that satisfies all m constraints. Note that a uniformly random assignment $x \in \{0,1\}^n$ is expected to satisfy $\frac{1}{2}m$ constraints. Strongly refuting the formula means proving that there does not exist an assignment satisfying more than $(\frac{1}{2} + \epsilon)m$ constraints, for some small constant $\epsilon > 0$.

Random k-XOR: To generate a random k-XOR formula, we fix a set of n Boolean variables $x_1, \ldots, x_n$, and generate m random constraints $C_1, \ldots, C_m$ as follows. For each constraint C_i,

1. choose k variables $(x_{i_1}, \ldots, x_{i_k})$ uniformly at random from all $\binom{[n]}{k}$ subsets of size k, and
2. choose $b_i \in \{0,1\}$ with equal probability.

Underlying Hypergraph: Given a k-XOR instance of n variables and m constraints, we consider the corresponding hypergraph $H = ([n], E)$ of n vertices and m hyperedges of size k, where each constraint $x_{i_1} + x_{i_2} + \cdots + x_{i_k} = b_i \mod 2$ corresponds to a hyperedge $(i_1, \ldots, i_k)$ in E.

For a random k-XOR instance, the corresponding hypergraph is a random hypergraph of n vertices and m hyperedges of size k in the standard model.

Spectral Approach

Product Form: For the spectral approach, we consider the equivalent product form of a constraint

$$x_{i_1} \cdot x_{i_2} \cdot \ldots \cdot x_{i_k} = b_i \quad \text{where each } x_{i_j} \in \{\pm 1\} \text{ and } b_i \in \{\pm 1\},$$

for each constraint $x_{i_1} + x_{i_2} + \cdots + x_{i_k} = b_i \mod 2$ of the additive form.

Bias Polynomial: Using this product form, we can express the number of constraints satisfied by an assignment $x \in \{\pm 1\}^n$ using the following bias polynomial:

$$\Psi(x) = \sum_{e \in H} b_e \prod_{i \in e} x_i, \tag{24.1}$$

where $e \in H$ denotes a hyperedge e in H, and $i \in e$ denotes a vertex i in the hyperedge e. For an assignment $x \in \{\pm 1\}^n$, note that $\Psi(x)$ is equal to the number of constraints satisfied by x minus the number of constraints unsatisfied by x. Thus, $\Psi(x) = m$ if and only if x is a satisfying assignment, and strongly refutation is equivalent to proving that

$$\max_{x \in \{\pm 1\}^n} \Psi(x) \leq \epsilon m \quad \text{for some small constant } \epsilon > 0.$$

2-XOR and Quadratic Form: For 2-XOR, the bias polynomial takes the simple form

$$\Psi(x) = \sum_{ij \in H} b_{ij} x_i x_j,$$

which can be expressed as the quadratic form $x^\top B x$ of the adjacency matrix, where

$$B(i,j) = \begin{cases} b_{ij} & \text{if } ij \in H \\ 0 & \text{otherwise.} \end{cases} \tag{24.2}$$

Therefore, strong refutation reduces to proving a strong upper bound on the maximum eigenvalue:

$$2 \max_{x \in \{\pm 1\}^n} \Psi(x) = \max_{x \in \{\pm 1\}^n} x^\top B x \le \max_{x \in \{\pm 1\}^n} \lambda_{\max}(B) \cdot \|x\|_2^2 = n \cdot \lambda_{\max}(B), \tag{24.3}$$

where the inequality follows from the characterization of the maximum eigenvalue in Lemma A.12. This naturally leads to a spectral approach for CSP refutation.

Matrix Concentration: To strongly refute a *random* 2-XOR formula, we can apply the matrix concentration inequalities in Chapter 23. For an edge $ij \in H$, let $A_{ij} = \chi_i \chi_j^\top + \chi_j \chi_i^\top$ where $\chi_i \in \mathbb{R}^n$ denotes the characteristic vector of the vertex i. Then

$$B = \sum_{ij \in H} b_{ij} A_{ij},$$

which is a random sum of the matrices $\{A_{ij}\}_{ij \in H}$. To bound $\lambda_{\max}(B)$, we can either apply the matrix Bernstein inequality in Theorem 23.12 or the matrix Khintchine inequality in Theorem 23.16. In either case, we need to compute

$$\mathbb{E}\left[B^2\right] = \sum_{ij \in H} A_{ij}^2 = \sum_{ij \in H} \left(\chi_i \chi_i^\top + \chi_j \chi_j^\top\right) = D_H, \tag{24.4}$$

where D_H denotes the diagonal degree matrix of the hypergraph H, with $D_H(i,i)$ equal to the number of hyperedges in H that contains vertex i. Since D_H is a diagonal matrix, it follows that

$$\left\|\mathbb{E}\left[B^2\right]\right\|_{\mathrm{op}} = \max_i \{D_H(i,i)\} = \deg_{\max}(H),$$

denoting the maximum degree of a vertex in H. The matrix Khintchine inequality then implies that

$$\mathbb{E}\left[\|B\|_{\mathrm{op}}\right] \lesssim \sqrt{\log n} \cdot \left\|\mathbb{E}\left[B^2\right]\right\|_{\mathrm{op}}^{\frac{1}{2}} = \sqrt{\log n} \cdot \sqrt{\deg_{\max}(H)}.$$

Strong Refutation of Random 2-XOR: It follows from the above bound and (24.3) that

$$\mathbb{E}\left[\max_{x \in \{\pm 1\}^n} \Psi(x)\right] \lesssim n \cdot \sqrt{\log n} \cdot \sqrt{\deg_{\max}(H)}.$$

Suppose H is a regular graph. Then $\deg_{\max}(H) = \frac{m}{n}$, and thus

$$\mathbb{E}\left[\max_{x \in \{\pm 1\}^n} \Psi(x)\right] \lesssim \sqrt{n \log n} \cdot \sqrt{m} \le \epsilon \cdot m, \qquad \text{when } m \ge \frac{n \log n}{\epsilon^2}.$$

Since H is a *random* graph, it follows from standard concentration arguments (i.e., Chernoff bound and a union bound) that $\deg_{\max}(H) \lesssim \frac{m}{n}$ when $m \gtrsim n \log n$ with high probability. We omit these details as they are standard and also because they can be bypassed; see Section 24.3. One can apply the matrix Bernstein inequality (with $t \asymp \sqrt{\frac{m}{n} \log n}$) to prove the following conclusion.

Theorem 24.1 (Strong Refutation of Random 2-XOR). *The spectral algorithm strongly refutes a random 2-XOR formula with n variables and $m \gtrsim n \log n$ constraints with high probability.*

Note that this yields a near-linear time algorithm for strongly refuting a random 2-XOR formula with $m \gtrsim n \log n$ constraints with high probability, by constructing the matrix B and computing its maximum eigenvalue using the power method.

We also note that there is a polynomial time semidefinite programming algorithm that strongly refutes a random 2-XOR formula with $m \gtrsim n/\epsilon^2$ constraints with high probability; see [AGK21].

24.2 Random k-XOR: Tensor Flattening

In this section, we extend the spectral approach to strongly refute random k-XOR formulas for even $k \geq 4$. To ease notation, we focus on the $k = 4$ case, and the generalization to larger even k is straightforward. For 4-XOR, the bias polynomial is a degree-4 polynomial

$$\Psi(x) = \sum_{(i,j,k,l) \in H} b_{ijkl} \cdot x_i x_j x_k x_l,$$

where $H = ([n], E)$ is a 4-uniform hypergraph with n vertices and m hyperedges, and $b_{ijkl} \in \{\pm 1\}$ for each hyperedge $(i, j, k, l) \in H$.

Tensor Norm

Strong refutation of 4-XOR can be reduced to a tensor injective norm problem (see e.g., [BGJ+25]). Let $T \in \mathbb{R}^{n \times n \times n \times n}$ be the 4-dimensional tensor associated with the 4-XOR formula, where

$$T(i,j,k,l) = \begin{cases} b_{ijkl} & \text{if } (i,j,k,l) \in H, \\ 0 & \text{otherwise,} \end{cases}$$

and let $x^{\otimes 4} \in \mathbb{R}^{n \times n \times n \times n}$ be the rank-one 4-tensor with $x^{\otimes 4}{}_{i,j,k,l} = x_i x_j x_k x_l$. Then

$$24 \max_{x \in \{\pm 1\}^n} \Psi(x) = \max_{x \in \{\pm 1\}^n} \langle T, x^{\otimes 4} \rangle \leq \max_{x \in \{\pm 1\}^n} \|T\|_{\text{inj}} \cdot \|x\|_2^4 = n^2 \cdot \|T\|_{\text{inj}},$$

where the tensor injective norm is defined as

$$\|T\|_{\text{inj}} := \sup_{\|x_1\|_2 \leq 1, \ldots, \|x_4\|_2 \leq 1} \langle T, x_1 \otimes x_2 \otimes x_3 \otimes x_4 \rangle.$$

Unfortunately, estimating the tensor injective norm is not nearly as well understood, and there are no known tensor concentration inequalities for random tensors as sharp as matrix concentration inequalities for random matrices.

Tensor Flattening

A natural approach to bypass the tensor injective norm problem is to reduce it to a matrix norm problem by "flattening" the tensor to a matrix. Let B be an $\binom{n}{2} \times \binom{n}{2}$ matrix associated with the 4-XOR formula, defined as

$$B((i,j),(k,l)) = \begin{cases} b_{ijkl} & \text{if } (i,j,k,l) \in H \\ 0 & \text{otherwise,} \end{cases} \tag{24.5}$$

where each row and column of B is indexed by a pair of distinct vertices (i,j) in H. Let $x^{\circ 2} \in \mathbb{R}^{\binom{n}{2}}$ be the vector with $x^{\circ 2}(i,j) = x_i x_j$ for each pair of distinct vertices (i,j). Then

$$6 \max_{x\in\{\pm 1\}^n} \Psi(x) = \max_{x\in\{\pm 1\}^n} \left(x^{\circ 2}\right)^\top B\left(x^{\circ 2}\right) \le \max_{x\in\{\pm 1\}^n} \|B\|_{\mathrm{op}} \cdot \left\|x^{\circ 2}\right\|_2^2 = \binom{n}{2} \cdot \|B\|_{\mathrm{op}}. \tag{24.6}$$

This reduces the problem to bounding the matrix norm of B.

Matrix Concentration and Strong Refutation

As for random 2-XOR in Section 24.1, we can write $B = \sum_{ijkl\in H} b_{ijkl} \cdot A_{ijkl}$ as a random sum of matrices, and compute $\mathbb{E}\left[B^2\right] = \sum_{ijkl\in H} (A_{ijkl})^2 = D_H$ as in (24.4), where D_H is a diagonal degree matrix with $D_H(ij,ij)$ equal to the number of hyperedges in H that contains the pair of vertices (i,j). The matrix Khintchine inequality in Theorem 23.16 then implies that

$$\mathbb{E}\left[\|B\|_{\mathrm{op}}\right] \lesssim \sqrt{\log n} \cdot \sqrt{\deg_{\max}(H)} \quad \text{where } \deg_{\max}(H) := \max_{(i,j)} \{D_H(ij,ij)\}.$$

It follows from the above bound and (24.6) that

$$\mathbb{E}\left[\max_{x\in\{\pm 1\}^n} \Psi(x)\right] \lesssim n^2 \cdot \sqrt{\log n} \cdot \sqrt{\deg_{\max}(H)}.$$

Assuming $\deg_{\max}(H) \lesssim m/\binom{n}{2}$ and $m \ge (n^2 \log n)/\epsilon^2$, then

$$\mathbb{E}\left[\max_{x\in\{\pm 1\}^n} \Psi(x)\right] \lesssim n\sqrt{\log n} \cdot \sqrt{m} \le \epsilon \cdot m.$$

Since H is a random hypergraph, it follows from standard concentration arguments that indeed $\deg_{\max}(H) \lesssim m/\binom{n}{2}$ when $m \gtrsim n^2 \log n$ with high probability. One can then apply the matrix Bernstein inequality to prove the following result formally.

Theorem 24.2 (Strong Refutation of Random k-XOR for even k)**.** *The tensor flattening algorithm strongly refutes a random k-XOR formula for even k with n variables and $m \gtrsim n^{\frac{k}{2}} \log n$ constraints with high probability.*

Using the XOR principle stated in Section 24.1, this yields a spectral algorithm to strongly refute a random k-SAT formula for even k when $m \gtrsim n^{\frac{k}{2}} \log n$ with high probability. This represents a significant improvement over the $\Omega(n^k)$ constraints required for refuting an unsatisfiable k-SAT formula in the worst case, highlighting the power of the spectral approach in the average-case setting.

24.3 Semi-Random CSP: Reweighting Trick

In this section, we extend the results in the previous sections to the more general semi-random setting, using the reweighting trick introduced by Hsieh, Kothari, and Mohanty [HKM23].

Random and Semi-Random k-XOR

In a random k-XOR instance of n variables and m constraints, we are given a random k-uniform hypergraph $H = ([n], E)$ with m hyperedges, where each hyperedge is sampled uniformly and independently from all subsets of size k, and a random value $b_e \in \{\pm 1\}$ for each hyperedge $e \in E(H)$.

In a semi-random k-XOR instance of n variables and m constraints, we are given an arbitrary k-uniform hypergraph $H = ([n], E)$ with m hyperedges, and a random value $b_e \in \{\pm 1\}$ for each hyperedge $e \in E(H)$. This problem is strictly harder, as the refutation algorithm needs to succeed with high probability (over the choices of b_e) for every k-uniform hypergraph with m hyperedges.

In the analyses for random k-XOR in the previous sections, we use the randomness of $E(H)$ and $\{b_e\}_{e \in E(H)}$ separately in two steps. In the first step, we consider a random matrix $B = \sum_{e \in E(H)} b_e M_e$ over the random choices of b_e, and apply the matrix Khintchine inequality to bound

$$\mathbb{E}\left[\|B\|_{\mathrm{op}}\right] \lesssim \sqrt{\log n} \cdot \sqrt{\deg_{\max}(H)}.$$

Note that this step holds for an arbitrary k-uniform hypergraph. In the second step, we use the randomness of $E(H)$ to show that, with high probability,

$$\text{the maximum degree} \lesssim \text{the average degree},$$

when the average degree is sufficiently large. This average degree depends on the setting: $2m/n$ for the adjacency matrix in 2-XOR (see Section 24.1), and $2m/\binom{n}{2}$ for the flattened matrix in 4-XOR (see Section 24.2). In all cases, the average degree bound is crucial in proving the theorems in the previous sections.

In a semi-random instance, however, the maximum degree can be much bigger than the average degree, and this approach appears to break down. The question of strongly refuting semi-random CSPs was posed as a direction for future work in [AOW15]. Subsequently, significant techniques (such as partitioning into pseudorandom instances) were developed by Abascal, Guruswami, and Kothari [AGK21] to extend the results from the random setting to the semi-random setting.

Reweighting Trick

It is therefore remarkable that Hsieh, Kothari, and Mohanty [HKM23] introduced a clever reweighting trick that overcomes this issue and extends the guarantees from the random setting to the semi-random setting almost effortlessly. The reweighting trick was originally introduced to analyze the combinatorial trace method for proving the hypergraph Moore bound. The adaptation to the setting of matrix concentration inequalities is from Kothari's course notes on matrix concentration.

Motivation: To motivate this trick, consider the simplest setting of the adjacency matrix for 2-XOR in Section 24.1. In this setting, recall that $A_{ij} = \chi_i \chi_j^\top + \chi_j \chi_i^\top$ for an edge $ij \in H$, and

$$B = \sum_{ij \in H} b_{ij} A_{ij} \quad \text{and} \quad \mathbb{E}\left[B^2\right] = \sum_{ij \in H} A_{ij}^2 = D_H,$$

where D_H is the diagonal degree matrix with $D_H(i,i)$ equal to the degree of i in H. It follows from the matrix Khintchine inequality in Theorem 23.16 that

$$\mathbb{E}\left[\|B\|_{\mathrm{op}}\right] \lesssim \sqrt{\log n} \cdot \left\|\mathbb{E}\left[B^2\right]\right\|_{\mathrm{op}}^{\frac{1}{2}} = \sqrt{\log n} \cdot \sqrt{\deg_{\max}(H)}.$$

The dependency on the maximum degree in the operator norm is unavoidable, and this step is essentially tight. However, observe that when the maximum degree is much larger than the average degree, any approximate maximizer x of the quadratic form $x^\top Bx$ must place significant mass on the high-degree vertices (e.g., consider a maximum eigenvector of a star graph), and hence vectors in $\{\pm 1\}^n$ are far from being approximate maximizers. This means that the previous relaxation in

$$2 \max_{x\in\{\pm1\}^n} \Psi(x) = \max_{x\in\{\pm1\}^n} x^\top Bx \le \max_{x\in\{\pm1\}^n} \|B\|_{\mathrm{op}} \cdot \|x\|_2^2 = n \cdot \|B\|_{\mathrm{op}} \tag{24.7}$$

cannot be tight.

Reweighting: The idea is to consider a reweighted matrix by applying a rescaling

$$\Gamma^{-\frac{1}{2}} B \Gamma^{-\frac{1}{2}} = \sum_{ij\in H} b_{ij} \cdot \Gamma^{-\frac{1}{2}} A_{ij} \Gamma^{-\frac{1}{2}}$$

to the rows and columns of B, so that $\mathbb{E}\big[\big(\Gamma^{-\frac{1}{2}} B \Gamma^{-\frac{1}{2}}\big)^2\big]$ is more balanced and the relaxation in (24.8) becomes tighter, where Γ is a diagonal matrix. We then proceed as follows:

$$\begin{aligned} 2 \max_{x\in\{\pm1\}^n} \Psi(x) &= \max_{x\in\{\pm1\}^n} x^\top \Gamma^{\frac{1}{2}} \big(\Gamma^{-\frac{1}{2}} B \Gamma^{-\frac{1}{2}}\big) \Gamma^{\frac{1}{2}} x \\ &\le \max_{x\in\{\pm1\}^n} \big\|\Gamma^{-\frac{1}{2}} B \Gamma^{-\frac{1}{2}}\big\|_{\mathrm{op}} \cdot \big\|\Gamma^{\frac{1}{2}} x\big\|_2^2 \\ &= \big\|\Gamma^{-\frac{1}{2}} B \Gamma^{-\frac{1}{2}}\big\|_{\mathrm{op}} \cdot \mathrm{Tr}(\Gamma). \end{aligned} \tag{24.8}$$

A natural choice is $\Gamma := D_H$, but this does not work, as we will see. The key choice in [HKM23] is

$$\Gamma = D_H + \overline{d} \cdot I \quad \text{where} \quad \overline{d} = \frac{2m}{n}. \tag{24.9}$$

To apply the matrix Khintchine inequality to bound the expected operator norm, we compute

$$\mathbb{E}\left[\big(\Gamma^{-\frac{1}{2}} B \Gamma^{-\frac{1}{2}}\big)^2\right] = \sum_{ij\in H} \left(\Gamma^{-\frac{1}{2}} A_{ij} \Gamma^{-\frac{1}{2}}\right)^2 = \mathrm{diag}\left(\left\{ \sum_{j: ij\in H} \frac{1}{\big(\deg(i)+\overline{d}\big)\big(\deg(j)+\overline{d}\big)} \right\}_{i\in[n]}\right),$$

as each $\big(\Gamma^{-\frac{1}{2}} A_{ij} \Gamma^{-\frac{1}{2}}\big)^2$ contributes $1/\big(\Gamma(i,i)\cdot\Gamma(j,j)\big)$ to the i-th and j-th diagonal entries. Thus,

$$\left\|\mathbb{E}\left[\big(\Gamma^{-\frac{1}{2}} B \Gamma^{-\frac{1}{2}}\big)^2\right]\right\|_{\mathrm{op}} = \max_{i\in[n]} \sum_{j: ij\in H} \frac{1}{\big(\deg(i)+\overline{d}\big)\big(\deg(j)+\overline{d}\big)} \le \max_{i\in[n]} \frac{1}{\deg(i)+\overline{d}} \cdot \frac{\deg(i)}{\overline{d}} \le \frac{1}{\overline{d}}. \tag{24.10}$$

It follows from the matrix Khintchine inequality in Theorem 23.16 that

$$\mathbb{E}\left[\big\|\Gamma^{-\frac{1}{2}} B \Gamma^{-\frac{1}{2}}\big\|_{\mathrm{op}}\right] \lesssim \sqrt{\log n} \cdot \left\|\mathbb{E}\left[\big(\Gamma^{-\frac{1}{2}} B \Gamma^{-\frac{1}{2}}\big)^2\right]\right\|_{\mathrm{op}}^{\frac{1}{2}} = \sqrt{\frac{\log n}{\overline{d}}} = \sqrt{\frac{n \log n}{2m}}.$$

Plugging this bound into (24.8) and noting that $\mathrm{Tr}(\Gamma) = \sum_{i\in[n]} (\deg(i)+\overline{d}) = 4m$, we conclude that

$$\mathbb{E}\left[\max_{x\in\{\pm1\}^n} \Psi(x)\right] \lesssim \sqrt{mn\log n} \le \epsilon m \quad \text{when } m \ge \frac{n\log n}{\epsilon^2}.$$

This proves that the spectral algorithm strongly refutes a semi-random 2-XOR formula whenever $m \gtrsim n \log n$, thereby extending Theorem 24.1 to the semi-random setting.

Remark 24.3 (Choice of Γ). *If we set $\Gamma := D_H$, then we can only bound $\big\|\mathbb{E}[\big(\Gamma^{-\frac{1}{2}} B \Gamma^{-\frac{1}{2}}\big)^2]\big\|_{\mathrm{op}} \lesssim 1$ while $\mathrm{Tr}(\Gamma) = 2m$. By setting $\Gamma := D_H + \overline{d}\cdot I$, we gain a factor of $\overline{d}$ from $\big\|\mathbb{E}[\big(\Gamma^{-\frac{1}{2}} B \Gamma^{-\frac{1}{2}}\big)^2]\big\|_{\mathrm{op}}$ while only losing a factor of 2 from $\mathrm{Tr}(\Gamma)$.*

24.4 Random 3-XOR: Cauchy-Schwarz Trick

So far, the spectral algorithms we have seen apply to refuting semi-random k-XOR formulas when k is an even integer. When k is odd, such as $k = 3$, there are no natural matrices associated with the k-XOR formula, and it is not clear how to flatten the tensor or to define a Kikuchi matrix.

Indeed, refuting semi-random k-XOR formulas when k is odd is considerably more challenging, requiring significantly more work even with the reweighting trick [AGK21, HKM23].

In this section, we only consider the simplest case of strongly refuting random 3-XOR formulas, illustrating the Cauchy-Schwarz trick [BM16] that underlies these works, following the presentation in [AGK21] via matrix concentration.

Theorem 24.4 (Strong Refutation of Random 3-XOR)**.** *There exists a spectral algorithm that strongly refutes a random 3-XOR formula with n variables and $m \gtrsim n^{\frac{3}{2}}\sqrt{\log n}$ constraints with high probability.*

Cauchy-Schwarz Trick

The starting point is to write a 3-XOR instance as a collection of 2-XOR instances. Given a 3-uniform hypergraph $H = ([n], E)$ and a value $b_e \in \{\pm 1\}$ for each hyperedge $e \in E(H)$, we define n matrices $B_1, \dots, B_n \in \mathbb{R}^{n \times n}$ where

$$B_\ell(i,j) = \begin{cases} b_{\ell ij} & \text{if } (\ell, i, j) \in E(H) \text{ and } \ell < i \text{ and } \ell < j, \\ 0 & \text{otherwise.} \end{cases}$$

Then the bias polynomial can be expressed as

$$2\Psi(x) = \sum_{\ell ij \in E(H)} b_{\ell ij} \cdot x_\ell x_i x_j = \sum_{\ell \in [n]} x_\ell \cdot \sum_{i,j > \ell \,:\, \ell ij \in E(H)} b_{\ell ij} \cdot x_i x_j = \sum_{\ell \in [n]} x_\ell \cdot x^\top B_\ell x.$$

By construction, each $b_{\ell ij}$ appears in exactly one matrix, so the matrices $B_1, \dots, B_n$ are independent random matrices.

The Cauchy-Schwarz trick is to decouple these 2-XOR instances using the Cauchy-Schwarz inequality and the fact that $x_\ell^2 = 1$ for $\ell \in [n]$, so that

$$4\Psi(x)^2 \le \sum_{\ell \in [n]} x_\ell^2 \cdot \sum_{\ell \in [n]} \left(x^\top B_\ell x\right)^2 = n \cdot \sum_{\ell \in [n]} \left(x^\top B_\ell x\right)^2 = nm + n \cdot \sum_{\ell \in [n]} \left(x^{\otimes 2}\right)^\top \left(B_\ell \tilde{\otimes} B_\ell\right)\left(x^{\otimes 2}\right),$$

where $x^{\otimes 2}$ is the n^2-dimensional vector with $x^{\otimes 2}(i,j) = x_i \cdot x_j$, and $B_\ell \tilde{\otimes} B_\ell$ is the usual $n^2 \times n^2$ tensor product matrix with $(B_\ell \tilde{\otimes} B_\ell)((i_1, i_2), (j_1, j_2)) = B_\ell(i_1, j_1) \cdot B_\ell(i_2, j_2)$, except that we zero out all the entries with $i_1 = i_2$ and $j_1 = j_2$ as will be justified below. Note that the contribution of these entries to the quadratic form is equal to $\sum_{\ell \in [n]} \sum_{i,j > \ell \,:\, \ell ij \in E(H)} B_\ell(i,j)^2 \cdot x_i^2 \cdot x_j^2 = m$, which is accounted for in the above calculation.

With this matrix formulation, we can apply the usual relaxation to bound the operator norm:

$$4 \max_{x \in \{\pm 1\}^n} \Psi(x)^2 \le mn + \max_{x \in \{\pm 1\}^n} n \cdot \left\| \sum_{\ell \in [n]} B_\ell \tilde{\otimes} B_\ell \right\|_{\mathrm{op}} \cdot \left\| x^{\otimes 2} \right\|_2^2 = mn + n^3 \cdot \left\| \sum_{\ell \in [n]} B_\ell \tilde{\otimes} B_\ell \right\|_{\mathrm{op}}. \quad (24.11)$$

Matrix Concentration

We cannot apply the matrix Khintchine inequality as usual, because each entry of the matrix $B_\ell \tilde{\otimes} B_\ell$ is a product of two random variables. We instead apply the matrix Bernstein inequality in Theorem 23.12. Let $Z = \sum_{\ell \in [n]} B_\ell \tilde{\otimes} B_\ell$ denote the random sum.

Zero Mean: By construction, $\mathbb{E}\left[B_\ell \tilde{\otimes} B_\ell\right] = 0$ for each $\ell \in [n]$, as all nonzero entries are products of two zero-mean independent random variables. This is the reason we zero out those entries in $B_\ell \otimes B_\ell$ with $B_\ell(i,j)^2$.

Random Hypergraph and Hyperedge Removal: Since H is a random 3-uniform hypergraph, when $m \le o(n^2)$, a standard concentration argument implies that, with high probability, there are no two hyperedges e_1, e_2 such that $|e_1 \cap e_2| \ge 2$ after removing at most $o(m)$ hyperedges. We remove these $o(m)$ hyperedges from the instance and assume that the corresponding constraints are all satisfied in the worst case.

Norm Bound: After removing those hyperedges, each row of B_ℓ has at most one nonzero entry. This implies that each row of $B_\ell \tilde{\otimes} B_\ell$ has at most one nonzero entry of the form $-1 \le b_{\ell i_1 j_1} \cdot b_{\ell i_2 j_2} \le 1$. Therefore, each $\|B_\ell \tilde{\otimes} B_\ell\|_{\mathrm{op}} = 1$.

Variance: Since each row of $B_\ell \tilde{\otimes} B_\ell$ has at most one nonzero entry with a value in $\{\pm 1\}$, each $(B_\ell \tilde{\otimes} B_\ell)^2$ is a diagonal matrix with each diagonal entry at most one, where the $((i_1,i_2),(i_1,i_2))$-entry is one only if ℓ is a common neighbor of i_1 and i_2. As each $B_\ell \tilde{\otimes} B_\ell$ is independent and zero-mean,

$$\sigma^2(Z) = \left\|\mathbb{E}\left[Z^2\right]\right\| = \left\|\sum_{\ell=1}^{n} \mathbb{E}\left[(B_\ell \tilde{\otimes} B_\ell)^2\right]\right\|_{\mathrm{op}} = \|D_H\|_{\mathrm{op}} \le \mathrm{codeg}_{\max}(H),$$

where D_H is a diagonal matrix with the $((i_1,i_2),(i_1,i_2))$-entry being at most the number of common neighbors of i_1 and i_2, and $\mathrm{codeg}_{\max}(H)$ denotes the maximum number of common neighbors of a pair of vertices. Since H is a random 3-uniform hypergraph, a standard probabilistic argument shows that, with high probability,

$$\mathrm{codeg}_{\max}(H) \lesssim \frac{m^2}{n^3} \quad \text{whenever } m \gtrsim n^{\frac{3}{2}}\sqrt{\log n}.$$

Matrix Bernstein: Therefore, by applying the matrix Bernstein inequality in Theorem 23.12 with $\sigma^2(Z) \lesssim m^2/n^3$ and $L = 1$, it follows that with high probability

$$\|Z\|_{\mathrm{op}} \lesssim \sqrt{\frac{m^2 \log n}{n^3}} \quad \text{whenever } m \gtrsim n^{\frac{3}{2}}\sqrt{\log n}.$$

Plugging back into (24.11), we conclude that, with high probability,

$$\max_{x \in \{\pm 1\}^n} \Psi(x)^2 \lesssim mn + n^3\sqrt{\frac{m^2 \log n}{n^3}} = mn + mn^{\frac{3}{2}}\sqrt{\log n} \le \epsilon^2 m^2 \quad \text{whenever } m \ge \frac{n^{\frac{3}{2}}\sqrt{\log n}}{\epsilon^2}.$$

This proves the strong refutation result for random 3-XOR in Theorem 24.4.

Semi-Random k-XOR: When k is odd, this approach requires dealing with $\mathrm{codeg}_{\max}(H)$, which adds considerable difficulty to the analysis. We refer the reader to [HKM23] for a proof of Theorem 25.3, along with the definition of the "colored" Kikuchi matrices when k is odd.

References

[AGK21] Jackson Abascal, Venkatesan Guruswami, and Pravesh K Kothari. Strongly refuting all semi-random boolean csps. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 454–472. SIAM, 2021. 298, 300, 302, 304

[AOW15] Sarah R Allen, Ryan O'Donnell, and David Witmer. How to refute a random csp. In *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*, pages 689–708. IEEE, 2015. 298, 302

[BGJ+25] Afonso S Bandeira, Sivakanth Gopi, Haotian Jiang, Kevin Lucca, and Thomas Rothvoss. Tensor concentration inequalities: A geometric approach. In *Proceedings of the 57th Annual ACM Symposium on Theory of Computing*, pages 822–832, 2025. 300

[BM16] Boaz Barak and Ankur Moitra. Noisy tensor completion via the sum-of-squares hierarchy. In *Conference on Learning Theory*, pages 417–445. PMLR, 2016. 304

[Fei02] Uriel Feige. Relations between average case complexity and approximation complexity. In *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, pages 534–543, 2002. 298

[FO07] Uriel Feige and Eran Ofek. Easily refutable subformulas of large random 3cnf formulas. *Theory of Computing*, 3(1):25–43, 2007. 298

[HKM23] Jun-Ting Hsieh, Pravesh K Kothari, and Sidhanth Mohanty. A simple and sharper proof of the hypergraph moore bound. In *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2324–2344. SIAM, 2023. 9, 301, 302, 303, 304, 306, 311, 315, 318

Chapter 25

Kikuchi Matrices and Applications

The sum-of-squares method has been the most powerful framework for designing algorithms for the tensor PCA problem and CSP refutation, achieving much stronger guarantees than standard spectral algorithms, and is conjectured to be optimal.

In a remarkable work, Wein, El Alaoui, and Moore [WEAM25] redeemed the spectral approach by introducing *Kikuchi matrices*, showing that they can be used to design simpler and faster algorithms that match the guarantees of sum-of-squares algorithms. Moreover, this approach considerably simplifies the analysis, reducing it to direct applications of matrix concentration inequalities.

Subsequently, Kikuchi matrices have found significant applications, including proving lower bounds for locally decodable codes and resolving Feige's conjecture on the hypergraph Moore bound. In this chapter, we present the Kikuchi matrix method and these applications.

25.1 Smooth Tradeoff for CSP Refutation

Probabilistic methods show that a random k-XOR formula is strongly refutable with high probability when $m \geq \omega(n)$. However, there remains a significant gap between the constraint-to-variable ratio at which efficient strong refutation algorithms are known and the ratio at which instances become strongly refutable with high probability.

Smooth Tradeoff

Using the sum-of-squares framework, Raghavendra, Rao, and Schramm [RRS17] establish a smooth tradeoff between time complexity and the constraint-to-variable ratio.

Theorem 25.1 (Smooth Tradeoff for Strong Refutation of Random k-XOR)**.** *For any $\delta \in [0, 1)$, there is an algorithm that strongly refutes a random k-XOR formula with n variables and m constraints with high probability, when*

$$m \gtrsim n^{\frac{k}{2}+\delta(1-\frac{k}{2})} \cdot \log n, \qquad \textit{with time complexity } 2^{O(n^\delta)}.$$

In particular, there is a spectral algorithm achieving the same guarantee by computing the maximum eigenvalue of a $2^{O(n^\delta)} \times 2^{O(n^\delta)}$ matrix.

For example, to strongly refute a random 4-XOR formula with $m \gtrsim n^{1.9} \log n$, the algorithm runs in time $2^{O(n^{0.1})}$, which is subexponential in n. In contrast, to strongly refute a random 4-XOR formula with $m \gtrsim n \log n$, the algorithm requires exponential time.

Theorem 25.1 matches the sum-of-squares lower bounds by Grigoriev [Gri01] and Schoenebeck [Sch08] up to polylogarithmic factors, and is conjectured to be essentially optimal. However, both the analysis and even the matrix description in [RRS17] are quite involved.

Kikuchi Matrices for Even Arity

Wein, El Alaoui, and Moore [WEAM25] introduced the Kikuchi matrix, providing a simple and elegant spectral algorithm that matches the guarantee in Theorem 25.1 when k is an even integer.

Limitation of Previous Approach: To see the limitation of the previous spectral approach for random 4-XOR in Section 24.2, note that the inequality in (24.6) is a tight relaxation if and only if the quadratic form of B is approximately maximized at a vector $x^{\circ 2}$ for some $x \in \{\pm 1\}^n$. In the regime where $m \ll \binom{n}{2}$, this cannot be the case, as B is an $\binom{n}{2} \times \binom{n}{2}$ matrix with only $O(m)$ nonzero entries (each hyperedge only contributes 6 nonzero entries to B), and so an approximate maximizer to the quadratic form must have small support.

Desired Properties: To make this spectral approach work, we would like to construct a matrix K associated with a random k-XOR formula with the following properties:

1. $\Psi(x)$ is related to $\left(x^{\circ \ell}\right)^\top K \left(x^{\circ \ell}\right)$ for some integer ℓ;
2. The quadratic form of K is approximately maximized at $x^{\circ \ell}$ for some $x \in \{\pm 1\}^n$.

Kikuchi Matrices: The Kikuchi matrix (or the symmetric difference matrix) can be viewed as a generalization of the matrix flattening in Section 24.2 that satisfies these two properties.

Definition 25.2 (Kikuchi Matrix for Even Arity)**.** *Given a k-XOR instance for an even integer k with underlying k-uniform hypergraph $H = ([n], E)$ and value $b_e \in \{\pm 1\}$ for each $e \in E(H)$, for $\frac{k}{2} \le \ell \le n$, the order-$\ell$ Kikuchi matrix K is an $\binom{n}{\ell} \times \binom{n}{\ell}$ matrix where each row and column corresponds to a subset $S \subseteq [n]$ with $|S| = \ell$, and*

$$K(S,T) = \begin{cases} b_e & \text{if } S \oplus T = e \text{ for some } e \in E(H), \\ 0 & \text{otherwise,} \end{cases} \tag{25.1}$$

where $S \oplus T = (S - T) \cup (T - S)$ denotes the symmetric difference of the two sets S and T.

Note that the matrix flattening in (24.5) is the order-2 Kikuchi matrix of the 4-XOR instance, and the adjacency matrix in (24.2) is the order-1 Kikuchi matrix of the 2-XOR instance.

For a fixed hyperedge e of size k, the number of pairs $S, T \subseteq [n]$ such that $|S| = |T| = \ell$ and $S \oplus T = e$ is equal to

$$\Delta := \binom{k}{\frac{k}{2}} \cdot \binom{n-k}{\ell - \frac{k}{2}}, \tag{25.2}$$

as such pairs S, T must satisfy $|S \cap e| = |T \cap e| = \frac{k}{2}$ and $S \setminus e = T \setminus e$.

Quadratic Form: Let $x^{\circ\ell} \in \mathbb{R}^{\binom{n}{\ell}}$ be the vector with

$$x^{\circ\ell}(S) = \prod_{i \in S} x_i, \qquad \text{for each } S \subseteq [n] \text{ with } |S| = \ell.$$

For $x \in \{\pm 1\}^n$, the quadratic form satisfies

$$\begin{aligned}
\left(x^{\circ\ell}\right)^\top K \left(x^{\circ\ell}\right) &= \sum_{S,T : |S|=|T|=\ell} K(S,T) \cdot x^{\circ\ell}(S) \cdot x^{\circ\ell}(T) \\
&= \sum_{S,T : |S|=|T|=\ell} K(S,T) \cdot x^{\circ\ell}(S \oplus T) \\
&= \Delta \cdot \sum_{e \in H} b_e \cdot \prod_{i \in e} x_i \\
&= \Delta \cdot \Psi(x), \qquad (25.3)
\end{aligned}$$

where the second line uses that $x_i^2 = 1$ for $i \in S \cap T$, and the third line follows by regrouping terms according to $e = S \oplus T$, noting that each hyperedge contributes exactly Δ pairs (S,T). This establishes the first desired property, namely that the bias polynomial can be expressed via the quadratic form of the Kikuchi matrix.

Eigenbasis: For some intuition for the second desired property, consider the full $2^n \times 2^n$ symmetric difference matrix F of a k-XOR instance, where each row and column of F is indexed by a subset $S \subseteq [n]$, with

$$F(S,T) = \begin{cases} b_e & \text{if } S \oplus T = e \text{ for some } e \in E(H), \\ 0 & \text{otherwise.} \end{cases}$$

For each $x \in \{\pm 1\}^n$, let $x^\star \in \{\pm 1\}^{2^n}$ be the vector with

$$x^\star(S) = \prod_{i \in S} x_i, \qquad \text{for each } S \subseteq [n].$$

Observe that $x^\star$ is an eigenvector of F with eigenvalue $\sum_{e \in E(H)} b_e \cdot x^\star(e)$, as

$$(Fx^\star)(S) = \sum_{T \subseteq [n]} F(S,T) \cdot x^\star(T) = \sum_{T \subseteq [n]} F(S,T) \cdot x^\star(S \oplus T) \cdot x^\star(S) = x^\star(S) \cdot \sum_{e \in E(H)} b_e \cdot x^\star(e),$$

where the second equality uses that $x_i^2 = 1$ for $i \in S \cap T$, and the last equality follows since $S \oplus T$ ranges over all subsets of $[n]$ as T ranges over all subsets of $[n]$.

Note that $x^\star$ and $y^\star$ are orthogonal for $x, y \in \{\pm 1\}^n$ with $x \neq y$. Since there are 2^n such vectors, $\{x^\star\}_{x \in \{\pm 1\}^n}$ forms an eigenbasis of the full symmetric difference matrix F (the Fourier basis). In particular, this implies that the quadratic form of F is maximized at $x^\star$ for some $x \in \{\pm 1\}^n$. Hence, by computing the maximum eigenvalue of F, one obtains an exact but exponential-time spectral algorithm for strongly refuting arbitrary k-XOR formulas.

Intuitions: To speed up the computation, we consider the order-ℓ Kikuchi matrix, which is a submatrix of F obtained by restricting to rows and columns indexed by sets of size ℓ. It is thus conceivable — especially when ℓ is sufficiently large — that the second desired property is satisfied.

More concretely, for 4-XOR, each hyperedge contributes Δ nonzero entries in the Kikuchi matrix, while only contributing 6 nonzero entries in the standard flattened matrix. One can interpret this as increasing the density and symmetry of the matrix, thereby "killing" large quadratic forms arising from vectors that are far from the product form $x^{\circ \ell}$. This is the perspective taken in [RRS17] to construct their matrix and to prove Theorem 25.1.

The reader is encouraged to consult [WEAM25] for various derivations and interpretations of the Kikuchi matrices (e.g., as a linearized message passing algorithm, or as the Hessian of an energy functional), as well as for the eigenbasis intuition described here.

Spectral Algorithms for Smooth Tradeoff

Once the right matrix is constructed, the analysis proceeds similarly to that in the previous chapter, as a direct consequence of matrix concentration inequalities. By (25.3),

$$\Delta \cdot \max_{x \in \{\pm 1\}^n} \Psi(x) = \max_{x \in \{\pm 1\}^n} \left(x^{\circ \ell}\right)^\top K \left(x^{\circ \ell}\right) \le \max_{x \in \{\pm 1\}^n} \|K\|_{\mathrm{op}} \cdot \left\|x^{\circ \ell}\right\|_2^2 = \binom{n}{\ell} \cdot \|K\|_{\mathrm{op}}. \tag{25.4}$$

We write the order-ℓ Kikuchi matrix $K = \sum_{e \in E(H)} b_e \cdot M_e$ as a random sum of matrices, where

$$M_e(S,T) = \begin{cases} 1 & \text{if } S \oplus T = e, \\ 0 & \text{otherwise.} \end{cases} \tag{25.5}$$

Then

$$\mathbb{E}\left[K^2\right] = \sum_{e \in E(H)} M_e^2 = \sum_{e \in E(H)} \sum_{S : |S \cap e| = k/2} \chi_S \chi_S^\top = D_K, \tag{25.6}$$

where $\chi_S \in \mathbb{R}^{\binom{n}{\ell}}$ is the characteristic vector of S with only one nonzero entry in the S-coordinate, and D_K is the diagonal degree matrix with $D_K(S,S) = |\{e \in E(H) \mid |S \cap e| = \frac{k}{2}\}|$ for $S \subseteq [n]$ with $|S| = \ell$. By the matrix Khintchine inequality in Theorem 23.16,

$$\mathbb{E}\left[\|K\|_{\mathrm{op}}\right] \lesssim \sqrt{\log \binom{n}{\ell}} \cdot \left\|\mathbb{E}\left[K^2\right]\right\|_{\mathrm{op}}^{\frac{1}{2}} = \sqrt{\ell \log n} \cdot \sqrt{\max_S D_K(S,S)}.$$

Since H is a random k-uniform hypergraph, by linearity of expectation,

$$\mathbb{E}\left[D_K(S,S)\right] = \sum_{e \in E(H)} \Pr\left(|e \cap S| = \frac{k}{2}\right) = m \cdot \frac{\binom{\ell}{k/2} \cdot \binom{n-\ell}{k/2}}{\binom{n}{k}} = \frac{m\Delta}{\binom{n}{\ell}}, \tag{25.7}$$

where the last equality is by a direct calculation using the definition of Δ in (25.2). Assuming $\max_S D_K(S,S) \lesssim m\Delta / \binom{n}{\ell}$, it follows from (25.4), (25.7), and (25.2) that

$$\mathbb{E}\left[\max_{x \in \{\pm 1\}^n} \Psi(x)\right] \le \frac{\binom{n}{\ell}}{\Delta} \cdot \mathbb{E}\left[\|K\|_{\mathrm{op}}\right] = \sqrt{\frac{m\ell\binom{n}{\ell}\log n}{\Delta}} \le \epsilon m,$$

whenever

$$m \ge \frac{\ell\binom{n}{\ell}\log n}{\Delta\epsilon^2} = \frac{\ell\binom{n}{\ell}\log n}{\binom{k}{k/2} \cdot \binom{n-k}{\ell-k/2} \cdot \epsilon^2} \gtrsim \frac{\ell n^k \log n}{\ell^{\frac{k}{2}} n^{\frac{k}{2}} \epsilon^2} = \frac{\ell^{1-\frac{k}{2}} \cdot n^{\frac{k}{2}} \log n}{\epsilon^2},$$

where the second inequality is by expanding the binomial coefficients and assuming $n \gg \ell \gg k$. Therefore, for $\delta \in (0,1)$, setting

$$\ell = n^{\delta} \quad \text{and} \quad m \gtrsim \epsilon^{-2} \cdot n^{\frac{k}{2}+\delta(1-\frac{k}{2})} \cdot \log n \quad \Longrightarrow \quad \mathbb{E}\left[\max_{x \in \{\pm 1\}^n} \Psi(x)\right] \leq \epsilon m.$$

Finally, it follows from standard concentration arguments that indeed $\max_S D_K(S,S) \lesssim \mathbb{E}\left[D_K(S,S)\right]$ with high probability, when $\mathbb{E}\left[D_K(S,S)\right] = m\Delta/\binom{n}{\ell} \gg \log\binom{n}{\ell}$ by our choice of m. One can then prove Theorem 25.1 formally by applying the matrix Bernstein inequality. To conclude, increasing the order ℓ interpolates smoothly between exponential-time exact algorithms and polynomial-time spectral algorithms.

Smooth Tradeoff for Semi-Random k-XOR

The reweighting trick in Section 24.3, with straightforward modifications, extends to the Kikuchi matrix. We leave the verification of the following extension of Theorem 25.1 to semi-random k-XOR for even k to Problem 25.8.

Theorem 25.3 (Smooth Tradeoff for Strong Refutation of Semi-Random k-XOR [HKM23])**.** *For any $\delta \in [0,1)$, there is an algorithm that strongly refutes a semi-random k-XOR formula with n variables and m constraints with high probability, when*

$$m \gtrsim n^{\frac{k}{2}+\delta(1-\frac{k}{2})} \cdot \log n, \qquad \textit{with time complexity } 2^{O(n^{\delta})}.$$

In particular, there is a spectral algorithm achieving this guarantee by computing the maximum eigenvalue of a $2^{O(n^{\delta})} \times 2^{O(n^{\delta})}$ reweighted Kikuchi matrix.

See Section 25.3 for a similar proof applying the reweighting trick to Kikuchi matrices to establish a hypergraph Moore bound.

25.2 Lower Bounds for Locally Decodable Codes

Interestingly, techniques developed for strongly refuting semi-random k-XOR can be used to prove lower bounds for locally decodable codes and locally correctable codes, leading to significant improvements on these long-standing open problems [AGKM23, KM24].

In this section, we only consider the simplest setting of proving lower bounds for locally decodable codes (with an even number of queries), to illustrate the connection to strong refutation of semi-random k-XOR formulas.

Locally Decodable Codes

An error correcting code is a function $f : \{0,1\}^k \to \{0,1\}^n$ that maps a k-bit message $b \in \{0,1\}^k$ to an n-bit codeword $f(b) \in \{0,1\}^n$, with the property that the receiver can recover b from a corrupted n-bit codeword y in which a small constant δ fraction of the bits of $f(b)$ are flipped. An error correcting code is q-locally decodable if, for any $i \in [k]$, the decoding algorithm can recover the bit b_i with probability $\frac{1}{2} + \epsilon$ by reading at most q bits of the corrupted codeword y.

The main question is to understand how small n can be in a q-query locally decodable code (q-LDC). In the case of $q = 2$, the Hadamard code is a 2-LDC with $n = 2^k$, and a lower bound of $n = 2^{\Omega(k)}$ is

known. In contrast, there is a wide gap in our understanding of higher-query LDCs. For $q = 3$, the best known constructions are based on matching vector codes with $n = 2^{k^{o(1)}}$, while the best known lower bound until recently was $n \gtrsim k^2/\log k$. See [AGKM23] for references to previous work.

Using spectral techniques for semi-random CSP refutations and the Kikuchi matrices, an improved lower bound $n \gtrsim k^3/\operatorname{polylog} k$ is proved for 3-LDCs [AGKM23]. Subsequently, these techniques were significantly extended to prove an exponential lower bound for 3-query locally correctable codes [KM24], where the requirement is to recover any bit of $f(b)$ with probability $\frac{1}{2} + \epsilon$ by reading at most q bits of the corrupted codeword y.

The goal of this section is to illustrate these techniques by recovering the known bounds for even q.

Theorem 25.4 (Lower Bounds for Locally Decodable Codes for Even q)**.** *For constants δ, ϵ and even $q \geq 4$, a q-locally decodable code $f : \{0,1\}^k \to \{0,1\}^n$ requires $n \geq k^{\frac{q}{q-2}}/\operatorname{polylog}(k)$.*

We remark that the spectral approach in this section gives a short proof of the exponential lower bound for 2-LDCs; see Problem 25.9.

Normal Form

General locally decodable codes can be quite complex, as the decoders may make adaptive queries and perform arbitrary computations. Fortunately, to prove lower bounds for locally decodable codes, we can reduce to the following simple normal form in which the queries are fixed and the decoder just performs additions.

Definition 25.5 (Normal Form [Yek12])**.** *A binary code $f : \{0,1\}^k \to \{0,1\}^n$ is (q, δ, ϵ)-normally decodable if, for each $i \in [k]$, there is a collection of δn disjoint q-tuples $e_{i,1}, \ldots, e_{i,\delta n} \subseteq [n]$ such that*

$$\Pr_{b \in \{0,1\}^k}\left[b_i = \sum_{j \in e_{i,\ell}} f(b)_j \mod 2\right] \geq \frac{1}{2} + \epsilon \quad \text{for every } \ell \in [\delta n],$$

where each $b \in \{0,1\}^k$ is chosen uniformly at random with probability $1/2^k$.

Note that normally decodable codes are weaker objects, as they only provide an average-case guarantee of correct decoding (and so it is potentially harder to prove their non-existence).

Yekhanin [Yek12] proved that to rule out a (q, δ, ϵ)-LDC encoding k-bit messages to n-bit codewords for constant $q \geq 2$ and constants $\delta, \epsilon \in (0,1)$, it suffices to rule out (q, δ', ϵ')-normally decodable codes encoding k-bit messages to $O(n)$-bit codewords for constants $\delta', \epsilon' \in (0,1)$.

Henceforth, we focus on proving the non-existence of $(q, \Omega(1), \Omega(1))$-normally decodable codes for constant even q, when $n \ll k^{\frac{q}{q-2}}/\operatorname{polylog}(k)$.

Connection to CSP Refutation

The normal form brings the q-LDC problem quite close to a collection of q-XOR problems. For each message $b \in \{0,1\}^k$, we interpret the search for the codeword $f(b) \in \{0,1\}^n$ as a q-XOR problem on n binary variables $x_1, \ldots, x_n \in \{0,1\}$, where each q-tuple $e_{i,\ell} \subseteq [n]$ for $i \in [k]$ and $\ell \in [\delta n]$ is a hyperedge of size q with value b_i. For each message $b \in \{0,1\}^k$, we denote this q-XOR instance by Ψ_b, with the intended solution being the codeword $f(b) \in \{0,1\}^n$ assigned to $x_1, \ldots, x_n$. So, there

are a total of 2^k instances of q-XOR, with the same set of variables and the same set of hyperedges, but the values on the hyperedges depend on b.

The formulation in Definition 25.5 does not require us to find a solution $f(b)$ for each $b \in \{0,1\}^k$ that satisfies all the XOR constraints in Ψ_b, but requires each XOR constraint to be satisfied by at least a $\frac{1}{2} + \epsilon$ fraction of solutions $\{f(b)\}_{b\in\{0,1\}^k}$. In particular, the existence of a q-normally decodable code implies that

$$\mathbb{E}_{b\in\{0,1\}^k}\big[\mathrm{opt}(\Psi_b)\big] \geq \Big(\frac{1}{2} + \epsilon\Big)m, \tag{25.8}$$

where $\mathrm{opt}(\Psi_b)$ denotes the maximum number of constraints in Ψ_b satisfied by an assignment. Thus, if we can "refute" (25.8) for a certain range of n, then we can prove that q-LDCs do not exist for that range of n. This is the goal of the remainder of this section.

Observe that the goal is almost the same as strongly refuting a semi-random q-XOR formula, but not exactly the same (as otherwise we would be done, and with an exponential lower bound!). The difference is that in a semi-random q-XOR instance, each hyperedge e has an independent random value b_e, while in the current setting all the hyperedges $e_{i,1}, \ldots, e_{i,\delta n}$ share the same random value b_i. Thus, this is a variant of semi-random q-XOR with limited randomness, which introduces some technical issues, as we will see, but the general spectral approach remains essentially the same.

Spectral Bound Using Kikuchi Matrices

In this subsection, we follow the same spectral approach as in previous sections, highlight the technical issues arising from limited randomness, and explain the fix in the next subsection.

For the spectral approach, we consider the equivalent setting where $b \in \{\pm 1\}^k$ and $x \in \{\pm 1\}^n$, so that each XOR constraint can be written in product form, and (25.8) can be expressed using the bias polynomial as in (24.1) as

$$\mathbb{E}_{b\in\{\pm1\}^k}\left[\max_{x\in\{\pm1\}^n} \Psi_b(x)\right] = \mathbb{E}_{b\in\{\pm1\}^k}\left[\max_{x\in\{\pm1\}^n} \sum_{i\in[k]} b_i \sum_{\ell\in[\delta n]} \prod_{j\in S_{i,\ell}} x_j\right] \geq 2\epsilon m. \tag{25.9}$$

We next express the bias polynomial as a quadratic form of a matrix. Here, we use the Kikuchi matrices, as they yield the tightest bounds, and in this application there is no concern about time complexity. Let $H_i = \{e_{i,1}, \ldots, e_{i,\delta n}\}$ be the hypergraph matching (since these hyperedges are disjoint) consisting of the hyperedges with value b_i. Note that we may assume that any two hypergraph matchings have no hyperedges in common, as this cannot occur in a normally decodable code in Definition 25.5. Let K_b be the order-r Kikuchi matrix where

$$K_b(S,T) = \begin{cases} b_i & \text{if } S \oplus T = e \text{ for some } e \in H_i, \\ 0 & \text{otherwise.} \end{cases}$$

Using the same calculations as in (25.3), (25.4) and the definition of Δ from (25.2),

$$\Delta \cdot \max_{x\in\{\pm1\}^n} \Psi_b(x) = \max_{x\in\{\pm1\}^n} \big(x^{\circ r}\big)^\top K_b\big(x^{\circ r}\big) \leq \max_{x\in\{\pm1\}^n} \|K_b\|_{\mathrm{op}} \cdot \big\|x^{\circ r}\big\|_2^2 = \binom{n}{r} \cdot \|K_b\|_{\mathrm{op}}. \tag{25.10}$$

To bound $\mathbb{E}_{b\in\{\pm1\}^k}\big[\|K_b\|_{\mathrm{op}}\big]$, we write the order-$r$ Kikuchi matrix $K_b = \sum_{i\in[k]} b_i \cdot M_i$ as a random sum of matrices, where

$$M_i(S,T) = \begin{cases} 1 & \text{if } S \oplus T = e \text{ for some } e \in H_i, \\ 0 & \text{otherwise.} \end{cases}$$

To apply the matrix Khintchine inequality, we need to compute $\mathbb{E}_{b\in\{\pm1\}^k}[K_b^2] = \sum_{i\in[k]} M_i^2$ and bound the operator norm of this matrix. Here is where the technical issue arises.

In the setting of (25.6), each hyperedge e has an independent random value b_e, and thus the matrix M_e in (25.5) has at most one nonzero entry in each row. Here, in this limited randomness setting, δn hyperedges are bundled in a matrix M_i. For a set $S \in \binom{[n]}{r}$ such that $|S \cap e| = \frac{q}{2}$ for many hyperedges $e \in H_i$, its row in the Kikuchi matrix contains many nonzero entries, and the operator norm of $\sum_{i\in[k]} M_i^2$ is indeed too large to yield any meaningful result for LDCs.

Truncated Kikuchi Matrix

The idea in [AGKM23] is to zero out some entries of K_b so that the resulting matrix $\tilde{K}_b$ can be written as $\sum_{i\in[k]} b_i \tilde{M}_i$ where each $\tilde{M}_i$ has at most one nonzero entry in each row. Then, by construction, $\tilde{M}_i^2$ is a diagonal matrix where each nonzero entry is one, and thus we have a good upper bound on the variance:

$$\left\|\mathbb{E}_{b\in\{\pm1\}^k}[\tilde{K}_b{}^2]\right\|_{\mathrm{op}} = \left\|\sum_{i\in[k]} \tilde{M}_i^2\right\|_{\mathrm{op}} \le k. \tag{25.11}$$

The other property that needs to be satisfied is that, for each hyperedge e, at most a constant fraction of the entries it contributes are zeroed out, so as to maintain the bound $\Delta \cdot \max_{x\in\{\pm1\}^n} \Psi_b(x) \asymp \max_{x\in\{\pm1\}^n} \left(x^{\circ r}\right)^\top \tilde{K}_b\left(x^{\circ r}\right)$ in (25.10) for the analysis to go through.

The formal definition is as follows. For each matching H_i, define the collection of good sets by

$$\mathcal{G}_i := \left\{ S \in \binom{[n]}{r} \;\middle|\; |S\cap e| = \frac{q}{2} \text{ for only one hyperedge in } H_i \right\}.$$

Let $\tilde{K}_b$ be the order-r truncated Kikuchi matrix where

$$\tilde{K}_b(S,T) = \begin{cases} b_i & \text{if } S \oplus T = e \text{ for some } e \in H_i \text{ and } S,T \in \mathcal{G}_i, \\ 0 & \text{otherwise.} \end{cases}$$

A technical counting argument in [AGKM23] shows that, for each hyperedge e, at most half of the entries it contributes are zeroed out, as long as

$$r \lesssim n^{1-\frac{2}{q}}.$$

We further zero out additional entries so that each hyperedge e contributes exactly $\frac{\Delta}{2}$ entries to $\tilde{K}_b$.

Lower Bounds for Locally Decodable Codes

Using this truncated order-r Kikuchi matrix with $r \asymp n^{1-\frac{2}{q}}$, we proceed to finish the analysis:

$$\frac{\Delta}{2} \cdot \max_{x\in\{\pm1\}^n} \Psi_b(x) = \max_{x\in\{\pm1\}^n} \left(x^{\circ r}\right)^\top \tilde{K}_b\left(x^{\circ r}\right) \le \max_{x\in\{\pm1\}^n} \|\tilde{K}_b\|_{\mathrm{op}} \cdot \left\|x^{\circ r}\right\|_2^2 = \binom{n}{r} \cdot \|\tilde{K}_b\|_{\mathrm{op}}.$$

Then we write $\tilde{K}_b = \sum_{i\in[k]} b_i \cdot \tilde{M}_i$ as a random sum, where $\tilde{M}_i$ is the corresponding truncation of the matrix M_i above. By construction, each $\tilde{M}_i$ has at most one nonzero entry in each row. Thus, by the matrix Khintchine inequality in Theorem 23.16 and (25.11),

$$\mathbb{E}_{b\in\{\pm1\}^k}\left[\left\|\tilde{K}_b\right\|_{\mathrm{op}}\right] \le \sqrt{\log\binom{n}{r}} \cdot \left\|\mathbb{E}_{b\in\{\pm1\}^k}[\tilde{K}_b{}^2]\right\|_{\mathrm{op}}^{\frac{1}{2}} \le \sqrt{kr\log n}.$$

It follows that

$$\mathbb{E}_{b\in\{\pm1\}^k}\left[\max_{x\in\{\pm1\}^n}\Psi_b(x)\right] \lesssim \frac{1}{\Delta}\binom{n}{r}\sqrt{kr\log n} \lesssim \left(\frac{n}{r}\right)^{\frac{q}{2}}\sqrt{kr\log n} \asymp n\sqrt{kn^{1-\frac{2}{q}}\log n},$$

where the second inequality follows from the definition of Δ in (25.2) and that q is a constant, and the last equality follows by substituting $r \asymp n^{1-\frac{2}{q}}$. Combining with the guarantee for normally decodable codes in (25.9) and using $m = k\delta n$,

$$2\epsilon k\delta n = 2\epsilon m \le \mathbb{E}_{b\in\{\pm1\}^k}\left[\max_{x\in\{\pm1\}^n}\Psi_b(x)\right] \lesssim n\sqrt{kn^{1-\frac{2}{q}}\log n}.$$

Rearranging and using that δ, ϵ are constants yields $n \ge k^{\frac{q}{q-2}}/\operatorname{polylog}(k)$, proving Theorem 25.4.

We remark that the restriction $r \asymp n^{1-\frac{2}{q}}$ (required to ensure the properties of the truncated Kikuchi matrix) is the bottleneck of the current approach.

25.3 Hypergraph Moore Bound

In this section, we show that the spectral approach using the Kikuchi matrix also has interesting applications to combinatorial problems.

The Moore bound for graphs [AHL02] states that a graph of average degree d has a cycle of length at most $2\log_{d-1}(n)$. Motivated by the problem of refuting random 3-SAT formulas, Feige [Fei08] conjectured the following generalization of the Moore bound to hypergraphs.

Conjecture 25.6 (Feige Conjecture [Fei08])**.** *Every k-uniform hypergraph $H = ([n], E)$ with average degree $(n/\ell)^{\frac{k}{2}-1}$ has a cycle of length $O(\ell\log n)$. Here, a cycle in a hypergraph is a subset of hyperedges $C \subseteq E$ such that every vertex appears in an even number of hyperedges in C.*

This conjecture was wide open until Guruswami, Kothari, and Manohar [GKM22] resolved it up to polylogarithmic factor using the Kikuchi matrix method. Our presentation follows the simplified and improved proofs in [HKM23, GKM25] and Kothari's course notes on matrix concentration.

Counting Walks

We highlight the main idea in this subsection, and present the formal proof in the next subsection.

Hypergraph Cycles: We consider the following formulation that brings the hypergraph cycle problem closer to the problems that we studied in the previous sections. Let $C = \{e_1, \ldots, e_t\}$ be a hypergraph cycle. Since every vertex appears in an even number of hyperedges in C, it follows that

$$e_1 \oplus e_2 \oplus \ldots \oplus e_t = \emptyset. \tag{25.12}$$

Note that each hyperedge e_i appears at most once in a hypergraph cycle.

Closed Walks in Kikuchi Graphs: Next, we observe that a closed walk in the "Kikuchi graph" also defines a collection of hyperedges satisfying the same relation in (25.12). Given a k-uniform hypergraph $H = ([n], E)$ for an even integer k, let A_ℓ be the $\binom{n}{\ell} \times \binom{n}{\ell}$ adjacency matrix of the order-ℓ Kikuchi matrix K_ℓ of H as defined in (25.1), where each nonzero entry of K_ℓ is set to 1 (i.e., $b_e = 1$ for every hyperedge $e \in E(H)$). Then the order-ℓ Kikuchi graph G_ℓ of H is the graph with vertex set $\binom{[n]}{\ell}$ and adjacency matrix A_ℓ. In particular, two vertices $S, T \in \binom{[n]}{\ell}$ are adjacent in G_ℓ if and only if $S \oplus T = e$ for some $e \in E(H)$.

Suppose $(S_1, S_2, \ldots, S_t, S_{t+1} = S_1)$ is a closed walk in G_ℓ. Since (S_i, S_{i+1}) is an edge in the Kikuchi graph, there exists a unique hyperedge $e_i \in E(H)$ such that $S_i \oplus S_{i+1} = e_i$ for $1 \le i \le t$. Note that

$$e_1 \oplus e_2 \oplus \ldots \oplus e_t = (S_1 \oplus S_2) \oplus (S_2 \oplus S_3) \oplus \ldots \oplus (S_t \oplus S_1) = \emptyset. \tag{25.13}$$

Thus, each closed walk defines a collection of hyperedges satisfying (25.13), but the collection may not contain a hypergraph cycle. For example, the walk $(S_1, S_2, S_3, S_4, S_3, S_2, S_1)$ defines the collection $(e_1, e_2, e_3, e_3, e_2, e_1)$, which does not contain a hypergraph cycle if $e_1 \oplus e_2 \oplus e_3 \neq \emptyset$.

Even Closed Walks and Non-Even Closed Walks: Call a closed walk an even closed walk if the corresponding collection of hyperedges contains each hyperedge an even number of times; otherwise, it is called a non-even closed walk.

Note that, for a non-even closed walk, the corresponding collection of hyperedges must contain a hypergraph cycle, as the subset of hyperedges that appear an odd number of times must satisfy (25.12).

Proof by Counting Walks: Suppose the hypergraph H has no hypergraph cycles with at most t hyperedges. Then, by the discussion above, the Kikuchi graph G_ℓ does not have a non-even closed walk with at most t steps. Therefore, the number of closed walks of length t in G_ℓ is equal to the number of even closed walks of length t.

On one hand, we know that the number of closed walks of length t in G_ℓ is equal to $\mathrm{Tr}(A_\ell^t)$, using the simple combinatorial argument in Lemma 1.5. On the other hand, we observe that the number of even closed walks of length t in G_ℓ is equal to $\mathbb{E}_b[\mathrm{Tr}(K_\ell^t)]$, where K_ℓ is as defined in (25.1) with an independent random variable $b_e \in \{\pm 1\}$ with equal probability for each hyperedge $e \in E(H)$. To see this, note that each term in $\mathbb{E}_b[\mathrm{Tr}(K_\ell^t)]$ is of the form $\mathbb{E}_b[b_{e_1}^{c_1} \cdot b_{e_2}^{c_2} \cdot \ldots \cdot b_{e_r}^{c_r}]$ for some r such that $c_1 + \ldots + c_r = t$. For a non-even closed walk, there is an odd c_i, and so the corresponding term contributes zero as $\mathbb{E}\left[b_e^{c_i}\right] = 0$ for odd c_i, while each even closed walk contributes one as $\mathbb{E}\left[b_e^{c_j}\right] = 1$ for even c_j.

Therefore, when H has no hypergraph cycles with at most t hyperedges, it holds that

$$\mathrm{Tr}\left(A_\ell^t\right) = \mathbb{E}_b[\mathrm{Tr}\left(K_\ell^t\right)]. \tag{25.14}$$

This is the main conceptual step of the proof. We sketch the remaining steps below and explain them formally in the next subsection. The proof proceeds by setting $t \asymp \ell \log n$ and giving a lower bound on the LHS and an upper bound on the RHS, yielding

$$d_{\mathrm{avg}} \le \mathrm{Tr}(A_\ell^t)^{\frac{1}{t}} = \left(\mathbb{E}_b[\mathrm{Tr}\left(K_\ell^t\right)]\right)^{\frac{1}{t}} \lesssim \sqrt{\ell \log n} \cdot \sqrt{d_{\max}},$$

where the lower bound follows from a simple argument and the upper bound follows from the non-commutative Khintchine inequality in Theorem 23.15 and $t \asymp \ell \log n$. Suppose $d_{\max} \asymp d_{\mathrm{avg}}$. Then

this implies that $d_{\mathrm{avg}} \lesssim \ell \log n$, which yields the desired bound

$$m \le n \cdot \left(\frac{n}{\ell}\right)^{\frac{k}{2}-1} \log n$$

up to a logarithmic factor, after some straightforward calculation.

Formal Proof via Reweighting

In the proof overview above, we assumed that $d_{\mathrm{avg}} \asymp d_{\max}$, which does not necessarily hold for the Kikuchi graph. To address this issue, we use the reweighting trick as in Section 24.3. Let

$$\Gamma = D_K + \overline{d} \cdot I_{\binom{n}{\ell}},$$

where D_K is the diagonal degree matrix defined in Section 25.1, with

$$D_K(S,S) = \left|\left\{e \in E(H) \;\middle|\; |S \cap e| = \frac{k}{2}\right\}\right| \quad \text{and} \quad \overline{d} = \frac{2m\Delta}{\binom{n}{\ell}}$$

being respectively the degree of S and the average degree in the Kikuchi graph G_ℓ of H, where $\Delta = \binom{k}{k/2} \cdot \binom{n-k}{\ell-k/2}$ as defined in (25.2).

We count weighted walks using the reweighted adjacency matrix $\Gamma^{-\frac{1}{2}} A_\ell \Gamma^{-\frac{1}{2}}$ and the reweighted Kikuchi matrix $\Gamma^{-\frac{1}{2}} K_\ell \Gamma^{-\frac{1}{2}}$. Since each walk in G_ℓ still has positive weight after reweighting, using the same arguments in establishing (25.14), it still holds that, when H has no hypergraph cycles with at most t hyperedges,

$$\operatorname{Tr}\left(\left(\Gamma^{-\frac{1}{2}} A_\ell \Gamma^{-\frac{1}{2}}\right)^t\right) = \mathbb{E}_b\left[\operatorname{Tr}\left(\left(\Gamma^{-\frac{1}{2}} K_\ell \Gamma^{-\frac{1}{2}}\right)^t\right)\right]. \tag{25.15}$$

To lower bound the LHS in (25.15), note that

$$\operatorname{Tr}\left(\left(\Gamma^{-\frac{1}{2}} A_\ell \Gamma^{-\frac{1}{2}}\right)^t\right)^{\frac{1}{t}} \ge \left\|\Gamma^{-\frac{1}{2}} A_\ell \Gamma^{-\frac{1}{2}}\right\|_{\mathrm{op}} \ge \frac{\left(\Gamma^{\frac{1}{2}}1\right)^\top \left(\Gamma^{-\frac{1}{2}} A_\ell \Gamma^{-\frac{1}{2}}\right)\left(\Gamma^{\frac{1}{2}}1\right)}{\left(\Gamma^{\frac{1}{2}}1\right)^\top \left(\Gamma^{\frac{1}{2}}1\right)} = \frac{1^\top A_\ell 1}{\operatorname{Tr}(\Gamma)} = \frac{1}{2}, \tag{25.16}$$

where the first inequality follows from the fact that the ℓ_t-norm of eigenvalues dominates the operator norm, the second inequality follows from the optimization formulation for eigenvalues in Lemma A.12 with the test vector $\Gamma^{\frac{1}{2}}1$, and the last equality follows from $1^\top A_\ell 1 = 2m\Delta$ and $\operatorname{Tr}(\Gamma) = 4m\Delta$.

To upper bound the RHS in (25.15), we apply the non-commutative Khintchine inequality in Theorem 23.15 to the random matrix $K_\ell = \sum_{e \in E(H)} b_e M_e$, where M_e is as defined in (25.5), and set $t \asymp \log \binom{n}{\ell} \asymp \ell \log n$ to obtain

$$\begin{aligned}
\mathbb{E}_b\left[\operatorname{Tr}\left(\left(\Gamma^{-\frac{1}{2}} K_\ell \Gamma^{-\frac{1}{2}}\right)^t\right)\right]^{\frac{1}{t}} &\lesssim \sqrt{t-1} \cdot \operatorname{Tr}\left[\left(\sum_{e \in E(H)} \left(\Gamma^{-\frac{1}{2}} M_e \Gamma^{-\frac{1}{2}}\right)^2\right)^{\frac{t}{2}}\right]^{\frac{1}{t}} \\
&\asymp \sqrt{t-1} \cdot \left\|\sum_{e \in E(H)} \left(\Gamma^{-\frac{1}{2}} M_e \Gamma^{-\frac{1}{2}}\right)^2\right\|_{\mathrm{op}}^{\frac{1}{2}} \\
&\le \sqrt{\frac{\ell \log n}{\overline{d}}},
\end{aligned} \tag{25.17}$$

where the second line follows from the moment method in (23.6) when $t \asymp \log \binom{n}{\ell}$, and the last inequality follows from the same calculation in (24.10).

Combining (25.15), (25.16), and (25.17), we conclude that when the hypergraph H does not have a hypergraph cycle with at most $t \asymp \ell \log n$ hyperedges, then $\overline{d} \lesssim \ell \log n$, which implies that

$$m \lesssim \frac{\ell \binom{n}{\ell} \log n}{\Delta} \lesssim \ell \left(\frac{n}{\ell}\right)^{\frac{k}{2}} \log n = n \left(\frac{n}{\ell}\right)^{\frac{k}{2}-1} \log n.$$

This proves Feige's conjecture up to a logarithmic factor for even k.

We remark that the proof in [HKM23] is based on counting walks directly using a combinatorial argument, while here we use the non-commutative Khintchine inequality for convenience.

In Problem 25.10 and Problem 25.11, we outline an alternative approach for proving the existence of a hypergraph cycle when the hypergraph is sufficiently dense, using the concept of "rainbow cycles".

Further References

As usual, the same result for odd k is significantly harder; we refer the reader to [HKM23].

See [GKM22] for applications of the hypergraph Moore bound in refuting semi-random CSPs.

It remains an open problem to remove the logarithmic factor. See [HKM+25] for improvements in the case when k is odd.

25.4 Tensor Principal Component Analysis

Removing noise from data is a fundamental task across many areas. Since the spectrum of random matrices is well understood, spectral methods are highly effective in separating noise from signal.

In this section, we first introduce the planted random model and review known results for two fundamental problems. We then introduce the tensor PCA problem, where the Kikuchi matrix can be used to design significantly more efficient spectral algorithms.

Planted Random Model

A well-studied setting in theoretical computer science is the planted random model (also known as the stochastic block model, or semi-random graph problems).

Planted Clique: A classical problem in this setting is the planted clique problem. In this problem, we first generate a random graph $G(n, \frac{1}{2})$, where there are n vertices and each pair of vertices is connected by an edge independently with probability $\frac{1}{2}$. Then we choose a random subset S of k vertices and add all edges between vertices in S. The clique can be viewed as the signal, and the random graph as noise. The algorithmic task is to recover this hidden clique S.

It is well known that the largest clique in $G(n, \frac{1}{2})$ is $(2 + o(1)) \cdot \log_2 n$ with high probability. Thus, exact recovery of the planted clique is information-theoretically possible whenever $k \gg \log n$. However, there remains a significant gap between what is information-theoretically possible and what can be achieved algorithmically.

The first nontrivial algorithm [AKS98] for the planted clique problem is based on the spectral method. The main observation is that the second eigenvalue of $G(n, \frac{1}{2})$ is at most $(2 + o(1)) \cdot \sqrt{n}$ with high probability by classical results in random matrix theory, whereas if $k \gg \sqrt{n}$, the second eigenvalue becomes $\Theta(k)$ by a simple spectral argument using the centered characteristic vector of the clique, which is orthogonal to the top eigenvector of the graph. For example, when $k \geq 10\sqrt{n}$, it can be shown that the k vertices corresponding to the largest absolute entries of the second eigenvector have a large overlap with the hidden clique, allowing exact recovery.

Since then, many different algorithms have been proposed to recover the planted clique, all of which succeed only when $k \gg \sqrt{n}$. It remains a long-standing open problem to recover a planted clique of size $o(\sqrt{n})$. There is now strong evidence that $\Theta(\sqrt{n})$ is optimal for polynomial-time algorithms, with a nearly matching sum-of-squares lower bound of $n^{\frac{1}{2}-o(1)}$ [BHK+19].

Planted Bisection: Another extensively studied problem is the planted bisection problem. In this problem, there is a hidden bisection $(S, V \setminus S)$ of the vertex set. A random graph is generated by connecting each pair of vertices within S and within $V \setminus S$ independently with probability p, while each pair with one vertex in S and one vertex in $V \setminus S$ is connected independently with probability q. The algorithmic task is to recover the hidden bisection $(S, V \setminus S)$.

MsSherry [MsS01] proposed a general spectral framework, based on matrix perturbation theory, for recovering the hidden partition in various models, including the planted clique model. In particular, if $p = \frac{a \log n}{n}$ and $q = \frac{b \log n}{n}$ with $a - b \geq \Omega(1)$, then a spectral algorithm exactly recovers the hidden bisection. We refer the reader to [Spi26, Ver26] for expositions of this matrix perturbation approach.

Indeed, a sharp threshold is known for exact recovery of the hidden bisection [ABH15, MNS15]. In the regime where $p = \frac{a \log n}{n}$ and $q = \frac{b \log n}{n}$, exact recovery of the hidden bisection is efficiently solvable if $|\sqrt{a} - \sqrt{b}| > \sqrt{2}$ and information-theoretically impossible if $|\sqrt{a} - \sqrt{b}| < \sqrt{2}$. Remarkably, it was shown in [AFWZ20] that the simple spectral algorithm that computes the second largest eigenvector x of the adjacency matrix and outputs the set $S := \{i \in [n] \mid x(i) \geq 0\}$ achieves exact recovery whenever $|\sqrt{a} - \sqrt{b}| > \sqrt{2}$. We refer the reader to the survey [Abb18] for detailed expositions of these results.

Tensor PCA

This subsection follows closely the presentation in [WEAM25], where references can be found.

In the order-k tensor PCA (or spiked tensor) problem, we observe a k-fold $n \times n \times \cdots \times n$ tensor

$$Y = \lambda \cdot x_*^{\otimes k} + G, \tag{25.18}$$

where $\lambda \geq 0$ is the signal-to-noise ratio, $x_* \in \{\pm 1\}^n$ is a planted signal vector, and G is a symmetric noise tensor with $N(0,1)$ entries (i.e., $G_{i_1,\ldots,i_k} = G_{i_{\pi(1)},\ldots,i_{\pi(k)}}$ for any permutation π of $[k]$).

The task of *strong detection* is to determine, with probability tending to 1, whether the observed tensor is sampled from the distribution in (25.18) for a given $\lambda > 0$ or from the case $\lambda = 0$. The task of *strong recovery* is to return an estimator $\hat{x} \in \{\pm 1\}^n$ such that, with high probability,

$$\mathrm{corr}(\hat{x}, x_*) = \frac{|\langle \hat{x}, x_* \rangle|}{\|\hat{x}\| \|x_*\|} = 1 - o(1),$$

while the task of *weak recovery* is to return an estimator $\hat{x} \in \{\pm 1\}^n$ such that $\mathrm{corr}(\hat{x}, x_*) = \Omega(1)$ with high probability.

The matrix case: The spiked tensor model reduces to the spiked Wigner model when $k = 2$. It is known that strong detection and weak recovery are possible when $\lambda > 1/\sqrt{n}$ by computing the largest eigenvalue and using the corresponding eigenvector. On the other hand, strong detection and weak recovery are statistically impossible when $\lambda < 1/\sqrt{n}$, so there is a sharp phase transition at $\lambda = 1/\sqrt{n}$.

The tensor case: Information-theoretically, there is a sharp phase transition similar to the matrix case: there exists a constant c depending on k such that if $\lambda > c\, n^{(1-k)/2}$ then strong detection and weak recovery are possible, while they are impossible when $\lambda < c\, n^{(1-k)/2}$.

Computationally, there are polynomial-time algorithms (e.g., sum-of-squares algorithms) that succeed at both strong detection and strong recovery when $\lambda \gg n^{-k/4}$, and there are sum-of-squares lower bounds suggesting that no polynomial-time algorithms can succeed at strong detection or weak recovery when $\lambda \ll n^{-k/4}$.

Thus, unlike the matrix case, there is a large statistical–computational gap, with a "possible-but-hard" regime when $n^{(1-k)/2} \ll \lambda \ll n^{-k/4}$. Interestingly, as in the problem of refuting random k-XOR in Section 25.1, the sum-of-squares framework exhibits a smooth tradeoff between time complexity and the signal-to-noise ratio in this regime.

Theorem 25.7 (Smooth Tradeoff for Tensor PCA [BGL17, HSS15])**.** *For any $1 \le \ell \le n$, there is an algorithm with running time $n^{O(\ell)}$ that achieves strong detection and strong recovery in the order-k tensor PCA problem whenever $\lambda \gg \ell^{\frac{1}{2}-\frac{k}{4}}\, n^{-\frac{k}{4}}\, \mathrm{polylog}(n)$.*

When $\ell = n^{\delta}$, this interpolates smoothly between the polynomial-time threshold $\lambda \asymp n^{-\frac{k}{4}}$ when $\delta = 0$ and the information-theoretic threshold $\lambda \asymp n^{(1-k)/2}$ when $\delta = 1$. This tradeoff is conjectured to be optimal, with a matching sum-of-squares lower bound.

Strong Detection

Prior to the work of Wein, El Alaoui, and Moore [WEAM25], there was a large gap between the power of the sum-of-squares approach and that of the statistical physics approach (e.g., approximate message passing and related spectral methods) for the tensor PCA problem. Using the concept of "Kikuchi free energies", they derived the Kikuchi matrix and used it to design a simple spectral algorithm that matches the guarantee in Theorem 25.7, thereby redeeming the statistical physics approach. Furthermore, this spectral approach is considerably easier to analyze, as the proof reduces to a straightforward application of the matrix Chernoff bound for Gaussian series in Theorem 23.10.

Algorithm: Let k be an even number. For a symmetric k-fold tensor Y and a subset $e = \{i_1, \dots, i_k\} \subseteq [n]$ of size k, denote $Y(e) := Y_{i_1,\dots,i_k}$. For $\frac{k}{2} \le \ell \le n$, let K_ℓ be the order-ℓ $\binom{n}{\ell} \times \binom{n}{\ell}$ Kikuchi matrix as in Definition 25.2, where

$$K_\ell(S,T) = \begin{cases} Y(S \oplus T) & \text{if } |S \oplus T| = k, \\ 0 & \text{otherwise.} \end{cases} \tag{25.19}$$

The detection algorithm is simply to compute the maximum eigenvalue of the Kikuchi matrix.

Algorithm 24 Strong Detection Algorithm for Tensor PCA

Require: Let $Y = \lambda \cdot x_*^{\otimes k} + G$ as described in (25.18).

1: Compute the maximum eigenvalue $\lambda_{\max}(K_\ell)$ of the Kikuchi matrix K_ℓ of Y defined in (25.19).

2: Return "no signal" if and only if

$$\lambda_{\max}(K_\ell) \le \frac{\lambda\, d_\ell}{2}, \quad \text{where } d_\ell = \binom{n-\ell}{k/2}\binom{\ell}{k/2}.$$

Proof by Matrix Concentration: Write the Kikuchi matrix as $K_\ell := \lambda X_* + Z$, where X_* is the signal part with

$$X_*(S,T) = \begin{cases} \prod_{i \in S \oplus T} x_*(i) & \text{if } |S \oplus T| = k, \\ 0 & \text{otherwise,} \end{cases}$$

and $Z := \sum_{e \subseteq [n]:|e|=k} g_e \cdot M_e$ is the noise part, which is a random sum of matrices where $g_e \sim N(0,1)$ and M_e is as defined in (25.5) with

$$M_e(S,T) = \begin{cases} 1 & \text{if } S \oplus T = e, \\ 0 & \text{otherwise.} \end{cases}$$

The no-signal case: Suppose there is no signal, i.e., $\lambda = 0$. Then $K_\ell = Z$. As in (25.6),

$$\mathbb{E}\left[Z^2\right] = \sum_{e \subseteq [n]\,:\,|e|=k} M_e^2 = \sum_{e \subseteq [n]\,:\,|e|=k} \; \sum_{S \subseteq [n], |S|=\ell\,:\,|S \cap e| = k/2} \chi_S \chi_S^\top = d_\ell \cdot I_{\binom{n}{\ell}},$$

where $\chi_S \in \mathbb{R}^{\binom{n}{\ell}}$ is the characteristic vector of S, with a 1 in the S-coordinate and 0 otherwise. The last equality holds since, for each $S \subseteq [n]$ of size ℓ, there are exactly d_ℓ sets $e \subseteq [n]$ of size k such that $|S \cap e| = k/2$. By the matrix Chernoff bound for Gaussian series in Theorem 23.10,

$$\Pr\left(\lambda_{\max}(K_\ell) \ge \frac{\lambda d_\ell}{2}\right) \le \Pr\left(\|Z\|_{\mathrm{op}} \ge \frac{\lambda d_\ell}{2}\right) \le 2\binom{n}{\ell} \cdot \exp\left(-\frac{(\lambda d_\ell/2)^2}{2d_\ell}\right) \le 2n^\ell \exp\left(-\frac{\lambda^2 d_\ell}{8}\right).$$

The signal case: Suppose there is a signal, i.e., $\lambda > 0$. Note that $\|X_*\|_{\mathrm{op}} = d_\ell$ with corresponding eigenvector $x_*^{\circ \ell} \in \mathbb{R}^{\binom{n}{\ell}}$, where $x_*^{\circ \ell}(S) = \prod_{i \in S} x_*(i)$. Then

$$\lambda_{\max}(K_\ell) \ge \lambda \|X_*\|_{\mathrm{op}} - \|Z\|_{\mathrm{op}} = \lambda \cdot d_\ell - \|Z\|_{\mathrm{op}}.$$

It follows from the same bound that

$$\Pr\left(\lambda_{\max}(K_\ell) \le \frac{\lambda d_\ell}{2}\right) \le \Pr\left(\|Z\|_{\mathrm{op}} \ge \frac{\lambda d_\ell}{2}\right) \le 2n^\ell \exp\left(-\frac{\lambda^2 d_\ell}{8}\right).$$

Conclusion: Note that $d_\ell \asymp n^{\frac{k}{2}} \ell^{\frac{k}{2}}$ when $\ell = o(n)$. In this regime, whenever

$$\lambda \gtrsim n^{-\frac{k}{4}}\, \ell^{-\frac{k-2}{4}} \sqrt{\log n}\, \sqrt{\log(1/\epsilon)}, \tag{25.20}$$

the strong detection algorithm succeeds with high probability, as the error probability satisfies

$$\Pr\left(\lambda_{\max}(K_\ell) \ge \frac{\lambda d_\ell}{2} \;\middle|\; \text{no signal}\right) + \Pr\left(\lambda_{\max}(K_\ell) \le \frac{\lambda d_\ell}{2} \;\middle|\; \text{signal}\right) \lesssim \epsilon.$$

This yields a simple spectral algorithm matching the guarantee in Theorem 25.7.

Removing the Logarithmic Factor: It was conjectured in [WEAM25] that the $\sqrt{\log n}$ factor in (25.20) can be removed. This conjecture was resolved by Kothari and Xu [KX26], who used a careful combinatorial trace method to obtain a tight bound on the operator norm of the Kikuchi matrix.

Strong Recovery

The strong recovery algorithm is more subtle, as the leading eigenvector of K_ℓ does not correlate well with that of X_*. Nevertheless, [WEAM25] designed a "rounding" algorithm that transforms the leading eigenvector $y \in \mathbb{R}^{\binom{n}{\ell}}$ of K_ℓ into a vector $x \in \mathbb{R}^n$ that correlates well with the signal vector $x_* \in \{\pm 1\}^n$.

Their idea is to use a *voting matrix*. Given a vector $y \in \mathbb{R}^{\binom{n}{\ell}}$, associate a symmetric $n \times n$ voting matrix $V(y)$ with entries

$$V_{ii}(y) = 0 \quad \forall i \in [n], \quad \text{and} \quad V_{ij}(y) = \frac{1}{2} \sum_{S,T \in \binom{[n]}{\ell} : S \oplus T = \{i,j\}} y(S) \cdot y(T) \quad \forall i \neq j.$$

Algorithm 25 Strong Recovery Algorithm for Tensor PCA

Require: Let $Y = \lambda \cdot x_*^{\otimes k} + G$ as described in (25.18).

1: Compute a leading eigenvector $y \in \mathbb{R}^{\binom{n}{\ell}}$ of the Kikuchi matrix K_ℓ of Y defined in (25.19).
2: Form the associated voting matrix $V(y)$.
3: Compute a leading eigenvector x of $V(y)$ and output x.

The proof uses Fourier analysis on a slice of the hypercube (i.e., on subsets of size ℓ) and is somewhat beyond the scope of this book. We refer the reader to [WEAM25] for an elegant proof.

25.5 Problems

Problem 25.8 (Smooth Tradeoff for Strongly Refuting Semi-Random k-XOR)**.** *Prove Theorem 25.3 using the reweighting trick in Section 24.3.*

Hint: Use $\Gamma = D_K + \overline{d} \cdot I_{\binom{n}{\ell}}$, where D_K is the diagonal matrix in Section 25.1 and $\overline{d} = 2m\Delta/\binom{n}{\ell}$.

Problem 25.9 (Exponential Lower Bounds for 2-LDCs [Gop18])**.** *Use the spectral approach in Section 25.2 to prove that a 2-locally decodable code $f : \{0,1\}^k \to \{0,1\}^n$ requires $n \geq 2^{\Omega(k)}$.*

Hint: Use the basic spectral approach for strongly refuting 2-XOR in Section 24.1.

Problem 25.10 (Hypergraph Cycles via Rainbow Cycles)**.** *Let $G = (V, E)$ be a properly edge-colored graph, where each edge has a color and the edges of each color form a matching. A cycle is called a rainbow cycle if its edges have distinct colors.*

It is proved in [ABS+25] that a rainbow cycle exists in a properly edge-colored graph whenever $|E| \gtrsim |V| \log |V| \log \log |V|$.

Apply this theorem to the Kikuchi graph to obtain an alternative proof of Feige's conjecture up to logarithmic factors.

Problem 25.11 (Finding Rainbow Cycles Using Random Walks)**.** *In this problem, we outline a simple proof of the existence of a rainbow cycle in a properly edge-colored graph $G = (V, E)$ whenever $|E| \gtrsim |V| \operatorname{polylog} |V|$.*

1. *Suppose G has mixing time $O(\log |V|)$ and minimum degree $\Omega(\log^2 |V|)$. Using random walks, prove that for any vertex v, there is a rainbow path of length $O(\log |V|)$ from v to more than half of the vertices.*

2. *Let G be a spectral expander (i.e., with a constant spectral gap). Using matrix concentration, prove that its edge set can be split into two disjoint subgraphs H_1, H_2 with disjoint color sets such that each H_i is a spectral expander.*

3. *Use (1) and (2) to prove that if G is a spectral expander, then there exists a rainbow cycle of length $O(\log |V|)$.*

4. *Extend these results to general graphs (with worse bounds) using expander decompositions in Chapter 12.*

Problem 25.12 (Weak Hypergraph Moore Bound for Odd k)**.** *Let $H = ([n], E)$ be a k-uniform hypergraph with odd k. Let $B = (\binom{[n]}{\ell}, \binom{[n]}{\ell+1}; F)$ be the bipartite Kikuchi graph, where $S \in \binom{[n]}{\ell}$ and $T \in \binom{[n]}{\ell+1}$ are adjacent if and only if $S \oplus T = e$ for some $e \in E(H)$. Use the approach in Section 25.3 applied to this bipartite Kikuchi graph to prove that H has a hypergraph cycle of length $O(\ell \log n)$ whenever*

$$|E| \gtrsim n\Big(\frac{n}{\ell}\Big)^{\lceil \frac{k}{2} \rceil - 1} \log n.$$

References

[Abb18] E Abbe. Community detection and stochastic block models. *Foundations and Trends in Communications and Information Theory. now Publishers Inc*, 14(1-2), 2018. 319

[ABH15] Emmanuel Abbe, Afonso S Bandeira, and Georgina Hall. Exact recovery in the stochastic block model. *IEEE Transactions on information theory*, 62(1):471–487, 2015. 319

[ABS+25] Noga Alon, Matija Bucić, Lisa Sauermann, Dmitrii Zakharov, and Or Zamir. Essentially tight bounds for rainbow cycles in proper edge-colourings. *Proceedings of the London Mathematical Society*, 130(4):e70044, 2025. 322

[AFWZ20] Emmanuel Abbe, Jianqing Fan, Kaizheng Wang, and Yiqiao Zhong. Entrywise eigenvector analysis of random matrices with low expected rank. *Annals of statistics*, 48(3):1452, 2020. 319

[AGKM23] Omar Alrabiah, Venkatesan Guruswami, Pravesh K Kothari, and Peter Manohar. A near-cubic lower bound for 3-query locally decodable codes from semirandom csp refutation. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, pages 1438–1448, 2023. 9, 311, 312, 314

[AHL02] Noga Alon, Shlomo Hoory, and Nathan Linial. The moore bound for irregular graphs. *Graphs and Combinatorics*, 18(1):53–57, 2002. 315

[AKS98] Noga Alon, Michael Krivelevich, and Benny Sudakov. Finding a large hidden clique in a random graph. *Random Structures & Algorithms*, 13(3-4):457–466, 1998. 319

[BGL17] Vijay Bhattiprolu, Venkatesan Guruswami, and Euiwoong Lee. Sum-of-squares certificates for maxima of random tensors on the sphere. *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, 2017. 320

[BHK+19] Boaz Barak, Samuel Hopkins, Jonathan Kelner, Pravesh K Kothari, Ankur Moitra, and Aaron Potechin. A nearly tight sum-of-squares lower bound for the planted clique problem. *SIAM Journal on Computing*, 48(2):687–735, 2019. 319

[Fei08] Uriel Feige. Small linear dependencies for binary vectors of low weight. In *Building Bridges: Between Mathematics and Computer Science*, pages 283–307. Springer, 2008. 315

[GKM22] Venkatesan Guruswami, Pravesh K Kothari, and Peter Manohar. Algorithms and certificates for boolean csp refutation: smoothed is no harder than random. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, pages 678–689, 2022. 9, 315, 318

[GKM25] Venkatesan Guruswami, Pravesh K Kothari, and Peter Manohar. New spectral algorithms for refuting smoothed k-sat. *Communications of the ACM*, 68(3):83–91, 2025. 9, 315

[Gop18] Sivakanth Gopi. *Locality in coding theory*. PhD thesis, Princeton University, 2018. 322

[Gri01] Dima Grigoriev. Linear lower bound on degrees of positivstellensatz calculus proofs for the parity. *Theoretical Computer Science*, 259(1-2):613–622, 2001. 308

[HKM23] Jun-Ting Hsieh, Pravesh K Kothari, and Sidhanth Mohanty. A simple and sharper proof of the hypergraph moore bound. In *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2324–2344. SIAM, 2023. 9, 301, 302, 303, 304, 306, 311, 315, 318

[HKM+25] Jun-Ting Hsieh, Pravesh K Kothari, Sidhanth Mohanty, David Munhá Correia, and Benny Sudakov. Small even covers, locally decodable codes and restricted subgraphs of edge-colored kikuchi graphs. *International Mathematics Research Notices*, 2025(5):rnaf045, 2025. 318

[HSS15] Samuel B Hopkins, Jonathan Shi, and David Steurer. Tensor principal component analysis via sum-of-square proofs. In *Conference on Learning Theory*, pages 956–1006. PMLR, 2015. 320

[KM24] Pravesh K Kothari and Peter Manohar. An exponential lower bound for linear 3-query locally correctable codes. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing*, pages 776–787, 2024. 311, 312

[KX26] Pravesh K Kothari and Jeff Xu. Smooth trade-off for tensor pca via sharp bounds for kikuchi matrices. In *Proceedings of the 2026 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2617–2632. SIAM, 2026. 322

[MNS15] Elchanan Mossel, Joe Neeman, and Allan Sly. Consistency thresholds for the planted bisection model. In *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*, pages 69–75, 2015. 319

[MsS01] Frank MsSherry. Spectral partitioning of random graphs. In *Proceedings 42nd IEEE Symposium on Foundations of Computer Science*, pages 529–537. IEEE, 2001. 319

[RRS17] Prasad Raghavendra, Satish Rao, and Tselil Schramm. Strongly refuting random csps below the spectral threshold. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 121–131, 2017. 307, 308, 310

[Sch08] Grant Schoenebeck. Linear level lasserre lower bounds for certain k-csps. In *2008 49th Annual IEEE Symposium on Foundations of Computer Science*, pages 593–602. IEEE, 2008. 308

[Spi26] Daniel Spielman. Spectral and algebraic graph theory. *Book draft*, 2026. 15, 319, 365

[Ver26] Roman Vershynin. *High-Dimensional Probability: An Introduction with Applications in Data Science*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 2 edition, 2026. 319

[WEAM25] Alexander Wein, Ahmed El Alaoui, and Cristopher Moore. The kikuchi hierarchy and tensor pca. *Journal of the ACM*, 72(5):1–40, 2025. 9, 307, 308, 310, 319, 320, 322

[Yek12] Sergey Yekhanin. Locally decodable codes. *Foundations and Trends® in Theoretical Computer Science*, 6(3):139–255, 2012. 312

Part VII

High-Dimensional Expanders and Mixing Time

The concept of high-dimensional expanders has led to major progress in several areas of theoretical computer science. We study its applications in the analysis of mixing time in Markov chains, developing a new framework for bounding spectral gaps and beyond.

Chapter 26

High-Dimensional Expanders

From Chapter 4 to Chapter 6, we have seen that expander graphs exhibit useful combinatorial, probabilistic, and algebraic properties, leading to a rich theory with connections and applications in diverse areas. An active research direction is to develop high-dimensional generalizations of expander graphs to further extend the theory and enhance its applications. Recent breakthroughs in this direction include constructions of locally testable codes [DEL+22], constructions of probabilistic checkable proofs [BMV24], and the resolution of the matroid expansion conjecture [ALOV19].

In this topic, we study a definition of high-dimensional expanders known as local spectral expanders and their applications in analyzing the mixing times of random walks. In this chapter, we introduce the basic concepts and prove a fundamental result by Oppenheim [Opp18], known as the trickling down theorem. The proof illustrates the local-to-global method, which is a common theme in the study of high-dimensional expanders.

26.1 Simplicial Complexes

A simplicial complex is a high-dimensional generalization of a graph.

Definition 26.1 (Simplicial Complex)**.** *A set system is a pair $X = (U, \mathcal{F})$ with U as the ground set and $\mathcal{F}$ is a family of subsets of U. A simplicial complex is a set system that is downward closed, meaning that if $\sigma \in \mathcal{F}$ and $\tau \subset \sigma$, then $\tau \in \mathcal{F}$.*

We follow the convention of using Greek letters τ, σ, η for subsets in $\mathcal{F}$, but reserve α, λ for eigenvalues. The following are some basic definitions related to simplicial complexes.

Definition 26.2 (Face, Dimension, Pure Simplicial Complex)**.** *Any subset $\sigma \in \mathcal{F}$ is called a face of the simplicial complex $X = (U, \mathcal{F})$. A face σ is of dimension k if its size is $|\sigma| = k + 1$. For example, a 0-dimensional face is a singleton (a vertex), a 1-dimensional face is a pair (an edge), a 2-dimensional face is a triple (a triangle), and so on.*

Given a simplicial complex $X = (U, \mathcal{F})$, we use $X(k)$ to denote the set of faces of dimension k. A simplicial complex is d-dimensional if the maximum face size is $d + 1$. A d-dimensional simplicial complex is pure if every maximal face has size $d + 1$.

Simplicial complex is a very general concept. We can associate a simplicial complex with many classes of combinatorial objects. The following is a relevant example of a pure simplicial complex.

Example 26.3 (Simplicial Complex from Spanning Trees). *Given a graph $G = (V, E)$, we can define a simplicial complex $X = (E, \mathcal{F})$ where the ground set in X is the edge set E of G. A subset of edges $F \subseteq E$ belongs to $\mathcal{F}$ if and only if F forms an acyclic subgraph in G.*

It is clear that X is a pure simplicial complex. When G is connected, the maximal faces correspond to spanning trees, which have size $|V| - 1$, so X is a $(|V| - 2)$-dimensional pure simplicial complex.

More generally, every matroid naturally corresponds to a simplicial complex.

Example 26.4 (Simplicial Complex from Matroids). *A matroid $M = (U, \mathcal{I})$ is a set system where U is the ground set and $\mathcal{I}$ is a collection of subsets of U satisfying the following two properties:*

1. *$\mathcal{I}$ is downward closed, i.e., if $S \in \mathcal{I}$ and $T \subseteq S$, then $T \in \mathcal{I}$.*
2. *If $S, T \in \mathcal{I}$ and $|S| > |T|$, then there exists $x \in S \setminus T$ such that $T \cup \{x\} \in \mathcal{I}$.*

By (1), $M = (U, \mathcal{I})$ forms a simplicial complex. By (2), $M = (U, \mathcal{I})$ is a pure simplicial complex. The sets in $\mathcal{I}$ are called independent sets, and the maximal sets are called bases of the matroid.

It can be checked that the simplicial complex from spanning trees is a matroid. A more general example is the class of linear matroids. Given a matrix $A \in \mathbb{F}^{m \times n}$, the linear matroid of A is defined as $M = ([n], \mathcal{I})$, where the ground set $[n]$ represents the columns of A, and a subset S of columns belongs to $\mathcal{I}$ if and only if the columns in S are linearly independent.

There are many more simplicial complexes that can be defined from combinatorial objects, such as simplicial complexes from cliques of graphs and simplicial complexes for graph coloring. We will discuss some of these in later chapters.

Weighted Simplicial Complexes

We consider pure simplicial complexes with weights on their faces. We follow the convention in [DDFH18] that the weights form a probability distribution on the faces of the same dimension.

Definition 26.5 (Weighted Simplicial Complexes). *A weighted pure simplicial complex (X, Π) is a pure simplicial complex with a probability distribution Π on the faces of maximal dimension.*

In applications of random sampling, the probability distribution in the maximal faces is usually uniform, but the following definition of induced distributions is generally non-uniform and plays a crucial role in our study.

Definition 26.6 (Induced Distributions). *Given a d-dimensional weighted pure simplicial complex (X, Π), a probability distribution Π_k on $X(k)$ for $0 \leq k \leq d$ is defined inductively as follows. The base case is $\Pi_d = \Pi$. For $d - 1 \geq k \geq 0$, the probability distribution $\Pi_k : X(k) \to \mathbb{R}$ is defined for each face $\tau \in X(k)$ as*

$$\Pi_k(\tau) = \frac{1}{|\tau| + 1} \sum_{\sigma \in X(k+1) : \sigma \supset \tau} \Pi_{k+1}(\sigma). \tag{26.1}$$

Equivalently, Π_k is the probability distribution arising from the following random process: First, sample a random face $\sigma \in X(d)$ using the probability distribution Π_d. Then, sample a uniform random subset τ of σ in $X(k)$, so that

$$\Pi_k(\tau) = \frac{1}{\binom{d+1}{|\tau|}} \sum_{\sigma \in X(d) : \sigma \supset \tau} \Pi_d(\sigma) = \frac{1}{\binom{d+1}{|\tau|}} \Pr_{\sigma \sim \Pi_d} [\sigma \supset \tau].$$

Note that $\Pi_k(\tau)$ is proportional to the marginal probability of the subset $\tau \subset \sigma$ when sampling a random maximal face $\sigma \sim \Pi_d$. Alternatively, $\Pi_k(\tau)$ can be interpreted as the (normalized) weighted degree of τ in the simplicial complex. We will often drop the subscript about the dimension of the face when it is clear or irrelevant from the context.

26.2 Local Spectral Expanders

There are various definitions of high-dimensional expanders, some using concepts from algebraic topology; see [Lub18, GK22] for a survey with motivations and applications. We study a more recent and elementary definition developed in [DK17, KO20], which was motivated by the study of random walks.

Links

The following is the key definition that enables a local-to-global approach for simplicial complexes.

Definition 26.7 (Links)**.** *Let $X = (U, \mathcal{F})$ be a simplicial complex. For a face $\tau \in \mathcal{F}$, the link X_τ is defined as*

$$X_\tau := \{\sigma \setminus \tau \mid \sigma \in \mathcal{F}, \sigma \supset \tau\}.$$

In words, X_τ consists of the faces η that can extend τ such that $\tau \cup \eta \in \mathcal{F}$.

If X is a pure d-dimensional simplicial complex and $\tau \in X$, then X_τ is a pure $(d - |\tau|)$-dimensional simplicial complex. In the spanning tree complex $X = (E, \mathcal{I})$, given a subset of edges $F \in \mathcal{I}$, the link X_F is defined such that a subset of edges F' is a face in X_F if and only if $F \cup F'$ is an acyclic subgraph. In matroid terminology, the link X_F is obtained by "contracting" the elements in F.

The probability distributions $\Pi_0, \dots, \Pi_d$ on X in Definition 26.6 can be used to define $\Pi_0^\tau, \dots, \Pi_{d-|\tau|}^\tau$ on X_τ using conditional probability, where $\Pi^\tau(\eta) \propto \Pr_{\sigma \sim \Pi_d}[\sigma \supset \eta \mid \sigma \supset \tau]$.

Definition 26.8 (Induced Distributions on Links)**.** *Let (X, Π) be a d-dimensional weighted pure simplicial complex. For any face τ and any $\eta \in X_\tau$,*

$$\Pi^\tau(\eta) := \Pr_{\sigma \sim \Pi_{|\tau|+|\eta|-1}} \left[\sigma = \tau \cup \eta \mid \sigma \supset \tau\right] = \frac{\Pi(\tau \cup \eta)}{\sum_{\sigma : |\sigma| = |\tau|+|\eta|, \sigma \supset \tau} \Pi(\sigma)} = \frac{\Pi(\tau \cup \eta)}{\binom{|\tau \cup \eta|}{|\tau|} \cdot \Pi(\tau)}, \tag{26.2}$$

where the last equality follows from Definition 26.6.

A general approach to study a simplicial complex is to decompose it into its links. For example, the following decomposition follows from the law of total probability $\Pr(\sigma) = \sum_\tau \Pr(\sigma \setminus \tau \mid \tau) \cdot \Pr(\tau)$.

Exercise 26.9 (Decomposition of Expectation Using Links)**.** *Let $f : X(k) \to \mathbb{R}$ be a function over faces of dimension k. For any $0 \le l < k$,*

$$\mathbb{E}_{\sigma \sim \Pi_k} f(\sigma) = \mathbb{E}_{\tau \sim \Pi_l} \, \mathbb{E}_{\eta \sim \Pi_{k-l-1}^\tau} f(\tau \cup \eta).$$

Skeletons and Graphs

Definition 26.10 (k-Skeletons)**.** *Given $X = (U, \mathcal{F})$, the k-skeleton of X is the simplicial complex $X_k = (U, \mathcal{F}_k)$, where $\mathcal{F}_k$ consists of the faces of $\mathcal{F}$ with dimension at most k. When there are weights on the faces in $\mathcal{F}$, we use the same weights on the faces in $\mathcal{F}_k$.*

The special case of the 1-skeleton is of particular interest, which can be interpreted as the underlying graph of the simplicial complex.

Definition 26.11 (Graph of Links)**.** *For a link X_τ, the graph $G_\tau = (X_\tau(0), X_\tau(1), \Pi_1^\tau)$ is defined as the 1-skeleton of X_τ. More explicitly, each singleton $\{i\}$ in X_τ corresponds to a vertex i in G_τ, each pair $\{i, j\}$ in X_τ corresponds to an edge ij in G_τ, and the weight of ij in G_τ is equal to $\Pi_1^\tau(\{i, j\})$.*

A simple observation is that if X is a pure d-dimensional simplicial complex and Π is the uniform distribution on $X(d)$, then for any $\tau \in X(d-2)$, the weighting Π_1^τ on the edges of G_τ is uniform. We will use this observation later.

Random Walk and Normalized Laplacian Matrices

The definition of local spectral expanders is based on the spectral gap of the random walk matrices or the normalized Laplacian matrices of the links.

Definition 26.12 (Random Walk and Normalized Laplacian Matrices of a Link)**.** *Given the graph $G_\sigma = (X_\sigma(0), X_\sigma(1), \Pi_1^\sigma)$ of a link X_σ, let A_σ be the adjacency matrix of G_σ and let D_σ be the diagonal degree matrix where*

$$D_\sigma(i,i) = \sum_{j \in X_\sigma(0)} A_\sigma(i,j) = \sum_{j \in X_\sigma(0)} \Pi_1^\sigma(\{i,j\}) = 2 \cdot \Pi_0^\sigma(\{i\}),$$

where the last equality follows from Definition 26.6 and Definition 26.8.

The random walk matrix W_σ of G_σ is defined as

$$W_\sigma := D_\sigma^{-1} A_\sigma \quad \text{where} \quad W_\sigma(i,j) = \frac{\Pi_1^\sigma(\{i,j\})}{2\Pi_0^\sigma(\{i\})} = \frac{\Pi(\sigma \cup \{i,j\})}{(|\sigma|+2) \cdot \Pi(\sigma \cup \{i\})} \quad \text{for all } \{i,j\} \in X_\sigma(1). \tag{26.3}$$

The stationary distribution of W_σ is Π_0^σ, which follows from the time reversibility condition

$$\Pi_0^\sigma(i) \cdot W_\sigma(i,j) = \Pi_0^\sigma(j) \cdot W_\sigma(j,i). \tag{26.4}$$

The Laplacian matrix L_σ and the normalized Laplacian matrix $\tilde{\mathcal{L}}_\sigma$ of G_σ are defined as

$$L_\sigma = D_\sigma - A_\sigma \quad \text{and} \quad \tilde{\mathcal{L}}_\sigma = I - W_\sigma.$$

Note that $\tilde{\mathcal{L}}$ is not the same as the normalized Laplacian matrix $\mathcal{L}$ in Definition 1.17 since $\tilde{\mathcal{L}}$ is not necessarily symmetric, but they are similar matrices.

Local Spectral Expanders

Finally, we can state the definition of high-dimensional expanders that we will use.

Definition 26.13 (Local Spectral Expanders [DK17, KO20])**.** *Let (X, Π) be a weighted pure d-dimensional simplicial complex. We say (X, Π) is a g-local-spectral expander if*

$$\lambda_2(\tilde{\mathcal{L}}_\sigma) \geq g \quad \text{or} \quad \alpha_2(W_\sigma) \leq 1 - g \quad \text{for all } \sigma \in X(k) \text{ for all } -1 \leq k \leq d-2.$$

where $\lambda_2(\tilde{\mathcal{L}}_\sigma)$ is the second smallest eigenvalue of the normalized Laplacian matrix $\tilde{\mathcal{L}}_\sigma$, and $\alpha_2(W_\sigma)$ is the second largest eigenvalue of the random walk matrix W_σ.

More generally, given $g_{-1}, \ldots, g_{d-2}$, we say (X, Π) is a $(g_{-1}, \ldots, g_{d-2})$-local-spectral expander if

$$\lambda_2(\tilde{\mathcal{L}}_\sigma) \geq g_k \quad \text{or} \quad \alpha_2(W_\sigma) \leq 1 - g_k \quad \text{for all } \sigma \in X(k) \text{ for all } -1 \leq k \leq d-2.$$

Remark 26.14 (Non-Standard Notations)**.** *The definitions in [DK17, KO20] assume that $\alpha_2(W_\sigma) \leq \gamma$, while the definition above assumes that $\lambda_2(\tilde{\mathcal{L}}_\sigma) \geq g$ or $\alpha_2(W_\sigma) \leq 1 - g$. This slight modification is motivated by the fact that the normalized Laplacian matrix simplifies calculations significantly, as demonstrated by Mohanty [Moh22]. We elaborate on these simplifications in the proof of Oppenheim's theorem in this chapter and the analysis of higher-order random walks in the next chapter.*

We can interpret this definition as requiring that the "local" random walks in each link graph mix rapidly. We can also interpret it as requiring that the "local" weighted graphs of the links have large edge conductance following from Cheeger's inequality.

Example 26.15 (Complete Complex)**.** *Consider the complete complex $X_d = (U, \mathcal{F}_d)$, where every subset $S \subseteq U$ with $|S| \leq d+1$ belongs to $\mathcal{F}_d$, equipped with the uniform distribution on the faces of dimension d. Then the graph of every link of dimension k is an unweighted complete graph with $d-k$ vertices, where the second smallest eigenvalue of the normalized Laplacian matrix being $1 + 1/(d-k-1)$.*

It is not surprising that a complete complex is a strong high-dimensional expander. Just as in expander graphs (Chapter 5), the goal is to construct high-dimensional expanders with few maximal faces. Unlike in the graph case, however, random simplicial complexes are *not* high-dimensional expanders with high probability. Constructing sparse high-dimensional expanders is a challenging task, and only a few explicit algebraic constructions are known [Lub18].

26.3 Oppenheim's Trickling Down Theorem

To show that a simplicial complex is a g-local-spectral expander, we need to bound the second eigenvalue for every link X_σ up to dimension $d-2$. In applications where the goal is uniform sampling of the maximal faces, it is usually much easier to work with the graphs of the top links of dimension $d-2$, because they are unweighted as we observed earlier. For lower links, just determining the edge weights may already involve intractable counting problems. Thus, it would be very convenient if we could bound the second eigenvalues of the lower links by those of the top links. Oppenheim's trickling down theorem [Opp18] provides such a general bound for any pure simplicial complex.

Theorem 26.16 (Oppenheim's Trickling Down Theorem [Opp18]). *Let (X, Π) be a pure d-dimensional weighted simplicial complex, where Π satisfies Definition 26.6 and Definition 26.8. Suppose the graph $G_\emptyset = (X(0), X(1), \Pi_1)$ is connected and $\lambda_2(\tilde{\mathcal{L}}_i) \geq g$ for all $i \in X(0)$. Then*

$$\lambda_2(\tilde{\mathcal{L}}_\emptyset) \geq 2 - \frac{1}{g}.$$

The condition that the graph $G_\emptyset$ is connected is necessary, as seen in the case of two disjoint cliques.

In general, the spectral gap bound deteriorates as we go to lower links. However, if we can establish that $g \geq 1$, then the trickling down theorem implies that the simplicial complex is a 1-local-spectral expander, which is nearly as strong as the complete complex in Example 26.15.

Applying Theorem 26.16 inductively gives the following bound, which implies that if $g \geq 1 - O(\frac{1}{d^2})$ then the simplicial complex is a $\left(1 - O(\frac{1}{d^2})\right)$-local-spectral expander. The proof is left as an exercise.

Theorem 26.17 (Oppenheim's Bound [Opp18]). *Let (X, Π) be a pure d-dimensional weighted simplicial complex, where Π satisfies Definition 26.6 and Definition 26.8. Suppose $\lambda_2(\tilde{\mathcal{L}}_\sigma) \geq g \geq 1 - \frac{1}{d}$ for every face σ of dimension $d-2$, and G_σ is connected for every face σ. Then, for every $k \leq d-2$,*

$$1 - \lambda_2(\tilde{\mathcal{L}}_\tau) \leq \frac{1-g}{1-(d-2-k)(1-g)} \quad \text{for every } \tau \in X(k).$$

Matroid Complex

The following result is a key step in establishing the matroid expansion conjecture, which will be proved in the next chapter.

Theorem 26.18 (Matroid Complex is 1-Local-Spectral Expander [ALOV19]). *The simplicial complex of any matroid (Example 26.4), with the uniform distribution on the maximal faces, is a 1-local-spectral expander.*

Proof. Let X be a pure d-dimensional simplicial complex from a matroid M. By Oppenheim's trickling down theorem, it suffices to prove that (i) the graph of every link is connected, and (ii) the second largest eigenvalue of the random walk matrix of the links of dimension $d-2$ is at most 0.

The first claim that the graph of every link is connected follows from the second axiom of matroids stated in Example 26.4 and is left as a simple exercise.

For the second claim, consider the adjacency matrix A_σ of a face σ of dimension $d-2$. Since the probability distribution on the maximal faces is uniform, every non-zero entry of A_σ has the same weight. Without loss of generality, we rescale the matrix such that $A_\sigma(i,j) = 1$ if $\sigma \cup \{i,j\}$ is a maximal face, and $A_\sigma(i,j) = 0$ otherwise. We aim to show that A_σ has at most one positive eigenvalue, which implies that the normalized adjacency matrix $\mathcal{A}_\sigma$ also has at most one positive eigenvalue by Courant-Fischer (Theorem A.14). Since W_σ and $\mathcal{A}_\sigma$ are similar matrices, this would further imply that the random walk matrix W_σ has at most one positive eigenvalue.

To establish that A_σ has at most one positive eigenvalue, we first illustrate this with the spanning tree complex from Example 26.3. Consider the spanning tree complex $X = (E, \mathcal{F})$ of a graph $G = ([n], E)$. The maximal faces have size $n-1$, so the dimension of the complex is $d = n-2$. Given a face $F \subseteq E$ of dimension $d-2$, with $|F| = n-3$, the subgraph formed by the edges in F consists of three connected components. The edges remaining in the link X_F are those with

endpoints in different components. Two edges e, f in X_F form a face of size 2 if and only if $F \cup \{e, f\}$ forms a spanning tree, which happens precisely when e and f are not parallel edges in the contracted graph obtained by contracting the three components into single vertices. This structure implies that the edges in X_F can be partitioned into three equivalent classes E_1, E_2, E_3, such that two edges e, f form a face of size 2 in X_F if and only if they do not belong to the same class. Thus, the adjacency matrix A_F can be written as

$$A_F = J - \chi_{E_1}\chi_{E_1}^\top - \chi_{E_2}\chi_{E_2}^\top - \chi_{E_3}\chi_{E_3}^\top,$$

where J is the all-one matrix, and χ_{E_i} is the characteristic vector of E_i for $i = 1, 2, 3$. Since A_F is a rank-one matrix minus three positive semidefinite matrices, it follows that A_F has at most one positive eigenvalue. This completes the proof for spanning tree complexes.

The same proof extends to linear matroids, where two columns i, j form a face of size 2 if and only if they are parallel in the linear algebraic sense. Thus, the columns can be partitioned into equivalence classes $E_1, E_2, \ldots, E_l$ (with l not necessarily equal to 3), leading to the decomposition $A = J - \sum_{i=1}^{l} \chi_{E_i}\chi_{E_i}^\top$. Again, this ensures that A has at most one positive eigenvalue.

In general, this property holds for all matroids, known as the matroid partition property, and thus the proof extends to general matroids. □

One may wonder what other simplicial complexes are 1-local-spectral expanders. Problem 26.22 shows that they must have very restrictive structures, requiring that the graphs of the top links be complete multi-partite graphs.

26.4 Local-to-Global Method

The main goal in this section is to prove Theorem 26.16. We follow the exposition in [Moh22], which simplifies the calculations by considering normalized Laplacian matrices rather than the random walk matrices.

In the following, we first introduce a local-to-global formulation of the spectral gap of the normalized Laplacian matrix and then use it to derive a proof of the trickling down theorem.

Local-to-Global Characterization of Spectral Gap

Recall from Section 3.3 that the random walk matrix W and the normalized adjacency matrix $\mathcal{A}$ of a graph are similar matrices. This implies that they share the same eigenvalues, but the eigenvectors of W are not necessarily orthonormal with respect to the standard inner product.

Given a random walk matrix $W = D^{-1}A$, a similar calculation as in Lemma 3.17 (where we considered $W^\top$) shows that the eigenvectors $u_1, \ldots, u_n \in \mathbb{R}^n$ of W satisfy $\langle u_i, u_j \rangle_D = 0$. It is more convenient to use the stationary distribution π, which is proportional to the degrees, to define the inner product

$$\langle f, g \rangle_\pi := \sum_{i=1}^{n} \pi(i) f(i) g(i) = \mathbb{E}_{i \sim \pi}[f(i)g(i)],$$

as it can be interpreted as an expected value.

The second eigenvalue can be characterized using Rayleigh quotient under this inner product. The proof is similar to that in Lemma 2.3.

Lemma 26.19 (Rayleigh Quotient Characterization of Spectral Gap). *Let $G = (V, E)$ be an undirected graph. Let $W = D^{-1}A$ be its random walk matrix with stationary distribution π, and let $\tilde{\mathcal{L}} := I - W$. The second eigenvalue of $\tilde{\mathcal{L}}$ is*

$$\lambda_2(\tilde{\mathcal{L}}) = \min_{f:\langle f,\vec{1}\rangle_\pi=0} \frac{\langle f, \tilde{\mathcal{L}} f\rangle_\pi}{\langle f, f\rangle_\pi} = \min_{f:\langle f,\vec{1}\rangle_\pi=0} \frac{\sum_{ij\in E} A(i,j)\cdot(f(i)-f(j))^2}{\sum_{i\in V} D(i,i)\cdot f(i)^2}.$$

The following geometric characterization of the second eigenvalue of the normalized Laplacian matrix is well-known and has applications in metric embeddings and approximation algorithms (see Chapter 20 and Problem 26.25). It provides a local-to-global characterization, where the numerator is the average squared edge length and the denominator is the average squared pairwise distance. This formulation aligns naturally with the local-to-global method in high-dimensional expanders.

Lemma 26.20 (Local-to-Global Characterization of Spectral Gap). *Let $G = (V, E)$ be an undirected graph. Let $W = D^{-1}A$ be its random walk matrix with stationary distribution π, and let $\tilde{\mathcal{L}} := I - W$. The second eigenvalue of $\tilde{\mathcal{L}}$ is*

$$\lambda_2(\tilde{\mathcal{L}}) = \min_f \frac{\mathbb{E}_{i\sim\pi, j\sim W(i,\cdot)}[(f(i)-f(j))^2]}{\mathbb{E}_{i\sim\pi,j\sim\pi}[(f(i)-f(j))^2]},$$

where the notation $i \sim \pi, j \sim W(i,\cdot)$ means first sampling a random vertex i from π and then sampling a random neighbor of i according to the distribution $W(i,\cdot)$.

Proof. Starting from the characterization in Lemma 26.19, we claim that

$$\lambda_2(\tilde{\mathcal{L}}) = \min_{f:\langle f,\vec{1}\rangle_\pi=0} \frac{\sum_{ij\in E} A(i,j)\cdot(f(i)-f(j))^2}{\sum_{i\in V} D(i,i)\cdot f(i)^2} = \min_f \max_c \frac{\sum_{ij\in E} A(i,j)\cdot(f(i)-f(j))^2}{\sum_{i\in V} D(i,i)\cdot(f(i)-c)^2}.$$

In one direction, an optimal function f on the left-hand side achieves the same objective value on the right-hand side, since shifting does not increase the objective value given that $\langle f, \vec{1}\rangle_\pi = 0 = \langle f, \vec{1}\rangle_D$ (see Lemma 2.8). In the other direction, for any function f, the optimal choice of c in the inner maximization problem is

$$c = \sum_{i\in V} D(i,i)\cdot f(i) \Big/ \sum_{i\in V} D(i,i).$$

Using $\pi(i) = D(i,i)/\sum_j D(j,j) = D(i,i)/(2|E|)$, we see that the function $f - c\vec{1}$ is a feasible solution on the left-hand side with the same objective value. Then, substituting the optimal choice of c, the denominator on the right-hand side simplifies to

$$\begin{aligned}
\sum_{i\in V} D(i,i)\cdot(f(i)-c)^2 &= 2|E|\cdot\sum_{i\in V}\pi(i)\cdot f(i)^2 - 2|E|\cdot\Big(\sum_{i\in V}\pi(i)\cdot f(i)\Big)^2 \\
&= |E|\sum_{i,j\in V}\pi(i)\cdot\pi(j)\cdot(f(i)-f(j))^2 \\
&= |E|\cdot\mathbb{E}_{i\sim\pi,j\sim\pi}[(f(i)-f(j))^2].
\end{aligned}$$

Finally, a simple calculation shows that

$$\sum_{ij\in E} A(i,j)\cdot(f(i)-f(j))^2 = |E|\cdot\mathbb{E}_{i\sim\pi,j\sim W(i,\cdot)}[(f(i)-f(j))^2].$$

Plugging these equalities into the first line proves the lemma. □

Proof of the Trickling Down Theorem

The usual proofs of the trickling down theorem rely on the Rayleigh quotient characterization of the random walk matrix, given by

$$\alpha_2(W_\sigma) = \max_{f:\langle f,\vec{1}\rangle_{\Pi_0^\sigma}=0} \frac{\langle f, W_\sigma f\rangle_{\Pi_0^\sigma}}{\langle f, f\rangle_{\Pi_0^\sigma}}. \tag{26.5}$$

The approach involves decomposing $W_\emptyset$ into the random walk matrices W_i of its singleton links, and decomposing f as $f = f_i^{\parallel} + f_i^{\perp}$, where $\langle f_i^{\perp}, \vec{1}\rangle_{\Pi_0^i} = 0$, to bound the quadratic form of $\langle f_i^{\perp}, W_i f_i^{\perp}\rangle$.

In [Moh22], the unconstrained characterization in Lemma 26.20 is used to bypass the decomposition of f. Moreover, the denominator in this formulation fits naturally with the local-to-global method in high-dimensional expanders.

Proof of Theorem 26.16. Let $\tilde{\mathcal{L}} := \tilde{\mathcal{L}}_\emptyset$ and $W := W_\emptyset$ be the normalized Laplacian matrix and the random walk matrix of the empty link. The stationary distribution of W is Π_0 as shown in Definition 26.12. Let f be an optimizer in the characterization of $\tilde{\mathcal{L}}$ in Lemma 26.20, normalized so that $\mathbb{E}_{i\sim\Pi_0, j\sim\Pi_0}[(f(i)-f(j))^2] = 1$.

In the following, we first outline the main steps leading to the conclusion, followed by detailed explanations. Readers may find it helpful to read each step, then refer to the explanations before proceeding further. We claim that

$$\begin{aligned}
\lambda_2(\tilde{\mathcal{L}}) &= \mathbb{E}_{i\sim\Pi_0, j\sim W(i,\cdot)}[(f(i)-f(j))^2] && (26.6)\\
&= \mathbb{E}_{l\in\Pi_0}\mathbb{E}_{i\sim\Pi_0^l, j\sim W_l(i,\cdot)}[(f(i)-f(j))^2] && (26.7)\\
&\geq g\cdot\mathbb{E}_{l\in\Pi_0}\mathbb{E}_{i\sim\Pi_0^l, j\sim\Pi_0^l}[(f(i)-f(j))^2] && (26.8)\\
&= g\cdot\mathbb{E}_{i\sim\Pi_0, j\sim W^2(i,\cdot)}[(f(i)-f(j))^2] && (26.9)\\
&\geq g\cdot\lambda_2(\tilde{\mathcal{L}})\cdot(2-\lambda_2(\tilde{\mathcal{L}}).) && (26.10)
\end{aligned}$$

Since the graph G of the empty link is connected, it follows from Proposition 1.15 that $\lambda_2(\tilde{\mathcal{L}}) > 0$, leading to the conclusion that $1 \geq g\cdot(2-\lambda_2(\tilde{\mathcal{L}}))$ and thus $\lambda_2(\tilde{\mathcal{L}}) \geq 2-\frac{1}{g}$. We now justify each step.

Explanation of (26.6): This follows from Lemma 26.20 since $\mathbb{E}_{i\sim\Pi_0, j\sim\Pi_0}[(f(i)-f(j))^2] = 1$ and f is an optimizer.

Explanation of (26.7): This is the key step of decomposing the quadratic form of the empty link into the quadratic forms of the singleton links. To verify this, observe that the weight of each term $(f(i)-f(j))^2$ in (26.7) is

$$\sum_l \Pi_0(l)\cdot\Big(\Pi_0^l(i)\cdot W_l(i,j) + \Pi_0^l(j)\cdot W_l(j,i)\Big) = \sum_l \Pi_0(l)\cdot\Pi_1^l(ij) = \frac{1}{3}\sum_l \Pi_2(\{l,i,j\}) = \Pi_1(ij),$$

where we used (26.3), (26.2), and (26.1). This matches the corresponding weight in (26.6), which is $\Pi_0(i)\cdot W(i,j) + \Pi_0(j)\cdot W(j,i) = \Pi_1(ij)$ by (26.3).

Explanation of (26.8): Apply Lemma 26.20 on the random walk matrix W_l of each link X_l, with the stationary distribution being Π_0^l from Lemma 26.20 and the assumption that $\lambda_2(\tilde{\mathcal{L}}_l) \geq g$.

Explanation of (26.9): The weight of each term $(f(i)-f(j))^2$ on (26.8) is

$$2\sum_l \Pi_0(l)\cdot\Pi_0^l(i)\cdot\Pi_0^l(j) = 2\sum_l \frac{\Pi_1(i,l)\cdot\Pi_1(l,j)}{4\cdot\Pi_0(l)} = 2\sum_l \Pi_0(i)\cdot W(i,l)\cdot W(l,j) = 2\Pi_0(i)\cdot W^2(i,j),$$

where we used (26.2) and (26.3). The notation $j \sim W^2(i,\cdot)$ means sampling a random vertex by starting a 2-step random walk of W from vertex i. Note that the stationary distribution of W^2 is still Π_0. This matches the weight of each term $(f(i)-f(j))^2$ on (26.9), which is

$$\Pi_0(i)\cdot W^2(i,j)+\Pi_0(j)\cdot W^2(j,i)=2\Pi_0(i)\cdot W^2(i,j),$$

where the time reversibility of W^2 follows from that of W.

Explanation of (26.10): Since f is an eigenvector of $I-W$, f is still an eigenvector of $I-W^2$, with eigenvalue

$$1-(1-\lambda_2(\tilde{\mathcal{L}}))^2=\lambda_2(\tilde{\mathcal{L}})\cdot(2-\lambda_2(\tilde{\mathcal{L}})).$$

Since the stationary distribution of W^2 is still Π_0, we can apply Lemma 26.20 to $I-W^2$ to obtain

$$\mathbb{E}_{i\sim\Pi_0,j\sim W^2(i,\cdot)}[(f(i)-f(j))^2]\geq\lambda_2(\tilde{\mathcal{L}})\cdot(2-\lambda_2(\tilde{\mathcal{L}}))\cdot\mathbb{E}_{i\sim\Pi_0,j\sim\Pi_0}[(f(i)-f(j))^2]=\lambda_2(\tilde{\mathcal{L}})\cdot(2-\lambda_2(\tilde{\mathcal{L}})),$$

where the last equality is by our assumption that $\mathbb{E}_{i\sim\Pi_0,j\sim\Pi_0}[(f(i)-f(j))^2]=1$. □

This approach of decomposing a global function into local functions on its links is often called Garland's method in the literature.

Further Developments

The spectral gap bound in the trickling down theorem deteriorates quickly as we go to lower links. For many simplicial complexes from combinatorial problems, however, it is conjectured that the spectral gap bound improves as we move to lower links. Recent work has developed improved trickling down theorems and applied them to obtain strong results in random sampling [ALO22, AO23, WZZ24, LLO25].

26.5 Problems

Exercise 26.21 (Product Distributions). *Let $X=(E,\mathcal{I})$ be a matroid complex. Suppose each element $e\in E$ has a weight w_e. Consider the probability distribution Π where each maximal face F has probability $\Pi(F)$ proportional to $\prod_{e\in F}w_e$. Show that (X,Π) is still a 1-local-spectral expander.*

Problem 26.22 (Complete Multi-Partite Graphs). *Prove that the adjacency matrix of a graph G has at most one positive eigenvalue if and only if G is a complete multi-partite graph.*

Problem 26.23 (Approximate Negative Correlation of Matroids). *In Problem 14.25, we have seen that the variables in a random spanning tree are negatively correlated, such that for any two edges $e\neq f$,*

$$\Pr_T[e\in T\mid f\in T]\leq\Pr_T[e\in T].$$

This is known to be not necessarily true for general matroids, but not all is lost. Use the result that any matroid complex is a 1-local-spectral expander in Theorem 26.18 to prove that for any two elements $i\neq j$ in a matroid,

$$\Pr_B[i\in B\mid j\in B]\leq 2\Pr_B[i\in B],$$

where B is a uniform random basis of the matroid.

Question 26.24. *It is an open question what is the best constant that one could prove for the approximate negative correlation property of matroids in Problem 26.23. There are examples showing that the constant is at least* $8/7$*, and some conjectured that this is tight.*

Problem 26.25 (Metric Embedding)**.** *A metric* $d(x, y)$ *can be embedded into* ℓ_2^2 *with distortion* D *if there exists a function* f *such that*

$$\|f(x) - f(y)\|_2^2 \leq d(x, y) \leq D \cdot \|f(x) - f(y)\|_2^2 \quad \text{for every } x, y.$$

Use Lemma 26.20 to show that any embedding of the shortest path distance of a constant-degree expander graph on n *vertices into* ℓ_2^2 *must have distortion* $\Omega(\log n)$*.*

References

[ALO22] Dorna Abdolazimi, Kuikui Liu, and Shayan Oveis Gharan. A matrix trickle-down theorem on simplicial complexes and applications to sampling colorings. In *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 161–172. IEEE, 2022. 338, 354

[ALOV19] Nima Anari, Kuikui Liu, Shayan Oveis Gharan, and Cynthia Vinzant. Log-concave polynomials II: high-dimensional walks and an FPRAS for counting bases of a matroid. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, STOC*, pages 1–12. ACM, 2019. 9, 329, 334, 344

[AO23] Dorna Abdolazimi and Shayan Oveis Gharan. An improved trickle down theorem for partite complexes. In *38th Computational Complexity Conference (CCC 2023)*, pages 10–1. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2023. 338

[BMV24] Mitali Bafna, Dor Minzer, and Nikhil Vyas. Quasi-linear size pcps with small soundness from hdx. *arXiv preprint arXiv:2407.12762*, 2024. 329

[DDFH18] Yotam Dikstein, Irit Dinur, Yuval Filmus, and Prahladh Harsha. Boolean function analysis on high-dimensional expanders. *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, 2018. 330

[DEL+22] Irit Dinur, Shai Evra, Ron Livne, Alexander Lubotzky, and Shahar Mozes. Locally testable codes with constant rate, distance, and locality. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, pages 357–374, 2022. 77, 83, 329

[DK17] Irit Dinur and Tali Kaufman. High dimensional expanders imply agreement expanders. In *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS*, pages 974–985. IEEE Computer Society, 2017. 331, 333

[GK22] Roy Gotlib and Tali Kaufman. Nowhere to go but high: a perspective on high-dimensional expanders. In *Proc. Int. Cong. Math*, volume 6, pages 4842–4871, 2022. 331

[KO20] Tali Kaufman and Izhar Oppenheim. High order random walks: Beyond spectral gap. *Comb.*, 40(2):245–281, 2020. 9, 331, 333, 343

[LLO25] Jonathan Leake, Kasper Lindberg, and Shayan Oveis Gharan. Optimal trickle-down theorems for path complexes via lorentzian polynomials with applications to sampling and log-concave sequences. *arXiv preprint arXiv:2503.01005*, 2025. 338

[Lub18] Alexander Lubotzky. High dimensional expanders. In *Proceedings of the International Congress of Mathematicians (ICM 2018)*, pages 705–730, 2018. 9, 331, 333

[Moh22] Sidhanth Mohanty. Local-to-global theorems for high-dimensional expansion. 2022. https://sidhanthm.com/pdf/local-to-global.pdf. 333, 335, 337, 345

[Opp18] Izhar Oppenheim. Local spectral expansion approach to high dimensional expanders part I: descent of spectral gaps. *Discrete & Computational Geometry*, 59(2):293–330, 2018. 9, 329, 333, 334

[WZZ24] Yulin Wang, Chihao Zhang, and Zihan Zhang. Sampling proper colorings on line graphs using $(1+o(1))\delta$ colors. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing*, pages 1688–1699, 2024. 338, 354

Chapter 27

Higher Order Random Walks

We study two related random walks on simplicial complexes, called the down-up walks and up-down walks. The main result is that they are fast mixing for simplicial complexes that are good local-spectral expanders. A consequence is that the natural random walks on matroid bases mix rapidly, proving the long-standing matroid expansion conjecture.

27.1 Random Walks on Simplicial Complexes

Kaufman and Mass [KM17] defined two natural random walks on faces of dimension k in a simplicial complex: the up-down walks, which pass through faces of dimension $k+1$, and the down-up walks, which pass through faces of dimension $k-1$. An intuitive way to describe these walks is through the following bipartite graphs.

Definition 27.1 (Bipartite Graph of a Layer)**.** *Let (X,Π) be a pure d-dimensional simplicial complex. For any $-1 \le k \le d-1$, the bipartite graph $H_k = (X(k), X(k+1); E)$ has one vertex for each face in $X(k) \cup X(k+1)$, with an edge between a face $\tau \in X(k)$ and a face $\sigma \in X(k+1)$ if and only if $\tau \subset \sigma$. The weight of this edge is given by $\frac{1}{k+2} \cdot \Pi_{k+1}(\sigma)$.*

Up and Down Operators

We consider the random walk matrix of these bipartite graphs and define the up and down operators, which correspond to one-step random walks on the bipartite graphs in Definition 27.1.

Definition 27.2 (Up and Down Operators)**.** *Let (X,Π) be a pure d-dimensional simplicial complex. Let A_k be the adjacency matrix of H_k with*

$$A_k(\tau,\sigma) = A_k(\sigma,\tau) = \frac{1}{k+2} \cdot \Pi_{k+1}(\sigma) \quad \text{if } \tau \subset \sigma \quad \text{for } \tau \in X(k), \sigma \in X(k+1)$$

and zero otherwise. For each face $\tau \in X(k)$, the weighted degree of τ is

$$\deg(\tau) := \sum_{\sigma \in X(k+1): \sigma \supset \tau} A_k(\tau,\sigma) = \sum_{\sigma \in X(k+1): \sigma \supset \tau} \frac{1}{k+2} \cdot \Pi_{k+1}(\sigma) = \Pi_k(\tau),$$

where the last equality is by Equation 26.1. For each face $\sigma \in X(k+1)$, the weighted degree of σ is

$$\deg(\sigma) := \sum_{\tau \in X(k): \tau \subset \sigma} A_k(\tau,\sigma) = \sum_{\tau \in X(k): \tau \subset \sigma} \frac{1}{k+2} \cdot \Pi_{k+1}(\sigma) = \Pi_{k+1}(\sigma).$$

The random walk matrix W_k of H_k is

$$W_k = \begin{pmatrix} 0 & D_{k+1} \\ U_k & 0 \end{pmatrix},$$

where D_{k+1} is a $X(k) \times X(k+1)$ matrix and U_k is a $X(k+1) \times X(k)$ matrix with

$$D_{k+1}(\tau, \sigma) = \frac{A_k(\tau, \sigma)}{\deg(\tau)} = \frac{\Pi_{k+1}(\sigma)}{(k+2) \cdot \Pi_k(\tau)} \quad \text{and} \quad U_k(\sigma, \tau) = \frac{A_k(\sigma, \tau)}{\deg(\sigma)} = \frac{1}{k+2}.$$

for $\tau \in X(k)$ and $\sigma \in X(k+1)$ satisfying $\tau \subset \sigma$. The matrix D_{k+1} is called the down operator from $X(k+1)$ to $X(k)$, and U_k is called the up operator from $X(k)$ to $X(k+1)$.

The following remark clarifies the naming convention of these operators.

Remark 27.3 (Down-Up Confusion)**.** *The term down operator comes from the perspective that D_{k+1} is an operator that maps a function $f : X(k+1) \to \mathbb{R}$ to a function $g = D_{k+1} f : X(k) \to \mathbb{R}$, so the output is one dimension lower, hence the name down operator. In other words, the naming convention arises from right-multiplication on the matrix.*

When considering random walks, however, we apply left-multiplication in the form $p^\top W_k$. From this perspective, D_{k+1} actually maps a distribution on $X(k)$ to a distribution on $X(k+1)$, so the output is one dimension higher. This may seem counterintuitive when thinking about random walks, but it is not a major issue since we rarely discuss these operators in isolation.

A useful property is the adjoint property of the up and down operators.

Exercise 27.4 (Adjoint Property)**.** *Let (X, Π) be a pure d-dimensional simplicial complex. Show that for any $f : X(k) \to \mathbb{R}$ and $g : X(k+1) \to \mathbb{R}$,*

$$\langle U_k f, g \rangle_{\Pi_{k+1}} = \langle f, D_{k+1} g \rangle_{\Pi_k}.$$

Up-Down Walks and Down-Up Walks

The two random walks defined by Kaufman and Mass correspond to two-steps random walks on the bipartite graphs in Definition 27.1.

Definition 27.5 (Up-Down Walks and Down-Up Walks [KM17])**.** *Let (X, Π) be a pure d-dimensional simplicial complex. Let H_k be the bipartite graph in Definition 27.1, and let W_k be the random walk matrix on H_k in Definition 27.2. Consider*

$$W_k^2 = \begin{pmatrix} D_{k+1}U_k & 0 \\ 0 & U_k D_{k+1} \end{pmatrix} =: \begin{pmatrix} P_k^\triangle & 0 \\ 0 & P_{k+1}^\triangledown \end{pmatrix},$$

where $P_k^\triangle \in \mathbb{R}^{X(k) \times X(k)}$ is called the up-down walk matrix and $P_{k+1}^\triangledown \in \mathbb{R}^{X(k+1) \times X(k+1)}$ is called the down-up walk matrix.

A simple but important property of $P_k^\triangle$ and $P_{k+1}^\triangledown$ is that they have the same spectrum (see Fact A.28). This will be used in an inductive proof to analyze the spectrum of $P_d^\triangledown$.

Fact 27.6 (Same Spectrum of $P_k^{\triangle}$ and $P_{k+1}^{\triangledown}$)**.** *There is a one-to-one correspondence between the non-zero eigenvalues of $P_k^{\triangle}$ and $P_{k+1}^{\triangledown}$.*

It is helpful to write out the entries of $P_k^{\triangle}$ and $P_{k+1}^{\triangledown}$ explicitly.

Fact 27.7 (Entries of $P_k^{\triangle}$ and $P_{k+1}^{\triangledown}$)**.** *Let (X, Π) be a pure d-dimensional simplicial complex. For $\tau, \tau' \in X(k)$,*

$$P_k^{\triangle}(\tau, \tau') = \begin{cases} \frac{1}{k+2} & \text{if } \tau = \tau' \\ \frac{\Pi_{k+1}(\tau \cup \tau')}{(k+2)^2 \cdot \Pi_k(\tau)} & \text{if } \tau \cup \tau' \in X(k+1) \\ 0 & \text{otherwise.} \end{cases}$$

For $\sigma, \sigma' \in X(k+1)$.

$$P_{k+1}^{\triangledown}(\sigma, \sigma') = \begin{cases} \sum_{\tau \in X(k): \tau \subset \sigma} \frac{\Pi_{k+1}(\sigma')}{(k+2)^2 \cdot \Pi_k(\tau)} & \text{if } \sigma = \sigma' \\ \frac{\Pi_{k+1}(\sigma')}{(k+2)^2 \cdot \Pi_k(\sigma \cap \sigma')} & \text{if } \sigma \cap \sigma' \in X(k) \\ 0 & \text{otherwise.} \end{cases}$$

Notice that $P_0^{\triangle}$ is just the standard lazy random walks on a graph.

The stationary distributions of $P_k^{\triangle}$ and $P_k^{\triangledown}$ are the same. This can be verified by direct computation or by checking that the time reversible condition (i.e., $\pi(i)P(i,j) = \pi(j)P(j,i)$ for all i, j) holds.

Fact 27.8 (Stationary Distributions)**.** *The stationary distributions of $P_k^{\triangle}$ and $P_k^{\triangledown}$ are Π_k.*

Random Walks on Matroid Bases

To sample a uniform random basis of a matroid, we consider the matroid complex with the uniform distribution on its bases and run the down-up walk $P_d^{\triangledown}$. By Fact 27.8, the stationary distribution is the uniform distribution.

The down-up walk $P_d^{\triangledown}$ corresponds to the natural algorithm where we start from an arbitrary basis B_0 and, in each iteration $t \geq 0$, drop a uniform random element i from the current basis and add a random element j so that $B_{t+1} := B_t - i + j$ remains a basis. This process is repeated iteratively. Observe that the random spanning tree algorithm in Algorithm 2 is a special case.

27.2 Kaufman-Oppenheim Theorem and Matroid Expansion

Kaufman and Oppenheim [KO20] proved that if the simplicial complex is a good local-spectral expander, then the up-down and down-up walks mix rapidly. Recall that we used a slightly different definition of a local spectral expander in Definition 26.13, where we considered the spectral gap instead of the second largest eigenvalue of the random walk matrix. Thus, the following statement differs slightly from those in the literature.

Theorem 27.9 (Kaufman-Oppenheim Spectral Gap Bound [KO20])**.** *Let (X, Π) be a g-local-spectral expander in Definition 26.13. Let $P_k^{\triangledown}$ be the down-up walk on $X(k)$ and $\tilde{\mathcal{L}}_k^{\triangledown} := I - P_k^{\triangledown}$ be its normalized Laplacian matrix. For any $0 \leq k \leq d$, the second smallest eigenvalue of $\tilde{\mathcal{L}}_k^{\triangledown}$ satisfies*

$$\lambda_2(\tilde{\mathcal{L}}_k^{\triangledown}) \geq \frac{1}{k+1} - k(1-g).$$

This theorem implies that if the simplicial complex is a 1-local-spectral expander, then the down-up walk $P_k^{\triangledown}$ has spectral gap at least $\frac{1}{k+1}$, leading to the resolution of the matroid expansion conjecture.

Matroid Expansion Conjecture

Recall from Theorem 26.18 that the matroid complex is a 1-local-spectral expander. By Kaufman-Oppenheim's Theorem 27.9,

$$\lambda_2(\tilde{\mathcal{L}}_{r-1}^{\triangledown}) \geq \frac{1}{r}, \tag{27.1}$$

where $r := d+1$ is the rank of the matroid. By the standard mixing time analysis in Theorem 3.15, the ϵ-mixing time of the down-up walks is at most $O(r \log \frac{N}{\epsilon}) = O(r^2 \log \frac{n}{\epsilon})$, where $N \leq n^r$ is the number of bases and n is the number of elements in the ground set. This provides a simple and efficient algorithm to sample a uniform matroid basis.

Theorem 27.10 (Sampling Matroid Bases by Down-Up Walks [ALOV19]). *For any matroid M or rank r with n elements, the mixing time of the down-up walk is at most*

$$O\Big(r^2 \log \frac{n}{\epsilon}\Big).$$

The matroid expansion conjecture, proposed by Mihail and Vazirani in 1989, states that the basis exchange graph (the underlying unweighted graph of the down-up walk matrix of the matroid complex) has edge expansion at least one. Their motivation was to obtain an efficient sampling algorithm for matroid bases, as in Theorem 27.10.

The conjecture follows from (27.1) and Cheeger's inequality in Theorem 2.2.

Corollary 27.11 (Proof of the Matroid Expansion Conjecture [ALOV19]). *The basis exchange graph $G = (V, E)$ has edge expansion at least one, i.e., $|\delta(S)|/|S| \geq 1$ for all $S \subseteq V$ with $|S| \leq |V|/2$.*

27.3 Spectral Gap Bound in Product Form

The techniques from high-dimensional expanders have led to a breakthrough in analyzing mixing times of Markov chains. This motivates the goal of obtaining sharper bounds on the spectral gap of higher-order random walks for other simplicial complexes.

The following product form ensures a positive spectral gap bound as long as all links are connected.

Theorem 27.12 (Spectral Gap in Product Form [AL20]). *Let (X, Π) be a $(g_{-1}, \ldots, g_{d-2})$-local-spectral expander in Definition 26.13. Let $P_k^{\triangledown}$ be the down-up walk on $X(k)$, and let $\tilde{\mathcal{L}}_k^{\triangledown} := I - P_k^{\triangledown}$ be its normalized Laplacian matrix. For any $0 \leq k \leq d$, the second smallest eigenvalue of $\tilde{\mathcal{L}}_k^{\triangledown}$ satisfies*

$$\lambda_2(\tilde{\mathcal{L}}_k^{\triangledown}) \geq \frac{1}{k+1} \prod_{j=-1}^{k-2} g_j.$$

Combining this with Oppenheim's Theorem 26.16, it follows that if the graphs of the top links have spectral gap at least $1 - \frac{1}{d+1}$, then the down-up walk remains rapidly mixing.

Corollary 27.13 (Top Links). *Let (X,Π) be a pure d-dimensional simplicial complex. Suppose $g_{d-2} \geq 1 - \frac{1}{d+1}$ and G_σ is connected for every face σ up to dimension $d-2$, then*

$$\lambda_2(\tilde{\mathcal{L}}_d^{\triangledown}) \geq \frac{1}{(d+1)^2}.$$

Another consequence is that the following spectral gap profile ensures polynomial mixing time.

Corollary 27.14 (Improving Profile from Top to Bottom). *Let (X,Π) be a pure d-dimensional simplicial complex. If there exists a constant $0 < c < 1$ such that*

$$(g_{d-2}, g_{d-3}, \ldots, g_0, g_{-1}) = \Big(1-c, 1-\frac{c}{2}, \ldots, 1-\frac{c}{d-1}, 1-\frac{c}{d}\Big),$$

then

$$\lambda_2(\tilde{\mathcal{L}}_d^{\triangledown}) \geq \frac{1}{d^{1+c}}.$$

This profile turns out to be very useful in analyzing mixing times as we will see in the next chapter.

Local-to-Global Method

We follow the exposition by Mohanty [Moh22] to prove Theorem 27.12.

Similar to the discussion in Section 26.4 for the trickling down theorem, the usual proofs of the random walk theorems (Theorem 27.9 and Theorem 27.12) also rely on the Rayleigh quotient characterization of the random walk matrix. This approach involves decomposing $P_k^{\triangle}$ into the random walk matrices W_τ of its $(k-1)$-dimensional links, decomposing $P_k^{\triangledown}$ into "complete graphs" J_τ of its $(k-1)$-dimensional links, and then comparing W_τ and J_τ term by term. In this comparison, a function f on $X(k)$ is decomposed into f_τ on its $(k-1)$-dimensional links, which is further split into $f_\tau = f_\tau^{\parallel} + f_\tau^{\perp}$ where $\langle f_\tau^{\perp}, \vec{1}\rangle_{\Pi_0^\tau} = 0$, to bound the quadratic form $\langle f_\tau^{\perp}, (W_\tau - J_\tau) f_\tau^{\perp}\rangle$. The improvement in Theorem 27.12 is obtained by recovering certain dropped terms involving $f_\tau^{\parallel}$ in the analysis of Theorem 27.9.

As in Section 26.4, the unconstrained characterization in Lemma 26.20 is used to bypass the decomposition of f. Moreover, the denominator in this characterization fits perfectly in the analysis of higher-order random walks. As a result, the calculations are significantly simplified.

Proof of Theorem 27.12. We prove by induction that

$$\lambda_2(\tilde{\mathcal{L}}_k^{\triangle}) = \lambda_2(\tilde{\mathcal{L}}_{k+1}^{\triangledown}) \geq \frac{1}{k+2} \prod_{j=-1}^{k-1} g_j,$$

where the equality follows from Fact 27.6. The base case when $k = 0$ is clear.

For the induction step, we first outline the main steps leading to the conclusion, followed by detailed explanations. Readers may find it helpful to read each step, then refer to the explanations before proceeding further.

Let f be any function satisfying $\mathbb{E}_{\sigma\sim\Pi_k,\sigma'\sim\Pi_k}\left[\left(f(\sigma)-f(\sigma')\right)^2\right]=1$. We claim that

$$\begin{aligned}
&\mathbb{E}_{\sigma\sim\Pi_k,\sigma'\sim P_k^{\triangle}(\sigma,\cdot)}\left[\left(f(\sigma)-f(\sigma')\right)^2\right] \\
=\;& \mathbb{E}_{\tau\sim\Pi_{k-1}}\mathbb{E}_{i\sim\Pi_0^\tau,j\sim\frac{k+1}{k+2}W_\tau(i,\cdot)+\frac{1}{k+2}\chi_i}\left[\left(f(\tau\cup\{i\})-f(\tau\cup\{j\})\right)^2\right] && (27.2)\\
\geq\;& \frac{k+1}{k+2}\cdot g_{k-1}\cdot\mathbb{E}_{\tau\sim\Pi_{k-1}}\mathbb{E}_{i\sim\Pi_0^\tau,j\sim\Pi_0^\tau}\left[\left(f(\tau\cup\{i\})-f(\tau\cup\{j\})\right)^2\right] && (27.3)\\
=\;& \frac{k+1}{k+2}\cdot g_{k-1}\cdot\mathbb{E}_{\sigma\in\Pi_k,\sigma'\sim P_k^{\triangledown}(\sigma,\cdot)}\left[\left(f(\sigma)-f(\sigma')\right)^2\right] && (27.4)\\
\geq\;& \frac{k+1}{k+2}\cdot g_{k-1}\cdot\frac{1}{k+1}\prod_{j=-1}^{k-2}g_j && (27.5)\\
=\;& \frac{1}{k+2}\prod_{j=-1}^{k-1}g_j.
\end{aligned}$$

Since this holds for any function f satisfying $\mathbb{E}_{\sigma\sim\Pi_k,\sigma'\sim\Pi_k}[(f(\sigma)-f(\sigma'))^2]=1$, we conclude from Lemma 26.20 that $\lambda_2(\tilde{\mathcal{L}}_k^{\triangle})\geq\prod_{j=-1}^{k-1}g_j/(k+2)$. It remains to justify each step.

Explanation of (27.2): Each pair σ,σ' contributing a nonzero term in the first line is of the form $\sigma=\tau\cup\{i\}$, $\sigma'=\tau\cup\{j\}$, where $\tau=\sigma\cap\sigma'\in X(k-1)$ and $i\neq j$. The key observation is that sampling such a pair from this distribution is equivalent to sampling τ from Π_{k-1}, then sampling a random edge ij using the random walk matrix W_τ of the link X_τ. The only difference is that $P_k^{\triangle}$ includes a self-loop probability of $\frac{1}{k+2}$, while there is no self-loop in W_τ. This is the reason for the adjustment factor $(k+1)/(k+2)$ in (27.2).

To verify formally, the weight of a nonzero term $(f(\sigma)-f(\sigma'))^2$ on (27.2) is

$$2\Pi(\tau)\Pi_0^\tau(i)\frac{k+1}{k+2}W_\tau(i,j)=2\Pi(\tau)\frac{\Pi(\tau\cup\{i\})}{(|\tau|+1)\Pi(\tau)}\frac{k+1}{k+2}\frac{\Pi(\tau\cup\{i,j\})}{(|\tau|+2)\Pi(\tau\cup\{i\})}=2\frac{\Pi(\sigma\cup\sigma')}{(k+2)^2}$$

where we used (26.4), (26.2), and (26.3). This matches the corresponding weight on the left-hand side, which is $2\Pi(\sigma)P_k^{\triangle}(\sigma,\sigma')=2\Pi(\sigma\cup\sigma)/(k+2)^2$ using Fact 27.7.

Explanation of (27.3): Applying the local-to-global characterization from Lemma 26.20 to each inner expectation in (27.2) yields (27.3).

Explanation of (27.4): Again, each pair σ,σ' contributing a non-zero term in (27.4) is of the form $\sigma=\tau\cup\{i\}$, $\sigma'=\tau\cup\{j\}$, where $\tau=\sigma\cap\sigma'\in X(k-1)$ and $i\neq j$. The weight of this term $(f(\sigma)-f(\sigma'))^2$ in (27.3) is

$$2\Pi(\tau)\Pi_0^\tau(i)\Pi_0^\tau(j)=2\Pi(\tau)\frac{\Pi(\tau\cup\{i\})}{(|\tau|+1)\Pi(\tau)}\frac{\Pi(\tau\cup\{j\})}{(|\tau|+1)\Pi(\tau)}=2\frac{\Pi(\sigma)\Pi(\sigma')}{(k+1)^2\cdot\Pi(\sigma\cap\sigma')}$$

where we used (26.2). This matches the corresponding weight on (27.3), which is $2\Pi(\sigma)P_k^{\triangledown}(\sigma,\sigma')=2\Pi(\sigma)\Pi(\sigma')/(k+1)^2\Pi(\sigma\cap\sigma')$ using Fact 27.7.

Explanation of (27.5): Since $\mathbb{E}_{\sigma\sim\Pi_k,\sigma'\sim\Pi_k}[(f(\sigma)-f(\sigma'))^2]=1$, it follows from Lemma 26.20 that

$$\mathbb{E}_{\sigma\in\Pi_k,\sigma'\sim P_k^{\triangledown}(\sigma,\cdot)}[(f(\sigma)-f(\sigma'))^2]\geq\lambda_2(\tilde{\mathcal{L}}_k^{\triangledown})=\lambda_2(\tilde{\mathcal{L}}_{k-1}^{\triangle})\geq\frac{1}{k+1}\prod_{j=-1}^{k-2}g_j,$$

where the equality is from Fact 27.6 and the last inequality is by the induction hypothesis. □

Combinatorial Interpretation

The key step of the proof is to compare the up-down walk $P_k^{\triangle}$ with the down-up walk $P_k^{\triangledown}$. The idea is to decompose $P_k^{\triangle}$ into the random walk matrix W_τ of its links X_τ, which measures the average squared edge length in (27.2), and to decompose $P_k^{\triangledown}$ into the "complete graph" matrix of its links X_τ, which measures the average squared pairwise difference in (27.4). Then Lemma 26.20 is used to compare the random walk matrix with the complete graph matrix through the spectral gap g_{k-1}.

If we interpret the proof combinatorially using edge conductance, then (27.2) decomposes the up-down walk graph into expanders, (27.4) decomposes the down-up walk graph into complete graphs, and replacing each complete graph with an expander preserves the edge conductance within a factor g_{k-1} in (27.3).

A subtle but important step is that $P_k^{\triangledown}$ and $P_{k-1}^{\triangle}$ have the same spectrum, which enables the inductive approach. One could visualize the overall proof as having a stack of bipartite graphs, one for each layer as described in Definition 27.1. We build the down-up walk $P_k^{\triangledown}$ graph inductively starting from $P_0^{\triangle} = \frac{1}{2}(W_\emptyset + I)$, relating the spectrum of $P_k^{\triangle}$ to $P_{k+1}^{\triangledown}$ through the bipartite graph in the k-th layer, and replacing each clique in $P_{k+1}^{\triangledown}$ with the link graphs W_σ for $\sigma \in X(k)$ to obtain $P_{k+1}^{\triangle}$, until we reach $P_k^{\triangledown}$.

One may understand the resolution of the matroid expansion conjecture as using the right induction for the problem, which may not be easy to come up with without the perspective of a simplicial complex and the concepts such as links. It would be interesting to see whether there exists a purely combinatorial proof (without using any linear algebra) of the matroid expansion conjecture using the combinatorial interpretation above.

Question 27.15 (Combinatorial Proof of Matroid Expansion Conjecture)**.** *Is there a purely combinatorial proof of the matroid expansion conjecture?*

References

[AL20] Vedat Levi Alev and Lap Chi Lau. Improved analysis of higher order random walks and applications. In *Proccedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, STOC*, pages 1198–1211. ACM, 2020. 9, 344

[ALOV19] Nima Anari, Kuikui Liu, Shayan Oveis Gharan, and Cynthia Vinzant. Log-concave polynomials II: high-dimensional walks and an FPRAS for counting bases of a matroid. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, STOC*, pages 1–12. ACM, 2019. 9, 329, 334, 344

[KM17] Tali Kaufman and David Mass. High dimensional random walks and colorful expansion. In *8th Innovations in Theoretical Computer Science Conference, ITCS*, volume 67, pages 4:1–4:27, 2017. 341, 342

[KO20] Tali Kaufman and Izhar Oppenheim. High order random walks: Beyond spectral gap. *Comb.*, 40(2):245–281, 2020. 9, 331, 333, 343

[Moh22] Sidhanth Mohanty. Local-to-global theorems for high-dimensional expansion. 2022. https://sidhanthm.com/pdf/local-to-global.pdf. 333, 335, 337, 345

Chapter 28

Spectral and Entropic Independence

The results from high-dimensional expanders in previous chapters provide a new approach to directly bounding the spectral gap of down-up walks. In this chapter, we introduce the notion of spectral independence, an elegant probabilistic reformulation of the results without the language of high-dimensional expanders. This framework has led to recent advances in analyzing the mixing times of Markov chains.

This approach is further developed into the notion of entropic independence, which directly bounds the entropy decay and the log-Sobolev constant of down-up walks. An interesting consequence is a near-linear time algorithm for sampling a random spanning tree.

28.1 Spectral Independence

Anari, Liu and Oveis Gharan [ALO20] introduced the notion of spectral independence, a probabilistic reformulation of Theorem 27.12 for sampling from a distribution, without the language of high-dimensional expanders.

The following correlation matrix captures the pairwise correlation between elements, analogous to the random walk matrix of the empty link of the corresponding simplicial complex.

Definition 28.1 (Correlation Matrix)**.** *Let $\mu : \{0,1\}^n \to \mathbb{R}$ be a probability distribution. The correlation matrix of μ is a $2n \times 2n$ matrix Ψ, whose rows and columns are indexed by $[n] \times \{0,1\}$, with*

$$\Psi\big((i,a_i),(j,a_j)\big) = \Pr_{Z\sim\mu}\big[Z(j) = a_j \mid Z(i) = a_i\big] - \Pr_{Z\sim\mu}\big[Z(j) = a_j\big]$$

for any $i \neq j$ and $a_i, a_j \in \{0,1\}$, while $\Psi\big((i,a_i),(j,a_j)\big) = 0$ if $i = j$.

Remark 28.2 (Influence Matrix)**.** *The following $n \times n$ influence matrix Φ was defined in [ALO20]. For any i, j,*

$$\Phi(i,j) = \Pr_{Z\sim\mu}\big[Z(j) = 1 \mid Z(i) = 1\big] - \Pr_{Z\sim\mu}\big[Z(j) = 1 \mid Z(i) = 0\big].$$

The influence matrix and the correlation matrix are closely related, with the same nonzero eigenvalues; see Problem 28.34. The correlation matrix, stated in [AASV21, CGSV21], is more directly connected to the corresponding simplicial complex.

The following conditional correlation matrices are the correlation matrices given a partial assignment, analogous to the random walk matrices of the links in the corresponding simplicial complex.

Definition 28.3 (Conditional Correlation Matrices)**.** *Let $\mu : \{0,1\}^n \to \mathbb{R}$ be a probability distribution. Let $S \subseteq [n]$ be a subset, and let $a_S \in \{0,1\}^{|S|}$ be a binary string specifying an assignment for each element $i \in S$. Define the event $Z(S) = a_S$ as $Z(i) = a_S(i)$ for all $i \in S$, when $Z \sim \mu$.*

The conditional correlation matrix Ψ_{a_S} is a $2(n-|S|) \times 2(n-|S|)$ matrix, whose rows and columns are indexed by $([n] \setminus S) \times \{0,1\}$, with

$$\Psi_{a_S}\big((i,a_i),(j,a_j)\big) = \Pr_{Z\sim\mu}\big[Z(j) = a_j \mid Z(i) = a_i, Z(S) = a_S\big] - \Pr_{Z\sim\mu}\big[Z(j) = a_j \mid Z(S) = a_S\big]$$

for any $i \neq j$ and $a_i, a_j \in \{0,1\}$, while $\Psi\big((i,a_i),(j,a_j)\big) = 0$ if $i = j$.

The definition of spectral independence is closely related to the local-spectral expansion of the corresponding simplicial complex of the probability distribution.

Definition 28.4 (Spectral Independence)**.** *A probability distribution $\mu : \{0,1\}^n \to \mathbb{R}$ is called ψ-spectrally independent if, for any $S \subseteq [n]$ with $|S| \leq n-2$ and any partial assignment $a_S \in \{0,1\}^{|S|}$,*

$$\lambda_{\max}\big(\Psi_{a_S}\big) \leq \psi.$$

Let's consider some examples before proceeding. First, it is easy to see that if μ is a product distribution (i.e., there exist $\lambda_1, \ldots, \lambda_n$ such that $\mu(S) \propto \sum_{i\in S} \lambda_i$), then μ is 0-spectrally independent. This suggests that spectral independence provides a linear algebraic way to quantity the independence of a probability distribution.

A more interesting example is the class of negatively correlated distributions (see Problem 14.25).

Claim 28.5 (Spectral Independence of Negatively Correlated Distributions)**.** *Let $\mu : \{0,1\}^n$ be a homogeneous distribution such that for all $i \neq j$,*

$$\Pr_{Z\sim\mu}\big[Z(i) = 1 \mid Z(j) = 1\big] \leq \Pr_{Z\sim\mu}\big[Z(i) = 1\big].$$

Then, $\lambda_{\max}(\Psi) \leq 1$, where Ψ is the correlation matrix of μ.

Glauber Dynamics

A natural random walk on a probability distribution $\mu : \{0,1\}^n \to \mathbb{R}$ is called Glauber dynamics.

Definition 28.6 (Glauber Dynamics)**.** *Let $\mu : \{0,1\}^n \to \mathbb{R}$ be a probability distribution on subsets of $[n]$. Start with an arbitrary subset $S_0 \in \operatorname{supp}(\mu)$. At each iteration $t \geq 1$, choose a uniformly random element $i \in [n]$ and update S_t as follows:*

$$S_t := \begin{cases} S_{t-1} \setminus \{i\} & \text{with probability } \frac{\mu\left(S_{t-1}\setminus\{i\}\right)}{\mu\left(S_{t-1}\setminus\{i\}\right)+\mu\left(S_{t-1}\cup\{i\}\right)} \\ S_{t-1} \cup \{i\} & \text{otherwise.} \end{cases}$$

Verify that this Markov chain has stationary distribution μ.

The main result of this formulation is a bound on the spectral gap of the transition matrix of the Glauber dynamics in terms of the spectral independence of the probability distribution.

Theorem 28.7 (Spectral Gap via Spectral Independence [ALO20]). *Let $\mu : \{0,1\}^n \to \mathbb{R}$ be a probability distribution that is ψ-spectrally independent. The random walk matrix of the Glauber dynamics for μ has a spectral gap of at least*

$$\frac{1}{n}\prod_{i=0}^{n-2}\left(1-\frac{\psi}{n-i-1}\right).$$

Consequently, the mixing time of the Glauber dynamics is at most $\tilde{O}(n^{2+\psi})$.

28.2 Simplicial Complex for Glauber Dynamics

The proof of Theorem 28.7 consists of the following steps:

1. Define a simplicial complex X^μ for the probability distribution $\mu : \{0,1\}^n \to \mathbb{R}$.
2. Show that the down-up walk $P^{\triangledown}_{n-1}$ of X^μ corresponds to the Glauber dynamics.
3. Bound the second largest eigenvalue of the random walk matrix W_τ of the links by the maximum eigenvalue of the conditional correlation matrices of μ.
4. Apply Theorem 27.12 for down-up walks to derive Theorem 28.7 for Glauber dynamics.

The following is a natural simplicial complex of assignments defined on the label extended graph.

Definition 28.8 (Simplicial Complex of Assignments). *Let $\mu : \{0,1\}^n \to \mathbb{R}$ be a probability distribution on subsets of $[n]$. The simplicial complex $X^\mu = ([n] \times \{0,1\}, \Pi)$ is defined with ground set $[n] \times \{0,1\}$, and a maximal face*

$$\zeta := \big((1, Z(1)), (2, Z(2)), \ldots, (n, Z(n))\big)$$

of dimension $n-1$, with weight $\Pi(\zeta) := \mu(Z)$ for each $Z \in \mathrm{supp}(\mu)$. In other words, each maximal face of X^μ corresponds to an assignment of the n binary variables with nonzero probability in μ.

Note that each face of X^μ corresponds to a partial assignment $a_S \in \{0,1\}^{|S|}$ on a subset $S \subseteq [n]$ of binary variables. Hence, we denote the links of X^μ by $X^\mu_{a_S}$ for $S \subseteq [n]$ and for $a_S \in \{0,1\}^{|S|}$.

It is easy to check that each step of the down-up walk $P^{\triangledown}_{n-1}$ on X^μ in Definition 27.5 (and Fact 27.7) corresponds to flipping a random coordinate in the current assignment, following the same distribution as the Glauber dynamics in Definition 28.6.

Claim 28.9 (Glauber Dynamics and Down-Up Walks). *The down-up walk matrix $P^{\triangledown}_{n-1}$ on X^μ is exactly the transition matrix of the Glauber dynamics on μ in Definition 28.6.*

Next, we establish the correspondence between the conditional correlation matrices of μ and the random walk matrices of links of X^μ. The key observation is that the matrix Ψ_{a_S} arises from the inequality in (27.3), which compares the random walk matrix W and the "complete walk matrix" J of a link, using the local-to-global characterization of normalized Laplacian eigenvalue in Lemma 26.20.

Lemma 28.10 (Conditional Correlation Matrices and Random Walk Matrices of Links)**.** *Let $\mu : \{0,1\}^n \to \mathbb{R}$ be a probability distribution, and let X^μ be the simplicial complex from Definition 28.8. For a partial assignment $a_S \in \{0,1\}^{|S|}$ on a subset $S \subseteq [n]$,*

$$\alpha_2(W_{a_S}) \le \frac{\lambda_{\max}(\Psi_{a_S})}{n - |S| - 1},$$

where W_{a_S} is the random walk matrix of the link $X^\mu_{a_S}$ in Definition 26.12, and Ψ_{a_S} is the conditional correlation matrix from Definition 28.3.

Proof. Let $J_{a_S} := \vec{1}(\Pi_0^{a_S})^\top$, where $\Pi_0^{a_S}$ is the stationary distribution of the random walk matrix W_{a_S}. We claim that, for any $i \neq j$ and any $a_i, a_j \in \{0,1\}$,

$$\Psi_{a_S}((i,a_i),(j,a_j)) = (n - |S| - 1) \cdot W_{a_S}((i,a_i),(j,a_j)) - (n - |S|) \cdot J_{a_S}((i,a_i),(j,a_j)). \tag{28.1}$$

Define $a_{S\cup\{i\}}$ as the partial assignment on $S \cup \{i\}$ extending a_S with the i-th variable assigned a_i, and define $a_{S\cup\{i,j\}}$ analogously. By Definition 26.12 and (26.1),

$$\begin{aligned} W_{a_S}\big((i,a_i),(j,a_j)\big) &= \frac{\Pi(a_{S\cup\{i,j\}})}{(|S|+2)\cdot\Pi(a_{S\cup\{i\}})} \\ &= \frac{\binom{n}{|S|+2}^{-1}\cdot \Pr_{Z\sim\mu}\left[Z\big(S\cup\{i,j\}\big) = a_{S\cup\{i,j\}}\right]}{(|S|+2)\cdot\binom{n}{|S|+1}^{-1}\cdot \Pr_{Z\sim\mu}\left[Z\big(S\cup\{i\}\big) = a_{S\cup\{i\}}\right]} \\ &= \frac{1}{n-|S|-1}\cdot \Pr_{Z\sim\mu}\left[Z(j) = a_j \mid Z(i) = a_i, Z(S) = a_S\right]. \end{aligned}$$

By (26.2) and a similar calculation,

$$J_{a_S}\big((i,a_i),(j,a_j)\big) = \Pi_0^{a_S}((j,a_j)) = \frac{\Pi(a_{S\cup\{j\}})}{(|S|+1)\cdot\Pi(a_S)} = \frac{1}{n-|S|}\cdot \Pr_{Z\sim\mu}\left[Z(j) = a_j \mid Z(S) = a_S\right].$$

It follows from the definition of Ψ_{a_S} (Definition 28.3) that (28.1) holds.

Since $W_{a_S}((i,a_i),(i,a_i')) = 0$ and $\Psi((i,a_i),(i,a_i')) = 0$ for all i and all a_i, a_i', (28.1) implies that

$$\Psi_{a_S} = (n - |S| - 1)\cdot W_{a_S} - (n - S)\cdot J_{a_S} + B_{a_S},$$

where B_{a_S} is a block-diagonal matrix with n blocks of 2×2 matrices, each equal to the corresponding block in $(n-S)\cdot J_{a_S}$. Using the characterization of $\alpha_2(W_{a_S})$ in (26.5), for any optimizer f in (26.5) satisfying $\langle f, f\rangle_{\Pi_0^{a_S}} = 1$ and $\langle f, 1\rangle_{\Pi_0^{a_S}} = 0$,

$$\begin{aligned} \lambda_{\max}(\Psi_{a_S}) \ge \langle f, \Psi_{a_S} f\rangle_{\Pi_0^{a_S}} &= \Big\langle f, \Big((n-|S|-1)\cdot W_{a_S} - (n-S)\cdot J_{a_S} + B_{a_S}\Big) f\Big\rangle_{\Pi_0^{a_S}} \\ &= (n-|S|-1)\cdot\alpha_2(W_{a_S}) + \langle f, B_{a_S} f\rangle_{\Pi_0^{a_S}} \ge (n-|S|-1)\cdot\alpha_2(W_{a_S}), \end{aligned}$$

where the first inequality follows from $\Pi_0^{a_S}\Psi_{a_S}$ being a symmetric matrix and Lemma A.12, and the second line follows from $J_{a_S} f = 0$ and $\Pi_0^{a_S} B_{a_S}$ being positive semidefinite. □

Now we are ready to apply Theorem 27.12 to complete the proof of Theorem 28.7.

Proof of Theorem 28.7. Let $\tilde{\mathcal{L}}_{a_S} := I - W_{a_S}$ be the normalized Laplacian matrix of the link X_{a_S}. By Lemma 28.10, we have

$$\lambda_2\big(\tilde{\mathcal{L}}_{a_S}\big) = 1 - \alpha_2(W_{a_S}) \geq 1 - \frac{\psi}{n - |S| - 1} \quad \Longrightarrow \quad g_j \geq 1 - \frac{\psi}{n - j - 2}.$$

Thus, the spectral gap bound follows directly from Theorem 27.12. The mixing time bound follows from Corollary 27.14 and the standard mixing time analysis in Theorem 3.15. □

28.3 Applications of Spectral Independence

Spectral independence has become a powerful tool for analyzing the mixing times of Markov chains. We briefly discuss some recent developments and point to the relevant references.

Sampling Independent Sets from Hardcore Distributions

The first major application of the spectral independence formulation is to prove fast mixing for sampling independent sets from the hardcore distribution.

Definition 28.11 (Hardcore Distributions)**.** *Given a graph $G = (V, E)$ and a parameter $\lambda > 0$, define the hardcore distribution $\mu_\lambda : \{0,1\}^{|V|} \to \mathbb{R}$ as*

$$\mu_\lambda(S) = \frac{\lambda^{|S|}}{Z_G(\lambda)}$$

for each independent set $S \subseteq V$, where $Z_G(\lambda) := \sum_{S \subseteq V : S \text{ is an independent set}} \lambda^{|S|}$ is the normalization constant called the partition function.

Estimating the partition function is a well-studied problem in statistical physics. For a graph of maximum degree Δ, there is a critical threshold $\lambda(\Delta) = (\Delta - 1)^{\Delta - 1}/(\Delta - 2)^{\Delta}$ called the tree uniqueness threshold, where $\lambda < \lambda(\Delta)$ corresponds to the regime where the influence of a vertex i on another vertex j in the infinite Δ-regular tree decays exponentially fast in the distance between i and j, while $\lambda > \lambda(\Delta)$ corresponds to the regime where long-range dependencies may appear.

The tree uniqueness threshold marks the phase transition of a mathematical property, but very interestingly, this coincides with the phase transition of the computational complexity. On one hand, Weitz [Wei06] showed that for any $\lambda < \lambda(\Delta)$, there is a deterministic fully polynomial time approximate scheme to estimate $Z_G(\lambda)$. On the other hand, Sly [Sly10] proved that for any $\lambda > \lambda(\Delta)$, there is no such scheme to estimate $Z_G(\lambda)$ unless NP = RP. Both proofs connect explicitly the mathematical property to the computational complexity.

It was conjectured that the simple Glauber dynamics in Definition 28.6 for the hardcore distributions mixes in polynomial time whenever $\lambda < \lambda(\Delta)$. Anari, Liu and Oveis Gharan [ALO20] introduced spectral independence and used this notion to resolve the conjecture positively. Their proof uses the self-avoiding walk tree defined by Weitz to write a recurrence, bounding the maximum row sum of the correlation matrices Ψ in Definition 28.1. This gives a bound on the maximum eigenvalue $\lambda_{\max}(\Psi)$, which suffices to apply Theorem 28.7 to conclude fast mixing.

This work has been significantly extended to establish an $O(n \log n)$ mixing time of the Glauber dynamics below the uniqueness threshold [CFYZ22], and polynomial mixing time at the uniqueness threshold [CCYZ24], thus completely resolving the problem.

Sampling Graph Coloring Using Glauber Dynamics

One natural generalization of spectral independence is to sample from distributions $\mu : [k]^n \to \mathbb{R}$ where the variables have larger arity. This class of distributions includes the problem of sampling a random vertex coloring of a graph.

The long-standing open problem for sampling graph coloring is whether the simple Glauber dynamics in Definition 28.6 mixes rapidly as long as the number of colors k is at least $\Delta + 2$, where Δ is the maximum degree of the input graph. Note that the Glauber dynamics may not be irreducible when $k \leq \Delta + 1$. The best known result, which essentially dates back to Vigoda [Vig99], shows that the Glauber dynamic mixes in polynomial time as long as $k \geq \frac{11}{6}\Delta$, recently improved to 1.809Δ in [CV25]. Thus, a significant gap remains between the upper bound and the lower bound.

The random graph coloring problem is very well-studied, with most previous results relying on coupling techniques to prove fast mixing. Using spectral independence with correlation decay arguments, previous results can be recovered with improved running time [CGSV21, CLV21]. Recently, significant improvements have been made for edge coloring using $(1+o(1))\Delta$ colors [ALO22, WZZ24] and for vertex coloring in large constant-girth graphs using $\Delta + 3$ colors [CLMM23]. Some of these results also rely on log-Sobolev inequalities and entropy techniques, which we will discuss in the next sections.

Spectral Independence as a Unifying Framework

A general question is how the spectral independence method relates to other methods for proving fast mixing, such as the widely-used coupling techniques. Recent works have shown that certain types of coupling methods [Liu21, BCC+22] and the polynomial interpolation method [CLV24] also imply spectral independence. Furthermore, [AJK+24] proved that if the down-up walk P_k^{∇} for a distribution μ has spectral gap $1/(ck)$, then μ is c-spectrally independent. This implies that any method that establishes a spectral gap also implies spectral independence, suggesting the spectral independence method is a universal framework for analyzing mixing times of Glauber dynamics.

28.4 Analyzing Mixing Time Using Entropy

In this section, we define relative entropy and log-Sobolev constants, discuss their roles in bounding mixing times, and provide some intuition behind these definitions.

Variance and Entropy

In Chapter 3, when we analyze the mixing time of random walks, we upper bound the total variation distance $d_{\mathrm{TV}}(p_t, \pi)$ by upper bounding $\|p_t - \pi\|_{D^{-1}}$ (see (3.1) and (3.2)). This can be understood as upper bounding the variance of $f := p_t/\pi$, the density of p_t with respect to π at time $t \geq 0$.

Definition 28.12 (π-Variance). *Let $f : [n] \to \mathbb{R}$ be a function and π be a probability distribution on $[n]$. The variance of f with respect to π is defined as*

$$\mathrm{Var}_\pi[f] := \mathbb{E}_\pi[f^2] - \big(\mathbb{E}_\pi[f]\big)^2,$$

where $\mathbb{E}_\pi[f] = \sum_{i\in[n]} \pi(i) f(i)$.

There are other ways to measure the closeness of two probability distributions. A well-known measure is the relative entropy between the two distributions.

Definition 28.13 (KL-Divergence)**.** *Let p and q be probability distributions on $[n]$ such that $q(i) = 0$ implies $p(i) = 0$ for $1 \le i \le n$. The Kullback-Liebler divergence, or relative entropy, between p and q is defined as*

$$\mathcal{D}_{\mathrm{KL}}(p \parallel q) = \sum_{i=1}^{n} p(i) \cdot \log \frac{p(i)}{q(i)},$$

where we follow the convention that $0 \log 0 = 0$. Check that $\mathcal{D}_{\mathrm{KL}}(p, q) \ge 0$ by Jensen's inequality.

Pinsker's inequality upper bounds the total variation distance by the KL-divergence.

Theorem 28.14 (Pinsker's Inequality)**.** *For any two probability distributions p, q on $[n]$,*

$$d_{\mathrm{TV}}(p, q)^2 \le 2\mathcal{D}_{\mathrm{KL}}(p \parallel q).$$

To analyze mixing time, it is also natural to consider the relative entropy betwen p_t and π.

Definition 28.15 (π-Entropy)**.** *Let $f : [n] \to \mathbb{R}$ be a function and π be a probability distribution on $[n]$. Define*

$$\mathrm{Ent}_\pi[f] := \mathbb{E}_\pi[f \log f] - \mathbb{E}_\pi\big[f \log(\mathbb{E}_\pi[f])\big].$$

Check that $\mathrm{Ent}_\pi[\frac{p}{\pi}] = \mathcal{D}_{\mathrm{KL}}(p \parallel \pi)$ for a probability distribution p.

Spectral Gap and Log-Sobolev Constants

The following definition in the mixing time literature corresponds to the quadratic form of the normalized Laplacian matrix $\langle f, \tilde{\mathcal{L}} f \rangle_\pi$ in Lemma 26.19.

Definition 28.16 (Dirichlet Form)**.** *Let $P \in \mathbb{R}^{n \times n}$ be the transition matrix of a reversible Markov chain whose stationary distribution is π. For two vectors $f, g \in \mathbb{R}^n$, the Dirichlet form is defined as*

$$\mathcal{E}_P(f, g) := \langle (I - P)f, g \rangle_\pi = \frac{1}{2} \sum_{1 \le i,j \le n} \pi(i) \cdot P(i, j) \cdot \big(g(i) - g(j)\big) \cdot \big(f(i) - f(j)\big).$$

The Dirichlet form is often called the energy of the function f, which measures the local variation of f along the edges of the underlying graph. In contrast, the variance in Definition 28.12 measures the global variation of f. The following characterization is equivalent to the local-to-global characterization of the spectral gap in Lemma 26.20.

Definition 28.17 (Variational Characterization of Spectral Gap)**.** *Let $P \in \mathbb{R}^{n \times n}$ be the transition matrix of a reversible Markov chain whose stationary distribution is π. Define the spectral gap as*

$$\lambda(P) := \inf \left\{ \frac{\mathcal{E}_P(f, f)}{\mathrm{Var}_\pi(f)} \;\middle|\; f : [n] \to \mathbb{R},\ \mathrm{Var}_\pi[f] \ne 0 \right\}.$$

We note that the spectral gap is often called the Poincaré constant in the mixing time literature. The log-Sobolev constant replaces the variance of f in the denominator of the spectral gap by the π-entropy of f in Definition 28.15.

Definition 28.18 (Log-Sobolev Constant [DSC96]). *Let $P \in \mathbb{R}^{n\times n}$ be the transition matrix of a reversible Markov chain whose stationary distribution is π. Define the log-Sobolev constant of P as*

$$\gamma(P) := \inf\left\{ \frac{\mathcal{E}_P(f,f)}{\mathrm{Ent}_\pi(f^2)} \;\middle|\; f : [n] \to \mathbb{R}_{\geq 0},\ \mathrm{Ent}_\pi[f^2] \neq 0 \right\}.$$

The modified log-Sobolev constant was introduced by Bobkov and Tetali.

Definition 28.19 (Modified Log-Sobolev Constant [BT06]). *Let $P \in \mathbb{R}^{n\times n}$ be the transition matrix of a reversible Markov chain whose stationary distribution is π. Define the modified log-Sobolev constant of P as*

$$\rho(P) := \inf\left\{ \frac{\mathcal{E}_P(f,\log f)}{\mathrm{Ent}_\pi(f)} \;\middle|\; f : [n] \to \mathbb{R}_{\geq 0},\ \mathrm{Ent}_\pi[f] \neq 0 \right\}.$$

These definitions may not appear intuitive. In the following, we first state the results of using log-Sobolev constants to bound mixing times, and then provide some intuition behind these definitions.

Bounding Mixing Time by Log-Sobolev Constants

Recall from Theorem 3.19 that the ϵ-mixing time can be bounded by the spectral gap as

$$\tau_\epsilon(P) \lesssim \frac{1}{\lambda(P)}\left(\log\frac{1}{\pi_{\min}} + \log\frac{1}{\epsilon}\right).$$

The significance of the log-Sobolov constant is that it provides a much better dependence on $1/\pi_{\min}$. The following theorem was proved by Diaconis and Saloff-Coste.

Theorem 28.20 (Mixing Time by Log-Sobolev Constant [DSC96]). *Let $P \in \mathbb{R}^{n\times n}$ be the transition matrix of a reversible Markov chain whose stationary distribution is π. Then*

$$\tau_\epsilon(P) \lesssim \frac{1}{\gamma(P)}\left(\log\log\frac{1}{\pi_{\min}} + \log\frac{1}{\epsilon}\right).$$

Bobkov and Tetali proved a similar result for the modified log-Sobolev constant.

Theorem 28.21 (Mixing Time by Modified Log-Sobolev Constant [BT06]). *Let $P \in \mathbb{R}^{n\times n}$ be the transition matrix of a reversible Markov chain whose stationary distribution is π. Then*

$$\tau_\epsilon(P) \lesssim \frac{1}{\rho(P)}\left(\log\log\frac{1}{\pi_{\min}} + \log\frac{1}{\epsilon}\right).$$

It was established in [BT06] that

$$2\lambda(P) \geq \rho(P) \geq 4\gamma(P),$$

showing that the lower bounds on these constants are increasingly difficult to obtain. The modified log-Sobolev constant has the advantage of providing the same upper bound on the mixing time while always being at least as large as the log-Sobolev constant.

Intuition from Continuous Time Random Walks

The definitions of the spectral gap and the modified log-Sobolev constant arise naturally from continuous time random walks.

Definition 28.22 (Continuous Time Random Walks)**.** *Let $P \in \mathbb{R}^{n\times n}$ be the transition matrix of a reversible Markov chain whose stationary distribution is π. For any $t \geq 0$, the transition matrix, or the heat kernel, is defined as*

$$H_t = e^{-t(I-P)} = \sum_{k=0}^{\infty} \frac{t^k (P-I)^k}{k!}.$$

Let $p_0 \in \mathbb{R}^n$ be an initial distribution. Then $p_t^\top = p_0^\top H_t$ is the distribution at time t.

We consider $f_t := p_t/\pi$ and keep track of how fast it converges to $\vec{1}$.

Claim 28.23 (Change of Density)**.** *Let $P \in \mathbb{R}^{n\times n}$ be the transition matrix of a reversible Markov chain whose stationary distribution is π. Let $f_t(i) = p_t(i)/\pi(i)$ for all $i \in [n]$ be the density of p_t with respect to π at time $t \geq 0$. For any initial distribution p_0 and all $t \geq 0$,*

$$f_t = H_t f_0 \quad \text{and} \quad \frac{df_t}{dt} = (P-I)f_t.$$

The change in variance is exactly the Dirichlet form.

Lemma 28.24 (Change of Variance)**.** *Let $P \in \mathbb{R}^{n\times n}$ be the transition matrix of a reversible Markov chain whose stationary distribution is π. Let $f_t = p_t/\pi$. Then*

$$\frac{d}{dt}\mathrm{Var}_\pi(f_t) = -2\mathcal{E}_P(f_t, f_t).$$

Proof. Note that $\mathrm{Var}_\pi(f_t) = \mathbb{E}_\pi[f_t^2] - 1$ by Definition 28.12, and so

$$\frac{d}{dt}\mathrm{Var}_\pi(f_t) = \sum_{i=1}^n \pi(i)\cdot \frac{d}{dt} f_t(i)^2 = 2\sum_{i=1}^n \pi(i)\cdot f_t(i)\cdot \big((P-I)f_t\big)(i) = -2\mathcal{E}_P(f_t, f_t).$$

□

The spectral gap in Definition 28.17 was *defined* to ensure that

$$\frac{d}{dt}\mathrm{Var}_\pi(f_t) = -2\mathcal{E}_P(f_t, f_t) \leq -2\lambda(P)\cdot \mathrm{Var}_\pi(f_t) \quad \Longrightarrow \quad \frac{d}{dt}\log\big(\mathrm{Var}_\pi(f_t)\big) \leq -2\lambda(P).$$

By integrating on both sides, the variance is exponentially decreasing as

$$\log\big(\mathrm{Var}_\pi(f_t)\big) - \log\big(\mathrm{Var}_\pi(f_0)\big) \leq -2\lambda(P)\cdot t \quad \Longrightarrow \quad \mathrm{Var}_\pi(f_t) \leq \mathrm{Var}_\pi(f_0)\cdot e^{-2\lambda(P)\cdot t}.$$

Since the initial variance $\mathrm{Var}_\pi(f_0) \leq \frac{1}{\pi_{\min}}$, Theorem 3.19 for continuous time random walks follows.

Bobkov and Tetali used the same logic to define the modified log-Sobolev constant by using relative entropy in place of variance.

Lemma 28.25 (Change of Relative Entropy). *Let $P \in \mathbb{R}^{n\times n}$ be the transition matrix of a reversible Markov chain whose stationary distribution is π. Let $f_t = p_t/\pi$. Then*

$$\frac{d}{dt}\mathrm{Ent}_\pi(f_t) = -\mathcal{E}_P(f_t, \log f_t).$$

Proof. Note that $\mathbb{E}_\pi[f] = 1$ and thus $\mathrm{Ent}_\pi[f] = \mathbb{E}_\pi[f \log f]$ by Definition 28.15, and hence

$$\frac{d}{dt}\mathrm{Ent}_\pi(f_t) = \sum_{i=1}^n \pi(i)\cdot \frac{d}{dt} f_t(i)\log f_t(i) = \sum_{i=1}^n \pi(i)\cdot\big(1+\log f_t(i)\big)\cdot\big((P-I)f_t\big)(i) = -\mathcal{E}_P(f_t, \log f_t),$$

where last equality uses $\sum_{i=1}^n \pi(i)\cdot((P-I)f_t)(i) = \langle \pi, (P-I)f_t\rangle = 0$. □

The modified log-Sobolev constant was *defined* to ensure that

$$\frac{d}{dt}\mathrm{Ent}_\pi(f_t) = -\mathcal{E}_P(f_t,\log f_t) \le -\rho(P)\cdot \mathrm{Ent}_\pi(f_t) \quad\Longrightarrow\quad \mathrm{Ent}_\pi(f_t) \le \mathrm{Ent}_\pi(f_0)\cdot e^{-\rho(P)\cdot t}.$$

Crucially, since the initial relative entropy is

$$\mathrm{Ent}_\pi[f_0] = \sum_{i=1}^n p_0(i)\cdot\log\frac{p_0(i)}{\pi(i)} \le \log\frac{1}{\pi_{\min}},$$

Theorem 28.21 for continuous time random walks follows from Pinsker's inequality in Theorem 28.14.

28.5 Log-Sobolev Constant for Strongly Log-Concave Distribution

It has been notoriously difficult to prove a lower bound on the log-Sobolev constant of a Markov chain. This is starting to change after the resolution of the matroid expansion conjecture. Not only do the techniques from high-dimensional expanders provide a direct way to establish lower bounds on the spectral gap, but recent developments have also extended these techniques to establish lower bounds on the modified log-Sobolev constant. The first result in this direction was obtained by Cryan, Guo and Mousa.

Theorem 28.26 (Modified Log-Sobolev Constant for Strongly Log-Concave Distribution [CGM21]). *Let μ be a d-homogeneous strongly log-concave distribution. Then the modified log-Sobolev constant of the down-up walk $P_d^\triangledown$ in Definition 27.5 is*

$$\rho(P_d^\triangledown) \ge \frac{1}{d}.$$

The proof shows that the relative entropy decreases exponentially after one step of the down-up walk, such that

$$\mathcal{D}_{\mathrm{KL}}(P_d^\triangledown p \,\|\, \pi) \le \left(1-\frac{1}{d}\right)\cdot \mathcal{D}_{\mathrm{KL}}(p\,\|\,\pi).$$

This provides an optimal mixing time analysis for the down-up walk on matroid bases.

Corollary 28.27 (Optimal Mixing Time for Sampling Matroid Bases [CGM21]). *The mixing time of the down-up walk in Chapter 27 for sampling uniform random matroid bases of size d on $[n]$ is*

$$\tau_\epsilon(P_d^\triangledown) \lesssim d\Big(\log d + \log\log n + \log\frac{1}{\epsilon}\Big).$$

Near-Linear Time Algorithm for Random Spanning Trees

An immediate consequence of Corollary 28.27 is that the mixing time of the down-up walk for sampling uniform random spanning trees is at most $O(n \log n)$.

To design a near-linear time algorithm, one needs to efficiently implement each iteration of the down-up walk. However, it is not known how to do this directly.

Fortunately, the trick in [ALO+21] is to consider the down-up walk on the *dual* matroid. Given a graph $G = (V, E)$, the rank of the dual matroid is $|E| - |V| + 1 \leq |E|$. By Corollary 28.27, the mixing time of the down-up walk on the dual matroid is $O(|E| \log \frac{|E|}{\epsilon})$.

The resulting algorithm is as follows. Let T_0 be an arbitrary spanning tree. In iteration $t \geq 0$, sample a uniform random edge $e \in E - T_t$. Then, sample a uniform random edge f in the unique cycle in $T_t + e$ and set $T_{t+1} := T_t + e - f$, and repeat. This algorithm was studied by Russo, Teixeira and Francisco, who showed that each iteration can be implemented in amortized $O(\log |E|)$ time using the cut-link trees data structures.

Theorem 28.28 (Near-Linear Time Algorithm for Sampling Random Spanning Trees [ALO+21])**.** *Given a graph $G = (V, E)$, there is an algorithm to sample a random spanning tree in G with distribution ϵ-close to the uniform distribution and running time $O(|E| \log |E| \log \frac{|E|}{\epsilon})$.*

The problem of designing a fast algorithm for sampling a uniform random spanning tree is well-studied. The previous best known algorithm, due to Schild, achieves an almost-linear running time of $O(m^{1+o(1)})$. This approach relies on simulating another Markov chain for generating random spanning trees using techniques from Laplacian solvers and electrical flows. The algorithm by Schild is sophisicated and complicated, and so Theorem 28.28 is a dramatic simplification based on a better analysis of the mixing time.

Concentration Inequality for Strongly Log-Concave Distribution

One main application of log-Sobolev inequalities is to prove concentration inequalities [BLM13, VH14]. The following result is a consequence of Theorem 28.26 for strongly log-concave distributions.

Theorem 28.29 (Concentration of Strongly Log-Concave Distributions [CGM21])**.** *Let μ be a d-homogeneous strongly log-concave distribution with support $\Omega \subseteq \{0,1\}^n$. For any observable function $f : \Omega \to \mathbb{R}$ and $a \geq 0$,*

$$\Pr_{x \sim \mu}\left[|f(x) - \mathbb{E}_\pi f| \geq a\right] \leq 2 \exp\Big(-\frac{a^2}{2d \cdot \nu(f)}\Big),$$

where $\nu(f)$ is the maximum of one-step variances

$$\nu(f) := \max_{x \in \Omega}\left\{\sum_{y \in \Omega} P_d^{\triangledown}(x, y) \cdot \big(f(x) - f(y)\big)^2\right\}.$$

The proof of Theorem 28.29 follows from the "standard" Herbst argument (see [BLM13]).

Entropic Independence

Anari, Jain, Koehler, Pham, and Vuong [AJK+22] introduced a general framework to derive entropy contraction and establish optimal mixing times for down-up walks.

Definition 28.30 (Down Operator). *For a ground set $[n]$ and $n \geq k \geq l$, define the row-stochastic down operator $D_{k\to l} \in \mathbb{R}^{\binom{[n]}{k}\times\binom{[n]}{l}}$ as*

$$D_{k\to l}(S,T) = \begin{cases} 1/\binom{k}{l} & \text{if } T \subseteq S, \\ 0 & \text{otherwise.} \end{cases}$$

The notion of spectral independence is about the variance contraction of the down operator, while the notion of entropic independence is about the entropy contraction of the down operator.

Definition 28.31 (Entropic Independence). *A probability distribution μ on $\binom{[n]}{k}$ is said to be $\frac{1}{\gamma}$-entropically independent for $\gamma \in (0,1]$, if for all probability distributions ν on $\binom{[n]}{k}$,*

$$\mathcal{D}_{\mathrm{KL}}\big(\nu D_{k\to 1} \parallel \mu D_{k\to 1}\big) \leq \frac{1}{\gamma k} \cdot \mathcal{D}_{\mathrm{KL}}(\nu \parallel \mu).$$

They proved that entropic independence implies a large modified log-Sobolev constant. The following is a special case of their result.

Theorem 28.32 (Local-to-Global Entropy Contraction [AJK+22]). *Suppose $\mu : \binom{[n]}{k} \to \mathbb{R}_{\geq 0}$ is $\frac{1}{\gamma}$-entropically independent for all conditional distributions. Then the down-up walk P_k^{∇} with respect to μ has modified log-Sobolev constant $\Omega(1-\frac{1}{\gamma k})$.*

They also proved that a probability distribution μ is $\frac{1}{\gamma}$-entropic independence if the distribution is "γ-fractionally log-concave". This result considerably generalizes the work of Cryan, Guo, and Mousa in Theorem 28.26 for strongly log-concave distributions and provides a general framework for obtaining optimal mixing time analysis for down-up walks.

All the proofs in this line of work are very elegant.

28.6 Problems

Problem 28.33 (Simplicial Complex for Graph Coloring). *Define a simplicial complex for graph coloring so that the down-up walk matrix corresponds exactly to the Glauber dynamics. Define the corresponding notion of spectral independence and compare to those defined in [CGSV21, CLV21].*

Problem 28.34 (Correlation Matrix and Influence Matrix). *Show that the correlation matrix in Definition 28.1 and the influence matrix in Remark 28.2 have the same nonzero eigenvalues.*

Hint: Show that the correlation matrix Ψ has the block structure $\begin{bmatrix} A & -A \\ B & -B \end{bmatrix}$ such that $\Phi = A - B$.

References

[AASV21] Yeganeh Alimohammadi, Nima Anari, Kirankumar Shiragur, and Thuy-Duong Vuong. Fractionally log-concave and sector-stable polynomials: counting planar matchings and more. In *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing,*, pages 433–446. ACM, 2021. 349

[AJK+22] Nima Anari, Vishesh Jain, Frederic Koehler, Huy Tuan Pham, and Thuy-Duong Vuong. Entropic independence: optimal mixing of down-up random walks. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1418–1430, 2022. 9, 360

[AJK+24] Nima Anari, Vishesh Jain, Frederic Koehler, Huy Tuan Pham, and Thuy-Duong Vuong. Universality of spectral independence with applications to fast mixing in spin glasses. In *Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 5029–5056. SIAM, 2024. 354

[ALO20] Nima Anari, Kuikui Liu, and Shayan Oveis Gharan. Spectral independence in high-dimensional expanders and applications to the hardcore model. In *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS*, pages 1319–1330. IEEE, 2020. 9, 349, 351, 353

[ALO+21] Nima Anari, Kuikui Liu, Shayan Oveis Gharan, Cynthia Vinzant, and Thuy-Duong Vuong. Log-concave polynomials IV: approximate exchange, tight mixing times, and near-optimal sampling of forests. In *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing,*, pages 408–420. ACM, 2021. 359

[ALO22] Dorna Abdolazimi, Kuikui Liu, and Shayan Oveis Gharan. A matrix trickle-down theorem on simplicial complexes and applications to sampling colorings. In *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 161–172. IEEE, 2022. 338, 354

[BCC+22] Antonio Blanca, Pietro Caputo, Zongchen Chen, Daniel Parisi, Daniel Stefankovic, and Eric Vigoda. On mixing of markov chains: Coupling, spectral independence, and entropy factorization. In *Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 3670–3692, 2022. 354

[BLM13] Stéphane Boucheron, Gábor Lugosi, and Pascal Massart. *Concentration inequalities: A nonasymptotic theory of independence*. Oxford university press, 2013. 359

[BT06] Sergey G Bobkov and Prasad Tetali. Modified logarithmic sobolev inequalities in discrete settings. *Journal of Theoretical Probability*, 19(2):289–336, 2006. 356

[CCYZ24] Xiaoyu Chen, Zongchen Chen, Yitong Yin, and Xinyuan Zhang. Rapid mixing at the uniqueness threshold. *arXiv preprint arXiv:2411.03413*, 2024. 353

[CFYZ22] Xiaoyu Chen, Weiming Feng, Yitong Yin, and Xinyuan Zhang. Optimal mixing for two-state anti-ferromagnetic spin systems. In *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 588–599. IEEE, 2022. 353

[CGM21] Mary Cryan, Heng Guo, and Giorgos Mousa. Modified log-sobolev inequalities for strongly log-concave distributions. *Annals of Probability*, 49(1):506–525, 2021. 358, 359

[CGSV21] Zongchen Chen, Andreas Galanis, Daniel Stefankovic, and Eric Vigoda. Rapid mixing for colorings via spectral independence. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1548–1557. SIAM, 2021. 349, 354, 360

[CLMM23] Zongchen Chen, Kuikui Liu, Nitya Mani, and Ankur Moitra. Strong spatial mixing for colorings on trees and its algorithmic applications. In *2023 IEEE 64th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 810–845. IEEE, 2023. 354

[CLV21] Zongchen Chen, Kuikui Liu, and Eric Vigoda. Optimal mixing of glauber dynamics: entropy factorization via high-dimensional expansion. In *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing,*, pages 1537–1550. ACM, 2021. 354, 360

[CLV24] Zongchen Chen, Kuikui Liu, and Eric Vigoda. Spectral independence via stability and applications to holant-type problems. *TheoretiCS*, 3, 2024. 354

[CV25] Charlie Carlson and Eric Vigoda. Flip dynamics for sampling colorings: Improving $11/6 - \epsilon$ using a simple metric. In *Proceedings of the 2025 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2194–2212. SIAM, 2025. 354

[DSC96] Persi Diaconis and Laurent Saloff-Coste. Logarithmic sobolev inequalities for finite markov chains. *The Annals of Applied Probability*, 6(3):695–750, 1996. 356

[Liu21] Kuikui Liu. From coupling to spectral independence and blackbox comparison with the down-up walk. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2021)*, pages 32–1. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2021. 354

[Sly10] Allan Sly. Computational transition at the uniqueness threshold. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pages 287–296. IEEE, 2010. 353

[SV23] Daniel Stefankovic and Eric Vigoda. Lecture notes on spectral independence and bases of a matroid: Local-to-global and trickle-down from a markov chain perspective. *arXiv preprint arXiv:2307.13826*, 2023.

[VH14] Ramon Van Handel. Probability in high dimension. Technical report, PRINCETON UNIV NJ, 2014. 359

[Vig99] Improved bounds for sampling colorings. In *40th Annual Symposium on Foundations of Computer Science (Cat. No. 99CB37039)*, pages 51–59. IEEE, 1999. 354

[Wei06] Dror Weitz. Counting independent sets up to the tree threshold. In *Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*, pages 140–149, 2006. 353

[WZZ24] Yulin Wang, Chihao Zhang, and Zihan Zhang. Sampling proper colorings on line graphs using $(1+o(1))\delta$ colors. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing*, pages 1688–1699, 2024. 338, 354

Appendix

Chapter A

Linear Algebra

A.1 Eigenvalues and Eigenvectors

Definition A.1 (Eigenvalues and Eigenvectors)**.** *Let A be an $n \times n$ matrix. A nonzero vector v is called an eigenvector of A if $Av = \lambda v$ for some scalar λ. A scalar λ is called an eigenvalue of A if there exists an eigenvector v with $Av = \lambda v$.*

The multi-set of eigenvalues of A is given by the roots of the characteristic polynomial. While this viewpoint is not often used in this course, it plays a central role in recent breakthroughs in spectral graph theory using interlacing polynomials (see [Spi19]).

Definition A.2 (Characteristic Polynomial)**.** *Let A be an $n \times n$ matrix. The characteristic polynomial of A is defined as $p_A(x) := \det(xI - A)$.*

Two matrices are said to be similar if one is obtained from another by a change of basis.

Definition A.3 (Similar Matrices)**.** *A matrix X is similar to another matrix Y if there exists a non-singular matrix B so that $X = BYB^{-1}$.*

It is well known that similar matrices have the same spectrum.

Fact A.4 (Spectrum of Similar Matrices)**.** *If X is similar to Y, then the multiset of eigenvalues of X and that of Y are the same.*

Proof. One way to see this is that they have the same characteristic polynomial:

$$p_X(x) = \det(xI - X) = \det(xI - BYB^{-1}) = \det(B(xI - Y)B^{-1}) = \det(xI - Y) = p_Y(x),$$

where the second-to-last equality follows from Fact A.27. □

Real Symmetric Matrices

In this course, we mostly work with real symmetric matrices, which have real eigenvalues and an orthonormal basis of eigenvectors.

Theorem A.5 (Spectral Theorem for Real Symmetric Matrices)**.** *Let $A \in \mathbb{R}^{n\times n}$ be a symmetric matrix. Then all eigenvalues of A are real numbers, and there exists an orthonormal basis of $\mathbb{R}^n$ consisting of eigenvectors of A.*

A proof of this fundamental theorem can be found in most linear algebra textbooks. We recommend the proofs in [GR, Tre17].

Remark A.6 (Undirected and Directed Graphs)**.** *The spectral theorem applies to the adjacency and Laplacian matrices of undirected graphs, but not to those of directed graphs. This is why the spectral theory for undirected graphs is much more devleoped. Developing a comparable spectral theory for directed graphs remains an open direction.*

Diagonalization: Using the spectral theorem, real symmetric matrices can be written in the following form. Let $A \in \mathbb{R}^{n\times n}$ be a symmetric matrix. Let $v_1, \ldots, v_n \in \mathbb{R}^n$ be an orthonormal basis of eigenvectors guaranteed by Theorem A.5 with corresponding eigenvalues $\lambda_1, \ldots, \lambda_n$. Let V be the $n \times n$ matrix whose i-th column is v_i. Let D be the $n \times n$ diagonal matrix with $D_{i,i} = \lambda_i$. Then the conditions $Av_i = \lambda_i v_i$ for $1 \le i \le n$ can be compactly written as $AV = VD$. Since the columns of V form an orthonormal basis, it follows that $V^\top V = I$ and thus $V^{-1} = V^\top$. Hence, we can rewrite $AV = VD$ as

$$A = VDV^{-1} = VDV^\top. \tag{A.1}$$

Power of Matrices: For a symmetric matrix $A \in \mathbb{R}^{n\times n}$, the diagonalized form $A = VDV^\top$ simplifies computations. To compute A^k, note that

$$A^k = (VDV^\top)^k = VD^kV^\top,$$

where D^k is computed by raising the diagonal entries of D to the k-th power.

This is particularly useful in analyzing random walks. For instance, P^t, the transition matrix of a random walk after t steps, can be expressed in terms of the eigenvalues of P to bound the mixing time.

Eigen-Decomposition and Pseudoinverse: If $v_1, \ldots, v_n$ form an orthonormal basis, then any $x \in \mathbb{R}^n$ can be written as a linear combination $c_1v_1 + \ldots + c_nv_n$. By orthonormality, for any $1 \le i \le n$,

$$\langle x, v_i\rangle = \langle c_1v_1 + \ldots + c_nv_n, v_i\rangle = \langle c_iv_i, v_i\rangle = c_i.$$

Therefore, for any $x \in \mathbb{R}^n$,

$$x = \langle x, v_1\rangle v_1 + \ldots + \langle x, v_n\rangle v_n = v_1v_1^\top x + \ldots v_nv_n^\top x = \big(v_1v_1^\top + \ldots + v_nv_n^\top\big)x.$$

Since this is true for all $x \in \mathbb{R}^n$, it follows that

$$v_1v_1^\top + \ldots + v_nv_n^\top = I_n.$$

Now, if $v_1, \ldots, v_n$ are also eigenvectors of a matrix $A \in \mathbb{R}^{n\times n}$, then for any $x \in \mathbb{R}^n$,

$$Ax = A(v_1v_1^\top + \ldots + v_nv_n^\top)x = \big(\lambda_1v_1v_1^\top + \ldots + \lambda_nv_nv_n^\top\big)x.$$

This implies that

$$A = \lambda_1 v_1 v_1^\top + \ldots + \lambda_n v_n v_n^\top.$$

Verify that we can also write the inverse using the eigen-decomposition as

$$A^{-1} = \frac{1}{\lambda_1} v_1 v_1^\top + \ldots + \frac{1}{\lambda_n} v_n v_n^\top.$$

This form will also be used to define the "pseudo-inverse" of a matrix A when A is not of full rank.

Definition A.7 (Moore-Penrose Pseudoinverse)**.** *Let $A \in \mathbb{R}^{n \times n}$ be a real symmetric matrix with eigen-decomposition $A = \sum_{i=1}^n \lambda_i v_i v_i^\top$. The pseudoinverse of A, denoted by $A^\dagger$, is defined as*

$$A^\dagger := \sum_{i:\lambda_i \neq 0} \frac{1}{\lambda_i} v_i v_i^\top.$$

Check the following properties of pseudoinverse.

Fact A.8 (Properties)**.** *Let A be a real symmetric matrix and $A^\dagger$ be its pseudoinverse. Then*

$$AA^\dagger A = A \quad \text{and} \quad A^\dagger A A^\dagger = A^\dagger \quad \text{and} \quad \left(A^\dagger\right)^\dagger = A.$$

Positive Semidefinite Matrices

An important class of real symmetric matrices is the class of positive semidefinite matrices. A real symmetric matrix is called positive semidefinite if all of its eigenvalues are nonnegative. This can be seen as a matrix analog of a non-negative number. The following are some equivalent characterizations of a positive semidefinite matrix.

Fact A.9 (Positive Semidefinite Matrix)**.** *Let $A \in \mathbb{R}^{n \times n}$ be a real symmetric matrix. The following statements are equivalent.*

1. *A is positive semidefinite, i.e., all eigenvalues of A are non-negative.*
2. *For any $x \in \mathbb{R}^n$, it holds that $x^\top A x \geq 0$, i.e., all quadratic forms are non-negative.*
3. *$A = B^\top B$ for some matrix $B \in \mathbb{R}^{m \times n}$ for some positive integer m.*

The notation $A \succcurlyeq 0$ is used to denote that A is a positive semidefinite matrix.

It is a good exercise to prove this fact. A matrix is called positive definite if all eigenvalues of A are positive. It is left as an exercise to find the equivalent characterizations for positive definite matrices as in Fact A.9.

Check that the set of positive semidefinite matrices forms a convex set. Optimizing a linear function over the set of positive semidefinite matrices with linear constraints is called semidefinite programming. This is an important class of convex optimization problems that can be solved in polynomial time and is a powerful tool in designing approximation algorithms. More background are provided in the relevant chapters.

Exercise A.10. *Prove that for any two positive semidefinite matrices $A, B \in \mathbb{R}^{n \times n}$,*

$$\langle A, B \rangle := \sum_{i=1}^n \sum_{j=1}^n A_{ij} B_{ij} \geq 0.$$

Optimization Formulation for Eigenvalues

The main reason why eigenvalues are related to optimization problems is through the following formulation, which is the quadratic form normalized by the vector length.

Definition A.11 (Rayleigh Quotient). *The Rayleigh quotient of a vector $x \in \mathbb{R}^n$ with respect to a matrix $A \in \mathbb{R}^{n \times n}$ is defined to be*

$$R_A(x) := \frac{x^\top A x}{x^\top x} = \frac{\sum_{i=1}^n \sum_{j=1}^n A_{ij} x_i x_j}{\sum_{i=1}^n x_i^2}.$$

The largest eigenvalue is the maximum value of the Rayleigh quotient, with the corresponding eigenvectors being an optimal solution.

Lemma A.12 (Optimization Formulation for α_1). *Suppose $A \in \mathbb{R}^{n \times n}$ is a real symmetric matrix with eigenvalues $\alpha_1 \geq \alpha_2 \geq \ldots \geq \alpha_n$. Then*

$$\alpha_1 = \max_{x \in \mathbb{R}^n} \frac{x^\top A x}{x^\top x}.$$

Proof. Let $v_1, v_2, \ldots, v_n$ be the corresponding orthonormal basis of eigenvectors guaranteed by Theorem A.5. As $v_1, \ldots, v_n$ forms a basis of $\mathbb{R}^n$, any vector $x \in \mathbb{R}^n$ can be written as a linear combination $x = c_1 v_1 + \cdots + c_n v_n$. Then, the numerator can be written as

$$x^\top A x = (c_1 v_1 + \cdots + c_n v_n)^\top A (c_1 v_1 + \cdots + c_n v_n) = (c_1 v_1 + \cdots + c_n v_n)^\top (c_1 \alpha_1 v_1 + \cdots + c_n \alpha_n v_n) = \sum_{i=1}^n c_i^2 \alpha_i,$$

where the second equality follows since $v_1, \cdots, v_n$ are eigenvectors and the last equality is because $v_1, \cdots, v_n$ are orthonormal. Similarly, the denominator can be written as

$$x^\top x = (c_1 v_1 + \cdots + c_n v_n)^\top (c_1 v_1 + \cdots + c_n v_n) = \sum_{i=1}^n c_i^2.$$

Thus, the Rayleigh quotient of x is

$$\frac{x^\top A x}{x^\top x} = \frac{\sum_{i=1}^n c_i^2 \alpha_i}{\sum_{i=1}^n c_i^2} \leq \frac{\alpha_1 \sum_{i=1}^n c_i^2}{\sum_{i=1}^n c_i^2} = \alpha_1.$$

On the other hand, note that v_1 attains the maximum, and the lemma follows. □

This result can be extended to characterize other eigenvalues. The following lemma is useful for characterizing the second largest eigenvalue of a matrix when the first eigenvector is known.

Lemma A.13 (Optimization Formulation for α_k). *Suppose $A \in \mathbb{R}^{n \times n}$ is a real symmetric matrix with eigenvalues $\alpha_1 \geq \alpha_2 \geq \ldots \geq \alpha_n$ and corresponding orthonormal eigenvectors $v_1, \ldots, v_n$. Let T_k be the set of vectors that are orthogonal to $v_1, v_2, \ldots, v_{k-1}$. Then*

$$\alpha_k = \max_{x \in T_k} \frac{x^\top A x}{x^\top x}.$$

Proof. Let $x \in T_k$. Write $x = c_1 v_1 + \cdots + c_n v_n$. Recall that $c_i = \langle x, v_i \rangle$ from the eigen-decomposition part. Since $x \in T_k$, it follows that $c_1 = c_2 = \cdots = c_{k-1} = 0$. Using the same calculation as in Lemma A.12,

$$\frac{x^\top A x}{x^\top x} = \frac{\sum_{i=k}^n c_i^2 \alpha_i}{\sum_{i=k}^n c_i^2} \leq \frac{\alpha_k \sum_{i=k}^n c_i^2}{\sum_{i=k}^n c_i^2} = \alpha_k.$$

On the other hand, $v_k \in T_k$ and $v_k^\top A v_k / v_k^\top v_k = \alpha_k$, so the lemma follows. □

The above result gives a characterization of α_k, but it requires knowledge of the previous eigenvectors. The Courant-Fischer theorem provides a characterization without requiring prior knowledge of the eigenvectors, and this is useful for proving upper and lower bounds on eigenvalues. In words, the Courant-Fischer theorem says that to prove a lower bound on α_k, one needs to exhibit a k-dimensional subspace in which every vector has large Rayleigh quotient, and the best k-dimensional subspace gives the tight lower bound. Similarly, to prove an upper bound on α_k, one needs to exhibit an $(n-k+1)$-dimensional subspace in which every vector has small Rayleigh quotient, and the best $(n-k+1)$-dimensional subspace gives the tight upper bound.

Theorem A.14 (Courant-Fischer Theorem). *Suppose $A \in \mathbb{R}^{n \times n}$ is a real symmetric matrix with eigenvalues $\alpha_1 \geq \alpha_2 \geq \ldots \geq \alpha_n$. Then*

$$\alpha_k = \max_{S \subseteq \mathbb{R}^n : \dim(S) = k} \min_{x \in S} \frac{x^\top A x}{x^\top x} = \min_{S \subseteq \mathbb{R}^n : \dim(S) = n-k+1} \max_{x \in S} \frac{x^\top A x}{x^\top x}.$$

Proof. We prove the max-min equality. The min-max equality is similar and is left as an exercise.

Let S_k be the k-dimensional subspace spanned by the first k orthonormal eigenvectors $v_1, \ldots, v_k$, i.e. $S_k = \{x \mid x = c_1 v_1 + \cdots + c_k v_k \text{ for some } c_1, \ldots, c_k \in \mathbb{R}\}$. For any $x \in S_k$,

$$\frac{x^\top A x}{x^\top x} = \frac{(c_1 v_1 + \cdots + c_k v_k)^\top A (c_1 v_1 + \cdots + c_k v_k)}{(c_1 v_1 + \cdots + c_k v_k)^\top (c_1 v_1 + \cdots + c_k v_k)} = \frac{\sum_{i=1}^k c_i^2 \alpha_i}{\sum_{i=1}^k c_i^2} \geq \frac{\alpha_k \sum_{i=1}^k c_i^2}{\sum_{i=1}^k c_i^2} = \alpha_k.$$

Therefore,

$$\max_{S \subseteq \mathbb{R}^n : \dim(S) = k} \min_{x \in S} \frac{x^\top A x}{x^\top x} \geq \min_{x \in S_k} \frac{x^\top A x}{x^\top x} \geq \alpha_k.$$

To prove that the maximum cannot exceed α_k, observe that any k-dimensional subspace must intersect the $(n-k+1)$-dimensional subspace T_k spanned by $\{v_k, v_{k+1}, \ldots, v_n\}$. For any $x \in T_k$,

$$\frac{x^\top A x}{x^\top x} = \frac{\sum_{i=k}^n c_i^2 \alpha_i}{\sum_{i=k}^n c_i^2} \leq \alpha_k.$$

Therefore,

$$\max_{S \subseteq \mathbb{R}^n : \dim(S) = k} \min_{x \in S} \frac{x^\top A x}{x^\top x} \leq \min_{x \in S \cap T_k} \frac{x^\top A x}{x^\top x} \leq \alpha_k.$$

□

One consequence of the Courant-Fischer theorem is the eigenvalue interlacing theorem.

Theorem A.15 (Cauchy's Interlacing Theorem)**.** *Let $A \in \mathbb{R}^{n\times n}$ be a real symmetric matrix, and B be a $(n-1)\times(n-1)$ principal submatrix of A. Then*

$$\alpha_1 \geq \beta_1 \geq \alpha_2 \geq \beta_2 \geq \ldots \geq \alpha_{n-1} \geq \beta_{n-1} \geq \alpha_n,$$

where $\alpha_1 \geq \alpha_2 \geq \ldots \geq \alpha_n$ and $\beta_1 \geq \beta_2 \geq \ldots \geq \beta_{n-1}$ are the eigenvalues of A and B, respectively.

Proof. Assume without loss of generality that B is in the top left corner of A, i.e., the first $n-1$ coordinates. It should be clear that $\alpha_k \geq \beta_k$ because the search space for α_k is larger than that for β_k. More formally,

$$\alpha_k = \max_{S\subseteq\mathbb{R}^n:\dim(S)=k} \min_{x\in S} \frac{x^\top Ax}{x^\top x} \geq \max_{S\subseteq\mathbb{R}^{n-1}:\dim(S)=k} \min_{x\in S} \frac{x^\top Ax}{x^\top x} = \max_{S\subseteq\mathbb{R}^{n-1}:\dim(S)=k} \min_{x\in S} \frac{x^\top Bx}{x^\top x} = \beta_k.$$

Next, we show $\beta_k \geq \alpha_{k+1}$. For any $S \subseteq \mathbb{R}^n$ with $\dim(S) = k+1$, its restriction to the first $n-1$ coordinates (i.e., $S \cap \mathbb{R}^{n-1}$) has dimension at least k. Thus, if there is a good $(k+1)$-dimensional subspace for A, then there is a good k-dimensional subspace for B, so β_k can do as well as α_{k+1}. More formally, let S_{k+1} be the $(k+1)$-dimensional subspace that attains maximum for α_{k+1},

$$\alpha_{k+1} = \min_{x\in S_{k+1}} \frac{x^\top Ax}{x^\top x} \leq \min_{x\in S_{k+1}\cap\mathbb{R}^{n-1}} \frac{x^\top Ax}{x^\top x} \leq \max_{S\subseteq\mathbb{R}^{n-1}:\dim(S)=k} \min_{x\in S} \frac{x^\top Ax}{x^\top x} = \beta_k.$$

□

Perron-Frobenius Theorem

The Perron-Frobenius theorem is an important result about the largest eigenvalue and its corresponding eigenvectors of non-negative matrices. To state it, we need the definitions of an irreducible matrix and the spectral radius of a matrix.

Definition A.16 (Irreducible Matrix)**.** *A matrix $A \in \mathbb{R}^{n\times n}$ is irreducible if its underlying directed graph $G = (V, E)$ is strongly connected, where the vertex set of G is $V = [n]$ and the edge set of G is $E = \{ij \mid A_{ij} \neq 0\}$.*

The spectral radius of a real symmetric matrix is simply the eigenvalue with largest absolute value. The following is the more general definition for matrices with complex eigenvalues.

Definition A.17 (Spectral Radius)**.** *The spectral radius $\rho(A)$ of a matrix A is the maximum of the moduli of its eigenvalues.*

The Perron-Frobenius theorem is particularly useful in the study of random walks. See Chapter 8.8 in [GR] and Chapter 8.4 in [HJ13] for more details and proofs.

Theorem A.18 (Perron-Frobenius Theorem)**.** *Let $A \in \mathbb{R}^{n\times n}$ be a non-negative irreducible matrix.*

1. *The spectral radius $\rho(A)$ is an eigenvalue of A with multiplicity one. In particular, if A is also a real symmetric matrix, then the largest eigenvalue is of multiplicity one, and its absolute value is the largest.*
2. *If v is an eigenvector with eigenvalue $\rho(A)$, then all the entries of v are nonzero and have the same sign.*

Matrix Norms

Definition A.19 (Operator Norm)**.** *Let A be an $m \times n$ matrix. The operator norm $\|A\|_{\text{op}}$ of A is defined as*

$$\|A\|_{\text{op}} := \sup_{x \in \mathbb{R}^n, x \neq 0} \frac{\|Ax\|_2}{\|x\|_2}.$$

This norm is also denoted as $\|A\|_2$ to indicate that it relates to the 2-norm of vectors. However, this can sometimes be confused with the Schatten 2-norm of A, also denoted as $\|A\|_2$, so we use the notation $\|A\|_{\text{op}}$ which is also commonly used.

Exercise A.20 (Operator Norm of a Real Symmetric Matrix)**.** *Show that $\|A\|_{\text{op}}$ is equal to the largest eigenvalue of A when A is a real symmetric matrix.*

The following are some simple properties that will be useful.

Fact A.21 (Properties of Operator Norm)**.** *Let $A \in \mathbb{R}^{m \times n}$.*

1. *$\|A\|_{\text{op}} \geq 0$ and $\|A\|_{\text{op}} = 0$ if and only if $A = 0$.*
2. *$\|cA\|_{\text{op}} = |c| \|A\|_{\text{op}}$ for every scalar c.*
3. *$\|A + B\|_{\text{op}} \leq \|A\|_{\text{op}} + \|B\|_{\text{op}}$.*
4. *$\|Ax\|_2 \leq \|A\|_{\text{op}} \|x\|_2$ for every $x \in \mathbb{R}^n$.*
5. *$\|BA\|_{\text{op}} \leq \|B\|_{\text{op}} \|A\|_{\text{op}}$*

A.2 Formulas and Inequalities

In this section, we record some useful formulas and inequalities in this section. A general reference is the book by Horn and Johnson [HJ13].

Inverse

The following formulas are for updating the inverse of a matrix. Proofs can be found online, such as the Wikipedia.

Fact A.22 (Sherman-Morrison Formula)**.** *Suppose $A \in \mathbb{R}^{n \times n}$ is an invertible square matrix and $u, v \in \mathbb{R}^n$ are column vectors. Then $A + uv^\top$ is invertible if and only if $1 + v^\top A^{-1} u \neq 0$, and*

$$(A + uv^\top)^{-1} = A^{-1} - \frac{A^{-1} u v^\top A^{-1}}{1 + v^\top A^{-1} u}$$

Fact A.23 (Woodbury Formula)**.** *Let A be a square invertible $n \times n$ matrix, U an $n \times k$ matrix, and V a $k \times n$ matrix. Assuming $(I_k + VA^{-1}U)$ is invertible, then*

$$(A + UV)^{-1} = A^{-1} - A^{-1} U \big(I_k + VA^{-1}U\big)^{-1} VA^{-1}.$$

The following formula is for inverting a block matrix, where the Schur complement is a useful definition.

Fact A.24 (Block Matrix Inversion). *Let A and D be square matrices and*

$$M = \begin{pmatrix} A & B \\ C & D \end{pmatrix}.$$

If A and the Schur complement $D - CA^{-1}B$ are invertible, then

$$M^{-1} = \begin{pmatrix} A^{-1} + A^{-1}B\big(D - CA^{-1}B\big)^{-1}CA^{-1} & -A^{-1}B\big(D - CA^{-1}B\big)^{-1} \\ -\big(D - CA^{-1}B\big)^{-1}CA^{-1} & \big(D - CA^{-1}B\big)^{-1} \end{pmatrix}.$$

Determinant

Fact A.25 (Laplace Co-Factor Expansion). *Let A be a $n \times n$ matrix. For every $1 \leq i \leq n$,*

$$\det(A) = \sum_{j=1}^{n} (-1)^{i+j} A_{i,j} \det\left(A_{[n]\setminus i,[n]\setminus j}\right),$$

where $A_{S,T}$ is the submatrix with rows in $S \subseteq [n]$ and columns in $T \subseteq [n]$.

Applying Laplace expansion recursively gives the Leibniz formula.

Fact A.26 (Leibniz Formula). *Let A be a $n \times n$ matrix. Then*

$$\det(A) = \sum_{\sigma \in S_n} \operatorname{sgn}(\sigma) \prod_{i=1}^{n} A_{\sigma(i),i},$$

where S_n is the set of permutations of the set $[n] = \{1, \ldots, n\}$ and $\operatorname{sgn}(\sigma)$ is the sign function of permutation σ, which returns $+1$ for even permutations and -1 for odd permutations.

The following is a simple fact.

Fact A.27 (Product Rule for Determinants).

$$\det(AB) = \det(A)\det(B).$$

The following result, sometimes known as Sylvester's determinant identity, can be used to deduce that the nonzero eigenvalues of AB and BA are the same (with multiplicity).

Fact A.28 (Weinstein-Aronszajn Identity).

$$\det(I + AB) = \det(I + BA).$$

The matrix determinant formula tracks how the determinant changes after a rank-one update.

Fact A.29 (Matrix Determinant Formula).

$$\det\left(M - uv^T\right) = \det(M) \cdot \left(1 - v^T M^{-1} u\right).$$

The Cauchy-Binet formula is a useful tool with applications such as computing the number of spanning trees of a graph.

Fact A.30 (Cauchy-Binet Formula)**.** *Let A be an $m \times n$ matrix and B be an $n \times m$ matrix. Then*

$$\det(AB) = \sum_{S \in \binom{[n]}{m}} \det\left(A_{[m],S}\right) \det\left(B_{S,[m]}\right)$$

The following formula provides the coefficients of the characteristic polynomials.

Fact A.31 (Characteristic Polynomial)**.** *Let A be an $n \times n$ matrix.*

$$\det(\lambda I_n - A) = \sum_{k=0}^{n} \lambda^{n-k}(-1)^k \sum_{S \in \binom{[n]}{k}} \det(A_{S,S}).$$

Using Fact A.31 and Cauchy-Binet formula in Fact A.30, we obtain the following identity for the characteristic polynomial of a sum of outer products.

Fact A.32 (Characteristic Polynomial of Sum of Outer Products)**.** *Let $u_1, \ldots, u_m \in \mathbb{R}^n$.*

$$\det\left(xI - \sum_{i=1}^{m} u_i u_i^\top\right) = \sum_{k=0}^{n} \lambda^{n-k}(-1)^k \sum_{S \in \binom{[m]}{k}} \det_k\left(\sum_{i \in S} u_i u_i^\top\right),$$

where $\det_k(A) = \sum_{S \in \binom{[n]}{k}} \det(A_{S,S})$.

Trace

Definition A.33 (Trace)**.** *The trace of a matrix $A \in \mathbb{R}^{n \times n}$, denoted by $\mathrm{Tr}(A)$, is the sum of the diagonal entries of A.*

Fact A.34 (Cyclic Property of Trace)**.** *For two matrices $A \in \mathbb{R}^{m \times n}$ and $B \in \mathbb{R}^{n \times m}$,*

$$\mathrm{Tr}(AB) = \mathrm{Tr}(BA).$$

By examining the coefficient of x^{n-1} in the characteristic polynomial $\det(xI - A)$ of $A \in \mathbb{R}^{n \times n}$ in two ways, one can derive the following result.

Fact A.35 (Trace is the Sum of Eigenvalues)**.** *Let $\lambda_1, \ldots, \lambda_n$ be the eigenvalues of $A \in \mathbb{R}^{n \times n}$. Then*

$$\mathrm{Tr}(A) = \sum_{i=1}^{n} \lambda_i.$$

Fact A.36. *If $A, B \succcurlyeq 0$, then $\mathrm{Tr}(AB) \geq 0$.*

Fact A.37. *If $A \succcurlyeq 0$, then*

$$\mathrm{Tr}(AB) \leq \|B\|_{\mathrm{op}} \cdot \mathrm{Tr}(A).$$

Lemma A.38. *Let A, B, C be symmetric matrices with $A, B \succcurlyeq 0$. Then*

$$\mathrm{Tr}(ACBC) \leq \mathrm{Tr}(A|C|) \cdot \mathrm{Tr}(B|C|).$$

Lemma A.39 (Matrix Hölder's Inequality). *For $p, q \geq 0$ with $\frac{1}{p} + \frac{1}{q} = 1$,*

$$\operatorname{Tr}(AB) \leq \|A\|_p \cdot \|B\|_q = \operatorname{Tr}\left(|A|^p\right)^{\frac{1}{p}}.$$

The following are two advanced results about traces; see [Bha97].

Theorem A.40 (Golden-Thompson Inequality).

$$\operatorname{Tr}(e^{A+B}) \leq \operatorname{Tr}(e^A) \cdot \operatorname{Tr}(e^B)$$

Theorem A.41 (Lieb-Thirring Inequality). *Let A and B be positive definite matrices and $q \geq 1$. Then*

$$\operatorname{Tr}\left((BAB)^q\right) \leq \operatorname{Tr}(B^q A^q B^q).$$

Matrix Calculus

The formula for differenting the inverse is derived by differentiating the identity $AA^{-1} = I$.

Fact A.42 (Inverse).

$$d(A^{-1}) = -A^{-1}(dA)A^{-1}.$$

Jacobi's formula is obtained by differentiating the cofactor expansion in Fact A.25.

Fact A.43 (Jacobi's Formula).

$$\frac{d}{dt} \det A(t) = \operatorname{Tr}\left(\operatorname{adj}(A(t)) \cdot \frac{dA(t)}{dt}\right) = \left(\det A(t)\right) \cdot \operatorname{Tr}\left(A(t)^{-1} \cdot \frac{dA(t)}{dt}\right).$$

References

[Bha97] Rajendra Bhatia. *Matrix Analysis*, volume 169. Springer, 1997. 374

[GR] Christopher D. Godsil and Gordon F. Royle. *Algebraic Graph Theory*. 366, 370

[HJ13] Roger A. Horn and Charles R. Johnson. *Matrix Analysis*. Cambridge University Press, Cambridge; New York, 2nd edition, 2013. 41, 47, 370, 371

[Spi19] Daniel A. Spielman. *Spectral and Algebraic Graph Theory*. 2019. 15, 319, 365

[Tre17] Luca Trevisan. *Lecture Notes on Graph Partitioning, Expanders and Spectral Methods*. 2017. 15, 68, 366

Chapter B

Notations

$[n]$: the set of integers $\{1, \ldots, n\}$.
$\mathbb{R}_+$: the set of non-negative real number.
$a \lesssim b$: denotes $a = O(b)$.
$a \gtrsim b$: denotes $a = \Omega(b)$.
$a \asymp b$: denotes $a = \Theta(b)$.

$v \in \mathbb{R}^n$: an n-dimensional column vector v.
$v(i)$: the i-th entry of a vector v.
$\vec{1}$: the all-ones vector.
χ_S: the characteristic vector of a subset S with $\chi_S(i) = 1$ if $i \in S$ and $\chi_S(i) = 0$ otherwise.
$\|v\|_1$: the ℓ_1-norm of v defined as $\sum_i |v(i)|$.
$\|v\|_2$: the ℓ_2-norm of v defined as $\sqrt{\sum_i v(i)^2}$.

$A_{i,j}$: the (i, j)-th entry of a matrix A.
$A(i, j)$: the (i, j)-th entry of a matrix A.
I: the identity matrix.
J: the all-ones matrix.
$M^\top$: the transpose of the matrix M.
$\mathrm{Tr}(M)$: the trace of the matrix M in Definition A.33.
$\|M\|_{\mathrm{op}}$: the operator norm of M.
$\|M\|_F$: the Frobenius norm of M.

$A(G)$: the adjacency matrix of graph G in Definition 1.1.
$\mathcal{A}(G)$: the normalized adjacency matrix of G in Definition 1.17.
α_i: the i-th largest eigenvalue of the adjacency matrix or the normalized adjacency matrix.
$L(G)$: the Laplacian matrix of a graph G in Definition 1.11.
$\mathcal{A}(G)$: the normalized Laplacian matrix of G in Definition 1.17.
λ_i: the i-th smallest eigenvalue of the Laplacian matrix or the normalized Laplacian matrix.
$R(x)$: the Rayleigh quotient of a vector x in Lemma 7.4.

$\deg(v)$: the degree of vertex v.
$\delta(S)$: the set of edges with one endpoint in S and the other endpoint not in S.
$\phi(S)$: the edge conductance of a subset S in Definition 2.1.

$\phi(G)$: the edge conductance of a graph G in Definition 2.1.
$\text{vol}(S)$: the volume of a subset S in Definition 2.1.
$\Phi(S)$: the edge expansion of a subset S in Definition 2.10.
$\Phi(G)$: the edge expansion of a graph G in Definition 2.10.
$\psi(S)$: the vertex expansion of a subset S in Definition 2.11.
$\psi(G)$: the vertex expansion of a graph G in Definition 2.11.
$\partial(S)$: the vertex boundary of a subset S in Definition 2.11.
$E(S,T)$: the set of edges with one endpoint in S and the other endpoint in T.

$\deg^+(v)$: the out-degree of vertex v.
$\deg^-(v)$: the in-degree of vertex v.
$\delta^+(S)$: the set of edges with the tail in S and the head not in S in Definition 10.1.
$\delta^-(S) = \delta^+(\overline{S})$: the set of edges with the tail not in S and the head in S.
$\vec{\psi}(S)$: the directed vertex expansion of a subset S in Definition 10.1.
$\vec{\psi}(G)$: the directed vertex expansion of a directed graph G in Definition 10.1.
$\vec{\phi}(S)$: the directed edge conductance of a subset S in Definition 10.6.
$\vec{\phi}(G)$: the directed edge conductance of a directed graph G in Definition 10.6.

$d_{TV}(\vec{p},\vec{q})$: the total variation distance of two distributions $\vec{p}$ and $\vec{q}$ in Definition 3.7.
$\tau_\epsilon(P)$: the ϵ-mixing time of a Markov chain P in Definition 3.14.
$\tau(P)$: the mixing time of a Markov chain P in Definition 3.14.