CS 798 - Algorithmic Spectral Graph Theory, Fall 2015, Waterloo

Lecture 18: Linear-size spectral sparsification

We study a polynomial time algorithm to construct a linear-size spectral sparsifier using barrier arguments.

Statements

The main theorem that we study today is by Batson, Spielman and Srivastava.

Theorem For any graph G, there is a $(1\pm\varepsilon)$-spectral sparsifier H with $O(\frac{n}{\varepsilon^2})$ edges.

This is an amazing result, especially when it was not know that linear-size cut sparsifiers exist. We will think about why this stronger result is easier to prove.

We will first reduce the problem of spectral sparsification to a purely linear algebraic question.

Recall that $L_G = \sum_{e\in G} w_e b_e b_e^T$ where $b_e = \chi_i - \chi_j$ for an edge $e=ij$ and w_e is the weight of edge e.

The problem of spectral sparsification is to find a reweighting of the edges $L_H = \sum_{e\in H} w'_e b_e b_e^T$ so that there are only $O(\frac{n}{\varepsilon^2})$ non-zero edges in H and $(1-\varepsilon)L_G \lesssim L_H \lesssim (1+\varepsilon)L_G$.

This can be reduced to the following simpler form.

Theorem Suppose $\sum_{i=1}^m v_i v_i^T = I$ where $v_i \in \mathbb{R}^n$. There exist $w_i \in \mathbb{R}$ with at most dn non-zeros such that $(1-\frac{1}{\sqrt{d}})^2 I \lesssim \sum_{i=1}^m w_i v_i v_i^T \lesssim (1+\frac{1}{\sqrt{d}})^2 I$.

To reduce the spectral sparsification problem to this theorem about identity matrix, we consider the vectors $y_e = L_G^{+/2} b_e$ and note that

$$\sum_{e\in G} w_e y_e y_e^T = \sum_{e\in G} w_e L_G^{+/2} b_e b_e^T L_G^{+/2} = L_G^{+/2}\left(\sum_{e\in G} w_e b_e b_e^T\right) L_G^{+/2}$$
$$= L_G^{+/2} L_G L_G^{+/2} = I',$$

where I' is the identity operator for vectors $\perp \vec{1}$,

We can apply the theorem in the subspace orthogonal to the all-one vector, and find scalars w'_i with only $d(n-1)$ non-zeros such that $(1-\frac{1}{\sqrt{d}})^2 \sum_e w_e y_e y_e^T \lesssim \sum_e w'_e y_e y_e^T \lesssim (1+\frac{1}{\sqrt{d}})^2 \sum_e w_e y_e y_e^T$.

Multiplying both sides by $L_G^{+/2}$ on the left and on the right, we get

$$(1-\frac{1}{\sqrt{d}})^2 \sum_e w_e b_e b_e^T \lesssim \sum_e w'_e b_e b_e^T \lesssim (1+\frac{1}{\sqrt{d}})^2 \sum_e w_e b_e b_e^T.$$

Let H be the graph with the subset of edges with $w'_e > 0$ and set $d = \frac{1}{\varepsilon^2}$,

we obtain the main result $(1-\varepsilon)^2 L_G \lesssim \sum_{e\in H} w'_e b_e b_e^T \lesssim (1+\varepsilon)^2 L_G$ where H has $\frac{n}{\varepsilon^2}$ edges.

So, we focus on the theorem about the identity matrix for the rest of today.

Random sampling

To understand the problem more, let us see how the random sampling algorithm by effective resistance works in this setting, and see that why it won't work to obtain linear-size sparsifiers.

Just consider the case where $w_e = 1$ for all $e \in G$.

We consider the vectors $y_e = L_G^{+/2} b_e$ such that $\sum_{e \in G} y_e y_e^T = I'$.

Notice that the length of y_e is $\|y_e\|_2^2 = b_e L_G^{+/2} L_G^{+/2} b_e = b_e L_G^{+} b_e = R_{eff}(e)$.

So, the sampling by effective resistance algorithm in the identity matrix setting is simply to sample a vector with probability proportional to its length.

This makes sense as we should be more conservative about long vectors, because if we assign it a small probability p_e and then put it with weight $\frac{1}{p_e} y_e y_e^T$, we may overshoot in this direction.

On the other hand, if the length is small, we can be more aggressive and set the sampling probability smaller, and we have concentration for a sum of many small variables.

However, this sampling algorithm will not work to give linear-size sparsifiers.

Consider the simple example where $\sum_{i=1}^{n} e_i e_i^T = I$ where e_i is the i-th standard unit vector.

Since all vectors are of the same length, the sampling probabilities are the same, i.e. $p_i = \frac{1}{n}$.

Now, following L17, if we sample $k = cn$ iterations and set $w_i := w_i + \frac{1}{c}$ for some constant c.

The resulting solution is equivalent to throwing cn balls into n bins.

This is a basic stochastic process and it is well known that if cn balls are thrown, the maximum load is $\Omega(\log n / \log\log n)$ while there is a constant fraction of empty bins.

So, the resulting solution is very unbalanced and not a good approximation to the identity at all.

To get a linear-size sparsifier, we should pick the vectors more carefully based on the partial solution we have so far (e.g. updating the sampling probability).

We will come back to this point later.

Potential functions

The approach by Batson, Spielman and Srivastava is deterministic.

The algorithm adds one edge at a time, and use a potential function to guide the

choice of the next edge.

The beauty of the algebraic formulation is that we only have two parameters to keep track of, the maximum eigenvalue and the minimum eigenvalue of our solution.

In some sense, they correspond to the most used cut and the least used cut in the graph problem, but it will be clear that generalizing to $\mathbb{R}^n$ allow us to reason about them elegantly.

We add one edge by one edge. Let A be the current matrix so far.

One natural attempt is to bound the maximum eigenvalue $\lambda_{max}(A)$, and make sure that it won't increase too quickly, e.g. to prove inductively $\lambda_{max}(A_{i+1}) \leq \lambda_{max}(A_i) + \varepsilon$.

This way of measuring progress does not work well for this problem, as the matrix is n-dimensional, just focusing on the maximum direction can't distinguish the case where every direction is large or where one direction is large and all other (orthogonal) directions are small.

Ideally, we hope to say something like after n iterations, every direction is increased one unit (think of the balls and bins example). To prove it inductively, we want a potential function to let us argue that the maximum direction is increased by $\frac{1}{n}$ unit per edge, but the maximum eigenvalue won't do this for us, since in worst case it increases by one unit.

So, we want a more holistic parameter that will take into consideration of all directions.

An analogy is like in the multiplicative update method we keep track of the progress of all experts (in an appropriately weighted way), instead of just focusing on the best expert so far.

One possible parameter of this kind is $Tr(A)/n$, the average eigenvalue. For this, we can easily argue that it increases slowly, but the problem is that we cannot conclude that the maximum eigenvalue is small by using that the average is small.

What about multiplicative update (by giving more weight to the best expert)?

I think it would correspond to considering the function $\log Tr(e^A)$, sum of the exponential of the eigenvalues and then take log.

I think it would work to give a deterministic algorithm to recover the $O(n \log n/\varepsilon^2)$

sparsifier in [17], and we have to lose a factor of $\log n$ because we can't distinguish the case when there is only one large eigenvalue or all eigenvalues are large.

I am not sure about this paragraph as I haven't worked out the details.

The key idea in BSS is to use the potential function $\Phi(A) = \mathrm{Tr}(uI-A)^{-1} = \sum_{i=1}^{n} \frac{1}{u-\lambda_i}$, where u is intended to be an upper bound of the maximum eigenvalue and $\lambda_1, \ldots, \lambda_n$ are the eigenvalues of A.

They maintain the invariant that $\Phi(A)$ is upper bounded, by increasing u slowly over time.

This ensures that the maximum eigenvalue will never get close to u, as otherwise the potential function will blow up and would not be bounded.

This is very similar to the use of barrier functions in interior point method, in which a constrained optimization problem is reduced to an unconstrained one using a barrier function such as $\log(A_i X - b_i)$, while guaranteeing that the constraints will not be violated (as otherwise the objective value is unbounded and will not be an optimal solution to the unconstrained problem).

Suppose we guarantee that $\Phi(A) = \sum_{i=1}^{n} \frac{1}{u-\lambda_i} \le 1$. This will ensure that at most one eigenvalue $\ge n-1$, at most two eigenvalues $\ge n-2$, etc. This gives a more holistic measure of the solution.

Outline

To give you a more concrete idea, we present the algorithm in a simple (but non-optimal) setting of parameters.

Initially, $A = 0$, $u_0 = n$ and $l_0 = -n$.

Invariants to maintain are $\Phi_u(A) = \mathrm{Tr}(uI-A)^{-1} = \sum_{i=1}^{n} \frac{1}{u-\lambda_i} \le 1$ and $\Phi_l(A) = \mathrm{Tr}(A-lI)^{-1} = \sum_{i=1}^{n} \frac{1}{\lambda_i - l} \le 1$.

(The second is the lower barrier function to lower bound the minimum eigenvalue.)

In each iteration, we choose a vector v_i and set $A \leftarrow A + t v_i v_i^T$ for some scalar t, so that $\Phi_{u+\varepsilon_u}(A + t v_i v_i^T) \le 1$ and $\Phi_{l+\varepsilon_l}(A + t v_i v_i^T) \le 1$.

Think of both ε_u and ε_l are constants, say $\varepsilon_u = 2$ and $\varepsilon_l = \frac{1}{3}$.

If all the iterations are possible, after $6n$ iterations, the maximum eigenvalue is at most $13n$, while the minimum eigenvalue is at least n, and thus we have a spectral sparsifier with $6n$ edges that approximate all quadratic forms up to a constant factor.

Analysis

The mathematics turns out to work very nicely.

The first thing to understand is how the potential functions change upon adding a vector. We would like to know for what δ_u and t such that $\Phi_{u+\delta_u}(A + tvv^T) \le \Phi_u(A)$.

By definition, $\Phi_{u+\delta_u}(A+tvv^T) = \mathrm{Tr}\left(u'I - A - tvv^T\right)^{-1}$ where $u' = u + \delta_u$.

There is a well-known formula for updating the inverse after a rank one update.

Sherman-Morrison-Woodbury formula $\left(M - txx^T\right)^{-1} = M^{-1} + \frac{tM^{-1}xx^TM^{-1}}{1 - tx^TM^{-1}x}$ for non-singular M.

Upper barrier

$$\begin{aligned}
\text{So, } \Phi_{u+\delta_u}(A+tvv^T) &= \mathrm{Tr}\left(u'I - A - tvv^T\right)^{-1}. \\
&= \mathrm{Tr}\left(\left(u'I-A\right)^{-1} + \frac{t(u'I-A)^{-1}vv^T(u'I-A)^{-1}}{1 - t\,v^T(u'I-A)^{-1}v}\right) \\
&= \mathrm{Tr}\left(u'I-A\right)^{-1} + t\,\mathrm{Tr}\left(\frac{v^T(u'I-A)^{-2}v}{1 - t\,v^T(uI-A)^{-1}v}\right) \qquad \text{as } \mathrm{Tr}(XY) = \mathrm{Tr}(YX) \\
&= \Phi_{u'}(A) + \frac{t\,v^T(u'I-A)^{-2}v}{1 - t\,v^T(u'I-A)^{-1}v}.
\end{aligned}$$

$$\begin{aligned}
\text{Therefore, } \Phi_{u'}(A+tvv^T) \le \Phi_u(A) &\iff \Phi_{u'}(A) - \Phi_u(A) + \frac{v^T(u'I-A)^{-2}v}{\frac{1}{t} - v^T(u'I-A)^{-1}v} \le 0 \\
&\iff \frac{1}{t} \ge \frac{v^T(u'I-A)^{-2}v}{\Phi_u(A) - \Phi_{u'}(A)} + v^T(u'I-A)^{-1}v \qquad (*)
\end{aligned}$$

So, for any v, we can increase t a bit without violating the upper barrier - which ensures that the maximum eigenvalue after adding tvv^T is at most $u+\delta_u$ as long as $\Phi_u(A)$ is bounded.

Lower barrier

This is mostly similar, with some subtle difference.

$$\begin{aligned}
\Phi_{l+\delta_l}(A+tvv^T) &= \mathrm{Tr}\left(A + tvv^T - l'I\right)^{-1} \\
&= \mathrm{Tr}\left(\left(A-l'I\right)^{-1} - \frac{t(A-l'I)^{-1}vv^T(A-l'I)^{-1}}{1 + t\,v^T(A-l'I)^{-1}v}\right) \\
&= \Phi_{l'}(A) - \frac{t\,v^T(A-l'I)^{-2}v}{1 + t\,v^T(A-l'I)^{-1}v}.
\end{aligned}$$

$$\begin{aligned}
\text{So, } \Phi_{l+\delta_l}(A+tvv^T) \le \Phi_l(A) &\iff \Phi_{l'}(A) - \Phi_l(A) - \frac{v^T(A-l'I)^{-2}v}{\frac{1}{t} + v^T(A-l'I)^{-1}v} \le 0 \\
&\iff \frac{1}{t} \le \frac{v^T(A-l'I)^{-2}v}{\Phi_{l'}(A) - \Phi_l(A)} - v^T(A-l'I)^{-1}v. \qquad (**)
\end{aligned}$$

The RHS could be negative, and so for some v, it may not be possible to find a $t \ge 0$ so that the lower barrier function doesn't increase.

Both barriers

We would like to find a v_i so that there exists t to satisfy both (*) and (**) simultaneously. The important idea here is to show that the sum of RHS of (**) is at least the sum of RHS of (*), and this would imply there exists a specific v_i with the RHS of (**) at least the RHS of (*), and so we can choose t for that v_i so that both potential functions don't increase.

Let's first consider the sum of RHS of (*), which is

$$\sum_i \left(\frac{v_i^T (u'I - A)^{-2} v_i}{\Phi_u(A) - \Phi_{u'}(A)} + v_i^T (u'I - A)^{-1} v_i \right)$$

$$= \sum_i \left(\frac{\mathrm{Tr}\left((u'I - A)^{-2} v_i v_i^T \right)}{\Phi_u(A) - \Phi_{u'}(A)} + \mathrm{Tr}\left((u'I - A)^{-1} v_i v_i^T \right) \right)$$

$$= \frac{\mathrm{Tr}\left((u'I - A)^{-2} \sum_i v_i v_i^T \right)}{\Phi_u(A) - \Phi_{u'}(A)} + \mathrm{Tr}\left((u'I - A)^{-1} \sum_i v_i v_i^T \right) \quad \text{as trace is linear}$$

$$= \frac{\mathrm{Tr}\left(u'I - A \right)^{-2}}{\Phi_u(A) - \Phi_{u'}(A)} + \mathrm{Tr}\left(u'I - A \right)^{-1} \quad \text{as } \sum_i v_i v_i^T = I \text{ by the assumption of the theorem}$$

$$= \frac{\sum_j \frac{1}{(u + \delta_u - \lambda_j)^2}}{\sum_j \frac{1}{u - \lambda_j} - \sum_j \frac{1}{u + \delta_u - \lambda_j}} + \Phi_{u'}(A)$$

$$= \frac{\sum_j (u + \delta_u - \lambda_j)^{-2}}{\sum_j \delta_u (u - \lambda_j)^{-1} (u + \delta_u - \lambda_j)^{-1}} + \Phi_{u'}(A)$$

$$< \frac{1}{\delta_u} + \Phi_u(A) \quad \text{as } \Phi_u(A) > \Phi_{u'}(A)$$

The sum of the RHS of (**) is similar but more involved:

$$\sum_i \left(\frac{v_i^T (A - \ell' I)^{-2} v_i}{\Phi_{\ell'}(A) - \Phi_\ell(A)} - v_i^T (A - \ell' I)^{-1} v_i \right)$$

$$= \frac{\mathrm{Tr}\left(A - \ell' I \right)^{-2}}{\Phi_{\ell'}(A) - \Phi_\ell(A)} - \mathrm{Tr}\left(A - \ell' I \right)^{-1} \quad \text{using trace is linear and } \sum_i v_i v_i^T = I$$

$$= \frac{\sum_j \left(\lambda_j - \ell - \delta_\ell \right)^{-2}}{\sum_j (\lambda_j - \ell - \delta_\ell)^{-1} - \sum_j (\lambda_j - \ell)^{-1}} - \sum_j \left(\lambda_j - \ell - \delta_\ell \right)^{-1}$$

$$\geq \frac{1}{\delta_\ell} - \Phi_\ell(A) \quad \text{hiding the calculations in Claim 3.6 of [BSS].}$$

Therefore, if we choose δ_u, δ_ℓ and set the initial potentials $\Phi_u(A), \Phi_\ell(A)$ in such a way that $\frac{1}{\delta_\ell} - \Phi_\ell(A) \geq \frac{1}{\delta_u} + \Phi_u(A)$, then we can choose some v_i and t such that $\Phi_{\ell'}(A + t v_i v_i^T) \leq \Phi_\ell(A)$ and $\Phi_{u'}(A + t v_i v_i^T) \leq \Phi_u(A)$, and thus the condition will

continue to satisfy, and we can repeat as many times as we want.

Setting the parameters

Set $\ell_0 = -\sqrt{d}\,n$ and $u_0 = \left(\frac{d+\sqrt{d}}{\sqrt{d}-1}\right)n$ so that initially $\Phi_{\ell_0}(A) = \frac{1}{\sqrt{d}}$ and $\Phi_{u_0}(A) = \frac{\sqrt{d}-1}{d+\sqrt{d}}$.

Set $\delta_L = 1$ and $\delta_U = \frac{\sqrt{d}+1}{\sqrt{d}-1}$ so that the condition $\frac{1}{\delta_\ell} - \Phi_{\ell_0}(A) \geq \frac{1}{\delta_u} + \Phi_{u_0}(A)$ holds.

Therefore, after dn steps, $\frac{\lambda_{\max}(A)}{\lambda_{\min}(A)} \leq \frac{\Phi_{u_0}(A) + dn\,\delta_u}{\Phi_{\ell_0}(A) + dn\,\delta_\ell} = \frac{d+2\sqrt{d}+1}{d-2\sqrt{d}+1}$.

New developments

One open question is how to compute a linear-size spectral sparsifier in near-linear time.

Directly implementing the above algorithm would take $\Omega(n^4)$ time, making it not useful in algorithmic applications, despite being an amazing mathematical result.

Recently, there is significant progress in resolving this question.

One paper interprets BSS using the (matrix) multiplicative update framework, and use this understanding to design a $\widetilde{O}(n^2)$-time algorithm.

More recently, Lee and Sun designed an almost-linear time algorithm for constructing a linear-size spectral sparsifier.

It takes $\widetilde{O}(qmn^{\frac{1}{q}})$ time to construct a spectral sparsifier with $O(qn/\varepsilon^2)$ edges.

The idea is quite neat, to combine the random sampling algorithm with the potential function algorithm.

The intuition is from balls and bins.

If we only put say $n^{0.99}$ balls into n bins, then with high probability the maximum load is only a constant.

Using matrix Chernoff bound, they show that if we only sample $n^{0.99}$ edges by effective resistance, the added edges satisfy $\lesssim \frac{1}{2}(uI - A)$.

Only after these edges are added, they update the sampling probability by setting the probability of v_i proportional to $v_i^T(uI-A)^{-1}v_i + v_i^T(A-\ell I)^{-1}v_i$, i.e. more conservative if the direction is close to boundary.

Therefore, it only takes $n^{0.01}$ iterations of random sampling, and hence the runtime.

This is only the main idea. There are also some technical differences, a more notable one is that they use the potential function $\mathrm{Tr}(uI-A)^{-g}$ for some larger g.

References

- Twice-Ramanujan sparsifiers (http://arxiv.org/abs/0808.0163), by Batson, Spielman, Srivastava, 2009.
- Spectral sparsification and regret minimization beyond matrix multiplicative updates (http://arxiv.org/abs/1506.04838), by Allen-Zhu, Liao, Orecchia, 2015.
- Constructing linear-sized spectral sparsification in almost-linear time (http://arxiv.org/abs/1508.03261), by Lee and Sun, 2015.