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

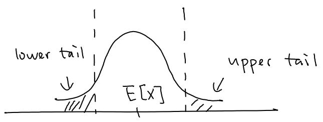
Lecture 16: Graph sparsification

We start the last part of the course with the first result in cut-preserving sparsifier, along with the proof of Chernoff bounds and Karger's randomized min-cut algorithm.

Concentration inequalities

On a high level, tail inequalities or concentration inequalies give upper bounds on the probability that the value of a random variable is far from its expected value, and these allow us to show that randomized algorithms behave like what we expect with high probability.

Sum of independent variables



The general question is to bound $Pr(X > (1+\varepsilon)E[X])$ (upper tail) and $Pr(X < (1-\varepsilon)E[X])$ (lower tail).

We consider the situation when X is the sum of many independent random variables, which is commonly seen in the analysis of randomized algorithms.

The law of large number asserts that the sum of n independent identically distributed variables is approximately $n\mu$, where μ is a typical mean.

The central limit theorem says that $\frac{X - n\mu}{\sqrt{n\sigma^2}} \rightarrow N(0,1)$. the deviations from $n\mu$ are typically of the order $\sqrt{n}$.

Chernoff bounds give us quantitative estimates of the probabilities that X is far from E[X] for any (large enough) value of n.

Consider a simple setting where there are n coin flips, each is head with probability p.

The expected number of heads is np.

To bound the upper tail, in principle we just need to compute $Pr(X \geq k) = \sum_{i \geq k} \binom{n}{i} p^i (1-p)^{n-i}$, and show that it is very small when k is much larger than np (say $k \geq (1+\varepsilon)np$), but this sum is not easy to work with and this method is not easy to be generalized.

Instead, we extend the approach of using Markov's inequality. The Markov's inequality is often too weak, but recall in the proof of Chebyshev's inequality we can strengthen it if we know the second moment of X.

To extend this, one can use the fourth moment or any 2k-th moment to get (why even?)

$$\Pr(|X-E[X]|>a) = \Pr((X-E[X])^{2k} > a^{2k}) \le E[(X-E[X])^{2k}]/a^{2k}$$

The idea in proving the Chernoff bounds is to consider:

$$\Pr(X \ge a) = \Pr(e^{tX} \ge e^{ta}) \le E[e^{tX}]/e^{ta} \quad \text{for any } t>0.$$

There are at least two reasons that we consider e^{tX}:

- Let $M_X(t) = E[e^{tX}] = E[\sum_{i\ge 0} \frac{t^i}{i!} X^i] = \sum_{i \ge 0} \frac{t^i}{i!} E[X^i]$ If we have $M_X(t)$, to compute $E[X^i]$, we can just compute $M_X^{(k)}(0)$, where $M_X^{(k)}(0)$ is the k-th derivative of $M_X(t)$ evaluated at $t=0$.

  So, $M_X(t)$ contains all the moments information, and is called the moment generating function.

  It gives a strong bound when applying Markov's inequality, as the denominator is exponentially large.
- If $X = X_1 + X_2$ and X_1, X_2 are independent, then $E[e^{tX}] = E[e^{tX_1} e^{tX_2}] = E[e^{tX_1}]\,E[e^{tX_2}]$.

  So, this function is easy to compute when X is the sum of independent random variables.

Chernoff Bounds

Roughly speaking, Chernoff-type bounds are the bounds obtained by $\Pr(X \ge a) \le E[e^{tX}]/e^{ta}$.

Let us consider a useful case when X is the sum of independent heterogenous coin flips.

Heterogenous coin flips:

Let $X_1, \dots, X_n$ be independent random variables with $X_i = 1$ with probability p_i and $X_i = 0$ otherwise.

Let $X = \sum_{i=1}^n X_i$. Let $\mu = E[X] = \sum_{i=1}^n E[X_i] = \sum_{i=1}^n p_i$ be the expected value.

Then $E[e^{tX}] = E[e^{tX_1} e^{tX_2} \cdots e^{tX_n}] = \prod_{i=1}^n E[e^{tX_i}]$ by independence

$$= \prod_{i=1}^n (p_i e^{t\cdot 1} + (1-p_i) e^{t\cdot 0}) = \prod_{i=1}^n (1 + p_i(e^t - 1)) \le \prod_{i=1}^n e^{p_i(e^t-1)} = e^{\mu(e^t-1)}.$$

We put in some specific parameters to get some useful bounds.

Theorem. In the heterogenous coin flipping setting, we have:

① for $\delta > 0$, $\Pr(X \ge (1+\delta)\mu) < \left(\frac{e^\delta}{(1+\delta)^{1+\delta}}\right)^\mu$.

② for $0 < \delta < 1$, $\Pr(X \ge (1+\delta)\mu) < e^{-\delta^2 \mu/3}$.

③ for $R \ge 6\mu$, $\Pr(X \ge R) \le 2^{-R}$.

③ for $R \geq 6\mu$, $\Pr(X \geq R) \leq 2^{-R}$.

<u>proof</u> ① $\Pr(X \geq (1+\delta)\mu) \leq E[e^{tX}]/e^{t(1+\delta)\mu} \leq e^{\mu(e^t-1)}/e^{t(1+\delta)\mu}$

By elementary calculus, we find out that this term is minimized when $t = \ln(1+\delta)$, and

this implies that $\Pr(X \geq (1+\delta)\mu) \leq e^{\mu\delta}/(1+\delta)^{(1+\delta)\mu}$, proving ①.

② When $0<\delta<1$, it holds that $e^{\delta}/(1+\delta)^{1+\delta} \leq e^{-\delta^2/3}$.

This can be verified by taking log of both sides and letting $f(\delta) = \delta - (1+\delta)\ln(1+\delta) + \frac{\delta^2}{3}$,

and show that $f'(\delta) \leq 0$ in the interval $[0,1]$, and thus $f(\delta) \leq 0$ in this interval

since $f(0)=0$, and this implies the claim. (see MU Theorem 4.4 for details.)

③ Let $R = (1+\delta)\mu$. When $R \geq 6\mu$, we have $\delta \geq 5$.

Hence, $\Pr(X \geq (1+\delta)\mu) \leq (e^{\delta}/(1+\delta)^{1+\delta})^{\mu} \leq (e/(1+\delta))^{(1+\delta)\mu} \leq (e/6)^R \leq 2^{-R}$. □

Similar bounds hold for the lower tail; very similar proof (by setting $t<0$). (see MU Thm 4.5)

<u>Theorem</u> In the heterogenous coin flipping setting, we have for $0<\delta<1$

① $\Pr(X \leq (1-\delta)\mu) \leq (e^{-\delta}/(1-\delta)^{1-\delta})^{\mu}$

② $\Pr(X \leq (1-\delta)\mu) \leq e^{-\mu\delta^2/2}$.

<u>Corollary</u> In the heterogenous coin flipping setting, $\Pr(|X-\mu| \geq \delta\mu) \leq 2e^{-\mu\delta^2/3}$ for $0<\delta<1$.

<u>Hoeffding extension</u> The same bounds hold when each X_i is a random variable taking values in $[0,1]$ with mean p_i. This is because the function e^{tx} is convex, and thus it always lies below the straight line joining the endpoints $(0,1)$ and $(1,e^t)$. This line has the equation $y = \alpha x + \beta$ for $\alpha = e^t - 1$ and $\beta = 1$. Therefore, $E[e^{tX_i}] \leq E[\alpha X_i + \beta] = p_i(\alpha+\beta) + (1-p_i)\beta = 1 + p_i(e^t-1)$, and the same calculations as above follow.

<u>Remarks</u>:

- The same method holds for other random variables, e.g. Poisson r.v., Gaussian r.v., etc.
- It is often an easier way to compute the moments by computing the moment generating functions.
- Chernoff bounds also hold for negatively correlated variables, because $E[e^{t(X+Y)}] \leq E[e^{tX}]E[e^{tY}]$ and then the same proof works. and this observation is very useful in some applications.

Graph Sparsification

Given an edge weighted undirected graph $G=(V,E,w)$, for a subset of vertices $S \subseteq V$,

let $\delta_G(S)$ be the set of edges with one endpoint in S and another endpoint in $V-S$, and

let $w(\delta_G(S))$ be the total weight of the edges in $\delta_G(S)$.

We say H is a $(1\pm\varepsilon)$-cut-approximator of G if $(1-\varepsilon)w(\delta_G(S)) \le w(\delta_H(S)) \le (1+\varepsilon)w(\delta_G(S))$

for all $S \subseteq V$. Note that H is on the same vertex set but may have different edge weights.

We are interested in finding a $(1\pm\varepsilon)$-cut-approximator of G that is sparse (having few edges).

Assumption: We consider a simple setting in which G is unweighted and has min-cut value $\Omega(\log n)$.

Algorithm: In this simple setting, the algorithm is very simple. Set a sampling probability p.

For every edge $e \in E(G)$, put it in H with weight $\frac{1}{p}$ with probability p.

Theorem Set $p = 9\ln n/(\varepsilon^2 c)$ where c is the min-cut value of G.

Then H is a $(1\pm\varepsilon)$-cut-approximator of G with $O(p\cdot|E(G)|)$ edges with prob $\ge 1-\frac{4}{n}$.

proof Consider a subset $S\subseteq V$. Say $\delta_G(S)$ has k edges. Note that $k \ge c$ by definition.

By linearity of expectation, $E[|\delta_H(S)|] = pk$ and thus $E[w(\delta_H(S))] = p\cdot k\cdot\frac{1}{p} = k = |\delta_G(S)|$.

Since each edge is an independent 0-1 random variable, by Cheroff, we have

$$\Pr[\,\big|\,|\delta_H(S)| - pk\,\big| > \varepsilon pk\,) \le 2e^{-pk\varepsilon^2/3} = 2e^{-(\frac{9\ln n}{\varepsilon^2 c})(\frac{k\varepsilon^2}{3})} = 2e^{-\frac{3k\ln n}{c}}$$

Since $k \ge c$ by definition, this implies that this probability is at most $2/n^3$.

So, the probability that one cut is violated is pretty small but there are exponentially many cuts.

The important observation here is that the probability that a large cut is violated is much smaller, and there are not too many small cuts.

Claim: The number of cuts with at most αc edges for $\alpha \ge 1$ is at most $n^{2\alpha}$.

proof: It will follow from the analysis of the randomized min-cut algorithm that we will describe in the next section.

Now, with the claim. we can bound

$\Pr($ some cut S is violated$)$

$$\le \sum_{S \subseteq V} \Pr(\text{cut } S \text{ is violated}) \quad (\text{this is called the union bound})$$

$$\le \sum_{\alpha \ge 1} \sum_{S \subseteq V : |\delta_G(S)| \le \alpha c} \Pr(\text{cut } S \text{ is violated})$$

$$\le \sum_{\alpha \ge 1} n^{2\alpha} \Pr(\text{cut } S \text{ is violated} \mid |\delta_G(S)| \le \alpha c)$$

$$\le \sum_{\alpha \ge 1} n^{2\alpha} 2 e^{-3\alpha \ln n}$$

$$= \sum_{\alpha \ge 1} 2n^{-\alpha} \le 4/n$$

It is easy to see that H has $O(p \cdot |E(G)|)$ edges with high probability, and the theorem follows. $\square$

<u>Applications</u>: We briefly sketch some applications of the theorem.

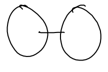
① Approximate min s-t cut: Given G, fix ε (say to 0.01), construct H with $\tilde{O}(\frac{m}{c})$ edges.

Solve min s-t cut in H. The same cut is a $(1+\varepsilon)$-approx solution in G.

② Approximate max s-t flow: Randomly color the edges by $\frac{1}{p}$ colors.

Each color subgraph has $O(p|E(G)|)$ edges, and the max s-t flow in each subgraph is at least $(1-\varepsilon)p \cdot OPT$ where OPT is the max s-t flow value in G, as each color subgraph has the same distribution as a random sampled subgraph of G.

Solve max s-t flow in each subgraph and combine the solutions to get a $(1-\varepsilon)$-approximate max s-t flow in G.

<u>Improvements</u>: Without the minimum cut assumption, then it is easy to see that a uniform sampling algorithm won't work, e.g. in , it is very likely that the cut edge is not picked.

Benczur and Karger designed a very clever non-uniform sampling algorithm, where the sampling probability for each edge is inversely proportional to the "connectivity" of the two endpoints.

They defined a notion called "strong connectivity" and proved that sampling inversely proportional to it will result in a $(1 \pm \varepsilon)$-cut-approximator with $O(n \log n)$ edges.

Furthermore, they showed that there is an almost linear-time algorithm to estimate the strong connectivity which is good enough for the purpose, leading to an $\tilde{O}(n^2)$-algorithm for approx min s-t cut.

The definition of "strong connectivity" is a bit unnatural, and they conjectured that it can be replaced by the more natural edge-connectivity (i.e. the max u-v flow value for edge uv).

This Conjecture is proven by Fung, Hariharan, Harvey and Panigrahi in 2011.

Randomized minimum cut

In the minimum cut problem, we are given an unweighted undirected graph $G=(V,E)$, and our objective is to find a minimum subset of edges $F \subseteq E$ so that $G-F$ is disconnected.

A simple observation is that a minimum cut is also a min s-t cut for some s,t. So, one can compute min s-t cut for all s,t and take a minimum one, and this naive algorithm requires n^2 calls of maximum flow.

Then, one observes that it is enough to just fix an arbitrary vertex as s and just compute min s-t cut for all possible t, and this gives an algorithm in n maxflow time.

For a while, it seems that this global min-cut problem is more difficult than min s-t cut.

Then, Matula (1987) gave a very nice algorithm that computes min-cut in $O(mn)$ time.

His new idea is to do a graph search and identify an edge that can be contracted and repeat.

Karger came up with the idea of random contraction and improved the runtime to $\tilde{O}(n^2)$, which is faster than computing min s-t cut. Eventually he gave a near-linear time $\tilde{O}(m)$ algorithm for the minimum cut problem, which is very interesting and surprising.

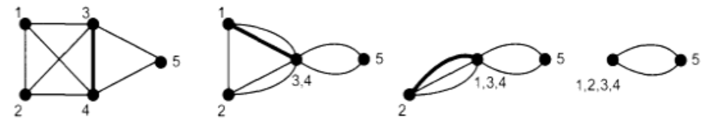
We will present an $O(n^4)$-time algorithm and mention how it can be improved to $\tilde{O}(n^2)$.

The <u>algorithm</u> is very simple.

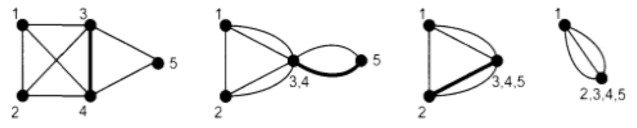
```
While there are more than two vertices in the graph
    Pick a random edge and contract the two endpoints
Output the edges between the remaining vertices.
```

By "contracting" an edge, we mean to identify the two endpoints as a single vertex.

See the picture below from the book by Mitzenmacher and Upfal.



(a) A successful run of min-cut.



(b) An unsuccessful run of min-cut.

Observation : Each vertex in an intermediate graph is a subset of vertices in the original graph.

So, each cut in an intermediate graph corresponds to a cut in the original graph.

Hence, a min-cut in an intermediate graph is at least a min-cut in the original graph.

Theorem The probability that the algorithm outputs a minimum cut is at least $2/n(n-1)$.

Proof Let F be a minimum cut and let $k = |F|$ be the number of edges in F.

If we never contract an edge in F until the algorithm ends, then the algorithm succeeds.

What is the probability that an edge in F is contracted in the i-th iteration?

By the observation, the min-cut value in the i-th iteration is at least k.

Note that it implies that every vertex is of degree at least k, as otherwise we can disconnect a single vertex from the graph by removing less than k edges.

Therefore, the number of edges in the i-th iteration is at least $(n-i+1)\cdot k/2$.

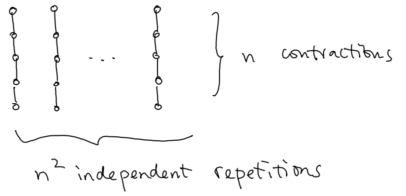
Since we pick a random edge to contract, the probability that we pick an edge in F is at most $k/\left((n-i+1)k/2\right) = \frac{2}{n-i+1}$.

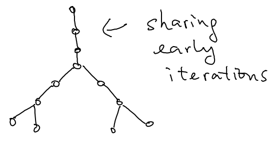
So, the probability that F survives until the end is at least

$$\left(1-\frac{2}{n}\right)\cdot\left(1-\frac{2}{n-1}\right)\cdots\left(1-\frac{2}{3}\right) = \frac{\cancel{n-2}}{n}\cdot\frac{\cancel{n-3}}{n-1}\cdot\frac{\cancel{n-4}}{\cancel{n-2}}\cdots\frac{2}{\cancel{4}}\cdot\frac{1}{\cancel{3}} = \frac{2}{n(n-1)}.\ \square$$

Improving success probability : By repeating the whole procedure t times, the failure probability is at most $\left(1-\frac{2}{n(n-1)}\right)^t$. By setting $t = 100n(n-1)$, this is at most e^{-200} which is tiny, using the inequality $1-x \le e^{-x}$.

Running time : One execution can be implemented in $O(n^2)$ time, and the total runtime is $O(n^4)$.

Improving running time : The observation is that the failure probability is only large near the end of one executation, while it is very small in the beginning. So, the idea is that we only repeat the later iterations but not the early iterations. Pictorically, the $O(n^4)$ algorithm is like 

, while it is much better to do 

<u>Combinatorial Structure:</u> An interesting corollary of the theorem is that there are at most $O(n^2)$ min-cuts in an undirected graph, because each min-cut survives with probability $\Omega(\frac{1}{n^2})$ and the events that two different min-cut survive are disjoint. This is a non-trivial statement to prove using other arguments. It is an exercise to prove the claim used for graph sparsification.

<u>Open questions</u> Can you design a fast algorithm for computing global vertex connectivity?

References

- Probability and computing, by Mitzenmacher and Upfal (for Chernoff bounds and randomized min-cut).
- Random sampling in cut, flow, and network design problems, by Karger.
- Randomized approximation schemes for cuts and flows in capacitated graphs, by Benczur and Karger.