18.409 An Algorithmist's Toolkit 10/8/2009

Lecture 9

Lecturer: Jonathan Kelner

At the end of the previous lecture, we began to motivate a technique called Sparsification. In this lecture, we describe sparsifiers and their use, and give an overview of Combinatorial and Spectral Sparsifiers. We also define Spectral Sparsifiers, and create tools and language with which to construct and analyze them.

1 Sparsification

Suppose we are given a graph $G = (V, E)$. We would like to solve some cut problem (i.e. min-cut, s-t min cut, sparsest cut) and so on. The running time of algorithms for these problems typically depends on the number of edges in the graph, which might be as high as $O(n^2)$. Is there any way to approximate our graph with a sparse graph G' in which all cuts are approximately of the same size?

We will describe two ways of "sparsifying" our graph. The first is the method of Benczur-Karger, and relies on random sampling of edges. The second technique is Spectral Sparsification, and uses spectral techniques to improve upon Benczur-Karger's algorithm.

1.1 First Try

Our first attempt at sparsifying will use random sampling. Let's start by sampling each edge with probability p. Then, if a cut has c edges crossing it in G, the expected value of edges crossing it in the new graph G' is pc. Our algorithm will solve the cut problem in G'. Say the answer is a cut with value S'; then our algorithm will output the estimate $S = S'/p$ for the original graph G.

Denoting the number of edges between S and $\bar{S}$ by $e(S) = pc$, we have the following concentration result due to Chernoff's inequality:

$$P(|e_{G'}(S) - pc| \geq \epsilon pc) \geq e^{-\epsilon^2 pc/2}. \tag{1}$$

So our result will be close to the correct answer provided pc is large. In particular, picking

$$p = \Omega(\frac{d \log n}{\epsilon^2 c}),$$

will make the right side of Eq. (1) at most n^{-d}. Summarizing, we can choose p to get an ϵ multiplicative approximation with probability at least $1 - n^{-d}$.

Is it possible to choose p to get this multiplicative approximation for *all* cuts, rather than just one as above? The answer is yes; the main ingredient is a result of Karger that the number of small cuts in a graph is not too large:

Theorem 1 (Karger) *If G has a min-cut of size c, then the number of cuts of value αc or less is at most $n^{2\alpha}$.*

1.2 Second try

The problem with this proposal is that it breaks for small cuts. Say c is small, but an edge e is only involved in cuts of size $\geq k$. What we want to do is to sample these edges with a small probability of failure.

The idea that we use is to sample edges, but with a "weight" of $1/p$. This method is called importance sampling. To do this, we need a slightly modified version of the Chernoff bound:

Theorem 2 (Chernoff Bound) *Let $X_1, \ldots, X_n$ be random variables so that $X_i \in [0,1]$, and let $X = \sum X_i$. Then,*

$$Pr[|X - E[X]| \geq \epsilon X] \leq 2e^{-\Theta(1)\epsilon^2 E[X]}$$

Proof The only difference here is that the random variables X_i are no longer discrete variables, but lie in the interval $[0,1]$. The proof is carried out the same as with the regular Chernoff bound. ■

What this allows us to do is to scale our random variables without changing the error bounds. Returning to our case, we assign to every edge e a random variable Y_e and a weight w_e. If e is in a cut of size c, we require that $w_e \leq c$. We will set $Y_e = 1$ with probability p/w_e; and $Y_e = 0$ with probability $1 - p/w_e$. Instead of counting how many edges cross a cut $(S, \bar{S})$, we will compute a weighted sum:

$$Y_S = \sum_{e \in \partial(S,\bar{S})} w_e Y_e$$

The expectation is still correct; if there are c edges across the cut $(S, \bar{S})$ in G, then

$$E[Y_S] = \sum_{e \in \partial(S,\bar{S})} w_e \frac{p}{w_e} = pc.$$

This scheme gives us an advantage: if an edge is present in only cuts of large size, we can keep it with low probability, which corresponds to setting w_e to be large. On the other hand, if an edge is present in cuts of small size, we will keep it with high probability, which corresponds to setting w_e to be small. In this way, we can approximate cut problems while throwing away more edges which are present in only cuts of high size.

Thus, a natural choice for w_e would be the size of the smallest cut containing e. Unfortunately, we do not know w_e; however, it is possible to approximate it quickly. The final result is an ϵ multiplicative approximation based on this scheme. We refer the reader to [1] for details.

2 Spectral Sparsifiers

The construction shown above is known as a *Combinatorial Sparsifier*. In the upcoming section and following lecture, we will see how to improve upon it with the spectral methods that we have been learning.

Let $G = (V, E)$ be our original graph. Recall that the laplacian has the property that

$$x^T L_G x = \sum_{(i,j) \in E} (x_i - x_j)^2,$$

for some $x \in \mathbb{R}^n$, and the sum is being taken over all edges in G. If x takes value 1 on the set S and -1 on the $\bar{S}$, this equation becomes

$$x^T L_G x = 4e(S).$$

Let G' be a combinatorial sparsifier of the graph G. The condition that all cuts in G are approximated with a multiplicative error of at most ϵ by cuts in G' can be restated as

$$(1-\epsilon) x^T L_{G'} x \leq x^T L_G x \leq (1+\epsilon) x^T L_{G'} x, \tag{2}$$

for all x that take on only the values 1 and -1. This is true for all such discrete values of x.

On the other hand, consider if Eq. (2) is true for all $x \in R^n$. Note that in this case we can limit ourselves to the instances $x \in [-1,1]^n$ by normalization. We now have a good definition for a spectral version of sparsification:

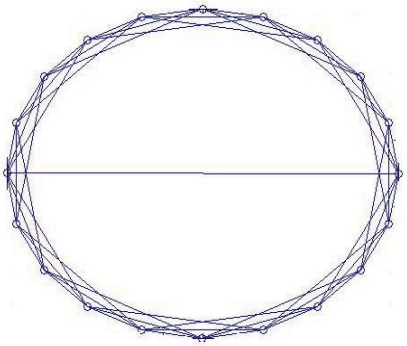
Definition 3 *A **Spectral Sparsifier** G' of a graph G is one for which the relation*

$$(1-\epsilon) x^T L_{G'} \leq x^T L_G x \leq (1+\epsilon) x^T L_{G'} x$$

for all $x \in [0,1]^n$

It is clear from this definition that spectral sparsifiers are combinatorial sparsifiers. A natural question is then to ask if all combinatorial sparsifiers also spectral sparsifiers.

The answer is no, and we provide a proof by counterexample. Consider the graph G' with vertex set $\{1, 2, \ldots, n\}$ and an edge between i, j when $i - j \bmod n \leq k$. G is G' with the edge $(1, n/2)$ added. The graph looks something like the figure below.



Then, for an appropriate ϵ, G' is a combinatorial sparsifier of G. Indeed, the min cut in G cuts $\Theta(k)$ edges; the min cut in G' cuts one less. With $\epsilon = \Theta(1/k)$, we have that G' is a combinatorial sparsifier of G.

On the other hand, G' is not a spectral sparsifier of G. Let

$$x = \begin{pmatrix} 0 & 1 & \ldots & n/2-1 & n/2-1 & \ldots & 1 & 0 \end{pmatrix}.$$

Then, we have that

$$x^T L_{G'} x = \Theta(nk^3)$$

since each vertex contributes $\Theta(\sum_{i=1}^{k} k^2)$ to the sum. On the other hand,

$$x^T L_G x = \Theta(nk^3) + (\frac{n}{2} - 1)^2$$

If k is constant, we get that we need $\epsilon = \Theta(1/n)$ for G' to be a spectral sparsifier of G.

2.1 Order Relations on Laplacians

In order to define spectral approximations, we first need to define the appropriate vocabulary. Earlier, we made error approximations based on cut size. In the spectral case, we will be using the laplacian of the graph instead - so a nice way to compare laplacians would be idea. That is to say, we want a good relation $\succeq$ on symmetric matrices that is an ordering on them, and also is somewhat consistent with the notions of cuts.

How will we define this ordering? An immediate idea is the following:

$$M \succeq N \Leftrightarrow m_{i,j} \geq n_{i,j} \forall i, j$$

Upon second thought, we realize that this is no good for our purposes. For one, spectral graph theory is all about eigenvalues, and this relation tells us nothing about the eigenvalues of the matrix! Furthemore, the values of individual entries are highly dependent on choice of basis, which would be bad. If such a definition were used, a process like diagonalizing the Laplacians could possibly affect the graph orders.

We try again with another definition:

$$M \succeq N \text{ if the } i^{th} \text{ eigenvalue of } M \text{ is } \geq \text{ the } i^{th} \text{ eigenvalue of } N \text{ for all indices } i$$

This is better in that it is basis independent - but it is too basis independent. Under this definition, we have both

$$\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \succeq \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

as well as

$$\frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \succeq \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

After this experimentation, we claim that the following is the "right" definition of order.

Definition 4 *We write that $M \succeq N$ if*

$$x^T M x \geq X^T N x \ \forall x \in \mathbb{R}^n$$

Note that this definition of order has the following properties:

1. If $M \succeq N$ and $N \succeq M$, then $M = n$
2. $M \succeq 0$ if M is a positive semidefinite matrix.
3. $M \succeq N$ if $M - N$ is positive semidefinite
4. If $M_1 \succeq N_1$ and $M_2 \succeq N_2$, then
$$M_1 + M_2 \succeq N_1 + N_2$$

These properties suffice for our purposes, and with this, we can define an associated order on graphs as well.

Definition 5 *Given graphs G and H, say that $G \succeq H$ if $L_G \succeq L_H$.*

Claim 6 *Let $G = (V, E_G, w_G)$ and $H = (V, E_H, w_H)$ be weighted graphs on the same vertex set such that $w_G(i,j) \geq w_H(i,j)$ for all edges $(i,j) \in E$. Then, $G \succeq H$*

2.2 Towards Spectral Sparsification

With this order relation on graphs, we can now restate the goal of spectral sparsification: Given a dense graph G, we want to create a sparse graph H where

$$L_h \preceq L_G \preceq (1+\epsilon) L_H$$

By "sparse," we mean that H has polylog(n) edges, where n is the number of nodes. We will show in this and the next lecture how to construct spectral sparsifiers with $O(nlogn)$ edges in Polynomial time. This can actually be improved to a linear time construction, but will use geometric techniques that we will learn. Moreover, it is possible to construct $O(n)$ edge sparsifiers in polynomial time. The benefits of this are that the problem is more geometrically flavored. It is also a nice example of how generalizing can make things easier sometimes.

The algorithm that we propose is very simple. It is similar in structure to the B-K algorithm, but we use different probabilities for sampling the edges.

- Compute probability p_e for each edge e.
- Sample each edge uniformly with probability p_e, and if an edge is selected, include it with weight $1/p_e$.

These probabilities are based on a linear algebra sense of importance, and have a nice interpretation in terms of effective resistance of circuits. To proceed with our analysis, however, we need to develop the ideas of pseudoinverses, calculating effective resistances, and a matrix version of the Chernoff Bound.

2.3 Pseudoinverses

In our analysis, we will come across the need to "invert" a singular matrix. Since this is obviously not possible, we redefine our question to one that makes more sense. Let M be a $n \times n$ symmetric matrix. We can diagonalize M:

$$M = \sum_{i=1}^{n} \lambda_i v_i v_i^T$$

If all the eigenvalues are nonzero, then it obviously invertible, and $M^{-1} = \sum_{i=1}^{n} \frac{1}{\lambda_i} v_i v_i^t$

The case we worry about is when there is a zero eigenvalue. But this is okay too: when M is degenerate, we define the *pseudoinverse* by throwing away the zero eigenvalues and eigenvectors. In that case, we have

$$M^+ = \sum_{i|\lambda_i \neq 0} \frac{1}{\lambda_i} v_i v_i^T$$

The pseudoinverse has many nice properties. Of these, we use:

- $ker(L) = ker(L^+$
- $MM^+ = \sum_{i|\lambda_i \neq 0} v_i v_i^T =$ the projection onto the nonzero eigenvectors.

It is easy to see that $MM^+ = I$ when restricted to the image of M.

2.4 Effective Resistance

We mentioned earlier that Spectral Sparsification also samples edges with different probability. It turns out that the correct way to do this is to sample each edge with probability proportional to its "effective resistance."

The basic idea is to treat each edge as a resistor with resistance 1. If the edge had a capacity of c, we give it a resistance of $1/c$. After calculating these values, we sample the edge (u, v) with probability proportional to the effective resistance between nodes u and v.

Students may recall learning methods to solve circuits from their previous classes. For example, students may use a combination of Ohm's law and Kirchoff's law, as well as the rules for calculating effective resistances of resistors in series and parallel. To those who are comfortable with solving circuits, this may be a good way to think about the problem. However, the students who don't like solving circuits are in luck too: now that we have the tools of Spectral Graph Theory, we can solve circuits with only linear algebra! In fact, we will combine our frequent use of the graph Laplacian with the pseudoinverse defined above.

Let U be the edge-vertex adjacency matrix, C be the diagonal matrix with the various capacitances, and $r_e = 1/c_e$.

That is, we define U as in:

$$U(e, v) = \begin{cases} 1 & \text{if } v \text{ is the head of } e \\ -1 & \text{if } v \text{ is the tail of } e \\ 0 & \text{otherwise} \end{cases}$$

Then, we have that $L = U^T C U$. From ohm's law, we have $i = CUv$ for $i \in \mathbb{R}^E$, and $v \in \mathbb{R}^v$. From the conservation of current, we have $i_{ext} = U^T i$, for $i_{ext} \in \mathbb{R}^V$. Finally, we have $i_{ext} = Lv$, and $v = L^+ i_{ext}$

We define $U(e, v)$ to be the adjacency matrix with ± 1 values. Let u_e be the e^{th} row, and $v = L^+ i_e xt$. We have

$$R_{eff}(e) = u_e L^+ u_e^T$$

and as a result,

$$R_{eff}(e) = (U L^+ U^T)_{e,e}$$

Thus, calculating the effective resistance of an edge is as simple as calculating the pseudoinverse of the Laplacian. Simple!

2.5 Error Bounds

The last tool that we need to build is a way to define error bounds for matrices. In particular, we will use the following theorem.

Theorem 7 *For distributions on vectors y where $\| y \| \leq t$ and $\| Eyy^t \|_2 \leq 1$ (where we are using the l_2 norm) then:*

$$E \| Eyy^T - \frac{1}{q} \sum_{i=1}^{q} y_i y_i^T \|_2 \leq kt \sqrt{\frac{\log q}{q}}$$

This is a "concentration of measure theorem, and we claim that it is similar to the Chernoff bound.

Now, onto approximation. For our sparisifier H to approximate the original dense graph G, we want that

$$1 - \epsilon \leq \frac{x^T L_H x}{x^T L_G x} \leq 1 + \epsilon$$

for all vectors x. Rather, it is sufficient to show that

$$1 - \epsilon \leq \frac{z^T M^T L_H M z}{z^T M^T L_G M z} \leq 1 + \epsilon$$

for all vectors z, provided that $x \perp (L_G) \Rightarrow x \in range(M)$. Choose M so that $M^T L_G M$ is a projection. Then, it suffices to show that

$$\| M^T L_H M - M^T L_G M \|_2 \leq \epsilon$$

From before, we have that $L_G = U^T C U$. Choose $M = L_G^+ U^T C^{1/2}$. Then, we have

$$\Pi = M^T L_G M = C^{1/2} U L_G^+ U^T C^{1/2} = \Pi\Pi$$

Now, recall that $L_G = U^T C U$. If we let d_e be the weight of e in the sparsifier H, set $S_{e,e} = \frac{d_e}{c_e}$. Then, we can write

$$L_H = U^T C S U = U^T C^{1/2} S C^{1/2} U$$

yielding

$$M^T L_H M = \Pi S \Pi$$

We need to choose a diagonal S such that the number of nonzero elements of S is $O(nlogn/\epsilon^2)$ With this choice, we have

$$\| \Pi S \Pi - \Pi \|_2 \leq \epsilon$$

Define π_e as the e^{th} column of Π: that is, $\pi_e = \Pi(\cdot, e)$. Then, $\Pi S \Pi = \sum S_{e,e} \pi_e \pi_e^T$, so

$$\| \pi_e \|^2 = \Pi_{e,e} = c_e R_{eff}(e)$$

(this is because $\Pi = \Pi^2 = C^{1/2}(U L_G^+ U^T) C^{1/2}$)

We then set $\tau_e = \sqrt{\frac{n-1}{c_e R_{eff}(e)}} \pi_e$ with $\| \tau_e \| = \sqrt{n-1}$. Choose edges with probability $p_e = \frac{c_2 R_{eff}(e)}{n-1}$. Recall that

$$\sum_e c_e R_{eff}(e) = \sum_e \Pi_{e,e} = n - 1$$

Then, we find that

$$E[\tau_e \tau_e^T] = \sum_e p_e \tau_e \tau_e^T = \sum_e \pi_e \pi_e^T = \Pi$$

Sample q times with replacement, and set $S(e,e) = \frac{1}{q c_e R_{eff}(e)} \times$ the number of times that e is chosen. Then, from the theorem above, we have

$$E[\| \Pi - \Pi S \Pi \|_2] \leq k\sqrt{n-1}\sqrt{\frac{logq}{q}} \leq \epsilon/2$$

for $q = O(n \log n/\epsilon^2)$. Thus, we see that our construction yields a spectral sparsifier as desired.

From the algorithmics of the construction, it is easy to see that this is a poly-time procedure. The whole procedure is constructive, and uses the standard linear algebra operations. The bottleneck in this procedure comes from computing effective resistances, and in particular, the matrix inversions and multiplications. We claim that the procedure can be improved to nearly linear time. Doing so would involve two components:

- Close to linear algorithms for solving linear equations of the form $Lx = b$ for a laplacian L.
- A way to compute all the effective resistances by solving logarithmically many linear systems. This uses the Johnson-Lindenstrauss Lemma.

References

[1] "Randomized Approximation Schemes for Cuts and Flows in Capacitated Graphs, " A. Benczur, D. Karger, manuscript.

18.409 Topics in Theoretical Computer Science: An Algorithmist's Toolkit
Fall 2009