Lecture 16

Spectral sparsifiers

16.1 Sampling via effective resistances

Recall the setup from the previous lecture: we are given an $n \times n$ real symmetric invertible M-matrix M, a real vector b, and our goal is to solve $Mx = b$. We managed to reduce this to a small number of vector-matrix multiplications by other M-matrices $M_i = D_i - A_i$. The only task left for us is to show the following.

Theorem 16.1. *For any $\varepsilon > 0$ and M-matrix $M = D - A$ there exists an M-matrix $\hat{M} = \hat{D} - \hat{A}$ such that $D \approx_\varepsilon \hat{D}$, $D - A \approx_\varepsilon \hat{D} - \hat{A}$, and $\hat{M}$ has $O(n/\varepsilon^2)$ nonzero entries.*

Not only do we want existence, as in the theorem, but we also want $\hat{M}$ to be efficiently constructible from M. You will see from the proof that this is the case, provided we have an efficient way to compute "effective resistances". This point I will gloss over. In fact it itself reduces to solving a system of linear equations...but an easier one!

Remark 16.2. Recall the interpretation of the Laplacian as a quadratic form, $x^T L x = \sum_{(i,j)\in E}(x_i - x_j)^2$. If $\hat{L} \approx_\varepsilon L$ then for any cut $(S, \overline{S})$, by setting $x_i = 1$ for $i \in S$ and $x_i = 0$ and using

$$e^{-\varepsilon} x^T L x \leq x^T \hat{L} x \leq e^{\varepsilon} x^T L x$$

we see that the size of the same cut in $\hat{G}$ will be a close approximation, up to a multiplicative $e^{\pm\varepsilon} \approx (1 \pm \varepsilon)$, to its size in G. So the notion of "spectral sparsifier" implicit in Theorem 16.1 is at least as strong as that of "combinatorial sparsifier" (which only requires approximately preserving the size of all cuts); in the homework you will show that it can be strictly stronger (i.e. not every combinatorial sparsifier is a spectral sparsifier).

So let's prove the theorem. We're going to use some form of importance sampling with a well-chosen set of weights. For any edge (a, b), let $r_{a,b} = (e_a - e_b)^T L^+ (e_a - e_b)$, where L^+ is the pseudo-inverse of the Laplacian matrix (the inverse of L on its range) and e_a the indicator vector with a 1 in the a-th coordinate and 0 elsewhere. This quantity is called the *effective resistance* of edge (a, b). Now set $p_{a,b} = \min(1, C r_{a,b} \log n/\varepsilon^2)$, for some constant C that will be defined later. Edge (a, b) is sampled with probability $p_{a,b}$, and it is given

a weight $1/p_{a,b}$. This way we get the expectation right: if we let $\hat{L}$ be the random matrix associated to the sampled graph then on expectation $\mathbf{E}[\hat{L}] = L$. Our goal is now to show that with high probability $\hat{G}$ has $O(n \log n/\varepsilon^2)$ edges and is an ε-approximation to G, in the sense that $L \approx_\varepsilon \hat{L}$.

Number of edges. Let's start by counting the expected number of edges in $\hat{G}$. We have

$$\begin{aligned}\sum_{(a,b)\in E} r_{a,b} &= \sum_{(a,b)\in E} (e_b - e_a)^T L^+ (e_b - e_a) \\ &= \sum_{(a,b)\in E} \mathrm{Tr}\big(L^+(e_b - e_a)(e_b - e_a)^T\big) \\ &= \mathrm{Tr}\big(L^+ L\big) \\ &= n - 1,\end{aligned}$$

the dimension of the range of L. The choice of weights $r_{a,b}$ we made is precisely so that this equation worked out. Once we scale by $C \log n/\varepsilon^2$ to obtain $p_{a,b}$, we see that the expected number of edges in $\hat{G}$ is $O(n \log n/\varepsilon^2)$, which is quasi-linear in n. Moreover, using that each edge is sampled independently a simple Chernoff bound shows that the probability we sample more than a constant times this number of edges is exponentially small.

Approximation quality. The main remaining question is whether we can show our sampling technique achieves $L \approx_\varepsilon \hat{L}$. Let $X_{a,b}$ be the random matrix defined as $\frac{1}{p_{a,b}} L^{+/2} L_{a,b} L^{+/2}$ with probability $p_{a,b}$, and 0 otherwise, where $L_{a,b} = (e_b - e_a)(e_b - e_a)^T$. If we ignore the "sandwiching" by $L^{+/2}$, $X_{a,b}$ is the Laplacian matrix corresponding to the random graph obtained when we flip the coin that lets us decide whether to include edge (a, b) in $\hat{G}$ or not. We have

$$\mathbf{E}\Big[\sum_{a,b} X_{a,b}\Big] \;=\; L^{+/2}\Big(\sum_{a,b} L_{a,b}\Big)L^{+/2} \;=\; L^{+/2} L L^{+/2} = \Pi,$$

the projector on the range of L. Let $X = \sum_{a,b} X_{a,b}$. We will prove that with high probability $\Pi \approx_\varepsilon X$, i.e. $e^{-\varepsilon}\Pi \leq X \leq e^{\varepsilon}\Pi$. Note that if we have this we can simply multiply on both sides by $L^{1/2}$ and deduce the conclusion we really want, $e^{-\varepsilon} L \leq \hat{L} \leq e^{\varepsilon} L$.

This is starting to look very much like a Chernoff bound...except we are dealing with matrices! As in the Chernoff (or, rather, Hoeffding) bound, the quality of concentration we get will very much depend on the maximal size of the matrices $X_{a,b}$, this time measured by

their operator norm. So let's make sure this is never too big:

$$\begin{aligned}\|X_{a,b}\| &\le \frac{1}{p_{a,b}}\left\|L^{+/2}L_{a,b}L^{+/2}\right\| \\ &= \frac{1}{p_{a,b}}\mathrm{Tr}\big(L^{+}(e_a-e_b)(e_a-e_b)^T\big) \\ &= \frac{r_{a,b}}{p_{a,b}} \\ &= \frac{\varepsilon^2}{C\log n},\end{aligned} \tag{16.1}$$

where we used that $L^{+/2}L_{a,b}L^{+/2}$ is positive semidefinite of rank 1 to equate its norm with its trace. Note how the norm is independent of (a,b). This uniformity is a very good sign that we'll be able to obtain good concentration, and another main advantage of sampling via effective resistances.

Matrix Chernoff bounds. We're almost done — we're only missing the appropriate *matrix Chernoff bound*! Luckily for us, such a thing exists. Here is a bound that will give us exactly what we want.

Theorem 16.3. *Let $X_1,\dots,X_m$ be independent random positive semidefinite n-dimensional matrices (not necessarily identically distributed) and $R>0$ such that $\|X_i\|\le R$ for all i. Let $X=\sum X_i$, and μ_{min},μ_{max} the smallest and largest eigenvalues of $\mathbf{E}[X]$ respectively. Then for any $0<\varepsilon<1$,*

$$\mathbf{Pr}\left(\lambda_{min}\Big(\sum_{i=1}^m X_i\Big)\le(1-\varepsilon)\mu_{min}\right)\le n e^{-\frac{\varepsilon^2\mu_{min}}{2R}},$$

$$\mathbf{Pr}\left(\lambda_{max}\Big(\sum_{i=1}^m X_i\Big)\ge(1+\varepsilon)\mu_{max}\right)\le n e^{-\frac{\varepsilon^2\mu_{max}}{3R}}.$$

Note the main difference in the above statement, with respect to the standard Hoeffding bound, is the dimensional factor n on the right-hand side. This is a bit annoying (it would have been nice to have a dimension-free bound!), but in general it is unavoidable.

Exercise 1. Give an example of $X_1,\dots,X_m$ independent and identically distributed random n-dimensional matrices such that $\|X_i\|\le 1$ for all $i\in\{1,\dots,m\}$ and $X=\sum_i X_i$ is such that

$$\mathbf{Pr}\left(\lambda_{max}(X)\ge(1+\varepsilon)\lambda_{max}(\mathbf{E}[X])\right)\ge n e^{-\varepsilon^2\mu_{max}/C},$$

for some $0<\varepsilon<1$ (possibly depending on n) and a constant $C>0$.

Let's apply the theorem to conclude our analysis. In our case, by (16.1) we can set $R=\varepsilon^2/(C\log n)$. Also for us $\mathbf{E}\,X=\Pi$, which has both its smallest and largest eigenvalues equal to 1 (there is the 0 eigenvalue, but it doesn't matter, as we can do all the analysis in

the subspace orthogonal to the associated $\mathbf{1}$ eigenvector; note $\mathbf{1}^T X_{a,b}\mathbf{1} = 0$ for all $(a,b) \in E$). The matrix Chernoff bound gives us

$$\mathbf{Pr}\left(\lambda_{min}(X) \leq (1-\varepsilon)\right) \leq ne^{-\frac{\varepsilon^2 C \log n}{2\varepsilon^2}} = n^{1-C/2},$$

and

$$\mathbf{Pr}\left(\lambda_{max}(X) \geq (1+\varepsilon)\right) \leq ne^{-\frac{\varepsilon^2 C \log n}{3\varepsilon^2}} = n^{1-C/3}.$$

If we choose say $C = 4$, then both probabilities are small, and we get the desired approximation $(1-\varepsilon)\Pi \leq X \leq (1+\varepsilon)\Pi$ with high probability.

16.2 Preconditioning

Looking at the three methods for solving $Ax = b$ we've seen, they all had a linear dependence on the sparsity of the matrix A, but different dependencies on the condition number $\kappa = \lambda_{max}\lambda_{min}$: linear in κ for first-order Richardson, $\sqrt{\kappa}$ for the improvement using Chebyshev polynomials, and $\ln^3 \kappa$ for M-matrices. So a natural question is, given an arbitrary matrix A, could we somehow efficiently preprocess A to make its condition number smaller? This would lead to a corresponding improvement in running time for each of the three methods.

This is the idea behind preconditioning. Consider any symmetric invertible $n \times n$ matrix B. Then $Ax = b \iff AB^{-1}(Bx) = b$. Let $y = Bx$, solving $Ax = b$ can be done by solving for y in $AB^{-1}y = b$, and then for x in $Bx = y$. (You might worry that AB^{-1} is not symmetric. We could simply multiply by BA on the left, and solve $BA^2B^{-1}y = BAb$ instead.) Can we find B such that both tasks are easier than the original one? We'd like $Bx = y$ to be easy, and $\kappa(AB^{-1})$ to be much smaller than $\kappa(A)$.

We're going to see a very nice idea on how to do this for the special case where A is a Laplacian matrix. (Recall that Laplacian matrices are not invertible, so when we write inverse we simply mean the pseudo-inverse, i.e. the inverse on the orthogonal complement of the nullspace — the $\mathbf{1}$ vector.)

The idea is the following. Suppose G is a given graph, and H a subgraph of G. Then I claim that $L_H \leq L_G$ in the semidefinite order. An easy way to see this is to remember the interpretation of the Laplacian as a quadratic form:

$$x^T L_G x = \sum_{(a,b)\in E(G)} w_{a,b}(x_b - x_a)^2 \geq \sum_{(a,b)\in E(H)} w_{a,b}(x_b - x_a)^2 = x^T L_H x.$$

In particular, we get that $\lambda_{min}(L_G L_H^{-1}) \geq 1$ (the matrix is not symmetric, but we can still talk about its eigenvalues), so the question is whether we can find H such that, (i) $L_H x = y$ is easy to solve, and (ii) $L_H^{-1}L_G$ has small norm. The idea to guarantee (i) is to use an H that is a *spanning tree* for G. (We need the tree to be spanning so that L_H is invertible on the same space as L_G is.)

Exercise 2. Give a linear time-algorithm to solve $L_H x = y$ (exactly) when H is a tree.

What kind of trees will be such that $\|L_H^{-1}L_G\|$ is small? We can get a naive bound by computing the trace:

$$\begin{aligned}\mathrm{Tr}\big(L_H^{-1}L_G\big) &= \sum_{(a,b)\in E(G)} w_{a,b}\mathrm{Tr}\big(L_H^{-1}(e_b-e_a)(e_b-e_a)^T\big)\\ &= \sum_{(a,b)\in E(G)} w_{a,b}(e_b-e_a)^T L_H^{-1}(e_b-e_a).\end{aligned}$$

Now, for fixed (a,b) we've seen this quantity before: it is the effective resistance between a and b, in the tree H. But it's easy to see that, since H is a tree, the effective resistance is just the sum of the resistances along the unique path in H from a to b. Since effective resistances are the inverse of weights, we are led to define the *stretch* of an edge $(a,b)\in E(G)$ with respect to the tree H as

$$s_{a,b} = w_{a,b}\sum_{i=1}^{k}\frac{1}{w_i},$$

where the sum is over the path in H from a to b. So what we are looking for is a *low-stretch spanning tree*...and such things can be computed efficiently!

Theorem 16.4 (Abraham and Neiman '12)**.** *Every weighted graph G has a spanning tree subgraph H such that the sum of the stretches of all edges of G with respect to H is at most $O(m\log n\log\log n)$, where m is the number of edges in G. Moreover, this tree can be computed in time $O(m\log n\log\log n)$.*

The upshot is that using such a tree, we can reduce to the case where $\kappa(L_H^{-1}L_G) = O(m\log n\log\log n)$. In Claim 11.7 we saw that for an arbitrary (unweighted) graph G the smallest eigenvalue of the (normalized) Laplacian had to be $\Omega(n^{-3})$, and that this bound could be achieved. So the condition number of an arbitrary Laplacian can be as large as $O(n^3)$: here we get an improvement even for the case of dense graphs.