15-852: Randomized Algorithms. Lecture notes February 17

Recycling Random bits

To review what we did last time:

- Have BPP alg using r random bits. Want to get error decrese of running k times independently, but without using so many random bits.
- Idea: set up implicit expander graph with one node for each string of length r, and do a random walk. Only need constant random bits per step. Sample every β steps where β is defined to make 2nd largest eigenvalue of $R = M^\beta$ at most $1/10$. Sample 7k times and take majority vote. If we color nodes "good" or "bad" depending on whether they cause the BPP algorithm to answer correctly or not, then what we want is for it to be very unlikely that more than half of samples are "bad" nodes.
- We'd like to say that no matter where you start, after running one step of R, there's at most 1/5 chance of being at a bad node. Can't quite get this. But, get something similar by looking at L_2 length. Showed: for any vector p, $||pR\bar{W}|| < ||p||/5$, where $\bar{W}$ is the matrix that zeroes out the "good" entries. Intuitively, if p is "spread out" then lots of p is zeroed out by multiplying with $\bar{W}$. On the other hand, if p is not spread out, then multiplying by R will spread it out, thereby reducing its L_2 length.
- Formally, eigenvectors: $e_1, e_2, ...$ where $e_1 = (1/n, ..., 1/n)$. All orthogonal.

  $p = x + y$, where $x = c_1 e_1$, $y = c_2 e_2 + \ldots + c_n e_n$

  $||xB\bar{W}|| = ||x\bar{W}|| \leq 1/10||x||$

  $||yB\bar{W}|| \ \leq \ ||yB|| \ = \ ||c_2\lambda_2^\beta e_2 + \ldots + c_n\lambda_n^\beta e_n|| \ \leq \ 1/10||y||$.

  So,

$$\begin{aligned}
||pB\bar{W}|| &= ||xB\bar{W} + yB\bar{W}|| \\
&\leq ||xB\bar{W}|| + ||yB\bar{W}|| \\
&\leq 1/10||x|| + 1/10||y|| \\
&\leq 1/5||p||.
\end{aligned}$$

Intuitively, in terms of expansion, look at "bad" set and its sequence of neighborhoods. Number of nodes, and number of edges, are increasing exponentially as you go out. So, somewhat like a random walk with bias outward.

Rapid Mixing

Basically this exact argument shows that an eigenvalue gap gives you rapid mixing.

Theorem 1 *Say M is a markov chain with real eigenvalues and orthogonal eigenvectors. Then, for any starting distribution, the L_2 distance between $q^{(t)}$ (the distribution after t steps) and the stationary distribution π is at most $|\lambda_2|^t$, where λ_2 is the eigenvalue of second-largest absolute value.*

So, if we have an eigenvalue gap of some constant ϵ, then takes only $O((\log n)/\epsilon)$ to get down to $1/n^c$.

For instance, symmetric matrices have real eigenvalues and orthogonal eigenvectors. E.g., say $M = (I/2 + A/2d)$ where A is matrix for a d-regular graph. Then M is symmetric, all eigenvalues of M are nonnegative, and M has eigenvalue gap $\epsilon/2d$ where ϵ is the eigenvalue gap for A.

In fact, can generalize to "time-reversible" Markov chains: $p_{ij}\pi_i = p_{ji}\pi_j$ for all i, j. E.g., random walk on a graph where nodes are not necessarily all the same degree.

Proof of theorem: Say orthogonal eigenvectors are $e_1, \ldots, e_n$. $e_1 = \pi$. Say we start at

$$q^{(0)} = c_1 e_1 + \ldots + c_n e_n.$$

(Actually, $c_1 = 1$ since entries in π sum to 1 and entries in all other eigenvectors sum to zero.) After t steps, we're at:

$$q^{(t)} = c_1 e_1 + (c_2(\lambda_2)^t e_2 + \ldots + c_n(\lambda_n)^t e_n).$$

The L_2 length of this second part is at most $|\lambda_2|^t$ times the length of $q^{(0)}$ (and $||q^{(0)}|| \leq 1$). ■ (which is another way of seeing that $c_1 = 1$).

Expanders and eigenvalues

What's going on in the above argument in terms of graph properties? What do eigenvalues have to do with the graph being an expander?

Alon's paper shows the following theorem:

Theorem 2 *Consider the following variation on an expander: G has degree d, and every set X of at most $n/2$ vertices has $|N(X) - X| \geq c|X|$ for some $c > 0$ (G doesn't have to be bipartite). Then, $A(G)$ satisfies:*

$$\lambda_2 \leq d - \frac{c^2}{4 + 2c^2}.$$

Proof is in paper. Can read if you want. Here, will give some intuition.

First, let's look at matrix $M = A(G)/d$. So, max eigenvalue is 1. Want to show next largest is bounded away from 1.

What does it mean to say $fM = \lambda f$? For each index v, this means:

$$(1/d) \sum_{u \in N(v)} f_u = \lambda f_v.$$

E.g., largest $\lambda = 1$, so this says that for its eigenvector, each entry is equal to average of its neighbors, which as we noted means they're all equal (assuming graph is connected). Rest of eigenvectors satisfy $\sum_v f_v = 0$. So, we could ask:

> Given that $\sum_v f_v = 0$, and $f \neq \vec{0}$, what is the largest value of λ such that for all v, $\sum_{u \in N(v)} f_u/d \geq \lambda f_v$?

Let's look at an example of the line graph. Say $\lambda = 1 - \epsilon$. Then we get:

$f_0 = 1, f_1 = 1 - \epsilon, f_2 \approx 1 - 4\epsilon, f_3 \approx 1 - 9\epsilon$, and roughly we can let $\epsilon = 1/n^2$.

In general, we're solving for:

$$\frac{f(x+\delta) + f(x-\delta)}{2} = (1-\epsilon) f(x),$$

viewing the line as a sequence of points separated by distance δ. In fact, if we set the left endpoint to be $x = 0$ and the right endpoint to be $x = \pi$, then $f(x) = cos(x)$ is a solution to this series of equations. In particular, if we plug in $f(x) = cos(x)$, and do a Taylor expansion about $f(x)$, we get:

$$cos(x)(1 - \delta^2/2! + \delta^4/4! - \ldots) = (1 - \epsilon) cos(x).$$

The line graph is not an expander, and we didn't get much eigenvalue gap. Here's another graph which we will "pretend" is an expander:

Consider a rooted complete binary tree of some depth d, connected at the leaves to the leaves of another identical rooted binary tree. This is not an expander, but it looks like one from the point of view of the two roots. Let's consider an eigenvector f such that $f_v = 1$ at one of the roots and $f_v = -1$ at the other root, $f_v = 0$ at the common leaves of the two trees, and more generally, for node v, the value of f_v is a function of just its distance to root # 1. Note: this is *not* the eigenvector of second-higest eigenvalue, but showing that its eigenvalue is bounded away from 1 gives us insight into what's going on in expanders.

At the leaves, $f_v = 0$. At their neighbors to the left, lets say $f_v = \alpha$. If we had $\lambda = 1$, then at *their* neighbors to the left we would have $f_v = 3\alpha$ so that $(3\alpha + 2 * 0)/3 = \alpha$. But, this means that the slope of f is *increasing* as we head towards the root. This cannot be since we know the curvature is the other way near the root. So, what value of λ is needed to get at least the curvature going in the right way? We need to replace the 3α by something at most 2α. Solving we get: $(2\alpha + 2 * 0)/3 \geq \lambda\alpha$, and therefore $\lambda \leq 2/3$.

This is in some sense the intuition behind the eigenvalue gap for expanders.

Approximation hardness

Slight digression related to homework assignment.

We talked before about an algorithm for approximating MAX 3-SAT. Also said that there are known constants such that approximating better than those is NP-hard. Specifically, several results have shown stronger and stronger versions of the following:

A poly time transformation T from 3CNF formulas to 3CNF formulas with the following property:

- If formula ϕ is satisfiable, then $\phi' = T(\phi)$ is satisfiable.
- If ϕ is not satisfiable, then not only is ϕ' not satisfiable, but also any assignment to the variables of ϕ' satisfies at most a $(1-\varepsilon)$ fraction of the clauses of ϕ'. The quantity ε is a specific constant greater than 0, like 1/113.

This transformation is the [ALMSS92] result, and we will discuss part of the proof of that result later.

Turns out you can use this to show that for the Clique problem, not only is it NP-hard, and not only can't you approximate to, say, a factor of 5, but you can't even approximate the optimal to a ratio of $n^{\epsilon'}$ for some $\epsilon' > 0$. (Actually, it's now recently been shown that you can't even approximate to $n^{1-\epsilon'}$).

Here's a neat randomized reduction. Show that if you could solve the Clique approximation, then could solve SAT in randomized poly time (so would imply NP $\subseteq$ RP). On the homework, you will derandomize this using the [IZ] result we proved last time. So, will show hardness based on P $\neq$ NP.

The reduction: Pick a clause at random from ϕ'. Repeat this $lg(n)$ times and call that set of at most $lg(n)$ clauses S_1. Repeat that n^3 times to create $S_2, \ldots, S_{n^3}$. For each set S_i create one node for each assignment to the variables in that set which satisfy the clauses in that set. So, there are at most $n^3 2^{3lg(n)} = n^6$ nodes. Let N be the number of nodes. Create the edges by connecting two vertices if their partial assignments are consistent.

If ϕ' is satisfiable, then there is a clique of size n^3. If ϕ' is not satisfiable, consider some fixed assignment A to all the variables. Let X_i be the random variable that equals 1 if A satisfies S_i and equals 0 if it does not. Over the random choice of the clauses in S_i, $\mathbf{Pr}[X_i = 1] < (1-\varepsilon)^{lg(n)} < \frac{1}{2}n^{-\varepsilon}$. Let $Y = X_1 + \ldots + X_{n^3}$, so $\mathbf{E}[Y] < \frac{1}{2}n^{3-\varepsilon}$. Each X_i is an independent $\{0,1\}$ random variable, so by Chernoff bounds, $\mathbf{Pr}[Y > n^{3-\varepsilon}] < e^{-n^{3-\varepsilon}/6} < e^{-n^2}$. Since there are "only" 2^n possible assignments A, with overwhelming probability $(> 1 - 2^n e^{-n^2})$ there is no assignment that satifies more than $n^{3-\varepsilon}$ sets S_i, and so there is no clique of size greater than $n^{3-\varepsilon}$. Thus, we have a ratio of n^{ε} or $N^{\varepsilon/6}$. ∎