CS 710: Complexity Theory 3/02/2010

Lecture 13: Expanders

Instructor: Dieter van Melkebeek Scribe: Amanda Hittson, Nathan Collins and Matthew Anderson

In the last lecture we introduced randomized computation in terms of machines that have access to a source of random bits and that return correct answers more than $\frac{1}{2}$ of the time. We showed that BPP has polynomial-size circuits and the conjecture in the community is that BPP = P.

Today we'll introduce expanders, a type of graph that is useful in improving (amplifying) randomized algorithms with little or no additional random bit overhead.

1 Expanders

The proof of Theorem 2 in Lecture 12 used the "majority vote" trick to get an exponential in k increase in accuracy using a linear in k increase in random bit usage. Expander graphs, which are introduced here, lead to an "amplification" technique that gives an exponential in k accuracy improvement using only a constant increase in random bit usage (rk versus $r+k$). Expanders have many other uses, some of which we mention in this lecture, and others we may see in later lectures.

Definition 1 ((k,c)-expanding)**.** *An (undirected) graph $G = (V, E)$ is* (k, c)-expanding *if every $S \subset V$ with $|S| \leq k$ satisfies $|\Gamma(S)| \geq c|S|$, where $\Gamma(S) = \{v \in V \mid \exists s \in S, (s, v) \in E\}$ is the* neighborhood *of S in G.*

Notice that any graph is trivially (k, c)-expanding for all k with $c \leq 1$. Since $\Gamma(V) = V$ one can see it is important not to let $|S|$ be too large. In other words, for $k \geq |V|$ and $c > 1$, no graph is (k, c)-expanding.

Definition 2 (Expander family)**.** *An* expander family *is an infinite sequence of graphs $G_1 = (V_1, E_1), G_2 = (V_2, E_2), \ldots$ with a fixed integer d real number $c > 1$ such that each G_n has degree $\leq d$ and is $(\frac{N}{2}, c)$-expanding, where $N = N_n = |V_n|$ is the number of vertices in G_n.*

Intuitively, expander graphs are "very connected" in that the number of vertices reachable from a given subset of vertices is proportional to the size of that subset, at least when the subsets aren't so large as to make this impossible.

1.1 Graph Theoretic Properties

From now on we'll assume all our graphs are d-regular (each vertex has d edges) and have N vertices. We describe a d-regular graph using a normalized adjacency matrix. This view of a expander graphs proves useful in the analysis of their properties and randomized algorithms that make use of them.

Definition 3 (Normalized adjacency matrix)**.** *Given a d-regular $G = (V, E)$ its $N \times N$* normalized adjacency matrix *A is defined by*

$$A_{ij} = \begin{cases} \frac{1}{d}, & (i,j) \in E \\ 0, & \textit{otherwise} \end{cases}$$

The normalized adjacency matrix describes the Markov chain of a random walk on G:

$$A_{ij} = \Pr[\text{go to state } i \text{ from state } j],$$

and if p is a column vector with p_i the probability of being at vertex i, then $(Ap)_i$ gives the probability of being at vertex i after one random step in G.

The normalized adjacency matrix has a number of properties that can be proved using basic linear algebra. We will use the following properties, but we do not prove them here.

Proposition 1. *A is real-symmetric and so: all eigenvalues of A are real, and A has a basis of orthogonal eigenvectors.*

It turns out that the eigenvalues of the normalized adjacency matrix of an expander are closely connected to the expander graph properties. The following are some basic properties, whose proofs we omit.

Proposition 2. *Let A be a normalized adjacency matrix of a graph on N vertices.*

1. *Each eigenvalue λ of A satisfies $|\lambda| \leq 1$.*
2. *1 is an eigenvalue of A, with corresponding eigenvector given by the uniform distribution on $(1/N, 1/N, ..., 1/N)$.*
3. *The multiplicity of the eigenvalue 1 is greater than 1 iff G is disconnected.*
4. *A has -1 for an eigenvalue iff G is bipartite.*

As the uniform distribution is always an eigenvector for eigenvalue 1, we look only at the remaining eigenvector/values. It turns out that the second largest eigenvalue in absolute value is often useful to work with.

Definition 4 ($\lambda(G)$)**.** *Let $\Lambda = \{\lambda \mid \exists e_\lambda \neq 0 \text{ such that } (Ae_\lambda = \lambda e_\lambda) \wedge (\langle e_\lambda, u\rangle = 0)\}$. Then*

$$\lambda(G) = \max_{\lambda \in \Lambda} |\lambda|$$

is the largest eigenvalue corresponding to an eigenvector orthogonal to u.

So, if G is connected and not bipartite then $\lambda(G) < 1$. The value $1 - \lambda(G)$ is called the *spectral gap* of the graph. The next two theorems will not be proved here. See the notes for CS 880 Spring 2006 for proofs.

Theorem 1. *There exists a function f(x,y) such that if $x > 1$ then $f(x, y) < 1$ and for any graph G that is $(\frac{N}{2}, c)$-expanding, $\lambda(G) < f(c, d)$. Note that the value of $f(c, d)$ depends only on c and d.*

Corollary 1. *Expanders have positive spectral gaps and expander families have spectral gaps bounded away from 0.*

Theorem 2. *If $\lambda(G) \leq \lambda < 1$ then G is $(\frac{N}{2}, g(\lambda))$-expanding where $g(x)$ is a function satisfying $g(\lambda) > 1$ when $\lambda < 1$. Note that $g(x)$ does not depend on G, only on λ.*

In light of Theorems 1 and 2 we see that we could just as well have defined expander families in Definition 2 as d-regular families with positive spectral gaps.

So, for expanders, the uniform distribution u is the only fixed point. The following Lemma tells us that the larger the spectral gap the faster an arbitrary probability distribution converges to the uniform distribution via a random walk on G.

Lemma 1. *For any probability distribution vector p*

$$\|A^t p - u\|_1 \leq \sqrt{N}\lambda^t,$$

where $\|v\|_q = \left[\sum_i |v_i|^q\right]^{1/q}$ is the q-norm of v and $\lambda = \lambda(G)$.

One way to understand this lemma is: given an initial distribution for a random walk p, after t steps of the random walk $A^t p$, we will be exponentially closer to the random distribution, u, provided $\lambda < 1$, that is G is connected and not bipartite.

Proof. Since $Au = u$ we have $A^t p - u = A^t(p - u)$. Now, $(p - u)\perp u$ because

$$\langle p - u, u\rangle = \langle p, u\rangle - \langle u, u\rangle = \sum_{i=1}^{N} p_i/N - \sum_{i=1}^{N} 1/N^2 = 1/N - 1/N,$$

which follows since p is a probability distribution.

Write $p - u = \sum_i a_i e_i$ for some real a_i, where the $\{e_i\}$ form an orthogonal basis of eigenvectors for A with $Ae_i = \lambda_i e_i$. Then, since the u-component of $p - u$ is 0, we have the following inequality

$$\|A^t(p-u)\|_2 = \left\|A^t \sum_i a_i e_i\right\|_2 = \left\|\sum_i \lambda_i^t a_i e_i\right\|_2 \leq \lambda^t \left\|\sum_i a_i e_i\right\|_2 = \lambda^t \|p - u\|_2$$

Now

$$\|p - u\|_2^2 + \|u\|_2^2 = \|p\|_2^2 \leq \|p\|_1^2 = 1,$$

where $(p - u)\perp u$ implies the first equality, and $\|v\|_2^2 \leq \|v\|_1^2$ for all vectors v implies the inequality. By the Cauchy-Schwartz inequality, we have

$$\begin{aligned}\|A^t(p-u)\|_1 &= |\langle(-1^{\sigma_1}, \ldots, -1^{\sigma_N}), A^t(p-u)\rangle| \\ &\leq \|(-1^{\sigma_1}, \ldots, -1^{\sigma_N})\|_2 \|A^t(p-u)\|_2 \\ &= \sqrt{N}\|A^t(p-u)\|_2,\end{aligned}$$

where $\sigma_k = \begin{cases} 0, & (A^t(p-u))_k \geq 0 \\ 1, & \text{otherwise} \end{cases}$. Combining all of the above we get

$$\begin{aligned}\|A^t(p-u)\|_1 &\leq \sqrt{N}\|A^t(p-u)\|_2 \\ &\leq \sqrt{N}\lambda^t\|(p-u)\|_2 \\ &\leq \sqrt{N}\lambda^t,\end{aligned}$$

completing the proof. □

This lemma can be used to prove that the random walk algorithm for the undirected path problem needs only polynomially many steps, i.e. that PATH ∈ BPL. The proof uses Lemma 1 and the following exercise.

Exercise 1. *If G is connected and not bipartite then $\lambda(G) \leq 1 - \frac{1}{dN^2}$.*

1.2 Constructions

To prove that expanders exist one can argue that a randomly chosen d-regular graph G (for each vertex, choose d edges randomly) has a high probability of being an expander (with c close to d). However, we want to use expanders to reduce the amount of randomness needed in random algorithms, so using randomness to construct expanders won't help us. We want explicit constructions for which given a vertex v and index i we can compute v's ith neighbor in time poly$(|v|, |i|)$.

Example: For any integer $m \geq 2$, we can get an expander G on vertices $V = \mathbb{Z}_m \times \mathbb{Z}_m$ with edges given by the relations

$$\Gamma(\{(x, y)\}) = \{(x, y \pm x), (x, y \pm (x + 1)), (x \pm y, y), (x \pm (y + 1), y)\}.$$

If $m \geq 4$ then the graph has degree 8. The proof that this construction works is non-trivial. See the notes for CS 880 for a partial proof using harmonic analysis. ⊠

There are other efficient constructions of expanders, but the aforementioned expander suffices for our needs. See the notes for CS880 Spring 2006 for other constructions.

2 Expander Properties

The second largest eigenvalue, $\lambda(G)$, determines how quickly random walks converge to the uniform distribution. Lemma 1 showed that for a probability distribution p, a normalized matrix A and the uniform distribution u:

$$\|A^t p - u\|_1 < \sqrt{N}(\lambda)^t. \tag{1}$$

If $\lambda < 1$, the distance between the uniform distribution and a random walk from an arbitrary distribution p decreases exponentially with the number of steps t. The left hand side of equation (1) can be written using the definition of the 1-norm:

$$\|A^t p - u\|_1 = 2 \cdot \max_{B \subseteq V} |\Pr[A^t p \in B] - \Pr[u \in B]| \tag{2}$$

Think of $\|A^t p - u\|_1$ as twice the distance between a uniform distribution and a random walk for any set of vertices B. Normally it would take $\log V$ random bits to select a vertex at random from V, using this property a vertex can be selected almost uniformly using $t \log d$ bits by fixing some start vertex and performing a random walk for t steps.

The next important algebraic property of expanders is called the *expander mixing lemma*:

Lemma 2 (Expander Mixing Lemma)**.** *For every pair of subsets $S, T \subseteq V$,*

$$\left| \frac{|E(S,T)|}{dN} - \mu(S)\mu(T) \right| \leq \lambda \sqrt{\mu(S)(1 - \mu(S))\mu(T)(1 - \mu(T))}. \tag{3}$$

This lemma bounds the difference in the distributions of picking two vertices uniformly at random (second term on left) with picking one vertex and one neighbor of the vertex at random (first term on left). In the first case $2 \log V$ random bits are used, in the second case only $\log V + \log d$ random bits are used. This idea allows the amount of randomness to be reduced at the cost of producing a distribution that is not quite uniform.

Proof. Expander Mixing Lemma

Recall that $A(G)$ is symmetric and real implying that it has a full orthonormal eigenbasis. The number of edges between two sets S and T can be written in terms of their relative characteristic vectors (i.e. $\chi_{S,i} = 1$ iff $v_i \in S$ and $\chi_{S,i} = 0$ otherwise):

$$|E(S,T)| = \chi_S^T (dA) \chi_T. \tag{4}$$

By definition of A, dA is the standard adjacency matrix for G. Both χ_S and χ_T can be rewritten in terms of their components parallel and perpendicular to the uniform vector u. Recall $u = (\frac{1}{N}, \frac{1}{N}, ..., \frac{1}{N})$ is an eigenvector corresponding to the eigenvalue 1. The inner product of χ_S with u is $(\chi_S, u) = \frac{|S|}{N} = \mu(S)$ and $\chi_S^{\parallel} = (\chi_S, \hat{u})\hat{u} = (\chi_S, \sqrt{N}u)\sqrt{N}u = |S|u$. Equation 4 can be rewritten:

$$\begin{aligned} |E(S,T)| &= (\chi_S^{\parallel} + \chi_S^{\perp})(dA)(\chi_T^{\parallel} + \chi_T^{\perp}) \\ &= \chi_S^{\parallel} dA \chi_T^{\parallel} + \chi_S^{\perp} dA \chi_T^{\perp} \\ &= d\chi_S^{\parallel} \chi_T^{\parallel} + \chi_S^{\perp} dA \chi_T^{\perp} \\ &= d\frac{|S||T|}{N} + \chi_S^{\perp} dA \chi_T^{\perp}. \end{aligned} \tag{5}$$

The second and third lines follow from the first because $\chi^{\parallel}$ is an eigenvector of A causing the first term to simplify and the cross terms to vanish. Dividing by dN, moving terms around and taking the absolute value gives:

$$\begin{aligned} \left| \frac{|E(S,T)|}{dN} - \mu(S)\mu(T) \right| &= \left| \frac{\chi_S^{\perp}(dA)\chi_T^{\perp}}{dN} \right| \\ &\leq \frac{\|\chi_S^{\perp}\|_2 \cdot \lambda d \cdot \|\chi_T^{\perp}\|_2}{N} \end{aligned} \tag{6}$$

The second line is reached by applying Cauchy-Schwarz to the RHS and using the fact that A decreases the magnitude of $\chi_T^{\perp}$ by at least λ as there is no component of $\chi_T^{\perp}$ along u. Applying the Pythagorean theorem and some simple algebra to $\|\chi_S\|_2$ we can derive the value of $\|\chi_S^{\perp}\|_2$:

$$\begin{aligned} \|\chi_S\|_2^2 &= \|\chi_S^{\parallel}\|_2^2 + \|\chi_S^{\perp}\|_2^2, \text{ therefore} \\ |S| &= |S|^2 \frac{1}{N} + \|\chi_S^{\perp}\|_2^2, \text{ and} \\ \|\chi_S^{\perp}\|_2^2 &= |S|(1 - \mu(S)), \\ \|\chi_S^{\perp}\|_2 &= \sqrt{|S|(1-\mu(S))}. \end{aligned} \tag{7}$$

Substituting this back in for $\|\chi_S^{\perp}\|_2$ and $\|\chi_T^{\perp}\|_2$ and pulling the factor of N into the square root completes the proof.

□

2.1 Next Time

The expander mixing lemma can be used to reduce the error probability of randomized algorithms while using little or no extra random bits. In particular, we'll use expanders for two purposes:

1. *Deterministic amplification*: Given a randomized algorithm R that uses r random bits we reduce the error to be less than some arbitrary ε, without using any additional random bits. This requires running R $\text{poly}(1/\varepsilon)$ times.

2. *Randomness efficient amplification*: With R and r as above we reduce the error to be less than some arbitrary ε using $r + O(\log(1/\varepsilon))$ additional random bits and running R $O(\log(1/\varepsilon))$ times.

To accomplish the above amplifications we use an expander graph G whose vertices are in one-to-one correspondence with the bit strings in $\{0,1\}^r$. For application 1 we run R, look at the vertex v in G corresponding to the random bits used by R's run, and then run R once for each neighbor v' of v, with random bits corresponding to v'. For application 2 we use the same G and v. Then, starting from v, we perform a random walk in G starting from v of length $t = \log(1/\varepsilon)$. This gives t vertices $v_1, \ldots, v_t$. We then run R once for each vertex v_i with random bits corresponding to v_i. In both cases, we finish by taking the "majority vote" of R's runs. We discuss these applications and prove their correctness in the next lecture.