CS 710: Complexity Theory 10/27/2011

Lecture 15: Expanders

Instructor: Dieter van Melkebeek Scribe: Li-Hsiang Kuo

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 somewhat more than half 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. First we'll prove a theorem relating BPP to the polynomial time hierarchy.

1 BPP and the Polynomial Time Hierarchy

Although we don't know if BPP = P, or even if BPP $\subseteq$ NP, we do know that BPP $\subseteq$ PH:

Theorem 1. BPP $\subseteq \Sigma_2^p \cap \Pi_2^p$

Proof. Fix M a randomized polytime machine that accepts $L \in$ BPP, and let r be the number of random bits M uses when running on an input x of size $n = |x|$. We will now show that BPP $\subseteq \Sigma_2^p$. Since BPP = *co*BPP it follows that BPP $\subseteq co\Sigma_2^p = \Pi_2^p$, completing the proof.

We need to remove randomness from the computation. For a given input x, the space $\{0,1\}^r$ of r-bit strings gets partitioned into two pieces: $Acc(x)$, the set of random strings on which M accepts x, and $Rej(x)$, the set of strings on which M rejects x. If the error rate ε of M is small, then $Acc(x)$ will be much larger than $Rej(x)$ when $x \in L$, and $Acc(x)$ will be much smaller than $Rej(x)$ when $x \notin L$. See Figure 1. If our random computation has a small error rate then it will be correct on most random bit strings. We'll turn "most" into "all." The idea is that when $x \in L$ a few "shifts" of $Acc(x)$ will cover the whole space $\{0,1\}^r$ of r-bit random sequences, while the same few shifts of $Acc(x)$ will fail to cover the whole space when $x \notin L$.

If $S \subseteq \{0,1\}^r$ and $\sigma \in \{0,1\}^r$, then $S \oplus \sigma = \{s \oplus \sigma | s \in S\}$, the *shift* of S by σ. Since shifting is invertible we see that $|S \oplus \sigma| = |S|$.

We will use shifts to give a Σ_2-predicate using the intuition discussed above. Namely, consider

$$x \in L \iff \exists \sigma_1, \ldots, \sigma_t \forall \rho \in \{0,1\}^r [\rho \in \bigcup_{i=1}^{t} Acc(x) \oplus \sigma_i]. \tag{1}$$

Now, r is poly in n, and so if we can pick t poly in n as well, then the above will be a Σ_2^p-predicate, provided that we can verify the membership of ρ in $\bigcup_{i=1}^{t} Acc(x) \oplus \sigma_i$ in time poly in n. The membership check is no problem since we can equivalently check that $\rho \oplus \sigma_i \in Acc(x)$ for some i, which is polytime since it corresponds to running M on x with the random bit string $\rho \oplus \sigma_i$, for poly-many σ_i.

We'll show we can pick t a suitable poly to make (1) true by showing that we can choose the σ_is randomly with a high rate of success. There are two cases to consider:

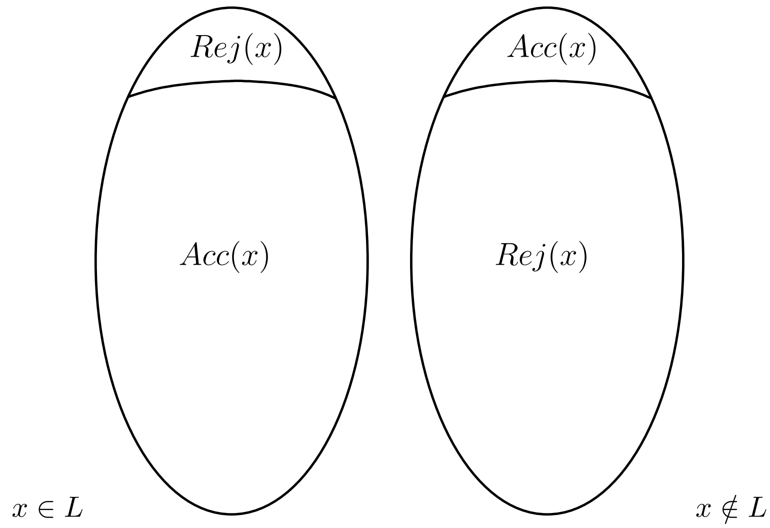

Figure 1: If the error rate ε of M is small, then $Acc(x)$ will be much larger than $Rej(x)$ when $x \in L$, and $Acc(x)$ will be much smaller than $Rej(x)$ when $x \notin L$.

1. $x \in L$: For $A \subseteq \{0,1\}^r$, define
$$\mu(A) = \frac{|A|}{2^r}.$$
For ρ fixed,
$$\Pr_{\sigma_1,\ldots,\sigma_t}[\rho \notin \bigcup_{i=1}^{t} Acc(x) \oplus \sigma_i] \leq (\mu(Rej(x))^t \leq \varepsilon^t,$$
where $\mu(Rej(x))$ is the probability that M rejects when it should accept, and is hence not larger than ε. So, by an union bound,
$$\Pr[\{0,1\}^r \not\subseteq \bigcup_{i=1}^{t} Acc(x) \oplus \sigma_i] \leq |\{0,1\}^r|\varepsilon^t = 2^r\varepsilon^t,$$
and hence if $2^r\varepsilon^t < 1$ then some choice of the σ_is must work.

2. $x \notin L$: We need to be sure that for any choice of the σ_is that none of the shifts of ρ make M accpet x. But
$$\mu(\bigcup_{i=1}^{t} Acc(x) \oplus \sigma_i) \leq \sum_{i=1}^{t} \mu(Acc(x) \oplus \sigma_i) = t\varepsilon,$$
since $\mu(Acc(x)) \leq \varepsilon$ when $x \notin L$, and hence we need $t\varepsilon < 1$.

So, we need to choose t so that both $t\varepsilon$ and $2^r\varepsilon^t$ are less than 1. So $t = r$ works, provided that $\varepsilon < \frac{1}{r}$. From the definition of BPP we only know that $\varepsilon < \frac{1}{3}$, but, using the "majority vote" trick introduced during the last lecture, we can in k runs reduce the error to $e^{-2k\delta^2}$, where $\delta = \frac{1}{2} - \varepsilon > 0$. Since k runs will need kr random bits, we see that it suffices to choose k poly and large enough that $e^{-2k\delta^2} < \frac{1}{rk}$. In fact, $k = O(\log r)$ works, and so an examination of the above shows that the time complexity is $O(T^2 \log T)$ if the original run-time was T.

$\square$

2 Expanders

The proof of Theorem 1 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.

2.1 Definitions

Here we introduce two definitions of expander graphs, namely the combinatorial definition and the algebraic definition.

Definition 1 (Combinatorial definition)**.** *A graph $G = (V, E)$ is a (k, c)-*expander *if $\forall S \subseteq V$ 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 it is easy to construct a graph which is (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. A typical setting is c is a constant > 1 and $k = \frac{N}{2}$ where $N = |V|$.

Definition 2 (Algebraic definition)**.** *Given a regular $G = (V, E)$ with degree d 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 step of the random walk on G.

$$(Ap)_i = \sum_{j=1}^{N} A_{ij} p_j.$$

2.2 Properties of the normalized adjacency matrix of a regular graph

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. *Let A be a normalized adjacency matrix of a regular graph G on N vertices.*

1. *A is real and symmetric: all eigenvalues of A are real, and there exists a full orthonormal basis of eigenvectors.*

2. *All eigenvalues λ of A satisfy $|\lambda| \leq 1$.*

3. *1 is an eigenvalue of A, with corresponding eigenvector $u = (\frac{1}{N}, \frac{1}{N}, ..., \frac{1}{N})$. The multiplicity of 1 as an eigenvalue equals the number of components of G.*

4. *-1 is an eigenvalue of A iff G is bipartite.*

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

Definition 3. $\lambda(G) = max\{|\lambda| \mid \lambda$ *is an eigenvalue of* G *with eigenvector* e_λ *orthogonal to* $u\}$

From Proposition 1 we have $\lambda(G) \leq 1$, and

$$\lambda(G) < 1 \iff G \text{ is connected and not bipartite.}$$

So, for expanders, the uniform distribution u is the only fixed point. The following theorem tells us that the larger the difference between 1 and $\lambda(G)$ the faster an arbitrary probability distribution converges to the uniform distribution via a random walk on G.

Theorem 2. *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 theorem 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 uniform 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

$$\begin{aligned} \|A^t(p-u)\|_2^2 &= \left\|A^t \sum_i a_i e_i\right\|_2^2 = \left\|\sum_i \lambda_i^t a_i e_i\right\|_2^2 = \sum_i \|\lambda_i^t a_i e_i\|_2^2 = \sum_i |\lambda_i|^{2t} \|a_i e_i\|_2^2 \\ &\leq \sum_i \lambda^{2t} \|a_i e_i\|_2^2 = \lambda^{2t} \sum_i \|a_i e_i\|_2^2 = \lambda^{2t} \|p - u\|_2^2. \end{aligned}$$

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 can be used to prove that the random walk algorithm for the undirected path problem needs only polynomially many steps, i.e. that PATH $\in$ BPL. The proof uses Theorem 2 and the following exercise.

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

3 Next Time

In the next lecture we will prove expander mixing lemma.

Lemma 1 (Expander Mixing Lemma)**.** *For every pair of subsets $S, T \subseteq V$,*

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

This lemma can be used in the following applications:

1. *Deterministic error reduction*: 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 poly$(1/\varepsilon)$ times.

2. *Randomness efficient error reduction*: 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.

We will discuss these applications and prove their correctness in the next lecture.

Acknowledgements

In writing the notes for this lecture, I perused the notes by Beth Skubak and Amanda Hittson for lecture 12 and 13 from the Spring 2010 offering of CS 710 and the notes by Nathan Collins for lecture 12 from the Spring 2007 offering of CS 810.