CS 810: Complexity Theory 2/13/2007

Lecture 13: Amplification

Instructor: Dieter van Melkebeek Scribe: Matthew Anderson

Last lecture we introduced expander graphs. From a combinatorial point of view these were relatively sparse graphs where each vertex had a constant degree and were non-trivially expanding. From an algebraic point of view expanders are graphs of constant degree which have $\lambda(G) < 1$. Recall $\lambda(G)$ is the largest absolute value eigenvalue of the normalized adjacency of G corresponding to an eigenvector which is perpendicular to the all ones vector.

Today we discuss two applications of expanders for performing correctness amplification on randomized algorithms. The first application decreases the error rate of an algorithm using no additional randomness. The second application decreases the error rate of an algorithm even further using only slightly more randomness.

1 Expander Properties

$\lambda(G)$ determines how quickly random walks converge to the uniform distribution. Last lecture we showed that for a probability distribution p, a normalized matrix A and the uniform distribution u:

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

If $\lambda(A) < 1$, the distance between the uniform distribution and a random walk from an arbitrary distribution p decreases exponentially with the number of steps t. The LHS of the above equation 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 1 (Expander Mixing Lemma)**.**

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

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):

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

dA is now the standard adjacency matrix for G. Since A is a real and symmetric matrix, A has a full orthonormal eigenbasis. 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 the eigenvector corresponding to the eigenvalue 1. The inner product of χ_S with u is $(\chi_S, u) = \frac{|S|}{N} = \mu(S)$ and $\chi_S^{\|} = (\chi_S, \hat{u})\hat{u} = (\chi_S, \sqrt{N}u)\sqrt{N}u = |S|u$. Equation 4 can be rewritten:

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

The second and third lines follow from the first because $\chi^{\|}$ 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 $\|\chi_S^{\perp}\|_2$:

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

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.

$\square$

Both of these properties can be used to reduce the error probability of randomized algorithms while using little or no extra random bits.

2 Deterministic Amplification

Using expanders it is possible to transform a random algorithm, R, that takes r random bits and errs with probability $\epsilon_0 \leq \frac{1}{3}$, into an equivalent algorithm, R', also using r random bits which errs with probability $\leq \epsilon$ by calling R poly$(\frac{1}{\epsilon})$ times.

The idea behind this transformation is to consider an explicit expander graph G with 2^r vertices with each vertex corresponding to a random bit string in $\{0,1\}^r$. Pick one vertex, ρ', at random from G using r random bits. Look at all neighbors, $\rho \in V$ at distance t from ρ'. Run R on each of these neighbors and return the majority vote of these neighbors as the output of R'.

Intuitively, because G is expanding the set of neighbors at distance t will be "spread out" on the graph and close to uniformly distributed.

2.1 Analysis

Let S be the set of the bad random strings for R (the strings that give the wrong answer for an input x). Let T be the set of bad random strings ρ' for R' (the strings whose majority of neighbors at distance t give the wrong answer).

$$\begin{aligned} S &= \{\rho | R \text{ gives incorrect answer}\} \\ T &= \{\rho' | \text{ majority of } \mathrm{N^t}(\rho') \text{ give the incorrect answer}\} \end{aligned} \tag{8}$$

By construction of S, $\mu(S) \leq \epsilon_0 \leq \frac{1}{3}$. In order for R' to have the required error we want $\mu(T) \leq \epsilon$. Consider fixing $t = 1$ then:

$$\frac{|E(S,T)|}{dN} \geq \frac{|T|\frac{d}{2}}{dN} = \frac{\mu(T)}{2}. \tag{9}$$

This is because at least half of the immediate neighbors ($t = 1$) of each ρ' in T are in S (they were incorrect for R). Applying the expander mixing lemma to this gives:

$$\mu(T)|\frac{1}{2} - \mu(S)| = |\frac{\mu(T)}{2} - \mu(S)\mu(T)| \leq \left|\frac{|E(S,T)|}{dN} - \mu(S)\mu(T)\right| \leq \lambda\sqrt{\mu(S)\mu(T)}. \tag{10}$$

The two extra terms in the last expression were dropped because they must be less than or equal to 1. Rearranging terms and bounding $\mu(S) \leq \epsilon_0$:

$$\mu(T) \leq \frac{\lambda^2\mu(S)}{(\frac{1}{2} - \mu(S))^2} \leq \frac{\lambda^2\epsilon_0}{(\frac{1}{2} - \epsilon_0)^2} \leq \epsilon. \tag{11}$$

This only gives a constant decrease in error. Consider the effect of taking neighbors at distance t instead of only immediate neighbors. Replace G with G' where G' has the same vertices as G but there is an edge between two vertices u and v in G' if and only if there is a path of length t between u and v in G (allow multiple edges between two vertices in G'). This has the effect of increasing the degree of G' to d^t and decreasing second eigenvalue $\lambda(G') = \lambda^t(G)$, and the prior analysis still holds with $\lambda(G')$ substituted for $\lambda(G)$ This changes the result of the Equation 11 to give:

$$\mu(T) \leq \frac{\lambda^{2t}\epsilon_0}{(\frac{1}{2} - \epsilon_0)^2} \leq \epsilon. \tag{12}$$

Selecting $t = O(\log \frac{1}{\epsilon})$ gives error less than ϵ. The number of neighbors of ρ' at distance t is $d^{O(\log \frac{1}{\epsilon})}$ which is polynomial in $\frac{1}{\epsilon}$. The complexity of determining the neighbors is polynomial because G is an explicit expander construction. Therefore we can reduce the error to $\frac{1}{\text{poly}}$ from a constant using polynomial time and the same randomness as the original algorithm.

3 Randomness Efficient Amplification

The idea of this second application is to reduce the error further by using a little additional randomness.

Given a random algorithm R which uses r random bits and errs with probability $\epsilon_0 \leq \frac{1}{3}$, it can be transformed into an equivalent algorithm R' which uses $r + O(\log \frac{1}{\epsilon})$ random bits with error $\leq \epsilon$ by calling R only $O(\log \frac{1}{\epsilon})$ times in total. Notice this construction allows us to acheive exponentially small error $(\frac{1}{2^n})$ in polynomial time with only slightly more randomness.

Such a transformation can be done trivially using $O(r \log \frac{1}{\epsilon})$ bits by sampling $\log \frac{1}{\epsilon}$ random strings, but does not achieve the additive result suggested. In order to improve on the trivial result, we perform a variation on approach in the previous application.

Again, pick $\rho' \in \{0,1\}^r$ uniform at random. Consider all ρ on a random walk of length t starting at ρ'. Run R on all such ρ and take the majority vote.

The idea here is that we can do a better job than in the previous application because there is more randomness to work with and neighbors at a further distance can be visited (because only one neighbor at each distance is visited).

3.1 Analysis

There is one key lemma that will allow us to bound the error of this approach:

Lemma 2. *Let P be a projection on those ρ for which R errs, then for any vector x:*

$$\|PAx\|_2 \leq \sqrt{\epsilon_0 + \lambda^2}\|x\|_2. \tag{13}$$

Proof. Consider the representation of $x = x^{\|} + x^{\perp}$ in the eigenbasis with respect to u as stated earlier. Then by the triangle inequality:

$$\|PAx\|_2 \leq \|PAx^{\|}\|_2 + \|PAx^{\perp}\|_2. \tag{14}$$

Using the facts that the bad set is small, P projects onto the bad set, and that $x^{\perp}$ contracts by at least λ:

$$\begin{aligned} \|PAx^{\|}\|_2 &= \|Px^{\|}\|_2 \leq \sqrt{\epsilon}\|x^{\|}\|_2, \\ \|PAx^{\perp}\|_2 &\leq \|Ax^{\perp}\|_2 \leq \lambda\|x^{\perp}\|_2. \end{aligned} \tag{15}$$

Substituting back into the original equation we have:

$$\|PAx\|_2 \leq \sqrt{\epsilon_0}\|x^{\|}\|_2 + \lambda\|x^{\perp}\|_2 = (\sqrt{\epsilon_0}, \lambda) \cdot (\|x^{\|}\|_2, \|x^{\perp}\|_2)^T \leq \sqrt{\epsilon_0 + \lambda^2}\|x\|_2 \tag{16}$$

The last inequality follows from Cauchy-Schwarz. □

This lemma can be used to bound the error probability of R'. Consider the probability that R' errs, this is the same as the probability that at least half of the steps in the random walk fell in the set where R errs.

$$\begin{aligned}
\Pr[R' \text{ errs}] &= \Pr[\text{At least } \frac{\text{t}}{2} \text{ of } \rho \text{ fall in the set on which R errs}] \\
&\le \sum_{B\subseteq[t],|B|\ge\frac{t}{2}} \Pr[(\forall i \in B) i^{\text{th}} \text{ step lies in the bad set for R}] \\
&= \sum_{B\subseteq[t],|B|\ge\frac{t}{2}} \|M_t A M_{t-1} A ... M_2 A M_1 A u\|_1 \\
&\le \sum_{B\subseteq[t],|B|\ge\frac{t}{2}} \sqrt{N}\|M_t A ... M_1 A u\|_2 \\
&\le \sum_{B\subseteq[t],|B|\ge\frac{t}{2}} \sqrt{N}\left(\sqrt{\epsilon_0+\lambda^2}\right)^{|B|}\|u\|_2 \\
&= \sum_{B\subseteq[t],|B|\ge\frac{t}{2}} \left(\sqrt{\epsilon_0+\lambda^2}\right)^{|B|} \\
&\le 2^t \cdot \left(\sqrt{\epsilon_0+\lambda^2}\right)^{\frac{t}{2}} \\
&= (4\sqrt{\epsilon_0+\lambda^2})^{\frac{t}{2}} \le \epsilon.
\end{aligned} \tag{17}$$

The second line upper bounds the actual probability because it is over counting the bad strings. This probability is rewritten as a matrix in the third line, where $M_i = P$ if and only if $i \in B$ and $M_i = I$ otherwise. The fifth line follows from repeated applications of Lemma 2. A constant number of iterations can decrease $\sqrt{\epsilon_0+\lambda^2}$ to less than $\frac{1}{4}$, making $|B| = \frac{t}{2}$ maximize the RHS. We can use the deterministic amplification procedure to acheive this. If $4\sqrt{\epsilon_0+\lambda^2} < 1$ walking for $t = O(\log\frac{1}{\epsilon})$ steps will give error less than ϵ. This procedure uses r random bits to pick the starting vertex, and $\log d$ bits for each of the $\log\frac{1}{\epsilon}$ steps in the random walk for a total of $r + O(\log\frac{1}{\epsilon})$ random bits for R'.

4 Other Results

A stronger result which considers the variance of random walks is known a the Expander Chernoff Bound. It states that if you take a random walk that the fraction of times the walk lands in the bad set does not vary much from the expected number. Let X_i be an indicator variable that indicates the event that the i^{th} step lies in some set $A_i \subseteq V$. Then the probability that the walk varies from the the expected number of steps in the bad sets can be written as:

$$\Pr[\sum_i^t (X_i - \mu(A_i)) \ge at] \le e^{-b(1-\lambda)a^2 t}. \tag{18}$$

where $a \ge 0$ and b is some universal constant. The probability that the walk varies from expected for a constant a decreases exponentially as t increases. This inequality reduces to the standard Chernoff Bound if G is a complete graph ($\lambda(G) = 0$ because $\text{rank}(G) = 1$ and all the X_i are independent).

5 Next Time

Next lecture we will discuss pseudorandom generators for space-bounded computation, as another application of expanders. This application leverages the power of being able to pick one vertex and a neighbor at random instead of picking two vertices uniformly at random. We will apply this procedure recursively to construct pretty good (though perhaps not the best) pseudorandom generators. Pseudorandom number generation is the dual of amplification. In pseudorandom number generation we will try to reduce the amount of randomness used while not making the error grow by too much more.