CS 710: Complexity Theory 3/4/2010

Lecture 14: Amplification

Instructor: Dieter van Melkebeek Scribe: Matthew Anderson & Tyson Williams

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

$$\lambda(G) = \max\{|\lambda| \mid (\exists 0 \neq x \perp u)\ Ax = \lambda x\}$$

where A is the normalized adjacency matrix of A and u is the uniform distribution.

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. Both applications use the expander mixing lemma that was proved in the previous lecture.

1 Deterministic Amplification

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

1.1 Construction

The idea behind this transformation is to consider an explicit expander graph G with 2^r vertices where each vertex corresponds to a random bit string in $\{0,1\}^r$. Pick a vertex ρ' at random from G using r random bits. Look at all neighbors ρ at distance t from ρ'. Run R on each of these neighbors ρ 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.

1.2 Analysis

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

$$B = \{\rho \mid R \text{ gives incorrect answer}\} \tag{1}$$
$$B' = \{\rho' \mid \text{majority of } N^t(\rho') \text{ give the incorrect answer}\} \tag{2}$$

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

$$\frac{|E(B, B')|}{dN} \geq \frac{|B'|\frac{d}{2}}{dN} = \frac{\mu(B')}{2}. \tag{3}$$

This is because at least half of the immediate neighbors ($t = 1$) of each ρ' in B' are also in B (they were incorrect for R). Applying the expander mixing lemma to (3) gives:

$$\mu(B')\left|\frac{1}{2} - \mu(B)\right| = \left|\frac{\mu(B')}{2} - \mu(B)\mu(B')\right| \tag{4}$$

$$\leq \left|\frac{|E(B, B')|}{dN} - \mu(B)\mu(B')\right| \tag{5}$$

$$\leq \lambda\sqrt{\mu(B)\mu(B')}. \tag{6}$$

The last two terms in (5) were dropped because they must be less than or equal to 1. Rearranging terms and bounding $\mu(B) \leq \epsilon_0$ gives:

$$\mu(B') \leq \frac{\lambda^2\mu(B)}{\left(\frac{1}{2} - \mu(B)\right)^2} \leq \frac{\lambda^2\epsilon_0}{\left(\frac{1}{2} - \epsilon_0\right)^2} \leq \epsilon. \tag{7}$$

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' iff 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 the second eigenvalue $\lambda(G') = \lambda^t(G)$. The prior analysis still holds with $\lambda(G')$ substituted for $\lambda(G)$. This changes the result of (7) to give:

$$\mu(B') \leq \frac{\lambda^{2t}\epsilon_0}{\left(\frac{1}{2} - \epsilon_0\right)^2} \leq \epsilon. \tag{8}$$

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.

2 Randomness Efficient Amplification

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

Given a random (BPP) algorithm R which uses r random bits and errs with probability $\epsilon_0 \leq \frac{1}{3}$, it can be transformed into an equivalent random (BPP) 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 achieve 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 upon the approach in the previous application.

2.1 Construction

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).

2.2 Analysis

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

Lemma 1. *Let P be a projection on those ρ for which R errs, then for any vector x:*

$$\|PAx\|_2 \leq \sqrt{\epsilon_0 + \lambda^2}\|x\|_2. \tag{9}$$

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

$$\|PAx\|_2 \leq \|PAx^{\|}\|_2 + \|PAx^{\perp}\|_2. \tag{10}$$

Using the facts that the uniform distribution is invariant under A and that the bad set is small, we get:

$$\|PAx^{\|}\|_2 = \|Px^{\|}\|_2 \leq \sqrt{\epsilon}\|x^{\|}\|_2. \tag{11}$$

Using the facts that P projects onto the bad set (so it does not affect a perpendicular vector) and that $x^{\perp}$ contracts by at least λ, we get:

$$\|PAx^{\perp}\|_2 \leq \|Ax^{\perp}\|_2 \leq \lambda\|x^{\perp}\|_2.. \tag{12}$$

Substituting (11) and (12) back into the (10), we have:

$$\begin{aligned}
\|PAx\|_2 &\leq \sqrt{\epsilon_0}\|x^{\|}\|_2 + \lambda\|x^{\perp}\|_2 && (13)\\
&= (\sqrt{\epsilon_0}, \lambda) \cdot \left(\|x^{\|}\|_2, \|x^{\perp}\|_2\right)^{\intercal} && (14)\\
&\leq \sqrt{\epsilon_0 + \lambda^2}\|x\|_2 && \text{(Follows from Cauchy-Schwarz)}
\end{aligned}$$

□

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.

$$\Pr[R' \text{ errs}] = \Pr[\text{At least } \frac{t}{2} \text{ of } \rho \text{ fall in the set on which } R \text{ errs}] \tag{15}$$

$$\leq \sum_{B \subseteq [t], |B| \geq \frac{t}{2}} \Pr[(\forall i \in B) i^{\text{th}} \text{ step lies in the bad set for } R] \tag{16}$$

$$= \sum_{B \subseteq [t], |B| \geq \frac{t}{2}} \|M_t A M_{t-1} A \dots M_2 A M_1 A M_0 u\|_1 \tag{17}$$

$$\leq \sum_{B \subseteq [t], |B| \geq \frac{t}{2}} \sqrt{2^r} \|M_t A \dots M_1 A M_0 u\|_2 \tag{18}$$

$$\leq \sum_{B \subseteq [t], |B| \geq \frac{t}{2}} \sqrt{2^r} \left(\sqrt{\epsilon_0 + \lambda^2}\right)^{|B|} \|u\|_2 \tag{19}$$

$$= \sum_{B \subseteq [t], |B| \geq \frac{t}{2}} \left(\sqrt{\epsilon_0 + \lambda^2}\right)^{|B|} \tag{20}$$

$$\leq 2^t \cdot \left(\sqrt{\epsilon_0 + \lambda^2}\right)^{\frac{t}{2}} \tag{21}$$

$$= (4\sqrt{\epsilon_0 + \lambda^2})^{\frac{t}{2}} \leq \epsilon. \tag{22}$$

Line (16) is an upper bounds the actual probability because it is over counting the bad strings. This probability is rewritten as a product of matrices in (17) where

$$M_i = \begin{cases} P & i \in B \\ I & \text{otherwise.} \end{cases}$$

Line (19) follows from repeated applications of Lemma 1. 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 $4\sqrt{\epsilon_0 + \lambda^2} \geq 1$, then we can decrease ϵ_0 by running R a constant number of times, which is the randomness ineffecient approach to amplification. The better option is to preform a random walk of length $c \cdot t$ where c is a constant. Just like in the deterministic case, this has the effect of replacing λ with λ^c. Because we are free to decrease ϵ_0 or λ, we can get $4\sqrt{\epsilon_0 + \lambda^2} < 1$.

If $4\sqrt{\epsilon_0 + \lambda^2} < 1$, then 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'.

3 Other Results

A stronger result which considers the variance of random walks is known as the Expander Chernoff Bound. It states that the fraction of times a random 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 $B_i \subseteq V$. Then the probability that the walk varies from the the expected

number of steps in the bad sets can be written as

$$\Pr\left[\sum_{i}^{t}(X_i - \mu(B_i)) \geq \alpha t\right] \leq e^{-b\alpha^2(1-\lambda)t}, \tag{23}$$

where b is some universal constant and $\alpha \geq 0$. The probability that the walk varies from the expectation for a constant α decreases exponentially as t increases linearly. 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).

4 Next Time

In the 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.