| | |
|---|---|
| **CS 880: Advanced Complexity Theory** | 2/17/06 |

# Lecture 10: Two-Sided Error Reduction with Expanders

Instructor: Dieter van Melkebeek — Scribe: Bess Berg and Jurgen Van Gael

We continue our discussion of applying expander graphs to derandomization. Previously we saw deterministic amplification and also randomness-efficient error reduction, both for the case of one-sided error. In this lecture we consider the case of two-sided error. The algorithms we use are almost the same as before with one-sided error, but the analyses are slightly more complicated. We will end with the statement and proof of the Chernoff bound for random walks on expanders, a result we will use in future lectures.

## 1 Deterministic Amplification with Two-Sided Error

Consider a probabilistic polynomial-time machine $M$ that decides some language $L$, taking input $x$ along with a random string $\rho$, $|\rho| = |x|^r$, such that

$$\begin{aligned} x \in L \quad &\Rightarrow \quad \Pr_\rho[M(x,\rho) = 1] \geq 1 - \epsilon \\ x \notin L \quad &\Rightarrow \quad \Pr_\rho[M(x,\rho) = 1] \leq \epsilon \end{aligned}$$

for some constant $\epsilon \in (0, 1/2)$. Our goal is to reduce the error $\epsilon$ without requiring more random bits than we are already using, though possibly increasing the running time a little bit. We will construct a machine $M'$ that takes input $x$ and random string $\rho'$, $|\rho'| = |x|^r$, such that $M'$ also decides $L$, but with error probability substantially less than $\epsilon$. As before, we assume $M'$ has efficient access to some expander graph on $2^{|\rho'|}$ vertices, each vertex corresponding to a binary string of length $|\rho'|$. $M'$ will consider the set of all vertices $\rho$ within distance $s$ of $\rho'$ and then run the original machine $M$ once on each such $\rho$ (along with the original input $x$), outputting the majority vote of these trials. Note that the running time increases (though it remains polynomial for the right choice of $s$) but the amount of randomness required remains the same as for the original machine $M$.

What kind of reduction in error can we expect? We first consider the case $s = 1$; it can later be generalized. For our analysis we call upon the expander mixing lemma. Recall that this lemma states that for any graph $G = (V, E)$ and any $S, T \subseteq V$, we have

$$\left| E(S,T) - \frac{d|S||T|}{N} \right| \leq \lambda_2 d \sqrt{|S||T|},$$

where $E(S,T)$ denotes the number of edges that go between the sets $S$ and $T$. Fix $x$ and let $S = B'$, $T = B$, where, as before, $B'$ is the set of $\rho'$ on which $M'$ errs, and $B$ is the set of $\rho$ on which $M$ errs. For a moment assume we are back in the case of one-sided error. Then we have $E(B', B) = d|B'|$, since by the definition of $B'$, for every $\rho' \in B'$ we have $\Gamma(\rho') \subseteq B$. The expander mixing lemma then gives us the following series of equations:

$$
\begin{aligned}
d|B'|\left(1-\frac{B}{N}\right) &\leq \lambda_2 \cdot d\sqrt{|B||B'|} \\
|B'| &\leq \frac{\lambda_2^2}{\left(1-\frac{|B|}{N}\right)^2}\cdot|B| \quad \leq \quad \frac{\lambda_2^2}{(1-\epsilon)^2}\cdot|B| \\
\epsilon' \equiv \frac{|B'|}{N} &\leq \frac{\lambda_2^2}{(1-\epsilon)^2}\cdot\epsilon
\end{aligned}
$$

Now, for the case of two-sided error, we can run the same analysis, but with $E(B,B') \geq \frac{d}{2}|B'|$ (by definition of $B'$, the vertices in $B'$ are those for which at least half of their neighbor set is in $B$). Carrying through we get

$$\epsilon' \leq \frac{\lambda_2^2}{(\frac{1}{2}-\epsilon)^2}\cdot\epsilon$$

Of course, for $\epsilon$ tending towards $\frac{1}{2}$, this may not be an improvement! To fix this, we can increase $s$, and note that the same analysis will go through if we think of still just taking one step out from the original $\rho'$ but now on the expander graph that is the $s^{th}$ power of the original expander. That is, we simply need to replace $\lambda_2$ with $\lambda_2^s$ to understand the reduction of error for $s > 1$. Therefore, though the above term $\epsilon/(\frac{1}{2}-\epsilon)^2$ can be very small, it is still just a constant and can be dominated by the $\lambda_2^s$ term with the appropriate choice of $s$. For appropriate parameters $d$ and $\lambda_2$ of the expander graph, we can set $s$ such that $\epsilon'$ is at most $1/p(n)$ for any polynomial $p$ and also such that the runtime increases only by some polynomial factor.

## 2 Randomness-Efficient Error Reduction

Our goal in this section is still to reduce the error, but this time we allow ourselves to use a little extra randomness in order to make $\epsilon'$ exponentially small. As in the case of one-sided error we will construct a machine $M'$ that takes input $x$ along with some random string $\rho'$. $M'$ uses this random string $\rho'$ as the start of a random walk on an expander graph, considering the nodes $\rho_0 = \rho', \rho_1, \ldots, \rho_s$ on this walk and running $M$ with each of them (always on the original input $x$), and outputting the majority vote of these $s$ trials.

Now how much error reduction can we expect? For a fixed input $x$, the probability that $M'$ errs is the probability that more than half of the nodes in its random walk are bad. That is, we have

$$\epsilon' \quad \equiv \quad \Pr[M' \text{ errs}] \quad = \quad \Pr[\text{more than } \frac{s}{2} \text{ of the } \rho_i\text{'s are in } B].$$

We break down this probability into a union of bad events, and once we have upper bounded the probability of a single bad event, we take a union bound. Consider a set of indices, $0 \leq i_0 < i_1 < \ldots < i_k \leq s$, for some $k \geq \frac{s}{2}$. We consider the probability that this set indexes bad steps of the random walk:

$$\Pr[\ \forall(0 \leq j \leq k)\ \rho_{i_j} \in B] \quad = \quad \left\| \left(\prod_{j=1}^{k} RA^{i_j - i_{j-1}}\right) RA^{i_0}u \right\|_1$$

As in earlier lectures, $A$ is the normalized adjacency matrix that defines one step of the random walk on the expander graph, $R$ is the projection onto the bad set $B$, and $u$ is the uniform distribution over all vertices. Really the righthand side of the above equation is just the probability weight that remains when we force the random walk to pass through bad vertices at the designated indices. We proceed to upperbound this righthand side by (1) switching from the 1-norm to the 2-norm and thus picking up a factor of $\sqrt{N}$, (2) making use of the fact that $u$ is invariant under the operator $A$ and thus $A^{i_0}u = u$, and (3) making use of the fact that $R^2 = R$ because $R$ is a projection. We therefore obtain the upperbound:

$$\left\| \left( \prod_{j=1}^{k} RA^{i_j - i_{j-1}} \right) RA^{i_0} u \right\|_1 \quad \leq \quad \sqrt{N} \left\| \left( \prod_{j=1}^{k} RA^{i_j - i_{j-1}} R \right) u \right\|_2 .$$

We proved in a previous lecture that $\|RARx\|_2 \leq (\epsilon + (1-\epsilon)\lambda_2)\|x\|_2$ for any vector $x$. This holds for any expander with random walk operator $A$, and so when we are working with powers of $A$, say $A^{i_j - i_{j-1}}$, we just consider the expander that is the appropriate power of the original expander and therefore has as its second largest eigenvalue $\lambda_2^{i_j - i_{j-1}} \leq \lambda_2$. We are therefore able to upperbound the above righthand term with

$$\sqrt{N}(\epsilon + (1-\epsilon)\lambda_2)^k \|u\|_2 \quad = \quad (\epsilon + (1-\epsilon)\lambda_2)^k.$$

We have now upperbounded the probability of the bad even associated with random walk steps $i_0, i_1, \ldots, i_k$. In fact, we only need consider the cases $k = \frac{s}{2}$, since their union certainly covers all cases $k > \frac{s}{2}$. We obtain with a union bound:

$$\begin{aligned}
\epsilon' &= \Pr[M' \text{ errs}] \\
&\leq \sum_{0 \leq i_0 < i_1 < \ldots < i_{s/2} \leq s} (\epsilon + (1-\epsilon)\lambda_2)^{\frac{s}{2}} \\
&\leq \frac{1}{2} \cdot 2^{s+1} (\epsilon + (1-\epsilon)\lambda_2)^{\frac{s}{2}} \\
&\leq (4(\epsilon + (1-\epsilon)\lambda_2))^{\frac{s}{2}}
\end{aligned}$$

The term $4(\epsilon + (1-\epsilon)\lambda_2)$ is a constant, and for small enough $\lambda_2$ it becomes less than 1. Then for example if we choose $s$ linear in the input size, we will err with exponentially small probability.