CS 830: Randomness and Complexity 2/1/05

Lecture 5: A linearly preserving reduction using expander graphs

Instructor: Jin-Yi Cai Scribe: Aparna Das, Giordano Fusco

1 Overview

The reduction we present today is similar to the reduction we saw in the previous lecture. Like the last reduction this one is also linearly preserving. The main advantage with today's reduction is that it uses only $O(n)$ public random bits (compared to the $O(n2\log(\frac{1}{w(n)}))$ used by the last reduction). We get away with using less public random bits mainly by using explicitly constructed constant degree expanders.
We first introduce expander graphs and then describe the reduction. We end the lecture with a few definitions pertaining to the next topic.

2 Explicit construction of expanders

The following two papers illustrate some details about the construction of expanders:

- N. Alon, *Eigenvalues and expanders*, Combinatorica, 6(2), pp. 83–96, 1986.
- O. Gabber, Z. Galil, *Explicit Constructions of Linear-Sized Superconcentrators*, J. Comput. Syst. Sci., 22(3), pp. 407–420, 1981.

In Alon's paper unimodular matrices are use to construct expanders form $\mathbb{Z}_p2$ to $\mathbb{Z}_p2$. The four matrices used are

$$A = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \qquad A^{-1} = \begin{pmatrix} 1 & -1 \\ 0 & 1 \end{pmatrix}$$

$$B = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} \qquad B^{-1} = \begin{pmatrix} 1 & 0 \\ -1 & 1 \end{pmatrix}$$

The following theorem is proved in this paper.

Theorem 2.1. *Given a bipartite graph (L, R, E) and $n = p2$, then for every $S \subseteq L$,*

$$\frac{|\Gamma(S)|}{|S|} \geq 1 + \left(1 - \frac{\sqrt{3}}{2}\right)\left(1 - \frac{|S|}{n}\right)$$

where $\Gamma(S)$ is the set of neighbors of S.

The meaning of this theorem is that if we consider a set $V = 2^n$, then $\forall S \subseteq V$ with $|S| \leq n/2$ there is an expander s.t. $|\Gamma(S)| \geq (1 + \theta)|S|$.

Exercise 2.2. *What is the probability that given a set T and a set S, $|S| < |T|$, taking 3 random points in S, their neighbors are in T? (Hint: $\binom{n}{|T|}\frac{1}{2^{3|S|}}$)*

Alon gives an explicit construction of the expander, but he does not give explicit constants. Instead Gabber and Galil give a very explicit construction with explicit constants.
Note that the matrices above are not the only possible ones. We can build an expander using any unimodular matrix, its inverse and its transpose.

3 Linearly preserving reduction using linear public random bits

The overall structure of today's reduction is similar to the one we saw last time. This reduction also uses a two step process. First we reduce a weak one-way function f to a $\frac{1}{8}$-weak one-way function g. Then we convert g into a strong one-way function h.
As before we take $\ell = O(\log \frac{1}{w(n)})$ and $N = 2^{\ell}$. We will view all the inputs $\{0,1\}^n$ of f as the vertices of a constant degree expander. We will replace the n bit strings $\pi_1 \dots \pi_\ell$ with $\rho_1, \dots, \rho_\ell$ where each $\rho_i \in_{\mathcal{U}} \{0,1\}^d$ and d is a constant that depends on the degree of the expander. Each ρ_i can be thought of as selecting some random neighbor of a vertex in the expander graph. Thus, whenever we would have computed $z' = z \oplus \pi$, in the previous reduction, we will now follow the ρ_i edge out of vertex z to end up on some other vertex z'.
We define g using the recursive function s as follows:

Definition 3.1.

$$g(x; \rho_1, \dots, \rho_\ell) = (s(x; \rho_i, \dots, \rho_\ell); \rho_1, \dots, \rho_\ell)$$

where

$$s(x; \rho_1, \dots, \rho_i) = s(\rho_i(s(x; \rho_1, \dots, \rho_{i-1})), \rho_1, \dots, \rho_{i-1})$$

and

$$s(x; \rho_1) = f(\rho_1(f(x)))$$

We can interpret $s(x; \rho_1, \dots, \rho_{i-1})$ as one of the 2^n nodes, and $\rho_i(s(x; \rho_1, \dots, \rho_{i-1}))$ as its ρ_i-th neighbor.
As seen last time, we can also compute g iteratively as:

$$\begin{aligned}
y_1 &= x \\
y_2 &= \rho_1(f(y_1)) \\
&\vdots \\
y_i &= \rho_{m_i}(f(y_{i-1})) \\
&\vdots \\
y_N &= \rho_{m_N}(f(y_{N-1}))
\end{aligned}$$

3.1 Going from f to g

We use the same $(L, R, E)-$bipartite graph as in the previous reduction. L contains 2^n vertices, each representing an input to f. R contains $2^n 2^{O(\ell)}$, each representing an input $\langle z; \rho_1, \ldots, \rho_\ell \rangle$ to g. As before we connect $x \in L$ to $\langle z; \rho_1, \ldots, \rho_\ell \rangle$ (on the right) iff x equals one of the y_i produced during the computation of g on $\langle z; \rho_1, \ldots, \rho_\ell \rangle$.
Take a set $S' \subseteq L$ and define F_i as:

$$F_i = F_{\{\rho_1, \ldots, \rho_\ell\}} = \{x \mid \text{ at least one of } y_1, \ldots, y_{2^i} \in S'\}$$

Our aim is to show that this graph has the forward expansion property. To do so we will show that the size of F_i increases geometrically with i. We will show that $|F_i| \geq (1 + \theta')|F_{i-1}|$. At this point we can use the same analysis from the previous reduction to conclude that the forward expansion property holds.
We start by noting that F_i consists of the points in F_{i-1} and also those points which make it into set S after another application of function s. So,

$$F_i = F_{i-1} \cup s_{i-1}^{-1}(\rho_i^{-1}(F_{i-1}); \rho_1, \ldots, \rho_{i-1})$$

Since s is a 1-1 function, applying s_{i-1} to the entire right hand side, will not affect the size of F_i. We get:

$$|F_i| = |s_{i-1}(F_{i-1}) \cup \rho_i^{-1}(F_{i-1})|$$

Using the inclusion-exclusion principle we obtain:

$$|F_i| = |s_{i-1}(F_{i-1})| + |\rho_i^{-1}(F_{i-1})| - |s_{i-1}(F_{i-1}) \cap \rho_i^{-1}(F_{i-1})|$$

Since ρ_i^{-1} is also a one-to-one mapping, $|F_{i-1}| = |\rho_i^{-1}(F_{i-1})|$. For the same reason $|s_{i-1}(F_{i-1}) \cap \rho_i^{-1}(F_{i-1})| = |\rho_i(s_{i-1}(F_{i-1})) \cap F_{i-1}|$.
This lets us simplify the above expression to:

$$|F_i| = 2|F_{i-1}| - |\rho_i(s_{i-1}(F_{i-1})) \cap F_{i-1}|$$

We like to argue that the intersection is not too big.

Claim 3.2.

$$E[|\rho_i(s_{i-1}(F_{i-1})) \cap F_{i-1}|] \leq (1 - \frac{\theta}{d}|F_{i-1}|)$$

We will use the claim for now and defer the proof until later. Applying the claim we get:

$$\begin{aligned} E[|F_i|] &\geq 2|F_{i-1}| - (1 - \frac{\theta}{d})|F_{i-1}| \\ &\geq (1 + \frac{\theta}{d})|F_{i-1}| \end{aligned}$$

This result says that, going from $i-1$ to the i-th step, $|F_i|$ grows geometrically by some amount greater than one. We will apply this "growing-step" $\ell = O(\log n)$ times. This is enough to get a $\frac{1}{8}$-weak g using the last part of analysis given in the previous lecture.
Now we go back and prove the claim stated above. For better readability we denote:

- $T = s_{i-1}(F_{i-1}; \rho_1, \ldots, \rho_{i-1})$
- $T_i = \rho_i(T)$ and
- $S = F_{i-1}$.

Proof. By the properties of the expander:

$$|T_1 \cup \cdots \cup T_d| \geq (1+\theta)|T| = (1+\theta)|S| \tag{1}$$

where $d =$ degree of the (constant degree) expander.
Note that for any i, $|T_i| = |T_i \setminus S| + |T_i \cap S| \leq |T_i \setminus S| + |S|$. From this it follows that

$$|T_1 \cup \cdots \cup T_d| \leq |S| + \sum_{i=1}^{d} |T_i \setminus S| \tag{2}$$

We wish to calculate the size of $|T_i \cap S|$ when i is picked uniformly. We are going to bound its expectation.

$$E|T_i \cap S| = |T_i| - E|T_i \setminus S| \tag{3}$$

To get a bound on $E[|T_i \setminus S|]$ we start with the lower and upper bounds for $|T_1 \cup \cdots \cup T_d|$ obtained in equations (1) and (2):

$$(1+\theta)|S| \leq |T_1 \cup \cdots \cup T_d| \leq |S| + \sum_{i=1}^{d} |T_i \setminus S| \tag{4}$$

Simplifying the above we get:

$$\theta|S| \leq \sum_{i=1}^{d} |T_i \setminus S|$$

Plugging in the fact that $E[|T_i \setminus S|] = \frac{1}{d}\sum_{i=1}^{d} |T_i \setminus S|$, we get:

$$E[|T_i \setminus S|] \geq \frac{\theta}{d}|S| \tag{5}$$

Since $|S| = |T_i|$, plugging in (5) into (3) we get

$$E[|T_i \cap S|] \leq |S| - \frac{\theta}{d}|S| = (1 - \frac{\theta}{d})|S|$$

□

3.2 Going from $\frac{1}{8}$-weak to strong one-way

The second step of the reduction is to build a strong one-way function h from the $\frac{1}{8}$-weak g derived above. We would like h to use only $O(n)$ public random bits.
In order to do this we will use a new constant degree expander. The expander will have $2^{n+d\ell}$ vertices, each corresponding to an input of g, which is of the form $\bar{x} = (x; \rho_1, \ldots \rho_\ell)$. h will use $m = O(n)$ random strings $\alpha_1, \ldots, \alpha_m$ where each $\alpha_i \in \{0,1\}^c$ and c is the degree of the expander. Each α_i can be thought of as selecting some random neighbor of a vertex in the expander graph.
We define h as follows:

Definition 3.3.

$$h(x;\rho_1,\dots,\rho_\ell;\alpha_1,\dots,\alpha_m) = \langle y_{m+1};\rho_1,\dots,\rho_\ell;\alpha_1,\dots,\alpha_m\rangle$$

where

$$g(y_i,\psi_i) = \langle y_i',\psi_{i+1}\rangle \text{ and } (y_{i+1},\psi_{i+1}) = \alpha_i(y_i',\psi_i)$$

and, $\psi_i = \langle\rho_1,\dots,\rho_\ell\rangle$ *and* $y_1 = x$.

Intuitively h works as follows. Given x, $\rho_1,\dots,\rho_\ell$ and $\alpha_1,\dots,\alpha_m$, h starts at the point in the expander defined by $\langle x;\rho_1,\dots,\rho_\ell\rangle$ and performs the next step m times: apply g to current point then move to a new random point by applying α_i. This is shown pictorially below:

$$\begin{array}{c}
(y_1,\psi_1)\\
\downarrow g\\
(y_1',\psi_1)\\
\downarrow \alpha_1\\
(y_2,\psi_2)\\
\downarrow g\\
(y_2',\psi_2)\\
\downarrow \alpha_2\\
\cdots\\
\downarrow \alpha_{m-1}\\
(y_m,\psi_m)
\end{array}$$

Note that h does not exactly perform a random walk since it intermediately transforms points using g.

As before we can think of a bipartite graph (L,R,E) where the nodes of L represent the inputs to g and the ones on the right represent inputs to h. For $x = \langle x;\rho_1,\dots\rho_\ell\rangle \in L$ there is an edge to a point $r = \langle z;\rho_1,\dots,\rho_\ell;\alpha_1,\dots,\alpha_m\rangle \in R$ if x equals any y_i produced during the computation of $h(r)$. (This includes all y's produces within the calls to all the intermediary g's).

We can be argue that after $O(n)$ steps we will have traversed set $S \subset L$, showing that the graph has the forward expanding property.

Exercise 3.4. *What would it happen if instead of*

$$s_{i-1}(\rho_i(s_{i-1}(x;\rho_1,\dots,\rho_{i-1})),\rho_1,\dots,\rho_{i-1})$$

we do something else? For example what if instead of applying a permutation to s_{i-1}*, we apply it to* s_1*?*

4 Definitions for next time

DEFINITION 4.1 (YAO). *A polynomial-time computable* pseudo random generator *is a function $g : 2^n \to 2^{l(n)}$, where $n < l(n) \leq n^c$. $g(x)$ is $s(n)$-secure if for all polynomial-time bounded adversary A*

$$\frac{T_A(n)}{\delta(n)} \geq s(n)$$

where,

$$\delta(n) = |\Pr_X[A(g(X)) = 1] - \Pr_Y[A(Y) = 1]|$$

with $X \in_{\mathcal{U}} \{0,1\}^n$, $g(x) : \{0,1\}^n \to \{0,1\}^{l(n)}$ and $Y \in_{\mathcal{U}} \{0,1\}^{l(n)}$.

DEFINITION 4.2. *A function $g(x)$ is $s(n)$-secure* next bit unpredictable *if for all polynomial-time bounded adversary B*

$$\frac{T_B(n)}{\delta(n)} \geq s(n)$$

where,

$$\delta(n) = \Pr_{I,X}[B(g(X)_{\{1,\ldots,I-1\}}, I) = g(X)_I] - \frac{1}{2}$$

with $X \in_{\mathcal{U}} \{0,1\}^n$, $I \in_{\mathcal{U}} \{0,\ldots,l\}$.

In the next lecture we will prove that these two concepts are equivalent.