CSE522/Math581EB Coding Theory **Fall 2001**

Lecture 9: November 6

Lecturer: Elchanan Mossel *Scribe: Karl Schwede*

What we were looking at in this lecture is linear time encoding and decoding since normally these processes take greater than n^2 time. Specifically, we'll only discuss now linear time decoding. The main idea is to use linear codes with sparse matrices, that is we looked at LDPC (Low Density Parity Check) codes. The main objective in the next couple of talks is to prove

Theorem 9.1 *There exists a family of "good" (constant relative rate and distance) codes which are encodable/decodable in linear time. They can be constructed in polynomial time.*

We start with description of codes with linear time decoding following [SS]. To do this we view codes as *bi-partite* graphs with left, and right vertices and edges between the sets. That is $G = (L, R, E)$, where L is the variables of the code, R is a set of a constraints and an edge (l, r) connects a variable to a restraint if the variable x_l is involved in the constraint r. For example if we had a constraint that said $x_1 + x_2 = 0$, then we would have an edge from that constraint to x_1 and an edge to x_2. Low density means we have a "small" number of edges. Also note that if $|L| = n$ and $|R| = m$, then $k \geq n - m$.

Now we have a few definitions. Note all our graphs as above, of the form (L, R, E) (left vertices, right vertices and edges).

Definition 9.2 *A graph G is (d_L, d_R) bounded if the degrees on the left are bounded by d_L and the degrees on the right are bounded by d_R. A graph G is (d_L, d_R) regular if all the degrees on the left are d_L and all the degrees on the right are d_R.*

Definition 9.3 *G is an (α, δ) expander if for all $S \subseteq L$ ($|L| = n$) such that $|S| < \delta n$, $|\partial(S)| \geq \alpha|S|$, where $\partial(S) = \{r \in R | \exists s \in S, (s, r) \in E\}$.*

Theorem 9.4 *If G is (d_R, d_L) bounded and an (α, δ) expander then C_G has distance $\frac{2\alpha\delta}{d_L}$, provided $2\alpha > d_L$.*

Note that a (d_L, d_R) bounded graph must have $\alpha \leq d_L$.

Proof: Choose $x \neq 0$ and suppose $wt(x) < \frac{2\alpha\delta}{d_L}$, we want to show that $x \notin C_G$. Let $S = \{i | x_i \neq 0\}$. Let $A = \{r \in \partial(S) | r \text{ has 1 neighbor in } S\}$ and let $B = \{r \in \partial(S) | r \text{ has more than 1 neighbor in } S\}$. Note a restriction must involve variables so $\partial(S) = A \cup B$, and since A and B are disjoint $|\partial(S)| = |A| + |B|$. The maximum number of left vertices that a variable can connect to is d_L (by assumption), so $|A| + 2|B| \leq d_L|S|$, since each element of B connects to at least 2 elements of S. So since $B = \partial(S) - A$, we have $|A| + 2(|\partial(S)| - |A|) \leq d_L|S|$, or $|A| \geq 2|\partial(S)| - d_L|S|$. Now we have 2 cases,

- If $s = |S| < \delta n$ then $2|\partial(S)| - d_L s \geq 2\alpha|S| - d_L|S| > 0$ since G is an (α, δ) expander, (recall $2\alpha > d_L$ by assumption), thus $|A| > 0$.
- On the other hand if $|S| \geq \delta n$ we have that $2|\partial(S)| - d_L|S| \geq 2\alpha\delta n - d_L|S| > 2\alpha\delta n - d_L(\frac{2\alpha\delta n}{d_L}) = 0$ (note $wt(x) = |S| < \frac{2\alpha\delta n}{d_L}$ our desired distance).

Therefore in either case $|A| > 0$ so there is a constraint that involves exactly one of x's non-zero bits, so x cannot satisfy it, and thus $x \notin C_G$. ■

Theorem 9.5 *If G is as before and additionally $\alpha > 3/4d_L$ then the code corrects up to $(\frac{2\alpha - d_L}{d_L}\delta)n$ errors in linear time.*

The algorithm we use is as follows. If we receive a word x and if x_l is a neighbor of more unsatisfied constraints than satisfied ones, flip x_l. We continue flipping variables until we cannot continue.

Proof: Set $\epsilon = \frac{2\alpha - d_L}{d_L}\delta$, we will now show that if we start with a word with less than ϵn ones, we terminate with 0 (the zero word). Set $S = \{i | x_i = 1\}$, $s = |S|$, $U = \{$ unsatisfied constraints $\}$, $u = |U|$. Set A as in the previous proof.

Claim 1: If $0 < s \leq \delta n$ then $u \geq (2\alpha - d_L)s$. This is exactly the same as it was done in the previous theorem since $|U| \geq |A|$ which was greater than the listed quantity.

Claim 2: At all stages of the algorithm $s < \delta n$. This is true at the first step since $s < \epsilon n < \delta n$. Also at the first step $u \leq d_L s < \epsilon n d_L = (2\alpha - d_L)\delta n$. Note at each step, the total number of unsatisfied constraints will decrease and s can increase by at most 1, so after one step $s \leq \delta n$ but then the previous claim is still valid (and so is the statement about u) so by the previous statement $s \leq \frac{u}{2\alpha - d_L} < \delta n$. Thus this claim is proved as well.

Finally we want to show that if $0 < s < \delta n$, there exists $i \in S$ with greater than $d_L/2$ unsatisfied constraints (obviously if $s = 0$ we are done). Again $u \geq (2\alpha - d_L)s > 2(3/4d_L - d_L)s = d_L s/2$ so there is $i \in S$ with more than $d_L/2$ neighbors in U (by the pigeon hole principle).

We should also comment on why the algorithm takes linear time. Note at each step of the algorithm u decreases. When u is zero, the algorithm terminates. Since $u = O(n)$, if follows that we perform only a linear number of steps. It remains to show that each step of the algorithm can be implemented in $O(1)$ operations. In order to do that, we have a data structure for the graph, and we construct a list of all variables which have more unsatisfied neighbors than satisfied ones (such a list is constructed in linear time). Whenever we flip a variable in the list we re-check all variables which may be affected by this flip (neighbors of the constraints which involve the variable), and update the list accordingly. Since this takes a constant number of operations, the claim follows. ■

Now the question is how to find (d_L, d_R) bounded expanders with $\alpha > 3/4d_L$. Unfortunately there are no explicit constructions which achieve that. However, random (d_L, d_R) regular graphs have the desired property with high probability.

In order to obtain explicit constructions, we will look at extended LDPC codes and derive some analogous results.

Definition 9.6 *Let $G = (L, R, E)$ be a (d_L, d_R) regular graph, and let D be a $[d_R, l, \Delta]$ code. The code $C_{G,D}$ is defined as follows: x is a codeword (string of $n = |L|$ bits) if for all $r \in R$, its neighbors $x_{l_1}, \ldots, x_{l_{d_R}}$ are a codeword of D.*

This time we have that if $|L| = n$ and $|R| = m$, then $k \geq n - (d_R - l)m$.

Theorem 9.7 *If G is an (α, δ) expander and D a code as above, then the code $C_{G,D}$ has distance $\frac{\alpha\Delta}{d_L}\delta n$ assuming $\alpha\Delta > d_L$.*

Proof: Take $x \neq 0$ with $wt(x) < \frac{\alpha\Delta}{d_L}\delta n$. Set $S = \{i | x_i = 1\}$. As before set $A = \{j \in R | j$ has less than Δ neighbors in $S\}$ and $B = \{j \in R | j$ has greater than or equal to Δ neighbors in $S\}$. We have $\partial(S) = A \cup B$ and

$|\partial(S)| = |A|+|B|$. We need to show that $|A| > 0$. Clearly, $|A|+\Delta|B| \leq d_L|S|$ or $|A|+\Delta(|\partial(S)|-|A|) \leq d_L|S|$ so that $|A| \geq \frac{\Delta|\partial(S)|-d_L|S|}{\Delta-1}$.

- If $|S| < \delta n$, we have $|A| \geq \frac{\Delta\alpha|S|-d_L|S|}{\Delta-1} > 0$, since $\alpha\Delta > d_L$.
- On the other hand, if $|S| \geq \delta n$, we have $\Delta\alpha|\partial(S)| - d_L|S| \geq \Delta\alpha\delta n - d_L|S| > \Delta\alpha\delta n - d_L\frac{\Delta\alpha\delta}{d_L}n > 0$, since $|S| < \frac{\Delta\alpha\delta}{d_L}n$.

Therefore in both cases $|A| > 0$, so there exists $r \in R$ with less than Δ neighbors in S, the set of nonzero coordinates of x. If $x \in C_{G,D}$, then r's neighbors (evaluated at x) are a codeword of D, but since D has distance Δ this is impossible, so $x \notin C_{G,D}$. ■

An analogous theorem to theorem 9.5 allows one to build explicit constructions of linear time decodable codes using extended LDPC and the Lubotzky-Phillips-Sarnak [LPS] explicit construction of expander graphs.

The proof that LPS construction actually gives an expander is complicated. However, the description of the construction is quite simple. The vertices of their expander are the elements of

$$PSL(2, F_q) = \{A \in M_{2x2}(F_q) | \det A = 1\}.$$

The edges are (M_1, M_2) where $M_2 = E_iM_1$, where $\{E_i\}$ is a set of generators. In other words the graph is just the Cayley graph of $PSL(2, F_q)$, with respect to the generators E_i.

Since we are interested in unbalanced bi-partite graphs, we look at the "edge vertex incident graphs" of the LPS construction. Taking $G = (V, E)$ to be such an expander, we construct a new graph $G' = (V', E')$ as follows, $V' = V \cup E$, and $(v, e) \in E'$ if v is an endpoint of e.

Using this graph together with good codes in the extended LDPC setting yields the required explicit construction of linear time decodable codes.

References

[1] M Sipser and D. A. Spielman (1996) Expander codes. Codes and complexity. *IEEE Trans. Inform. Theory* **42 no. 6, part 1**, 1710–1722.

[2] A. Lubotzky, R. Phillips and P. Sarnak (1988) Ramanujan graphs. *Combinatorica* **8, no. 3**, 261–277.