6.895 Essential Coding Theory	November 1, 2004

Lecture 15: Expander Codes

Lecturer: Madhu Sudan *Scribe: Vinod Vaikuntanathan*

Overview of this lecture: We study a new family of codes, based on expander graphs, that gives us "good" rate and relative distance, and *linear time decoding algorithms.*

Some History

How efficient can decoding of linear codes be ? Gallager addressed this problem in his thesis (from 1965): he came up with codes that have a sparse parity-check matrix (also called Low-density Parity Check or LDPC codes), and gave a decoding algorithm for such codes that he observed works well in practice. However, he did not prove rigorous bounds on the performance of the decoding algorithm. He also proved that a random LDPC code is asymptotically "good" with high probability.

Tanner, in 80's studied LDPC codes, and proved that the performance of Gallager's algorithm is related to the girth of the (natural) bipartite graph representation of the parity-check matrix. Sipser and Spielman were the first to analyze the codes in terms of expansion. They showed that good expansion implies good distance of the code, as well as a good decoding algorithm.

1 Graph-Theoretic Codes

Lets look at a candidate construction of LDPC codes. Choose the $n \times m$ parity-check matrix H as follows: (say $m = \frac{n}{2}$)

$$H_{ij} = \begin{cases} 0 & \text{with probability } 1 - \frac{10}{n} \\ 1 & \text{with probability } \frac{10}{n} \end{cases}$$

The probability that a row of H is the all 0 vector, is $(1 - \frac{10}{n})^{n/2}$, which is some constant. The expected number of rows that are all 0 is, therefore, cn. What does this do to the distance of the code ? An all 0 row in the parity-check matrix means that the corresponding bit of the message does not affect the codeword. In other words, take a codeword, flip a bit and we get another codeword (The distance is 1 !!)

Nevertheless, at this point, we assert that there is a way to show that $\exists$ parity-check matrices H such that H has a constant number of 1's per column, that gives an asymptotically good code. The following are roughly the assertions made in Gallager's thesis about such codes. (The latter assertion was proved)

1. There is a natural algorithm that "seems to" correct a large fraction of errors.
2. We know that for a relative distance δ, random codes give a rate of $1 - H(\delta)$. The assertion about the graph-theoretic codes is that $\forall \epsilon\ \exists H$ having at most d $1's$ per column, such that H is the parity-check matrix of a code of relative distance δ and rate $1 - H(\delta) - \epsilon$.

2 Graph Theoretic Framework

The $n \times m$ parity-check matrix H can be considered as a bipartite graph G_H, in a natural way. Each row of H is a left vertex of the graph and each column is a right vertex. If $H_{ij} = 1$, then there is an edge between the i^{th} left vertex and the j^{th} right vertex. A codeword $x \in \{0,1\}^n$ is associated with an assignment to the left vertices. The corresponding assignment to the right vertices is the result of the product $H \cdot x$. Therefore, an assignment to the left vertices is a codeword iff the assignment induced on the right vertices is 0^m. Thus, the bipartite graph G_H defines a code in a natural way. The requirement of low-density says that each right-vertex should have small (constant) degree.

To get a large rate, we require that m is much less than n. How do we get large distance ? Consider a subset of the left vertices S, and the set of their neighbors $\Gamma(S)$. If $x \in \Gamma(S)$ is connected to an even number of vertices in S, then the assignment that (naturally) corresponds to the subset S is a codeword. Therefore, informally, we want that, for each sufficiently small subset S of the left vertices, all the vertices in $\Gamma(S)$ have an even number of neighbors in S.

Before we move on, we want to introduce some notation: A bipartite graph is defined as $G = (L, R, E \subseteq L \times R)$. For a set $S \subseteq L$, define its neighborhood $\Gamma(S) = \{v \in R \mid \exists u \in S \ s.t \ (u, v) \in E\}$. Further, define $\Gamma_{odd}(S) = \{v \in R \mid \#u \in S \ s.t \ (u, v) \in E \text{ is odd }\}$.

Claim 1 *G yields a code of distance δn if $\forall S$ such that $|S| < \delta n$, $|\Gamma_{odd}(S)| > 0$.*

Proof The proof of the claim is immediate from the fact that each subset $S \subseteq L$ corresponds to a vector in $\{0, 1\}^n$ with Hamming weight $|S|$. ■

Expander Graphs: with $|L| = n$, $|R| = m$ is a (γ, δ) expander if $\forall S, S \subseteq L, |S| \leq \delta n$, $|\Gamma(S)| \geq \gamma|S|$. Graph G is c-regular iff the degree of every left vertex is c.

What can we say about the possible values of γ ? Clearly $\gamma \leq c$. By similar arguments, we can rule out the case that $\gamma > c - 1$. Moreover, this is the best that we can rule out because,

Theorem 2 *$\forall \epsilon$, for all sufficiently large c, there is a c-regular graph with expansion $(1 - \epsilon)c$ and $\delta > 0$.*

δ cannot be too large either. It is easy to see that $\delta n \leq \frac{m}{\gamma}$.

Now, we give a rough sketch of the (non-constructive) proof that such graphs exist. Take a random matching on $2cn$ vertices. Consider cn vertices with no edges between them. Group them into n chunks each of size c. Group the remaining cn vertices into m chunks (Assume, for convenience, that $\frac{cn}{m}$ is an integer). This process defines a c-left-regular bipartite graph with $|L| = n, |R| = m$. We claim is that this is an expander with a high probability.

We introduce some more notations. Recall that $\Gamma(S)$ for $S \subseteq L$ stands for $\{v \in R \mid \exists u \in L, (u, v) \in E\}$ and $\Gamma_{odd}(S)$ stands for $\{v \in R \mid \#u \in Ss.t(u, v) \in E \text{ is odd }\}$. Now, define $\Gamma_{unique}(S) = \{v \in R \mid \exists u \in S \ s.t \ (u, v) \in E\}$. It is easy to see that $\Gamma_{unique}(S) \subseteq \Gamma_{odd}(S)$ for all $S \subseteq L$.

Lemma 3 *In a (γ, δ)-expander with left-degree c, $\Gamma_{unique}(S) \geq (2\gamma - c)|S|$ for all S such that $|S| \leq \delta n$.*

Note: Observe that, for this bound to be non-trivial, we need $\gamma > \frac{c}{2}$. As a historical aside, we note that $\gamma = \frac{c}{2}$ was the classical barrier on the expansion.

Theorem 4 *If G is a c-regular graph with m left vertices and n right vertices, and it is a (γ, δ) expander for some $\gamma > \frac{c}{2}$, then G gives a code of rate $\frac{n-m}{n}$ and (relative) distance δ.*

Proof of Theorem 4 From Lemma 3, we have that for each $S \subseteq L$ with $|S| \leq \delta n$, $\Gamma_{odd}(S) \geq \Gamma_{unique}(S) > 0$. Therefore, no codeword has weight less than δn. Since the code is linear, this implies that the distance is at least δn. Since the parity check matrix is $n \times m$, the rate is $\frac{n-m}{n}$. ■

Proof of Lemma 3 For a set S, write $\Gamma(S)$ as the disjoint union of sets A and B. i.e, $\Gamma(S) = A \cup B$, where $A = \{v \in R \mid \#u \in S \ s.t \ (u, v) \in E \ is \ 1\}$ and $B = \{v \in R \mid \#u \in S \ s.t \ (u, v) \in E \text{ is more than } 1\}$.

$$|A| + 2|B| \leq \text{ \#edges from S to } \Gamma(S) = c|S| \tag{1}$$
$$|A| + |B| \geq \gamma|S| \tag{2}$$

where the first equation is due to counting the number of edges between S and $\Gamma(S)$ in two ways, and the second equation is a restatement of the expansion property of the graph.

$$\begin{aligned} |A| \geq & \quad \gamma|S| - |B| \\ \geq & \quad \gamma|S| - \left(\tfrac{c|S| - |A|}{2}\right) \\ |A| \geq & \quad (2\gamma - c)|S| \end{aligned}$$

■

If the graph is a sufficiently good expander, this gives us that $|A|$ is close to $c|S|$. That is, each vertex on the neighborhood of S has a unique neighbor in S.

3 Sipser-Spielman Algorithm

In this section, we sketch the decoding algorithm for the expander codes due to Sipser and Spielman.

Algorithm Flip(r)
$\mathbf{r}$ is the received vector (of length m). $\mathbf{r}$ defines a labeling of the left vertices with 1 or 0. Let H denote the parity check matrix defined by the expander graph.

1. Label the right vertices as SAT if $H \cdot \mathbf{r} = 0$. Otherwise, label them UNSAT.
2. If $\exists v$ on the left side with more neighbors marked UNSAT than SAT, flip the value of v. Repeat Step 2.

Note that the number of iterations of the algorithm is at most m, since each vertex on the left side is flipped at most once. More precisely, if there are τn errors in $\mathbf{r}$, then the number of iterations is at most $c\tau n$.

Claim 5 *If $\tau \leq \frac{\delta}{2}$, we stop at the nearest codeword to* $\mathbf{r}$.

Claim 6 *Assume $\gamma > \frac{3c}{4}$ and S is any set of size at most δn on the left. Then $|\Gamma_{unique}(S)| > \frac{c}{2}|S|$.*

Claim 6 is immediate from Lemma 3. We now turn to prove Claim 5.

Proof of Claim 5: At any stage, let S be the set of left vertices "in error". We know that $|S| \leq \delta n$. This implies that $\Gamma_{unique}(S) > \frac{c}{2}|S|$ from Claim 6. Therefore, there is a vertex $v \in S$ such that more than half the edges leaving v enter $\Gamma_{unique}(S)$. This just says that, if there are errors, then there exists a candidate vertex that could be flipped.

What if $|S|$ becomes $\geq \delta n$ at some point in the decoding process ? That is, what if the algorithm flips many uncorrupted vertices ? We prove that this cannot happen. Assume, for the sake of contradcition, that at some point $|S| = \delta n$. At this point, $\Gamma_{unique}(S) > \frac{c\delta n}{2}$. That is, the number of "unsatisfied constraints" is more than $\frac{c\delta n}{2}$. At the beginning, this value was at most $\frac{c\delta n}{2}$, and each step of the algorithm decreases this. ■

As proved above, this procedure could correct $\frac{\delta}{2}$ fraction of errors, which is optimal for a code of minimum distance δ.

We remark that the decoding algorithm given above is linear-time. On the other hand, it is not clear whether there is an encoding algorithm for this code that runs in linear time. However, Spielman, building on these ideas, gave a code for which both encoding and decoding are linear-time. This will be the topic for the next lecture.