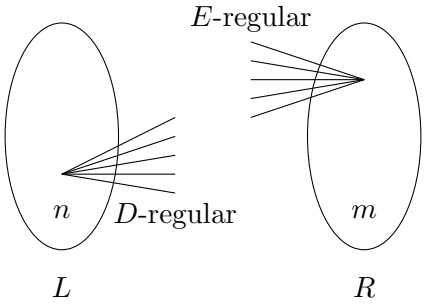
Lecture 6: Expander Codes

Error-Correcting Codes (Spring 2016)
Rutgers University
Swastik Kopparty
Scribes: Abhishek Bhrushundi and Alex Conway

In this lecture we will look at another way of constructing codes of constant rate and distance. These codes are non-algebraic, and are fairly recent. We begin by introducing the concept of expander graphs.

1 Expander graphs



Consider a bipartite graph $G = (L, R, \mathcal{E})$ that is D-regular on the left, and E-regular on the right, where L denotes the set of left vertices, R the set of right vertices, and $\mathcal{E}$ the edge set. Let $|L| = m$, and $|R| = n$. Then G is an (α, h)-expander graph if for every $S \subseteq L$ such that $|S| \leq \alpha n$, we have that $|N(S)| \geq h|S|$, where $N(S)$ is the set of neighbours of S in R.

A simple application of the probabilistic method shows the existence of "good" expanders.

Theorem 1. *For every $D \geq 3, E \geq 1$, $\epsilon \geq \frac{\log(\frac{E}{D})}{D}$, and large enough n, there are bipartite graphs $G = (L, R, \mathcal{E})$ with $|L| = n$, $|R| = m = \frac{Dn}{E}$, that are D-regular on the left, E-regular on the right, and are $(\frac{\epsilon}{E}, D(1-\epsilon))$-expanders.*

In fact, Capalbo-Reingold-Vadhan-Widgerson showed that expander graphs with slightly weaker parameters can be constructed efficiently[1] using Zig-Zag products of graphs. Since we will not be much concerned with the slight weakening of parameters, we may assume that the expanders guaranteed by Theorem 1 are efficiently constructible[2].
This construction is purely combinatorial and was the first of its kind. Earlier constructions were algebraic and had a limitation - h could not exceed $\frac{D}{2}$.

[1] What exactly "efficient" means will not be discussed here, but the reader is assured that the constructions are efficient enough for our purpose, i.e. polynomial time encoding and decoding of codes we are about to construct (in fact, linear time decoding).

[2] For the careful reader, we state the difference between the parameters guaranteed by Theorem 1 and those given by the result of Capalbo et al.: the latter gives $(\frac{\epsilon^2}{E}, D(1-\epsilon))$, whereas the former gives $(\frac{\epsilon}{E}, D(1-\epsilon))$.

2 Expander codes

Suppose we want to construct a good code of length n. Let $G = (L, R, \mathcal{E})$ be an $(\alpha, D(1-\epsilon))$-expander graph with $|L| = n$ guaranteed by Theorem 1. Let us assume that $L = [n]$.

Definition 2 (Expander codes)**.** *The expander code based on the graph G is defined as*

$$C = \{x \in \{0,1\}^n | \forall v \in R, \bigoplus_{i \in N(v)} x_i \equiv 0 \ mod\ 2\}$$

In other words, we can think of the R to be a set of parity checks, and C to be set of all strings that satisfy every parity check in R. It's easy to see that C is linear and that $|C| \geq 2^{n-m}$.

For any $S \subset L$, we define $U(S) \subseteq R$ as follows:

$$U(S) = \{v \in R | N(v) \cap S = 1\}$$

Lemma 3. *Let G be a $(\alpha, D(1-\epsilon))$-expander. Then for every for every $S \subseteq L$ satisfying $|S| \leq \alpha n$ we have that*

$$|U(S)| \geq D(1-2\epsilon)|S|$$

Before we go on to the proof, notice that the lemma gives us nothing nontrivial if G is such that $\epsilon > \frac{1}{2}$, so we really need expanders with $\epsilon < \frac{1}{2}$. Fortunately, the combinatorial construction of Capalbo et al. do give us expanders with this range of parameters. Recall that earlier algebraic constructions had a limitation in this sense.

Proof of Lemma 3. Let $S \subseteq L$ and $|S| \leq \alpha n$. Note that the total number of edges coming out of S is $D|S|$. Also note that the vertices in $U(S)$ receive exactly one edge each from S, implying that the rest of the edges go to the vertices in $N(S) \setminus U(S)$. Thus,

$$\#\text{ of edges from } S \text{ to vertices in } N(S) \setminus U(S) \ = D|S| - |U(S)| \tag{1}$$

Since every vertex in $N(S) \setminus U(S)$, by definition of $U(S)$, receives at least two edges from S, we also have that

$$\#\text{ of edges from } S \text{ to vertices in } N(S) \setminus U(S) \ \geq 2|N(S) \setminus U(S)| \tag{2}$$

Combining Equation (1) and Equation (2), we get

$$D|S| - |U(S)| \geq 2(|N(S)| - |U(S)|)$$

which after a bit of manipulation gives us

$$|U(S)| \geq 2|N(S)| - D|S| \tag{3}$$

Since G is an $(\alpha, D(1-\epsilon))$-expander, we have that $|N(S)| \geq D(1-\epsilon)|S|$. Combining this with Equation (3), gives us

$$|U(S)| \geq D(1-2\epsilon)|S|$$

□

We will now sketch an argument that proves that $dist(C) > \alpha n$. For the sake of contradiction, assume that there is a nonzero codeword x with Hamming weight $\leq \alpha n$. Let S be the support of x. Since $|S| \leq \alpha n$, using Lemma 3 we have that $|U(S)| \geq D(1-2\epsilon)|S|$, which implies that $U(S) \neq \emptyset$, and that there is a $v \in U(S)$. Note that v has exactly one neighbour, say j, in S. Also note that all the other neighbours of v are indices that are outside S. Thus, $\oplus_{i \in N(v)} x_i = x_j$, and since $j \in S$, we have that $x_j = 1$. But this means that the parity check corresponding to v is violated by x which is a contradiction.

We will now prove a better bound on the distance.

Lemma 4. $dist(C) \geq 2\alpha(1-\epsilon)n$

Proof. Let $x \in C$ such that the Hamming weight of x is $< 2\alpha(1-\epsilon)n$. Let T be the support of x, and let $S \subseteq T$ be an arbitrary subset satisfying $|S| = \alpha n$. Using Lemma 3, we have that

$$|U(S)| \geq D(1-2\epsilon)|S|$$

We know that the vertices in $U(S)$ receive exactly one edge from the vertices in S, but what about edges coming from vertices in $T \setminus S$. Since the total number of edges emanating from $T \setminus S$ is $d|T \setminus S|$, if we could show that $D|T \setminus S| < |U(S)|$, it would imply that there is at least one vertex in $U(S)$ that recieves exactly one edge, not just from S, but from the whole of T. In other words, it would show that $|U(T)| \neq 0$. It turns out this is indeed the case:

$$D(|T| - |S|) < D\left(2|S|(1-\epsilon) - |S|\right) = D(1-2\epsilon)|S| \leq |U(S)|$$

This shows that $U(T) \neq \emptyset$, and as we have seen before, this would imply that $x \notin C$, a contradiction! □

It might be worth noting that in the above argument(s), we really don't have to work with the set of vertices in R that have *exactly one* neighbour in S: the argument would work even if we were to analyze the set of vertices in R that have an odd number of neighbours in S. Unfortunately, we don't know (yet) how to leverage this to prove better bounds on the distance.

Lemma 4 proves that the relative distance of expander codes is

$$\delta(C) \geq 2\alpha(1-\epsilon) = 2\frac{\epsilon}{E}(1-\epsilon)$$

What can be said about the rate of these codes? It is easy to see that the rate is

$$R(C) \geq 1 - \frac{m}{n} = 1 - \frac{D}{E}$$

Recall that we need $\epsilon < \frac{1}{2}$ for the code to have good distance (via Lemma 3), and Theorem 1 requires that $\epsilon > \frac{\log\left(\frac{E}{D}\right)}{D}$. This can be achieved by choosing D and E to be constants such that

$$E < 2^{\frac{D}{2}+1}$$

The reader can check that expander codes with rate R for any $R < 1$ may be constructed by carefully choosing D and E. Recall that concatenated Reed-Solomon codes cannot achieve rate $R > \frac{1}{2}$.

3 Decoding

Strategy: Pick a left vertex that is involved in more than $D/2$ violated constraints. Flip it and repeat. See Figure 1.

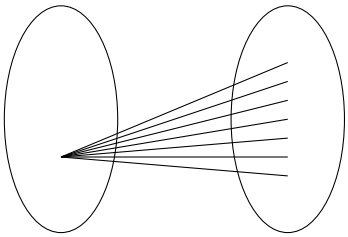


Figure 1: If one left vertex is flipped, then all the constraints will be violated. Intuitively then, if a vertex is involved in many violated constraints, then it *suggests* it should be flipped.

Claim: If we are given x such that

$$\Delta(x, C) < \alpha(1 - \epsilon)n, \tag{4}$$

and $\epsilon < 1/4$, then this algorithm decodes correctly in $O(n)$ time.

Note that this depends on the the condition that $\Delta(x, C) < \alpha(1-\epsilon)n$. Otherwise the linear system could be solved to get just 1 violation.

Let $c \in C$ be the codeword near x. Let S be the coordinates where x and c differ. Since $|S| < \alpha n$ and $\epsilon < 1/4$,

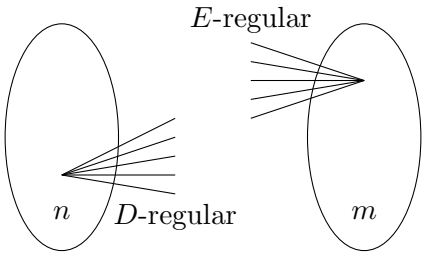
$$|U(S)| \geq D(1 - 2\epsilon) \cdot |S| > \frac{D}{2}|S|.$$

Therefore, some $v \in S$ has more than $D/2$ neighbors in $U(S)$. Now we have made (possibly indirect) progress as the total number of violated constraints has gone down.

It remains to show that if we move away from the code while lowering the total number of violated constraints that Equation (4) will still hold. This is left as the following exercise: let $y \in \{0,1\}^n$ such that $\Delta(y, C) < \alpha n$, then there exists some bit of y involved in more than $D/2$ violated constraints.

4 Expander Codes II: Tanner Construction

We begin with an expander graph with parameters indicated by the following figure:



Pick a linear code $C_0 \subseteq \{0,1\}^E$ of dimension k_0 and distance d_0. Define

$$C = \{x \in \{0,1\}^n \text{ such that every vertex } v \in R \text{ “sees” a codeword of } C_0\}.$$

(By “see” we mean that its neighbors in the induced order form a C_0 codeword.)

Because there are $m(E-k_0)$ linear constraints, we have

$$\mathrm{Dim}(C) \geq n - m(E-k_0).$$

We will now show that

$$\mathrm{Dist}(C) \geq \alpha, \text{ provided } \epsilon < 1 - \frac{1}{d_0}.$$

Let $c \in \{0,1\}^n$ with $\mathrm{wt}(x) < \alpha n$, and let S be the set of non-zero bits of x. We want to show that there exists a constraint that see less than d_0 but at least 1 neighbors in S. We define $U_{[1,d_o)}(S)$ to be the set of such constraints. We have:

$$\begin{aligned} D \cdot |S| &\geq |U_{[1,d_o)}(S)| + d_0(|\Gamma(S) \setminus U_{[1,d_o)}(S)|) \\ &\geq d_0 D(1-\epsilon) \cdot |S| - (d_0 - 1) \cdot |U_{[1,d_o)}(S)| \end{aligned}$$

Because

$$(d_0-1) \cdot |U_{[1,d_o)}(S)| \geq (d_0-1)D \cdot |S| - d_0 D\epsilon \cdot |S|,$$

we have

$$|U_{[1,d_o)}(S)| \geq D\left(1 - \frac{d_0}{d_0 - 1}\epsilon\right) \cdot |S|.$$

Thus, if $|S| < \alpha n$ and $\epsilon < 1 - 1/d_0$, then there exists a violated constraint, and we have proven $\mathrm{Dist}(C) \geq \alpha$.

A key point of this construction is that the expansion is only D/d_0, so in particular less is required.

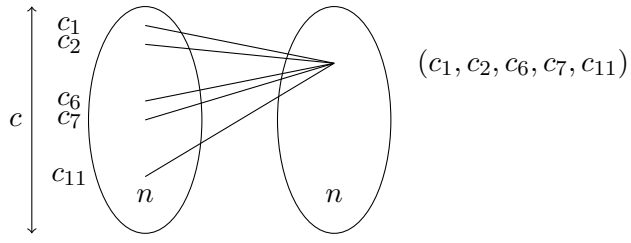
5 Expander-based Distance Amplification

Suppose we are given a code $C \subseteq \{0,1\}^n$. Our goal is to construct a code over a larger alphabet with *improved* distance.

Note there is the trivial map $\{0,1\}^n \to (\{0,1\}^k)^{n/k}$. This preserves the rate, but the distance δ goes to δ/k, which is very bad as we expect distance to improve over a larger alphabet.

Expander Symbol Pushing – Alon, Bruck, Naor, Naor and Roth

Start with an expander graph with n vertices on each side. Take a codeword $c \in C$ and write it "down the vertices" of L:



The symbols of c are pushed across to R, where we obtain length D strings at each vertex. This is a new codeword $c^* \subseteq \epsilon^n$, where $|\epsilon| = 2^D$. In this new code, $|C^*| = |C|$, but $\text{Rate}(C^*) = \frac{1}{D}\text{Rate}(C)$. To compute $\text{Dist}(C^*)$, notice that for all $c \in C$, the number of nonzero coordinates in the corresponding codeword of C^* is at least $D(1-\epsilon) \cdot \min(\text{wt}(c), \alpha n)$. Thus if C has small distance $\delta < \alpha$, then C^* has distance $D(1-\epsilon)\delta$.

It is possible to use an additional error correcting code on the left to improve the rate without hurting the distance. This can be used to match Reed-Solomon codes and the singleton bound, but with linear decoding.