Computational Complexity II Course Instructor: V. Arvind

Expander Codes

Lecturer: Bireswar Das *Scribe: Ramprasad Saptharishi*

1 Introduction

"Expander Codes" for a subclass of "Low Density Parity Check" codes. They are linear codes, which have rates and minimum distance measures very close to optimal. Added to these nice properties, they have very efficient decoding algorithms, in this lecture we shall see one linear time decodable expander code and a linear time parallelly decodable expander code.

2 Linear Codes: Recall

Definition 1. *A linear code, $\mathcal{C}$, is a subspace of $\mathbb{F}_q^n$ of dimension k. This also referred to a $[n, k]$ linear code. The code can also be thought of a mapping from $\mathbb{F}_q^k$ to $\mathbb{F}_q^n$, expanding k column vectors to n column vectors. k is also referred to as the block size of the code.*

Definition 2. *An $k \times n$ matrix G is called the generating matrix of a $[n, k]$ linear code $\mathcal{C}$ if the rows of G for a basis for $\mathcal{C}$.*

And since this generator matrix defines a linear transformation from $\mathbb{F}_q^k \to \mathbb{F}_q^n$, the linear transformation can be thought of as the encoding function for k-column vectors into n-space.

Also, we can think of $\mathcal{C}$ as a kernel of a homomorphism,

$$\begin{aligned} H &: \mathbb{F}_q^n \to \mathbb{F}_q^{n-k} \\ \mathcal{C} &= \{x : Hx^T = 0\} \end{aligned}$$

The matrix H forms a basis of the orthogonal complement of $\mathcal{C}$ and is called the parity check matrix of $\mathcal{C}$.

Definition 3. *The minimum distance (ρ) for any code $\mathcal{C}$ is defined as the least hamminng distance between any two codewords,*

$$\rho = \min_{x,y} \Delta(\mathcal{C}(x), \mathcal{C}(y))$$

And for any linear code, $\rho = \min_x wt(\mathcal{C}(x))$

3 Bipartite Expanders and Expander Codes

Definition 4. *A bipartite graph $G = (V_L \cup V_R, E)$ is called a (c, d)-regular graph if every vertex in V_L has degree c and every vertex in V_R has degree d.*

We shall be referring to the vertices on the left as "variables" and those on the right as "constraints". The edges from x on the left to y on the right will be interpretted as if the y depends on variable x.

Definition 5. *A (c, d, ϵ, δ)-expander is a (c, d)-regular graph such that for every $S \subseteq V_L$ such that $|S| \leq \epsilon|V_L| \implies |\Gamma(S)| \geq \delta|S|$*

Let B be a (c, d)-regular graph, with $V_L = \{v_1, \cdots, v_n\}$ variables and $V_R = \{p_1, \cdots, p_m\}$ with $m = \frac{cn}{d}$. And similar to the rotation maps, we define a map $b : \left[\frac{cn}{d}\right] \times [d] \rightarrow V_L$ such that $b(i, j)$ returns the j-th neighbour of vertex $i \in V_R$

Definition 6. *Let S be an error correcting code of block length d. The expander code $\mathcal{C}(B, S)$ consists of all code words $[x_1, \cdots, x_n]$ such that for all $i \in \left[\frac{nc}{d}\right]$*

$$\left[x_{b(i,1)}, x_{b(i,2)}, \cdots, x_{b(i,d)}\right] \in S$$

Lemma 7. *Suppose B is a $\left(c, d, \alpha, \frac{c}{d\epsilon}\right)$-expander and S is a code with rate r greater than $\frac{c-1}{c}$ and minimum distance ϵ, then the expander code $\mathcal{C}(B, S)$ has*

- *rate$(\mathcal{C}) \geq cr - (c - 1)$*
- *minimum relative distance $\rho(\mathcal{C}) \geq \alpha$.*

Proof. Let H be the parity check matrix of S; H will be a $(d-k) \times d$ matrix where $r = \frac{k}{d}$. Each constraint imposes $(1 - r)d = d - k$ linear restrictions. Hence the total number or linear restrictions from all the constraints is atmost

$$\frac{nc}{d}(1 - r)d = cn(1 - r)$$

Hence the total degrees of freedom is atleast $n - cr(1 - r)$. And hence

$$\begin{aligned} dim(\mathcal{C}) &\geq n - nc(1 - r) \\ \text{rate}(\mathcal{C}) &\geq 1 - c(1 - r) \\ &= cr - (c - 1) \end{aligned}$$

Suppose the minimum relative distance is not α, then there exists a codeword w such that $wt(w) < \alpha n$. Let V_w be the set of indices i such that $w_i = 1$; $|V_w| < \alpha n$. The total number of edges coming out of V_w is $c|V_w|$. And by our size bound on $|V_w|$ and the definition of expansion, $|\Gamma(V_w)| > |V_w|\frac{c}{d\epsilon}$. Hence the average number of edges per constraints is $< d\epsilon$, and therefore should exists a constraint which has less than $d\epsilon$ neighbours in V_w. But by the definition of minimum relative distance, every code word in S *must* have atleast $d\epsilon$ ones, which gives us the required contrdiction.

Hence, minimum relative distance of $\mathcal{C}$ is atleast α. □

4 "Easily" Decodable Expander Codes

In this section we shall look as two choices of our base code S to construct an expander code with very efficient decoding algorithms.

4.1 The Even-Parity Expander Code

Definition 8. *The even parity code of block length d is the set of all code words x such that*

$$\sum_{i=1}^{d} x_i \equiv 0 \pmod 2$$

The dimension of the even-parity code of block length d is $d-1$ and hence the rate is $(d-1)/d$, and the minimum relative distance is $\epsilon = 2/d$.

Let B be a (c,d)-regular expander whose expansion parameters shall be fixed soon, and let S be the even parity code. If B were a (c,d,α,x) expander, we would need x to be atleast $c/2$ for the expander code to have minimum relative distance of α. But in order to be able to decode efficiently, we need more than $c/2$ expansion, we would need $3c/4$.

Decoding Algorithm:

Input: A corrupted assignment x on the variables.
Algorithm:

- If there exists a variable v that is incident to more unsatisfied constraints than satisfied, flip its value.
- Repeat until no more flips are possible.

Claim 9. *The algorithm runs in linear time*

Proof. Clearly at every flip the number of satisfied constraints come down by 1 and hence this has to stop after linear number of flips. We only need to argue that every flip takes only constant number of steps.

Let $S_0, S_1, \cdots, S_c$ be sets such that

$$S_i = \{v | v \text{ has } i \text{ unsatisfied constraints}\}$$

Pick a variable v from the right-most non-empty set (most number of unsatisfied constraints), say S_i; a flip is possible only if $i > \frac{c}{2}$. If v was flipped, the only variable that could get affected would be those that are at a distance 2 from v, and there are atmost cd of them. Hence, we would have to move atmost cd many variables from one S_j to another, and this takes only constant time.

Hence the algorithm runs in time $O(size(B))$. □

We now need to show that the above "Decoding" Algorithm *decodes* considerable errors.

Lemma 10. *If B is a $\left(c, d, \alpha, \frac{3c}{4}\right)$, the decoding algorithm corrects upto a $\alpha/2$ fraction of errors.*

Proof. Suppose w is the corrupted codeword and w_0 is the nearest codeword. A variable where w and w_0 don't match shall be referred to as the corrupt variable, let X be the set of corrupt variables of w. Define the "stage" of the algorithm at any stage as a tuple (u, v) where u is the number of unsatisfied constraints and v is the number of corrupt variables; we want v to become 0.

Let us first consider the stages where $v \leq \alpha u$. Let s be the number of satisfied neighbours of X. $|\Gamma(X)| = u + s > (3c/4)v$ Each satisfied constraints should be connected to atleast 2 elements of X (since X is a set of corrupt variables, a non-zero even number of flips should have happened), and each unsatisfied constraint must be connected to atleast 1 element of X. And hence, the number of edges going out $= cv \geq u + 2s$. With this, and the earlier inequality that $u + s > (3c/4)v$, we get $u > cv/2$. And now by our averaging argument, there will exist one variable that will be flipped since half or more of it's neighbours are unsatisfied, hence the algorithm will flip some corrupt variable.

Now we shall show that the algorithm will successfully decode to the nearest codeword if we begin with atmost $\alpha n/2$ corrupt variable. The only way it can flip an uncorrupt variable is when $v > \alpha n$. Hence if this were to happen, there should be some point when $v = \alpha n$. But at this point, the earlier inequalities work and it would tell us that $u > cv/2 > c\alpha n/2$. But initially u is atmost $c\alpha n/2$ and throughout the algorithm this can only decrease, which leads to a contradiction.

Hence the algorithm will decode correctly if there are atmost $\alpha n/2$ corrupt variables. □

4.2 Explicit Constructions of Expander Codes

Let $G = (V, E)$ be a (n, d, λ) spectral expander (λ is the second largest eigenvalues of the un-normalized adjacency matrix). From this a bipartite expander can be constructed by the "mid-point" partition.

For every $e \in E$, add a vertex for that edge and attach that vertex to both the end points of the edge. Hence your new vertex set will be $E \cup V$ and you have B, a $(2, d)$-regular graph.

Let S be an error correcting code of block length d with ϵ as the minimum relative distance and r as the rate. Hence our expander code $\mathcal{C}(B, S)$ will have rate $\geq 2r - 1$.

Theorem 11 (Alon, Chung)**.** *Let G be a d regular spectral expander with λ as its second largest eigenvalue of the un-normalized adjacency matrix. Let $X \subseteq V$, $|X| = \gamma|V|$. Then the number of edges in the subgraph induced by X is atmost*

$$\frac{d|V|}{2}\left(\gamma^2 + \frac{\lambda}{d}\left(\gamma - \gamma^2\right)\right)$$

We would be using this theorem to prove the following claim about the expander code.

Claim 12. *The minimum relative distance of $\mathcal{C}B, S$ is atleast*

$$\left(\frac{\epsilon - \frac{\lambda}{d}}{1 - \frac{\lambda}{d}}\right)^2$$

Proof. Suppose there exists a w such that

$$wt(w) \leq \frac{dn}{2}\left(\gamma^2 + \frac{\lambda}{d}(\gamma - \gamma^2)\right)$$

then by the theorem w must be adjacent to greater than γn constraints. Since each constraint has 2 neighbours, the average number of variables per constraints is

$$\frac{2\frac{dn}{2}\left(\gamma^2 + \frac{\lambda}{d}(\gamma - \gamma^2)\right)}{\gamma n}$$

Thus if

$$d\left(\gamma^2 + \frac{\lambda}{d}(\gamma - \gamma^2)\right) < \epsilon d$$

then a word of relative weight $\left(\gamma^2 + \frac{\lambda}{d}(\gamma - \gamma^2)\right)$ cannot be a codeword of $\mathcal{C}(G, S)$ and this inequality is satisfied for

$$\gamma < \left(\frac{\epsilon - \frac{\lambda}{d}}{1 - \frac{\lambda}{d}}\right)$$

Hence, in particular (relaxing the inequalities), there can't be a non-zero codeword of weight $< \frac{dn}{2}\gamma^2$, and hence the minimum relative distance of $\mathcal{C}(G, S)$ is atleast $\left(\frac{\epsilon - (\lambda/d)}{1 - (\lambda/d)}\right)^2$. □

Parallel Decoding Algorithm

1. If for any constraint, the variables in that constraint differ in atmost $\frac{d\epsilon}{4}$ places from the nearest codeword, send a "FLIP" message to that vertex.

2. If a variable v receives atleast one "FLIP", flip it's value.

3. Repeat this round.

Claim 13. *If α fraction variables be corrupt relative to the nearest codeword, then after one round of the parallel algorithm, then fraction of corrupt variables (relative to the same codeword) is atmost*

$$\alpha\left(\frac{2}{3} + \frac{16\alpha}{\epsilon^3} + \frac{4\lambda}{\epsilon d}\right)$$

Proof. Let G be the d-regular graph from which B was derived, so $\mathcal{C}(B, S)$ has $\frac{dn}{2}$ variables and n constraints. Let X be the set of $\frac{\alpha n d}{2}$ corrupt variables. The variables that remain corrupt at the end of one round are those that don't receive a "FLIP" message. We shall call a constraint "confused" if it sends a "FLIP" message to something that's not in X, and we shall call a constraint “unhelpful” if it is adjacent to a vertex in X but does not send a “FLIP” message to it.

For a constraint to be confused, atleast $\frac{3d\epsilon}{4}$ variables from X must be its neighbours. And since each variable of X is a neighbour of 2 constraints, the number of confused constraints is atmost

$$\frac{2\frac{\alpha dn}{2}}{\frac{3\epsilon d}{4}} = \frac{4\alpha n}{3\epsilon}$$

Each of these can send atmost $\frac{d\epsilon}{4}$ “FLIP” signals and hence the number of variables outside X that would receive “FLIP” signals is atmost

$$\frac{4\alpha n}{3\epsilon}\frac{d\epsilon}{4} = \frac{dn}{2}\cdot\frac{2\alpha}{3}$$

For a constraint to be unhelpful, it must have more than $\frac{d\epsilon}{4}$ neighbours in X. And hence the number of unhelpful constraints is atmost

$$\frac{2\frac{\alpha dn}{2}}{\frac{\epsilon d}{4}} = \frac{4\alpha n}{\epsilon}$$

These constraints are vertices in our original graph and by Alon’s theorem, there can be atmost

$$\frac{dn}{2}\left(\left(\frac{4\alpha}{\epsilon}\right)^2 + \frac{\lambda}{d}\left(\frac{4\alpha}{\epsilon}\right)\right)$$

variables such that both its neighbours are unhelpful (and hence doesn’t receive a “FLIP” message).

And hence, the fraction of corrupt variables at the end one round is atmost

$$\alpha\left(\frac{2}{3} + \frac{16\alpha}{\epsilon^3} + \frac{4\lambda}{\epsilon d}\right)$$

□

And now with suitable choice of parameters now, we can show that this decoding is also linear time. We shall choose $\alpha < \frac{\epsilon^2}{48}$ and $\lambda = 2\sqrt{d-1}$, showing that the algorithm is linear time for these parameters is left to the interested reader.