CS 59000 CTT Current Topics in Theoretical CS Oct 16, 2012

Lecture 17

Lecturer: Elena Grigorescu *Scribe: Venkata Gandikota*

In this lecture we'll show how to construct a family of good error correcting codes from expander graphs. We'll also use properties of expanders to give a linear time decoding algorithm.

Introduction

Recall the definition of a code.

Definition 1 *A $(n, k, d)_q$ code C is a collection of strings over alphabets Σ, where $|\Sigma| = q$, block length $= n$, message length $= k$ (dimension), minimum distance $= d = \Delta(C)$*

The notation $[n, k, d]_q$ denotes that the code is linear, namely can be defined in terms of a parity check matrix as follows: $C = \{c \mid cH = 0 \; where, \; H \in F_q^{n\times(n-k)}\}$. Here we'll talk about binary codes, so $q = 2$. Recall that a good code is one where the rate $R(C_n) = k/n > R_0 > 0$ and $\delta(C_n) > \delta_0 > 0$ for some absolute constants R_0 and δ_0.
A linear code given by its parity check matrix H can also be represented by a bipartite graph whose left vertices represent the n coordinates of a word and whose right vertices represent the constraints given by the parity matrix, one for each column. There is an edge between c_i and h_l (a column vector of H) if $H_{il} \neq 0$. So c is a codeword if for any constraint h_l it is the case that $\sum_{i:H_{il}\neq 0} c_i = 0$. In other words, a constraint is satisfied for a word w if the constraint is adjacent to an even number of positions that take value 1 in w. Note that H is the adjacency matrix of the bipartite graph constructed this way.
Recall the following definition.

Definition 2 *Let $G = (L \cup R, E)$ be a bipartite graph. G is a (d, c)-regular if $\forall\, v \in L$, $deg(v) = d$, and $\forall\, v \in R$, $deg(v) = c$. G is d-left regular if $\forall\, v \in L$, $deg(v) = d$.*

Definition 3 *$G = (L \cup R, E)$ be a (d)-left regular graph with $|L| = n$. G is a (k, γ) - expander if $\forall S \subset L, |S| <= k$ is such that $|N(S)| > \gamma|S|$.*

Recall that we want $k = \delta n$, $\gamma > d/2$, $|R| = \beta n$.

Construction of good ECCs from Expanders

We will next show that expanders with $|S| > \delta n$ give good error correcting codes.
Note that the minimum distance $\Delta(C)$ is the minimum number of 1's in an assignment to the coordinates that satisfies all the constraints.
So if we want a code with $\Delta(C) = \delta n$ (for some constant δ) we must have that for any set $S \subset [n]$ with $|S| < \delta n$ no word whose coordinates take value 1 exactly at the positions in S can satisfy all the constraints. So for each such set there is some constraint that is adjacent to an odd number of vertices in S.

Definition 4 *Let $N_1(S)$ be the set of neighbours of S that are adjacent to exactly one neighbour in S This is called the set of unique neighbours of S.*

Theorem 5 *If $G = (L \cup R, E)$ is a d-left regular graph that is a $(\delta n, \gamma)$- expander with $\gamma > d/2$, then $\forall S \subset L$, s.t. $|S| < \delta n$, we have that $|N_1(S)| \geq 1$.*

Proof Fix S. Assume that there are no unique neighbours, i.e. $|N_1(S)| = 0$. Hence, all vertices in $N(S)$ have at least 2 neighbours in S. Note that the total number of edges $= dn$. We now try to bound the number of edges between S and $N(S)$: $|E(S, N(S))| \leq d|S|$. Also, $|E(S, N(S))| \geq 2|N(S)|$. Since G is a $(\delta n, \gamma)$ expander, $|N(S)| > \gamma|S| \Rightarrow |E(S, N(S))| \geq 2\gamma|S|$. So $2\gamma|S| \leq |E(S, N(S))| \leq d|S|$, which implies $\gamma \leq d/2$, a contradiction. Hence, our assumption was false. ■

Therefore, whenever $|S| < \delta n$, there will be a vertex v in R that has odd number of neighbours. Hence, assignments supported on a set S with $|S| < \delta n$ will not be satisfied and so the minimum weight of the code is at least δn, which is what we wanted. Also, to ensure good rate we need that $k = \beta n$ for some constant β.
Therefore, we have shown that good expanders give good error correcting codes. By probabilistic arguments similar to the ones we made in a previous lecture one can show that expanders with all these parameters do exist and so good codes exist.
We finally remark that a stronger version of Theorem 5 can be shown for the expanders graphs discussed. We will need this stronger version to show the linear time decoding algorithm for expander based codes.

Theorem 6 *If $G = (L \cup R, E)$ is a d-left regular graph that is a $(\delta n, \gamma)$- expander with $\gamma > d/2$, then $\forall S \subset L$, s.t. $|S| < \delta n$, we have that $|N_1(S)| \geq (2\gamma - d)|S|$.*

Proof Let $n_1 = |N_1(S)|$ and n_2 be the number of neighbors of S with more than 2 edges to vertices in S. So $n_1 + n_2 = |N(S)| \geq \gamma|S|$. Since $|E(S, N(S)| = d|S|$ we have that $n_1 + 2n_2 \leq d|S|$. Therefore $n_1 \geq (2\gamma - d)|S|$.
■

A Decoding Algorithm for expander based codes

In this section we'll show a decoding algorithm for codes corresponding to (d, c)-regular graphs. As mentioned before, such graphs are w.h.p expanders.
We have seen in a previous lecture how the parity check matrix was used to decode the Hamming code. It turns out that the parity matrix and its properties can be used to decode more general linear codes. We will use again the representation of a code with (d, c)-sparse parity matrix H by its corresponding (d, c)-regular graph.
A first try for a decoding algorithm could be to look for unsatisfied constraints and correct coordinate bits so as to satisfy each unsatisfied constraint. This might not work since correcting one bit might make some other constraints unsatisfied and so the algorithm might not terminate. The following decoding algorithm is due to Sipser and Spielman.
Algorithm (FLIP):

- (Suppose the vector r is received, so the values of its coordinates represent an assignment to the vertices (variables) on the left).

- If there is a variable that is adjacent to more unsatified than satisfied constraints, then flip the value of that variable.
- Repeat until no such variable remains.

Claim 7 *The algorithm terminates in finite number of steps.*

Proof Suppose we start with k unsatisfied constraints. Each flip increases the number of satisfied constraints by at least 1 since the constraints that are not adjacent to the flipped vertex do not change their status. Hence, the algorithm terminates in at most k steps. ■

Claim 8 *The algorithm runs in linear time.*

Proof cn operations are enough to compute all the parity checks (for the vertices on the right). For each left vertex, if flipped we need to update at most d checks. Hence, running time of the algorithm $= cn + dn$. ■

Theorem 9 *Let G be a (d, c)- regular, $(\delta n, \gamma)$-expander with $\gamma > 3d/4$. Suppose there are at most $\epsilon n < \delta n/2d$ errors in the codeword, then FLIP terminates with the transmitted codeword.*

Note that since δn is the distance of code, the radius for unique decoding is $\delta n/2$ but this algorithm can correct only up to some constant fraction of unique decoding distance.
Proof Let,
y: transmitted code word
r: received word
z: decoded word from r using the FLIP algorithm.
We want to show that $z = y$. Note that $d(y, r) \leq \epsilon n \leq \delta n/2d$ since this is the number of error that could have occurred.
Let, $S = \{i | r_i \neq y_i\}$. Since all vertices from the right that are not adjacent to vertices in S must be satisfied, it follows that the number of unsatisfied constraints $\leq |N(S)| \leq d|S| \leq d\epsilon n$. So FLIP terminates in $d\epsilon n$ steps. The number of bits flipped to get from r to z is at most the number of unsatisfied constraints, so $d(r, z) \leq d|S| \leq d\epsilon n$.
From the triangle inequality, we have $d(y, z) \leq d(y, r) + d(r, z) < \delta n/2d + \delta n/2 < \delta n$.
Since distance of the code is δn z must be either y or not a codeword.

Claim 10 *FLIP does not terminate in a non-codeword.*

Proof We'll show that if z is not a codeword, by the expansion of the graph it will follow that there is a non-satisfied constraint after FLIP terminates, a contradiction. Let $y + z = e$. If $e = (0, 0, \cdots 0) = \bar{0}$ then $y = z$ so assume this is not the case. By the above argument $wt(e) < \delta n$.
Consider z and $y + z$. Since y is a codeword, all constaints are satisfied, so the constraints satisfied by z should be exactly the same as the constraints satisfied by $z + y = e$.
Let $S = support(e)$. By Theorem 6 the number of unique neighbours is at least $(\gamma - d/2)|S|$. Therefore, using our assumption that $\gamma > 3d/4$ we get, $|N_1(S)| \geq (\gamma - d/2)|S| > d|S|/2$.

Therefore by averaging over all vertices in S, we can say that $\exists v \in S$ such that $N(v)$ contains at least $d/2$ unique neighbours. Since each vertex has exactly d neighbors, it follows that for v the number of unsatisfied constraints is larger than the number of satisfied constraints, a contradiction since we assumed that FLIP terminates. ■ ■