18.409 Applied Extremal Combinatorics November 3, 1998

Lecture 14

Lecturer: Daniel A. Spielman *Scribe: Sae-Young Chung*

1 Introduction

In this lecture, we finish the construction of the Golay code and show some of its properties. Then, we show how to construct error-correcting codes from expander graphs by using random bipartite graphs and prove there exist good encoding and decoding algorithms for the codes.

2 The Golay code

We first define the quadratic residue character and construct the Golay code from this.

Definition 2.1 *When a prime $p \equiv 3 \bmod 4$ is given, the quadratic residue character $\chi(X)$ is defined as $\chi(X) = 1$ if $\exists a$ such that $a^2 \equiv X \bmod p$, and $\chi(X) = -1$ otherwise.*

Using this definition, we define a $p \times p$ matrix M'_p as the following.

$$M'_p(i,j) = \begin{cases} 1 & \text{if } i = j \\ 1 & \text{if } \chi(i-j) = 1 \\ 0 & \text{if } \chi(i-j) = -1 \end{cases}$$

It can be shown that any two rows of M'_p agree in exactly $\frac{p-1}{2}$ columns and each row of M'_p has $\frac{p+1}{2}$ 1's. As a special case, we think about $p = 11$. We can construct a 24×12 matrix G by using M'_{11} and a 12×12 identity matrix.

$$M'_{11} = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \end{pmatrix}$$

$$G = \begin{pmatrix} & 0 & 11111111111 \\ & 1 & \\ & 1 & \\ & 1 & \\ & 1 & \\ I & 1 & M'_{11} \\ (12 \times 12) & 1 & (11 \times 11) \\ & 1 & \\ & 1 & \\ & 1 & \\ & 1 & \\ & 1 & \end{pmatrix}$$

We prove that G is the generator matrix of the extended Golay code. The extended Golay code has 2^{12} code words, $d_{\min} = 8$, and length 24. It is obvious from the matrix G that the weight of the first row is 12 and the rest of the rows have weight 8.

Proposition 2.1 *All pairs of rows have inner product 0.*

proof) If the top row is not involved, then 2 rows in M'_p agree in 5 columns and have common 1's in 3 columns. (see proposition 2.3) Since there is an extra column in G where the 2 rows have a common 1, the inner product is 0 (modulo 2). The top row and another row have common 1's in 6 places, because M'_p agrees with all 1 vector of length 11 in 6 places. Therefore, the inner product for this case is also 0. $\square$

We show that all codewords of the extended Golay code have weight divisible by 4 by proving the following lemma and we also show that the extended Golay code has the minimum distance 8 by proving that no codeword of G has weight 4.

Lemma 2.2 *Let v and w be vectors with inner product 0 (modulo 2) and each has weight divisible by 4, then $4|wt(v+w)$.*

proof) Let t be the number of places that both are 1, then

$$wt(v+w) = wt(v) + wt(w) - 2t.$$

Since $4|wt(v), 4|wt(w), 2|t$, the lemma follows. □

Proposition 2.3 *No codeword of G has weight 4.*

proof) Since there is a 12×12 identity matrix in G, it is enough to check if the sum of less than or equal to 4 rows of G has weight ≤ 4. Since every row has weight ≥ 8, we only have to prove this for sums of 2, 3 or 4 rows.

- 3 rows in last 11 rows: Since we get 4 1's from I_{12} and the $[011111111111]^T$ column, we need to have sum of 3 rows of $M'_p = 0$. We can use the same argument for sum of 4 rows because we also get 4 1's from 13 columns excluding M'_p. Therefore, if we can prove that no sum of 3 or 4 rows of M'_p is 0, then this proves no sum of 3 or 4 rows in last 11 rows has weight 4.

  How many solutions are there that satisfy $v \cdot M'_p = (0, \ldots, 0)$? It is clear that $v = (1, \ldots, 1)$ is a solution because every row has weight 6. In fact, there are no other solutions, because

  $$M'_p \cdot {M'}^T_p = J + I,$$

  where J is the 11×11 all 1 matrix. Since J has rank 0 and I has rank 11, J+I should have rank at least 10 and this implies the rank of M'_p is 10.

- 2 rows in last 11 rows: The only way to have 2 length 11 vectors with weight 6 and agreeing in 5 places is the following.

  11111100000
  00011111100

  Therefore, all sums of 2 rows in last 11 rows have weight 8.

The rest of analysis include the top row can be done similarly. □

The (23,12,7) Golay code can be constructed from the (24,12,8) extended Golay code by dropping any column. It has length 23, 2^{12} codewords, and $d_{\min} = 7$ and meets the sphere packing bound. Since

$$1 + \binom{23}{1} + \binom{23}{2} + \binom{23}{3} = 2^{11},$$

every ball of radius 3 contains 2^{11} words and any two balls centered at two different codeword vectors are disjoint because the minimum distance is 7 and $2^{11} \cdot 2^{12} = 2^{23}$. There are not many codes that meet the sphere packing bound. The Golay code, the Hamming code, all 0 code, and repetition code are the only codes that are optimal in the sphere packing bound sense, which can be proved.

3 Construction of Error-Correcting Codes from Expander Graphs

We now show how to construct error-correcting codes from expander graphs by using random bipartite graphs and prove there exist fast encoding and decoding algorithms for the codes. To do this, we consider certain types of expanders, namely uneven bipartite graphs which have different number of nodes on each side. This code is called the "Low Density Parity Check" (LDPC) code and is from Gallager's thesis.

The left hand side of the graph has n nodes each having degree d, and the other side of the graph has $\frac{n}{2}$ nodes each having degree $2d$. The left hand side nodes correspond to bit positions and the right hand side nodes correspond to parity checks. A parity check $\oplus$ is satisfied if the parity of its neighbors is equal to 0. Since there are n variables and $\frac{n}{2}$ constraints, there are at least $2^{\frac{n}{2}}$ code words. Now, we show that this is a good expander and is also a good error-correcting code.

Let us assume that for all sets A of at most αn nodes on the left, the number of neighbors of $A = |N(A)| > \frac{d}{2}|A|$, although we do not know how to construct codes to satisfy this. We claim that the relative minimum distance of the code is at least α.

Proposition 3.1 *The relative minimum distance of LDPC code is $\geq \alpha$.*

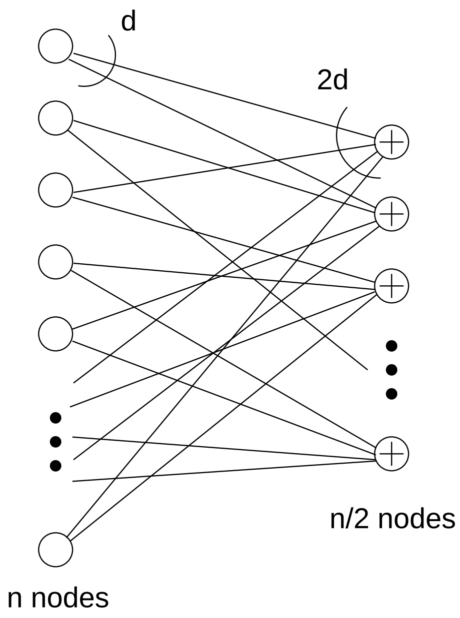

Figure 1: Low Density Parity Check code

proof) We prove this by contradiction. Assume $\exists$ a codeword of weight $\leq \alpha n$ and let A be the set of nodes that are 1. Then, by assumption, $|N(A)| > \frac{d}{2}|A|$. Since this is a codeword, all parity checks are satisfied. If a parity check is in $N(A)$, then it has to have even number of nodes of A and, therefore, has to be a neighbor of at least 2 nodes in A. This implies the number of edges between A and the neighbors of A should be at least $2|N(A)|$. On the other hand, $d|A|$ is the number of edges between A and the neighbors of A and this implies $N(A) \leq \frac{d}{2}|A|$. This contradicts the assumption. $\Box$

We now show that this code can be decoded quickly. We say a check is unsatisfied if the check sum $\oplus$ of neighbors is 1 when a word is received which is assumed to be error corrupted. We can decode (correct) errors by first finding a node on the left such that most neighbors are unsatisfied and then flip the bit of that node and iterate until there is nothing more can be done. It is clear that the number of unsatisfied checks will be decreased at each iteration of this algorithm. We claim that this decoding algorithm is a linear time algorithm since it runs at most $\frac{n}{2}$ times because the number of unsatisfied checks decreases each time. We assume that constant time is required per operation because we only need to update $d \cdot 2d$ nodes each time. From now on, we assume that for $\forall$ sets A on the left such that $|A| \leq \alpha n$, we have $|N(A)| > \frac{3}{4}d|A|$.

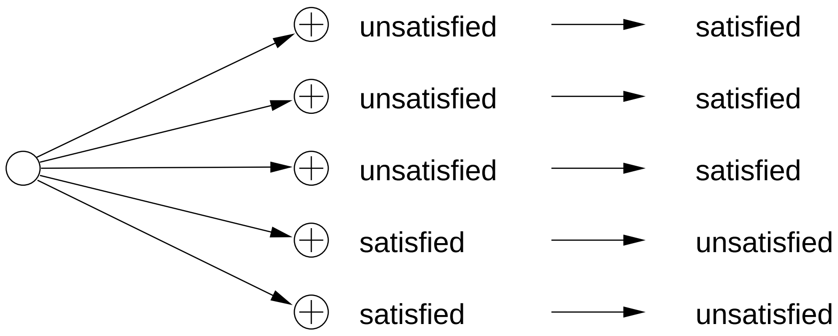

Figure 2: Updating rule

Lemma 3.2 *If there are $1 \leq a \leq \alpha n$ errors, then there are more than $\frac{da}{2}$ unsatisfied checks.*

proof) Let A be the error nodes on the left, let U be the set of unsatisfied checks, and let S be the set of satisfied checks with neighbors in errors. Let us define $u = |U|$ and $s = |S|$. From the assumption on the size of the neighbors of A, we get $u + s > \frac{3}{4}da$. Since every node in S should have at least 2 nodes of A as neighbors to have a satisfied check, we have $u + 2s \leq da$. By combining these two inequalities, we obtain $u > \frac{d}{2}a$. $\square$

The algorithm can begin because we have enough number of unsatisfied checks. In figure 3, we see how the number of unsatisfied checks is decreased. Sometimes, however, the number of errors could be increased even though the number of unsatisfied checks is decreased and this could result in catastrophe if we started from A because the number of errors can exceed αn during the decoding. However, if C is the initial point, i.e. the received word has less than $\frac{\alpha n}{2}$ errors, we can make sure that this catastrophe will not happen and the decoding algorithm will be successful.

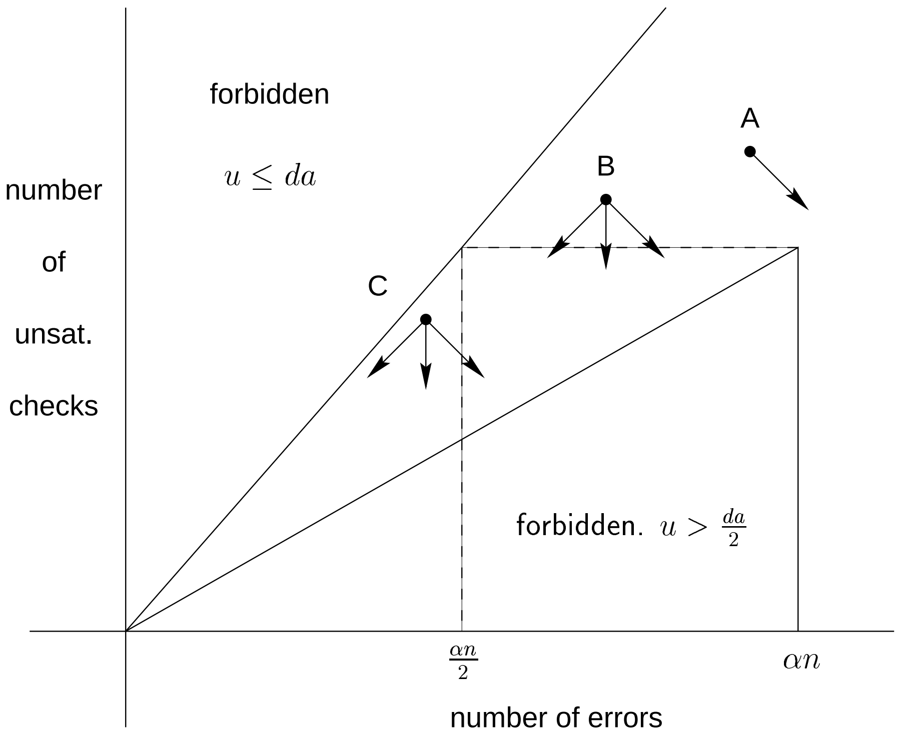

Figure 3: Number of errors and decoding algorithm