6.440 Essential Coding Theory Spring 2013

Lecture 19 - April 22, 2013

Lecturer: Madhu Sudan *Scribe: Mohsen Ghaffari*

1 Announcement

Sign up for the project presentation slots. Related Information are available on the website of the course. The presentation dates are Tuesday 5/7 and Thursday 5/9, 10 am to 1 pm. Each team will have a 20 minutes slot for presentation.

2 Overview: Codes From Other Codes

So far in the course, we have seen three types of codes, summarized as follows:

1. **Algebraic Codes**
   - Work for worst-case errors.
   - Have polynomial time encoding and decoding algorithms
2. **LDPCs**
   - Work for worst-case errors as well, but a less many errors (smaller distance) compared to the Algebraic Codes.
   - Have linear time encoding and decoding algorithms and are simpler.
3. **Polar Codes**
   - Work for random errors. when viewed in the context of worst-case errors, these codes do not have good distance.
   - Have $O(n \log n)$ encoding and decoding algorithms. Note that this bounds sits somewhere in between the time complexities of the Algebraic Codes and LDPCs.

In today's lecture, we will look into the question of "how can we combine codes to get new codes?". In this regard, we will consider five types of operations:

- Puncturing
- Restriction
- Tensor
- Concatenation

- A graph-theoretic method

Puncturing and restriction are simple and basic operations, which we will review quickly, and we have already seen concatenation in the past lectures (refer to Lecture 7). The focus in this lecture will be on the *Tensor method*, presented in Section 3.2 and the *graph-theoretical method*, presented in 3.4.

3 Operations on Codes

3.1 Puncturing & Restriction (simple)

To explain puncturing and restriction, consider an $(n, k, d)_\Sigma$ code C and fix a coordinate $i \in \{1, 2, \ldots, n\}$.

Puncturing In the puncturing operation, for each codeword $c_j \in C$, we eliminate the i^{th} coordinate of c_j. It is easy to see that this operation produces a $(n-1, k, d-1)_\Sigma$ code.

Restriction Consider the i^{th} coordinate of all the codewords in C and let $a \in \Sigma$ be the symbol that appears in the i^{th} coordinate with the highest frequency. Let C' be the set of all the codewords $c_j \in C$ such that the i^{th} coordinate of c_j is equal to a. In the restriction operation, the new code C_{new} is derived from C' by eliminating the i^{th} coordinate of each codeword $c_j \in C'$. It is easy to see that C_{new} is a $(n-1, k-1, d)_\Sigma$ code. In particular, to see that C_{new} has dimension $k-1$, note that by definition of a, the number of codewords in C' (and thus in C_{new}) is at least $|\Sigma|^k/\Sigma = |\Sigma|^{k-1}$.

3.2 Tensor (new)

We define the tensor operation for linear codes. For two linear codes $C_1 = [n_1, k_1, d_1]_\Sigma$, $C_2 = [n_2, k_2, d_2]_\Sigma$, the tensor code of C_1 and C_2 is denoted by $C_1 \otimes C_2$. We view the codewords of $C_1 \otimes C_2$ as a $n_1 \times n_2$ matrix [1]. A given $n_1 \times n_2$ matrix A with entries in Σ is a codeword of code $C_1 \otimes C_2$ iff each column A is a codeword of C_1 and each row of A is a codeword of C_2.

$$A_{n_1,n_2} = \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n_2} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n_2} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n_1,1} & a_{n_1,2} & \cdots & a_{n_1,n_2} \end{bmatrix}$$

Theorem 1 *$C_1 \otimes C_2$ is a $[n_1 n_2,\ k_1 k_2,\ d_1 d_2]_\Sigma$ code.*

[1] Note that this can be equivalently viewed as a vector of length $n_1 n_2$.

Proof First of all, it is easy to see that $C_1 \otimes C_2$ is a linear code. Intuitively, this is because $C_1 \otimes C_2$ can be described as a collection of linear constraints on the entries of the matrix. To formally prove the linearity, one can see that the summation of two matrices (codewords) in $C_1 \otimes C_2$ is in $C_1 \otimes C_2$, and also, that scaling up a matrix in $C_1 \otimes C_2$ by an element of Σ is in $C_1 \otimes C_2$, as well.

We next verify that $C_1 \otimes C_2$ has distance $d_1 d_2$. First consider a codeword c_1 of C_1 that has only d_1 non-zero elements, and a codeword c_2 of C_2 that has only d_2 non-zero elements. Then, consider the matrix A where the entry of A in the intersection of the i^{th} row and the j^{th} column, i.e., $a_{i,j}$, is equal to the product of the i^{th} element of c_1 and the j^{th} element of c_2. Note that $A \in C_1 \otimes C_2$ because each column of A is a multiple of c_1 and is thus a codeword of C_1, and each row of A is a multiple of c_2 and is thus a codeword of C_2. Now the number of nonzero elements of A is exactly $d_1 d_2$. This shows that the distance of $C_1 \otimes C_2$ is at most $d_1 d_2$. To see that this distance is also at least $d_1 d_2$, consider an arbitrary non-zero matrix B in $C_1 \otimes C_2$ and pick a nonzero entry of B; let it be $b_{i,j}$. Since the i^{th} row of B is a non-zero codeword of C_2, this row has at least d_2 nonzero elements. Consider the d_2 columns of B that intersect the i^{th} row of B in these nonzero elements (at least d_2 columns). Each of these columns is z nonzero codeword of C_1 and thus, has at least d_1 nonzero elements. Therefore, matrix B in total has at least $d_1 d_2$ nonzero entries. Hence, we conclude that the distance of code $C_1 \otimes C_2$ is $d_1 d_2$.

Finally, we verify that $C_1 \otimes C_2$ has dimension $k_1 k_2$. Before going there, note that just using the method described at the start of the previous paragraph, we can find $|\Sigma|^{k_1} \cdot |\Sigma|^{k_2} = |\Sigma|^{k_1+k_2}$ matrices in $C_1 \otimes C_2$, which already shows that the dimension of $C_1 \otimes C_2$ is at least $k_1 + k_2$. To prove that the dimension is $k_1 k_2$, we show a method to produce $|\Sigma|^{k_1 k_2}$ matrices in $C_1 \otimes C_2$. Note that we can describe code C_1 by an $n_1 \times k_1$ matrix G_1 such that for each $k_1 \times 1$ vector x with entries from Σ, $G_1 x$ is a codeword of C_1. Similarly, we can describe code C_2 by an $n_2 \times k_2$ matrix G_2 such that for each $k_2 \times 1$ vector y with entries from Σ, $y^T G_2^T$ is a codeword of C_2. Now, for each $k_1 \times k_2$ matrix A with entries from Σ, $G_1 A G_2^T$ is a codeword of $C_1 \otimes C_2$.

$$\begin{bmatrix} \\ G_1 \\ \\ \end{bmatrix}_{n_1 \times k_1} \quad \begin{bmatrix} \\ A \\ \\ \end{bmatrix}_{k_1 \times k_2} \begin{bmatrix} \\ G_2^T \\ \\ \end{bmatrix}_{k_2 \times n_2} \in C_1 \otimes C_2$$

Moreover, each matrix A produces a distinct matrix $G_1 A G_2^T$ of $C_1 \otimes C_2$. Hence, we get $|\Sigma|^{k_1 k_2}$ matrices which shows that $C_1 \otimes C_2$ has dimension $k_1 k_2$. ■

3.3 Concatenation, and Recap

In the past lectures, we defined concatenation. As a short summary, the operations that we have studied so far can be summarized as follows:

- Puncturing: $(n, k, d)_{\mathbb{F}_q} \to (n-1, k, d-1)_{\mathbb{F}_q}$
- Restriction: $(n, k, d)_{\mathbb{F}_q} \to (n-1, k-1, d)_{\mathbb{F}_q}$
- Tensor: $[n_1, k_1, d_1]_{\mathbb{F}_q} \otimes [n_2, k_2, d_2]_{\mathbb{F}_q} \to [n_1 n_2, k_1 k_2, d_1 d_2]_{\mathbb{F}_q}$
- Concatenation: $\{n_1, k_1, d_1\}_{\mathbb{F}_q^{k_2}} \circ [n_2, k_2, d_2]_{\mathbb{F}_q} \to [n_1 n_2, k_1 k_2, d_1 d_2]_{\mathbb{F}_q}$

In the final bullet-point, $\{n, k, d\}_{\mathbb{F}_q^{k'}}$ denotes an *Additive Code*, which can be viewed as a weaker variant of linear code, and is defined as follows:

Definition 2 ***(Additive Code)*** *$C \subseteq \mathbb{F}_q^{k'}$ is a $\{n, k, d\}_{\mathbb{F}_q^{k'}}$ code iff (1) for each codeword $(x_1, \ldots, x_n) \in C$ and each $\alpha \in \mathbb{F}_q$, we have $(\alpha x_1, \ldots, \alpha x_n) \in C$, and (2) for each two codewords $(x_1, \ldots, x_n)$ and $(y_1, \ldots, y_n)$ in C, we have $(x_1 + y_1, \ldots, x_n + y_n) \in C$.*

Note that each of the above operations loses in 'performance' (e.g., when measured by rate plus distance). Also, amongst them, only concatenation has been useful for us thus far. In the next section, we present a graph-theoretic method of deriving new codes from given codes which actually leads to improvements in 'performance'.

3.4 The Graph-Theoretic Method

The method that we explain in this section was first presented by Alon, Bruck, Naor, Naor and Roth in 1992 [2], and later extended by Alon, Edmonds, and Luby in 1995 [3]. These two papers focus on the construction of these two codes and will be the focus point of the rest of this lecture. Guruswami and Indyk [1] studied the time complexity of the decoding of codes constructed by this method.

We start with presenting the idea of the ABNNR work [2], which takes a 'good' code over alphabet $\mathbb{F}_2$ and builds an 'excellent' code over $\mathbb{F}_2^d$. Consider a bipartite d-regular graph H with n nodes on each side that is a (γ, δ)-expander [2]. Let $C \in \mathbb{F}_2^n$ be a code with distance at least $\delta' n$. For each codeword $c \in C$, we get a codeword $c' \in C_{new}$, as follows: label the n nodes on the left side of H with the bits of the codeword c. Then, for each node v on the right side of H, label v with the binary string of length d that is made of the labels of the d neighbors of v (ordered consistent with the ordering of nodes of the left side). The codeword $c' \in C_{new}$ consists of the labels of the nodes of the right side, which are n symbols in $\mathbb{F}_2^d$. See Figure 3.4 for an illustration.

[2] Recall from the previous lectures that H is a (γ, δ)-expander if for each set S of nodes on the left such that $|S| \leq \delta n$, we have $|\Gamma(S)| \geq \delta |S|$, where $\Gamma(S)$ denotes the set of nodes on the right side that have at least one neighbor in S.

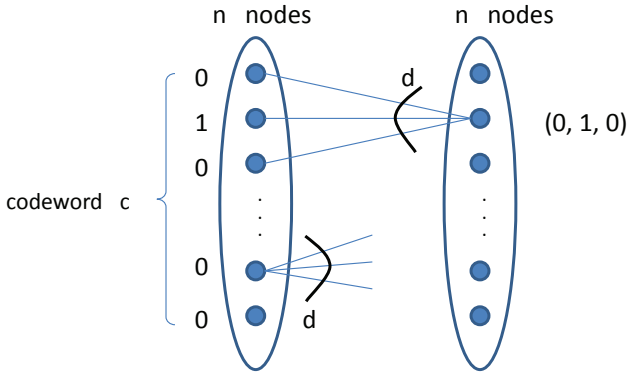

Figure 1: Creating a codeword $c' \in C_{new}$ from codeword $c \in C$

Theorem 3 *If C is a $[n, k, \delta' n]_2$ code and H is a bipartite d-regular (γ, δ)-expander graph with n nodes on each side such that $\delta \geq \delta'$, then the new code C_{new} derived as above is a $\{n, \frac{k}{d}, \gamma\delta' n\}_{2^d}$ code.*

Proof It is easy to check that C_{new} is an additive code. For the dimension, note that C_{new} has 2^k codewords, one for each codeword of C. These codewords correspond to $2^k = (2^d)^{\frac{k}{d}}$ messages in alphabet $\mathbb{F}_2^d$, which can be each expressed via $\frac{k}{d}$ symbols in this alphabet. Thus the dimension of the new code is $\frac{k}{d}$.

For the distance, note that each nonzero codeword c of C produces a codeword of C_{new} with at least $\gamma\delta'$ nonzero symbols[3]. This is because, let S be a set of δ' nonzero elements of c (exists because C has distance $\delta' n$). Then the labels of all nodes in $\Gamma(S)$ are non-zero, and we know that since $|S| = \delta' n \leq \delta n$ and H is a (γ, δ)-expander, $|\Gamma(S)| \geq \gamma\delta'$. ■

If we choose a 'good' expander H, then γ would be close to d which means that in C_{new}, we have increased the distance by (almost) a factor of d, when compared to C. This however comes at the cost of decreasing the rate by a factor of d and increasing the alphabet size exponentially to 2^d.

Alon, Edmonds, Luby [3] presented a new interpretation of this method. In this interpretation, we combine two codes to get one new code. We explain later that putting the ABNNR work in this framework, one of the initial codes is simply a repetition code. First we present a definition:

Definition 4 ((d, ϵ)-***Regular Bipartite Graph***)*: Consider a d-regular bipartite graph H with n nodes on each side. H is called a (d, ϵ)-Regular Bipartite Graph if for each set X of the left nodes of H and each set Y of the right nodes of H, we have*

$$|E_H(X, Y)| \geq (\frac{|X|}{n}\frac{|Y|}{n} - \epsilon) \cdot dn$$

[3]Here, the latter nonzero is with respect to F_2^d and means an element of $\mathbb{F}_2^d$ that is not equal to $\underbrace{(0, 0, \ldots, 0)}_{d \text{ times}}$.

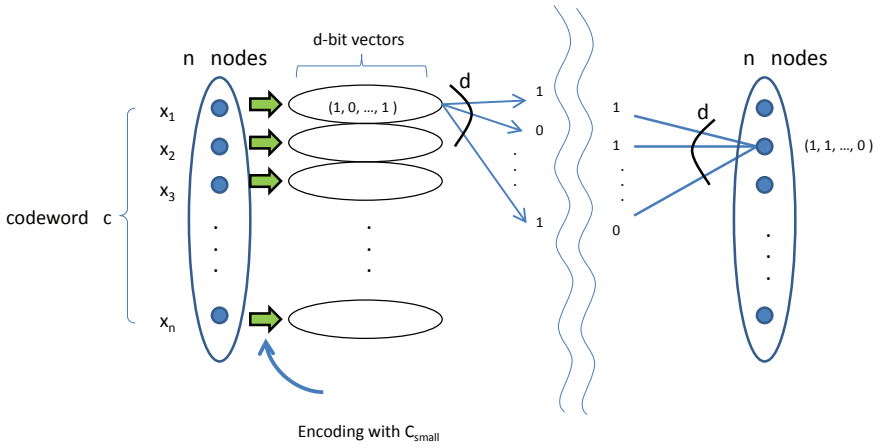

Figure 2: New way of creating a codeword $c' \in C_{new}$ from codeword $c \in C$

, where $E_H(X,Y)$ denotes the edges of H with one endpoint in X and the other in Y.[4]

We define the new code C_{new} using a given a (d,ϵ)-regular bipartite graph H, a given additive code $C = \{n,k,\delta' n\}_{\mathbb{F}_2^{R_1 d}}$ and a given linear code $C_{small} = [d, R_1 d, \delta_1 d]_{\mathbb{F}_2}$. For each codeword $c \in C$, we define a codeword $c' \in C_{new}$, as follows: label the n nodes on the left side of H with the n symbols of the codeword c. Then, encode each of these symbols using C_{small}. Thus, now, each node on the left is labeled with a d-bit vector. Each left-side node sends the d-bits of its labels to its d neighbors on the right side of graph H, one bit to each and with the order consistent with the ordering of the bits of the label and the ordering of the right-side nodes. Each node on the right-side receives exactly d bits, one from each of its neighbors. Putting these bits in a d-bit vector (ordered consistent with the ordering of the left-side nodes) gives the label of each right-side node. The n labels of the right-side nodes constitute the codeword c' in $C_{new} \subseteq \mathbb{F}_2^d$. See Figure 3.4 for an illustration.

Theorem 5 *If C is a $\{n,k,\delta' n\}_{\mathbb{F}_2^{R_1 d}}$ code, C_{small} is a $[d, R_1 d, \delta_1 d]_{\mathbb{F}_2}$ code, and H is a (d,ϵ)-regular bipartite graph, then the new code C_{new} derived as above is a $\{n, R_1 k, (\delta_1 - \epsilon/\delta')\}_{\mathbb{F}_2^d}$ code.*

Note that this theorem shows that the new code will have rate and relative distance almost equal to those of the code C_{small}. In the case of ABNNR work, C_{small} was simply a repetition code, i.e., we had $R_1 d = 1$ and $C_{small} = [d, 1, d]_2$.

We will see the proof of Theorem 5 in the next lecture. We will also study some algorithms for decoding the codes constructed via this method.

[4]Note that on a random d-regular bipartite graph H', $\mathbb{E}[|E_{H'}(X,Y)|] = (\frac{|X|}{n}\frac{|Y|}{n}) \cdot dn$.

References

[1] V. Guruswami, and P. Indyk, *"Expander-Based Constructions of Efficiently Decodable Codes,"* In the proceedings of 42nd Annual Symposium on Foundations of Computer Science (FOCS'01), IEEE Computer Society, Las Vegas, Nevada, USA, 658-667.

[2] N. Alon, N., J. Bruck, J. Naor, M. Naor, and R.M. Roth, *"Construction of asymptotically good low-rate error-correcting codes through pseudo-random graphs,"* Information Theory, IEEE Transactions on , vol.38, no.2, pp.509-516, Mar 1992.

[3] N. Alon, J. Edmonds, and M. Luby, *"Linear time erasure codes with nearly optimal recovery,"* In Proceedings of the 36th Annual Symposium on Foundations of Computer Science (FOCS'95). IEEE Computer Society, Washington, DC, USA, 512-520.