

Lecture 12

*Instructor: Sruthi Sekar**Scribe: Siddharth Patil*

1 Overview

In this lecture, we look at some linear code facts and use them to try to come up with the Hamming code. We also look at the Hadamard code and the Reed-Solomon codes, their benefits and their shortcomings and look at some measures of the utility of codes. Then we define concatenation codes that help us to overcome the shortcomings of the aforementioned codes.

2 Recap

2.1 ECCs

2.1.1 Definition

Error Correcting Codes are ways to encode messages, of length k , to code words, of length n , such that we can correct up to t errors in the code words.

2.1.2 Notation

ECCs are denoted by a triple $(n, k, d)_q$, where n, k, d are as defined above, and q is the field over which the computation occurs.

2.1.3 Properties

1. **Rate of ECC** is defined as the length of the message divided by the length of the code word, i.e. k/n . We aim to maximize the rate.
2. **Minimum Distance of C**, $d(C)$, is defined as the minimum Hamming Distance between any two distinct code words. $t = \lfloor (Minimum\ Distance - 1)/2 \rfloor$, therefore we aim to maximize the minimum distance.

2.2 Linear ECCs

2.2.1 Definition

Linear ECCs are Eccs that could be represented as a matrix multiplication, i.e. $Enc(x) = xG$, where G is a $k \times n$ matrix.

2.2.2 Notation

ECCs are denoted as $[n, k, d]_q$, which is similar to that of general ECCs, the only change being that the round brackets are replaced by square brackets.

2.2.3 Error Detection

Since the encoding is a matrix multiplication, the code word lies in the row span of the generator matrix G . Therefore, we can define $C^\perp = \text{Nullspace}(G)$. Now, we construct a $(n - k) \times n$ matrix H such that the row span of H is $C^\perp$, then for all code words z , $H z = 0$. Hence, this H is called the parity check matrix. But this H could also be used as an encoding matrix, i.e. $\text{Enc}^\perp(x) = xH$, and these code words lie in $C^\perp$, therefore, $C^\perp$ is called the dual code.

3 Linear Code Facts

3.1 Minimum Distance in terms of Hamming Weight

Definition 1 (Hamming Weight). The Hamming Weight of a word is the Hamming Distance of that word from 0, i.e.

$$wt(x) = \Delta(x, 0)$$

Theorem 2. The Minimum Distance of a linear code C , $d(C)$, is the minimum Hamming Weight of any non-zero code word, i.e.

$$d(C) = \min_{y \in C, y \neq 0} wt(y)$$

Proof. From the definition of Minimum Distance,

$$d(C) = \min_{y, y' \in C, y \neq y'} \Delta(y, y')$$

From linearity,

$$\begin{aligned} \Delta(y, y') &= \Delta(y - y', 0) \\ &= wt(y - y') \end{aligned}$$

therefore,

$$d(C) = \min_{y, y' \in C, y \neq y'} wt(y - y')$$

but, since C is linear,

$$y, y' \in C \implies (y - y') \in C$$

therefore, we get,

$$d(C) = \min_{y \in C, y \neq 0} wt(y)$$

□

Hence, proved.

3.2 Minimum Distance in terms of H

Theorem 3. The Minimum Distance of a linear code C , $d(C)$, is the minimum number of columns of H that are linearly dependent.

Proof. From the previous theorem,

$$d(C) = \min\{wt(y) | y \in C, y \neq 0\}$$

but,

$$y \in C \implies Hy = 0$$

therefore,

$$d(C) = \min\{wt(y) | Hy = 0, y \neq 0\}$$

Note that Hy is just a linear combination of the columns of H , where the weight of each column H_j is equal to y_j , i.e.,

$$Hy = \sum_{j=1}^n y_j H_j$$

Therefore, finding minimum $wt(y)$, such that $\sum_{j=1}^n y_j H_j = 0$ and $y \neq 0$, is equivalent to finding the size of the smallest subset of columns of H whose linear combination, with non zero weights, could yield 0.

This implies that any subset of $d(C) - 1$ columns of H must be linearly independent.

Therefore, $d(C)$ is the minimum number of linearly dependent columns of H . □

4 Hamming Code

We will only discuss the $q = 2$ case.

4.1 Goal

Main goal: High rate, high minimum distance linear code.

But, to start with, let us relax this goal.

Intermediate goal: High rate, 'decent' minimum distance linear code.

To achieve this goal,

1. **For higher rate**, we need to maximize k , or equivalently, minimize $n - k$, given n . Hence, the matrix H should be wide.
2. **For higher minimum distance**, from Theorem 3, we want a large minimum number of linearly dependent columns in H . Also, for error correction, we need $d(C) \geq 3$.
To get $d(C) \geq 3$, our H should not have any 0 column, and no two columns should be identical.

4.2 Definition

Using the aforementioned information, a natural H that comes to mind is the matrix whose columns are $n - k$ bit natural numbers in their base 2 representation, which is the Hamming Parity Matrix.

Definition 4 (Hamming Code). A linear code having the parity matrix

$$H = \begin{bmatrix} 0 & 0 & 0 & \dots & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ 0 & 1 & 1 & \dots & 1 \\ 1 & 0 & 1 & \dots & 1 \end{bmatrix}$$

i.e.,

$$H = [c_1 \quad c_2 \quad \dots \quad c_{2^{n-k}-1}]$$

where $c_i = \text{BitRepresentation}(i)$ for $1 \leq i \leq 2^{n-k} - 1$.

As we can see, in this case, if we define $r = n - k$, then $n = 2^r - 1$, $k = n - r = 2^r - r - 1$ and $d(C) = 3$, therefore, C is a $[2^r - 1, 2^r - r - 1, 3]_2$ code.

Hence,

$$\text{Rate}(C) = \frac{2^r - r - 1}{2^r - 1} \approx 1 - \frac{\log_2(n + 1)}{n}$$

which is very high.

However, the minimum distance is only 3, which only allows us to detect up to 2 errors and correct up to 1 error.

4.3 Error locating property

Although Hamming Code cannot correct multiple errors, the construction of H gives rise to a very nice property:

If z is a codeword that is corrupted at the i th position, i.e. only one error, then $H z = c_i$.

So, just by computing $H z$, we get the exact location of the error in binary format, provided there are not multiple errors.

Proof. Let y be a codeword and e_i be the single bit error at location i , i.e. $e_i(i) = 1$ and $e_i(j) = 0$ for all $j \neq i$. Now, let $z = y + e_i$ be the codeword with error, Then,

$$\begin{aligned} H z &= H(y + e_i) \\ &= H y + H e_i \\ &= 0 + \sum_{j=1}^n H_j e_i(j) \\ &= \sum_{j=1}^n c_j e_i(j) \\ &= c_i \end{aligned}$$

Therefore, $H z = c_i = \text{BitRepresentation}(i)$. □

5 Hadamard Code

We will only discuss the $q = 2$ case.

5.1 Goal

We will flip the goal from the Hamming code case.

Intermediate goal: High minimum distance, 'decent' rate.

5.2 Definition

To achieve the given goal, we can take the dual of Hamming Code, i.e. $[c_0 \ H_{\text{hamming}}]$, is the generator matrix for this code.

Following are three equivalent definitions of Hadamard Code:

Definition 5 (Hadamard Code). A linear code having the generator matrix

$$G = \begin{bmatrix} 0 & 0 & 0 & 0 & \dots & 1 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 1 \\ 0 & 0 & 1 & 1 & \dots & 1 \\ 0 & 1 & 0 & 1 & \dots & 1 \end{bmatrix}$$

i.e.,

$$G = [c_0 \ c_1 \ \dots \ c_{2^{n-k}-1}]$$

where $c_i = \text{BitRepresentation}(i)$ for $1 \leq i \leq 2^{n-k} - 1$.

Definition 6 (Hadamard Code). A linear code whose encoding, $Enc : \mathbb{F}_2^r \rightarrow \mathbb{F}_2^{2^r}$, is defined as

$$Enc(x) := (x \cdot a)_{a \in \mathbb{F}_2^r}$$

where, $x \cdot a$ is the dot product of x and c , and $(x \cdot a)_{a \in \mathbb{F}_2^r}$ denotes the sequence of evaluations of $x \cdot a$ for all $a \in \mathbb{F}_2^r$, i.e. 2^r evaluations.

Definition 7 (Hadamard Code). A linear code whose encoding, $Enc : \mathbb{F}_2^r \rightarrow \mathbb{F}_2^{2^r}$, is defined as the truth table of the linear r -variate polynomial with the input message as the coefficients.

That is, for $x \in \mathbb{F}_2^r$, define $L_x : \mathbb{F}_2^r \rightarrow \mathbb{F}_2$ as $L_x(a) = x_1 a_1 + x_2 a_2 + \dots + x_r a_r$. Then,

$$Enc(x) := \text{Truth table of } L_x$$

Remark: This code is named the Hadamard code because if we replace $\mathbb{F}_2$ with $\{-1, +1\}$ and then list all the code words in a row, then we get the Hadamard Matrix H_{2^r} . From the definitions of Hadamard Code, it is easy to see that $n = 2^r$ and $k = r$, but the minimum distance is not that clear.

5.3 Minimum Distance

Theorem 8. The minimum distance for Hadamard code, C , with $n = 2^r$ is 2^{r-1} , i.e.

$$d(C) = \frac{n}{2}$$

Proof. We have,

$$L_x(a) = x_1 a_1 + x_2 a_2 + \dots + x_r a_r$$

and,

$$Enc(x) = \text{Truth table of } L_x$$

Using Schwartz-Zippel Lemma with $d = 1$,

$$\Pr_{\alpha_1, \dots, \alpha_r \sim \mathbb{F}_2} [L_x(\alpha_1, \dots, \alpha_r) = 0] \leq \frac{1}{2}$$

Therefore, for any two distinct messages x and x' , due to linearity of L_x , we get,

$$\begin{aligned} & \Pr_{\alpha_1, \dots, \alpha_r \sim \mathbb{F}_2} [(L_x - L_{x'}) (\alpha_1, \dots, \alpha_r) = 0] \leq \frac{1}{2} \\ \implies & \Pr_{\alpha_1, \dots, \alpha_r \sim \mathbb{F}_2} [(L_x - L_{x'}) (\alpha_1, \dots, \alpha_r) \neq 0] \geq \frac{1}{2} \end{aligned}$$

This implies that L_x and $L_{x'}$ disagree for at least $2^r/2$ values of a , which in turn implies that $Enc(x)$ and $Enc(x')$ differ in at least 2^{r-1} locations.

Therefore, $d(C) = 2^{r-1} = n/2$. □

Therefore, Hadamard is a $[2^r, r, 2^{r-1}]_2$, or equivalently, $[n, \log_2 n, n/2]_2$ code.

Remark: Hadamard code has great min distance but bad rate. But, for binary codes, it is known that if $d(C) \geq (0.5 + \varepsilon)n$, where $\varepsilon \rightarrow 0$, then C will have only $O(1)$ code words, i.e. $O(1/n)$ rate, which is worse than Hadamard's rate $O((\log_2 n)/n)$. Therefore, the Hadamard code, although giving a bad rate, optimizes minimum distance, in the binary setting.

6 Reed-Solomon Codes

Here, we do not constraint q to be 2.

6.1 Goal

In Hamming code and Hadamard code, we used intermediate goals to optimize rate or minimum distance respectively, and now we will move back to our main goal.

Goal: High rate, high minimum distance linear code, without constraints on q .

To achieve this goal, we will try to generalize the Hadamard Code.

6.2 Definition

Definition 9 (Reed-Solomon Codes). A family of linear codes with $1 \leq k < n$ and $q \geq n$, whose encoding, $Enc : \mathbb{F}_q^k \rightarrow \mathbb{F}_q^n$ is defined as

$$Enc(x) := (P_x(a))_{a \in S}$$

where $S \subseteq \mathbb{F}_q$ with $|S| = n$, $P_x(a)$ is a uni-variate polynomial of degree $k - 1$, in a , defined as

$$P_x(a) := x_0 + x_1 a + x_2 a^2 + \dots + x_{k-1} a^{k-1}$$

and $(P_x(a))_{a \in S}$ denotes the list of evaluations of $P_x(a)$ for all $a \in S$, i.e. n evaluations.

We could see that this definition is an amalgamation of Definitions 6 and 7 of Hadamard code where we move from a degree 1, r -variate polynomial to a degree $(k - 1)$, uni-variate polynomial.

That is, we are encoding the message x , by taking n evaluations of a $(k - 1)$ degree polynomial represented by that message.

Remark: Usually we take $q = n$, and $S = \mathbb{F}_n$.

6.3 Linearity

Although $P_x(a)$ is a degree $(k - 1)$ polynomial, this code is still linear because this degree is with respect to a , and all the terms with x_i appear with degree 1.

Another way to look at this is by denoting $Enc(x)$ as a matrix multiplication of x with a Vandermonde matrix

$$Enc(x) = \begin{bmatrix} x_0 & x_1 & \dots & x_{k-1} \end{bmatrix} \begin{bmatrix} 1 & 1 & \dots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ a_1 & a_2 & \dots & a_n \\ a_1^{k-1} & a_2^{k-1} & \dots & a_n^{k-1} \end{bmatrix}$$

6.4 Minimum Distance

Theorem 10. *The minimum distance of Reed-Solomon codes is at least $n - (k - 1)$*

Proof. We have,

$$P_x(a) = x_0 + x_1 a + x_2 a^2 + \dots + x_{k-1} a^{k-1}$$

and,

$$Enc(x) := (P_x(a))_{a \in S}$$

Using Schwartz-Zippel Lemma with $d = k - 1$,

$$\Pr_{a \sim \mathbb{F}_q} [P_x(a) = 0] \leq \frac{k - 1}{q}$$

Therefore, for any two distinct messages x and x' , due to linearity of P_x in x , we get,

$$\begin{aligned}
& \Pr_{a \sim \mathbb{F}_q} [(P_x - P_{x'})(a) = 0] \leq \frac{k-1}{q} \\
\Rightarrow & \Pr_{a \sim Y, Y \subseteq \mathbb{F}_q} [(P_x - P_{x'})(a) = 0] \leq \frac{k-1}{|Y|} \\
\Rightarrow & \Pr_{a \sim S} [(P_x - P_{x'})(a) = 0] \leq \frac{k-1}{n} \\
\Rightarrow & \Pr_{a \sim S} [(P_x - P_{x'})(a) \neq 0] \geq 1 - \frac{(k-1)}{n}
\end{aligned}$$

This implies that P_x and $P_{x'}$ disagree for at least $n - (k-1)$ values of $a \in S$, which in turn implies that $Enc(x)$ and $Enc(x')$ differ in at least $n - (k-1)$ locations.

Therefore, $d(C) = n - (k-1)$ □

Therefore, Reed-Solomon codes are $[n, k, n - (k-1)]_q$ codes, where $q \geq n$.

6.5 Rate - Minimum Distance Trade-off

Using the singleton bound (11), it could be shown that for $q \geq n$, the rate - minimum distance trade-off is optimal for Reed-Solomon codes.

Theorem 11 (Singleton Bound). *For $(n, k, d)_q$ code (need not be linear), $k \leq n - d + 1$, i.e.*

$$d \leq n - k + 1$$

Proof. For any code, we can delete the first $(d-1)$ letters from each code word and they will still be distinct, because the minimum distance is d .

The length of the new code words is $n - d + 1$ and since they are distinct, there is a one to one relation from the message space to the new code word space.

This implies that the size of the new code word space has to be at least as large as that of the message space. Therefore,

$$\begin{aligned}
q^k & \leq q^{n-d+1} \\
\Rightarrow k & \leq n - d + 1 \\
\Rightarrow d & \leq n - k + 1
\end{aligned}$$

Hence, proved. □

Note: Although Reed-Solomon codes give great rate and minimum distance, they use a large alphabet, $q \geq n$.

7 Analysis

To analyze the codes and compare their performance/utility, we will define asymptotic rate and minimum distance.

Definition 12 (Asymptotic Rate). For a code $[n, k, d]_q$, its asymptotic rate is defined as,

$$\mathcal{R}(C) = \lim_{n \rightarrow \infty} \frac{k(n)}{n}$$

Definition 13 (Asymptotic Minimum Distance). For a code $[n, k, d]_q$, its asymptotic minimum distance is defined as,

$$\delta(C) = \lim_{n \rightarrow \infty} \frac{d(n)}{n}$$

Then, we have

$$\begin{array}{ll} \mathcal{R}(\text{Hamming}) = 1 & \delta(\text{Hamming}) = 0 \\ \mathcal{R}(\text{Hadamard}) = 0 & \delta(\text{Hadamard}) = \frac{1}{2} \\ \mathcal{R}(\text{Reed Solomon}) = R_0 & \delta(\text{Reed Solomon}) = 1 - R_0 \end{array}$$

where R_0 is any value in the range $(0, 1)$.

So, the Reed-Solomon codes look like they achieve our main goal, but they require a variable alphabet size, which is generally not feasible. Hence, we should alter our goal to account for this.

Dream Goal: Asymptotically good code with a fixed alphabet size, i.e., C with

fixed q

$$\mathcal{R}(C) \geq R_0 > 0$$

$$\delta(C) \geq \delta_0 > 0$$

8 Toward the Dream Goal

8.1 Feasibility

To show that good codes exist, the following theorem was shown without proof

Theorem 14 (Feasibility). *For all q and for all $0 \leq \delta_0 \leq 1 - 1/q$, there exists a C with $\delta(C) = \delta_0$ and $\mathcal{R}(C) = 1 - h_q(\delta_0)$.*

Where, $h_q(\delta_0)$ is the q -ary entropy function, such that the number of q -ary strings in a Hamming ball of radius $\delta_0 n$ is $\tilde{\Theta}(q^{h_q(\delta_0)n})$.

8.2 Concatenation Codes

Now that we know that good codes exist, the next natural question is whether they are efficiently encodable and decodable. The following theorem, again without proof, shows that this is possible.

Theorem 15 (Justesen). *For $q = 2$, there exist $\text{poly}(n)$ -time encodable and decodable asymptotically good linear codes. That is, there exist C with*

$$q = 2$$

$$\mathcal{R}(C) = R_0$$

$$\delta(C) \approx 0.11(1 - R_0)$$

for all R_0 in the range $(0, 1)$.

This theorem uses concatenation codes which are defined below.

8.3 Definition

Definition 16 (Concatenation Codes). A concatenation code C of outer code C_o and inner code C_i is obtained by first encoding the message using C_o and then encoding each alphabet of this intermediate

encoded code word using C_i .

That is, let C_o be $[n_o, k_o, d_o]_{q_o}$ and C_i be $[n_i, k_i, d_i]_{q_i}$, with $q_o = q_i^{k_i}$, then the encoding of C is

$$\begin{bmatrix} m_{1,1} \\ m_{1,2} \\ \vdots \\ m_{1,k_i} \\ m_{2,1} \\ m_{2,2} \\ \vdots \\ m_{k_o,k_i} \end{bmatrix} \xrightarrow{\text{Enc}} \begin{bmatrix} c_{1,1} \\ c_{1,2} \\ \vdots \\ c_{1,n_i} \\ c_{2,1} \\ c_{2,2} \\ \vdots \\ c_{n_o,n_i} \end{bmatrix}$$

where the encoding procedure is as follows

$$\begin{aligned} & \begin{bmatrix} m_{i,1} \\ m_{i,2} \\ \vdots \\ m_{i,k_i} \end{bmatrix} \xrightarrow[\text{from } q_i \text{ to } q_o]{\text{Base conversion}} m_i \\ & \begin{bmatrix} m_1 \\ m_2 \\ \vdots \\ m_{k_o} \end{bmatrix} \xrightarrow{\text{Enc}_o} \begin{bmatrix} t_1 \\ t_2 \\ \vdots \\ t_{n_o} \end{bmatrix} \\ & t_j \xrightarrow[\text{from } q_o \text{ to } q_i]{\text{Base conversion}} \begin{bmatrix} t_{j,1} \\ t_{j,2} \\ \vdots \\ t_{j,k_i} \end{bmatrix}, \forall 1 \leq j \leq n_o \\ & \begin{bmatrix} t_{j,1} \\ t_{j,2} \\ \vdots \\ t_{j,k_i} \end{bmatrix} \xrightarrow{\text{Enc}_i} \begin{bmatrix} c_{j,1} \\ c_{j,2} \\ \vdots \\ c_{j,n_i} \end{bmatrix}, \forall 1 \leq j \leq n_o \end{aligned}$$

8.4 Notation ($\diamond$)

Given the outer and inner codes $[n_o, k_o, d_o]_{q_o}$ and $[n_i, k_i, d_i]_{q_i}$, respectively, then their concatenation code is denoted as $[n_o, k_o, d_o]_{q_o} \diamond [n_i, k_i, d_i]_{q_i}$.

8.5 Resulting code

Theorem 17. Let C_o be $[n_o, k_o, d_o]_{q_o}$ and C_i be $[n_i, k_i, d_i]_{q_i}$, then their concatenation code $C = C_o \diamond C_i$ is a $[n_o n_i, k_o k_i, d_o d_i]_{q_i}$ code.

Proof. Let C be a $[n, k, d]_q$ code.

From the definition of concatenation codes, it is easy to see that $n = n_o n_i$, $k = k_o k_i$ and $q = q_i$.

Now, to prove that $d = d_o d_i$, we will first use

$$\begin{bmatrix} m_{i,1} \\ m_{i,2} \\ \vdots \\ m_{i,k_i} \end{bmatrix} \xrightarrow[\text{from } q_i \text{ to } q_o]{\text{Base conversion}} m_i$$

$$\begin{bmatrix} m_1 \\ m_2 \\ \vdots \\ m_{k_o} \end{bmatrix} \xrightarrow{\text{Enc}_o} \begin{bmatrix} t_1 \\ t_2 \\ \vdots \\ t_{n_o} \end{bmatrix}$$

This implies that by changing the message, at least d_o values of t_j are changed.

Then, we use

$$t_j \xrightarrow[\text{from } q_o \text{ to } q_i]{\text{Base conversion}} \begin{bmatrix} t_{j,1} \\ t_{j,2} \\ \vdots \\ t_{j,k_i} \end{bmatrix}, \forall 1 \leq j \leq n_o$$

$$\begin{bmatrix} t_{j,1} \\ t_{j,2} \\ \vdots \\ t_{j,k_i} \end{bmatrix} \xrightarrow{\text{Enc}_i} \begin{bmatrix} c_{j,1} \\ c_{j,2} \\ \vdots \\ c_{j,n_i} \end{bmatrix}, \forall 1 \leq j \leq n_o$$

which implies that when a t_j is changed, then at least d_i values of $c_{j,k}$ are changed.

These two results combined give us that when the message is changed, at least $d_o d_i$ values of $c_{j,k}$ are changed, which implies that the minimum distance of C is at least $d_o d_i$.

Hence, proved. $\square$

Example 18 (Reed-Solomon $\diamond$ Hadamard). *Reed-Solomon is a $[q, k, q - k + 1]_q$ code, and Hadamard is a $2^r, r, 2^{r-1}$ code. Now, let $q = 2^r$, therefore, the Hamadard code could be written as a $[q, \log_2 q, q/2]_2$ code. Then the resulting concatenation code is a $[q^2, k \log_2 q, q(q - k + 1)/2]$ code. Now, if $k = \varepsilon q$, ($0 < \varepsilon < 1$), then $\delta \approx 1/2$ and the rate is $O((\log_2 q)/q)$.*