A STUDY ON LINEAR ERROR CORRECTING CODES

Project report submitted to

MAHATMA GANDHI UNIVERSITY

In partial fulfilment of the requirement for the award of the degree of

MASTER OF SCIENCE IN MATHEMATICS

Submitted by

SHILPA VB
Reg. No. 200011014786

Under the supervision of

Mrs.Karthika V

DEPARTMENT OF MATHEMATICS

Bharata Mata College, Thrikkakara
2020-2022

ACKNOWLEDGEMENT

I sincerely thank each and everyone who has helped me for the successful completion of this project. I thank God Almighty for all his blessings showered upon me.

I would like to place on record all my gratitude to Mrs.Karthika V, Department of Mathematics, Bharata Mata College, Thrikkakara, who has provided her valuable guidance throughout this project work.

Also, I thank all other teachers of the department for their encouragement. My friends were also helpful to me through their suggestions for which I am grateful towards them.

Place : thrikkakara

Date : 26/09/2022 SHILPA VB

CERTIFICATE

This is to certify that the dissertation entitled “STUDY ON LINEAR ERROR CORRECTING CODES “submitted by Shilpa VB is a record of work done by the candidate during the period of her study under my supervision and guidance .

Mrs. karthika V
Department of mathematics
Bharata mata college,
Thrikkakara.

Place : Thrikkakara
Date : 26/09/2022

DECLARATION

I hereby declare that the project report entitled “STUDY ON LINEAR ERROR CORRECTING CODES” submitted for the M.sc Degree is my orginal work done under the supervision of Mrs.Karthika V and the project has not formed the basis for the award of any academic qualification fellowship or other similar title of any other university or board.

Place :Thrikkakara

Date : 26/09/2022 SHILPA VB

CONTENTS

ABSTRACT

Coding theory is the study of the methods for efficient and accurate of information from one place to another .The theory has been developed for such diverse application as the minimization of noise from compact disc recordings data trasfer from one computer to another or from memory to central processor,and information transmission from a distant source such as weather or communication satellite or the voyager spacecraft which sent pictures of jupitor and saturn to Earth.

Moving some data from one place to another requires the data to move through some medium. What we wish is that the data which one party sends is exactly the same as the data which the other party receives. But in general, there is not always a guarantee that the data really stays the same as it goes through the medium. For example, a satellite sends data in the form of radiowaves to some station on earth, but some interference can easily change that signal. This means the data received at the station might not be the same as the original. This is because of the presence of different kinds of noises. The subject of error correcting codes (ECC) arose originally in response to practical problem in the reliable communication of digitally encoded information. Since then algebraic coding has developed many connections with portions of algebra and combinatorics. The history of ECC started with the introduction of hamming codes (Hamming 1974), at or about the same time as the seminal work of Shannon (1948). Shortly after, Golay codes were invented (Golay 1947). These two first classes of codes are optimal, and will be discussed in this project.

All error correcting codes are based on the same basic principle: *redundancy* is added to information in order to correct any error that may occur in the process of transmission or storage. The message to be communicated is first “encoded”, i.e. turned into a codeword, by

adding redundancy. The codeword is then sent through the channel and the received message is “decoded” by the receiver into a message resembling, as closely as possible, the original message. The degree of resemblance will depend on how good the code is in relation to the channel. There are various types of ECC but in this project we restrict our attention to linear block codes (or simply linear codes) only, which has wide applications. Linear codes allow for more efficient encoding and decoding algorithms than other codes.

In the first chapter of this project a more detailed explanation of linear codes with definitions and examples to clarify them are explained. Then in the second chapter, some of the encoding and decoding algorithms such as nearest neibourhood(maximum likelihood) decoding, and syndrome decoding are discussed. The third chapter is left to discuss some important classes of linear codes such as Hamming codes and Golay codes. Some of the applications of linear ECC codes in various fields are also discussed in this chapter.

INTRODUCTION

Information media ,such as communinication systems and storage devices of data are not absolutely reliable in practice because noise or other forms of introduced interference.One of the task in coding theory is to detect,or even correct errors.

The common feature of communication channels is that information is emanating from a source and is sent over the channel to a receiver at the other end.

A communication channel is illustrated in the figure 1.1 at source ,a message denoted **x** in the figure is to be send.If no modification is made to the message and it is transmitted directly over channel,any noise would disort the message so that it is not recoverable.In order to avoid this we add some redundancy to the orginal message so that hopefully the received message is the orginal message that was sent.The redundancy is added by the encoder called a code word **c** in the figure in the form of an error vector e distords the code word producing a code word $\mathbf{y}^1$ .The received vector is then sent to be decorded where the errors are removed. The redundancy is then removed and we will get an estimate $\hat{\mathbf{x}}$ of the orginal message.Hopefully $\hat{\mathbf{x}} = \mathbf{x}$

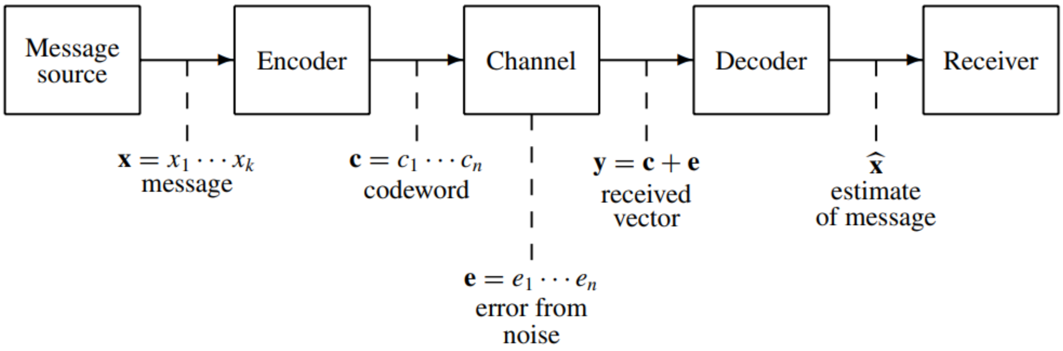


Figure 1.1

The goal of channel coding is to construct encoders and decoders in such a way as to effect:

1) Fast encoding of messages

2) Easy transmission of encoded messages

3) Fast decoding of received messages

4) Maximum transfer of information per unit time

5) Maximal detection or correction capability

Linear codes are flexible and powerful class of error detection and correction codes which enable efficient encoding and decoding procedures. To understand more about linear codes first familiarize with basic definitions.

1.LINEAR CODES

1.1 Linear codes

Definition 1.1: Let A_q^n denote the veotor space of all n-tuple over the field A_q $A_q = \{a_1, a_2, \ldots a_q\}$ be an alphabet we call the a_i values symbols. A block code C of length n over A_q is a subset ofA_q^n . A vector $c \in C$ is called a codeword. The number of elements in C, denoted $|C|$, is called the size of the code. If C is not only a subset of, A_q^n but a subspace as well, then C is a linear code.$|C|$ is called the dimension of linear code. A code of length n and size k is called a (n, k) code.

Example 1.1: A code over $A = \{0,1\}$ is a binary code. Here A is an alphabet, 0 and 1 are symbols.
$C = \{(0,0,1)(1,0,0)(1,1,1)\}$ subset of A^3 is a block code and in this case $|C| = 3$. Any element in C is called a binary code. In general, if A contains q elements, then it is called a q-ary code.

Definition 1.2: Let $S \subseteq A_q^n$. The set of all vectors orthogonal to S is denoted by $S^\perp$ is called the orthogonal complement of S. If $C =< S >$, then $C^\perp =< S^\perp>$ which is also a linear code and is called dual code of C. If $C = C^\perp$, it is called a self dual code.

Definition 1.3: Hamming Weight of a vector v in A_q^n is given by the number of non zero entries in v and is denoted by $wt(x)$.

Example 1.2: If U=(10110) then wt(U)=3
If U=(1100) then wt(U)=2

Definition 1.4: Minimum Weight is the minimum of hamming weight of all codewords in a block code C.
$wt(C) = min\{wt(x) \colon x \in C, x \neq 0\}$.

Definition 1.5: Hamming distance between vector u and v in A_q^n is given by the number of non zero entries in their difference i.e., d: $A_q^n \times A_q^n \rightarrow \mathbb{Z}$ is given by

$d(u, v) = wt(u - v)$.

Example 1.3: If U=(10101010) and v=(10111000) then u and v differs in two places so that d(u,v)=2

Remark 1.1: The Hamming distance between x and y is the same as the Hamming weight of $(x \oplus y)$. The symbol $\oplus$ means the bitwise XOR operator.

Definition 1.6: Minimum distance of a code C is the smallest distance between distinct pairs of vectors of C. If C is a linear code the difference of u and v is also in C. So the minimum distance is then the minimum weight over all non zero vectors in C. Minimum distance of a codeword C is denoted by $d(C)$

Example 1.4: Consider C = {100,011,111} which is a subset of A_q^n. Let x = 100 , y = 011 , z = 111 . Then,

$$wt(x) = 1$$

$$wt(y) = 2$$

$$wt(z) = 3$$

$$\begin{aligned} wt(C) &= min\{wt(x), wt(y), wt(z)\} \\ &= min\{1,2,3\} \\ &= 1 \end{aligned}$$

$$d(x, y) = 3$$

$$d(y, z) = 1$$

$$d(z, x) = 2$$

$$\begin{aligned} d(C) &= min\{d(x, y), d(y, z), d(z, x)\} \\ &= min\{3,2,1\} \\ &= 1 \end{aligned}$$

Theorem 1.1: If C is a linear code in, then $d(C) = wt(C)$

.

Proof. Let x and y are any two different codewords. Then by using Remark 1.1 and the fact that $x - y$ is some codeword $z \neq 0$, we have:

$$\begin{aligned} d(x, y) &= wt(x - y) \\ &= wt(z) \\ &\geq \min\{ wt(z) | z \in C, z \neq 0 \} \\ &= wt(C) \end{aligned}$$

This holds for any two different x and y, thus

$$d\ (C) \geq wt(C).$$

On the other hand, fixing one codeword to be the null codeword, we have:

$$\begin{aligned} d(C) &= \min\{d(x, y) \mid x, y \in C, x \neq y \} \\ &\leq \min\{d(0, y)|y \in C, y \neq 0\} \\ &= \min\{ wt(y)| \ y \in C, y \neq 0\} \\ &= wt(C) \end{aligned}$$

i.e., $d(C) \leq wt(C)$

Combining the two inequalities gives us $d(C) = wt(C)$.

Theorem 1.2 : Let C be a linear code of length n over Aq. Then,

i) $|C| = q^{\ dim(C)}$

ii) $C^{\perp}$ is a linear code and $dim(C) = log\ |C|$

iii) $(C^{\perp})^{\perp}$ = C

Definition 1.7: The rate of a code is the ratio $R = \frac{k}{n}$ where k is the dimension and n is the length of the code.

Definition 1.8: The relative minimum distance of a code is the ratio $\delta = \frac{d}{n}$ where d is the minimum distance and n is the length of the code.

Definition 1.9: The information rate of C is defined to be $R(C) = (log_q\ k)/\ n$ where k is the size and n is the length of the code.

Definition 1.10: Two (n, k) codes over Aq are equivalent if one can be obtained from the other by a combination of operations of the following types:

(i) permutation of the n digits of the codewords;

(ii) multiplication of the symbols appearing in a fixed position by a nonzero scalar.

The most common ways to represent a linear code are with either a generator matrix or a parity check matrix .

1.2Generator matrix

In Coding Theory, a generator matrix is a matrix whose rows form a basis for a linear code. The codewords are all of the linear combinations of the rows of this matrix. That is, the linear code is the row space of its generator matrix.

A $k \times n$ matrix G is a generator matrix for some linear code, if the rows of G are linearly independent; that is, if the rank of G equals k. A linear code generated by a $k \times n$ generator matrix G is called a $[n, k]$ code . An $[n, k]$ code with distance d is said to be an $[n, k, d]$ code. If G_1 is row equivalent to G then G_1 also generates the same linear code C. If G_2 is column equivalent to G, then the linear code C_2 generated by G_2 is not equal to C, but equivalent to C.

Standard form of a generator matrix: A generator matrix of an code $[n, k]$ C is in standard form if G= [I,A] where I is the k×k identity matrix A is a k×(n-k) matrix.

Example 1.5: Example 1.3: Consider the 3×3 matrix

$$H = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}$$

H forms a generator matrix for A_2^3 since {100, 010, 111} is a basis of A_2^3

Example 1.6:consider the 3×5 generator matrix

$$G = \begin{pmatrix} 0 & 0 & 11 & 0 \\ 0 & 1 & 01 & 0 \\ 1 & 0 & 01 & 0 \end{pmatrix}$$

of rank 3. By interchanging the first row and third row, we obtain another generator matrix,

$$G_1 = \begin{pmatrix} 1 & 0 & 01 & 0 \\ 0 & 1 & 01 & 0 \\ 0 & 0 & 11 & 0 \end{pmatrix} =[I_3|A]$$

for the same linear code. Note that G and G_1 are in reduced row echelon form (RREF). This linear code has an information rate of 3/5 (i.e., G and G_1 accept all the messages in A_2^3 and changes them into words of length 5. The generator matrix $G_1 = [I_3|$ A] is said to be in standard form, and the code C generated by G is called systematic code. Not all linear codes have a generator matrix in standard form.

For example, the linear code C ={000,100,001,101} has six generator matrice

$G_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ $G_2 = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$ $G_3 = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 1 \end{bmatrix}$ $G_4 = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 1 \end{bmatrix}$ $G_5 = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$
$G_6 = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix}$

None of these matrices are in standard form. Note that the matrix G'= $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$

Is in standard form generates the code C'={000,100,010,110} which is equivalent to C.

1.3Parity-check matrix

A matrix H is called a parity-check matrix for a linear code C of length n generated by the matrix G, if the columns of H form a basis for the dual code $C^{\perp}$· The parity-check matrix for a given binary linear code can be derived from its generator matrix (and vice versa). If the generator matrix for an [n, k] code is in standard form $G = (I_k|P)$, then the parity check matrix is given by,

$$H = \begin{pmatrix} P \\ I_{n-k} \end{pmatrix}$$

For example, if a binary code has the generator matrix,

$G = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{pmatrix}$, then its parity check matrix is $= \begin{pmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$

If G is a generator matrix of a self-dual code, then $H = G^T$. both the generator matrices

$$G = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \end{pmatrix} \text{ and } G_1 = \begin{pmatrix} 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \end{pmatrix}$$

generate self-dual codes but only G_1 is in *RREF*. If G=[I | B] is a generator of a selfdual code, then$B^{\,2} = I$.

Proposition 1.1:

Let H be a parity-check matrix for a linear code C generated by the k× n matix G. Then

1) The rows of G are linearly independent

2) The columns of H are linearly independent

3) $GH = Z_k$, where Z_k is the k× k zero matrix

4) By permuting columns of H, we obtain another parity-check matrix of G

5) $dim(C) = rank(G),\ dim(C^{\perp}) = rank(H)$,and$dim(C) + dim(C^{\perp}) = n$

6) $H^T$is a generator matrix for $C^{\perp}$ with G^T its parity-check matrix

7) If C is self-dual with $G = [I_k \mid B]$ its generator, then $G_1 = [B \mid I_k]$ also generates C;

8) C has distance d if and only if any set of $d-1$ rows of $H$is linearly independent, and at least one set of $d$rows of H is linearly dependent.

1.4 Algorithm for Finding Generator and Parity-Check Matrices

An algorithm to find generator matrix and parity-check matrix of a linear code can be better explained with the following examples.

Example 1.5 : Let S={01100,01010,11100,00110} be a subset of F_2^5generating the

linear code C.By using the words in S,we define the matrix

$$M = \begin{bmatrix} M_1 \\ M_2 \\ M_3 \\ M_4 \end{bmatrix} = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \end{pmatrix}$$

Note that

$$M_1 + M_2 = 0\,1\,1\,0\,0 + 0\,1\,0\,1\,0 = 0\,0\,1\,1\,0 = M_4$$

Thus the linear binary code C generated by S has dimension 3; so the matrix

$$G = \begin{bmatrix} M_1 \\ M_2 \\ M_3 \end{bmatrix} = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 \end{pmatrix}$$

is a generator matrix.

Now we use some row operations on G, to obtain a generator matrix in standard form. Let

$$E_1 = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad E_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}, \quad E_3 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}$$

be elementary matrices, then

$$G_1 = E_3 E_2 E_1 G = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 \end{pmatrix}$$
$$= \begin{pmatrix} 1 & 0 & 0 & \vdots & 0 & 0 \\ 0 & 1 & 0 & \vdots & 1 & 0 \\ 0 & 0 & 1 & \vdots & 1 & 0 \end{pmatrix} \quad .$$

is a generator matrix in standard form.

To obtain a parity-check matrix of the linear code, we form the matrix

$$B = \begin{pmatrix} 0 & 0 \\ 1 & 0 \\ 1 & 0 \end{pmatrix}$$

Form the last two columns of $\mathrm{G}_{1;}$ the matrix

$$H_1 = \begin{bmatrix} B \\ I_2 \end{bmatrix} = \begin{pmatrix} 0 & 0 \\ 1 & 0 \\ 1 & 0 \\ \cdots & \cdots \\ 1 & 0 \\ 0 & 1 \end{pmatrix}$$

will be a parity-check matrix associated to the generator matrix $G_{1.}$

Example 1.7:

$Let\ S = \{1010010101,0001010001,0000100100,0000001001,0000000011\}$ be a linearly independent set generating C. The generator matrix

$$G = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}$$

is in *RREF* but not in standard form. We permute the columns of G into order 1,4,5,7,9,2,3,6,8,10 to form the matrix

$$G_1 = G * P = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$= \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & \vdots & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & \vdots & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & \vdots & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & \vdots & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & \vdots & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Then we form the matrix H_1 and finally rearrange the rows of H_1in to their natural order to form the parity check matrix H

$$H_1 = \begin{bmatrix} B \\ I_5 \end{bmatrix} = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \; ; \; H = P * H_1 = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

The columns of H form a basis for $C^{\perp}$

Thus it is clear that parity-check matrix can be generated from generator matrix and vice versa.

2.ENCODING AND DECODING OF LINEAR CODES

2.1 Encoding

Suppose we have a block code with n data bits and k extra bits. The encoding function is simply a function which works on the n data bits and creates a longer binary string, consisting of $n + k$ bits.

So the encoder just maps each data string to a longer string of length $n + k$, which is called a codeword. It should also be clear that the encoder should be an injective function, since we do not want the same codeword for different data strings. Therefore k should not be negative. It also does not make sense to have $k = 0$, since it would then just permute the data strings. So the problem of designing a block code really lies in choosing the codewords of the bigger set wisely. To demonstrate the main concept of what a good block code is, let us suppose we have two different data strings d_1 and d_2 which the encoder will respectively encode as codewords c_1 and c_2. In the next two figures we also assume that each circle in the set represents a binary string, and that they are ordered such that two circles are neighbours only if the Hamming distance between them is 1.

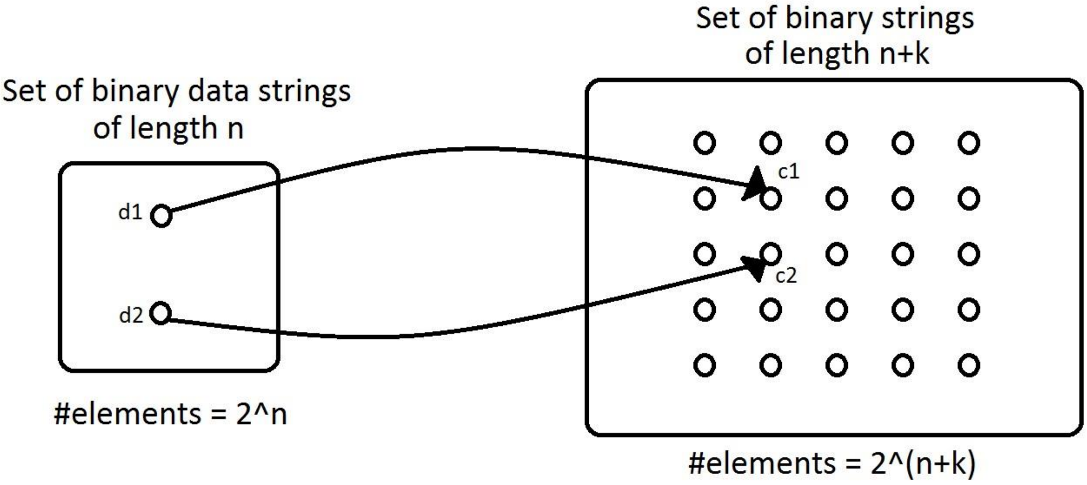


Figure 2.1

Scenario 1: The Hamming distance of c_1 and c_2 is very small. In other words, the two codewords c_1 and c_2 look a lot like each other. This means that when the codeword c_1 goes through the noisy channel, a single bitflip can turn c_1 into c_2. So when the decoder gets the binary string c_2, the obvious strategy would be to decode c_2 as the data string d_2 which is wrong.

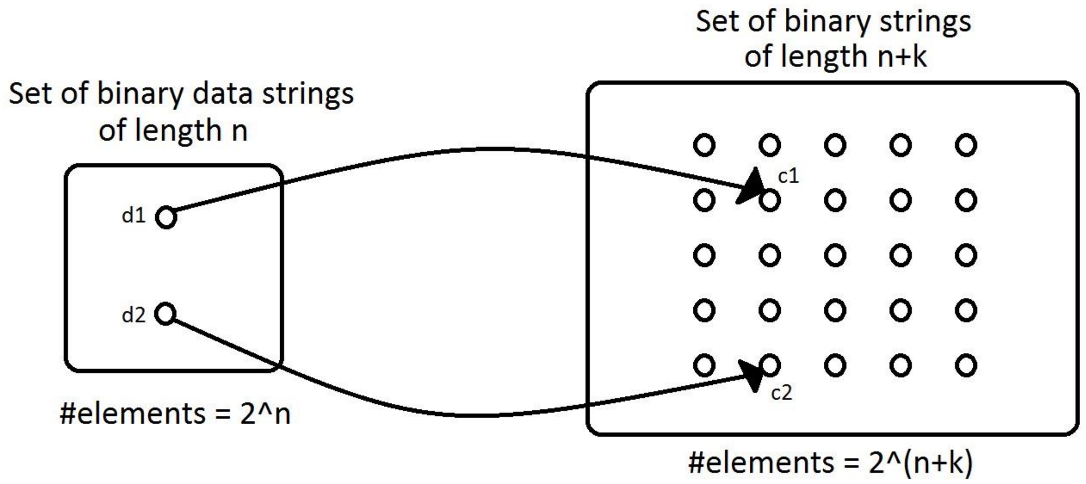

Set of binary data strings of length n
d1
d2
#elements = 2^n
Set of binary strings of length n+k
c1
c2
#elements = 2^(n+k)

Figure 2.2

Scenario 2: The Hamming distance of c_1 and c_2 is big. In this case, even if there is a biflip, the corrupted codewords $\tilde{c}$ will still be close to c_1. The decoder can then use the strategy: Pretend that the received binary string is just the codeword which lies closest to it, and then continue decoding as normal. In this case, it means the corrupted codeword $\tilde{c}$ will still be decoded as d_1, which is the error-correcting ability we want. If the strategy is to only accept binary strings which are exactly equal to a codeword, then the decoder would detect the error and we have an error-detecting code.

So we see that if the encoding function distributes the codewords in such a way that the Hamming distance for all pairs of codewords is as big as possible, then it will be more difficult for one codeword to turn into another codeword and the decoder is thus able to detect or correct more errors.

Let C be a $[n, k, d]$ linear code over the finite field A_q. Each codeword of C can represent one piece of information so C can represent q^k distinct pieces of information. Once a basis $\{r_1, r_2, \dots, r_k\}$ is fixed for C each codeword v or equivalently, each of the q^k pieces of information can be uniquely written as the linear combination

$$v = u_1 r_1 + \cdots + u_k r_k \text{where } u_1, \dots, u_k \in F_q$$

Equivalently, we may set G to be the generator matrix of C whose i^{th} row is the vector $r_i$in the chosen basis . Given a vector $u = (u_{1.} \dots . u_k) \in A_q^k$. It is clear that

$$v = uG = u_1r_1 + \cdots + u_kr_k = \sum_{i=1}^{k} u_ir_i$$

is a codeword in C. Conversely, any v∈ Ccan be written uniquely as $v = uG$ where

$u = (\, u_1, \dots, u_k) \in A_q^k$. Hence, every word $u \in A_q^k$ can be encoded as $v = uG$.The

process of representing the elements u of A_q^k as codewords $v = uG$ in C is called encoding.

Remark 2.1:the encoding rule is even simpler if G is in standard form .Suppose

G = [I_k |A],where A= (a_{ij}) is a k×(n-k) matrix .Then the message vector u is encoded as

$$X = uG = x_1x_2 \dots . x_kx_{k+1} \dots . x_n$$

Where x_i=u_i $1 \leq i \leq k,$ are the message digits and

$x_{k+1} = \sum_{j=1}^{k} a_{ij}\, u_{ij}, 1 \leq \; i \leq n - k$

are the check digits. The check digit represents redundancy which has been added to the message to give protection against noises.

Example 2.1: Given the message $(0,1,0,1)$, the encoded code word

$$(0,1,0,1) * \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix} = (0,1,0,1,1,0,0)$$

is just the sum of the second and fourth row of G.

For a general linear code, we summarize the encoding part of the communication scheme in the following diagram

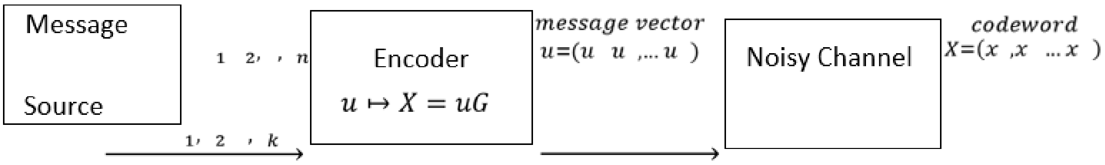


Figure 2.3

2.2. Decoding

A code is of practical use only if an efficient decoding scheme can be applied to it. There are so many methods to decode an encoded code. But some methods are exclusively for certain types of codes. In this section, we discuss a rather simple but elegant nearest neighbor (maximum likelihood) decoding for linear codes, as well as a modification that improves its performance when the length of the code is large.

2.2.1. Cosets

We begin with the notion of a coset. Cosets play a crucial role in the decoding schemes to be discussed in this chapter.

Definition 2.1: Let C be a linear code of length n over A_q^n , and let $u \in A_q^n$, be any vector of length n;we define the coset of C determined by u to be the set

$$C + u = \{v + u : v \in C\}$$

Definition 2.2: A word of least (Hamming weight) in a coset is called a *cosetleader*.

Remark 2.2: By considering the vector addition A_q^n is a finite abelian group, and a linear code C over A_q^n, of length n is also a subgroup of A_q^n . Therefore, the coset of a linear code defined above coincides with the usual notion of a coset in group theory.

Example 2.2.1 :Let q= 2 and C = {000,101,010,111}.Then,

$$C + 000 = \{000, 101, 010, 111\},$$
$$C + 001 = \{001, 100, 011, 110\},$$
$$C + 010 = \{010, 111, 000, 101\},$$
$$C + 011 = \{011, 110, 001, 100\},$$
$$C + 100 = \{100, 001, 110, 011\},$$
$$C + 101 = \{101, 000, 111, 010\},$$
$$C + 110 = \{110, 011, 100, 001\},$$
$$C + 111 = \{111, 010, 101, 000\}.$$

Note that

C+000 = C+010 = C+101 = C+111 = C

C+001= C+011=C+100=C+110=$A_3^2\backslash C$

Proposition 2.1: *Let C be a* [n,k,d] *linear code over the finite field A_q^n . Then,*

(i) Every vector of A_q^n , is contained in some coset of C

(ii) For all $u \in A_{q.}^n$ $|c + u| = |c| = q^k$

(iii) For all u, v∈ A_q^n *, $u \in C + v$ implies that $C + u = C + v$*

(iv) Two cosets are either identical or they have empty intersection

(v) There are q^{n-k}different cosets of C

(vi) For all u,v∈ A_q^n ,u-v∈ C, *if and only if u and* v*are in the same coset.*

2.2.2. Maximum likelihood decoding

The method of decoding a received vector to the closest code vector is called maximum likelihood decoding.

Let C be a linear code. Assume the codeword $v$is transmitted and the word $w$is

received, resulting in the error pattern (or error string)

$e = w - v \in w + C.$

Then $w - e = v \in C$, so the error pattern e and the received word w are in the same coset. Since error patterns of small weight are the most likely to occur, nearest neighbor decoding works for a linear code C in the following manner. Upon receiving the word w, we choose a word e of least weight in the coset $w + C$ and conclude that $v = w - e$was the codeword transmitted.

Definition 2.2 : We define a sphere of rarius ‘r’ about a vector u, denoted by $S_r(u)$ as

$$S_r(u) = \{v \in V/d(u,v) \leq r\}$$

Note: Let [x] denote greatest integer less than or equal to x.

Theorem 2.1 : If d the minimum weight of a code C then C can correct t =$[\frac{d-1}{2}]$ or fewer errors and conversely.

Proof : First we prove that spheres of radius ‘t’ about codewords are disjoint.If possible suppose that ,they are not.Then there exist distinct vectors u,w $\in C$ such that , $S_t(u) \cap S_t(w) \neq \emptyset$.

Sppose v $\in S_t(u) \cap S_t(w)$

Thus v$\in S_t(u)$ and v$\in S_t(w)$ so d(u,v) $\leq t$ and d(v,w) $\leq t$

Now d(u,w) $\leq d(u,v) + d(v,w)$

$\leq t + t$

=2t

But 2t$\leq d - 1$

∴d(u,w) $\leq d - 1$ → (1)

As u,w are distinct ,u-w is a non zero vector.So

Wt(u-w) $\geq d$

ie ; d(u,w) $\geq d$

This contrdicts (1)

∴ Our assumption is wrong..

So sphere of radius t about the code words are disjoint.

If t or fewer errors occur,then the number of the positions in which the received vector v and the orginal vector say u differ is less than or equal to t.

Ie; d(u,v) $\leq t$

As sphere of radius t about code words are disjoint ,the received vector v is in a sphere of radius t about a unique closet code word u.We decode v to u.So t or fewer errors can be corrected.

Converse follows immediately.

Definition 2.3 : -If $[\frac{d-1}{2}]$ or fewer errors have occurred ,the received vector v will be ina shere of radius $[\frac{d-1}{2}]$ about unique code word.If more errors have occurred ,there could be several code words at a smallest distance from v.If we need to decode every received vector we could choose any one of these and and decode v to it .This is called **complete decoding.**

Definition 2.4 : Another decoding technique is to decode only those received messages that have $[\frac{d-1}{2}]$ or fewer errors and detect the others .This is called **incomplete decoding.**

Example 2.3: Let q = 2 and C = {0000,1011,0101,1110}.Decode the following received word:

W = 1101 w = 1111

First we write down the standard array of C

$$\begin{array}{l} 0000 + C : 0000\ \ 1011\ \ 0101\ \ 1110 \\ 0001 + C : 0001\ \ 1010\ \ 0100\ \ 1111 \\ 0010 + C : 0010\ \ 1001\ \ 0111\ \ 1100 \\ 1000 + C : 1000\ \ 0011\ \ 1101\ \ 0110 \end{array}$$

i) $w = 1101$: $w + C$ is the fourth coset. The word of least weight (coset leader) in this coset is 1000. Hence,$1101 - 1000 = 1101 + 1000 = 0101$ was the most liklihood codeword transmitted.

ii) W = 1111:w+c is the second coset. There are two words of smallest weight, $0001$and 0100, in this coset. This means that there are two choices for the coset leader. When the coset of the received word has more than one possible leader, the approach we take for decoding depends on the decoding scheme (i.e., incomplete or complete) used. If we are doing incomplete decoding, we ask for a retransmission. If we are doing complete decoding, we arbitrarily choose one of the words of smallest weight, say 0001,to be the error pattern and conclude that ,

1111-0001 = 1111+0001 = 1110 was a most likely codeword sent.

2.2.2 Syndrome decoding

Decoding schemes designed for specific codes are more efficient than decoding schemes that can be used for any codes .Syndrome decoding is a decoding scheme that can be used for any code but it is more efficient than the completing listing method .If we advise a decoding scheme for specific code or family of codes it is agood idea to compare it with a syndrome decoding.

The decoding scheme based on the standard array works reasonably well when the length n of the linear code is small, but it may take a considerate amount of time when n is large. Some time can be saved by making use of the syndrome to identify the coset to which the received word belongs.

Definition 2.5 : Let C be an [n,k,d] linear code over A and let H be a parity check matrix for C. For any w ∈ A_q^n.The syndrome of w is the word S(w) = wH^T ∈ A_q^n.

Remark 2.3: Strictly speaking, as the syndrome depends on the choice of the paritycheck matrix H, it is more appropriate to denote the syndrome of w by S_H(w) to emphasize this dependence. However, for simplicity of notation , the suffix is dropped whenever there is no risk of ambiguity.

Proposition 2.2: *Let C be an* [n,k,d] *linear code and let* H *be a parity-check matrix for C. For* u,v ∈ A_q^n

1) S(u+v) = s(u) + s(v)
2) S(u) = 0 if and only if u is a code word in C .
3) S(u) = S(v) if and only if u and v are in the same coset.

Remark 2.4 :

i) Properties (1) and (3) says that we can identify a coset by its syndrome; conversely, all the words in a given coset yield the same syndrome. So the syndrome of a coset is the syndrome of any word in the coset. In other words, there is a one-to-one correspondence between the cosets and the syndrome.

ii) Since the syndromes are in A_q^{n-k} there are at most q^{n-k}syndromes.

Syndrome decoding scheme : In this scheme ,we choose a set of coset leaders of an [n,k] code C and list them with their syndromes.Since all the vectors in a coset have the same syndrome this list contains all possible q^{n-k} syndromes.The code itself has the zerovector as coset leader .We can then choose vectors of weight 1 as coset leaders . We computer their syndrom whenever we get a new syndrome ,we hve a new coset leader .When we we finish with vectors of weight 1 and there are more coset leaders,we go on to see if vectors weight 2 can be coset leaders .Thus each time we get a new coset leader of weigt i that gave rise to it.After we complete the vectors of weight i ,we continue with vectors of weight i+1 .Until we reach our q^{n-k} syndromes.To decode a recived vector w ,compute S(w),locate this in the syndrome list .Substract the coset leader u corresponding to this syndrome from w.Decode w as w-u = v.

Definition 2.6: A table which matches each coset leader with its syndrome is called a syndrome look-up table. Sometimes such a table is called a standard decoding array (SDA).

Example 2.4:

Coset leader u	syndrome S(u)
0000	00
0001	01
0010	10
1000	11

Table 2.1

Steps to construct a syndrome look-up table assuming complete nearest neighbor decoding :-

Step 1: List all the cosets for the code, choose from each coset a word of least weight as coset leader u.

Step 2: Find a parity-check matrix H for the code and for each coset leader u, calculate its syndrome

$$S(u) = uH^T.$$

Remark 2.5: For incomplete nearest neighbor decoding, if we find more than one word of the smallest weight in *Step 1* of the above procedure, place the symbol '*' in that entry of the syndrome look-up table to indicate that retransmission is required.

Decoding procedure for syndrome decoding :-

Step 1: For the received word *w*, compute the syndrome *S(w).*

Step 2: Find the coset leader u next to the syndrome $S(w) = S(u)$ in the syndrome look-up table.

Step 3: Decode w as $v = w - u$.

Example 2.5: Let *q* = *2* and let $C = \{0000{,}1011{,}0101{,}1110\}$. Construct a syndrome look-up table and decode (i) $w = 1101$; $(ii)\ w = 1111.$

From the cosets computed in Example 2.4, we choose the words $0000, 0001, 0010$ and 1000 as coset leaders. Next, a parity-check matrix for *C* is

$$H = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{pmatrix}$$

Now we construct a syndrome look-up table for C

Coset leader u	syndrome S(u)
0000	00
0001	01
0010	10
1000	11

Table 2.2

$(i)\ w = 1101$. The syndrome is $S(w) = wH^T = 11$. From Table 2.2 we see that the coset leader is 1000. Hence was $1101 + 1000 = 0101$ most likely codeword sent.

$(ii)\ w = 1101$ .The syndrome is $S(w) = wH^T = 01$. From Table 2.2 we see that the coset leader is 0001. Hence 1111 + 0001 = 1110 was a most likely codeword sent.

Theorem 2.2 : Syndrome decoding is a maximum likelihood decoding scheme .

Proof : In syndrome decoding scheme we choose a set of coset leaders of an [n,k] code C and list them with their syndromes .Since all the vectors in a coset have the same syndrome there are altogether q^{n-k} possible syndromes.The code contains the zero vector .We can use the zero vector as the coset leader and find its syndrome.We can then choose vectors of weight one as coset leader.We compute their syndromes. Whenever we get a new syndrome we have a new coset leader .When we finish the coset leader of weight one,we go onto see if vectors of weigt two can be cosetleaders.If there are coset leader of weight two we compute their syndrome. We continue this process until we reach our q^{n-k} syndrome. To decode a recived vector w ,compute S(w),locate this in the syndrome list .Substract the coset leader u corresponding to this syndrome from w.Decode w as w-u = v.Thus if w is received we decode w to v for some v code with w=u+v,where u is the vector of smallest weight.So it follows that syndrome decoding is a maximum likelihood decoding scheme.

Advantages of syndrome decoding:

Consider a binary [100,60] code .For syndrome decoding we store 2^{40} cosetleaders and their syndromes .This is quite a saving of 2^{60} items .It is also easier to search through 2^{40} syndromes rather than 2^{60} codewords .If we need a complete decoding algorithm we can decode each received vector as described .If our code has minimum weight 'd' and t = [$\frac{d-1}{2}$] , an alternative incomplete decoding scheme would be to decode all vectors whose coset leaders have weight t or less and detect otherwise.

3.SOME IMPORTANT TYPES OF BINARY CODES

3.0 Introduction

There are different types of error correcting codes. While creating and comparing codes there are many different aspect which have to be considered before deciding a good code. Given a q-ary (n,M,d) code, where n is fixed, the size M is a measure of the efficiency of the code, and the distance d is an indication of its error-correcting capability. It would be nice if both M and d could be as large as possible, but, as we shall see shortly in this chapter, this is not quite possible, and a compromise needs to be struck.

3.1 The main coding problem

Consider the q-ary code $C = F_q^n$. It is easy to see that

$$(n, M, d) = (n, q^n, 1).$$

Hence,

$$R(C) = (log_q(q^n))/n = 1$$

$$\delta(C) = 0.$$

This code has the maximum possible information rate, while its relative minimum distance is 0. As the minimum distance of a code is related closely to its error-correcting capability, a low relative minimum distance implies a relatively low error-correcting capability. Thus, it is clear that for an efficient code, a comprise has to be made between information rate and relative minimum distance.

Definition 3.1: For a given code alphabet A of size q (with $q > 1$) and given values of n and d, let $A_q(n,d)$ denote the largest possible size M for which there exists an (n,M,d) code over A. Thus,

$$A_q(n, d) = max\{M : there\ exists\ an\ (n, M, d)\ code\ over\ A\}.$$

(n,M,d) code C that has the maximum size, that is, for which $M = A_q(n, d)$ is called an optimal code.

Remark 3.1: (i) Note that $A_q(n, d)$ depends only on the size of A, n and d . It is independent of A.

(ii) The numbers $A_q(n, d)$ play a central role in coding theory, and much effort has been made in determining their values. In fact, the problem of determining the values of $A_q(n, d)$ is sometimes known as the *main coding theory problem.*

Instead of considering all codes, we may restrict ourselves to linear codes and obtain the following definition:

Definition 3.2: For a given prime power q and given values of n and d, let $B_q(n, d)$ denote the largest possible size q^k for which there exists an $[n, k, d]$ code over F_q . Thus,

$$B_q(n, d) = max\{q^k : there\ exists\ an\ [n, k, d] code\ over\ F_q \}.$$

3.2 Bounds in Coding Theory

We discuss here two well known lower bounds: the sphere-covering bound (for $A_q(n, d)$) and the Gilbert–Varshamov bound (for $B_q(n, d)$) and an upper bound called Hamming bound for codes.

Definition 3.3: Let A be an alphabet of size q, where $q > 1$. For any vector $u \in A_q^n$ and any integer $r \geq 0$, the *sphere* of radius r and centre u, denoted $S_A(u, r)$, is the set

$$\{v \in A_q^n : d(u, v) \leq r \}$$

Definition 3.4: For a given integer $q > 1$, a positive integer n and an integer $r \geq 0$, define V_q^n(r) to be

$$V_q^n(r) = \begin{cases} \binom{n}{0} + \binom{n}{1}(q-1) + \binom{n}{2}(q-1)^2 + \cdots + \binom{n}{r}(q-1)^r & if\ 0 \leq r \leq n \\ q^n & if\ n \leq r \end{cases}$$

Theorem 3.1: *For all integers $r \geq 0$, a sphere of radius r inA_q^n contains exactlyV_q^n(r) vectors, where A is an alphabet of size $q > 1$.*

Proof. Fix a $u \in A_q^n$ vector . We determine the number of vectors v$\in A_q^n$ such that $d(u, v) = m$; i.e., the number of vectors in A_q^n of distance exactly m from u. The number of ways in which to choose the m coordinates where v differs from u is given by $\binom{n}{m}$.

For each coordinate, we have $q - 1$ choices for that coordinate in v. Therefore, the total number of vectors of distance m from u is given by ,

$\binom{n}{m}(q - 1)^m$ For $0 \leq r \leq n$, The theorem now follows

When $r \geq n, S_A(u, r) = A_q^n$, hence it contains $V_q^n\ (r)$ vectors.

Definition 3.5: For an integer $q > 1$ and integers n, d such that $1 \leq d \leq n$, we have

$$\frac{q^n}{\sum_{i=0}^{d-1}\binom{n}{i}(q-1)^i} \leq A_q(n, d)$$

This is called *sphere-covering bound* for a linear code.

Definition 3.6: Let n, k and d be integers satisfying $2 \leq d \leq n$ and $1 \leq k \leq n$. If

$$\sum_{i=0}^{d-2}\binom{n-1}{i}(q-1)^i < q^{n-k}$$

then there exists an $[n, k]$ linear code over F_q with minimum distance at least d. This is called the *Gilbert-Varshamov* bound for linear code.

Remark 3.2: The Gilbert–Varshamov bound, a lower bound for $B_q(n, d)$ (i.e., for linear codes) is known since the 1950s. There is also an asymptotic version of the Gilbert

Varshamov bound, which concerns infinite sequences of codes whose lengths tend to infinity. For a long time, the asymptotic Gilbert–Varshamov bound was the best lower bound known to be attainable by an infinite family of linearcodes, so it became a sort of benchmark for judging the ‘goodness’ of an infinite sequence of linear codes. Between 1977 and 1982, V. D. Goppa constructed algebraic-geometry codes using algebraic curves over finite fields with many rational points. A major breakthrough in coding theory was achieved shortly after these discoveries, when it was shown that there are sequences of algebraic-geometry codes that perform better than the asymptotic Gilbert– Varshamov bound for certain sufficiently large q.

An upper bound for $A_q(n, d)$ that we are going to discuss is the Hamming bound, also known as the sphere-packing bound.

Definition 3.7: For an integer $q > 1$ and integers n, d such that $1 \leq d \leq n$, we have

$$A_q(n,d) \leq \frac{q^n}{\sum_{i=0}^{\lfloor (d-1)/2 \rfloor} \binom{n}{i} (q-1)^i}$$

This is called hamming bound or sphere packing bound.

Definition 3.8: A q-ary code that attains the Hamming (or sphere-packing) bound, i.e., one which has $\frac{q^n}{\sum_{i=0}^{\lfloor (d-1)/2 \rfloor} \binom{n}{i} (q-1)^i}$ codewords, is called a *perfect code*.

Remark 3.3: The following codes are obviously perfect codes and are called *trivial perfect codes*:

(i) The $C = A_q^n (d = 1)$ linear code

(ii) any C $|C| = 1$ $(d = \infty)$ with

(iii)binary repetition codes of odd lengths consisting of two codewords at distance $n$from each other $(d = n)$

The Hamming codes and the Golay codes are examples of nontrivial perfect codes. Various constructions of nonlinear perfect codes with the same parameters as the *q*-ary Hamming codes have also been found.

3.3 Hamming codes

Hamming codes were discovered by R. W. Hamming and M. J. E. Golay. They form an important class of codes – they have interesting properties and are easy to encode and decode. While Hamming codes are defined over all finite fields F_q, we begin by discussing specifically the binary Hamming codes. These codes form a special case of the general *q*-ary Hamming codes, but because they can be described more simply than the general *q*-ary Hamming codes, and because they are arguably the most interesting Hamming codes, it is worthwhile discussing them separately from the other Hamming codes.

Definition 3.8: Let $r \geq 2$. A binary linear code of length $n = 2^r - 1$, with paritycheck matrix H whose columns consist of all the non zero vectors of A_2^r, is called a *binary Hamming code* of length $2^r - 1$. It is denoted by $Ham(r, 2)$. The dual of the binary

Hamming code $Ham(r, 2)$ is called a binary *simplex code*. It is sometimes denoted by $S(r, 2)$.

Remark 3.4: (i) The order of the columns of H has not been fixed in definition. Hence, for each r ≥ 2, the binary Hamming code $Ham(r, 2)$ is only well defined up to equivalence of codes.

(iii) Note that the rows of H are linearly independent since H contains all the r columns of weight 1 words. Hence, H is indeed a parity-check matrix.

Proposition 3.1: (Properties of the Binary Hamming Codes)

(i)All the binary Hamming codes of a given length are equivalent.

(ii) The dimension of $Ham(r, 2)$ is $k = 2^r - 1 - r$.

(iii)The distance of $Ham(r, 2)$ is $d = 3$, hence $Ham(r, 2)$ is exactly single errorcorrecting.

(iv) Binary Hamming codes are perfect codes.

Since $Ham(r, 2)$ is perfect single-error-correcting, the coset leaders are precisely the 2^r vectors of length n of weight ≤ 1. Let e_j denote the vector with 1 in the jth coordinate and 0 elsewhere. Then the syndrome of e_j is just e_jH^T, i.e., the transpose of the jth column of H. Hence, if the columns of H are arranged in the order of increasing binary number, the decoding is given by:

Step 1: When w is received, calculate its syndrome $S(w) = wH^T$.

Step 2: If $S(w) = 0$, assume w was the codeword sent.

Step 3: If $S(w) \neq 0$, then $S(w)$ is the binary representation of j, for some $1 \leq j \leq 2^r - 1$. Assuming a single error, the word e_j gives the error, so we take the sent word to be $w - e_j$ (or, equivalently, $w + e_j$).

Example 3.1: $Ham(3, 2)$ is a Hamming code of length 7 with a parity-check matrix

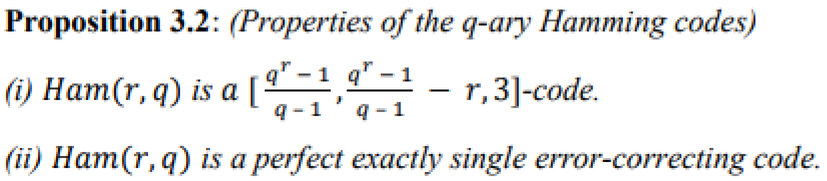
$$H = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}$$

Definition 3.9: Let $r \geq 2$. A q-ary linear code, whose parity-check matrix H has the property that the columns of H are made up of precisely one nonzero vector from each vector subspace of dimension 1 of $A^r{}_q$, is called a *q-ary Hamming code*, often denoted as $Ham(r, q)$. The dual of the q-ary Hamming code $Ham(r, q)$ is called a q-ary *simplex code*. It is sometimes denoted by $S(r, q)$.

Remark 3.5: (i) when $q = 2$, the code defined here is the same as the Binary Hamming Code (ii) An easy way to write down a parity-check matrix for $Ham(r, q)$ is to list as columns all the nonzero r -tuples in A_q^r whose first nonzero entry is 1.

Example 3.2: Ham (2,3) is an example of 3-ary Hamming code with parity matrix

$$H = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 2 \end{pmatrix}$$



Proposition 3.2: *(Properties of the q-ary Hamming codes)*

(i) $Ham(r, q)$ *is a* $[\frac{q^r-1}{q-1}, \frac{q^r-1}{q-1} - r, 3]$*-code.*

(ii) $Ham(r, q)$ *is a perfect exactly single error-correcting code.*

3.4 Golay codes

The Golay codes were discovered by M. J. E. Golay in the late 1940s. The (unextended) Golay codes are examples of perfect codes. It turns out that the Golay codes are essentially unique in the sense that binary or ternary codes with the same parameters as them can be shown to be equivalent to them.

Definition 3.10: Let G be the 12×24 matrix $G = (I_{12}|A)$, where is I_{12} the 12×12 identity matrix and A is the 12×12 matrix

$$A = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{pmatrix}$$

The binary linear code with generator matrix G is called the extended binary Golay code and will be denoted by G_{24}.

Proposition 3.3: (Properties of the extended binary Golay codes)

(i) The length of G_{24} is 24 and its dimension is 12.

(ii) A parity-check matrix for G_{24} is the 12×24 matrix

$$H = (A|I_{12})$$

(iii) The code G_{24} is self-dual, i.e., $G^{\perp} = G_{24}$

(iv) Another parity-check matrix for G_{24} is the 12 ×24 matrix

$$H' = (I_{12}|A) = G$$

(v) Another generator matrix for G_{24} is the 12 ×24 matrix

$$G' = (A|I_{12}) = H$$

(vi) The weight of every codeword in G_{24} is a multiple of 4.

(vii) The code G_{24} has no codeword of weight 4, so the distance of G_{24} is

$$d = 8.$$

(viii) The code G_{24} is an exactly three error-correcting code.

Definition 3.11: Let $\widehat{G}$ be the 12×23 matrix

$$\hat{G} = (I_{12}|\hat{A}),$$

Where I_{12} is the 12×12 identity matrix and $\widehat{A}$ is the 12×11 matrix obtained from the matrix A by deleting the last column of A. The binary linear code with generator matrix $\widehat{G}$ is called the *binary Golay code* and will be denoted by G_{23}.

Remark 3.6: Alternatively, the binary Golay code can be defined as the code obtained from G_{24} by deleting the last coordinate of every codeword.

Proposition 3.3:

(i) The length of G_{23} *is 23 and its dimension is 12.*

(ii)A parity-check matrix for G_{23} *is the* 11×23 *matrix*

$$\widehat{H} = \left(\hat{A}^T \middle| I_{11}\right).$$

(iii) The extended code of G_{23} *is* G_{24}.

(iv) The distance of G_{23} *is* $d = 7$.

(v) The code G_{23} *is a perfect exactly three-error-correcting code.*

Definition 3.12: The *extended ternary Golay code*, denoted by G_{12}, is the ternary linear code with generator matrix $G = (I_6|B)$, where B is the 6×6 matrix.

$$B = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 2 & 2 & 1 \\ 1 & 1 & 0 & 1 & 2 & 2 \\ 1 & 2 & 1 & 0 & 1 & 2 \\ 1 & 2 & 2 & 1 & 0 & 1 \\ 1 & 1 & 2 & 2 & 1 & 0 \end{pmatrix}$$

Remark 3.6: Any linear code that is equivalent to the above code is also called an extended ternary Golay code.

As already mentioned above Hamming codes and Golay codes form a very important class of binary codes and they have many applications. A detailed study on perfect codes led to the following theorem, proof of which requires an advanced knowledge in coding theory.

Theorem 3.1: ($Van\ Lint\ and\ Tietäväinen.$)

When $q \geq 2$ is a prime power ,a nontrivial perfect code over Fq must have the same parameters as one of the Hamming or Golay codes.

3.5 Applications of linear codes

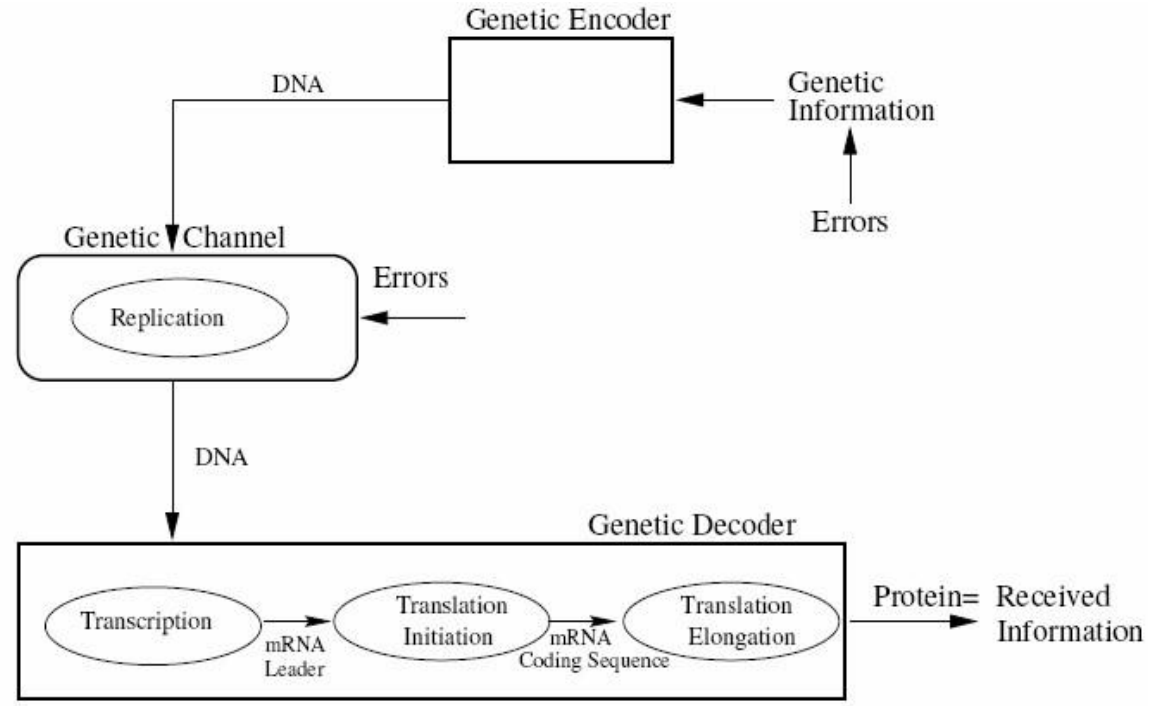
There are many applications of coding theory in the modern world. In computer science, where coding theory originated, powerful error detection and correction codes are used in the transmission of digital data.

3.6.1 Hamming codes and DRAM

Traditional DRAM (dynamic random-access memory) uses Hamming codes for error correction purposes. However, hamming codes have a minimum distance $d = 3$ which enables them to correct only one bad bit per codeword. As computers have progressed from 8-bit machines to 16-bits, 32-bits or even 64-bits, the ability to correct only a single bit error introduces the increasing possibility of data corruption. In the presence of ever-increasing data throughput even extended Hamming codes (obtained from $Ham\ (2,3)$ by adding a parity check coordinate and having $d = 4$ seem to fall short of the required error correction.

3.6.2 Coding Theory and Genetic Research

There are many new frontiers of science that coding theory is finding applications in. One such application which arouses great excitement is the use of coding theory in the study of evolution and genetic mutations. Genetic information in the form of DNA is taken as input, transmitted via the process of replication and then ultimately output as amino acid proteins. Errors are introduced by fluxuations in heat, radioactivity and other factors. One model used in an attempt to validate the processes was a (n, k) block code which outputted a parity check code. Based on the known genetic bases, codewords of length $n = 5$ and $n = 8$ were developed and evaluated based on a minimum distance (nearest neighbour) decoding scheme. The following figure describes the proposed structure of the encoding/decoding process:



(Fig 3.1)

3.6.3 Photographs from spacecrafts

The codes initially used for transmitting photographs from spacecrafts were first order Reed Muller codes, which can be constructed as orthogonal extended Hamming codes. But later Golay codes was used. Hundreds of colour pictures of Jupiter and Saturn in the 1979, 1980, 1981 fly-bys of Voyager 1 and 2 spacecraft would be transmitted within a constrained telecommunication bandwidth. Colour image transmission requires three times the amount of data as black and white images and Golay (24,12,8) code was used for this purpose. This Golay code is only triple-error correcting, but it could be transmitted at a much higher rate than Hadamard code (non-linear) that was used during Mariner mission.

Above mentioned are some of the thousands of applications of linear ECC. Apart from linear codes, there are several families of nonlinear codes that are well known and important in coding theory. Non-linear codes are used to obtain the largest possible number of codewords with a given minimum distance.

4.CONCLUSION

Coding Theory is concerned with successfully transmitting data through a noisy channel and correcting errors in corrupted messages. This project is giving a brief introduction to linear Error Correcting Codes (ECC) whilst only assuming basic linear algebra. It contains a rigorous introduction to the theory of block codes. In this study, different performance measurement parameters which makes one class of ECC different from other classes is explained. Various encoding and decoding techniques which helps in the efficient use of linear codes and perfect codes are also discussed. Giving more focus to Binary Hamming Codes and Golay Codes, which constitute a very important class of linear ECC, some of the major applications of linear ECC are also discussed in this project. In essence, from this project we can understand that linear Error Correcting Codes play a vital role in controlling and correcting errors which are caused due to different noises in the channel.

Error control coding applications have grown rapidly in the past several years in various field of communication and information storage mechanism. There are number of techniques of error correction based on applied mathematics which correct various types of errors. These codes have some limitations in mathematical or practical considerations or in other ways. It is impossible to correct all the errors but these errors can be minimized. Still no error correcting code is available which can correct all the random errors and burst errors. When number of errors was increased designed codes turn out to be inefficient. Small error correction codes with desired correction capabilities can be easily developed but with large error correction capability; developing a code is real practical problem. Now future work can be done on such error correcting code which would be capable to correct the errors with high probability.

BIBILOGRAPHY

- Vera Pless, *Introduction to the Theory of Error-Correcting Codes*, A WileyInterscience Publication, Third Edition.
- San Ling and Chaoping Xing, *Coding Theory – A First Course*, Cambridge.
- John Baylis, *Error Correcting Codes – A Mathematical Introduction.*
- Todd K. Moon, *Error Correction Coding: Mathematical Methods and Algorithms.*
- W. Wesley Peterson and E. J. Weldon, *Error Correcting Codes*, Second Edition.
- F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error – Correcting Codes.*