Introduction

In our day life we are not very much concerned with accurate transmission of information. In general conversations, lectures, radio or telephone communication many words or sentences may be missed still not distorting the meaning of the message. However when we are to transmit intelligence- more information in a shorter time, we wish to eliminate unnecessary redundancy. Our language becomes less redundant and errors in transmission become more serious. While we are talking about numerical data, misreading of even a single digit could have a marked effect on the intent of the message. Thus the primary objective of coding for transmission of intelligence would be two fold – Increase the efficiency and reduce the transmission errors. Also, we would like our technique to ensure security and reliability. Transmission of classical information in time and space is nowadays very easy. It took centuries and ingenious developments and discoveries and the idea of the digitalization of all forms of information to discover fully this property of information. In 1948 Claude Shannon published a mathematical theory of communication, an article in two parts in the July and October issues of the bell system technical journal. This work focused on the problem of how best to encode the information, a sender wants to transmit. In this fundamental work he used tools in probability theory, developed by Norbert Wener, which were in their nascent stages of being

applied to communication theory at that time. Shannon developed information entropy, a measure for the uncertainty in a message while essentially inventing the field of information theory. Basically the main areas the coding associated with are data compression, error correction codes. Coding theory develops method to protect information against a noise. Coding also ensures the security in our communication. In this project we present some techniques for source encoding and connection between coding and information theory in the light of Shannon's investigations.

Chapter 1

CODING THEORY

1.1 DEFINITION OF CODES

Encoding' or 'Enciphering' is a procedure for associating words constructed from a finite alphabet of a language with given words of another language in a one-to-one manner. Let the source characterized by the set of symbols

$$S = s_1, s_2, \ldots, s_q$$

We shall call 'S' as the "source alphabet". Consider another set, X, comprising of 'r' symbols.

$$X = (x_1, x_2, \ldots\ldots, x_r)$$

We shall call 'X" as the "code alphabet".

- Coding can be defined as the mapping of all possible sequences of symbols of S into symbols of X.

- Any finite sequence of symbols from an alphabet will be called a "word".
- Any sequence from the alphabet 'X' forms a "code word".
- The total number of symbols contained in the 'word' will be called a "word length".

Example 1.1.1. x_1; $x_1x_3x_4$; $x_3x_5x_7x_9$; $x_1x_1x_2x_2x_2$ form code words. Their word lengths are respectively 1; 3; 4; and 5.

1.2 BASIC PROPERTIES OF CODES

1.2.1 Block codes

A block code is one in which a particular message of the source is always encoded in the same "fixed sequence" of the code symbols. The code can be 'fixed length code' or a variable length code.

Example 1.2.1.

$$S = (s_1, s_2, s_3, s_4)\, X = (0, 1)\ Codes, X_1 = (0, 11, 10, 11)$$

1.2.2 Non singular codes

A block code is said to be non singular code if all the words of the code set X_1, are "distinct".

Example 1.2.2.

$$S = (s_1, s_2, s_3, s_4),\ X = (0,1)\ \ Codes, X_1 = (0, 00, 10, 11)$$

1.2.3 Uniquely decodable codes

A non-singular code is uniquely decipherable if every word immersed in a sequence of words can be uniquely identified. [2]

Example 1.2.3. Second extension of the codes

$$S = (s_1, s_2, s_3, s_4),\ X = (0,1)\ \ Codes, X_1 = (0, 00, 10, 11)$$

$$S^2 = (s_1s_1,\ s_1s_2,\ s_1s_3,\ s_1s_4,\ s_2s_1,\ s_2s_2,\ s_2s_3,\ s_2s_4,\ s_3s_1,\ s_3s_2,\ s_3s_3,\ s_3s_4,\ s_4s_1,\ s_4s_2,\ s_4s_3,\ s_4s_4)$$

Source symbols	Codes	Source symbols	codes	Source symbols	Codes	Source symbols	Codes
s_1s_1	00	s_2s_1	000	s_3s_1	100	s_4s_1	110
s_1s_2	000	s_2s_2	0000	s_3s_2	1000	s_4s_2	1100
s_1s_3	010	s_2s_3	0010	s_3s_3	1010	s_4s_3	1110
s_1s_4	011	s_2s_4	0011	s_3s_4	1011	s_4s_4	1111

The codes of the source sequences $s_1s_2$and $s_2s_1$are not distinct and hence the code is “singular in the large”. Since such singularity properties introduce ambiguity in the decoding stage, we therefore require, in general, for unique decodability of our codes that “The nth extension of the code be non singular for every finite n”

1.2.4 Instantaneous or Irreducible codes

A uniquely decodable code is said to be "Instantaneous" if the end of any code word is recognizable without the need of inspection of succeeding code symbols.

Source symbols	Code A	Code B	Code C
s_1	00	0	0
s_2	01	10	01
s_3	10	110	011
s_4	11	1110	0111

Example 1.2.4. Code A is the simplest possible uniquely decipherable code. It is non singular and all the code words have the same length. The decoding can be done as soon as we receive two code symbols without any need to receive succeeding code symbols.

Code B is also uniquely decodable with a special feature that the '0' s indicate the termination of a code word. It is called the "comma code". When scanning a sequence of code symbols, we may use the comma to determine the end of a code word and the beginning of the other. Accordingly, notice that the codes can be decoded as and when they are received and there is, once again, no time lag in the decoding process.

Whereas code 'c' is also non singular and uniquely decodable but cannot be decoded word by word as it is received. For example, if we receive '01', we cannot decode it as s_2 until we receive the next code symbol. If the next code symbol is '0' indeed, the previous words corresponds to s_2, while if it is a '1' it may be the symbol s_3, which can be concluded so only if we receive a '0' in the fourth place. Thus, there is a definite 'time lag' before a word can be decoded. Such a 'time waste' is not there if use either code A or code B.

Prefix property- "no encoded word can be obtained from each other by the addition of more letters". This property is called "prefix property" or "Irreducibility".

"A necessary and sufficient condition for a code to be 'instantaneous' is that no complete code word be prefix of some other code word".

1.2.5 Optimal codes

An instantaneous code is said to be optimal if it has 'minimum average length', for a source with given probability assignment for the source symbol.

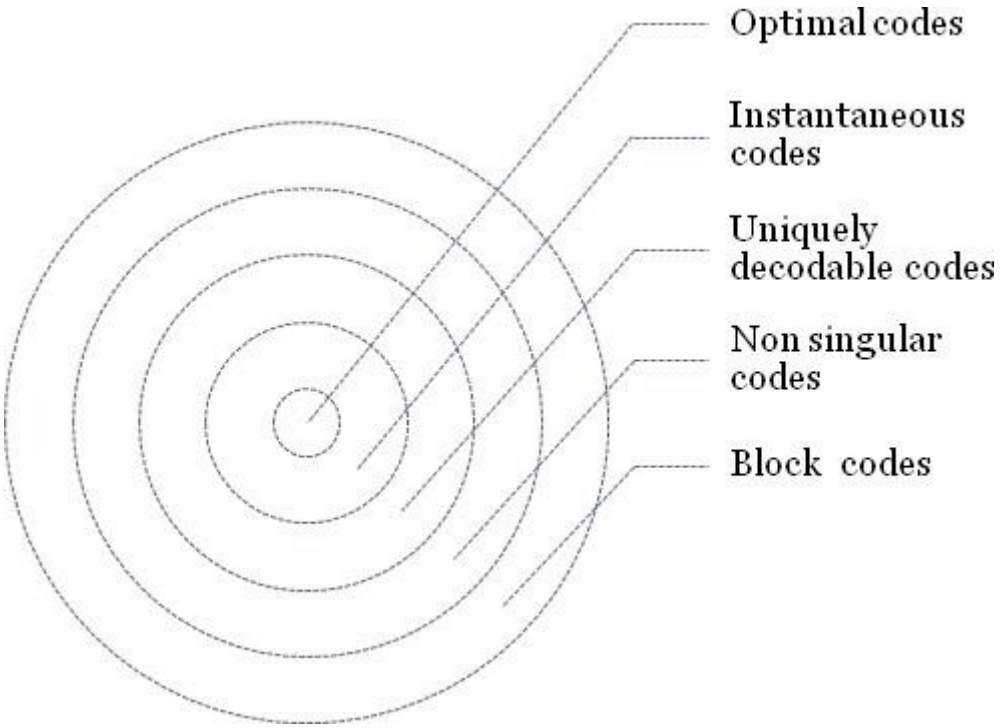

Figure 1.1: Figure 1

Chapter 2

BLOCK CODES

2.1 Defnition of block codes

Error-correcting codes are used to reliably transmit digital data over unreliable communication channels subject to channel noise. When a sender wants to transmit a possibly very long data stream using a block code, the sender breaks the stream up into pieces of some fixed size. Each such piece is called message and the procedure given by the block code encodes each message individually into a codeword, also called a block in the context of block codes. The sender then transmits all blocks to the receiver, who can in turn use some decoding mechanism to (hopefully) recover the original messages from the possibly corrupted received blocks. The performance and success of the overall transmission depends on the parameters of the channel and the block code. [4]

Formally, a block code is an injective mapping

$$C : \Sigma^k \to \Sigma^n$$

. Here, Σ is a finite and nonempty set and k and n are integers. The meaning and significance of these three parameters and other parameters related to the code are described below.

The alphabet Σ

The data stream to be encoded is modeled as a string over some alphabet Σ . The size $|\Sigma|$ of the alphabet is often written as q. If $q = 2$, then the block code is called a binary block code. In many applications it is useful to consider q to be a prime power, and to identify Σ with the finite field F_q.

The message length k

Messages are elements m of Σ^k, that is, strings of length k. Hence the number k is called the message length or dimension of a block code.

The block length n

The block length n of a block code is the number of symbols in a block. Hence, the elements c of Σ^n are strings of length n and correspond to blocks that may be received by the receiver. Hence they are also called received words. If $c = C(m)$ for some message m, then c is called the codeword of m.

The rate R

The rate of a block code is defined as the ratio between its message length and its block length:

$R = k/n$.

A large rate means that the amount of actual message per transmitted block is high.

In this sense, the rate measures the transmission speed and the quantity $1 - R$ measures the overhead that occurs due to the encoding with the block code. It is a simple information theoretical fact that the rate cannot exceed 1 since data cannot in general be losslessly compressed. Formally, this follows from the fact that the code C is an injective map.

The distance d

The distance or minimum distance d of a block code is the minimum number of positions in which any two distinct codewords differ, and the relative distance δ is the fraction d/n. Formally, for received words $c_1, c_2 \in \Sigma^n$, let $\Delta(c_1, c_2)$ denote the Hamming distance between c_1 and c_2, that is, the number of positions in which c_1 and c_2 differ. Then the minimum distance d of the code C is defined as

$$d := \min_{\substack{m_1, m_2 \in \Sigma^k \\ m_1 \neq m_2}} \Delta[C(m_1), C(m_2)]$$

. Since any code has to be injective, any two codewords will disagree in at least one position, so the distance of any code is at least 1. Besides, the distance equals the minimum weight for linear block codes because:

$$\min_{\substack{m_1, m_2 \in \Sigma^k \\ m_1 \neq m_2}} \Delta[C(m_1), C(m_2)] = \min_{\substack{m_1, m_2 \in \Sigma^k \\ m_1 \neq m_2}} \Delta[\mathbf{0}, C(m_1) + C(m_2)] = \min_{\substack{m \in \Sigma^k \\ m \neq \mathbf{0}}} w[C(m)] = w_{\min}$$

A larger distance allows for more error correction and detection. For example, if we only consider errors that may change symbols of the sent codeword but never erase or add them, then the number of errors is the number of positions in which the sent codeword and the received word differ. A code with distance d allows the

receiver to detect up to $d-1$ transmission errors since changing $d-1$ positions of a codeword can never accidentally yield another codeword. Furthermore, if no more than $(d-1)/2$ transmission errors occur, the receiver can uniquely decode the received word to a codeword. This is because every received word has at most one codeword at distance $(d-1)/2$. If more than $(d-1)/2$ transmission errors occur, the receiver cannot uniquely decode the received word in general as there might be several possible codewords. One way for the receiver to cope with this situation is to use list decoding, in which the decoder outputs a list of all codewords in a certain radius.

Popular notation

The notation $(n, k, d)_q$ describes a block code over an alphabet Σ of size q, with a block length n, message length k, and distance d . If the block code is a linear block code, then the square brackets in the notation $[n, k, d]_q$ are used to represent that fact. For binary codes with $q = 2$, the index is sometimes dropped. For maximum distance separable codes, the distance is always $d = n - k + 1$, but sometimes the precise distance is not known, non-trivial to prove or state, or not needed. In such cases, the d-component may be missing.

Sometimes, especially for non-block codes, the notation $(n, M , d)_q$ is used for codes that contain M codewords of length n. For block codes with messages of length k over an alphabet of size q, this number would be $M = q^k$.

2.2 Definition and Simple Properties of Linear Codes

Definition 2.2.1. An $(n,k)_q$ code C over F_q is called a linear code, if the sum of two codewords is another codeword:

$$a,b \in C \Rightarrow a+b \in C$$

. In addition, non-binary codes with $q > 2$ must satisfy

$$a \in C, \alpha \in \mathsf{F}_q \Rightarrow \alpha \cdot a \in C$$

. In other words, C is to be a vector space.

Consider some simple examples ($q = 2$):

$C = \{000, 100, 010, 001\}$ is non-linear,

$C = \{000, 110, 011, 111\}$ is non-linear,

$C = \{000, 110, 101, 011\}$ is linear.

Example 2.2.1. (1) The $(n,1)_2$ code

$$C = \{00...0, 11...1\} \tag{2.1}$$

is called a repetition code. The linearity is apparent. The code rate is $R = 1/n$ and the minimum distance is $d_{min} = n$. Systematic encoding can use $u_0 \mapsto a = (u_0, ..., u_0)$.

(2) The $(n, n-1)_2$ code

$$C = \{(a_0, ..., a_{n-1}) | \sum_{i=0}^{n-1} a_i = 0\} \tag{2.2}$$

is called a parity-check code (or single parity-check code, SPCC). This code is linear with $R = (n-1)/n = 1 - 1/n$. Since 000...0 and 110...0 are codewords, $d_{min} = 2$. When systematically encoding with $(u_0, ..., u_{n-2}) \mapsto a = (u_0, ..., u_{n-2}, u_0 + \cdots + u_{n-2})$, the sum of the information bits is attached as a parity-check bit.

Theorem 2.2.1. A linear code C is invariant under additive shifts, i.e., $C + b = \{a + b | a \in C\} = C$ for all $b \in C$.

Due to $d_H(a, b) = w_H(a - b)$ the minimum distance of a code is equal to the minimum weight of the codewords, thus for determining d_min, only $q^k - 1$ words need to be considered instead of $q^k(q^k - 1)$ pairs. This implies:

Theorem 2.2.2. For a linear $(n, k, d_{min})_q$ code C the minimum Hamming distance is equal to the minimum Hamming weight:

$$d_{min} = min\{d_H(a, b) | a, b \in C, a \neq b\} \tag{2.3}$$

$$= min\{w_H(a) | a \in C, a \neq 0\}. \tag{2.4}$$

Chapter 3

CONSTRUCTION OF INSTANTANEOUS CODES

Consider encoding of five symbol source into binary instantaneous codes. i.e

$$S = \{s_1, s_2, s_3, s_4, s_5\}, X = \{0, 1\}$$

We may start by assigning

$$S_1 \rightarrow 0$$

Then by prefix property all other source symbol must correspond to code words beginning with 1.

Then we might have,

$$S_2 \rightarrow 10$$

This in turn would require the remaining code words to start with 11. If

$$S_3 \rightarrow 110$$

Then the only 3 binit prefix unused is 111 and we might set

$$S_4 \rightarrow 1110$$

$$S_5 \rightarrow 1111$$

The other possible instantaneous code is

$$S_1 \rightarrow 00$$

$$S_2 \rightarrow 01$$

$$S_3 \rightarrow 10$$

$$S_4 \rightarrow 110$$

$$S_5 \rightarrow 111$$

3.1 Kraft Inequality:

Theorem 3.1.1 (Kraft Inequality)**.** Given a source $S = \{s_1, s_2, ..., s_q\}$. Let the word length of code corresponding to these symbols be $l_1, l_2, ..., l_q$ and the code alphabet be $X = \{x_1, x_2, , x_r\}$. Then an instantaneous code for the source exists if and only if

$$\sum_{k=1}^{q} r^{-l_k} \leq 1 \tag{3.1}$$

This is called Kraft inequality. [1]

Proof. Let us assume that the word lengths have been arranged in the ascending order that is

$$l_1 \leq l_2 \leq \cdots \leq l_q$$

Since our code alphabet has only $'r'$ symbols. We can have at most $'r'$ instantaneously decodable sequences of length one so as to satisfy the prefix property. Let n_k denote the actual number of messages encoded into code words of length k. then it follows

$$n_1 \leq r$$

The number of actual instantaneous codes of word length 2 must obey the rule.

$$n_2 \leq (r - n_1) \cdot r = r^2 - n_1 r$$

As the first symbol can be only $(r - n_1)$ symbols that are not used in forming the code words of length one and the second symbol of the sequence can be any of the r-code alphabet symbols. Similarly, the actual number of codes of length 3, that are distinguishable from each other and from the n_1 and n_2 words must obey

$$n_3 \leq [(r - n_1)r - n_2]r = r^3 - n_1 r^2 - n_2 r$$

As the first two symbols may be chosen in $[(r - n_1)r - n_2]$ ways and third element in r- ways. Following this way, we arrive at

$$n_k \leq r^k - n_1 r^{k-1} - n_2 r^{k-2} \cdot \cdots - n_{k-1} r \tag{3.2}$$

Multiplying (3.2) throughout by r^{-k} and re writing we obtain

$$n_k r^{-k} + n_{k-1} r^{-k-1} + n_{k-2} r^{-k-2} + \cdots\cdots + n_1 r^{-1} \leq 1$$

or

$$\sum_{j=1}^{k} n_j r^{-j} \leq 1 \tag{3.3}$$

Remembering that n_k is a 'a positive integer we may rewrite (3.3) as

$$\sum_{j=1} n_j r^{-j} = \frac{(r^{-1} + r^{-1} + \cdots + r^{-1})}{n_1 times} + \frac{(r^{-2} + r^{-2} + \cdots + r^{-2})}{n_2 times} + \cdots\cdots + \frac{(r^{-k} + r^{-k} + \cdots + r^{-k})}{n_k times}$$
$$= \sum_{i=1}^{n_1} r^{-1} + \sum_{i=1}^{n_2} r^{-2} + \cdots + \sum_{i=1}^{n_k} r^{-k}$$

Here each grouping corresponds to code cords of length l_j and accordingly the subscript of $'r'$ in the j^{th} grouping indeed is $-l_j$. Further since $n_1 + n_2 + \cdots + n_k = q$ it follows

$$\sum_{k=1}^{q} r^{-l_k} \leq 1$$

□

The inequality just tells whether the instantaneous code we are seeking will exist or not. It does not show how to construct the code nor does this guarantee that any code has word lengths satisfying this inequality would be automatically instantaneous

Example 3.1.1. A six symbol source is encoded into binary codes shown below. Which of these codes are instantaneous?