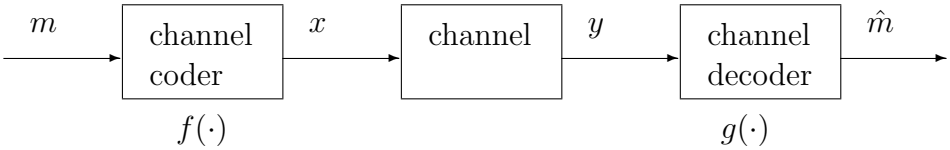
Lecture of January 14th 2004
Scribe: Michael Wood

1 Introduction

Our general system block diagram is:

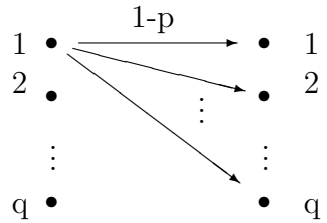


For a discrete memoryless channel (DMC):
$m \in \Omega = \{1, 2, ..., M\}$ (the message/information set)
$x \in X$ (the finite input alphabet for the channel)
$y \in Y$ (the finite output alphabet for the channel)
$\hat{m} \in \Omega' = \Omega \cup \{0\} = \{0, 1, 2, ..., M\}$ (the output alphabet)

Here 0 represents decoding failure, i.e. no decision was made. Also note that $\hat{m}$ is an estimate of m

In block oriented coding: $P(\underline{y}|\underline{x}) = \prod_{i=1}^{n} P(\underline{y_i}|\underline{x_i})$
where $P(\underline{y_i}|\underline{x_i})$ = probability of receiving y_i given x_i was transmitted.

Example: A q symmetric channel: $|X| = |Y| = q$



Transition Probabilities

$$P(y|x) = \begin{cases} 1-p, y = x \\ \frac{p}{1-q}, y \neq x \end{cases}$$

e.g.: $q = 2 \Rightarrow BSC$

Encoding:

The encoder $f(\cdot)$ implements a mapping from Ω to X ($\Omega \rightarrow X$). In a block-oriented scheme, the encoder maps a message from Ω to a vector $\underline{x}$ of fixed length n, i.e. $\underline{x}$ is a vector or block or n-tuple. Hence $\underline{x} \in X^n$ and, the range of $f(\cdot)$ = the image of $f(\cdot)$ under Ω $= \{f(1), f(2), ..., f(M)\} = C$, where C is the codebook or code, and members of C are codewords. $f(\cdot)$ is assumed to be one to one (i.e. an injection) then $|C| = |\Omega| = |M|$.

Example: $M = 4, \quad \Omega = \{00, 01, 10, 11\}$

$$R(C) = code\ rate = \frac{number\ of\ information\ bits}{number\ of\ times\ we\ use\ the\ channel} = \frac{2}{n}\ \frac{bits}{use}$$

In general, the rate of the codebook $R(C) = \frac{log_2 M}{n}\ \frac{bits}{use} = \frac{log_2|C|}{n}$. This is because of our one to one assumption. We have also assumed that maximum entropy is achieved. In general our attention will be on the codebook, not the structure of the encoder.

Decoding:

The decoder $g(\cdot)$ implements a mapping from Y^n to $\hat{m}$ ($Y^n \rightarrow \Omega' = \Omega \cup \{0\} = \{0, 1, 2, ..., M\}$). Remember that a 0 represents decoding failure.

Definition: Decoding failure is when the decoder cannot, or chooses not to make a decoding decision.

Example: An error detecting decoder outputs a zero when the received vector $\underline{y}$ is not a codeword. An error has occurred when $\hat{m} \neq m$. Hence we have the following figure of merit $P[\hat{m} \neq m] \downarrow$ which reads: we wish to minimize the probability that $\hat{m} \neq m$.

2 Block Codes

An alphabet is a mathematical term for a finite set. The cartesian product of two alphabets is as follows: $A \times B = \{(a, b) : a \in A,\ b \in B\}$. A^n is an n-fold cartesian product of A with itself, i.e. $A^3 = A \times A \times A$. Cartesian products of alphabets also have the following cardinality properties: $|A \times B| = |A| \cdot |B|$ and $|A^n| = |A|^n$.

A block code C of length n is a non-empty subset of A^n where A is a finite alphabet. $C \subseteq A^n, C \neq \emptyset$.

Examples:

$$C_1 = \{000, 111\} \subset \{0, 1\}^3$$

$$C_2 = \{000, 001, 101, 110\} \subset \{0, 1\}^3$$

$$C_3 = \{001, 100, 010\} \subset \{0, 1\}^3$$

Note: C_1 and C_2 are linear codes because they are closed, whereas C_3 is non-linear because it is open.

3 Error Detection

1. The simplest application of error control codes.

We transmit a codeword $\underline{c} \in C$ and check if the received value $\underline{r} \in C$. If $\underline{r} \notin C$ then we have detected an error. Note that $\underline{r} \in C$ does not imply that no error occurred. Clearly the probability of undetected error is as follows: $P[\underline{c} \in C \ \wedge \ \underline{r} \in C \ \wedge \ \underline{c} \neq \underline{r}]$ where $\underline{c}$ was transmitted and $\underline{r}$ was received. We like codes with small probability of undetectable error.

Definition: Hamming Distance, $d_H(\underline{x}, \underline{y})$ or $d(\underline{x}, \underline{y})$, is simply the number of positions that $\underline{x}$ and $\underline{y}$ differ. Formally:

$$d_H(\underline{x}, \underline{y}) = |\{i : x_i \neq y_i\}|$$
$$0 \leq d_H(\underline{x}, \underline{y}) \leq n$$
$$e.g.\ d_H(0001, 1000) = 2$$

Definition: A 'predicate' or 'boolean proposition' over a domain D is a proposition which returns either true or false at each point of D.

Example:

$D = \mathbb{Z}^+ = \{1, 2, ..., \infty\}$

Predicate P :'$a \in D$ is even' $\begin{cases} for\ a \in \{2, 4, 6, ...\} \ \Rightarrow P\ returns\ true \\ for\ a \in \{1, 3, 5, ...\} \ \Rightarrow P\ returns\ false \end{cases}$

Definition: Function $\delta[\cdot]$ or the predicate function: Let P be a predicate over the domain D, with $a \in D$. Then the function $\delta[P(a)] : D \rightarrow \{0, 1\}$ is as follows:

$$\delta[P(a)] = \begin{cases} 1\ if\ P(a)\ is\ true \\ 0\ if\ P(a)\ is\ false \end{cases}$$

The following redefinition of the hamming function will be useful in future proofs:

$$d_H(\underline{x}, \underline{y}) = \sum_{i=1}^{n} \delta[x_i \neq y_i]$$

It is also convenient to discuss the minimum hamming distance of a code, which is defined as the minimum hamming distance between any two codewords in the code. Formally:

$$d_{Hmin}(C) = d_{min} = d = min\{d_H(\underline{c_i}, \underline{c_j}) : \underline{c_i} \in C \wedge \underline{c_j} \in C \wedge \underline{c_i} \neq \underline{c_j}\}$$

Examples:
$C = \{000, 111\}$ $d = 3$
$C = \{0001, 1000, 0100\}$ $d = 2$