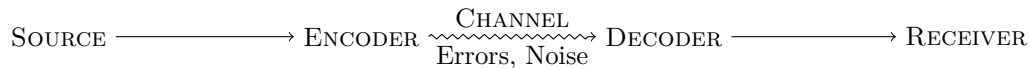


Coding & Cryptography

January 27, 2020

0 Communication Channels

This course will be about modelling communication. In general, we have the following idea:



For example, the channel might be an optical or electrical telegraph, modems, audio CDs, satellite relays. The encoding and decoding might be something like ASCII, so that each character in the email “Call at 2pm” would be encoded into 8 bits using $a = 1100101, \dots$, giving an 84 bit message to be transmitted via the internet, and decoded by the receiver’s email client. Our general aim here will be, given some source and channel (modelled probabilistically), to design an encoder and decoder to send messages economically and reliably.

Examples

- (Noiseless coding) Morse Code. In this code, more common letters are assigned shorter codes, so that we have $A = \cdot - *, E = \cdot *, Q = - - \cdot - *, Z = - - \cdot \cdot *$. This is adapted to the *source*, in the sense that we chose the codes based off the expected distribution of letters that we will have to transmit.
- (Noisy coding) ISBN. In the ISBN encoding, every book is given a 10 digit number $a_1 a_2 \dots a_{10}$, with $\sum_{i=1}^{10} (11-i)a_i \equiv 0 \pmod{11}$. This is adapted to the *channel*, in the sense that the likely errors to occur will be 1 incorrect digit, or accidentally transposing two digits, which this code is resistant to (will return an error rather than an erroneous result).

A **communication channel** accepts symbols from some alphabet $\mathcal{A} = \{a_1, a_2, \dots, a_r\}$ (e.g. $\{0, 1\}, \{a, b, \dots, z\}$), and outputs symbols from an alphabet $\mathcal{B} = \{b_1, \dots, b_s\}$. The channel is modelled by the probabilities:

$$\mathbb{P}(y_1, y_2, \dots, y_n \text{ received} | x_1, x_2, \dots, x_n \text{ sent}) = \prod_{i=1}^n \mathbb{P}(y_i \text{ received} | x_i \text{ sent})$$

A **discrete memoryless channel (DMC)** is a channel with $p_{ij} = \mathbb{P}(b_j \text{ received} | a_i \text{ sent})$ the same for each channel usage and independent of any past or future channel usages.

The **channel matrix** is $P = (p_{ij})$, an $r \times s$ stochastic matrix.

Examples: The **binary symmetric channel (BSC)** with error probability $p \in [0, 1]$ has $\mathcal{A} = \mathcal{B} = \{0, 1\}$. The channel matrix is $\begin{pmatrix} 1-p & p \\ p & 1-p \end{pmatrix}$. A symbol is transmitted correctly with probability $1-p$.

The **binary erasure channel** has input alphabet $\{0, 1\}$, and output alphabet $\{0, 1, *\}$, where we miss a bit is probability p , giving channel matrix $\begin{pmatrix} 1-p & 0 & p \\ 0 & 1-p & 0 \end{pmatrix}$. We can model n uses of a channel by the n^{th} **extension** with input alphabet \mathcal{A}^n , and output alphabet \mathcal{B}^n .

A **code** c of **length** n is a function $c : M \rightarrow \mathcal{A}^n$ where M is the set of all possible messages. Implicitly, we also have a decoding rule $\mathcal{B}^n \rightarrow M$.

The **size** of c is $m = |M|$.

The **information rate** is $\rho(c) = \frac{1}{n} \log_2(m)$.

The **error rate** is $\hat{e}(c) = \max_{x \in M} \{\mathbb{P}(\text{error} | x \text{ sent})\}$.

A channel can transmit reliably at a rate R if there exists a sequence of codes $(c_n : n \geq 1)$ with c_n a code of length n , $\lim_{n \rightarrow \infty} (\rho(c_n)) = R$, $\lim_{n \rightarrow \infty} (\hat{e}(c_n)) = 0$. The capacity of a channel is the supremum of all reliable transmission rates.

Theorem 0.1. *A BSC with error probability $p < \frac{1}{2}$ has a non-zero capacity (i.e. good codes exist).*

Proof. See 9.3 □

1 Noiseless Coding

1.1 Prefix-free Codes

For an alphabet \mathcal{A} , $|\mathcal{A}| < \infty$, let $\mathcal{A}^* = \bigcup_{n \geq 0} \mathcal{A}^n$, the set of all finite strings from \mathcal{A} . The **concatenation** of strings $x = x_1 \dots x_r$ and $y = y_1 \dots y_s$ is $xy = x_1 \dots x_r y_1 \dots y_s$.

Let \mathcal{A}, \mathcal{B} , be alphabets. A **code** is a function $c : \mathcal{A} \rightarrow \mathcal{B}^*$. The strings $c(a)$ for $a \in \mathcal{A}$ are called **codewords** (cws). If $x, y \in \mathcal{B}^*$ then x is a **prefix** of y if $y = xz$ for some $z \in \mathcal{B}^*$.

For example, we have the Greek fire code, found in the writings of Polybius around 280 BC. $\mathcal{A} = \{\alpha, \beta, \dots, \omega\}$, $\mathcal{B} = \{1, 2, 3, 4, 5\}$, with code $\alpha \mapsto 11, \beta \mapsto 12, \dots, \psi \mapsto 53, \omega \mapsto 54$, where xy means “ x torches held up, and another y torches nearby”.

The English language is even a code: we can let \mathcal{A} be words in a given dictionary, and $\mathcal{B} = \{a, b, \dots, z, \square\}$, where the coding function is to spell the word and follow it with a space.

We send a message $x_1 \dots x_n \in \mathcal{A}^*$ as $c(x_1) \dots c(x_n) \in \mathcal{B}^*$. So c extends to a function $c^* : \mathcal{A}^* \rightarrow \mathcal{B}^*$.

c is **decipherable/decidable** if c^* is injective, so that each string in \mathcal{B}^* could have come from at most one message. Note that it isn’t sufficient to just have c injective, although clearly this is necessary:

$\mathcal{A} = \{1, 2, 3, 4\}$, $\mathcal{B} = \{0, 1\}$, $c : 1 \mapsto 0, 2 \mapsto 1, 3 \mapsto 00, 4 \mapsto 01$. Then $c^*(114) = 0001 = c^*(312)$.

If $|\mathcal{A}| = m$, $|\mathcal{B}| = a$, then we say c is an **a -ary code of size m** . 2-ary = **binary**, 3-ary = **ternary**.

We aim to construct decipherable codes with short word lengths. Assuming c is injective, the following are always decipherable:

- Block codes, where every codeword has the same length (e.g. Greek fire, ASCII)

- Comma codes, where we have an “end of word” character (e.g. English language)
- Prefix-free codes, where no codeword is a prefix of any other distinct words.

Note that both of the first two are special cases of prefix-free codes. Prefix-free codes are often called *instantaneous* or *self-punctuating* codes. Note that not all decipherable codes are prefix-free: $0 \mapsto 01, 1 \mapsto 011$ is decipherable but not prefix free.

Theorem 1.1 (Kraft’s Inequality). *Let $|\mathcal{A}| = m, |\mathcal{B}| = a$. A prefix-free code $c : \mathcal{A} \rightarrow \mathcal{B}^*$ with word lengths ℓ_1, \dots, ℓ_m exists if and only if:*

$$\sum_{i=1}^m a^{-\ell_i} \leq 1 \quad (*)$$

Proof. Rewrite $(*)$ as $\sum_{\ell=1}^s n_\ell a^{-\ell} \leq 1$, where n_ℓ is the number of codewords of length ℓ and $s = \max_{1 \leq i \leq m} \ell_i$.

\implies If $c : \mathcal{A} \rightarrow \mathcal{B}^*$ is prefix-free, then $n_1 a^{s-1} + n_2 a^{s-2} + \dots + n_s \leq a^s$, since the LHS is the number of strings of length s in \mathcal{B} with some codeword of c as a prefix, and RHS is the number of strings of length s . Dividing by a^s gives $(*)$.

\impliedby Given n_1, \dots, n_s satisfying $(*)$, we need to construct a prefix-free code c with n_ℓ codewords of length ℓ for all $\ell \leq s$. We use induction on s . The case $s = 1$ is clear: we have $(*)$ gives $n_1 \leq a$, so we can choose a code.

By the induction hypothesis there is a prefix-free code \hat{c} with n_ℓ codewords of length ℓ for all $\ell \leq s - 1$. Then $(*)$ gives:

$$n_1 a^{s-1} + n_2 a^{s-2} + \dots + n_{s-1} a + n_s \leq a^s$$

where the first $s - 1$ terms on LHS sum to the number of strings of length s with some codeword of \hat{c} as a prefix, and the RHS is the number of strings of length s . Hence we can add at least n_s new codewords of length s to \hat{c} and maintain the prefix-free property, giving our code. □

Theorem 1.2 (McMillan). *Any decipherable code satisfies Kraft’s inequality*

Karush. Let $c : \mathcal{A} \rightarrow \mathcal{B}^*$ be a decipherable code with codewords of lengths ℓ_1, \dots, ℓ_m . Let $s = \max_{1 \leq i \leq m} \ell_i$. Then for $R \in \mathbb{N}$:

$$\left(\sum_{i=1}^m a^{-\ell_i} \right)^R = \sum_{\ell=1}^{Rs} b_\ell a^{-\ell}$$

where $b_\ell = |\{x \in \mathcal{A}^R : c^*(x) \text{ has length } \ell\}| \leq |\mathcal{B}^\ell| = a^\ell$, using the fact that c^* is injective. Then:

$$\begin{aligned} \left(\sum_{i=1}^m a^{-\ell_i} \right)^R &\leq \sum_{\ell=1}^R a^\ell a^{-\ell} = Rs \\ \sum_{i=1}^m a^{-\ell_i} &\leq (Rs)^{\frac{1}{R}} \rightarrow 1 \text{ as } R \rightarrow \infty \end{aligned}$$

□

Corollary 1.3. *A decipherable code with prescribed word lengths exists iff a prefix-free code with same word lengths exists.*

Proof.

\Rightarrow Use **1.2** to generate a prefix-free code by **1.1**

\Leftarrow Prefix-free codes are decipherable.

□

2 Shannon's Noiseless Coding Theorem

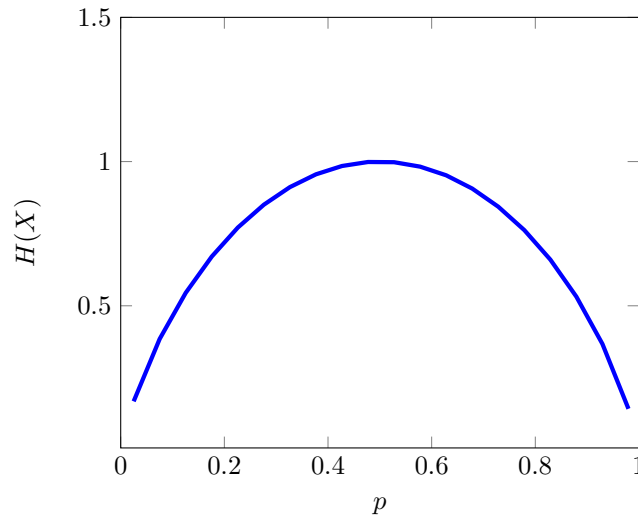
Entropy is a measure of ‘randomness’ or ‘uncertainty’. Suppose we have a random variable X that takes values x_1, \dots, x_n with probabilities p_1, \dots, p_n . Then the **entropy** (roughly speaking) is the expected number of fair coin tosses needed to simulate X .

Examples:

- $p_1 = p_2 = p_3 = p_4 = \frac{1}{4}$. We can identify $\{x_1, x_2, x_3, x_4\}$ with $\{HH, HT, TH, TT\}$, and so the entropy of this random variable is 2.
- $(p_1, p_2, p_3, p_4) = (\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{8})$. Here, the entropy is $1 \cdot \frac{1}{2} + 2 \cdot \frac{1}{4} + 3 \cdot \frac{1}{8} + 3 \cdot \frac{1}{8} = \frac{7}{4}$. We might say then, since the entropy is greater, that the first example is “more random” than the second.

More concretely, the **Shannon (or information) entropy** of X is $H(X) = -\sum_{i=1}^n p_i \log_2 p_i$. Note that $H(X) \geq 0$ with equality if and only if $\mathbb{P}(X = x_i) = 1$. This is measured in **bits**. We take the convention that $0 \log 0 = 0$.

Example: Consider a biased coin, where $\mathbb{P}(X = H) = p, \mathbb{P}(X = T) = 1 - p$. Then $H(X) = -p \log p - (1 - p) \log(1 - p) = p \log(\frac{1-p}{p}) - \log(1 - p)$.



Proposition 2.1 (Gibb's Inequality). *Let (p_1, \dots, p_n) and (q_1, \dots, q_n) be probability distributions. Then:*

$$-\sum_{i=1}^n p_i \log p_i \leq -\sum_{i=1}^n p_i \log q_i$$

with equality if and only if $p_i = q_i$ for all i .

Proof. Since $\log x = \frac{\ln x}{\ln 2}$, we may replace \log by \ln in the proof. Put $I = \{1 \leq r \leq n : p_i \neq 0\}$. Now $\ln x \leq x - 1$ with equality if and only if $x = 1$. So we have $\ln \frac{q_i}{p_i} \leq \frac{q_i}{p_i} - 1$, and hence:

$$\begin{aligned} \sum_{i \in I} p_i \ln \frac{q_i}{p_i} &\leq \sum_{i \in I} q_i - \sum_{i \in I} p_i \\ &= \sum_{i \in I} q_i - 1 \leq 0 \\ \therefore -\sum_{i \in I} p_i \ln p_i &\leq -\sum_{i \in I} p_i \ln q_i \\ \therefore -\sum_{i=1}^n p_i \log p_i &\leq -\sum_{i=1}^n p_i \log q_i \end{aligned}$$

If equality holds, then $\sum_{i \in I} p_i = 1$ and $\frac{p_i}{q_i} = 1$ for all $i \in I$, so $p_i = q_i$ □

Corollary 2.2. $H(p_1, \dots, p_n) \leq \log n$ with equality if and only if $p_1 = \dots = p_n = \frac{1}{n}$.

Proof. Take $q_1 = \dots = q_n = \frac{1}{n}$ in 2.1. □

Let $\mathcal{A} = \{\mu_1, \dots, \mu_m\}$, and $|\mathcal{B}| = a$, where $m, a \geq 2$. The random variable X takes values μ_1, \dots, μ_m with probabilities p_1, \dots, p_m . We say a code $c : \mathcal{A} \rightarrow \mathcal{B}^*$ is **optimal** if it is a decipherable code with smallest possible expected word length, $\mathbb{E}S = \sum_i p_i \ell_i$.

Theorem 2.3 (Shannon's Noiseless Coding Theorem). *The expected word length $\mathbb{E}S$ of an optimal code satisfies:*

$$\frac{H(X)}{\log a} \leq \mathbb{E}S < \frac{H(X)}{\log a} + 1$$

Proof. For the lower bound, take $c : \mathcal{A} \rightarrow \mathcal{B}^*$ decipherable with word lengths ℓ_1, \dots, ℓ_m . Then set $q_i = \frac{a^{-\ell_i}}{D}$ where $D = \sum_{i=1}^m a^{-\ell_i}$. Now we have that $\sum_{i=1}^m q_i = 1$. By Gibbs,

$$\begin{aligned} H(X) &\leq -\sum_{i=1}^m p_i \log q_i \\ &= -\sum_{i=1}^m p_i (-\ell_i \log a - \log D) \\ &= \left(\sum_{i=1}^m p_i \ell_i \right) \log a + \log D \end{aligned}$$

By McMillan, $D \leq 1$, so $\log D \leq 0$, and so $H(X) \leq (\sum_{i=1}^m p_i \ell_i) \log a = \mathbb{E}S \cdot \log a$, and we have equality if and only if $p_i = a^{-\ell_i}$ for some integers ℓ_1, \dots, ℓ_m .

For the upper bound, take $\ell_i = \lceil -\log_a p_i \rceil$. Then $-\log_a p_i \leq \ell_i \implies p_i \geq a^{-\ell_i}$.

Now $\sum_{i=1}^m a^{-\ell_i} \leq \sum_{i=1}^m p_i = 1$. By Kraft, there is some prefix-free code c with word lengths ℓ_1, \dots, ℓ_m , and the expected word length of c is $\mathbb{E}S = \sum p_i \ell_i < \sum p_i (-\log_a p_i + 1) = \frac{H(X)}{\log a} + 1$. \square

Example: *Shannon-Fano coding*

We mimic the above proof: given probabilities p_1, \dots, p_n , set $\ell_i = \lceil -\log_a p_i \rceil$. Construct the prefix-free code with word lengths ℓ_1, \dots, ℓ_m by choosing in order of increasing length, ensuring that previous codewords are not prefixes. For example, if $a = 2, m = 5$ we have:

i	p_i	$\lceil -\log_2 p_i \rceil$	Codewords
1	0.4	2	00
2	0.2	3	010
3	0.2	3	011
4	0.1	4	1000
5	0.1	4	1001

$$\mathbb{E}S = \sum p_i \ell_i = 2.8, \text{ entropy} = 2.12$$

3 Huffman Coding Algorithm

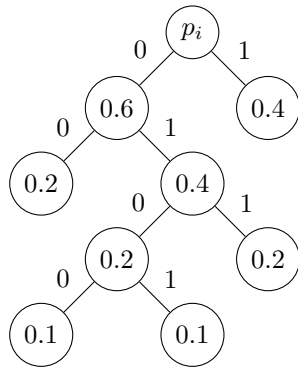
Huffman was a student of Fano, and was thinking about how to construct an optimal code. For simplicity, we will take $a = 2$. Suppose we get messages with orders $p_1 \geq p_2 \geq \dots \geq p_m$. Huffman gave a recursive definition of codes that we can prove are optimal. If $m = 2$, then take codewords 0 and 1.

If $m > 2$, we first have a Huffman code for messages $\mu_1, \dots, \mu_{m-2}, \nu$ with probabilities $p_1, \dots, p_{m-2}, p_{m-1} + p_m$, then append 0 and 1 to give codewords for μ_{m-1} and μ_m .

Note:

- Huffman codes are prefix-free.
- We have some choices to make if some of the p_j are equal, so Huffman codes are not unique.

Example: Reconsider the previous example:



i	p_i	Codewords
1	0.4	1
2	0.2	00
3	0.2	011
4	0.1	0100
5	0.1	0101

This code has expected length 2.2, which is less than Shannon-Fano gave.

Theorem 3.1 (Huffman, 1952). *Huffman codes are optimal.*

Proof. We show this by induction on m . The case of $m = 2$ is trivial. For $m > 2$, let c_m be a Huffman code for source X_m which takes values μ_1, \dots, μ_m with probabilities $p_1 \geq \dots \geq p_m$. Then c_{m-1} is constructed from a Huffman code c_{m-1} for values $\mu_1, \dots, \mu_{m-1}, \nu$ with probabilities $p_1, \dots, p_{m-2}, p_{m-1} + p_m$.

Observe that $\mathbb{E}S_m = \mathbb{E}S_{m-1} + p_{m-1} + p_m$ by construction of c_m from c_{m-1} .

Now let c'_m be an optimal code for X_m . Without loss of generality, we may take c'_m to be prefix-free and the last two codewords of c'_m have maximal length and differ only in the last digit (see 3.2 below). Say $c'_m(\mu_{m-1}) = y0, c'_m(\mu_m) = y1$ for some $y \in \{0, 1\}^*$.

Let c'_{m-1} be the prefix free code for X_{m-1} given by $c'_{m-1}(\mu_i) = c'_m(\mu_i), c'_{m-1}(\nu) = y$.

Then the expected word length is $\mathbb{E}S'_m = \mathbb{E}S'_{m-1} + p_{m-1} + p_m \geq \mathbb{E}S_{m-1} + p_{m-1} + p_m = \mathbb{E}S_m$ by the inductive hypothesis, and so c_m is optimal. \square

Lemma 3.2. *Suppose messages μ_1, \dots, μ_m are sent with probabilities p_1, \dots, p_m , with an optimal code c with word lengths ℓ_1, \dots, ℓ_m . Then:*

1. *If $p_i > p_j$ then $\ell_i \leq \ell_j$.*
2. *Among all codewords of maximal length, there are two that differ only in the last digit.*

Proof. Otherwise, modify c by swapping the i^{th} and j^{th} codewords, or deleting the last letter of each codeword of maximal length. The modified code is still prefix-free but has shorter expected word length, contradicting optimality of c . \square

4 Joint Entropy

If X, Y are random variables with value in \mathcal{A} and \mathcal{B} . Then (X, Y) is also a random variable with entropy $H(X, Y)$, the **joint entropy** of X, Y .

$$H(X, Y) = - \sum_{x \in \mathcal{A}} \sum_{y \in \mathcal{B}} \mathbb{P}(X = x, Y = y) \log \mathbb{P}(X = x, Y = y)$$

We can of course generalise this to any finite number of random variables. We will use Gibb's (2.1) to prove:

Lemma 4.1. *Let X, Y be random variables taking values in \mathcal{A}, \mathcal{B} . Then:*

$$H(X, Y) \leq H(X) + H(Y)$$

with equality if and only if X and Y are independent.

Proof. Let $\mathcal{A} = \{x_1, \dots, x_m\}$, $\mathcal{B} = \{y_1, \dots, y_n\}$. Set $p_{ij} = \mathbb{P}(X = x_i, Y = y_j)$, $p_i = \mathbb{P}(X = x_i)$, $q_j = \mathbb{P}(Y = y_j)$. Then Gibb's inequality with $\{p_{ij}\}$ and $\{p_i q_j\}$ gives:

$$\begin{aligned} -\sum_{i,j} p_{ij} \log p_{ij} &\leq -\sum_{i,j} p_{ij} \log(p_i q_j) = -\sum_i \left(\sum_j p_{ij} \right) \log p_i - \sum_j \left(\sum_i p_{ij} \right) \log q_j \\ &= -\sum_i p_i \log p_i - \sum_j q_j \log q_j \end{aligned}$$

i.e. $H(X, Y) \leq H(X) + H(Y)$, with equality if and only if $p_{ij} = p_i q_j$ for all i, j , i.e. when X, Y are independent. \square

Example: Let X be a random variable that takes D values with probability $\frac{1}{D}$. Then $H(X) = \log_2(D)$. Suppose X_1, \dots, X_N are i.i.d. with the same distribution as X . Then $H(X_1, \dots, X_N) = N \log_2 D$.

5 Error Correcting Codes

5.1 Noisy Channels and Hamming's Code

A **binary $[n, m]$ -code** is a subset $C \subseteq \{0, 1\}^n$ of **size** $m = |C|$, **length** n . The elements of C are called **codewords**. We use an $[n, m]$ -code to send one of m messages through a binary symmetric channel, making n uses of the channel. Clearly $1 \leq m \leq 2^n$, so $0 \leq \frac{1}{n} \log m \leq 1$. If $|C| = 1$ then $\rho(C) = 0$, and if $C = \{0, 1\}^n$ then $\rho(C) = 1$.

For $x, y \in \{0, 1\}^n$, the **Hamming distance** $d(x, y) = |\{i : 1 \leq i \leq n, x_i \neq y_i\}|$, i.e. the number of positions where x and y differ.

We have three possible decoding rules:

1. The **ideal observer** decoding rule decodes $x \in \{0, 1\}^n$ as $c \in C$ maximising $\mathbb{P}(c \text{ sent} | x \text{ received})$.
2. Then **maximum likelihood** decoding rule decodes $x \in \{0, 1\}^n$ as $c \in C$ maximising $\mathbb{P}(x \text{ received} | c \text{ sent})$.
3. The **minimum distance** decoding rule decodes $x \in \{0, 1\}^n$ as $c \in C$ minimising $d(x, c)$.

Lemma 5.1.

1. If all the messages are equally likely, then 1. and 2. agree.
2. If $p < \frac{1}{2}$, then 2. and 3. agree.

Proof.

1. By Bayes' Rule:

$$\mathbb{P}(c \text{ sent} | x \text{ received}) = \frac{\mathbb{P}(c \text{ sent})}{\mathbb{P}(x \text{ received})} \mathbb{P}(x \text{ received} | c \text{ sent})$$

By Hypothesis, $\mathbb{P}(c \text{ sent})$ is independent of $c \in C$, and so for fixed x , maximising $\mathbb{P}(c \text{ sent} | x \text{ received})$ is the same as maximising $\mathbb{P}(x \text{ received} | c \text{ sent})$.

Let $r = d(x, c)$. Then $\mathbb{P}(x \text{ received} | c \text{ sent}) = p^r (1-p)^{n-r} = (1-p)^n \left(\frac{p}{1-p}\right)^r$. Since $p < \frac{1}{2}$, $\frac{p}{1-p} < 1$, and so maximising $\mathbb{P}(x \text{ received} | c \text{ sent})$ is the same as minimising $d(x, c)$. \square

For instance, suppose 000 is sent with probability $\frac{9}{10}$, and 111 with probability $\frac{1}{10}$, through a binary symmetric channel with error probability $\frac{1}{4}$. If we receive 110, the ideal receiver computes $\mathbb{P}(000 \text{ sent} | 110 \text{ received}) = \frac{3}{4}$; $\mathbb{P}(110 \text{ sent} | 110 \text{ received}) = \frac{1}{4}$, and so decodes it as 000. But the minimum distance (and so maximal likelihood) code is 111. Henceforth, we will decide to use minimal distance decoding.

Note that minimal distance decoding can be expensive in terms of time and storage if $|C|$ is large, and we also need to specify a convention in the case of a tie (e.g. make a random choice, request the message again).

A code is ***d-error detecting*** if changing up to d digits in each codeword can never produce another codeword. It is ***e-error correcting*** if, knowing that $x \in \{0, 1\}^n$ differs from some codeword in at most e places, we can deduce uniquely what the codeword is.

Examples

1. A ***repetition code*** of length n has codewords $00 \dots 0, 11 \dots 1$. This is an $[n, 2]$ -code. It is $(n-1)$ error detecting and $\lfloor \frac{n-1}{2} \rfloor$ -error correcting. But the information rate is only $\frac{1}{n}$.
2. A ***simple parity check code*** or ***paper tape code***: identify $\{0, 1\}$ with \mathbb{F}_2 (i.e. arithmetic modulo 2), and let $C = \{(x_1, \dots, x_n) \in \{0, 1\}^n : \sum x_i = 0\}$. This is an $[n, 2^{n-1}]$ -code. It is 1-error detecting, but cannot correct errors. Its information rate is $\frac{n-1}{n}$.
3. ***Hamming's Original Code*** is a 2-error detecting and 1-error correcting binary $[7, 16]$ -code:

$$C = \left\{ c \in \mathbb{F}_2^7 : \begin{array}{l} c_1 + c_3 + c_5 + c_7 = 0 \\ c_2 + c_3 + c_6 + c_7 = 0 \\ c_4 + c_5 + c_6 + c_7 = 0 \end{array} \right\}$$

The bits c_3, c_5, c_6, c_7 are arbitrary and c_1, c_2, c_4 are forced. The information rate is $\frac{4}{7}$.

Given $x \in \mathbb{F}_2^7$, we form the ***syndrome*** $z = (z_1, z_2, z_4) \in \mathbb{F}_2^3$, where $z_1 = x_1 + x_3 + x_5 + x_7$, $z_2 = x_2 + x_3 + x_6 + x_7$, $z_4 = x_4 + x_5 + x_6 + x_7$. If $x \in C$ then $z = (0, 0, 0)$. If $d(x, c) = 1$ for some $c \in C$ then x_i and c_i differ for $i = z_1 + 2z_2 + 4z_4$.