

Part II – Coding and Cryptography

Based on lectures by Dr R. Camina

Notes taken by Bhavik Mehta

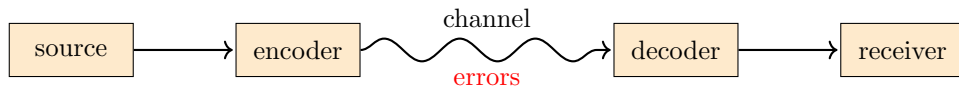
Lent 2017

Contents

1 Noiseless Coding	4
1.1 Mathematical Entropy	6
2 Error Control Codes	15

Introduction to communication channels and coding

For example, given a message m = ‘Call me!’ which we wish to send by email, first encode as binary strings using ASCII. So, $f(C) = 1000011$, $f(a) = 1100001$, and $f^*(m) = 1000011\ 1100001 \dots 0100001$.



Basic problem: Given a source and a channel (described probabilistically) we aim to design an encoder and a decoder in order to transmit information both economically and reliably (coding) and maybe also to preserve privacy (cryptography).

Example.

- ‘economically’ Morse code: common letters have shorter codewords:
- ‘reliably’ Every book has an ISBN of form $a_1a_2 \dots a_{10}$ where $a_i \in \{0, 1, \dots, 9\}$ for $1 \leq i \leq 9$, $a_{10} \in \{0, 1, \dots, 9, X\}$ such that

$$10a_1 + 9a_2 + \dots + a_{10} \equiv 0 \pmod{11}$$

so errors can be detected (but not corrected). Similarly a 13-digit ISBN has

$$x_1 + 3x_2 + x_3 + 3x_4 + \dots + 3x_{12} + x_{13} \equiv 0 \pmod{10}$$

for $0 \leq x_i \leq 10$, doesn’t necessarily spot transpositions.

- ‘preserve privacy’ e.g. RSA.

A **communication channel** takes letters from an input alphabet $\Sigma_1 = \{a_1, \dots, a_r\}$ and emits letters from an output alphabet $\Sigma_2 = \{b_1, \dots, b_s\}$.

A channel is determined by the probabilities

$$P(y_1 \dots y_k \text{ received} \mid x_1 \dots x_k \text{ sent})$$

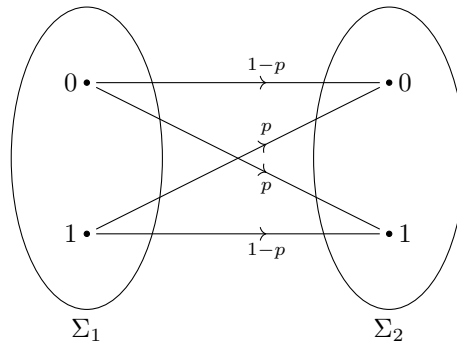
Definition (Discrete memoryless channel). A **discrete memoryless channel** (DMC) is a channel for which

$$P_{ij} = P(b_j \text{ received} \mid a_i \text{ sent})$$

is the same each time the channel is used and is independent of all past and future uses.

The channel matrix is the $r \times s$ matrix with entries p_{ij} (note the rows sum to 1).

Example. Binary Symmetric Channel (BSC) has $\Sigma_1 = \Sigma_2 = \{0, 1\}$, $0 \leq p \leq 1$:

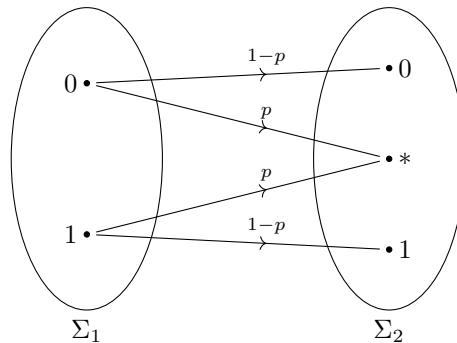


with channel matrix

$$\begin{pmatrix} 1-p & p \\ p & 1-p \end{pmatrix}$$

i.e. p is the probability a symbol is mistransmitted.

Another example is given by the Binary Erasure channel, $\Sigma_1 = \{0, 1\}$, $\Sigma_2 = \{0, 1, *\}$ and $0 \leq p \leq 1$.



with channel matrix

$$\begin{pmatrix} 1-p & p & 0 \\ 0 & p & 1-p \end{pmatrix}$$

i.e. p is the probability a symbol can't be read.

Informally, a channel's capacity is the highest rate at which information can be reliably transmitted over the channel. Rate refers to units of information per unit time, which we want to be high. Similarly, reliably means we want an arbitrarily small error probability.

1 Noiseless Coding

Notation. For Σ an alphabet, let $\Sigma^* = \bigcup_{n \geq 0} \Sigma^n$ be the set of all finite strings of elements of Σ .

If $x = x_1 \dots x_r$, $y = y_1 \dots y_s$ are strings from Σ , write xy for the concatenation $x_1 \dots x_r y_1 \dots y_s$. Further, $|x_1 \dots x_r y_1 \dots y_s| = r + s$, the length of the string.

Definition (Code). Let Σ_1, Σ_2 be two alphabets. A **code** is a function $f : \Sigma_1 \rightarrow \Sigma_2^*$. The strings $f(x)$ for $x \in \Sigma_1$ are called **codewords**.

Example.

- 1) Greek fire **code**:

$$\begin{aligned}\Sigma_1 &= \{\alpha, \beta, \gamma, \dots, \omega\} & 24 \text{ letters} \\ \Sigma_2 &= \{1, 2, 3, 4, 5\}\end{aligned}$$

so, $\alpha \mapsto 11, \beta \mapsto 12, \dots, \omega \mapsto 54$.

- 2) $\Sigma_1 = \{\text{all words in the dictionary}\}$, and $\Sigma_2 = \{A, B, \dots, Z, [\text{space}]\}$ and $f = \text{'spell the word and a space'}$.

Send a message $x_1 \dots x_n \in \Sigma_1^*$ as $f(x_1) \dots f(x_n) \in \Sigma_2^*$ i.e. extend f to $f^* : \Sigma_1^* \rightarrow \Sigma_2^*$.

Definition (Decipherable). A **code** f is **decipherable** if f^* is injective, i.e. every string from Σ_2^* arises from at most one message. Clearly we need f injective, but this is not enough.

Example. Take $\Sigma_1 = \{1, 2, 3, 4\}$, $\Sigma_2 = \{0, 1\}$ with

$$f(1) = 0, f(2) = 1, f(3) = 00, f(4) = 01.$$

f injective but $f^*(312) = 0001 = f^*(114)$ so f^* not **decipherable**.

Notation. If $|\Sigma_1| = m$, $|\Sigma_2| = a$, then we say f is an a -ary code of size m . (If $a = 2$ we say binary).

Aim. Construct **decipherable codes** with short word lengths.

Provided $f : \Sigma_1 \rightarrow \Sigma_2^*$ is injective, the following codes are always decipherable.

- (i) A **block code** is a code with all codewords of the same length (e.g. **Greek fire code**).
- (ii) In a **comma code**, we reserve one letter from Σ_2 that is only used to signal the end of the codeword (e.g. **Example 2 above**).
- (iii) A **prefix-free code** is a code where no codeword is a prefix of another (if $x, y \in \Sigma_2^*$, x is a prefix of y if $y = xz$ for some $z \in \Sigma_2^*$).

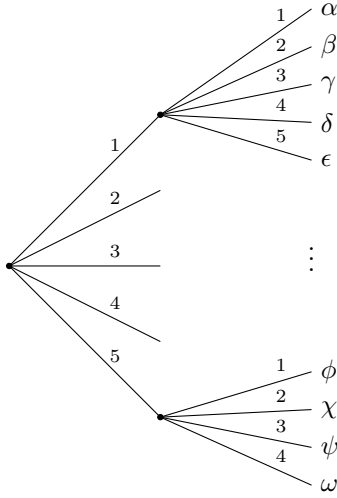
Remark. (i) and (ii) are special cases of (iii).

Prefix-free codes are also known as **instantaneous codes** (i.e. a word can be recognised as soon as it is complete) or **self-punctuating codes**.

Theorem 1.1 (Kraft's inequality). Let $|\Sigma_1| = m, |\Sigma_2| = a$. A **prefix-free code** $f : \Sigma_1 \rightarrow \Sigma_2^*$ with word lengths s_1, \dots, s_m exists iff

$$\sum_{i=1}^m a^{-s_i} \leq 1.$$

Proof. (\Rightarrow) Consider an infinite tree where each node has a descendant, labelled by the elements of Σ_2 . Each **codeword** corresponds to a node, the path from the root to this node spelling out the codeword. For example,



Assuming f is **prefix-free**, no codeword is the ancestor of any other. Now view the tree as a network with water being pumped in at a constant rate and dividing the flow equally at each node.

The total amount of water we can extract at the codewords is $\sum_{i=1}^m a^{-s_i}$, which is therefore ≤ 1 .

(\Leftarrow) Conversely, suppose we can construct a prefix-free code with word lengths s_1, \dots, s_m , wlog $s_1 \leq s_2 \leq \dots \leq s_m$. We pick codewords of lengths s_1, s_2, \dots sequentially ensuring previous codewords are not prefixes. Suppose there is no valid choice for the r th codeword. Then reconstructing the tree as above gives $\sum_{i=1}^{r-1} a^{-s_i} = 1$, contradicting our assumption. So we can construct a prefix-free code. (There is a more algebraic proof in Welsh.) \square

Theorem 1.2 (McMillan). Every **decipherable code** satisfies **Kraft's inequality**.

Proof. (Karush) Let $f : \Sigma_1 \rightarrow \Sigma_2^*$ be a **decipherable code** with word lengths s_1, \dots, s_m , let $s = \max_{1 \leq i \leq m} s_i$. Let $r \in \mathbb{N}$,

$$\left(\sum_{i=1}^m a^{-s_i} \right)^r = \sum_{l=1}^{rs} b_l a^{-l}$$

where b_l is the # of ways of choosing r codewords of total length l . f decipherable $\implies b_l \leq |\Sigma_2|^l = a^l$.

Thus

$$\begin{aligned} \left(\sum_{i=1}^m a^{-s_i} \right)^r &\leq \sum_{l=1}^{rs} a^l a^{-l} = rs \\ \Rightarrow \sum_{i=1}^m a^{-s_i} &\leq (rs)^{\frac{1}{r}} \rightarrow 1 \text{ as } r \rightarrow \infty. \end{aligned}$$

(As $\frac{\log r + \log s}{r} \rightarrow 0$ as $r \rightarrow \infty$).

$$\therefore \sum_{i=1}^m a^{-s_i} \leq 1.$$

□

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

So we can restrict our attention to prefix-free codes.

1.1 Mathematical Entropy

Definition (Entropy). The entropy of X :

$$H(X) = H(p_1, \dots, p_n) = - \sum_{i=1}^n p_i \log p_i$$

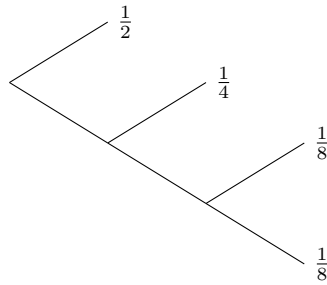
where, in this course, $\log = \log_2$.

Remark.

- (i) If $p_i = 0$, we take $p_i \log p_i = 0$.
- (ii) $H(x) \geq 0$.

Example.

1. Suppose $p_1 = p_2 = p_3 = p_4 = \frac{1}{4}$. We identify $\{x_1, x_2, x_3, x_4\}$ with $\{\text{HT}, \text{HT}, \text{TH}, \text{TT}\}$. Then $H(X) = 2$.
2. Take $(p_1, p_2, p_3, p_4) = (\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{8})$.

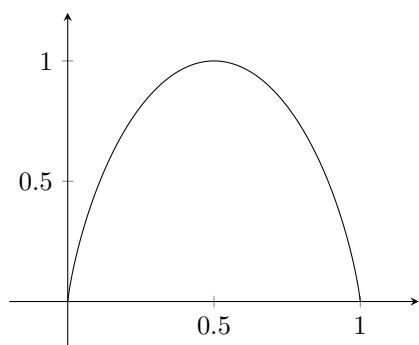


$$H(X) = \frac{1}{2} \times 1 + \frac{1}{4} \times 2 + \frac{1}{8} \times 3 + \frac{1}{8} \times 3 = \frac{7}{4}.$$

So example 1 is more random than example 2.

Entropy is a measure of ‘randomness’ or ‘uncertainty’. Consider a random variable X taking values x_1, \dots, x_n with probability p_1, \dots, p_n ($\sum p_i = 1, 0 \leq p_i \leq 1$). The entropy $H(X)$ is roughly speaking the expected number of tosses of a fair coin needed to simulate X (or the expected number of yes/no questions we need to ask in order to establish the value of X).

Example. We toss a biased coin, $P(\text{heads}) = p, P(\text{tails}) = 1-p$. Write $H(p) = H(p, 1-p) = -p \log p - (1-p) \log(1-p)$. If $p = 0$ or 1 , the outcome is certain and so $H(p) = 0$. **Entropy** is maximal where $p = \frac{1}{2}$, i.e. a fair coin.



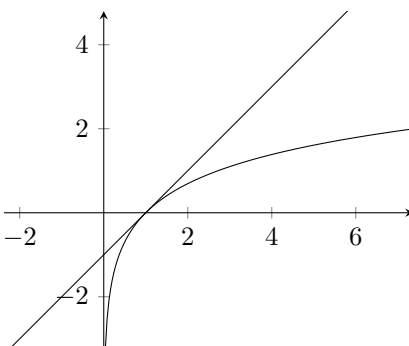
Note the **entropy** can also be viewed as the expected value of the information of X , where information is given by $I(X = x) = -\log_2 P(X = x)$. For example, if a coin always lands heads we gain no information from tossing the coin. The entropy is the average amount of information conveyed by a random variable X .

Lemma 1.3 (Gibbs’ Inequality). Let p_1, \dots, p_n and q_1, \dots, q_n be probability distributions. Then

$$-\sum p_i \log p_i \leq -\sum p_i \log q_i$$

with equality iff $p_i = q_i$.

Proof. Since $\log x = \frac{\ln x}{\ln 2}$ it suffices to prove the inequality with \log replaced with \ln . Note $\ln x \leq x - 1$, equality iff $x = 1$.



Let $I = \{1 \leq i \leq n \mid p_i \neq 0\}$

$$\begin{aligned}
& \ln \frac{q_i}{p_i} \leq \frac{q_i}{p_i} - 1 \quad \forall i \in I \\
& \sum_{i \in I} p_i \ln \frac{q_i}{p_i} \leq \sum_{i \in I} q_i - \underbrace{\sum_{i \in I} p_i}_{=1} \leq 0 \\
& \implies -\sum_{i \in I} p_i \ln p_i \leq -\sum_{i \in I} p_i \ln q_i \\
& \implies -\sum_{i=1}^n p_i \ln p_i \leq -\sum_{i=1}^n p_i \ln q_i
\end{aligned}$$

If equality holds then $\frac{q_i}{p_i} = 1 \quad \forall i \in I$. So, $\sum_{i \in I} q_i = 1$ and hence $p_i = q_i$ for $1 \leq i \leq n$. \square

Corollary. $H(p_1, \dots, p_n) \leq \log n$ with equality iff $p_1 = p_2 = \dots = p_n = \frac{1}{n}$.

Proof. Take $q_1 = q_2 = \dots = q_n = \frac{1}{n}$ in [Gibbs' Inequality](#). \square

Suppose we have two alphabets Σ_1, Σ_2 with $|\Sigma_1| = m$ and $|\Sigma_2| = a$, for $m \geq 2$ and $a \geq 2$. We model the source as a sequence of random variables X_1, X_2, \dots taking values in Σ_1 .

Definition (Memoryless source). A **Bernoulli** or **memoryless** source is a sequence of independently, identically distributed random variables.

That is, for each $\mu \in \Sigma_1$, $P(X_i = \mu)$ is independent of i and independent of all past and future symbols emitted. Thus

$$P(X_1 = x_1, X_2 = x_2, \dots, X_k = x_k) = \prod_{i=1}^k P(X_i = x_i).$$

Let $\Sigma_1 = \{\mu_1, \dots, \mu_n\}$, $p_i = P(X = \mu_i)$ (assume $p_i > 0$).

Definition (Expected word length). The **expected word length** of a code $f : \Sigma_1 \rightarrow \Sigma_2^*$ with word lengths s_1, \dots, s_m is $E(S) = \sum_{i=1}^m p_i s_i$.

Definition (Optimal code). A [code](#) $f : \Sigma_1 \rightarrow \Sigma_2^*$ is **optimal** if it has the shortest possible [expected word length](#) among [decipherable](#) codes.

Theorem 1.4 (Shannon's Noiseless Coding Theorem). The minimum [expected word length](#) of a [decipherable code](#) $f : \Sigma_1 \rightarrow \Sigma_2^*$ satisfies

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

Proof. The lower bound is given by combining [Gibbs' Inequality](#) and [Kraft's inequality](#). Let $q_i = \frac{a^{-s_i}}{c}$ where $c = \sum a^{-s_i} \leq 1$ by Kraft's inequality. Note $\sum q_i = 1$.

$$\begin{aligned}
H(X) &= -\sum p_i \log p_i \leq -\sum_i p_i \log q_i \\
&= \sum p_i (s_i \log a + \log c) \\
&= \left(\sum p_i s_i \right) \log a + \underbrace{\log c}_{\leq 0} \leq E(S) \log a \\
&\implies \frac{H(X)}{\log a} \leq E(S)
\end{aligned}$$

We get equality $\iff p_i = a^{-s_i}$ for some integers s_i . For the upper bound put

$$s_i = \lceil -\log_a p_i \rceil$$

where $\lceil x \rceil$ means least integer $\geq x$.

We have

$$\begin{aligned}
-\log_a p_i &\leq s_i < -\log_a p_i + 1 \\
\implies a^{-s_i} &\leq p_i \implies \sum a^{-s_i} \leq \sum p_i \leq 1.
\end{aligned}$$

So by [Theorem 1.1](#), \exists a prefix-free code with word lengths s_1, \dots, s_m . Also,

$$\begin{aligned}
E(S) &= \sum p_i s_i \\
&< \sum p_i (-\log_a p_i + 1) \\
&= \frac{H(X)}{\log a} + 1
\end{aligned}$$

□

Remark. The lower bound holds for all [decipherable codes](#).

Shannon-Fano coding

Follows from above proof. Set $s_i = \lceil -\log_a p_i \rceil$ and construct a [prefix-free code](#) with word lengths s_1, \dots, s_m by taking the s_i in increasing order ensuring that previous [codewords](#) are not prefixes. [Kraft's inequality](#) ensures there is enough room.

Example. Suppose μ_1, \dots, μ_5 are emitted with probabilities 0.4, 0.2, 0.2, 0.1, 0.1.

A possible [Shannon-Fano](#) code (with $a = 2$, $\Sigma_2 = \{0, 1\}$) has

p_i	$\lceil -\log_2 p_i \rceil$	
0.4	2	00
0.2	3	010
0.2	3	100
0.1	4	1100
0.1	4	1110

This has [expected word length](#)

$$\begin{aligned}
&= 2 \times 0.4 + 3 \times 0.2 + 3 \times 0.2 + 4 \times 0.1 + 4 \times 0.1 \\
&= 2.8.
\end{aligned}$$

compare $H(X) \approx 2.12$.

Huffman coding

For simplicity, take $a = 2$. Take $\Sigma_1 = \{\mu_1, \dots, \mu_m\}$ with $p_i = P(X = \mu_i)$. Without loss of generality, $p_1 \geq p_2 \geq \dots \geq p_m$. Huffman coding is defined inductively.

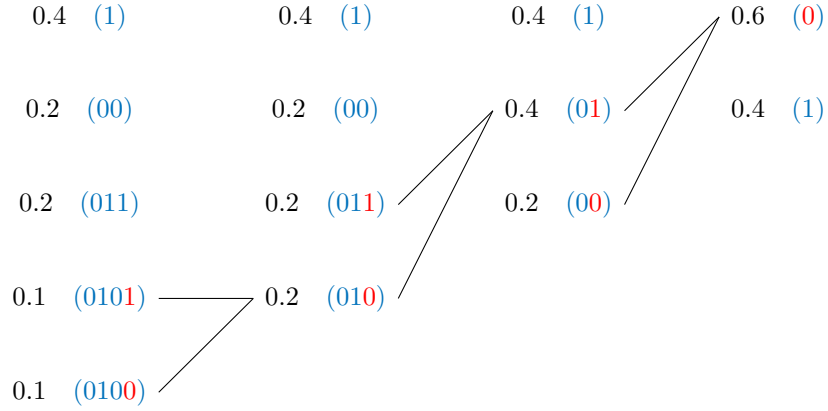
If $m = 2$, assign codewords 0 and 1. If $m > 2$, find a Huffman coding in the case of messages μ_1, μ_2, \dots, ν , with probabilities $p_1, p_2, \dots, p_{m-1} + p_m$.

Append 0 (resp, 1) to the codeword for ν to give a codeword for μ_{m-1} (resp, μ_m).

Remark.

- i) This construction gives a **prefix-free** code.
- ii) We exercise some choice when some of the p_i are equal. So **Huffman codes** are not unique.

Example. Use the same **example probabilities** as earlier.



So $\{1, 00, 011, 0101, 0100\}$ is the **prefix-free code** constructed. The **expected word length** is:

$$\begin{aligned}
 &= 1 \times 0.4 + 2 \times 0.2 + 2 \times 0.2 + 4 \times 0.1 + 4 \times 0.1 \\
 &= 0.4 + 0.4 + 0.6 + 0.4 + 0.4 \\
 &= 2.2.
 \end{aligned}$$

This is better than **Shannon-Fano**, which gave 2.8.

Theorem 1.5. **Huffman coding** is **optimal**.

Lemma 1.6. Suppose we have $\mu_1, \dots, \mu_m \in \Sigma_1$ emitted with probabilities p_1, \dots, p_m . Let f be an **optimal prefix-free code**, with word lengths s_1, \dots, s_m . Then

- i) If $p_i > p_j$, then $s_i \leq s_j$.
- ii) \exists two **codewords** of maximal length which are equal up to the last digit.

Proof.

- i) If not, then swap the i th and j th **codewords**. This decreases the **expected word length**, contradicting f **optimal**.

- ii) If not, then either only one codeword of maximal length, or any two codewords of maximal length differ before the last digit. In either case, delete the last digit of each codeword of maximal length. This maintains the [prefix-free](#) condition, contradicting f optimal. \square

Proof of Theorem 1.5 ($a = 2$). We show by induction on m that any [Huffman code](#) of size m is [optimal](#).

For $m = 2$, [codewords](#) 0, 1 are optimal.

For $m > 2$, say source X_m emits μ_1, \dots, μ_m with probabilities $p_1 \geq p_2 \geq \dots \geq p_m$ while source X_{m-1} emits $\mu_1, \dots, \mu_{m-2}, \nu$ with probabilities $p_1, \dots, p_{m-2}, p_{m-1} + p_m$.

We construct a Huffman coding f_{m-1} for X_{m-1} and extend to a Huffman coding for X_m . Then the [expected codeword length](#) satisfies:

$$E(s_m) = E(s_{m-1}) + p_{m-1} + p_m \quad (\dagger)$$

Let f'_m be an optimal code for X_m , WLOG f'_m prefix-free. [Lemma 1.6](#) tells us that by shuffling codewords, we may assume that the last two codewords are of maximal length and differ only in the last digit, say y_0 and y_1 for some string y .

We define a code f'_{m-1} for X_{m-1} with

$$\begin{aligned} f'_{m-1}(\mu_i) &= f'_m(\mu_i) \quad \forall 1 \leq i \leq m-2 \\ f'_{m-1}(\nu) &= y. \end{aligned}$$

Then f'_{m-1} is a prefix-free code, and the expected word length satisfies

$$E(s'_m) = E(s'_{m-1}) + p_{m-1} + p_m \quad (\ddagger)$$

Induction hypothesis tells us f_{m-1} is optimal, so $E(s_{m-1}) \leq E(s'_{m-1})$. Hence $E(s_m) \leq E(s'_m)$ by (\dagger) , (\ddagger) . So f_m optimal. \square

Remark. Not all [optimal](#) codes are [Huffman](#). For instance, take $m = 4$, and probabilities 0.3, 0.3, 0.2, 0.2. An optimal code is given by 00, 01, 10, 11, but this is not Huffman.

But, the previous result says that if we have a prefix-free optimal code with word lengths s_1, \dots, s_m and associated probabilities p_1, \dots, p_m , \exists a Huffman code with these word lengths.

Definition (Joint entropy). The **joint entropy** of X and Y is

$$H(X, Y) = - \sum_{x \in \Sigma_1} \sum_{y \in \Sigma_2} P(X = x, Y = y) \log P(X = x, Y = y).$$

Lemma 1.7. $H(X, Y) \leq H(X) + H(Y)$, with equality $\iff X, Y$ independent.

Proof. Take $\Sigma_1 = \{x_1, \dots, x_m\}$, $\Sigma_2 = \{y_1, \dots, y_n\}$. Let $p_{ij} = P(X = x_i, Y = y_j)$, as well as $p_i = P(X = x_i)$, $q_j = P(Y = y_j)$. Apply [Gibbs' Inequality](#) to the distributions p_{ij} and $p_i q_j$:

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

That is, $H(X, Y) \leq H(X) + H(Y)$. Equality $\iff p_{ij} = p_i q_j \forall i, j \iff X, Y$ independent. \square

Suppose a source Ω produces a stream X_1, X_2, \dots of random variables with values in Σ . The probability mass function (p.m.f.) of $X^{(n)} = (X_1, \dots, X_n)$ is given by

$$p_n(x_1, \dots, x_n) = P(X_1, \dots, X_n = x_1, \dots, x_n) \quad \forall x_1, \dots, x_n \in \Sigma^n$$

Now,

$$\begin{aligned} p_n &: \Sigma^n \rightarrow \mathbb{R} \\ X^{(n)} &: \Omega \rightarrow \Sigma^n \end{aligned}$$

can form

$$p_n(X^{(n)}) : \Sigma \xrightarrow{X^{(n)}} \Sigma^n \xrightarrow{p_n} \mathbb{R}$$

a random variable sending $\omega \mapsto p_n(X^{(n)} = X^{(n)}(\omega))$.

Example. Take $\Sigma = \{A, B, C\}$, with

$$X^{(2)} = \begin{cases} AB & p = 0.3 \\ AC & p = 0.1 \\ BC & p = 0.1 \\ BA & p = 0.2 \\ CA & p = 0.25 \\ CB & p = 0.05 \end{cases}$$

So, $p_2(AB) = 0.3$, etc, and $p_2(X^{(2)})$ takes values

$$p_2(X^{(2)}) = \begin{cases} 0.3 & p = 0.3 \\ 0.1 & p = 0.2 \\ 0.2 & p = 0.2 \\ 0.25 & p = 0.25 \\ 0.05 & p = 0.05 \end{cases}$$

Definition (Convergence in probability). A sequence of random variables X_1, X_2, \dots **converges in probability** to $c \in \mathbb{R}$, written $X_n \xrightarrow{p} c$ as $n \rightarrow \infty$, if

$$\forall \epsilon > 0 \quad P(|X_n - c| \leq \epsilon) \rightarrow 1 \quad \text{as } n \rightarrow \infty.$$

So, X_n and c can take very different values for large n , but only on a set with small probability.

Theorem (Weak law of large numbers). X_1, X_2, \dots an independent, identically distributed sequence of random variables with finite expected value μ , then

$$\frac{1}{n} \sum_{i=1}^n X_i \xrightarrow{p} \mu \quad \text{as } n \rightarrow \infty.$$

Example (Application). Take X_1, X_2, \dots a **Bernoulli source**. Then $p(X_1), p(X_2), \dots$ are i.i.d. random variables

$$\begin{aligned} p(X_1, \dots, X_n) &= p(X_1) \dots p(X_n) \\ -\frac{1}{n} \log p(X_1, \dots, X_n) &= -\frac{1}{n} \sum_{i=1}^n \log p(X_i) \xrightarrow{p} E(-\log p(X_1)) = H(X_1) \quad \text{as } n \rightarrow \infty. \end{aligned}$$

Definition (Asymptotic Equipartition Property).

A source X_1, X_2, \dots satisfies the **Asymptotic Equipartition Property** (AEP) if for some $H \geq 0$ we have

$$-\frac{1}{n} \log p(X_1, \dots, X_n) \xrightarrow{p} H \quad \text{as } n \rightarrow \infty.$$

Example. Consider a coin, $p(H) = p$. If coin tossed N times, expect approximately pN heads and $(1-p)N$ tails.

$$\begin{aligned} & P(\text{particular sequence of } pN \text{ heads and } (1-p)N \text{ tails}) \\ &= p^{pN} (1-p)^{(1-p)N} \\ &= 2^{N(p \log p + (1-p) \log(1-p))} = 2^{-NH(A)} \end{aligned}$$

where A is the result of an independent coin toss. So, with high probability we will get a typical sequence, and its probability will be close to $2^{-NH(A)}$.

Lemma 1.8. A source X_1, \dots, X_n satisfies **AEP** iff it satisfies the following.

$\forall \epsilon > 0, \exists n_0(\epsilon)$ such that $\forall n \geq n_0(\epsilon) \exists$ a ‘typical set’ $T_n \subset \Sigma^n$ with

$$\begin{aligned} & P((X_1, \dots, X_n) \in T_n) > 1 - \epsilon \\ & 2^{-n(H+\epsilon)} \leq p(x_1, \dots, x_n) \leq 2^{-n(H-\epsilon)} \quad \forall (x_1, \dots, x_n) \in T_n \end{aligned} \quad (*)$$

Sketch proof. **AEP** \Rightarrow $(*)$. Take

$$\begin{aligned} T_n &= \left\{ (x_1, \dots, x_n) \in \Sigma^n \mid \left| -\frac{1}{n} \log p(x_1, \dots, x_n) - H \right| < \epsilon \right\} \\ &= \left\{ (x_1, \dots, x_n) \in \Sigma^n \mid 2^{-n(H+\epsilon)} \leq p(x_1, \dots, x_n) \leq 2^{-n(H-\epsilon)} \right\} \end{aligned}$$

$(*) \Rightarrow$ **AEP**

$$P\left(\left| -\frac{1}{n} \log p(X_1, \dots, X_n) - H \right| < \epsilon\right) \geq P(T_n) \rightarrow 1 \quad \text{as } n \rightarrow \infty. \quad \square$$

Definition (Reliably encodable). A source X_1, X_2, \dots is **reliably encodable** at rate r if $\exists A_n \subset \Sigma^n$ for each n such that

- (i) $\frac{\log |A_n|}{n} \rightarrow r \quad \text{as } n \rightarrow \infty$
- (ii) $P((X_1, \dots, X_n) \in A_n) \rightarrow 1 \quad \text{as } n \rightarrow \infty$.

So, in principle you can encode at rate almost r with negligible error for long enough strings.

So, if $|\Sigma| = a$, you can **reliably encode** at rate $\log a$. However you can often do better. For example, consider telegraph English with 26 letters and a space. $27 \approx 2^{4.756}$, so can encode at rate of 4.76 bits/letter. But much lower rates suffice, as there is a lot of redundancy in the English language. Hence the following definition.

Definition (Information rate). The **information rate** H of a source is the infimum of all values at which it is **reliably encodable**.

Roughly, nH is the number of bits required to encode (X_1, \dots, X_n) .

Theorem 1.9 (Shannon's first coding theorem). If a source satisfies [AEP](#) with some constant H , then the source has information rate H .

Proof. Let $\epsilon > 0$ and let $T_n \subset \Sigma^n$ be [typical sets](#). Then for sufficiently large $n \geq n_0(\epsilon)$,

$$p(x_1, \dots, x_n) \geq 2^{-n(H+\epsilon)} \quad \forall (x_1, \dots, x_n) \in T_n.$$

So,

$$\begin{aligned} P((X_1, \dots, X_n) \in T_n) &\geq 2^{-n(H+\epsilon)} |T_n| \\ \implies |T_n| &\leq 2^{n(H+\epsilon)} \end{aligned}$$

therefore the source is [reliably encodable](#) at rate $H + \epsilon$.

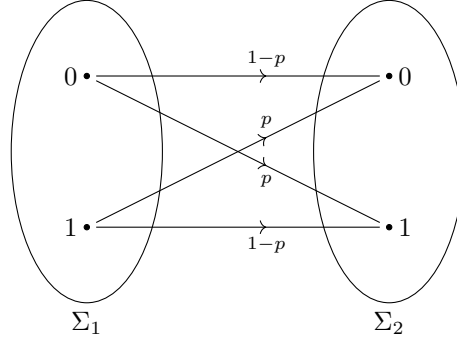
Conversely, if $H = 0$, done. Otherwise, pick $0 < \epsilon < \frac{H}{2}$. Suppose the source is reliably encodable at rate $H - 2\epsilon$, say with sets A_n .

$$\begin{aligned} p(x_1, \dots, x_n) &\leq 2^{-n(H-\epsilon)} \quad \forall (x_1, \dots, x_n) \in T_n \\ \implies P((x_1, \dots, x_n) \in A_n \cap T_n) &\leq 2^{-n(H-\epsilon)} |A_n| \\ \implies \frac{\log P(A_n \cap T_n)}{n} &\leq -(H - \epsilon) + \frac{\log |A_n|}{n} \\ &\rightarrow -(H - \epsilon) + (H - 2\epsilon) = -\epsilon \quad \text{as } n \rightarrow \infty. \quad \square \end{aligned}$$

2 Error Control Codes

Definition (Binary code). A $[n, m]$ **binary code** is a subset $C \subset \{0, 1\}^n$ of size $|C| = m$. We say C has length n . The elements of C are called codewords.

We use a $[n, m]$ -code to send one of m possible messages through a **BSC** making use of the channel n times.



Definition (Information rate). The **information rate** of C is

$$\rho(C) = \frac{\log m}{n}$$

where we continue to use $\log = \log_2$.

Note since $C \subset \{0, 1\}^n$, $\rho(C) \leq 1$, with equality iff $C = \{0, 1\}^n$. A code with size $m = 1$ has **information rate** 0. We aim to design code with both a large information rate and a small error rate, which are contradicting aims. The error rate depends on the decoding rule. We consider 3 possible rules.

- (i) The **ideal observer** decoding rule decodes $x \in \{0, 1\}^n$ as the codeword c maximising $P(c \text{ sent} \mid x \text{ received})$.
- (ii) The **maximum likelihood** decoding rule decodes $x \in \{0, 1\}^n$ as the codeword c maximising $P(x \text{ received} \mid c \text{ sent})$.
- (iii) The **minimum distance** decoding rule decodes $x \in \{0, 1\}^n$ as the codeword c minimising $\#\{1 \leq i \leq n \mid x_i \neq c_i\}$.

Remark. Some convention should be agreed in the case of a ‘tie’, e.g. choose at random, or ask for message to be sent again.

Lemma 2.1. If all messages are equally likely, then the **ideal observer** and **maximum likelihood rule** agree.

Proof. By Bayes’ rule,

$$\begin{aligned} P(c \text{ sent} \mid x \text{ received}) &= \frac{P(c \text{ sent}, x \text{ received})}{P(x \text{ received})} \\ &= \frac{P(c \text{ sent})}{P(x \text{ received})} P(x \text{ received} \mid c \text{ sent}). \end{aligned}$$

We suppose $P(c \text{ sent})$ is independent of c . So for fixed x , maximising $P(c \text{ sent} \mid x \text{ received})$ is the same as maximising $P(x \text{ received} \mid c \text{ sent})$. \square

Definition. Let $x, y \in \{0, 1\}^n$. Then **Hamming distance** between x and y is $d(x, y) = \#\{1 \leq i \leq n \mid x_i \neq y_i\}$.

Lemma 2.2. If $p < \frac{1}{2}$, then **maximum likelihood** and **minimum distance** agree.

Proof. Suppose $d(x, c) = r$.

$$P(x \text{ received} \mid c \text{ sent}) = p^r(1-p)^{n-r} = (1-p)^n \left(\frac{p}{1-p} \right)^r$$

Since $p < \frac{1}{2}$, $\frac{p}{1-p} < 1$. So choosing c to maximise $P(x \text{ received} \mid c \text{ sent})$ is the same as choosing c to minimise $d(x, c)$. \square

Note we assumed $p < \frac{1}{2}$, which is not unreasonable.

Example. Suppose codewords 000 and 111 are sent with probabilities $\alpha = \frac{9}{10}$ and $1-\alpha = \frac{1}{10}$ respectively. We use a **BSC** with $p = \frac{1}{4}$. If we receive 110, how should it be decoded? Clearly **minimum distance** and therefore **maximum likelihood** (by **Lemma 2.2**) say decode as 111. For **ideal observer**:

$$\begin{aligned} P(000 \text{ sent} \mid 110 \text{ received}) &= \frac{P(000 \text{ sent}, 110 \text{ received})}{P(110 \text{ received})} \\ &= \frac{\alpha p^2(1-p)}{\alpha p^2(1-p) + (1-\alpha)p(1-p)^2} \\ &= \frac{\alpha p}{\alpha p + (1-\alpha)(1-p)} \\ &= \frac{9/40}{9/40 + 3/40} = \frac{3}{4} \\ P(000 \text{ sent} \mid 110 \text{ received}) &= \frac{(1-\alpha)p^2(1-p)}{(1-\alpha)p^2(1-p) + \alpha p(1-p)^2} \\ &= \frac{(1-\alpha)(1-p)}{(1-\alpha)(1-p) + \alpha p} \\ &= \frac{3/40}{9/40 + 3/40} = \frac{1}{4}. \end{aligned}$$

So the ideal observer rule says decode as 000.

Remark. The **ideal observer** rule is also known as the minimum-error rule. But it does rely on knowing the probabilities of the codewords sent.

From now on, we use **minimum distance** decoding.

Definition. For a **binary code** C ,

- (i) C is **d -error detecting** if changing at most d letters of a codeword cannot give another codeword.
- (ii) C is **e -error correcting** if knowing that the string received has at most e errors is sufficient to determine which codeword was sent.

Example.

1. The repetition code of length n has:

$$C = \{\underbrace{0 \cdots 0}_n, \underbrace{1 \cdots 1}_n\}$$

This is an $[n, 2]$ -code. It is $n - 1$ error detecting, and $\lfloor \frac{n-1}{2} \rfloor$ error correcting. But it has information rate $= \frac{1}{n}$.

2. The simple parity check code of length n (also known as paper tape code). We view $\{0, 1\} = \mathbb{F}_2 = \mathbb{Z}/2\mathbb{Z}$ (i.e. do arithmetic modulo 2).

$$C = \left\{ (x_1, \dots, x_n) \in \{0, 1\}^n \mid \sum_{i=1}^n x_i = 0 \right\}.$$

This is a $[n, 2^{n-1}]$ code. It is 1-error detecting and 0-error correcting. It has information rate $\frac{n-1}{n}$.

Remark. Suppose we change our code $C \subset \{0, 1\}^n$ by using the same permutation to reorder each codeword. This gives a code with the same mathematical properties (e.g. information rate, error detection rate). We say such codes are equivalent.