Part II – Coding and Cryptography

Based on lectures by Dr R. Camina

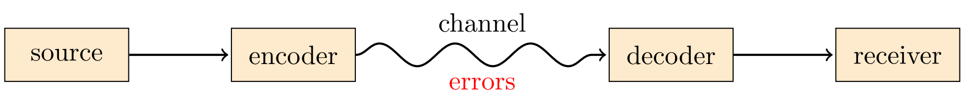
Notes taken by Bhavik Mehta

Lent 2017

Contents

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 $a_1 a_2 \dots a_{10}$ where $a_i \in \{0, 1, \dots, 9\}$ for $1 \le i \le 9$, $a_{10} \in \{0, 1, \dots, 9, X\}$ such that
$$10a_1 + 9a_2 + \ldots + a_{10} \equiv 0 \pmod{11}$$
so errors can be detected (but not corrected). Similarly a 13-digit ISBN has
$$x_1 + 3x_2 + x_3 + 3x_4 + \ldots + 3x_{12} + x_{13} \equiv 0 \pmod{10}$$
for $0 \le x_i \le 10$, 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 form an output alphabet $\Sigma_2 = \{b_1, \dots, b_s\}$.

A channel is determined by the probabilities

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

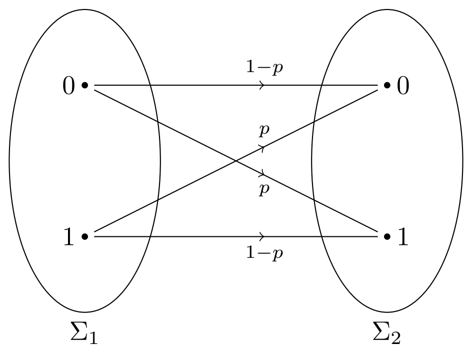
Definition (Discrete memoryless channel)**.** A **discrete memoryless channel** (DMC) is a channel for which

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

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 $\Sigma_1 = \Sigma_2 = \{0, 1\}$, $0 \leq p \leq 1$:

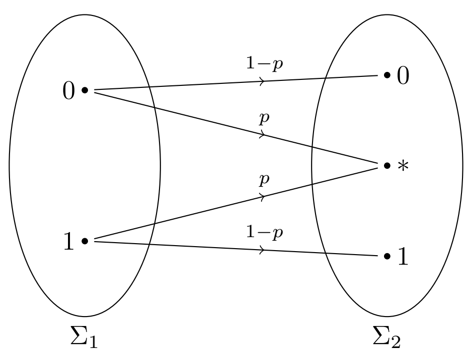


with channel matrix

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

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.

I 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, $|x_1 \dots x_r y_1 \dots y_s| = r + s$, length of string.

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

Example.

1) Greek fire code:

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

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

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

We send a message $x_1, \dots, x_n \in \Sigma_1^*$ as $f(x_1)f(x_2)\cdots f(x_n) \in \Sigma_2^*$ i.e. extend f to $f^* : \Sigma_1^* \to \Sigma_2^*$.

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 $f : \Sigma_1 \to \Sigma_2^*$ 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 (Example 2).

(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 $z \in \Sigma_2^*$.)

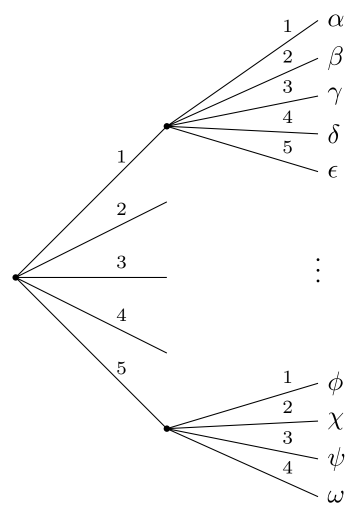
Remark. (i) and (ii) are special cases of (iii).

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 I.1 (Kraft's inequality)**.** Let $|\Sigma_1| = m, |\Sigma_2| = a$. A prefix-free code $f : \Sigma_1 \to \Sigma_2^*$ with word lengths $s_1, \dots, s_m$ exists iff

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

Proof. ($\Rightarrow$) Consider an ∞ tree where each 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 $s_1 \leq s_2 \leq \cdots \leq s_m$. 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 rth 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 I.2 (McMillan)**.** Every decipherable code satisfies Kraft's inequality.

Proof. (Karush) Let $f : \Sigma_1 \to \Sigma_2^*$ 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

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

$$\implies \sum_{i=1}^{m} a^{-s_i} \leq (rs)^{\frac{1}{r}} \to 1 \text{ as } r \to \infty.$$

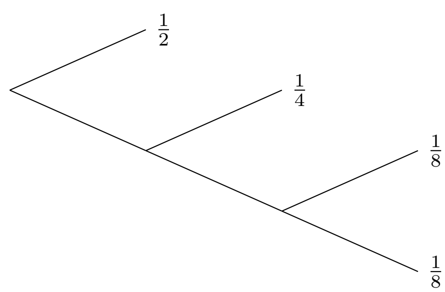
(As $\frac{\log r + \log s}{r} \to 0$ as $r \to \infty$).

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

□

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 $\{HH, HT, TH, HH\}$. $H(X) = 2$.
2. Take $(p_1, p_2, p_3, p_4) = (\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{8})$.



So example 1 is more random than example 2.

Definition (Entropy)**.** The entropy of X:

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

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

Remark.

(i) If $p_i = 0$, we take $p_i \log p_i = 0$.

(ii) $H(x) \geq 0$.

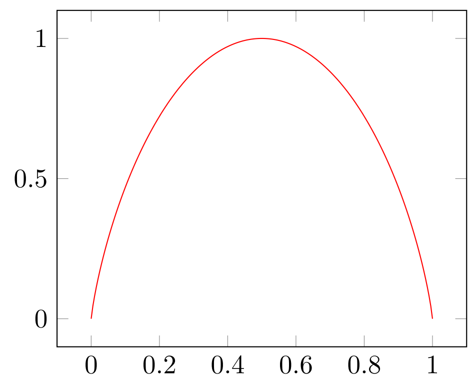
Corollary. A decipherable code with prescribed word lengths exists iff there exists a prefix-free code with the same word lengths.

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

I.1 Mathematical Entropy

Entropy is a measure of 'randomness' or 'uncertainty'. Consider a random variable X taking values $x_1, \dots, x_n$ with probability $p_1, \dots, p_n$ ($\sum p_i = 1, 0 \le p_i \le 1$). 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 entropy=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 I.3 (Gibbs' Inequality)**.** Let $p_1, \dots, p_n$ and $q_1, \dots, q_n$ be probability distributions. Then

$$-\sum p_i \log p_i \le -\sum p_i \log q_i$$

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 $\ln x \le x - 1$, equality iff $x = 1$. Let $I = \{ 1 \le i \le n \mid p_i \ne 0 \}$

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

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

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

Proof. Take $q_1 = q_2 = \ldots = q_n = \frac{1}{n}$ in previous lemma. □

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 $X_1, X_2, \ldots$ 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

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

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

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

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

Theorem I.4 (Shannon's Noiseless Coding Theorem)**.** The minimum expected word length of a decipherable code $f : \Sigma_1 \to \Sigma_2^*$ 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 $\sum q_i = 1$.

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

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

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

So by Theorem I.1, $\exists$ a prefix-free code with word lengths $s_1, \ldots, s_m$. Also,

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

$\square$

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 $s_1, \ldots, s_m$ by taking the s_i in increasing order ensuring that previous codewords are not prefixes. The Kraft inequality ensures there is enough room.

Example. Suppose $\mu_1, \ldots, \mu_5$ are emitted with probabilities $0.4, 0.2, 0.2, 0.1, 0.1$. A 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 $H(X) \approx 2.12$.