Introduction

We model communication:

$$\underbrace{\text{SOURCE}}_{\text{message}} \to \underbrace{\text{ENCODER}}_{\text{codewords}} \xrightarrow[\text{errors,noise}]{\text{CHANNEL}} \underbrace{\text{DECODER}}_{\text{recieved word error correction}} \to \underbrace{\text{RECIEVER}}_{\text{message}}.$$

Examples: optical signals, electrical telegraph, SMS (compression), postcodes, CDs (error correction), zip/gz files (compression).
Given a source and a channel, modelled probabilistically, the basic problem is to design an encoder and decoder to transmit messages economically (noiseless coding; compression) and reliably (noisy coding).

Examples:

- Noiseless coding: Morse code: common letters are assigned shorter codewords, e.g $A \mapsto \bullet -$, $E \mapsto \bullet$, $Q \mapsto - - \bullet -$, $S \mapsto \bullet\bullet\bullet$, $O \mapsto - - -$, $Z \mapsto - - \bullet\bullet$. Noiseless coding is adapted to source.
- Noisy coding: Every book has an ISBN $a_1, a_2, \dots, a_9, a_{10}$, $a_i \in \{0, 1, \dots, 9\}$ for $1 \leq i \leq 9$ and $a_{10} \in \{0, 1, \dots 9, X\}$ with $\sum_{j=1}^{10} ja_j \equiv 0 \pmod{11}$. This detects common errors - e.g one incorrect digit, transposition of two digits. Noisy coding is adapted to the channel.

Plan:

(I) Noiseless coding - entropy

(II) Error correcting codes - noisy channels

(III) Information theory - Shannon's theorems

(IV) Examples of codes

(V) Cryptography

Books: [GP], [W], [CT], [TW], Buchmann, Körner. Online notes: Carne, Körner.

Basic Definitions

Definition (Communication channel)**.** A *communication channel* accepts symbols from a alphabet $\mathcal{A} = \{a_1, \dots, a_r\}$ and it outputs symbols from alphabet $\mathcal{B} = \{b_1, \dots, b_s\}$. Channel modelled by the probabilities $\mathbb{P}(y_1 \dots y_n \text{ recieved} | x_1 \dots x_n \text{sent})$.
A *discrete memoryless channel* (DMC) is a channel with

$$p_{ij} = \mathbb{P}(b_j \text{ recieved} | a_i \text{ sent})$$

the same for each channel use and independent of all past and future uses. The channel matrix is $P = (b_{ij})$, a $r \times s$ stochastic matrix.

Definition (Binary symmetric channel)**.** The *binary symmetric channel* (BSC) with error probability $p \in [0, 1)$ from $\mathcal{A} = \mathcal{B} = \{0, 1\}$. The channel matrix is

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

A symbol is transmitted correctly with probability $1 - p$. Usually assume $p < 1/2$.

The *binary erasure channel* (BEC) has $\mathcal{A} = \{0, 1\}$, $\mathcal{B} = \{0, 1, *\}$. The channel matrix is

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

So $p = \mathbb{P}(\text{symbol can't be read})$.

Definition. We model n uses of a channel by the nth extension, with input alphabet $\mathcal{A}^n$ and output alphabet $\mathcal{B}^n$. A *code C of length n* is a function $\mathcal{M} \to \mathcal{A}^n$ where $\mathcal{M}$ is the set of possible messages. Implicitly we also have a decoding rule $\mathcal{B}^n \to \mathcal{M}$. The *size* of C is $m = |\mathcal{M}|$. The *information rate* is $\rho(C) = \frac{1}{n} \log_2 m$. The *error rate* is $\hat{e}(C) = \max_{x \in \mathcal{M}} \mathbb{P}(\text{error}|x \text{ sent})$.

Remark. For the remainder of the course we write log instead of $\log_2$.

Definition. A channel can *transmit reliably at rate R* if there exists $(C_n)_{n=1}^{\infty}$ with each C_n a code of length n such that

$$\lim_{n \to \infty} \rho(C_n) = R \ \& \ \lim_{n \to \infty} \hat{e}(C_n) = 0.$$

The *capacity* is the supremum of all reliable transmission rates. We'll see in Chapter 9 that a BSC with error probability $p < 1/2$ has non-zero capacity.

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 $xy = x_1 \dots x_r y_1 \dots y_s$.

Definition. Let $\mathcal{A}, \mathcal{B}$ be alphabets. A code is a function $c : \mathcal{A} \to \mathcal{B}^*$. The strings $c(a)$ for $a \in \mathcal{A}$ are called *codewords* or *words* (CWS).

Example 1.1 (Greek fire code)**.** $\mathcal{A} = \{\alpha, \beta, \dots, \omega\}$ (greek alphabet), $\mathcal{B} = \{1, 2, 3, 4, 5\}$., $c : \alpha \mapsto 11, \beta \mapsto 12, \dots, \psi \mapsto 53$, $\omega \mapsto 54$. xy means hold up x torches and another y torches nearby.

Example 1.2. $\mathcal{A}$ = words in a dictionary, $\mathcal{B} = \{A, B, \dots, Z, \omega\}$. $c : \mathcal{A} \to \mathcal{B}$ splits the word and follows with a space. Send 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 $c^* : \mathcal{A}^* \to \mathcal{B}^*$.

Definition. c is said to be *decipherable* if the induced map c^* (as in the previous example) is injective. In other words, each string from $\mathcal{B}$ corresponds to at most one message.

Clearly if c is decipherable, it is necessary for c to be injective. However it is not sufficient:

Example 1.3. $\mathcal{A} = \{1, 2, 3, 4\}$, $\mathcal{B} = \{0, 1\}$. Define $c : 1 \mapsto 0$, $2 \mapsto 1$, $3 \mapsto 00$, $4 \mapsto 01$. Then $c^*(114) = 0001 = c^*(312) = c^*(144)$ yet c is injective.

Notation: $|\mathcal{A}| = m$, $|\mathcal{B}| = a$, call c am a-ary code of size m. For example a 2-ary code is a binary one, and a 3-ary code is a ternary code.

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

(i) A <u>block code</u> has all codewords of the same length (e.g Greek fire code);

(ii) A <u>comma code</u> reserves a letter from $\mathcal{B}$ to signal the end of a word (e.g Example 1.2);

(iii) A <u>prefix-free code</u> is a code where no codeword is a prefix of any other distinct word (if $x, y \in \mathcal{B}^*$ then x is a prefix of y if $y = xz$ for some string $z \in \mathcal{B}^*$).

(i) and (ii) are special cases of (iii). As we can decode the message as it is recieved, prefix-free codes are sometimes called *instantaneous.*

Exercise: find a decipherable code which is not prefix-free.

Definition (Kraft's inequality)**.** $|\mathcal{A}| = m$, $|\mathcal{B}| = a$, $c : \mathcal{A} \to \mathcal{B}^*$ has word lengths $l_1, \dots, l_m$. Then Kraft's inequality is

$$\sum_{i=1}^{m} a^{-l_i} \leq 1. \tag{$*$}$$

Theorem 1.1. *A prefix-free code exists if and only if Kraft's inequality* $(*)$ *holds.*

Proof. Rewrite $(*)$ as

$$\sum_{l=1}^{s} n_l a^{-l} \leq 1, \tag{$**$}$$

where n_l is the number of codewords with length l, and $s = \max_{1 \le i \le m} l_i$.

Now if $c : \mathcal{A} \to \mathcal{B}^*$ is prefix-free,

$$n_1 a^{s-1} + n_2 a^{s-2} + \ldots + n_{s-1} a + n_a \le a^s.$$

Indeed the LHS is the number of strings of length s in B with some codeword of c as a prefix, and the RHS is the total number of strings of length S. Dividing through by a^s we get $(**)$.

Now given $n_1, \ldots, n_s$ satisfying $(**)$, we try to construct a prefix-free code c with n_l codewords of length l, $\forall l \le s$. Proceed by induction on s, $s = 1$ is clear (since $(**)$ gives $n_1 \le a$ so can construct code).

By the induction hypothesis, there exists a prefix-code $\hat{c}$ with n_l codewords of length l for all $l \le s-1$. Then $(**)$ implies

$$n_1 a^{s-1} + n_2 a^{s-2} + \ldots n_{s-1} a + n_s \le a^s.$$

The first $s-1$ terms on the LHS sum to the number of strings of length s with a 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.

$\square$

Remark. This proof is constructive: just choose codewords in order of increasing length, ensuring that no previous codeword is a prefix.

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

Proof (Karush, 1961). Let $c : \mathcal{A} \to \mathcal{B}^*$ be a decipherable code with word lengths $l_1, \ldots, l_m$. Set $s = \max_{1 \le i \le m} l_i$. For $R \in \mathbb{N}$

$$\left(\sum_{i=1}^{m} a^{-l_i} \right)^R = \sum_{l=1}^{Rs} b_l a^{-l}, \tag{$\dagger$}$$

where b_l is the number of ways of choosing R codewords of total length l. Since c is decipherable, any string of length l formed from codewords must correspond to at most one sequence of codewords, i.e $b_l \le |\mathcal{B}^l| = a^l$. Subbing this into $(\dagger)$

$$\left(\sum_{i=1}^{m} a^{-l_i} \right)^R \le \sum_{i=1}^{Rs} a^l a^{-l} = Rs,$$

so

$$\sum_{i=1}^{m} a^{-l_i} \le (Rs)^{1/R} \to 1 \text{ as } R \to \infty.$$

Hence $\sum_{i=1}^{m} a^{-l_i} \le 1$. $\square$

Corollary 1.3. *A decipherable code with prescribed word lengths exists if and only if a prefix-free code with the same word lengths exists.*

Proof. Combine previous two theorems. □

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

2 Shannon's Noiseless Coding Theorem

Entropy is a measure of 'randomness' or 'uncertainty'. Suppose we have a random variable X taking a finite set of values $x_1, \dots, x_n$ with probabilities $p_1, \dots, p_n$ respectively. The *entropy* $H(X)$ of X is the expected number of fair coin tosses needed to simulate X (roughly speaking).

Example 2.1. Suppose $p_1 = p_2 = p_3 = p_4 = 1/4$. Identify (x_1, x_2, x_3, x_4) with (HH, HT, TH, TT). Then the entropy is 2.

Example 2.2. Suppose $(p_1, p_2, p_3, p_4) = (1/2, 1/4, 1/8, 1/8)$. Identify (x_1, x_2, x_3, x_4) with (H, TH, TTH, TTT). Then the entropy is

$$\frac{1}{2} \times 1 + \frac{1}{4} \times 2 + \frac{1}{8} \times 3 + \frac{1}{8} \times 3 = \frac{7}{4}.$$

In a sense, the previous example (2.1) was 'more random' than this.

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

$$H(X) = -\sum_{i=1}^{b} p_i \log p_i.$$

(Recall that $\log =: \log_2$ here.) Note $H(X) \geq 0$. It is measured in *bits* (binary digits). Conventionally, we take $0 \log 0 = 0$.

Example 2.3. Take a biased coin $\mathbb{P}(H) = p$, $\mathbb{P}(T) = 1-p$. Write $H(p, 1-p) := H(p)$. Then

$$H(p) = -p \log p - (1-p) \log(1-p).$$

Note that $H'(p) = \log \frac{1-p}{p}$. Hence the entropy is maximised for $p = 1/2$ (giving entropy 1).

Proposition 2.1 (Gibbs' inequality)**.** *Let $(p_1, \dots, p_n), (q_1, \dots, q_n)$ be probability distributions. Then*

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

(The RHS is sometimes called the cross entropy or mixed entropy) Furthermore we have equality iff $p_i = q_i$ for all i.

Proof. Since $\log x = \frac{\ln x}{\ln 2}$, we may replace log with ln. Put $I = \{1 \leq i \leq n : p_i \neq 0\}$. Now $\ln x = x - 1$ for all $x > 0$ with equality iff $x = 1$. Hence $\ln \frac{q_i}{p_i} \leq \frac{q_i}{p_i} - 1$ for all $i \in I$. So

$$\sum_{i \in I} p_i \ln \frac{q_i}{p_i} \leq \underbrace{\sum_{i \in I} q_i}_{\leq 1} - \underbrace{\sum_{i \in I} p_i}_{=1} \leq 0$$

$$\implies -\sum_{i \in I} p_i \ln p_i \leq -\sum_{i \in I} p_i \ln q_i$$

$$\implies -\sum_{i=1}^{n} p_i \ln p_i \leq -\sum_{i=1}^{n} p_i \ln q_i.$$

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

Corollary 2.2. *$H(p_1, p_2, \ldots, p_n) \leq \log n$ with equality iff $p_1 = p_2 = \ldots = p_n = 1/n$.*

Proof. Take $q_1 = q_2 = \ldots = q_n = 1/n$ in Gibbs' inequality. $\square$

Let $\mathcal{A} = \{\mu_1, \ldots, \mu_m\}$, $|\mathcal{B}| = a$ $(m, n \geq 2)$. The random variable X takes values $\mu_1, \ldots, \mu_m$ with probabilities $p_1, \ldots, p_m$.

Definition. If $c : \mathcal{A} \to \mathcal{B}^*$ is a code, we say it is *optimal* if has the smallest possible expected word length. i.e $\mathbb{E}S := \sum_{i=1}^{n} p_i l_i$ is minimal amongst all decipherable codes.

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

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

Remark. The lower bound is actually true for any decipherable code.

Proof. We first get the lower bound. Let $c : \mathcal{A} \to \mathcal{B}^*$ be decipherable with word lengths $l_1, \ldots, l_m$. Let $q_i = \frac{a^{-l_i}}{D}$ where $D = \sum_{i=1}^{m} a^{-l_i}$. Note $\sum_{i=1}^{m} q_i = 1$. By Gibbs' inequality

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

By McMillan, $D \leq 1$ so $\log D \leq 0$. Hence

$$H(X) \leq \left(\sum_{i=1}^{m} p_i l_i\right) \log a \implies \frac{H(X)}{\log a} \leq \mathbb{E}S.$$

And we have equality iff $p_i = a^{-l_i}$ for some integers $l_1, \ldots, l_m$. Note we have only used decipherability so far.

Now we get the upper bound. Take $l_i = \lceil -\log_a p_i \rceil$. Then

$$-\log_a p_i \leq l_i < -\log_a p_i + 1.$$

Hence $\log_a p_i \geq -l_i$, so $p_i \geq a^{-l_i}$. Therefore $\sum_{i=1}^m a^{-l_i} \leq \sum_{i=1}^m p_i = 1$. By Kraft's inequality, there exists a prefix-free code c with word lengths $l_1, \ldots, l_m$. c has expected word length

$$\mathbb{E}S = \sum_{i=1}^m p_i l_i < \sum_{i=1}^m p_i(-\log_a p_i + 1) = \frac{H(X)}{\log a} + 1.$$

$\square$

Example 2.4 (Shannon-Fano Coding)**.** We mimic the above proof: given $p_1, \ldots, p_m$, set $l_i = \lceil -\log_a p_i \rceil$. Construct a prefix-free code with word lengths l_i by choosing codewords in order of increasing length, ensuring any new codeword has no previous codeword as a prefix (Kraft's inequality ensures we can do this).

Example 2.5. Take $a = 2, m = 5$.

i	p_i	$\lceil -\log_2 p_i \rceil$	
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

Then $\mathbb{E}S = \sum_{i=1}^m p_i l_i = 2.8$, $H = H/\log a = 2.12$. [See also Carne p13.]