Coding and Cryptography

Cambridge University Mathematical Tripos: Part II

21st May 2024

Contents

1 Modelling communication

To reason about communication, we use the following model. We have a *source* which knows a message, that uses an *encoder* to produce some *code words*. The code words are sent through a *channel*, but errors and noise may be introduced in this channel. The code words are received by a *decoder*, which performs some form of error detection and correction. The message is finally received by a *receiver*.

The source is often named *Alice*, and the receiver is named *Bob*. There may be an agent watching the channel called *Eve*, short for eavesdropper.

Examples of these ideas include the optical and electrical telegraph, SMS, postcodes, CDs and their error correction, compression algorithms such as `gzip`, and PINs.

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).

An example of noiseless coding is Morse code, where every letter is assigned a unique sequence of dots and dashes, where more common letters are assigned shorter strings. Noiseless coding is adapted to the source.

Here is an example of noisy coding. Each book has an ISBN $a_1 a_2 \dots a_9 a_{10}$ where the $a_1, \dots, a_9$ are digits in $\{0, \dots, 9\}$, and $a_{10} \in \{0, \dots, 9, X\}$ such that $11 \mid \sum_{j=1}^{10} j a_j$. This coding system detects the common human errors of writing an incorrect digit and transposing two adjacent digits. Noisy coding is adapted to the channel, which in this case is the human reading the number and typing it into a computer.

Definition. A *communication channel* accepts a string of symbols from a finite alphabet $\mathcal{A} = \{a_1, \dots, a_r\}$ and outputs a string of symbols from another finite alphabet $\mathcal{B} = \{b_1, \dots, b_s\}$. It is modelled by the probabilities $\mathbb{P}(y_1 \dots y_n \text{ received} \mid x_1 \dots x_n \text{ sent})$.

Definition. A *discrete memoryless channel* is a channel where $p_{ij} = \mathbb{P}\left(b_j \text{ received} \mid a_i \text{ sent}\right)$ are the same for each channel use, and independent of all past and future uses of the channel. Its *channel matrix* is the $r \times s$ stochastic matrix $P = (p_{ij})$.

Example. The *binary symmetric channel* with error probability $p \in [0, 1]$ is a discrete memoryless channel with input and output alphabets $\{0, 1\}$, where the channel matrix is

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

Here, a symbol is transmitted correctly with probability $1 - p$. Usually, we assume $p < \frac{1}{2}$.

Example. The *binary erasure channel* has $\mathcal{A} = \{0, 1\}$ and $\mathcal{B} = \{0, 1, \star\}$. The channel matrix is

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

p can be interpreted as the probability that the symbol received is unreadable. If $\star$ is received, we say that we have received a *splurge error*.

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 a set of messages. Implicitly, we also have a decoding rule $\mathcal{B}^n \to \mathcal{M}$.

- The *size* of this code is $m = |\mathcal{M}|$.
- The *information rate* of the code is $\rho(C) = \frac{1}{n} \log_2 m$.
- The *error rate* of the code is $\hat{e}(C) = \max_{x \in \mathcal{M}} \mathbb{P}(\text{error} \mid x \text{ sent})$.

Definition. A channel can *transmit reliably at rate R* if there is a sequence of codes $(C_n)_{n=1}^{\infty}$ with each C_n a code of length n such that $\lim_{n\to\infty} \rho(C_n) = R$ and $\lim_{n\to\infty} \hat{e}(C_n) = 0$. The *capacity* of a channel is the supremum of all reliable transmission rates.

It is a nontrivial fact that the capacity of the binary symmetric channel with $p < \frac{1}{2}$ is nonzero. This is one of Shannon's theorems, proven later.

2 Noiseless coding

2.1 Prefix-free codes

Let $\mathcal{A}$ be a finite alphabet. We write $\mathcal{A}^\star$ for the set of strings of elements of $\mathcal{A}$, defined by $\mathcal{A}^\star = \bigcup_{n\geq 0} \mathcal{A}^n$. The *concatenation* of two strings $x = x_1 \dots x_r$ and $y = y_1 \dots y_s$ is the string $xy = x_1 \dots x_r y_1 \dots y_s$.

Definition. Let $\mathcal{A}, \mathcal{B}$ be alphabets. A *code* is a function $c : \mathcal{A} \to \mathcal{B}^\star$. The *codewords* of c are the elements of $\operatorname{Im} c$.

Example (Greek fire code). Let $\mathcal{A} = \{\alpha, \beta, \dots, \omega\}$, and $B = \{1, 2, 3, 4, 5\}$. We map $c(\alpha) = 11, c(\beta) = 12, \dots, c(\psi) = 53, c(\omega) = 54$. xy means to hold up x torches and another y torches nearby. This code was described by the historian Polybius.

Example. Let $\mathcal{A}$ be a set of words in some dictionary. Let $\mathcal{B}$ be the letters of English $\{A, \dots, Z, \_\}$ The code is to spell the word and follow with a space.

The general idea is to send a message $x_1, \dots, x_n \in \mathcal{A}^\star$ as $c(x_1) \dots c(x_n) \in \mathcal{B}^\star$. So c extends to a function $c^\star : \mathcal{A}^\star \to \mathcal{B}^\star$.

Definition. A code c is *decipherable* (or *uniquely decodable*) if $c^\star$ is injective.

If c is decipherable, each string in $\mathcal{B}^\star$ corresponds to at most one message. It does not suffice to require that c be injective. Consider $\mathcal{A} = \{1, 2, 3, 4\}, \mathcal{B} = \{0, 1\}$, and let $c(1) = 0, c(2) = 1, c(3) = 00, c(4) = 01$. Then $c^\star(114) = 0001 = c^\star(312)$.

Typically we define $m = |\mathcal{A}|$ and $a = |\mathcal{B}|$. We say c is an a-ary code of size m. A 2-ary code is a binary code, and a 3-ary code is a ternary code. We aim to construct decipherable codes with short word lengths. Assuming that c is injective, the following codes are always decipherable.

(i) a *block code*, where all codewords have the same length, such as in the Greek fire code;

(ii) a *comma code*, which reserves a letter from $\mathcal{B}$ to signal the end of a word;

(iii) a *prefix-free code*, a code in which no codeword is a prefix of another codeword.

Block codes and comma codes are examples of prefix-free codes. Such codes require no lookahead to determine if we have reached the end of a word, so such codes are sometimes called *instantaneous* codes. One can easily find decipherable codes that are not prefix-free.

2.2 Kraft's inequality

Definition. Let $\mathcal{A}$ be an alphabet of size m, and $\mathcal{B}$ be an alphabet of size a. Let $c : \mathcal{A} \to \mathcal{B}^\star$ be a code with codewords are of length $\ell_1, \dots, \ell_m$. Then, *Kraft's inequality* is

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

Theorem. A prefix-free code (with given codeword lengths) exists if and only if Kraft's inequality holds.

Proof. Let us rewrite Kraft's inequality as $\sum_{\ell=1}^{s} n_\ell a^{-\ell} \leq 1$, where n_ℓ is the number of codewords of length ℓ, and s is the length of the longest codeword. Suppose $c : \mathcal{A} \to \mathcal{B}^\star$ is prefix-free. Then,

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

since the left hand side counts the number of strings of length s in $\mathcal{B}$ with some codeword of c as a prefix, and the right hand side counts the total number of strings of length s. Dividing by a^s gives the desired result.

Now, suppose that $\sum_{\ell=1}^{s} n_\ell a^{-\ell} \leq 1$. We aim to construct a prefix-free code c with n_ℓ codewords of length ℓ for all $\ell \leq s$. Proceed by induction on s. The case $s = 1$ is clear; in this case, the inequality gives $n_1 \leq a$. By the inductive hypothesis, we have constructed a prefix-free code $\hat{c}$ with n_ℓ codewords of length ℓ for all $\ell < s$. The inequality gives $n_1 a^{s-1} + \dots + n_{s-1} a + n_s \leq a^s$. The first $s - 1$ terms on the left hand side gives the number of strings of length s with some codeword of $\hat{c}$ as a prefix. So we are free to add n_s additional codewords of length s to $\hat{c}$ to form c without exhausting our supply of a^s total strings of length s. □

Remark. The proof of existence of such a code is constructive; one can choose codewords in order of increasing length, ensuring that we do not introduce prefixes at each stage.

2.3 McMillan's inequality

Theorem. Any decipherable code satisfies Kraft's inequality.

Proof. Let $c : \mathcal{A} \to \mathcal{B}^\star$ be decipherable with word lengths $\ell_1, \dots, \ell_m$. Let $s = \max_{i \leq m} \ell_i$. For $R \in \mathbb{N}$, we have

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

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 exactly one sequence of codewords. Hence, $b_\ell \leq |\mathcal{B}^\ell| = a^\ell$. The inequality therefore gives

$$\left(\sum_{i=1}^{m} a^{-\ell_i}\right)^R \leq Rs \implies \sum_{i=1}^{m} a^{-\ell_i} \leq (Rs)^{\frac{1}{R}}$$

As $R \to \infty$, the right hand side converges to 1, giving Kraft's inequality as required. □

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

We can therefore restrict our attention to prefix-free codes.

2.4 Entropy

Entropy is a measure of 'randomness' or 'uncertainty' in an input message. Suppose that we have a random variable X taking a finite number of values $x_1, \dots, x_n$ with probability $p_1, \dots, p_n$. Then, the entropy of this random variable is the expected number of fair coin tosses required to determine X.

Example. Suppose $p_1 = p_2 = p_3 = p_4 = \frac{1}{4}$. Identifying $\{x_1, x_2, x_3, x_4\} = \{00, 01, 10, 11\}$, we would expect $H(X) = 2$.

Example. Suppose $p_1 = \frac{1}{2}$, $p_2 = \frac{1}{4}$, and $p_3 = p_4 = \frac{1}{8}$. Identifying $\{x_1, x_2, x_3, x_4\} = \{0, 10, 110, 111\}$, we obtain $H(X) = \frac{1}{2} \cdot 1 + \frac{1}{4} \cdot 2 + \frac{1}{8} \cdot 3 + \frac{1}{8} \cdot 3 = \frac{7}{4}$.

In a sense, the first example is 'more random' than the second, as its entropy is higher.

Definition. The *entropy* of a random variable X taking a finite number of values $x_1, \dots, x_n$ with probabilities $p_1, \dots, p_n$ is defined to be

$$H(X) = H(p_1, \dots, p_n) = -\sum_{i=1}^{n} p_i \log p_i = -\mathbb{E}\left[\log p_i\right]$$

where the logarithm is taken with base 2.

Note that $H(X) \geq 0$, and equality holds exactly when X is constant with probability 1. It is measured in *bits*, binary digits. By convention, we write $0 \log 0 = 0$ (note that $x \log x \to 0$ as $x \to 0$).

Example. For a biased coin with probability p of a head, we write $H(p, 1-p) = H(p)$. We find

$$H(p) = -p \log p - (1-p)\log(1-p); \quad H'(p) = \log \frac{1-p}{p}$$

This graph is concave, taking a maximum value of 1 when $p = \frac{1}{2}$. If $p = 0, 1$ then $H(p) = 0$.

2.5 Gibbs' inequality

Proposition. Let $(p_1, \dots, p_n), (q_1, \dots, q_n)$ be discrete probability distributions. Then,

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

with equality if and only if $p_i = q_i$.

The right hand side is sometimes called the *cross entropy*, or *mixed entropy*.

Proof. Since $\log x = \frac{\ln x}{\ln 2}$, we may replace the inequality with

$$-\sum p_i \ln p_i \leq -\sum p_i \ln q_i$$

Define $I = \{i \mid p_i \neq 0\}$. Now, $\ln x \leq x - 1$ for all $x > 0$, with equality if and only if $x = 1$. Hence, $\ln \frac{q_i}{p_i} \leq \frac{q_i}{p_i} - 1$ for all $i \in I$. Then,

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

As the p_i form a probability distribution, $\sum_{i \in I} p_i = 1$ and $\sum_{i \in I} q_i \leq 1$, so the right hand side is at most 0. Therefore,

$$-\sum_{i=1}^{n} p_i \ln p_i = -\sum_{i \in I} p_i \ln p_i \leq -\sum_{i \in I} p_i \ln q_i \leq -\sum_{i=1}^{n} p_i \ln q_i$$

If equality holds, we must have $\sum_{i \in I} q_i = 1$ and $\frac{q_i}{p_i} = 1$ for all $i \in I$, giving that $p_i = q_i$ for all i. □

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

2.6 Optimal codes

Let $\mathcal{A} = \{\mu_1, \dots, \mu_m\}$ be an alphabet of $m \geq 2$ messages, and let $\mathcal{B}$ be an alphabet of length $a \geq 2$. Let X be a random variable taking values in A with probabilities $p_1, \dots, p_m$.

Definition. A code $c \colon \mathcal{A} \to \mathcal{B}^\star$ is called *optimal* if it has the smallest possible expected word length $\sum p_i \ell_i = \mathbb{E}[S]$ among all decipherable codes.

Theorem (Shannon's noiseless coding theorem)**.** The expected word length $\mathbb{E}[S]$ of a decipherable code satisfies

$$\overbrace{\frac{H(X)}{\log a} \leq \mathbb{E}[S]}^{\text{for decipherable codes}} \underbrace{< \frac{H(X)}{\log a} + 1}_{\text{for optimal codes}}$$

Moreover, the left hand inequality is an equality if and only if $p_i = a^{-\ell_i}$ with $\sum a^{-\ell_i} = 1$ for some integers $\ell_1, \dots, \ell_m$.

Proof. First, we consider the lower bound. Let $c : \mathcal{A} \to \mathcal{B}^\star$ be a decipherable code with word lengths $\ell_1, \dots, \ell_m$. Let $q_i = \frac{a^{-\ell_i}}{D}$ where $D = \sum a^{-\ell_i}$, so $\sum q_i = 1$. By Gibbs' inequality,

$$H(X) \leq -\sum p_i \log q_i = -\sum p_i(-\ell_i \log a - \log D) = \log D + \log a \sum p_i \ell_i$$

By McMillan's inequality, $D \leq 1$ so $\log D \leq 0$. Hence, $H(X) \leq \log a \sum p_i \ell_i = \log a \mathbb{E}[S]$ as required. Equality holds exactly when $D = 1$ and $p_i = q_i = \frac{a^{-\ell_i}}{D} = a^{-\ell_i}$ for some integers $\ell_1, \dots, \ell_m$.

Now, consider the upper bound. We construct a code called the *Shannon–Fano code*. Let $\ell_i = \lceil -\log_a p_i \rceil$, so $-\log_a p_i \leq \ell_i < -\log_a p_i + 1$. Therefore, $\log_a p_i \geq -\ell_i$, so $p_i \geq a^{-\ell_i}$. Thus, Kraft's inequality $\sum a^{-\ell_i} \leq 1$ is satisfied, so there exists a prefix-free code c with these word lengths $\ell_1, \dots, \ell_m$. c has expected word length

$$\mathbb{E}[S] = \sum p_i \ell_i < \sum p_i(-\log p_i + 1) = \frac{H(X)}{\log a} + 1$$

as required. □

Example (Shannon–Fano coding). For probabilities $p_1, \dots, p_m$, we set $\ell_i = \lceil -\log_a p_i \rceil$. Construct a prefix-free code with these word lengths by choosing codewords in order of size, with smallest codewords being selected first to ensure that the prefix-free property holds. By Kraft's inequality, this process can always be completed.

Example. Let $a = 2, m = 5$, and define

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

Here, $\mathbb{E}[S] = \sum p_i \ell_i = 2.8$, and $H(X) = \frac{H(X)}{\log 2} \approx 2.12$. Clearly, this is not optimal; one could take $c(4) = 100, c(5) = 101$ to reduce the expected word length.

2.7 Huffman coding

Let $\mathcal{A} = \{\mu_1, \dots, \mu_m\}$ and $p_i = \mathbb{P}(X = \mu_i)$. We assume $a = 2$ and $\mathcal{B} = \{0, 1\}$ for simplicity. Without loss of generality, we can assume $p_1 \geq p_2 \geq \dots \geq p_m$. We construct an optimal code inductively.

If $m = 2$, we take codewords 0 and 1. If $m > 2$, first we take the Huffman code for messages $\mu_1, \dots, \mu_{m-2}, \nu$ with probabilities $p_1, p_2, \dots, p_{m-2}, p_{m-1} + p_m$. Then, we append 0 and 1 to the codeword for ν to obtain the new codewords for μ_{m-1}, μ_m.

Remark. By construction, Huffman codes are prefix-free. In general, Huffman codes are not unique; we require a choice if $p_i = p_j$.

Example. Consider the example Let $a = 2, m = 5$, and consider as before

i	p_i
1	0.4
2	0.2
3	0.2
4	0.1
5	0.1

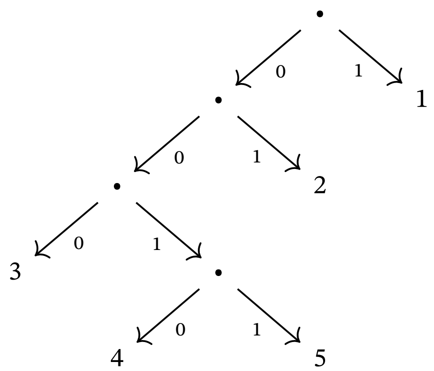
Merging 4 and 5, as they have the lowest probabilities,

i	p_i
1	0.4
2	0.2
3	0.2
45	0.2

Continuing, we obtain

i	p_i
(3(45))2	0.6
1	0.4

giving codewords



This gives $\mathbb{E}[S] = 2.2$, better than the Shannon–Fano code found above.

Lemma. Let $\mu_1, \dots, \mu_m$ be messages in $\mathcal{A}$ with probabilities $p_1, \dots, p_m$. Let c be an optimal prefix-free code for c with word lengths $\ell_1, \dots, \ell_m$. Then,

(i) if $p_i > p_j$, $\ell_i \le \ell_j$; and
(ii) among all codewords of maximal length, there exist two which differ only in the last digit.

Proof. If this were not true, one could modify c by

(i) swapping the ith and jth codewords; or
(ii) deleting the last letter of each codeword of maximal length

which yields a prefix-free code with strictly smaller expected word length. □

Theorem. Huffman codes are optimal.

Proof. The proof is by induction on m. If $m = 2$, then the codewords are 0 and 1, which is clearly optimal. Assume $m > 2$, and let c_m be the Huffman code for X_m which takes values $\mu_1, \dots, \mu_m$ with probabilities $p_1 \geq \dots \geq p_m$. c_m is constructed from a Huffman code c_{m-1} with random variable X_{m-1} taking values $\mu_1, \dots, \mu_{n-2}, \nu$ with probabilities $p_1, \dots, p_{m-2}, p_{m-1} + p_m$. The code c_{m-1} is optimal by the inductive hypothesis. The expected word length $\mathbb{E}[S_m]$ is given by

$$\mathbb{E}[S_m] = \mathbb{E}[S_{m-1}] + p_{m-1} + p_m$$

Let c_m' be an optimal code for X_m, which without loss of generality can be chosen to be prefix-free. Without loss of generality, the last two codewords of c_m' can be chosen to have the largest possible length and differ only in the final position, by the previous lemma. Then, $c_m'(\mu_{m-1}) = y0$ and $c_m'(\mu_m) = y1$ for some $y \in \{0,1\}^\star$. Let c_{m-1}' be the prefix-free code for X_{m-1} given by

$$c_{m-1}'(\mu_i) = \begin{cases} c_m'(\mu_i) & i \leq m-2 \\ y & i = m-1, m \end{cases}$$

The expected word length satisfies

$$\mathbb{E}[S_m'] = \mathbb{E}[S_{m-1}'] + p_{m-1} + p_m$$

By the inductive hypothesis, c_{m-1} is optimal, so $\mathbb{E}[S_{m-1}] \leq \mathbb{E}[S_{m-1}']$. Combining the equations,

$$\mathbb{E}[S_m] \leq \mathbb{E}[S_m']$$

So c_m is optimal as required. □

Remark. Not all optimal codes are Huffman codes. However, we have proven that, given a prefix-free optimal code with prescribed word lengths, there is a Huffman code with these word lengths.

2.8 Joint entropy

Let X, Y be random variables with values in $\mathcal{A}, \mathcal{B}$. Then, the pair (X, Y) is also a random variable, taking values in $\mathcal{A} \times \mathcal{B}$. This has entropy $H(X, Y)$, called the *joint entropy* for X and Y.

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

This construction generalises to finite tuples of random variables.

Lemma. 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 $\mathcal{A} = \{x_1, \dots, x_m\}$ and $\mathcal{B} = \{y_1, \dots, y_n\}$. Let $p_{ij} = \mathbb{P}(X = x_i, Y = y_j)$, $p_i = \mathbb{P}(X = x_i)$, and

$q_j = \mathbb{P}\left(Y = y_j\right)$. By Gibbs' inequality applied to $\{p_{ij}\}$ and $\{p_i q_j\}$,

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

Equality holds if and only if $p_{ij} = p_i q_j$ for all i, j, or equivalently, if X, Y are independent. □

3 Noisy channels

3.1 Decoding rules

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

We use an $[n,m]$-code to send one of m messages through a channel using n bits. For instance, if the channel is a binary symmetric channel, we use the channel n times. Note that $1 \leq m \leq 2^n$, so the information rate $\rho(C) = \frac{1}{n}\log m$ satisfies $0 \leq \rho(C) \leq 1$. If $m = 1$, $\rho(C) = 0$, and if $C = \{0,1\}^n$, $\rho(C) = 1$.

Definition. Let $x, y \in \{0,1\}^n$. The *Hamming distance* between x and y is

$$d(x,y) = |\{i \mid x_i \neq y_i\}|$$

In this section, we consider only the binary symmetric channel with probability p.

Definition. Let C be a binary $[n,m]$-code.

- The *ideal observer* decoding rule decodes $x \in \{0,1\}^n$ as the $c \in C$ maximising the probability that c was sent given that x was received;
- The *maximum likelihood* decoding rule decodes $x \in \{0,1\}^n$ as the $c \in C$ maximising the probability that x was received given that c was sent;
- The *minimum distance* decoding rule decodes $x \in \{0,1\}^n$ as the $c \in C$ minimising the Hamming distance $d(x,c)$.

Lemma. Let C be a binary $[n,m]$-code.

(i) If all messages are equally likely, the ideal observer and maximum likelihood decoding rules agree.

(ii) If $p < \frac{1}{2}$, then the maximum likelihood and minimum distance decoding rules agree.

Note that the hypothesis in part (i) is reasonable if we first encode a message using noiseless coding. The hypothesis in part (ii) is reasonable, since a channel with $p = \frac{1}{2}$ can carry no information, and a channel with $p > \frac{1}{2}$ can be used as a channel with probability $1-p$ by inverting its outputs. Channels with $p = 0$ are called *lossless channels*, and channels with $p = \frac{1}{2}$ are called *useless channels*.

Proof. Part (i). By Bayes' rule,

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

By hypothesis, $\mathbb{P}(c \text{ sent})$ is independent of c. Hence, for some fixed received message x, maximising $\mathbb{P}(c \text{ sent} \mid x \text{ received})$ is the same as maximising $\mathbb{P}(x \text{ received} \mid c \text{ sent})$.

Part (ii). Let $r = d(x, c)$. Then,

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

As $p < \frac{1}{2}$, $\frac{p}{1-p} < 1$. Hence, maximising $\mathbb{P}(x \text{ received} \mid c \text{ sent})$ is equivalent to minimising $r = d(x, c)$. ☐

We can therefore choose to use minimum distance decoding from this point.

Example. Suppose codewords $000, 111$ are sent with probabilities $\alpha = \frac{9}{10}$ and $1 - \alpha = \frac{1}{10}$, through a binary symmetric channel with error probability $p = \frac{1}{4}$. Suppose that we receive 110. Clearly, an error has been introduced.

$$\mathbb{P}(000 \text{ sent} \mid 110 \text{ received}) = \frac{\alpha p^2(1-p)}{\alpha p^2(1-p) + (1-\alpha)p(1-p)^2} = \frac{3}{4}$$
$$\mathbb{P}(111 \text{ sent} \mid 110 \text{ received}) = \frac{1}{4}$$

The ideal observer therefore decodes 110 as 000. The maximum likelihood or minimum distance decoding rules decode 110 as 111.

Remark. Minimum distance decoding may be expensive in terms of time and storage if $|C|$ is large, since the distance to all codewords must be calculated *a priori*. One must also specify a convention in case of a tie between the probabilities or distances, for instance, using a random choice, or requesting a retransmission.

3.2 Error detection and correction

The aim when constructing codes for noisy channels is to detect errors, and if possible, to correct them.

Definition. A binary $[n, m]$-code C is
- *d-error detecting* if, when changing up to d digits in each codeword, we can never produce another codeword;
- *e-error correcting* if, knowing that $x \in \{0,1\}^n$ differs from a codeword in at most e positions, we can deduce the codeword.

Example. A *repetition code* of length n has codewords $0^n, 1^n$. This is an $[n, 2]$-code. It is $(n-1)$-error detecting, and $\left\lfloor \frac{n-1}{2} \right\rfloor$-error correcting. Its information rate is $\frac{1}{n}$.

Example. A *simple parity check code* or *paper tape code* of length n identifies the set $\{0, 1\}$ with the field $\mathbb{F}_2$ of two elements, and defines $C = \{(x_1, \dots, x_n) \in \mathbb{F}_2^n \mid \sum x_i = 0\}$. This is an $[n, 2^{n-1}]$-code. This is 1-error detecting and 0-error correcting, but has information rate $\frac{n-1}{n}$.

Example. Hamming's original code is a 1-error correcting binary $[7, 16]$-code, defined on a subset of $\mathbb{F}_2^7$ by

$$C = \left\{c \in \mathbb{F}_2^7 \mid c_1 + c_3 + c_5 + c_7 = 0; c_2 + c_3 + c_6 + c_7 = 0; c_4 + c_5 + c_6 + c_7 = 0\right\}$$

The bits c_3, c_5, c_6, c_7 are chosen arbitrarily, and c_1, c_2, c_4 are check digits, giving a size of $2^4 = 16$. Suppose that we receive $x \in \mathbb{F}_2^7$. We form the *syndrome* $z = z_x = (z_1, z_2, z_4) \in \mathbb{F}_2^3$ where

$$z_1 = x_1 + x_3 + x_5 + x_7; \quad z_2 = x_2 + x_3 + x_6 + x_7; \quad z_4 = x_4 + x_5 + x_6 + x_7$$

By definition of C, if $x \in C$ then $z = (0, 0, 0)$. If $d(x, c) = 1$ for some $c \in C$, then the place where x and c differ is given by $z_1 + 2z_2 + 4z_4$ (not modulo 2). Indeed, if $x = c + e_i$ where e_i is the zero vector with a one in the ith position, $z_x = z_{e_i}$, and one can check that this holds for each $1 \leq i \leq 7$. Therefore, Hamming's original code is 1-error correcting.

Lemma. The Hamming distance is a metric on $\mathbb{F}_2^n$.

Proof. Clearly, $d(x, y) \geq 0$ and equality holds if and only if $x = y$, and $d(x, y) = d(y, x)$. Let $x, y, z \in \mathbb{F}_2^n$. Then,

$$\{i \mid x_i \neq z_i\} \subseteq \{i \mid x_i \neq y_i\} \cup \{i \mid y_i \neq z_i\}$$

Hence $d(x, z) \leq d(x, y) + d(y, z)$. □

Remark. We could write $d(x, y)$ as $\sum d_1(x_i, y_i)$ where d_1 is the discrete metric on $\mathbb{F}_2$.

3.3 Minimum distance

Definition. The *minimum distance* of a code is the minimum value of $d(c_1, c_2)$ for codewords $c_1 \neq c_2$.

Lemma. Let C be a code with minimum distance $d > 0$. Then,
(i) C is $(d-1)$-error detecting, but cannot detect all sets of d errors;
(ii) C is $\left\lfloor \frac{d-1}{2} \right\rfloor$-error correcting, but cannot correct all sets of $\left\lfloor \frac{d-1}{2} \right\rfloor + 1$ errors.

Proof. Part (i). If $x \in \mathbb{F}_2^n$ and c is a codeword with $1 \leq d(x, c) \leq d - 1$. Then $x \notin C$, so $d - 1$ errors are detected. Suppose c_1, c_2 are codewords with $d(c_1, c_2) = d$. Then c_1 can be corrupted into c_2 with only d errors, and this is undetectable.

Part (ii). Let $e = \left\lfloor \frac{d-1}{2} \right\rfloor$. By definition, $e \leq \frac{d-1}{2} < e + 1$, so $2e < d \leq 2(e + 1)$. Let $x \in \mathbb{F}_2^n$. If $c_1 \in C$ with $d(x, c_1) \leq e$, we want to show that $d(x, c_2) > e$ for all $c_2 \neq c_1$. By the triangle inequality, $d(x, c_2) \geq d(c_1, c_2) - d(x, c_1) \geq d - e > e$ as required. Hence, C is e-error correcting.

Let $c_1, c_2 \in C$ with $d(c_1, c_2) = d$. Let $x \in \mathbb{F}_2^n$ differ from c_1 in precisely $e+1$ places that c_1 and c_2 differ. Then $d(x, c_1) = e + 1$, and $d(x, c_2) = d - (e+1) \leq e + 1$. Hence, C cannot correct all sets of $e+1$ errors. $\square$

Definition. An $[n, m]$-code with minimum distance d is called an $[n, m, d]$-code.

Note that $m \leq 2^n$ with equality if and only if $C = \mathbb{F}_2^n$. Similarly, $d \leq n$, with equality in the case of the repetition code.

Example. The repetition code of length n is an $[n, 2, n]$-code. The simple parity check code of length n is an $[n, 2^{n-1}, 2]$-code. The trivial code on n bits is an $[n, 2^n, 1]$-code. Hamming's original code is 1-error correcting, so has minimum distance at least 3. The minimum distance can easily be shown to be exactly 3 as $0000000, 1110000$ are codewords, so it is a $[7, 16, 3]$-code.

3.4 Covering estimates

Definition. Let $x \in \mathbb{F}_2^n$ and $r \geq 0$. Then, we denote the *closed Hamming ball* with centre x and radius r by $B(x, r)$. We write $V(n, r) = |B(x, r)| = \sum_{i=0}^{r} \binom{n}{i}$ for the *volume* of this ball.

Lemma (Hamming's bound; sphere packing bound)**.** An e-error correcting code C of length n has

$$|C| \leq \frac{2^n}{V(n, e)}$$

Proof. C is e-error correcting, so $B(c_1, e) \cap B(c_2, e)$ is empty for all codewords $c_1 \neq c_2$. Hence,

$$\sum_{c \in C} |B(c, e)| \leq |\mathbb{F}_2^n| \implies |C| V(n, e) \leq 2^n$$

as required. $\square$

Definition. An e-error correcting code C of length n such that $|C| = \frac{2^n}{V(n,e)}$ is called *perfect*.

Remark. Equivalently, a code is perfect if for all $x \in \mathbb{F}_2^n$, there exists a unique $c \in C$ such that $d(x, c) \leq e$. Alternatively, $\mathbb{F}_2^n$ is a union of disjoint balls $B(c, e)$ for all $c \in C$, or that any collection of $e + 1$ will cause the message to be decoded incorrectly.

Example. Consider Hamming's $[7, 16, 3]$-code. This is 1-error correcting, and

$$\frac{2^n}{V(n, e)} = \frac{2^7}{V(7, 1)} = \frac{2^7}{1 + 7} = 2^4 = |C|$$

So Hamming's original code is perfect.

Example. The binary repetition code of length n is perfect if and only if n is odd.

Remark. If $\frac{2^n}{V(n,e)}$ is not an integer, there does not exist a perfect e-error correcting code of length n. The converse is false; the case $n = 90, e = 2$ is discussed on the second example sheet.

Definition. $A(n, d)$ is the largest possible size m of an $[n, m, d]$-code.

The values of the $A(n, d)$ are unknown in general.

Example. $A(n, 1) = 2^n$, considering the trivial code. $A(n, 2) = 2^{n-1}$, maximised at the simple parity check code. $A(n, n) = 2$, maximised at the repetition code.

Lemma. $A(n, d + 1) \leq A(n, d)$.

Proof. Let $m = A(n, d + 1)$, and let C be an $[n, m, d + 1]$-code. Let $c_1, c_2 \in C$ be distinct codewords such that $d(c_1, c_2) = d + 1$. Let c_1' differ from c_1 in exactly one of the places where c_1 and c_2 differ. Then $d(c_1', c_2) = d$. If $c \in C$ is any codeword not equal to c_1, then $d(c, c_1) \leq d(c, c_1') + d(c_1', c_1)$ hence $d + 1 \leq d(c, c_1') + 1$, so the code given by $C \cup \{c_1'\} \setminus \{c_1\}$ has minimum distance d, but has length n and size m. This is therefore an $[n, m, d]$-code as required. $\square$

Corollary. Equivalently, $A(n, d) = \max\{m \mid \exists [n, m, d']\text{-code, for some } d' \geq d\}$.

Theorem.

$$\frac{2^n}{V(n, d-1)} \leq A(n, d) \leq \frac{2^n}{V\left(n, \left\lfloor \frac{d-1}{2} \right\rfloor\right)}$$

The upper bound is Hamming's bound; the lower bound is known as the GSV (Gilbert–Shannon–Varshamov) bound. The upper bound can be thought of as a sphere packing bound, and the lower bound is a sphere covering bound.

Proof. We prove the lower bound. Let $m = A(n, d)$, and let C be an $[n, m, d]$-code. Then, there exists no $x \in \mathbb{F}_2^n$ with $d(x, c) \geq d$ for all codewords. Indeed, if such an x exists, we could consider the code $C \cup \{x\}$, which would be an $[n, m + 1, d]$-code, contradicting maximality of m. Then,

$$\mathbb{F}_2^n \subseteq \bigcup_{c \in C} B(c, d - 1) \implies 2^n \leq \sum_{c \in C} |B(c, d - 1)| = mV(n, d - 1)$$

as required. $\square$

Example. Let $n = 10, d = 3$. Then $V(n, 1) = 11$ and $V(n, 2) = 56$, so the GSV bound is $\frac{2^{10}}{56} \leq A(10, 3) \leq \frac{2^{10}}{11}$. Hence, $19 \leq A(10, 3) \leq 93$. It was known that the lower bound could be improved to 72. We now know that the true value of $A(10, 3)$ is exactly 72. In this case, the GSV bound was not a sharp inequality.

3.5 Asymptotics

We study the information rate $\frac{\log A(n,\lfloor n\delta \rfloor)}{n}$ as $n \to \infty$ to see how large the information rate can be for a fixed error rate.

Proposition. Let $0 < \delta < \frac{1}{2}$. Then,
(i) $\log V(n, \lfloor n\delta \rfloor) \leq nH(\delta)$;
(ii) $\frac{1}{n} \log A(n, \lfloor n\delta \rfloor) \geq 1 - H(\delta)$;
where $H(\delta) = -\delta \log \delta - (1-\delta)\log(1-\delta)$.

Proof. (i) implies (ii). By the GSV bound, we find

$$A(n, \lfloor n\delta \rfloor) \geq \frac{2^n}{V(n, \lfloor n\delta \rfloor - 1)} \geq \frac{2^n}{V(n, \lfloor n\delta \rfloor)}$$

Taking logarithms,

$$\frac{1}{n} \log A(n, \lfloor n\delta \rfloor) \geq 1 - \frac{\log V(n, \lfloor n\delta \rfloor)}{n} \geq 1 - H(\delta)$$

Part (i). $H(\delta)$ is increasing for $\delta < \frac{1}{2}$. Therefore, without loss of generality, we may assume $n\delta$ is an integer. Now, as $\frac{\delta}{1-\delta} < 1$,

$$\begin{aligned}
1 &= (\delta + (1-\delta))^n \\
&= \sum_{i=0}^{n} \binom{n}{i} \delta^i (1-\delta)^{n-i} \\
&\geq \sum_{i=0}^{n\delta} \binom{n}{i} \delta^i (1-\delta)^{n-i} \\
&= (1-\delta)^n \sum_{i=0}^{n\delta} \binom{n}{i} \left(\frac{\delta}{1-\delta}\right)^i \\
&\geq (1-\delta)^n \sum_{i=0}^{n\delta} \binom{n}{i} \left(\frac{\delta}{1-\delta}\right)^{n\delta} \\
&= \delta^{n\delta} (1-\delta)^{n(1-\delta)} V(n, n\delta)
\end{aligned}$$

Taking logarithms,

$$0 \geq n\delta \log \delta + n(1-\delta)\log(1-\delta) + \log V(n, n\delta)$$

as required. □

The constant $H(\delta)$ in the proposition is optimal.

Lemma. $\lim_{n\to\infty} \frac{\log V(n, \lfloor n\delta \rfloor)}{n} = H(\delta)$.

Proof. Exercise. Follows from Stirling's approximation to factorials. □

3.6 Constructing new codes from old

Let C be an $[n, m, d]$-code.

Example. The *parity check extension* is an $[n+1, m, d']$-code given by

$$C^+ = \left\{ \left(c_1, \dots, c_n, \sum_{i=1}^{n} c_i \right) \middle| (c_1, \dots, c_n) \in C \right\}$$

where d' is either d or $d+1$, depending on whether d is odd or even.

Example. Let $1 \leq i \leq n$. Then, deleting the ith digit from each codeword gives the *punctured code*

$$C^- = \{(c_1, \dots, c_{i-1}, c_{i+1}, \dots, c_n) \mid (c_1, \dots, c_n) \in C\}$$

If $d \geq 2$, this is an $[n-1, m, d']$-code where d' is either d or $d-1$.

Example. Let $1 \leq i \leq n$ and let $\alpha \in \mathbb{F}_2$. The *shortened code* is

$$C' = \{(c_1, \dots, c_{i-1}, c_{i+1}, \dots, c_n) \mid (c_1, \dots, c_{i-1}, \alpha, c_{i+1}, \dots, c_n) \in C\}$$

This is an $[n-1, m', d']$ with $d' \geq d$ and $m' \geq \frac{m}{2}$ for a suitable choice of α.

4 Information theory

4.1 Sources and information rate

Definition. A *source* is a sequence of random variables $X_1, X_2, \dots$ taking values in $\mathcal{A}$.

Example. The *Bernoulli* (or *memoryless*) source is a source where the X_i are independent and identically distributed according to a Bernoulli distribution.

Definition. A source $X_1, X_2, \dots$ is *reliably encodable* at rate r if there exist subsets $A_n \subseteq \mathcal{A}^n$ such that
(i) $\lim \frac{\log |A_n|}{n} = r$;
(ii) $\lim \mathbb{P}\left((X_1, \dots, X_n) \in A_n\right) = 1$.

Definition. The *information rate* H of a source is the infimum of all reliable encoding rates.

Example. $0 \leq H \leq \log |\mathcal{A}|$, with both bounds attainable. The proof is left as an exercise.

Shannon's first coding theorem computes the information rate of certain sources, including Bernoulli sources.

Recall from IA Probability that a probability space is a tuple $(\Omega, \mathcal{F}, \mathbb{P})$, and a discrete random variable is a function $X : \Omega \to \mathcal{A}$. The probability mass function is the function $p_X : \mathcal{A} \to [0,1]$ given by $p_X(x) = \mathbb{P}(X = x)$. We can consider the function $p(X) : \Omega \to [0,1]$ defined by the composition $p_X \circ X$, which assigns $p(X)(\omega) = \mathbb{P}(X = X(\omega))$; hence, $p(X)$ is also a random variable.

Similarly, given a source $X_1, X_2, \dots$ of random variables with values in $\mathcal{A}$, the probability mass function of any tuple $X^{(n)} = (X_1, \dots, X_n)$ is $p_{X^{(n)}}(x_1, \dots, x_n) = \mathbb{P}(X_1 = x_1, \dots, X_n = x_n)$. As $p_{X^{(n)}} : \mathcal{A}^n \to [0,1]$, and $X^{(n)} : \Omega \to \mathcal{A}^n$, we can consider $p(X^{(n)}) = p_{X^{(n)}} \circ X^{(n)}$ defined by $\omega \mapsto p_{X^{(n)}}(X^{(n)}(\omega))$.

Example. Let $\mathcal{A} = \{A, B, C\}$. Suppose

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

Then, $p_{X^{(2)}}(AB) = 0.3$, and so on. Hence,

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

We say that a source $X_1, X_2, \dots$ converges in probability to a random variable L if for all $\varepsilon > 0$, $\lim_{n\to\infty} \mathbb{P}(|X_n - L| > \varepsilon) = 0$. We write $X_n \xrightarrow{\mathbb{P}} L$. The weak law of large numbers states that if $X_1, X_2, \dots$ is a sequence of independent identically distributed real-valued random variables with finite expectation $\mathbb{E}[X_1]$, then $\frac{1}{n}\sum_{i=1}^n X_i \xrightarrow{\mathbb{P}} \mathbb{E}[X]$.

Example. Let $X_1, X_2, \dots$ be a Bernoulli source. Then $p(X_1), p(X_2), \dots$ are independent and identically distributed random variables, and $p(X_1, \dots, X_n) = p(X_1) \dots p(X_n)$. Note that by the weak law of large numbers,

$$-\frac{1}{n}\log p(X_1, \dots, X_n) = -\frac{1}{n}\sum_{i=1}^n \log p(X_i) \xrightarrow{\mathbb{P}} \mathbb{E}[-\log p(X_1)] = H(X_1)$$

Lemma. The information rate of a Bernoulli source $X_1, X_2, \dots$ is at most the expected word length of an optimal code $c : \mathcal{A} \to \{0,1\}^\star$ for X_1.

Proof. Let $\ell_1, \ell_2, \dots$ be the codeword lengths when we encode $X_1, X_2, \dots$ using c. Let $\varepsilon > 0$. Let

$$A_n = \{x \in \mathcal{A}^n \mid c^\star(x) \text{ has length less than } n(\mathbb{E}[\ell_1] + \varepsilon)\}$$

Then,

$$\mathbb{P}((X_1, \dots, X_n) \in A_n) = \mathbb{P}\left(\sum_{i=1}^n \ell_i \leq n(\mathbb{E}[\ell_1] + \varepsilon)\right) = \mathbb{P}\left(\left|\frac{1}{n}\sum_{i=1}^n \ell_i - \mathbb{E}[\ell_i]\right| < \varepsilon\right) \to 1$$

Now, c is decipherable so $c^\star$ is injective. Hence, $|A_n| \leq 2^{n(\mathbb{E}[\ell_1]+\varepsilon)}$. Making A_n larger if necessary, we can assume $|A_n| = \lfloor 2^{n(\mathbb{E}[\ell_1]+\varepsilon)} \rfloor$. Taking logarithms, $\frac{\log|A_n|}{n} \to \mathbb{E}[\ell_1] + \varepsilon$. So $X_1, X_2, \dots$ is reliably encodable at rate $r = \mathbb{E}[\ell_1] + \varepsilon$ for all $\varepsilon > 0$. Hence the information rate is at most $\mathbb{E}[\ell_1]$. $\square$

Corollary. A Bernoulli source has information rate less than $H(X_1) + 1$.

Proof. Combine the previous lemma with the noiseless coding theorem. □

Suppose we encode $X_1, X_2, \dots$ in blocks of size N. Let $Y_1 = (X_1, \dots, X_N), Y_2 = (X_{N+1}, \dots, X_{2N})$ and so on, such that $Y_1, Y_2, \dots$ take values in $\mathcal{A}^N$. One can show that if the source $X_1, X_2, \dots$ has information rate H, then $Y_1, Y_2, \dots$ has information rate NH.

Proposition. The information rate H of a Bernoulli source is at most $H(X_1)$.

Proof. Apply the previous corollary to the Y_i to obtain

$$NH < H(Y_1) + 1 = H(X_1, \dots, X_N) + 1 = NH(X_1) + 1 \implies H < H(X_1) + \frac{1}{N}$$

as required. □

4.2 Asymptotic equipartition property

Definition. A source $X_1, X_2, \dots$ satisfies the *asymptotic equipartition property* if there exists a constant $H \geq 0$ such that

$$-\frac{1}{n} \log p(X_1, \dots, X_n) \xrightarrow{\mathbb{P}} H$$

Example. Suppose we toss a biased coin with probability p of obtaining a head. Let $X_1, X_2, \dots$ be the results of independent coin tosses. If we toss the coin N times, we expect pN heads and $(1-p)N$ tails. The probability of any particular sequence of pN heads and $(1-p)N$ tails is

$$p^{pN}(1-p)^{(1-p)N} = 2^{N(p\log p + (1-p)\log(1-p))} = 2^{-NH(X)}$$

Not every sequence of tosses is of this form, but there is only a small probability of 'atypical sequences'. With high probability, it is a 'typical sequence' which has a probability close to $2^{-NH(X)}$.

Lemma. The asymptotic equipartition property for a source $X_1, X_2, \dots$ is equivalent to the property that for all $\varepsilon > 0$, there exists $n \in \mathbb{N}$ such that for all $n \geq n_0$, there exists a 'typical set' $T_n \subseteq \mathcal{A}^n$ such that
(i) $\mathbb{P}\left((X_1, \dots, X_n) \in T_n\right) > 1 - \varepsilon$;
(ii) $2^{-n(H+\varepsilon)} \leq p(x_1, \dots, x_n) \leq 2^{-n(H-\varepsilon)}$ for all $(x_1, \dots, x_n) \in T_n$.

Proof sketch. First, we show that the asymptotic equipartition property implies the alternative definition. We define

$$T_n = \left\{(x_1, \dots, x_n) \,\middle|\, \left|-\frac{1}{n} \log p(x_1, \dots, x_n) - H\right| \leq \varepsilon\right\} = \{(x_1, \dots, x_n) \mid \text{condition (ii) holds}\}$$

For the converse,

$$\mathbb{P}\left(\left|\frac{1}{n} \log p(x_1, \dots, x_n) - H\right| < \varepsilon\right) \geq \mathbb{P}\left(T_n\right) \to 1$$

□

4.3 Shannon's first coding theorem

Theorem. Let $X_1, X_2, \dots$ be a source satisfying the asymptotic equipartition property with constant H. Then this source has information rate H.

Proof. Let $\varepsilon > 0$, and let $T_n \subseteq \mathcal{A}^n$ be typical sets. Then, for all $n \geq n_0(\varepsilon)$, for all $(x_1, \dots, x_n) \in T_n$ we have $p(x_1, \dots, x_n) \geq 2^{-n(H+\varepsilon)}$. Therefore, $1 \geq \mathbb{P}(T_n) \geq 2^{-n(H+\varepsilon)} \cdot |T_n|$, giving $\frac{1}{n} \log |T_n| \leq H + \varepsilon$. Taking $A_n = T_n$ in the definition of reliable encoding shows that the source is reliably encodable at rate $H + \varepsilon$.

Conversely, if $H = 0$ the proof concludes, so we may assume $H > 0$. Let $0 < \varepsilon < \frac{H}{2}$, and suppose that the source is reliably encodable at rate $H - 2\varepsilon$ with sets $A_n \subseteq \mathcal{A}^n$. Let $T_n \subseteq \mathcal{A}^n$ be typical sets. Then, for all $(x_1, \dots, x_n) \in T_n$, $p(x_1, \dots, x_n) \leq 2^{-n(H-\varepsilon)}$, so $\mathbb{P}(A_n \cap T_n) \leq 2^{-n(H-\varepsilon)}|A_n|$, giving

$$\frac{1}{n} \log \mathbb{P}(A_n \cap T_n) \leq -(H - \varepsilon) + \frac{1}{n} \log |A_n| \to -(H - \varepsilon) + (H - 2\varepsilon) = -\varepsilon$$

Then, $\log \mathbb{P}(A_n \cap T_n) \to -\infty$, so $\mathbb{P}(A_n \cap T_n) \to 0$. But $\mathbb{P}(T_n) \leq \mathbb{P}(A_n \cap T_n) + \mathbb{P}(\mathcal{A}^n \setminus A_n) \to 0 + 0$, contradicting typicality. So we cannot reliably encode at rate $H - \varepsilon$, so the information rate is at least H. □

Corollary. A Bernoulli source $X_1, X_2, \dots$ has information rate $H(X_1)$.

Proof. In a previous example we showed that for a Bernoulli source, $-\frac{1}{n} \log p(X_1, \dots, X_n) \xrightarrow{\mathbb{P}} H(X_1)$. So the asymptotic equipartition property holds with $H = H(X_1)$, giving the result by Shannon's first coding theorem. □

Remark. The asymptotic equipartition property is useful for noiseless coding. We can encode the typical sequences using a block code, and encode the atypical sequences arbitrarily.

Many sources, which are not necessarily Bernoulli, also satisfy the property. Under suitable hypotheses, the sequence $\frac{1}{n} H(X_1, \dots, X_n)$ is decreasing, and the asymptotic equipartition property is satisfied with constant $H = \lim_{n\to\infty} \frac{1}{n} H(X_1, \dots, X_n)$.

4.4 Capacity

Consider a communication channel with input alphabet $\mathcal{A}$ and output alphabet $\mathcal{B}$. Recall the following definitions. A *code* of length n is a subset $C \subseteq \mathcal{A}^n$. The *error rate* is

$$\hat{e}(C) = \max_{c \in C} \mathbb{P}(\text{error} \mid c \text{ sent})$$

The *information rate* is $\rho(C) = \frac{\log |C|}{n}$. A channel can *transmit reliably at rate R* if there exist codes $C_1, C_2, \dots$ where C_n has length n such that $\lim_{n\to\infty} \rho(C_n) = R$ and $\lim_{n\to\infty} \hat{e}(C_n) = 0$. The *(operational) capacity* of a channel is the supremum of all rates at which it can transmit reliably.

Suppose we are given a source with information rate r bits per second that emits symbols at a rate of s symbols per second. Suppose we also have a channel with capacity R bits per transmission that transmits symbols at a rate of S transmissions per second. Usually, information theorists take $S = s = 1$. We will show that reliable encoding and transmission is possible if and only if $rs \leq RS$.

We will now compute the capacity of the binary symmetric channel with error probability p.

Proposition. A binary symmetric channel with error probability $p < \frac{1}{4}$ has nonzero capacity.

Proof. Let δ be such that $2p < \delta < \frac{1}{2}$. We claim that we can reliably transmit at rate $R = 1 - H(\delta) > 0$. Let C_n be a code of length n, and suppose it has minimum distance $\lfloor n\delta \rfloor$ of maximal size. Then, by the GSV bound,

$$|C_n| = A(n, \lfloor n\delta \rfloor) \geq 2^{-n(1-H(\delta))} = 2^{nR}$$

Replacing C_n with a subcode if necessary, we can assume $|C_n| = \lfloor 2^{nR} \rfloor$, with minimum distance at least $\lfloor n\delta \rfloor$. Using minimum distance decoding,

$$\begin{aligned}
\hat{e}(C_n) &\leq \mathbb{P}\left(\text{in } n \text{ uses, the channel makes at least } \left\lfloor \frac{\lfloor n\delta \rfloor - 1}{2} \right\rfloor \text{ errors}\right) \\
&\leq \mathbb{P}\left(\text{in } n \text{ uses, the channel makes at least } \left\lfloor \frac{n\delta - 1}{2} \right\rfloor \text{ errors}\right)
\end{aligned}$$

Let $\varepsilon > 0$ be such that $p + \varepsilon < \frac{\delta}{2}$. Then, for n sufficiently large, $\frac{n\delta - 1}{2} = n\left(\frac{\delta}{2} - \frac{1}{2n}\right) > n(p + \varepsilon)$. Hence, $\hat{e}(C_n) \leq \mathbb{P}\left(\text{in } n \text{ uses, the channel makes at least } n(p+\varepsilon) \text{ errors}\right)$. We show that this value converges to zero as $n \to \infty$ using the next lemma. $\square$

Lemma. Let $\varepsilon > 0$. A binary symmetric channel with error probability p is used to transmit n digits. Then,

$$\lim_{n \to \infty} \mathbb{P}\left(\text{in } n \text{ uses, the channel makes at least } n(p+\varepsilon) \text{ errors}\right) = 0$$

Proof. Consider random variables $U_i = \mathbb{1}[\text{the } i\text{th digit is mistransmitted}]$. The U_i are independent and identically distributed with $\mathbb{P}(U_i = 1) = p$. In particular, $\mathbb{E}[U_i] = p$. Therefore, the probability that the channel makes at least $n(p+\varepsilon)$ errors is

$$\mathbb{P}\left(\sum_{i=1}^n U_i \geq n(p+\varepsilon)\right) \leq \mathbb{P}\left(\left|\frac{1}{n}\sum_{i=1}^n U_i - p\right| \geq \varepsilon\right)$$

so the result holds by the weak law of large numbers. $\square$

4.5 Conditional entropy

Definition. Let X, Y be random variables taking values in alphabets $\mathcal{A}, \mathcal{B}$ respectively. Then, the *conditional entropy* is defined by

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

and

$$H(X \mid Y) = \sum_{y \in \mathcal{B}} \mathbb{P}(Y = y) H(X \mid Y = y)$$

Note that $H(X \mid Y) \geq 0$.

Lemma. $H(X, Y) = H(X \mid Y) + H(Y)$.

Proof.

$$\begin{aligned}
H(X \mid Y) &= -\sum_{y \in \mathcal{B}} \sum_{x \in \mathcal{A}} \mathbb{P}(X = x \mid Y = y) \mathbb{P}(Y = y) \log (\mathbb{P}(X = x \mid Y = y)) \\
&= -\sum_{y \in \mathcal{B}} \sum_{x \in \mathcal{A}} \mathbb{P}(X = x \mid Y = y) \mathbb{P}(Y = y) \log \left(\frac{\mathbb{P}(X = x, Y = y)}{\mathbb{P}(Y = y)} \right) \\
&= -\sum_{y \in \mathcal{B}} \sum_{x \in \mathcal{A}} \mathbb{P}(X = x, Y = y) (\log \mathbb{P}(X = x, Y = y) - \log \mathbb{P}(Y = y)) \\
&= -\sum_{y \in \mathcal{B}} \sum_{x \in \mathcal{A}} \mathbb{P}(X = x, Y = y) \log \mathbb{P}(X = x, Y = y) \\
&+ \sum_{y \in \mathcal{B}} \sum_{x \in \mathcal{A}} \mathbb{P}(X = x, Y = y) \log \mathbb{P}(Y = y) \\
&= -\sum_{y \in \mathcal{B}} \sum_{x \in \mathcal{A}} \mathbb{P}(X = x, Y = y) \log \mathbb{P}(X = x, Y = y) \\
&+ \sum_{y \in \mathcal{B}} \mathbb{P}(Y = y) \log \mathbb{P}(Y = y) \\
&= H(X, Y) - H(Y)
\end{aligned}$$

□

Example. Let X be a uniform random variable on $\{1, \dots, 6\}$ modelling a dice roll, and Y is defined to be zero if X is even, and one if X is odd. Then, $H(X, Y) = H(X) = \log 6$ and $H(Y) = \log 2$. Therefore, $H(X \mid Y) = \log 3$ and $H(Y \mid X) = 0$.

Corollary. $H(X \mid Y) \leq H(X)$, with equality if and only if X and Y are independent.

Proof. Combine this result with the fact that $H(X, Y) \leq H(X) + H(Y)$ where equality holds if and only if $H(X), H(Y)$ are independent. □

Now, replace random variables X and Y with random vectors $X^{(r)} = (X_1, \dots, X_r)$ and $Y^{(s)} = (Y_1, \dots, Y_s)$. Similarly, we can define $H(X_1, \dots, X_r \mid Y_1, \dots, Y_s) = H(X^{(r)} \mid Y^{(s)})$. Note that $H(X, Y \mid Z)$ is the entropy of X and Y combined, given the value of Z, and is not the entropy of X, together with Y given Z.

Lemma. Let X, Y, Z be random variables. Then, $H(X \mid Y) \leq H(X \mid Y, Z) + H(Z)$.

Proof. Expand $H(X, Y, Z)$ in two ways.

$$H(Z \mid X, Y) + \underbrace{H(X \mid Y) + H(Y)}_{H(X,Y)} = H(X, Y, Z) = H(X \mid Y, Z) + \underbrace{H(Z \mid Y) + H(Y)}_{H(Y,Z)}$$

Since $H(Z \mid X, Y) \geq 0$, we have

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

□

Proposition (Fano's inequality)**.** Let X, Y be random variables taking values in $\mathcal{A}$. Let $|\mathcal{A}| = m$, and let $p = \mathbb{P}(X \neq Y)$. Then $H(X \mid Y) \leq H(p) + p\log(m-1)$.

Proof. Define Z to be zero if $X = Y$ and one if $X \neq Y$. Then, $\mathbb{P}(Z = 0) = \mathbb{P}(X = Y) = 1 - p$, and $\mathbb{P}(Z = 1) = \mathbb{P}(X \neq Y) = p$. Hence, $H(Z) = H(p)$. Applying the previous lemma, $H(X \mid Y) \leq H(X \mid Y, Z) + H(p)$, so it suffices to show $H(X \mid Y, Z) \leq p\log(m-1)$.

Since $Z = 0$ implies $X = Y$, $H(X \mid Y = y, Z = 0) = 0$. There are $m - 1$ remaining possibilities for X. Hence, $H(X \mid Y = y, Z = 1) \leq \log(m-1)$.

$$\begin{aligned}
H(X \mid Y, Z) &= \sum_{y \in \mathcal{A}} \sum_{z \in \{0,1\}} \mathbb{P}(Y = y, Z = z) H(X \mid Y = y, Z = z) \\
&\leq \sum_{y \in \mathcal{A}} \mathbb{P}(Y = y, Z = 1) \log(m-1) \\
&= \mathbb{P}(Z = 1) \log(m-1) \\
&= p\log(m-1)
\end{aligned}$$

as required. □

Let X be a random variable describing the input to a channel and Y be a random variable describing the output of the channel. $H(p)$ provides the information required to decide whether an error has occurred, and $p\log(m-1)$ gives the information needed to resolve that error in the worst possible case.

4.6 Shannon's second coding theorem

Definition. Let X, Y be random variables taking values in $\mathcal{A}$. The *mutual information* is $I(X; Y) = H(X) - H(X \mid Y)$.

This is nonnegative, as $I(X; Y) = H(X) + H(Y) - H(X, Y) \geq 0$. Equality holds if and only if X, Y are independent. Clearly, $I(X; Y) = I(Y; X)$.

Definition. Consider a discrete memoryless channel with input alphabet $\mathcal{A}$ of size m and output alphabet $\mathcal{B}$. Let X be a random variable taking values in $\mathcal{A}$, used as the input to this channel. Let Y be the random variable output by the channel, depending on X and the channel matrix. The *information capacity* of the channel is $\max_X I(X; Y)$.

The maximum is taken over all discrete random variables X taking values in $\mathcal{A}$, or equivalently. This maximum is attained since I is continuous and the space

$$\left\{(p_1, \dots, p_m) \in \mathbb{R}^m \,\middle|\, p_i \geq 0, \sum_{i=1}^m p_i = 1\right\}$$

is compact. The information capacity depends only on the channel matrix.

Theorem. For a discrete memoryless channel, the (operational) capacity is equal to the information capacity.

We prove that the operational capacity is at most the information capacity in general, and we will prove the other inequality for the special case of the binary symmetric channel.

Example. Assuming this result holds, we compute the capacity of certain specific channels.

(i) Consider the binary symmetric channel with error probability p, input X, and output Y. Let $\mathbb{P}(X = 0) = \alpha, \mathbb{P}(X = 1) = 1-\alpha$, so $\mathbb{P}(Y = 0) = (1-p)\alpha p(1-\alpha), \mathbb{P}(Y = 1) = (1-p)(1-\alpha)+p\alpha$. Then, as $H(Y \mid X) = \mathbb{P}(X = 0)\,H(p) + \mathbb{P}(X = 1)\,H(p)$,

$$\begin{aligned} C &= \max_\alpha I(X;Y) = \max_\alpha [H(Y) - H(Y \mid X)] \\ &= \max_\alpha \left[H(\alpha(1-p) + (1-\alpha)p) - H(p)\right] = 1 - H(p) \end{aligned}$$

with the maximum attained at $\alpha = \frac{1}{2}$. Hence, the capacity of the binary symmetric channel is $C = 1 + p\log p + (1-p)\log(1-p)$. If $p = 0$ or $p = 1$, $C = 1$. If $p = \frac{1}{2}$, $C = 0$. Note that $I(X;Y) = I(Y;X)$; we can choose which to calculate for convenience.

(ii) Consider the binary erasure channel with erasure probability p, input X, and output Y. Let $\mathbb{P}(X = 0) = \alpha, \mathbb{P}(X = 1) = 1-\alpha$, so $\mathbb{P}(Y = 0) = (1-p)\alpha, \mathbb{P}(Y = 1) = (1-p)(1-\alpha), \mathbb{P}(Y = \star) = p$. We obtain

$$H(X \mid Y = 0) = 0; \quad H(X \mid Y = 1) = 0; \quad H(X \mid Y = \star) = H(\alpha)$$

Therefore, $H(X \mid Y) = pH(\alpha)$, giving

$$\begin{aligned} C &= \max_\alpha I(X;Y) = \max_\alpha \left[H(X) - H(X \mid Y)\right] \\ &= \max_\alpha \left[H(\alpha) - pH(\alpha)\right] = (1-p)\max_\alpha H(\alpha) = 1 - p \end{aligned}$$

with maximum attained at $\alpha = \frac{1}{2}$.

We will now model using a channel n times as the *nth extension*, replacing $\mathcal{A}$ with $\mathcal{A}^n$ and $\mathcal{B}$ with $\mathcal{B}^n$, and use the channel matrix defined by

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

Lemma. Consider a discrete memoryless channel with information capacity C. Then, its nth extension has information capacity nC.

Proof. Let $X_1, \dots, X_n$ be the input producing an output $Y_1, \dots, Y_n$. Since the channel is memoryless,

$$H(Y_1, \dots, Y_n \mid X_1, \dots, X_n) = \sum_{i=1}^{n} H(Y_i \mid X_1, \dots, X_n) = \sum_{i=1}^{n} H(Y_i \mid X_i)$$

Therefore,

$$\begin{aligned} I(X_1, \dots, X_n; Y_1, \dots, Y_n) &= H(Y_1, \dots, Y_n) - H(Y_1, \dots, Y_n \mid X_1, \dots, X_n) \\ &\leq \sum_{i=1}^{n} H(Y_i) - \sum_{i=1}^{n} H(Y_i \mid X_i) \\ &= \sum_{i=1}^{n} [H(Y_i) - H(Y_i \mid X_i)] \\ &= \sum_{i=1}^{n} I(X_i; Y_i) \leq nC \end{aligned}$$

Equality is attained by taking $X_1, \dots, X_n$ independent and identically distributed such that $I(X_i; Y_i) = C$. Indeed, if $X_1, \dots, X_n$ are independent, then so are $Y_1, \dots, Y_n$, so $H(Y_1, \dots, Y_n) = \sum_{i=1}^{n} H(Y_i)$. Therefore,

$$\max_{X_1, \dots, X_n} I(X_1, \dots, X_n; Y_1, \dots, Y_n) = nC$$

as required. □

We now prove part of Shannon's second coding theorem, that the operational capacity is at most the information capacity for a discrete memoryless channel.

Proof. Let C be the information capacity. Suppose reliable transmission is possible at a rate $R > C$. Then, there is a sequence of codes $(C_n)_{n \geq 1}$ where C_n has length n and size $\lfloor 2^{nR} \rfloor$, such that $\lim_{n\to\infty} \rho(C_n) = R$ and $\lim_{n\to\infty} \hat{e}(C_n) = 0$.

Recall that $\hat{e}(C_n) = \max_{c \in C_n} \mathbb{P}(\text{error} \mid c \text{ sent})$. Define the *average error rate* $e(C)$ by $e(C) = \frac{1}{|C_n|} \sum_{c \in C} \mathbb{P}(\text{error} \mid c \text{ sent})$. Note that $e(C_n) \leq \hat{e}(C_n)$. As $\hat{e}(C_n) \to 0$, we also have $e(C_n) \to 0$.

Consider an input random variable X distributed uniformly over C_n. Let Y be the output given by X and the channel matrix. Then $e(C_n) = \mathbb{P}(X \neq Y) = p$. Hence, $H(X) = \log |C_n| = \log \lfloor 2^{nR} \rfloor \geq nR - 1$ for sufficiently large n. Also, by Fano's inequality, $H(X \mid Y) \leq H(p) + p \log(|C_n| - 1) \leq 1 + pnR$.

Recall that $I(X; Y) = H(X) - H(X \mid Y)$. By the previous lemma, $nC \geq I(X; Y)$, so

$$nC \geq nR - 1 - 1 - pnR \implies pnR \geq n(R - c) - 2 \implies p \geq \frac{n(R - C) - 2}{nR}$$

As $n \to \infty$, the right hand side converges to $\frac{R-C}{R} > 0$. This contradicts the fact that $p = e(C_n) \to 0$. Hence, we cannot transmit reliably at any rate which exceeds C, hence the capacity is at most C. □

To complete the proof of Shannon's second coding theorem for the binary symmetric channel with error probability p, we prove that the operational capacity is at least $1 - H(p)$.

Proposition. Consider a binary symmetric channel with error probability p, and let $R < 1 - H(p)$. Then there exists a sequence of codes $(C_n)_{n\geq 1}$ with C_n of length n and size $\lfloor 2^{nR} \rfloor$ such that $\lim_{n\to\infty} \rho(C_n) = R$ and $\lim_{n\to\infty} e(C_n) = 0$.

Remark. This proposition deals with the average error rate, instead of the error rate $\hat{e}$.

Proof. We use the method of random coding. Without loss of generality let $p < \frac{1}{2}$. Let $\varepsilon > 0$ such that $p + \varepsilon < \frac{1}{2}$ and $R < 1 - H(p + \varepsilon)$. We use minimum distance decoding, and in the case of a tie, we make an arbitrary choice. Let $m = \lfloor 2^{nR} \rfloor$, and let $C = \{c_1, \dots, c_m\}$ be a code chosen uniformly at random from $\mathcal{C} = \{[n, m]\text{-codes}\}$, a set of size $\binom{2^n}{m}$.

Choose $1 \leq i \leq m$ uniformly at random, and send c_i through the channel, and obtain an output Y. Then, $\mathbb{P}(Y \text{ not decoded as } c_i)$ is the average value of $e(C)$ for C ranging over $\mathcal{C}$, giving $\frac{1}{|\mathcal{C}|} \sum_{C\in\mathcal{C}} e(C)$.
We can choose a code $C_n \in \mathcal{C}$ such that $e(C_n) \leq \frac{1}{|\mathcal{C}|} \sum_{C\in\mathcal{C}} e(C)$. So it suffices to show $\mathbb{P}(Y \text{ not decoded as } c_i) \to 0$.

Let $r = \lfloor n(p + \varepsilon) \rfloor$. Then if $B(Y, r) \cap C = \{c_i\}$, Y is correctly decoded as c_i. Therefore,

$$\mathbb{P}(Y \text{ not decoded as } c_i) \leq \mathbb{P}(c_i \notin B(Y, r)) + \mathbb{P}(B(Y, r) \cap C \supsetneq \{c_i\})$$

We consider the two cases separately.

In the first case with $d(c_i, Y) > r$, $\mathbb{P}(d(c_i, Y) > r)$ is the probability that the channel makes more than r errors, and hence more than $n(p + \varepsilon)$ errors. We have already shown that this converges to zero as $n \to \infty$.

In the second case with $d(c_i, Y) \leq r$, if $j \neq i$,

$$\mathbb{P}\left(c_j \in B(Y, r) \mid c_i \in B(Y, r)\right) = \frac{V(n, r) - 1}{2^n - 1} \leq \frac{V(n, r)}{2^n}$$

Therefore,

$$\begin{aligned}
\mathbb{P}(B(Y, r) \cap C \supsetneq \{c_i\}) &\leq \sum_{j\neq i} \mathbb{P}\left(c_j \in B(Y, r), c_i \in B(Y, r)\right) \\
&\leq \sum_{j\neq i} \mathbb{P}\left(c_j \in B(Y, r) \mid c_i \in B(Y, r)\right) \\
&\leq (m - 1)\frac{V(n, r)}{2^n} \\
&\leq \frac{mV(n, r)}{2^n} \\
&\leq 2^{nR} 2^{nH(p+\varepsilon)} 2^{-n} \\
&= 2^{n(R-(1-H(p+\varepsilon)))} \to 0
\end{aligned}$$

as required. ∎

Proposition. We can replace e with $\hat{e}$ in the previous result.

Proof. Let R' be such that $R < R' < 1 - H(p)$. Then, apply the previous result to R' to construct a sequence of codes $(C'_n)_{n\geq 1}$ of length n and size $\lfloor 2^{nR'} \rfloor$, where $e(C'_n) \to 0$. Order the codewords of C'_n by the probability of error given that the codeword was sent, and delete the worst half. This gives a code C_n with $\hat{e}(C_n) \leq 2e(C'_n)$. Hence $\hat{e}(C_n) \to 0$ as $n \to \infty$. Since C_n has length n, and size $\frac{1}{2}\lfloor 2^{nR'} \rfloor = \lfloor 2^{nR'-1} \rfloor$. But $2^{nR'-1} = 2^{n(R'-\frac{1}{n})} \geq 2^{nR}$ for sufficiently large n. So we can replace C'_n with a code of smaller size $\lfloor 2^{nR} \rfloor$ and still have $\hat{e}(C_n) \to 0$ and $\rho(C_n) \to R$ as $n \to \infty$. □

Therefore, a binary symmetric channel with error probability p has operational capacity $1 - H(p)$, as we can transmit reliably at any rate $R < 1 - H(p)$, and the capacity is at most $1 - H(p)$. The result shows that codes with certain properties exist, but does not give a way to construct them.

4.7 The Kelly criterion

Let $0 < p < 1$, $u > 0$, $0 \leq w < 1$. Suppose that a coin is tossed n times in succession with probability p of obtaining a head. If a stake of k is paid ahead of a particular throw, the return is ku if the result is a head, and the return is zero if the result is a tail.

Suppose the initial bankroll is $X_0 = 1$. After n throws, the bankroll is X_n. We bet wX_n on the $(n+1)$th coin toss, retaining $(1-w)X_n$. The bankroll after the toss is

$$X_{n+1} = \begin{cases} X_n(wu + (1-w)) & (n+1)\text{th toss is a head} \\ X_n(1-w) & (n+1)\text{th toss is a tail} \end{cases}$$

Define $Y_{n+1} = \frac{X_{n+1}}{X_n}$, then the Y_i are independent and identically distributed. Then $\log Y_i$ is a sequence of independent and identically distributed random variables. Note that $\log X_n = \sum_{i=1}^n \log Y_i$.

Lemma. Let $\mu = \mathbb{E}[\log Y_1]$, $\sigma^2 = \operatorname{Var}(\log Y_1)$. Then, if $a > 0$,

(i) $\mathbb{P}\left(\left|\frac{1}{n}\sum_{i=1}^n \log Y_i - \mu\right| \geq a\right) \leq \frac{\sigma^2}{na^2}$ by Chebyshev's inequality;

(ii) $\mathbb{P}\left(\left|\frac{\log X_n}{n} - \mu\right| \geq a\right) \leq \frac{\sigma^2}{na^2}$;

(iii) given $\varepsilon > 0$ and $\delta > 0$, there exists N such that $\mathbb{P}\left(\left|\frac{\log X_n}{n} - \mu\right| \geq \delta\right) \leq \varepsilon$ for all $n \geq N$.

Consider a single coin toss, with probability $p < 1$ of a head. Suppose that a bet of k on a head gives a payout of ku for some payout ratio $u > 0$. Suppose further that we have an initial bankroll of 1, and we bet w on heads, retaining $1-w$, for some $0 \leq w < 1$. Then, if Y is the expected fortune after the throw, $\mathbb{E}[\log Y] = p\log(1+(u-1)w) + (1-p)\log(1-w)$. One can show that the value of $\mathbb{E}[\log Y]$ is maximised by taking $w = 0$ if $up \leq 1$, and setting $w = \frac{up-1}{u-1}$ if $up > 1$.

Let $q = 1-p$. If $up > 1$, at the optimum value of w, we find

$$\mathbb{E}[\log Y] = p\log p + q\log q + \log u - q\log(u-1) = -H(p) + \log u - q\log(u-1)$$

Kelly's criterion is that in order to maximise profit, $\mathbb{E}[\log Y]$ should be optimised, given that we can bet arbitrarily many times.

One can show that if w is set below the optimum, the bankroll will still increase, but does so more slowly. If w is set sufficiently high, the bankroll will tend to decrease.

5 Algebraic coding theory

5.1 Linear codes

Definition. A binary code $C \subseteq \mathbb{F}_2^n$ is *linear* if $0 \in C$, and whenever $x, y \in C$, we have $x + y \in C$.

Equivalently, C is a vector subspace of $\mathbb{F}_2^n$.

Definition. The *rank* of a linear code C, denoted $\operatorname{rank} C$, is its dimension as an $\mathbb{F}_2$-vector space. A linear code of length n and rank k is called an (n,k)-code. If it has minimum distance d, it is called an (n,k,d)-code.

Let $v_1, \dots, v_k$ be a basis for C. Then $C = \left\{\sum_{i=1}^k \lambda_i v_i \mid \lambda_i \in \mathbb{F}_2\right\}$. The size of the code is therefore 2^k, so an (n,k)-code is an $[n, 2^k]$-code, and an (n,k,d)-code is an $[n, 2^k, d]$-code. The information rate is $\frac{k}{n}$.

Definition. The *weight* of $x \in \mathbb{F}_2^n$ is $w(x) = d(x, 0)$.

Lemma. The minimum distance of a linear code is the minimum weight of a nonzero codeword.

Proof. Let $x, y \in C$. Then, $d(x,y) = d(x+y, 0) = w(x+y)$. Observe that $x \neq y$ if and only if $x + y \neq 0$, so $d(C)$ is the minimum $w(x+y)$ for $x + y \neq 0$. $\square$

Definition. Let $x, y \in \mathbb{F}_2^n$. Define $x \cdot y = \sum_{i=1}^n x_i y_i \in \mathbb{F}_2$. This is symmetric and bilinear.

There are nonzero x such that $x \cdot x = 0$.

Definition. Let $P \subseteq \mathbb{F}_2^n$. The *parity check code* defined by P is

$$C = \{x \in \mathbb{F}_2^n \mid \forall p \in P,\ p \cdot x = 0\}$$

Example. (i) $P = \{11 \dots 1\}$ gives the simple parity check code.

(ii) $P = \{1010101, 0110011, 0001111\}$ gives Hamming's original $[7, 16, 3]$-code.

(iii) C^+ and C^- are linear if C is linear.

Lemma. Every parity check code is linear.

Proof. $0 \in C$ as $p \cdot 0 = 0$. If $p \cdot x = 0$ and $p \cdot y = 0$ then $p \cdot (x + y) = 0$, so $x, y \in C$ implies $x + y \in C$. $\square$

Definition. Let $C \subseteq \mathbb{F}_2^n$ be a linear code. The *dual code* $C^\perp$ is defined by

$$C^\perp = \{x \in \mathbb{F}_2^n \mid \forall y \in C,\ x \cdot y = 0\}$$

By definition, $C^\perp$ is a parity check code, and hence is linear. Note that $C \cap C^\perp$ may contain elements other than 0.

Lemma. $\operatorname{rank} C + \operatorname{rank} C^\perp = n$.

Proof. One can prove this by defining $C^\perp$ as an annihilator from linear algebra. A proof using coding theory is shown later. □

Corollary. Let C be a linear code. Then $(C^\perp)^\perp = C$. In particular, all linear codes are parity check codes, defined by $C^\perp$.

Proof. If $x \in C$, then $x \cdot y = 0$ for all $y \in C^\perp$ by definition, so $x \in (C^\perp)^\perp$. Then $\operatorname{rank} C = n - \operatorname{rank} C^\perp = n - (n - \operatorname{rank}(C^\perp)^\perp) = \operatorname{rank}(C^\perp)^\perp$, so $C = (C^\perp)^\perp$. □

Definition. Let C be an (n,k)-code. A *generator matrix* G for C is a $k \times n$ matrix where the rows form a basis for C. A *parity check matrix* H for C is a generator matrix for the dual code $C^\perp$, so it is an $(n-k) \times n$ matrix.

The codewords of a linear code can be viewed either as linear combinations of rows of G, or linear dependence relations between the columns of H, so $C = \{x \in \mathbb{F}_2^n \mid Hx = 0\}$.

Definition. Let C be an (n,k)-code. The *syndrome* of $x \in \mathbb{F}_2^n$ is Hx.

If we receive a word $x = c + z$ where $c \in C$ and z is the error pattern, $Hx = Hz$ as $Hc = 0$. If C is e-error correcting, we precompute Hz for all z for which $w(z) \leq e$. On receiving x, we can compute the syndrome Hx and find this entry in the table of values of Hz. If successful, we decode $c = x - z$, with $d(x,c) = w(z) \leq e$.

Definition. Codes $C_1, C_2 \subseteq \mathbb{F}_2^n$ are *equivalent* if there exists a permutation of bits that maps codewords in C_1 to codewords in C_2.

Codes are typically only considered up to equivalence.

Lemma. Every (n,k)-linear code is equivalent to one with generator matrix with block form $\begin{pmatrix} I_k & B \end{pmatrix}$ for some $k \times (n-k)$ matrix B.

Proof. Let G be a $k \times n$ generator matrix for C. Using Gaussian elimination, we can transform G into row echelon form

$$G_{ij} = \begin{cases} 0 & j < \ell(i) \\ 1 & j = \ell(i) \end{cases}$$

for some $\ell(1) < \ell(2) < \dots < \ell(k)$. Permuting the columns replaces C with an equivalent code, so without loss of generality we may assume $\ell(i) = i$. Hence,

$$G = \begin{pmatrix} 1 & & \star & \\ & \ddots & & B \\ & & 1 & \end{pmatrix}$$

Further row operations eliminate $\star$ to give G in the required form. $\square$

A message $y \in \mathbb{F}_2^k$ viewed as a row vector can be encoded as yG. If $G = \begin{pmatrix} I_k & B \end{pmatrix}$, then $yG = (y, yB)$ where y is the message and yB is a string of check digits. We now prove the following lemma that was stated earlier.

Lemma. $\operatorname{rank} C + \operatorname{rank} C^\perp = n$.

Proof. Let C have generator matrix $G = \begin{pmatrix} I_k & B \end{pmatrix}$. G has k linearly independent columns, so there is a linear map $\gamma : \mathbb{F}_2^n \to \mathbb{F}_2^k$ defined by $x \mapsto Gx$ which is surjective. Its kernel is $C^\perp$. By the rank-nullity theorem, $\dim \mathbb{F}_2^n = \dim \ker \gamma + \dim \operatorname{Im} \gamma$, so $n = \operatorname{rank} C + \operatorname{rank} C^\perp$ as required. $\square$

Lemma. An (n,k)-code with generator matrix $G = \begin{pmatrix} I_k & B \end{pmatrix}$ has parity check matrix H of the form $\begin{pmatrix} B^\intercal & I_{n-k} \end{pmatrix}$.

Proof.

$$GH^\intercal = \begin{pmatrix} I_k & B \end{pmatrix} \begin{pmatrix} B \\ I_{n-k} \end{pmatrix} = B + B = 2B = 0$$

So the rows of H generate a subcode of $C^\perp$. But $\operatorname{rank} H = n - k$, and $\operatorname{rank} C^\perp = n - k$. So $H = C^\perp$, and $C^\perp$ has generator matrix H. $\square$

Lemma. Let C be a linear code with parity check matrix H. Then, $d(C) = d$ if and only if

(i) any $d - 1$ columns of H are linearly independent; and
(ii) a set of d columns of H are linearly dependent.

The proof is left as an exercise.

5.2 Hamming codes

Definition. Let $d \geq 1$, and let $n = 2^d - 1$. Let H be the $d \times n$ matrix with columns given by the nonzero elements of $\mathbb{F}_2^d$. The *Hamming $(n, n-d)$-linear code* is the code with parity check matrix H.

Lemma. The Hamming $(n, n-d)$-code C has minimum distance $d(C) = 3$, and is a perfect 1-error correcting code.

Proof. Any two columns of H are linearly independent, but there are three linearly dependent columns. Hence, $d(C) = 3$. Hence, C is $\left\lfloor \frac{3-1}{2} \right\rfloor = 1$-error correcting. A perfect code is one such that $|C| = \frac{2^n}{V(n,e)}$. In this case, $n = 2^d - 1$ and $e = 1$, so $\frac{2^n}{1+2^d-1} = 2^{n-d} = |C|$ as required. $\square$

5.3 Reed–Muller codes

Let $X = \{p_1, \dots, p_n\}$ be a set of size n. There is a correspondence between the power set $\mathcal{P}(X)$ and $\mathbb{F}_2^n$.

$$\mathcal{P}(X) \xrightarrow{A \mapsto \mathbb{1}_A} \{f : X \to \mathbb{F}_2\} \xrightarrow{f \mapsto (f(p_1), \dots, f(p_n))} \mathbb{F}_2^n$$

The *symmetric difference* of two sets A, B is defined to be $A \triangle B = A \setminus B \cup B \setminus A$, which corresponds to vector addition in $\mathbb{F}_2^n$. Intersection $A \cap B$ corresponds to the *wedge product* $x \wedge y = (x_1 y_1, \dots, x_n y_n)$.

Let $X = \mathbb{F}_2^d$, so $n = 2^d - |X|$. Let $v_0 = (1, \dots, 1)$, and let $v_i = \mathbb{1}_{H_i}$ where $H_i = \{p \in X \mid p_i = 0\}$ is a coordinate hyperplane.

Definition. Let $0 \leq r \leq d$. The *Reed–Muller code* $RM(d, r)$ of *order* r and length 2^d is the linear code spanned by v_0 and all wedge products of at most r of the the v_i for $1 \leq i \leq d$.

By convention, the empty wedge product is v_0.

Example. Let $d = 3$, and let $X = \mathbb{F}_2^3 = \{p_1, \dots, p_8\}$ in binary order.

X	000	001	010	011	100	101	110	111
v_0	1	1	1	1	1	1	1	1
v_1	1	1	1	1	0	0	0	0
v_2	1	1	0	0	1	1	0	0
v_3	1	0	1	0	1	0	1	0
$v_1 \wedge v_2$	1	1	0	0	0	0	0	0
$v_2 \wedge v_3$	1	0	0	0	1	0	0	0
$v_1 \wedge v_3$	1	0	1	0	0	0	0	0
$v_1 \wedge v_2 \wedge v_3$	1	0	0	0	0	0	0	0

A generator matrix for Hamming's original code is a submatrix in the top-right corner.

$RM(3, 0)$ is spanned by v_0, and is hence the repetition code of length 8. $RM(3, 1)$ is spanned by v_0, v_1, v_2, v_3, which is equivalent to a parity check extension of Hamming's original $(7, 4)$-code. $RM(3, 2)$ is an $(8, 7)$-code, and can be shown to be equivalent to a simple parity check code of length 8. $RM(3, 3)$ is the trivial code $\mathbb{F}_2^8$ of length 8.

Theorem. (i) The vectors $v_{i_1} \wedge \dots \wedge v_{i_s}$ for $i_1 < \dots < i_s$ and $0 \leq s \leq d$ form a basis for $\mathbb{F}_2^n$.

(ii) The rank of $RM(d, r)$ is $\sum_{s=0}^{r} \binom{d}{s}$.

Proof. Part (i). There are $\sum_{s=0}^{d}\binom{d}{s} = 2^d = n$ vectors listed, so it suffices to show they are a spanning set, or equivalently $RM(d,d)$ is the trivial code. Let $p \in X$, and let y_i be v_i if $p_i = 0$ and $v_0 + v_i$ if $p_i = 1$. Then $\mathbb{1}_{\{p\}} = y_1 \wedge \dots \wedge y_d$. Expanding this using the distributive law, $\mathbb{1}_{\{p\}} \in RM(d,d)$. But the set of $\mathbb{1}_{\{p\}}$ for $p \in X$ spans $\mathbb{F}_2^n$, as required.

Part (ii). $RM(d,r)$ is spanned by $v_{i_1} \wedge \dots \wedge v_{i_s}$ where $i_1 < \dots < i_s$ and $0 \leq s \leq r$. Since these are linearly independent, the rank of $RM(d,r)$ is the number of such vectors, which is $\sum_{s=0}^{d}\binom{d}{s}$. □

Definition. Let C_1, C_2 be linear codes of length n where $C_2 \subseteq C_1$. The *bar product* is $C_1 \mid C_2 = \{(x \mid x+y) \mid x \in C_1, y \in C_2\}$.

This is a linear code of length $2n$.

Lemma. (i) $\operatorname{rank}(C_1 \mid C_2) = \operatorname{rank} C_1 + \operatorname{rank} C_2$.
(ii) $d(C_1 \mid C_2) = \min\{2d(C_1), d(C_2)\}$.

Proof. Part (i). If C_1 has basis $x_1, \dots, x_k$ and C_2 has basis $y_1, \dots, y_\ell$, then $C_1 \mid C_2$ has basis

$$\{(x_i \mid x_i) \mid 1 \leq i \leq k\} \cup \{(0 \mid y_i) \mid 1 \leq i \leq \ell\}$$

Part (ii). Let $0 \neq (x \mid x+y) \in C_1 \mid C_2$. If $y \neq 0$, then $w(x \mid x+y) = w(x) + w(x+y) \geq w(y) \geq d(C_2)$. If $y = 0$, then $w(x \mid x+y) = w(x \mid x) = 2w(x) \geq 2d(C_1)$. Hence, $d(C_1 \mid C_2) \geq \min\{2d(C_1), d(C_2)\}$.

There is a nonzero $x \in C_1$ with $w(x) = d(C_1)$, so $d(C_1 \mid C_2) \leq w(x \mid x) = 2d(C_1)$. There is a nonzero $y \in C_2$ with $w(y) = d(C_2)$, giving $d(C_1 \mid C_2) \leq w(0 \mid 0+y) = d(C_2)$, giving the other inequality as required. □

Theorem. (i) $RM(d,r) = RM(d-1,r) \mid RM(d-1,r-1)$ for $0 < r < d$.
(ii) $RM(d,r)$ has minimum distance 2^{d-r} for all r.

Proof. Part (i). Exercise.

Part (ii). If $r = 0$, then $RM(d,r)$ is the repetition code of length 2^d, which has minimum distance 2^d. If $r = d$, $RM(d,r)$ is the trivial code of length 2^d, which has minimum distance $1 = 2^{d-d}$. We prove the remaining cases by induction on d. From part (i), $RM(d,r) = RM(d-1,r) \mid RM(d-1,r-1)$. By induction, the minimum distance of $RM(d-1,r)$ is 2^{d-1-r} and the minimum distance of $RM(d-1,r-1)$ is 2^{d-r}. By part (ii) of the previous lemma, the minimum distance of $RM(d,r)$ is $\min\{2 \cdot 2^{d-1-r}, 2^{d-r}\} = 2^{d-r}$. □

5.4 Cyclic codes

If F is a field and $f \in F[X]$, ${}^{F[X]}\!/\!_{(f)}$ is in bijection with F^n where $n = \deg f$, since ${}^{F[X]}\!/\!_{(f)}$ is represented by the set of functions of degree less than $\deg f$.

Definition. A linear code $C \subseteq \mathbb{F}_2^n$ is *cyclic* if

$$(a_0, a_1, \dots, a_{n-1}) \in C \implies (a_{n-1}, a_0, \dots, a_{n-2}) \in C$$

We identify $\mathbb{F}_2[X] / (X^n - 1)$ with $\mathbb{F}_2^n$, letting $\pi(a_0 + a_1 X + \dots + a_{n-1}X^{n-1}) = (a_0, a_1, \dots, a_{n-1})$.

Lemma. A code $C \subseteq \mathbb{F}_2^n$ is cyclic if and only if $\pi(\mathcal{C}) = C$ satisfies

(i) $0 \in \mathcal{C}$;
(ii) $f, g \in \mathcal{C}$ implies $f + g \in \mathcal{C}$;
(iii) $f \in \mathbb{F}_2[X], g \in \mathcal{C}$ implies $fg \in \mathcal{C}$.

Equivalently, $\mathcal{C}$ is an ideal of $\mathbb{F}_2[X] / (X^n - 1)$.

Proof. If $g(X) = a_0 + a_1 X + \dots + a_{n-1}X^{n-1}$, multiplication by X gives $Xg(X) = a_{n-1} + a_0 X + \dots + a_{n-2}X^{n-1}$. So $\mathcal{C}$ is cyclic if and only if (i) and (ii) hold and $g(X) \in C$ implies $Xg(X) \in C$. Linearity then gives (iii). $\square$

We will identify C with $\mathcal{C}$. The cyclic codes of length n correspond to ideals in $\mathbb{F}_2[X] / (X^n - 1)$. Such ideals correspond to ideals of $\mathbb{F}_2[X]$ that contain $X^n - 1$. Since $\mathbb{F}_2[X]$ is a principal ideal domain, these ideals correspond to polynomials $g(X) \in \mathbb{F}_2[X]$ dividing $X^n - 1$.

Theorem. Let $C \trianglelefteq \mathbb{F}_2[X] / (X^n - 1)$ be a cyclic code. Then, there exists a unique *generator polynomial* $g(X) \in \mathbb{F}_2[X]$ such that

(i) $C = (g)$;
(ii) $g(X) \mid X^n - 1$.

In particular, $p(X) \in \mathbb{F}_2[X]$ represents a codeword if and only if $g \mid p$.

Proof. Let $g(X) \in \mathbb{F}_2[X]$ be the polynomial of smallest degree that represents a nonzero codeword of C. Note that $\deg g < n$. Since C is cyclic, $(g) \subseteq C$. Now let $p(X) \in \mathbb{F}_2[X]$ represent a codeword. By the division algorithm, $p = qg + r$ for $q, r \in \mathbb{F}_2[X]$ where $\deg r < \deg g$. Then, $r = p - qg \in C$ as C is an ideal. But $\deg r < \deg g$, so $r = 0$. Hence, $g \mid p$. For part (ii), let $p(X) = X^n - 1$, giving $g \mid X^n - 1$.

Now we show uniqueness. Suppose $C = (g_1) = (g_2)$. Then $g_1 \mid g_2$ and $g_2 \mid g_1$. So $g_1 = cg_2$ where $c \in \mathbb{F}_2^\star$, so $c = 1$. $\square$

Lemma. Let C be a cyclic code of length n with generator $g(X) = a_0 + a_1 X + \dots + a_k X^k$ with $a_k \neq 0$. Then C has basis $\{g, Xg, X^2 g, \dots, X^{n-k-1}g\}$. In particular, $\operatorname{rank} C = n - k$.

Proof. Exercise. $\square$

Corollary. Let C be a cyclic code of length n with generator $g(X) = a_0 + a_1 X + \dots + a_k X^k$

with $a_k \neq 0$. Then, a generator matrix for C is given by

$$G = \begin{pmatrix} a_0 & a_1 & a_2 & \cdots & a_k & 0 & 0 & \cdots & 0 \\ 0 & a_0 & a_1 & \cdots & a_{k-1} & a_k & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & a_0 & a_1 & \cdots & a_k \end{pmatrix}$$

This is an $(n-k) \times n$ matrix.

Definition. Let g be a generator for C. The *parity check polynomial* is the polynomial h such that $g(X)h(X) = X^n - 1$.

Corollary. Writing $h(X) = b_0 + b_1X + \dots + b_{n-k}X^{n-k}$, the parity check matrix is

$$H = \begin{pmatrix} b_{n-k} & b_{n-k-1} & b_{n-k-2} & \cdots & b_1 & b_0 & 0 & 0 & \cdots & 0 \\ 0 & b_{n-k} & b_{n-k-1} & \cdots & b_2 & b_1 & b_0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & b_{n-k} & b_{n-k-1} & b_{n-k-2} & \cdots & b_0 \end{pmatrix}$$

which is a $k \times n$ matrix.

Proof. One can check that the inner product of the ith row of the generator matrix and the jth row of the parity check matrix is the coefficient of $X^{n-k-i+j}$ in $g(X)h(X) = X^n - 1$. Since $1 \leq i \leq n-k$ and $1 \leq j \leq k$, $0 < n-k-i+j < n$, and such coefficients are zero. Hence, the rows of G are orthogonal to the rows of H. Note that as $b_{n-k} \neq 0$, $\operatorname{rank} H = k = \operatorname{rank} C^\perp$, so H is the parity check matrix. $\square$

Remark. Given a polynomial $f(X) = \sum_{i=0}^m f_iX_i$ of degree m, the *reverse* polynomial is $\check{f}(X) = f_n + f_{n-1}X + \dots + f_0X^M = X^m f\left(\frac{1}{X}\right)$. The cyclic code generated by $\check{h}$ is the dual code $C^\perp$.

Lemma. If n is odd, $X^n - 1 = f_1(X) \dots f_t(X)$ where the $f_i(X)$ are distinct irreducible polynomials in $\mathbb{F}_2[X]$. Thus, there are 2^t cyclic codes of length n.

This is false if n is even, for instance, $X^2 - 1 = (X-1)^2$. The proof follows from Galois theory.

5.5 BCH codes

Recall that if p is a prime, $\mathbb{F}_p = {}^{\mathbb{Z}}\!/_{p\mathbb{Z}}$ is a field, and if $f(X) \in \mathbb{F}_p[X]$ is irreducible, the quotient $K = {}^{\mathbb{F}_p[X]}\!/_{(f)}$ is a field and has order $p^{\deg f}$. Moreover, any finite field arises in this way.

If $q = p^\alpha$ is a prime power where $\alpha \geq 1$, there exists a unique field $\mathbb{F}_q$ of order q, up to isomorphism. Note that $\mathbb{F}_q \not\simeq {}^{\mathbb{Z}}\!/_{q\mathbb{Z}}$ if $\alpha > 1$. The multiplicative group $\mathbb{F}_q^\times$ is cyclic; there exists $\beta \in \mathbb{F}_q$ such that $\mathbb{F}_q^\times = \langle \beta \rangle = \left\{1, \beta, \dots, \beta^{q-2}\right\}$. Such a β is called a *primitive element*.

Let n be an odd integer, and let $r \geq 1$ such that $2^r \equiv 1 \bmod n$, which always exists as 2 is coprime to n. Let $K = \mathbb{F}_{2^r}$, and define $\mu_n(K) = \{x \in K \mid x^n = 1\} \leq K^\times$, which is a cyclic group. Since

$n \mid (2^r - 1) = |K^\times|$, $\mu_n(K)$ is the cyclic group of order n. Hence, $\mu_n(K) = \left\{1, \alpha, \alpha^2, \dots, \alpha^{n-1}\right\}$ for some primitive nth root of unity $\alpha \in K$.

Definition. The cyclic code of length n with *defining set* $A \subseteq \mu_n(K)$ is the code

$$C = \left\{ f(X) \in {}^{\mathbb{F}_2[X]}\!/\!_{(X^n - 1)} \,\middle|\, \forall a \in A,\ f(a) = 0 \right\}$$

The generator polynomial $g(X)$ is the nonzero polynomial of least degree such that $g(a) = 0$ for all $a \in A$. Equivalently, g is the least common multiple of the minimal polynomials of the elements of A.

Definition. The cyclic code of length n with defining set $\left\{\alpha, \alpha^2, \dots, \alpha^{\delta-1}\right\}$ is a *BCH code* with *design distance* δ.

Theorem. A BCH code C with design distance δ has minimum distance $d(C) \geq \delta$.

This proof needs the following result.

Lemma. The Vandermonde matrix satisfies

$$\det \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ x_1 & x_2 & x_3 & \cdots & x_n \\ x_1^2 & x_2^2 & x_3^2 & \cdots & x_n^2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_1^{n-1} & x_2^{n-1} & x_3^{n-1} & \cdots & x_n^{n-1} \end{pmatrix} = \prod_{1 \leq j < i \leq n} (x_i - x_j)$$

Proof of theorem. Consider

$$H = \begin{pmatrix} 1 & \alpha & \alpha^2 & \cdots & \alpha^{n-1} \\ 1 & \alpha^2 & \alpha^4 & \cdots & \alpha^{2(n-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha^{\delta-1} & \alpha^{2(\delta-1)} & \cdots & \alpha^{(\delta-1)(n-1)} \end{pmatrix}$$

This is a $(\delta-1)\times n$ matrix. Any collection of $(\delta-1)$ columns is independent as it forms a Vandermonde matrix. But any codeword of C is a dependence relation between the columns of H. Hence every nonzero codeword has weight at least δ, giving $d(C) \geq \delta$. □

Note that H in the proof above is not a parity check matrix, as its entries do not lie in $\mathbb{F}_2$.

Let C be a cyclic code with defining set $\left\{\alpha, \alpha^2, \dots, \alpha^{\delta-1}\right\}$ where $\alpha \in K$ is a primitive nth root of unity. Its minimum distance is at least δ, so we should be able to correct $t = \left\lfloor \frac{\delta-1}{2} \right\rfloor$ errors. Suppose we send $c \in C$ through the channel, and receive $r = c + e$ where e is the error pattern with at most t nonzero errors. Note that r, c, e correspond to polynomials $r(X), c(X), e(X)$, and $c(\alpha^j) = 0$ for $j \in \{1, \dots, \delta - 1\}$ as c is a codeword. Hence, $r(\alpha^j) = e(\alpha^j)$.

Definition. The *error locator polynomial* of an error pattern $e \in \mathbb{F}_2^n$ is

$$\sigma(X) = \prod_{i \in \mathcal{E}} (1 - \alpha^i X) \in K[X]$$

where $\mathcal{E} = \{i \mid e_i = 1\}$.

Assuming that $\deg \sigma = |\mathcal{E}|$, where $2t + 1 \leq \delta$, we must recover σ from $r(X)$.

Theorem. Suppose $\deg \sigma = |\mathcal{E}| \leq t$ where $2t + 1 \leq \delta$. Then $\sigma(X)$ is the unique polynomial in $K[X]$ of least degree such that

(i) $\sigma(0) = 1$;

(ii) $\sigma(X) \sum_{j=1}^{2t} r(\alpha^j) X^j = \omega(X) \bmod X^{2t+1}$ for some $\omega \in K[X]$ of degree at most t.

Proof. Define $\omega(X) = -X\sigma'(X)$, called the *error co-locator*. Hence,

$$\omega(X) = \sum_{i \in \mathcal{E}} \alpha^i X \prod_{j \neq i} (1 - \alpha^j X)$$

This polynomial has $\deg \omega = \deg \sigma$. Consider the ring $K[\![X]\!]$ of formal power series. In this ring,

$$\frac{\omega(X)}{\sigma(X)} = \sum_{i \in \mathcal{E}} \frac{\alpha^i X}{1 - \alpha^i X} = \sum_{i \in \mathcal{E}} \sum_{j=1}^{\infty} (\alpha^i X)^j = \sum_{j=1}^{\infty} X^j \sum_{i \in \mathcal{E}} (\alpha^j)^i = \sum_{j=1}^{\infty} e(\alpha^j) X^j$$

Hence $\sigma(X) \sum_{j=1}^{\infty} e(\alpha^j) X^j = \omega(X)$. By definition of C, we have $c(\alpha^j) = 0$ for all $1 \leq j \leq \delta - 1$. Hence $c(\alpha^j) = 0$ for $1 \leq j \leq 2t$. As $r = c + e$, $r(\alpha^j) = e(\alpha^j)$ for all $1 \leq j \leq 2t$, hence $\sigma(X) \sum_{j=1}^{2t} r(\alpha^j) X^j = \omega(X) \bmod X^{2t+1}$. This verifies (i) and (ii) for this choice of ω, so $\deg \omega = \deg \sigma = |\mathcal{E}| \leq t$.

For uniqueness, suppose there exist $\tilde{\sigma}, \widetilde{\omega}$ with the properties (i), (ii). Without loss of generality, we can assume $\deg \tilde{\sigma} \leq \deg \sigma$. $\sigma(X)$ has distinct nonzero roots, so $\omega(X) = -X\sigma'(X)$ is nonzero at these roots. Hence σ, ω are coprime polynomials. By property (ii), $\tilde{\sigma}(X)\omega(X) = \sigma(X)\widetilde{\omega}(X) \bmod X^{2t+1}$. But the degrees of $\sigma, \tilde{\sigma}, \omega, \widetilde{\omega}$ are at most t, so this congruence is an equality. But $\sigma(X)$ and $\omega(X)$ are coprime, so $\sigma \mid \tilde{\sigma}$, but $\deg \tilde{\sigma} \leq \deg \sigma$ by assumption, so $\tilde{\sigma} = \lambda \sigma$ for some $\lambda \in K$. By property (i), $\sigma(0) = \tilde{\sigma}(0)$ hence $\lambda = 1$, giving $\tilde{\sigma} = \sigma$. □

Suppose that we receive $r(X)$ and wish to decode it.

- Compute $\sum_{j=1}^{2t} r(\alpha^j) X^j$.
- Set $\sigma(X) = 1 + \sigma_1 X + \dots + \sigma_t X^t$, and compute the coefficients of X^i for $t + 1 \leq i \leq 2t$ to obtain linear equations for $\sigma_1, \dots, \sigma_t$, which are of the form $\sum_0^t \sigma_j r(\alpha^{i-j}) = 0$.
- Then solve these polynomials over K, keeping solutions of least degree.
- Compute $\mathcal{E} = \left\{i \mid \sigma(\alpha^{-i}) = 0\right\}$, and check that $|\mathcal{E}| = \deg \sigma$.
- Set $e(X) = \sum_{i \in \mathcal{E}} X^i$, then $c(X) = r(X) + e(X)$, and check that c is a codeword.

Example. Consider $n = 7$, and $X^7-1 = (X+1)(X^3+X+1)(X^3+X^2+1)$ in $\mathbb{F}_2[X]$. Let $g(X) = X^3+X+1$, so $h(X) = (X+1)(X^3+X^2+1) = X^4+X^2+X+1$. The parity check matrix is

$$H = \begin{pmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{pmatrix}$$

The columns are the elements of $\mathbb{F}_2^3 \setminus \{0\}$. This is the Hamming $(7,4)$-code.

Let K be a splitting field for X^7-1; we can take $K = \mathbb{F}_8$. Let $\beta \in K$ be a root of g. Note that $\beta^3 = \beta+1$, so $\beta^6 = \beta^2+1$, so $g(\beta^2) = 0$, and hence $g(\beta^4) = 0$. So the BCH code defined by $\{\beta, \beta^2\}$ has generator polynomial $g(X)$, again proving that this is Hamming's $(7,4)$-code. This code has design distance 3, so $d(C) \geq 3$, and we know Hamming's code has minimum distance exactly 3.

5.6 Shift registers

Definition. A *(general) feedback shift register* is a map $f : \mathbb{F}_2^d \to \mathbb{F}_2^d$ given by

$$f(x_0, \dots, x_{d-1}) = (x_1, \dots, x_{d-1}, C(x_0, \dots, x_{d-1}))$$

where $C : \mathbb{F}_2^d \to \mathbb{F}_2$. We say that the register has length d. The *stream* associated to an *initial fill* $(y_0, \dots, y_{d-1})$ is the sequence $y_0, \dots$ with $y_n = C(y_{n-d}, \dots, y_{n-1})$ for $n \geq d$.

Definition. The general feedback shift register $f : \mathbb{F}_2^d \to \mathbb{F}_2^d$ is a *linear feedback shift register* if C is linear, so

$$C(x_0, \dots, x_{d-1}) = \sum_{i=0}^{d-1} a_i x_i$$

We usually set $a_0 = 1$.

The stream produced by a linear feedback shift register is now given by the recurrence relation $y_n = \sum_{i=0}^{d-1} a_i y_{n-d+i}$. We can define the auxiliary polynomial $P(X) = X^d + a_{d-1}X^{d-1} + \dots + a_1 X + a_0$. We sometimes write $a_d = 1$, so $P(X) = \sum_{i=0}^{d} a_i X^i$.

Definition. The *feedback polynomial* is $\check{P}(X) = a_0 X^d + \dots + a_{d-1}X + 1 = \sum_{i=0}^{d} a_{d-i} X^i$. A sequence $y_0, \dots$ of elements of $\mathbb{F}_2$ has *generating function* $\sum_{j=0}^{\infty} y_j X^j \in \mathbb{F}_2[\![X]\!]$.

Theorem. The stream $(y_n)_{n \in \mathbb{N}}$ comes from a linear feedback shift register with auxiliary polynomial $P(X)$ if and only if its generating function is (formally) of the form $\frac{A(X)}{\check{P}(X)}$ with $A \in \mathbb{F}_2[X]$ such that $\deg A < \deg \check{P}$.

Note that $\check{P}(X) = X^{\deg P} P(X^{-1})$.

Proof. Let $P(X)$ and $\check{P}(X)$ be as above. We require

$$\left(\sum_{j=0}^{\infty} y_j X^j\right)\left(\sum_{i=0}^{d} a_{d-i} X^i\right)$$

to be a polynomial of degree strictly less than d. This holds if and only if the coefficient of X^n in $G(X)\check{P}(X)$ is zero for all $n \geq d$, which is $\sum_{i=0}^{d} a_{d-i} y_{n-i} = 0$. This holds if and only if $y_n = \sum_{i=0}^{d-1} a_i y_{n-d+i}$ for all $n \geq d$. This is precisely the form of a stream that arises from a linear feedback shift register with auxiliary polynomial P. □

The problem of recovering the linear feedback shift register from its stream and the problem of decoding BCH codes both involve writing a power series as a quotient of polynomials.

5.7 The Berlekamp–Massey method

Let $(x_n)_{n\in\mathbb{N}}$ be the output of a binary linear feedback shift register. We wish to find the unknown length d and values $a_0, \dots, a_{d-1}$ such that $x_n + \sum_{i=1}^{d} a_{d-i} x_{n-i} = 0$ for all $n \geq d$. We have

$$\underbrace{\begin{pmatrix} x_d & x_{d-1} & \cdots & x_1 & x_0 \\ x_{d+1} & x_d & \cdots & x_2 & x_1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ x_{2d-1} & x_{2d-2} & \cdots & x_d & x_{d-1} \\ x_{2d} & x_{2d-1} & \cdots & x_{d+1} & x_d \end{pmatrix}}_{A_d} \begin{pmatrix} a_d \\ a_{d-1} \\ \vdots \\ a_1 \\ a_0 \end{pmatrix} = 0$$

We look successively at $A_0 = \begin{pmatrix} x_0 \end{pmatrix}, A_1 = \begin{pmatrix} x_1 & x_0 \\ x_2 & x_1 \end{pmatrix}, \dots$, starting at A_r if we know $d \geq r$. For each A_i, we compute its determinant. If $|A_i| \neq 0$, then $d \neq i$. If $|A_i| = 0$, we solve the system of linear equations on the assumption that $d = i$, giving a candidate for the coefficients $a_0, \dots, a_{d-1}$. This candidate can be checked over as many terms of the stream as desired.

6 Cryptography

6.1 Cryptosystems

We want to modify a message such that it becomes unintelligible to an eavesdropper Eve. Certain secret information is shared between two participants Alice and Bob, called the *key*, chosen from a set of possible keys $\mathcal{K}$. The unencrypted message is called the *plaintext*, which lies in a set $\mathcal{M}$, and the encrypted message is called the *ciphertext*, and lies in a set $\mathcal{C}$. A *cryptosystem* consists of $(\mathcal{K}, \mathcal{M}, \mathcal{C})$ together with the *encryption* function $e : \mathcal{M} \times \mathcal{K} \to \mathcal{C}$ and *decryption* function $d : \mathcal{C} \times \mathcal{K} \to \mathcal{M}$. These maps have the property that $d(e(m,k),k) = m$ for all $m \in \mathcal{M}, k \in \mathcal{K}$.

Example. Suppose $\mathcal{M} = \mathcal{C} = \{A, B, \dots, Z\}^\star = \Sigma^\star$. The *simple substitution cipher* defines $\mathcal{K}$ to be the set of permutations of Σ. To encrypt a message, each letter of plaintext is replaced with its image under a chosen permutation $\pi \in \mathcal{K}$.

The *Vigenère cipher* has $\mathcal{K} = \Sigma^d$ for some d. We identify Σ and $\mathbb{Z}/26\mathbb{Z}$. Write out the key repeatedly below the plaintext, and add each plaintext letter with the corresponding key letter to produce a letter of ciphertext. For instance, encrypting the plaintext ATTACKATDAWN with the key LEMON gives ciphertext LXFOPVEFRNHR. Note, for instance, that each occurrence of the letter A in the plaintext corresponds to a letter of the key in the ciphertext. If $d = 1$, this is the *Caesar cipher*.

6.2 Breaking cryptosystems

Eve may know e and d, as well as the probability distributions of $\mathcal{K}, \mathcal{M}$, but she does not know the key itself. She seeks to recover the plaintext from a given string of ciphertext. There are three possible attack levels.

1. (ciphertext-only) Eve only knows some piece of ciphertext.
2. (known-plaintext) Eve knows a considerable length of plaintext and its corresponding ciphertext, but not the key. In other words, she knows m and $e(m, k)$, but not k.
3. (chosen plaintext) Eve can acquire the ciphertext for any plaintext message; she can generate $e(m, k)$ for any m.

Remark. The simple substitution cipher and Vigenère cipher fail at Level 1 in English if the messages are sufficiently long, as we can perform frequency analysis. Even if the plaintext is suitably random, both examples can fail at Level 2. For modern applications, Level 3 security is desirable.

Consider a cryptosystem $(\mathcal{M}, \mathcal{K}, \mathcal{C})$. We model the keys and messages as independent random variables K, M taking values in $\mathcal{K}, \mathcal{M}$. The ciphertext random variable is $C = e(K, M) \in \mathcal{C}$.

Definition. A cryptosystem $(\mathcal{M}, \mathcal{K}, \mathcal{C})$ has *perfect secrecy* if $H(M \mid C) = H(M)$, or equivalently, M and C are independent, or $I(M; C) = 0$.

One can show that perfect secrecy implies that $|\mathcal{K}| \geq |\mathcal{M}|$.

Definition. The *message equivocation* is $H(M \mid C)$. The *key equivocation* is $H(K \mid C)$.

Lemma. $H(M \mid C) \leq H(K \mid C)$.

Proof. Note that $M = d(C, K)$, hence $H(M \mid C, K) = 0$. Therefore, $H(C, K) = H(M, C, K)$. So

$$\begin{aligned} H(K \mid C) &= H(K, C) - H(C) \\ &= H(M, C, K) - H(M \mid K, C) - H(C) \\ &= H(M, K, C) - H(C) \\ &= H(K \mid M, C) + H(M, C) - H(C) \\ &= H(K \mid M, C) + H(M \mid C) \end{aligned}$$

Hence $H(K \mid C) \geq H(M \mid C)$. $\square$

Let $\mathcal{M} = \mathcal{C} = \mathcal{A}$, and suppose we send n messages modelled as $M^{(n)} = (M_1, \dots, M_n)$ encrypted as $C^{(n)} = (C_1, \dots, C_n)$ using the same key K.

Definition. The *unicity distance* is the least n such that $H\big(K \mid C^{(n)}\big) = 0$; it is the smallest number of encrypted messages required to uniquely determine the key.

Now,

$$\begin{aligned} H\big(K \mid C^{(n)}\big) &= H\big(K, C^{(n)}\big) - H\big(C^{(n)}\big) \\ &= H\big(K, M^{(n)}, C^{(n)}\big) - H\big(C^{(n)}\big) \\ &= H\big(K, M^{(n)}\big) - H\big(C^{(n)}\big) \\ &= H(K) + H\big(M^{(n)}\big) - H\big(C^{(n)}\big) \end{aligned}$$

as $K, M^{(n)}$ are independent. We make the following assumptions.

(i) All keys are equally likely, so $H(K) = \log|\mathcal{K}|$.

(ii) $H\big(M^{(n)}\big) \approx nH$ for some constant H and sufficiently large n.

(iii) All sequences of ciphertext are equally likely, so $H\big(C^{(n)}\big) = n \log|\mathcal{A}|$.

Hence,

$$H\big(K \mid C^{(n)}\big) = \log|\mathcal{K}| + nH - n\log|\mathcal{A}|$$

This is nonnegative if and only if

$$n \leq U = \frac{\log|\mathcal{K}|}{\log|\mathcal{A}| - H}$$

Equivalently, $\frac{\log|\mathcal{A}|}{R\log|\mathcal{A}|}$ where $R = 1 - \frac{H}{\log|\mathcal{A}|}$ is the *redundancy* of the source. Recall that $0 \leq H \leq \log|\mathcal{A}|$. To make the unicity distance large, we can make the number of keys large, or use a message source with little redundancy.

6.3 One-time pad

Consider streams in $\mathbb{F}_2$ representing the plaintext $p_0, p_1, \dots$, the key stream $k_0, k_1, \dots$, and the ciphertext $z_0, z_1, \dots$ where $z_n = p_n + k_n$.

Definition. A *one-time pad* is a cryptosystem where k is generated randomly; the k_i are independent and take values of 0 or 1 with probability $\frac{1}{2}$.

$z = p + k$ is now a stream of independent and identically distributed random variables taking values of 0 or 1 with probability $\frac{1}{2}$. Hence, without the key stream, deciphering is impossible, so the unicity distance is infinite. One can show that a one-time pad has perfect secrecy.

In order to effectively use a one-time pad, we need to generate a random key stream. We then need to share the key stream to the recipient, which is exactly the initial problem. In most applications, the one-time pad is not practical. Instead, we share an initial fill $k_0, \dots, k_{d-1}$ to be used in a shared feedback shift register of length d to generate k. We then apply the following result.

Lemma. Let $x_0, x_1, \dots$ be a stream in $\mathbb{F}_2$ produced by a feedback shift register of length d. Then there exist $M, N \leq 2^d$ such that $x_{N+r} = x_r$ for all $r \geq M$.

Proof. Let the register be $f : \mathbb{F}_2^d \to \mathbb{F}_2^d$, and let $v_i = (x_i, \dots, x_{i+d-1})$. Then for all i, we have $f(v_i) = v_{i+1}$. Since $\left|\mathbb{F}_2^d\right| = 2^d$, the tuples $v_0, v_1, \dots, v_{2^d}$ cannot all be distinct. Let $a < b \leq 2^d$ such that $v_a = v_b$. Let $M = a$ and $N = b - a$, so $v_M = v_{M+N}$ so by induction we have $v_r = v_{r+N}$ for all $r \geq M$. □

Remark. The maximum period of a feedback shift register of length d is 2^d. For a linear feedback shift register, the maximum period is $2^d - 1$; this result is shown on the fourth example sheet.

Stream ciphers using linear feedback shift registers fail at level 2 due to the Berlekamp–Massey method. However, this cryptosystem is cheap, fast, and easy to use. Encryption and decryption can be performed on-the-fly, without needing the entire codeword first, and it is error-tolerant.

Recall that the stream produced by a linear feedback shift register is given by

$$x_n = \sum_{i=1}^{d} a_{d-i} x_{n-i}$$

for all $n \geq d$, and has auxiliary polynomial

$$P(X) = X^d + a_{d-1}X^{d-1} + \dots + a_0$$

with $a_d = 1$. The solutions to the recursion relations are linear combinations of powers of roots of P. Over $\mathbb{C}$, the general solution is a linear combination of $\alpha^n, n\alpha^n, \dots, n^{t-1}\alpha^n$ where α is a root of $P(X)$ with multiplicity t.

As $n^2 = n$ in $\mathbb{F}_2$, we cannot use this method directly. First, we must work in a splitting field K of P, a field containing $\mathbb{F}_2$ in which P is expressible as a product of linear factors. In addition, we replace the $n^i\alpha^n$ term with $\binom{n}{i}\alpha^n$. The general solution is now a linear combination of these terms in K.

We can also generate new key streams from old ones.

Lemma. Let $(x_n), (y_n)$ be outputs from linear feedback shift registers of length M, N respectively. Then,

(i) the sequence $(x_n + y_n)$ is the output of a linear feedback shift register of length $M + N$;
(ii) the sequence $(x_n y_n)$ is the output of a linear feedback shift register of length MN.

The following proof is non-examinable.

Proof. Assume for simplicity that the auxiliary polynomials $P(X), Q(X)$ each have distinct roots α_1, α_M and $\beta_1, \dots, \beta_N$ in a field K extending $\mathbb{F}_2$. Then $x_n = \sum_{i=1}^{M} \lambda_i \alpha_i^n$ and $y_n = \sum_{i=1}^{N} \mu_j \beta_j^n$ where $\lambda_i, \mu_j \in K$. Now, $x_n + y_n = \sum_{i=1}^{M} \lambda_i \alpha_i^n \sum_{i=1}^{N} \mu_j \beta_j^n$ is produced by a linear feedback shift register with auxiliary polynomial $P(X)Q(X)$. For the second part, $x_n y_n = \sum_{i=1}^{M} \sum_{j=1}^{n} \lambda_i \mu_j (\alpha_i \beta_j)^n$ is the output of a linear feedback shift register with auxiliary polynomial $\prod_{i=1}^{N} \prod_{j=1}^{M} (X - \alpha_i \beta_j)$. □

Adding outputs of linear feedback shift registers is no more economical than producing the same string with a single linear feedback shift register. Muliplying streams does increase the effective length of the linear feedback shift register, but $x_n y_n = 0$ when either x_n or y_n are zero, so we gain little extra data. Nonlinear feedback shift registers are in general hard to analyse; in particular, an eavesdropper may understand the feedback shift register better than Alice and Bob.

6.4 Asymmetric ciphers

Stream ciphers are examples of symmetric cryptosystems. In such a system, the decryption process is the same, or is easily deduced from, the encryption process. In an asymmetric cryptosystem, the key is split into two parts: the *private key* for decryption, and the *public key* for encryption. Knowing the encryption and decryption processes and the public key, it should still be hard to find the private key or to decrypt the messages. This aim implies security at level 3. In this case, there is also no key exchange problem, since the public key can be broadcast on an open channel.

We base asymmetric cryptosystems on certain mathematical problems in number theory which are believed to be 'hard', such as the following.

(i) Factoring. Let $N = pq$ for p, q large prime numbers. Given N, the task is to find p and q.

(ii) Discrete logarithm problem. Let p be a large prime and g be a primitive root mod p (a generator of $\mathbb{F}_p^\star$). Given x, we wish to find a such that $x \equiv g^a \bmod p$.

Definition. An algorithm runs in *polynomial time* if the number of operations needed to perform the algorithm is at most cN^d where N is the input size, and c, d are constants.

Example. An algorithm for factoring N has input size $\log_2 N$, roughly the number of bits in its binary expansion. Polynomial time algorithms include arithmetic operations on integers including the division algorithm, computation of greatest common divisors, and the Euclidean algorithm. We can also compute $x^\alpha \bmod N$ in polynomial time using repeated squaring; this is called modular exponentiation. Primality testing can be performed in polynomial time.

Polynomial time algorithms are not known for examples (i) and (ii) above. However, we have elementary methods for computing (i) and (ii) that take exponential time. If $N = pq$, dividing N by successive primes up to $\sqrt{N}$ will find p and q but takes $O(\sqrt{N}) = O(2^{\frac{B}{2}})$ steps where $B = \log_2 N$.

We describe the *baby-step, giant-step* algorithm for the discrete logarithm problem. Set $m = \lceil \sqrt{p} \rceil$, and write $a = qm + r$ for $0 \leq q, r < m$. Then, $x \equiv g^a = g^{qm+r} \bmod p$, so $g^{qm} = g^{-r}x \bmod p$. We list all values of g^{qm} and $g^{-r}x \bmod p$; we then sort the lists and search for a match. This takes $O(\sqrt{p} \log p)$ steps.

The best known methods for solving the examples above use a factor base method, called the *modular number sieve*. It has running time

$$O\left(\exp\left(c(\log N)^{\frac{1}{3}}(\log\log N)^{\frac{2}{3}}\right)\right)$$

where c is a known constant.

6.5 Rabin cryptosystem

Recall that *Euler's totient function* is denoted φ, where $\varphi(n)$ is the number of integers less than n which are coprime to n. Equivalently, $\varphi(n) = \left|\left(\mathbb{Z}/n\mathbb{Z}\right)^\times\right|$. By Lagrange's theorem, $a^{\varphi(N)} \equiv 1 \bmod N$ for each a coprime to N; this result is sometimes known as the Fermat–Euler theorem. If $N = p$ is prime, $a^{p-1} \equiv 1 \bmod p$, which is Fermat's little theorem.

Lemma. Let $p = 4k - 1$ be a prime, and let $d \in \mathbb{Z}$. If $x^2 \equiv d \bmod p$ is soluble, one solution is $x \equiv d^k \bmod p$.

Proof. Suppose x_0 is a solution, so $x_0^2 \equiv d \bmod p$. Without loss of generality we can assume $x_0 \not\equiv 0$, or equivalently, $x_0 \nmid p$. Then $x_0^2 \equiv d$ so $d^{2k-1} \equiv x_0^{2(2k-1)} \equiv x_0^{p-1} \equiv 1$. Hence, $\left(d^k\right)^2 \equiv d$. $\square$

In the Rabin cryptosystem, the private key consists of two large distinct primes $p, q \equiv 3 \bmod 4$. The public key is $N = pq$. $\mathcal{M} = \mathcal{C} = \{1, \dots, N-1\} = \mathbb{Z}_N^\times$. We encrypt a plaintext message m as $c = m^2 \bmod N$. Usually, we restrict our messages so that $(m, N) = 1$ and $m > \sqrt{N}$.

Receiving ciphertext c, we can solve for x_1, x_2 such that $x_1^2 \equiv c \bmod p$ and $x_2^2 \equiv c \bmod q$ using the previous lemma. Then, applying the Chinese remainder theorem, we can find x such that $x \equiv x_1 \bmod p$ and $x \equiv x_2 \bmod q$, hence $x^2 \equiv c \bmod N$. Indeed, running the Euclidean algorithm on p, q gives integers r, s such that $rp + sq = 1$, then we can take $x = sqx_1 + rpx_2$.

Lemma. (i) Let p be an odd prime, and let $(d, p) = 1$. Then $x^2 \equiv d \bmod p$ has no solutions or exactly two solutions.
(ii) Let $N = pq$ where p, q are distinct odd primes, and let $(d, N) = 1$. Then $x^2 \equiv d \bmod N$ has no solutions or exactly four solutions.

Proof. Part (i). $x^2 \equiv y^2 \bmod p$ if and only if $p \mid (x^2 - y^2) = (x-y)(x+y)$, so either $p \mid x - y$ or $p \mid x + y$, so $x = \pm y$.

Part (ii). If x_0 is a solution, then by the Chinese remainder theorem, there exist solutions x with $x \equiv \pm x_0 \bmod p$ and $x \equiv \pm x_0 \bmod q$. This gives four solutions as required. By (i), these are the only possible solutions. □

Hence, to decrypt the Rabin cipher, we must find all four solutions to $x^2 \equiv c \bmod N$. Messages should include enough redundancy to uniquely determine which of these four solutions is the intended plaintext.

Theorem. Breaking the Rabin cryptosystem is essentially as difficult as factoring N.

Proof. If we can factorise N as pq, we have seen that we can decrypt messages. Conversely, suppose we can break the cryptosystem, so we have an algorithm to find square roots modulo N. Choose $x \bmod N$ at random, and use the algorithm to find y such that $y^2 \equiv x^2 \bmod N$. With probability $\frac{1}{2}$, $x \not\equiv \pm y \bmod N$. Then, $(N, x-y)$ is a nontrivial factor of N. If this fails, choose another x, and repeat until the probability of failure $\left(\frac{1}{2}\right)^r$ is acceptably low. □

6.6 RSA cryptosystem

Suppose $N = pq$ where p, q are distinct odd primes. We claim that if we know a multiple m of $\varphi(N) = (p-1)(q-1)$, then factoring N is 'easy'. Write $o_p(x)$ for the order of x as an element of $\left(\mathbb{Z}/p\mathbb{Z}\right)^\times$. Write $m = 2^a b$ where $a \geq 1$, b odd. Let

$$X = \left\{x \in \left(\mathbb{Z}/N\mathbb{Z}\right)^\times \,\middle|\, o_p(x^b) \neq o_q(x^b)\right\}$$

Theorem. (i) If $x \in X$, then there exists $0 \leq t < a$ such that $(x^{2^t b} - 1, N)$ is a nontrivial factor of N.
(ii) $|X| \geq \frac{1}{2}\left|\left(\mathbb{Z}/N\mathbb{Z}\right)^\times\right| = \frac{1}{2}(p-1)(q-1)$.

Proof. Part (i). By the Fermat–Euler theorem, $x^{\varphi(N)} \equiv 1 \bmod N$. Hence $x^m \equiv 1 \bmod N$. But $m = 2^a b$, so setting $y = x^b \bmod N$, we obtain $y^{2^a} \equiv 1 \bmod N$. In particular, $o_p(y)$ and $o_q(y)$ are powers of 2. Since $x \in X$, $o_p(y) \neq o_q(y)$, so without loss of generality suppose $o_p(y) < o_q(y)$. Let $o_p(y) = 2^t$, so $0 \leq t < a$. Then $y^{2^t} \equiv 1 \bmod p$, but $y^{2^t} \not\equiv 1 \bmod q$. So $(y^{2^t} - 1, N) = p$ as required. □

The proof of part (ii) will be seen later.

In the RSA cryptosystem, the private key consists of large distinct primes p, q chosen at random. Let $N = pq$, and choose the *encrypting exponent* e randomly such that $(e, \varphi(N)) = 1$, for instance taking e prime larger than p, q. By Euclid's algorithm, there exist d, k such that $de - k\varphi(N) = 1$; d is called the *decrypting exponent*.

The public key is (N, e), and we encrypt $m \in \mathcal{M}$ as $c \equiv m^e \bmod N$. The private key is (N, d), and we decrypt $c \in \mathcal{C}$ as $x \equiv c^d \bmod N$. By the Fermat–Euler theorem, $x \equiv m^{de} \equiv m^{1+k\varphi(N)} \equiv m \bmod N$, noting that the probability that $(m, N) \neq 1$ is small enough to be ignored. Hence, the decrypting function is inverse to the encrypting function.

Corollary. Finding the RSA private key (N, d) is essentially as difficult as factoring N.

Proof. We have already shown that if we can factorise N, we can find d. Conversely, suppose there is an algorithm to find d given N and e. Then $de \equiv 1 \bmod \varphi(N)$. Taking $m = de - 1$ in the proof of part (i) of the theorem above, we can factorise N. If this fails, repeat until the probability of failure is acceptably low. After r such random choices, we find a factor of N with probability $1 - \left(\frac{1}{2}\right)^r$. □

We now prove part (ii) of the above theorem.

Proof. The Chinese remainder theorem provides a multiplicative group isomorphism

$$\left(\mathbb{Z}/N\mathbb{Z}\right)^\times \to \left(\mathbb{Z}/p\mathbb{Z}\right)^\times \times \left(\mathbb{Z}/q\mathbb{Z}\right)^\times$$

mapping x to $(x \bmod p, x \bmod q)$. We claim that if we partition $\left(\mathbb{Z}/p\mathbb{Z}\right)^\times$ according to the value of $o_p(x^b)$, then each equivalence class has size at most

$$\frac{1}{2}\left|\left(\mathbb{Z}/p\mathbb{Z}\right)^\times\right| = \frac{1}{2}(p-1)$$

We show that one of these subsets has size exactly $\frac{1}{2}(p-1)$. Let g be a primitive root mod p, so $\left(\mathbb{Z}/p\mathbb{Z}\right)^\times = \langle g \rangle$. By Fermat's little theorem, $g^{p-1} \equiv 1 \bmod p$, so $g^m = g^{2^a b} \equiv 1 \bmod p$. Hence, $o_p(g^b)$ is a power of 2, say $2^t \leq a$. Let $x = g^k$ for some $0 \leq k \leq p-2$, then $x^b = (g^b)^k$, so $o_p(x^b) = \frac{2^t}{(2^t, k)}$. So $o_p(x^b) = 2^t$ if and only if k is odd, so

$$o_p(x^b) = o_p(g^{bk}) = \begin{cases} o_p(g^b) = 2^t & \text{if } k \text{ odd} \\ < 2^t & \text{if } k \text{ even} \end{cases}$$

Thus, $\{g^k \bmod p \mid k \text{ odd}\}$ is the set as required, proving the claim. To finish, for each $y \in \left(\mathbb{Z}/q\mathbb{Z}\right)^\times$, the set

$$\left\{x \in \left(\mathbb{Z}/p\mathbb{Z}\right)^\times \,\middle|\, o_p(x^b) \neq o_q(x^b)\right\}$$

has at least $\frac{1}{2}(p-1)$ elements. Applying the Chinese remainder theorem,

$$|X| = \left|\left\{(x,y) \in \left(\mathbb{Z}/p\mathbb{Z}\right)^\times \times \left(\mathbb{Z}/q\mathbb{Z}\right)^\times \,\middle|\, o_p(x^b) \neq o_q(x^b)\right\}\right| \geq \frac{1}{2}(p-1)(q-1) = \frac{1}{2}\varphi(N)$$

□

Remark. We have shown that finding (N, d) from the public key (N, e) is as hard as factoring N. It is unknown whether decrypting messages sent via RSA is as hard as factoring.

RSA avoids the issue of needing to share keys, but it is slow. Symmetric ciphers are often faster.

Example (Shamir's padlock example). Let $\mathcal{A} = \mathbb{Z}_p$. Alice chooses $a \in \mathbb{Z}_{p-1}^\star$ and computes g^a. She finds a' such that $aa' = 1 \bmod p-1$. Bob chooses $b \in \mathbb{Z}_{p-1}^\star$ and computes g^b. He similarly finds b' such that $bb' = 1 \bmod p-1$.

Let m be a message in $\mathbb{Z}_p$. She encodes m as $c = m^a \bmod p$. She then sends this to Bob, who computes $d = c^b \bmod p$. He sends this back to Alice, who computes $e = d^{a'} \bmod p$. She sends this back to Bob, who computes $e^{b'} \bmod p$. By Fermat's little theorem, $e^{b'} \equiv d^{a'b'} \equiv c^{ba'b'} \equiv m^{aba'b'} \equiv m$.

$$m \xrightarrow{A} m^a \xrightarrow{B} c^b \xrightarrow{A} d^{a'} \xrightarrow{B} e^{b'}$$

Example (Diffie–Hellman key exchange). Alice and Bob wish to agree on a secret key k. Let p be a large prime, and g a primitive root mod p. Alice chooses an exponent $\alpha \in \mathbb{Z}_{p-1}$ and sends g^α mod p to Bob. Bob chooses an exponent β and sends g^β mod p to Alice. Both Alice and Bob compute $k = g^{\alpha\beta}$, which can be used as their secret key. An eavesdropper must find $g^{\alpha\beta}$ knowing g, g^α, and g^β. Diffie and Hellman conjectured that this problem is as difficult as solving the discrete logarithm problem.

6.7 Secrecy and attacks

Consider a message m sent by Alice to Bob. Here are some possible aims that the participants may have in communication.

(i) *Secrecy*: Alice and Bob can be sure that no third party can read the message.

(ii) *Integrity*: Alice and Bob can be sure that no third party can alter the message.

(iii) *Authenticity*: Bob can be sure that Alice sent the message.

(iv) *Non-repudiation*: Bob can prove to a third party that Alice sent the message.

Example (authenticity using RSA). Suppose Alice uses a private key (N, d) to encrypt m. Anyone can decrypt m using the public key (N, e) as $(m^d)^e = (m^e)^d = m$, but they cannot forge a message sent by Alice. Suppose Bob picks a random message m and sends it to Alice; if Bob then receives a message back from Alice which after decryption ends in m, then he can be sure it comes from Alice.

Signature schemes preserve integrity and non-repudiation. They also prevent tampering in the following sense.

Example (homomorphism attack)**.** Suppose a bank sends messages of the form (M_1, M_2) where M_1 represents the client's name and M_2 represents an amount of money to be transferred into their account. Suppose that messages are encoded using RSA as $(Z_1, Z_2) = (M_1^e, M_2^e)$, where all calculations are performed modulo N. A client C transfers £100 to their account, and observes the encrypted message (Z_1, Z_2). Then, sending (Z_1, Z_2^3) to the bank, C becomes a millionaire without breaking RSA. Alternatively, one could simply send (Z_1, Z_2) to the bank many times, gaining more money each time; this particular attack is defeated by timestamping the messages.

Definition. A message m is *signed* as (m, s) where the *signature* $s = s(m, k)$ is a function of m and the private key k.

The recipient can check the signature using the public key to verify authenticity of the message. The signature function or *trapdoor* function $s : \mathcal{M} \times \mathcal{K} \to \mathcal{S}$ is designed such that without knowledge of the private key, one cannot sign messages, but anyone can check whether a signature is valid. Note that the signature is associated to each message, not to each sender.

Example (signatures using RSA)**.** Suppose Alice has a private key (N, d), and broadcasts a public key (N, e). She signs a message m as (m, s) where $s = m^d \bmod N$. The signature is verified by checking $s^e = m$.

This technique is vulnerable to the homomorphism attack. This is also vulnerable to the *existential forgery* attack, in which an attacker produces valid signed messages of the form $(s^e \bmod N, s)$ after choosing s first. Hopefully, such messages are not meaningful.

To solve these problems, we could use a better signature scheme. In addition, rather than signing a message m, we instead sign the *digest* $h(m)$ where $h : \mathcal{M} \to \{1, \dots, N-1\}$ is a *hash* function. A hash function is a publicly known function for which it is very difficult to find pairs of messages with matching hashes; such a pair is called a *collision*. Examples of hash functions include MD5 and the SHA family.

6.8 Elgamal signature scheme

Alice chooses a large prime p and a random integer u with $1 < u < p$. Let g be a primitive root mod p. The public key is $p, g, y = g^u \bmod p$. The private key is u. Let $h : \mathcal{M} \to \{1, \dots, p-1\}$ be a collision-resistant hash function.

To send a message m with $0 \leq m \leq p-1$, Alice randomly chooses k with $1 \leq k \leq p-2$ coprime to $p-1$. She computes r, s with $1 \leq r \leq p-1$ and $1 \leq s \leq p-2$ satisfying

$$r \equiv g^k \mod p; \quad h(m) \equiv ur + ks \mod (p-1)$$

Since k is coprime to $p-1$, the congruence for s always has a solution. Alice signs the message with the signature (r, s). Now,

$$g^{h(m)} \equiv g^{ur+ks} \equiv (g^u)^r (g^k)^s \equiv y^r r^s \mod p$$

Bob accepts a signature if $g^{h(m)} \equiv y^r r^s \bmod p$. To forge a signature, obvious attacks involve the discrete logarithm problem, finding u from $y = g^u$.

Lemma. Let $a, b, m \in \mathbb{N}$ and consider the congruence $ax \equiv b \bmod m$. This has either no solutions or $\gcd(a, m)$ solutions for $x \bmod m$.

Proof. Let $d = \gcd(a, m)$. If $d \nmid b$, there is no solution. If $d \mid b$, we can rewrite the congruence as $\frac{a}{d}x \equiv \frac{b}{d} \bmod \frac{m}{d}$. Note that $\frac{a}{d}, \frac{m}{d}$ are coprime, so this congruence has a unique solution. $\square$

It is vital that Alice chooses a new value of k to sign each message. Suppose she sends m_1, m_2 using the same value of k. Denote the signatures (r, s_1) and (r, s_2); note that r depends only on k and is hence fixed.

$$h(m_1) \equiv ur + ks_1 \mod (p-1); \quad h(m_2) \equiv ur + ks_2 \mod (p-1)$$

Hence,

$$h(m_1) - h(m_2) \equiv k(s_1 - s_2) \mod (p-1)$$

Let $d = \gcd(p-1, s_1 - s_2)$. By the previous lemma, this is the number of solutions for k modulo $p-1$. Choose the solution that gives the correct value in the first congruence $r \equiv g^k \bmod p$. Then,

$$s_1 \equiv \frac{h(m_1) - ur}{k} \mod (p-1)$$

This gives $ur \equiv h(m_1) - ks_1$. Hence, using the lemma again, there are $\gcd(p-1, r)$ solutions for u. Choose the solution for u that gives $y \equiv g^u$. This allows us to deduce Alice's private key u, as well as the exponent k used in both messages.

6.9 The digital signature algorithm

The digital signature algorithm is a variant of the Elgamal signature scheme developed by the NSA. The public key is (p, q, g) constructed as follows.

- Let p be a prime of exactly N bits, where N is a multiple of 64 such that $512 \leq N \leq 1024$, so $2^{N-1} < p < 2^N$.
- Let q be a prime of 160 bits, such that $q \mid p - 1$.
- Let $g \equiv h^{\frac{p-1}{q}} \bmod p$, where h is a primitive root mod p; in particular, g is an element of order q in $\mathbb{Z}_p^\times$.
- Alice chooses a private key x with $1 < x < q$ and publishes $y = g^x$.

Let m be a message with $0 \leq m < q$. She chooses a random k with $1 < k < q$, and computes

$$s_1 \equiv (g^k \bmod p) \bmod q; \quad s_2 \equiv k^{-1}(m + xs_1) \bmod q$$

The signature is (s_1, s_2). To verify a signature, we perform the following procedure. Bob computes $w \equiv s_2^{-1} \bmod q$, $u_1 \equiv mw \bmod q$, $u_2 \equiv s_1 w \bmod q$, and $v = (g^{u_1} y^{u_2} \bmod p) \bmod q$. He accepts the signature if $v = s_1$.

Proposition. If a message is signed with the DSA and the message is not manipulated, the signature is accepted.

Proof. First, note that $(m + xs_1)w = ks_2s_2^{-1} \bmod q$. Now, as $g^q = 1 \bmod p$,

$$
\begin{aligned}
v &= (g^{u_1} y^{u_2} \bmod p) \bmod q \\
&= (g^{mw} g^{xs_1 w} \bmod p) \bmod q \\
&= (g^{(m+xs_1)w} \bmod p) \bmod q \\
&= (g^k \bmod p) \bmod q \\
&= s_1
\end{aligned}
$$

Hence, for a correctly signed message, the verification succeeds. □

Suppose that Alice sends m_1 to Bob and m_2 to Carol, and provides signatures for each message using the DSA. One can show that if Alice uses the same value of k for both transmissions, it is possible for an eavesdropper to recover the private key x from the signed messages.

6.10 Commitment schemes

Suppose Alice wants to send a bit $m \in \{0, 1\}$ to Bob in such a way that

(i) Bob cannot determine the value of m without Alice's help; and

(ii) Alice cannot change the bit once she has sent it.

Such a system can be used for coin tossing: suppose Alice and Bob are in different rooms, where Alice tosses a coin and Bob guesses the result. The result of the coin and Bob's guess can be viewed as messages of this form. As another example, consider a poll whose result cannot be viewed until everyone has voted. We will see two examples of such a *commit-and-reveal* strategy, known as *bit commitment*.

Suppose that we have a publicly known encryption function e_A and a decryption function d_A known only to Alice. Alice makes a choice for her message m, and commits to Bob the ciphertext $c = e_A(m)$. Under the assumption that the cipher is secure, Bob cannot decipher the message. To reveal her choice, Alice sends her private key to Bob, who can then use it to decipher the message $d_A(c) = d_A(e_A(m)) = m$. He can also check that d_A, e_A are inverse functions and thus ensure that Alice sent the correct private key.

Alternatively, suppose that Alice has two ways to communicate to Bob: a clear channel which transmits with no errors, and a binary symmetric channel with error probability p. Suppose $0 < p < \frac{1}{2}$, and the noisy channel corrupts bits independent of any action of Alice or Bob, so neither can affect its behaviour. Bob publishes a binary linear code C of length N and minimum distance d, and Alice publishes a random non-trivial linear map $\theta \colon C \to \mathbb{F}_2$. To send a bit $m \in \mathbb{F}_2$, Alice chooses a random codeword $c \in C$ such that $\theta(c) = m$, and sends c to Bob via the noisy channel. Bob receives $r = c + e \in \mathbb{F}_2^N$ where e is the error pattern. The expected value of $d(r, c) = d(e, 0)$ is Np. N is chosen such that $Np \gg d$, so Bob cannot tell what the original codeword c was, and hence cannot find $\theta(c) = m$.

To reveal, Alice sends c to Bob using the clear channel. Bob can check that $d(c, r) \approx Np$; if so, he accepts the message. It is possible that many more or many fewer bits of c were corrupted by the noisy channel, which may make Bob reject the message even if Alice correctly committed and revealed the message. N, d should be chosen such that the probability of this occurring is negligible.

We have shown that Bob cannot read Alice's guess until she reveals it. In addition, Alice cannot cheat by changing her guess, because she knows c but not how it was corrupted by the noisy channel. All she knows is that the received message r has distance approximately Np from c. If she were to send $c' \neq c$, she must ensure that $d(r, c') \approx Np$, but the probability that this happens is small unless she chooses c' very close to c. But any two distinct codewords have distance at least d, so she cannot cheat.

6.11 Secret sharing schemes

Suppose that the CMS is attacked by the MIO. The Faculty will retreat to a bunker known as MR2. Entry to MR2 is controlled by a *secret*, which is a positive integer S. This secret is known only to the Leader. Each of the n members of the Faculty knows a pair of numbers, called their *shadow* or *share*. It is required that, in the absence of the Leader, any k members of the Faculty can reconstruct the secret from their shadows, but any $k-1$ cannot.

Definition. Let $k, n \in \mathbb{N}$ with $k < n$. A (k, n)*-threshold scheme* is a method of sharing a message S among a set of n participants such that any subset of k participants can reconstruct S, but no subset of smaller size can reconstruct S.

We discuss Shamir's method for implementing such a scheme. Let $0 \leq S \leq N$ be the secret, which can be chosen at random by the Leader. The Leader chooses and publishes a prime $p > n, N$. They then choose independent random coefficients $a_1, \dots, a_{k-1}$ with $0 \leq a_j \leq p-1$ where we take $a_0 = S$, and distinct integers $x_1, \dots, x_n$ with $1 \leq x_j \leq p-1$. Define

$$P(r) \equiv a_0 + \sum_{j=1}^{k-1} a_j x_r^j \mod p$$

choosing $0 \leq P(r) \leq p-1$. The rth participant is given their shadow pair $(x_r, P(r))$ to be kept secret. The Leader can then discard their computations.

Suppose k members of the Faculty assemble with shadow pairs $(y_j, Q(j)) = (x_{i_j}, P(i_j))$ for $1 \leq j \leq k$. By properties of the Vandermonde determinant,

$$\det \begin{pmatrix} 1 & y_1 & \cdots & y_1^{k-1} \\ 1 & y_2 & \cdots & y_2^{k-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & y_k & \cdots & y_k^{k-1} \end{pmatrix} = \prod_{1 \leq j < i \leq k} (y_i - y_j)$$

The y_i are distinct, so this determinant does not vanish. Hence, we can uniquely solve the system of k simultaneous equations

$$\begin{aligned} z_0 + y_1 z_1 + y_1^2 z_2 + \dots + y_k^{k-1} z_{k-1} &\equiv Q(1) \\ z_0 + y_2 z_1 + y_2^2 z_2 + \dots + y_2^{k-1} z_{k-1} &\equiv Q(2) \\ &\vdots \\ z_0 + y_k z_1 + y_k^2 z_2 + \dots + y_k^{k-1} z_{k-1} &\equiv Q(k) \end{aligned}$$

In particular, $z_0 = a_0 = S$ is the secret, as $(a_0, \dots, a_{k-1})$ is also a solution to these equations by construction. Suppose $k-1$ people attempt to reconstruct the secret. In this case, the Vandermonde determinant gives

$$\det\begin{pmatrix} y_1 & y_1^2 & \cdots & y_1^{k-1} \\ y_2 & y_2^2 & \cdots & y_2^{k-1} \\ \vdots & \vdots & \ddots & \vdots \\ y_{k-1} & y_{k-1}^2 & \cdots & y_{k-1}^{k-1} \end{pmatrix} = y_1 y_2 \dots y_{k-1} \prod_{1\le j<i\le k-1} (y_i - y_j)$$

This is nonzero modulo p, so the system of equations

$$\begin{aligned} z_0 + y_1 z_1 + y_1^2 z_2 + \cdots + y_k^{k-1} z_{k-1} &\equiv Q(1) \\ z_0 + y_2 z_1 + y_2^2 z_2 + \cdots + y_2^{k-1} z_{k-1} &\equiv Q(2) \\ &\vdots \\ z_0 + y_{k-1} z_1 + y_{k-1}^2 z_2 + \cdots + y_{k-1}^{k-1} z_{k-1} &\equiv Q(k-1) \end{aligned}$$

has solutions for $z_1, \dots, z_{k-1}$ regardless of the value of z_0. Thus, $k-1$ members of the Faculty cannot reconstruct the secret S, or even tell which values are more likely than others.

Remark. Note that a polynomial of degree $k-1$ can be recovered from its values at k points, but not on fewer points; this technique is known as Lagrange interpolation. The secret shadow pairs can be changed without altering the secret S; the Leader simply chooses a different random polynomial with the same constant term. Changing the polynomial frequently can increase security, since any eavesdropper who has gathered some shadow pairs generated from one polynomial cannot use that information to help decrypt a different polynomial.

Example. Consider a $(3, n)$-threshold scheme, where ordinary workers in a company have single shares, the vice presidents have two shares, and the Leader has three. In this case, the secret can be recovered by any three ordinary workers, any two workers if one of them is a vice president, or the Leader alone. In such *hierarchical schemes*, the 'importance' of individuals determines how many of them are required to recover the secret.

Example. Suppose Alice has a private key that she wishes to store securely and reliably. She uses a $(k, 2k-1)$-threshold scheme, where she forms $2k-1$ shadow pairs and stores them in different locations. As long as she does not lose more than half of the pairs, she can recover her key, hence the scheme is reliable. An eavesdropper needs to steal more than half of the pairs in order to recover the key, hence the scheme is secure.