18/01/13

Coding and Cryptography ①

SM/37

Information

This course deals with the transmission of information from a sender A to a reciever B. To be transmitted via some channel, the information needs to be encoded in the correct form, and possibly encrypted, if it is to remain confidential.

Definition

An alphabet A is a finite set of symbols.

Examples

$\{a, b, c, \ldots, z, \text{"space"}\}$ or $\{0, 1\}$.

Symbols from A are called letters. A message or word is a finite sequence of letters from A, and is usually denoted by juxtaposition of letters (e.g. $m = a_1 a_2 \ldots a_{N-1} a_N$). The set of all words from A is denoted A^*.

To convey a message m successfully, A needs to transform it into a suitable form. So A encodes the message into a new alphabet B via a coding function $c : A \to B^*$, that codes each letter from alphabet A into a finite sequence from B. These finite sequences are codewords (CWS).

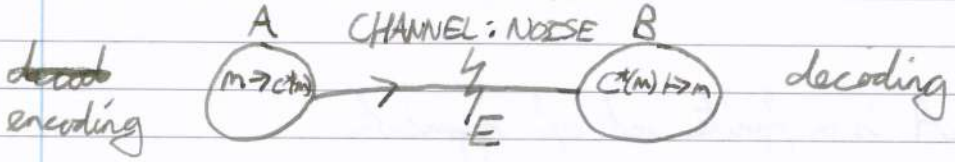
The entire message m is then encoded as

$$c^*(m) := c(a_1)\, c(a_2) \ldots c(a_{N-1})\, c(a_N) \in B^*$$

This is then transmitted via a channel to B who decodes it to recover m.

If $|A| = n$, $|B| = a$, we say that c is an a-ary code of size n.

The function c may be chosen so that only the intended recipient can decode it (called encryption). But it could be encoded simply to compress the message or reduce the likelihood of errors being introduced in the transmission.



Example

- A sends m = "Call at 2pm"
- Encode as binary strings using ASCII, e.g. $c(C) = 1000011$

  $c^*(m) = 1000011 | \ldots | \ldots$ | blocks of length 7
- Transmit via the internet
- Decoded by B.

Examples

- Morse code, audio CDs (error correction), SMS (text compression), zip files (compression), PIN numbers (cryptography), jpeg files (compression and data

Prerequisites

- IA Probability (expectation, Law of Large Numbers)
- Numbers and Sets (modular arithmetic, CRT, FLT)
- Linear Algebra (vector spaces)
- GRM (finite fields)

18/01/13

Coding and Cryptography ①

Books

Goldie-Pinch, Welsh, Keith Carne's notes, Körner's notes

Plan

1. Noiseless Coding
2. Error correcting codes
3. Shannon's Theorems
4. Linear codes, cyclic codes
5. Cryptography "The science of encipherment"

1. Noiseless Coding

1 Prefix-free Codes

Encode $m = a_1 \dots a_N$ using function $c : A \to B^*$ to obtain cw $c^*(m) := c(a_1) \dots c(a_N)$. Here $c^* : A^* \to B^*$. We need to decode $c^*(m)$ and recover the original m. Hence we want c^* injective, in which case we say that c is decodable (decipherable).

Remark

For c to be decodable, $c : A \to B^*$ must be injective, but this is not sufficient.

$A = \{1, 2, 3, 4\}$, $B = \{0, 1\}$

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

$c^*(114) = 0001 = c^*(312)$, so c is injective but not decodable.

1.1 Definition

Take $w \in B^*$. Another word w' is a prefix of w if w is

the concatenation of w' with another word w'' such that $w = w'w''$

A code $c: A \to B^*$ is prefix-free if no codeword $c(a)$ is a prefix of a different codeword $c(a')$.

Example

$c: \{0,1,2,3\} \to \{0,1\}^*$ $\quad c: \begin{matrix} 0 \mapsto 0 \\ 1 \mapsto 10 \\ 2 \mapsto 110 \\ 3 \mapsto 111 \end{matrix}$ is prefix-free.

If we reversed each cw this is <u>not</u> prefix-free.

Assuming that c is injective, the following codes are always decodable

- Prefix-free code is decodable (if we recieve $w \in B^*$ we can decode by examining prefixes of w. One of these must be $c(a_1)$ for some $a_1 \in A$. Since the code is prefix free, a is unique. Delete this process and repeat).

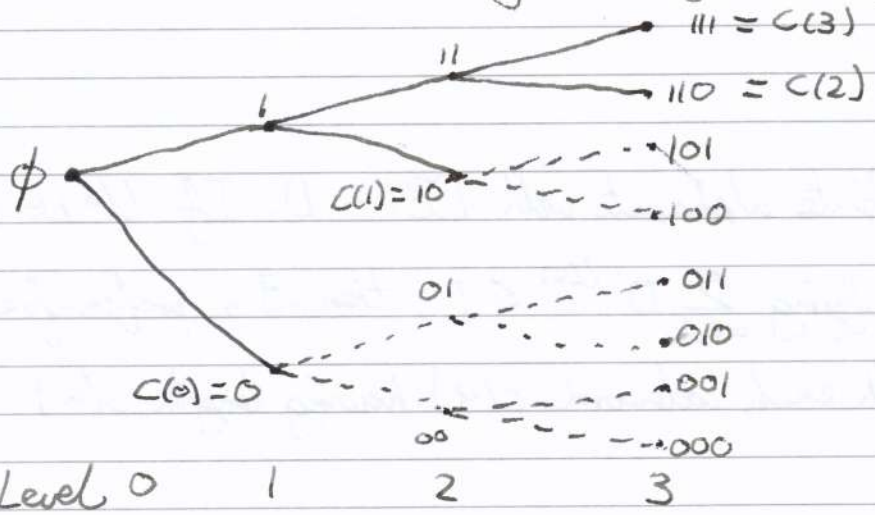
Notice that we can decode a message as it is recieved; we do not require the whole message to decode the first letter.

e.g. $\begin{matrix} 0 & 110 & 111 & 100 \\ \downarrow & \downarrow & \downarrow & \downarrow\downarrow \\ 0 & 2 & 3 & 1\,0 \end{matrix}$ | Prefix free codes are sometimes called <u>instantaneous</u> / <u>self punctuating</u>

- Block codes (all cws are the same length)
- Comma codes (reserves a special letter from B to signal the end of the word).

21/01/13

Coding and Cryptography ②



$C : \{0,1,2,3\} = \{0,1\}^*$

$0 \mapsto 0$
$1 \mapsto 10$
$2 \mapsto 110$
$3 \mapsto 111$

Prefixes of w are all vertices of the tree in the path from the base vertex to w. This is prefix free $\Leftrightarrow$ the path from any codeword to ϕ contains no other codeword

The Kraft Inequality

(1-3) $C : A \to B^*$ be a prefix-free code with codeword $c(a)$ having length $l(a)$. Then $\sum_{a \in A} D^{-l(a)} \leq 1$ where $D = |B|$

Proof

Let $L = \max\{l(a) : a \in A\}$. Consider the tree constructed from C. Vertices at level L correspond to words of length L

$\therefore \exists\ D^L$ vertices.

A codeword $c(a)$, length $l(a)$ is a prefix of $D^{L-l(a)}$ such vertices

Since C is prefix-free, no level L vertex can have two codewords as prefixes. Hence the total number of level L vertices with codewords as prefixes is $\sum_{a \in A} D^{L-l(a)}$ and this can be at most the number of level L vertices $= D^L \cdot \sum_{a \in A} D^{-l(a)} \leq 1$.

Conversely,

Proposition 1.4

Let A, B be finite alphabets with $|B| = D$. If $l(a) \in \mathbb{N}$ for each $a \in A$ satisfying $\sum_{a \in A} D^{-l(a)} \leq 1$, then $\exists$ a prefix-free code $c : A \to B^*$ with each codeword $c(a)$ having length $l(a)$.

Proof

Reorder letters in A : $a_1, a_2, \ldots, a_K$ such that $l_j = l(a_j)$ satisfying $l_1 \leq l_2 \leq \ldots \leq l_K$. We define c inductively.

Choose any word of length l_1 as $c(a_1)$. Suppose that $c(a_j)$ are all chosen for $j < k$ so that no $c(a_i)$ is a prefix of $c(a_j)$ for $i < j < k$. Consider words of length l_k. There are D^{l_k} of them and of these $D^{l_k - l_j}$ have $c(a_j)$ as prefix.

By assumption $1 + \sum_{j=1}^{k-1} D^{l_k - l_j} = \sum_{j=1}^{k} D^{l_k - l_j} \leq D^{l_k}$

So we can choose a word $c(a_k)$ of length l_k which does not have any $c(a_j)$, $j < k$, as prefix. Hence, the result follows by induction.

1.5 Proposition (McMillan's Inequality)

Let $c : A \to B^*$ be a decodable code with codeword $c(a)$ with length $l(a)$. Then $\sum_{a \in A} D^{-l(a)} \leq 1$, where $D = |B|$.

Proof

Let $L = \max \{ l(a) : a \in A \}$. Consider $\left(\sum_{a \in A} D^{-l(a)} \right)^N$ for some $n \in \mathbb{N}$.

Expanding, $\sum D^{-(L(a_1)+\dots+L(a_N))}$, summed over all sequences $a_1, a_2, \dots, a_N$ of length N.

The word $w = c(a_1)\dots c(a_N) \in B^*$ has length

$L(a_1) + \dots + L(a_N) \leq NL$

Since c is decodable, every word $w \in B^*$ of length M can come from at most one sequence $a_1, a_2, \dots, a_N$.

Hence $\left(\sum_{a \in A} D^{-L(a)}\right)^N \leq \sum_{M=1}^{NL} n(M) D^{-M}$

where $n(M)$ = # words of length M of the form $c(a_1)\dots c(a_N)$.

Now, the total number of words of length M is $\leq D^M$, so $n(M) \leq D^M$

$\therefore \left(\sum_{a \in A} D^{-L(a)}\right)^N \leq \sum_{M=1}^{NL} D^M D^{-M} = NL$

Take the N^{th} root.

$\sum_{a \in A} D^{-L(a)} \leq (NL)^{\frac{1}{N}}$ and let $N \to \infty$. □

Combining these results

Corollary 1.6

If $\exists$ a decodable code of n words c_j of length l_j, then there exists a prefix free code consisting of n words c_j' of length l_j.

2 Entropy

We will use $(\mathbb{P}, \Omega)$, (discrete) probability on a sample space Ω.

Definition 2.1

For each $A \in \Omega$ the *information* of A is $I(A) = -\log_2 \mathbb{P}(A)$

So $I(A) \geq 0$, with equality $\Leftrightarrow \mathbb{P}(A) = 1$.

Example

(X_n) a family of independent Bernoulli random variables, each taking values $0, 1$ with probability $\frac{1}{2}$. Given a sequence $x_1, x_2, \ldots, x_N$, the set $A = \{X_1 = x_1, \ldots, X_N = x_N\}$ has $\mathbb{P}(A) = (\frac{1}{2})^N$, thus $I(A) = N$ (= # binary digits, or bits, that are specified)

Definition 2.2

Let X be a discrete random variable on Ω. For each X, the set $\{X = x\}$ has information $I(X=x) = -\log_2 \mathbb{P}(X=x)$

The expected value of this is the *entropy* of X.

$$H(X) := -\sum_{x \in X} \mathbb{P}(X=x) \log \mathbb{P}(X=x)$$

Remarks

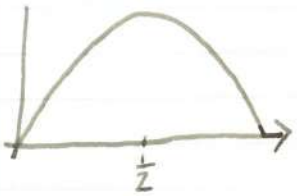
1. $H(X)$ depends only on the probability distribution of X and not on the individual values of X.
2. The definition fails when one of the $\{X = x\}$ has probability 0, so note that $h(p) = -p \log p \to 0$ as $p \downarrow 0$, so we interpret $-\mathbb{P}(X=x) \log \mathbb{P}(X=x)$ as 0 in this case.
3. $H(X) \geq 0$ with equality $\Leftrightarrow X$ is almost surely constant.

Example: $X : \Omega \to \{0, 1\}$, Bernoulli r.v. taking value 1 with

0 with probability $q = 1-p$.

$H(X) = -p\log p - q\log q = H(p, 1-p)$

Max at $p = \frac{1}{2}$



Entropy is the average amount of information gained by knowing a value of X. We will use it to measure the amount of information in a message, and hence decide how much it can be compressed.

<u>Lemma 23</u> (Gibb's Inequality)

X is a discrete random variable taking different values with probabilities $(p_j)_{j=1}^N$, so $\sum_{j=1}^N p_j = 1$. If $(q_j)_{j=1}^N$ is another set of positive numbers whose sum is equal to 1, then

$$-\sum_{j=1}^N p_j \log p_j \leq -\sum_{j=1}^N p_j \log q_j$$

with equality $\Leftrightarrow p_i = q_i \ \forall i$. (for some ordering)

Shannon-entropy — Mixed Entropy

23/01/13

Coding and Cryptography (3)

Joint Entropy

Let X, Y be random variables. (X, Y) is a vector-valued random variable with entropy $H(X, Y)$, the joint entropy of X, Y.

We can prove using P2.3, that

Corollary 2.4

If X, Y are discrete, then $H(X, Y) \leq H(X) + H(Y)$ with equality $\Leftrightarrow$ X, Y are independent.

Example

X takes D values, each with probability $\frac{1}{D}$.

Then $H(X) = \log_2 D$. Suppose that $X_1, \ldots, X_N$ are iid (to X).

Then $H(X_1, \ldots, X_N) = \sum_{i=1}^{N} H(X_i) = N \log_2 D$

Conditional Entropy

Suppose that x is a value taken by X with non-zero probability.

Then the conditional distribution of another random variable Y given that $X = x$ is $\mathbb{P}(Y = y \mid X = x) = \frac{\mathbb{P}(X=x, Y=y)}{\mathbb{P}(X=x)}$. Then we have an rv $(Y \mid X = x)$

$$H(X, Y) = -\sum_{x,y} \mathbb{P}(X=x, Y=y) \log_2 \mathbb{P}(X=x, Y=y)$$

~~$H(X) = -\sum_{x,y} \mathbb{P}(X=x) \log_2 \mathbb{P}(X=x, Y=y)$~~

$$H(X) = -\sum_{x,y} \mathbb{P}(X=x, Y=y) \log_2 \mathbb{P}(X=x)$$

Then $H(X, Y) - H(X) = -\sum_{x,y} \mathbb{P}(X=x, Y=y) \log_2 \frac{\mathbb{P}(X=x, Y=y)}{\mathbb{P}(X=x)}$

$$= -\sum_{x,y} \mathbb{P}(X=x)\, \mathbb{P}(Y=y \mid X=x) \log_2 \mathbb{P}(Y=y \mid X=x)$$

$$= \sum_{x} \mathbb{P}(X=x)\, H(Y \mid X=x), \text{ the conditional entropy}$$

$$H(Y \mid X) = H(X, Y) - H(X)$$

Now, for each x, $H(Y|X=x) \geq 0$ with equality iff $(Y|X=x)$ is constant (almost surely).

$\therefore \quad H(X,Y) - H(X) = H(Y|X) \geq 0$

with equality $\Leftrightarrow$ Y is a function of X (almost surely).

Proposition 2.5

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

Proof

Apply the above results.

3 Efficient Codes

We try to find codes where the expected length of codewords is as short as possible. Such codes are called optimal (best).

Shannon-Fano Codes

Let $c : A \to B^*$ be a code, assumed prefix-free (and thus injective) decodable. Given $a \in A$, the codeword $c(a)$ will have length $|c(a)|$. We seek codes where the expected value of length is as small as possible.

3.1 Definition

The expected length of a codeword is $\sum_{a \in A} p(a)\,|c(a)|$. Let A be a random variable taking values in A, where $P(A=a) = p(a)$. Then the expected length of a codeword is $E(|c(A)|)$.

Coding and Cryptography ③

1·5 shows that the lengths $L(a) = |c(a)|$ satisfy

$$\sum_{a \in A} D^{-L(a)} \leq 1 \qquad (*)$$

Kraft II (1·4) shows that if we have positive integers $L(a)$, satisfying $(*)$ then there is a prefix-free code with codeword length $L(a)$. We solve

$$\min \sum_{a \in A} p(a) L(a) \text{ such that } \sum_{a \in A} D^{-L(a)} \leq 1, \text{ each } L(a) \in \mathbb{N}.$$

3·2 Example

We will try to solve this without insisting that $L(a) \in \mathbb{N}$.

$$\min \sum_{a \in A} p(a) L(a) \quad . \quad \sum_{a \in A} D^{-L(a)} \leq 1.$$

Solution: we claim that the minimum occurs when

$$L(a) = -\log_D p(a) = -\frac{\log_2 p(a)}{\log_2 D}$$

If so then $\sum_{a \in A} D^{-L(a)} = \sum_{a \in A} p(a) = 1$ and expected length is

$$\sum_{a \in A} p(a) L(a) = -\sum_{a \in A} \frac{p(a) \log_2 p(a)}{\log_2 D} = \frac{H(A)}{\log_2 D}$$

We must show that for any values $L(a)$ such that $\sum_{a \in A} D^{-L(a)} \leq 1$ we have $-\frac{1}{\log_2 D} \sum p(a) \log_2 p(a) \leq \sum p(a) L(a)$

Put $S = \sum D^{-L(a)}$ and $q(a) = \frac{D^{-L(a)}}{S}$. Then $\sum q(a) = 1$ by construction, so by Gibbs (2·3)

$$-\sum p(a) \log_2 p(a) \leq -\sum p(a) \log_2 q(a)$$

$$= \sum p(a) \left[L(a) \log_2 D + \log_2 S \right]$$

$$= \left(\sum p(a) L(a) \right) \log_2 D + \log_2 S \leq \left(\sum p(a) L(a) \right) \log_2 D$$

Proposition 3·3

Suppose that A is a random-variable taking values in a finite $\mathcal{A}$.

For any decodable $c : \mathcal{A} \to \mathcal{B}^*$

$\mathbb{E}(|c(A)|) \geq \frac{H(A)}{\log_2 D}$ where $D = |\mathcal{B}|$.

Proof

1·3 shows that $l(a) = |c(a)|$ satisfy $\sum D^{-l(a)} \leq 1$.

Hence (as in 3·2) $\sum p(a) l(a) \geq \frac{H(A)}{\log_2 D}$ □

3·4 Proposition (Shannon-Fano Encoding)

Suppose that A is an rv taking values in a finite alphabet $\mathcal{A}$. Then there is a prefix-free code $c : \mathcal{A} \to \mathcal{B}^*$ satisfying

$\mathbb{E}(|c(A)|) < 1 + \frac{H(A)}{\log_2 D}$ where $D = |\mathcal{B}|$

Proof

1·4 shows that we need to find integers $l(a)$ such that

$\sum p(a) l(a) < 1 + \frac{H(A)}{\log_2 D}$ and $\sum D^{-l(a)} \leq 1$.

If the $l(a) \in \mathbb{N}$, then we can't always take $l(a) = -\log_D p(a)$

so set $l(a) = \lceil -\log_D p(a) \rceil$

Then $-\log_D p(a) \leq l(a)$, so $p(a) \geq D^{-l(a)}$. This implies $\sum D^{-l(a)} \leq 1$, so there is a prefix free code with these word lengths. Moreover $\sum p(a) l(a) < \sum p(a)(1 - \log_D p(a))$

$= 1 - \sum p(a) \log_D p(a) = 1 + \frac{H(A)}{\log_2 D}$

23/01/13 Coding and Cryptography (3)

Given word lengths $l(a)$ we can apply 1.4 to construct the the desired c.

Such codes are called Shannon-Fano codes. They aren't quite optimal but they are easy to construct.

Combining 3.3, 3.4, we get

Theorem 3.6 (Shannon's Noiseless Coding Theorem)

Suppose that A is a random variable taking values in a finite alphabet $\mathcal{A}$. Then an optimal code $c: \mathcal{A} \to \mathcal{B}^*$ for $D = |\mathcal{B}|$ satisfies

$$\frac{H(A)}{\log_2 D} \leq \mathbb{E}(|c(A)|) < 1 + \frac{H(A)}{\log_2 D}$$

Example 3.6

$\mathcal{A} = \{1,2,3,4,5\}$ with probabilities $0.4, 0.2, 0.2, 0.1, 0.1$

$\mathcal{B} = \{0,1\}$

i	p_i	$-\log_2 p_i$	$s_i = \lceil -\log_2 p_i \rceil$	c
1	0.4	1.32	2	00
2	0.2	2.32	3	010
3	0.2	2.32	3	011
4	0.1	3.32	4	1000
5	0.1	3.32	4	1001

$E(|c(A)|) = \sum p_i s_i = 2.8 \geq 2.12 = H(p_1, p_2, p_3, p_4, p_5)$

Huffman Codes

How to construct optimal code? Assume $\mathcal{B} = \{0,1\}$.

Let $\mathcal{A}$ be a finite alphabet with probabilities $p(a)$.

$\mathcal{A} = \{a_1, \ldots, a_K\}$. $p_i = p(a_i)$, such that $p_1 \geq p_2 \geq \ldots \geq p_K$.

$K = 2$ (Base case) The Huffman code is $h: \{a_1, a_2\} \to \{0,1\}$, $h(a_1) = 0$, $h(a_2) = 1$. This is clearly optimal.

Recursively, define a Huffman code on alphabets of size $K-1$.

Choose two letters in $\mathcal{A}$ that have the smallest probabilities attached to them: [illegible], i.e. p_{K-1}, p_K. Form $\tilde{\mathcal{A}} = \{a_1, a_2, \ldots, a_{K-2}, a_{K-1} \cup a_K\}$, by combining a_{K-1} and a_K to give the letter $a_{K-1} \cup a_K \in \tilde{\mathcal{A}}$, that is given the probability $p_{K-1} + p_K$.

Let $\tilde{h}: \tilde{\mathcal{A}} \to \{0,1\}^*$ be a Huffman code for this new alphabet.

Then the Huffman code for $\mathcal{A}$ is obtained by adding a 0 or 1 to the end of the $\tilde{h}$-codewords for a_{K-1}, a_K.

$$h(a_j) = \begin{cases} \tilde{h}(a_j) & j = 1, 2, \ldots, K-2 \\ \tilde{h}(a_{K-1} \cup a_K)\,0 & j = K-1 \\ \tilde{h}(a_{K-1} \cup a_K)\,1 & j = K \end{cases}$$

Definition 3.7

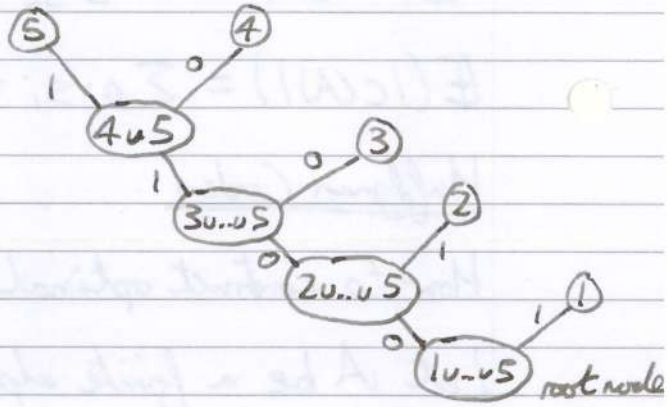
This is the (binary) Huffman code. Clearly this is prefix free.

Theorem 3.8

A Huffman code is indeed optimal (i.e. the average length of codewords is as small as possible for any decodable code $A \to \{0,1\}^*$

Example 3.9

i	p_i	$c(i)$	l_i
1	0.4	1	1
2	0.2	01	2
3	0.2	000	3
4	0.1	0010	4
5	0.1	0011	4



$\mathbb{E}(|h(A)|) = 2.2$

The proof of 3.8 requires a lemma giving properties of optimal codes.

Let $A = \{a_1, \ldots, a_K\}$ with probabilities $p_1 \geq p_2 \geq \ldots \geq p_K$.

The random variable A takes values a_j with probability p_j.

Average codeword length $\mathbb{E}(|c(A)|)$.

Lemma 3.10

There exists an optimal code $c: A \to \{0,1\}^*$ such that

a) $l_j = |c(a_j)|$ are ordered inversely to the probabilities i.e. $l_1 \leq \ldots \leq l_K$

b) The codewords $c(a_{K-1})$ and $c(a_K)$ have the same length and differ only in their last bit (or final digit).

25/01/13 Coding and Cryptography ④

Proofs

We know that ∃ some prefix-free code for A. Let the expected code length = λ. Then, ∃ only finitely many codes c such that $\sum p_j l_j \leq \lambda$. So ∃ a prefix-free code achieving minimum value for $\sum p_j l_j$. This is optimal.

a) If $p_i \geq p_j$ but $l_i > l_j$ then we could reduce $\sum p_j l_j$ by swapping the codewords $c(a_i)$ and $c(a_j)$.
Hence, we must have $l_1 \leq l_2 \leq \ldots \leq l_K$.

b) Let Λ be the maximum codeword length, $\Lambda = l_K$. Write $c(a_K) = wb$ where w is a word of length $\Lambda - 1$ and b is the final bit. Form a new code by changing $c(a_K)$ to w. The length reduces, so reduces $\sum p_j l_j$. Since c was optimal, this new code cannot be prefix-free. Hence there exists $c(a_j)$ with w as a prefix. Thus $c(a_j)$ and $c(a_K)$ both have length Λ and differ only in their last bit. Finally, permute the codewords of length Λ so that $c(a_{K-1})$ and $c(a_K)$ differ only in the last bit.

Proof that Huffman is Optimal: (Huffman, 1952)

Use induction on $k = |A|$. $k = 2$ is simple.

Assume true for all alphabets of size $k-1$. Let $h: A \to \{0,1\}^*$ be a Huffman code, and $c: A \to \{0,1\}^*$ is an optimal code.

Construction of the Huffman code shows that ∃ a Huffman code

$\tilde{h}$ on $\tilde{A} = \{a_1, a_2, \ldots, a_{k-2}, a_{k-1} \cup a_k\}$ of size $k-1$. Letters have probabilities $p_1, p_2, \ldots, p_{k-2}, p_{k-1}+p_k$ respectively. Write $\tilde{A}$ for the random variable taking letters in $\tilde{A}$ with these probabilities.

$$h(a_j) = \begin{cases} \tilde{h}(a_{k-1}) 0 & j = k-1 \\ \tilde{h}(a_{k-1} \cup a_k) 1 & j = k \\ \tilde{h}(a_j) & j \leq k-2 \end{cases}$$

So the expected codeword length for h and $\tilde{h}$ satisfy

$$\mathbb{E}(|h(A)|) = \mathbb{E}(|\tilde{h}(\tilde{A})|) + (p_{k-1} + p_k)$$

For optimal code C, apply 3.10. Choose c with $c(a_{k-1})$, $c(a_k)$ differing only in their last bit, say $c(a_{k-1}) = w0$, $c(a_k) = w1$, for a word $w \in \{0,1\}^*$. Define a new code on $\tilde{A}$ via $\tilde{c}(a_{k-1} \cup a_k) = w$ and $\tilde{c}(a_j) = c(a_j)$ $\forall j \leq k-2$. $\tilde{C}$ is prefix-free and has

$$\mathbb{E}(|c(A)|) = \mathbb{E}(|\tilde{c}(\tilde{A})|) + (p_{k-1} + p_k).$$

Induction says that $\tilde{h}$ is optimal. So $\mathbb{E}(|\tilde{h}(\tilde{A})|) \leq \mathbb{E}(|\tilde{c}(\tilde{A})|)$

Thus $\mathbb{E}(|h(A)|) \leq \mathbb{E}(|c(A)|)$.

But $\mathbb{E}(|c(A)|)$ is minimal average codeword length, so we must have equality. Hence h is optimal.

Chapter 2 : Error Correcting Codes

4 Noisy Channels and Hamming's Code

Up until now, we have assumed that each coded message is transmitted without error through the channel to B, who then decodes it

encoder $A \to \mathcal{A} -$ [Channel] $- \mathcal{B} -$ decoder B

What if there is a small, but appreciable chance of error?

We would like to devise codes than can detect, or better still, correct such errors.

Discrete Memoryless Channels

Definition 4.1

We have a coded message $b_1, b_2, \dots, b_N \in \mathcal{B}^*$ sent through a channel, where each letter of b_i may be altered to another. Assume that changes to one letter are independent of others. The channel is memoryless. Let B_n be a random variable that gives the n^{th} letter of our message, and B_n' the random variable giving the n^{th} letter received, then $\mathbb{P}(b_1', b_2', \dots, b_N' \text{ received} \mid b_1, \dots, b_N \text{ sent})$

$$= \prod_{n=1}^{N} \mathbb{P}(B_n' = b_n' \mid B_n = b_n)$$

We have a transition matrix $(\mathbb{P}(B_n' = i \mid B_n = j))_{ij}$, $i, j \in \mathcal{B}$.

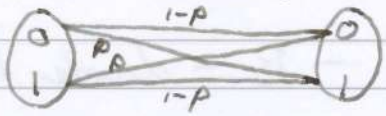
As the probability of error is small, the $(i,i)^{th}$ entry should be 'close' to 1. Assume that the channel is time independent

i.e. the chances of a particular alteration do not depend on n.

Example 4.2

a) Binary Symmetric Channels (BSC)

$A = B = \{0, 1\}$. A BSC is a time independent, memory-less channel with transition matrix $\begin{pmatrix} q & p \\ p & q \end{pmatrix}$



p = probability of error in a single bit (usually small, $< \frac{1}{2}$), $q = 1-p$

If $p = \frac{1}{2}$, no information is transmitted, and we have all noise. This channel is called useless.

If $p = 0$, or 1, there is no noise and no errors to be corrected. We call these channels lossless or perfect

b) Paper-Tape Codes

8-track paper tape is used. $A = \{0, 1\}$, with a 1 represented by a hole punched in the tape. The first 7 bits $x_1, \ldots, x_7$ carry data while x_8 is a check digit, $\sum_{i=1}^{8} x_i \equiv 0 \quad (2) \quad (*)$

We model the chances of error in tape using BSC, with p the probability of error. $\mathbb{P}(\text{recieve } x_1, \ldots, x_7 \text{ without error}) = (1-p)^7$

If there is one error in our 8-bits, $x_1, \ldots, x_8$, then the check $(*)$ fails for the recieved message and the error is detected. If there are two errors then $(*)$ is true and we do not detect the error.

Remark

Check digits are widely used to detect errors, see sheet 2.

Definition 4.3

A binary $[n, m]$-code is a subset $C \subseteq [0,1]^n$ comprising codewords c_i (i.e. a fixed length code of length n). We say that C has size $m = |C|$. If m is large, we can send lots of messages (more information) but as m increases, it is harder to distinguish between different messages, for example when errors occur.

Example

$m = 1$: errors cause no problem, but no information is transmitted.

$m = 2^n$: lots of messages but any error moves $c_i \to c_j$

Definition 4.4

The information rate of C is $\rho(C) = \frac{\log_2 m}{n}$

Note that $m \leq 2^n$, so $\rho(C) \leq 1$; if $m = 1$ then $\rho(C) = 0$, and if $m = 2^n$ then $\rho(C) = 1$.

Assume that all messages are equally likely and that the errors are independent. B's strategy to decode is to guess that the codeword sent is one which differs in the fewest places from the n-string received.

Definition 4.5

Let $x, y \in [0,1]^n$. Write $d(x,y) = \sum_{j=1}^{n} |x_j - y_j| = \sum_{j=1}^{n} 1_{\{x_j \neq y_j\}}$ and call it the Hamming Distance between x, y.

Check that d is a metric on $\{0,1\}^n$. $d(x,y)$ counts the number of coordinates where x, y differ.

When $x \in \{0,1\}^n$ is transmitted through a noisy channel and we recieve an altered word y, then $d(x,y)$ counts the number of letters altered.

Decoding With Errors

Suppose that $b \in \{0,1\}^n$ is sent through a noisy channel and recieved as $v \in \{0,1\}^n$. Errors mean that the word may not be a codeword at all, or may be the wrong codeword. We have 3 possible decoding rules:

i) Ideal Observer

Decode v with $\mathbb{P}(b \text{ sent} \mid v \text{ recieved})$ maximal.

ii) Maximum Likelihood

Decode v with $\mathbb{P}(v \text{ recieved} \mid b \text{ sent})$ maximal

iii) Minimum Distance

Decode v as b with $d(b,v)$ minimal.

Proposition 4.6

If all codewords are equally likely, then i) and ii) give the same result.

Proof

$$\mathbb{P}(v \text{ recieved} \mid b \text{ sent}) = \frac{\mathbb{P}(v \text{ recieved and } b \text{ sent})}{\mathbb{P}(b \text{ sent})}$$

$$= \mathbb{P}(b \text{ sent} \mid v \text{ recieved}) \frac{\mathbb{P}(v \text{ recieved})}{\mathbb{P}(b \text{ sent})}$$

So if all codewords are equally likely, we have the same result.

Proposition 4.7

If letters are transmitted through a BSC with error probability $p < \frac{1}{2}$ then ML and Min-distance rules give the same result.

Proof:

$$\mathbb{P}(v \text{ recieved} \mid b \text{ sent}) = \prod_{j=1}^{n} \mathbb{P}(v_j \text{ recieved} \mid b_j \text{ sent})$$

$$= p^{d}(1-p)^{n-d} = (1-p)^{n}\left(\frac{p}{1-p}\right)^{d}$$

where d = # terms where $v_j \neq b_j$. Hence $d = d(v,b)$.

Since $0 \leq \frac{p}{1-p} < 1$, we see that $\mathbb{P}(v \text{ recieved} \mid b \text{ sent})$ is maximal when $d(v,b)$ is minimal. □

We generally use minimum distance decoding.

This is time consuming if the size of the code is large.

We must specify some convention in the case where $d(b_1,v) = d(b_2,$

(make a random choice, resend the message).

Definition 4.8 (closed Hamming ball)

The ball of radius $r \geq 0$ centred on $b \in \{0,1\}^n$ is

$B(b,r) = \{v \in \{0,1\}^n : d(b,r) \leq r\}$

Its *volume* is the number of points that it contains.

$V(n,r) = |B(b,r)| = \sum_{0 \leq j \leq r} \binom{n}{j}$, clearly independent of the centre b. We will estimate the size of $V(n,r)$ in Chapter 6.

Definition 4.9

Let C be a code as in 4.3. C is

i) *d-error detecting* if changing up to d-digits in each codeword can never produce another codeword.

ii) *e-error correcting* if, knowing that $v \in \{0,1\}^n$ differs from a codeword in at most e-places, we can deduce the codeword.

4.10 Examples

a) *Repetition code of length n* (an $[n,2]$ code)

$C = \{(c\, c\, c \ldots c) \text{ with } c = 0 \text{ or } 1\}$

C is $(n-1)$-error detecting and $\left[\frac{n-1}{2}\right]$-error correcting. The ML-decoder chooses the symbol occurring most often. $\rho(C) = \frac{1}{n}$ (low

b) *Paper-tape code* (4.2b) (an $[n, 2^{n-1}]$ code)

$C = \{(c_1 \ldots c_n) : c_1 + c_2 + \ldots + c_n \equiv 0 \ (2)\}$

(We're identifying $\{0,1\}^n$ with $\mathbb{F}_2^n$). The code is 1-error detecting (if $x \in \mathbb{F}_2^n$ is obtained from $c \in C$ by a single error we have $x_1 + \ldots + x_n = 1$)

It is not error-correcting, since if $\sum_{i=1}^{n} x_i = 1$, then $\exists$ a codewords y with distance $(x, y) = 1$. $\rho(C) = \frac{n-1}{n}$

Example 4.11 (Hamming's Original Code)

This is a 1-error correcting $[7, 16]$ $\overset{[7, 2^4]}{=}$ code with $\rho(C) = \frac{4}{7}$

$C \subseteq \mathbb{F}_2^7$ defined by

$$C = \left\{ c \in \mathbb{F}_2^7 \;\middle|\; \begin{array}{l} c_1 + c_3 + c_5 + c_7 = 0 \ (2) \\ c_2 + c_3 + c_6 + c_7 = 0 \ (2) \\ c_4 + c_5 + c_6 + c_7 = 0 \ (2) \end{array} \right\}$$

We may choose c_3, c_5, c_6, c_7 freely, then c_1, c_2, c_4 are completely determined.

$\therefore |C| = 2^4 = 16$, $\rho(C) = \frac{\log_2 |C|}{n} = \frac{4}{7}$

Suppose that we reciere $\mathbb{F}_2^7$. Form the syndrome $z = (z_1, z_2, z_4) \in \mathbb{F}_2$

$z_1 = x_1 + x_3 + x_5 + x_7 \ (2)$, $z_2 = x_2 + x_3 + x_6 + x_7 \ (2)$

$z_4 = x_4 + x_5 + x_6 + x_7$

If $z = (0, 0, 0)$, then $x \in C$ is correct. If not, and $c \in C$ such that $d(x, c) = 1$, then x_i, c_i differ for $i = z_1 + 2z_2 + 4z_4$ (ordinary addition, z is interpreted as the binary number which gives the number of the bit which is in error). Simply check case by case the 7 binary sequences x containing one 1 and six 0's.

Thus

Proposition 4.12 (Hamming's Binary Code)

Hamming's Binary $[7, 2^4]$ code is 1-error-correcting. $\square$

Remark

If 2 errors occur then the decoder always gives the wrong answer.

30/01/13 Coding and Cryptography ⑥

5. Error-Detecting and Error-Correcting

How good can error-correcting and error-detecting codes be?

Definition 5.1

The (minimum) distance of a code $C \subseteq \{0,1\}^n$ is the minimum value of $d(c_1, c_2)$ for c_1, c_2 distinct codewords.

5.2 Notation

A code C of length n, size M, with distance d is written as an $[n, M, d]$ code

$(C \leq \mathbb{F}_2^n, |C| = M, d = \min\{d(x,y) : x, y \in C, x \neq y\})$

Lemma 5.3

Let C have minimum distance d.

i) C is $(d-1)$-error detecting, but cannot detect all sets of d-errors

ii) C is $\lfloor \frac{d-1}{2} \rfloor$ error-correcting, but cannot correct all sets of $\lfloor \frac{d-1}{2} \rfloor + 1$ errors

Proof

i) $d(c_1, c_2) \geq d$ for all distinct $c_1, c_2 \in C$ $\therefore$ C is $(d-1)$-error-detecting

But $d(c_1, c_2) = d$ for some $c_1, c_2 \in C$ $\therefore$ C cannot detect all sets of d-errors.

ii) Recall (4.8), $B(x, r) = \{y \in \mathbb{F}_2^n, d(x,y) \leq r\}$. C is e-error correcting iff for all distinct $c_1, c_2 \in C$, $B(c_1, e) \cap B(c_2, e) = \emptyset$.

But if $x \in B(c_1, e) \cap B(c_2, e)$ then $d(c_1, c_2) \leq d(c_1, x) + d(c_2, x)$,

$d(c_1, c_2) \leq 2e$. So if $d \geq 2e+1$ then C is e-error-correcting.

Take $e = \lfloor \frac{d-1}{2} \rfloor$. Let $c_1, c_2 \in C$, $d(c_1, c_2) = d$. Let $x \in \mathbb{F}_2^n$,

differ from c_1 in the e-digits where c_1, c_2 also differ. Then $d(x, c_1) = e$ and $d(x, c_2) = d - e$ and if $d \leq 2e$ then $B(c_1, e) \cap B(c_2, e) \neq \emptyset$ i.e. C cannot correct all sets of e-errors. Take $e = \lceil \frac{d}{2} \rceil = \lfloor \frac{d-1}{2} \rfloor + 1$ □

5.4 Examples

1. Repetition code of length n is an $[n, 2, n]$ code
2. Paper tape code of length n is an $[n, 2^{n-1}, 2]$ code (check!)
3. Hamming code (4.11) as a $[7, 16, 3]$-code (1-error-correcting)

Error Correcting Codes

$d \geq 3$, but also $\overbrace{0000000}^{7}$ and $\overbrace{1110000}^{7}$ are both codewords, so $d = 3$.

It is 2-error-detecting.

6. Bounds on Codes

Theorem 6.11 (Hamming's Bound)

Let C be e-error correcting of length n. Then $|C| \leq \frac{2^n}{V(n,e)}$

Proof

C is e-error-correcting $\Rightarrow B(c, e)$, $c \in C$ are pairwise disjoint.

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

$\Rightarrow |C| \, V(n, e) \leq 2^n$ □

Definition 6.2

A code C of length n, size m which is e-error correcting is called *perfect* if $m = 2^n / V(n,e)$

$\Leftrightarrow \forall x \in \mathbb{F}_2^n \; \exists! \, c \in C$ with $d(x,c) \leq e$

$\Leftrightarrow \mathbb{F}_2^n = \bigcup_{c \in C} B(c,e)$ i.e. any $e+1$ errors will make you decode wrongly.

Example

The Hamming Code (4.11) is 1-error-correcting, $n=7$, $m=2^4$,

$V(7,1) = \binom{7}{0} + \binom{7}{1} = 8 = 2^3 \quad \therefore |C| = 2^4 = \frac{2^7}{2^3}$

Lemma 6.3

Hamming's Original $[7,16,3]$ code is perfect.

Remark

If $2^n / V(n,e) \notin \mathbb{Z}$ then no perfect e-error-correcting code of length n can exist. Sheet 3, q.8, 17 deal with the converse, a Golay code.

Now a lower bound:

Definition 6.4

$A(n,d) = \max\{m : \exists [n,m,d] \text{ code}\}$

Examples

$A(n,n) = 2$, $A(n,1) = 2^n \geq A(n,2) = 2^{n-1}$. In fact,

Lemma 6.5

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

Proof:

Let $m = A(n, d+1)$. So $\exists$ a code C with parameters $[n, m, d+1]$.

Let $c_1, c_2 \in C$ with $d(c_1, c_2) = d+1$. Let c_1' be obtained from c_1 by switching one of the digits where c_1, c_2 differ.

$d(c_1', c_2) = d$. If $c \in C \setminus \{c_1\}$ then $d(c, c_1) \leq d(c, c_1') + d(c_1', c_1)$

$\Rightarrow d+1 \leq d(c, c_1') + 1 \Rightarrow d(c, c_1') \geq d$.

Replacing c_1 by c_1' gives $[n, m, d]$-code, so $A(n, d) \geq m$ □

Corollary 6.6

$A(n, d) = \max \{m : \exists\ [n, m, d']$-code for some $d' \geq d\}$

Theorem 6.7

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

Gilbert Sullivan Varshamov (GSV) Bound — Sphere Covering Bound

Hamming Bound — Sphere Packing Bound

Proof:

Let $m = A(n, d)$. $\exists\ [n, m, d]$-code, C say, i.e. no word can be added to C without decreasing the minimum distance d.

Now $\nexists x \in \mathbb{F}_2^n$ such that $d(x, c) \geq d\ \forall c \in C$, otherwise replace C by $C' = C \cup \{x\}$, an $[n, m+1, d]$ code.

So $\mathbb{F}_2^n = \bigcup_{c \in C} B(c, d-1) \Rightarrow 2^n \leq \sum_{c \in C} |B(c, d-1)| = |C| V(n, d-1)$, $|C| = m$

$\Rightarrow m = |C| \geq \frac{2^n}{V(n, d-1)}$ □

Example

$n = 10$, $d = 3$, $V(n, 2) = \binom{10}{0} + \binom{10}{1} + \binom{10}{2} = 56$, $V(n, 1) = 11$

$6.7 \Rightarrow \frac{2^{10}}{56} \leq A(10, 3) \leq \frac{2^{10}}{11}$ i.e. $19 \leq A(10, 3) \leq 93$

The best that we can say is that $72 \leq A(10,3) \leq 79$ but $A(10,3)$ is unknown.

Asymptotics for $V(n,r)$

We study $\frac{1}{n}\log A(n, \lfloor n\delta \rfloor)$ as $n \to \infty$ to see how large the information rate can be for a given error rate.

Proposition 6.8

Let $\delta \in (0, \frac{1}{2})$

i) $\log V(n, \lfloor n\delta \rfloor) \leq n H(\delta)$

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

where $H : [0,1] \to \mathbb{R}$, $p \mapsto \begin{cases} -p\log p - q\log q & p \in (0,1) \quad q = 1-p \\ 0 & p \in \{0,1\} \end{cases}$

The entropy of a Bernoulli rv with parameter p, Chapter 2

Proof

i) $\Rightarrow$ ii) GSV: $A(n, \lfloor n\delta \rfloor) \geq \frac{2^n}{V(n, \lfloor n\delta \rfloor)}$ (by i))

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

i) $H(\delta)$ is increasing for $\delta \leq \frac{1}{2}$. WLOG, assume $n\delta \in \mathbb{Z}$.

$$1 = (\delta + (1-\delta))^n = \sum_{j=0}^{n} \binom{n}{j} \delta^j (1-\delta)^{n-j} \geq \sum_{j=0}^{n\delta} \binom{n}{j} \delta^j (1-\delta)^{n-j}$$

$$= (1-\delta)^n \sum_{j=0}^{n\delta} \binom{n}{j} \left(\frac{\delta}{1-\delta}\right)^j \qquad \frac{\delta}{1-\delta} \leq 1 \text{ as } \delta < \tfrac{1}{2}$$

$$\geq (1-\delta)^n \sum_{j=0}^{n\delta} \binom{n}{j} \left(\frac{\delta}{1-\delta}\right)^{n\delta} = \delta^{n\delta}(1-\delta)^{n(1-\delta)} V(n, n\delta)$$

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

i.e. $0 \geq -n H(\delta) + \log V(n, n\delta)$ □

Actually, the constant $H(\delta)$ in 6.8 i) is the best possible.

Theorem 6.9

For $\delta \in (0, \frac{1}{2})$, $\frac{1}{n} \log V(n, \lfloor n\delta \rfloor) \to H(\delta)$ as $n \to \infty$.

Proof

Let $\delta \in (0, \frac{1}{2})$. Let $r \in [0, \frac{n}{2}]$, and recall that $V(n,r) = \sum_{i=0}^{r} \binom{n}{i}$

$\binom{n}{r} \leq V\binom{n}{r} \leq (r+1)\binom{n}{r}$ (*)

Recall Stirling's Formula: $\ln n! = n \ln n - n + O(\ln n)$ (IA)

and by substituting for $\log_2 n!$ in $\log_2 \binom{n}{r}$

$$\ln \binom{n}{r} = (n \ln n - n) - (r \ln r - r) - \left((n-r)\ln(n-r) - (n-r)\right) + O(\ln n)$$

$$\Rightarrow \log_2 \binom{n}{r} = -r \log_2 \tfrac{r}{n} - (n-r)\log_2 \left(\tfrac{n-r}{n}\right) + O(\log_2 n) = n H(\tfrac{r}{n}) + O(\log_2 n)$$

$$\Rightarrow H(\tfrac{r}{n}) + O(\tfrac{1}{n}\log n) \leq \tfrac{1}{n}\log_2 V(n,r) \leq H(\tfrac{r}{n}) + O(\tfrac{1}{n}\log n)$$

$$\therefore \boxed{\lim_{n\to\infty} \frac{1}{n} V(n, \lfloor n\delta \rfloor) = H(\delta)} \qquad \square$$

Theorem 6.10

For $\delta \in (0, \frac{1}{2})$, define the "asymptotic bound" $\alpha(\delta) = \liminf_{n\to\infty} \frac{1}{n} \log_2 (A(n, \lfloor n\delta \rfloor))$

Then $1 - H(\delta) \leq \alpha(\delta) \leq 1 - H(\frac{\delta}{2})$. Use a combination of the Hamming bound, GSV bound, and 6.9. $\square$

The Hamming Bound and 6.10 suggest that it is not possible to have an information rate $> 1 - H(\delta)$ for an error rate $\delta < \frac{1}{2}$. But the GSV bound and 6.10 suggests that it is always possible to have an information rate $> 1 - H(2\delta)$ for an error rate $\delta < \frac{1}{2}$. So Shannon realised how to improve the information rate while retaining the error rate low...

04/02/13 Coding and Cryptography (8)

3. Shannon's Theorem

7. Capacity

Given a time-independent memory-less channel (4.1, 4.2), a random letter $B \in \mathcal{B} \xrightarrow{\text{channel}} B'$ is sent and recieved. A noisy channel means that maybe $B' \neq B$. The average amount of information given by B is $H(B)$, and recieved by B' is $H(B')$. The conditional entropy $H(B'|B)$ is the info due to noise in the channel.

The remaining info is $H(B') - H(B'|B)$, the part due to the letter being sent.

$$H(Y|X) = H(X,Y) - H(X)$$
$$= \sum_{x \in \mathcal{X}} \mathbb{P}(X=x) H(Y|X=x)$$

Definition 7.1

The mutual information of B, B' is $I(B',B) = H(B') - H(B'|B)$, the amount of uncertainty about B' that is removed by knowing B.

Lemma 7.2

a) $I(B',B) = H(B') + H(B) - H(B',B) = I(B',B)$

b) $I(B',B) \geq 0$ with equality $\Leftrightarrow$ B, B' independent

c) $I(B',B) \leq H(B')$ with equality iff B' is a function of B

d) $I(B',B) \leq H(B)$ with equality iff B is a function of B'.

Proof

a) $H(B'|B) = H(B',B) - H(B)$

$\therefore I(B',B) = H(B') - H(B'|B) = H(B') - H(B',B) + H(B)$

So this is symmetric in B, B'.

b) is (2.4)

c) $H(B'|B) \geq 0$, equal iff B' is a function of B.

d) From c), a) □

We know that $\mathbb{P}(B=b)$, and transition probabilities $\mathbb{P}(B'=b'|B=b)$. We can calculate both

$$\mathbb{P}(B'=b', B=b) = \mathbb{P}(B'=b'|B=b)\,\mathbb{P}(B=b)$$

$$\mathbb{P}(B'=b') = \sum_{b\in B} \mathbb{P}(B'=b', B=b)$$

and hence obtain $H(B)$, $H(B')$, $H(B',B)$, $I(B',B)$. If we have no noise then $B'=B$ and (7.2) implies that $I=H(B)$.

Definition 7.3

The information capacity is $\sup\{I(B',B)\}$ over all probability distributions for B. The distribution $p = (p_1(a_1), \dots, p_K(a_K)) \in [0,1]^K$ (compact) and I is a continuous function of p. Therefore, the sup is attained for some distribution. So Capacity depends only on the transition probabilities $\mathbb{P}(B'|B)$.

Theorem 7.4 (Information Capacity for BSC)

If a BSC has error probability p then its info capacity is

$$C(p) = 1 - h(p) \qquad (= 1 + p\log p + q\log q)$$

Proof

$\mathbb{P}(B=1) = t$, $\mathbb{P}(B=0) = 1-t$ distribution of B

$H(B) = H(1-t, t) = -t\log t - (1-t)\log(1-t)$ entropy of B.

$$H(B'|B) = \mathbb{P}(B=1)H(B'|B=1) + \mathbb{P}(B=0)H(B'|B=0)$$

conditional entropy

$$= tH(1-p, p) + (1-t)H(p, 1-p)$$

$$\begin{cases} \mathbb{P}(B'=1) = t(1-p) + (1-t)p = t+p-2tp \\ \mathbb{P}(B'=0) = tp + (1-t)(1-p) = (1-p) - t + 2tp \end{cases}$$

distribution of B'

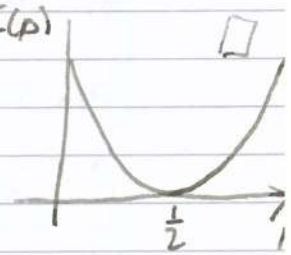
entropy of B': $H(B') = h(t+p-2tp)$

Mutual info: $I(B', B) = H(B') - H(B'|B) = h(t+p-2tp) - h(p)$

We can choose any distribution for B i.e. any $t \in (0, 1)$.

Max $\{h(t+p-2tp)\} = 1$, attained when $t = \frac{1}{2}$, $t+p-2tp = \frac{1}{2}$

So the info capacity $= 1 - h(p)$



$p = 0$ or 1, $C(0) = C(1) = 1$ (perfect transmission)

$p = \frac{1}{2}$, $C(\frac{1}{2}) = 0$ (perfect scrambler)

The Noisy Coding Theorem

Given any code C and any decoding scheme for C, then

$$e(C) = \text{error probability} = \frac{1}{k}\sum_{i=1}^{k} \mathbb{P}(\text{error} \mid c_i \text{ transmitted})$$

where there are k codewords $c_1, \ldots, c_k$ in C (i.e. $e(C)$ is the average probability of error on the assumption that all codewords are equally likely to have been sent).

For binary codes, we assume that ML (= min distance) decoding rule is used and refer to the "error probability" of code without mentioning the decoding rule. We seek codes with small $e(C)$, requiring

Definition 7.5

The maximum error probability $\hat{e}(C) = \max \mathbb{P}(\text{error} \mid c_i \text{ transmitted})$

to be small. $\hat{e}(C) \geq e(C)$.

Recall that the rate of transmission for C is $R = p(C)$ which we want as large as possible. We aim to relate this to the info capacity of the chann

For the BSC, Capacity $= C(p) = 1 + p\log p + q\log q$

Theorem 7.6 (Shannon, 1948)

Given a BSC of capacity C and any R with $0 < R < C$, then, if $\{M_n : 1 \leq n < \infty\}$ is any sequence of integers satisfying $1 \leq M_n \leq 2^{nR}$ $(1 \leq n < \infty)$ and any $\epsilon > 0$, then $\exists$ a sequence of codes $\{C_n : 1 \leq n < \infty\}$ and an integer $N_0(\epsilon)$ with C_n having M_n codewords of length n, and with max error probability $\hat{e}(C_n) \leq \epsilon \quad \forall n \geq N_0(\epsilon)$.

So provided the transmission rate is below the channel capacity, we achieve arbitrarily high reliability.

06/02/13 Coding and Cryptography (9)

Lemma 7.7 (Chebyshev's Inequality)

If X is a real valued random variable, mean μ, finite variance σ^2, then for any $a > 0$, $\mathbb{P}(|X-\mu| \geq a) \leq \frac{\sigma^2}{a^2}$

Lemma 7.8 (Tail Inequality)

Given δ, $0 \leq \delta \leq \frac{1}{2}$, $\sum_{j=0}^{\lfloor n\delta \rfloor} \binom{n}{j} \leq 2^{n h(\delta)}$

where $h(p) = -p\log p - q \log q$, see 6.8 i).

Proofs of 7.6 (Welsh)

We use the method of "random coding". Fix $n \in \mathbb{N}$ and work with binary codes in the space $V_n = \{0,1\}^n$ over $\mathbb{F}_2$. We seek a code with m codewords $b_i \in V_n$. Choose these b_i by just selecting a vector at random from V_n independently for each $1 \leq i \leq m$.

Decode as follows: fix $r > 0$ and consider $B(v, r)$; if we recieve v we decode as the codeword b_j if b_j is the unique codeword in $B(v,r)$, otherwise decode v as some arbitrary codeword, say b_1.

Notation

Send a codeword b → recieve v (i.e. an error has occurred)

Let $E = \{v$ is decoded as a codeword other than $b\}$

This can happen if, either

A) $d(b,v) > r$, or (i.e. too far from any other codeword)

B) $d(b,v) \leq r$ and $d(b',v) \leq r$, for some other codeword b' (i.e. codeword conflict)

$E = A \cup B$, so $\mathbb{P}(E) = \mathbb{P}(A \cup B) \leq \mathbb{P}(A) + \mathbb{P}(B)$

Now B occurs if both

B_1) $\leq r$ errors are made in transmission, and

B_2) one of the codewords other than b is within r of v.

$B = B_1 \cap B_2 \quad \therefore \quad \mathbb{P}(B) \leq \mathbb{P}(B_2)$ ①

For B_2, codewords are chosen randomly $\therefore \mathbb{P}(b_i \text{ within } r \text{ of } v) = \frac{V(n,r)}{2^n}$

Hence the probability that at least one of the other $m-1$ codewords ($\neq b$) is within distance r of v satisfies $\mathbb{P}(B_2) \leq \frac{m-1}{2^n} V(n,r)$ ②

So $\forall \varepsilon > 0$, taking $r = \lfloor n\rho + n\varepsilon \rfloor$, from ① and 7.8

$\mathbb{P}(B) \leq \frac{m}{2^n} 2^{nh(\rho+\varepsilon)}$ ③

For A, let U = # symbols in error when sending codeword b

$\mathbb{P}(A) = \mathbb{P}(U > r)$ and U is Binomial (n, ρ).

Hence, by Chebyshev (7.7),

$\mathbb{P}(A) = \mathbb{P}(U > (\rho+\varepsilon)n) \leq \mathbb{P}(|U - n\rho| > n\varepsilon) \leq \frac{Var(U)}{n^2\varepsilon^2}$

Now $Var(U) = npq$, so

$\mathbb{P}(E) \leq m\, 2^{-n(1-h(\rho+\varepsilon))} + \frac{pq}{n\varepsilon^2}$ for sufficiently large n.

But, from 7.4, $C(\rho+\varepsilon) = 1 - h(\rho+\varepsilon)$ so this reduces to

$\mathbb{P}(E) \leq \frac{pq}{n\varepsilon^2} + m\, 2^{-nC(\rho+\varepsilon)}$

Since $\varepsilon > 0$, this error probability is arbitrarily small for n large, provided that m ($= m(n)$) grows at a rate no faster than $2^{nC(\rho)}$.

This only bounds the average error probability. ∎

Claim

There exist codes C_n with m_n codewords where $m_n \leq 2^{R_n}$, $\hat{e}(C_n) < \varepsilon$

Let $\varepsilon' = \varepsilon/2$ and $m_n' = 2m_n$.

Since $m_n \leq 2^{Rn}$ and $R < C$, $\exists R'$ such that $R < R' < C$ and N_0' such that $\forall n \geq N_0'$, $m_n' \leq 2^{nR'}$, and a sequence of codes C_n' such that C_n' has m_n' codewords and average error probability $< \varepsilon'$ $\forall n \geq N_0'$.

But if $x_1, \ldots, x_{m_n'}$ are codewords of ~~length~~ C_n', this means that

$$\sum_{i=1}^{m_n'} \mathbb{P}(E \mid x_i) \leq \varepsilon' m_n' \qquad (4)$$

Hence at least half of the x_i must satisfy $\mathbb{P}(E \mid x_i) \leq 2\varepsilon' \leq \varepsilon$

Let C_n be any m_n of these codewords satisfying (4). We now have the required code and $\hat{e} \leq \varepsilon$. □

7.9 Remarks and Examples

i) 7.6 tells us that such codes exist, but not how to construct them

ii) 7.6 extends to more general noisy channels (e.g. when noise is generated by a Markov Chain).

iii) The orders of magnitude for the rate at which error probabilities approach 0 are known (Shannon 1957).

iv) Shannon's Theorem has a converse:

For a discrete, memoryless channel of capacity C, and any $R > C$, there cannot exist a sequence of codes C_n with the property that C_n has 2^{Rn} codewords of length n and error probabilities $\hat{e}(C_n) \to 0$ as $n \to \infty$. (Actually, Wolfowitz (1961) showed that the maximum error probabilities $\to 1$ as $n \to \infty$).

8 The Wizard of Odds

Every day at noon, you make a bet with me for any amount k that you choose. Whatever happens, I keep the stake.

I toss a coin, and if Heads, I pay you ku.

If Tails, I pay you nothing.

$\mathbb{P}(H) = p$. What is the best strategy?

Expected Winnings : $-k + pku$

If $pu < 1$, $\mathbb{E}(\text{winings}) < 0$, don't bet.

If $pu > 1$, you bet. But how much? If you bet your entire fortune (bankroll) and win, then you are better off than if you had bet a smaller sum; but if you lose, you are bankrupt.

08/02/13

Coding and Cryptography ⑩

What proportion, w, to bet each time? Choice of w is always the same. Suppose that your fortune after n plays is X_n, and initial bankroll is $X_0 = 1$.

Consider $Z_n = \frac{X_{n+1}}{X_n}$, independent identically distributed random variables, say to Z, where $Z_{n+1} = \begin{cases} uw + (1-w) & (n+1)^{th} \text{ toss a He} \\ 1-w & \text{" " " Tai} \end{cases}$

Now $X_n = \prod_{i=0}^{n-1} Z_i$, so $\log X_n = \sum_{i=0}^{n-1} \log Z_i$

$\Rightarrow \frac{\log X_n}{n} = \frac{\sum \log Z_i}{n} \to \mathbb{E}(\log(Z))$, meaning, by the Weak Law of Large Numbers:

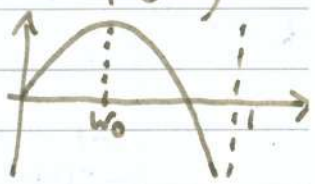
Lemma 8.1
Suppose that $Z_1, Z_2, \ldots$ are iid taking values in $[a,b]$, $0 < a < b$. If $X_n = \prod_{i=1}^{n} Z_i$ then

$$\mathbb{P}\left(\left| \frac{\log X_n}{n} - \mathbb{E}(\log(Z)) \right| > \varepsilon \right) \to 0 \text{ as } n \to \infty.$$

We want to choose w to maximise
$\mathbb{E}(\log Z_n)$ = expected value of the random variable taking value $\log(1-w)$ if we lose, and $\log((1-w)+uw)$ if you win.

$$= (1-p)\log(1-w) + p\log((1-w)+uw) =: f(w)$$

One method: $f'(w) = \frac{(pu-1)+(u-1)w}{((u-1)w+1)(1-w)}$



So if $u \leq 1$, obviously we don't bet. So assume $u > 1$.
If $pu \leq 1$, then $f(w)$ is decreasing for +ve w, so don't bet.
If $pu > 1$, f takes a maximum, so we bet when
$w = w_0 = \frac{up-1}{u-1}$

$$\text{Then } E(\log Z_n) = (1-p)\log(1-w_0) + p\log(uw_0 + (1-w_0))$$
$$= p\log up + q\log uq - q\log(u-1)$$
$$= \log u + p\log p + q\log q - q\log(u-1)$$

Exercise: Show that if you bet < optimum, your bankroll increases, but more slowly than optimum, but if you bet more than some proportion w_1, your bankroll decreases.

We have seen $-(p\log p + q\log q)$ before as the Shannon entropy for BSC. Kelly (1956) gave interpretations in gambling, and the stock market by which an agent recieves info over a noisy channel, about, for example, which horse will win.

In Shannon's World, information can be transmitted through a noisy channel at a rate close to the capacity with negligible error, provided that the messages are long enough.

In Kelly's World, the agent can, with arbitarily high probability, increase her bankroll at a certain optimum rate provided she can continue long enough.

Definition 8.2
The idea that agents making a long sequence of bets should aim to maximise the expectation of the log is called the Kelly Criterion. (Ed Thorp: Beat the Market, Beat the Dealer)

HAL
↓↓↓
IBM

Chapter 4: Linear and Cyclic Codes

9 Linear Codes

* Polynomials and Fields

What follows are a few results from Numbers + Sets, Linear Algebra, GRM

28/02/13 Coding and Cryptography (10)

- A ring R is a set with operations $+, \times$ satisfying the usual axioms
- A field k is a ring such that every non-zero element has a multiplicative inverse, e.g. $\mathbb{Q}$, $\mathbb{R}$, $\mathbb{C}$, $\mathbb{F}_p = \mathbb{Z}/p\mathbb{Z}$, p prime

- Every field is either an extension of $\mathbb{F}_p$, or of $\mathbb{Q}$.

- k has q elements $\Leftrightarrow q = p^r$, p prime. These are unique up to isomorphism and will be denoted $\mathbb{F}_q$.

Examples

- p prime, $\mathbb{Z}/p\mathbb{Z}$ a finite field, $\mathbb{F}_p$

- $n \geq 2$, note that $\mathbb{F}_{p^n} \neq \mathbb{Z}/p^n\mathbb{Z}$ (e.g. $\alpha \in \mathbb{F}_{2^2}$, $\alpha \neq 0, 1$, $\alpha + 1 \neq 0, 1$, $\alpha^2 = \alpha + 1$, $\mathbb{F}_{2^2} = \{0, 1, \alpha, \alpha^2\}$)

- The non-zero elements of $\mathbb{F}_q$ form an abelian group under $\times$, $\mathbb{F}_q^*$, which is actually cyclic, of order $q-1$. Any generator of $\mathbb{F}_q^*$ is a primitive element of $\mathbb{F}_q$.

If α is a primitive element, then $\mathbb{F}_q^* = \{\alpha^j \mid 1 \leq j \leq q-1\}$.
Note that $o(\alpha^j) = \frac{q-1}{(q-1, j)}$

primitive elements $= \varphi(q-1) = |\{j : j = 1, \ldots, q-1, (q-1, j) = 1\}|$

A primitive r^{th} root of unity is an element of k of order r in k^*.

We want to work with alphabets that are finite dimensional vector spaces over $\mathbb{F}_q$ (e.g. $\mathbb{F}_q^d$ is a d-dimensional vector space over $\mathbb{F}_q$). Any d-dimensional vector space has q^d vectors. We have a scalar product in $\mathbb{F}_q^d$.

$x \cdot y = \sum_{i=1}^{d} x_i y_i$ (*). Every linear map $\alpha : \mathbb{F}_q^d \to \mathbb{F}_q$ has the form $x \mapsto x \cdot y$ (for some $y \in \mathbb{F}_q^d$).

So (*) allows an identification of $(\mathbb{F}_q^d)^*$ with $\mathbb{F}_q^d$. If $|S| < \infty$, $V = \mathbb{F}_q^S = \{f : S \to \mathbb{F}_q\}$ is $|S|$-dimensional as a vector space over $\mathbb{F}_q$.

A polynomial over $\mathbb{F}_q$ is a formal expression

$$P(X) = \sum_{j=0}^{d} a_j X^j \quad \text{with } a_j \in \mathbb{F}_q \text{ and } X \text{ an indeterminate}$$

Its degree $\partial(P)$ is the largest m with $a_m \neq 0$. By convention, $\partial(0) = -\infty$.

Manipulate polynomials in X using the standard rules but beware of anomalies.

Example

Over $\mathbb{F}_2$, $X^2 + X$ is a non-zero polynomial but $t^2 + t = 0$ $\forall t \in \mathbb{F}_2$

Finite Fields

The multiplicative group of a finite field is cyclic. Fa field.

Proof

Lagrange: For a group H, $x \in H$, $|H| = n$, we have $o(x) \mid n$.

Therefore, for $H \subset F$ finite, $x \in H$, $x^n = 1$.

Suppose that there are $H[d]$ elements of order d.

For such elements, $x^d = 1$, and they are solutions to $X^d - 1$. These generate the subgroup of solutions to $X^d - 1$ in the field, so there can be at most $\varphi(d)$ of them.

Thus

$$n = \sum_{d \mid n} H[d] \leq \sum_{d \mid n} \varphi(d) = n$$

So $H[d] = \varphi(d) \quad \forall d, d \mid n$.

Then, $H[n] = \varphi(n) > 0$.

So there is an element of order n, and this must generate H.

11/02/13 Coding and Cryptography (11)

$(\mathbb{F}_q[x], +, \cdot)$ is a ring, and the 'degree' is a Euclidean function for this ring $\Rightarrow$ Euclidean domain. So for $P, D \in \mathbb{F}_q[x]$, $D \neq 0$, $\exists!\ Q, R \in \mathbb{F}_q[x]$ with $P = QD + R$, $\partial(R) < \partial(D)$

Remainder Theorem

i) Given $P \in \mathbb{F}_q[x]$, $a \in \mathbb{F}_q$, $\exists\ Q \in \mathbb{F}_q[x]$, $r \in \mathbb{F}_q$ such that $P(x) = (x-a)Q(x) + r$

ii) If $P(x) \in \mathbb{F}_q[x]$ and $a \in \mathbb{F}_q$ is such that $P(a) = 0$, then $\exists Q$ such that $P(x) = (x-a)Q(x)$ □

- A polynomial P is reducible if $\exists$ non-constant Q, $\partial(Q) < \partial(P)$ which divides P. A non-constant P that is not reducible is irreducible.
- $f(x) \in \mathbb{F}_q[x]$ irreducible $\Rightarrow \frac{\mathbb{F}_q[x]}{(f(x))}$ is a field. If $d = \partial(f)$, this is a field of order q^d. (e.g. $f(x) = x^2+x+1 \in \mathbb{F}_2[x]$, $\frac{\mathbb{F}_2[x]}{(x^2+x+1)} = \mathbb{F}_{2^2}$)
- $\mathbb{F}_q[x]$ is a PID: if $I \lhd \mathbb{F}_q[x]$ then either $I = 0$, or $\exists\ 0 \neq D \in I$ of smallest degree. Given any $P \in I$, $P = QD + R$, $\partial(R) < \partial(D)$. But $R \in I \Rightarrow R = 0$, so $P = QD$. Hence $I = D\mathbb{F}_q[x]$ for some D, $I = (D)$.
- Given $I = (D)$, the quotient $\frac{\mathbb{F}_q[x]}{I}$ is a ring, and $\exists$ a canonical ring homomorphism $\mathbb{F}_q[x] \to \frac{\mathbb{F}_q[x]}{I} = R$.
- We will take $D = x^n - 1$ and consider R. Divide any $P \in \mathbb{F}_q[$ by $x^n - 1$ to obtain a remainder of degree $\lneq n$.

So each coset $P + (X^n - 1)\mathbb{F}_q[X]$ contains a unique polynomial of degree $n-1$. Hence, we can represent R by

$\{a_0 + a_1 X + \dots + a_{n-1} X^{n-1} : a_0, a_1, \dots, a_{n-1} \in \mathbb{F}_q\}$

Multiplication in R corresponds to multiplying polynomials, and then reducing mod $(X^n - 1)$. In this case, R is a vector space of dimension $n = \partial(D)$, i.e. $R = \mathbb{F}_q^n$

Linear Codes

Definition 9.1

A linear code $C \leq \mathbb{F}_2^n$ is such that

i) $0 \in C$ ii) $x, y \in C \Rightarrow x+y \in C$

i.e. $\Leftrightarrow C$ is an $\mathbb{F}_2$-subspace of $\mathbb{F}_2^n$.

The rank of C is $k = \dim_{\mathbb{F}_2} C$. A linear code of length n, rank k, is an (n,k)-code, and if the minimum distance is d it is an (n,k,d)-code.

If $e_1, \dots, e_k$ is a basis of $C/\mathbb{F}_2$ then $C = \{\sum_{i=1}^{n} \lambda_i e_i \mid \lambda_i \in \mathbb{F}_2\}$ so $|C| = 2^k$. Thus an (n,k) code is an $[n, 2^k]$-code.

The information rate $= \frac{k}{n} = \rho(C)$

Examples of Linear Codes

i) Repetition Code $C = \{\underline{x} : \underline{x} = (x, x, \dots, x), x \in \mathbb{F}_2\}$

ii) Paper Tape Code $C = \{\underline{x} : \sum_{i=0}^{n} x_i = 0\}$

iii) Hamming's Original Code.

Lemma 9.2

The weight of $x \in \mathbb{F}_2^n$ is $wt(x) = d(x, 0) = |\{i : x_i \neq 0\}|$

The minimum distance of a linear code

= The minimum weight of the non-zero codewords

Proof:

If $x, y \in C$, then $d(x, y) = d(x+y, 0) = wt(x+y)$

Since $x+y \in C$, $\min\{d(x,y) : x, y \in C, x \neq y\}$

$= \min\{wt(c) : c \in C, c \neq 0\}$ □

Codes C_1, C_2 are equivalent if reordering each codeword of C_1 using the same permutation gives the codewords of C_2.

Definition 9.3

Let C be an (n, k) code. A generator matrix for C is a $k \times n$ matrix G whose rows form a basis for C. If G is a generator matrix for C and G' is any other matrix obtained from G by any finite sequence of elementary row or column operations of the form:

i) Permuting rows or permuting columns

ii) Multiplying a row or column by a non-zero scalar (i.e. 1)

iii) Adding to a row a multiple of another row.

} (*)

then the following is routine:

Lemma 9.4

G' is the generator matrix of C' equivalent to C.

Lemma 9.5

Let G be any $k \times n$ matrix whose rows are linearly independent. Then, by applying a sequence of operations (*) to G, it is possible to transform G to a matrix of type $[I_k \mid B]$, I_k the $k \times k$ identity

Proof:

Gaussian Elimination transforms G into row-echelon form $G_{ij} = \begin{cases} 0 & j < l(i) \\ 1 & j = l(i) \end{cases}$ for some $l(1) < \ldots < l(k)$. Permuting columns replaces the code with an equivalent code. So WLOG, $l(i) = i$ $(1 \leq i \leq k)$. Thus $G = \begin{pmatrix} 1 & & \\ & \ddots & \\ 0 & & 1 \end{pmatrix} \mid (\times\times)$. Further row operations give $G = [I_k \mid B]$.

Theorem 9.6

If C is a linear (n,k)-code then $\exists$ an equivalent code C' with generator matrix $[I_k \mid B]$. Combine 9.4, 9.5 $\square$

Remark

Theorem 9.6 says that any codeword can be written as $(y, z) = (y \mid yB)$ where $y = (y_1, \ldots, y_k)$ can be considered the message and $z = yB$ of length $n-k$ may be considered as check digits. Any code whose codewords can be split in this way is called systematic.

Definition 9.7

Let $P \leq \mathbb{F}_2^n$. Then the parity check code defined by P is

$C = \{x \in \mathbb{F}_2 : p \cdot x = 0 \ \forall p \in P\}$

Examples

- $P = \{(1,1,\ldots,1)\}$ gives the simple parity check code.
- $P = \{1010101, 0110011, 0001111\}$ gives Hamming's $[7,16,3]$-code

Lemma 9.8

Every parity-check code is linear.

Proof:

$0 \in C$ since $p \cdot 0 = 0 \quad \forall p \in P$

$x, y \in C \Rightarrow p \cdot (x+y) = p \cdot x + p \cdot y = 0 \quad \forall p \in P$ □

Definition 9.9

Let $C \leq \mathbb{F}_2^n$, linear. Define the dual code

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

This is a parity check code, hence linear. Note that $C \cap C^\perp \neq \{0\}$ is possible.

Theorem 9.10

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

Proof

Note that $C^\perp$ is the annihilator of C, and use linear algebra.

OR, WLOG, C has generator matrix $G = [I_k, B]$. G has k linearly independent columns, so the linear map $\alpha : \mathbb{F}_2^n \to \mathbb{F}_2^k$, $x \mapsto Gx$ is injective with $\ker \alpha = C^\perp$, hence $\dim \mathbb{F}_2^n = \dim \ker \alpha + \dim \operatorname{im} \alpha$ by rank-nullity, so $n = \operatorname{rank} C^\perp + \operatorname{rank} C$ □

Lemma 9.11

Let C be linear. Then $(C^\perp)^\perp = C$. In particular, C is a parity check code.

Proof

Let $x \in C$. Then $x \cdot y = 0 \ \forall y \in C^\perp$. So $x \in (C^\perp)^\perp$, hence $C \leq (C^\perp)^\perp$. By 9.10, $\operatorname{rank} C = n - \operatorname{rank} C^\perp$

$= n - (n - \operatorname{rank}(C^\perp)^\perp) = \operatorname{rank}(C^\perp)^\perp$. So $C = C^\perp$ □

Definition 9.12

A parity check matrix H for C is a generator matrix for $C^\perp$. It is an $(n-k)\times n$ matrix. Note that codewords in C can now be thought of as dependence relations between columns of H i.e.

$$C = \{x \in \mathbb{F}_2^n : Hx = 0\}$$

9.13 Syndrome Decoding

The syndrome of $x \in \mathbb{F}_2^n$ is Hx. Suppose that you recieve $x = c + z$ where c is the codeword, and z is the error. Then $Hx = Hc + Hz = Hz$. If C is known to be e-error correcting, we precompute Hz for all z with $wt(z) \leq e$. On recieving $x \in \mathbb{F}_2^n$, we look for Hx in our list. $Hx = Hz$, i.e. $H(x-z) = 0$, so $c = x - z \in C$, with $d(x,c) = wt(z) \leq e$ (c.f. 4.11 with $e = 1$).

Example

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

Codewords: 0000, 1010, 0111, 1101

Syndrome Lookup Table

Syndrome	Corrector = coset leader
0 0	0 0 0 0
1 0	0 0 1 0
0 1	0 0 0 1
1 1	0 1 0 0

Recieve 1111, Syndrome 10. The Leader = 0010, decode as 1101
But, the leader of 10 might have been 1000, then we would decode as 0111.

Finally, we try to exploit the fact that if C is linear and $a \in C$, then $a + C \in C$. Recall that $wt(x) = d(x,0)$ $(x \in \mathbb{F}_2^n)$.

Weight Enumeration for Linear Codes

Lemma 9.14

i) $wt(x) \geq 0$, and $wt(x) = 0 \Leftrightarrow x = 0$

ii) $wt(\lambda x) = wt(x)$ $(\lambda \in \mathbb{F}^*)$

iii) $wt(x+y) \leq wt(x) + wt(y)$

This is immediate, since d defines a metric on $\mathbb{F}_2^n$. □

Recall 9.2: the non-zero, min-distance of a linear code $\equiv$ the minimum non-zero weight.

Definition 9.15

The weight enumerator polynomial of a linear code C is the homogeneous, 2-variable polynomial $W_C(s,t) = \sum_{i=0}^{n} A_i s^i t^{n-i}$ where A_i = # elements of C of weight i.

Lemma 9.16

i) $W_C(s,t)$ is homogeneous of degree n.

ii) If rank $C = k$, then $W_C(1,1) = 2^k$.

iii) $W_C(0,1) = 1$

iv) $W_C(1,0) = 0$ or 1

v) $W_C(s,t) = W_C(t,s)$ $\forall s,t \Leftrightarrow W_C(1,0) = 1$

Proof

i) Clear from the definition.

ii) $W_C(1,1) = \sum A_i$ = # codewords in C $= 2^k$

iii) $W_C(0,1) = A_0 = 1$ ($\underline{0}$ is the unique codeword of weight 0)

iv) $W_C(1,0) = A_n = \begin{cases} 0 & (1,1,\dots,1) \in C \\ 1 & \dots\dots \notin C \end{cases}$.

v) $W_C(s,t) = W_C(t,s) \overset{(\Rightarrow)}{\Rightarrow} 1 = W(0,1) = W(1,0)$

$(\Leftarrow)$ $W_C(1,0) = 1 \Rightarrow (1,1,\dots,1) \in C$

$\Rightarrow (x + (1,1,\dots,1) \in C \Leftrightarrow x \in C)$

$\Rightarrow ((1-x_1,\dots,1-x_n) \in C \Leftrightarrow (x_1,\dots,x_n) \in C)$

$\Rightarrow A_j = A_{n-j} \Rightarrow W(s,t) = W(t,s)$ □

9.17 Examples

i) BSC, probability p of error. Let C be a linear code of length n.

$W_C(p,1-p) = \mathbb{P}$(recieve cw | cw transmitted)

$\mathbb{P}$(recieve incorrect cw | cw transmitted) $= W_C(p,1-p) - (1-p)^n$

ii) C the repetition code of length n, $W_C(s,t) = s^n + t^n$

iii) C the paper tape code, length n. $W_C(s,t) = \frac{1}{2}\left((s+t)^n + (t-s)^n\right)$

To see i), $W_C(p,1-p) = \sum A_j p^j (1-p)^{n-j}$

$= \sum$ (# cws length j) $\times \mathbb{P}$(we make j mistakes transmitting $(0,\dots,0)$)

$= \mathbb{P}$(codeword recieved | $(0,\dots,0)$ sent)

$= \mathbb{P}$(codeword recieved | codeword sent)

i.e. $\mathbb{P}$(recieve incorrect cw | cw sent) $= \mathbb{P}$(recieve cw | cw sent) $-\mathbb{P}$(no mistakes)

$= W(p,1-p) - (1-p)^n$

ii), iii) are special cases of the following Theorem.

Theorem 9.18 (MacWilliams Identity)

If $C \leq \mathbb{F}_2^n$ a linear code, and $C^\perp$ is its dual, then

$$W_{C^\perp}(s,t) = 2^{-\dim C} W_C(t-s, s+t)$$

Proof

Sheet 3, question 9 □

10 Constructing Codes from other codes

Hamming Codes

Definition 10.1

For $d \geq 1$, let $n = 2^d - 1$. Let $D = \mathbb{F}_2^d$, the (column) vector space of dimension d over $\mathbb{F}_2$. Let H be the $d \times n$ matrix whose columns are the n distinct, non-zero vectors of D.

The Hamming $(n, n-d)$-code is the linear code C of length n defined by the parity check matrix H.

(This is only defined up to equivalence).

Example

Hamming's [7, 16] example is one, with $d = 3$.

Lemma 10.2

The Hamming $(n, n-d)$ code is a linear code C of length n, rank $n-d$, having minimum weight 3, and is a perfect 1-error-correcting code ($n = 2^d - 1$).

Example

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

Proof of 10·2

By 9·12, the rows of C are dependence relations between columns of H. Any two columns of H are linearly independent, so there are no non-zero codewords of weight ≤ 2. Hence $d(C) \geq 3$.

If $H = \begin{pmatrix} 1&0&\cdots \\ 0&1&\cdots \\ \cdots&\cdots&\cdots \end{pmatrix}$ then $1110\ldots0 \in C$, hence $d(C) = 3$.

By 5·3, C is 1-error-correcting.

$$2^n / V(n,1) = 2^n / (n+1) = 2^{n-d} = |C| \qquad \square$$

Definition 10·3

If C is an $[n, m, d]$ code

a) the parity check extension C^+ of C is the code of length $n+1$ given by $C^+ = \{x \in \mathbb{F}_2^{n+1} : (x_1, \ldots, x_n) \in C, \sum_{j=1}^{n+1} x_j \equiv 0 \ (2)\}$

b) the truncation C^- of C is the code of length $n-1$ given by $C^- = \{(x_1, \ldots, x_{n-1}) : (x_1, \ldots, x_n) \in C \text{ for some } x_n \in \mathbb{F}_2\}$
$[n-1, m, d']$ code, $d-1 \leq d' \leq d$

c) the shortening (puncturing) C' of C by the symbol $\alpha \in \{0,1\}$ is the code of length $n-1$ given by $C' = \{(x_1, \ldots, x_{n-1}) : (x_1, \ldots, x_{n-1}, \alpha) \in C\}$

This is an $[n-1, m', d']$ code, $d' \geq d$, $m' \geq \frac{m}{2}$ for suitable α.

Lemma 10·4

If C is linear, so are C^+, C^-, C' (for $\alpha = 0$). $\square$

d) The repetition $C^{(m)}$ is the code of length mn obtained by repeating each word of C m times.

Combining Two Linear Codes C_1, C_2

The direct sum $C_1 \oplus C_2 = \{(x, y),\ x \in C_1,\ y \in C_2\}$ is of little interest because

Lemma 10.5

$$d(C_1 \oplus C_2) = \min\{d(C_1), d(C_2)\}$$

Proof

$(x, 0) \in C_1 \oplus C_2$, $wt(x, 0) = wt(x)$

$(0, y) \in C_1 \oplus C_2$, $wt(0, y) = wt(y)$

$\Rightarrow \min wt(C_1 \oplus C_2) \leq \min\{\min wt(C_1), \min wt(C_2)\}$

However, if $(x\ y) \in C_1 \oplus C_2$, $(x\ y) \neq 0$, then

$x \neq 0$, $wt(x\ y) \geq wt(x)$, and/or $y \neq 0$, $wt(x, y) \geq wt($

$\min wt(C_1 \oplus C_2) \geq \min\{\min wt(C_1), \min wt(C_2)\}$

Hence we have equality.

Finally, $(0, 0) \in C_1 \oplus C_2$. $(x\ y) + (x'\ y') = (x+x'\ y+y')$ $\in C_1 \oplus C_2$

$\therefore C_1 \oplus C_2$ linear $\therefore$ min weight = min distance (9.2)

$\therefore \min dist(C_1 \oplus C_2) = \min\{d(C_1), d(C_2)\}$

Note $rk(C_1 \oplus C_2) = rk(C_1) + rk(C_2)$ and length $(C_1 \oplus C_2)$ $=$ length (C_1) + length (C_2)

Definition 10.6

C_1, C_2 linear, length n, $C_2 \subseteq C_1$. The <u>bar product</u> $C_1 | C_2$ is the code of length $2n$ with words $(x \mid x+y)$, $x \in C_1$, $y \in C$

Lemma 10.7

$C_1 | C_2$ is linear, rank = rank (C_1) + rank (C_2)

$d(C_1 | C_2) = \min\{2d(C_1), d(C_2)\}$

Proofs

Let C_1 have a basis $x_1, \dots, x_k$, C_2 a basis $y_1, \dots, y_l$.

C_1/C_2 has basis $\{(x_i|x_i) \mid 1 \le i \le k\} \cup \{(0, y_j) \mid 1 \le j \le l\}$

Let $z = (x|x+y) \in C_1/C_2$. Then $wt(z) = wt(x) + wt(x+y)$

Suppose that $z \neq 0$, so that $x, x+y$ may not both be 0.

If $x = 0$, then $y \neq 0$, $wt(z) = wt(y) \geq d(C_2)$

If $x \neq 0$, $y = -x$, $wt(z) = wt(x) \geq d(C_2)$,

since in this case $x = -y \in C_2$.

If $x \neq 0$, $y \neq -x$, then $x+y \in C_1$, so

$wt(z) = wt(x) + wt(x+y) \geq 2d(C_1)$

Now, $\exists x \in C_1$ with $wt(x) = d(C_1)$. Then, $d(C_1/C_2) \le wt(x|x) = 2d(C_1)$

$\exists y \in C_2$ with $wt(y) = d(C_2)$. Then $d(C_1/C_2) \le wt(0|y) = d(C$

$\therefore \; d(C_1/C_2) \le \min\{2d(C_1), d(C_2)\}$

Decoding rules

Knowing decoders for C_1, C_2,

send $(x|x+y) \to (u|v)$ recieved, with e-errors, $e \le \frac{1}{2}d(C_1/C$

See Goldie + Pinch, page 157.

Reed-Muller Codes

These are linear codes suitable when the error-rate is high and we are less interested in the information rate. These were famously used by NASA's early planetary probes.

Preliminaries

Let X be a set, $X = \{P_0, \dots, P_{n-1}\}$, $|X| = n$. We have a correspondan

$\mathcal{P}(X) \leftrightarrow \mathbb{F}_2^n$, namely $\mathcal{P}(X) \leftrightarrow \{f: X \to \mathbb{F}_2\} \longleftrightarrow \mathbb{F}_2^n$

$A \mapsto 1_A$, then $f \mapsto (f(P_0), \ldots, f(P_{n-1}))$

Symmetric difference $A \triangle B \leftrightarrow x + y$ vector addition

Intersection $A \cap B \leftrightarrow$ "wedge product" $x \wedge y = (x_0 y_0, \ldots, x_{n-1} y_{n-1})$ pointwise product

$\emptyset \leftrightarrow (0, 0, \ldots, 0)$, $v_0 = (1, \ldots, 1) (\leftrightarrow 1_X)$

Take $X = \mathbb{F}_2^d$ and list the points of this d-dimensional space with some $P_0, \ldots, P_{n-1}$. So $|X| = 2^d$.

$v_0 = (1, \ldots, 1)$ $(\leftrightarrow 1_X)$

coordinate hyperplane

$v_i = 1_{H_i}$ for $1 \le i \le d$ where $H_i = \{P \in X : i^{th}$ coordinate of P is $0\}$

Definition 10.8

The Reed-Muller code of order r, $(0 \le r \le d)$, length $n = 2^d$, denoted $RM(d, r)$, is the linear code spanned by v_0, and all wedge products of r or fewer of the v_i (convention: the empty wedge is v_0).

Example 10.9

$d = 3$, $X = \mathbb{F}_2^3$, $P_0 = 000$, $P_1 = 001$, ..., $P_7 = 111$

H_1 = elements of X with 0 in the first coordinate

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
$v_1 \wedge v_3$	1	0	1	0	0	0	0	0
$v_2 \wedge v_3$	1	0	0	0	1	0	0	0
$v_1 \wedge v_2 \wedge v_3$	1	0	0	0	0	0	0	0

These are the generators of $RM(3, r)$

$RM(3,0)$ = repetition code of length 8 = $\langle v_0 \rangle$

$RM(3,1)$ = equivalent to a parity check extension of Hamming's original code

$RM(3,2)$ = paper tape code of length 8 - simple parity check

$RM(3,3)$ = trivial code = all elements of $\mathbb{F}_2^3$

Proposition 10.10

i) The vectors $v_{i_1} \wedge \ldots \wedge v_{i_s}$ for $1 \le i_1 < \ldots < i_s \le d$ and $0 \le s \le d$ are a basis of $\mathbb{F}_2^X$

ii) Rank $RM(d, r) = \sum_{s=0}^{r} \binom{d}{s}$

Proof

i) We have listed in total $\sum_{s=0}^{d} \binom{d}{s} = (1+1)^d = 2^d = |X|$ vectors so it is enough to check that $RM(d,d) = \mathbb{F}_2^X$.

Let $P \in X$ and $y_i = \begin{cases} v_i & \text{if the } i^{th} \text{ coordinate of } P \text{ is } 1 \\ v_0 + v_i & \text{if the } i^{th} \text{ coordinate of } P \text{ is } 0 \end{cases}$

Then $1_{\{P\}} = y_1 \wedge \ldots \wedge y_d$. Expand using the distributive law

to see that $1_{\{P\}} \in RM(d,d)$. But $1_{\{P\}}$ for $P \in X$ spans $\mathbb{F}_2^n$ $\therefore$ the given vectors form a basis.

ii) $RM(d,r) = \text{span}\{v_{i_1} \wedge \dots \wedge v_{i_s} \mid 1 \le i_1 < \dots < i_s \le d,\ 0 \le s \le r\}$

These vectors are linearly independent by i), so they form a basis. Hence rank $RM(d,r) = 1 + \binom{d}{1} + \dots + \binom{d}{r}$ □

<u>Proposition 10·11</u>

$RM(d,r) = RM(d-1,r) \mid RM(d-1,r-1)$, $0 < r < d$

<u>Proof</u>

$RM(d-1,r-1) \subseteq RM(d-1,r)$, so the bar product (10·6) makes sense.

Arrange vectors in $X = \mathbb{F}_2^d$ so that $v_d = (0,\dots,0 \mid 1,\dots,1)$ with 2^{d-1} 0 digits then 2^{d-1} 1 digits.

Let $z \in RM(d,r)$, a sum of wedges $v_{i_1} \wedge \dots \wedge v_{i_s}$, some of which involve v_d. Write the sum of these terms as $y \wedge v_d$ and the sum of the remaining terms in z as x. (x, y do not involve v_d).

$z = x + (y \wedge v_d)$ $(x \in RM(d,r),\ y \in RM(d,r-1))$

Let x' be the vector containing the first 2^{d-1} components of x and y' containing the first 2^{d-1} components of y.

Now $x = (x' \mid x')$, $y = (y' \mid y')$ with $x' \in RM(d-1,r)$ and $y' \in RM(d-1,r-1)$.

We now have $y \wedge v_d = ~~(y' \mid 0)~~ (0 \mid y')$, $z = (x' \mid x'+y')$

$= (x' \mid x') + (0 \mid y')$ □

Proposition 10·12

The minimum weight of an $RM(d,r)$ code is 2^{d-r} $(0 \le r \le d)$

Proof

Clearly $wt(v_1 \wedge \dots \wedge v_r) = 2^{d-r}$ $\therefore$ min weight $\le 2^{d-r}$

$r = d \Rightarrow RM$ is the trivial code, min weight $1 = 2^0$

$r = 0 \Rightarrow RM$ is the repetition code, min weight $n = 2^d$

For $0 < r < d$ we use induction on d.

$d = 1$ is clear. Suppose that $d > 1$, and the result is true for RM codes of length $< 2^d$.

By 10·11, $RM(d,r) = RM(d-1,r) \mid RM(d-1, r-1)$.

By induction, min weights of $RM(d-1,r)$, $RM(d-1,r-1)$ are 2^{d-1-r} and 2^{d-r} respectively. By 10·7,

min weight $RM(d,r) = \min\{2 \cdot 2^{d-1-r}, 2^{d-r}\} = 2^{d-r}$ □

Example 10·13

$RM(5,1)$ is a $(32, 64, 16)$ code used by NASA for the Mariner missions to Mars, 1969 – 1976.

11 Cyclic Codes

Definition 11·1

A linear code $C \le \mathbb{F}_2^n$ is called *cyclic* if

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

Examples

Repetition, paper tape, Hamming's (7,4) code are all cyclic codes.

Establish a correspondence between $\mathbb{F}_2^n$ and $\frac{\mathbb{F}_2[x]}{(x^n-1)}$ as follows: $\mathbb{F}_2^n \leftrightarrow \{f \in \mathbb{F}_2[x] : \partial(f) < n\} \leftrightarrow \frac{\mathbb{F}_2[x]}{(x^n-1)}$

vector $\underline{a} = (a_0, \dots, a_{n-1}) \leftrightarrow a(x) = \sum_{i=0}^{n-1} a_i x^i$ polynomial

Lemma 11.2

A code $C \subseteq \frac{\mathbb{F}_2[x]}{(x^n-1)}$ is cyclic $\Leftrightarrow$

i) $0 \in C$ ii) $f, g \in C \Rightarrow f+g \in C$

iii) $f(x) \in \mathbb{F}_2[x]$, $g(x) \in C \Rightarrow f(x)g(x) \in C$

(i.e. C is an ideal in $\mathbb{F}_2[x]/(x^n-1)$

Proof

If $g(x) = a_0 + a_1 x + \dots + a_{n-1} x^{n-1} \mod (x^n-1)$

$xg(x) = a_{n-1} + a_0 x + \dots + a_{n-2} x^{n-1} \mod (x^n-1)$

C is cyclic $\Leftrightarrow$ i), ii) hold (C is linear) and

iii') $g \in C \Rightarrow xg \in C$ (special case $f(x) = x$ of iii))

In general $f(x) = \sum b_i x^i = \sum \underbrace{b_i x^i g(x)}_{\in C \text{ by iii'}} \in C$ by ii)

The converse is immediate.

Problem

Classify all cyclic codes of length n.

$\{$cyclic codes length $n\} \leftrightarrow \{$ideals of $\frac{\mathbb{F}_2[x]}{(x^n-1)}\}$

$\leftrightarrow \{$ideals in $\mathbb{F}_2[x]$ containing $x^n-1\}$

$\leftrightarrow \{$polynomials in $\mathbb{F}_2[x]$ dividing $x^n-1\}$

Note on 10.11 Proof

$V_d = (0,\dots,0\,|\,1,\dots,1)$, $V_i = (V_i'\,|\,V_i')$ $1 \le i \le d-1$

$z = (y \wedge V_d) + x$, where x, y are sums of wedges of $V_1,\dots,V_{d-1}$, so $z = (x'\,|\,x') + (y'\,|\,y') \wedge V_d$

Theorem 11.3

Let $C \le \frac{\mathbb{F}_2[x]}{(x^n-1)}$ be a cyclic code. Then $\exists!\ g(x) \in \mathbb{F}_2[x]$ such that

i) $C = \{f(x)g(x) \pmod{x^n-1} : f(x) \in \mathbb{F}_2[x]\}$

ii) $g(x) \mid x^n - 1$ ($p(x) \in \mathbb{F}_2[x]$ represents a codeword $\Leftrightarrow g(x) \mid p(x)$.

Definition 11.4

$g(x)$ is called the generator polynomial of C.

Proof

i) Consider $g(x)$, a non-zero polynomial of least degree representing a non-zero codeword. Note $\partial(g) < n$. C is cyclic, so $\supseteq$ in i) holds.

Let $p(x) \in \mathbb{F}_2[x]$ represent a codeword. By the division algorithm, $p(x) = q(x)g(x) + r(x)$ for some $q, r \in \mathbb{F}_2[x]$, $\partial(r) < \partial(g)$. Hence $r(x) = p(x) - q(x)g(x) \in C$

$\Rightarrow r(x) = 0$ as $\partial(r) < \partial(g) < n$. Hence $g(x) \mid p(x)$ proving $\subseteq$ in i).

ii) Take $p(x) = x^n - 1$ in the above to get ii)

Uniqueness : If g_1, g_2 are two generator polynomials then $g_1 \mid g_2$, $g_2 \mid g_1$ $\therefore g_1 = u g_2$ for $u \in \mathbb{F}_2^* = \{1\}$

11.3 tells us that we should seek generators $g(x)$ from amongst the factors of $X^n - 1$. With no conditions on n, life is dull, e.g. in $\mathbb{F}_2[x]$, $X^{2^r} - 1 = (x-1)^{2^r}$. To avoid this, we only consider the separable cyclic codes, i.e. when n is odd.

Lemma 11.5

Let $n \in \mathbb{N}$ be odd, and $k \geq \mathbb{F}_2$, in which $X^n - 1 = \prod_{j=1}^{t} f_j(x)$ with f_j linear (i.e. $k = Spl(X^n - 1)$). Then $f_1, \ldots, f_t$ are all distinct. Hence $X^n - 1$ has n distinct roots which form a cyclic group.

Note

1. In particular there are 2^t cyclic codes of length n.
2. This is false if n is even : $(X^2 - 1) = (x-1)^2$ over $\mathbb{F}_2$.

Proof

Suppose that $X^n - 1$ has a repeated factor. Then $\exists$ an extension $k \geq \mathbb{F}_2$ such that $X^n - 1 = (x-\alpha)^2 P(x)$, $\alpha \in k$, $P(x) \in k[x]$. Taking formal derivatives

$$nX^{n-1} = 2(x-\alpha)P(x) + (x-\alpha)^2 P'(x)$$
$$= (x-\alpha)\left[2P(x) + (x-\alpha)P'(x)\right]$$

Since n is odd, $n|_k \neq 0_k$. But $n\alpha^{n-1} = 0$

$\therefore \alpha = 0$ since n is odd. Hence $0 = \alpha^n = 1$ ⚡

If $S = \{\text{roots of } X^n - 1 \text{ in } k\}$, $|S| = n$.

Clearly $S \leq k^*$, and k^* is cyclic. Hence S is cyclic □

Lemma 11.6

Let C be a cyclic code of length n with generator polynomial $g(x) = g_0 + g_1 X + \dots + g_k X^k$ of degree k (i.e. $g_k \neq 0$). Then a basis for C is $\{g(x), Xg(x), \dots, X^{n-k-1} g(x)\}$ and in particular $\operatorname{rank} C = n - k$.

Proof.

i) Linear Independence: Suppose that $f(x)g(x) \equiv 0 \ (X^n - 1)$ for some $f(x) \in \mathbb{F}_2[X]$, $\partial(f) < n-k$. Then $\partial(fg) < n$, so $f(x)g(x) = 0$, hence $f(x) = 0$, i.e. every dependence relation is trivial.

ii) Spanning: Let $p(x) \in \mathbb{F}_2[X]$ represent a codeword. WLOG $\partial(p) < n$. Since $g(x)$ is generator polynomial, $g(x) | p(x)$ i.e. $p(x) = f(x)g(x)$ for some $f(x)$. Now $\partial(f) = \partial(p) - \partial(g)$ so $\partial(f) < n-k$, i.e. $p(x) \in \operatorname{span}\{g(x), Xg(x), \dots, X^{n-k} g(x)\}$

Corollary 11.7

A generator matrix for C is

$$G = \begin{pmatrix} g_0 & g_1 & \dots & g_k & 0 & \dots & 0 \\ 0 & g_0 & \dots & g_{k-1} & g_k & \dots & 0 \\ \vdots & & \ddots & & & \ddots & \\ & & & & g_{k-1} & g_k & \end{pmatrix}$$

Definition 11.8

The parity check polynomial $h(x) \in \mathbb{F}_2[X]$ is defined as $g(x)h(x) = X^n - 1$. If $h(x) = h_0 + h_1 X + \dots + h_{n-k} X^{n-k}$ then the parity check matrix (9.12) is

$$H = \begin{pmatrix} h_{n-k} & h_{n-k-1} & \dots & h_1 & h_0 & 0 & \dots & 0 \\ 0 & h_{n-k} & \dots & h_1 & h_0 & 0 & \dots & 0 \\ \vdots & & \ddots & & & \ddots & & \\ & & & & & & h_1 & h_0 \end{pmatrix}$$

Proof

Take the dot product of the i^{th} row of G and the j^{th} row of H and compare coefficients $\square$

Lemma 11.9

Suppose $X^n - 1$ is $g(x)h(x)$. Then g is a generator of a cyclic code (of length n) and h is a generator for another cyclic code which is the reverse of a dual code $C^\perp$. In particular, the dual of a cyclic code is cyclic.

Proof

Write $X^n - 1 = \prod_{i=1}^{t} f_i(x)$ and let β_i be a root of f_i for some extension of $\mathbb{F}_2$. Then $f_i(x)$ is the minimal polynomial of β_i, and the corresponding code is just the set of polynomials $a(x)$ such that $a(\beta_i) = 0$.

Consider $\{\alpha_j\}_{1 \le j \le s}$ elements in some extension of $\mathbb{F}_2$ and let $g(x) = \text{lcm}$ of their min. polys. over $\mathbb{F}_2$. Then the cyclic code with gen. poly. $g(x)$ consists of polys. $a(x)$ such that $a(\alpha_j) = 0$.

$\therefore$ prescribing a cyclic code is the same as prescribing common roots of its codewords.

Definition 11.10

A defining set for cyclic code C over $\mathbb{F}_2$ is a set A of elements in some field $K \supseteq \mathbb{F}_2$ such that: $a(x) \in C$

$\iff$ Each element of A is a root of a.

Equivalently, C is generated by the least common multiples of the min. polys. of the elements of A.

12 BCH Codes

Let n be odd. Pick $r \geq 1$ such that $2^r \equiv 1 \ (n)$. $K = \mathbb{F}_{2^r}$.

Let $\mu_n(K) = \{x \in K : x^n = 1\} \leq K^*$, a cyclic group of order n (since $n \mid 2^r - 1 = |K^*|$).

So $\mu_n(K) = \{1, \alpha, \ldots, \alpha^{n-1}\}$

12.1 Definition

The BCH (Bose - Chaudhuri - Hocquenghem) code with design distance δ is the cyclic code C of length n with defining set $A = \{\alpha, \alpha^2, \ldots, \alpha^{\delta-1}\}$

Here $\delta \in \mathbb{N}$ such that $1 \leq \delta \leq n$.

BCH Codes

Theorem 12.2 (BCH Bound)

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

Lemma 12.3

Work over any field k. The vander Monde determinant

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

Proof of 12.2

Let $P(x) = b_0 + b_1 x + \ldots + b_{n-1} x^{n-1}$ be a non-zero codeword.

$$H = \begin{pmatrix} 1 & \alpha & \alpha^2 & \cdots & \alpha^{n-1} \\ 1 & \alpha^2 & \alpha^4 & \cdots & \alpha^{2(n-1)} \\ \vdots & & & & \vdots \\ 1 & \alpha^{\delta-1} & \cdots & \cdots & \alpha^{(\delta-1)(n-1)} \end{pmatrix} \qquad H \begin{pmatrix} b_0 \\ b_1 \\ \vdots \\ b_n \end{pmatrix} = 0$$

By 12.3 any $\delta - 1$ columns of H are linearly independent over $\mathbb{F}_2$ since no two of the n powers of α are equal or 0 (11.5, But any codeword of C is a dependence relation between columns of H. Hence $wt(P(x)) \geq \delta$, so $d(C) \geq \delta$.

Remark

H is not a parity check matrix as in 9.12 since the entries do not lie in $\mathbb{F}_2$.

Decoding BCH Codes

Let C be a BCH code, length n, over $\mathbb{F}_2$, design distance δ, defining set $A = \{\alpha, \alpha^2, \ldots, \alpha^{\delta-1}\}$ where α is the n^{th} root of unity lying in $k \geq \mathbb{F}_2$. We know by 12.2 that C is t-error correcting, for $t = \lfloor \frac{1}{2}(\delta - 1) \rfloor$. Suppose that $c = (c_0, \ldots, c_{n-1})$ is sent and we recieve $r + e$ where

the error vector e has $\leq t$ non-zero entries, i.e. $|\mathcal{E}| \leq t$ where

$\mathcal{E} = \{0 \leq j \leq n-1 : e_j = 1\}$

Problem

Find e, hence recover c. Under $\mathbb{F}_2^n \Leftrightarrow \frac{\mathbb{F}_2[x]}{(x^n-1)}$,

$c(x) = \sum_{j=0}^{n-1} c_j x^j$, $r(x) = \sum_{j=0}^{n-1} r_j x^j$, $e(x) = \sum_{j=0}^{n-1} e_j x^j$

polynomials of degrees $< n$, corresponding to c, r, e.

We know that $c(x)$ satisfies $c(\alpha) = c(\alpha^2) = \ldots = c(\alpha^{\delta-1}) = 0$

$\therefore r(\alpha) = e(\alpha)$, $r(\alpha^2) = e(\alpha^2), \ldots, r(\alpha^{\delta-1}) = e(\alpha^{\delta-1})$

Idea: Calculate $r(\alpha^j)$ $(1 \leq j \leq \delta-1)$. If all zero then $r(x)$ is a codeword and there are no errors (or $\geq \delta+1$ errors)

Otherwise, assume $0 < |\mathcal{E}| < t$

Definition 12.4

The error-locator polynomial is

$$\sigma(x) = \prod_{j \in \mathcal{E}} (1 - \alpha^j x) \in k[x]$$

If we know $\sigma(x)$, we can find which α^{-j} are roots of $\sigma(x)$ and hence find the indices where errors have occurred. Correct the errors by changing the entries. How do we find $\sigma(x)$?

Theorem 12.5

Assume $\partial(\sigma) = |\mathcal{E}| \leq t$. Then $\sigma(x)$ is the unique polynomial over k, of least degree, such that:

i) $\sigma(0) = 1$

ii) $\exists\, \omega(x) \in k[x]$, $\partial(\omega) \leq t$, such that

$$\sigma(x) \times \sum_{j=1}^{2t} r(\alpha^j) x^j \equiv \omega(x) \pmod{x^{2t+1}}$$

The Algorithm

1. Input: $r(x)$ (received)
2. Compute: $\sum_{j=0}^{2t} r(\alpha^j) x^j$
3. Put $\sigma(x) = \sum_{j=0}^{t} \sigma_j x^j$ with $\sigma(0) = 1$ and compute the coefficients of x^i for $t+1 \leq i < 2t$ to obtain t linear equations for the σ_i ($1 \leq i \leq t$).
4. Solve these equations over k and keep the solutions of least degree.
5. Compute $\mathcal{E} = \{0 \leq i \leq n-1 \mid \sigma(\alpha^{-i}) = 0\}$ and check that $|\mathcal{E}| = \partial(\sigma)$
6. Put $e(x) = \sum_{i \in \mathcal{E}} x^i$, $c(x) = r(x) + e(x)$ and check that $c(x)$ is a codeword.

Proof of 12.5

Existence: Let $\omega(x) = -x\sigma'(x) = \sum_{i \in \mathcal{E}} \alpha^i x \prod_{j \in \mathcal{E}, j \neq i} (1 - \alpha^j x)$ (the error-co-locator polynomial).

Formally, we work in ~~$k[[x]]$~~ $\subseteq \mathrm{Frac}\, k[x]$, $k(x)$

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

$$= \sum_{j=1}^{\infty} e(\alpha^j) x^j \Rightarrow \sigma(x) \cdot \sum_{j=1}^{\infty} e(\alpha^j) x^j = \omega(x)$$

By definition of C, $c(\alpha^j) = 0 \quad \forall j \leq \delta - 1$, and as $r = c + e$, $r(\alpha^j) = e(\alpha^j) \quad \forall j \leq 2t$.

We have shown that $\sigma(x) \sum_{j=1}^{2t} r(\alpha^j) x^j \equiv \omega(x) \pmod{x^{2t}}$.

i), ii) hold with $\omega(x) = -x\sigma'(x)$ and so both $\sigma(x)$ and $\omega(x)$ have degree $|\mathcal{E}| \leq t$.

Uniqueness:

Suppose $\tilde{\sigma}(x), \tilde{\omega}(x) \in k[x]$ also satisfies i), ii), and $\partial(\tilde{\sigma}(x)) \leq \partial(\sigma(x))$. If $i \in \mathcal{E}$, $\omega(\alpha^{-i}) = \prod_{j \in \mathcal{E}, j \neq i} (1 - \alpha^{j-i}) \neq 0$

So $\sigma(x), \omega(x)$ are coprime. By ii),

$$\tilde{\sigma}(x)\omega(x) = \sigma(x)\tilde{\omega}(x) = \sigma(x)\tilde{\sigma}(x) \sum_{j=1}^{2t} r(\alpha^j) x^j \pmod{x^{2t+1}}$$

But $\sigma, \tilde{\sigma}, \omega, \tilde{\omega}$ all have deg $\leq t$, hence

$$\tilde{\sigma}(x)\omega(x) = \sigma(x)\tilde{\omega}(x)$$

Finally, $\sigma(x) \mid \tilde{\sigma}(x)$ since $\sigma(x), \omega(x)$ coprime.

But $\partial(\tilde{\sigma}) \leq \partial(\sigma)$ $\therefore \tilde{\sigma}(x) = k\sigma(x)$, $k \in K$

By i), $\sigma = \tilde{\sigma}$. □

13 One-Time Pads and Shift Registers

We are leading towards cryptography starting with Vernam and Mauborgne's

Definition 13.1 (one-time pad) 1976

Let A be an alphabet, and take a "random" sequence $k_1, k_2, \ldots, k_n$ of letters from A "the key"

k_j is chosen uniformly from the elements of A independently for each j.

We encipher $m = a_1 \ldots a_n$ as $b_1 \ldots b_n$ where

$$b_j = a_j + k_j \pmod{|A|}$$

One Time Pad

We have a key-stream $k_1, k_2, \ldots, k_n$, with $m = a_1 \ldots a_n$ enciphered as $b_j = a_j + k_j \pmod{|A|}$. Each random k_j is only used once.

These were used to encrypt White House/Kremlin hotline in the Cold War.

Flaws

1. Generating a truly random string of letters.
2. To decipher, the key must be known (so the key must also be sent in a secure way).

We will be replacing the random key stream, by a suitable pseudo-random sequence, such as:

Shift Registers

Definition 13.2

A (general) feedback shift register is a function $f: \mathbb{F}_2^d \to \mathbb{F}_2^d$ of the form

$f: (x_0, x_1, \ldots, x_{d-1}) = (x_1, x_2, \ldots, x_{d-1}, C(x_0, x_1, \ldots, x_{d-1}))$,

for some map $C: \mathbb{F}_2^d \to \mathbb{F}_2$. This register has length d.

f is linear (LFSR) if C is a linear map.

$(x_0, \ldots, x_{d-1}) \mapsto \sum_{i=0}^{d-1} a_i x_i$ (N.B. if a_0 was 0, then the value of x_0 does not ~~appear~~ affect any later values and we could consider the remaining $d-1$ registes as a simpler LFSR).

We will assume that $a_0 = 1$ in the definition.

The initial feed $(y_0, \dots, y_{d-1})$ produces the output stream $(y_n)_{n \geq}$
such that $y_{n+d} = C(y_n, y_{n+1}, \dots, y_{n+d-1}) = \sum_{i=0}^{d-1} a_i y_{n+i}$
i.e. we have a sequence determined by a linear recurrence
relation with auxiliary polynomial $C(X) = X^d + a_{d-1} X^{d-1} + \dots + a_1 X + a$

Solving the recurrence relations over $\mathbb{F}_2$ are similar to solving over $\mathbb{R}$ or $\mathbb{C}$, but more complicated over $\mathbb{F}_2$ (since $n^2 \equiv n \ (2)$).

See Sheet 4, question 20.

Definition 13.3

The feedback polynomial is $\tilde{C}(X) = a_0 X^d + a_1 X^{d-1} + \dots + a_{d-1} X +$

Now suppose that we have a sequence of elements $(y_n)_{n \geq 0} \subset \mathbb{F}_2$

Definition 13.4

The generating function $G(X)$ of $(y_n)_{n \geq 0}$ is $G(X) = \sum_{j=0}^{\infty} y_j X^j$

Compare the following with 12.5

Theorem 13.5

The sequence $(y_n)_{n \geq 0}$ in $\mathbb{F}_2$ is the output from a LFSR
with auxiliary polynomial $C(X)$
$\Leftrightarrow G(X) = \frac{B(X)}{\tilde{C}(X)}$, $B(X) \in \mathbb{F}_2[X]$, with $\partial(B) < \partial(C)$
and $\tilde{C}(X) = X^{\partial(C)} C(\frac{1}{X}) \in \mathbb{F}_2[X]$

Remark

If we can recover C or $\tilde{C}$ from G, then we have
recovered the LFSR from the stream.

Proof

$C(X) = \sum_{j=0}^{d} a_j X^j$. $\tilde{C}(X) = a_0 X^d + a_1 X^{d-1} + \ldots + a_{d-1} X + a_d$

The condition is that $\left(\sum_{j\geq 0} y_j X^j\right) \tilde{C}(X)$ is a polynomial of degree $< d$. This holds $\Leftrightarrow \sum_{j=0}^{d-1} a_j y_{n-d+j} = y_n \quad \forall n \geq d$.

$\Leftrightarrow \sum_{j=0}^{d-1} a_j y_{n+j} = y_{n+d} \quad \forall n \geq 0$

$\Leftrightarrow (y_n)_{n\geq 0}$ is the output from a LFSR. □

Problem

Suppose that $(y_n)_{n\geq 0}$ is the output of a LFSR. Can you find $a_0, \ldots, a_{d-1} \in \mathbb{F}_2$ such that $y_{n+d} = \sum_{j=0}^{d-1} a_j y_{n+j} \quad \forall n \geq 0$?

13.6 Berlekamp - Massey Method

Observe that $\sum_{j=0}^{d} a_j y_{n-j} = 0 \quad (a_0 = 1) \quad \forall n \geq d$.

$$\underbrace{\begin{pmatrix} y_d & y_{d-1} & \cdots & y_1 & y_0 \\ y_{d+1} & y_d & \cdots & y_2 & y_1 \\ \vdots & & & & \vdots \\ y_{2d} & y_{2d-1} & \cdots & y_{d+1} & y_d \end{pmatrix}}_{A_d} \begin{pmatrix} a_1 \\ \vdots \\ a_d \end{pmatrix} = 0 \qquad (*)$$

If we know the register has length $\geq r$, start at $i = r$.

Compute $\det A_i$.

i) If $\det A_i \neq 0$, then $d > i$. Replace i by $i+1$ and repeat.

ii) If $\det A_i = 0$, solve $(*)$ for the a_i, by Gaussian Elimination, and test the solution over as many terms of the sequence as we like. If it fails this test, then $d > i$ so replace i by $i+1$ and repeat.

13.7 Secret Sharing (Tappe, Washington)

If Cambridge University is attacked by the GOVE virus, all the dons retreat to the CMS. Entry to the CMS requires entering

$m \in \mathbb{N}$ ("the secret"), known only to their leader Dr Cowley : C.
Each of the n dons knows a pair of numbers (their "shadow").
We require that in C's absence, any k dons can reconstruct m from their shadows but no $k-1$ can. Any method to solve this is known as a (k, n) threshold scheme.

Shamir's Threshold Scheme (1979)

Suppose that $0 < m < N$ (n random).

i) C chooses a prime $p > m, N$; then chooses $k-1$ integers $a_1, \ldots, a_{k-1}$ at random and distinct integers $x_1, \ldots, x_n$ at random such that $0 \leq a_j \leq p-1$, $1 \leq x_j \leq p-1$; then sets $a_0 = m$.

ii) C computes $P(r) = a_0 + a_1 x_r^1 + a_2 x_r^2 + \ldots + a_{k-1} x_r^{k-1} \pmod p$ choosing $0 \leq P(r) \leq p-1$ $(1 \leq r \leq n)$

iii) C now gives the r^{th} don their shadow pair $(x_r, P(r))$, secret from everybody else.

iv) p is known to all participants. C then burns his calculations

Claim

Any k participants can recover m, but not $k-1$.

Suppose that there are k-dons with shadows $(y_j, Q_j) = (x_{r_j}, P(r_j))$, $1 \le j \le k$, are together. Rewrite the system as

$$\begin{pmatrix} 1 & y_1 & \cdots & y_1^{k-1} \\ 1 & y_2 & \cdots & y_2^{k-1} \\ \vdots & & & \\ 1 & y_k & \cdots & y_k^{k-1} \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_{k-1} \end{pmatrix} = \begin{pmatrix} Q_1 \\ \vdots \\ Q_k \end{pmatrix} \pmod p$$

V is a van der Monde matrix and from 12·3 we know that it has a unique solution mod p

$$\Leftrightarrow \prod_{1 \le j < i \le k} (y_i - y_j) = \det V \not\equiv 0 \pmod p$$

Hence the system $z_0 + y_1 z_1 + \ldots + y_1^{k-1} z_{k-1} = Q_1, \ldots,$ $z_0 + y_k z_1 + \ldots + y_k^{k-1} z_{k-1} = Q_k$ has a unique solution $\underline{z}$. But we know that $\underline{a}$ is a solution, so $\underline{z} = \underline{a}$, and the secret $m = z_0$.

On the other hand, $\begin{vmatrix} y_1 & y_1^2 & \cdots & y_1^{k-1} \\ y_2 & y_2^2 & \cdots & y_2^{k-1} \\ \vdots & & & \\ y_{k-1} & y_{k-1}^2 & \cdots & y_{k-1}^{k-1} \end{vmatrix} = \left(\prod_{l=1}^{k-1} y_l\right) \prod_{1 \le j < i \le k-1} (y_i - y_j) \not\equiv 0 \pmod p$

So the system of equations $z_0 + y_1 z_1 + y_1^2 z_2 + \ldots + y_1^{k-1} z_{k-1} = Q_1$ $, \ldots,$ $z_0 + y_{k-1} z_1 + \ldots + y_{k-1}^{k-1} z_{k-1} = Q_{k-1}$ has a solution, whatever value of z_0 is taken. So k-1 dons cannot decide if any value of m is more likely than any other.

Exercise

Is the secret m compromised if the x_j become known to everybody?

5 Cryptography

14 Introduction to Cryptosystems

This is the art of codemaking, where are message is enciphered (or encrypted) so that it can only be read by the intended recipient.

$$A \xrightarrow{E} B$$

$M \in \mathcal{M}$ set of all messages $\quad C \in \mathcal{C}$ set of all encrypted messages

14.1 Terminology

Message M (plaintext), a string from some finite alphabet $\mathcal{A}$.

We encipher M using a key from a finite set $\mathcal{K}$ of possible keys

We have

i) An encrypting function $e_K : \mathcal{A} \to \mathcal{B} \quad (K \in \mathcal{K})$

ii) Encrypted plaintext is called ciphertext

iii) Decrypting function $d_K : \mathcal{B} \to \mathcal{A}$ with $d_K(e_K(m)) = m \quad \forall m$ $(K \in \mathcal{K})$

iv) $\langle \mathcal{M}, \mathcal{K}, \mathcal{C} \rangle$ is called a cryptosystem

Examples

i) Simple Substitution "monoalphabetic cipher"

key = a permutation π of $\mathcal{A}$. Each letter m is replaced by $\pi(m)$

e.g. a 26-letter string, UXEB..., $A \mapsto U$, $B \mapsto X$, etc

'easy' to decipher/break the code: check the frequency of letters in English and compare this to the frequencies in the ciphertext. The context of the message also helps to find the permutation.

ii) Vigenère Cipher (1523, "le chiffre indechiffrable")

The key is a string of D letters $k_1, \ldots, k_D$ from $\mathcal{A}$ repeated to give an infinite sequence $k_1, k_2, \ldots, k_D, k_{D+1}, \ldots, k_{2D}$

where $a' \equiv a \ (D) \Rightarrow k_{a'} = k_a$. The plaintext $M = a_1 \ldots a_N$ enciphered as $b_1 \ldots b_N$ such that

$b_j \equiv a_j + k_j \pmod{|\mathcal{A}|}$

D=1 : Caesar cipher "polyalphabetic"

e.g. D=4, key: BEER

M: OLD PECULIAR
k: BEE RBEERBEE
C: QQI HGM ZEKFW

For a long key, a given letter is encrypted differently depending on its place in M, so a simple frequency analysis will not work. Babbage used a more clever frequency analysis to break it (Singh, the Code Book).

Breaking a Cryptosystem

E knows possible functions e_K, d_K but not k. If E intercepts some ciphertext C she will try to decode it (or work out the key so that she can decode all future ciphertexts). 3 forms of attack:

i) Ciphertext only (L1)

E has a single piece of ciphertext

ii) Known Plaintext (L2)

E has a piece of plaintext and matching ciphertext, and seeks the key.

iii) Chosen plaintext (L3)

E can obtain arbitary plaintexts of choice, and their corresponding ciphertexts.

Examples

In the English alphabet, simple substitution and the Vigenère Cipher are both vulnerable at L1. E can find the key, if the ciphertext is long enough.

All ciphers are vulnerable at L3 : the system is finite i.e.

only a finite number of letters and keys and E can carry out an exhaustive search. To withstand L3 we want the effect of searching and/or the time it takes to be prohibitively large.

Equivocation

Suppose that we have a random plaintext M chosen from $\mathcal{M}$, the set of possible messages, with a certain probability distribution. Then choose some random $k \in \mathcal{K}$ independently of M. The ciphertext is a random variable $C = e_k(M) \in \mathcal{C}$

Definition 14.2

Modelled as above as transmitting M through a channel to produce C, with noise coming from the choice of k.

$H(M|C)$ is the message equivocation.

$H(k|C)$ is the key equivocation.

Theorem 14.3

$H(k|C) \geq H(M|C)$

Proof

$H(k|C) = H(k,C) - H(C) = H(k,M,C) - H(C)$ (since $M = d_k(C)$ is a function of (k,C)).

$H(k|C) = H(k,M,C) - H(M,C) + H(M,C) - H(C)$

$= \underbrace{H(k|M,C)}_{\to 0} + H(M|C) \geq H(M|C)$ □

Definition 14.4

Shannon defines perfect secrecy for a cryptosystem, if the ciphertext gives no information about the plaintext, i.e. $H(M|C) = H(M)$. Thus $H(M,C) = H(M) + H(C)$, i.e. M, C are independent.

Remark

Equivalently, $I(M,C) = H(C) - H(C|M) = H(C) + H(M) - H(M,C) = 0$ (7.1)

Proposition 14.5

Perfect secrecy $\Rightarrow |K| \geq |M|$

Proof:

Fix $m_0 \in M$, and $k_0 \in K$, both with strictly positive probability.

Then $c_0 = e_{k_0}(m_0)$ also has strictly positive probability.

For any $m \in M$, we have $\mathbb{P}(C = c_0 | M = m) = \mathbb{P}(C = c_0)$

$= \mathbb{P}(C = c_0 | M = m_0) = \mathbb{P}(k = k_0) > 0$

So $\exists k \in K$ with $c_0 = e_k(m)$. If m_1, m_2 give the same key k, then $e_k(m_1) = c_0 = e_k(m_2)$ $\therefore m_1 = m_2$

Hence $m \mapsto k$ is injective. □

Example 14.6

The One-Time Pad (13.1) has perfect secrecy. Suppose $|A| = q$

$\mathbb{P}(M = \underline{m}, C = \underline{c}) = \mathbb{P}(M = \underline{m}, k = \underline{c} - \underline{m} \pmod{q})$

$= \mathbb{P}(M = \underline{m})\, \mathbb{P}(k = \underline{c} - \underline{m}) = \mathbb{P}(M = m) \frac{1}{q^n}$

$\therefore$ M, C independent. (n is the length of string)

We might replace the key by a pseudo-random sequence (e.g. using LFSR). But this is vulnerable at L2.

Theorem 14.7

The use of the output sequence from LFSR as a pseudo-random one time pad is vulnerable at L2.

Proofs

Message $\underline{m} = m_1 m_2 \ldots$, $m_i \in \mathbb{F}_2$, $\underline{x} = x_1 x_2 \ldots$, stream from LFSR

Ciphertext $c_i = m_i + x_i \pmod 2$, $1 \le i < \infty$.

If m_i, c_i are known, the key sequence x_i is known ($\equiv m_i + c_i$ (2)

If the register has feedback coefficients $a_0, \ldots, a_{d-1}$ as in 13·3, then once E knows any $2d$ consecutive x_i the output stream, the a_i can be found by solving d linear equations (BM, 13·6). □

Unicity Distance

In attacking the simple substitution cipher, E [redacted]:

- examines letter frequencies
- knows that the original message makes sense
- (for reasonable length) $\exists!$ k, a key giving a sensible message from known ciphertext.

How long a message does E need for this argument to apply.

Take $\mathcal{M} = \mathcal{C} = \Sigma$ say. Send n messages $M^{(n)} = (M_1, \ldots, M_n)$ encrypted as $C^{(n)} = (C_1, \ldots, C_n)$ using the same key.

Definition 14·8

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

Now $H(k \mid C^{(n)}) = H(k, C^{(n)}) - H(C^{(n)}) = H(k, M^{(n)}) - H(C^{(n)})$

$= H(k) + H(M^{(n)}) - H(C^{(n)})$

Assume: i) that all keys are equally likely, so $H(k) = \log_2 |\mathcal{K}|$

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

iii) All sequences of ciphertext are equally likely, so $H(C^{(n)}) = n \log_2 |\Sigma|$

Then $H(k|C^{(n)}) = \log_2 |\mathcal{K}| + nH - n\log_2 |\Sigma| \geq 0$

$\Leftrightarrow n \leq U := \frac{\log |\mathcal{K}|}{\log |\Sigma| - H}$

Now $0 \leq H \leq \log_2 |\Sigma|$. To make U large, make $|\mathcal{K}|$ large, or use a message source with little redundancy.

Example

The English entropy per letter is roughly 1-2 bits. Take a substitution cipher with key in S_{27}.

$|\mathcal{K}| = 27!$, $\log |\mathcal{K}| \approx 93.14$, $|\mathcal{C}| = 27$.

$U = \frac{\log(27!)}{\log 27 - 1.2} \approx 26.2$ $\therefore$ we expect to determine the key uniquely if the ciphertext is > 26 letters.

15 Stream Ciphers

Recall that a stream is a sequence in $\mathbb{F}_2$. For a plaintext stream $p_0, p_1, p_2, \ldots$ and a key stream $k_0, k_1, \ldots$, we set ciphertext stream to be $z_0, z_1, \ldots$, where $z_n = p_n + k_n$ (2).

Remark

This is an example of a private key / symmetric system (both A and B need the key k in order to encrypt or decrypt).

Example Without k, deciphering is impossible ($U = \infty$) for the One time pad: key stream is a random sequence $k_i = K_i$ where K_i are iid with $\mathbb{P}(K_i = 0) = \mathbb{P}(K_i = 1) = \frac{1}{2}$

Ciphertext $Z_i = p_i + K_i$, $\{Z_i\}$ are iid with $\mathbb{P}(Z_i = 0) = \mathbb{P}(Z_i = 1) = \frac{1}{2}$

Lemma 15.1

Let $(x_n)_{n\geq 0}$ be a stream produced by a (general) feedback shift register of length d. Then $\exists N, M \leq 2^d$ such that $x_{N+r} = x_r$, $\forall r \geq M$.

Proof

Let $f: \mathbb{F}_2^d \to \mathbb{F}_2^d$ be a register. Let $V_j = (x_j, x_{j+1}, \dots, x_{j+d-1})$ so $V_{j+1} = f(V_j)$. Since $|\mathbb{F}_2^d| = 2^d$, $V_0, V_1, \dots, V_{2^d}$ are not all distinct. So $\exists\, 0 \leq a < b \leq 2^d$ such that $V_a = V_b$. Let $M = a$, $N = b - a$. So $V_M = V_{N+M}$ and $V_r = V_{r+N}$ $\forall r \geq N$ (by induction, applying f), so $x_r = x_{r+N}$ $\forall r \geq M$ $\square$

Remarks

1. The maximum period of a feedback shift register of length d is 2^d.
2. The maximum period of a LFSR of length d is $2^d - 1$ (Sheet 4, Q18)
3. Stream ciphers using LFSR of length d are vulnerable at L2 due to Berlekamp-Massey.

However, they have advantages:

- Cheap (computationally), fast, easy to use
- Messages can be encrypted/decrypted "on the fly"
- Error tolerant

Lemma 15.2

If (x_n), (y_n) are outputs from LFSR of lengths M, N respectively. Then

a) $(x_n + y_n)$ is the output from a LFSR of length $M+N$.

b) $(x_n y_n)$ is the output from a LFSR of length MN

Proof (Sketch, complete using Galois Theory)

Assume that the auxiliary polynomials P, Q each have distinct roots $\alpha_1, \ldots, \alpha_M$ and $\beta_1, \ldots, \beta_N$ in some extension $K/\mathbb{F}_2$.

Then $x_n = \sum_{i=1}^{M} \lambda_i \alpha_i^n$, $y_n = \sum_{j=1}^{N} \mu_j \beta_j^n$ where $\lambda_i, \mu_i \in K$.

$x_n + y_n = \sum_{i=1}^{M} \lambda_i \alpha_i^n + \sum_{j=1}^{N} \mu_j \beta_j^n$ is produced by a LFSR with auxiliary polynomials $P(X)Q(X)$.

$x_n y_n = \sum_{i=1}^{M} \sum_{j=1}^{N} \lambda_i \mu_j (\alpha_i \beta_j)^n$, produced by a LFSR with auxiliary polynomial $\prod_{i=1}^{M} \prod_{j=1}^{N} (X - \alpha_i \beta_j)$ lying in $\mathbb{F}_2[X]$ (which must be proved) □

i) Adding outputs of 2 LFSRs is no more economical than producing the same stream with a single LFSR.

ii) Multiplying streams looks more promising until we realise that $x_n y_n = 0$ about 75% of the time (see Körner).

ii) Non-linear feedback shift registers are hard to analyse, and E may understand them better than we do.

Lemma 15.3

Suppose that (x_n) is a periodic stream with period N, and (y_n) similarly with period M. Then $(x_n + y_n)$, $(x_n y_n)$ are periodic with periods dividing $\text{lcm}(N, M)$.

Proof

Easy exercise.

16 Public Key Cryptography

An example of an asymmetric cryptosystem (i.e. we have 2 keys, one for encrypting and one for decrypting). We have the private key (for decrypting) and the public key (for encrypting).
Anyone can encrypt the plaintext message and send it to B, but only B knows the private key to allow decryption.

Easy: encrypt $m \in M$ as $c = e_K(m)$ using known e_K.
Hard: given $c \in \mathcal{C}$, find $m \in M$ such that $c = e_K(m)$.
e_K is a "one way function": it is easy to find $e_K(m)$, difficult to find $e_K^{-1}(c)$. We aim to be secure at L3.
This relies on two problems which are hard (to solve quickly).

i) Factoring: Let $N = pq$, p, q large primes. Given N, find p, q
ii) Discrete Logarithm: Let p be a large prime, g a primitive root mod p (i.e. $\mathbb{F}_p^* = \langle g \rangle$). Given $x \in \mathbb{F}_p^*$, find a such that $x \equiv g^a \pmod{p}$.

Definition 16.1

An algorithm runs in polynomial time if
operations $\leq c(\text{input size})^d$ for constants c, d.

Examples

i) Integer arithmetic ($+, -, \times$, division algorithm)
ii) Computation of gcd by the Euclidean Algorithm
iii) Primality Testing (Agrawal et al, 2002)
iv) Modular Exponentiation, computing $x^{(N)}$ using repeated squaring.

Polynomial time algorithms are not known for factoring and discrete logarith

- Trial division (properly organised) takes time $O(\sqrt{n})$
- Baby Step, Giant Step algorithm (set $m = \lceil \sqrt{p} \rceil$, $a = qm + r$, $0 \le q, r < m$. Then $x \equiv g^a \equiv g^{qm+r} \ (p)$ $\therefore g^{qm} \equiv g^{-r} x \ (p)$
  List $g^{qm} \ (p)$ for $q \equiv 0, \ldots, m-1$ and $g^{-r} x \ (p)$ for $r = 0, \ldots, m-1$. Sort the two lists and look for a match.
  We can find discrete logs in time and storage $O(\sqrt{p} \log p)$
- Factor base (number field sieve), $O(e^{c(\log N)^{\frac{1}{2}} (\log\log N)^{\frac{2}{3}}})$, c known

RSA Labs offered prizes for factoring large numbers.

Challenge No.	# Decimal Digits	Factored	Prize
RSA-576	174	Dec 2003	\$10k
RSA-640	193	Nov 2005	\$20k
RSA-704	212	?	\$30k

Recall $\phi(N) = |\{1 \le a \le N \mid (a,N) = 1\}| = |(\mathbb{Z}/N\mathbb{Z})^*|$

Euler-Fermat

$a^{\phi(N)} \equiv 1 \ (N)$ for each a with $(a,N) = 1$

Fermat's Little Theorem

$a^p \equiv a \ (p)$ $\forall a$ or $a^{p-1} \equiv 1 \ (p)$ for all a with $(a,p) = 1$

Lemma 16.2

Let $p = 4k-1$ prime, $d \in \mathbb{Z}$. If $x^2 \equiv d \ (p)$ is soluble (i.e. $(\frac{d}{p}) = 1$) then solutions are $x \equiv \pm d^k \ (p)$.

Proof

Let x_0 be a solution. WLOG $x_0 \not\equiv 0 \ (p)$.

$d^{2k-1} = {x_0}^{2(2k-1)} \equiv {x_0}^{p-1} \equiv 1 \ (p)$ $\therefore (d^k)^2 \equiv d \ (p)$ $\square$

16.3 Robin (Williams) Cipher (1979)

i) Choose two large distinct primes $p, q \equiv 3 \ (4)$

ii) Private key (p, q), Public key $N = pq$

iii) $\mathcal{M} = \mathcal{C} = \{0, 1, \ldots, N-1\}$

iv) $m \in \mathcal{M}$ is encrypted as $c \equiv m^2 \pmod{N}$

v) We restrict the alphabet so that $(m, N) = 1$, $m > \sqrt{N}$

If B recieves $c \equiv m^2 \ (N)$. We use 16.2 to solve

$m = \delta c^k \ (p)$, $m = \varepsilon c^l \ (q)$, $\delta, \varepsilon = \pm 1$.

$\exists u, v$ with $up + vq = 1$. CRT:

$$m = \delta c^k + up(\varepsilon c^l - \delta c^k) \quad (N)$$

[CRT: $x \equiv c \ (p)$, $x \equiv d \ (q)$, ($ap + cq = 1$ where) $\Leftrightarrow x \equiv c + ap(d - c) \ (pq)$

We claim that all 4 posible values can occur and are distinct.

To decipher c we find all 4 square roots mod N and choose the one making sense. Messages should contain enough redundancy for only one of these four choices to make sense.

Lemma 16.4

Let $N = pq$, a product of distinct primes. $\left(\frac{m}{N}\right) = 1$. Then

i) Either $(m, N) = 1$ and $\exists$ exactly 4 square roots of $m \pmod{N}$

ii) Or $(m, N) = p$, or q, and $\exists$ exactly 2 square roots

iii) $m \equiv 0 \ (N)$ so 0 is the unique square root of 0 $(\text{mod } N)$.

Proof

$x \in \mathbb{Z}$. $x^2 \equiv m \ (N) \Leftrightarrow x^2 \equiv m \ (p)$, $x^2 \equiv m \ (q)$ (CRT)

$(m, N) = 1 \Rightarrow$ 2 solutions to each of $x^2 \equiv m \ (p)$, $x^2 \equiv m \ (q)$

CRT $\Rightarrow$ 4 solutions to $x^2 \equiv m\ (N)$, giving i).

$p \mid m \Rightarrow$ unique solution to $x^2 \equiv m\ (p)$, giving ii), iii) □

Theorem 16.5

An algorithm to decipher Rabin gives an algorithm to factorise N.

Proof

An algorithm to decipher Rabin must give one of the 4 square roots of a ciphertext $c\ (\text{mod } N)$ which we obtain as $c \equiv x^2\ (N)$. Choose $m \in (\mathbb{Z}/n\mathbb{Z})^*$ at random. The algorithm gives a particular square root x of $c \equiv x^2\ (N)$. So $x^2 \equiv m^2\ (N)$. $\exists$ 4 distinct choices of m giving the same value of c. Two give $x \equiv \pm m$ (
and the others do not.

For the latter two $x^2 - m^2 \equiv (x-m)(x+m) \equiv 0\ (N)$ but neither $x+m, x-m$ are divisible by N. Thus $(N, x-m) \neq 1$. With probability $\frac{1}{2}$ we find a non-trivial factor $(N, x-m)$ of N.

Repeating r times means that the probability of finding a factor of N is $\geq 1 - (\frac{1}{2})^r$ □

RSA Cipher (Rivest-Shamir-Adleman, 1978) / (Cocks, 1973)

i) Choose two large distinct primes p, q. Set $N = pq$.

ii) Pick e randomly with $\gcd(e, \varphi(N)) = 1$ (e.g. e prime, $> p, q$
"encrypting exponent"

iii) $\exists d, k \in \mathbb{Z}$ such that $de - k\varphi(N) = 1$ "decrypting exponent"

iv) Public key (N, e), encrypt with $m \mapsto m^e\ (N)$

v) Private key d, decrypt with $C \mapsto C^d \ (N)$

Euler-Fermat: $(m^e)^d = m^{de} \equiv m^{k\varphi(N)+1} \equiv m \ (N)$

Theorem 16.6

Finding the RSA private key from public key (N, e) is essentially as difficult as factoring N.

Proof

Assume that there is an algorithm give d in terms of N, e.

Then $m^{de} \equiv m \ (N)$, $\forall m \in \mathbb{Z}/n\mathbb{Z}$. Write $de - 1 = 2^s r$, r odd

Let $|x|_p$ = (order of x in $(\mathbb{Z}/p\mathbb{Z})^*$). Let

$X = \{x \in (\mathbb{Z}/n\mathbb{Z})^* \mid |x^r|_p \neq |x^r|_q\}$

Lemma 16.7

If $x \in X$, we can factorise N.

Proof

Let $x \in X$. Let $y = x^r$. This satisfies $y^{2^s} = x^{2^s r} = x^{de-1} \equiv 1 \ (1$

$\therefore |y|_p, |y|_q$ are powers of 2. Suppose $|y|_p = 2^t < |y|_q$.

Then $y^{2^t} \equiv 1 \ (p)$ but $y^{2^t} \not\equiv 1 \ (q)$ so $(y^{2^t} - 1, N) = p$

and we could use Euclid to find one factor of N. To factorise N when we have $x \in X$, compute $(y^{2^t} - 1, N)$ for $0 \leq t \leq s$.

It must be p for one of these choices of t, so we factor N □

Lemma 16.8

$|\{x \in (\mathbb{Z}/p\mathbb{Z})^* : |x|_p = c\}| \leq \frac{1}{2}(p-1)$ for every possible order C.

Proof

Let α be a primitive root mod p. $(\mathbb{Z}/p\mathbb{Z})^* = \langle \alpha \rangle$, $\alpha^{2^s r} \equiv 1 \ (N)$

so certainly $|\alpha^r|_p \mid 2^s$. Let $|\alpha^r|_p = 2^t$, $0 \leq t \leq s$.

If $x = \alpha^k$ then $x^r = (\alpha^r)^k$ $\therefore |x^r|_p = \frac{2^t}{(2^t, k)}$.

Hence $|x^r|_p = 2^t \Leftrightarrow k$ is odd. There are $\frac{1}{2}(p-1)$ such values for $x \in (\mathbb{Z}/p\mathbb{Z})^*$. The remaining $\frac{1}{2}(p-1)$ values for x have $|x|_p < 2^t$.

Hence there are no more than $\frac{1}{2}(p-1)$ elements from $(\mathbb{Z}/p\mathbb{Z})^*$ with $|x^r|_p = c$.

Lemma 16.9

$|X| \geq \frac{1}{2}(p-1)(q-1)$ $\left(= \frac{1}{2}\varphi(N) = \frac{1}{2}\left|(\mathbb{Z}/N\mathbb{Z})^*\right|\right)$

Proof

CRT gives $(\mathbb{Z}/N\mathbb{Z})^* \to (\mathbb{Z}/p\mathbb{Z})^* \oplus (\mathbb{Z}/q\mathbb{Z})^*$, a multiplicative group isomorphism

$x \mapsto (x(p), x(q))$. So $|X| = \left\{(x,y) \in (\mathbb{Z}/p\mathbb{Z})^* \oplus (\mathbb{Z}/q\mathbb{Z})^* : |x^r|_p \neq |y^r|_q\right\}$

For each $y \in (\mathbb{Z}/q\mathbb{Z})^*$ we know $\left|\{x \in (\mathbb{Z}/p\mathbb{Z})^* : |x^r|_p \neq |y^r|_q\}\right| \geq \frac{1}{2}(p-1)$.

Hence the result. □

To complete the proof of Theorem 16.6, choose $x \in (\mathbb{Z}/N\mathbb{Z})^*$ at random. $\mathbb{P}(x \in X) \geq \frac{1}{2}$. If $x \in X$ we use 16.7 to find the factors of N. If $x \notin X$, choose another random value for x. After k such random choices, we will find a factor of N with probability $\geq 1 - (\frac{1}{2})^k$. □

Remark

It is unknown as to whether decrypting RSA messages without knowledge of d is essentially as hard as factoring N.

17 Discrete Logarithm Ciphers

i) Take p, a large prime, and g, a primitive root mod p. These are both fixed and public.

17.1 Diffie-Hellman Key-Exchange (1976)

ii) A, B wish to agree a secret key $k \in \mathbb{Z}/p\mathbb{Z}$ to use. A chooses random $a \in (\mathbb{Z}/(p-1)\mathbb{Z})^*$, computes $g^a = \alpha$, then publishes α.

iii) B chooses random $b \in (\mathbb{Z}/(p-1)\mathbb{Z})^*$, computes $g^b = \beta$, then publishes β.

iv) A knows a and β and computes β^a (p).
B knows b and α and computes α^b (p)

v) $\beta^a = (g^b)^a = (g^a)^b = \alpha^b$ (p) so A, B use the common value k of β^a, α^b, for their key.

vi) E wants to compute $k = g^{ab}$ from p, g, α, β, published data. Diffie-Hellman conjectured that this was equivalent to solving the discrete logarithm problem.

Elgamal Cipher

i) A wishes to send encrypted messages to B.

ii) B chooses random private key $b \in (\mathbb{Z}/(p-1)\mathbb{Z})^*$, and computes $\beta = g^b$ (p), publishing (p, g, β) as public key.

iii) A wants to send $0 < m < p$, so chooses random $a \in (\mathbb{Z}/(p-1)\mathbb{Z})^*$ and sends the pair $(c_0, c_1) = (g^a, \beta^a m) \in \mathbb{Z}/p\mathbb{Z} \oplus \mathbb{Z}/p\mathbb{Z}$

iv) B deciphers by computing $c_1 c_0^{-b}$ (p):

$c_0^b = (g^a)^b = g^{ba} = \beta^a$ (p), so $c_1 c_0^{-b} \equiv m$ (p)

v) If E knows one plaintext m, and the corresponding ciphertext (c_0, c_1) then the public keys $\beta = g^b \ (p)$, $\alpha = g^a \ (p)$ used in Diffie-Hellman. Breaking the Elgamal cipher is tantamount to breaking the Diffie-Hellman Key Exchange, which we hope is as hard as computing discrete logarithms.

18 Trapdoors and Signatures

Guarantees

A sends a message to B. Both are concerned about the following issues:

i) Secrecy : no third-party can read their message

ii) Integrity : no third-party has altered their message

18.1 Example

A is a customer at a bank owned by B. A sends an instruction to pay £100 to C. B must be sure that the amount and the recipient have not been altered. Suppose that the bank creates messages of the form (a, m), where a specifies the recipient, m the amount to pay them.

Suppose that the messages are encrypted using RSA:

$(c_0, c_1) = (a^e, m^e) \pmod N$

E then enters into a transaction with the bank which adds £100 to her account. She observes the resulting encrypted message (c_0', c_1'), then transmits (c_0', c_1^3).

We say that RSA is vulnerable to a homomorphism attack !!

(i.e. uses knowledge of the form of the cipher) (c.f copy attack, same message resent to the bank. Beat using timestamps).

iii) Authenticity : B wants to be sure that A sent the message (we would like some kind of digital signature)

iv) Non-Repudiation : B can prove to C that A sent the message.

Trapdoors

18.2 Terminology

Suppose $h: A^* \to B$. There are infinitely many messages, but B is finite, so $\exists$ many messages with the same h values. It should be hard to find 2 messages with the same h value : a collision. If it is easy to compute $h(m)$ (the digest) for any message m, but hard to find a collision then h is called a signature, trapdoor, or hash function.

Example

i) md5sum, a 128-bit binary sequence computed for every file

ii) RSA, public key (N, e), private key d.

For any letter a, compute $a^e (N)$, then apply a permutation to the binary bits of a^e to give $\sigma(a^e)$. For $m = a_1 \dots a_k$ compute $\sigma_k(a_k^e)$ for a sequence of different permutations σ_k, then add the results mod N. Changes to letters have an unpredictable effect on the digest so it seems hard to find collisions.

18.3 RSA Signatures

A wants to sign a message m to show that she sent it. She uses a public key cipher. Suppose e_A, d_A, are the public and private encryption and decryption functions, so that $e_A d_A = d_A e_A = 1$.

A takes her name n, computes $s = d_A(n)$, and appends s to m.
B can calculate $e_A(s) = e_A(d_A(n)) = n$ to get A's name.
E does not have access to A's (private) d_A, so will find it hard to sign m in this way. But E could separate s from m and attach it to other messages (existential forgery). How do we guarantee integrity?

i) A wants to send m, and adds signature $s = d_A(h(m))$ where h is a (public) trapdoor, i.e. sends (m, s).

ii) B recieves (m, s) and uses the public key to compute $e_A(s)$, then checks $h(m) = e_A(s)$. Only A has the decrypting function d_A, so only she could have signed m with signature such that $e_A(s) = h(m)$.
Any alteration to m would require a corresponding change to $d_A(h(m))$.

18.4 Elgamal Signature Scheme

i) Take a large prime p, and let g be a primitive root mod p.

ii) A chooses $a \in \mathbb{Z}_{p-1}^*$ and publishes $\alpha = g^a \in \mathbb{Z}_p^*$

iii) Let h be a hash function. A sends $1 \leq m \leq p-1$ by choosing random k coprime to $p-1$ and puts $r \equiv g^k (p)$, $s = \frac{h(m) - ar}{k} (p-1)$

iv) A's signature is (r, s). (m, r, s) is sent to B.

v) B recieves (m, r, s) and checks that $g^{h(m)} \equiv \alpha^r r^s (p)$

(because $g^{h(m)} \equiv g^{ar+sk} \equiv (g^a)^r (g^k)^s = \alpha^r r^s (p)$)

It is believed that the only way to forge this signature is to solve the discrete logarithm problem.

Lemma 18.5

In 18.4, one must choose different random k for each message.

Proof:

Suppose that m_1, m_2 are signed using the same k giving (m_1, r, s_1) and (m_2, r, s_2). Then

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

(Recall that $xs \equiv h(m)$ has either no solutions for x or else has (m, s) solutions mod m).

Hence there are $(p-1, s_1 - s_2)$ solutions for k mod $p-1$.

Choose the one that gives the correct value for $r \equiv g^k (p)$.

Then $s_1 \equiv \frac{h(m) - ar}{k} (p-1) \Rightarrow ar = h(m_1) - ks_1 (p-1)$

This gives $(p-1, r)$ solutions for a. Choose the one giving $\alpha = g^a (p)$. We have found A's private key a and the

exponent k that is used for signatures.

19 Bit-Commitment

A wants to send a message to B such that

i) B cannot read the message until Alice sends further information

ii) A cannot change the message.

For example, B tosses a coin, and A calls, both in different rooms so that A cannot see B tossing the coin. How do we ensure that A does not change the call on discovering the result, and that B cannot change the reported result, when A's guess is announced

Commit and Reveal Process

A writes down a guess, and places it in an envelope, and gives it to B. B tosses a coin and reports the result. Together, they open the envelope to see if A's guess was correct.

19.1 Bit Commitment using a Public Key Cipher

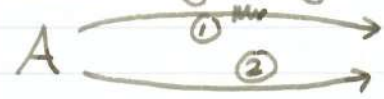
Let e_A, d_A be encrypting and decrypting functions for the public key cipher used by A.

i) e_A is published. d_A is secret, known only to A.

ii) A chooses $m \in \mathbb{F}_2$ and commits to B the encrypted message $c = e_A(m)$ which B cannot decipher.

iii) To reveal the choice, A sends to B the private key so that d_A can be found and $d_A(c) = m$ computed, revealing A's choice. B can check that d_A, e_A are inverse functions, hence can be confident that A has not sent the wrong private key.

19-2 Using Noisy Coding



C_1, Channel 1, a BSC, error probability $p < \frac{1}{2}$

C_2, Channel 2, a clear (error-free) channel.

i) B chooses some linear (binary) code $C \leq \mathbb{F}_2^n$, min. distance δ.

ii) A chooses some random, non-trivial linear map $\varphi : C \to \mathbb{F}_2$

These are both made public.

iii) To send $m \in \mathbb{F}_2$, A chooses a (random) $c \in C$ such that $\varphi(c) = m$, and sends c to B using C_1.

iv) B recieves $r = c + e$ (certain components of c may be altered).

The expected value for $d(r, c) = wt(e) \approx np$ (provided that the variance of the BSC can be made sufficiently small).

v) Later on, A sends C using C_2, and B checks that $d(r, c) \approx np$

If so, then $\varphi(C)$ is accepted as A's choice.

1. Why can B not read the original message?

We can arrange that $\delta << n$. This means that B cannot tell what the original codeword C was, hence can't find $\varphi(C) = m$ and determine A's choice.

2. Why can't A change choice?

A knows that C was sent but does not know r. If later on, A sends $c' \neq C$, it will only be accepted if $d(c', r) \not\approx np$

A's only safe option is to choose c' very close to C, but if δ is sufficiently large, $\delta >> np$, this forces $c = c'$.

20 Quantum Cryptography (Cane, Chapter 22)

Problems with public-key cryptosystems:

i) Based on the belief that some maths problems are "hard". We could be wrong

ii) Computers get faster. Yesterday's securely encrypted message is easily read tomorrow.

We aim to construct a key exchange that is secure conditional on the laws of physics. Consider a single quantum system which describes the state of a single polarised photon. The photon is described by a wave-function Ψ.

Definition 20.1

A *quantum bit* or *qubit* is a linear combination

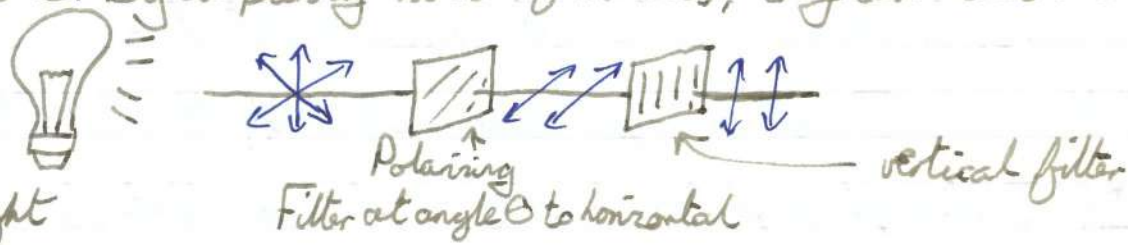
$$\Psi = \underbrace{\alpha |0\rangle}_{\text{vertical polarisation}} + \underbrace{\beta |1\rangle}_{\text{horizontal polarisation}} \qquad \alpha, \beta \in \mathbb{C}$$

$$|\alpha|^2 + |\beta|^2 = 1$$

Two states $|0\rangle$, $|1\rangle$ are a basis for state space.
Denote the basis by $+$. Measuring Ψ gives $|0\rangle$ with probability $|\alpha|^2$ and $|1\rangle$ with probability $|\beta|^2$. After measurement the qubit collapses to the state observed, i.e. either $|0\rangle$ or $|1\rangle$.
There are many other possible bases, for example:

$+\frac{\pi}{4}$ $|+\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$
$-\frac{\pi}{4}$ $|-\rangle = \frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle$ } denote by $\times$.

Basic idea: A generates a sequence of qubits and sends them to B. By comparing notes afterwards, they can detect E's presence



Each photon passing through the first filter then passes through the second with probability $\cos^2\theta$. Endow $\mathbb{C}^2 = \{\alpha|0\rangle + \beta|1\rangle, \alpha,\beta\in\mathbb{C}\}$ with an inner-product $(\alpha_1,\beta_1)\cdot(\alpha_2,\beta_2) = \alpha_1\bar{\alpha}_2 + \beta_1\bar{\beta}_2$
If $\Psi = r|+\rangle + s|-\rangle$ then observation gives $|+\rangle$ with probability $|r|^2$ and $|-\rangle$ with probability $|s|^2$.

20.2 BB84 Protocol for Quantum key Exchange (Bennet, Brassard, 1984)

A and B want to agree a secret key of n binary bits.

Step 1

A sends B a stream of $4n$ qubits with randomly chosen polarisations $|0\rangle$, $|1\rangle$, $|+\rangle$, $|-\rangle$ with probability $\frac{1}{4}$.

Step 2

B measures the qubits using either the +-basis or the x-basis, deciding which at random.

Afterwards, A ~~and B~~ announces which bases ~~they have~~ were used and B announces which bits were measured with the right bases (we expect roughly $2n$ of these).

A, B share $2n$ bits. They compare n of these bits, and if they agree, they use the other n bits as their key.

If E can predict which basis A is using to send, or B is using to measure, she will remain undetected.

Otherwise, E will change about 25% of the $2n$ bits shared.

(If E intercepts a photon $A \leftrightarrow B$, and knows that it is either vertically or horizontally polarised, she can use a vertical filter. If the photon passes through, then it was vertically polarised when sent by A, and E can pass on a vertically polarised photon to B. If not, then it was polarised the other way, and a horizontally polarised photon is sent to B. If A's photon was actually diagonally polarised, then the procedure results in E sending B a photon "wrongly polarised"

Problems and Remarks

i) There is high noise in the system (so bits can be corrupted even when E is not present). One can use error-correcting codes to deal with this. So it is possible to exchange keys with

arbitrarily small probability of error and arbitrarily small probability that an enemy has intercepted it undetected. It is secure because of the laws of quantum mechanics rather than because a maths problem is "hard".

ii) There is no device (yet) for producing single photons (low power lasers are used so that each qubit consists of many photons each with the same polarisation.

iii) "Man in the middle" attacks : If E can intercept all the data sent between A and B, E can impersonate each of them to the other, so that each of A, B will exchange a key with E.

iii) We have sketched a protocol for exchanging keys using the polarisation of photons. Other methods exist such as Ekert's Quantum Entanglement Protocol.