Information and Coding Michaelmas 2006/Michaelmas 2007

O.Johnson/Y. Suhov

Lecture 1: Introduction to entropy

1.1 Course Administration

Unsurprisingly, given its title, the themes of this course are the two related ideas of information and coding. In the first part of the course, we will see the definition of entropy – which gives a rigorous measure of randomness or disorder, and will see results concerning when entropy is maximised, giving an equivalence between the Central Limit Theorem and the Second Law of Thermodynamics. In the second part of the course, we will consider coding – that is, how to protect information from corruption in the presence of noise. We explicitly construct several good codes, including the perfect codes.

Some of those who were at Cambridge last year may notice overlaps between this course and last year's Part IIC Coding and Cryptography course. However, that course is not a pre-requisite for this one, and I will attempt to introduce as much new material as possible to people who feel that they've "been there, done that" last year.

This course is also similar to the old Part IIB Information Theory course.

1.2 Introduction to entropy

The idea of information-theoretic entropy was developed by Shannon. This quantified the idea that 'some random variables are more random than others'. Entropy gives a numerical measure of how far from deterministic a random variable is.

Definition 1.1 *If event A occurs with probability $\mathbb{P}(A)$, define the information $I(A)$ gained by knowing that A has occurred to be*

$$I(A) = -\log_2 \mathbb{P}(A).$$

The intuitive idea is that the rarer an event A, the more information we gain if we know it has occurred. For example, the unlikely event that occurred in 1938 when fishermen caught a coelacanth (a prehistoric fish believed to be extinct) required a significant change to beliefs about evolution and biology. On the other hand, the likely event of catching a herring would hardly imply any change in theories.

Of course, we could use any decreasing function of $\mathbb{P}(A)$ in the definition of information. An axiomatic approach (see Section 4.1) gives an post hoc justification of the choice of $\log_2$, a choice which dates back to Hartley.

Definition 1.2 *Given a discrete random variable X taking values in the finite set $\mathcal{X} = \{x_1, x_2, \ldots, x_n\}$ with probabilities $\mathbf{p} = (p_1, p_2, \ldots, p_n)$, we define the entropy of X to be the expected amount of information gained on learning the value of X:*

$$h(X) = \mathbb{E}I(\{X = x_i\}) = -\sum_{i=1}^{n} p_i \log_2 p_i.$$

It is clear that the entropy $h(X)$ depends on the probability distribution $\mathbf{p}$, but not on the values of X, so we will write interchangably $h(X)$, $h(p_1, \ldots, p_m)$ or even just $h(\mathbf{p})$.

Considerations of continuity lead us to the convention that $0 \log 0 = 0$ (that is, since the limit $\lim_{x \to 0} x \log x = \lim_{t \to \infty} e^{-t}(-t) = 0$, since $e^t \geq t^2/2$ for $t > 0$), so the sum may be reduced to those i for which $p(x_i) > 0$.

Example 1.3 *For X a Bernoulli(p) random variable (that is taking the value 0 with probability $1-p$ and 1 with probability p),*

$$h(X) = -p \log_2 p - (1-p) \log_2 (1-p).$$

Many authors write $h(p)$ for this function, further abusing the notation. We too will use this notation on occasions.

Notice that for $p = 0$ or 1, X is deterministic and $h(X) = 0$. On the other hand, for $p = 1/2$, X is 'as random as it can be' and $h(X) = 1$, the maximum value for random variables taking 2 values, see the graph Figure 1.1. This fits in with our intuition as to how a sensible measure of uncertainty should

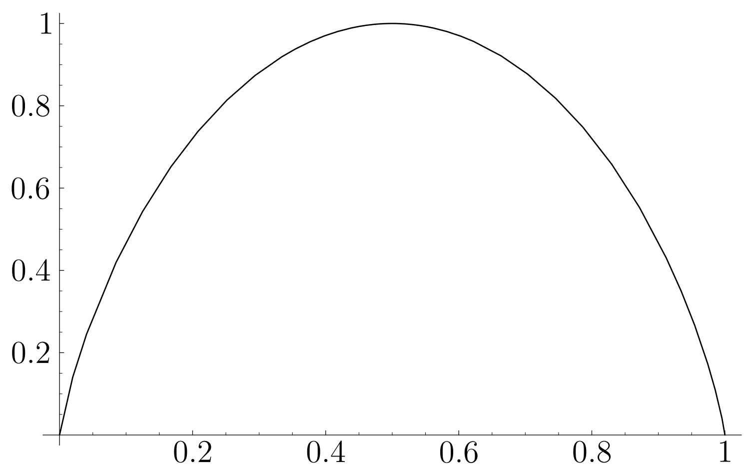

Figure 1.1: The entropy of a Bernoulli(p) variable for different p

behave. Further, $h''(p) = -\log e/(p(1-p)) \leq 0$, that is the function $h(p)$ is concave.

We extend Definition 1.2 to the case of random variables taking countably many values by replacing the sum from 1 to n with a sum from 1 to ∞ (and adopt the convention that if this sum diverges, then the entropy is infinite):

Example 1.4 *For X a random variable with a geometric(p) distribution, that is with $\mathbb{P}(X = r) = (1-p)p^r$ for $r = 0, 1, \ldots$*

$$\begin{aligned} h(X) &= \sum \mathbb{P}(X = r) \left(-\log_2(1-p) - r\log_2 p\right) \\ &= -\log_2(1-p) - \frac{p}{1-p}\log_2 p, \end{aligned}$$

since $\mathbb{E}X = p/(1-p)$.

Shannon entropy is named in analogy with the thermodynamic entropy developed by authors such as Carnot, Boltzmann, Gibbs and Helmholtz (see Section 4.3 for more information). Just as maximum entropy states play a significant role within statistical physics, maximum entropy distributions naturally arise in probability and statistics.

1.3 Basic properties of entropy

Some simple properties of entropy are as follows:

Lemma 1.5

1. *Since the entropy of X does not depend on the values $\{x_i\}$, discrete entropy is both shift and scale invariant, that is for all $a \neq 0$ and for all b:*
$$h(aX + b) = h(X).$$
2. *The entropy of a discrete random variable is always non-negative, and is strictly positive unless X is deterministic.*
3. *The entropy of a random variable taking n values is maximised by the uniform distribution $p_i \equiv 1/n$, so that $h(X) \leq \log n$.*

The first two properties are clear, the last is best proved using the positivity of relative entropy defined in Definition 3.4.

Next, we introduce the concept of the uncertainty contained in a pair of random variables.

Definition 1.6 *Given a pair of random variables (Y, Z), we can define the joint entropy*

$$h(Y, Z) = -\sum_{r,s} \mathbb{P}((Y, Z) = (r, s)) \log \mathbb{P}((Y, Z) = (r, s)).$$

Whilst this is often presented as a new definition, it is in fact a consequence of Definition 1.2, since we can think of (Y, Z) as a single variable X and the pair (r, s) as a single label i (appealing to the idea that the values themselves are irrelevant, and only the probabilities matter).

Notice that if Y and Z are independent then:

$$\begin{aligned} h(Y, Z) &= -\sum_{r,s} \mathbb{P}\big((Y, Z) = (r, s)\big) \log \big(\mathbb{P}(Y = r)\mathbb{P}(Z = s)\big) \\ &= h(Y) + h(Z). \end{aligned}$$

In general $h(Y, Z) \leq h(Y) + h(Z)$ (see Lemma 3.8), with equality if and only if Y and Z are independent.

1.4 Motivation via data compression

We refer to any process which produces a 'random-looking' output as a data source. For example, we could consider the sequence of outcomes of a coin toss or the bits produced by digitised music as the output of data sources. Given a data source producing a string of n outputs, there will often be redundancy in the message. This means that we can summarise the output in a smaller number of symbols.

For example, if we consider 100 outputs of a very biased coin with probability 0.01 of providing heads, instead of listing the whole string "TTTTTTTT ... THT ...", we could summarise the output by saying "12/68", to represent the fact that the heads occurred in those two places. This is the essence of data compression, and suggests an obvious question: given a series of n independent outputs from a random variable, by what factor do we expect to be able to compress the output?

Another question worth considering is this: suppose a data compression program can achieve a compression factor of 50%. Suppose we use it 2,3,4, ... times in succession on the same file – will we achieve a 50% compression each time?

In data compression we try to encode independent realisations of the random variable X in the most efficient way possible, representing the outcome x_i by a binary string or 'word' y_i, of length r_i. Since we will store a concatenated series of words $y_{i_1} \vee y_{i_2} \vee \ldots$, the codewords need to be decipherable (that is, that the concatenated series of codewords can be uniquely parsed back to the original codewords). One way to achieve this is to insist that the code be 'prefix-free', that is, there do not exist codewords y_i and y_j such that $y_i = y_j \vee z$ for some z.

This can be equivalently phrased in terms of trees – when written as a binary tree, there is never more than one word on any branch (see Figure 1.2).

By counting how many possible words are made impossible by each choice of a codeword, it turns out that this constraint implies the Kraft inequality:

Lemma 1.7 *Given positive integers $r_1, \ldots, r_n$, there exists a decipherable*

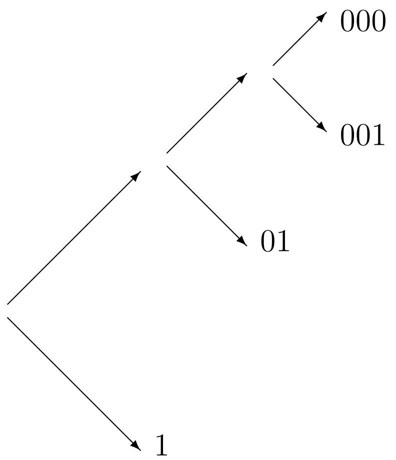

Figure 1.2: Example of a collection of decipherable codewords

binary code with codeword lengths $r_1, \ldots, r_n$ *if and only if*

$$\sum_{i=1}^{n} 2^{-r_i} \leq 1.$$

Since we want to minimise the expected codeword length we naturally consider an optimisation problem:

$$\begin{array}{rl} \text{minimise:} & \sum_{i=1}^{n} r_i p_i \\ \text{such that:} & \sum_{i=1}^{n} 2^{-r_i} \leq 1 \\ \text{and:} & \text{all the } r_i \in \mathbb{Z}. \end{array}$$

Standard techniques of Lagrangian optimisation indicate that, relaxing the requirement that r_i should be an integer, the optimal choice would be $r_i = -\log p_i$ and the average word length required would be $\sum p_i r_i = -\sum p_i \log p_i$. (There is another argument to show this, but it requires the relative entropy, which is not introduced until Definition 3.4).

We can always get close to this optimal bound, by picking $r_i = \lceil -\log p_i \rceil$, so that the minimum expected codeword length lies between $h(X)$ and $h(X)+1$.

By an argument based on coding a block of length n from the random variable, we can in fact code arbitrarily close to the entropy. That is, for blocks of length n, the space required is within 1 bit per block of the entropy, so when normalised, this becomes $1/n$ bits per symbol of the entropy.

This idea of coding a source using a binary alphabet means that the natural units for entropy are 'bits', and this explains why we use the non-standard logarithm to base 2. Throughout this course, the symbol log will refer to logarithms taken to base 2, and $\log_e$ will represent the natural logarithm. Similarly, we will sometimes use the notation $\exp_2(x)$ to denote 2^x, and $\exp(x)$ for e^x.

Lecture 2: Shannon's First coding theorem

2.1 Small sets

To be able to data compress efficiently, we would like to find small sets that occur with high probability. Instead of just working in binary, think about symbols coming from a general set I (known as the alphabet), and write I^n for $I \times \ldots (n \text{ times}) \ldots \times I$.

Definition 2.1 *For a given $R > 0$, a source generating strings $U^{(n)}$ is said to be reliably encodable at rate R if, for any n you can find a set $A_n \in I^n$ such that*

$$\sharp A_n \leq 2^{nR} \textit{ and } \lim_{n \to \infty} \mathbb{P}(U^{(n)} \in A_n) = 1.$$

The idea here is that we can label the members of the set A_n with a label of length nR, and so since we are 'nearly always' in A_n, the average length of a compressed string will be close to nR.

Definition 2.2 *The information rate H of a given source is the smallest reliable encoding rate*

$$H = \inf\, [R : R \textit{ is reliable}].$$

Theorem 2.3 *For a source with alphabet of size m,*

$$0 \leq H \leq \log m,$$

both bounds being attainable.

Proof The LH inequality trivially follows from Definition 2.2. On the other hand, for any A_n, the size $\sharp A_n \leq \sharp I^n = m^n = 2^{n \log m}$, hence the RH inequality.

The LH equality is attained for a source where always $x_1 = x_2 = x_3 \ldots$. Then A_n contains at most m strings, which is eventually less than 2^{nR} for any $R > 0$.

The RH equality is attained for a source with uniformly, independent and identically distributed x_i: in this case for any set of strings $\mathbb{P}(A_n) = (1/m^n)\sharp A_n$, which goes to zero when $\sharp A_n \leq 2^{nR}$ and $R < \log m$. $\square$

Although the quantity H seems useful, it also seems hard to write down. We will prove results about it by first introducing a quantity D_n which tells us about H, and then a quantity ξ_n which tells us about D_n. Then, ξ_n is easier to find, and from it we deduce the value of H.

Definition 2.4 *Define the subset maximum by*

$$D_n(R) := \max_{A:\sharp A \leq 2^{nR}} \mathbb{P}(U^{(n)} \in A). \tag{2.1}$$

Lemma 2.5 *For any $\epsilon > 0$, the information rate H satisfies*

$$\lim_{n\to\infty} D_n(H+\epsilon) = 1, \quad \textit{and, if } H > 0, \quad D_n(H-\epsilon) \not\to 1.$$

Proof By definition, $H + \epsilon$ is a reliable encoding rate. There must therefore exist a sequence of sets $A_n \subset I^n$, with $\sharp A_n \leq 2^{nR}$ and $\mathbb{P}(U^{(n)} \in A_n) \to 1$, as $n \to \infty$. Since $D_n(R) \geq \mathbb{P}(U^{(n)} \in A_n)$, then $D_n(R) \to 1$.

Alternatively, for any $H > 0$, we can pick ϵ such that $H - \epsilon$ is > 0, but not reliable. Then choosing C_n to be the sets for which the maximum in (2.1) is attained, the $\sharp C_n \leq 2^{nR}$, but $\mathbb{P}(C_n) \not\to 1$. $\square$

Given a string $u^{(n)} = u_1 \ldots u_n$, consider its log–likelihood per source letter:

$$\xi_n(u^{(n)}) = -\frac{1}{n} \log_+ p_n(u^{(n)}),$$

where $p_n(u^{(n)})$ is the probability assigned to string $u^{(n)}$. Here, and below, $\log_+ x = \log x$, if $x > 0$, and $= 0$, if $x = 0$. For a random string $U^{(n)}$,

$$\xi_n(U^{(n)}) = -\frac{1}{n} \log_+ p_n(U^{(n)}) \tag{2.2}$$

is a random variable.

Lemma 2.6 *For any R, $\epsilon > 0$,*

$$\mathbb{P}(\xi_n \le R) \le D_n(R) \le \mathbb{P}(\xi_n \le R + \epsilon) + 2^{-n\epsilon}. \tag{2.3}$$

Proof For simplicity, we leave out the superscript (n). Define

$$\begin{aligned} B_n &= \{u \in I^n : p_n(u) \ge 2^{-nR}\} = \{u \in I^n : -\log p_n(u) \le nR\} \\ &= \{u \in I^n : \xi_n(u) \le R\}. \end{aligned}$$

Then

$$1 \ge \mathbb{P}(U \in B_n) = \sum_{u \in B_n} p_n(u) \ge 2^{-nR} (\sharp B_n),$$

and so $\sharp B_n \le 2^{nR}$. Thus,

$$D_n(R) = \max_{\sharp A \le 2^{nR}} \mathbb{P}(U \in A) \ge \mathbb{P}(U \in B_n) = \mathbb{P}(\xi_n \le R),$$

which proves the LH in (2.3).

On the other hand, there exists $C_n \subseteq I^n$ where the maximum in (2.1) is attained. For such a C_n,

$$\begin{aligned} D_n(R) &= \mathbb{P}(U \in C_n) \\ &= \mathbb{P}(U \in C_n, \xi_n \le R + \epsilon) + \mathbb{P}(U \in C_n, \xi_n > R + \epsilon) \\ &\le \mathbb{P}(\xi_n \le R + \epsilon) + \sum_{u \in C_n} \mathbb{I}\left(p_n(u) < 2^{-n(R+\epsilon)}\right) p_n(u) \\ &\le \mathbb{P}(\xi_n \le R + \epsilon) + 2^{-n(R+\epsilon)} \sharp C_n \\ &\le \mathbb{P}(\xi_n \le R + \epsilon) + 2^{-n(R+\epsilon)} 2^{nR} = \mathbb{P}(\xi_n \le R + \epsilon) + 2^{-n\epsilon}, \end{aligned}$$

as required. □

2.2 Shannon's first coding theorem

Definition 2.7 *We say that a sequence of random variables $\{\eta_n\}$ converges in probability to a constant r if for any $\epsilon > 0$,*

$$\lim_{n \to \infty} \mathbb{P}\left(\left|\eta_n - r\right| \ge \epsilon\right) = 0.$$

Convergence in probability is denoted by $\eta_n \xrightarrow{\mathbb{P}} r$.

Convergence in probability is established in the law of large numbers (see Theorem 2.11 below).

Theorem 2.8 (Shannon's First Coding Theorem) *If ξ_n converges in probability to a constant γ then $\gamma = H$, the information rate of a source.*

Proof Let $\xi_n \xrightarrow{\mathbb{P}} \gamma$. Since $\xi_n \geq 0$, its limit $\gamma \geq 0$. By Lemma 2.6, for any $\epsilon > 0$,

$$\begin{aligned} D_n(\gamma+\epsilon) &\geq \mathbb{P}(\xi_n \leq \gamma+\epsilon) \geq \mathbb{P}(\gamma-\epsilon \leq \xi_n \leq \gamma+\epsilon) \\ &= \mathbb{P}\left(|\xi_n - \gamma| \leq \epsilon\right) = 1 - \mathbb{P}\left(|\xi_n-\gamma| > \epsilon\right), \end{aligned}$$

which converges to 1 as $n \to \infty$. We deduce that $H \leq \gamma$.

In particular, if $\gamma = 0$ then $H = 0$. If $\gamma > 0$, by the opposite bound in Lemma 2.6

$$\begin{aligned} D_n(\gamma-\epsilon) &\leq \mathbb{P}(\xi_n \leq \gamma - \epsilon/2) + 2^{-n\epsilon/2} \\ &\leq \mathbb{P}\left(|\xi_n-\gamma| \geq \epsilon/2\right) + 2^{-n\epsilon/2}, \end{aligned}$$

which tends to $2^{-n\epsilon/2}$, so by Lemma 2.5, $H \geq \gamma$ and the result follows. □

Remark 2.9 *Convergence $\xi_n \xrightarrow{\mathbb{P}} \gamma = H$ is equivalent to the following asymptotic equipartition property (AEP): for any $\epsilon > 0$,*

$$\lim_{n\to\infty} \mathbb{P}\left(2^{-n(H+\epsilon)} \leq p_n(U^{(n)}) \leq 2^{-n(H-\epsilon)}\right) = 1.$$

In fact,

$$\begin{aligned} &\mathbb{P}\left(2^{-n(H+\epsilon)} \leq p_n(U^{(n)}) \leq 2^{-n(H-\epsilon)}\right) \\ &\quad= \mathbb{P}\left(H - \epsilon \leq -\frac{1}{n}\log p_n(U^{(n)}) \leq H+\epsilon\right) \\ &\quad= \mathbb{P}\left(|\xi_n - H| \leq \epsilon\right) = 1 - \mathbb{P}\left(|\xi_n - H| > \epsilon\right). \end{aligned}$$

In other words, for any $\epsilon > 0$ there exists $n_0(\epsilon)$ such that, for any $n > n_0(\epsilon)$, we can identify a typical set T_n with the properties that

(a) $\mathbb{P}\left(U^{(n)} \notin T_n\right) < \epsilon$

(b) for any $u^{(n)} \in T_n$, the $2^{-n(H+\epsilon)} \leq \mathbb{P}\left(U^{(n)} = u^{(n)}\right) \leq 2^{-n(H-\epsilon)}$

For a source with the AEP, encode the typical strings with codewords of length $n(H+\epsilon)$, and the rest however you like. You will then have an effective encoding rate $H + o(1)$ bits/source-letter.

2.3 Information rate of a Bernoulli source

Theorem 2.10 *For a Bernoulli source, the information rate equals the entropy, that is*

$$H = h.$$

Proof For an IID sequence U_1, $U_2, \ldots$,

$$p_n(u^{(n)}) = \prod_{i=1}^{n} p(u_i).$$

Hence,

$$-\log p_n(u) = \sum_i -\log p(u_i).$$

For a random string $U^{(n)} = (U_1, \ldots, U_n)$,

$$\xi_n = -\log p_n(U^{(n)}) = -\sum_{i=1}^{n} \log p(U_i) = \sum_{i=1}^{n} \sigma_i,$$

where random variables $\sigma_i = -\log p(U_i)$ are independent and have the same distribution (we refer to this as IID).

Observe that $\mathbb{E}\sigma_i = -\sum_j p(j) \log p(j) = h$ and so $\mathbb{E}\xi_n = \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}\sigma_i = h$. Hence $\xi_n \xrightarrow{\mathbb{P}} h$ is a direct corollary of Theorem 2.11 below. □

Theorem 2.11 (Weak Law of Large Numbers) *Consider any sequence of IID random variables $X_1, X_2, \ldots$ with finite mean $\mathbb{E}X_i = \mu$ and finite variance $\mathbb{E}(X_i - \mu)^2 = \sigma^2$. Then for any $\epsilon > 0$,*

$$\lim_{n\to\infty} \mathbb{P}\left(\left| \frac{1}{n} \sum_{i=1}^{n} X_i - \mu \right| \geq \epsilon \right) = 0. \tag{2.4}$$

Proof By Chebyshev, for any random variable η and any $\epsilon > 0$,

$$\mathbb{P}(\eta \geq \epsilon) \leq \frac{1}{\epsilon^2} \mathbb{E}\eta^2.$$

Applying this to the LHS of (2.4) gives

$$\mathbb{P}\left(\left|\frac{1}{n}\sum_{i=1}^{n} X_i - \mu\right| \geq \epsilon\right) \leq \frac{1}{\epsilon^2}\mathbb{E}\left(\frac{1}{n}\sum_{i=1}^{n} X_i - \mu\right)^2 = \frac{1}{\epsilon^2 n^2}\mathbb{E}\left(\sum_{i=1}^{n}(X_i - \mu)\right)^2. \tag{2.5}$$

Since the sum $\sum_{i=1}^{n}(X_i - \mu)$ has mean zero, the RHS of (2.5) is nothing but

$$\frac{1}{\epsilon^2 n^2}\mathrm{Var}\left(\sum_{i=1}^{n}(X_i - \mu)\right) = \frac{1}{\epsilon^2 n^2}\sum_{i=1}^{n}\mathrm{Var}(X_i - \mu) = \frac{1}{n\epsilon^2}\mathrm{Var}X_1,$$

which goes to 0 when $n \to \infty$. Theorem 2.11 is proved, and so is Theorem 2.10. $\square$

Note that we refer to this as the weak law, because of the (relatively) weak sense in which convergence is proved, and also because of the strong assumption of finite variance.

Stronger laws are possible: for example, Theorem 12.10 of *Probability with Martingales* by David Williams shows that if $\mathbb{E}|X_i| < \infty$, then for $X_1, X_2, \ldots$ IID:

$$\frac{\sum_{i=1}^{n} X_i}{n} \to (\mathbb{E}X_i) \text{ almost surely.}$$

("Almost surely" means "with probability 1").

However, for our purposes, the weak law will be enough.

Information and Coding Michaelmas 2006/Michaelmas 2007
O.Johnson/Y. Suhov

Lecture 3: More entropy-like quantities

3.1 Differential entropy

We can edit Definition 1.2 to give a definition of entropy for continuous random variables Y with density p. One obvious idea is to rewrite Equation (1.2) by replacing the sum by an integral and the probabilities by densities. In fact this turns out to be exactly the right thing to do. A quantisation argument that justifies this is as follows.

Consider a random variable Y with a continuous density p. We can produce a discrete version Y^δ. We split the real line into intervals $I_t = (t\delta, (t+1)\delta)$, and let

$$\mathbb{P}(Y^\delta = t) = \int_{I_t} p(y)dy = \delta p(y_t), \text{ for some } y_t \in I_t.$$

(The existence of such a y_t follows by the mean value theorem, since the density f is continuous.) Then

$$\begin{aligned} h(Y^\delta) &= -\sum_t \mathbb{P}(Y^\delta = t) \log \mathbb{P}(Y^\delta = t) \\ &= -\sum_t \delta p(y_t) \log p(y_t) - \log \delta, \end{aligned}$$

so by Riemann integrability, $\lim_{\delta \to 0} \left(h(Y^\delta) + \log \delta \right) = -\int p(y) \log p(y) dy$. We take this limit to be the definition of differential entropy.

Definition 3.1 *The differential entropy of a continuous random variable Y with density p is:*

$$h(Y) = h(p) = -\int p(y) \log p(y) dy.$$

Again, $0 \log 0$ is taken as 0. For a random variable Y with no density, the $h(Y) = \infty$.

We know that to encode a real number to arbitrary precision will require an infinite number of bits. However, using similar arguments as before, if we wish to represent the outcome of a continuous random variable Y to an accuracy of n binary places, we will require an average of $h(Y) + n$ bits.

Although we use the same symbol, h, to refer to both discrete and differential entropy, it will be clear from context which type of random variable we are considering.

Example 3.2

1. *If Y has a uniform distribution on $[0, c]$:*
$$h(Y) = -\int_0^c \frac{1}{c} \log\left(\frac{1}{c}\right) dx = \log c.$$

2. *If Y has a normal distribution with mean 0 and variance σ^2:*
$$h(Y) = \int_{-\infty}^{\infty} \phi(x) \left(\frac{\log(2\pi\sigma^2)}{2} + \frac{x^2 \log e}{2\sigma^2}\right) dx = \frac{\log(2\pi e\sigma^2)}{2}.$$

3. *If Y has an n-dimensional multivariate normal distribution with mean 0 and covariance matrix C:*
$$\begin{aligned} h(Y) &= \int \phi(\mathbf{x}) \left(\frac{\log((2\pi)^n \det C)}{2} + \frac{(\mathbf{x}^T C^{-1}\mathbf{x}) \log e}{2}\right) d\mathbf{x} \\ &= \frac{\log((2\pi)^n \det C)}{2} + \frac{n \log e}{2} = \frac{\log((2\pi e)^n \det C)}{2}. \end{aligned}$$

These examples indicate that differential entropy does not retain all the useful properties of the discrete entropy.

Lemma 3.3

1. *Differential entropies can be both positive and negative. In fact (see Exercise 3 for an example) $h(Y)$ can even be minus infinity.*

2. *Although the entropy again only depends on the densities and not the values, since the density itself is not scale invariant, for all $a \neq 0$ and for all b:*
$$h(aY + b) = h(Y) + \log a$$
(so shift, but not scale invariance holds).

Proof

1. See Example 3.2(1).

2. Follows from the scaling of the density. The distribution function of $aY+b$ satisfies $F_{aY+b}(y) = \mathbb{P}(aY+b \leq y) = \mathbb{P}(Y \leq (y-b)/a) = F_Y((y-b)/a)$. This means that the density satisfies: $f_{aY+b}(y) = f_Y((y-b)/a)/a$. Hence:

$$\begin{aligned} h(aY+b) &= -\int f_{aY+b}(y) \log f_{aY+b}(y) dy \\ &= -\int \frac{1}{a} f_Y\left(\frac{y-b}{a}\right)\left(\log f_Y\left(\frac{y-b}{a}\right) - \log a\right) dy \\ &= -\int f_Y(z)\left(\log f_Y(z) - \log a\right) dz, \end{aligned}$$

taking $z = (y-b)/a$, so $dy/a = dz$. The result follows.

□

3.2 Relative entropy

We can recover scale invariance by introducing the idea of relative entropy:

Definition 3.4 *For two discrete random variables taking values in* $\{x_1, \ldots x_n\}$ *with probabilities* $\mathbf{p} = (p_1, \ldots p_n)$ *and* $\mathbf{q} = (q_1, \ldots q_n)$ *respectively, we define the relative entropy distance from* $\mathbf{p}$ *to* $\mathbf{q}$ *to be:*

$$D(\mathbf{p}\|\mathbf{q}) = \sum_i p_i \log\left(\frac{p_i}{q_i}\right).$$

In the case of continuous random variables with densities p *and* q*, define the relative entropy distance from* p *to* q *to be:*

$$D(p\|q) = \int p(x) \log\left(\frac{p(x)}{q(x)}\right) dx.$$

For discrete random variables, if for some x, the $q(x) = 0$ but $p(x) \neq 0$, then we must take $D(p\|q) = \infty$. In the same way, for continuous random variables, if there exists an interval I of positive length such that for $x \in I$, the $q(x) = 0$ but $p(x) \neq 0$, then again we take $D(p\|q) = \infty$.

For those who know what the support of a random variable is (it is roughly the set on which the measure is positive, though there are some technicalities to do with measurability), we can say that if the support $\mathrm{supp}(p) \not\subseteq \mathrm{supp}(q)$, then $D(p\|q) = \infty$.

Relative entropy goes by a variety of other names, including Kullback-Leibler distance, information divergence and information discrimination.

Lemma 3.5

1. *Although we refer to it as a distance, note that D is not a metric: it is not symmetric and does not obey the triangle rule.*
2. *However, D is positive semi-definite: for any probability densities p and q, $D(p\|q) \geq 0$ with equality if and only if $p = q$ almost everywhere.*

Proof The key is the fact that there exists a constant $c > 0$ such that $\log(y) \geq c(1 - 1/y)$. Defining $B = \mathrm{supp}(p) = \{x : p(x) > 0\}$:

$$\begin{aligned} D(p\|q) &= \int p(x) \log\left(\frac{p(x)}{q(x)}\right) \mathbb{I}(x \in B) dx \\ &\geq c \int p(x) \left(1 - \frac{q(x)}{p(x)}\right) \mathbb{I}(x \in B) dx = c(1 - q(B)) \geq 0, \end{aligned}$$

with equality if and only if $q(x) = p(x)$ almost everywhere. □

The asymmetry in the definition of D represents the asymmetry between accepting and rejecting a hypothesis. For example, D appears as the limit in Stein's lemma:

Lemma 3.6 *Take $X_1 \ldots X_n$ IID with distribution Q. Consider the hypotheses $H_0 : Q = Q_0$ and $H_1 : Q = Q_1$, and critical region A_n with Type I and II error probabilities: $\alpha_n = \mathbb{P}_{Q_0^{\times n}}(A_n^c)$ and $\beta_n = \mathbb{P}_{Q_1^{\times n}}(A_n)$. Define:*

$$\beta_n^\epsilon = \inf_{\alpha_n < \epsilon} \beta_n.$$

Then

$$\lim_{\epsilon \to 0} \lim_{n \to \infty} \frac{1}{n} \log \beta_n^\epsilon = -D(Q_0 \| Q_1).$$

That is, the larger the distance D, the smaller the β that can be achieved for a given α, so the easier it is to tell the distributions apart.

This Lemma helps to explain why the relative entropy D also occurs as an exponent in large deviation theory, in Cramér's theorem and Sanov's theorem. For example

Lemma 3.7 *For $X_1, \ldots X_n$ a collection of independent identically distributed Bernoulli(q), for any $1 > p > q > 0$:*

$$-\lim_{n \to \infty} \frac{1}{n} \log \mathbb{P}\left(\frac{\sum_{i=1}^n X_i}{n} \geq p \right) = (1-p) \log \left(\frac{1-p}{1-q} \right) + p \log \left(\frac{p}{q} \right),$$

the relative entropy distance from a Bernoulli(p) to a Bernoulli(q).

Another motivation for the relative entropy, in the discrete case at least, is to suppose we code optimally in the belief that the distribution is $q(x)$, and hence use codewords of length $\lceil -\log q(x) \rceil$. If in fact the true distribution is $p(x)$, our coding will no longer be optimal. The average number of extra bits required, compared with the optimal coding, is about $D(p\|q)$.

3.3 Other entropy-like quantities

The positivity of the relative entropy offers the easiest proof of many properties of entropy. For example the final part of Lemma 1.5 can be proved simply by considering p, our probability distribution taking n values, and q, uniform on these same values. Then $0 \leq D(p\|q) = -h(p) + \sum_i p_i \log n$, so the result follows. Similarly:

Lemma 3.8 *For any random variables X and Y:*

$$h(X, Y) \leq h(X) + h(Y),$$

with equality if and only if X and Y are independent.

Proof Consider the distance from the joint distribution to the product of the marginals:

$$\begin{aligned} 0 &\le D(p(x,y)\|p(x)p(y)) \\ &= \sum_{x,y} p(x,y)\left(\log p(x,y) - \log p(x) - \log p(y)\right) \\ &= -h(X,Y) + h(X) + h(Y) \end{aligned}$$

as required. □

This quantity, $h(X) + h(Y) - h(X,Y)$, occurs commonly enough that it is given its own name: mutual information $i(X,Y)$. It represents how easy it is to make inference about random variable X from a knowledge of random variable Y. Interestingly, the symmetry present means that it also measures how easy it is to make inference about Y from a knowledge of random variable X.

Definition 3.9 *For random variables X and Y, define:*

1. *mutual information $i(X,Y) = h(X) + h(Y) - h(X,Y)$*
2. *conditional entropy $h(Y|X) = h(X,Y) - h(X)$.*

These different entropies are illustrated schematically in Figure 3.1. In particular, notice that for any random variables X, Y:

$$i(X,Y) = h(Y) - h(Y|X) = h(X) - h(X|Y). \tag{3.1}$$

In fact, this schematic diagram suggests an interesting relationship, known as the Hu correspondence, which indicates that our understanding of entropy as 'the amount of information gained' really carries through. Specifically, the correspondence works by a 1-1 matching between entropies and the size μ of sets. That is

$$\begin{aligned} h(X) &\longleftrightarrow \mu(A) \\ i(X,Y) &\longleftrightarrow \mu(A \cap B) \\ h(X,Y) &\longleftrightarrow \mu(A \cup B) \\ h(X|Y) &\longleftrightarrow \mu(A \setminus B) \end{aligned}$$

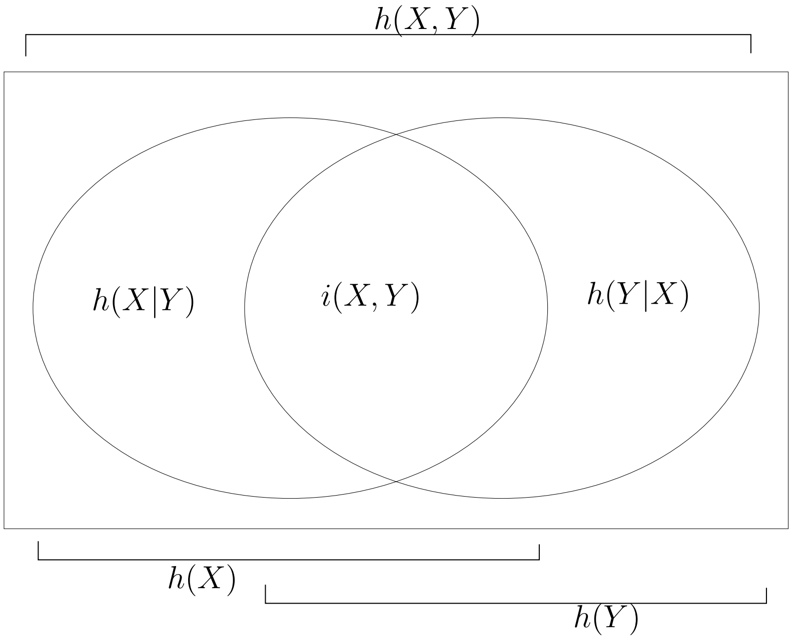

Figure 3.1: Different types of entropies

Then Hu shows that any linear relationship in these h and i is true, if and only if the corresponding relationship in terms of set sizes also holds. Thus for example, the statement of Lemma 3.8 is clear since

$$h(X,Y) \leq h(X) + h(Y) \longleftrightarrow \mu(A \cup B) \leq \mu(A) + \mu(B).$$

In fact the situation is slightly more complicated for intersections of 3 or more sets. It turns out that μ can be negative, so we have to think of it as a signed measure.

Conditional entropy gains its name since:

$$\begin{aligned}
h(X,Y) - h(X) &= \sum_{x,y} -p(x,y) \log p(x,y) + \sum_{x} p(x) \log p(x) \\
&= \sum_{x,y} -p(x,y) \log \left(\frac{p(x,y)}{p(x)} \right) \\
&= \sum_{x} p(x) \sum_{y} -p(y|x) \log p(y|x) \\
&= \sum_{x} p(x) h(Y|X=x) \qquad (3.2)
\end{aligned}$$

(here treating $Y|X = x$ as a random variable, for each x). The conditional entropy allows a neat proof of the fact that entropy increases on convolution.

Lemma 3.10 *For X and Y independent random variables:*

$$h(X+Y) \geq \max(h(X), h(Y)).$$

Proof Since only the probabilities matter

$$h(X+Y|X) = h(Y|X) = h(Y),$$

so we know that

$$h(Y) = h(X+Y|X) \leq h(X+Y),$$

since entropy decreases on conditioning. This is a direct consequence of Lemma 3.8, since $h(U|V) = h(U,V) - h(V) \leq h(U)$. $\square$

Lecture 4: Axiomatic definitions, maximum entropy and the Second Law

4.1 Axiomatic definition of entropy

Some authors have tried to motivate the choice of h as the 'one true entropy' by showing that it is the only measure of uncertainty to possess certain plausible properties. It is reasonable to ask whether Definition 1.2 represents the only possible definition of entropy. Rényi introduced axioms on how we would expect a measure of information, or 'randomness' to behave, and then discusses which functions satisfy them.

Rényi's axioms are:

Definition 4.1 *For any collection of positive numbers $\mathcal{P} = \{p_1, \ldots p_k\}$, such that $0 < \sum_r p_r \leq 1$:*

1. *Symmetry: $h(\mathcal{P})$ is symmetric in indices of p.*
2. *Continuity: $h(\{p\})$ is continuous in p, for $0 < p \leq 1$.*
3. *Standardisation: $h(\{1/2\}) = 1$.*
4. *Independence: for X, Y independent $h(X, Y) = h(X) + h(Y)$.*
5. *Decomposition:*

$$h(p_1, \ldots p_m, q_1, \ldots q_n) = g^{-1}\Big(\alpha h(p_1 \ldots p_m) + (1-\alpha) h(q_1 \ldots q_n)\Big),$$

   where $\alpha = \sum p_i / (\sum p_i + \sum q_i)$ and $g(x)$ is some fixed function.

He shows that if $g(x) = x$, the only possible function h satisfying these properties is a generalised version of the discrete entropy of Definition 1.2:

$$h(p_1, \ldots, p_k) = \frac{-\sum_i p_i \log p_i}{\sum_j p_j}.$$

Further, if $g(x) = 2^{(\alpha-1)x}$, then the only possible h is the so-called Rényi α-entropy defined by:

Definition 4.2 *Given a probability distribution* $\mathbf{p} = (p_1, \dots p_k)$ *define the Rényi* α*-entropy to be:*

$$h_\alpha(\mathbf{p}) = \frac{1}{1-\alpha} \log \left(\sum_k p_k^\alpha \right).$$

Note that by L'Hôpital's rule, since $\frac{d}{dx} a^x = a^x \log_e a$,

$$\lim_{\alpha \to 1} h_\alpha(\mathbf{p}) = (\log e) \lim_{\alpha \to 1} \frac{-\sum_k p_k^\alpha \log_e p_k}{\sum_k p_k^\alpha} = \frac{-\sum_k p_k \log p_k}{\sum_k p_k} = h(\mathbf{p}).$$

4.2 Maximum entropy distributions

Using relative entropy, we can see that various well-known distributions can be seen to maximise entropy under very natural constaints. In this lecture, we will describe a variety of such results, and then go on to see parallels with the Second Law of Thermodynamics. As the name suggests, the relative entropy D generalises entropy, by considering random variables with respect to a different reference measure. For example, in Definition 3.4, if $q_i = 1/n$ then

$$\begin{aligned} D(p\|q) &= \sum p(x) \log p(x) + \sum p(x) \log n \\ &= \log n - h(p), \end{aligned}$$

so (up to a linear transformation), $h(p)$ corresponds to taking the relative entropy distance from p to a uniform distribution. This idea leads to a variational principle.

Lemma 4.3 *Given a probability distribution* p *taking a finite set of values* $\{x_1, \dots x_n\}$*, writing* q *for the uniform distribution on this set:*

$$h(p) \leq h(q),$$

with equality iff p *is uniformly distributed.*

Proof Using the positivity of the relative entropy, taking $q(x) = 1/n$:

$$\begin{aligned}
0 &\leq D(p\|q) \\
&= \sum p(x) \log p(x) + \sum p(x) \log n \\
&= \log n - h(p) \\
&= h(q) - h(p),
\end{aligned}$$

as required. Equality holds if and only if $D(p\|q) = 0$, which holds precisely when $p(x) = q(x)$ for all x, that is when p is uniform. $\square$

Other variational principles provides more maximum entropy results.

Lemma 4.4 *If p is the density of a random variable with mean μ and variance σ^2, and ϕ_{μ,σ^2} is the density of a $N(\mu, \sigma^2)$ random variable then:*

$$h(p) \leq h(\phi_{\mu,\sigma^2}) = \frac{\log(2\pi e \sigma^2)}{2},$$

with equality if and only if p is a Gaussian density.

Proof By shift invariance we may assume p has mean 0. Then, because $\log \phi_{0,\sigma^2}(x)$ is a quadratic function in x:

$$\begin{aligned}
0 &\leq D(p\|\phi_{0,\sigma^2}) = \int p(x) \left(\log p(x) + \frac{\log(2\pi\sigma^2)}{2} + \frac{x^2}{2\sigma^2} \log e \right) dx \\
&= -h(p) + \frac{\log(2\pi e \sigma^2)}{2} = -h(p) + h(\phi_{0,\sigma^2}).
\end{aligned}$$

$\square$

This property gave the initial motivation to consider the Central Limit Theorem, in which the Gaussian distribution plays a special role, in terms of Shannon's entropy. See Section 5.3 below.

4.3 Definition of thermodynamic entropy

Information-theoretic entropy is named by analogy with the better-known thermodynamic entropy. Apparently it was von Neumann who suggested to Shannon that he should use the same name:

> You should call it 'entropy' and for two reasons; first, the function is already in use in thermodynamics under that name; second, and more importantly, most people don't know what entropy really is, and if you use the word 'entropy' in an argument you will win every time.

Statistical physics contrasts macrostates (properties of large numbers of particles, such as temperature and pressure) and microstates (properties of individual molecules, such as position and momentum). The link comes as follows:

Definition 4.5 *Suppose there are Ω microstates corresponding to a particular macrostate. Then the thermodynamic entropy of the macrostate is*

$$S = k \log_e \Omega.$$

Here k is Boltzmann's constant (since we are only concerned with changes, we can ignore constants).

Suppose each microstate r can occur with probability p_r. Consider a very large collection of v replicas of the same system (we call this an ensemble), then on average there will be v_r replicas in the rth microstate, where v_r is the closest integer to vp_r. In this case, by Stirling's formula

$$\Omega = \frac{v!}{v_1! v_2! \dots v_k!} \simeq \frac{v^v}{v_1^{v_1} v_2^{v_2} \dots v_k^{v_k}}.$$

Hence, since $\sum v_r = v$, the thermodynamic entropy of the ensemble will be:

$$\begin{aligned} S_v &= k \log_e \Omega \simeq k \left(v \log_e v - \sum v_r \log_e v_r \right) \\ &= k \left(-\sum v_r (\log_e v_r - \log v) \right) \\ &= k \left(-\sum v_r \log_e (v_r / v) \right) \\ &= -kv \sum p_r \log_e p_r. \end{aligned}$$

Since the entropy of a compound system is the sum of the parts of the system, $S_v = vS$, so that

$$S = S_v / v = -k \sum p_r \log_e p_r,$$

which, up to a constant, is the formula from Definition 1.2 for information-theoretic entropy.

4.4 Maximum entropy and the Second Law

Lagrangian methods show that the entropy S is maximised subject to an energy constraint by the so-called Gibbs states.

Lemma 4.6 *The maximum of*

$$-\sum p_r \log p_r \text{ such that } \sum p_r = 1, \sum p_r E_r = E$$

comes at $p_i = \exp(-\beta E_i)/Z_\beta$*, for some* β *determined by* E *and where the partition function* $Z_\beta = \sum_i \exp(-\beta E_i)$*.*

The Second Law of Thermodynamics states that the thermodynamic entropy always increases with time, implying some kind of convergence to the Gibbs state. Conservation of energy means that E remains constant during this time evolution, so we can tell from the start which Gibbs state will be the limit.

In the next lecture we will show that the Central Limit Theorem can be considered in the same way. That is, as we take convolutions it can be shown that the information-theoretic entropy increases to its maximum. Normalising appropriately means that the variance remains constant at σ^2, and we know from Lemma 4.4 that for fixed variance the maximum entropy is achieved by the normal distribution.

Note, however, that in this course we only use the Second Law as an analogy, not least because of its controversial status. Whilst it might sound surprising to refer to such a famous result in such a way, there remains ongoing argument about it. In different places, the Second Law can be treated as something so obvious as to not require discussion, or as something that might not even be true! For example, Eddington writes:

> if your theory is found to be against the second law of thermodynamics, I can give you no hope; there is nothing for it but to collapse in deepest humiliation.

In contrast, Truesdell complains that

> Seven times in the past thirty years have I tried to follow the argument Clausius offers [...] and seven times has it blanked and gravelled me [...]. I cannot explain what I cannot understand.

From our point of view, this will not be a problem; we will never require a rigorous statement of the Second Law. However, it is striking that the Central Limit Theorem (probably the best-known result in probability and statistics) can be seen as similar to the Second Law of Thermodynamics (one of the best-known results of physics).

Information and Coding Michaelmas 2006/Michaelmas 2007 O.Johnson/Y. Suhov

Lecture 5: Inequalities and the Central Limit Theorem

5.1 Entropy Power Inequality

Recall that Lemma 3.10 showed that entropy increases on convolution. In fact, better bounds exist on the entropy of a sum, in the form of the Entropy Power Inequality. Shannon stated the Entropy Power inequality and sketched a "proof". New proofs were offered by Blachman and later by Dembo, Cover and Thomas. Recall that $\exp_2(x)$ denotes 2^x.

Theorem 5.1 *For* $\mathbf{X}$ *and* $\mathbf{Y}$ *independent* n*-dimensional random vectors:*

$$\exp_2(2h(\mathbf{X}+\mathbf{Y})/n) \geq \exp_2(2h(\mathbf{X})/n) + \exp_2(2h(\mathbf{Y})/n),$$

with equality if and only if $\mathbf{X}$ *and* $\mathbf{Y}$ *are normally distributed with proportional covariance matrices.*

The proof of the Entropy Power inequality given by Dembo, Cover and Thomas is based on two observations. Firstly, Shannon entropy is a limiting case of the Rényi α-entropy of Definition 4.2.

Secondly, Beckner shows the exact values of the constants in the Hausdorff-Young inequality:

Lemma 5.2 *Writing* $\|f\|_p = \left(\int |f(\mathbf{x})|^p d\mathbf{x}\right)^{1/p}$ *for the* p*-norm of* f*, if* $1/r + 1 = 1/p + 1/q$ *and* $1 \leq p, q, r$ *then:*

$$\sup_{f,g} \frac{\|f \star g\|_r}{\|f\|_p \|g\|_q} = \left(\frac{c_p c_q}{c_r}\right)^{n/2}, \tag{5.1}$$

where $c_p = |p|^{1/p}/|p'|^{1/p'}$*, for* p' *satisfying* $1/p + 1/p' = 1$*, and the supremum is achieved by certain normal densities.*

Proof of Theorem 5.1 Armed with this, Dembo, Cover and Thomas complete the proof, since $h_p(f) = -p' \log \|f\|_p$, so taking the right limiting regime for p, q, the Entropy Power inequality follows.

We write $V_p(f) = \exp_2(2h_p(f)/n) = \exp_2(-(2p'/n) \log \|f\|_p)$, since by continuity $V_1(f) = \exp_2(2h(f)/n)$.

Taking the logarithm of the Hausdorff-Young inequality (Lemma 5.2), and multiplying through by r', we obtain:

$$\log V_r(f \star g) \geq \frac{r'}{p'} \log V_p(f) + \frac{r'}{q'} \log V_q(g) + r' \log c_r - r' \log c_p - r' \log c_q. \quad (5.2)$$

Notice that $1/r + 1 = 1/p + 1/q$ implies that $1/r' = 1/p' + 1/q'$. This means that if we take $\lambda = r'/p'$, then $1 - \lambda = r'/q'$. Further

$$p = r/(r + \lambda(1-r)) \text{ and } q = r/(r + (1-\lambda)(1-r)). \quad (5.3)$$

Further, notice that $r/r' = r(1 - 1/r) = (r-1)$. Putting this all together, the RHS of Equation (5.2) becomes $\lambda \log V_p(f) + (1-\lambda) \log V_q(g)$ plus

$$\begin{aligned}
&\frac{r'}{r} \log r - \log r' - \frac{r'}{p} \log p + \frac{r'}{p'} \log p' - \frac{r'}{q} \log q + \frac{r'}{q'} \log q' \\
= \; & \frac{1}{r-1} \log r - (\lambda + (1-\lambda)) \log r' - \frac{r'}{p} \log p + \lambda \log p' \\
& - \frac{r'}{q} \log q + (1-\lambda) \log q' \\
= \; & \frac{1}{r-1} \log r - \frac{r'}{p} \log p + \lambda \log(p'/r') - \frac{r'}{q} \log q + (1-\lambda) \log(q'/r') \\
= \; & \frac{1}{r-1} \log r + h(\lambda) - \frac{r + \lambda(1-r)}{r-1} \log \left(\frac{r}{r + \lambda(1-r)} \right) \\
& - \frac{r + (1-\lambda)(1-r)}{r-1} \log \left(\frac{r}{r + (1-\lambda)(1-r)} \right), \qquad (5.4)
\end{aligned}$$

where the last line follows using Equation (5.3).

Since this is true for all λ, we fix a particular value of λ and let $r \to 1$. Using L'Hopital, $\log r/(r-1) \to \log e$.
Similarly $(r + \lambda(1-r))/(r-1) \log (r/(r + \lambda(1-r))) \to \lambda \log e$
and $(r + (1-\lambda)(1-r))/(r-1) \log (r/(r + (1-\lambda)(1-r))) \to (1-\lambda) \log e$.

Hence, consider the limit for fixed λ as $r \to 1$, since Equation (5.3) gives that $p, q \to 1$. We deduce from Equation (5.4) that for any λ:

$$\log V_1(f \star g) \geq \lambda \log V_1(f) + (1-\lambda) \log V_1(g) + h(\lambda).$$

We now exploit our free choice of λ, by taking $\lambda = V_1(f)/(V_1(f) + V_1(g))$, to obtain that

$$V_1(f \star g) \geq V_1(f) + V_1(g),$$

the Entropy Power Inequality. $\square$

5.2 Extensions and applications of the EPI

Note that using other limiting regimes, the Entropy Power inequality can be seen to be one of a family of results that include the Brunn-Minkowski inequality.

Theorem 5.3 (Brunn-Minkowski) *For any n-dimensional sets A, B, writing μ for Lebesgue measure, and writing $A + B = \{x + y : x \in A, y \in B\}$ for the set sum of A and B:*

$$\mu(A+B)^{1/n} \geq \mu(A)^{1/n} + \mu(B)^{1/n}.$$

[note that for A, B spheres around the origin of radius r_A and r_B, $A + B$ will be a sphere around the origin of radius $r_A + r_B$, and this inequality will be sharp].

Proof This can be proved in precisely the same way, using a different limiting regime. The only change required is to note that for $0 < p, q, r < 1$, the inequality is reversed in Equation (5.1). We use the fact that considerations of continuity show that $V_0(f) = \mu(\{x : f(x) > 0\})^{2/n}$. Hence, consider any densities f and g.

Again using Equation (5.3) gives that for fixed λ, as $r \to 0$, so do $p, q \to 0$.

Once more using L'Hopital, we deduce that Equation (5.4) tends to $2h(\lambda)$. This means that we can deduce that

$$\begin{aligned}
\log V_0(f \star g) &\geq \lambda \log\left(\frac{V_0(f)}{\lambda^2}\right) + (1-\lambda)\log\left(\frac{V_0(g)}{(1-\lambda)^2}\right) \\
&= (\lambda + (1-\lambda)) \log((\sqrt{V_0(f)} + \sqrt{V_0(g)})^2)
\end{aligned}$$

by taking $\lambda = \sqrt{V_0(f)}/(\sqrt{V_0(f)} + \sqrt{V_0(g)})$. In other words,

$$\sqrt{V_0(f+g)} \geq \sqrt{V_0(f)} + \sqrt{V_0(g)},$$

and choosing f and g such that $A = \{x : f(x) > 0\}$ and $B = \{y : g(y) > 0\}$, the result follows. □

Now, the fact that (see Example 3.2) a multivariate normal with covariance matrix C has entropy $\log((2\pi e)^n(\det C))/2$ leads to simple proofs of results concerning determinants.

Theorem 5.4 *For symmetric, positive definite matrices C, C_1, C_2*

1. **Ky Fan**: *$\log\det C$ is a concave function of C*
2. **Minkowski**: $(\det(C_1 + C_2))^{1/n} \geq (\det(C_1))^{1/n} + (\det(C_2))^{1/n}$
3. **Hadamard**: *$\det C \leq \prod_i C_{ii}$, with equality iff C is diagonal*

Proof In the first two parts, consider X_1 and X_2 independent normals with mean zero and covariance C_1 and C_2 respectively.

1. To prove the Ky Fan theorem, choose random variable T independent of X_1, X_2 and taking values $1, 2$ with probability p, $1-p$. Further, define $Z = X_T$, with covariance $pC_1 + (1-p)C_2$. Then, since the entropy of Z is less than that of a normal with the same covariance:

$$\begin{aligned}
&\frac{1}{2}\log((2\pi e)^n \det(pC_1 + (1-p)C_2) \\
&\quad \geq h(Z) \\
&\quad \geq h(Z|T) \\
&\quad = \frac{p}{2}\log((2\pi e)^n \det C_1) + \frac{1-p}{2}\log((2\pi e)^n \det C_2)
\end{aligned}$$

and the result follows.

2. To prove Minkowski's result, use the Entropy Power Inequality:

$$\begin{aligned}
(2\pi e)(\det(C_1 + C_2))^{1/n} &= \exp_2(2h(X_1 + X_2)/n) \\
&\geq \exp_2(2h(X_1)/n) + \exp_2(2h(X_2)/n) \\
&= (2\pi e)(\det C_1)^{1/n} + (2\pi e)(\det C_2)^{1/n}
\end{aligned}$$

3. Finally, to prove the Hadamard Theorem, consider $\mathbf{X} = (X_1, \ldots X_n)$ a multivariate normal $N(0, C)$. Simply note that

$$\frac{1}{2} \log((2\pi e)^{1/n} \det C) = h(\mathbf{X}) \leq \sum_{i=1}^{n} h(X_i) = \sum_{i=1}^{n} \frac{1}{2} \log((2\pi e)^{1/n} C_{ii}),$$

with equality iff X_i are independent. $\square$

5.3 Entropy and the Central Limit Theorem

Notice that the Entropy Power Inequality Theorem 5.1 combines with the scaling result produced in Lemma 3.3 as follows. For X_1, X_2 independent and identically distributed, the one-dimensional version of the EPI gives that

$$\exp_2(2h(X_1 + X_2)) \geq 2 \exp_2(2h(X_1)),$$

so that

$$h\left(\frac{X_1 + X_2}{\sqrt{2}}\right) = h(X_1 + X_2) - \frac{1}{2} \log 2 \geq h(X_1).$$

Thus in general,

Lemma 5.5 *Taking $Y_1, Y_2, \ldots$ IID with variance σ^2, writing $S_n = (Y_1 + \ldots Y_n)/\sqrt{n}$, doubling the sample size increases the entropy:*

$$h(S_n) \leq h(S_{2n}).$$

Further, by Lemma 4.4, we know that the entropy of S_n is bounded above by that of a normal Z_{σ^2} with the same variance. Therefore in the Central Limit Theorem regime, along the 'powers-of-2 sequence', $h(S_{2^k})$ is increasing and bounded above by $h(Z_{\sigma^2})$. This means that

Lemma 5.6 *The sequence $h(S_{2^k})$ converges to a limit h.*

It is a natural question to try to identify the limit h, and to wonder if the whole sequence $h(S_n)$ will converge to the same limit. Of course, although we hope that $h = h(Z_{\sigma^2})$, nothing we have done so far shows that this will be the case.

In fact the result was proved by Barron in 1986, a result referred to as the information-theoretic Central Limit Theorem:

Theorem 5.7 *Taking $Y_1, Y_2, \ldots$ IID with variance σ^2, writing $S_n = (Y_1 + \ldots Y_n)/\sqrt{n}$, if for some m, $h(S_m) > -\infty$, then*

$$\lim_{n\to\infty} h(S_n) = h(Z_{\sigma^2}).$$

Equivalently, taking Z_{μ,σ^2} to be a normal random variable with the same mean and variance as Y_i, then if $D(S_m \| Z_{\mu,\sigma^2})$ is finite for some m then

$$\lim_{n\to\infty} D(S_n \| Z_{\mu,\sigma^2}) = 0.$$

Information and Coding Michaelmas 2006/Michaelmas 2007 O.Johnson/Y. Suhov

Lecture 6: Transmission in channels

6.1 Introduction to channels

In this part of the course, we discuss problems of information transmission. That is, we consider whether we can reliably send information through a noisy communication channel. A channel can be defined in a very abstract way to be "anything which changes things".

The process can be divided into several stages. For example, consider Alice talking to Bob on their mobile phones.

1. Consider a signal generated by a source (so for example, Alice's voice will be converted by the microphone into a series of 0s and 1s).

2. We use an encoder (any function f, random or deterministic) to convert these 0s and 1s into another series of 0s and 1s. The key will be to make a clever choice of function, to somehow "pad out" the information in the signal, and protect it from errors.

3. Next, the output of the function is sent to Bob's mobile phone, and errors are introduced in transmission. (For example, the mobile phone signal will be sent via masts and base stations to another mobile phone. Errors can be introduced by a variety of processes, but from our point of view, that won't matter - we won't differentiate between errors introduced by cosmic rays, software errors at the base station, pigeons flying into the mast and so on). This is what we refer to as the communication channel.

4. Bob's phone now attempts to decode the signal. The decoder will, roughly speaking, attempt to apply the inverse of the encoding function f, to recover something close to the original message. However, we will need a decoding function that is robust to the introduction of these errors.

Many other processes can be regarded in the same way, and the model of a channel is in fact very abstract.

When we use a channel we hope to:

1. correct as many errors as possible (deal with as noisy a channel as possible),

2. make effective use of the channel available (maximum transfer of information per unit time),

3. quickly encode and decode information (a mobile phone conversation must take place in real time).

A source emits a random text $U_1, U_2, \ldots$, where $U_i \in I$, and we encode a message $\mathbf{u}^{(n)}$ by a codeword $\mathbf{x}^{(N)}$, by using a code $f_n : I^n \to J^N$, (often $J = \{0, 1\}$) known to the receiver. The key is to understand the relationship between n and N.

A channel is understood as any physical process subject to 'noise' which distorts the messages transmitted: a message at the output differs in general from the message at the input. Formally, a channel is characterized by a conditional distribution

$$\mathbb{P}_{\text{channel}}\left(\text{receive word } \mathbf{y}^{(N)} \middle| \text{codeword } \mathbf{x}^{(N)} \text{ sent}\right), \tag{6.1}$$

which we again suppose is known to both sender and receiver. These conditional probabilities define a channel. We will often deal with the simplest case

Definition 6.1 *We say that a channel is memoryless if*

$$\mathbb{P}_{\text{channel}}\left(\mathbf{y}^{(N)} \middle| \mathbf{x}^{(N)}\right) = \prod_{i=1}^{N} P(y_i|x_i),$$

if $\mathbf{y}^{(N)} = y_1 \ldots y_N$*,* $\mathbf{x}^{(N)} = x_1 \ldots x_N$*. Here,* $P(y|x)$ *is a symbol-to-symbol channel probability (the conditional probability to have symbol* y *at the output of the channel given that symbol* x *has been sent). We think of* P *as a stochastic matrix, often called the channel matrix.*

If the rows of the channel matrix P are permutations of each other, we say that the channel is symmetric.

In particular, we often consider the memoryless binary symmetric channel, where $x, y \in \{0, 1\}$, so that $\{P(y|x)\}$ is a 2×2 transition probability matrix of the form

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

and p is called the row error probability (or the symbol-error probability).

6.2 Transmission rates and capacity

We want to introduce a decoding rule $\widehat{f}_N : J^N \to I^n$ so that the overall probability of error

$$\begin{aligned} \epsilon &= \sum_{u^{(n)}} \mathbb{P}\left(\widehat{f}(\mathbf{y}^{(N)}) \neq \mathbf{u}^{(n)}, \mathbf{u}^{(n)} \text{ emitted}\right) \\ &= \sum_{\mathbf{u}^{(n)}} \mathbb{P}_{\text{source}}(\mathbf{U}^{(n)} = \mathbf{u}^{(n)})\mathbb{P}_{\text{channel}}\left(\widehat{f}_N(\mathbf{y}^{(N)}) \neq \mathbf{u}^{(n)} | f_n(\mathbf{u}^{(n)}) \text{ sent}\right) \qquad (6.2) \end{aligned}$$

is small. In fact, we will try to have quantity (6.2) tending to zero as $n \to \infty$.

The construction is based on two facts:

1. For a source with the AEP the number of distinct n-strings emitted is $2^{n(H+o(1))}$ where $H \leq \log m$ is the information rate of the source. That is, we don't need to encode $m^n = 2^{n \log m}$ messages, but rather only $2^{n(H+o(1))}$ which may be considerably less. That implies a codeword length of $\lceil nH \rceil$.

2. We increase the length of the codewords used from $\lceil nH \rceil$ to $\lceil \overline{R}^{-1} nH \rceil$, for some constant $\overline{R}$. That is, we introduce redundancy in code f_n, hoping to reduce the overall error probability (6.2). We want to have the factor of increase $\overline{R}$ as small as possible, that is $\overline{R}$ as large as possible.

As the codeword-length is a crucial parameter, write N instead of $\overline{R}^{-1} Hn$ and $\overline{R}N$ instead of Hn. It is convenient to consider a 'typical' set $\mathcal{U}_N$ of distinct strings emitted by the source, with $\sharp\, \mathcal{U}_N = 2^{N(\overline{R}+o(1))}$.

Definition 6.2 *Value $\overline{R} \in (0,1)$ is a reliable transmission rate (for a given channel) if, given that the source strings take equiprobable values from a set $\mathcal{U}_N$ of size $2^{N(\overline{R}+o(1))}$, there exists an encoding rule $f_N : \mathcal{U}_N \to \mathcal{X}_N \subseteq \{0,1\}^N$ and a decoding rule $\widehat{f}_N : \{0,1\}^N \to \mathcal{U}_N$ with error probability*

$$\sum_{\mathbf{u}^{(N)} \in \mathcal{U}_N} \frac{1}{\sharp\, \mathcal{U}_N} \mathbb{P}_{\text{channel}} \left(\widehat{f}_N(\mathbf{y}^{(N)}) \neq \mathbf{u}^{(N)} \middle| f_N(\mathbf{u}^{(N)}) \text{ sent} \right)$$

tending to zero as $N \to \infty$.

Equivalently, that is, we look for

$$\overline{R} \sim (\log \sharp\, \mathcal{U}_N)/N. \tag{6.3}$$

Definition 6.3 *The channel capacity is defined as the supremum*

$$C = \sup\, [\overline{R} : \text{ is a reliable transmission rate }].$$

The reason for considering the equiprobable distribution on $\mathcal{U}_N$ in Definition 6.2 is that it is the worst case. The argument uses the idea of randomising, which proves to be very powerful later in the chapter.

Theorem 6.4 *Fix a channel, by choosing $\mathbb{P}_{\text{channel}}$ in (6.1). Fix a set $\mathcal{U}$ of source strings and denote by $\epsilon(\mathbb{P})$ the overall error probability (6.2) for the source with probability distribution $\mathbb{P}$ on $\mathcal{U}$, minimized over all encoding and decoding rules. Then*

$$\epsilon(\mathbb{P}) \leq \epsilon(\mathbb{P}^{\text{eq}}), \tag{6.4}$$

where $\mathbb{P}^{\text{eq}}$ is equidistribution over $\mathcal{U}$.

Proof Consider particular encoding and decoding rules, f and $\widehat{f}$ and let a string $\mathbf{u} \in \mathcal{U}$ have probability $\mathbb{P}(\mathbf{u})$. Define the error-probability when $\mathbf{u}$ is emitted to be

$$\beta(\mathbf{u}) = \sum_{\mathbf{y} : \widehat{f}(\mathbf{y}) \neq \mathbf{u}} \mathbb{P}_{\text{channel}}(\mathbf{y} | f(\mathbf{u})).$$

The overall error probability therefore equals $\epsilon(\mathbb{P}, f, \widehat{f}) = \sum_{\mathbf{u} \in \mathcal{U}} \mathbb{P}(\mathbf{u}) \beta(\mathbf{u})$.

Now, consider a permutation π of the codewords (that is, instead encode $\mathbf{u}$ by $f(\mathbf{v})$ where $\mathbf{v} = \pi(\mathbf{u})$). The new overall error probability $\epsilon(\pi) =$

$\sum_{\mathbf{u}\in\mathcal{U}} \mathbb{P}(\mathbf{u})\beta(\pi(\mathbf{u}))$. In the case of $\mathbb{P} = \mathbb{P}^{\rm eq}$ (equidistribution) $\epsilon(\pi)$ does not depend on π and equals $\bar{\epsilon} = \frac{1}{\sharp\mathcal{U}} \sum_{\mathbf{u}\in\mathcal{U}} \beta(\mathbf{u}) = \epsilon(\mathbb{P}^{\rm eq}, f, \widehat{f})$.

For each probability distribution $\mathbb{P}$ there exists π such that $\epsilon(\pi) \leq \bar{\epsilon}$. In fact, take a random permutation, Π, equidistributed among all permutations of codewords. Then

$$\begin{aligned} \min_{\pi} \epsilon(\pi) &\leq \mathbb{E}\epsilon(\Pi) = \mathbb{E}\sum_{\mathbf{u}\in\mathcal{U}} \mathbb{P}(\mathbf{u})\beta(\Pi\mathbf{u}) \\ &= \sum_{\mathbf{u}\in\mathcal{U}} \mathbb{P}(\mathbf{u})\mathbb{E}\beta(\Pi\mathbf{u}) = \sum_{\mathbf{u}\in\mathcal{U}} \mathbb{P}(\mathbf{u})\frac{1}{\sharp\,\mathcal{U}}\sum_{\mathbf{v}\in\mathcal{U}} \beta(\mathbf{v}) = \bar{\epsilon}. \end{aligned}$$

Hence, given any f and $\widehat{f}$, you can find new encoding and decoding rules with overall error probability $\leq \epsilon(\mathbb{P}^{\rm eq}, f, \widehat{f})$. Minimizing over f and $\widehat{f}$ leads to (6.4). $\square$

6.3 Simple decoding rules

It is now time to discuss possible decoding rules. As was noted before, a decoding rule (or a decoder) is a map $\widehat{f}_N\colon \{0,1\}^N \to \mathcal{U}_N$.

Equivalently, a decoding rule is given by fixing, for each codeword $\mathbf{x}_i^{(N)}$, a set $A(\mathbf{x}_i^{(N)}) \subset \{0,1\}^N$, so that $A(\mathbf{x}_i^{(N)})$ and $A(\mathbf{x}_j^{(N)})$ are disjoint for distinct codewords $\mathbf{x}_i^{(N)}$ and $\mathbf{x}_j^{(N)}$, and the union $\bigcup_i A(\mathbf{x}_i^{(N)})$ gives the whole $\{0,1\}^N$. Given that $\mathbf{y}^{(N)} \in A(\mathbf{x}^{(N)})$, we decode by $\widehat{f}_N(\mathbf{y}^{(N)}) = \mathbf{x}^{(N)}$. See Figure 6.1 for an schematic illustration of this.

Although in the definition of the channel capacity we assume that the source messages are equidistributed (which gives the worst case) of course the source does not always follow this assumption. We may need to distinguish between two situations: (i) the receiver knows the probability distribution $\mathbb{P}_{\rm source}$ and hence the probability distribution $p_N(\mathbf{x}^{(N)})$ of the codewords (ii) the receiver does not know $p_N(\mathbf{x}^{(N)})$.

Definition 6.5 *Two natural decoding rules are:*

1. *The ideal observer rule. Decode a received word* $\mathbf{y}^{(N)}$ *by a codeword*

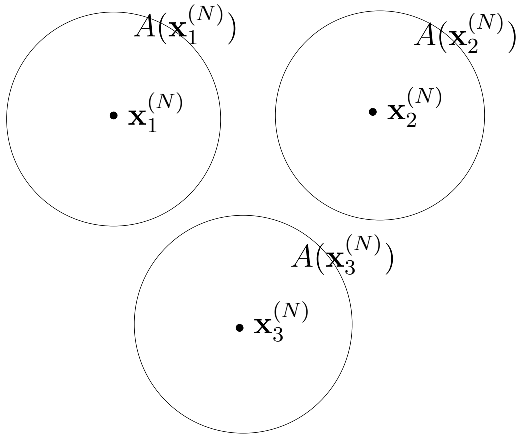

Figure 6.1: Decoding rule described in terms of sets around each codeword

$\mathbf{x}^{(N)\star}$ *that maximizes the posterior probability*

$$\mathbb{P}\left(\mathbf{x}^{(N)} \text{ sent } \middle| \mathbf{y}^{(N)} \text{ received}\right) = \frac{p_N(\mathbf{x}^{(N)})\mathbb{P}_{\text{channel}}(\mathbf{y}^{(N)}\big|\mathbf{x}^{(N)})}{p_{\mathbf{Y}^{(N)}}(\mathbf{y}^{(N)})}, \quad (6.5)$$

where

$$p_{\mathbf{Y}^{(N)}}(\mathbf{y}^{(N)}) = \sum_{\mathbf{u}^{(N)} \in \mathcal{X}_N} p_N(\mathbf{u}^{(N)})\mathbb{P}_{\text{channel}}(\mathbf{y}^{(N)}\big|\mathbf{u}^{(N)}).$$

2. *The maximum likelihood rule: Decode a received word* $\mathbf{y}^{(N)}$ *by codeword* $\mathbf{x}^{(N)\star}$ *that maximizes the prior probability*

$$\mathbb{P}_{\text{channel}}(\mathbf{y}^{(N)}\big|\mathbf{x}^{(N)}).$$

Theorem 6.6 *Suppose that an encoding rule f is defined for all messages that occur with positive probability and is one-to-one.*

1. *For any encoding rule, the ideal observer decoder minimizes the overall error probability among all decoders.*

2. *If the source message* $\mathbf{U}$ *is equiprobable on a set* $\mathcal{U}$*, and encoding rule* f *is* $\mathcal{U} \to \mathcal{X}$ *then codeword* $\mathbf{X} = f(\mathbf{U})$ *is equiprobable on* $\mathcal{X}$*, and the ideal observer and maximum likelihood decoders coincide.*

Proof 1. Note that, given a received word $\mathbf{y}$, the denominator in (6.5) is fixed, so the ideal observer rule maximizes the joint probability $p(\mathbf{x})\mathbb{P}_{\text{channel}}(\mathbf{y}|\mathbf{x})$ Suppose you use an encoding rule f and decoding rule $\widehat{f}$. Then the overall error probability (see (6.2)) is

$$\begin{aligned}
&\sum_{\mathbf{u}} \mathbb{P}_{\text{source}}(\mathbf{U}=\mathbf{u})\mathbb{P}_{\text{channel}}\left(\widehat{f}(\mathbf{y}) \neq \mathbf{u} \middle| f(\mathbf{u}) \text{ sent}\right) \\
&= \sum_{\mathbf{x}} p(\mathbf{x}) \sum_{\mathbf{y}:\widehat{f}(\mathbf{y})\neq\mathbf{x}} \mathbb{P}_{\text{channel}}(\mathbf{y}|\mathbf{x}) \\
&= \sum_{\mathbf{y}} \sum_{\mathbf{x}:\mathbf{x}\neq\widehat{f}(\mathbf{y})} p(\mathbf{x})\mathbb{P}_{\text{channel}}(\mathbf{y}|\mathbf{x}) \\
&= \sum_{\mathbf{y}}\sum_{\mathbf{x}} p(\mathbf{x})\mathbb{P}_{\text{channel}}(\mathbf{y}|\mathbf{x}) - \sum_{\mathbf{y}} p\left(\widehat{f}(\mathbf{y})\right)\mathbb{P}_{\text{channel}}\left(\mathbf{y}\middle|\widehat{f}(\mathbf{y})\right) \\
&= 1 - \sum_{\mathbf{y}} p\left(\widehat{f}(\mathbf{y})\right)\mathbb{P}_{\text{channel}}\left(\mathbf{y}\middle|\widehat{f}(\mathbf{y})\right)
\end{aligned}$$

Now each term in the sum $\sum_{\mathbf{y}} p\left(\widehat{f}(\mathbf{y})\right)\mathbb{P}_{\text{channel}}\left(\mathbf{y}\middle|\widehat{f}(\mathbf{y})\right)$ is maximized when $\widehat{f}$ coincides with the ideal observer rule. Hence, the whole sum is maximized, and the overall error probability minimized.

2. The probabilities being constant means that they can be cancelled, and the result is clear. □

As we assume, in the definition of the channel capacity, that the source messages are equidistributed, it is natural to use the maximum likelihood decoder. We are going to do so in the rest of the course.

Information and Coding Michaelmas 2006/Michaelmas 2007 O.Johnson/Y. Suhov

Lecture 7: Shannon's Second Coding Theorem

7.1 Result and motivation

In this and the next lecture, we will prove Shannon's Second Coding Theorem (in the case of a memoryless channel). It turns out that a key quantity in the theorem is the mutual information i of Definition 3.9.

The result is as follows:

Theorem 7.1 *For a memoryless channel, the capacity*

$$C = \sup_{p_X} i(X, Y).$$

Proof The proof is divided into two parts, Theorem 7.6 and 8.6, establishing an upper and lower bound respectively for C. □

First of all, we motivate matters by considering the so-called "noisy typewriter". That is:

Example 7.2 *Pick integers k and l. Consider a channel which has the input and output symbols $\{0, 1, 2, \dots kl-1\}$. The channel takes symbol i to $i, i+1, \dots, i+(k-1)$ with equal probability.*

Now (see Section 8.1 for more examples of this kind of calculation), for any input distribution p_X:

$$i(X,Y) = h(Y) - h(Y|X) = h(Y) - \sum_x \mathbb{P}(X=x) h(Y|X=x) = h(Y) - \log k.$$

(The first equality follows by Equation (3.1), the second by Equation (3.2)). The largest that $h(Y)$ could be is $\log(kl)$ – one way that we can achieve this is to take X equidistributed on all the possible inputs, so then Y is equidistributed on all the possible outputs. This means that $\sup_{p_X} i(X,Y) = \log l$.

Now, we can achieve error probability of zero, with this amount of redundancy introduced. That is, if we only use input symbols $\{0, k, 2k, \ldots (l-1)k\}$, *then the outputs of the channel will be distinct for distinct symbols. That is, we can send (and guarantee to decode)* l *messages with each symbol. Hence, in Equation (6.3), this corresponds to* $\sharp\, \mathcal{U}_1 = l$ *(and indeed* $\sharp\, \mathcal{U}_N = l^N$*), so that*

$$\overline{R} = (\log \sharp\, \mathcal{U}_N)/N = \log l$$

is a reliable transmission rate.

This argument doesn't really prove anything yet, but we can motivate our proof by saying that the property of joint typicality introduced in Definition 8.2 equates to saying that the source and channel is roughly like the noisy typewriter.

7.2 Fano inequalities and upper bound

We introduce two results concerning entropy; the first shows that if X is nearly constant, then the entropy of X is very small, the second shows that if X is nearly a function of Y, then the entropy of $X|Y$ is very small. (We know that for each statement that if the 'nearly' is replaced by an 'exactly' then each entropy is zero – this is a stronger result).

Theorem 7.3

1. **(Fano inequality)** *Suppose a random variable* X *takes* $m \geq 2$ *values, and one of them has probability* $(1-\epsilon)$. *Then*

$$h(X) \leq h(\epsilon) + \epsilon \log(m-1).$$

2. **(Generalized Fano inequality)** *For a pair of random variables,* X *and* Y, *if we can rearrange the values so that they pair up with* $x_1, \ldots, x_m$ *and* $y_1, \ldots, y_m$

$$\sum_{j=1}^{m} p(X = x_j, Y = y_j) = 1 - \epsilon,$$

then

$$h(X|Y) \leq h(\epsilon) + \epsilon \log(m-1).$$

In each case, $h(\epsilon)$ is the entropy function,

$$h(\epsilon) = -\epsilon \log \epsilon - (1-\epsilon)\log(1-\epsilon). \tag{7.1}$$

Proof We prove Theorem 7.3.1 first. Without loss of generality, $p_1 = 1-\epsilon$. Then

$$\begin{aligned}
h(X) &= -\sum_{i=1}^{m} p_i \log p_i \\
&= -p_1 \log p_1 - (1-p_1)\log(1-p_1) + (1-p_1)\log(1-p_1) - \sum_{i=2}^{m} p_i \log p_i \\
&= -p_1 \log p_1 - (1-p_1)\log(1-p_1) - \sum_{i=2}^{m} p_i \log\left(\frac{p_i}{1-p_1}\right) \\
&= h(p_1, 1-p_1) + (1-p_1) h\left(\frac{p_2}{1-p_1}, \ldots, \frac{p_m}{1-p_1}\right).
\end{aligned}$$

The first term is $h(\epsilon)$ and the second does not exceed $\epsilon \log(m-1)$.

To prove Theorem 7.3.2, we write $\epsilon_j = p(X \neq x_j | Y = y_j)$, and notice that we can write

$$\sum_j p_Y(y_j)\epsilon_j = \sum_j p(X \neq x_j, Y = y_j) = \epsilon.$$

By definition of the conditional entropy (Equation (3.2), the Fano inequality (the first part of this Theorem, already established), and concavity of function $h(\epsilon)$ (see Figure 1.1),

$$\begin{aligned}
h(X|Y) &= \sum_{y_j} p_Y(y_j) h(X|Y = y_j) \\
&\leq \sum_j p_Y(y_j)\left(h(\epsilon_j) + \epsilon_j \log(m-1)\right) \\
&= \left(\sum_j p_Y(y_j) h(\epsilon_j)\right) + \epsilon \log(m-1) \\
&\leq h(\epsilon) + \epsilon \log(m-1).
\end{aligned}$$

□

In fact, we only require a weaker result, that:

$$h(X|Y) \leq 1 + \epsilon \log m. \tag{7.2}$$

7.3 Entropy inequalities

Using this, we can establish an upper bound on the channel capacity, which gives half of Shannon's Second Coding Theorem. First, we give a general result on the way that conditioning changes uncertainty.

Lemma 7.4 *For any random variables U and V:*

$$h(U) \geq h(U|V),$$

with equality iff U and V are independent.

Proof That is, knowing V reduces (or at least fails to increase) the uncertainty about U. In fact, this is a simple corollary of Lemma 3.8 above. By Equation (3.1):

$$h(U) - h(U|V) = i(U, V).$$

Now Lemma 3.8 tells us that $i(U, V) \geq 0$, with equality if and only if U and V are independent. $\square$

The power of this result is that the U and V can be almost any random variables. As we have already argued that it is only the probabilities that matter, and not the values, U and V could in fact be pairs (X_1, X_2), or even strings $(X_1, \ldots X_n)$.

Exploiting this allows us to prove a technical bound on the mutual information.

Lemma 7.5 *For an memoryless channel:*

$$i\left(\mathbf{X}^{(N)}, \mathbf{Y}^{(N)}\right) \leq \sum_{j=1}^{N} i(X_j, Y_j),$$

with equality if the input symbols $X_1, \ldots, X_N$ are independent.

Proof Notice that

$$
\begin{aligned}
i\left(\mathbf{X}^{(N)}, \mathbf{Y}^{(N)}\right) &= h\left(\mathbf{Y}^{(N)}\right) - h\left(\mathbf{Y}^{(N)} \middle| \mathbf{X}^{(N)}\right) && (7.3)\\
&= h\left(\mathbf{Y}^{(N)}\right) - \sum_{j=1}^{N} h\left(Y_j | Y_1, \dots Y_{j-1}, \mathbf{X}^{(N)}\right) && (7.4)\\
&= h\left(\mathbf{Y}^{(N)}\right) - \sum_{j=1}^{N} h(Y_j | X_j). && (7.5)
\end{aligned}
$$

The justifications for these equalities are as follows:

(7.3) comes from substituting the right values in Equation (3.1).

(7.4) is proved by induction. That is $h(Y_1, Y_2)|\mathbf{X}^{(N)}) = h(Y_1, Y_2, \mathbf{X}^{(N)}) - h(\mathbf{X}^{(N)}) = h(Y_1, \mathbf{X}^{(N)}) + h(Y_2|Y_1, \mathbf{X}^{(N)}) - h(\mathbf{X}^{(N)}) = h(Y_1|\mathbf{X}^{(N)}) + h(Y_2|Y_1, \mathbf{X}^{(N)})$, and so on.

(7.5) comes from the case of equality in Lemma 7.4. That is, for a memoryless channel, Y_j depends on X_j, but is independent of all other terms. We are effectively taking $U = R|S$ and $V = T$ in Lemma 7.4, and concluding that if R and T are are independent, then $h(R|S) = h(R|S, T)$. You can either believe this argument, or check it from first principles in Exercise 9.

Further (by induction, using Lemma 3.8)

$$h\left(\mathbf{Y}^{(N)}\right) \leq \sum_{j=1}^{N} h(Y_j),$$

and rearranging, we obtain the result. Equality holds iff $Y_1, \dots, Y_N$ are independent, which is certainly true if $X_1, \dots, X_N$ are. □

Theorem 7.6 *For a memoryless channel, the channel capacity C obeys*

$$C \leq \sup_{p_X} i(X, Y).$$

Proof Suppose we have a source which produces equidistributed symbols on a set $\mathcal{U}_N$, of size 2^{NR_N}. We must show that for a sequence of codes with error probability ϵ_N converging to zero, then $R = \lim_{N\to\infty} R_N \leq \sup_{p_X} i(X, Y)$.

Now, suppose that the encoding rule f is 1-1 (if not, the bounds will be even tighter, since the error probability will be higher).

$$\begin{aligned} NR_N &= \log(\#\mathcal{U}_N) \\ &= h\left(\mathbf{X}^{(N)}\right) \\ &= h\left(\mathbf{X}^{(N)}|\mathbf{Y}^{(N)}\right) + i\left(\mathbf{X}^{(N)}, \mathbf{Y}^{(N)}\right) \\ &\leq (1 + (NR_N)\epsilon_N) + Ni(X, Y), \end{aligned}$$

where the first result follows by Equation (7.2), and the second by Lemma 7.5.

Hence, dividing through by N, since $R_N \to R$ and $\epsilon_N \to 0$, we can deduce the result. $\square$

Lecture 8: Shannon's Second Coding Theorem (cont)

8.1 Calculating channel capacities

We have a brief pause in the proof, to mention how to use Theorem 7.1 to calculate channel capacities.

A useful formula is $i(X,Y) = h(Y) - h(Y|X)$ (see Definition 3.9). In the case of a symmetric channel, then $h(Y|X=i)$ does not depend on i, and so $h(Y|X)$ just equals the entropy of one row. Hence, we only need to maximise $h(Y)$ over input distributions X – sometimes we will have a free choice of achievable Y, sometimes it will be constrained and we have to use other tricks.

Example 8.1 *For an memoryless binary symmetric channel, for each x,*

$$h(Y|X=x) = -\sum_{y=0,1} P(y|x)\log P(y|x) = -p\log p - (1-p)\log(1-p) = h(p,1-p)$$

and hence $h(Y|X) = h(p,1-p)$ does not depend on input distribution p_X. Thus, in this case

$$C = \sup_{p_X} h(Y) - h(p,1-p).$$

But $\sup_{p_X} h(Y)$ is equal to $\log 2 = 1$: it is attained at $p_X(0) = p_X(1) = 1/2$, because then

$$p_Y(0) = p_Y(1) = 1/2(p + (1-p)) = 1/2.$$

Therefore, $C = 1 - h(p,1-p)$.

8.2 Jointly typical sequences

We introduce the idea of jointly typical sequences, an idea which extends the definition of typical sequences and AEP of Remark 2.9.

Definition 8.2 *Given a probability distribution $p(x, y)$, we say that the sequences $(\mathbf{x}^{(N)}, \mathbf{y}^{(N)})$ are jointly typical if the following three properties hold:*

$$\begin{aligned}
\left| -\frac{1}{N} \log p\left(\mathbf{x}^{(N)}\right) - h(X) \right| &< \epsilon, \\
\left| -\frac{1}{N} \log p\left(\mathbf{y}^{(N)}\right) - h(Y) \right| &< \epsilon, \\
\left| -\frac{1}{N} \log p\left(\mathbf{x}^{(N)}, \mathbf{y}^{(N)}\right) - h(X, Y) \right| &< \epsilon,
\end{aligned}$$

where $p\left(\mathbf{x}^{(N)}\right) = \prod_{i=1}^{N} p(x_i)$ and so on.

We can establish a joint version of the Asymptotic Equipartition Property:

Theorem 8.3 *Writing $\mathbf{A}_\epsilon^{(N)}$ for the set of jointly typical sequences:*

1. *The probability of the set $\mathbb{P}\left(\mathbf{A}_\epsilon^{(N)}\right)$ converges to 1 as $N \to \infty$.*
2. *The size $\#\mathbf{A}_\epsilon^{(N)} \leq \exp_2(N(h(X, Y) + \epsilon))$.*
3. *Given random variables $\widetilde{\mathbf{X}}^{(N)}$, $\widetilde{\mathbf{Y}}^{(N)}$ independent and with the same marginals as $\mathbf{X}^{(N)}$ and $\mathbf{Y}^{(N)}$ then*

$$\mathbb{P}\left((\widetilde{\mathbf{X}}^{(N)}, \widetilde{\mathbf{Y}}^{(N)}) \in \mathbf{A}_\epsilon^{(N)}\right) \leq \exp_2(-N(i(X, Y) - 3\epsilon)).$$

Proof As before, the Law of Large Numbers will give that there exists a number N' such that for $N \geq N'$,

$$\begin{aligned}
\mathbb{P}\left(\left| -\frac{1}{N} \log p\left(\mathbf{x}^{(N)}\right) - h(X) \right| > \epsilon \right) &< \epsilon/3, \\
\mathbb{P}\left(\left| -\frac{1}{N} \log p\left(\mathbf{y}^{(N)}\right) - h(Y) \right| > \epsilon \right) &< \epsilon/3, \\
\mathbb{P}\left(\left| -\frac{1}{N} \log p\left(\mathbf{x}^{(N)}, \mathbf{y}^{(N)}\right) - h(X, Y) \right| > \epsilon \right) &< \epsilon/3
\end{aligned}$$

so that the probability of the union of these 3 sets is less than ϵ, and so the first part of the theorem holds.

To show the second part of the theorem, note that:

$$1 \geq \sum_{\mathbf{A}_\epsilon^{(N)}} p\left(\mathbf{x}^{(N)}, \mathbf{y}^{(N)}\right) \geq \left(\#\mathbf{A}_\epsilon^{(N)}\right) 2^{-N(h(X,Y)+\epsilon)}.$$

The third part follows by considering:

$$\begin{aligned}
&\mathbb{P}\left((\widetilde{\mathbf{X}}^{(N)}, \widetilde{\mathbf{Y}}^{(N)}) \in \mathbf{A}_\epsilon^{(N)}\right) \\
&= \sum_{\mathbf{x}^{(N)}, \mathbf{y}^{(N)} \in \mathbf{A}_\epsilon^{(N)}} \mathbb{P}\left((\widetilde{\mathbf{X}}^{(N)}, \widetilde{\mathbf{Y}}^{(N)}) = (\mathbf{x}^{(N)}, \mathbf{y}^{(N)})\right) \\
&= \sum_{\mathbf{x}^{(N)}, \mathbf{y}^{(N)} \in \mathbf{A}_\epsilon^{(N)}} \mathbb{P}\left((\widetilde{\mathbf{X}}^{(N)} = \mathbf{x}^{(N)}\right) \mathbb{P}\left(\widetilde{\mathbf{Y}}^{(N)}) = \mathbf{y}^{(N)}\right) \\
&\leq \sum_{\mathbf{x}^{(N)}, \mathbf{y}^{(N)} \in \mathbf{A}_\epsilon^{(N)}} \exp_2(-N(h(X) - \epsilon)) \exp_2(-N(h(Y) - \epsilon)) \\
&\leq \exp_2(N(h(X,Y) + \epsilon)) \exp_2(-N(h(X) - \epsilon)) \exp_2(-N(h(Y) - \epsilon)) \\
&= \exp_2(-N(i(X,Y) - 3\epsilon))
\end{aligned}$$

□

Now, notice that there are about $2^{Nh(X)}$ typical X sequences, about $2^{Nh(Y)}$ typical Y sequences, but only around $2^{Nh(X,Y)}$ jointly typical sequences. This means that if we pick a random typical X sequence, and a random typical Y sequence, the probability that sequence (X, Y) is jointly typical is about $2^{-Ni(X,Y)}$.

Definition 8.4 *To prove Shannon's Second Coding Theorem, we take any probability distribution $p(x)$, and generate a code randomly according to it; that we make a list of 2^{NR} codewords, where each symbol of each codeword is chosen independently according to the same distribution $p(x)$.*

Now, we can send 2^{NR} messages, so choose one codeword $\mathbf{x} = (x_1, \ldots x_n)$, taken uniformly from the set of 2^{NR} possible codewords. The receiver will receive a sequence $\mathbf{Y}^{(N)}$, distributed according to the channel probability matrix. Now, we introduce yet another decoding rule; typical set decoding.

Definition 8.5 *On receiving word $\mathbf{Y}^{(N)}$, look for a word $\mathbf{X}^{(N)}$ such that $(\mathbf{X}^{(N)}, \mathbf{Y}^{(N)})$ forms a jointly typical sequence. If there is a unique such word, return it as the best guess of the transmitted codeword. If there is more than one, or none at all, return an error.*

Theorem 8.6 *For a memoryless channel, the capacity*

$$C \geq \sup_{p_X} i(X, Y).$$

Proof We will show that typical set decoding is asymptotically optimal. We do this by taking a probability distribution p, and considering the average performance of the random codes generated in Definition 8.4, averaged across all possible codewords. This significantly simplifies the analysis. That is, we calculate the average across all random codes $\mathcal{C}$:

$$\begin{aligned}
\text{overall error prob.} &= \sum_{\mathcal{C}} \mathbb{P}(\mathcal{C})\mathbb{P}(\text{ error } |\mathcal{C}) \\
&= \sum_{\mathcal{C}} \mathbb{P}(\mathcal{C} \text{ used}) \frac{1}{2^{NR}} \sum_{w} \mathbb{P}(\text{ error } \mid \mathcal{C} \text{ used, word } w \text{ sent}) \\
&= \frac{1}{2^{NR}} \sum_{w} \sum_{\mathcal{C}} \mathbb{P}(\mathcal{C} \text{ used}) \mathbb{P}(\text{ error } \mid \mathcal{C} \text{ used, word } w \text{ sent}),
\end{aligned}$$

now notice that since we average across all random codes, it doesn't matter which codeword was actually sent, so without loss of generality, we can assume $w = 1$.

Note that two types of errors are possible; firstly if $\mathbf{X}^{(N)}$ and $\mathbf{Y}^{(N)}$ are not jointly typical, and secondly if there exists some other codeword which is jointly typical with $\mathbf{Y}^{(N)}$. Now, the trick will be to use the joint AEP; which shows that the first type of error has probability tending to zero, and that the probability that a given codeword and received word are jointly typical is about 2^{-Ni}, so we can afford to use about 2^{Ni} codewords. That is, if we write

$$E_i = \{\text{codeword } i \text{ is jointly typical with received } \mathbf{Y}^{(N)}\},$$

then the probability of an error is

$$\mathbb{P}(E_1^c \cup E_2 \cup E_3 \cup \ldots \cup E_{2^{NR}}) \leq \mathbb{P}(E_1^c) + \sum_{i=2}^{2^{NR}} \mathbb{P}(E_i).$$

Now, by the first part of Theorem 8.3, for N sufficiently large, the first event has probability less than ϵ. Similarly, the third part of the same Theorem shows that for $i \neq 1$, $\mathbb{P}(E_i) \leq \exp_2(-N(i(X,Y) - 3\epsilon))$, so that the error probability is less than

$$\epsilon + \exp_2(N(R - i(X,Y) + 3\epsilon)),$$

so that if $R < i(X,Y) - 3\epsilon$, this quantity goes to zero.

Now, notice that we have a free choice of probability distributions p, so we can choose the one which makes $i(X,Y)$ as large as possible. Further, since the average code has error probability tending to zero, we can guarantee that we will be able to choose some code which has its error probability tending to zero. $\square$

Lecture 9: Hamming spaces. Geometry of codes.

9.1 Hamming space

In this second part of the course, we consider coding; that is, practical ways of protecting information. We make a very natural definition.

Definition 9.1 *The Hamming space $H_N = \{0,1\}^N$ is the set of all binary strings (or 'words') of length N. The Hamming distance between two strings $\mathbf{x}^{(N)}, \mathbf{y}^{(N)} \in H_N$ is the number of places in which they differ and is given by:*

$$d(\mathbf{x}^{(N)}, \mathbf{y}^{(N)}) = \sharp\{i : x_i \neq y_i\} = \sum_i \mathbb{I}(x_i \neq y_i).$$

The weight of a binary string of length N is the number of non-zero elements it contains

$$w(\mathbf{x}^{(N)}) = d(\mathbf{x}^{(N)}, \mathbf{0}^{(N)}).$$

We introduce the dot product:

$$\mathbf{x} \cdot \mathbf{y} = \sum_{i=1}^{N} x_i y_i \mod 2.$$

Here obviously $\mathbf{0}^{(N)} = (0, 0, \ldots 0)$, the word of N zeroes, and $\mathbf{1}^{(N)} = (1, 1, \ldots 1)$, the word of N ones. Since we nearly always deal with words of a fixed length, we drop the superscript (N) where it will be clear from context.

The dot product is very similar to the Euclidean scalar product in $\mathbb{R}^N$, however it is not positive definite (and therefore does not define a norm). That is, there are non-zero vectors $\mathbf{x} \in \{0,1\}^N$ with $\mathbf{x} \cdot \mathbf{x} = 0$. Luckily, we do not need the positive definiteness.

As well as thinking of H_N as a vector space, it is natural to consider H_N as an Abelian group under pointwise addition: $(\mathbf{x} + \mathbf{y})_i = x_i + y_i$.

Lemma 9.2 *Note that $d(\mathbf{x}, \mathbf{y})$ defines a metric on H_N:*

(i) $0 \leq d(\mathbf{x}, \mathbf{y}) \leq N$ and $d(\mathbf{x}, \mathbf{y}) = 0$ iff $\mathbf{x} = \mathbf{y}$.

(ii) $d(\mathbf{x}, \mathbf{y}) = d(\mathbf{y}, \mathbf{x})$.

(iii) $d(\mathbf{x}, \mathbf{z}) \leq d(\mathbf{x}, \mathbf{y}) + d(\mathbf{y}, \mathbf{z})$.

Further d is translation-invariant:

(iv) For any $\mathbf{u} \in H_N$ the $d(\mathbf{x}, \mathbf{y}) = d(\mathbf{x} + \mathbf{u}, \mathbf{y} + \mathbf{u})$.

Geometrically, the Hamming space may be identified with the collection of the vertices of a unit cube in N dimensions. The Hamming distance can be thought of the number of edges you have to pass on the shortest path from one vertex to another.

We can motivate our interest in the Hamming distance as follows. A code is simply a subset $\mathcal{X}_N$ of the Hamming space H_N. We refer to the members of $\mathcal{X}_N$ as codewords, whether they correspond to letters or words or some other part of the original message.

Remark 9.3 *In many natural models of communication, we suppose that we transmit a codeword $\mathbf{x}$ of length N and receive a word $\mathbf{y}$, which is related to it by $\mathbf{y} = \mathbf{x} + \mathbf{e}$. It is plausible that the error vector $\mathbf{e}$ should 'mostly' consist of 0s, with a small number of 1s.*

For example, for the memoryless binary symmetric channel, the elements of $\mathbf{e}$ are independent, and for each element $e_i = 1$ with probability p. Hence, the maximum likelihood decoder looks for a codeword $\mathbf{x}_\star$ that has maximum number of digits coinciding with the received word $\mathbf{y}$. In fact, if $\mathbf{y}$ is received, the maximum likelihood decoder compares the probabilities

$$\mathbf{P}(\mathbf{y}|\mathbf{x}) = p^{d(\mathbf{x},\mathbf{y})}(1-p)^{N-d(\mathbf{x},\mathbf{y})} = (1-p)^N \left(\frac{p}{1-p}\right)^{d(\mathbf{x},\mathbf{y})}$$

for different codewords $\mathbf{x}$.

The first factor $(1-p)^N$ does not depend on $\mathbf{x}$, the decoder seeks to maximize the second factor, that is (in the case $p < 1/2$) to minimize $d(\mathbf{x}, \mathbf{y})$ (since $0 < p/(1-p) < 1$.)

In this case, the natural maximum likelihood decoding rule becomes a minimum distance decoding rule: "on receiving word $\mathbf{y}$, we decode it with the codeword $\mathbf{x} \in \mathcal{X}_N$ which minimises $d(\mathbf{x}, \mathbf{y})$".

This motivates us to study the balls B_r of radius r defined as:

$$B_r(\mathbf{x}) = \{\mathbf{y} : d(\mathbf{x}, \mathbf{y}) \leq r\}.$$

Of course, if $p > 1/2$, we use maximum distance decoding. We can think that as pre-processing we just reverse the parity of each bit, and then use minimum-distance decoding. If $p = 1/2$, the signal becomes IID noise, and we cannot decode the message – in particular, the $\mathbf{P}(\mathbf{y}|\mathbf{x})$ is constant for all $\mathbf{x}$, and so no choice of $\mathbf{x}$ is better than any other.

9.2 Codes

Recall that we consider codes $\mathcal{X}_N \subseteq H_N$.

Definition 9.4 *Parameter N is called the length of code $\mathcal{X}_N$, $r = \sharp\mathcal{X}_N$ the size and $\rho = (\log_2 r)/N$ the information rate (or transmission rate). A code has minimum distance (or just distance) δ if*

$$\delta = \min\{d(\mathbf{x}, \mathbf{y}) : \mathbf{x}, \mathbf{y} \in \mathcal{X}_N, \mathbf{x} \neq \mathbf{y}\}.$$

If the code has distance δ, we call the ratio δ/N the error control rate.

We often summarise this information by referring to $[N, r, \delta]$ codes.

Example 9.5 *Two simple codes are:*

1. *The repetition code consists of the two words $\mathbf{0} = (0, 0, \ldots, 0)$ and $\mathbf{1} = (1, 1, \ldots, 1)$. It has length N, size 2, information rate $1/N$ and minimum distance N, and so is $[N, 2, N]$.*
2. *The parity code consists of all the words $\mathbf{x}$ such that $\sum_i x_i \equiv 0 \mod 2$. It has length N, size 2^{N-1}, information rate $1 - 1/N$ and minimum distance 2, and so is $[N, 2^{N-1}, 2]$.*

As before we drop the subscript N from $\mathcal{X}_N$, unless that will prove confusing.

Definition 9.6 *A code $\mathcal{X}_N$ is said to be*

1. *D-error detecting if making up to D changes in any codeword does not produce another codeword*

2. *E-error correcting if making up to E changes in any codeword $\mathbf{x}$ produces a word which is still strictly closer to $\mathbf{x}$ than to any other codeword.*

Theorem 9.7 *Under these definitions:*

1. *Code $\mathcal{X}_N$ is D-error detecting iff its distance $\delta \geq D + 1$.*

2. *Code $\mathcal{X}_N$ is E-error correcting iff the balls of radius E about the codewords are disjoint, so $E = \lfloor (\delta - 1)/2 \rfloor$.*

Proof The first result is obvious. To prove the second result, first assume that the balls of radius E are disjoint. Then, changing up to E digits in a codeword produces a word that is still in the corresponding ball, and hence is further from any other codeword.

Conversely, let our code be E-error correcting. Then any word obtained by changing precisely E digits in a codeword remains in the ball of radius E around the original codewords but not any other ball of radius E. Making fewer changes cannot leave you in any other ball, for if it does then, moving towards the second centre will produce, soon or later, a word that is at distance E from the original codeword and at distance $< E$ from the second one, which is impossible. $\square$

Clearly, if a code detects D errors and D is even then it corrects $E = D/2$ errors, and if D is odd than it corrects $(D - 1)/2$ errors.

Shannon's Second Coding Theorem 7.1 shows that, under certain conditions, there exist asymptotically good codes. That is, codes which attain the limits imposed by the capacity of a channel. In fact, almost all random codes are asymptotically good. However, in practice, existence theorems are of little use. We will want to focus on good codes that can be explicitly described, and rapidly encoded and decoded.

9.3 New codes for old

Since codes are in general quite hard to construct from scratch, there is interest in creating new codes from old ones, since the parameters of new and old codes will often be related. In each case, we start with a code $\mathcal{X}$ of length N, size r and minimum distance δ.

Example 9.8 *The simplest example is parity extension. We produce a code* $\mathcal{X}^+$ *of length* $N+1$ *by taking each word* $\mathbf{x} = (x_1, \ldots, x_N) \in \mathcal{X}$ *and adding an extra bit* $x_{N+1} = x_1 + \ldots + x_N \mod 2$*.*

Clearly the size of the code is unchanged by this procedure. In general, the minimum distance will increase by 1, if δ *was odd (since any two words* $\mathbf{x}, \mathbf{y}$ *such that* $d(\mathbf{x}, \mathbf{y})$ *is odd will have a different parity bit) and stay the same, if* δ *was even.*

Example 9.9 *Another means of creating a new code is truncation. Given a codeword* $\mathbf{x} = (x_1, \ldots, x_N)$ *from* $\mathcal{X}$*, drop the last bit* x_N*, leaving a collection of vectors of length* $N-1$*. The new code,* $\mathcal{X}^-$*, has minimal distance* $\geq \delta - 1$ *and if* $\delta \geq 2$ *has the same size as* $\mathcal{X}$*. Repeated truncation will give the Singleton bound (Theorem 10.4) below.*

Example 9.10 *We can use the related procedure of shortening. We take the code* $\mathcal{X}$*, and pick a position* $i \in \{1, \ldots N\}$ *and a value* $j \in \{0, 1\}$*. Then, we simply make a sublist containing the codewords that have their* i*th bit equal to* j*, before deleting that bit from the list.*

Clearly the length of the code is reduced from N *to* $N-1$*. The minimum distance of the new code is* $\geq \delta$*. By the pigeonhole principle, given any* i *we can choose* j *to ensure that the new code has size at least* $r/2$*.*

The final construction that we shall introduce is the bar product, in Example 13.5. However, this only applies to linear codes, and so must wait until later.

Lecture 10: Bounds for codes

10.1 Hamming bound and perfect codes

It is natural to ask how many codewords we can have, given that we want to correct a given number of errors? In general, this turns out to be a very hard problem, though we can give bounds on how well we can do.

Observe that the volume of the ball in the Hamming space is

$$\nu_N(R) = \mathrm{vol}\,(B_R(z^{(N)})) = \sum_{i=0}^{R} \binom{N}{i}.$$

This gives the simplest bound on the size of a code.

Theorem 10.1 (Hamming bound) *If a code $\mathcal{X}_N$ corrects E errors then its size r obeys*

$$r \leq \frac{2^N}{\nu_N(E)}.$$

Proof By Theorem 9.7, the E-balls about the codewords $\mathbf{x} \in \mathcal{X}_N$ must be disjoint. Hence, the total number of points covered by these balls is $r\nu_N(E)$ which must be $\leq 2^N$, the cardinality of the Hamming space $\{0,1\}^N$. $\square$

If there exists a good code $\mathcal{X}_N$ correcting E errors, then we must be able to pack the Hamming space efficiently with balls of radius E. Finding good codes reduces to a purely geometrical problem. Of course, the best we can possibly do is to fill the whole space with codeword balls. If this is achieved, we will always be able to make a good estimate of the codeword $\mathbf{x}$ that led to received word $\mathbf{y}$.

Definition 10.2 *An E-error correcting code $\mathcal{X}_N$ of size $\sharp\,\mathcal{X}_N = r$ is called perfect when equality is achieved in the Hamming bound:*

$$r = \frac{2^N}{\nu_N(E)}\ .$$

If a code $\mathcal{X}_N$ is perfect, every word $\mathbf{y} \in \{0,1\}^N$ belongs to a unique ball $B_E(\mathbf{x})$. That is, you are able to decode every received word $\mathbf{y}$ by some codeword: this leads to the correct answer if the number of errors is $\leq E$, and to a wrong answer if it is $> E$.

In fact, perfect codes are very rare. A series of papers in the 1970s established this – in particular, papers by van Lint and Tietäväinen played a significant role in the proof. They showed that perfect codes only exist in the binary case for

1. $E = 0$, $r = 2^N$ (trivial code – everything is a codeword), for N odd, $E = (N-1)/2$, $r = 2$ (repetition code of odd length – see Example 9.5).
2. $E = 1$: here $N = 2^l - 1$, $r = 2^{2^l-1-l}$; the corresponding codes are called the Hamming codes – see Section 13.1
3. $E = 3$: here $N = 23$, $r = 2^{12}$; the corresponding code is called the (binary) Golay code – see Section 16.2.

(Allowing codes in alphabets of size ≥ 2 does not create many more perfect codes. The only really interesting new perfect code has three symbols, and is called the ternary Golay code– see Exercises 13 and 20).

10.2 Two more bounds

We now offer a lower bound on codes in the binary case.

Theorem 10.3 (Gilbert–Varshamov bound) *There exists a code $\mathcal{X}_N$ with minimal distance δ such that*

$$r = \sharp\, \mathcal{X}_N \geq \frac{2^N}{\nu_N(\delta - 1)}.$$

Proof Consider a code of maximal size among the codes of minimal distance δ (and length N). Then any word $\mathbf{y} \in \{0,1\}^N$ must be distant $\leq \delta - 1$ from some codeword: otherwise you can add $\mathbf{y}$ to the code without changing the minimal distance. Hence, the balls of radius $\delta - 1$ about the codewords cover the whole Hamming space $\{0,1\}^N$. That is, $r\nu_N(\delta-1) \geq 2^N$. $\square$

The problem with this bound is that although in one sense it is simple and constructive, to find balls with no codewords in is a computing-intensive problem, and rapidly becomes unfeasible as the length of the code increases. It took many years to provide an explicit construction for a family of codes that asymptotically achieved the bound. This problem was solved in 1982 by Tsfasman, Vlădut and Zink, using algebraic geometry codes, where finding codes with large numbers of codewords corresponds to the (solved) problem of finding elliptic curves with large numbers of rational points.

Theorem 10.4 (Singleton bound) *Any code $\mathcal{X}_N$ with minimal distance δ has*

$$r = \sharp\,\mathcal{X}_N \leq 2^{N-\delta+1}.$$

Proof Repeat the truncation procedure of Example 9.9 a total of $\delta - 1$ times, still preserving the size of code $\mathcal{X}_N$. The remaining code will be contained in the Hamming space $\{0,1\}^{N-\delta+1}$ that contains $2^{N-\delta+1}$ elements. $\square$

10.3 Plotkin bound

The Hamming and Singleton bounds are not tight when $\delta \sim N/2$. For example, in the case $N = 7$ and $\delta = 4$, the Hamming bound gives $r \leq 2^7/(1+7) = 16$, and the Singleton bound gives $r \leq 2^{7+1-4} = 16$. Overall then, the best upper bound proved so far is $r \leq 16$. In fact, the Plotkin bound (Theorem 10.5 below) proves that $r \leq 8$.

In fact the Plotkin bound turns out to be sharp in this case, in the sense that there exists a code with these $N = 7$, $\delta = 4$ and $r = 8$. The codewords are as follows:

$$\begin{array}{cccc} 0000000 & 1110001 & 1001101 & 0111100 \\ 0101011 & 1011010 & 1100110 & 0010110 \end{array}$$

We can tell this code has distance 4, since the truncated version has distance 3. We can see this by comparing all pairs of words, or by observing that the truncated code is linear, and using Theorem 13.1 below.

The Gilbert–Varshamov lower bound Theorem 10.3 gives that there must exist a code of size at least 2, which shows that Gilbert–Varshamov is not sharp in this region either.

Theorem 10.5 (Plotkin bound) *For any code $\mathcal{X}_N$ with minimum distance δ satisfying $N < 2\delta$,*

$$r = \sharp\mathcal{X} \leq 2\left\lfloor \frac{\delta}{2\delta - N} \right\rfloor. \tag{10.1}$$

Proof Consider the quantity $T = \sum_{\mathbf{x},\mathbf{y}\in\mathcal{X}} d(\mathbf{x},\mathbf{y})$ in two ways. Firstly,

$$T = \sum_{\mathbf{x}\in\mathcal{X}} \sum_{\mathbf{y}\in\mathcal{X}} d(\mathbf{x},\mathbf{y}) \geq r(r-1)\delta \tag{10.2}$$

On the other hand, write code $\mathcal{X}$ as an $r \times N$ matrix with rows as codewords. Suppose that column i of the matrix contains s_i zeroes and $r - s_i$ ones. Then

$$T = \sum_{\mathbf{x}\in\mathcal{X}} \sum_{\mathbf{y}\in\mathcal{X}} d(\mathbf{x},\mathbf{y}) = 2\sum_{i=1}^{N} s_i(r - s_i). \tag{10.3}$$

If r is even, the RHS of (10.3) is maximized when $s_i = r_i/2$ which, together with (10.2), yields

$$r(r-1)\delta \leq \frac{1}{2}Nr^2,$$

or (cancelling a factor of r from each side)

$$r \leq \frac{2\delta}{2\delta - N}.$$

As r is even, this implies

$$r \leq 2\left\lfloor \frac{\delta}{2\delta - N} \right\rfloor.$$

If r is odd, the RHS of (10.3) is maximised at $s_i = (r \pm 1)/2$, and hence is $\leq N(r^2-1)/2$ which, together with (10.2) yields

$$r \leq \frac{N}{2\delta - N} = \frac{2\delta}{2\delta - N} - 1.$$

This implies in turn that

$$r \leq \left\lfloor \frac{2\delta}{2\delta - N} \right\rfloor - 1 \leq 2\left\lfloor \frac{\delta}{2\delta - N} \right\rfloor,$$

because $\lfloor 2x \rfloor \leq 2\lfloor x \rfloor + 1$. $\square$

Lecture 11: More bounds

11.1 Asymptotic bounds

We give the asymptotic form of these bounds. First, a technical lemma:

Lemma 11.1 *Let* $\lambda \in (0, 1/2)$. *Then*

$$\lim_{N\to\infty} \frac{1}{N} \log \nu_N([\lambda N]) = h(\lambda) = -\lambda \log \lambda - (1-\lambda)\log(1-\lambda).$$

Proof Since $R = [\lambda N] \leq N/2$, in the sum

$$\nu_N(R) = \sum_{i=0}^{R} \binom{N}{i},$$

the last term is the largest. To check this, consider the ratio of two successive terms:

$$\frac{\binom{N}{i+1}}{\binom{N}{i}} = \frac{N-i}{i+1} \geq 1 \quad \text{for } 0 \leq i \leq R-1.$$

Hence,

$$\binom{N}{R} \leq \nu_N(R) \leq (R+1)\binom{N}{R}.$$

Now use Stirling's formula: $N! \sim N^{N+1/2}e^{-N}\sqrt{2\pi}$. Then

$$\log \binom{N}{R} = -(N-R)\ \log \frac{N-R}{N} - R \log \frac{R}{N} + O\left(\log N\right)$$

and hence:

$$\begin{aligned}
\frac{1}{N}\ \log \nu_N(R) &\geq \frac{1}{N} \log \binom{N}{R} \\
&= -\left(1 - \frac{R}{N}\right) \log\left(1 - \frac{R}{N}\right) - \frac{R}{N}\ \log \frac{R}{N} + \frac{O(\log N)}{N} \\
&= h(R/N) + \frac{O(\log N)}{N}.
\end{aligned}$$

Similarly

$$\frac{1}{N}\log\nu_N(R) \;\le\; \frac{1}{N}\log\binom{N}{R} = h(R/N) + \frac{\log(R+1)}{N} + \frac{O(\log N)}{N}.$$

The limit $R/N \to \lambda$ yields the result. □

We define $r^*(N, [\lambda N])$ to be the maximal size of the code of length N and distance $[\lambda N]$ and consider

$$\lim_{N\to\infty}\frac{1}{N}\log r^*(N, [\lambda N]) = \alpha(\lambda), \quad 0 < \lambda < 1/2.$$

Theorem 11.2 *The asymptotic form of the bounds are:*

1. $\alpha(\lambda) \le 1 - h(\lambda/2),\ 0 \le \lambda \le 1/2,$ *(Hamming),*
2. $\alpha(\lambda) \le 1 - \lambda,\ 0 \le \lambda \le 1/2,$ *(Singleton),*
3. $\alpha(\lambda) \ge 1 - h(\lambda),\ 0 \le \lambda \le 1/2,$ *(Gilbert-Varshamov),*
4. $\alpha(\lambda) = 0,\ 1/2 \le \lambda \le 1.$ *(Plotkin).*

Proof By inspection. □

By using more elaborate bounds (also due to Plotkin), one can show that

$$\alpha(\lambda) \le 1 - 2\lambda,\ 0 \le \lambda \le 1/2.$$

Figure 11.1 shows the asymptotic behaviour of the bounds established.

Until 1972, all known families of codes with asymptotic error control rate α had asymptotic information rate zero. The development of codes which escaped this regime was due to Justesen.

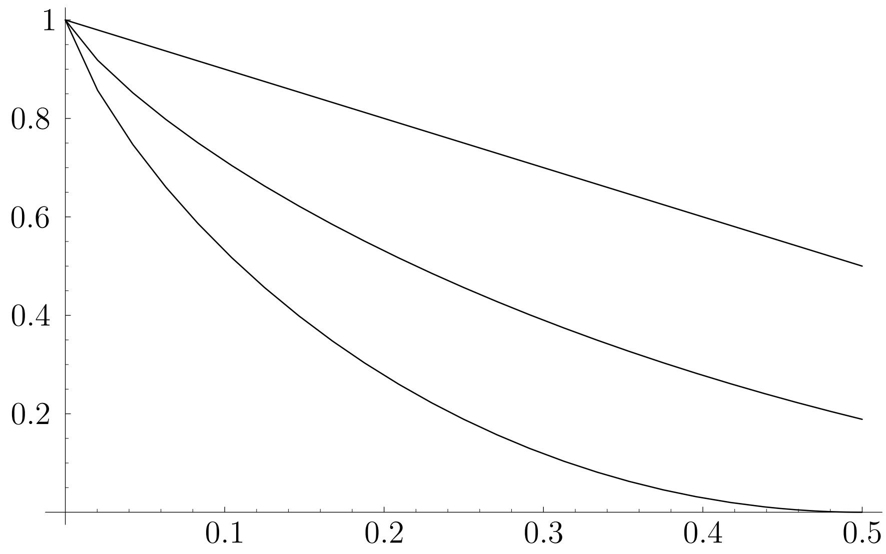

Figure 11.1: Asymptotic forms of the bounds

11.2 Elias bound

There exist many more bounds, where as you might expect, increased tightness comes at the cost of increased difficulty of proof. For example, the Elias bound can be seen as a more complicated version of the Plotkin bound.

Given a code $\mathcal{X}_N$, and given a value s, for each word $\mathbf{y} \in H_N$ we can define $K_{\mathbf{y}}$ to be the number of codewords in $\mathcal{X}_N$ in the ball $B_s(\mathbf{y})$.

Lemma 11.3 *For all $\mathcal{X}_N$ and s, there exists a word $\mathbf{y} \in H_N$ such that*

$$K_{\mathbf{y}} \geq \frac{r\nu_N(s)}{2^N},$$

which we refer to as a 'critical' word.

Proof Notice that (by symmetry of the distance) $\mathbf{x} \in B_s(\mathbf{y})$ if and only if $\mathbf{y} \in B_s(\mathbf{x})$. This means that each codeword $\mathbf{x}$ appears in $\nu_N(s) = \sharp B_s(\mathbf{y})$

balls of radius s. Hence counting the same expression two different ways:

$$\begin{aligned}\sum_{\mathbf{y}\in H_N} K_{\mathbf{y}} &= \sum_{\mathbf{y}\in H_N}\sum_{\mathbf{x}\in\mathcal{X}_N} \mathbb{I}(\mathbf{x}\in B_s(\mathbf{y})) \\ &= \sum_{\mathbf{x}\in\mathcal{X}_N}\sum_{\mathbf{y}\in H_N} \mathbb{I}(\mathbf{x}\in B_s(\mathbf{y})) \\ &= \sum_{\mathbf{x}\in\mathcal{X}_N}\sum_{\mathbf{y}\in H_N} \mathbb{I}(\mathbf{y}\in B_s(\mathbf{x})) = r\nu_N(s).\end{aligned}$$

Dividing through by 2^N, we see that the average value of $K_{\mathbf{y}}$ is at least $r\nu_N(s)/2^N$, and so there must at least one $\mathbf{y}$ where $K_{\mathbf{y}}$ is greater or equal to this value. □

Using this we can obtain the Elias bound:

Theorem 11.4 *For any integer s such that $1 \leq s < N/2$ and $2s^2 - 2Ns + Nd > 0$, the size of a code of length N and distance d satisfies*

$$r \leq \left(\frac{Nd}{2s^2 - 2Ns + Nd}\right)\frac{2^N}{\nu_N(s)}.$$

Proof From the original code, we create a new one $\mathcal{X}'$ as follows. Without loss of generality, $\mathbf{0}$ is a critical word in the original code (if not, we can pick a critical word $\mathbf{y}$ and translate the entire code using it). Now, define the new code $\mathcal{X}' = \mathcal{X} \cap B_s(\mathbf{0})$.

Now, this new code $\mathcal{X}'$ has minimum distance $D \geq d$ and size $R \geq r\nu_N(s)/2^N$, and we can consider it in the same way as we did in proving the Plotkin bound. That is, consider the quantity

$$T = \sum_{\mathbf{x},\mathbf{y}\in\mathcal{X}'} d(\mathbf{x},\mathbf{y}).$$

As in the Plotkin bound, we know that $T \geq R(R-1)D$.

We introduce some more notation – we write k_i for the total number of zeroes in the ith position in all the words in $\mathcal{X}'$ and P for the total number of zeroes in all the codewords – that is, $P = \sum_{i=1}^n k_i$. We know that

$$T = \sum_{i=1}^n 2k_i(R-k_i) = 2PR - 2\sum_{i=1}^n k_i^2. \tag{11.1}$$

By Cauchy-Schwarz:

$$P^2 = \left(\sum_{i=1}^{N} k_i \right)^2 \leq \left(\sum_{i=1}^{N} k_i^2 \right) N,$$

which means (in Equation (11.1)) that

$$T \leq 2PR - 2P^2/N. \tag{11.2}$$

Now, we will regard Equation (11.2) as a quadratic in P. Note that it is decreasing for $P \geq RN/2$.

Recall that since all the words in $\mathcal{X}'$ are in $B_s(\mathbf{0})$, they each have at least $N - s$ zeroes. This means that P, the total number of zeroes satisfies $P \geq R(N - s)$. Since $s \leq N/2$, this means that we are in the range $P \geq NR/2$, so we should take P as small as possible. That is, we should substitute $P = R(N - s)$ into Equation (11.2) to obtain that

$$T \leq \frac{2R^2(N - s)s}{N}.$$

Combining this with the Plotkin-style result we see that

$$R(R - 1)D \leq T \leq \frac{2R^2(N - s)s}{N},$$

which means that (cancelling the R):

$$(R - 1)DN \leq 2R(N - s)s,$$

or

$$(2s^2 - 2sN + DN)R \leq DN,$$

and the result follows on replacing D by d. □

The asymptotic form of the Elias bound is more complicated than those already described, since there are different choices of s possible. See the Example Sheet.

11.3 Codes in larger alphabets

Although we have described coding in the binary case so far, most of what we have described will pass through for larger alphabets. That is, we pick a q, and replace the Hamming space H_N by the set $\{0, 1, \dots q-1\}^N$. We can describe distance (and hence weight) in the same way as before. This means that the description of error detecting and correcting abilities in terms of balls goes through in the same way.

We do need to adapt the definition of the dot product $\cdot$, to give

$$\mathbf{x} \cdot \mathbf{y} = \sum_i x_i y_i \mod q.$$

Having done this, we can adapt the bounds previously described to the case of alphabets of size q.

Theorem 11.5

1. **Hamming bound** *If a code $\mathcal{X}_N$ corrects E errors then*

$$r = \sharp\, \mathcal{X}_N \leq \frac{q^N}{\sum_{i=0}^{E} \binom{N}{i} (q-1)^i}.$$

2. **Gilbert–Varshamov bound** *There exists a code $\mathcal{X}_N$ with minimal distance δ such that*

$$r = \sharp\, \mathcal{X}_N \geq \frac{q^N}{\sum_{i=0}^{\delta} \binom{N}{i} (q-1)^i}.$$

3. **Singleton bound** *Any code $\mathcal{X}_N$ with minimal distance δ has*

$$r = \sharp\, \mathcal{X}_N \leq q^{N-\delta+1}.$$

4. **Plotkin bound** *For any code $\mathcal{X}_N$ with minimum distance δ, writing $\theta = (q-1)/q$, if $\delta > \theta N$ then*

$$r = \sharp \mathcal{X} \leq \frac{\delta}{\delta - \theta N}.$$

Proof The proofs of most of these results are very straightforward.

We have to adapt the proof of the Plotkin bound by letting k_{ij} be the number of symbol j in position i. Then the contribution from the ith position is

$$\sum_{j=0}^{q-1} k_{ij}(M - k_{ij}) = M^2 - \sum_{j=0}^{q-1} k_{ij}^2 \leq M^2 - M^2/q = \theta M^2,$$

(since $\sum_{ij} k_{ij}^2$ is minimised by taking $k_{ij} = M/q$). $\square$

Lecture 12: Linear and Hamming codes

12.1 Linear codes

As discussesd previously, we are interested in finding large codes $\mathcal{X}_N$ with given minimum distance, that is, the problem of efficiently packing the Hamming space with balls. This is a hard (geometric) problem though, as we have seen, bounds exist to tell us how well we can expect to do.

However there is a secondary problem, that of decoding the message, often in real time (for example in a CD player, or a mobile phone message). For example, if you switch from the analogue version of a TV channel to the digital, you will find that the digital version is delayed by a few seconds compared with the analogue. This is because the set-top box takes a few seconds to decode the compressed digital signal. Such a delay is deemed acceptable to viewers – a delay of a few minutes would not be.

If a code has 2^{100} codewords (which is perfectly plausible in some applications), it would take more storage than exists in the universe to even list the codewords, and would take longer than the lifetime of the universe to find the closest codeword $\mathbf{x}$ to a received word $\mathbf{y}$ using an exhaustive search. We will need to consider codes with linear structure which makes the decoding easier (see later).

Definition 12.1 *A code $\mathcal{X}$ is called linear if, for every pair of codewords $\mathbf{x}$ and $\mathbf{y}$, $\mathcal{X}$ also contains the sum $\mathbf{x}+\mathbf{y}$ (recall this means the word with digits $(x_i + y_i) \mod 2$).*

Remark 12.2 *In other words, instead of just being a subset as before, $\mathcal{X}$ is a linear subspace of $\{0,1\}^N$, over the field $\{0,1\}$. A field is a set endowed with operations of addition and multiplication so that natural properties of commutativity, associativity and distributivity are valid.*

Linear codes are popular because they are easy to work with. For example, to describe a linear code $\mathcal{X}$ you only have to give a basis (that is, a maximal linearly independent set of words or vectors).

Definition 12.3 *A linear subspace is generated by its basis, and all bases for a given subspace have the same number of words: this number is called the dimension or the rank of the subspace. A linear code of length N, rank k and distance d is called an (N, K, d)-code.*

Note that any non-empty linear code contains the zero codeword $\mathbf{0} = \mathbf{x} + \mathbf{x}$.

12.2 Generators and parity checks

Lemma 12.4 *Any linear subspace of $\{0,1\}^N$ of rank k contains 2^k vectors.*

Proof A basis of the subspace contains k linearly independent vectors. The whole subspace is generated by the basis, that is it consists of the sums of basis vectors. There are precisely 2^k sums (the number of subsets of $\{1, \dots, k\}$), and they all give different vectors. $\square$

Hence an (N, k) code may be used for encoding all possible source strings of length k; the information rate of an (N, k)-code is k/N. As before, if the code has distance d, we call the ratio d/N the error control rate.

Definition 12.5 *A linear code of rank k is characterized by an $k \times N$ matrix of 0s and 1s, whose rows are linearly independent:*

$$G = \begin{pmatrix} g_{11} & \cdots & \cdots & g_{1N} \\ g_{21} & \cdots & \cdots & g_{2N} \\ & \vdots & \vdots & \\ g_{k1} & \cdots & \cdots & g_{kN} \end{pmatrix}$$

The rows $(g_{i1}, \dots, g_{iN})$ are the basis vectors of the linear code. Matrix G is called a generating matrix or generator of a linear code.

There is great freedom in the choice of basis; many matrices may generate the same code.

Definition 12.6 *A linear (N, k)-code $\mathcal{X}$ may alternatively be described as the kernel of a matrix H, again with entries 0 and 1. Matrix H is called a* *parity check (or simply check) matrix of code $\mathcal{X}$.*

$$H = \begin{pmatrix} h_{11} & \dots & h_{1(N-k)} \\ h_{21} & \dots & h_{2(N-k)} \\ & \vdots & \\ h_{N1} & \dots & h_{N(N-k)} \end{pmatrix}$$

with

$$\ker H = \{\mathbf{x} = x_1 \dots x_N : \mathbf{x}H = 0\}$$

The columns of H are vectors orthogonal to $\mathcal{X}$, in the sense of the dot product from Definition 9.1.

The parity check matrix is again not unique. If code $\mathcal{X}_N$ has rank k then H must contain $N-k$ linearly independent columns. For definiteness, we discard redundant columns and always think of H as an $N \times (N-k)$ matrix with linearly independent columns.

In many cases, the description of a code by a check matrix is more convenient than by a generator.

Example 12.7 (Hamming (7,4) code) *The code is determined by a 7×3 parity check matrix. The rows of the check matrix are all the non-zero words of length 3. If you order these words lexicographically, you obtain*

$$H^{\text{lex}} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}.$$

The corresponding generating matrix may be written as

$$G^{\text{lex}} = \begin{pmatrix} 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

In many cases it is convenient to write the check matrix of a linear (N,k) code in a canonical (or standard) form:

$$H^{\mathrm{can}} = \begin{pmatrix} I_{N-k} \\ H' \end{pmatrix}. \tag{12.1}$$

Example 12.8 *In the case of the Hamming (7,4) code*

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

with a corresponding generating matrix also in canonical form:

$$G^{\mathrm{can}} = (G' \ \ I_k) = \begin{pmatrix} 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{pmatrix}. \tag{12.2}$$

Note that in this case $G' = H'$. As Theorem 12.9 shows, this is not a coincidence.

Theorem 12.9 *For any linear code $\mathcal{X}$ there exists an equivalent code $\mathcal{X}'$ (obtained from $\mathcal{X}$ by permuting digits), with the generating matrix G^{can} and the check matrix H^{can} in standard form (12.2), (12.1) and $G' = H'$.*

Proof Assume, without loss of generality, that code $\mathcal{X}$ is non-trivial (that is, consists of more than just the zero word $\mathbf{0} = 0...0$). Write a basis for $\mathcal{X}$ and take the corresponding generating matrix G. By performing row operations we change the basis, but do not change the code. Our matrix G contains a nonzero column, say l_1: perform row operations to make g_{1l_1} the only non-zero entry in this column. Permute columns, to place column l_1 at position $N-k$. Drop row 1 and column $N-k$ (i.e., the old column l_1) and perform a similar procedure with the rest, ending up with the only non-zero entry g_{2l_2} in column l_2. Place column l_2 at position $N-k+1$. Continue until you form a triangular $k \times k$ submatrix. Further operations may be reduced to this matrix only. If this matrix is a unit matrix, stop. If not, pick the first column with more than one non-zero entry. Add the corresponding rows from the bottom to 'kill' redundant non-zero entries. Repeat until you form a unit submatrix. You end up with a generating matrix in a standard form, and the new code is equivalent to the original one.

To complete the proof, observe that matrices $G^{\rm can}$ and $H^{\rm can}$ figuring in (12.2), (12.1), with $G' = H'$, have k independent rows and $N-k$ independent columns. Besides, the $k \times (N-k)$ matrix $G^{\rm can}H^{\rm can}$ vanishes, since

$$\begin{aligned}(G^{\rm can}H^{\rm can})_{ij} &= (\text{row } i \text{ of } G).(\text{column } j \text{ of } H)\\ &= g'_{ij} \cdot 1_j + 1_{N-k+i}h'_{ij} = g'_{ij} + g'_{ij} = 0.\end{aligned}$$

Hence, $H^{\rm can}$ is a check matrix for $G^{\rm can}$. $\square$

Formally, $G^{\rm lex}$ and $G^{\rm can}$ determine different codes. However, these codes are equivalent:

Definition 12.10 *Two codes are called equivalent if they differ only in the order of digits.*

In general, since they have the same parameters, unless otherwise stated, we do not distinguish between equivalent linear codes.

12.3 Encoding and decoding messages

The advantage of writing G in canonical form is that a source string $\mathbf{u}^{(k)}$ can be encoded as an N-vector $\mathbf{u}^{(k)}G^{\rm can}$; by Equation (12.2) the last k digits form word $\mathbf{u}^{(k)}$ (called the information digits) whereas the first $N-k$ are called parity-check digits. The parity-check digits carry the redundancy that allows the decoder to detect and correct errors.

Decoding linear codes is also simple. Recall that maximum likelihood decoding corresponds to decoding received word $\mathbf{y} = y_1 \dots y_N$ by the closest codeword $\mathbf{x} \in \mathcal{X}$. We can think of this in terms of error patterns. That is, when a codeword $\mathbf{x}$ is transmitted, the received word $\mathbf{y}$ can be expressed as $\mathbf{x}+\mathbf{e}$, and we want to look for the error pattern $\mathbf{e}$ which has the least weight.

Now, given a received word $\mathbf{y}$, we can calculate the syndrome of $\mathbf{y}$, that is, the product $\mathbf{y}H$. There are 2^{N-k} possible syndromes.

For each codeword $\mathbf{x}$, we know that $\mathbf{y}H = (\mathbf{x}+\mathbf{e})H = \mathbf{e}H$. Hence, there are a certain number of possible error patterns $\mathbf{e}$, each of which give a product $\mathbf{e}H$. Thus, for a given syndrome $\mathbf{s}$, we simply look for the error pattern $\mathbf{e}^*$ which gives the minimum in

$$\text{minimise } w(\mathbf{e}) : \mathbf{e}H = \mathbf{s}.$$

Then, the minimum distance decoder decodes word $\mathbf{y}$ to the codeword $\mathbf{y}+\mathbf{e}^*$.

In practice, the decoding rule (known as syndrome decoding) is implemented as follows. The syndromes are listed in the syndrome table, and for each one a leader of the corresponding coset is calculated. In fact, it is easiest to take all the plausible leaders (that is, error vectors of low weight), and work out the syndrome for each one. This can be done as preprocessing, although we have to store a large sorted table of the syndromes and leaders.

Then, on receiving a word $\mathbf{y}$, you calculate its syndrome $\mathbf{y}H$ and find, in the syndrome table, the corresponding leader $\mathbf{u}$. Then set, as before, $\mathbf{x} = \mathbf{y} + \mathbf{u}$.

Example 12.11 *In the case of the Hamming* $(7,4)$ *code, the syndrome table would be:*

syndrome			*leader*						
0	0	0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	0	0	0
0	1	0	0	1	0	0	0	0	0
0	1	1	0	0	1	0	0	0	0
1	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	1

Hence, for example on receiving $\mathbf{y} = (1,1,1,1,1,0,0)$*, we calculate* $\mathbf{y}H = (0,0,1)$*, so the leader is* $\mathbf{u} = (1,0,0,0,0,0,0)$*, and the closest codeword is* $\mathbf{z} - \mathbf{u} = (0,1,1,1,1,0,0)$.

Theorem 12.12 *For the Hamming code, for each syndrome the leader* $\mathbf{u}$ *is unique and contains not more than one non-zero digit. More precisely, if the syndrome* $\mathbf{y}H = \mathbf{s}$ *gives row* i *of the check matrix then the leader of the corresponding coset has the only non-zero digit* i.

Proof The leader minimizes the distance between the received word and the code. The Hamming code is perfect 1-error correcting (see Theorem 13.3). Hence, every word is either a codeword or within distance 1 of a unique codeword. Hence, the leader is unique and contains at most one non-zero digit. If the syndrome $\mathbf{y}H = \mathbf{s}$ occupies position i among the rows of the parity-check matrix then, for word $\mathbf{e}_i = (0\ldots1\ 0\ldots0)$ with the non-zero digit i,

$$(\mathbf{y} + \mathbf{e}_i)H = \mathbf{s} + \mathbf{s} = 0,$$

so $(\mathbf{y} + \mathbf{e}_i) \in \mathcal{X}$. $\square$

Lecture 13: Hamming, Reed–Muller and dual codes

13.1 Hamming codes

Recall from Definition 9.1 that the weight $w(\mathbf{x})$ of a word $\mathbf{x}$ is the number of non-zero digits in $\mathbf{x}$:

$$w(\mathbf{x}) = \{i : 1 \le i \le N, x_i \ne 0\}.$$

Theorem 13.1 *The distance of a linear code is equal to:*

1. *The minimal weight of its non-zero codewords.*
2. *The minimal size of a linearly dependent sets of rows from the check matrix.*

Proof 1. As code $\mathcal{X}$ is linear, the sum $\mathbf{x}+\mathbf{y} \in \mathcal{X}$ for each pair of codewords $\mathbf{x}, \mathbf{y} \in \mathcal{X}$. The invariance property of Lemma 9.2 gives that $d(\mathbf{x}, \mathbf{y}) = d(\mathbf{0}, \mathbf{x}+\mathbf{y}) = w(\mathbf{x}+\mathbf{y})$ for any pair of codewords. Hence, the minimal distance of $\mathcal{X}$ equals the minimal distance between $\mathbf{0}$ and the rest of the code, i.e., the minimal weight of a non-zero codeword from $\mathcal{X}$.

2. Let $\mathcal{X}$ be a linear code with a parity-check matrix H and minimal distance δ. Then there exists a codeword $\mathbf{x} \in \mathcal{X}$ with exactly δ non-zero digits. Since $\mathbf{x}H = \mathbf{0}$, there are δ rows of H which are linearly dependent (they correspond to the non-zero digits in $\mathbf{x}$).

On the other hand, if there exist $(\delta - 1)$ rows of H which are linearly dependent then their sum is zero. But that means that there exists a non-zero word $\mathbf{y}$, of weight $w(\mathbf{y}) = \delta - 1$, such that $\mathbf{y}H = 0$. Then $\mathbf{y}$ must belong to $\mathcal{X}$ which is impossible, since $\min[w(\mathbf{x}) : \mathbf{x} \in \mathcal{X}, \mathbf{x} \ne \mathbf{0}] = \delta$. □

Remark 13.2 *Note that if a code has minimum distance δ, then there exists a set of δ linearly dependent rows of H, so there exists a $\delta \times \delta$ singular submatrix of H.*

Theorem 13.3 *The Hamming (7,4) code has minimal distance 3, and so detects 2 errors and corrects 1. It is perfect.*

Proof Rows 1, 6, 7 of the parity check matrix H^{lex} are linearly dependent (in fact taking any pair of rows and their sum gives a linearly dependent triple). No two rows are linearly dependent because they are distinct ($\mathbf{x}+\mathbf{y}=\mathbf{0}$ means that $\mathbf{x}=\mathbf{y}$). Also, the volume $\nu_7(1)$ equals $1+7=8=2^3$, and the size of the code is 2^4. Of course $2^{4+3}=2^7$. $\square$

Notice that we can easily generalise the construction of the Hamming (7,4) code.

Theorem 13.4 *Consider a matrix whose rows are all the non-zero words of length l:*

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

Using this as a parity check matrix gives the Hamming $(2^l-1, 2^l-1-l)$ *linear code, with minimal distance 3, which is perfect.*

Proof The columns of this matrix are linearly independent, and hence it may be considered as a check matrix of a linear code of length $N=2^l-1$ and rank $N-l=2^l-1-l$. Any two rows of H are linearly independent but there exist linearly dependent triples of rows, such as the first three. Hence, the code with check matrix H has a minimal distance 3, and so detects 2 errors and corrects 1.

This is a perfect 1-error correcting code: the volume of the 1-ball $\nu_{2^l-1}(1)$ equals $1+2^l-1=2^l$, and (size) $\times$ (volume) $=2^{2^l-1-l}\times 2^l=2^{2^l-1}=2^N$. The information rate is $(2^l-l-1)/(2^l-1)$, which tends to 1 as $l\to\infty$, but the error-correcting capabilities are asymptotically zero. $\square$

13.2 Bar product and Reed-Muller codes

As promised, we give the bar product construction, that turns two linear codes $\mathcal{X}_1, \mathcal{X}_2$ into a new one, $\mathcal{X}_1|\mathcal{X}_2$.

Example 13.5 *Given two linear codes of the same length N, with $\mathcal{X}_1 \subseteq \mathcal{X}_2$, define the bar product*

$$\mathcal{X}_1|\mathcal{X}_2 = \{(\mathbf{x}|\mathbf{x}+\mathbf{y}),\ \mathbf{x} \in \mathcal{X}_2,\ \mathbf{y} \in \mathcal{X}_1\}.$$

Lemma 13.6 *The bar product $\mathcal{X}_1|\mathcal{X}_2$ defines a new code of length $2N$, rank $k(\mathcal{X}_1) + k(\mathcal{X}_2)$ and minimum distance* $\min(\delta_1, 2\delta_2)$.

Proof The only result that requires proof is the minimum distance. Now, if $\mathbf{z} = \mathbf{x}|\mathbf{x}+\mathbf{y}$, and if $\mathbf{z} \neq \mathbf{0}$, then it cannot be that $\mathbf{x} = \mathbf{y} = \mathbf{0}$.

Hence we can consider three separate cases:

(i) $\mathbf{x} = \mathbf{0}$, and then $\mathbf{y} \neq \mathbf{0}$, so $w(\mathbf{z}) = w(\mathbf{y}) \geq \delta_1$.

(ii) $\mathbf{x} \neq \mathbf{0}$, and $\mathbf{y} = \mathbf{x} \in \mathcal{X}_1$, so $w(\mathbf{z}) = w(\mathbf{x}) \geq \delta_1$.

(iii) $\mathbf{x} \neq \mathbf{0}$ and $\mathbf{x}+\mathbf{y} \neq \mathbf{0}$. Hence, since $\mathbf{x}$ and $\mathbf{x}+\mathbf{y}$ are both in $\mathcal{X}_2$, then $w(\mathbf{z}) \geq 2\delta_2$. □

Using this, we can define the family of Reed-Muller codes:

Definition 13.7 *Define the $RM(e,0)$ to be the repetition code of length 2^e, define the $RM(e,e)$ code to be the trivial code of length 2^e with every word a codeword and then*

$$RM(e,r) = RM(e-1,r-1)|RM(e-1,r),$$

for $e \geq r$.

Lemma 13.8 *The $RM(e,r)$ code has length $n = 2^e$, rank $\binom{e}{0} + \binom{e}{1} + \ldots + \binom{e}{r}$ and minimum distance 2^{e-r}.*

The Reed-Muller codes are nested, in the sense that $RM(e,r) \subseteq RM(e,r+1)$.

Proof We work by induction. By definition, the only code with $e = 0$ is the $RM(0,0)$ code, consisting of the set $\{0,1\}$. So all the claims are true for all codes with $e = 0$.

Hence, we assume that the claims are true for all codes with $e \leq E-1$. Then, by Lemma 13.6, for $r \leq E-1$;

$$RM(E,r) = RM(E-1,r-1)|RM(E-1,r),$$

has length $2(2^{E-1})$, minimum distance $= \min(2^{E-r}, 2(2^{E-1-r})) = 2^{E-r}$, and rank:

$$\sum_{s=0}^{r-1}\binom{E-1}{s}+\sum_{s=0}^{r}\binom{E-1}{s}=\sum_{s=1}^{r}\binom{E-1}{s-1}+\sum_{s=1}^{r}\binom{E-1}{s}+1=\sum_{s=0}^{r}\binom{E}{s},$$

using Pascal's identity $\binom{n}{r} = \binom{n-1}{r-1} + \binom{n-1}{r}$.

The $RM(E,E)$ code has the right properties by definition.

The nesting property also follows by induction on e, since if $G_1 \subseteq G_2$ and $H_1 \subseteq H_2$ then $G_1|H_1 \subseteq G_2|H_2$. □

13.3 Dual codes

Notice that the code $\mathcal{X}$ has two equivalent descriptions, as the image of an $k \times N$ matrix G, or as the kernel of a $N \times (N-k)$ matrix H. This allows us to create another code, the dual code.

Definition 13.9 *Given a (N,k) code $\mathcal{X}$ with generator matrix G and parity check matrix H, the dual code $\mathcal{X}^\perp$ has generator matrix H^T and parity check matrix G^T, and is a $(N, N-k)$ code.*

Example 13.10 *The repetition code of Example 9.5 has generating matrix*

$$G = \begin{pmatrix} 1 & 1 & \dots & 1 \end{pmatrix}$$

with one choice of parity check matrix being

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

It is dual to the parity code of Example 9.5 which has generating matrix:

$$H^{\mathrm{T}} = \begin{pmatrix} 1 & 1 & 0 & \dots \\ 0 & 1 & 1 & \dots \\ 0 & 0 & 1 & \dots \\ & & \vdots & \end{pmatrix}$$

and parity check matrix

$$G^{\mathrm{T}} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ \vdots \end{pmatrix}.$$

13.4 Weight-enumerator polynomials

Many properties of the code $\mathcal{X}$ can be deduced from the weight-enumerator polynomial, which can be thought of as an averaged version of the minimum distance:

Definition 13.11 *For a code $\mathcal{X}$, with A_i words of weight i, define the weight-enumerator polynomial:*

$$W_{\mathcal{X}}(z) = \sum_{i=0}^{N} A_i z^i = \sum_{\mathbf{x} \in \mathcal{X}} z^{w(\mathbf{x})}.$$

The primal code and its dual are linked at the level of their weight-enumerator polynomials, via the following result:

Theorem 13.12 (MacWilliams Identity) *For a code $\mathcal{X}$ and its dual $\mathcal{X}^{\perp}$:*

$$(\sharp\mathcal{X}) W_{\mathcal{X}^{\perp}}(z) = (1+z)^n W_{\mathcal{X}}\left(\frac{1-z}{1+z}\right).$$

Proof First, notice that for a given $\mathbf{y}$:

$$\sum_{\mathbf{x} \in \mathcal{X}} (-1)^{\mathbf{x}\cdot\mathbf{y}} = (\sharp\mathcal{X}) \mathbb{I}(\mathbf{y} \in \mathcal{X}^{\perp}),$$

(this result is proved next, in Lemma 13.13)

Hence we can consider the function $g(\mathbf{x}) = \sum_{\mathbf{y} \in H_n} (-1)^{\mathbf{x}\cdot\mathbf{y}} z^{w(\mathbf{y})}$ in two ways. Firstly:

$$\begin{aligned} \sum_{\mathbf{x} \in \mathcal{X}} g(\mathbf{x}) &= \sum_{\mathbf{x} \in \mathcal{X}} \sum_{\mathbf{y} \in H_n} (-1)^{\mathbf{x}\cdot\mathbf{y}} z^{w(\mathbf{y})} = \sum_{\mathbf{y} \in H_n} z^{w(\mathbf{y})} \sum_{\mathbf{x} \in \mathcal{X}} (-1)^{\mathbf{x}\cdot\mathbf{y}} \\ &= \sum_{\mathbf{y} \in H_n} (\sharp\mathcal{X}) \mathbb{I}(\mathbf{y} \in \mathcal{X}^{\perp}) z^{w(\mathbf{y})} = (\sharp\mathcal{X}) W_{\mathcal{X}^{\perp}}(z) \end{aligned}$$

Alternatively:

$$
\begin{aligned}
g(\mathbf{x}) &= \sum_{\mathbf{y}\in H_n} ((-1)^{x_1} z)^{y_1} \dots ((-1)^{x_n} z)^{y_n} = \sum_{\mathbf{y}\in H_n} \prod_{i=1}^{n} ((-1)^{x_i} z)^{y_i} \\
&= \prod_{i=1}^{n} \left(\sum_{y=0}^{1} ((-1)^{x_i} z)^{y} \right) = \prod_{i=1}^{n} (1 + (-1)^{x_i} z) = (1+z)^n \left(\frac{1-z}{1+z} \right)^{w(\mathbf{x})}.
\end{aligned}
$$

$\square$

Lemma 13.13 *If $\mathcal{X} \subset \{0,1\}^n$ is a linear code and $\mathcal{X}^\perp = \{\mathbf{y} : \mathbf{y}\cdot\mathbf{x} = 0 \;\; \forall \mathbf{x} \in \mathcal{X}\}$ is its dual then:*

$$
\sum_{\mathbf{x}\in\mathcal{X}} (-1)^{\mathbf{y}\cdot\mathbf{x}} = \begin{cases} \#\mathcal{X} = 2^{\dim \mathcal{X}}, \textit{ if } \;\; \mathbf{y} \in \mathcal{X}^\perp, \\ 0, \textit{ otherwise.} \end{cases}
$$

Proof The case $\mathbf{y} \in \mathcal{X}^\perp$ is clear, so assume $\mathbf{y} \notin \mathcal{X}^\perp$.

If $\mathcal{X} = \{0,1\}^n$, then $\mathbf{y} \notin \mathcal{X}^\perp$ means that $\mathbf{y} \neq \mathbf{0}$. In this case pick j such that $y_j = 1$. In the sum, the summation over $x_j = 0, 1$ gives $(-1)^0 + (-1)^1 = 0$, hence the whole sum is 0.

If $\mathcal{X} \subset \{0,1\}^n$, with $\dim \mathcal{X} = k < n$, let G be a generating $k \times n$ matrix of $\mathcal{X}$. Then we can express $\mathbf{x}^T = \mathbf{u}^T G$, for some u, so that

$$
\sum_{\mathbf{x}\in\mathcal{X}} (-1)^{\mathbf{y}\cdot\mathbf{x}} = \sum_{\mathbf{u}\in\{0,1\}^k} (-1)^{\mathbf{u}^T G \mathbf{y}} = \sum_{\mathbf{u}\in\{0,1\}^k} (-1)^{\mathbf{u}\cdot\mathbf{G}^{\mathbf{T}}\mathbf{y}},
$$

which is zero as in the previous paragraph since $G\mathbf{y} \neq 0$. (We know this since $\mathbf{y} \in \mathcal{X}^\perp \Leftrightarrow \mathbf{y}^T H^\perp = 0 \Leftrightarrow \mathbf{y}^T G^T = 0 \Leftrightarrow G\mathbf{y} = 0$. $\square$

As a check, it is worthwhile listing the codewords of the dual Hamming $(7,4)$ code and showing that it satisfies the MacWilliams Identity.

Lecture 14: Cyclic codes

14.1 Definitions and generators

Next we study the cyclic codes, a class of codes which contains the Hamming codes.

Definition 14.1 *Given a word* $\mathbf{a} = a_0 \ldots a_{N-1}$*, we define its cyclic shift* $\pi\mathbf{a}$ *to be the word* $a_{N-1}a_0 \ldots a_{N-2}$*.*

A linear code $\mathcal{X}$ *is called cyclic if the cyclic shift of each codeword is again a codeword.*

Lemma 14.2 *A code* $\mathcal{X}$ *is cyclic if and only if, for any vector* $\mathbf{a}$ *from a basis of* $\mathcal{X}$*,* $\pi\mathbf{a} \in \mathcal{X}$*.*

Proof Each vector $\mathbf{u} \in \mathcal{X}$ is a sum of vectors from the basis. But $\pi(\mathbf{u}+\mathbf{v}) = \pi\mathbf{u} + \pi\mathbf{v}$, hence the result. □

One way to form a cyclic code as follows: take a word $\mathbf{a}$, form cyclic shifts $\pi\mathbf{a}, \pi^2\mathbf{a}, \ldots$ and take all linear combinations of these vectors. Such a construction allows you to build a code from a single word. We shall see how all the properties of the code may be inferred from the properties of word $\mathbf{a}$.

We shall see that every cyclic code may be obtained in such a way: the word $\mathbf{a}$ is called a generator of a cyclic code.

14.2 Polynomial algebra

We will assume the theory of polynomial algebra over finite fields.

Specifically, in the binary case, we assume that we can add and multiply polynomials, taking coefficients $\bmod 2$. Further, we can carry out division of one polynomial into another (leaving a quotient and remainder, which are both polynomials).

Having defined this division, we can say that two polynomials, $f_1(X)$ and $f_2(X)$ are called equivalent modulo $h(X)$ if their remainders, after division by $h(X)$, coincide. Addition and multiplication of polynomials respect this equivalence.

Every binary linear code $\mathcal{X}_N$ corresponds to a set of polynomials of degree $N-1$ with coefficients in $\{0,1\}$, which is closed under addition $\mod 2$.

That is, relabelling the indices to run from 0 to $N-1$, we write $a(X) = a_0 + a_1 X \cdots + a_{N-1}X^{N-1}$ for the word $\mathbf{a} = a_0 \ldots a_{N-1}$. We write $a(X) \in \mathcal{X}$ whenever $a^{(N)} = a_0 \ldots a_{N-1}$ belongs to $\mathcal{X}$.

Notice that the sum of codewords corresponds to summing polynomials, and taking coefficients $\mod 2$. That is, the coefficient of X^i in $a(X)+b(X)$ is a_i+b_i, as we would expect on taking the pointwise sum. We now see how to translate results about shifts of codewords into the language of polynomials.

Lemma 14.3 *If word* $\mathbf{a}$ *corresponds to a polynomial* $a(X)$ *then word* $\pi\mathbf{a}$ *corresponds to* $Xa(X) \mod (1+X^N)$.

Proof Write

$$\begin{aligned} Xa(X) &= a_0X + a_1X^2 + \cdots + a_{N-2}X^{N-1} + a_{N-1}X^N \\ &= a_{N-1} + a_0X + a_1X^2 + \cdots + a_{N-2}X^{N-1} + a_{N-1}X^N + a_{N-1} \\ &= a_{N-1} + a_0X + a_1X^2 + \cdots + a_{N-2}X^{N-1} + a_{N-1}(X^N+1), \end{aligned}$$

which means that the polynomial

$$a_{N-1} + a_0X + \cdots + a_{N-2}X^{N-1}$$

corresponding to $\pi\mathbf{a}$ equals $Xa(X) \mod (1+X^N)$. $\square$

Obviously word $\pi^2\mathbf{a}$ corresponds to $X^2a(X) \mod (1+X^N)$ and so on. We will write $\star$ for the operation of multiplication of polynomials $\mod (1+X^N)$,

$$a(X) \star b(X) = a(X)b(X) \mod (1+X^N).$$

A corollary of these facts is the following theorem.

Theorem 14.4 *A cyclic code $\mathcal{X}$ contains, with each pair of polynomials $a(X)$ and $b(X)$, the sum $a(X)+b(X)$ and any polynomial $v(X)\star a(X)$.*

Proof The sum $a(X)+b(X)\in\mathcal{X}$ because of linearity. Each polynomial $X^k a(X) \mod (1+X^N)$ corresponds to $\pi^k a$ and hence belongs to $\mathcal{X}$. Taking $v(X)=v_0+v_1X+\cdots v_{N-1}X^{N-1}$ then linearity gives that

$$v(X)a(X) \mod (1+X^N) = \sum_{i=0}^{N-1} v_i X^i a(X) \mod (1+X^N)$$

the LHS belongs to $\mathcal{X}$. $\square$

In other words, the polynomials of degree $\leq N-1$, with $\star$-multiplication and the usual addition, form a commutative ring, and cyclic codes are precisely the ideals of this ring.

Theorem 14.5 *Let $c(X)=\sum_{i=0}^{N-k} c_i X^i$ be a non-zero polynomial of minimum degree in a cyclic code $\mathcal{X}$. Then*

1. *$c(X)$ is the unique polynomial of minimal degree.*
2. *$a(X)\in\mathcal{X}$ if and only if $a(X)=v(X)\star c(X)$ for some polynomial $v(x)$ of degree $<k$ (that is, $c(X)$ is a divisor of every polynomial from $\mathcal{X}$).*
3. *The codewords corresponding to $c(X),\dots,X^{k-1}c(X)$, form a basis in $\mathcal{X}$; they are cyclic shifts of word $c_0\dots c_{N-k}0\dots 0$, so the code $\mathcal{X}$ has rank k.*

Proof 1. Suppose $c'(X)=\sum_{i=0}^{N-k} c'_i X_i$ is another polynomial of minimal degree $N-k$ in $\mathcal{X}$. Then $c_{N-k}=c'_{N-k}=1$, and hence deg $(c'(X)+c(X))<N-k$. But as $N-k$ is the minimal degree, $c'(X)+c(X)$ must equal zero, so $c(X)=c'(X)$. Hence, $c(X)$ is unique.

2. We know that each polynomial $a(X)\in\mathcal{X}$ has degree $\geq$ deg $c(X)$. By the Division Algorithm,

$$a(X)=v(X)c(X)+r(X).$$

Here, we must have

$$\deg v(X)<k \text{ and } \deg r(X)<\deg c(X)=N-k.$$

But then $v(X)c(X)$ belongs to $\mathcal{X}$ owing to Theorem 14.4 (as $v(X)c(X)$ has degree $\leq N-1$, it coincides with $v(X) \star c(X)$). Hence,

$$r(X) = a(X) + v(X)c(X) \in \mathcal{X}$$

by linearity. But $c(X)$ is the polynomial in $\mathcal{X}$ of minimum degree, so $r(X) = 0$.

3. By 2., each polynomial $a(X) \in \mathcal{X}$ has the form

$$c(X)v(X) = \sum_{i=1}^{r} v_i X^i c(X), \quad r < k.$$

Hence, each polynomial $a(X) \in \mathcal{X}$ is a linear combination of polynomials $c(X), Xc(X), \ldots, X^{k-1}c(X)$. These polynomials belong to $\mathcal{X}$, have distinct degrees and hence are linearly independent. Therefore words $\mathbf{c}, \pi\mathbf{c}, \ldots, \pi^{k-1}\mathbf{c}$, corresponding to $c(X), Xc(X), \ldots, X^{k-1}c(X)$, form a basis in $\mathcal{X}$. $\square$

14.3 Generators

Definition 14.6 *The unique polynomial $c(X)$ of minimal degree in $\mathcal{X}$ is called a generator of the cyclic code $\mathcal{X}$.*

Every word in the cyclic code is obtained from the generator, by cyclic shifts and linear combinations. There may be other polynomials that generate $\mathcal{X}$ in this sense, however the minimum degree polynomial is unique.

Theorem 14.7 *A polynomial $c(X)$ of degree $\leq N-1$ is the generator of a cyclic linear code of length N if and only if $c(X)$ divides $1 + X^N$. That is,*

$$1 + X^N = h(X)c(X)$$

for some polynomial $h(X)$ (of degree $N - 1 - \deg c(X)$).

Proof (**Only if**) By the Division Algorithm, there exists some r with $\deg r(X) < \deg c(X)$ such that

$$1 + X^N = h(X)c(X) + r(X),$$

That is,

$$r(X) = h(X)c(X) \mod (1 + X^N),$$

so by Theorem 14.5, $r(X)$ belongs to the cyclic code $\mathcal{X}$ generated by $c(X)$. However, $c(X)$ is the polynomial of minimum degree in $\mathcal{X}$, so we deduce that $r(X) = 0$ and $1 + X^N = h(X)c(X)$.

(**If**) Suppose that $1+X^N = v(X)c(X)$, $\deg v(X) = N - \deg c(X)$. Consider the set $\{a(X) = c(X) \star u(X)\}$ (the principal ideal in the $\star$-multiplication polynomial ring corresponding to $c(X)$). This set contains $c(X), Xc(X), \dots, X^{k-1}c(X)$ where $k = \deg v(X)$. Since $X^k c(X) = 1 + X^N + \sum_{j=0}^{k-1} v_j X^j c(X)$, it also lies in the set, so c generates a linear code $\mathcal{X}$. □

Corollary 14.8 *The cyclic codes of length N are in one-to-one correspondence with the divisors of polynomial $1 + X^N$.*

14.4 Examples of cyclic codes

Example 14.9 *Polynomial $1 + X^N$ always has two divisors*

$$1 + X^N = (1 + X)(1 + X + \ldots + X^{N-1})$$

Polynomial $1+X$ generates the parity-check code $\{\mathbf{x} = x_1 \ldots x_N : \sum_i x_i = 0\}$, whereas $1 + X + \ldots + X^{N-1}$ the repetition code $\{\mathbf{0}, \mathbf{1}\} = \{00\ldots0, 11\ldots1\}$.

Example 14.10 *We see that it is important to know the factorisation of polynomial $1 + X^N$ into irreducible factors. For $N = 1, 3, 5, 11, 13, 19$ the only factors are the trivial ones of Example 14.9. In general, we can factor:*

N	$(1 + X^N)/(1 + X)$
7	$(1 + X + X^3)(1 + X^2 + X^3)$
9	$(1 + X + X^2)(1 + X^3 + X^6)$
15	$(1 + X + X^2)(1 + X + X^4)(1 + X^3 + X^4)(1 + X + X^2 + X^3 + X^4)$
17	$(1 + X^3 + X^4 + X^5 + X^8)(1 + X + X^2 + X^4 + X^6 + X^7 + X^8)$
21	$(1 + X + X^2)(1 + X + X^3)(1 + X^2 + X^3)$
	$\times(1 + X + X^2 + X^4 + X^6)(1 + X^2 + X^4 + X^5 + X^6)$
23	$(1 + X + X^5 + X^6 + X^7 + X^9 + X^{11})$
	$\times(1 + X^2 + X^4 + X^5 + X^6 + X^{10} + X^{11})$

For N even, $1+X^N$ can have multiple roots. For example, $1+X^{2^l} = (1+X)^{2^l}$.

Example 14.11 (Hamming (7,4) code) *Take the generating and check matrices in lexicographic form. If you re-order the digits $x_4x_7x_5x_3x_2x_6x_1$ (which leads to an equivalent code) then the rows of the generating matrix become subsequent cyclic shifts of each other:*

$$G^{\mathrm{cycl}} = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix}$$

and the cyclic shift of the last row is again in the code:

$$\begin{aligned} \pi(0\,0\,0\,1\,1\,0\,1) &= (1\,0\,0\,0\,1\,1\,0) \\ &= (1\,1\,0\,1\,0\,0\,0) + (0\,1\,1\,0\,1\,0\,0) + (0\,0\,1\,1\,0\,1\,0) \end{aligned}$$

By Lemma 14.2, the code is cyclic. By Theorem 14.5 the generating polynomial $c(X)$ corresponds to 1101, so the generator $c(X) = 1 + X + X^3$.

A similar argument can be used to show that an equivalent cyclic code is obtained from the word $1011 \sim 1 + X^2 + X^3$. There is no contradiction: it was not claimed that the polynomial ideal of a cyclic code is the principal ideal of a unique element. See Definition 14.6.

Example 14.12 *For $N = 23$, observe that the two polynomials of degree 11 are reciprocal. [Polynomials $a(X)$ and $\widetilde{a}(X)$ of degree k are called reciprocal $\widetilde{a}(X) = X^k a\left(X^{-1}\right)$; they clearly generate equivalent codes.] The corresponding cyclic codes $\mathcal{X}_i$, $i = 1, 2$, are $(23, 12)$, they have minimal distance 7 and correct 3 errors. These are the perfect Golay codes (see Section 16.2 for more details).*

14.5 Dual cyclic codes

It turns out that the dual of a cyclic code $\mathcal{X}$ is also cyclic. Since we know that the generator $g(X)|X^N+1$, if we write the quotient $h(X) = (X^N+1)/g(X)$, it is tempting to hypothesise that $\mathcal{X}^\perp$ should be generated by $h(X)$. This is nearly right!

Lemma 14.13 *Write $k(X)$ for the reversal of $h(X)$, that is take coefficients $k_i = h_{r-i}$, where r is the degree of h. Then $k(X)$ generates $\mathcal{X}^\perp$.*

Proof We know that $g(X)h(X) = X^N + 1$, so comparing coefficients of X^s, we know that for all s:

$$\sum_i g_i h_{s-i} = 0. \tag{14.1}$$

We want it to be the case that any shift of k be orthogonal to g, that is we want that for all t:

$$\sum_j g_j k_{t+j} = 0. \tag{14.2}$$

Notice that for the coefficients of polynomial multiplication Equation (14.1), one goes up and the other goes down, whereas for the coefficients of the dot product Equation (14.2), the coeffients go up together. Thus, defining k to be the reversal of h, the result we know (Equation (14.1)) implies the result we want (Equation (14.2)), and the Lemma is proved. $\square$

Lecture 15: Codes and finite fields. BCH codes

15.1 Introduction to finite fields

This lecture requires some knowledge of the theory of finite fields – if necessary, consult the Part 1B Groups, Rings and Fields notes. We will show how to construct the BCH (Bose–Chaudhuri–Hocquengham) codes, a family of codes which can be designed to have any length and any minimum distance.

We use the fact that the words of length l form a finite field, by thinking of the words as polynomials. We take a polynomial $h(X)$ of degree l and consider the polynomial ring (that is, the operations of addition and multiplication) $\mod h(X)$. Some useful facts are as follows.

1. The polynomial ring $\mod h(X)$ is a field if and only if $h(X)$ is irreducible, and any two irreducible polynomials of the same degree generate isomorphic fields. There exist irreducible polynomials of any degree.
2. The number of elements in the field equals 2^l, and it is denoted by $\mathbb{F}_{2^l}$.
3. Any subfield of $\mathbb{F}_{2^l}$ is of the form $\mathbb{F}_{2^m}$ with $m < l$; we obtain a series of finite fields
$$\{0,1\} = \mathbb{F}_2 \to \mathbb{F}_{2^2} \to \mathbb{F}_{2^3} \to \dots,$$
where $\to$ means an isomorphic embedding.
4. Any field $\mathbb{F}_{2^l}$ is at the same time a linear vector space of dimension l over $\mathbb{F}_2$.
5. Another way to understand $\mathbb{F}_{2^l}$ is to note that $\mathbb{F}^*_{2^l}$, the multiplicative group of the non-zero elements of $\mathbb{F}_{2^l}$, is isomorphic to $\mathbb{Z}_{2^l-1}$, the cyclic group of order $2^l - 1$.

Example 15.1 *Field* $\mathbb{F}_{2^2}$. *There is a unique irreducible polynomial of degree 2:* $X^2 + X + 1$. *The corresponding field is described as follows*

element of polynomial field	*word*	*element of* $\mathbb{Z}_3$
$\mathbf{0}$	00	$-$
$\mathbf{1}$	10	ω
X	01	ω^2
$1 + X$	11	ω^3.

Example 15.2 *Field* $\mathbb{F}_{2^3}$. *There are two irreducible polynomials of degree 3:* $1 + X + X^3$ *and* $1 + X^2 + X^3$. *The corresponding fields are isomorphic and have the multiplicative group* $\mathbb{Z}_7$. *If you take polynomial* $X^3 + X + 1$ *then the field is described as follows*

element of polynomial field	*word*	*element of* $\mathbb{Z}_7$
$\mathbf{0}$	000	$-$
$\mathbf{1}$	100	e
X	010	ω
X^2	001	ω^2
$1 + X$	110	ω^3
$X + X^2$	011	ω^4
$1 + X + X^2$	111	ω^5
$1 + X^2$	101	ω^6

The first hint of a relationship between codes and finite fields comes from the following remark. When we write ω, we really mean the binary vector 010, and so on.

Remark 15.3 *From the* $\mathbb{Z}_7$*-point of view, the check matrix of the Hamming* $(7, 4)$ *code is*

$$H_{\text{Ham}} = \begin{pmatrix} e \\ \omega \\ \omega^2 \\ \dots \\ \omega^6 \end{pmatrix} .$$

15.2 Roots of generating polynomials

A convenient way to describe a cyclic code of length N and rank k is to list all the roots $\beta_1, \ldots, \beta_{N-k}$, of its generating polynomial $c(X) = \prod_j \left(X - \beta_j\right)$.

Since the generating polynomial divides $X^N + 1$, the β must be Nth roots of unity. We will assume that N is odd, so that the roots are distinct (if not, notice that since $X^N + 1 = (X - \beta)^2 g(X)$, so taking a derivative

$$NX^{N-1} = (X - \beta)^2 g'(X),$$

giving a contradiction).

Definition 15.4 *The β belong to some field $\mathbb{F}_{2^d}$. The minimal field containing them (that is, the minimal d) is called the splitting field of polynomial $c(X)$, or the splitting field for $\beta_1, \ldots, \beta_{N-k}$.*

By definition, the generating polynomial $c(X)$ vanishes at each β_i. The same condition holds for any polynomial $a(X) \in \mathcal{X}$, since $\mathcal{X} = \{a(X) = c(X) \star u(X)$ for some $u(X)\}$.

So, a cyclic code $\mathcal{X}$ of length N may be described as

$$\mathcal{X} = \{a(X) : \deg a(X) \leq N - 1 \text{ with } a(\beta_j) = 0,\ 1 \leq j \leq N - k\}.$$

In other words, $\mathcal{X}$ consists of the vectors $\mathbf{a} = a_0 \ldots x_{N-1}$ for which $a(\beta_i) = a_0 + a_1\beta_i + \ldots a_{N-1}\beta_i^{N-1} = \langle \mathbf{a} \cdot \boldsymbol{\beta}_i \rangle = 0$, where $\boldsymbol{\beta}_i = (\beta_i^0, \ldots, \beta_i^u)$. Therefore,

Theorem 15.5 *An $N \times (N - K)d$ matrix H given by*

$$H = \begin{pmatrix} \beta_1^0 & \beta_2^0 & \ldots \beta_{N-k}^0 \\ \beta_1^1 & \beta_2^1 & \ldots \beta_{N-k}^1 \\ \beta_1^2 & \beta_2^2 & \ldots \beta_{N-k}^2 \\ & \vdots & \\ \beta_1^{N-1} & \beta_2^{N-1} & \ldots \beta_{N-k}^{N-1} \end{pmatrix} \tag{15.1}$$

is a parity-check matrix of a cyclic code of length N with zeroes $\{\beta_1, \ldots, \beta_{N-k}\}$.

Proof Row s of H is a concatenation of $N - k$ binary d-dimensional vectors $\beta_1^s, \ldots, \beta_{N-k}^s$. Condition $\mathbf{a}H = 0$ means that $a(\beta_j) = 0,\ 1 \leq j \leq N - k$, which occurs if and only if $\mathbf{a} \in \mathcal{X}$. $\square$

In general, having $(N-k)d$ columns is too many, but we can remove linear dependencies. The right number of linearly independent columns is $N-k$.

15.3 Defining sets and BCH codes

Conversely, if $\{\beta_1, \ldots, \beta_u\}$ are Nth roots of unity, with the splitting field $\mathbb{F}_{2^d}$, then matrix (15.1) determines a cyclic code.

Again, the number of columns is excessive; one way to remove excessive columns (and identify the generator $c(X)$) is to go back to the decomposition of $1 + X^N$ into irreducible factors $m_j(X)$. Then each β_i is a root of one of $m_j(X)$.

Now, if you pick a single root of an irreducible factor $m(X)$ of $1+X^N$ then all its roots are automatically included. Hence, we might as well take one root from each irreducible factor. That is, we can reorder the m_j so that,

$$c(X) = \prod_{1 \leq j \leq t} m_j(X),$$

choose β_i to be some root of m_i, and write

$$\mathcal{X} = \{a(X) : \deg a \leq N : a(\beta_1) = a(\beta_2) = \ldots = a(\beta_t)\}.$$

Definition 15.6 *We say that such a set $\{\beta_1, \ldots \beta_t\}$ is the defining set of a code.*

Example 15.7 *The Hamming code corresponds to a defining set $\{\omega\}$.*

Lemma 15.8 *If f is a polynomial with coefficients in $\{0, 1\}$ which has root α, then α^2 is also a root of f.*

Proof

$$\begin{aligned} 0 &= (f(\alpha))^2 = \sum_{i,j} f_i f_j \alpha^i \alpha^j \\ &= \sum_i f_i^2 (\alpha^2)^i + 2 \sum_{i<j} f_i f_j \alpha^{i+j} \\ &= \sum_i f_i (\alpha^2)^i \end{aligned}$$

as required. $\square$

Hence, notice that for the Hamming code of length 7, the possible defining sets are $\{\omega\}$, $\{\omega^2\}$ and $\{\omega^4\}$, corresponding to the roots of $X^3 + X + 1$. Whichever defining set we choose, the code has the so-called BCH property, as follows:

Theorem 15.9 (BCH theorem) *If for some integer t (often $t = 1$) and some α, the $\{\alpha^t, \alpha^{t+1}, \dots \alpha^{t+\delta-2}\}$ are all roots, then we say the code is a BCH code of design distance δ. The minimum distance of the code is at least δ.*

Proof The key observation is the fact that the Vandermonde determinant is non-singular, that is:

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

The determinant is clearly zero if any $x_i = x_j$ so all these terms must be factors. Further, all terms in the determinant are of order $0+1+\dots+(n-1) = n(n-1)/2$, so there are no other non-constant factors on the RHS. Comparing coefficients of $x_2 x_3^2 \dots x_n^{n-1}$ (the terms on the LHS come from the diagonal only, the terms from the RHS must come from taking x_j from each product.

Now, suppose that we take a parity check matrix H of the following form

$$H = \begin{pmatrix} 1 & 1 & \dots & 1 \\ \beta^t & \beta^{t+1} & \dots & \beta^{t+\delta-2} \\ \beta^{2t} & \beta^{2(t+1)} & \dots & \beta^{2(t+\delta-2)} \\ & \vdots & & \\ \beta^{(N-1)t} & \beta^{(N-1)(t+1)} & \dots & \beta^{(N-1)(t+\delta-2)} \end{pmatrix}$$

By Remark 13.2, if the minimum distance of the code was $\leq \delta - 1$ then there would exist a singular $(\delta - 1) \times (\delta - 1)$ submatrix. If we consider selecting $\delta - 1$ rows of H, say rows $i_1, i_2, \dots i_{\delta-1}$, we will be left with a matrix of the form

$$\begin{pmatrix} \beta^{i_1 t} & \beta^{i_1(t+1)} & \dots & \beta^{i_1(t+\delta-2)} \\ \beta^{i_2 t} & \beta^{i_2(t+1)} & \dots & \beta^{i_2(t+\delta-2)} \\ & \vdots & & \\ \beta^{i_{\delta-1} t} & \beta^{i_{\delta-1}(t+1)} & \dots & \beta^{i_{\delta-1}(t+\delta-2)} \end{pmatrix}$$

Now, taking out factors of $\beta^{i_j t}$ from row j, the determinant of this is

$$\beta^{(i_1+i_2+\dots i_{\delta-1})t} \det \begin{pmatrix} 1 & \beta^{i_1} & \dots & \beta^{i_1(\delta-2)} \\ 1 & \beta^{i_2} & \dots & \beta^{i_2(\delta-2)} \\ & \vdots & & \\ 1 & \beta^{i_{\delta-1}} & \dots & \beta^{i_{\delta-1}(\delta-2)} \end{pmatrix} \neq 0.$$

(by the Vandermonde theorem, since i_j are distinct). $\square$

This gives a wide class of codes corecting any given number of errors. Indeed, for many given N, k, the code with the largest distance is a BCH code. However, the BCH codes are asymptotically bad: it can be shown that for any sequence of BCH codes of length $N \to \infty$, either k/N or $\delta/N \to 0$. In other words, they lie at the bottom of the diagram of Figure 11.1.

Example 15.10 *Take $l = 4$ and $N = 15$, and consider multiplication determined by an irreducible polynomial $1 + X + X^4$. The corresponding field*

$GF(2^4)$ is described as follows

element of polynomial field	*word*	*element of* $\mathbb{Z}_{15}$
$\mathbf{0}$	0000	$-$
$\mathbf{1}$	1000	e
X	0100	ω
X^2	0010	ω^2
X^3	0001	ω^3
$1+X$	1100	ω^4
$X+X^2$	0110	ω^5
X^2+X^3	0011	ω^6
$1+X+X^3$	1101	ω^7
$1+X^2$	1010	ω^8
$X+X^3$	0101	ω^9
$1+X+X^2$	1110	ω^{10}
$X+X^2+X^3$	0111	ω^{11}
$1+X+X^2+X^3$	1111	ω^{12}
$1+X^2+X^3$	1011	ω^{13}
$1+X^3$	1001	ω^{14}

Then write matrix $\widetilde{H}$:

$$\widetilde{H} = \begin{pmatrix} 1000 & 1000 \\ 0100 & 0001 \\ 0010 & 0011 \\ 0001 & 0101 \\ 1100 & 1111 \\ 0110 & 1000 \\ 0011 & 0001 \\ 1101 & 0011 \\ 1010 & 0101 \\ 0101 & 1111 \\ 1110 & 1000 \\ 0111 & 0001 \\ 1111 & 0011 \\ 1011 & 0101 \\ 1001 & 1111 \end{pmatrix} = \begin{pmatrix} e & e \\ \omega & \omega^3 \\ \omega^2 & \omega^6 \\ \omega^3 & \omega^9 \\ \omega^4 & \omega^{12} \\ \omega^5 & e \\ \omega^6 & \omega^3 \\ \omega^7 & \omega^6 \\ \omega^8 & \omega^9 \\ \omega^9 & \omega^{12} \\ \omega^{10} & e \\ \omega^{11} & \omega^3 \\ \omega^{12} & \omega^6 \\ \omega^{13} & \omega^9 \\ \omega^{14} & \omega^{12} \end{pmatrix}. \tag{15.2}$$

Theorem 15.11 *The linear code with parity check matrix $\widetilde{H}$ given by (15.2) is a 2-error correcting $(15, 7)$ code.*

Proof The rank 7 is due to the linear independence of the columns of $\widetilde{H}$. This is a tedious although staightforward calculation; we omit it because of lack of time.

We can check that the code corrects up to two errors, using Theorem 15.9. Note that since ω and ω^3 are roots, then so are $\{\omega, \omega^2, \omega^3, \omega^4\}$. $\square$

Information and Coding Michaelmas 2006/Michaelmas 2007 O.Johnson/Y. Suhov

Lecture 16: Golay codes

16.1 Self-dual codes

In this lecture, we deduce many properties of the Golay code, which is a perfect code of length 23, rank 12 and minimum distance 7. First of all, notice that is feasible that such a code could exist. Since $v_{23}(3) = \binom{23}{0} + \binom{23}{1} + \binom{23}{2} + \binom{23}{3} = 1 + 23 + 253 + 1771 = 2048 = 2^{11}$, the total volume of 2^{12} balls is 2^{23}. We "just" need to find a way to pack this many balls into the 23 dimensional Hamming space.

We start in an abstract way, by considering two important categories of codes, the so-called self-dual and self-orthogonal codes.

Definition 16.1 *A code $\mathcal{X}$ is said to be self-dual if $\mathcal{X} = \mathcal{X}^\perp$, that is if its generator matrix G and parity check matrix H satisfy $G^T = H$.*

A code $\mathcal{X}$ is said to be self-orthogonal if $\mathbf{x} \cdot \mathbf{y} = 0$ for any $\mathbf{x}, \mathbf{y} \in \mathcal{X}$.

Lemma 16.2 *A self-dual code (i) has even length (ii) is self-orthogonal and (iii) contains no words of odd weight.*

Proof The first part follows simply by the fact that if a code $\mathcal{X}$ has rank k, then its dual code $\mathcal{X}^\perp$ has rank $n - k$ (by the rank-nullity theorem). Hence, if they are the same code, then it must be the case that $k = n - k$, and so n is even (and $k = n/2$).

The second part follows since any word $\mathbf{x}$ in the primal code $\mathcal{X}$ is perpendicular to any word $\mathbf{y}$ in the dual code $\mathcal{X}^\perp$ (in the sense of the dot product $\cdot$ of Definition 9.1 - that is for any $\mathbf{x} \in \mathcal{X}, \mathbf{y} \in \mathcal{X}^\perp$, $\mathbf{x} \cdot \mathbf{y} = 0$). That is because $\mathbf{y}$ must be a linear combination $\sum_i a_i \mathbf{h}_i$ of columns $\mathbf{h}_i$ of H, and $\mathbf{x} \cdot \mathbf{h_i} = 0$ for each i. Hence, if $\mathcal{X}$ and $\mathcal{X}^\perp$ coincide then for any $\mathbf{x}, \mathbf{y} \in \mathcal{X}$, the $\mathbf{x} \cdot \mathbf{y} = 0$.

In particular, by self-orthogonality $\mathbf{x} \cdot \mathbf{x} = 0$, so $\mathbf{x}$ has even weight. □

Note that a self-orthogonal code $\mathcal{X}$ has $\mathcal{X} \subseteq \mathcal{X}^\perp$, so if a self-orthogonal code of length n has dimension $n/2$, then it is self-dual.

Lemma 16.3 *If a self-orthogonal code $\mathcal{X}$ has a generator matrix in which all the rows have weight* $\equiv 0 \mod 4$*, then all words in the code have weight* $\equiv 0 \mod 4$.

Proof The proof proceeds by induction on the number of generators added. That is, if $\mathbf{g}$ is a generator, where $4|w(\mathbf{g})$ and $\mathbf{c}$ is any word with $4|w(\mathbf{c})$, then by self-orthogonality,

$$w(\mathbf{g}+\mathbf{c}) = w(\mathbf{g}) + w(\mathbf{c}) - 2w(\text{intersection}).$$

Here intersection refers to the set of places where $\mathbf{c}$ and $\mathbf{g}$ are both 1. But $\mathbf{c} \cdot \mathbf{g} = 0$, so the intersection has even size, and the result follows. □

16.2 Parameters of the Golay code

We can now combine a variety of results to obtain the properties of the Golay code. Recall from Example 14.10 that $X^{23}+1$ factors as $(X+1)g_0(X)g_1(X)$, where g_0 and g_1 are irreducible polynomials of degree 11. In fact, it will not matter which polynomial we take as generator, but for definiteness we make the following choice.

Definition 16.4 *The Golay code $\mathcal{G}$ is a code of length 23 with generator* $g_0(X) = 1 + X + X^5 + X^6 + X^7 + X^9 + X^{11}$.

For completeness, note that $g_1(X) = 1 + X^2 + X^4 + X^5 + X^6 + X^{10} + X^{11}$, which is the reversal of $g_0(X)$. That is, writing $g_0(X) = \sum_i g_0^{(i)} X^i$, and $g_1(X) = \sum_j g_1^{(j)} X^j$, the

$$g_1^{(j)} = g_0^{(11-j)}. \tag{16.1}$$

Theorem 16.5 *The Golay code $\mathcal{G}$ is perfect, with length 23, and minimum distance* 7.

We prove this result by considering the extended Golay code $\mathcal{G}^+$. That is a code of length 24, with a parity bit added to the rows of $\mathcal{G}$. The key is to notice that the Golay code $\mathcal{G}$ has rank 12 (size 4096), so if we can show it has minimum distance 7, that will be enough. In fact, we show an even stronger result, as follows:

Theorem 16.6 *The extended Golay code $\mathcal{G}^+$ is self-dual and has weight-enumerator polynomial* $1 + 759z^8 + 2576z^{12} + 759z^{16} + z^{24}$.

Proof First, notice that the Golay code is a BCH code with design distance 5. This follows by Lemma 15.8, since writing α for one root of g_0, the roots are $\{\alpha, \alpha^2, \alpha^4, \alpha^8, \alpha^{16}, \alpha^9, \alpha^{18}, \alpha^{13}, \alpha^3, \alpha^6, \alpha^{12}\}$.

This means by Theorem 15.9 that it has minimum distance at least 5. Hence, so does the extended Golay code (in fact the minimum distance is at least 6, by Example 9.8).

Next, we show that $\mathcal{G}^+$ is self-dual. As previously remarked, we only need show that it is self-orthogonal, since it has rank 12 and length 24. Since the generators of $\mathcal{G}$ are $g_0, Xg_0, X^2g_0, \ldots$, all of weight 7, the generators of $\mathcal{G}^+$ are $(g_0|1), (Xg_0|1), (X^2g_0|1), \ldots$. Now for any i, j, the dot-product of two generators of $\mathcal{G}$ is

$$\begin{aligned}
(X^i g_0) \cdot (X^j g_0) &= \sum_r g_0^{(i+r)} g_0^{(j+r)} \\
&= \sum_r g_0^{(i+r)} g_1^{(11-j-r)} \\
&= \text{coefficient of } X^{11+i-j} \text{ in } g_0(X) \times g_1(X) \\
&= \text{coefficient of } X^{11+i-j} \text{ in } 1 + X + \ldots + X^{22} \\
&= 1
\end{aligned}$$

(by Equation (16.1), understanding these superscripts $\bmod 23$). Hence, the the dot-product of two generators of $\mathcal{G}^+$ is zero.

Since the code $\mathcal{G}^+$ is self-dual, and all its generators are of weight 8, we can appeal to Lemma 16.3, to see it can only have words of weight $0, 4, 8, 12, 16, 20, 24$. In fact, by the remark above, the minimum distance is at least 6, so there can only be words of weight $0, 8, 12, 16, 24$.

Next, we can consider the weight-enumerator polynomial of $\mathcal{G}^+$. Since $\mathcal{G}^+$ contains the word $\mathbf{1}$ of full weight (consisting of 24 1s), the weight-enumerator is symmetric (for any codeword $\mathbf{x}$ of weight n, there is a corresponding codeword $\mathbf{1} + \mathbf{x}$ of weight $24 - n$). That is the polynomial is of the form:

$$W_{\mathcal{G}^+}(z) = 1 + Pz^8 + Qz^{12} + Pz^{16} + z^{24},$$

for some P, Q such that $2 + 2P + Q = 4096$. Now, we can apply the McWilliams identity, Theorem 13.12, using the fact that the code is self-dual, so

$$W_{\mathcal{G}^+}(z) = W_{\mathcal{G}^{+\perp}} = \frac{(1+z)^{24}}{4096} W_{\mathcal{G}^+}\left(\frac{1-z}{1+z}\right).$$

That is:

$$\begin{aligned}
&4096\left(1 + Pz^8 + Qz^{12} + Pz^{16} + z^{24}\right)\\
&= (1+z)^{24} + P(1+z)^{16}(1-z)^8 + Q(1+z)^{12}(1-z)^{12}\\
&\quad + P(1+z)^8(1-z)^{16} + (1-z)^{24},
\end{aligned}$$

and comparing coefficients of z^2, we can see that

$$\begin{aligned}
0 &= 2\binom{24}{2} + 2P\left(\binom{8}{2} - 8\right) - 12Q\\
\Rightarrow 0 &= 552 + 40P - 12Q,
\end{aligned}$$

so we deduce that $P = 759$ and $Q = 2576$ as required. $\square$

We can deduce that the original Golay code $\mathcal{G}$ has weight enumerator:

$$1 + 759\left(\frac{1}{3}z^7 + \frac{2}{3}z^8\right) + 2576\left(\frac{1}{2}z^{11} + \frac{1}{2}z^{12}\right) + 759\left(\frac{2}{3}z^{15} + \frac{1}{3}z^{16}\right) + z^{23}.$$

This follows because $\mathcal{G}$ contains the word $\mathbf{1}$, so that the coefficient of z^r must be the same as the coefficient of z^{23-r}. Further, now we know the code is perfect, we know that each of the $\binom{23}{4}$ words of weight 4 lies within distance 3 of a codeword of weight 7. Each codeword has $\binom{7}{4}$ such words of weight 4 lying close to it. Since $\binom{23}{4}/\binom{7}{4} = 253$, we deduce that there are 253 codewords of weight 7.

16.3 Properties of the Golay code

The Golay code has many interesting properties. For example in $\mathcal{G}^+$ there are 759 words of weight 8, called the octads. Now, each octad contains $\binom{8}{5}$ quintuples (that is, subwords of weight 5). In total therefore, between all the octads, there are $759\binom{8}{5}$ quintuples (two distinct octads cannot share a quintuple, else their sum would have weight 6, contradicting the minimum distance being 8). However, there are exactly $\binom{24}{5}$ quintuples in total. And, remarkably, $\binom{24}{5} = 759\binom{8}{5}$.

This means that every quintuple is contained in a unique octad. This is referred to as a Steiner $(5, 8, 24)$ system. Such a construction has applications on experimental design (you can think of wanting to apply different treatments to lots of different samples) or trying to buy lottery tickets to guarantee a prize).

A more difficult result concerns the automorphism group of $\mathcal{G}^+$. The automorphism group of a code is defined as follows. Write a list of all the codewords. The automorphism group is the set of permutations of the columns which leaves the code invariant (though obviously the rows of the table will be reordered).

It can be shown that the automorphism group of $\mathcal{G}^+$ is the Mathieu group M_{24}, a group of order $244823040 = 24.23.22.21.20.48$ – one of the sporadic groups. Recall that finite simple groups either fall into one of 4 infinite families, or are one of the 26 sporadic groups – see

`http://mathworld.wolfram.com/SporadicGroup.html`

In the same way, it can be shown that the automorphism group of the extended ternary Golay code (see Exercise 20) is another Mathieu group, M_{12}, of order $95040 = 2^6.3^3.5.11$). This ternary Golay code corresponds to a Steiner $(5, 6, 12)$ system.

We might hope to find a perfect code corresponding to each of these 26 sporadic simple groups. Of course, as we know there are no more perfect codes, so this is not possible. Although perfect codes are not common enough to be useful for everyday communication, they are still beautiful objects with deep symmetry and structure and I think they are enough on their own to make the study of coding theory worthwhile.

Information and Coding Michaelmas 2006/Michaelmas 2007

O.Johnson/Y. Suhov

A collection of exercises with solutions

This part of the notes contains a set of exercises with solutions. Some of these exercises were in the Example sheets.

Exercise 1 Let X be a integer-valued random variable with $p(x)$ as its probability mass function. Prove that, for all $d \geq 0$,

$$\mathbb{P}(p(X) \leq d) \leq \frac{h(\mathbf{p})}{-\log d}.$$

Solution. Use the Markov inequality. That is, we know that since $p(x) \leq 1$, so $-\log p(x) \geq 0$ for each x:

$$\begin{aligned} h(X) &= \sum_x p(x)\,(-\log p(x)) \geq \sum_x p(x)\,(-\log p(x))\,\mathbb{I}(p(x) \leq d) \\ &\geq \sum_x p(x)\,(-\log d)\,\mathbb{I}(p(x) \leq d) = (-\log d)\,\mathbb{P}(p(x) \leq d), \end{aligned}$$

(since $p(x) \leq d$ implies that $-\log p(x) \geq -\log d$).

Exercise 2 Suppose that Z is a disjoint mixture of random variables X and Y. That is, suppose that X takes values in $\{1, \ldots, m\}$ with probabilities p_X and Y takes values in $\{m+1, \ldots, m+n\}$ with probabilities p_Y. Then for parameter α,

$$Z = \begin{cases} X, & \text{with probability } \alpha \\ Y, & \text{with probability } 1-\alpha \end{cases}.$$

Find $h(Z)$ in terms of $h(X), h(Y)$ and α. Show that

$$2^{h(Z)} \leq 2^{h(X)} + 2^{h(Y)}.$$

Solution. We can write that Z has probabilities $p_Z(i)$, where

$$\begin{aligned} p_Z(i) &= \alpha p_X(i) && \text{for } 1 \leq i \leq m, \\ p_Z(i) &= (1-\alpha) p_Y(i) && \text{for } m+1 \leq i \leq m+n. \end{aligned}$$

Hence

$$
\begin{aligned}
h(Z) &= -\sum_{i=1}^{m} \alpha p_X(i) \log(\alpha p_X(i)) \\
&\quad -\sum_{j=1}^{n}((1-\alpha)p_Y(m+j)) \log((1-\alpha)p_Y(m+j)) \\
&= \alpha \sum_{i=1}^{m} p_X(i)(-\log\alpha - \log p_X(i)) \\
&\quad +(1-\alpha)\sum_{j=1}^{n} p_Y(m+j)(-\log(1-\alpha) - \log p_Y(m+j)) \\
&= (-\alpha\log\alpha - (1-\alpha)\log(1-\alpha)) + \alpha h(X) + (1-\alpha)h(Y).
\end{aligned}
$$

Alternatively, define $T = \mathbb{I}(Z \leq m)$ and use the fact that $h(Z) = h(T) + h(Z|T) = h(T) + \mathbb{P}(T=1)h(Z|T=1) + \mathbb{P}(T=0)h(Z|T=0)$.

Differentiating with respect to α, we obtain that this is minimised when $\log((1-\alpha)/\alpha) = h(Y) - h(X)$, or $1/\alpha - 1 = 2^{h(Y)-h(X)}$, so that

$$
\alpha = \frac{2^{h(X)}}{2^{h(X)} + 2^{h(Y)}}.
$$

Substituting this above, we obtain the result.

Exercise 3 If $p_r(x) = C_r x^{-1}(-\log_e x)^{-(r+1)}$ on $0 \leq x \leq e^{-1}$, prove that then for $0 < r \leq 1$, $h(p_r) = -\infty$.

Solution. We work with logarithms to the base e rather than base 2 throughout. By making the substitution $y = -\log_e x$, we deduce that

$$
\int_0^{e^{-1}} \frac{1}{x(-\log_e x)^{r+1}} dx = \int_1^{\infty} \frac{1}{y^{r+1}} dy
$$

is $1/r$ if $r > 0$ and ∞ otherwise. Hence we know that $C_r = r$. Further, using $z = \log_e(-\log_e x)$,

$$
\int_0^{e^{-1}} \frac{\log_e(-\log_e x)}{x(-\log_e x)^{r+1}} dx = \int_0^{\infty} z \exp(-rz) dz,
$$

which is $1/r^2$ if $r > 0$ and ∞ otherwise. Hence:

$$\begin{aligned}
& - \int p_r(x) \log_e p_r(x) dx \\
= & \int p_r(x) \left(-\log_e r + \log_e x + (r+1) \log_e(-\log_e x) \right) dx \\
= & -\log_e r - \int_0^{e^{-1}} \frac{r x^{-1}}{(-\log_e x)^r} - r(1+r) \frac{x^{-1} \log_e(-\log_e x)}{(-\log_e x)^{r+1}} dx,
\end{aligned}$$

so for $0 < r \leq 1$, the second term is infinite, and the others are finite.

Exercise 4 Consider a monotone increasing function g. Show that the differential entropy of $g(Y)$ satisfies

$$h(g(Y)) = h(Y) + \mathbb{E}(\log g'(Y)).$$

(Check that this is consistent with Lemma 3.3).

Solution. The distribution function of $g(Y)$ satisfies $F_{g(Y)}(y) = \mathbb{P}(g(Y) \leq y) = \mathbb{P}(Y \leq g^{-1}(y)) = F_Y(g^{-1}(y))$. Now, recall that $g^{-1}(y)$ has derivative $1/g'(g^{-1}(y))$. This means that the density satisfies:

$$f_{g(Y)}(y) = f_Y(g^{-1}(y))/g'(g^{-1}(y)).$$

Hence:

$$\begin{aligned}
h(g(Y)) &= -\int f_{g(Y)}(y) \log f_{g(Y)}(y) dy \\
&= -\int \frac{f_Y(g^{-1}(y))}{g'(g^{-1}(y))} \left(\log f_Y(g^{-1}(y)) - \log g'(g^{-1}(y)) \right) dy \\
&= -\int \frac{f_Y(z)}{g'(z)} \left(\log f_Y(z) - \log g'(z) \right) g'(z) dz \\
&= -\int f_Y(z) \left(\log f_Y(z) - \log g'(z) \right) dz \\
&= h(Y) + \int f_Y(z) \log g'(z) dz,
\end{aligned}$$

taking $z = g^{-1}(y)$, so $y = g(z)$ and $dy = g'(z) dz$. The result follows.

For $g(y) = ay + b$, the $g'(y) = a$, so $\mathbb{E}(\log g'(Y)) = \log a$, so this result generalises Lemma 3.3.)

Exercise 5 If p is the density of a random variable supported on the positive half-line and with mean μ, then show that

$$h(p) \leq \log(\mu e),$$

with equality if and only if p is an exponential density.

Solution. Take q_μ to be the density of an exponential distribution with mean μ. Expanding in the same way as in Lemma 4.4:

$$\begin{aligned} 0 \;\leq\; & D(p\|q_\mu) = \int_0^\infty p(x) \left(\log p(x) + \frac{x}{\mu} \log e + \log \mu \right) dx \\ =\; & -h(p) + \log(e\mu) = -h(p) + h(q_\mu). \end{aligned}$$

Exercise 6 The relative entropy $D(p\|q)$ measures how easy it is to distinguish samples taken from densities p and q. Define $p_t(x) = p(x+t)$ (a shift of the density p). Show that (if p and p' are well-behaved at $\pm\infty$):

$$\left. \frac{\partial^2}{\partial t^2} D(p\|p_t) \right|_{t=0} = (\log e) \int p(x) \rho(x)^2 dx,$$

where $\rho(x) = p'(x)/p(x)$. The RHS is $\log e$ times a quantity referred to as the Fisher information with respect to location parameter, and given the symbol J.

Solution. The factor of $\log e$ emerges from taking logs to base 2. Switch to logarithms to base e, for the sake of simple calculations.

Clearly $D(p\|p_t) = 0$ for $t = 0$.

Next, we can consider

$$\begin{aligned} \frac{\partial}{\partial t} D(p\|p_t) &= \frac{\partial}{\partial t} \int p(x) \left(\log p(x) - \log p(x+t) \right) dx \\ &= -\int p(x) \frac{p'(x+t)}{p(x+t)} dx \end{aligned}$$

At $t = 0$, this reduces to $-\int p'(x) dx = 0$ (assuming that $p(x) \to 0$ at $\pm\infty$).

Finally,

$$\frac{\partial^2}{\partial t^2} D(p\|p_t) = -\int p(x) \frac{p''(x+t)}{p(x+t)} dx - \int p(x) p'(x+t) \left(\frac{-p'(x+t)}{p(x+t)^2} \right) dx$$

At $t = 0$, the first term is $-\int p''(x)dx = 0$ (assuming that $p(x) \to 0$ at $\pm\infty$). The second term reduces to the promised Fisher information.

Exercise 7 This question will establish some simple properties of the Fisher information introduced in Exercise 6, which suggest that it is similar to relative entropy in many ways.

1. Given a random variable U with mean 0 and variance σ^2, show (by considering $\mathbb{E}(\rho(U) + Ua)^2$, for some a) that the Fisher information is bounded below:
$$J(U) \geq 1/\sigma^2.$$
with equality if and only if U is $N(0, \sigma^2)$.

2. If U, V are independent random variables and $W = U + V$ with score functions ρ_U, ρ_V and ρ_W then show that
$$\rho_W(w) = \mathbb{E}[\rho_U(U)|W = w] = \mathbb{E}[\rho_V(V)|W = w].$$
Hint: consider the derivative of the density of W.

3. If U, V are independent random variables then show that
$$\frac{1}{J(U+V)} \geq \frac{1}{J(U)} + \frac{1}{J(V)}.$$
Hint: use the previous result - that is for some β, consider $\rho_W = \mathbb{E}[\beta\rho_U(U) + (1-\beta)\rho_V(V)|W = w]$.

Solution. 1. The key is to notice that $\mathbb{E}(\rho(U) + aU)^2$ is non-negative. Now, notice that $\mathbb{E}\rho(U)U = \int p(x)(p'(x)/p(x))x dx = \int p'(x)x dx = -\int p(x)dx = -1$. This means that (writing J for the Fisher information) $0 \leq J(U) - 2a + a^2\sigma^2$, so picking $a = 1/\sigma^2$, the result follows. Equality holds iff $\rho(x) \equiv -x/\sigma^2$, which characterises the normal distribution.

2. If U, V have densities $p(u), q(v)$, $U + V$ has the convolution density $r(w) = \int p(u)q(w-u)du$, so that

$$r'(w) = \int p(u)\frac{\partial q}{\partial w}(w-u)du = -\int p(u)\frac{\partial q}{\partial u}(w-u)du = \int p'(u)q(w-u)du$$

and hence:

$$\begin{aligned}
\frac{r'(w)}{r(w)} &= \int \frac{p'(u)q(w-u)}{r(w)} du = \int \frac{p'(u)}{p(u)} \frac{p(u)q(w-u)}{r(w)} du \\
&= \mathbb{E}\left[\frac{p'(U)}{p(U)} \Big| W = w \right].
\end{aligned}$$

Similarly, we can produce an expression in terms of the score function $q'(V)/q(V)$.

3. Using the hint, Jensen's inequality gives:

$$\begin{aligned}
J(W) = \mathbb{E}\rho_W^2(W) &= \mathbb{E}[\mathbb{E}(\beta\rho_U(U) + (1-\beta)\rho_V(V)|W)^2] \\
&\leq \mathbb{E}[\mathbb{E}((\beta\rho_U(U) + (1-\beta)\rho_V(V))^2|W)] \\
&= \beta^2\mathbb{E}\rho_U^2(U) + (1-\beta)^2\mathbb{E}\rho_V^2(V) + 2\beta(1-\beta)\mathbb{E}\rho_U(U)\mathbb{E}\rho_V(V) \\
&= \beta^2\mathbb{E}\rho_U^2(U) + (1-\beta)^2\mathbb{E}\rho_V^2(V),
\end{aligned}$$

Now, we have a free choice of β. The result follows on $\beta = J(V)/(J(U) + J(V))$, which is the optimal value of β for given $J(U), J(V)$.

Exercise 8 More on the Asymptotic Equipartition Property:

1. **(Continuous random variables).** Consider $\mathbf{X} = X_1, \ldots X_n$ IID with density f. Write $h(f)$ for the differential entropy of h and define the typical set

$$T_\epsilon^{(n)} = \{(x_1, \ldots x_n) : \left| \frac{-1}{n} \log f(x_1, \ldots, x_n) - h(f) \right| \leq \epsilon\}.$$

   Show that

   (a) For all n sufficiently large, $\mathbb{P}(\mathbf{X} \in T_\epsilon^{(n)}) \geq 1 - \epsilon$.

   (b) The set $T_\epsilon^{(n)}$ has Lebesgue measure $\leq 2^{n(h(f)+\epsilon)}$.

2. **(k-dependent random variables).** We say that a collection of random variables $, \ldots, X_{-1}, X_0, X_1, \ldots$ are k-dependent, if X_i and X_j are independent for $|i-j| > k$. (An example of such a collection is found by taking independent $\ldots Y_{-1}, Y_0, Y_1, \ldots$, and defining $X_i = \sum_{r=0}^k a_r Y_{i+r}$ for some coefficients a_r).

   Show that a collection of stationary (i.e. when the joint distribution of $X_i, \ldots X_{i+j}$ is the same for all i) k-dependent random variables taking values on a finite set have the AEP.

Solution. We need to duplicate the proof of the AEP for discrete variables.

1. Specifically:

$$\frac{-1}{n}\log f(X_1,\ldots,X_n)=\sum_{i=1}^{n}\frac{-\log f(X_i)}{n},$$

a sum of IID variables, which converges in probability to its mean $h(f)$ by the Law of Large Numbers.

To see the Lebesgue measure result, note that on $T_\epsilon^{(n)}$, $f(\mathbf{x})=f(x_1,\ldots x_n)\geq 2^{-n(h(f)+\epsilon)}$ so that:

$$\begin{aligned}1 &\geq \int \mathbb{I}(\mathbf{x}\in T_\epsilon^{(n)})f(\mathbf{x})d\mathbf{x}\\ &\geq 2^{-n(h(f)+\epsilon)}\int \mathbb{I}(\mathbf{x}\in T_\epsilon^{(n)})d\mathbf{x}\\ &= 2^{-n(h(f)+\epsilon)}\mathrm{vol}(\mathrm{T}_\epsilon^{(\mathrm{n})})\end{aligned}$$

as required.

2. The key is to show that k-dependent random variables Z_i satisfy the Law of Large Numbers. As in the proof of Theorem 2.11, we need to control $\mathrm{Var}(\sum_{i=1}^n Z_i)$, and see that it grows linearly with n. But since $\mathrm{Var}(Z_i,Z_j)=0$ for $|i-j|>k$

$$\begin{aligned}\mathrm{Var}(\sum_{i=1}^n Z_i) &= \sum_{i=1}^n\left(\mathrm{Var}Z_i+\sum_{j\neq i}\mathrm{Var}(Z_i,Z_j)\right)\\ &\leq \sum_{i=1}^n(\mathrm{Var}Z_i+2k\mathrm{Var}Z_i)=n(1+2k)\mathrm{Var}Z_i.\end{aligned}$$

Exercise 9 Checking the details of something claimed in the proof of Lemma 7.5. Show that for random variables R,S,T:

$$h(R|S)\geq h(R|S,T),$$

with equality if R and T are conditionally independent given S.

Solution. Use Definition 3.9 to expand out the RHS. It becomes $h(R,S,T)-h(S,T)=(h(R,T,S)-h(S))-(h(S,T)-h(S))=h(R,T|S)-h(T|S)$. Now,

$$0\leq h(R|S)+h(T|S)-h(R,T|S) \tag{1.1}$$

so substituting this above, the result follows.

To justify Equation (1.1), note that it is the equivalent of Lemma 3.8, and can be proved the same way. Specifically, if can be rewritten as a sum over s:

$$\sum_s \mathbb{P}(S = s) \left(h(R|S = s) + h(T|S = s) - h(R, T|S = s) \right).$$

Hence, for each fixed s, we can repeat the argument that proved Lemma 3.8, writing $p_s(x, y)$ for the joint probability distribution of $R, T|S = s$ for that particular s.

Exercise 10 Suppose that we have a family of random variables $f_\theta(y)$ indexed by a parameter θ (for example, normal distributions with unknown mean). Suppose that θ takes a random value distributed according to random variable Θ.

Using Exercise 9, show that for any function T

$$i(\Theta, Y) \geq i(\Theta, T(Y)),$$

with equality if and only if Θ and Y are independent, given $T(Y)$.

Use this to give a proof of the (Part 1B Statistics) result that $T(y)$ is a sufficient statistic (it contains as much information about θ as y does) if and only if the factorization criterion holds.

Solution. The key is to notice that $h(\Theta|Y) = h(\Theta|Y, T(Y)) \leq h(\Theta|T(Y))$, so that $i(\Theta, Y) - i(\Theta, T(Y)) = (h(\Theta) - h(\Theta|Y)) - ((h(\Theta) - h(\Theta|T(Y))) \geq 0$.

The factorization criterion requires that the density $f_\theta(y) = g(T(y), \theta)h(y)$, which can be rearranged to show that Θ and Y are independent given $T(Y)$.

Exercise 11 A channel has binary input and output alphabets and transition probabilities $p(y|x)$ given by the following matrix

$$\begin{pmatrix} 1 & 0 \\ 1/2 & 1/2 \end{pmatrix}.$$

Find the capacity C of the channel and the maximizing input probability distribution.

Suppose that instead of using this optimal distribution, we use a code where we generate 2^{nR} codewords, where the letters are chosen independently with probability $1/2$. For what R does the probability of error tend to zero.

Solution. If $p_X(0) = p$, then

$$i(X,Y) = h(Y) - h(Y|X) = -\frac{1}{2}(1+p)\log\frac{1+p}{2} - \frac{1}{2}(1-p)\log\frac{1-p}{2} - 1 + p, \tag{1.2}$$

which is maximised by $p = 3/5$ with $C = \log 5 - 2 \simeq 0.32198$.

For general choice of distribution of X, the proof of Theorem 8.6 shows that we can take $R = i(X,Y) - \epsilon$, for arbitrary X. Taking $p = 1/2$ in Equation (1.2), we obtain

$$i(X,Y) = -3/4\log(3/4) - 1/4\log(1/4) - 1/2 = -3/4\log 3 + 3/2 \simeq 0.31128$$

It is not clear whether any larger R will work. Any ideas?

Exercise 12 Consider a linear binary code of length N, rank k and minimum distance d. By reordering the columns if necessary, find a generating matrix G of the form

$$G = \begin{pmatrix} 1\dots1 & 0\dots0 \\ G_1 & G_2 \end{pmatrix}.$$

Show that the code with generating matrix G_2 has minimum distance $\geq d/2$. Prove the Griesmer bound: that

$$N \geq \sum_{i=0}^{k-1} \lceil d/2^i \rceil.$$

Solution. Consider any two words $\mathbf{x}, \mathbf{y}$ in the code G_2. There must be words of the form $(\mathbf{u}|\mathbf{x})$ and $(\mathbf{v}|\mathbf{y})$ in the original code. Further, by linearity $(\mathbf{u} + \mathbf{1}|\mathbf{x})$ must also be in the code.

Now, we know that

$$d \leq d((\mathbf{u}|\mathbf{x}), (\mathbf{v}|\mathbf{y})) = d(\mathbf{u}, \mathbf{v}) + d(\mathbf{x}, \mathbf{y}) \tag{1.3}$$

$$d \leq d((\mathbf{1} + \mathbf{u}|\mathbf{x}), (\mathbf{v}|\mathbf{y})) = d(\mathbf{1} + \mathbf{u}, \mathbf{v}) + d(\mathbf{x}, \mathbf{y}) \tag{1.4}$$

Now, we know that for every entry where $d(\mathbf{u}, \mathbf{v})$ are different, $d(\mathbf{u} + \mathbf{1}, \mathbf{v})$ will be the same (and vice versa). Hence $d(\mathbf{u}, \mathbf{v}) + d(\mathbf{u} + \mathbf{1}, \mathbf{v}) = d$.

Adding Equations (1.3) and (1.4) together, we obtain

$$2d \leq d + 2d(\mathbf{x}, \mathbf{y}),$$

and the result follows.

We can repeat this argument; the code generated by G_2 has rank $k-1$, length d and minimum distance $\geq d/2$. Repeating, we obtain a code with rank $k-2$, length $\geq d/2$, minimum distance $\geq d/4$, and so on. The total length N is $\geq$ the sum of the lengths of these intermediate codes. At each stage, since the length is $\geq d/2^i$ and is an integer, it is in fact $\geq \lceil d/2^i \rceil$.

Exercise 13 Generalise the constructions of the simplest types of perfect codes (that is, the trivial, repetition and Hamming codes) in Section 10.1 to the case of an alphabet of general size q, where q is prime. What are the parameters of the new codes? Are they perfect?

Solution. The trivial code generalises to $E = 0$, with $\delta = 1$, $r = q^N$. Hence the volume of each ball of radius 0 is 1, and the code is perfect (in fact it is perfect for any q, not just q prime).

The repetition code has size $r = q$ – the codewords are $\{0\ldots 0, 1\ldots 1, \ldots, (q-1)\ldots(q-1)\}$. The distance $\delta = N$, but the code is not perfect (consider a received word with $\lfloor N/q \rfloor$ each of zeroes, ones, twos, ... It is not within distance $\lfloor (N-1)/2 \rfloor$ of any codewords for $q > 2$.

The Hamming code has a parity check matrix constructed as follows. Make a list of every word of non-zero word of length r, but throw out words which are multiples of words already in the list. Since q is prime, this list will be of length $(q^r-1)/(q-1)$, so writing it as parity check matrix, the length of the code is $N = (q^r-1)/(q-1)$, and the rank is $N - r = (q^r-1)/(q-1) - r$. As before the distance is 3, and so the code can correct $E = 1$ error. The volume of the ball of radius 1 is $1 + N(q-1) = 1 + (q^r - 1) = q^r$, so we can fit q^{N-r} balls of this radius into the space, and the code is perfect.

Exercise 14 Establish an asymptotic form of the Elias bound. That is, pick x and δ, and take $d = \lfloor N\delta \rfloor$, and $s = \lfloor Nx \rfloor$. Ensuring that d, s satisfy the given constraints, give the asymptotic result.

Solution. Recall that the Elias bound needs $1 \leq s < N/2$, which corresponds to requiring $0 < x < 1/2$.

Further, we need $2s^2 - 2Ns + Nd > 0$. Since $d/N \to \delta$, $s/N \to x$, this will hold if $2x^2 - 2x + \delta > 0$. The roots of this in x are at $(1 \pm \sqrt{1-2\delta})/2$, so overall, x must satisfy $0 < x < (1 - \sqrt{1-2\delta})/2$.

Substituting this into the Elias bound, we obtain that

$$\frac{\log r}{N} \leq \frac{1}{N}\log\left(\frac{Nd}{2s^2-2Ns+Nd}\right)+1-\frac{1}{N}\log\nu_N(s).$$

Now the first term acts like $\log(\delta/(2x^2-2x+\delta))/N$, and from Lemma 11.1, the third term converges to $h(x)$.

Overall then, the asymptotic Elias bound is $1-h(x)$, with a free choice of x. To make the bound as tight as possible, take $x=(1-\sqrt{1-2\delta})/2)_+$.

Exercise 15 Show that the $RM(3,1)$ code is the parity extension of Hamming's original code. In general, show that the $RM(d,d-2)$ code is the parity extension of the Hamming code of length 2^d-1. Show that the $RM(3,2)$ code is the parity code of length 8. In general, show that the $RM(d,d-1)$ code is the parity code of length 2^d.

Solution. By Lemma 13.8, the $RM(d,d-2)$ code has length 2^d, rank $\binom{d}{0}+\binom{d}{1}+\ldots+\binom{d}{d-2}=2^d-\binom{d}{d-1}-\binom{d}{d}=2^d-d-1$, and minimum distance $2^{d-(d-2)}=2^2=4$. Hence, the parameters are right – now the truncated code has length 2^d-1, rank 2^d-d-1 and minimum distance ≥ 3. Hence, by the Hamming bound, it can only be the Hamming code.

In the same way, the code $RM(d,d-1)$ has length 2^d, rank 2^d-1 and minimum distance 2. Further $RM(d,d-1)=RM(d-1,d-2)|RM(d-1,d-1)$, so by induction every word is of even weight.

Exercise 16 Show that for each l, $\mathcal{X}_l^\perp$, the dual of the Hamming $[2^l-1,2^l-l-1]$ code has the property that all non-zero words have weight 2^{l-1}, and hence any two non-equal codewords are distance 2^{l-1} apart.

Solution. We can use induction. That is, if we add a column of zeroes to each generator matrix, then we obtain the parity extension $\mathcal{X}_l^{\perp+}$.

For each l we can write G_{l+1} in the form:

$$G_{l+1}=\begin{pmatrix} 1\ldots1 & 0\ldots0 \\ G_l & G_l \end{pmatrix}.$$

The code generated by G_{l+1} will contain words of the form $(\mathbf{x}|\mathbf{x})$ and $(\mathbf{1}+\mathbf{x}|\mathbf{x})$, where $\mathbf{x}\in\mathcal{X}_l^{\perp+}$.

Now, by assumption we know that $\mathbf{x}$ (and hence $\mathbf{x}+\mathbf{1}$) both have weight 2^{l-1}, so the result follows.

Exercise 17 Show that for the Reed-Muller codes $RM(e,r)$ is dual to another $RM(e,r^*)$ code. First, by considering sums of ranks, show what r^* would have to be.

Then by induction on e, complete the result.

Solution. We will show that $RM(e,r)$ is dual to the $RM(e,e-r-1)$ code.

By Lemma 13.8, the sum of the ranks is

$$\begin{aligned}
&\operatorname{rank}(RM(e,r)) + \operatorname{rank}(RM(e,e-r-1)) \\
&= \left(\binom{e}{0}+\binom{e}{1}+\ldots+\binom{e}{r}\right)+\left(\binom{e}{0}+\binom{e}{1}+\ldots+\binom{e}{e-r-1}\right) \\
&= \left(\binom{e}{0}+\binom{e}{1}+\ldots+\binom{e}{r}\right)+\left(\binom{e}{e}+\binom{e}{e-1}+\ldots+\binom{e}{r+1}\right) = 2^e.
\end{aligned}$$

Hence, r^* can be $e-r-1$.

We will show that these codes are really dual, by induction on e. That is, take

$$\begin{aligned}
RM(e,r) &= \{(\mathbf{x}|\mathbf{x}+\mathbf{y}) : \mathbf{x}\in RM(e-1,r), \mathbf{y}\in RM(e-1,r-1)\} \\
RM(e,e-r-1) &= \{(\mathbf{x}'|\mathbf{x}'+\mathbf{y}') : \\
&\qquad \mathbf{x}'\in RM(e-1,e-r-1), \mathbf{y}'\in RM(e-1,e-r-2)\}
\end{aligned}$$

Now, by the inductive hypothesis, $RM(e-1,r)$ is dual to $RM(e-1,e-r-2)$, so $\mathbf{x}\cdot\mathbf{y}'=0$. Further, $RM(e-1,r-1)$ is dual to $RM(e-1,e-r-1)$, so $\mathbf{x}'\cdot\mathbf{y}=0$. Finally, since by Lemma 13.8, $RM(e-1,r-1)\subseteq RM(e-1,r)$, we know $\mathbf{y}\cdot\mathbf{y}'=0$.

This means that

$$(\mathbf{x},\mathbf{x}+\mathbf{y})\cdot(\mathbf{x}',\mathbf{x}'+\mathbf{y}') = 2(\mathbf{x}\cdot\mathbf{x}')+\mathbf{x}\cdot\mathbf{y}'+\mathbf{y}\cdot\mathbf{x}'+\mathbf{y}\cdot\mathbf{y}' = 0.$$

Exercise 18 Show that if a binary linear code $\mathcal{X}$ can correct at least 1 error, then the words of the dual code $\mathcal{X}^\perp$ have average weight $n/2$.

Solution. If the weight-enumerator polynomial of a code χ is $W(z) = \sum_i A_i z_i$, the average weight is $W'(1)/(\#\mathcal{X}) = (\sum_i A_i i)/(\#\mathcal{X})$. Now, for the dual code, the MacWilliams Identity gives:

$$W_{\mathcal{X}^\perp}(z) = \frac{1}{\#\mathcal{X}} \sum_i A_i (1-z)^i (1+z)^{n-i},$$

so

$$W'_{\mathcal{X}^\perp}(z) = \frac{1}{\#\mathcal{X}} \sum A_i \left(-i(1-z)^{i-1}(1+z)^{n-i} + (n-i)(1-z)^i(1+z)^{n-i-1}\right).$$

Thus substituting in $z = 1$, we obtain $(-A_1 2^{n-1} + nA_0 2^{n-1})/(\#\mathcal{X})$. The result follows since $A_0 = 1$ (code is linear), $A_1 = 0$ (code can correct one error), and since $(\#\mathcal{X}^\perp)(\#\mathcal{X}) = 2^n$.

Exercise 19 Suppose a code $\mathcal{X}$ is cyclic with generator g, where $(X+1)$ does not divide $g(X)$. Identify the code $\mathcal{X}'$ generated by $(X+1)g(X)$.

Solution. The code $\mathcal{X}'$ is the even weight subcode of $\mathcal{X}$. Notice that if codeword $\mathbf{u}$ is represented by polynomial $u(X)$ then $w(\mathbf{u}) = \sum_{i=1}^N u_i = u(1)$.

Now, $\mathcal{X}'$ consists of all multiples $c(X) = a(X)((X+1)g(X))$. This means that $w(\mathbf{c}) = c(1) = a(1) * 0 * g(1) \equiv 0 \mod 2$, so all words in this code have even weight. Further, any word in $\mathcal{X}'$ is a multiple of $(X+1)g(X)$, and hence a multiple of $g(X)$, and so in the original code $\mathcal{X}$.

Conversely, any word in $\mathcal{X}$ of even weight must be a multiple $p(X)g(X)$. Since it has even weight, $p(1)g(1) = 0$, but $g(1) \neq 0$, so $p(1) = 0$, so we can write $p(X) = (X+1)q(X)$, so the word is a multiple of $(X+1)g(X)$, and hence belongs in $\mathcal{X}'$.

Exercise 20 We give a construction for the ternary Golay code . Define the symmetric matrix B to be the square root of $-I \mod 3$ (that is $B^2 \equiv -I \mod 3$). In fact (check?),

$$B = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 2 & 2 & 1 \\ 1 & 1 & 0 & 1 & 2 & 2 \\ 1 & 2 & 1 & 0 & 1 & 2 \\ 1 & 2 & 2 & 1 & 0 & 1 \\ 1 & 1 & 2 & 2 & 1 & 0 \end{pmatrix}.$$

Define the ternary Golay code $\mathcal{G}_{12}$ to have generator matrix $G = (I_6|B)$ (where I_6 is the identity matrix of size 6).

Show that (a) $\mathcal{G}_{12}$ is self-dual (b) $\mathcal{G}_{12}$ has length 12, rank 6 and minimum distance 6 (c) $\mathcal{G}_{11}$, the truncation of $\mathcal{G}_{12}$, is perfect.

Solution. (a) A code is self-dual if for all i, j, $g_i \cdot g_j = 0$, where g_i is the ith row of G. Now (working $\mod 3$)

$$g_i \cdot g_j = \sum_{k=1}^{6} g_{ik} g_{jk} + \sum_{k=7}^{12} g_{ik} g_{jk} = (I_6 I_6^T)_{ij} + (BB^T)_{ij} = \delta_{ij} - \delta_{ij} = 0,$$

as required.

(b) The code $\mathcal{G}_{12}$ clearly has length 12, and rank 6 (the presence of the identity matrix makes it clear that the rows are linearly independent).

Notice that all the generators have weight 6. We will show that this is the minimum distance.

In fact all the codewords $\mathbf{c}$ have weight a multiple of 3 (since working mod 3, the $0^2 = 0, 1^2 = 1, 2^2 = 1$, so $c_i^2 = I(c_i \neq 0)$. Then

$$w(\mathbf{c}) = \sum_i \mathbb{I}(c_i \neq 0) = \sum_i c_i^2 = \mathbf{c} \cdot \mathbf{c} = 0.$$

Warning: this bit gets ugly - any better ideas? Now, no codeword has weight 3; if so, then there would be a set of columns of G which were LD. Clearly, they can't all be from I_6, nor can 2 of them be from I_6 (since then one would be from B). All linear combinations of 2 columns of B have weight ≥ 2 (because of the diagonal of zeroes). Hence, all 3 columns would have to come from B.

There must be 3 indices i_1, i_2, i_3 such that $a_{i_1}\mathbf{b}_{i_1} + a_{i_2}\mathbf{b}_{i_2} + a_{i_3}\mathbf{b}_{i_3} = \mathbf{0}$. Now, by the pigeonhole principle, without loss, $a_{i_1} = a_{i_2} = 1$. But the sum of no two columns has weight 5.

(c) The truncation $\mathcal{G}_{11}$ has length 11, rank 6, and distance 5 or 6. The Hamming bound shows that distance 6 is impossible, but that distance 5 is possible, with equality holding – i.e. the code $\mathcal{G}_{11}$ is perfect.