UNIVERSITY OF OXFORD

MATHEMATICAL INSTITUTE

Information Theory

Dr. Hanqing Jin[1]

Hilary Term Term 2021

[1]This lecture note is from Dr Harald Oberhause with minor revision.

Contents

Introduction

Communication theory is a relatively young subject. It played an important role in the rise of the current information/digital/computer age and still motivates much research. Every time you make a phone call, store a file on your computer, query an internet search engine, watch a DVD, stream a movie, listen to a CD or mp3 file, etc., algorithms run that are based on topics we discuss in this course. However, independent of such applications, the underlying mathematical objects arise naturally as soon as one starts to think about "information", its representation and how to transfer and store information. In fact, a large part of the course deals with two fundamental questions:

(1) How much information is contained in a signal/data/message? (source coding)

(2) What are the limits to information transfer over a channel that is subject to noisy perturbations? (channel coding)

To answers to above questions requires us to develop new mathematical concepts. These concepts also give new interpretations of important results in probability theory. Moreover, they are intimately connected to

- Physics: Thermodynamics, Statistical mechanics, Quantum theory,
- Computer Science: Kolmogorov complexity, etc.
- Statistics and Machine learning,
- Large deviation theory,
- Economics, finance, gambling.

Textbook and Literature. For most parts of the course we follow the classic textbook

- Cover, T. (2012). Elements of information theory. John Wiley & Sons.

Another excellent book is

- MacKay, D. J. (2003). Information theory, inference and learning algorithms. Cambridge University Press, which has a more informal approach but many applications and is freely available on David MacKay's old webpage[2]. A concise treatment, focused on the theory is

- Csiszar, Körner (2011). Information Theory: Coding Theorems for Discrete Memoryless Systems. Cambridge University Press.

[2] `https://www.inferernce.phy.cam.ac.uk/mackay/itila/`

Chapter 1

Entropy, Divergence and Mutual Information

We will use a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ to describe the randomness, on which we define random variables, which are functions from $(\Omega, \mathcal{F})$ to $(\mathbb{R}, \mathcal{B}(\mathbb{R}))$. We will often omit these notations and only talk about random variables. Furthermore, we will focus on discrete random variables which take values in a discrete subset $\mathcal{X} \subset \mathbb{R}$, whose distributions can be described by their probability mass functions (pmf), e.g., for a random variable X, its pmf is $p(x) = \mathbb{P}(X = x)$ for $x \in \mathcal{X}$.

1.1 Entropy

Definition 1.1. *The entropy $H_b(X)$ in base b of a discrete random variable X is defined as*

$$H_b(X) = -\sum_{x \in \mathcal{X}} \mathbb{P}(X = x) \log_b \mathbb{P}(X = x), \tag{1.1.1}$$

where we use the convention that $0 \times \log_b(0) = 0$. For $b = 2$ we usually write $H(X)$ instead $H_2(X)$, and write $\log(q)$ instead $\log_2(q)$.

Some remarks on the concept:

- The notation $H(X)$ is somewhat misleading since the entropy only depends on the pmf of the random variable X, i.e., for two different random variables X and $\hat{X}$ with the same pmf, their entropies are the same. However, this notation is standard in the literature and the choice of $\mathbb{P}$ is usually unambigious in our applications. We also use the notation $H(P_X)$ or $H(p_X)$ for the entropy of X, where$P_X = \mathbb{P} \circ X^{-1}$ is the distribution of X and $p_X(x) = \mathbb{P}(X = x)$ is the pmf of X.
- Above reads[1] $H(X) = -\mathbb{E}[\log(p(X))]$ where $p(\cdot) = p_X(\cdot)$ is the pmf of X.

[1]Attention: often one uses X as an index for the pmf, i.e. $p_X = (x) = \mathbb{P}(X = x)$. In this case the entropy is written as $H(X) = -\mathbb{E}[\log(p_X(X))] = -\sum_{x \in \mathcal{X}} p_X(x) \log(p_X(x))$ but we emphasise that $p_X : \mathcal{X} \mapsto [0, 1]$ is a function and not random (does not depend on $\omega \in \Omega$)! A better notation would be to enumerate r.v. values x_i with $i \in \mathbb{N}$ and to denote the pmf of X with $p_i = \mathbb{P}(X = x_i)$, though this is less standard.

- The choice of base 2 for the logarithm is common (due to computers using two states) but not essen tial. Since $\log(x) = \log_b(x) = \frac{\log_a(x)}{\log_a(b)}$, we have $H_b(X) = \frac{1}{\log_a(b)} H_a(X)$.
- The unit of entropy in base 2 is called a `bit`, in base e `nat`, in base 256 a `byte`. As usual in mathematics, we do not use units but dimension checking is a useful sanity check for many calculations.

One way (among many!) to motivate above definition, is to think of $H(X)$ as a measure of the average uncertainty we have about the value of X: the less certain we are, the bigger $H(X)$. To see this, we first derive a function $s(A)$ to measure the "surprise" of observing the event $\{X \in A\}$ for a set $A \subset \mathcal{X}$. It seems to natural to demand that

(1) $s(A)$ depends continuously on $\mathbb{P}(X \in A)$,

(2) $s(A)$ is decreasing in $\mathbb{P}(X \in A)$,

(3) $s(A \cap B) = s(A) + s(B)$ for $\mathbb{P}(X \in A \cap B) = \mathbb{P}(X \in A)\mathbb{P}(X \in B)$, i.e., the surprise about the occurrence of two independent events $\{X \in A\}, \{X \in B\}$ is the sum of the surprises of each of these events.

Using that $\mathbb{P}(X \in A \cap B) = \mathbb{P}(X \in A)\mathbb{P}(X \in B)$, it follows that $s(A) = -\log(\mathbb{P}(A))$ fulfills these properties and is the unique function with these properties (up to choice of a multiplicative constant and base of the logarithm). In some books, $s(A)$ is also called the Shannon information content of the outcome A. Hence, we can regard the entropy $H(X)$ as the "average surprise" over the events $\{X = x\}$ for $x \in \mathcal{X}$. We will encounter other motivations for the definition of $H(X)$ later (e.g. as a compression bound, as number of yes-no-questions to determine a value, etc).

Example 1.2. If $\mathcal{X} = \{H, T\}$ and $\mathbb{P}(X = H) = p$, then

$$H(X) = -p\log(p) - (1-p)\log(1-p). \tag{1.1.2}$$

If $p \in \{0, 1\}$, then $H(X) = 0$. Differentiating in p shows that the entropy as a function of p increases on $(0, 0.5)$ and decreasing on $(0.5, 1)$. Hence, the entropy is maximised if $p = 0.5$ with $H(X) = \log(2) = 1$ bits.

Example 1.3. If X is a 2-dim vector in the form (X_1, X_2) with $X_i \in \mathcal{X}_i$ for $i = 1, 2$, then

$$H(X) = H(X_1, X_2) = -\sum_{x_1 \in \mathcal{X}_1, x_2 \in \mathcal{X}_2} p_{X_1,X_2}(x_1, x_2)\log(p_{X_1,X_2}(x_1, x_2). \tag{1.1.3}$$

If additionally, X_1 and X_2 are independent, i.e., $p_{X_1,X_2}(x_1, x_2) = p_{X_1}(x_1)p_{X_2}(x_2)$, then

$$H(X) = H(X_1) + H(X_2). \tag{1.1.4}$$

If X_1 and X_2 are independent and identically distributed (i.i.d.), then

$$H(X) = 2H(X_1) = 2H(X_2). \tag{1.1.5}$$

Now assume, X models a coin flip as in Example 1.2, i.e. X takes values in $\mathcal{X} = \{H, T\}$. Given knowledge about p, we want store the results of a sequence of n independent coin flips. One extreme case is $p \in \{0, 1\}$, in which case we need $H(X) = 0$ bits, the other extreme is $p = 0.5$ in which it is at least intuitive that we need n bits. This hints at another interpretation of entropy , namely as a storage/compression bound of information. We make this connection rigorous in Chapter 3.

1.2 Divergence

Definition 1.4. *Let p and q be pmfs on $\mathcal{X}$. We call*

$$D(p\|q) = \sum_{x\in\mathcal{X}} p(x)\log\left(\frac{p(x)}{q(x)}\right) \tag{1.2.1}$$

the divergence between p and q and set by convention $0\times\log(0) = 0$ and $D(p\|q) = \infty$ if $\exists x \in X$ such that $q(x) = 0$, $p(x) > 0$. (Divergence is also known as information divergence, Kullback–Leibler divergence, relative entropy).

Note that, given $X \sim p$ (which means the pmf of X is p),

$$\begin{aligned} D(p\|q) &= \mathbb{E}\left[\log\left(\frac{p(X)}{q(X)}\right)\right] \\ &= \mathbb{E}\left[\log\left(\frac{1}{q(X)}\right)\right] - \mathbb{E}\left[\log\left(\frac{1}{p(X)}\right)\right] \\ &= \mathbb{E}\left[\log\left(\frac{1}{q(X)}\right)\right] - H(X). \end{aligned}$$

In Example 1.2 we hinted at entropy as a measure for storage cost and from this perspective we can think of divergence as the cost we incur if we use the distribution q to encode a random variable X with distribution p. (Again we make all this rigorous in Chapter 3). Further, note that while we will show below that divergence is always non-negative it is not a metric: in general it is not symmetric and can take the value ∞. These properties are actually useful and desirable as the following example shows.

Example 1.5. (Asymmetry and infinite values are useful). Let $\mathcal{X} = \{0, 1\}$ and $p(0) = 0.5$, $q(0) = 1$. We are given independent samples from one of these two distributions but we do not know which one. If we observe 0000001, we can immediately infer that p is the underlying pmf. On the other hand, if we observe 0000000 it is likely that the sample comes from q but we cannot exclude that it comes from p. This is reflected in the divergence since $D(p\|q) = \infty$ but $D(q\|p) = 1$.

1.3 Mutual information

Definition 1.6. *Let X, Y be discrete random variables taking values in $\mathcal{X}$ and $\mathcal{Y}$ respectively. The mutual information $I(X; Y)$ between X and Y is defined as*

$$I(X;Y) = \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} \mathbb{P}(X = x, Y = y)\log\left(\frac{\mathbb{P}(X = x, Y = y)}{\mathbb{P}(X = x)\mathbb{P}(Y = y})\right).$$

Some movitations:

- Denote with $p_{X,Y}, p_X, p_Y$ the pmfs of $(X, Y), X$ and Y. Then

$$I(X;Y) = D(p_{X,Y}\|p_X p_Y).$$

  Hence, we can regard the mutual information as a measure on how much dependence there is between two random variables.

- Unlike covariance $\mathrm{Cov}(X,Y) = \mathbb{E}[(X-\mathbb{E}[X])(Y-\mathbb{E}[Y])]$, the mutual information $I(X;Y)$ takes into account higher order dependence (not just second order dependence).
- It is bviously that $I(X;Y) = I(Y;X)$.
- Another way to think about mutual information is in terms of entropies

$$I(X;Y) = \mathbb{E}\left[\log\left(\frac{\mathbb{P}(X=x,Y=y)}{\mathbb{P}(X=x)\mathbb{P}(Y=y}\right)\right] = \mathbb{E}[\log(p_{X,Y}(X,Y)) - \log(p_X(X) - \log(p_Y(Y))] = H(X)+H(Y)-H(X,Y).$$

1.4 Conditional entropy/divergence/mutual information

Often we are given additional knowledge by knowing the outcome of another random variable. This motivates to generalise the concepts of entropy, divergence and information by conditioning on this extra information.

Definition 1.7. *Let X, Y be discrete random variables taking values in $\mathcal{X}$. The conditional entropy of Y given X is defined as*

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

In analogy to entropy, it holds that

$$\begin{aligned} H(Y|X) &= -\sum_{x\in\mathcal{X}} \mathbb{P}(X=x)\sum_{y\in\mathcal{X}} \mathbb{P}(Y=y|X=x)\log(\mathbb{P}(Y=y|X=x) \\ &= -\sum_{x\in\mathcal{X}} \mathbb{P}(X=x)\mathbb{E}[\log(p_{Y|X=x}(Y)] \\ &= -\mathbb{E}[\log(p_{Y|X}(Y)]. \end{aligned}$$

An intuitive way to think about $H(X|Y)$ is as the average surprise we have about Y after having observed X (e.g. if $Y=X$ there's no surprise).

Definition 1.8. *Let X be a discrete random variable taking values in $\mathcal{X}$ with pmf p_X, $p(\cdot|x)$ and $q(\cdot|x)$ are two (conditional on the parameter x) pmfs on $\mathcal{X}$ for any $x\in\mathcal{X}$. The divergence between $p(\cdot|X)$ and $q(\cdot|X)$ conditioned on p_X (also known as conditional divergence, conditional Kullback-Leibler divergence, condition relative entropy) is defined as*

$$D(p_{Y|X}\|q_{Y|X}\,|\,p_X) = \sum_{x\in\mathcal{X}} p_X(x)D(p_{Y_1|X=x}\|q_{Y_2|X=x})$$

where random variables X, Y, Y_1, Y_2 are all constructed, such that $p_{Y|X}(y|x) = p(y|x) = p_{Y_1|X}(y|x)$, $q_{Y|X}(y|x) = q(y|x) = p_{Y_2|X}(y|x)$.

Above can be written as

$$D(p_{Y|X}\|q_{Y|X}\,|\,p_X) = \mathbb{E}[D(p_{Y_1|X}(\cdot|X)\|p_{Y_2|X}(\cdot|X))].$$

Definition 1.9. *Let X, Y, Z be discrete random variables taking values in $\mathcal{X}$. The conditional mutual information $I(X;Y|Z)$ (conditioned on Z) between X and Y is defined as*

$$I(X;Y|Z) := H(X|Z) - H(X|Y,Z).$$

Again, we can write this as $I(X;Y|Z) = \mathbb{E}\left[\log\left(\frac{p_{X,Y|Z}(X,Y)}{p_{X|Z}(X)p_{Y|Z}(Y)}\right)\right]$, by which we can see that $I(X;Y|Z) = I(Y;X|Z)$.

In the same way we regard mutual information as measure of dependence, we can regard conditional mutual information as a measure of dependence of two r.v.'s (X,Y) conditional on knowing another random variable (Z).

1.5 Basic properties and inequalities

We prove some basic properties of entropy, divergence and mutual information. We prepare this with two elementary but important inequalities

Lemma 1.10. *(Gibbs' inequality) Let p and q be pmfs on $\mathcal{X}$. Then*

$$-\sum_{x\in\mathcal{X}} p(x)\log(p(x)) \le -\sum_{x\in\mathcal{X}} p(x)\log(q(x))$$

and the equality holds if and only if (iff) $p = q$.

Proof. Denote X a r.v. following the pmf p. Adding $\sum_{x\in\mathcal{X}} p(x)\log(p(x))$ on both sides, we estimate

$$\begin{aligned}\sum_{x\in\mathcal{X}} p(x)\log\left(\frac{p(x)}{q(x)}\right) &= \mathbb{E}\left[-\log\left(\frac{p(X)}{q(X)}\right)\right] \\ &\ge -\log\left(\mathbb{E}\left[\frac{p(X)}{q(X)}\right]\right) \\ &= -\log\left(\sum_{x\in\mathcal{X}} p(x)\frac{q(x)}{p(x)}\right) \\ &= -\log(1) = 0.\end{aligned}$$

where the inequality follows by Jensen's inequality applied to $f(x) = -log(x)$ (a strictly convex function). Note that by Jensen's equality holds iff $\frac{q(x)}{p(x)}$ is constant. □

Put differently, Gibbs' inequality tells us that the minimiser of the map

$$q \mapsto -\mathbb{E}[\log(q(X))]$$

is the pmf p_X and the minimum is $H(X)$.

Lemma 1.11. *(Log sum inequality) Let $a_1, \cdots, a_n; b_1, \cdots, b_n$ are all nonnegative. Then*

$$\sum_{i=1}^{n} a_i \log\left(\frac{a_i}{b_i}\right) \ge \left(\sum_{i=1}^{n} a_i\right)\log\left(\frac{\sum_{i=1}^{n} a_i}{\sum_{i=1}^{n} b_i}\right)$$

with equality holds iff $\frac{a_i}{b_i}$ is constant.

We start with basic properties of the divergence.

Theorem 1.12. *(Divergence properties). Let (X,Y) and $(\hat{X},\hat{Y})$ be 2-dim discrete random variables taking values in $\mathcal{X}\times\mathcal{Y}$. Then*

(1) *(Information inequality)* $D(p_X \| p_{\hat{X}}) \geq 0$ *with equality holds iff* $p_X = p_{\hat{X}}$.

(2) *(Chain rule)* $D(p_{X,Y} \|, p_{\hat{X},\hat{Y}}) = D(p_{Y|X} \| p_{\hat{Y}|\hat{X}} \,|\, p_X) + D(p_X \| p_{\hat{X}})$.

(3) $D(p_{X,Y} \| p_{\hat{X},\hat{Y}}) \geq D(p_X \| p_{\hat{X}})$.

(4) $D(p_{Y|X} \| p_{\hat{Y}|\hat{X}} \,|\, p_X) = D(p_X p_{Y|X} \| p_X p_{\hat{Y}|\hat{X}})$.

(5) *(Convexity) For pmfs* p_1, p_2, q_1, q_2, *we have* $D(\lambda p_1 + (1-\lambda)p_2 \| \lambda q_1 + (1-\lambda)q_2) \leq \lambda D(p_1\|q_1) + (1-\lambda)D(p_2\|q_2)$ *for* $\forall\, \lambda \in [0,1]$.

Proof. Point (1) follows from Gibbs' inequality; Point (2) follows from

$$\begin{aligned}
D(p_{X,Y} \| p_{\hat{X},\hat{Y}}) &= \sum_{x\in\mathcal{X}, y\in\mathcal{Y}} p_{X,Y}(x,y) \log\left(\frac{p_{X,Y}(x,y)}{p_{\hat{X},\hat{Y}}(x,y)}\right) \\
&= \sum_{x\in\mathcal{X}, y\in\mathcal{Y}} p_{X,Y}(x,y) \log\left(\frac{p_X(x)p_{Y|X}(y|x)}{p_{\hat{X}}(x)p_{\hat{Y}|\hat{X}}(y|x)}\right) \\
&= \sum_{x\in\mathcal{X}, y\in\mathcal{Y}} p_{X,Y}(x,y) \log\left(\frac{p_X(x)p_{Y|X}(y|x)}{p_{\hat{Y}|\hat{X}}(y|x)}\right) + \sum_{x\in\mathcal{X}, y\in\mathcal{Y}} p_{X,Y}(x,y) \log\left(\frac{p_X(x)}{p_{\hat{X}}(x)}\right) \\
&= \sum_{x\in\mathcal{X}} p_X(x) \sum_{y\in\mathcal{Y}} p_{Y|X}(y|x) \log\left(\frac{p_X(x)p_{Y|X}(y|x)}{p_{\hat{Y}|\hat{X}}(y|x)}\right) + D(p_X \| p_{\hat{X}}) \\
&= \sum_{x\in\mathcal{X}} p_X(x) D(p_{Y|X} \| p_{\hat{Y}|\hat{X}}) + D(p_X \| p_{\hat{X}}) \\
&= D(p_{Y|X} \| p_{\hat{Y}|\hat{X}}) \,| p_X) + D(p_X \| p_{\hat{X}}).
\end{aligned}$$

With Point (2), and the fact $D(p_1 \| p_2 | p) \geq 0$ for any pmf's p_1, p_2, q, we have Point (3).

Point 4 follows since

$$\begin{aligned}
D(p_{Y|X} \| p_{\hat{Y}|\hat{X}} \,|\, p_X) &= \sum_{x\in\mathcal{X}} p_X(x) \sum_{y\in\mathcal{Y}} p_{Y|X}(y|x) \log\left(\frac{p_{Y|X}(y|x)}{p_{\hat{Y}|\hat{X}}(y|x)}\right) \\
&= \mathbb{E}\left[\log\left(\frac{p_{Y|X}(Y|X)}{p_{\hat{Y}|\hat{X}}(Y|X)}\right)\right] \\
&= \mathbb{E}\left[\log\left(\frac{p_X(X)p_{Y|X}(Y|X)}{p_X(X)p_{\hat{Y}|\hat{X}}(Y|X)}\right)\right] \\
&= D(p_X p_{Y|X} \,\|\, p_X p_{\hat{Y}|\hat{X}}).
\end{aligned}$$

For Point (5), we just need to apply Lemma 1.11 to

$$(\lambda p_1 + (1-\lambda)p_2) \log\left(\frac{\lambda p_1 + (1-\lambda)p_2}{\lambda q_1 + (1-\lambda)q_2}\right),$$

and summ over $x \in \mathcal{X}$.

□

Theorem 1.13. *(Mutual Information properties).*

(1) $I(X;Y) \geq 0$ *with equality holds iff* $X \perp Y$.

(2) $I(X;Y) = I(Y;X) = H(X) - H(X|Y) = H(Y) - H(Y|X)$.

(3) (Information chain rule)

$$I(X_1, \cdot, X_n; Y) = \sum_{i=1}^{n} I(X_i; Y \mid X_{i-1}, \cdots, X_1).$$

(4) (Data-processing inequality) If[2] $(X \perp Z) \mid Y$, *then*

$$I(X;Y) \geq I(X;Z).$$

(5) Let $f : \mathcal{X} \mapsto \mathcal{Y}$. *Then* $I(X;Y) \geq I(X; f(Y))$.

Proof. Point (1) follows since $I(X;Y) = D(p_{X,Y} \| p_X p_Y) \geq 0$ by the information inequality in Theorem 1.12.

The first equality in Point (2) follows from the definition of mutual information. The others follow since

$$\begin{aligned} I(X;Y) &= \mathbb{E}\left[\log\left(\frac{p_{X,Y}(X,Y)}{p_X(X)p_Y(Y)}\right)\right] \\ &= H(X) + H(Y) - H(X,Y), \end{aligned}$$

and

$$\begin{aligned} H(X,Y) &= -\sum_{x\in\mathcal{X}, y\in\mathcal{Y}} \mathbb{P}(X=x, Y=y) \log(\mathbb{P}(Y=y, X=x)) \\ &= -\sum_{x\in\mathcal{X}, y\in\mathcal{Y}} \mathbb{P}(X=x, Y=y)[\log(\mathbb{P}(Y=y|X=x)) + \log(\mathbb{P}(X=x))] \\ &= H(Y|X) + H(X). \end{aligned}$$

Notice that the last equality can be easily extended to

$$H(X_1, \cdots, X_n) = H(X_n|X_{n-1}, \cdots, X_1) + H(X_{n-1}, \cdots, X_1) = \sum_{i=1}^{n} H(X_i|X_{i-1}, \cdots, X_1).$$

with the notation $H(X_1|X_0) = H(X_1)$. Furthermore, we can have the conditional version

$$H(X_1, \cdots, X_n \mid Y) = \sum_{i=1}^{n} H(X_i|X_{i-1}, \cdots, X_1, Y).$$

Point (3) follows since

$$\begin{aligned} I(X_1, \cdots, X_n; Y) &= H(X_1, \cdots, X_n) - H(X_1, \cdots X_n \mid Y) \\ &= \sum_{i=1}^{n} \{H(X_i|X_{i-1}, \cdots, X_1) - H(X_i|X_{i-1}, \cdots, X_1, Y)\} \\ &= \sum_{i=1}^{n} I(X_i; Y \mid X_{i-1} \cdots, X_1), \end{aligned}$$

[2] Recall that X and Z are conditionally independent given Z (denoted as $(X \perp Z) \mid Y$) if $p_{(X,Z)|Y}(x, z|y) = p_{X|Y}(x|y)p_{Z|Y}(z|y)$. This is equivalent to (X, Y, Z) is a Markovian process with 3 time spots, which can be described by $p_{X,Y,Z}(x, y, z) = p(x)p(y|z)p(z|y)$.

where the last line follows directly by definition of conditional entropy. For Point (4) we use the chain rule (3) to write $I(Y, Z; X) = I(Y; X) + I(Z; X|Y) = I(Y; X)$. On the other hand, $I(Y, Z; X) = I(Z, Y; X) = I(Z; X) + I(Y; X|Z) \geq I(Z; X)$, so $I(X; Y) \geq I(X; Z)$.

Finally, Point (5) follows from the data-processing inequality in Point (4) by taking $Z = f(Y)$. □

Remark 1.14.

- Point (2) applied with $X = Y$ shows $I(X; X) = H(X)$ which explains why entropy is sometimes referred to as self-information.
- Point (2) motivates $I(X; Y)$ as a measure of the reduction in uncertainty that knowing either variable gives about the other.
- Despite its simple form and proof, the data processing inequality in Point (5) formalises the intuitive but fundamental concept: post-processing cannot increase information; e.g., if Z is a r.v. that depends only on Y , then Z can not contain more information about X than Y.
- Recall from Statistics that an estimator $T(X)$ for a parameter $\theta \in \Theta$ is called sufficient if conditional on $T(X)$, the distribution of X does not depend on θ. This is equivalent to $I(\theta; X) = I(\theta; T(X))$ under all distributions in $\{p_\theta : \theta \in \Theta\}$.

Theorem 1.15. *(Entropy properties). Let X, Y be discrete random variables taking values in $\mathcal{X}$. $|\mathcal{X}|$ is the number of elements in $\mathcal{X}$.*

(1) $0 \leq H(X) \leq \log(|\mathcal{X}|)$. *The upper bound is attained iff X is uniformly distributed on X, the lower bound is attained iff X is constant with probability 1.*

(2) $0 \leq H(X|Y) \leq H(X)$ *and* $H(X|Y) = H(X)$ *iff X and Y are independent,* $H(X|Y) = 0$ *iff. $X = f(Y)$ for some function f.*

(3) (Chain rule) $H(X_1, \cdots, X_n) = \sum_{i=1}^n H(X_i|X_{i-1}, \cdots, X_1) \leq \sum_{i=1}^n H(X_i)$ *with equality iff the X_i are independent.*

(4) For $f : \mathcal{X} \mapsto \mathcal{Y}$, $H(f(X)) \leq H(X)$ with equality iff f is bijective.

(5) Let X and Y be i.i.d., then

$$\mathbb{P}(X = Y) \geq 2^{-H(X)}$$

with equality iff they are uniformly distributed.

(6) $H(X)$ is concave in p_X.

Proof. For Point (1), the lower bound follows by definition of entropy; for the upper bound, we apply Gibbs' inequality with $q(x) = |\mathcal{X}|^{-1}$ to get

$$H(X) \leq -\sum_{x \in \mathcal{X}} p(x) \log(q(x)) = \log(|\mathcal{X}|).$$

Since equality holds in Gibbs' inequality iff $p_X = q$, it follows that X must be uniformly distributed to attain the upper bound. Similarly, since each term in the sum is zero iff $p(x) = 0$ or $p(x) = 1$ and there can be just one x with $p(x) = 1$, which shows that X must be constant to have zero entropy.

For Point (2), we use that $0 \leq I(X;Y) = H(X) - H(X|Y)$ by Theorem 1.13 so both bounds follow. The upper bound is attained iff X, Y are independent. For the lower bound, note that by definition

$$H(X|Y) = \sum_{y \in \mathcal{Y}} p_Y(y) H(X|Y = y),$$

where $H(X|Y = y) = -\sum_{x \in \mathcal{X}} p_{X|Y}(x|y) \log(p_{X|Y}(x|y))$. Hence, $H(X|Y) = 0$ iff $H(X|Y = y) = 0$ for all y in the support of Y. But by Point(1) this only happens if $\mathbb{P}(X = x|Y = y) = 1$ for some constant $x = f(y)$. This implies that $X = f(Y)$.

Point (3) follows as in the proof of the Point (3) in Theorem 1.13, and the fact that $H(X_i|X_{i-1}, \cdots, X_1) = H(X_i)$ iff X_i and $X_{i-1}, \cdots, X_1$ are independent.

Point (4) follows since

$$H(X, f(X)) = H(X) + H(f(X)|X) = H(X)$$

and

$$H(X, f(X)) = H(f(X), X) = H(f(X)) + H(X|f(X)) \geq H(f(X)).$$

So $H(f(X)) \leq H(X)$, and the equality holds iff $H(X|f(X)) = 0$, which is equivalent to that f is bijective.

Point (5) follows from Jensen's inequality,

$$2^{-H(X)} = 2^{\mathbb{E}[\log(p_X(X))]} \leq \mathbb{E}[2^{\log(p_X(X))}] = \mathbb{E}[p_X(X)] = \sum_{x \in \mathcal{X}} p_X(x) p_X(x) = \mathbb{P}(X = Y).$$

Point(6) follows from $g(x) = -x \log x$ is a concave function over $x \in (0, 1)$. □

Remark 1.16.

- Point (1) is especially intuitive if we think of entropy as the average surprise we have about X.
- Point (2) formalises "more information is better".
- Point (4) shows that entropy is invariant under relabelling of observations.

1.6 Fano's inequality

A common situation is that we use an observation of a random variable Y to infer the value of a random variable X. If $\mathbb{P}(X \neq Y) = 0$, then $H(X|Y) = 0$ by Point (2) in Theorem 1.15. We expect that if $\mathbb{P}(X \neq Y)$ is small, then $H(X|Y)$ should be small. Fano's inequality makes this precise.

Theorem 1.17. *(Fano's inequality, 1966). Let X, Y be discrete random variables taking values in $\mathcal{X}$. Then*

$$H(X|Y) \leq H(\mathbf{1}_{X \neq Y}) + \mathbb{P}(X \neq Y) \log(|\mathcal{X}| - 1).$$

Alternatively we can interpret Fano's inequality as giving a lower bounds on the error probability $\mathbb{P}(X|Y)$ and this is how we will apply to get bounds on information transmission over noisy channels in Chapter 3.

Proof. Set $Z = \mathbf{1}_{X \neq Y}$ and note that $H(Z|X,Y) = 0$. Now

$$\begin{aligned}
H(X|Y) &= H(X|Y) + H(Z|X,Y) \\
&= H(X,Z|Y) \\
&= H(Z|Y) + H(X|Y,Z) \\
&\leq H(Z) + H(X|Y,Z) \\
&\leq H(Z) + \sum_{y \in \mathcal{X}} [\mathbb{P}(Y=y, Z=0) H(X|Y=y, Z=0) + P(Y=y, Z=1) H(X|Y=y, Z=1)].
\end{aligned}$$

Now $\{Y = y, Z = 0\}$ implies $\{X = y\}$, hence $H(X|Y = y, Z = 0) = 0$. On the other hand, $\{Y = y, Z = 1\}$ implies that $\{X \in \mathcal{X} \backslash \{y\}\}$ which contains $|\mathcal{X}| - 1$ elements. Therefore,

$$H(X|Y = y, Z = 1) \leq \log(|\mathcal{X}| - 1).$$

It follows that

$$\begin{aligned}
H(X|Y) &\leq H(Z) + \sum_{y \in \mathcal{X}} \mathbb{P}(Y=y, Z=1) H(X|Y=y, Z=1) \\
&\leq H(Z) + \mathbb{P}(Z=1) \log(|\mathcal{X}| - 1).
\end{aligned}$$

□

Corollary 1.18. $H(X|Y) \leq 1 + \mathbb{P}(X \neq Y) \log(|\mathcal{X}| - 1)$.

Chapter 2

Typical Sequences

Given a discrete distribution, what can we infer about one sample from this distribution? Not much! An elementary but far reaching insight of Shannon is that this changes drastically if we deal with sequences of observations and that the entropy $H(X_1, \cdots, X_n) = nH(X)$ measures the average storage cost of sequences of length n.

Example 2.1. Denote with X a discrete r.v. with state space $\mathcal{X} = \{0, 1\}$ and $X_1, \cdots, X_n$, i.i.d. copies of X. A sequence $(x_1, \cdots, x_n) \in \{0, 1\}^n$ occurs withs probability

$$\mathbb{P}((X_1, \cdots, X_n) = (x_1, \cdots, x_n)) = p^{z(x_1, \cdots, x_n)} q^{o(x_1, \cdots, x_n)}, \tag{2.0.1}$$

where $p = \mathbb{P}(X = 0)$, $q = 1 - p$ and $z(x_1, \cdots, x_n) = \sum_i \mathbf{1}_{x_i = 0}$, $o(x_1, \cdots, x_n) = \sum_i \mathbf{1}_{x_i = 1}$. Now for a "typical sequence" $(x_1, \cdots, x_n)$, we can approximate the numbers of 0's and 1's by $z(x_1, \cdots, x_n) \approx \mathbb{E}[z(X_1, \cdots, X_n))] = np$ and $o(x_1, \cdots, x_n) \approx \mathbb{E}[o(X_1, \cdots, X_n))] = nq$. Hence,

$$\mathbb{P}((X_1, \cdots, X_n) = (x_1, \cdots, x_n)) \approx p^{np} q^{nq}.$$

Taking the logaritim on both sides of these approximiation, we get

$$-\log(\mathbb{P}((X_1, \cdots, X_n) = (x_1, \cdots, x_n))) \approx np \log(p) + nq \log(q) = nH(X).$$

Thus for a "typical sequence" $(x_1, \cdots, x_n) \in \{0, 1\}^n$,

$$\mathbb{P}((X_1, \cdots, X_n) = (x_1, \cdots, x_n)) \sim 2^{-nH(X)}.$$

Therefore the set of typical sequences of length n consists of approximately $2^{nH(X)}$ elements, each occurring with approximate probability $2^{-nH(X)}$. Finally, note that $2^{nH(X)} \leq 2^n$, and this difference can be very large.

Above informal calculation suggests to partition $\mathcal{X}^n$ in two sets,

- "typical sequences" and
- "atypical sequences".

The set of "typical sequences" forms a potentially relatively small subset of $\mathcal{X}^n$, that however carries most of the probability mass and its elements occur with approximately the same probability. This elementary but fundamental insight is due to Shannon and has important consequences for coding.

In the rest of this section, we extend and make above informal discussion rigorous.

2.1 Weak typicality and the asymptotic equipartition property (AEP)

Theorem 2.2. *(Weak AEP 1) Let X be a discrete random variable. Then*

$$-\frac{1}{n}\log(p_{X_1,\cdots X_n}(X_1,\cdots X_n)) \overset{\text{in prob.}}{\longrightarrow} H(X) \qquad \text{as } n\to+\infty. \tag{2.1.1}$$

Proof. By independence, $-\log(p_{X_1,\cdots X_n}(X_1,\cdots X_n)) = \sum_{i=1}^n \log(p_X(X_i))$ and $\mathbb{E}[-\log(p_X(X_i))] = H(X)$. The result follows from the (weak) law of large numbers. □

Theorem 2.2 suggests the following definition of "typical sequences".

Definition 2.3. *For any $n\in\mathbb{N}$, any $\varepsilon>0$, we call*

$$\mathcal{T}_n^\varepsilon := \left\{(x_1,\cdots,x_n)\in\mathcal{X}^n : \left|-\frac{1}{n}\log(p_{X_1,\cdots X_n}(x_1,\cdots x_n)) - H(X)\right| \le \varepsilon\right\}$$

the set of (weakly) typical sequences of length n of the random variable X (with error ε).

Theorem 2.4. *(Weak AEP 2). For all $\varepsilon>0$, there exists an $n_0\in\mathbb{N}$ such that for every $n>n_0$,*

(1) $p_{X_1,\cdots X_n}(x_1,\cdots x_n)\in[2^{-n(H(X)+\varepsilon)}, 2^{-n(H(X)-\varepsilon)}]$ for any $(x_1,\cdots,x_n)\in\mathcal{T}_n^\varepsilon$;

(2) $\mathbb{P}((X_1,\cdots,X_n)\in\mathcal{T}_n^\varepsilon)\ge 1-\varepsilon$;

(3) $|\mathcal{T}_n^\varepsilon|\in[(1-\varepsilon)2^{n(H(X)-\varepsilon)}, 2^{n(H(X)+\varepsilon)}]$.

Moreover, for Point (1) one can take $n_0=0$.

Proof. Point (1) follows directly from Definition 2.3 for $n_0=0$. Point (2) follows by Theorem 2.2, since for every $\varepsilon>0$,

$$\mathbb{P}((X_1,\cdots,X_n)\notin\mathcal{T}_n^\varepsilon) = \mathbb{P}\left(|\log(p_{X_1,\cdots,X_n}(X_1,\cdots,X_n) - H(X)|>\varepsilon\right),$$

which converges to 0 as $n\to+\infty$.

For the upper bound in Point (3), it follows since

$$\begin{aligned} 1 &= \sum_{(x_1,\cdots,x_n)\in\mathcal{X}^n} p_{X_1,\cdots,X_n)}(x_1,\cdots,x_n) \\ &\ge \sum_{(x_1,\cdots,x_n)\in\mathcal{T}_n^\varepsilon} p_{X_1,\cdots,X_n}(x_1,\cdots,x_n) \\ &\ge \sum_{(x_1,\cdots,x_n)\in\mathcal{T}_n^\varepsilon} 2^{-n(H(X)+\varepsilon)}. \end{aligned}$$

For the lower bound, we know by Point (2) that the probability $\mathbb{P}((X_1, \cdots, X_n) \in \mathcal{T}_n^\varepsilon)$ converges to 1, so for n large enough,

$$1 - \varepsilon \leq \mathbb{P}((X_1, \cdots, X_n) \in \mathcal{T}_n^\varepsilon) \leq \sum_{(x_1, \cdots, x_n) \in \mathcal{T}_n^\varepsilon} 2^{-n(H(X)-\varepsilon)} = 2^{-n(H(X)-\varepsilon)} |\mathcal{T}_n^\varepsilon|,$$

and then we get the lower bound. □

Remark 2.5.

- When n is large, above suggests to think of $(X_1, \cdots, X_n)$ as being drawn uniformly from $\mathcal{T}_n^\varepsilon$ with probability $2^{-nH(X)}$.
- Theorem 2.4 does not imply that most sequences are elements of $\mathcal{T}_n^\varepsilon$: $\mathcal{T}_n^\varepsilon$ has rather small cardinality compared to $\mathcal{X}^n$ since
$$\frac{|\mathcal{T}_n^\varepsilon|}{|\mathcal{X}^n|} \approx \frac{2^{nH(X)}}{2^{n \log(|\mathcal{X}|)}} = 2^{-n(\log(|\mathcal{X}|) - H(X))},$$
and the last ratio converges to 0 when $n \to +\infty$ unless $H(X) = \log(|\mathcal{X}|)$, which holds iff X is uniformly distributed by Theorem 1.15. However, $\mathcal{T}_n^\varepsilon$ carries most of the probability mass, as shown in Point (2) in Theorem 2.4.
- Theorem 2.4 allows to prove a property for typical sequences and then conclude that this property holds for random sequences $(X_1, \cdots, X_n)$ with high probability.
- The most likely sequence $x^* = \text{argmax}_x \mathbb{P}((X_1, \cdots, X_n) = x)$ is in general not an element of $\mathcal{T}_n^\varepsilon$. For example, take $\mathcal{X} = \{0, 1\}$ and $\mathbb{P}(X = 1) = 0.9$, then $(1, \cdots, 1)$ is the most likely sequence but not typical since $-\frac{1}{n} \log(p_{X_1, \cdots, X_n}(1, ..., 1) = -\log(0.9)) \approx 0.11$ is not close to $H(X) = -0.1 \log(0.1) - 0.9 \log(0.9) \approx 0.46$. Note that as $n \to +\infty$, the probability of every sequence, thus also the most likely sequence, tends to 0.

2.2 Source coding with block codes

We receive sequence in set $\mathcal{X}$ (e.g. a sequence of letter from the english alphabet) and we want to store this message, e.g. on our computer so using a sequence of 0's and 1's.

Definition 2.6. *For a finite set A, denote with A^* the set of finite sequences in A. For $a = a_1 \cdots a_n \in A^*$ with all $a_i \in A$, we call $|a| = n$ the length of the sequence $a \in A^*$.*

That is, to encode $\mathcal{X}$, we look for a map $c : \mathcal{X} \longrightarrow A^*$ that allows to recover any sequence in $\mathcal{X}$ from the associated sequence in A^*. If we have knowledge about the distribution of the sequence in $\mathcal{X}$ we can try to minimise the expected storage cost (e.g. $A = \{0, 1\}$ the number of bits on our computer needed to store this message). Using the AEP we associate short codewords with sequences in the typical set, and long codewords with the remaining atypical sequence. This gives a bound on the expected length of the encoded sequence by the entropy.

Theorem 2.7. *(Source coding 1, Shannon's first theorem). Let X be discrete random variable with state space $\mathcal{X}$. For every $\varepsilon > 0$, there exists an integer n, and a map*

$$c : \mathcal{X}^n \longrightarrow \{0, 1\}^*$$

such that

(1) the map $\cup_{k\geq 0}\mathcal{X}^{nk} \longrightarrow \{0,1\}^*$ *given by* $(x_1,\cdots,x_k)\mapsto c(x_1)\cdots c(x_k)\in\{0,1\}^*$ *is injective;*

(2) $\frac{1}{n}\mathbb{E}[|c(X_1,\cdots,X_n)|]\leq H(X)+\varepsilon$.

Proof. For some $\varepsilon_0>0$, we split $\mathcal{X}^n$ into the disjoint sets $\mathcal{T}_n^{\varepsilon_0}$ and $\mathcal{X}^n\backslash\mathcal{T}_n^{\varepsilon_0}$, and order the elements in $\mathcal{T}_n^{\varepsilon_0}$ and $\mathcal{X}^n\backslash\mathcal{T}_n^{\varepsilon_0}$ (in some arbitrary order; e.g. lexicographic). By the AEP, there are at most $2^{n(H(X)+\varepsilon_0)}$ elements in $\mathcal{T}_n^{\varepsilon_0}$, hence we can associate with every element of $\mathcal{T}_n^{\varepsilon_0}$ a string consisting of $l_1:=\lceil n(H(X)+\varepsilon_0)\rceil$ bits[1]; similarly we associate with every element of $\mathcal{X}^n\backslash\mathcal{T}_n^{\varepsilon_0}$ a unique string of $l_2=\lceil n\log(|\mathcal{X}|)\rceil$ bits. Now define $c(x_1,\cdots,x_n)$ as these strings with length l_1 resp. l_2 bits, prefixed by a 0 if $(x_1,\cdots,x_n)$ is in $\mathcal{T}_n^{\varepsilon_0}$, and prefixed by 1 otherwise. Clearly, this injective (hence a bijection on its image) and the prefix 0 or 1 indicates how many bits follow. This block code has expected length

$$\begin{aligned}
\mathbb{E}[|c(X_1,\cdots,X_n)|] &= \sum_{x\in\mathcal{T}_n^{\varepsilon_0}} p(x)(l_1+1)+\sum_{x\notin\mathcal{T}_n^{\varepsilon_0}} p(x)(l_2+1)\\
&\leq \sum_{x\in\mathcal{T}_n^{\varepsilon_0}} p(x)(n(H(X)+\varepsilon_0)+2)+\sum_{x\notin\mathcal{T}_n^{\varepsilon_0}} p(x)(n\log(|X|)+2))\\
&\leq \mathbb{P}((X_1,\cdots,X_n)\in\mathcal{T}_n^{\varepsilon_0})(n(H(X)+\epsilon_0)+2)+\mathbb{P}((X_1,\cdots,X_n)\notin\mathcal{T}_n^{\varepsilon_0})(n\log(|X|)+2)\\
&\leq n(H(X)+\varepsilon_0)+2+\varepsilon_0 n\log(|X|)\\
&= n(H(X)+\varepsilon_1)
\end{aligned}$$

with $\varepsilon_1:=\varepsilon_0(1+\log(|X|))+\frac{2}{n}$. For a given $\varepsilon>0$, we first choose ε_0 small enough such that $\varepsilon_0(1+\log(|X|))<\varepsilon/2$, and then n sufficiently large such that $\frac{2}{n}\leq\varepsilon/2$. □

Shannon's first theorem shows that we encode sequence $X_1,\cdots,X_n$ using on average no more than $nH(X)$. Put it differently: on average we need $H(X)$ bits to encode one symbol from this sequence. We will prove in Chapter 3, Theorem 3.9, that above bound is sharp. Hence, this leads to another, more operational interpretation of entropy of a random variable, namely as a compression bound of messages that are generated by sampling from a distribution.

2.3 Non i.i.d. source coding (not examinable)

Of course, the assumption that the sequence is generated by i.i.d. draws from the same distribution is not realistic (e.g. sentence seen as sequences of letters, etc). However, this assumption can be significantly weakened and this is the content of the Shannon–McMillan–Breiman Theorem[2]:

Theorem 2.8. *(Shannon–McMillan–Breiman). Let* $X_1,X_2,\cdots$ *be an ergodic and stationary sequence of random variables in a finite state space* $\mathcal{X}$. *Then*

$$-\frac{1}{n}\log(p_{X_1,\cdots,X_n}(X_1,\cdots,X_n)) \overset{\text{in prob.}}{\longrightarrow} \bar{H},\ \text{as } n\to+\infty,$$

where $\bar{H}:=\lim_{n\to+\infty}\frac{1}{n}H(X_1,\cdots,X_n)$.

[1] here $\lceil x\rceil$ means the lowest integer no less than x.

[2] The version here is due to Breiman, there are many extension (a.s. convergence, non-stationary, etc); see [1] for reference

(A sequence is stationary if $X_i, \cdots, X_{i+n}$ has the same law for all i. Loosely speaking, a sequence is ergodic if the time average over one realisaition equals the expectation. The class of of stationary and ergodic processes is large and covers many important processes). One can then modify Theorem 2.4 and adapt Shannon's block coding argument of Theorem 2.7.

2.4 Strong typicality

Above relies on the idea that we associate with sequences that appear often short codewords, and with rare sequence long codewords. Hence, we would ask if there are sets with smaller cardinality than $\mathcal{T}_n^\varepsilon$ that still carry most of the pmf.

Definition 2.9. *Denote with* $\mathcal{S}_n^\varepsilon$ *the smallest subset of* $\mathcal{X}^n$ *such that*

$$\mathbb{P}((X_1, \cdots, X_n) \in \mathcal{S}_n^\varepsilon) \geq 1 - \varepsilon.$$

We can construct this set by ordering sequences by their probability and adding them until the probability mass is greater or equal $1 - \varepsilon$.

Proposition 2.10. *Let* $(\varepsilon_n)_n$ *be a strictly positive sequence such that* $\lim_{n \to +\infty} \varepsilon_n = 0$. *Then*

$$\lim_{n \to +\infty} \left\{ \lim_{m \to +\infty} \frac{1}{m} \log \left(\frac{|\mathcal{S}_m^{\varepsilon_n}|}{|\mathcal{T}_m^{\varepsilon_n}|} \right) \right\} = 0.$$

In other words, the set of strong and weak typical sequences have the same number of elements up to first order in the exponent. Hence, we do not gain by working with strong typical sequences instead of weak typical sequences although its construction appears at first sight to be more efficient than that of $\mathcal{T}_n^\varepsilon$. Nevertheless, one could argue that the definition of $\mathcal{S}_n^\varepsilon$ is simpler and that we should have derived the source coding Theorem, Theorem 2.7, directly using $\mathcal{S}_n^\varepsilon$ instead of $\mathcal{T}_n^\varepsilon$. However, note that the proof relies on counting the elements of the set of "typical sequences": using $\mathcal{T}_n^\varepsilon$ this is trivial due to the "uniform distribution" elements in $\mathcal{T}_n^\varepsilon$, but this is much harder for $\mathcal{S}_n^\varepsilon$ and only Proposition 2.10 tells us the answer.

Chapter 3

Optimal Codes

We have used the AEP to construct a block code that compresses messages generated by i.i.d. samples from a random variable X. In this section we want to use symbol codes to compress, that is to associate with every element of $\mathcal{X}$ a sequence of bits (or more generally, a sequence of elements in a given set).

3.1 Symbol codes and Kraft–McMillan

Definition 3.1. *For a finite set $\mathcal{X}$, denote with $\mathcal{X}^*$ the set of finite sequences (also called strings) in $\mathcal{X}$. For $x = x_1 \cdots x_n \in \mathcal{X}^*$ with $x_i \in \mathcal{X}$ for all $i = 1, \cdots, n$, we call $|x| = n$ the length of the sequence $x \in \mathcal{X}^*$. Given two finite sets $\mathcal{X}$ and $\mathcal{Y}$, we call a function $c : \mathcal{X} \longrightarrow \mathcal{Y}^*$ a symbol code, and call $c(x) \in \mathcal{Y}^*$ the codeword of $x \in \mathcal{X}$. In this context, $\mathcal{Y}$ is called a d-ary if $|\mathcal{Y}| = d$.*

Since we need to recover the original sequence $x_1 \cdots x_n \in \mathcal{X}^*$ given $c(x_1) \cdots c(x_n) \in \mathcal{Y}^*$, we need to restrict attention to codes c that are injective. However, this is not sufficient

Example 3.2. Let $\mathcal{X} = \{1, \cdots, 6\}$ and $c(x)$ be the binary expansion, i.e. the source code is a binary code with codewords $\{1, 10, 11, 100, 101, 110\}$. In general, we can not recover the original sequence, e.g. 110 might correspond to $x_1 = 6$ or $x_1 x_2 = 12$.

Ideally, we are looking for a code that allows to recover the original message, and it is easy to decode in practice and compresses the original message as much as possible. To make all this rigorous, we define different classes of codes.

Definition 3.3. *Let $c : \mathcal{X} \longrightarrow \mathcal{Y}^*$ be a source code. We denote with $c^* : \mathcal{X}^* \longrightarrow \mathcal{Y}^*$ the extension of c to $\mathcal{X}^*$ by $c^*(x_1 \cdots x_n) = c(x_1) \cdots c(x_n)$. We say that c is*

(1) unambiguous if c is injective,

(2) uniquely decodable if c^* is injective,

(3) a prefix code, if no codeword of c is the prefix of another codeword of c. That is, there does not exist $x_1 \in \mathcal{X}, x_2 \in \mathcal{X}$ such that $c(x_1)y = c(x_2)$ for some $y \in \mathcal{Y}^*$. Prefix codes are also known as instantaneous codes.

Clearly,

$$\{\text{ prefix codes}\} \subset \{\text{ uniquely decodable codes }\} \subset \{\text{ unambigiuous codes}\},$$

and we are just interested in uniquely decodable codes. In general it is not easy to check if a given code is unique decodable; moreover, even if a code is uniquely decodable it can be very difficult/computationally expensive do decode.

Example 3.4. Take $\mathcal{X} = \{A, B, C, D\}, \mathcal{Y} = \{0, 1\}$. Then $c(A) = 0, c(B) = 01, c(C) = 011, c(D) = 111$ is uniquely decodable although this not completely trivial to see. Note that describing a decoding algorithm is not easy either.

On the other hand, prefix codes are trivial to decode. A surprising result is that we can restrict attention to the design of prefix codes without increasing the length of code words.

Theorem 3.5. *(1) Let $c : \mathcal{X} \longrightarrow \mathcal{Y}^*$ be uniquely decodable and set $l_x = |c(x)|$. Then*

$$\sum_{x\in\mathcal{X}} |\mathcal{Y}|^{-l_x} \leq 1. \tag{3.1.1}$$

(2) Conversely, given $(l_x)_{x\in\mathcal{X}} \subset \mathbb{N}$ and a finite set $\mathcal{Y}$ such that (3.1.1) holds, there exists a prefix code $c : \mathcal{X} \longrightarrow \mathcal{Y}^*$ such that $|c(x)| = l_x$ for $\forall x \in \mathcal{X}$.

Proof. Set $d = |\mathcal{Y}|$ and $l_{max} = \max_{x\in\mathcal{X}} |c(x)|$, $l_{min} = \min_{x\in\mathcal{X}} |c(x)|$. If we denote with $a(k)$ the number of source sequences mapping to codewords of length k, then

$$\left(\sum_{x\in\mathcal{X}} d^{-|c(x)|}\right)^n = \sum_{k=n*l_{min}}^{n*l_{max}} a(k)d^{-k}.$$

Unique decodability implies $a(k) \leq d^k$, hence $\sum_{x\in\mathcal{X}} d^{-|c(x)|} \leq (n(l_{max} - l_{min}) + 1)^{1/n}$. Letting $n \to +\infty$ shows the result.

Let $(l_x)_{x\in\mathcal{X}}$ be a set of integers that fulfils (3.1.1) and set $\mathcal{Y}$. By relabelling, identify $\mathcal{X}$ as the set $\{1, \cdots, |\mathcal{X}|\} \subset \mathbb{N}$ and assume $l_1 \leq l_2 \leq \cdots \leq l_{|\mathcal{X}|}$. Define $r_m := \sum_{i=1}^{m-1} |\mathcal{Y}|^{-l_i}$ for any $m \leq |\mathcal{X}|$, which satisfies $r_m \leq 1$ by the assumption. Define $c(m)$ as the first l_m digits in the $|\mathcal{Y}|$-ary expansion[1] of the real number $r_m \in [0, 1)$, that is c$(m) := y_1 \cdots y_{l_m}$, where

$$r_m = \sum_{i\geq 1} y_i |\mathcal{Y}|^{-i}.$$

This must be a prefix code: if not, there exists m, n with $m < n$, and $c(m)$ a prefix of $c(n)$ and therefore the first l_m digits of r_m and r_n in the $|\mathcal{Y}|$-ary expansion coincide which in turn implies $r_n - r_m < |\mathcal{Y}|^{-l_m}$; on the other hand, by the very definition of r_m and r_n we have $r_n - r_m = \sum_{i=m}^{n-1} |\mathcal{Y}|^{-l_i} \geq |\mathcal{Y}|^{-l_m}$, which is a contradiction. □

Remark 3.6. The inequality (3.1.1) is called Kraft-McMillan inequality. Under the stronger assumption that p is a prefix code in Point (1), the above Theorem 3.5 has a nice proof using trees (plan to appear in Sheet 3). Kraft showed above theorem under this extra assumption. Theorem 3.5 as stated above is due to McMillan (based on Kraft's work). Yet another proof of Point (1) can be given using the "probabilistic method" (also planned to appear in Sheet 3) which we will encounter again.

Corollary 3.7. *For any uniquely decodable code there exists a prefix code with the same codeword lengths.*

[1] With the usual convention that an infinite number of zeros appears, e.g. with $d = 2$, $\frac{1}{2}$ has the expansion 2^{-1} and not $\sum_{i=2}^{+\infty} 2^{-i}$.

3.2 Optimal codes

So far, we have not made any assumptions on how the messages that we want to encode are generated. We now study the case, when the messages are generated by independent samples from a discrete random variable X and our goal is to minimise the average codeword length.

Definition 3.8. *We call a symbol code $c : \mathcal{X} \longrightarrow \mathcal{Y}^*$ optimal for a random variable X with pmf p on $\mathcal{X}$ and a finite set $\mathcal{Y}$, if it minimises $\mathbb{E}[|c'(X)|]$ among all uniquely decodable codes $c' : \mathcal{X} \longrightarrow \mathcal{Y}^*$.*

In view of Kraft–McMillan inequality, given a set $\mathcal{Y}$ a code $c : \mathcal{X} \longrightarrow \mathcal{Y}^*$ is optimal if it solves the constraint minimisation problem

$$\begin{array}{ll} \text{Minimise} & \sum_{x\in\mathcal{X}} p(x) l_x \\ \text{s.t.} & \sum_{x:p(x)>0} d^{-l_x} \le 1 \text{ and } (l_x)_{x\in\mathcal{X}} \subset \mathbb{N}. \end{array} \tag{3.2.1}$$

This is an integer programming problem, and such problems are in general (computationally) hard to solve. To get an idea about what to expect, let us first neglect the integer constraint $l_x \in \mathbb{N}$ and assume $\sum d^{-l_x} = 1$. This in turn is a simple optimisation problem that can for example be solved using Lagrangian multipliers, i.e. differentiating

$$\sum_{x\in\mathcal{X}} p(x) l_x - \lambda \left(\sum_{x\in\mathcal{X}} d^{-l_x} - 1 \right)$$

after l_x and setting the derivative to 0 gives $l_x = -\log_d(p(x))$ and it remains to verify that this is indeed a minimum. This would give (still ignoring the integer constraint) an expected length $\mathbb{E}[|c(X)|] = -\sum p(x)\log_d(p(x)) = H_d(X)$. Instead of using Lagrange multipliers we make this rigorous using a direct argument involving just basic properties of entropy and divergence from Chapter 1.

Theorem 3.9. *(Source coding for symbol codes). Let X be a random variable taking values in a finite set $\mathcal{X}$ and c a uniquely decodable, d-ary source code. Then*

$$H_d(X) \le \mathbb{E}[|c(X)|],$$

and the equality holds iff $|c(x)| = -\log_d(\mathbb{P}(X = x))$.

Proof. Set $l_x := c(x)$ and $q(x) = \frac{d^{-l_x}}{\sum_{x\in\mathcal{X}} d^{-l_x}}$, we have,

$$\begin{aligned} \mathbb{E}[|c(X)|] - H_d(X) &= \sum_{x\in\mathcal{X}} p(x) l_x + \sum_{x\in\mathcal{X}} p(x)\log_b(p(x)) \\ &= -\sum_{x\in\mathcal{X}} p(x)\log_d(d^{l_x}) + \sum_{x\in\mathcal{X}} p(x)\log_b(p(x)) \\ &= -\sum_{x\in\mathcal{X}} p(x)\log_d\left(q(x)\sum_{x'\in\mathcal{X}} d^{-l_{x'}}\right) + \sum_{x\in\mathcal{X}} p(x)\log_b(p(x)) \\ &= -\sum_{x\in\mathcal{X}} p(x)\log_d\left(\sum_{x'\in\mathcal{X}} d^{-l_{x'}}\right) + \sum_{x\in\mathcal{X}} p(x)\log_b\left(\frac{p(x)}{q(x)}\right) \\ &= -\log_b\left(\sum_{x'\in\mathcal{X}} d^{-l_{x'}}\right) + D_d(p\| q) \\ &\ge 0, \end{aligned}$$

where used that by Kraft–McMillan's inequality (3.1.1) $\sum_{x'\in\mathcal{X}} d^{-l_{x'}} \leq 1$ and that divergence is non-negative. Note that the equality holds iff $\sum_{x'\in\mathcal{X}} d^{-l_{x'}} = 1$ and $D(p\| q) = 0$. Since $D(p\| q) = 0$ implies $p = q$, the result follows by definition of q. □

Proposition 3.10. *Let X be a random variable taking values in a finite set $\mathcal{X}$ and $\mathcal{Y}$ a d-ary set. There exists an optimal code c^* and*

$$H_d(X) \leq \mathbb{E}[|c^*(X)|] < H_d(X) + 1. \tag{3.2.2}$$

Proof. Set $l_x := \lceil -\log_d(p(x))\rceil$ and note that $\sum_{x\in\mathcal{X}} d^{-l_x} \leq \sum_{x\in\mathcal{X}} d^{-(-\log_d(p(x)))} = \sum_{x\in\mathcal{X}} p(x) = 1$. Hence, by Theorem 3.5, there exists a (not necessarily optimal) prefix code c with word lengths $(l_x)_{x\in\mathcal{X}}$. Now by definition

$$-\log_d(p(x)) \leq l_x < -\log_d(p(x)) + 1,$$

so conclude by multiplying this inequality with $p(x)$ and taking summing over $x \in \mathcal{X}$ to get (3.2.2). There are countably many prefix codes with expected length less than a given finite number, so we can sort them by expected length and take a code that achieves the minimum. The optimal code can only have an expected length less or equal to that of c. □

3.3 Approaching the lower bound by block codes

If c is an optimal code we are only guaranteed that

$$H_d(X) \leq \mathbb{E}[|c(X)|] < H_d(X) + 1.$$

The overhead of 1 is negligible if X has high entropy but it can be the dominating term for low entropies. By encoding sequences, we get arbitrary close to the lower bound: if we apply Proposition 3.10 to the $\mathcal{X}^n$-valued random variable $(X_1, \cdots, X_n)$ with X_i being i.i.d. copies of X, then the code $c_n : \mathcal{X}^n \longrightarrow \mathcal{Y}^*$ satisfies $\mathbb{E}[|c_n(X_1, \cdots, X_n)|] < H_d(X_1, \cdots, X_n) + 1$. But $H_d(X_1, \cdots, X_n) = nH(X)$, hence

$$\frac{1}{n}\mathbb{E}[|c_n(X_1, \cdots, X_n)|] < H_d(X) + \frac{1}{n} \to H_d(X) \text{ as } n \to +\infty.$$

Put differently, one needs at least $H_d(X)$ symbols to encode one symbol in the source and this bound is attainable (at asymptotically using block codes). This gives us the Shannon's code.

3.4 Shannon's code

In view of Theorem 3.9, a natural approach to construct a code is to assign with $x \in \mathcal{X}$ a codeword of length $\lceil -\log(p_X(x))\rceil$. Shannon gave an explicit algorithm that does this in his seminal 1948 paper: given a pmf p on $\mathcal{X} = \{1, \cdots, m\}, p_i = p(x_i)$, and a finite set $\mathcal{Y}$

(1) Order the probabilities p_i decreasingly and assume (by relabelling) that $p_1 \geq \cdots \geq p_m$,

(2) Define $c_S(x_r)$ as the first $l_r := \lceil -\log_{|\mathcal{Y}|}(p_r)\rceil$ digits in the $|\mathcal{Y}|$-ary expansion of the real number $\sum_{i=1}^{r-1} p_i$.

The above construction is the so-called Shannon code c_S. Following the proof of Theorem 3.5, one verifies that this is indeed a prefix code. As in Proposition 3.10, we also see that $H_{|\mathcal{Y}|}(X) \leq \mathbb{E}[|c_S(X)|] < H_{|\mathcal{Y}|}(X) + 1$. However,

- the Shannon code is in general not optimal,
- ordering a set of cardinality k needs $O(k \log(k))$ computational steps. This gets prohibitively expensive when combined with above block coding trick where we need to order $|\mathcal{X}|^n$ probabilities if we use blocks of length n; for example, already for uppercase English letters $\mathcal{X} = \{A, B, \cdots, Z\}$, using blocks of length $n = 100$, $|\mathcal{X}|^{100} = 26^{100}$ would require to order and store(!) a set that contains more elements than there are particles in the universe.

The Shannon code depends highly on the distribution of X. In practice, we usually have to infer the underlying probability distribution and work in a two step approach: firstly, read the whole message to infer the distribution; secondly, use the estimated pmf p to construct a code. The first step leads to errors, hence we need to ask how robust Shannon codes are.

Proposition 3.11. *Let p and q be pmf's on $\mathcal{X}$ and $X \sim p$ and $\mathcal{Y}$ a finite set of cardinality $|\mathcal{Y}| = d$. If we denote with $c_q : \mathcal{X} \longrightarrow \mathcal{Y}^*$ a Shannon code for the distribution q, then*

$$H_d(X) + D_d(p\|q) \leq \mathbb{E}[|c_q(X)|] < H_d(X) + D_d(p\|q) + 1.$$

Proof. We have

$$\begin{aligned}
\mathbb{E}[|c_q(X)|] &= \sum_{x \in \mathcal{X}} p(x) \lceil -\log_d(q(x)) \rceil \\
&< \sum_{x \in \mathcal{X}} p(x)(-\log_d(q(x)) + 1) \\
&= \sum_{x \in \mathcal{X}} p(x) \left(-\log_d \left(\frac{p(x)}{q(x)} \frac{1}{p(x)} \right) + 1 \right) \\
&= \sum_{x \in \mathcal{X}} p(x) \left(-\log_d \left(\frac{p(x)}{q(x)} \right) \right) + \sum_{x \in \mathcal{X}} p(x) \log_d \left(\frac{1}{p(x)} \right) + 1 \\
&= D_d(p\|q) + H_d(X) + 1.
\end{aligned}$$

Since the lower bound is attained iff $\lceil -\log_d(q(x)) \rceil = -\log_d(q(x))$ the lower bound follows similarly. $\square$

3.5 Fano's code [not examinable]

Fanon suggested a different construction that is also very simple to implement. Given a pmf p on $\mathcal{X} = \{1, \cdots, m\}$ with $X \sim p$ and $p_i = p(x_i)$, and a finite set $\mathcal{Y}$ with $d = |\mathcal{Y}|$, Fano gave an explicit construction for a d-ary prefix code. In the case of a binary encoding the construction is as follows:

(1) Order the symbols by their probability decreasingly, and assume (by relabelling) that $p_1 \geq \cdots \geq p_m$;

(2) Find r that minimises $|\sum_{i \leq r} p_i - \sum_{i > r} p_i|$ and split $\mathcal{X}$ into two groups $\mathcal{X}_0 := \{x_i : i \leq r\}$ and $\mathcal{X}_1 := \{x_i : i > r\}$;

(3) Define the first digit of the codewords for $\mathcal{X}_0$ as 0 and for $\mathcal{X}_1$ as 1,

(4) Repeat Steps (2) and (3) recursively until we can not split anymore.

Above construction leads to the so-called Fano-code (also called Shannon–Fano code) $c_F : \mathcal{X} \longrightarrow \mathcal{Y}^*$. As for the Shannon code, it is not hard to show that $\mathbb{E}[|c_F(X)|] \leq H_d(X) + 1$, that the Fano code is a prefix code and that in general the Fano code is not optimal.

3.6 Elias' code

Given a pmf p on $\mathcal{X} = \{1, \cdots, m\}$ with $p_i = p(x_i)$ and $X \sim p$, and a set $\mathcal{Y}$ of cardinality d. Define the Elias code (alos Shannon–Fano–Elias code) $c_E(x_i)$ as the first $\lceil -\log_d(p_i) \rceil + 1$ digits in the d-ary expansion of the real number $\sum_{i<r} p_i + \frac{p_r}{2}$. As above, one can show that $H_d(X) + 1 \leq \mathbb{E}[|c_E(X)|] \leq H_d(X) + 2$. Although it is less efficient than above codes, this construction has the big advantage that we do not need to order the elements of $\mathcal{X}$ by their probabilities. Further, it is a precursor of so-called arithmetic coding.

3.7 Huffman codes: optimal and a simple construction

Huffman was a student of Fano and realised that prefix codes corresponds to certain graphs, called rooted trees and that previous constructions such as Fano's code builds a tree starting at its root. As Huffman showed in 1952, by starting instead at the leaves of the tree, one gets a very simple algorithm that turns out to produce an optimal code!

Definition 3.12. *A undirected graph (V, E) is a tuple consisting of a set V and a set of two-element subsets of E. We call elements of V vertices and elements of E edges. For $v \in V$ we denote with $\deg(v)$ the number of edges that contain v and call $\deg(v)$ the degree of v. We call a graph d-ary if the maximal degree of its vertices is d.*

We now define a subset of the set of graphs.

Definition 3.13. *The set of rooted trees $\mathcal{T}$ is a subset of all graphs and defined recursively as:*

(1) The graph τ consisting of a single vertex r is a rooted tree. We call r the root and the leaf of τ.

(2) If $\tau_i \in \mathcal{T}$ for $i = 1, \cdots, n$, then the graph τ formed by starting with a new vertex r and adding edges to each of the roots of $\tau_1, \cdots, \tau_n$ is also a rooted tree. We call r the root of τ and we call the leaves of $\tau_1, \cdots, \tau_n$ the leaves of τ.

We can think of the set of prefix codes as the set of rooted trees: identify the codewords with leaves, the empty message with the root node and labelling the edges by letters that are in the codeword at the leave the end up at.

Lemma 3.14. *There is a bijection from the set of d-ary prefix codes to the set of d-ary rooted trees.*

As remarked in Section 3.2, to find a prefix code with minimal expected length we have to deal with a integer programming problem. Surprisingly, there exists a simple algorithm that construct the prefix code of shortest expected length for a given distribution in linear complexity. This the so-called Huffman code: we construct a rooted tree starting from the nodes of the least likely letters. For brevity of presentation, we describe only the binary Huffman code in detail: fix a pmf p on $\mathcal{X} = \{1, \cdots, m\}$ and a random variable $X \sim p$, and assume (by relabelling) that $p_1 \geq \cdots \geq p_m$ with $p_i := p(x_i)$. Then

(1) Associate with the two least likely symbols, two leaves that are joined into a vertex,

(2) Build a new distribution on $m-1$ symbols p, where $p'_1 = p_1, \cdots, p'_{m-2} = p_{m-2}$ and $p'_{m-1} := p_{m-1} + p_m$ (i.e. symbols $m-1$ and m are merged into one new symbol with probability $p'_{m-1} = p_{m-1} + p_m$), and relabel the resulted pmf by non-increasing order.

(3) Repeat above two steps of merging the two least likely symbols until we have a rooted tree.

Note that

- The algorithm can be seen as construction the codetree bottom up: Step 2 amounts to joining two leaves with a new node.
- Above algorithm terminates in $|\mathcal{X}| - 1$ steps and once we have build the rooted tree the code assignment is done by assigning 0 or 1 to the branches. Hence the complexity is $O(|\mathcal{X}|)$ if we are given a sorted pmf p; if we need to sort the pmf then the complexity of construction the Huffman code is $O(|\mathcal{X}| \log |\mathcal{X}|)$.
- If two symbols have same probability at every iteration, the resulting Huffman code may not be unique. However, they have the same expected length/
- In the d-ary case, the construction is analogous: we merge d nodes at every step. It may happen that we need to introduce dummy variables since there might not be enough nodes to merge d nodes. See [1] for details.

Proposition 3.15. *Let $\mathcal{X}, \mathcal{Y}$ be finite sets and p a pmf on $\mathcal{X}$ with a random variable $X \sim p$. The Huffman code $c : \mathcal{X} \longrightarrow \mathcal{Y}^*$ for p is optimal, i.e. if c' is another uniquely decodable code $c' : \mathcal{X} \longrightarrow \mathcal{Y}^*$ then*

$$\mathbb{E}[|c(X)|] \leq \mathbb{E}[|c'(X)|].$$

We prepare the proof with a Lemma about general properties of a certain optimal prefix code. In itself it is not an important code but it is a useful tool to prove optimality of other codes (such as Huffman as we will see in the proof Proposition 3.15).

Lemma 3.16. *Let p be a pmf on $\mathcal{X} = \{x_1, \cdots, x_m\}$ and assume wlog that $p_1 \geq \cdots \geq p_m$ for $p_i := p(x_i)$. There exists an optimal prefix code that has the property that*

(1) $p_j > p_k$ implies $|c(x_j)| \leq |c(x_k)|$,

(2) the two longest codewords have the same length,

(3) the two longest codewords only differ in the last digit.

We call c with these properties a canonical code for the pmf p.

Proof. The existence of an optimal prefix code holds since the set of prefix codes is well-ordered by expected length, hence there exists a (not necessarily unique) optimal code. For Point (1), fix an optimal code c and consider the code c' given by interchanging the codewords of c for x_j and x_k for some j, k with $j < k$ resp. $p_k < p_j$. Then

$$\begin{aligned} 0 &\leq \sum_i p_i |c'(x_i)| - \sum_i p_i |c(x_i)| \\ &= p_j |c(x_k)| + p_k |c(x_j)| - p_j |c(x_j)| - p_k |c(x_k)| \\ &= (p_j - p_k)(|c(x_k)| - |c(x_j)|). \end{aligned}$$

Hence $|c(x_k)| \geq |c(x_j)|$.

For Point (2), assume the contrary and remove the last digit from the longest codeword. This would still give a prefix code and this new prefix code would have strictly smaller expected length. Hence, the two longest codewords must have the same expected length.

For Point (3), identify a prefix code with a rooted tree. A codeword of maximum length must have a sibling (a leaf connecting to same vertex; otherwise, we could remove the last digit and get a prefix code of shorter expected length). Now exchange codewords until the two elements of $\mathcal{X}$ with lowest probabilities are associated with two siblings on the tree. □

We now use this to prove that the Huffman code is optimal.

Proof of Proposition 3.15. Fix a pmf p with $p_1 \geq \cdots \geq p_m$ on m symbols. Denote with p' the pmf on $m-1$ symbols given by merging the lowest probabilities, $p'_i = p_i$ for $i \in \{1, \cdots, m-2\}$ and $p'_{m-1} = p_{m-1} + p_m$. Let c^p be the canonical optimal code for p. Define $c^{p'}$ as the code for p' given by merging the leaves for p_{m-1} and p_m in the rooted tree representing c_p (by Lemma 3.16, p_{m-1}, p_m are siblings so this is possible). Then the difference in expected lengths is

$$L(c^p) - L(c^{p'}) = p_{m-1} l + p_m l - p'_{m-1}(l-1) \quad (3.7.1)$$

$$= p_{m-1} + p_m. \quad (3.7.2)$$

where l denotes the codeword lengths of symbols $m-1$ and m under c^p. On the other hand, let $e^{p'}$ be any optimal (prefix) code for p'. We again represent it as a rooted tree and define e^p by replacing the leaf for p'_{m-1} with a rooted tree consisting of two leaves p_m and p_{m-1}. Then

$$L(e^p) - L(e^{p'}) = p_{m-1} + p_m. \quad (3.7.3)$$

Substracting (3.7.1) from (3.7.3) yields

$$(L(e^p) - L(c^p)) + (L(c^{p'}) - L(e^{p'})) = 0.$$

By assumption, c^p and $e^{p'}$ are optimal, hence both terms are non-negative so both must equal 0. We conclude that $L(e^p) = L(c^p)$, hence e^p is an optimal code for p. The above shows, that expanding any optimal code e' for p' leads to an optimal code e^p for p. Now note that the Huffman code is constructed by a repeated application of such an expansion. Further, for $m = 2$ the Huffman code is clearly optimal, hence the result follows by induction on m. □

The Huffman code has a simple construction and is optimal. It is used in mainstream compression formats (such as gzip, jpeg, mp3, png, etc). However, it is not the final answer to source coding.

- Not every optimal code is Huffman; e.g. is optimal but not Huffman (since c can be obtained by

p(x)	0.3	0.3	0.2	0.2
c(x)	00	10	01	11

  permutating leaves of same length of the Huffman code for p).

- Huffman (and all the other prefix codes we have discussed so far, except Elias' code) requires (ordering) knowledge of p. Further, optimality was defined for messages that are drawn by i.i.d. samples. When compressing text (source symbols are english letters) this does not apply since e.g. the probability of sampling e is much higher if the previous two letters were “th” compared with say “xy”.

- Optimality just guarantees $H_d(X) \leq \mathbb{E}[|c(X)|] < H_d(X) + 1$. This is a good bound if $H_d(X)$ is large but for small entropies the term $+1$ on the right hand side is dominant. One can again use the block coding trick discussed in Section 3.3 to encode sequences of length n to reduce the overhead to $1/n$ bits but this again leads to a combinatorial explosion since we need to sort $|\mathcal{X}|^n$ probabilty masses.

Chapter 4

Channel Coding: Shannon's Second Theorem

In Chapter 2 we studied how much information is contained in sequences and used this to derived codes to store such sequences. In many real-world situations we are confronted with the problem of transmitting information from one place to another, typically through a medium that is subject to noise and perturbations.

4.1 Discrete memoryless channels

Definition 4.1. *A discrete memoryless channel (DMC) is a triple* $(\mathcal{X}, M, \mathcal{Y})$ *consisting of*

- *a finite set* $\mathcal{X}$*, called the input alphabet,*
- *a finite set* $\mathcal{Y}$*, called the output alphabet,*
- *a stochastic*[1] $|\mathcal{X}| \times |\mathcal{Y}|$*-matrix* M*.*

We say that a pair of random variables X, Y *defined on some probability space* $(\Omega, \mathcal{F}, \mathbb{P})$ *realises the DMC, if the conditional distribution of* Y *given* X *equals* M*, i.e.* $M = \big(p_{Y|X}(y|x)\big)_{x\in\mathcal{X}, y\in\mathcal{Y}}$.

Example 4.2. $\mathcal{X} = \{0, 1\}$, $\mathcal{Y} = \{a, b, c, d, e\}$ and $M = \begin{pmatrix} 0.2 & 0 & 0.5 & 0 & 0.3 \\ 0 & 0.2 & 0 & 0.8 & 0 \end{pmatrix}$.

Above is an example of a lossless channel: knowing the output $\mathcal{Y}$ allows to uniquely identify the input X (e.g. output is b or d iff the input is 1). More generally, we call $(\mathcal{X}, M, \mathcal{Y})$ a lossless channel if we can divide $\mathcal{Y}$ into disjoint sets $\mathcal{Y}_1, \cdots, \mathcal{Y}_{|X|}$ such that

$$\mathbb{P}(Y \in \mathcal{Y}_i | X = x_i) = 1 \text{ for } \forall\, 1 \le i \le |\mathcal{X}|.$$

For a lossless channel $(\mathcal{X}, M, \mathcal{Y})$, similar to Point (2) in Theorem 1.15, we have $H(X|Y) = 0$ (since $X = f(Y)$ for $f(y) = x_i \mathbf{1}_{y\in\mathcal{Y}_i}$, i.e., X is a deterministic function of Y).

[1] A stochastic matrix is a matrix with non-negative entries and the sum of entries in each row equals 1.

Another extreme is a channel that is completely useless for transmitting information, i.e. the output Y contains no information about the input X. This means X and Y are independent, which is (again by Point (2) in Theorem 1.15) equivalent to $H(X|Y) = H(X)$.

Here are some important examples of DMCs.

Example 4.3. Let $q \in [0, 1]$.

(1) **Binary symmetric channel**: $\mathcal{X} = \mathcal{Y} = \{0, 1\}$ and the stochastic matrix is given as

$\mathcal{X} \setminus \mathcal{Y}$	0	1
0	$1-q$	q
1	q	$1-q$

(2) **Binary erasure channel**: $\mathcal{X} = \{0, 1\}$, $\mathcal{Y} = \{0, 1, ?\}$ and the stochastic matrix is given as

$\mathcal{X} \setminus \mathcal{Y}$	0	?	1
0	$1-q$	q	0
1	0	q	$1-q$

(3) **Noisy typewriter**: $\mathcal{X} = \mathcal{Y} = \{A, \cdots, Z\}$ and the stochastic matrix is given as

$\mathcal{X} \setminus \mathcal{Y}$	A	B	C	D	$\cdots$	$\cdots$	Y	Z
A	1/3	1/3	0	0	$\cdots$	$\cdots$	0	1/3
B	1/3	1/3	1/3	0	$\cdots$	$\cdots$	0	0
$\vdots$		$\ddots$		$\ddots$		$\ddots$		$\ddots$
Y	0	0	0	$\cdots$	0	1/3	1/3	1/3
Z	1/3	0	0	$\cdots$	0	0	1/3	1/3

4.2 Channel capacity

We want to measure how much our uncertainty about the input X is reduced by knowing the output Y. We have seen that a lossless channel $H(X|Y) = 0$ and a useless channel $H(X|Y) = H(X)$. Motivated by this, an intuitive measure for the quality of a channel is

$$H(X) - H(X|Y) = I(X; Y).$$

A DMC only specifies the distribution of the output conditional on the input. To use the channel for information transmission, we have freedom to choose the distribution of the input. This motivates the definition of channel capacity.

Definition 4.4. *Let $(\mathcal{X}, M, \mathcal{Y})$ be a DMC. We call $C := \sup I(X;Y)$ the channel capacity of DMC $(\mathcal{X}, M, \mathcal{Y})$, where the supremum is taken over all pairs of random variables X, Y that realise the DMC $(\mathcal{X}, M, \mathcal{Y})$.*

From $I(X;Y) = H(X) - H(X|Y) = H(Y) - H(Y|X)$, it follows that

$$0 \le C \le \min\{\log(|\mathcal{X}|), \log(|\mathcal{Y}|)\}.$$

To make the definition well-defined, we have the following proposition of $I(X;Y)$.

Proposition 4.5. *Fix $p_{Y|X}$, $I(X;Y)$ is concave in p_X; Fix p_X, $I(X;Y)$ is convex in $p_{Y|X}$.*

Proof. For the first statement, recall $I(X;Y) = H(Y) - H(Y|X)$. Given $p_{Y|X}$, p_Y is linear in p_X, and $H(Y)$ is concave in p_Y, so $H(Y)$ is concave in p_X.

For the second statement, recall $I(X;Y) = D(p_{X,Y} \| p_X * p_Y)$. Given p_X, $p_{X,Y}$ is linear in $p_{Y|X}$, so is $p_X * p_Y$. By Point (4) in Theorem 1.12, we know $D(p\|q$ is convex in (p,q). So $I(X;Y)$ is convex in $p_{Y|X}$ for any fixed p_X. □

Below we calculate the capacity of some simple channels.

Example 4.6. For the binary symmetric channel specified in Example 4.3.(1), we have a transmission error with probability q. To calculate its capacity, we need to estimate $I(X;Y)$.

$$\begin{aligned} I(X;Y) &= H(Y) - H(Y|X) \\ &= H(Y) - \sum_{x\in\mathcal{X}} p(x) H(Y|X=x) \\ &= H(Y) - \sum_{x\in\mathcal{X}} p(x) H(q) \\ &\le 1 - H(q), \end{aligned}$$

where $H(q) = -q\log(q) - (1-q)\log(1-q)$ is the entropy of the pmf $(q, 1-q)$ (Bernoulii distribution). Note that if Y is uniform, $\mathbb{P}(Y=0) = \mathbb{P}(Y=1) = 1/2$, then $H(Y) = 1$ and above is an equality. Since

$$\begin{aligned} p_Y(0) &= (1-q)p_X(0) + q p_X(1) \\ p_Y(1) &= q p_X(0) + (1-q) p_X(1), \end{aligned}$$

we know that $\mathbb{P}(Y=0) = 1/2$ is equivalent to $\mathbb{P}(X=0) = 1/2$ by the symmetry between the roles of X and Y. Hence the maximum is $C = 1 - H(q)$, which is attained iff $\mathbb{P}(X=0) = 1/2$.

Example 4.7. In Example 4.3.(2), a binary erasure channel is specified by $\mathcal{X} = \{0,1\}$ and $\mathcal{Y} = \{0, e, 1\}$, where e can be interpreted as an error occurred in the transmission, and the stochastic matrix M given as

$\mathcal{X} \setminus \mathcal{Y}$	0	?	1
0	$1-q$	q	0
1	0	q	$1-q$

The binary channel erases a fraction of q bits that are transmitted and the receiver knows if which bits have been erased. Hence, we can only hope to recover $1-q$ bits in proportion. Now as before $I(X;Y) = H(Y) - H(Y|X) = H(Y) - H(q)$ with $H(q)$ the same as in the previous example. Set $\pi = \mathbb{P}(X=1)$, then $p_Y(0) = (1-\pi)(1-q)$, $p_Y(e) = (1-\pi)q + \pi q = q$, $p_Y(1) = \pi(1-q)$, so

$$\begin{aligned} H(Y) &= -(1-\pi)(1-q)\log((1-\pi)(1-q)) - q\log(q) - \pi(1-q)\log(\pi(1-q)) \\ &= -(1-\pi)(1-q)\log(1-\pi) - (1-\pi)(1-q)\log(1-q) \\ &\quad -q\log(q) \\ &\quad -\pi(1-q)\log(\pi) - \pi(1-q)\log(1-q) \\ &= H(q) + (1-q)H(\pi). \end{aligned}$$

Now

$$I(X;Y) = H(q) + (1-q)H(\pi) - H(q) = (1-q)H(\pi)$$

and therefore the capacity is $C = 1-q$ achieved with $\pi = \mathbb{P}(X=1) = 1/2$.

4.3 Channel codes, rates and errors

We want to use the channel to reliably transmit a message from a given set of possible messages. We are allowed to use the channel several times. Hence, we are looking for a map that transforms the message into a sequence symbols in $\mathcal{X}$ (encoding), we then send this sequence through the channel and upon receiving the corresponding sequence symbols in $\mathcal{Y}$, transforms this back to a message (decoding) with a small probability of error.

Definition 4.8. *Fix $m, n \geq 1$. A (m,n)-channel code for a DMC $(\mathcal{X}, M, \mathcal{Y})$ is a tuple (c,d) consisting of*

- *a map $c : \{1, \cdots, m\} \longrightarrow \mathcal{X}^n$, called the encoder,*
- *a map $d : \mathcal{Y}^n \longrightarrow \{1, \cdots, m\}$, called the decoder.*

We call $\{1, \cdots, m\}$ the message set, $c(i)$ the codeword for message $i \in \{1, \cdots, m\}$ and the collection $\{c(i) : i = 1, \cdots, m\}$ the codebook.

That is to say, a (m,n) channel transmits one out of m messages by using the channel n times.

Definition 4.9. *Let $(\mathcal{X}, M, \mathcal{Y})$ be a DMC. We call $\rho(c,d) := \frac{1}{n}\log_{|\mathcal{X}|}(m)$ the rate of the (m,n)-code (c,d).*

Definition 4.10. *Let (c,d) be a (m,n)-channel code for a DMC $(\mathcal{X}, M, \mathcal{Y})$. Set*

$$\varepsilon_i = \mathbb{P}(d(\mathbf{Y}) \neq i \,|\, c(i) = \mathbf{X}) \text{ for } i = 1, \cdots, m,$$

where $\mathbf{X} = (X_1, \cdots, X_n)$ and $\mathbf{Y} = (Y_1, \cdots, Y_n)$ with $\{(X_i, Y_i)\}_{i=1,\cdots,n}$ consisting of i.i.d. copies of random variables (X,Y) that realise the DMC. We say that the channel code has

(1) a maximal probability of error $\varepsilon_{max} := \max_{i \in \{1,\cdots,m\}} \varepsilon_i$,

(2) an arithmetic error $\bar{\varepsilon} := \frac{1}{m}\sum_{i=1}^{m} \varepsilon_i$.

Remark 4.11. For applications we clearly care about ε_{max} and *a priori* it is not clear that $\bar{\varepsilon}$ is a useful quantity to consider. Note that $\bar{\varepsilon} \leq \varepsilon_{max}$ and that $\bar{\varepsilon}$ is the expectation of the error ε_i, if an element i is chosen uniformly at random. It turns out that good estimates on $\bar{\varepsilon}$ imply good estimates on ε_{max} and that bounds on $\bar{\varepsilon}$ are easy to establish (we are going to use this in the proof of the noisy channel coding theorem).

Already a simple repetition code (represent the message i in its $|\mathcal{X}|$-ary expansion and transmit each digit multiple times) can achieve an arbitrary small error for the cost of a vanishing rate. We therefore need to understand the tradeoff between the error probability ε_{max} (which we want to make small) and the rate R(which we want to keep large). That is, we ask what points in (ε_{max}, R)-plane can be reached by channel codes (with a sufficiently large n)? Before Shannon, a common belief was that that as ε_{max} goes to 0 so does the rate. A big surprise was Shannon's noisy channel coding theorem, that showed that any rate below channel capacity can be achieved!

4.4 Shannon's second theorem: noisy channel coding

Definition 4.12. *A rate* $R > 0$ *is achievable for a DMC* $(\mathcal{X}, M, \mathcal{Y})$*, if for any* $\varepsilon > 0$ *there exists sufficiently large* m, n *and a* (m, n)*-channel code* (c, d) *with*

$$\rho(c, d) > R - \varepsilon \text{ and } \varepsilon_{max} < \varepsilon,$$

where ε_{max} *denotes the maximal error of* (c, d).

In other words, a rate R is achievable if there exists a sequence of codes whose rates approach R and whose maximal errors approach zero. *A priori* it is by no means obvious that a message may be transmitted over a DMC at a given rate with as small probability of error as desired! Shannon's result not only shows that this is possible but also shows that the set of rates that can be achieved is exactly those that are bounded by the channel capacity C. We already saw that the channel capacity can be explicitly computed for some important channels. All these are reasons why Theorem 4.13 is considered a (maybe even the) major result of communication theory.

Theorem 4.13. *(Shannon's second theorem: noisy channel coding). Let* $(\mathcal{X}, M, \mathcal{Y})$ *be a DMC with capacity* C*. Then a rate* $R > 0$ *is achievable iff* $R \leq C$.

An analogy that is often used is to compare a channel to a water pipe. If we pump water through a pipe above its capacity, then the pipe will burst and water will be lost. Similarly, if information flows through a channel at rate higher than channel capacity, the error is strictly bounded away from zero which means we loose information.

To be concise, we take $d = |\mathcal{X}| = 2$ in the rest of this section, so $\log_{|\mathcal{X}|}$ will be replaced by log.

Let us first give an informal "proof" of Shannon's channel coding theorem. The idea is to use a "typical set decoder": define a decoder by partitioning $\mathcal{Y}^n$ into disjoint subsets $\mathcal{Y}_1, \cdots, \mathcal{Y}_m$ of $\mathcal{Y}^n$, and associate each set with an input sequence $x_1, \cdots, x_m \in \mathcal{X}^n$. That is upon a receiving a sequence $y \in \mathcal{Y}^n$, if we find an i such that $y \in \mathcal{Y}_i$, then we decode it as message i. How can find a partition that is efficient and robust to the noise in the channel? The key insight is similar to source coding: sequences can be divided into a set of typical sequences that carries most of the probability mass. There are approximately $2^{nH(Y)}$ typical output sequences. Similarly, to a given typical input sequence x correspond approximately $2^{nH(Y|X)}$ output sequences that are likely (i.e. y's such that (x,y) is typical wrt to $p_{X,Y}$). But for two different typical input sequences, these subsets of $\mathcal{Y}^n$ might overlap. To account for this we restrict ourselves further to a subset of typical input sequences such that the corresponding sets of typical output sequences do not overlap (but still cover nearly all of) $\mathcal{Y}^n$. There are at most

$$\frac{2^{nH(Y)}}{2^{nH(Y|X)}} = 2^{n(H(Y)-H(Y|X))} = 2^{nI(X;Y)}$$

such typical input sequences. Hence, there are at most $2^{nI(X;Y)}$ codewords which gives a rate of $\frac{\log(2^{nI(X;Y)})}{n} = I(X;Y) \leq C$ bits per channel use. This shows (very heuristically) why we can expect to achieve any rate $R \leq C$.

Definition 4.14. *Let (X,Y) be a $\mathcal{X} \times \mathcal{Y}$-valued random variable with pmf $p_{X,Y}$. For $n \in \mathbb{N}$ and $\varepsilon > 0$, set $\mathbf{X} = (X_1, \cdots, X_n), \mathbf{Y} = (Y_1, \cdots, Y_n)$ with entries i.i.d. copies of X, Y, and*

$$\mathcal{J}_\varepsilon^{(n)} = \left\{(x,y) \in \mathcal{X}^n \times \mathcal{Y}^n : \max\left(\left|\frac{-\log(p_{\mathbf{X},\mathbf{Y}}(x,y))}{n} - H(X,Y)\right|, \left|\frac{-\log(p_{\mathbf{X}}(x))}{n} - H(X)\right|, \left|\frac{-\log(p_{\mathbf{Y}}(y))}{n} - H(Y)\right|\right) < \varepsilon\right\}.$$

We call $\mathcal{J}_\varepsilon^{(n)}$ the set of jointly typical sequences of length n and tolerance ε.

Theorem 4.15. *(Joint AEP). Let $\mathbf{X} = (X_1, \cdots, X_n), \mathbf{Y} = (Y_1, \cdots, Y_n)$ with entries i.i.d. copies of X, Y. Then*

(1) $\lim_{n\to+\infty} \mathbb{P}((\mathbf{X}, \mathbf{Y}) \in \mathcal{J}_\varepsilon^{(n)}) = 1$;

(2) $|\mathcal{J}_\varepsilon^{(n)}| \leq 2^{n(H(X,Y)+\varepsilon)}$;

(3) If X' and Y' are independent, and X and Y have the same margins as X' and Y' respectively, i.e., $(X', Y') \sim p_X p_Y$. Construct $\mathbf{X}'$ and $\mathbf{Y}'$ similarly based on (X', Y'), then $\exists\, n_0$ such that $\forall\, n \geq n_0$, we have,

$$(1-\varepsilon)2^{-n(I(X;Y)+3\varepsilon)} \leq \mathbb{P}\left((\mathbf{X}', \mathbf{Y}') \in \mathcal{J}_\varepsilon^{(n)}\right) \leq 2^{-n(I(X;Y)-3\varepsilon)},$$

where the upper bound holds for all $n \geq 1$.

Proof. Point (1) follows by independence and weak law of large numbers: $\frac{\log(p(X_1,\cdots,X_n))}{n} = \frac{\sum_{i=1}^n \log(p(X_i))}{n} \to H(X)$, hence

$$\mathbb{P}\left(\left|\frac{\log(p_X(X_1,\cdots,X_n))}{n} - H(X)\right| \geq \varepsilon\right) < \frac{\varepsilon}{3} \text{ for all } n \geq n_1$$

for some integer n_1, and similarly

$$\mathbb{P}\left(\left|\frac{\log(p_Y(Y_1,\cdots,Y_n))}{n} - H(Y)\right| \geq \varepsilon\right) < \frac{\varepsilon}{3} \text{ for all } n \geq n_2,$$

$$\mathbb{P}\left(\left|\frac{\log(p_{X,Y}(X_1,\cdots,X_n,Y_1,\cdots,Y_n))}{n} - H(X,Y)\right| \geq \varepsilon\right) < \frac{\varepsilon}{3} \text{ for all } n \geq n_3.$$

for some integers n_2, n_3. Taking $n \geq \max(n_1, n_2, n_3)$ shows the result.

Point (2) follows since

$$1 = \sum_{\mathcal{X}^n \times \mathcal{Y}^n} p_{X,Y}(x,y) \geq \sum_{\mathcal{J}_\varepsilon^{(n)}} p_{X,Y}(x,y) \geq |\mathcal{J}_\varepsilon^{(n)}| 2^{-n(H(X,Y)+\varepsilon)},$$

and therefore $|\mathcal{J}_\varepsilon^{(n)}| \leq 2^{n(H(X,Y)+\varepsilon)}$.

Point (3): for the upper bound,

$$\begin{aligned} \mathbb{P}\left((X',Y') \in \mathcal{J}_\varepsilon^{(n)}\right) &= \sum_{\mathcal{J}_\varepsilon^{(n)}} p_X(x) p_Y(y) \\ &\leq 2^{n(H(X,Y)+\varepsilon)} 2^{-n(H(X)-\varepsilon)} 2^{-n(H(Y)-\varepsilon)} \\ &= 2^{-n(I(X;Y)-3\varepsilon)}. \end{aligned}$$

For the lower bound, we have for large enough n that $\mathbb{P}\left((X,Y) \in \mathcal{J}_\varepsilon^{(n)}\right) \geq 1-\varepsilon$, hence

$$1-\varepsilon \leq \sum_{\mathcal{J}_\varepsilon^{(n)}} p_{X,Y}(x,y) \leq \left|\mathcal{J}_\varepsilon^{(n)}\right| 2^{-n(H(X,Y)-\varepsilon)},$$

and we get $\left|\mathcal{J}_\varepsilon^{(n)}\right| \geq (1-\varepsilon) 2^{n(H(X,Y)-\varepsilon)}$. Using this, we get similar to above,

$$\begin{aligned} \mathbb{P}\left((X',Y') \in \mathcal{J}_\varepsilon^{(n)}\right) &= \sum_{\mathcal{J}_\varepsilon^{(n)}} p_X(x) p_Y(y) \\ &\geq (1-\varepsilon) 2^{n(H(X,Y)-\varepsilon)} 2^{-n(H(X)+\varepsilon)} 2^{-n(H(Y)+\varepsilon)} \\ &= 2^{-n(I(X,Y)+3\varepsilon)}. \end{aligned}$$

□

We now use the above to give a rigorous proof of Shannon's channel coding theorem.

Proof of Theorem 4.13. Fix a pmf p on $\mathcal{X}$ and let $\mathcal{J}_\varepsilon^{(n)}$ be the jointly typical set of $p_{X,Y} = p_{Y|X} p_X$. We generate a random (m,n)-channel code as follows:

(1) Generate m random codewords in $\mathcal{X}^n$, by sampling independently from $\Pi_{i=1}^n p_X(x_i)$;

(2) For each message $i \in \{1, \cdots, m\}$, define its encoding by sampling uniformly from this set of random codewords;

(3) Define the decoder as a typical-set decoder: upon receiving $\mathbf{Y}$, check if there exists a unique element $\mathbf{X}$ in the set of random codewords such that $(\mathbf{X},\mathbf{Y}) \in \mathcal{J}_{\varepsilon/6}^{(n)}$. In this case, decode as the message that was in step (2) associated with the codeword $\mathbf{X}$. If this is not the case (there does not exists such a codeword or it is not unique) the decoder outputs m.

Denote this random (m,n)-channel code with $(\mathcal{C},\mathcal{D})$. Now,

(1) Sample from the channel code $(\mathcal{C},\mathcal{D})$;

(2) Sample a message W uniformly from $\{1, \cdots, m\}$;

(3) Send the sequence $\mathbf{X} = \mathcal{C}(W)$ through the channel;

(4) Decode the channel output using $\mathcal{D}$, denote the decoded message with $\hat{W}$.

For any $\varepsilon > 0$, applied with $m = 2^{n(R-2\varepsilon/3)+1}$, the random (m, n)-channel code $(\mathcal{C}, \mathcal{D})$ has rate $R - \frac{2}{3}\varepsilon + \frac{1}{n}$. By Lemma 4.18 (coming soon), for any $\varepsilon > 0$ we can choose n large enough such that

$$\mathbb{P}(W \neq \hat{W}) < \frac{\varepsilon}{2}.$$

By conditioning

$$\mathbb{P}(W \neq \hat{W}) = \sum_{(c,d)} \mathbb{P}(W \neq \hat{W} | (\mathcal{C}, \mathcal{D}) = (c, d)) \mathbb{P}((\mathcal{C}, \mathcal{D}) = (c, d)) < \frac{\varepsilon}{2},$$

it follows that there must exist at least one channel code (c^*, d^*) such that

$$\mathbb{P}(W \neq \hat{W} | (\mathcal{C}, \mathcal{D}) = (c^*, d^*)) < \frac{\varepsilon}{2}.$$

Recall that W was sampled uniformly and the arithmetic error is the expected error over all messages if the input is uniformly distributed. Hence, above inequality can be restated as $\bar{\varepsilon} < \frac{\varepsilon}{2}$, where $\bar{\varepsilon}$ denotes the arithmetic error of (c^*, d^*). Thus we have shown the existence of a (m, n)-channel code, rate $R + \frac{1}{n}$ and arithmetic error $\bar{\varepsilon} < \frac{\varepsilon}{2}$. Further,

$$\bar{\varepsilon} = \frac{1}{m} \sum_{i=1}^{m} \varepsilon_i < \frac{\varepsilon}{2},$$

or equivalently, $\sum_{i=1}^{m} \varepsilon_i < \frac{m}{2}\varepsilon$ (here ε_i denotes the probability of an error in decoding message i using channel code (c^*, d^*)). Now sort the codewords by their error probabilities ε_i. Each of the probabilities in the better half of the m codewords must be less than ε since otherwise the sum over the other half would be at least $\frac{m}{2}\varepsilon$ which contradicts $\sum_{i=1}^{m} \varepsilon_i < \frac{m}{2}\varepsilon$. Therefore throwing away the worse half the codewords modifies (c^*, d^*) into into a $(\frac{m}{2}, n)$-channel code with rate $\rho(c^*, d^*) = R - \frac{2}{3}\varepsilon > R - \varepsilon$ and $\varepsilon_{max} < \varepsilon$ as required.

For the optimality, we fix $\varepsilon > 0$ and assume for sufficiently large n there exists a (m, n) channel code with

$$\frac{\log(m)}{n} > R - \varepsilon \text{ and } \varepsilon_{max} < \varepsilon. \tag{4.4.1}$$

Let W be a random variable that is uniformly distributed on the messages $\{1, \cdots, m\}$ and as above, denote with the $\hat{W}$ the decoded message. Then

$$\begin{aligned}
\log(m) &= H(W) \\
&= H(W|\hat{W}) + I(W; \hat{W}) \\
&\leq H(W|\hat{W}) + I(X; Y) \\
&\leq H(W|\hat{W}) + \sum_{i=1}^{n} I(X_i; Y_i) \\
&\leq H(W|\hat{W}) + nC \\
&< 1 + \bar{\varepsilon} \log(m) + nC, \qquad (4.4.2)
\end{aligned}$$

where the first inequality uses that $I(W; \hat{W}) \leq I(X; Y)$ by the data processing inequality, the second inequality follows since $I(X; Y) \leq \sum_{i=1}^{n} I(X_i; Y_i)$, and the third inequality is just the definition of channel capacity. The last inequality is Fano's inequality. Using $\varepsilon \leq \varepsilon_{max} < \varepsilon$ and rearranging above inequality gives

$$\frac{\log(m)}{n} < \frac{C + 1/n}{1 - \varepsilon}.$$

Using the assumption (4.4.1), this implies $R - \varepsilon < \frac{C+1/n}{1-\varepsilon}$. By sending $n \to +\infty$ and $\varepsilon \to 0$ we conclude $R \leq C$. □

Remark 4.16. Above proof even gives an asymptotic bound on the arithmetic error $\bar{\varepsilon}$ for a code (c, d) with rate $\rho(c, d) > C$. Rearranging the estimate (4.2) implies

$$\bar{\varepsilon} \geq 1 - \frac{1 + nC}{\log(m)} = 1 - \frac{C + 1/n}{\frac{1}{n}\log(m)}. \tag{4.4.3}$$

For large n, the right hand side is well approximated by $1 - \frac{C}{\frac{1}{n}\log(m)} = 1 - \frac{C}{\rho(c,d)}$.

Remark 4.17. The bound (4.4.3) implies a strictly positive arithmetic error for any $n \geq 1$ if the rate is bigger than C. To see this, assume by contradiction that the arithmetic error equals 0 for some n_0. Then we could transform this into a new (m^k, kn_0)-channel code by concatenating k codewords. But this channel has the same rate. Hence choosing k large enough contradicts the estimate (4.4.3). This is often called the weak converse of the channel coding theorem. There also exists a strong converse (which do not prove) which shows that $\bar{\varepsilon} \to 1$ as $n \to +\infty$ if $\frac{\log(m)}{n} \geq C + \varepsilon$ for some $\varepsilon > 0$.

Lemma 4.18. *Let $W, \hat{W}$ be defined as in the proof of Theorem 4.13. Then $\mathbb{P}(W \neq \hat{W}) \to 0$ as $n \to +\infty$.*

Proof. [not examinable] . Denote with E_i the event that the random codeword for i and the channel output are jointly typical (in $J^{(n)}_{\varepsilon/6}$). By construction of the random code, ε_i is the same for all messages $i = 1, \cdots, m$, hence $\varepsilon_{max} = \varepsilon_1$ (both errors are expectations over the draw of the codewords). By the union bound for probabilities

$$\varepsilon_{max} = \varepsilon_1 = \mathbb{P}(\hat{W} \neq 1 | W = 1) = \mathbb{P}\left(E_1^c \cup (\cup_{i=2}^m E_i) \,|\, W = 1\right) \leq \mathbb{P}(E_1^c \,|\, W = 1) + \sum_{i=2}^m \mathbb{P}(E_i | W = 1).$$

By joint typicality, $\mathbb{P}(E_1^c | W = 1) < \frac{\varepsilon}{6}$ and $\mathbb{P}(E_i | W = 1) \leq 2^{-n(I(X;Y) - 3\varepsilon/6)} = 2^{-n(I(X;Y) - \varepsilon/2)}$ for n large enough. Hence,

$$\begin{aligned}
\varepsilon_{max} &\leq \frac{\varepsilon}{6} + \sum_{i=2}^m 2^{-n(I(X;Y) - \varepsilon/2)} \\
&= \frac{\varepsilon}{6} + m 2^{-n(I(X;Y) - \varepsilon/2)} \\
&= \frac{\varepsilon}{6} + 2^{-n\varepsilon/6} 2^{-n(I(X;Y) - R) + 1} \\
&\leq \varepsilon/2
\end{aligned}$$

for large enough n (such that $\frac{\varepsilon}{3} \geq 2^{1 - n\varepsilon/6}$) and $R \leq I(X;Y)$. □

4.5 Channel codes

How to find a good channel code?

- If n is fixed we could try to search all possible codebooks. There are $|\mathcal{X}|^n$ codewords and approximately $|\mathcal{X}|^{mn}$ injective codes. If the rate of the code is assumed to be close to C then m is approximately $|\mathcal{X}|^{nC}$, hence we need to search over approximately $|\mathcal{X}|^{n|\mathcal{X}|^{nC}}$, which is computationally infeasible.
- We could try to use a randomly generated channel code as in above proof. Above argument shows that is likely to be a good channel code for large n. Unfortunately, such a code is difficult to use in practice:
  - there are 2^{nR+1} codewords, i.e. to encode a message we need to store a table that grows exponentially with n;
  - the decoder needs to decide which of the 2^{nR+1} messages was transmitted, which again takes an exponential amount of time.

In fact, it took a long time after Shannon's proof of the existence of codes achieving rate C to find useful constructions. Breakthroughs are '72 Justesen, '93 Berrou et al, and '97 MacKay and Neal. The unifying idea of all these codes it introduce some redundancy such that a perturbed message can still be recovered. There are two big classes of codes used nowadays:

(1) block codes: to encode a block of information into a codeword but there is no dependence on past information. Examples include Hamming codes, Reed–Muller/Solomon codes, BCH codes, etc;

(2) convolutional codes: they are more complicated since they use dependicy on the past inputs.

The search for optimal and practical codes is still an active area of research. In general this is a complicated topic that requires lots of algebra. We only study Hamming codes on Sheet 4.

4.6 Channel coding with non-iid input

It is natural to ask whether one can combine Shannon's two theorems: given a signal such as digitised speech, the obvious approach is to first apply symbol coding (Theorem 2.7) for compression, and then apply channel coding (Theorem 4.13) to send this compressed signal through our channel. Two questions arise: firstly, is this two-stage approach optimal? An alternative is to directly feed the digitised signal into channel coding without an extra compression layer before. Secondly, the channel input will not be an i.i.d. sequence. This is a case that needs discussion even without the first stage.

We first address the second question by showing that the notion of entropy extends to sequences of (possibly dependent) random variables. We use this in the next section to answer the first question of optimality.

Definition 4.19. *A discrete stochastic process is a sequence $X = (X_i)_{i\geq 1}$ of discrete random variables. We say that a stochastic process is stationary if*

$$\mathbb{P}(X_1 = x_1, \cdots, X_n = x_n) = \mathbb{P}(X_{1+j} = x_1, \cdots, X_{n+j} = x_n)$$

for all integers n, j and $x_1, \cdots, x_n \in X$.

A special case is a stochastic process with X_i i.i.d., but much more complicated statistical dependencies can occur between the X_i.

Definition 4.20. *The entropy rate of a stochastic process $X = (X_i)_i$ is defined as*

$$\mathcal{H}(X) = \lim_{n\to+\infty} \frac{1}{n} H(X_1, \cdots, X_n),$$

whenever this limit exists.

Obviously, if X_i are i.i.d., then the entropy rate exists and $\mathcal{H}(X) = \lim_{n\to+\infty} \frac{1}{n}(H(X_1)+\cdots+H(X_n)) = H(X_1)$. However, for the case when X_i are independent but not identically distributed the above limit does not necessarily exists. For example, the binary variables X_i with

$$\mathbb{P}(X_i = 1) = 0.5 \text{ for } \log(\log(i)) \in (2k, 2k+1] \quad \text{and } \mathbb{P}(X_i = 1) = 0 \text{ for } \log(\log(i)) \in (2k+1, 2k+2]$$

where k can be any number in $\{0, 1, 2, \cdots\}$. This construction gives long stretches with $H(X_i) = 1$ followed by exponentially longer stretches of $H(X_i) = 0$, hence the running average will oscillate between 0 and 1.

Theorem 4.21. *For a stationary stochastic processes X, the entropy rate exists and*

$$\mathcal{H}(X) = \lim_{n\to+\infty} H(X_n|X_{n-1}, \cdots, X_1).$$

We prepare the proof with two Lemmas.

Lemma 4.22. *For a stationary stochastic process X, $n \mapsto H(X_n|X_{n-1}, \cdots, X_1)$ is non-increasing and $\lim_{n\to+\infty} H(X_n|X_{n-1}, \cdots, X_1)$ exists.*

Proof. For any integer n,

$$H(X_{n+1}|X_n, \cdots, X_1) \le H(X_{n+1}|X_n, \cdots, X_2) = H(X_n|X_{n-1}, \cdots, X_1),$$

where we used that conditioning reduces entropy for the inequality, and the equality is due to the stationarity of the process X. Since $H(X_n|X_{n-1}, \cdots, X_1) \ge 0$, the limit exists. □

Lemma 4.23. *(Cesaro mean). If $\lim_{n\to+\infty} a_n = a$, then $\lim_{n\to+\infty} \frac{1}{n}\sum_{i=1}^n a_i = a$.*

Proof. For any $\varepsilon > 0$, there exists a n_0 such that for all $n \ge n_0$, , $|a_n - a| < \varepsilon$. Hence

$$\begin{aligned} \left|\frac{1}{n}\sum_{i=1}^n a_i - a\right| &\le \frac{1}{n}\sum_{i=1}^n |a_i - a| \\ &\le \frac{1}{n}\sum_{i=1}^{n_0} |a_i - a| + \frac{n-n_0}{n}\varepsilon \\ &\le \frac{1}{n}\sum_{i=1}^{n_0} |a_i - a_0| + \varepsilon. \end{aligned}$$

Sending $n \to +\infty$ makes the first term vanish and then result follows. □

We now can give a proof of Theorem 4.21.

Proof of Theorem 4.21. By the chain rule for conditional entropy,

$$\frac{H(X_1, \cdots, X_n)}{n} = \frac{1}{n}\sum_{i=1}^{n} H(X_i|X_{n-1}, \cdots, X_1).$$

By above Lemma 4.22 the conditional entropies converge. Using Cesaro means, Lemma 4.23, the above running average of conditional entropies converges to $\lim_{n\to+\infty} H(X_n|X_{n-1}, \cdots, X_1)$. □

Example 4.24. A discrete stochastic process $X = (X_i)_{i\geq 1}$ is a Markov chain if

$$\mathbb{P}(X_{n+1} = x_{n+1} \,|\, X_n = x_n, \cdots, X_1 = x_1) = \mathbb{P}(X_{n+1} = x_{n+1} \,|\, X_n = x_n)$$

for all n and all $x_1, \cdots, x_n \in \mathcal{X}$.

A Markov chain is time-invariant if

$$\mathbb{P}(X_{n+1} = b|X_n = a) = \mathbb{P}(X_2 = b \,|\, X_1 = a)$$

for all n and all $a, b \in \mathcal{X}$.

A time-invariant Markov chain with state space $\mathcal{X} = \{x_1, \cdots, x_m\}$ is characterised by its initial state X_1 and its probability transition matrix $P = (P_{i,j})_{m\times m}$, where $P_{i,j} := \mathbb{P}(X_2 = x_i \,|\, X_1 = j)$. In this case, the pmf of X_{n+1} is given as $p_{X_{n+1}}(x_j) = \sum_i p_{X_n}(x_i)P_{i,j}$.

Given a time-invariant Markov process X, the distribution p_{X_n} on $\mathcal{X}$ is called stationary distribution of X if $p_{X_{n+1}} = p_{X_n}$. Hence, a pmf μ on $\mathcal{X}$ is a stationary distribution, if $\mu_j = \sum_i \mu_i P_{i,j}$ for all j, where $\mu_i = \mu(x_i)$, or in matrix notation (with $\mu = (\mu_1, \cdots, \mu_m)$)

$$\mu = \mu P.$$

A time-invariant Markov chain with stationary distribution μ and initial state $X_1 \sim \mu$ is a stationary stochastic process and its entropy rate is given by

$$\mathcal{H}(X) = \lim_{n\to+\infty} H(X_n \,|\, X_{n-1}, \cdots, X_1) = \lim H(X_n \,|\, X_{n-1}) = H(X_2|X_1).$$

Using the definition of conditional entropy this becomes

$$\mathcal{H}(X) = \sum_i \mathbb{P}(X_1 = x_i)H(X_2|X_1 = x_i) = -\sum_i \mu_i \left(\sum_j P_{i,j}\log(P_{i,j})\right) = -\sum_{i,j} \mu_i P_{i,j}\log(P_{i,j}).$$

Example 4.25. Let $X = (X_i)$ be Markov chain with two states $\mathcal{X} = \{a, b\}$ and $\mathbb{P}(X_2 = b|X_1 = a) = \alpha, \mathbb{P}(X_2 = a|X_1 = b) = \beta$, that is

$$P = \begin{pmatrix} 1-\alpha & \alpha \\ \beta & 1-\beta \end{pmatrix}.$$

Then the stationary distributions is $\mu(a) = \frac{\beta}{\alpha+\beta}, \mu(b) = 1 - \mu(\alpha)$. If $X_1 \sim \mu$, then

$$\mathcal{H}(X) = \frac{\beta}{\alpha+\beta}h(\alpha) + \frac{\alpha}{\alpha+\beta}h(\beta),$$

where $h(\alpha)$ resp. $h(\beta)$ denotes the entropy of a Bernoulli random variable with probability α resp. β.

Example 4.26. Consider a connected[2] graph (V, E) with vertices $V = \{1, \cdots, m\}$ and without self-connection. Associate with the edge connecting node i and j a weight $w_{i,j} = w_{j,i} \geq 0$ (if there's no edge, set $w_{i,j} = 0$). Define a Markov chain on the set of vertices V by

$$P_{i,j} = \mathbb{P}(X_{n+1} = j \,|\, X_n = i) = \frac{w_{i,j}}{\sum_{k=1}^m w_{i,k}}.$$

(Choose the next vertex at random from the neighbouring vertices, with probabilities proportional to the weight of the connecting edge). We can guess the stationary distribution: the probability of being at vertex i should be proportional to the total weight of the edges emanating from this vertex. That is, if we denote the total weight of edges connecting to vertex i by $w_i = \sum_j w_{j,i}$, and the sum of weight of all edges by $w = \sum_{j>i} w_{i,j}$. Then $\sum_i w_i = 2w$ and we expect that $\mu_i = \frac{w_i}{2w}$. Indeed, we can directly verify $\mu P = \mu$:

$$\sum_i \mu_i P_{i,j} = \sum_i \frac{w_i}{2w} \frac{w_{i,j}}{w_i} = \frac{1}{2} \sum_i \frac{w_{i,j}}{2} = \frac{w_j}{2w} = \mu_j.$$

It is interesting to note that μ_i does not change if the edge weights connecting to vertex i stay the same, but the other weights are changed subject to having the same total weight. To calculate the entropy rate

$$\begin{aligned}
\mathcal{H}(X) &= H(X_2|X_1) = -\sum_i \mu_i \sum_j P_{i,j} \log(P_{i,j}) \\
&= -\sum_i \frac{w_i}{2w} \sum_j \frac{w_{i,j}}{w_i} \log\left(\frac{w_{i,j}}{w_i}\right) \\
&= -\sum_{i,j} \frac{w_{i,j}}{2w} \log\left(\frac{w_{i,j}}{w_i}\right) \\
&= -\sum_{i,j} \frac{w_{i,j}}{2w} \log\left(\frac{w_{i,j}}{w_i}\frac{2w}{2w}\right) \\
&= -\sum_{i,j} \frac{w_{i,j}}{2w} \log\left(\frac{w_{i,j}}{2w}\right) + \sum_{i,j} \frac{w_{i,j}}{2w} \log\left(\frac{w_i}{2w}\right) \\
&= -\sum_{i,j} \frac{w_{i,j}}{2w} \log\left(\frac{w_{i,j}}{2w}\right) + \sum_{i} \frac{w_i}{2w} \log\left(\frac{w_i}{2w}\right) \\
&= H\left(\cdots, \frac{w_{i,j}}{2w}, \cdots\right) - H\left(\cdots, \frac{w_i}{2w}, \cdots\right).
\end{aligned}$$

In practice, one is often not directly interested in the Markov chain $X = (X_i)$ but to understand the process Y defined by a function of X, i.e., $Y_i = \phi(X_i)$. For example, think of X as a complicated system that evolves over time but we only observe the current state of the system partially. A basic question is to determine the entropy rate of the stochastic process Y . This is a complicated question since in general Y itself is not a Markov chain so we can't directly apply the results of the previous section (as an exercise, you can prove that Y is Markov iff $\phi(\cdot)$ is injective or constant). However, we know that $H(Y)$ is well-defined since Y is stationary.

A first approach is to simply estimate $\mathcal{H}(Y)$ by the first n observations as $H(Y_n \,|\, Y_{n-1}, \cdots, Y_1)$. However, the convergence $\mathcal{H}(Y) = \lim_n H(Y_n|Y_{n-1}, \cdots, Y_1)$ can be very slow so we have no means to decide whether this estimate is good for a given n! The theorem below shows that the difference $H(Y_n \,|\, Y_{n-1}, \cdots, Y_1) - H(Y_n|Y_{n-1}, \cdots, Y_1, X_1)$ gives guarantees for the goodness of this estimation.

[2] A graph is connected if every pair of vertices can be connected by a path of edges.

Theorem 4.27. *Let $X = (X_i)_{i\geq 1}$ be a stationary Markov chain and $\phi : \mathcal{X} \longrightarrow \mathcal{Y}$. Let $Y = (Y_i)_{i\geq 1}$ with $Y_i := \phi(X_i)$. Then*

$$H(Y_n|Y_{n-1}, \cdots, Y_1, X_1) \leq \mathcal{H}(Y) \leq H(Y_n|Y_{n-1}, \cdots, Y_1)$$

and $\mathcal{H}(Y) = \lim_{n\to+\infty} H(Y_n|Y_{n-1}, \cdots, Y_1, X_1) = \lim_{n\to+\infty} H(Y_n|Y_{n-1}, \cdots, Y_1)$.

Since $H(Y_n|Y_{n-1}, \cdots, Y_1)$ converges monotonically from above to $\mathcal{H}(Y)$, the theorem follows by combining the following two lemmas.

Lemma 4.28. $H(Y_n|Y_{n-1}, \cdots, Y_2, X_1) \leq \mathcal{H}(Y)$.

Proof. Using that $Y_1 = \phi(X_1)$, the Markovianity of X, that $Y_i = \phi(X_i)$ we get for any integer k that

$$\begin{aligned} H(Y_n|Y_{n-1}, \cdots, Y_2, X_1) &= H(Y_n|Y_{n-1}, \cdots, Y_2, Y_1, X_1) \\ &= H(Y_n|Y_{n-1}, \cdots, Y_2, Y_1, X_1, X_0, X_{-1}, \cdots, X_{-k}) \\ &= H(Y_n|Y_{n-1}, \cdots, Y_2, Y_1, X_1, X_0, X_{-1}, \cdots, X_{-k}, Y_0, \cdots, Y_{-k}) \\ &\leq H(Y_n|Y_{n-1}, \cdots, Y_1, Y_0, \cdots, Y_{-k}) \\ &= H(Y_{n+k+1}|Y_{n+k}, \cdots, Y_1), \end{aligned}$$

where the inequality is because the conditioning reduces entropy. So

$$H(Y_n|Y_{n-1}, \cdots, Y_2, Y_1) \leq \lim_k H(Y_{n+k+1}|Y_{n+k}, \cdots, Y_1) = \mathcal{H}(Y).$$

□

Lemma 4.29. $H(Y_n|Y_{n-1}, \cdots, Y_1) - H(Y_n|Y_{n-1}, \cdots, Y_1, X_1) \to 0$ *as* $n \to +\infty$.

Proof. $I(X_1; Y_n \,|\, Y_{n-1}, \cdots, Y_1) = H(Y_n \,|\, Y_{n-1}, \cdots, Y_1) - H(Y_n \,|\, Y_{n-1}, \cdots, Y_1, X_1)$.
Since $I(X_1; Y_n, Y_{n-1}, \cdots, Y_1) \leq H(X_1)$ and $n \mapsto I(X_1; Y_n, Y_{n-1}, \cdots, Y_1)$ increases, the limit

$$\lim_n I(X_1; Y_n, Y_{n-1}, \cdots, Y_1) \leq H(X_1)$$

exists. By the chain rule,

$$I(X_1; Y_n, Y_{n-1}, \cdots, Y_1) = \sum_{i=1}^{n} I(X_1; Y_i \,|\, Y_{i-1}, \cdots, Y_1),$$

so combining with the above we get

$$+\infty > H(X_1) \geq \sum_{i=1}^{+\infty} I(X_1; Y_i \,|\, Y_{i-1}, \cdots, Y_1),$$

thus $\lim_{n\to+\infty} I(X_1; Y_n \,|\, Y_{n-1}, \cdots, Y_1) = 0$. □

4.7 Combining symbol and channel coding for DMCs [not examinable]

Consider a source that generates symbols from a finite set $\mathcal{V}$. We model this source as a discrete stochastic process $V = (V_i)$ with state space $\mathcal{V}$. Our goal is to transmit a sequence of symbols $V^n := (V_1, \cdots, V_n)$

over a DMC. Therefore we use an encoder $c : \mathcal{V}^n \longrightarrow \mathcal{X}^n$ and recover V^n from the output sequence Y^n by using a decoder $d : \mathcal{Y}^n \longrightarrow \mathcal{V}^n$. We want to do this in such away that $\mathbb{P}(V^n \neq \hat{V}^n)$ is small.

Theorem 4.30. *Let $(\mathcal{X}, M, \mathcal{Y})$ be a DMC with channel capacity C. Let $V = (V_i)_{i \geq 1}$ be a discrete stochastic process in a finite state space $\mathcal{V}$. If V satisfies the AEP and*

$$\mathcal{H}(V) < C,$$

then for every $\varepsilon > 0$ there exists an $n \geq 1$, a map $c : \mathcal{V}^n \longrightarrow \mathcal{X}^n$, and a map $d : \mathcal{Y}^n \longrightarrow \mathcal{V}$ such that $\mathbb{P}(V^n \neq \hat{V}^n) < \varepsilon$. Conversely, for any stationary stochastic process V, if $\mathcal{H}(V) > C$, there exists a constant $\delta > 0$ such that $\mathbb{P}(V^n \neq \hat{V}^n) > \delta$ for any coder-decoder pair, for any $n \geq 1$.

Sketch of Proof. There exists a typical set $\mathcal{T}_\varepsilon^{(n)}$ of size $|\mathcal{T}_\varepsilon^{(n)}| \leq 2^{n(\mathcal{H}(V)+\varepsilon)}$ such that and $\mathbb{P}(V^n \in \mathcal{T}_\varepsilon^{(n)}) \geq 1 - \varepsilon$. Now consider a coder that only encodes elements in $\mathcal{T}_\varepsilon^{(n)}$ and elements in $\mathcal{V}^n \backslash \mathcal{T}_\varepsilon^{(n)}$ are all encoded as the same codeword (representing error). We need at most

$$n(\mathcal{H}(V) + \varepsilon)$$

bits to index elements in $\mathcal{T}_\varepsilon^{(n)}$. Using channel coding we can transmit such an index with probability of error less than

$$\mathcal{H}(V) + \varepsilon = R < C.$$

The decoder reconstructs V^n by enumerating the typical set $\mathcal{T}_\varepsilon^{(n)}$ and decoding the received index $Y^n = (Y_1, \cdots, Y_n)$ to get $\hat{V}^n$. Then for a large enough n,

$$\mathbb{P}(V^n \neq \hat{V}^n) \leq \mathbb{P}(V^n \notin \mathcal{T}_\varepsilon^{(n)}) + \mathbb{P}(d(Y^n) \neq V^n \mid V^n \in \mathcal{T}_\varepsilon^{(n)}) \leq \varepsilon + \varepsilon.$$

This shows the first part of the theorem (achievability). For the second part (optimality) we need to show that

$$\mathbb{P}(V^n \neq \hat{V}^n) \to 0$$

implies $\mathcal{H}(V) \leq C$ for any sequenced (c^n, d^n) of channel codes. By Fano's inequality,

$$\begin{aligned} H(V^n|\hat{V}^n) &\leq 1 + \mathbb{P}(\hat{V}^n \neq V) \log(|\mathcal{V}^n|) \\ &= 1 + \mathbb{P}(\hat{V}^n \neq V) n \log(|\mathcal{V}^n|). \end{aligned}$$

Now

$$\begin{aligned} \mathcal{H}(V) &\leq \frac{H(V_1, \cdots, V_n)}{n} \\ &= \frac{1}{n} H(V_1, \cdots, V_n \mid \hat{V}_1, \cdots, \hat{V}_n) + \frac{1}{n} I(V^n; \hat{V}^n) \\ &\leq \frac{1}{n} \Big[1 + \mathbb{P}(V^n \neq \hat{V}^n) n \log(|\mathcal{V}|)\Big] + \frac{1}{n} I(V^n; \hat{V}^n) \\ &\leq \frac{1}{n} \Big[1 + \mathbb{P}(V^n \neq \hat{V}^n) n \log(|\mathcal{V}|)\Big] + \frac{1}{n} I(X_1, \cdots, X_n; Y_1, \cdots, Y_n) \\ &\leq \frac{1}{n} + \mathbb{P}(V^n \neq \hat{V}^n) \log(|\mathcal{V}|) + C, \end{aligned}$$

where we used: the definition of entropy rate, the definition of mutual information, Fano's inequality, the data processing inequality, and finally, the definition of capacity of a DMC. Letting $n \to +\infty$ finishes the proof since

$$\mathcal{H}(V) \leq \log(|\mathcal{V}|) \lim_{n \to +\infty} \mathbb{P}(V^n \neq \hat{V}^n) + C = C.$$

□

We emphasise that above theorem makes no assumptions on the stochastic process V other than that the AEP holds; the sequence of random variables $(V_1, \cdots, V_n)$ can have very complicated dependencies. Most importantly, the theorem implies that a two-stage approach — given by firstly using symbol coding and then applying channel coding — achieves the same rates as applying source coding alone. This two-stage approach is advantageous from an engineering perspective since it divides a complicated problem into two smaller problems.

To sum up: source coding compresses the information using that by the AEP there exists a set of small cardinality $\approx 2^{nH}$ that carries most of the probability mass. Hence, we can use H bits per symbol to use a symbol code to compress the source. Channel coding uses that by the joint AEP, we have for large n with high probability that input and output are jointly typical; only with probability $\approx 2^{-nI}$ any other codeword will be jointly typical. Thus we can 2^{nI} codewords. Theorem 4.30 shows that we can design source code and channel code separately without loss of performance.

Appendix A

Probability theory

We briefly recall and introduce basic notation from probability theory. We refer the reader to [3, 4] for an elementary introduction to probability theory and to [2, 5] for a more exhaustive treatment.

A.1 Measure theory

A *measurable space* $(\mathcal{X}, \mathcal{A})$ consists of a set $\mathcal{X}$ and and a *σ-algebra* $\mathcal{A}$, that is a collection of subsets of $\mathcal{X}$ such that

(1) $\mathcal{X} \in \mathcal{A}$;

(2) $A \in \mathcal{A}$ implies $A^c \in \mathcal{A}$;

(3) if $A_n \in \mathcal{A}$ then $\cup_{n \in \mathbb{N}} A_n \in \mathcal{A}$.

Example A.1. Take $\mathcal{X} = \{a, b, c, d]\}$ and $\mathcal{A} = \{\emptyset, \{a, b\}, \{c, d\}, \{a, b, c, d\}\}$, then $(\mathcal{X}, \mathcal{A})$ is a discrete measurable space.

Take $\mathcal{X} = \mathbb{R}$ and $\mathcal{A}$ the smallest σ-algebra that contains all open sets ("Borel σ-algegra"), then $(\mathcal{X}, \mathcal{A})$ is one of the most often-used measurable spaces.

Given two measurable spaces $(\mathcal{X}_1, \mathcal{A}_1)$ and $\mathcal{X}_2, \mathcal{A}_2)$, we call a map $X : \mathcal{X}_1 \longrightarrow \mathcal{X}_2$ measurable with respect to $\mathcal{A}_1 \backslash \mathcal{A}_2$ if

$$X^{-1}(A) \in \mathcal{A}_1 \text{ for } \forall A \in \mathcal{A}_2.$$

It is a good exercise to show that the space of measurable maps (with respect to $\mathcal{A}_1 \backslash \mathcal{A}_2$) is closed under addition, scalar multiplication, $\liminf, \limsup$, etc.

A.2 Probability spaces

A *probability space* $(\Omega, \mathcal{F}, \mathbb{P})$ is a measurable space $(\Omega, \mathcal{F})$ together with a map $\mathbb{P} : \mathcal{F} \longrightarrow [0, 1]$ such that

(1) $\mathbb{P}(\Omega) = 1$,

(2) (σ-additivity) $\mathbb{P}(\cup_n A_n) = \sum \mathbb{P}(A_n)$ for disjoint $(A_n) \subset \mathcal{F}$ (i.e. $A_i \cap A_j = \emptyset$ for any $i \neq j$).

We refer to Ω as *sample space*, to elements of $\mathcal{F}$ as *events*, and to $\mathbb{P}$ as the *probability measure*. An $\mathcal{F}\backslash\mathcal{A}$-measurable map $X : \Omega \longrightarrow \mathcal{X}$ from Ω to another measurable space $\mathcal{X}$ with σ-algebra $\mathcal{A}$ is called a *random variable*, and $\mathcal{X}$ is called the *state space*.

Example A.2. A player flips a coin and wins one pound if it is a head, otherwise the player wins nothing. We can model this as follows: Let $\Omega = \{H, T\}, \mathcal{F} = \{\emptyset, \{H\}, \{T\}, \{H, T\}\}$, and

$$X(\omega) = \begin{cases} 1 & \text{if } \omega = H \\ 0 & \text{if } \omega = T \end{cases}.$$

Given any number $p \in [0, 1]$, we can define a probability measure by $\mathbb{P}(H) = p, \mathbb{P}(T) = 1 - p$[1] Notice that with different value $p \in [0, 1]$ we get different probability measure $\mathbb{P}$,

Example A.3. For an integer N, let $\Omega = \{H, T\}^N$ and $\mathcal{F}$ be the class of all subsets of Ω. Then $X_i(\omega) := \begin{cases} 1 & \text{if } \omega_i = H \\ 0 & \text{if } \omega_i = T \end{cases}$ is a random variable on $(\Omega, \mathcal{F})$ and so is

$$X_1 + \cdots + X_n$$

(the number of heads in n coin tosses).

We call two events $A, B \in \mathcal{F}$ *independent* events if $\mathbb{P}(A \cap B) = \mathbb{P}(A)\mathbb{P}(B)$. Otherwise, we call them dependent. Given two random variables on $(\Omega, \mathcal{F}, \mathbb{P})$ we say that X and Y are independent if $\{X \in A\}$[2] and $\{Y \in B\}$ are independent for all $A, B \in \mathcal{A}$. In the case of the discrete random variables, it is sufficient to require $\mathbb{P}(X = x, Y = y) = \mathbb{P}(X = x)\mathbb{P}(Y = y)$ for all $x \in X(\Omega), y \in Y(\Omega)$.

A.3 Discrete random variables

Throughout this course, we are mostly interested in random variables that take values in a countable set. More precisely, we call $X : \Omega \longrightarrow \mathbb{R}$ a discrete random variable, if the image $X(\Omega)$ is a countable subset of $\mathbb{R}$ and $X^{-1}(x) \in \mathcal{F}$ for all $x \in \mathbb{R}$. In this course, we often denote the image space of X with $\mathcal{X}$. Given a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ and a discrete random variable X, we call

$$p_X(x) := \mathbb{P}(X = x)$$

the probability mass function (pmf) of X (also distribution of X).

Example A.4. In Example A.2, X is a discrete random variable with the image space $X(\Omega) = \{0, 1\}$.

We can regard two discrete random variable X, Y with image spaces $\mathcal{X}, \mathcal{Y}$ as one discrete random varible (X, Y) with image space $\mathcal{X} \times \mathcal{Y}$. We call

$$p_{X,Y}(x, y) := \mathbb{P}(X = x, Y = y) = \mathbb{P}((X, Y) = (x, y))$$

the joint pmf of X and Y. Given a pmf on $\mathcal{X} \times \mathcal{Y}$ we call

$$p_X(x) := \sum_{y \in \mathcal{Y}} p_{X,Y}(x, y)$$

the marginal on $\mathcal{X}$, and the marginal on $\mathcal{Y}$ is defined similarly.

[1] To be rigorous, we should write $\mathbb{P}(\{H\}) = p$, which is often simplified to the notation $\mathbb{P}(H) = p$.

[2] The rigorous expression for $\{X \in A\}$ is $\{\omega : X(\omega) \in A\}$.

A.4 Expectation

Given a probaiblity space $(\Omega, \mathcal{F}, \mathbb{P})$ and a discrete random variable $X : \Omega \longrightarrow \mathcal{X} \subset \mathbb{R}$, we call

$$\mathbb{E}[X] := \sum_{x \in \mathcal{X}} x\mathbb{P}(X = x)$$

the expectation of X whenever this sum converges absolutely. If X and Y are discrete random variable defined on $(\Omega, \mathcal{F}, \mathbb{P})$, so are (X, Y) and any measurable function of (X, Y).

We call $\mathrm{Var}[X] := \mathbb{E}[(X - E[X])^2]$ the variance of X (if this expectation exists) and

$$\mathrm{Cov}[X, Y] := \mathbb{E}[(X - \mathbb{E}[X])(Y - \mathbb{E}[Y])]$$

the covariance of X and Y.

It is well-known that X and Y are independent (denoted as $X \perp Y$) iff

$$\mathbb{E}[f(X)g(Y)] = \mathbb{E}[f(X)]\mathbb{E}[g(Y)]$$

for all functions f, g for which the two expectations on the right hand side exists.

A.5 Conditional probabilities and conditional expectations

Given a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ and $A \in \mathcal{F}$ with $\mathbb{P}(A) > 0$, we define the conditional probability $\mathbb{P}(\cdot|A) : \mathcal{F} \longrightarrow [0, 1]$ as

$$\mathbb{P}(B|A) = \frac{\mathbb{P}(A \cap B)}{\mathbb{P}(A)}.$$

Note that $(\Omega, \mathcal{F}, \mathbb{Q}_A)$ is a probability space with $\mathbb{Q}_A(\cdot) := \mathbb{P}(\cdot|A)$. Given two discrete random variables X and Y, we call

$$p_{Y|X}(y|x) := p_{Y|X=x}(y) := \mathbb{P}(Y = y|X = x) = \begin{cases} \frac{p_{X,Y}(x,y)}{p_X(x)} & \text{if } p_X(x) > 0 \\ 0 & \text{otherwise} \end{cases}$$

the conditional pmf of Y given X.

For $A \in \mathcal{F}$ and a discrete random variable X, define the conditional expectation of X given A as

$$\mathbb{E}[X|A] := \sum_{x \in \mathcal{X}} xP(X = x|A).$$

We often apply this with $A = \{Y = y\}$ where Y is another discrete random variable, i.e. $\mathbb{E}[X|A] = \mathbb{E}[X|Y = y]$.

Appendix B

Convexity

Definition B.1. *We call* $f : \mathbb{R} \longrightarrow \mathbb{R}$ *be convex, if*

$$f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda)f(y)$$

for all $x, y \in \mathbb{R}$ *and* $\lambda \in [0,1]$. *We call* f *strictly convex if above is a strict inequality for all* $\lambda \in (0,1)$.

Theorem B.2. *(Jensen's inequality). Let* X *be a real-valued random variable such that* $\mathbb{E}[X]$ *exists. If* $\phi : \mathbb{R} \longrightarrow \mathbb{R}$ *is a convex function such that* $\mathbb{E}[|\phi(X)|] < +\infty$, *then*

$$\phi(\mathbb{E}[X]) \leq \mathbb{E}[\phi(X)].$$

If ϕ *is strictly convex, then the equality holds iff* X *is constant with probability one.*

Bibliography

[1] Thomas Cover, *Elements of information theory.* John Wiley & Sons, 2012.

[2] Richard M Dudley. *Real analysis and probability*, volume 74. Cambridge University Press, 2002.

[3] Geoffrey Grimmett and David Stirzaker. *Probability and random processes.* Oxford university press, 2001.

[4] Geoffrey Grimmett and Dominic Welsh. *Probability: an introduction.* Oxford University Press, 2014.

[5] Paul Malliavin. *Integration and probability*, volume 157. Springer Science & Business Media, 1995.