Marco Tomamichel

EE5139/EE6139:
Information Theory and its Applications

(Semester I, 2021–2022)

Disclaimer: These notes are not yet free of typos or always presented in the most clear way. Any comments that help reduce these deficiencies are very much appreciated. Parts of Chapters 0 and Chapter 7 are based on notes by Vincent Y. F. Tan. Most figures were contributed by Michael X. Cao.

Contents

$\emptyset$	empty set $\{\}$
$[M]$	the set $\{1, 2, \ldots, M\}$
$\mathcal{P}(\mathcal{X})$	the power set of $\mathcal{X}$, i.e. $\{A : A \subseteq \mathcal{X}\}$
$\mathcal{X} \times \mathcal{Y}$	the set of tuples $\{(x, y) : x \in \mathcal{X}, y \in \mathcal{Y}\}$
$\mathcal{X}^n$	the set of n-tuples with each element taking values in $\mathcal{X}$, e.g., $\mathcal{X}^2 = \mathcal{X} \times \mathcal{X}$
$\{0,1\}^n$	the set of n-bit strings
$\{0,1\}^*$	the set of bit strings of arbitrary length
$\max \mathcal{X}$	largest $x^* \in \mathcal{X}$, might not always exist
$\sup \mathcal{X}$	smallest $x^* \in \mathbb{R}$ such that $x \leq x^*$ for all $x \in \mathcal{X}$; equals the maximum, $\max \mathcal{X}$, if it exists
$\min \mathcal{X}$	smallest $x^* \in \mathcal{X}$, might not always exist
$\inf \mathcal{X}$	largest $x^* \in \mathbb{R}$ such that $x \geq x^*$ for all $x \in \mathcal{X}$; equals the minimum, $\min \mathcal{X}$, if it exists
$\mathbf{1}\{x = y\}$	indicator function, evaluates to 1 if the condition is true and 0 otherwise, so that, for example, $\mathbf{1}\{x = y\} + \mathbf{1}\{x \neq y\} = 1$
δ_{xy}	shorthand for $\mathbf{1}\{x = y\}$
$P_X(x)$	probability mass function (pmf), $P_X(x) = P[X = x]$
$p_X(x)$	probability density function (pdf), i.e. $P[X \in (1,2)] = \int_1^2 p_X(x)\mathrm{d}x$
$P[X \in \mathcal{A}]$	probability of a random variable X being in some set $\mathcal{A}$, i.e. $P[X \in \mathcal{A}] = \mathbb{P}(\{\omega : X(\omega) \in \mathcal{A}\}) = \sum_{x \in \mathcal{A}} P_X(x)$
$P[5 \leq X < 6]$	another way of writing $P[X \in [5, 6)]$
log	logarithm; in these notes we take the logarithm to base 2, i.e. $\log = \log_2$

Table 1: Some basic notation used in this module.

pmf	probability mass function
pdf	probability density function
cdf	cumulative density function
rv	random variable
DMS	discrete memoryless source
DMC	discrete memoryless channel

Table 2: Some abbreviations used in this module.

Chapter 0

Review of mathematical notation and foundations

[Week 1]

Intended learning outcomes:

- You are familiar with common notation used throughout the lecture.
- You are comfortable with the main mathematical concepts needed in this module, namely basic probability theory including random variables, conditional probabilities and Markov chains.
- You can apply basic bounds on tail probabilities, and can prove the weak law of large numbers.
- You can compute vector norms and apply the Cauchy-Schwarz inequality.
- You know what convex and concave functions are and can apply Jensen's inequality.
- You know what finite fields are and how to come up with the multiplication table for simple examples.

0.1 Notation

We will use standard notation and abbreviations that you should be familiar with from other modules. Some of the less frequently encountered mathematical expressions are summarised in Tables 1 and 2 on the previous page.

0.2 Probability theory

We will not directly need the framework of probability theory in its most abstract formulation as presented in the following, but it is good to know that both discrete and continuous random variables can be seen as emanating from a shared mathematical framework.

0.2.1 Probability space

A probability space is represented by a triple $(\Omega, \Sigma, \mathbb{P})$. Here Ω is a set that is called the *sample space*. Moreover, Σ is a σ-algebra, i.e. a collection of subsets of Ω, called events, with the following properties:

- $\Omega \in \Sigma$
- If $A \in \Sigma$, then its *complement*, $A^c = \Omega \setminus A$ is also in Σ, i.e. $A^c \in \Sigma$.
- If $A_1, A_2, \ldots, A_n, \ldots \in \Sigma$, then $\bigcup_{i=1}^{\infty} A_i \in \Sigma$

Question 0.1. *Show that the above also implies that $\emptyset \in \Sigma$ and $\bigcap_{i=1}^{\infty} A_i \in \Sigma$.*

For example, let $\Omega = [0, 1]$, and we are interested in the probability of subsets of Ω that are intervals of the form $[a, b]$ where $0 \leq a < b \leq 1$, but not individual points in Ω. Then we should also be able to say something about the probability of the union, intersections, complement and so on of such intervals. This is captured by the definition of a σ-algebra. Think of Σ as the properties of Ω that can actually be observed.

Example 0.2. *If your random variable is the location an athlete lands after a long jump then it makes sense to take Ω to be positive real numbers, $\mathbb{R}_+$ indicating the distance jumped (say, in meters). However, even with arbitrarily good equipment we cannot actually measure a real number, we can only ever say that he landed in some interval, the size of which is given by our measurement precision. Thus, Σ, comprised of the events we can actually observe, is built up by including all (arbitrarily small) intervals in $\mathbb{R}_+$ and their unions and complements. Or another way of looking at this is that the probability of the jumper landing exactly at $9m$ is always zero — it is simply the wrong question to ask. But the probability of landing within $1cm$ or some arbitrarily small interval around $9m$ might very well be nonzero.*

Question 0.3. *For the advanced reader: Note however that in the above example $\{x\} \in \Sigma$ for any $x \in \mathbb{R}^+$, that is, single points are also elements of the σ-algebra. Can you see why? Use an infinite intersection to construct it.*

Finally, the probability measure $\mathbb{P}$ is a function $\mathbb{P} : \Sigma \to [0, 1]$ defined on the measurable space (Ω, Σ), and represents your "belief" about the events in Σ. In order for $\mathbb{P}$ to be called a probability measure, it must satisfy the following two properties:

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

2. For $A_1, A_2, \ldots$ such that $A_i \cap A_j = \emptyset$ for all $i \neq j$, i.e. for mutually *disjoint* sets, we have

$$\mathbb{P}\left(\bigcup_{i=1}^{\infty} A_i\right) = \sum_{i=1}^{\infty} \mathbb{P}(A_i) \tag{1}$$

Some basic and very useful properties that can be derived from the above definition. The union bound in particular is very often used when analysing problems in information theory.

Proposition 0.4. *Let $(\Omega, \Sigma, \mathbb{P})$ be a probability space. The following holds true:*

1. $\mathbb{P}(A^c) = 1 - \mathbb{P}(A)$
2. *If $A \subset B$, then $\mathbb{P}(A) \leq \mathbb{P}(B)$*
3. $\mathbb{P}(A \cup B) = \mathbb{P}(A) + \mathbb{P}(B) - \mathbb{P}(A \cap B) \leq \mathbb{P}(A) + \mathbb{P}(B)$, *which is called the* union bound. *Clearly, by induction, the union bound works for finitely many sets $A_i, i = 1, \ldots, k$, namely*

$$\mathbb{P}\left(\bigcup_{i=1}^{\infty} A_i\right) \leq \sum_{i=1}^{\infty} \mathbb{P}(A_i). \tag{2}$$

Proof. Property 1 follows since $A^c \cap A = \emptyset$, and thus $\mathbb{P}(A) + \mathbb{P}(A^c) = \mathbb{P}(\Omega) = 1$ by (1). For Property 2, note that $B \setminus A = B \cap A^c \in \Sigma$ and since $A \cap (B \setminus A) = \emptyset$ we again argue that $\mathbb{P}(A) + \mathbb{P}(B \setminus A) = \mathbb{P}(B)$, from which the desired inequality follows.

For Property 3 note that $A \cup B$ can be decomposed in three different ways into mutually disjoint sets:

$$A \cup B = A \cup (B \setminus A) = B \cup (A \setminus B) = (A \setminus B) \cup (B \setminus A) \cup (A \cap B)\,. \tag{3}$$

Again using (1) for each of these decompositions we have

$$\begin{aligned} 2\mathbb{P}(A \cup B) &= \mathbb{P}(A) + \mathbb{P}(B) + \mathbb{P}(B \setminus A) + \mathbb{P}(A \setminus B) && (4)\\ &= \mathbb{P}(A) + \mathbb{P}(B) + \mathbb{P}(A \cup B) - \mathbb{P}(A \cap B)\,, && (5) \end{aligned}$$

which implies the desired equality. □

Question 0.5. *Show that $0 \leq \mathbb{P}(A) \leq 1$ for every $A \in \Sigma$.*

Sometimes we have two conflicting beliefs, or models, about the underlying probability distribution, and so we will consider two compatible probability spaces $(\Omega, \Sigma, \mathbb{P})$ and $(\Omega, \Sigma, \mathbb{Q})$. They offer different predictions about the probability with which the events in Σ occur, and one fundamental task in statistics is to find out which model is the correct one from the frequency with which certain events occur. We will cover this later in the module.

0.2.2 Random variables

We will usually not deal directly with the probability space but with random variables. A *random variable* (rv) $X : \Omega \to \mathcal{X}$ is a function from the space (Ω, Σ) to a measurable space $(\mathcal{X}, \Sigma_X)$. In order for X to make any sense, the mapping has to ensure that $\{\omega \in \Omega : X(\omega) \in \mathcal{B}\} \in \Sigma$ for all $\mathcal{B} \in \Sigma_X$, because we are restricted to observing events in Σ and our random variable can thus not be more fine-grained than what Σ allows. Functions satisfying this property are called a *measurable function*. A random variable is then more formally defined as a measurable mapping from (Ω, Σ) to $(\mathcal{X}, \Sigma_X)$.

The only two examples of interest for us in the following are discrete and continuous random variables:

discrete rv: $\mathcal{X}$ is a discrete set and Σ_X is the power set $\mathcal{P}(\mathcal{X})$ of $\mathcal{X}$, i.e. the set of all subsets of $\mathcal{X}$.

continuous rv: $\mathcal{X} = \mathbb{R}$ and $\Sigma_X = \mathcal{B}$, the Borel σ-algebra. This is the smallest σ-algebra containing all open intervals in $\mathbb{R}$.

The probability measure $\mathbb{P}$ induces a probability measure P_X on $(\mathcal{X}, \Sigma_X)$, given by

$$P_X(B) = P[X \in B] = \mathbb{P}(\{\omega \in \Omega : X(\omega) \in B\}) \tag{6}$$

for all $B \in \Sigma'$. P_X is called the *distribution* of the random variable X.

If $\mathcal{X} = \{a_1, \ldots, a_d\}$ is discrete (and Σ_X the power set of $\mathcal{X}$), then we say that X is a *discrete random variable*. The distribution of X is then also known as the *probability mass function (pmf)* of X and is fully characterised by all the events consisting of a single value, i.e. the values $P_X(a_1), P_X(a_2), \ldots, P_X(a_d)$.

Question 0.6. *Can you give a formal argument why the values at these points are sufficient?*

Some random variables are not random at all. If there is an a_i with $P_X(a_i) = 1$ (and thus $P_X(a_j) = 0$ for all $j \neq i$), then we call this random variable *deterministic*. On the other extreme we have *uniformly distributed* random variables, where $P_X(a_i) = \frac{1}{d}$ for all $i \in [d]$.

Example 0.7. *The simplest example is the Bernoulli random variable. It is defined on a binary alphabet $\mathcal{X} = \{0, 1\}$ and we write $X \sim \mathrm{Bern}(\epsilon)$ to denote the rv with $P[X = 1] = \epsilon$ and $P[X = 0] = 1 - \epsilon$.*

Let us now consider a real-valued random variable X. If there exists a function $p_X : \mathbb{R} \to [0, \infty)$ such that for all $A \in \Sigma_X$, we have

$$P[X \in A] = \int_A p_X(x)\,\mathrm{d}x \tag{7}$$

then we say that X is a *continuous random variable*. The function p_X is called the *probability density function (pdf)* of X. We also define the *cumulative distribution function* (cdf) by integrating $p_X(x)$, that is, the cdf is given by $F_X(a) = \mathbb{P}[X \leq a] = \int_{-\infty}^{a} p_X(x)\,\mathrm{d}x$.

Question 0.8. *Show that* $\int_{\mathcal{X}} p_X(x) = 1$. *Moreover, if* p_X *is continuous at some point* x, *it must satisfy* $p_X(x) \geq 0$. *Can* $p_X(x)$ *ever be larger than* 1*?*

In this class, we deal mainly with discrete rvs, although we will also encounter Gaussian random variables, which are continuous, later on.

Example 0.9. *We denote the pdf of a* Gaussian *random variable* X *as*

$$\mathcal{N}(x; \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, \tag{8}$$

where μ *is the* mean *and* σ *the* standard deviation *of* X. *The* variance *of* X *is* σ^2. *A* normal Gaussian *random variable has* $\mu = 0$ *and* $\sigma = 1$. *The corresponding cdf is denoted as*

$$\Phi(y) = \int_{-\infty}^{y} \mathcal{N}(x; 0, 1)\, \mathrm{d}x. \tag{9}$$

Some additional notations and definitions for discrete random variables are given below. The counterparts for continuous random variables can be obtained by simply replacing pmfs with pdfs. Thus assume now that X and Y are discrete random variables taking on values in $\mathcal{X}$ and $\mathcal{Y}$ respectively. The joint pmf of X and Y is defined as

$$P_{X,Y}(x, y) = P[X = x \wedge Y = y] = \mathbb{P}\big(\{\omega \in \Omega : X(\omega) = x \wedge Y(\omega) = y\}\big). \tag{10}$$

Question 0.10. *Verify that* $P_Y(y) = \sum_{x' \in \mathcal{X}} P_{X,Y}(x', y)$.

With this in hand we can define conditional pmf's and a notion of independence of random variables.

- The conditional pmf is given by

$$P_{X|Y}(x|y) = \frac{P_{X,Y}(x, y)}{P_Y(y)} = \frac{P_{Y|X}(y|x) P_X(x)}{P_Y(y)}, \quad \text{for} \quad P_Y(y) > 0\,, \tag{11}$$

  where the second expression is often referred to as Bayes' rule. If $P_Y(y) = 0$ then the conditional pmf is simply not defined.

- X and Y are *independent random variables*, if and only if, for all $x \in \mathcal{X}$ and $y \in \mathcal{Y}$,

$$P_{X,Y}(x, y) = P_X(x) P_Y(y) \tag{12}$$

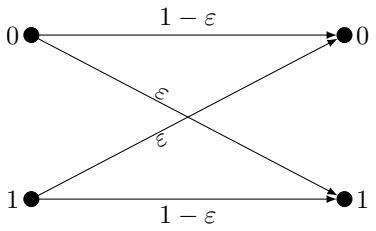
  or equivalently $P_{X|Y}(x|y) = P_X(x)$. The latter condition simply states that the conditional distribution $P_{X|Y}(x|y)$ does not depend on y.

Example 0.11 (Binary symmetric channel)**.** $X \sim \mathrm{Bern}(p)$ *is a bit that is sent over channel and is corrupted by additive noise* $Z \sim \mathrm{Bern}(\epsilon)$, *where* X *and* Z *are independent. The output of the channel is* $Y = X \oplus Z$. *The channel is fully defined by the conditional distribution* $P_{Y|X}$, *which we can compute as follows:*

$$P_{Y|X}(y|x) = P[X \oplus Z = y \mid X = x] = P[Z = y \oplus x \mid X = x] = P[Z = y \oplus x] = P_Z(y \oplus x) \tag{13}$$

Hence, the channel can be given as a matrix or pictorially as follows:

x	y	$P_{Y\|X}$
0	*0*	$1-\epsilon$
1	*0*	ϵ
0	*1*	ϵ
1	*1*	$1-\epsilon$



0.2.3 Expectation and variance

The expectation of a random variable X is defined to be

$$\mathbb{E}[X] = \int_{\Omega} X(\omega)\,\mathrm{d}\mathbb{P}(\omega). \tag{14}$$

This definition has a very precise mathematical meaning in measure theory, but here we are only interested in two special cases. If X is a discrete random variable this reduces to the familiar formula

$$\mathbb{E}[X] = \sum_{x\in\mathcal{X}} xP_X(x)\,. \tag{15}$$

If X is a continuous random variable with pdf $f_X(x)$, we have

$$\mathbb{E}[X] = \int_{\mathbb{R}} xp_X(x)\,\mathrm{d}x\,. \tag{16}$$

Note that the expectation is a statistical summary of the distribution of X, rather than depending on the realised value of X. If there are two different models $\mathbb{P}$ and $\mathbb{Q}$ we need to specify which probability measure we are using. We only do this when necessary (because the the model is not obvious from context) by adding a subscript $\mathbb{E}_P$ or $\mathbb{E}_Q$.

If g is a function, the expectation of $g(X)$ is given by

$$\mathbb{E}[g(X)] = \int_{\mathbb{R}} g(x)p_X(x)\,\mathrm{d}x. \tag{17}$$

Question 0.12. *Show that the expectation is linear, i.e.* $\mathbb{E}[aX+b] = a\mathbb{E}[X]+b$.

The variance of X is the expectation of $g(X) = (X-\mathbb{E}[X])^2$. Thus,

$$\mathrm{Var}(X) = \mathbb{E}[(X-\mathbb{E}[X])^2] = \int_{\mathbb{R}} (x-\mathbb{E}[X])^2 p_X(x)\,\mathrm{d}x\,. \tag{18}$$

Question 0.13. *Check from the above definition that the variance can also be expressed as*

$$\mathrm{Var}(X) = \mathbb{E}[X^2] - \mathbb{E}[X]^2. \tag{19}$$

Question 0.14. *Verify that* $\mathcal{N}(x;\mu,\sigma^2)$ *indeed has expectation* μ *and variance* σ^2.

0.2.4 Markov chains

Markov chains describe a notion of conditional independence. Let's start with the three random variables X, Y and Z. They are said to form a *Markov chain in the order*

$$X - Y - Z$$

if their joint distribution P_{XYZ} satisfies

$$P_{XYZ}(x, y, z) = P_X(x)P_{Y|X}(y|x)P_{Z|Y}(z|y) \qquad \text{for all} \qquad (x, y, z) \in \mathcal{X} \times \mathcal{Y} \times \mathcal{Z}. \tag{20}$$

This the same as saying that X and Z are *conditionally independent given* Y.

Question 0.15. *Assume $X - Y - Z$. Show that it is also true that $Z - Y - X$.*

Notice that if we do not assume anything about the joint distribution P_{XYZ}, then it factorizes (by repeated applications of Bayes rule) as

$$P_{XYZ}(x, y, z) = P_X(x)P_{Y|X}(y|x)P_{Z|XY}(z|x, y) \qquad \text{for all} \qquad (x, y, z) \in \mathcal{X} \times \mathcal{Y} \times \mathcal{Z} \tag{21}$$

so what Markovianty in the order $X - Y - Z$ buys us is that $P_{Z|XY}(z|x, y) = P_{Z|Y}(z|y)$ (i.e., we can drop the conditioning on X). In essence all the information that we can learn about Z is already contained in Y. No other information about Z can be gleaned from knowing X if we already know Y. Another way of saying this is that the conditional distribution of X and Z given $Y = y$ can be factorised as

$$P_{XZ|Y}(x, z|y) = P_{X|Y}(x|y)P_{Z|Y}(z|y) \qquad \text{for all} \qquad (x, y, z) \in \mathcal{X} \times \mathcal{Y} \times \mathcal{Z}. \tag{22}$$

Notice that this is in direct analogy to the situation where X and Z are (marginally) independent. Simply set Y to be a deterministic random variable (with only one possible outcome) to recover the definition of independence.

Question 0.16. *If Z is a deterministic function of Y, show that $X - Y - Z$ is true.*

Question 0.17. *If X and Z are conditionally independent given Y, this does not imply that X and Z are marginally independent (in general). Construct a counterexample.*

0.3 Tail bounds

In this section, we summarise some bounds on probabilities that we use extensively in the sequel. More precisely, we are interested in showing that the probability of a random variable deviating too far from its expectation value is small.

0.3.1 Basic bounds

We start with the familiar Markov and Chebyshev inequalities.

Proposition 0.18 (Markov's inequality)**.** *Let X be a real-valued non-negative random variable with pdf p_X. Then for any $a > 0$, we have*

$$P[X > a] \leq \frac{\mathbb{E}[X]}{a}. \tag{23}$$

Proof. By the definition of the expectation, we have

$$\mathbb{E}[X] = \int_0^\infty x p_X(x)\,\mathrm{d}x \geq \int_a^\infty x p_X(x)\,\mathrm{d}x \geq a\int_a^\infty p_X(x)\,\mathrm{d}x = aP[X > a]. \tag{24}$$

and we are done. □

Note that this bound only becomes nontrivial if a exceeds the expectation value $\mathbb{E}[X]$.

Question 0.19. *In which step is non-negativity of X used?*

Question 0.20. *Can you do the proof also for discrete random variables?*

If we let X above be the non-negative random variable $(X - \mathbb{E}[X])^2$, we obtain Chebyshev's inequality.

Proposition 0.21 (Chebyshev's inequality)**.** *Let X be a real-valued random variable with mean μ and variance σ^2. Then for any $a > 0$, we have*

$$P\big[|X - \mu| > a\sigma\big] \leq \frac{1}{a^2}. \tag{25}$$

Proof. Let X in Markov's inequality be the random variable $g(X) = (X - \mathbb{E}[X])^2$. This is clearly non-negative and the expectation of $g(X)$ is $\mathrm{Var}(X) = \sigma^2$. Thus, by Markov's inequality, we have

$$P[g(X) > a^2\sigma^2] \leq \frac{\sigma^2}{a^2\sigma^2} = \frac{1}{a^2}. \tag{26}$$

Now, $g(X) > a^2\sigma^2$ if and only if $|X - \mu| > a\sigma$ so the claim is proved. □

We now consider a collection of real-valued random variables that are independent and identically distributed (i.i.d.). In particular, let $X^n = (X_1, \ldots, X_n)$ be a collection of independent random variables where each X_i has distribution P with zero mean and finite variance σ^2.

Proposition 0.22 (Weak Law of Large Numbers)**.** *For every $\epsilon > 0$, we have*

$$\lim_{n\to\infty} P\left[\left|\frac{1}{n}\sum_{i=1}^n X_i\right| > \epsilon\right] = 0. \tag{27}$$

Consequently, the average $\frac{1}{n}\sum_{i=1}^n X_i$ converges to 0 in probability.

Note that for a sequence of random variables $\{S_n\}_{n=1}^{\infty}$, we say that this sequence *converges to a number* $b \in \mathbb{R}$ *in probability* if for all $\epsilon > 0$,

$$\lim_{n\to\infty} P\big[|S_n - b| > \epsilon\big] = 0. \tag{28}$$

We also write this as $S_n \xrightarrow{\text{p}} b$. Contrast this to convergence of numbers: We say that a sequence of numbers $\{s_n\}_{n=1}^{\infty}$ *converges to a number* $b \in \mathbb{R}$ if we have $\lim_{n\to\infty} |s_n - b| = 0$.

Proof. Let $\frac{1}{n}\sum_{i=1}^{n} X_i$ take the role of X in Chebyshev's inequality. Clearly, the mean is zero. The variance of X is

$$\mathrm{Var}\left(\frac{1}{n}\sum_{i=1}^{n} X_i\right) = \frac{1}{n^2}\mathrm{Var}\left(\sum_{i=1}^{n} X_i\right) = \frac{1}{n^2}\sum_{i=1}^{n}\mathrm{Var}(X_i) = \frac{\sigma^2}{n}. \tag{29}$$

Thus, we have

$$P\left(\left|\frac{1}{n}\sum_{i=1}^{n} X_i\right| > \epsilon\right) \leq \frac{\sigma^2}{n\epsilon^2} \longrightarrow 0 \tag{30}$$

as $n \to \infty$, which proves the claim. □

Some further useful bounds are derived in the homework.

0.3.2 Central limit theorem

We can actually say quite a bit more than the weak law of large numbers dictates. If the scaling in front of the sum in the statement of the law of large numbers Proposition 0.22 is $1/\sqrt{n}$ instead of $1/n$, the resultant random variable $\frac{1}{\sqrt{n}}\sum_{i=1}^{n} X_i$ converges in distribution to a Gaussian random variable. As in Proposition 0.22, let X^n be a collection of i.i.d. random variables where each X_i is zero mean with finite variance σ^2.

Proposition 0.23 (Central limit theorem)**.** *For any* $a \in \mathbb{R}$*, we have*

$$\lim_{n\to\infty} P\left(\frac{1}{\sigma\sqrt{n}}\sum_{i=1}^{n} X_i < a\right) = \Phi(a). \tag{31}$$

In other words,

$$\frac{1}{\sigma\sqrt{n}}\sum_{i=1}^{n} X_i \xrightarrow{\text{d}} Z \tag{32}$$

where $\xrightarrow{\text{d}}$ *means convergence in distribution and* Z *is a standard Gaussian random variable.*

For a sequence of random variables $\{S_n\}_{n=1}^{\infty}$, we say that this sequence of random variables *converges in distribution* to another random variable $\bar{S}$ if

$$\lim_{n\to\infty} P(S_n < a) = P(\bar{S} < a)$$

for all $a \in \mathbb{R}$. The proof of this statement requires tools that are outside the scope of these notes, but can be found in any textbook on probability theory.

0.4 Vector norms and Cauchy-Schwarz inequality

We can naturally interpret pmf's on an alphabet with d symbols as row vectors in a d-dimensional inner-product space. Without loss of generality we take the alphabet to be $\mathcal{X} = \{1, 2, \ldots, d\} = [d]$ and define the vector $p \in \mathbb{R}^d$ by its elements $p_x = P_X(x)$ for $x \in [d]$. The inner product is denoted by $\langle \cdot, \cdot \rangle$. For two general vectors $u, v \in \mathbb{R}^d$, it evaluates to

$$\langle u, v \rangle = uv^T = \sum_{i=1}^{d} u_i v_i \,, \tag{33}$$

where v^T denotes the transpose of the vector v, and is a column vector. The Cauchy-Schwarz inequality then states that for any two vectors we have

$$|\langle u, v \rangle|^2 \leq \langle u, u \rangle \langle v, v \rangle \,. \tag{34}$$

On these vector spaces we can also define the p-norms for $p \geq 1$ as

$$\|u\|_p = \left(\sum_{x=1}^{d} |u_x|^p \right)^{\frac{1}{p}} \tag{35}$$

We will mostly encounter the 1-norm and the 2-norm, the latter being the usual Euclidian norm of the vector. The following special case of the Cauchy-Schwarz inequality will be encountered later.

Lemma 0.24. *Let $u, v \in \mathbb{R}^d$. Then,*

$$\|u \cdot v\|_1 \leq \|u\|_2\| \, \|v\|_2 \,, \tag{36}$$

where $\cdot$ denotes the element-wise product of the vectors, i.e. $(u \cdot v)_i = u_i v_i$.

Proof. Define $k \in$ using $k_i = \operatorname{sgn}^*(u_i v_i)$, where sgn^* is the modified sign function, i.e.

$$\operatorname{sgn}^*(x) = \begin{cases} -1 & \text{if } x < 0 \\ 1 & \text{if } x \geq 0 \end{cases} . \tag{37}$$

Then since $\operatorname{sgn}^*(x)^2 = 1$ for all $x \in \mathbb{R}$, the Cauchy-Schwarz inequality yields

$$\big|\langle k \cdot u, v \rangle\big| \leq \langle u, u \rangle \langle v, v \rangle = \|k \cdot u\|_2 \|v\|_2 = \|u\|_2 \|v\|_2 \,. \tag{38}$$

Moreover, we have

$$\langle k \cdot u, v \rangle = \sum_{x=1}^{d} k_i u_i v_i = \sum_{x=1}^{d} |u_i v_i| = \|u \cdot v\|_1 \,. \tag{39}$$

□

Question 0.25. *Using the above, can you show that $\|u\|_1 \leq \sqrt{d}\|u\|_2$?*

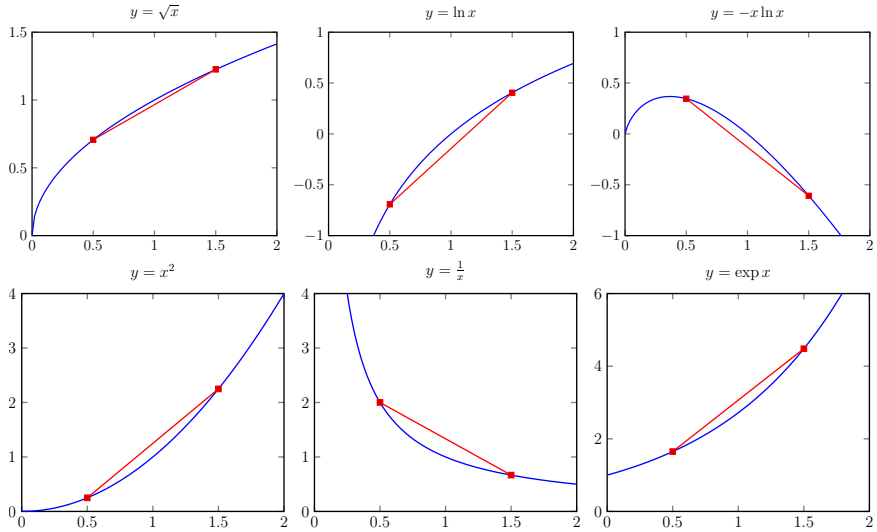


Figure 0.1: Examples of concave (upper ones) and convex (lower ones) functions. The straight line between two points of the curve is either below or above the plot of the function, which is exactly what the definition requires.

0.5 Convexity and Jensen's inequality

A function $f(x)$ is said to be *convex* on [a, b] if for all $x, y \in [a, b]$ and $\lambda \in [0, 1]$, we have

$$f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y). \tag{40}$$

If we do not mention any interval then we mean that the function is convex on its full domain, i.e. the statement $\log(x)$ is concave should be understood as $\log(x)$ is concave on $(0, \infty)$.

The function f is *strictly convex* if equality in (40) holds only if $\lambda = 0$ or 1, or $x = y$. The function f is *concave* if $-f$ is convex, and *strictly concave* if $-f$ is strictly convex.

In the homework you will show the following lemma:

Lemma 0.26. *If f is convex on $[a, b]$, then for any $a \leq x_1 < x_2 \leq x_3 < x_4 \leq b$, we have*

$$\frac{f(x_2) - f(x_1)}{x_2 - x_1} \leq \frac{f(x_4) - f(x_3)}{x_4 - x_3} \tag{41}$$

Proposition 0.27 (Jensen's inequality)**.** *If $f(x)$ is convex and X is a random variable on $\mathbb{R}$, then*

$$f(\mathbb{E}[X]) \leq \mathbb{E}[f(X)] \tag{42}$$

We only give a proof for discrete distributions here.

Proof. We give a proof by induction. Due to convexity, we have

$$p_1 f(x_1) + p_2 f(x_2) \geq f(p_1 x_1 + p_2 x_2), \tag{43}$$

which proves the statement if $|\mathcal{X}| = 2$.

Suppose the statement $\mathbb{E}[f(X)] \geq f(\mathbb{E}[X])$ is true when $|\mathcal{X}| = k - 1$. Then consider a pmf with k mass points $\{p_1, p_2, \ldots, p_k\}$. Define another pmf on $k - 1$ points given by the probabilities

$$p'_i = \frac{p_i}{1 - p_k}, \quad i = 1, \ldots, k-1\,. \tag{44}$$

We then have

$$\sum_{i=1}^{k} p_i f(x_i) = p_k f(x_k) + (1 - p_k) \sum_{i=1}^{k-1} p'_i f(x_i) \tag{45}$$

$$\geq p_k f(x_k) + (1 - p_k) f\left(\sum_{i=1}^{k-1} p'_i x_i \right) \tag{46}$$

$$\geq f\left(p_k x_k + (1 - p_k) \sum_{i=1}^{k-1} p'_i x_i \right) \tag{47}$$

$$= f\left(\sum_{i=1}^{k} p_i x_i \right) \tag{48}$$

where the first inequality is from the induction hypothesis and the second by convexity (of two points). By the definition of expectation we have $\mathbb{E}[f(X)] \geq f(\mathbb{E}[X])$. □

Often it is hard to check convexity directly. But for twice differentiable functions, this is easy.

Proposition 0.28. *Let $f : [a, b] \to \mathbb{R}$ be twice differentiable. The function f is convex if and only if $f''(x) \geq 0$ for all $x \in (a, b)$, and strictly convex if $f''(x) > 0$ for all $x \in (a, b)$.*

Proof. Assume $f''(x) > 0$ for all $x \in [a, b]$. By Taylor expansion of f around $x_0 \in (a, b)$, we have

$$f(x) = f(x_0) + f'(x_0)(x - x_0) + \frac{f''(x^*)}{2}(x - x_0)^2 \tag{49}$$

where $x^* \in [x_0, x]$. By assumption $f''(x^*) > 0$ so the quadratic term is strictly positive unless $x = x_0$, in which case it is still non-negative. Now let $x_0 = \lambda x_1 + (1 - \lambda) x_2$. Further let $x = x_1$. Then we have

$$f(x_1) \geq f(x_0) + f'(x_0)((1 - \lambda)(x_1 - x_2)). \tag{50}$$

Now let $x = x_2$. Then we have

$$f(x_2) \geq f(x_0) + f'(x_0)(\lambda(x_2 - x_1)). \tag{51}$$

Both of these inequalities are strict unless $\lambda \in \{0, \}]$ or $x_1 = x_2$. Multiplying the first inequality by λ and the second by $1 - \lambda$ and adding them up, we recover the definition of strict convexity. If we instead had assumed only $f''(x) \geq 0$ the same argument would ensure convexity (but no longer strict convexity).

For the other direction, choose $a < x_1 < x_2 < x_3 < x_4 < b$. By the property shown in Lemma 0.26,

$$\frac{f(x_2) - f(x_1)}{x_2 - x_1} \leq \frac{f(x_4) - f(x_3)}{x_4 - x_3} \tag{52}$$

Now let $x_2 \searrow x_1$ and $x_3 \nearrow x_4$. We see that $f'(x_1) \leq f'(x_4)$, and since these were arbitrary points, f' is increasing on (a, b). So $f''(x) \geq 0$ for all $x \in (a, b)$. □

0.6 Finite field arithmetic

This is a rather informal discussion, but it is sufficient for our purposes.

A finite field is a field (on which addition, subtraction, multiplication and division are defined) with a finite number of elements. Such fields are denoted by F_q where q is the number of elements in the field, or its *dimension*. For each dimension, the field is unique up to a relabelling of the elements. The idea is that such fields behave like $\mathbb{Q}$, $\mathbb{R}$ or $\mathbb{C}$, with the usual rules for addition and multiplication.

A bit more formally, we have two binary operations on F_q denoted by $+$ and $\cdot$ and the following properties (here a, b and c are any elements of F_q):

Associativity: $a + (b + c) = (a + b) + c$ and $a \cdot (b \cdot c) = (a \cdot b) \cdot c$.

Commutativity: $a + b = b + a$ and $a \cdot b = b \cdot a$.

Identities: There exist two different elements $0, 1$ such that $a + 0 = a$ and $a \cdot 1 = a$.

Additive inverse: Every a has an additive inverse, denoted $-a$, such that $a + (-a) = 0$.

Multiplicative inverse: Every $a \neq 0$ has a multiplicative inverse, denoted a^{-1}, such that $a \cdot a^{-1} = 1$.

Distributivity: $a \cdot (b + c) = (a \cdot b) + (a \cdot c)$.

Such fields exist only for particular numbers of elements, namely when $q = p^\ell$ for some prime p and $\ell \in \mathbb{N}$.

For F_q where q is prime we can always simply denote the elements of F_q by the integers $\{0, 1, \ldots, q - 1\}$ and use integer addition and multiplication modulo q as our operations.

Question 0.29. *Verify the above properties for F_2 and F_3. Can you find the inverse of 2 for a general F_q with prime q? Use that $q+1$ is even...*

This strategy fails when $q = 4$ is not a prime. The problem is that using multiplication modulo 4 we would, for example, get

$$2 \times 0 = 0 \tag{53}$$
$$2 \times 1 = 2 \tag{54}$$
$$2 \times 2 = 4 \mod 4 = 0 \tag{55}$$
$$2 \times 3 = 6 \mod 4 = 2\,, \tag{56}$$

and hence 2 does not have a multiplicative inverse.

When q is a prime power $q = p^\ell$ we can derive the arithmetic using a polynomial ring. We give the construction here; but we do not attempt to show that it actually works or that it is unique. First, we denote the elements of F_q by strings of length ℓ with elements in F_p. In particular, if the underlying prime is 2, these are simply binary strings, e.g., $F_4 = \{00, 01, 10, 11\}$. We can then interpret these elements as polynomials of degree $\ell - 1$ with coefficients in F_p. Again, for F_4 the polynomials corresponding to the four elements would be $00 \to 0$, $01 \to 1$, $10 \to x$ and $11 \to x + 1$. We can add these polynomials modulo p for each coefficient individually, so in particular for the binary case the negation of each number is just the number itself. For multiplication, we simply do this modulo an irreducible polynomial. (An irreducible polynomial is one that has no roots in F_p.) The choice of irreducible polynomial turns out not to matter — the resulting fields are equivalent up to relabelling of elements. For F_4 we can take the irreducible polynomial to be $x^2 + x + 1$.

Question 0.30. *Verify that $x^2 + x + 1$ is indeed irreducible over F_2? Is it also irreducible over F_3?*

So for the above labelings $\{00, 01, 10, 11\}$ of elements, we get

$$10 \times 00 \to x \times 0 = 0 \to 00 \implies 10 \times 00 = 00 \tag{57}$$
$$10 \times 01 \to x \times 1 = x \to 10 \implies 10 \times 01 = 10 \tag{58}$$
$$10 \times 10 \to x \times x = x^2 \mod x^2 + x + 1 = x + 1 \to 11 \implies 10 \times 10 = 11 \tag{59}$$
$$10 \times 11 \to x \times (x+1) = x^2 + x \mod x^2 + x + 1 = 1 \to 01 \implies 10 \times 11 = 01\,. \tag{60}$$

Hence, 10 and 11 are multiplicative inverses of each other. The full addition and multiplication tables can then be written down as follows:

+	00	01	10	11
00	00	01	10	11
01	01	00	11	10
10	10	11	00	01
11	11	10	01	00

×	00	01	10	11
00	00	00	00	00
01	00	01	10	11
10	00	10	11	01
11	00	11	01	10

Similar constructions can be done for every prime power, and, quite importantly for practical applications, all of this arithmetic can be implemented highly efficiently in computer programs.

Chapter 1

Information measures

[Week 2]

Intended learning outcomes:

- You can compute the entropy and conditional entropy for any discrete random variable and understand the basic properties of these two quantities, e.g. you can apply the chain rule or sub-additivity.
- You can compute mutual information and now how it relates to entropy and conditional entropy. You can apply the data-processing inequality for mutual information.
- You can compute the relative entropy and understand how entropy and mutual information can be expressed in terms of the relative entropy.

Book reference: Chapter 2 in Cover & Thomas [1], but we are not following it too closely.

1.1 Surprisal and entropy

It is not immediately clear how to model our intuitive notion of "information" in a mathematical language. In this chapter we take a somewhat axiomatic approach to information measures, i.e. we try to build them up from our intuitive understanding of what entropy and information "should" be. But we will only really be able to justify the choices we make here once we start analysing practical problems in information theory, and see that the quantities we derive here pop up again and again.

1.1.1 Surprisal

It turns out to be fruitful to start not by finding an expression for the information contained in a random variable, but rather the lack of information, or uncertainty inherent in a random experiment. Let us consider a discrete random variable X taking values in $\mathcal{X}$ following the pmf $P_X(x) = p_x$. How surprised are we to see a particular outcome $x \in \mathcal{X}$ of this random

experiment? Clearly this depends on the probability p_x and not the value of x itself. In fact, we do not even need to know what $\mathcal{X}$ really is. On the one hand, if $p_x = 1$ we are not surprised at all since we already knew that we would see x. On the other hand, the smaller p_x is the more surprised we are to see this particular outcome. If $p_x = 0$ we will never see x, so our surprise when seeing it anyway would be literally off the scale. Furthermore — and this turns out to be a very convenient choice — if we do a random experiment twice independently and both times observe x, we say that we will be twice as surprised as if we had seen x once in a single random experiment.

The above notions can be formalised, and that is essentially what Shannon did when he introduce the notion of *surprisal*. Let us denote the surprisal of x as $s(p_x)$. We want this function to satisfy the following three conditions:

1. **Monotonicity:** $s(p_x) = 0$ if $p_x = 1$ and $s(p_x)$ increases monotonically as p_x decreases.
2. **Additivity:** The surprisal of seeing a pair of outcomes of independent random experiments is simply the sum of the individual surprisals, i.e. $s(p_x p_y) = s(p_x) + s(p_y)$.
3. **Normalisation:** $s(\frac{1}{2}) = 1$

Question 1.1. *We do not really need the condition $s(p_x) = 0$ if $p_x = 1$ under Point 1 as it follows directly from additivity. Can you see how?*

It turns out that the only function that satisfies these three properties is the logarithm. To show this one uses a result by Erdös that characterises additive functions, but that is beyond the scope here. We therefore pick

$$s(p_x) = \log \frac{1}{p_x} \,. \tag{1.1}$$

where the logarithm is taken to base 2 (as everywhere in these notes) so that the normalisation requirement is satisfied.

We can see the surprisal as another random variable, say S, that takes the value $s(p_x) = \log \frac{1}{p_x}$ with probability p_x. Since $S = S(X)$ is a function of X we usually simply write this new random variable as

$$S(X) = \log \frac{1}{P_X(X)} \,. \tag{1.2}$$

1.1.2 Entropy

Entropy measures how much we can learn by looking at the outcome of a random experiments, or, in other words, how much uncertainty there is about the outcome. It is simply the expected surprisal of X.

Definition 1.2. *Given a discrete random variable X, the* entropy *of X is defined as*

$$H(X) := \mathbb{E}[S(X)] = \mathbb{E}\left[\log \frac{1}{P_X(X)}\right] = \sum_{x \in \mathcal{X}} p_x \log \frac{1}{p_x} \tag{1.3}$$

Here and throughout we use the convention that $0 \log 0 = 0$. This is reasonable since $\lim_{\epsilon \to 0} \epsilon \log \epsilon = 0$, and thus we simply continuously extend the function to the point 0.

Question 1.3. *Can you verify $\epsilon \log \epsilon \to 0$ as $\epsilon \to 0$?*

Note again that the entropy X is really only a function of the pmf of X, and in particular independent of the alphabet $\mathcal{X}$, in contrast to potential alternative uncertainty measures like the variance of X.

Sometimes we are interested in more than just the expected surprisal. The minimum surprisal, or min-entropy, for example, has applications in cryptography (see Chapter 3) and the variance of $S(X)$ has itself operational meaning in many information-theoretic problems when we go beyond first order asymptotics.

Question 1.4. *Can you find an expression for $\mathrm{Var}[S(X)]$ in terms of the probabilities p_x?*

Now let us explore the entropy a bit. First we want to show the following basic property.

Proposition 1.5. *Let X be a discrete random variable taking values in $\mathcal{X}$. We have*

$$H(X) \geq 0, \tag{1.4}$$

with equality if and only if X is deterministic.

Proof. Since $p_x \leq 1$, we have $\log \frac{1}{p_x} \geq 0$ for every $x \in \mathcal{X}$, so the expectation of this quantity over x must be non-negative too. In fact, $\log \frac{1}{p_x}$ equals 0 if and only if $p_x = 1$ and hence $H(X) = 0$ only if there exists an $x \in \mathcal{X}$ for which $p_x = 1$, which is the hallmark of a deterministic rv. □

The entropy is a strictly concave function of the probability mass function P_X. To see this, we first verify that $f(t) = t \log \frac{1}{t} = -t \log t$ is concave on $(0, 1)$ by taking its second derivative:

$$f'(t) = -\log t - \log e, \qquad f''(t) = -\frac{\log e}{t}. \tag{1.5}$$

Since the latter is always negative for $t \in (0, 1)$, the function is indeed strictly concave according to Lemma 0.28. Now since the entropy is simply the sum $\sum_{x \in \mathcal{X}} f(p_x)$ it is indeed a concave function of the pmf. This simple property, together with Jensen's inequality, has profound implications. The first one is that the entropy has a unique maximum. Intuitively we would want that entropy is maximal when uncertainty about the outcome is greatest, namely when the rv is uniformly distributed. And this is indeed the case.

Proposition 1.6. *Let X be a discrete random variable taking values in $\mathcal{X}$. We have*

$$H(X) \leq \log |\mathcal{X}|, \tag{1.6}$$

with equality if and only if X is uniformly distributed.

The general case will be covered in the homework but here we give a proof for the case when the set $\mathcal{X}$ is a bit, i.e. when the random variable is binary.

Proof for $\mathcal{X} = \{0, 1\}$. It is easy to verify by a simple computation that $H(X) = 1$ for a uniformly distributed random variable, so the difficulty is only in showing that this is the maximum and only achieved for the uniform distribution.

Let now $\{p, 1-p\}$ for $p \in [0, 1]$ be a general pmf for the random variable X. We use the function $f(t) = -t \log t$ to simplify notation. Then we can write

$$\begin{aligned} H(X) &= f(p) + f(1-p) && (1.7)\\ &= \frac{1}{2}\left(f(p) + f(1-p)\right) + \frac{1}{2}\left(f(p) + f(1-p)\right) && (1.8)\\ &\leq f\left(\frac{1}{2}p + \frac{1}{2}(1-p)\right) + f\left(\frac{1}{2}p + \frac{1}{2}(1-p)\right) && (1.9)\\ &= f\left(\frac{1}{2}\right) + f\left(\frac{1}{2}\right) && (1.10)\\ &= 1\,. && (1.11) \end{aligned}$$

The inequality is due to Jensen's inequality and the strict concavity of f, and equality holds only if $p = 1 - p = \frac{1}{2}$, i.e. when the random variable X follows the uniform distribution. □

Concavity in fact has even stronger consequences, and we will show a few additional properties of entropy later on using it.

1.2 Conditional entropy, and mutual information

1.2.1 Joint entropy

For two discrete random variables X and Y with joint pmf $P_{XY}(x, y) = p_{xy}$ we can simply consider (X, Y) as one single random variable and use the same construction to define the surprisal of a tuple (X, Y) as $S(X, Y) = -\log P_{XY}(X, Y)$. Its expectation is the *joint entropy*, $H(XY)$, given by

$$H(XY) := \mathbb{E}[S(X, Y)] = \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p_{xy} \log \frac{1}{p_{xy}} \tag{1.12}$$

The first thing to note is that — if X and Y are independent — then $p_{xy} = p_x \cdot p_y$ and thus the expression simplifies to

$$H(XY) = \sum_{x\in\mathcal{X}} \sum_{y\in\mathcal{Y}} p_{xy} \log \frac{1}{p_x p_y} \tag{1.13}$$

$$= \sum_{x\in\mathcal{X}} \sum_{y\in\mathcal{Y}} p_{xy} \log \frac{1}{p_x} + \log \frac{1}{p_y} \tag{1.14}$$

$$= \sum_{x\in\mathcal{X}} p_x \log \frac{1}{p_x} + \sum_{y\in\mathcal{Y}} p_y \log \frac{1}{p_y} \tag{1.15}$$

$$= H(X) + H(Y)\,. \tag{1.16}$$

This is not true in general though if the two random variables are correlated.

Question 1.7. *Find an example for which $H(XY) = H(X) = H(Y) = 1$.*

1.2.2 Conditional entropy

So why do these entropies not just add up? Fundamentally, this is because once we learn X we might not be so surprised seeing some particular outcomes of the random variable Y anymore. In fact, in the most extreme case, we have $Y = f(X)$ for some function f; hence, once we know that X takes on the value x, we can immediately deduce that Y will take on the value $f(x)$ with probability one, and thus there is no surprisal anymore! We model this "conditional suprisal" using the conditional pmfs, $P_{Y|X}(y|x) = p_{y|x}$, which leads us to conditional entropy.

Definition 1.8. *The* conditional entropy *of Y given X is defined as*

$$H(Y|X) = \mathbb{E}\left[\log \frac{1}{P_{Y|X}(Y|X)}\right] = \sum_{x\in\mathcal{X}} \sum_{y\in\mathcal{Y}} p_{xy} \log \frac{1}{p_{y|x}}\,. \tag{1.17}$$

This can be interpreted as the expectation of the entropy of Y over all outcomes X. We sometimes use the notation $H(Y|X = x) = H(Y_x)$ to denote the entropy of the random variable Y_x that follows the pmf $\{p_{y|x}\}_{y\in\mathcal{Y}}$, i.e., the pmf of Y when we already know that $X = x$. Using this and the expression in (1.17) we can write the conditional entropy as

$$H(Y|X) = \sum_{x\in\mathcal{X}} \sum_{y\in\mathcal{Y}} p_{xy} \log \frac{1}{p_{y|x}} \tag{1.18}$$

$$= \sum_{x\in\mathcal{X}} p_x \sum_{y\in\mathcal{Y}} p_{y|x} \log \frac{1}{p_{y|x}} \tag{1.19}$$

$$= \sum_{x} p_x\, H(Y|X = x)\,. \tag{1.20}$$

The last line which expresses the conditional entropy in terms of an average of (unconditional) entropies is particularly useful since it allows us to immediately conclude that the conditional entropy is also bounded from below and above, like the entropy. We thus have

$$0 \leq H(Y|X) \leq \log|\mathcal{Y}|\,. \tag{1.21}$$

Moreover, our definition of conditional entropy also allows us to establish a *chain rule* for the conditional entropy, which sometimes is in fact used as the definition of conditional entropy itself. This rule is very useful because it allows us to write the joint entropy as a sum of its parts, even if the two random variables are not independent.

Proposition 1.9. *We have* $H(XY) = H(X) + H(Y|X)$.

Proof. We take advantage of $p_{xy} = p_x p_{y|x}$ to write

$$\begin{aligned} H(XY) &= \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p_{xy}\log\frac{1}{p_{xy}} && (1.22)\\ &= \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p_{xy}\log\frac{1}{p_x} + \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p_{xy}\log\frac{1}{p_{y|x}} && (1.23)\\ &= \sum_{x\in\mathcal{X}} p_x\log\frac{1}{p_x} + \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p_{xy}\log\frac{1}{p_{y|x}} && (1.24)\\ &= H(X) + H(Y|X)\,. && (1.25) \end{aligned}$$

□

Question 1.10. *Show that* $H(Y|X) = H(Y)$ *for independent random variables. Using the chain rule, find a different proof that* $H(XY) = H(X) + H(Y)$ *in this case.*

Now we have put everything in place to show our first entropic inequality, which relates the entropy of two random variables with their joint entropy. This result shows the *subadditivity* of entropy.

Proposition 1.11. *Let* X *and* Y *be two discrete random variables. Then*

$$H(XY) \leq H(X) + H(Y) \qquad \textit{or, equivalently,} \qquad H(X|Y) \leq H(X)\,. \tag{1.26}$$

Equality holds in either statement only if X *and* Y *are independent.*

Proof. The equivalence of the two relations follows directly from the chain rule, we thus only need to show the second statement.

We start with Eq. (1.20), which states that

$$\begin{aligned} H(Y|X) &= \sum_x p_x H(Y|X=x) && (1.27)\\ &= \sum_x p_x \sum_y p_{y|x}\log\frac{1}{p_{y|x}} && (1.28)\\ &= \mathbb{E}\left[\sum_y p_{y|X}\log\frac{1}{p_{y|X}}\right] && (1.29) \end{aligned}$$

Note that sum inside the expectation is simply another expectation, as in the definition of entropy — but since we only want to apply Jensen's inequality on the outer expectation we spell this one out explicitly. Moreover, by definition of the conditional pmf we have $\mathbb{E}[p_{y|X}] = \sum_x p_x p_{y|x} = \sum_x p_{xy} = p_y$. Hence, using concavity of the entropy as a function of the pmf and Jensen's inequality for the outer expectation, we find

$$H(Y|X) = \mathbb{E}\left[\sum_y p_{y|X} \log \frac{1}{p_{y|X}}\right] \tag{1.30}$$

$$\leq \sum_y \left(\mathbb{E}[p_{y|X}]\right) \log \frac{1}{\mathbb{E}[p_{y|X}]} \tag{1.31}$$

$$= \sum_y p_y \log \frac{1}{p_y} \tag{1.32}$$

$$= H(Y)\,. \tag{1.33}$$

Equality in Jensen's inequality only holds if either X is deterministic or if $p_{y|x} = p_y$ for all x and y, but this only holds if X and Y are in fact independent. □

The second relation in Eq. (1.26) can be strengthened by considering three random variables X, Y and Z. In that case, we have

$$H(X|YZ) \leq H(X|Z)\,. \tag{1.34}$$

This is sometimes referred to as *strong sub-additivity*. The proof follows from (regular) sub-additivity, applied to the entropies $H(X|Y, Z = z)$ and $H(X|Z = z)$, and averaging the resulting inequalities.

Question 1.12. *Can you construct a formal proof out of the above sketch?*

1.2.3 Mutual information

We have already established that $H(XY) \neq H(X) + H(Y)$ in general, and hence also $H(Y|X) \neq H(Y)$ by the chain rule. The difference between these two quantities clearly tells us something about how much the uncertainty about Y changes when we learn X, or in other words, about how much information X contains about Y. This leads us to the definition of mutual information,

Definition 1.13. *The* mutual information *between X and Y is defined as*

$$I(X:Y) := H(Y) - H(Y|X) \tag{1.35}$$

It is not evident immediately from the way we defined it here but this expression is symmetric between X and Y. Namely, using the chain rule for conditional entropy (recall Proposition 1.9) twice, we can write

$$I(X:Y) = H(Y) - H(Y|X) = H(Y) + H(X) - H(XY) = H(X) - H(X|Y)\,. \tag{1.36}$$

The mutual information is thus a symmetric measure of the correlation between the two random variables.

Using these various equivalent expressions it is then easy to derive some bounds on the mutual information. First, sub-additivity of the entropy directly implies that $I(X:Y) \geq 0$, so the mutual information is non-negative, and it vanishes only if the two random variables are independent (a consequence of Proposition 1.11). This is consistent with our intuitive notion of information — we cannot know less than nothing after all! We also cannot know more than everything, i.e. the mutual information can never exceed the minimal entropy of its constituent parts.

Question 1.14. *Using the bounds on entropies established in the previous sections, show that* $I(X:Y) \leq \log \min\{|\mathcal{X}|, |\mathcal{Y}|\}$. *Give an example that saturates the bound.*

If we have three random variables X, Y and Z we can ask for the mutual information between X and Y conditioned on knowing Z, the *conditional mutual information*. It is defined as

$$I(X:Y|Z) := \sum_z P_Z(z) I(X:Y|Z=z)\,. \tag{1.37}$$

Various equivalent expressions can then be readily derived, e.g.,

$$I(X:Y|Z) = H(Y|Z) - H(Y|XZ) = H(X|Z) - H(X|YZ)\,. \tag{1.38}$$

Moreover, the *chain rule* for the mutual information states that

$$I(X:YZ) = I(X:Y) + I(X:Z|Y)\,, \tag{1.39}$$

which can be verified by a close inspection of the definition of both conditional and unconditional mutual information.

Consider now the special case where $X - Z - Y$ form a Markov chain. In this case $P_{X|YZ} = P_{X|Z}$ and thus $H(X|YZ) = H(X|Z)$. As a consequence, the conditional mutual information $I(X:Y|Z)$ as written in (1.38) vanishes.

One of the most intriguing properties of the mutual information is the *data-processing inequality* for mutual information. It states that the mutual information can never increase when we apply an operation that only acts on one of the parts. Intuitively this tells us that by manipulating one of the random variables without looking at the other we cannot increase the correlations between the pair.

We can formalise this using the notion of Markov chains.

Proposition 1.15. *Let* $X - Y - Z$ *form a Markov chain. Then,* $I(X:Y) \geq I(X:Z)$.

Proof. Since $I(X:Z|Y) = 0$, the chain rule for mutual information implies that $I(X:Y) = I(X:YZ)$. It thus remains to show that

$$I(X:Z) \leq I(X:YZ)\,. \tag{1.40}$$

But, since $I(X:Z) = H(X) - H(X|Z)$ and $I(X:YZ) = H(X) - H(X|YZ)$, the relation in Eq. (1.40) is equivalent to the condtion $H(X|Z) \geq H(X|YZ)$, which is in turn ensured by the strong sub-additivity of entropy. □

Question 1.16. *Can you also show that* $I(Y:Z) \geq I(X:Z)$ *under the same assumption?*

1.3 Relative entropy

The relative entropy appears when we want to compare two different probability distributions. We define it here only for discrete random variables (or rather the respective pmfs), but this can be readily generalised to other probability measures.

Definition 1.17. *Let P and Q be two pmfs on an alphabet $\mathcal{X}$. The* relative entropy *of P with regards to Q is defined as*

$$D(P\|Q) := \sum_{\substack{x\in\mathcal{X}\\ P(x)>0}} P(x)\log\frac{P(x)}{Q(x)}\,. \tag{1.41}$$

if $P(x) > 0 \implies Q(x) > 0$ for all $x \in \mathcal{X}$, and $D(P\|Q) = +\infty$ otherwise.

In the following, instead of restricting the sum, we will use the convention that $0\log\frac{0}{0} = 0$.

We can alternatively see the relative entropy as the expectation value of the *log-likelihood ratio*, namely we can write

$$D(P\|Q) = \mathbb{E}\left[Z(X)\right], \qquad \text{where} \qquad Z(X) = \log\frac{P(X)}{Q(X)} \tag{1.42}$$

and X is distributed according to P. The random variable $Z(X)$ is called the log-likelihood ratio. It takes on the role of the surprisal in the definition of entropy. We will explore this random variable and its distribution much more when we discuss the information spectrum method and hypothesis testing later on.

Just by manipulating the definition, we are able to show the following equivalences.

Proposition 1.18. *Let X and Y be random variables on alphabets $\mathcal{X}$ and $\mathcal{Y}$. Moreover, let U be a uniform random variable on $\mathcal{X}$. Then the following relations are true:*

$$H(X) = \log|\mathcal{X}| - D(P_X\|U_X) \tag{1.43}$$
$$H(X|Y) = \log|\mathcal{X}| - D(P_{XY}\|U_X\times P_Y) \tag{1.44}$$
$$I(X:Y) = D(P_{XY}\|P_X\times P_Y)\,. \tag{1.45}$$

You will prove these equivalences in the homework. They turn out to be very useful because they essentially tell us that once we established properties of the relative entropy this has immediate consequences also for the derived quantities

We will need two important properties of the relative entropy. The first proposition establishes that the relative entropy is always positive.

Proposition 1.19. *For any two pmfs P and Q, we have $D(P\|Q) \geq 0$ with equality if and only if $P = Q$.*

Proof. We can assume without loss of generality that the quantity is finite, as otherwise the statement is trivially true. We first note that $x \mapsto -\log x$ is strictly convex. Hence,

$$D(P\|Q) = \sum_{x:P(x)>0} P(x) \log \frac{P(x)}{Q(x)} \tag{1.46}$$

$$= \sum_{x:P(x)>0} P(x) \left(-\log \frac{Q(x)}{P(x)} \right) \tag{1.47}$$

$$\geq -\log \left(\sum_{x:P(x)>0} P(x) \frac{Q(x)}{P(x)} \right) \tag{1.48}$$

$$= -\log \left(\sum_{x:P(x)>0} Q(x) \right) \tag{1.49}$$

$$\geq -\log \left(\sum_{x} Q(x) \right) = 0\,. \tag{1.50}$$

Equality in the second inequality only holds if P and Q have the same support. Moreover, equality in the first inequality holds if $\frac{Q(x)}{P(x)}$ is independent of x for any x in the support of P. These two statements are both true only if $P(x) = Q(x)$ for all $x \in \mathcal{X}$, and thus $P = Q$. □

An immediate corollary of Propositions 1.18 and 1.19 is that $I(X : Y)$ is positive and zero only if X and Y are independent.

Question 1.20. *Can you see why?*

Finally, there is one property of the relative entropy that implies all other properties of both entropy and mutual information. It states that applying a noisy operation, i.e. a stochastic map or channel, on both arguments of the relative entropy will never increase the relative entropy. Together with the positivity of relative entropy this justifies that we think of it as a measure of similarity or distinguishability. If the relative entropy is small the two pmfs are similar and hard to distinguish by observing the outcomes of a random experiment. Observing the outcomes after further noise has been applied should make distinguishing them even harder, and that is exactly what the *data-processing inequality* for relative entropy tells us.

Proposition 1.21. *Let P_X and Q_X be two pmfs on an alphabet $\mathcal{X}$ (the input distributions), and let $P_{Y|X}$ be a conditional pmf (the channel). Define the marginals (the output distributiions)*

$$P_Y(y) = \sum_{x\in\mathcal{X}} P_{Y|X}(y|x)P_X(x) \qquad \text{and} \qquad Q_Y(y) = \sum_{x\in\mathcal{X}} P_{Y|X}(y|x)Q_X(x)\,. \tag{1.51}$$

Then, the data-processing inequality *(DPI) states that*

$$D(P_X\|Q_X) \geq D(P_Y\|Q_Y)\,. \tag{1.52}$$

Proof. Consider now the joint distributions $P_{XY}(x,y) = P_{Y|X}(y|x)P_X(x)$ and $Q_{XY}(x,y) = P_{Y|X}(y|x)Q_X(x)$, using the usual shorthand notation for conditional and marginal distributions. We first show that

$$D(P_{XY}\|Q_{XY}) - D(P_Y\|Q_Y) = \left(\sum_{x,y} p_{xy}\log\frac{p_{xy}}{q_{xy}}\right) - \left(\sum_y p_y \log\frac{p_y}{q_y}\right) \tag{1.53}$$

$$= \sum_{x,y} p_{xy}\left(\log\frac{p_{xy}}{q_{xy}} - \log\frac{p_y}{q_y}\right) \tag{1.54}$$

$$= \sum_y p_y \sum_x p_{x|y}\log\frac{p_{x|y}}{q_{x|y}} \tag{1.55}$$

$$= \sum_y p_y\, D(P_{X|Y=y}\|Q_{X|Y=y}) \geq 0\,, \tag{1.56}$$

where we have used the positivity of relative entropy in the last step. Similarly, we have

$$D(P_{XY}\|Q_{XY}) - D(P_X\|Q_X) = \sum_x p_x\, D(P_{Y|X=x}\|Q_{Y|X=x}) = 0 \tag{1.57}$$

since $Q_{Y|X} = P_{Y|X}$ by construction of the joint distribution. Combining Eqs. (1.53)–(1.56) and (1.57) yields the desired inequality. □

It turns out that all the properties of entropy, conditional entropy and mutual information we discussed previously can be derived form the DPI. As an example we give here a strengthening of the strong sub-additivity, which we call the data-processing inequality for conditional entropy. It intuitively states that any processing of the side information can at most increase the conditional entropy.

Corollary 1.22. *Let P_{XY} be a joint pmf and $P_{Z|Y}$ a conditinal pmf. Define now the pmf*

$$P_{XZ}(x,z) = \sum_y P_{XY}(x,y)P_{Z|Y}(z|y)\,. \tag{1.58}$$

Then, we have $H(X|Y) \leq H(X|Z)$.

Proof. Let us express the inequality in terms of relative entropies using Proposition 1.18. This reads

$$\log|\mathcal{X}| - D(P_{XY}\|\, U_X \times P_Y) \leq \log|\mathcal{X}| - D(P_{XZ}\|U_X \times P_Z)\,. \tag{1.59}$$

or simply $D(P_{XY}\|\, U_X \times P_Y) \geq D(P_{XZ}\|U_X \times P_Z)$. But this is imply the DPI applied to the channel $P_{Z|Y}$ that happens to leave X untouched. □

Chapter 2

Source coding

[Week 3–5]

Intended learning outcomes:

- You can determine if a code is instantaneous and if the codeword lengths are optimal using the McMillan–Kraft inequality.
- You can evaluate the quality of a variable-length code by comparing it to the fundamental limits.
- You can construct a Huffman code for any discrete source, and understand the algorithm and its properties.
- You understand the mathematical model used to study block codes asymptotically, and can compute the code rate.
- You understand Fano's inequality and typical sets and how they can be used to derive the fundamental limits of compression.

Book reference: Chapter 5 in Cover & Thomas [1].

2.1 Problem setup and definitions

In this chapter we are concerned with removing redundancy. In the first section we will introduce the formal mathematical model we use to investigate source coding, or compression.

2.1.1 Data source

We are given data as a sequence of symbols, for example these could be numbers, letters, colours of pixels, etc., and we would like to store that data in a (preferably short) sequence of bits. (We could generalise to larger alphabets, but conceptually nothing changes so we restrict ourselves to bits here to simplify presentation.) We start by formally defining what we mean by a *data source*, or simply *source* in the remainder of this chapter.

Definition 2.1. *A* data source *is an infinite sequence of random variables*

$$\boldsymbol{X} = X_1, X_2, \ldots, X_k, \ldots . \tag{2.1}$$

- *A source is called* discrete *if the random variables are discrete, i.e. if the source outputs in each iteration $i \in \mathbb{N}$ values from a finite set $\mathcal{X}$.*
- *A source is furthermore called* memoryless *if the X_i are independent and identically distributed (i.i.d.) according to the same pmf P_X, i.e., if we have $P_{X_1X_2\ldots} = P_{X_1} \times P_{X_2} \times \ldots$ with $P_{X_1} = P_{X_2} = \ldots = P_X$.*

Memoryless here refers to the fact that the distribution of X_i does not depend on the value of X_{i-1} or any other symbol in the sequence, or, formally, $P_{X_i|X_1X_2\ldots X_{i-1}} = P_{X_i} = P_X$. We will not consider sources that output continuous values in this module.

Question 2.2. *Can you see why we cannot expect to store and perfectly recover a continuous variable using finite (digital) memory?*

For most of our theoretical analysis, we will consider a *discrete memoryless source (DMS)*. An example of such a source is the sequence of face values one gets by throwing the same (fair or unfair) die repeatedly. Generally, the assumption that a source is memoryless is simplifying the analysis but in fact most sources do not satisfy this exactly. For example, think of a book (written in English) as a sequence of letters and a source reproducing them one by one. If $X_{i-1} =$ 'q', then $X_i =$ 'u' with much higher probability than the frequency of 'u' in English text would otherwise suggest. Hence, this source is far from memoryless and the corresponding distribution of the random variable is not i.i.d.. Nonetheless, understanding the simple case of discrete memoryless sources properly will allow us to get an intuition for more loosely structured sources as well. Various more complicated models of sources have been analysed in the literature.

2.1.2 Source codes

Next we introduce the notion of *source codes*, or simply *codes* in the remainder of this chapter.

Definition 2.3. *A* source code *is a map e and that maps outputs of the source $x \in \mathcal{X}$ to bit strings of variable length, $C(x) \in \{0,1\}^*$. We denote by $\ell(x)$ the length of the* codeword *$C(x)$.*

- *A code is called a* fixed-length *code if $\ell(x) = \ell$ is constant, otherwise it is called a* variable-length *code.*
- *A code is called* non-singular *if C is injective, i.e. if every $x \in \mathcal{X}$ is mapped to a unique bit string.*
- *A code is called a* prefix code *if for any pair $x, x' \in \mathcal{X}$ with $x \neq x'$, the codeword $C(x)$ is not a prefix of the codeword $C(x')$.*[1]

[1] We say that a bit string is a prefix of another bit string if the latter starts with the former, e.g. '01' is a prefix to '0100'.

- *A code is* uniquely decodable *if there exists a decoder that, for any $n \in \mathbb{N}$ and any sequence $x^n \in \mathcal{X}^n$, can uniquely recover x^n from the bit string $C(x_1)C(x_2)\dots C(x_n)$.*
- *A code is* instantaneous *if it is uniquely decodable and if the decoder can deduce the k-th symbol x_k as soon it has seen the bit string $C(x_1)C(x_2)\dots C(x_k)C(x_{k+1})\dots$ up to $C(x_k)$, even if there is no guarantee that the string is complete.*
- *A* block code *of length n takes a sequence of n source outputs $x^n \in \mathcal{X}^n$ as input and outputs a binary string $C(x^n)$.*

Let us note that the codes we consider here are not as general as they could be. In fact, we could also consider codes that take a variable length sequence of input symbols to a codeword (of either fixed or variable length). Such codes are in fact often used in practical applications.

Let us now discuss some fo the interrelations between all these properties. Clearly, any prefix code is non-singular by definition. First, we observe that not every non-singular code is uniquely decodable. Consider a code on $\mathcal{X} = \{0, 1, 2, 3\}$ that yields the binary representation

$$C(0) = 0, \quad C(1) = 1, \quad C(2) = 10, \quad C(3) = 11\,. \tag{2.2}$$

This code is non-singular but not a prefix code. The codeword string 110 could either be produced by the source sequence $(3, 0)$, by $(1, 2)$ or even by $(1, 1, 0)$, so there is no way for a decoder to distinguish between these three cases.

Proposition 2.4. *A code is instantaneous if and only if it is a prefix code*

Proof. We first show that a prefix code is instantaneous by constructing a decoder. The decoder will read the sequence $C(x_1)C(x_2)\dots$ bit by bit. Once $C(x_1)$ is fully read we can immediately deduce that the first source symbol was x_1 since $C(x_1)$ cannot be a prefix for a longer codeword. Similarly, with this rule in mind, since there is no other codeword that is a prefix to $C(x_1)$ we can be assured that this is indeed the first symbol we will decode. The same procedure continues for the remainder of the string with $C(x_2)C(x_3)\dots$. We know where every codeword ends and can decode them individually and instantaneously.

To verify the other direction, simply note that if a decoder can decide instantly once it has seen the codeword $C(x)$ this implies that $C(x)$ cannot be a prefix to any other codeword $C(x')$. Since this is true for any $x \neq x'$, the code must be a prefix code. □

Thus, any prefix code is uniquely decodable. However, the converse is not true in general, i.e. not every uniquely decodable code is a prefix code. Consider as an example the code

$$C(\text{‘a’}) = 1, \quad C(\text{‘b’}) = 10, \quad C(\text{‘c’}) = 00\,. \tag{2.3}$$

After seeing 10 we cannot decide if the first symbol was ‘a’ or ‘b’ even though we have seen the full codeword of the first symbol, hence this code is not instantaneous. However, once we have seen a full sequence of codewords, for example 100, we can decode uniquely by looking at the parity of the numbers of 0’s between two 1’s.

Question 2.5. *Can you come up with a formal decoder for this code?*

Codes can be conveniently represented by binary trees. Binary trees are connected graphs without cycles (trees) where each node (except the root) has exactly one parent and either zero (in which case it is called a leaf) or two children. The two branches emanating from the root and each node are assigned values '0' or '1' and codewords are composed by following a path from the root to a node.

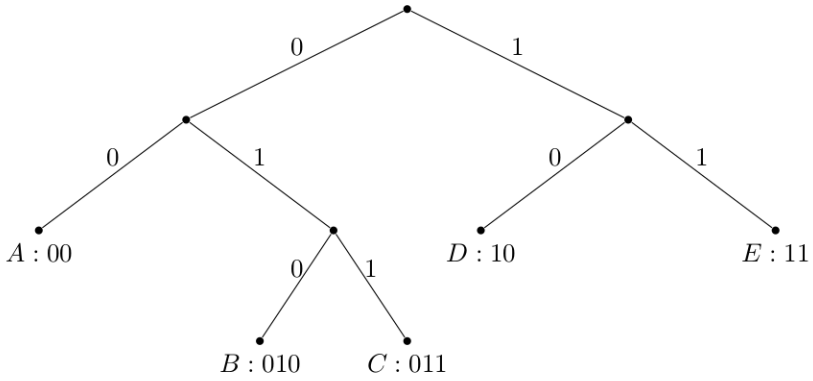


Figure 2.1: Example of a code tree.

The codeword length is equivalent to the depth (i.e. the distance from the root) of the node in the three. For a fixed-length code all the codewords are at the same depth (or level) of the tree. A code is a prefix code if and only if all the codewords are on leaves of the tree.

Question 2.6. *Can you see why this is the case?*

The next two sections will be devoted to variable-length and (fixed-length) block codes, respectively.

2.2 Variable-length codes

2.2.1 Optimal codeword lengths

The Kraft inequality gives a lower bound on the lengths of codewords in any instantaneous code. It is the first fundamental limit we will establish, it shows us that no code with shorter codeword lengths can exist, and thus if a code achieves equality in (2.4) we know it is optimal in this regard. We present a slightly more general result, MicMillan–Kraft inequality, which applies for any uniquely decodable code (and not just instantaneous codes).

Proposition 2.7 (McMillan–Kraft inequality)**.** *Any uniquely decodable code must satisfy the inequality*

$$\sum_{x\in\mathcal{X}} 2^{-\ell(x)} \leq 1\,. \tag{2.4}$$

Conversely, given a set of codewords lengths satisfying Eq. (2.4), it is possible to construct a prefix code with these lengths.

We show the inequality only for instantaneous codes, and the general proof for uniquely decodable (not necessarily instantaneous) codes will be covered in the homework.

Proof. We use the one-to-one correspondence of instantaneous codes with binary trees for which the codewords are leaves. For any tree we may assign to every node a value 2^{-d} where d is the depth in the tree. The root thus gets the value 1. To show the Kraft inequality we simply need to show that the sum of all the leaf values in a tree cannot exceed 1. To see that this is correct, simply note that by our construction any parent node's value is simply the sum of its children's values, so as we build up the binary tree from the root the sum of the values on all leaves is always exactly 1.

Conversely, given a set of codeword lengths satisfying the inequality, we can always create a binary tree and populate its leaves with codewords. If the inequality is strict there will be unused leaves in the tree. □

2.2.2 Optimal expected codeword length

Finding codewords with short lengths is however only half of the problem — we also need to assign those codewords to elements of $\mathcal{X}$. And we want to do this in such a way as to minimise the expected length of the codeword. That is, for a code $C(x)$ with codeword lengths $\ell(x)$, we define the *expected codeword length* as

$$\bar{\ell}(C) := \sum_x P_X(x)\ell(x) = \mathbb{E}[\ell(X)] \tag{2.5}$$

Using the McMillan–Kraft inequality from Proposition 2.7, we can lower bound $\bar{\ell}(C)$ with the entropy $H(X)$ for any uniquely decodable code.

Proposition 2.8. *For any uniquely decodable code C for a discrete source X with distribution P_X, we have*

$$\bar{\ell}(C) \geq H(X)\,. \tag{2.6}$$

Moreover, the equality is saturated if only if the codeword lengths saturate the McMillan-Kraft inequality and $P_X(x) = 2^{-\ell_x}$ for some set of numbers $\ell_x \in \mathbb{N}$.

Proof. We evaluate

$$\begin{aligned}
\bar{\ell}(C) - H(X) &= \mathbb{E}\left[\ell(X) - \log \frac{1}{P_X(X)}\right] && (2.7)\\
&= \mathbb{E}\left[\log \frac{P_X(X)}{2^{-\ell(X)}}\right] && (2.8)\\
&\geq \mathbb{E}\left[\log \frac{t \cdot P_X(X)}{2^{-\ell(X)}}\right] && (2.9)\\
&= D(P_X \| Q_X) && (2.10)\\
&\geq 0, && (2.11)
\end{aligned}$$

where we introduced the constant $t = \sum_x 2^{-\ell(x)} \leq 1$ (by the McMillan–Kraft inequality) and the pmf $Q(x) = \frac{1}{t} \cdot 2^{-\ell(x)}$. The final inequality is simply due to the non-negativity of relative entropy.

If the conditions for saturation are met we can see that the two inequalities become equalities as $t = 1$ and $P_X = Q_X$ if we choose $\ell(x) = \ell_x$. Conversely, using the positive definiteness of the relative entropy, we see that these conditions are in fact necessary to achieve equality. □

The above result allows us to show that certain codes have optimal expected codeword lengths. For example for the source X that outputs symbols 'a', 'b' and 'c' with probabilities $\frac{1}{2}, \frac{1}{4}$ and $\frac{1}{4}$, respectively, the code

$$C(\text{'a'}) = 0, \quad C(\text{'b'}) = 10, \quad C(\text{'c'}) = 11 \tag{2.12}$$

satisfies $\bar{\ell}(C) = \frac{1}{2} + 2 \cdot \frac{1}{4} \cdot 2 = \frac{3}{2}$ and $H(X) = \frac{1}{2}\log 2 + 2 \cdot \frac{1}{4} \log 4 = \frac{3}{2}$, and thus, we know that it is optimal thanks to Proposition 2.8.

The above example is constructed in such a way that all the probabilities are negative powers of 2 and the codeword lengths satisfy the Kraft inequality with equality, and in this particular case it is easy to see from the proof of Proposition 2.8 that $\bar{\ell}(C) = H(X)$. If the probabilities do not have this form the same construction does not generally work. However, we can show the following.

Proposition 2.9. *For a discrete source X with distribution P_X there always exists a prefix code C with $\bar{\ell}(C) \leq H(X) + 1$.*

Proof. The code we construct here is called the Shannon code, and it not optimal in general. We can choose codeword lengths $\ell(x) = \left\lceil \log \frac{1}{P_X(x)} \right\rceil$. These satisfy the Kraft inequality since

$$\sum_x 2^{-\ell(x)} = \sum_x 2^{-\left\lceil \log \frac{1}{P_X(x)} \right\rceil} \leq \sum_x 2^{-\log \frac{1}{P_X(x)}} = \sum_x P_X(x) = 1\,. \tag{2.13}$$

Hence, by Proposition 2.7 they correspond to a prefix code. Moreover, for this code we have

$$\bar{\ell}(C) = \sum_x P_X(x) \left\lceil \log \frac{1}{P_X(x)} \right\rceil \tag{2.14}$$

$$\leq \sum_x P_X(x) \left(\log \frac{1}{P_X(x)} + 1 \right) \tag{2.15}$$

$$= H(X) + 1\,. \tag{2.16}$$

□

The Shannon code is however not guaranteed to be optimal, just close to it.

2.2.3 Huffman codes

In this section we will construct prefix codes with optimal expected codeword length, so-called Huffman codes. We will first learn how to construct the codes and then use this construction to show optimality. Interestingly, the codes were invented by a student that was in the same situation as you are right now!

> In 1951, David Huffman and his MIT information theory classmates were given the choice of a term paper or a final exam. The professor, Robert Fano, assigned a term paper on the problem of finding the most efficient binary code. Huffman, unable to prove any codes were the most efficient, was about to give up and start studying for the final when he hit upon the idea of using a frequency-sorted binary tree and quickly proved this method the most efficient. In doing so, Huffman outdid Fano, who had worked with information theory inventor Claude Shannon to develop a similar code.

The code is constructed using Algorithm 2.1, which step-by-step merges a forest of trivial binary trees into a single binary tree.

The construction is not unique because in each step we can assign the labels '0' and '1' in either way to the two children. Moreover, we are asked to select the two trees with smallest probabilities, but there might be different such pairs, e.g. if we start with a source X with symbols and probabilities ('a', $\frac{1}{3}$), ('b', $\frac{1}{3}$), ('c', $\frac{1}{6}$), ('d', $\frac{1}{6}$) then both of these codes, C_1 and C_2, are possible Huffman codes:

$$C_1(\text{'a'}) = 00, \qquad C_1(\text{'b'}) = 01, \qquad C_1(\text{'c'}) = 10, \qquad C_1(\text{'d'}) = 11 \tag{2.17}$$

$$C_2(\text{'a'}) = 0, \qquad C_2(\text{'b'}) = 10, \qquad C_2(\text{'c'}) = 110, \qquad C_2(\text{'d'}) = 111 \tag{2.18}$$

$$\tag{2.19}$$

Question 2.10. *Can you retrace how they are created step-by-step?*

Input: List of symbols $x \in \mathcal{X}$ with probabilities $p_x = P_X(x)$
Output: Binary tree for the Huffman code

% initialise forest
for *each* $x \in \mathcal{X}$ **do**
    Create a tree with a root node labelled by the probability p_x (and the symbol x) and no other nodes;
    Add this tree to the forest;
end
% condense forest into a single tree
while *number of trees in the forest are larger than* 1 **do**
    select the two trees whose roots have the smallest probabilities, say p and p';
    join the two trees by adding a new root with probability $p + p'$ with the two trees as children, the edges are labelled with '0' and '1';
end
return last remaining tree in the forest;

Algorithm 2.1: Construction of a Huffman code tree.

The codeword lengths for both codes are optimal according to the Kraft inequality, that is, we have

$$\sum_x 2^{-\ell(x)} = 4 \cdot 2^{-2} = 1 \qquad \text{and} \tag{2.20}$$

$$\sum_x 2^{-\ell(x)} = 2^{-1} + 2^{-2} + 2 \cdot 2^{-3} = 1 \tag{2.21}$$

for C_1 and C_2, respectively. We can further compute their respective expected codeword lengths. This yields

$$\bar{\ell}(C_1) = 2 \tag{2.22}$$

$$\bar{\ell}(C_2) = \frac{1}{3} \cdot 1 + \frac{1}{3} \cdot 2 + \frac{1}{6} \cdot 3 + \frac{1}{6} \cdot 3 = 2 \tag{2.23}$$

Let us compare this to the entropy $H(X) = 2 \cdot \frac{1}{3} \log 3 + 2 \cdot \frac{1}{6} \log 6 \approx 1.92$. So from the entropy bound alone we cannot deduce that these codes are optimal in terms of the expected codeword length — but they in fact are!

Proposition 2.11. *Given a source X with probability distribution P_X, any code constructed using Algorithm 2.1 achieves the minimal expected codeword length for any prefix code.*

We call such a code with minimal expected length an *optimal prefix code*. The proof relies on the following lemma, which we show first.

Lemma 2.12. *There exists an optimal prefix code with the following property:*

- *The two longest codewords are siblings and their respective source symbols have the two smallest probabilities (this is not always unique).*

Proof. An optimal code always corresponds to a binary tree with no unused leaves — if not we can compress the tree by removing the parent of the unused leaf, reducing the expected codeword length. There is always at least one pair of leaves at maximum depth, and those are thus occupied with codewords. If those codewords would not correspond to the two symbols with smallest probability we could exchange symbols to move them there, a process which clearly cannot increase the expected codeword length. □

RecursiveHuffman:
Input: Forest f_{in}
Output: Forest f_{out}

if *number of trees in* $f_{\text{in}} = 1$ **then**
  return $f_{\text{out}} = f_{\text{in}}$;
else
  select the two trees in f_{in} whose roots have the smallest probabilities, say p and p';
  create new forest f' from f_{in} by joining the selected trees as in Algorithm 2.1;
  return $f_{\text{out}} =$ **RecursiveHuffman**(f');
end

Algorithm 2.2: Recursive formulation of the Huffman algorithm.

Proof of Proposition 2.11. The recursive formulation of the Huffman algorithm in Algorithm 2.2 is useful here. Clearly the algorithm produces an optimal code when $|\mathcal{X}| = 2$ since in this case the optimal code simply assigns the codewords 0 and 1 to the two symbols, and this is exactly what the output of the Huffman algorithm will be.

We will thus prove optimality by induction as follows. Let us assume, without loss of generality, that our source symbols have probabilities $p_1 \geq p_2 \geq \ldots \geq p_n$. By induction we may assume that the recursive Huffman algorithm provides us with an optimal tree when we call it for $n-1$ symbols with the probabilities $p_1, \ldots, p_2, \ldots, p_{n-2}, p_{n-1} + p_n$. We call the expected codeword length of this optimal tree L^*_{n-1}.

The Huffman algorithm for n symbols, by definition in its recursive form, produces exactly this tree but with the the $(n-1)$-th node split into two siblings with probabilities p_{n-1} and p_n (see Figure 2.2 for an example). Its expected codeword length, L_n, thus satisfies

$$L_n = L^*_{n-1} + p_{n-1} + p_n \,. \tag{2.24}$$

Note that we added $p_{n-1} + p_n$ as compared to the tree for $n-1$ symbols those two leaves are now one level deeper, which increases the expected codeword length by 1 with probability $p_{n-1} + p_n$. On the other hand, we have

$$L^*_{n-1} \leq L^*_n - p_{n-1} - p_n \tag{2.25}$$

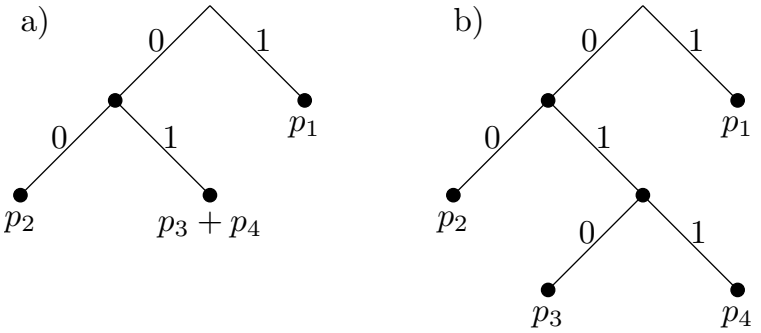


Figure 2.2: **Optimality of Huffman coding, recursive step.** a) Example of a construction with an optimal code tree for $(p_1, p_2, p_3 + p_4)$ with $L_3^* = p_1 + 2p_2 + 2(p_3 + p_4)$. We must have $p_3 + p_4 \leq p_1$ since otherwise this tree would not be optimal. b) The Huffman tree for (p_1, p_2, p_3, p_4) with $L_4 = p_1 + 2p_2 + 3p_3 + 3p_4 = L_3^* + p_3 + p_4$.

since a valid (but not necessarily optimal) prefix code for $n-1$ symbols can be constructed from an optimal prefix code for n symbols by merging the two leaves at maximum depth with minimal probability (which exist due to Lemma 2.12) into a single leaf. Such a code has expected codeword length $L_n^* - p_{n-1} - p_n$, and thus in particular the optimal length of such a tree for $n-1$ symbols must satisfy $L_{n-1}^* \leq L_n^* - p_{n-1} - p_n$.

Combining Eqs. (2.24) and (2.25) yields $L_n \leq L_n^*$, and since L_n can never be smaller than the optimal codeword length (by definition of the latter), these two quantities must in fact be equal. This proves that the Huffman code construction is optimal. □

2.3 Fixed-length block codes

Fixed-length codes have the property that all codewords are equally long. If we require error-free compression, then there is not much flexibility: the expected codeword length has to be equal to $\lceil \log |\mathcal{X}| \rceil$ (assuming that every source symbol appears with strictly positive probability). The picture gets dramatically more interesting if we encode a whole block of n source symbols and only require that the probability of a decoding error vanishes as $n \to \infty$.

2.3.1 Setup for block coding

As we are observing a long sequence of symbols $X_1, X_2, \ldots, X_k, \ldots$, one thing we can do is to treat a block of, say n, symbols as a single symbol (with a much larger alphabet of size $|\mathcal{X}|^n$) and then try to find efficient codes for such blocks. Obviously then we can no longer encode and decode instantaneously as we will need to wait for the full block to perform the encoding, and the decoding operation will in turn yield a full block as well.

So for a block code, we observe a sequence of random variables $X^n = (X_1, \ldots, X_n)$ from a discrete memoryless source and we would like to compress it into a random variable $M \in \{0,1\}^L$, the codeword, using an encoder, a function e from X^n to M. Later on, the

Bibliography

[1] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley, 1991.

[2] T. S. Han. *Information-Spectrum Methods in Information Theory*. Applications of Mathematics. Springer, 2002.

[3] Masahito Hayashi and Hiroshi Nagaoka. General Formulas for Capacity of Classical-Quantum Channels. *IEEE Transactions on Information Theory*, 49(7):1753–1768, jul 2003.

[4] T. Lattimore and C. Szepesvári. *Bandit Algorithms*. Cambridge University Press, 2020.

[5] Yury Polyanskiy, H. Vincent Poor, and Sergio Verdú. Channel Coding Rate in the Finite Blocklength Regime. *IEEE Transactions on Information Theory*, 56(5):2307–2359, may 2010.

[6] C. Shannon. A Mathematical Theory of Communication. *Bell System Technical Journal*, 27:379–423, 1948.

[7] Maurice Sion. On General Minimax Theorems. *Pacific Journal of Mathematics*, 8:171–176, 1958.

[8] Marco Tomamichel and Vincent Y. F. Tan. A Tight Upper Bound for the Third-Order Asymptotics for Most Discrete Memoryless Channels. *IEEE Transactions on Information Theory*, 59(11):7041–7051, nov 2013.