B8.4 INFORMATION THEORY

HARALD OBERHAUSER

PLEASE send any typos/comments/suggestions to oberhauser@maths.ox.ac.uk
This version: Saturday 29th December, 2018, 16:41
29.12.2018: added notes for section 4.6 and 4.7. Minor typos throughout lecture notes.

Contents

Introduction 3
Literature 3
1. Entropy, Divergence and Mutual information 4
1.1. Entropy 4
1.2. Divergence 5
1.3. Mutual information 6
1.4. Conditional entropy/divergence/mutual information 6
1.5. Basic properties and inequalities. 7
1.6. Fano's inequality 12
2. Typical sequences 13
2.1. Weak typicality and the asymptotic equipartition property (AEP) 13
2.2. Source coding with block codes 15
2.3. Non iid source coding (not examinable) 16
2.4. Strong typicality 16
3. Symbol codes 18
3.1. Symbol codes and Kraft–McMillan 18
3.2. Optimal codes 19
3.3. Approaching the lower bound by block codes 21
3.4. Shannon's code 21
3.5. Fano's code [not examinable] 22
3.6. Elias' code 23
3.7. Huffman codes: optimal and a simple constrution 23
4. Channel coding: Shannon's second theorem 27
4.1. Discrete memoryless channels 27
4.2. Channel capacity 28
4.3. Channel codes, rates and errors 29
4.4. Shannon's second theorem: noisy channel coding 30
4.5. Channel codes 35
4.6. Channel coding with non-iid input 36

4.7. Combining symbol and channel coding for DMCs [Section 4.7 is not examinable] 40
References 42
References 42
Appendix A. Probability theory 42
A.1. Measure theory 42
A.2. Probabilty spaces 43
A.3. Discrete random variables 43
A.4. Expectation 44
A.5. Conditional Probabilties and conditional Expectations 44
Appendix B. Convexity 45

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 probablity 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,

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[1]. A concise treatment, focused on the theory is

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

[1] http://www.inference.phy.cam.ac.uk/mackay/itila/

1. Entropy, Divergence and Mutual information

1.1. **Entropy.**

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

$$H_b(X) = -\sum_{x\in\mathcal{X}} \mathbb{P}(X=x)\log_b \mathbb{P}(X=x) \tag{1.1}$$

where we use the convention that $0\cdot\log 0 = 0$. For $b=2$ we usually write $H(X)$ instead of $H_2(X)$.

Some remarks:

- The notation $H(X)$ is somewhat misleading since a random variable is a measurable map $X:\Omega\to\mathcal{X}$ but the entropy $H(X)$ depends on the pmf $p(x)=\mathbb{P}(X=x)$. 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[2] $H(X) = -\mathbb{E}[\log p(X)]$ where p is the pmf of X.
- The choice of base 2 for the logarithm is common (due to computers using two states) but not essential. Since $\log_b x = \frac{\log_a x}{\log_a b}$ 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 $A\cap B=\emptyset$, i.e. the surprise about the occurence of two independent events $\{X\in A\}$, $\{X\in B\}$ is the sum of the surprises of each of these events.

Using that $\mathbb{P}(X\in A\cap B) = \mathbb{P}(X\in A)\mathbb{P}(X\in B)$ for $A\cap B=\emptyset$, 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 $\{X=x\}$, $x\in\mathcal{X}$. We will

[2] *Attention*: often one uses X as an index for the pmf, i.e. $p_X = \mathbb{P}(X=x)$. 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 emphasize that $p_X:\mathcal{X}\to[0,1]$ is a function and not random (does not depend on $\omega\in\Omega$; $p_{X(\omega)}$ does not make any sense)! A better notation would be enumerate rv X_i with $i\in\mathbb{N}$ and denote the pmf of X_i with p_i, though this is less standard.

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

$$H(X) = p\log p + (1-p)\log(1-p) \tag{1.2}$$

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

Example 1.3. If $X = (X_1, X_2)$ with $X_1 \in \mathcal{X}_1$, $X_2 \in \mathcal{X}_2$ then

$$H(X) = H(X_1, X_2) = \sum_{(x_1,x_2)\in\mathcal{X}_1\times\mathcal{X}_2} p(x_1,x_2)\log p(x_1,x_2) \tag{1.3}$$

If additionally, X_1 and X_2 are independent ($p(x_1,x_2) = p(x_1)\,p(x_2)$), then

$$H(X) = H(X_1, X_2) = H(X_1) + H(X_2)\,. \tag{1.4}$$

If X_1 and X_2 are independent and identically distributed, then

$$H(X) = 2H(X_1) = 2H(X_2)\,. \tag{1.5}$$

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 $p = \frac{1}{2}$ in which it is at least intuitive that we need n bits. This hints at another interpretation of entropym, namely as a storage/compression bound of information. We make this connection rigorous in Section 3.

1.2. **Divergence.**

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

$$D(p \parallel q) = \sum_{x\in\mathcal{X}} p(x)\log\frac{p(x)}{q(x)} \tag{1.6}$$

the *divergence* between p and q and set by convention $0\log\frac{0}{0} = 0$ and $D(p \parallel q) = \infty$ if $\exists x \in \mathcal{X}\ q(x) = 0, p(x) > 0$. (Divergence is also known as *information divergence, Kullback–Leibler divergence, relative entropy*).

Note that

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

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 Section (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 $p(0) = \frac{1}{2}$, $q(0) = 1$. 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 $D(p \parallel q) = \infty$ but $D(q \parallel p) = 1$.

1.3. **Mutual information.**

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

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

Some motivation:

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

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

  Hence, we can regard the mutual information as a measure on how much dependence there is between two random variables.
- Unlike covariance $Cov(X,Y) = \mathbb{E}[(X-\mathbb{E}[X])(Y-\mathbb{E}[Y])]$, the mutual information $\mathrm{I}(X;Y)$ takes into account higher order dependence (not just second order dependence).
- Another way to think about mutual information is in terms of entropies

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

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

Definition 1.7. Let X,Y be discrete random variables. The *conditional entropy of Y given X* is defined as

$$H(Y|X) = -\sum_x\sum_y \mathbb{P}(X=x,Y=y)\log\mathbb{P}(Y=y|X=x).$$

In analogy to entropy, it holds that $H(Y|X) = -\mathbb{E}\left[\log p_{Y|X}(Y|X)\right]$. An intuitive way to think about $H(X|Y)$ is as the average surprise we have about X after having observed Y (e.g. if $Y = X$ there's not surprise).

Definition 1.8. Let X,Y be discrete rv with distributions $p_{X,Y}$, $q_{X,Y}$. We call

$$D(p_{Y|X} \parallel q_{Y|X}|p_X) = \sum_{x\in\mathcal{X}} p_X(x)\, D(p_{Y|X=x} \parallel q_{Y|X=x}) \tag{1.7}$$

the *divergence* between p_Y and q_Y *conditional on* X (Also known as *conditinal information divergence, conditional Kullback–Leibler divergence, conditional relative entropy*).

Above can be written as

$$D(p_{Y|X} \parallel q_{Y|X}|p_X) = \mathbb{E}_{X\sim p_X}\left[D(p_{Y|X}(\cdot|X) \parallel q_{Y|X}(\cdot|X))\right].$$

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

$$\mathrm{I}(X;Y|Z) := H(X|Z) - H(X|Y,Z).$$

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

In the same way we regard mutual information as measure of dependence, we can reguard conditional mutual information as a measure of dependence of two rv conditional on knowing another random variable.

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 w be pmf on $\mathcal{X}$. Then*

$$-\sum_{x\in\mathcal{X}} p(x)\log p(x) \le -\sum_{x\in\mathcal{X}} p(x)\log q(x) \tag{1.8}$$

and equality holds iff $p = q$.

Proof. Adding $\sum_{x\in\mathcal{X}} p(x)\log p(x)$ on both sides, we estimate

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

where the inequality follows by Jensen's inequality applied to $f(x) = -\log x$ ($f''(x) > 0$ for $x > 0$ hence strictly convex). Note that by Jensen, equality holds iff $\frac{q(X)}{p(X)}$ is constant. □

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

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

is the pmf p of X and that this minimum equals $H(X)$.

Lemma 1.11 (Log sum inequality)**.** *Let $a_1, \ldots, a_n, b_1, \ldots, b_n \geq 0$. Show that*

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

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

We start with basic properties of the divergence.

Theorem 1.12 (Divergence properties)**.** *Let $p_{X,Y}$ and $q_{X,Y}$ be pmf on $\mathcal{X} \times \mathcal{Y}$ of[3] of discrete random variables X,Y. Then*

(1) **(Information inequality)** $D(p_X \parallel q_X) \geq 0$ with equality iff $p_X = q_X$,
(2) **(Chain rule)** $D(p_{X,Y} \parallel q_{X,Y}) = D(p_{Y|X} \parallel q_{Y|X} | p_X) + D(p_X \parallel q_X)$,
(3) $D(p_{X,Y} \parallel q_{X,Y}) \geq D(p_X \parallel q_X)$
(4) $D(p_{Y|X} \parallel q_{Y|X} | p_X) = D(p_X p_{Y|X} \parallel p_X q_{Y|X})$,
(5) **(Convexity)** for pmfs p_1, p_2, q_1, q_2 we have

$$D(\lambda p_1 + (1-\lambda) p_2 \parallel \lambda q_1 + (1-\lambda) q_2) \leq \lambda D(p_1 \parallel q_1) + (1-\lambda) D(p_2 \parallel q_2) \;\; \forall \lambda \in [0,1]$$

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

$$\begin{aligned}
D(p_{X,Y} \parallel q_{X,Y}) &= \sum_{x,y \in \mathcal{X} \times \mathcal{Y}} p_{X,Y}(x,y) \log \frac{p_{X,Y}(x,y)}{q_{X,Y}(x,y)} \\
&= \sum_{x,y \in \mathcal{X} \times \mathcal{Y}} p_{X,Y}(x,y) \log \frac{p_X(x) p_{Y|X}(y|x)}{q_X(x) q_{Y|X}(y|x)} \\
&= \sum_{x,y \in \mathcal{X} \times \mathcal{Y}} p_{X,Y}(x,y) \log \frac{p_X(x)}{q_X(x)} + \sum_{x,y \in \mathcal{X} \times \mathcal{Y}} p_{X,Y}(x,y) \log \frac{p_{Y|X}(y|x)}{q_{Y|X}(y|x)} \\
&= D(p_X \parallel q_X) + \sum_{x \in \mathcal{X}} p_X(x) \sum_{y \in \mathcal{Y}} p_{Y|X}(y|x) \log \frac{p_{Y|X}(y|x)}{q_{Y|X}(y|x)} = \\
&= D(p_X \parallel q_X) + \sum_{x} p_X(x) D(p_{Y|X=x} \parallel p_{Y|X=x}) \\
&= D(p_X \parallel q_X) + D(p_{Y|X} \parallel q_{Y|x} | p_X).
\end{aligned}$$

Point (3) follows from the chain rule, Point (2) and that

$$D(p_{Y|X} \parallel q_{Y|X} | p_X) = \mathbb{E}_{X \sim p_X} \left[D(p_{Y|X}(\cdot|X) \parallel q_{Y|X}(\cdot|X)) \right] \geq 0$$

by the information inequality (1).

Point 4 follows since

$$D(p_{Y|X} \parallel q_{Y|X} | p_X) = \sum_{x} p_X(x) D(p_{Y|X=x} \parallel q_{Y|X=x})$$

[3] Recall that these determine distributions p_X, p_Y,$p_{Y|X}$ and q_X, q_Y, $q_{Y|X}$.

$$
\begin{aligned}
&= \sum_x p_X(x)\,\mathbb{E}_{Y\sim p_{Y|X=x}}\left[\log\frac{p_{Y|X=x}(Y)}{q_{Y|X=x}(Y)}\right]\\
&= \sum_x p_X(x)\,\mathbb{E}_{Y\sim p_{Y|X=x}}\left[\log\frac{p_{Y|X=x}(Y)}{q_{Y|X=x}(Y)}\frac{p_X(x)}{p_X(x)}\right]\\
&= \mathbb{E}_{X\sim p_X, Y|X\sim p_{Y|X}}\left[\log\frac{p_{Y|X}(Y|X)\,p_X(X)}{q_{Y|X}(Y|X)\,p_X(X)}\right].
\end{aligned}
$$

For Point 5 apply Lemma 1.11 to

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

and sum over $x\in\mathcal{X}$. □

Theorem 1.13 (Mutual Information properties)**.**

(1) $\mathrm{I}(X;Y)\geq 0$ with equality iff $X \perp\!\!\!\perp Y$,

(2)

$$
\begin{aligned}
\mathrm{I}(X;Y) &= \mathrm{I}(Y;X)\\
&= H(X)-H(X|Y)\\
&= H(Y)-H(Y|X).
\end{aligned}
$$

(3) **(Information chain rule)**

$$\mathrm{I}(X_1,\ldots,X_n;Y)=\sum_{i=1}^{n}\mathrm{I}(X_i;Y|X_{i-1},\ldots,X_1)$$

,

(4) **(Data-processing inequality)** If[4] $(X \perp\!\!\!\perp Z)|Y$ then

$$\mathrm{I}(X;Y)\geq\mathrm{I}(X;Z).$$

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

Proof. Point (1) follows since

$$\mathrm{I}(X;Y)=D(p_{X,Y}\parallel p_Xp_Y)\geq 0$$

by the information inequality, (1) of Theorem 1.12.

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

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

[4]Recall that X and Z are conditionally independent given Z, $(X \perp\!\!\!\perp Z)|Y$, if $p_{X,Z|Y}(x,z|y)=p_{X|Y}(x|y)p_{Z|Y}(z|y)$. This is equivalent to $p_{X,Y,Z}(x,y,z)=p(x)p(y|z)p(z|y)$.

Now,

$$p(x,y)\log p(x,y) = p(x,y)\log p_X(x)p_{Y|X}(y|x) = p(x,y)\log p_X(x) + p(x,y)\log p_{Y|X}(y|x)$$

Taking the sum over $x, y \in \mathcal{X}\times\mathcal{Y}$ shows $H(X,Y) = H(X) + H(Y|X)$ and the result follows. Note that this immedialy generalizes to $H(X_1,\ldots,X_n) = \sum_{i=1}^n H(X_i|X_{i-1},\ldots,X_1)$; similarly we get $H(X_1,\ldots,X_n|Y) = \sum_{i=1}^n H(X_i|X_{i-1},\ldots,X_1,Y)$.

Point (3) follows since

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

where the last line follows directly by definition of conditional entropy. For point (4) we use the chain rule (3) to write

$$\begin{aligned}
\mathrm{I}(X;Y,Z) &= \mathrm{I}(X;Z) + \mathrm{I}(X;Y|Z) \\
&= \mathrm{I}(X;Y) + \mathrm{I}(X;Z|Y).
\end{aligned}$$

Now X and Z are conditionally independent given Y, $\mathrm{I}(X;Z|Y) = 0$ but since $\mathrm{I}(X;Y|Z) \geq 0$ we must have

$$\mathrm{I}(X;Y) \geq \mathrm{I}(X;Z).$$

Point (5) follows from the data processing inequality applied with $Z = f(Y)$. □

Remark 1.14.

- Point 2 applied with $X = Y$ shows $\mathrm{I}(X;X) = H(X)$ which explains why entropy is sometimes referred to as *self-information,*
- Point 2 motivates $\mathrm{I}(X;Y)$ 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 (5) formalizes the intuitive but fundamental concept: *post-processing cannot increase information*; e.g. if Z is a rv 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 $\mathrm{I}(\theta;X) = \mathrm{I}(\theta;T(X))$ under all distributions $p_\theta, \theta \in \Theta$.

Theorem 1.15 (Entropy properties)**.** *Let X, Y be discrete rv.*

(1) $0 \leq H(X) \leq \log|\mathcal{X}|$. The upper bound is attained iff X is uniformly distributed on $\mathcal{X}$, 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)** $H(X_1,\ldots,X_n) = \sum_{i=1}^n H(X_i|X_{i-1},\ldots,X_1) \leq \sum_i H(X_i)$ with equality iff the X_i are independent,
(4) For $f : \mathcal{X} \to \mathcal{Y}$, $H(f(X)) \leq H(X)$ with equality iff f is bijective,
(5) Let X and Y be iid. Then

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

with equality iff they are uniformly distributed.

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

$$H(X) \leq -\sum p(x) \log |\mathcal{X}|^{-1} = \log |\mathcal{X}| .$$

Since equality holds in Gibbs' inequality iff $p_X = q$, it follows that X must be uniformly distributed to attain the upper bound. Similary, 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) use that $0 \leq \mathrm{I}(X;Y) = H(X) - H(X|Y)$ 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 $H(X|Y = y) = -\sum_{x \in \mathcal{X}} p_{X|Y}(x|y) \log_{X|Y} p(x|y)$. 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 $X|\{Y = x\}$ is constant, i.e. $(X|\{Y = x\}) = f(x)$.

Point (3) follows as in the proof of Theorem 1.13, Point (3), from the factorisation

$$p_{X_1,\ldots,X_n} = p_{X_1} p_{X_2|X_1} p_{X_3|X_2,X_1} \cdots p_{X_n|X_{n-1},\ldots,X_1} .$$

Point (4) follows since

$$H(f(X)) = H(X, f(X)) \leq H(X) + H(f(X))$$

where inequality holds by Point (3) and $H(f(X)) \geq 0$ by Point (1). If f is bijective, note that with $p_{f(X)}(y) = p_X\left(f^{-1}(y)\right)$ and $H(X) = \sum_x p(x) \log p(x) = \sum_{y \in f(\mathcal{X})} p_{f(X)}(y) \log p_{f(X)}(y) = H(f(Y))$.

Point (5) follows from Jensen's inequality,

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

□

Remark 1.16.

- Point (1) is especially intuitive if we think of entropy as the average surprise we have about X.
- Point (2) formalizes "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 Theorem 1.15, Point (2). We expect that if $\mathbb{P}(X \neq Y)$ is small, then $H(X|Y)$ should be small. Fano's inquality makes this precise.

Theorem 1.17 (Fano's inequality, 1966)**.** *Let X,Y be discrete rv taking values in the same state space. Then*

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

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 Section 3.

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

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

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 $\{X \in \mathcal{X}\setminus\{y\}\}$ which contains $|\mathcal{X}| - 1$ elements. Therefore,

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

It follows that

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

□

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

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 $H(X_1, \dots, X_n) = nH(X)$ measures the average storage cost of sequences of length n.

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

$$\mathbb{P}\left((X_1, \dots, X_n) = \left(x_{i_1}, \dots, x_{i_n}\right)\right) = p^{z(x_1,\dots,x_n)} q^{o(x_1,\dots,x_n)} \tag{2.1}$$

where $p := \mathbb{P}(X = 0)$, $q := \mathbb{P}(X = 1)$ and $z(x_1, \dots, x_n)$ denotes the number of 0s in the sequence $(x_1, \dots, x_n)$ and $o(x_1, \dots, x_n)$ denotes the number of 1s. Now for a "typical sequence" $(x_1, \dots, x_n)$, we can approximate the number of 0s and 1s by $z(x_1, \dots, x_n) \simeq \mathbb{E}[z(X_1, \dots, X_n)] = np$ and $o(x_1, \dots, x_n) \simeq \mathbb{E}[o(X_1, \dots, X_n)] = nq$. Hence,

$$\mathbb{P}((X_1, \dots, X_n) = (x_1, \dots, x_n)) \simeq p^{np} q^{nq}$$

and taking the logarithm on both sides gives

$$-\log \mathbb{P}\left((X_1, \dots, X_n) = \left(x_{i_1}, \dots, x_{i_n}\right)\right) \simeq -np \log p - nq \log q = nH(X).$$

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

$$\mathbb{P}\left((X_1, \dots, X_n) = \left(x_{i_1}, \dots, x_{i_n}\right)\right) \simeq 2^{-nH(X)}.$$

Therefore the set of typical sequences of length n consists of approximately $2^{nH(X)}$ elements, each occuring 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).

By the theorem below,

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

$$-\frac{1}{n} \log p(X_1, \dots, X_n) \to H(X) \quad \text{in probability as } n \to \infty \tag{2.2}$$

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

Theorem 2.2 suggests the following definition of "typical sequences"

Definition 2.3. For $n \in \mathbb{N}$ let

$$\mathcal{T}_n^\epsilon = \left\{(x_1, \dots, x_n) \in \mathcal{X}^n : \left| -\frac{1}{n} \log p(x_1, \dots, x_n) - H(X) \right| < \epsilon \right\}$$

We call $\mathcal{T}_n^\epsilon$ the *set of (weakly) typical sequences of length n of the rv X*.

Theorem 2.4 (Weak AEP 2)**.** *For all $\epsilon > 0$ there exists a n_0 such that for every $n > n_0$*

(1) $p(x_1, \dots, x_n) \in \left[2^{-n(H(X)+\epsilon)}, 2^{-n(H(X)-\epsilon)}\right]$,
(2) $\mathbb{P}\left((X_1, \dots, X_n) \in \mathcal{T}_n^\epsilon\right) \geq 1 - \epsilon$,
(3) $\left|\mathcal{T}_n^\epsilon\right| \in \left[(1-\epsilon)2^{n(H(X)-\delta)}, 2^{n(H(X)+\epsilon)}\right]$.

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

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

$$\mathbb{P}\left((X_1, \dots, X_n) \notin \mathcal{T}_n^\epsilon\right) = \mathbb{P}\left(\left|\frac{1}{n} \log p(X_1, \dots, X_n) - H(X)\right| > \epsilon\right)$$

converges to 0 as $n \to \infty$. Point (3) follows since we get the upper bound from the estimate

$$\begin{aligned} 1 &= \sum_{(x_1,\dots,x_n) \in \mathcal{X}^n} p(x_1, \dots, x_n) \geq \sum_{(x_1,\dots,x_n) \in \mathcal{T}_n^\epsilon} p(x_1, \dots, x_n) \\ &\geq \sum_{(x_1,\dots,x_n) \in \mathcal{T}_n^\epsilon} 2^{-n(H(X)+\epsilon)} \geq 2^{-n(H(X)+\epsilon)} \left|\mathcal{T}_n^\epsilon\right| \end{aligned}$$

and the lower bound follows since by Theorem 2.2 the probability $\mathbb{P}\left((X_1, \dots, X_n) \in \mathcal{T}_n^\epsilon\right)$ converges to 1 so that for large enough n,

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

Remark 2.5.

- When n is large, above suggests to think of $(X_1, \dots, X_n)$ as being drawn uniformly from $\mathcal{T}_n^\epsilon$ with probability $2^{-nH(X)}$,
- Theorem 2.4 does not imply that most sequences are elements of $\mathcal{T}_n^\epsilon$: $\mathcal{T}_n^\epsilon$ has rather small cardinality compared to $\mathcal{X}^n$ since

$$\frac{\left|\mathcal{T}_n^\epsilon\right|}{|\mathcal{X}|^n} \approx \frac{2^{nH(X)}}{2^{n \log |\mathcal{X}|}} = 2^{-n(\log |\mathcal{X}| - H(X))}$$

  (Note that $\log |\mathcal{X}|$ is attained iff X is uniformly distributed by Theorem 1.15). However, $\mathcal{T}_n^\epsilon$ carries most of the probablity mass, Theorem 2.4,(2).
- Theorem 2.4 allows to prove a property for typical sequences and then conclude that this property holds for random sequences $(X_1, \dots, X_n)$ with high probability.

- The most likely sequence $\boldsymbol{x}^\star = \operatorname{argmax}_{\boldsymbol{x}} \mathbb{P}((X_1,\dots,X_n)=\boldsymbol{x})$ is in general not an element of $\mathcal{T}_n^\epsilon$; e.g. $\mathcal{X}=\{0,1\}$ $\mathbb{P}(X=1)=0.9$ then $(1,\dots,1)$ is the most likely sequence but not typical since $\frac{1}{n}\log p(1,\dots,1) = \log\frac{1}{0.9} \approx 0.11$ is not close to $H(X) = 0.1\log\frac{1}{0.1} + 0.9\log\frac{1}{0.9} \approx 0.46$. 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 0s and 1s.

Definition 2.6. For a finite set $\mathcal{A}$, denote with $\mathcal{A}^\star$ the set of finite sequences in $\mathcal{A}$. For $\boldsymbol{a} = a_1\cdots a_n \in \mathcal{A}^\star$ with $a_1,\dots,a_n \in \mathcal{X}$, we call $|\boldsymbol{a}| = n$ the *length* of the sequence $\boldsymbol{a}\in\mathcal{A}^\star$.

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

Theorem 2.7 (Source coding 1)**.** *Let X be discrete rv with state space $\mathcal{X}$. For every $\epsilon>0$ there exists an integer n, and a map*

$$c:\mathcal{X}^n \to \{0,1\}^\star$$

such that

(1) *the map $\bigcup_{k\geq 0}\mathcal{X}^{nk}\to\{0,1\}^\star$ given by $(\boldsymbol{x}_1,\dots,\boldsymbol{x}_{nk})\mapsto c(\boldsymbol{x}_1)\cdots c(\boldsymbol{x}_{nk})\in\{0,1\}^\star$ is injective,*
(2) $\frac{1}{n}\mathbb{E}[|c(X_1,\dots,X_n)|] \leq H(X)+\epsilon$.

Proof. Split $\mathcal{X}^n$ into the disjoint sets $\mathcal{T}_n^{\epsilon_0}$ and $\mathcal{X}^n\backslash\mathcal{T}_n^{\epsilon_0}$ and order the elements in $\mathcal{T}_n^{\epsilon_0}$ and $\mathcal{X}^n\backslash\mathcal{T}_n^{\epsilon_0}$ (in some arbitrary order; e.g. lexicographic). By the AEP, there at most $2^{n(H(X)+\epsilon_0)}$ elements in $\mathcal{T}_n^{\epsilon_0}$, hence we can associate with every element of $\mathcal{T}_n^{\epsilon_0}$ a string consisting of $l_1 := \lceil n(H(X)+\epsilon_0)\rceil$ bits; similarly we associate with every element of $\mathcal{X}^n\backslash\mathcal{T}_{\epsilon_0}^n$ a unique string of $l_2 = \lceil n\log|\mathcal{X}|\rceil$ bits. Now define $c(x_1,\dots,x_n)$ as this strings l_1 resp. l_2 bits, prefixed by a 0 if $(x_1,\dots,x_n)$ is in $\mathcal{T}_n^{\epsilon_0}$ and prefixed by 1 if $(x_1,\dots,x_n)\notin\mathcal{T}_{\epsilon_0}^n$. 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

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

$$
\begin{aligned}
&+ \sum_{\boldsymbol{x}\in\mathcal{X}^n\setminus\mathcal{T}_n^{\epsilon_0}} p(\boldsymbol{x})(n\log|\mathcal{X}|+2)\\
\leq\ & \mathbb{P}\left((X_1,\dots,X_n)\in\mathcal{T}_n^{\epsilon_0}\right)(n(H(X)+\epsilon_0)+2)\\
&+\mathbb{P}\left((X_1,\dots,X_n)\notin\mathcal{T}_n^{\epsilon_0}\right)(n\log|\mathcal{X}|+2)\\
\leq\ & n(H(X)+\epsilon_0)+2+\epsilon_0 n\log|\mathcal{X}|\\
=\ & n(H(X)+\epsilon_1)
\end{aligned}
$$

with $\epsilon_1 := \epsilon_0(1+\log|\mathcal{X}|)+\frac{2}{n}$. For given ϵ we choose first ϵ_0 small enough such that $\epsilon_0(1+\log|\mathcal{X}|)<\frac{\epsilon}{2}$ and then n sufficiently large. □

Shannon's first theorem shows that we encode sequence $X_1,\dots,X_n$ using on average not more than $nH(X)$; put differently: on average we need $H(X)$ bits to encode one symbol from this sequence. We will prove in Section 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 iid source coding (not examinable).** Of course, the assumption that the sequence is generated by iid draws from the same distribution is not realistic (e.g. sentence seen as sequences of letters, etc). However, these assumptions can be significantly weakend and this is the content of the Shannon–McMillan–Breiman Theorem[5]:

Theorem 2.8 (Shannon–McMillan–Breiman (not examinable))**.** *Let $X_1, X_n\dots$, be an ergodic and stationary sequence of rv in a finite state space $\mathcal{X}$. Then*

$$-\frac{1}{n}\log p(X_1,\dots,X_n)\to\overline{H}\quad \textit{in probability as } n\to\infty$$

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

(A sequence is stationary if $X_i,\dots,X_{i+n}$ has the same law for all i,j; loosely speaking a sequence is ergodic if the time average over one realizaition equals the expectation. The class of of stationary and ergodic processes is large and covers many imporant 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 $\mathcal{T}_\epsilon^n$ that still carry most of the pmf.

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

$$\mathbb{P}\left((X_1,\dots,X_n)\in\mathcal{S}_n^\epsilon\right)\geq 1-\epsilon.$$

[5]The version below is due to Breiman, there many extension (a.s. convergence, non-stationary, etc); see [1] for references.

We can construct this set by ordering sequences by their probability and adding themuntil the probablity mass is greater or equal $1-\epsilon$.

Proposition 2.10. *Let* $(\epsilon_n)_n$ *be sequence such that* $\lim_{n\to\infty}\epsilon_n = 0$*. Then*

$$\lim_{n\to\infty}\frac{1}{n}\log\frac{\left|\mathcal{S}_n^{\epsilon_n}\right|}{\left|\mathcal{T}_n^{\epsilon_n}\right|} = 0.$$

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 $\mathcal{T}_n^\epsilon$. Nevertheless, one could argue that the definition of $\mathcal{S}_n^\epsilon$ is simpler and that we should have derived the source coding Theorem, Theorem 2.7, directly using $\mathcal{S}_n^\epsilon$ instead of $\mathcal{T}_n^\epsilon$. However, note that the proof relies on counting the elements of the set of "typical sequences": using $\mathcal{T}_n^\epsilon$ this is trivial due to the "uniofrm distribution" elements in $\mathcal{T}_n^\epsilon$, but for $\mathcal{S}_n^\epsilon$ this is much harder and (only Proposition 2.10 tells us the answer).

3. Symbol codes

We have used the AEP to construct a block code that compresses messages generated by iid samples from a random variable X. In this section we want to use symbol codes to compress, that is 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 $\mathcal{X}^\star$ the set of finite sequences (also called strings) in $\mathcal{X}$. For $\boldsymbol{x} = x_1 \cdots x_n \in \mathcal{X}^\star$ with $x_1, \dots, x_n \in \mathcal{X}$, we call $|\boldsymbol{x}| = n$ the *length* of the sequence $\boldsymbol{x} \in \mathcal{X}^\star$. Given two finite sets $\mathcal{X}$,$\mathcal{Y}$ we call a function $c : \mathcal{X} \to \mathcal{Y}^\star$ a *symbol code*. We call c d-ary if $|\mathcal{Y}| = d$. We call $c(x) \in \mathcal{Y}^\star$ the *codeword* of $x \in \mathcal{X}$.

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

Example 3.2. Let $\mathcal{X} = \{1, \dots, 6\}$ 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 $x_1 = 5$ or $x_1 x_2 = 12$.

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

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

(1) *unambiguous* if c is injective,
(2) *uniquely decodable* if $c^\star$ 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 $x_1, x_2 \in \mathcal{X}$ such that $c(x_1)\boldsymbol{y} = c(x_2)$ for some $\boldsymbol{y} \in \mathcal{Y}^\star$. Prefix codes are also known as *instantaneous codes*.

Clearly,

$$\text{prefix codes} \subset \text{uniquely decodeable codes} \subset \text{unambigious codes}$$

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

Example 3.4. $\mathcal{X} = \{A, B, C, D\}$, $\mathcal{Y} = \{0, 1\}$. 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 an the 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 (Kraft–McMillan)**.**

(1) *Let $c : \mathcal{X} \to \mathcal{Y}^\star$ be uniquely decodeable and set $\ell_x = |c(x)|$. Then*

$$\sum_{x\in\mathcal{X}} |\mathcal{Y}|^{-\ell_x} \leq 1. \tag{3.1}$$

(2) *Conversely, given $(\ell_x)_{x\in\mathcal{X}} \subset \mathbb{N}$ and a finite set $\mathcal{Y}$ such that (3.1) holds, there exists a prefix code $c : \mathcal{X}^\star \to \mathcal{Y}^\star$ such that $|c(x)| = \ell_x \ \forall x \in \mathcal{X}$.*

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

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

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 $(\ell_x)_{x\in\mathcal{X}}$ be a set of integers that fulfills (3.1) and set $\mathcal{Y}$. By relabeling, identify $\mathcal{X}$ as the set $\{1, \ldots, |\mathcal{X}|\} \subset \mathbb{N}$ and assume $\ell_1 \leq \ell_2 \leq \cdots \leq \ell_{|\mathcal{X}|}$. By assumption, $r_m := \sum_{i=1}^{m-1} |\mathcal{Y}|^{-\ell_i} \leq 1$ and we define $c(m)$ as the first ℓ_m digits in the $|\mathcal{Y}|$-ary expansion[6] of the real number $r_m \in (0, 1]$, that is $c(m) := y_1 \cdots y_{\ell_m}$ where

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

This must be a prefix code: if not, there exists m, n, $m < n$, with $c(m)$ 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 $r_n - r_m \leq |\mathcal{Y}|^{-\ell_m}$; on the other hand, by the very definition of r_m and r_n we have $r_n - r_m = \sum_{i=m}^{n-1} |\mathcal{Y}|^{-\ell_i} > |\mathcal{Y}|^{-\ell_m}$ which is a contradiction. □

Remark 3.6. Under the stronger assumption that p is a prefix code in Point (1), the above Theorem 3.5 has a nice proof using trees (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 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.*

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 minimize the average codeword length.

Definition 3.8. We call a symbol code $c : \mathcal{X} \to \mathcal{Y}^\star$ *optimal* for a pmf p on $\mathcal{X}$ and a finite set $\mathcal{Y}$, if it minimizes $\mathbb{E}_{X\sim p}[|c'(X)|]$ among all uniquely decodeable codes $c' : \mathcal{X} \to \mathcal{Y}^\star$.

[6] 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 $\sum_{i=2}^{\infty} 2^{-i}$.

In view of Kraft–McMillan inqequality, Theorem 3.5, given a set $\mathcal{Y}$ a code $c : \mathcal{X} \to \mathcal{Y}^\star$ is optimal if it solves the constraint minimization problem

$$\begin{aligned} &\text{minimize } \sum_{x\in\mathcal{X}} p(x)\,\ell_x, \\ &\text{subject to } \sum_{x:p(x)>0} d^{-\ell_x} \le 1 \text{ and } (\ell_x)_{x\in\mathcal{X}} \subset \mathbb{N}. \end{aligned} \tag{3.2}$$

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 $\ell_x \in \mathbb{N}$ and assume $\sum d^{-\ell_x} = 1$. This in turn is a simple optimization problem that can for example be solved using Lagrangian multipliers, i.e. differentiating

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

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

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

$$\begin{aligned} \mathbb{E}[|c(X)|] - H(X) &= \sum_{x\in\mathcal{X}} p(x)\,\ell_x + \sum_{x\in\mathcal{X}} p(x)\log p(x) \\ &= -\sum_{x\in\mathcal{X}} p(x)\log d^{-\ell_x} + \sum_{x\in\mathcal{X}} p(x)\log p(x) \\ &= -\sum_{x\in\mathcal{X}} p(x)\log\left(q(x)\sum_{x'\in\mathcal{X}} d^{-\ell_{x'}}\right) + \sum_{x\in\mathcal{X}} p(x)\log p(x) \\ &= -\sum p(x)\log\left(\sum_{x'} d^{-\ell_{x'}}\right) + \sum_{x\in\mathcal{X}} p(x)\log\frac{p(x)}{q(x)} \\ &= -\log\left(\sum_{x'} d^{-\ell_{x'}}\right) + D(p\parallel q) \ge 0. \end{aligned}$$

where used that by Kraft–McMillan $\sum_{x'} d^{-\ell_{x'}} \leq 1$ and that divergence is non-negative. Note that equality holds iff $\sum_{x'} d^{-\ell_{x'}} = 1$ and $D(p \parallel q) = 0$. Since $D(p \parallel q) = 0$ 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 $c^\star$ and*

$$H_d(X) \leq \mathbb{E}\left[\left|c^\star(X)\right|\right] < H_d(X) + 1. \tag{3.3}$$

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

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

so conclude by multiplying this inequality with $p(x)$ and taking summing over $x \in \mathcal{X}$ to get (3.3). There are countably many prefix codes, 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 neglibile if X has high entropy but it can be the dominanting term for low entropies. By encoding sequences, we get arbitrary close to the lower bound: Proposition 3.10 applied to the $\mathcal{X}^n$-valued random variable $(X_1, \ldots, X_n)$, $X_1, \ldots, X_n$ iid copies of X, gives a code $c_n : \mathcal{X}^n \to \mathcal{Y}^\star$ such that $\mathbb{E}[|c_n(X_1, \ldots, X_n)|] < H(X_1, \ldots, X_n) + 1$. But $H(X_1, \ldots, X_n) = nH(X)$, hence

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

Put differently, one needs at least $H(X)$ symbols to encode one symbol in the source and this bound is attainable (at asymptotically using block codes). This give

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 $\lceil -\log p_i \rceil$. Shannon gave an explicit algorithm that does this in his seminal 1948 paper: given a pmf p on $\mathcal{X} = \{1, \ldots, m\}$, $p_i = p(x_i)$, and a finite set $\mathcal{Y}$

(1) Order the probabilities and assume (by relabelling) that $p_1 \geq \cdots \geq p_m$,
(2) Define $c_S(x_r)$ as the first $\ell_r := \left\lceil -\log_{|\mathcal{Y}|} p_r \right\rceil$ 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 $H(X) \leq \mathbb{E}[|c_S(X)|] < H(X) + 1$. 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 $|\mathcal{X}|^n$ probablities if we use blocks of length n; for example, already for uppercase English letters $\mathcal{X} = \{A, B, \ldots, Z\}$, 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 on $\mathcal{X}$ and $X \sim p$ and $\mathcal{Y}$ a finite set of cardinality $|\mathcal{Y}| = d$. If we denote with $c_q : \mathcal{X} \to \mathcal{Y}^\star$ a Shannon code for the distribution q, then*

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

Proof. We have

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

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

3.5. **Fano's code [not examinable].** Fanon suggest a different construction that is also very simple to implement. Given a pmf p on $\mathcal{X} = \{1, \ldots, m\}$, $p_i = p(x_i)$, and a finite set $\mathcal{Y}$. 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 and assume (by relabelling) that $p_1 \geq \cdots \geq p_m$,
(2) Find r that minimizes $\left|\sum_{i\leq r} p_i - \sum_{i>r} p_i\right|$ and split $\mathcal{X}$ into two groups $\mathcal{X}_0 := \{x_i : i \leq r\}$, $\mathcal{X}_1 := \{x_i : i > r\}$,
(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) $c_F : \mathcal{X} \to \mathcal{Y}^\star$. As for the Shannon code, it is not hard to show that $\mathbb{E}[|c_F(X)|] \leq H(X)+1$, 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 $\mathcal{X} = \{1,\dots,m\}$, $p_i = p(x_i)$, and a set $\mathcal{Y}$ of cardinality d. Define the Elias code (alos Shannon–Fano–Elias code) $c_E(x_i)$ as the first $\lceil -\log p_i \rceil + 1$ digits in the d-ary expansion of the real number $\sum_{i<r} p_i + \frac{1}{2}p_r$. As above, one can show that $H_d(X)+1 \leq \mathbb{E}[|c_E|] \leq H_d(X)+2$. 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 probablities. Further, it is a precursor of so-called arithmetic coding.

3.7. **Huffman codes: optimal and a simple constrution.** Huffman was a student of Fano and realized that prefix codes corresponds to certain graphs, called rooted trees and that previous constructions such as Fano's build the 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! Recall that a graph

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 $\tau_1,\dots,\tau_n \in \mathcal{T}$, then the graph τ formed by starting with a new vertex r and adding edges to each of the roots of $\tau_1,\dots,\tau_n$ is also a rooted tree. We call r the root of τ and we call the leaves of $\tau_1,\dots,\tau_n$ 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 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 exptected 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 $\mathcal{X} = \{1,\dots,m\}$ and assume (by relabelling) that $p_1 \geq \cdots \geq p_m$ with $p_i := p(x_i)$. Then

(1) Associate with the two leasty likely symbols, two leaves that are joined in a vertex,
(2) Build a new distribution on $m-1$ symbols, $p'_1 \geq p'_2 \geq \cdots \geq p'_{m-2} \geq p'_{m-1}$ where $p'_1 = p_1,\ldots,p'_{m-2} = p_{m-2}$ and $p'_{m-1} := p_{m-1} + p_m$ (i.e. symbols $m-1$ and m are merged into one new symbol with probability $p'_{m-1} = p_{m-1} + p_m$),
(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 assignement is done by assigning 0,1 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 probablity 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 for details.

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

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

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 $\mathcal{X} = \{x_1, \ldots, x_m\}$ 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 $\left|c\left(x_j\right)\right| \leq |c(x_k)|$,*
(2) *the two longest codewords have the same length,*
(3) *the two longest codewords only differ in the last digit.*

We call c the canonical code for the pmf p.

Proof of Lemma 3.16. 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

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

$$= \quad \left(p_j - p_k\right)\left(|c\left(x_k\right)| - \left|c\left(x_j\right)\right|\right).$$

Hence, $|c\left(x_k\right)| - \left|c\left(x_j\right)\right| \geq 0$.

For (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 exptected 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,

$$p'_i = p_i \text{ for } i \in \{1, \ldots, m-1\} \text{ and } p'_{m-1} = p_{m-1} + p_m.$$

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, p_{m-1},p_m are siblings so this is possible). Then the difference in expecteds length is

$$L\left(c^p\right) - L\left(c^{p'}\right) = p_{m-1}\ell + p_m\ell - p'_{m-1}\left(\ell - 1\right) \tag{3.4}$$

$$= p_{m-1} + p_m. \tag{3.5}$$

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

$$L\left(e^p\right) - L\left(e^{p'}\right) = p_{m-1} + p_m. \tag{3.6}$$

Substracting (3.4) from (3.6) yields

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

By assumption, c^p and $e^{p'}$ are optimal, hence both terms are non-negative so both must equal 0. We conclude that $L\left(e^p\right) = L\left(c^p\right)$, 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.

$$\begin{array}{cccccc} p(x) & 0.3 & 0.3 & 0.2 & 0.2 \\ c(x) & 00 & 10 & 01 & 11 \end{array}$$

  is optimal but not Huffman (since c can be obtained by 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) require knowledge p. Further, optimality was defined for messages that are drawn by iid samples. Already 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 $H(X) < \mathbb{E}[|c(X)|] < H(X)+1$. This is a good bound if $H(X)$ 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 on page 21 to encode sequences of length n to reduce the overhead to $\frac{1}{n}$ bits but this again leads to a combinatorial explosion since we need to sort $|\mathcal{X}|^n$ probabilty masses.

4. Channel coding: Shannon's second theorem

In Chapter 3 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

- a finite set $\mathcal{X}$, called the *input alphabet,*
- a finite set $\mathcal{Y}$, called the *input alphabet,*
- a stochastic[7] $|\mathcal{X}| \times |\mathcal{Y}|$-matrix M

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

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 Y allows to uniquely identify the input X (e.g. b,c iff the input is 1). More generally, in a lossless channel we can divide $\mathcal{Y}$ into disjoint sets $\mathcal{Y}_1, \ldots, \mathcal{Y}_{|\mathcal{X}|}$such that

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

which is by Theorem 2 the same as $H(X|Y) = 0$ (since $X = f(Y)$ for $f(y) = \sum_i x_i 1_{y\in\mathcal{Y}_i}$, i.e. X is a deterministic function of Y). The other 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 Theopem 2 equivalent to $H(X|Y) = H(X)$.

Some important examples of channels are

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 ,

$\mathcal{X}\backslash\mathcal{Y}$	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

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

[7]A stochastic matrix has non-negative entries and for each row, the sum over column entries equals 1

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

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

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 intuitve measure for the quality of a channel is

$$H(X) - H(X|Y) = \mathrm{I}(X;Y).$$

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 $C := \sup \mathrm{I}(X;Y)$ the *channel capacity* of DMC$(\mathcal{X}, M, \mathcal{Y})$ where the supremum is taken over all pairs of random variables X, Y that realize the DMC $(\mathcal{X}, M, \mathcal{Y})$.

From $\mathrm{I}(X;Y) = H(X) - H(X|Y) = H(Y) - H(Y|X)$ it follows that

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

Remark 4.5. $\mathrm{I}(X;Y)$ is a convave function in the pmf p_X of X. Also it is bounded by the entropy of X. Hence if we take a supremum over a closed convex set it attains its maximum which shows that $C = \max \mathrm{I}(X;Y)$ is indeed well-defined. Secondly, note only the distribution of X matters (not the specific form of the probabilty space, or the map $(X,Y) : \Omega \to \mathcal{X}\times\mathcal{Y}$, as long as the distribution of $Y|X$ matches M). Therefore one sometimes finds in the literature the notation $C = \max_{p_X} \mathrm{I}(X;Y)$.

Below we calculate the capacity of some simple channels.

Example 4.6 (Binary noisy channel)**.** $\mathcal{X} = \mathcal{Y} = \{0, 1\}$ and

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

that is with probability $q \in [0,1]$ with have a transmission error. Estimate

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

$$\leq \quad 1 - H(q)$$

where $H(q) := -q\log q - (1-q)\log(1-q)$. Note that if Y is uniform, $\mathbb{P}(Y=0) = \mathbb{P}(Y=0) = \frac{1}{2}$, then $H(Y) = 1$ and above is an equality. Since

$$\begin{aligned}\mathbb{P}(Y=0) &= (1-q)\mathbb{P}(X=0) + q\mathbb{P}(X=1)\\ \mathbb{P}(Y=1) &= q\mathbb{P}(X=0) + (1-q)\mathbb{P}(X=1)\end{aligned}$$

by symmetry $\mathbb{P}(Y=0) = \mathbb{P}(Y=0) = \frac{1}{2}$ is equivalent to $\mathbb{P}(X=0) = \mathbb{P}(X=0) = \frac{1}{2}$. Hence, the maximimum is attained if $\mathbb{P}(X=0) = \mathbb{P}(X=0) = \frac{1}{2}$ and it equals

$$C = 1 - H(q).$$

Example 4.7 (Binary erasure channel)**.** As above $\mathcal{X} = \{0,1\}$ but $\mathcal{Y} = \{0, e, 1\}$ where e denotes that an error has occured, i.e.

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

resp. $\mathbb{P}(Y=0|X=0) = q$ and $\mathbb{P}(Y=e|X=0) = 1-q$. This 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. Now as before

$$\mathrm{I}(X;Y) = H(Y) - H(Y|X) = H(Y) - H(q)$$

with $H(q) = q\log q + (1-q)\log(1-q)$. Set $\pi = \mathbb{P}(X=1)$ and calculate

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

Now

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

and therefore the capacity is achievied with $\pi = \mathbb{P}(X=1) = 0.5$ and equals $C = 1-q$.

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

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

That is 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

$$\epsilon_i = \mathbb{P}(d(\boldsymbol{Y}) \neq i | c(i) = \boldsymbol{X}) \text{ for } i = 1,\dots,m$$

where $\boldsymbol{X} = (X_1,\dots,X_n)$ and $\boldsymbol{Y} = (Y_1,\dots,Y_n)$ consist of iid copies of random variables X,Y that realize the DMC. We say that the channel code has

(1) *a maximal probablity of error* $\epsilon_{\max} := \max_{i\in\{1,\dots m\}} \epsilon_i$,
(2) *an arithmetic error* $\overline{\epsilon} := \frac{1}{m}\sum_{i=1}^m \epsilon_i$.

Remark 4.11. For applications we clearly care about $\epsilon_{\max}$ and apriori it is not clear that $\overline{\epsilon}$ is a useful quantity to consider. Note that $\overline{\epsilon} \leq \epsilon_{\max}$ and that $\overline{\epsilon}$ is the expectation of the error ϵ_i, if an element i is choosen uniformly at random. It turns out that good estimates on $\overline{\epsilon}$ imply good estimates on $\epsilon_{\max}$ and that bounds on $\overline{\epsilon}$ 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 $\epsilon_{\max}$ (which we want to make small) and the rate (which we want to keep large). That is, we ask what points in $(\epsilon_{\max}, R)$-plane can be reached by channel codes (with a sufficiently large n)? Before Shannon, a common belief was that that as $\epsilon_{\max}$ goes to 0 so does the rate. A big suprise 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 $\epsilon > 0$ there exists sufficiently large m,n and a (m,n)-channel (c,d) with

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

where $\epsilon_{\max}$ 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. Apriori 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. Al 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 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.

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 sets $\mathcal{Y}_1, \ldots, \mathcal{Y}_m \subset \mathcal{Y}^n$ and associate each set with an input sequence $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m \in \mathcal{X}^n$. That is upon a receiving a sequence $\boldsymbol{y} \in \mathcal{Y}^n$, if we find an i such that $\boldsymbol{y} \in \mathcal{Y}_i$ then we decode 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 approximately $2^{nH(Y)}$ typical output sequences. Similarly, to a given typical input sequence $\boldsymbol{x}$ correspond approximately $2^{nH(Y|X)}$ output sequences that are likely (i.e. $\boldsymbol{y}$'s such that $(\boldsymbol{x}, \boldsymbol{y})$ is typical wrt to $p_{\boldsymbol{X},\boldsymbol{Y}}$) . But for two different typical input sequences, these subsets of $\mathcal{Y}^n$ might overlap, see Figure. 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

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

such typical input sequences, see Figure. Hence, there are at most $2^{n\,\mathrm{I}(X;Y)}$ codewords which gives a rate of $\frac{\log 2^{n\,\mathrm{I}(X;Y)}}{n} = \mathrm{I}(X;Y) \leq C$ 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 rv with pmf $p_{X,Y}$. For $n \in \mathbb{N}$, $\epsilon > 0$ set

$$\begin{aligned} \mathcal{J}_\epsilon^{(n)} \quad = \quad & \left\{ (\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{X}^n \times \mathcal{Y}^n : \max\left(\left| \frac{1}{n} \log p_{\boldsymbol{X},\boldsymbol{Y}}(\boldsymbol{x}, \boldsymbol{y}) - H(X,Y) \right|, \right.\right. \\ & \left.\left. \left| \frac{1}{n} \log p_{\boldsymbol{X}}(\boldsymbol{x}) - H(X) \right|, \left| \frac{1}{n} \log p_{\boldsymbol{Y}}(\boldsymbol{y}) - H(Y) \right| \right) < \epsilon \right\}. \end{aligned}$$

We call $\mathcal{J}_n^\epsilon$ the set of *jointly typical sequences of length n and tolerance ϵ.*

Theorem 4.15 (Joint AEP)**.** *Let $\boldsymbol{X} = (X_1, \ldots, X_n)$, $\boldsymbol{Y} = (Y_1, \ldots, Y_n)$ with entries iid copies of X,Y. Then*

(1) $\lim_{n\to\infty} \mathbb{P}\left((\boldsymbol{X}, \boldsymbol{Y}) \in \mathcal{J}_n^\epsilon \right) = 1$,

(2) $\left|\mathcal{J}_\epsilon^n\right| \le 2^{n(H(X,Y)+\epsilon)}$,

(3) *If X', Y' are independent and X and Y have same margins as X' and Y', that is (X,Y) has pmf $p_X p_Y$, then $\exists n_0$ such that $\forall n \ge n_0$*

$$(1-\epsilon)2^{-n(\mathrm{I}(X;Y)+3\epsilon)} \le \mathbb{P}\left((\boldsymbol{X}', \boldsymbol{Y}') \in \mathcal{J}_\epsilon^{(n)}\right) \quad \le \quad 2^{-n(\mathrm{I}(X;Y)-3\epsilon)}$$

The upper bound holds for all $n \ge 1$.

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

$$\mathbb{P}\left(\left|\frac{1}{n}\log p_{\boldsymbol{X}}(X_1,\ldots,X_n) - H(X)\right| \ge \epsilon\right) < \frac{\epsilon}{3} \text{ for all } n \ge n_1$$

and similarly

$$\begin{aligned}
\mathbb{P}\left(\left|\frac{1}{n}\log p_{\boldsymbol{Y}}(Y_1,\ldots,Y_n) - H(Y)\right| \ge \epsilon\right) &< \frac{\epsilon}{3} \text{ for all } n \ge n_2, \\
\mathbb{P}\left(\left|\frac{1}{n}\log p_{\boldsymbol{X},\boldsymbol{Y}}(X_1,\ldots,X_n,Y_1,\ldots,Y_n) - H(X,Y)\right| \ge \epsilon\right) &< \frac{\epsilon}{3} \text{ for all } n \ge n_3.
\end{aligned}$$

Taking $n \ge \max(n_1, n_2, n_3)$ shows the result.

Point (2) follows since

$$1 = \sum_{\mathcal{X}^n \times \mathcal{Y}^n} p_{\boldsymbol{X},\boldsymbol{Y}}(\boldsymbol{x},\boldsymbol{y}) \ge \sum_{\mathcal{J}_\epsilon^{(n)}} p_{\boldsymbol{X},\boldsymbol{Y}}(\boldsymbol{x},\boldsymbol{y}) \ge \left|\mathcal{J}_\epsilon^{(n)}\right| 2^{-n(H(X,Y)+\epsilon)}$$

and therefore $\left|\mathcal{J}_\epsilon^{(n)}\right| \le 2^{n(H(X,Y)+\epsilon)}$.

Point (3): for the upper bound

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

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

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

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

$$\begin{aligned}
\mathbb{P}\left((\boldsymbol{X}', \boldsymbol{Y}') \in \mathcal{J}_\epsilon^{(n)}\right) &= \sum_{(\boldsymbol{x},\boldsymbol{y}) \in \mathcal{J}_\epsilon^{(n)}} p_{\boldsymbol{X}}(\boldsymbol{x}) p_{\boldsymbol{Y}}(\boldsymbol{y}) \\
&\ge (1-\epsilon) 2^{n(H(X,Y)-\epsilon)} 2^{-n(H(X)+\epsilon)} 2^{-n(H(Y)+\epsilon)}
\end{aligned}$$

$$= \quad 2^{-n(\mathrm{I}(X;Y)+3\epsilon)}.$$

□

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_X on $\mathcal{X}$ and let $\mathcal{J}_\epsilon^{(n)}$ 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 $\prod_{i=1}^n p_X(x_i)$,
(2) For each message $i \in \{1,\dots,m\}$, define its encoding by sampling uniformly from this set of random codewords,
(3) Define the decoder as a typical-set decoder: upon receiving $\boldsymbol{Y}$, check if there exists a unique element $\boldsymbol{X}$ in the set of random codewords such that $(\boldsymbol{X},\boldsymbol{Y}) \in \mathcal{J}_\epsilon^{(n)}$. In this case, decode as the message that was in step 2 associated with the codeword $\boldsymbol{X}$. 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 $(C,\mathcal{D})$. Now,

(1) Sample from the channel code $(C,\mathcal{D})$,
(2) Sample a message W uniformly from $\{1,\dots,m\}$,
(3) Send the sequence $\boldsymbol{X} = C(W)$ through the channel,
(4) Decode the channel output using $\mathcal{D}$, denote the decoded message with $\hat{W}$.

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

$$\mathbb{P}\left(W \neq \hat{W}\right) < \frac{\epsilon}{2}.$$

By conditioning

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

it follows that there must exist a least one channel code $\left(c^\star, d^\star\right)$ such that

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

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 $\overline{\epsilon} < \frac{\epsilon}{2}$ where $\overline{\epsilon}$ denotes the arithmetic error of $\left(c^\star, d^\star\right)$. Thus we have shown the existence of a (m,n)-channel code, rate $R+\frac{1}{n}$ and arithmetic error $\overline{\epsilon} < \frac{\epsilon}{2}$. Further,

$$\overline{\epsilon} = \frac{1}{m}\sum_{i=1}^m \epsilon_i < \frac{\epsilon}{2},$$

or equivalently $\sum_{i=1}^m \epsilon_i < \frac{m}{2}\epsilon$ (ϵ_i denotes the probability of an error in decoding message i using channel code $\left(c^\star, d^\star\right)$). 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 $\frac{m}{2}\epsilon$ which contradicts $\sum_{i=1}^{m}\epsilon_i < \frac{m}{2}\epsilon$. Therefore throwing away the worse half the codewords modifies $\left(c^\star, d^\star\right)$ into a $\left(\frac{m}{2}, n\right)$-channel code with $\epsilon_{\max} < \epsilon$ as required.

Optimality. Fix $\epsilon > 0$ and assume for sufficiently large n there exists a (m,n) channel code with

$$\frac{\log m}{n} > R - \epsilon \text{ and } \epsilon_{\max} < \epsilon \tag{4.1}$$

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

$$\begin{aligned}
\log m &= H(W) \\
&= H\left(W|\hat{W}\right) + \mathrm{I}(W;\hat{W}) \\
&\leq H\left(W|\hat{W}\right) + \mathrm{I}(\boldsymbol{X};\boldsymbol{Y}) \\
&\leq H\left(W|\hat{W}\right) + \sum_{i=1}^{n}\mathrm{I}(X_i;Y_i) \\
&\leq H\left(W|\hat{W}\right) + nC \\
&< 1 + \overline{\epsilon}\log m + nC, \qquad (4.2)
\end{aligned}$$

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

$$\frac{\log m}{n} < \frac{\frac{1}{n} + C}{1-\epsilon}.$$

Using the assumption (4.1), this implies $R - \epsilon < \frac{\frac{1}{n}+C}{1-\epsilon}$. By letting $n \to \infty$ and $\epsilon \to 0$ we conclude that $R \leq C$. □

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

$$\overline{\epsilon} \geq 1 - \frac{1+nC}{\log m} = 1 - \frac{\frac{1}{n}+C}{\frac{1}{n}\log m} \tag{4.3}$$

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

Remark 4.17. The bound (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 $\left(m^k, kn_0\right)$-channel code by concatenating k codewords. But this channel has the same rate. Hence choosing k

large enough contradicts the estimate (4.3). This is often called the *weak converse* of the channel coding theorem. There also exists a *strong converse* (which do not prove) that shows that $\overline{\epsilon} \to 1$ as $n \to \infty$ if $\frac{\log m}{n} \geq C + \epsilon$ for some $\epsilon > 0$.

Lemma 4.18. *Let $W, \hat{W}$ be defined as in the proof of Theorem 4.13. Then $\mathbb{P}\left(W \neq \hat{W}\right) \to 0$ 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. By construction of the random code, ϵ_i is the same for all messages $i \in \{1, \ldots, m\}$, hence $\epsilon_{\max} = \epsilon_1$ (both errors are expectations over the draw of the codewords). By the union bound for probabilities

$$\epsilon_{\max} = \epsilon_1 = \mathbb{P}\left(\hat{W} \neq 1 | W = 1\right) = \mathbb{P}\left(E_1^c \cup \bigcup_{i=2}^{m} E_i | W = 1\right) \leq \mathbb{P}\left(E_1^c | W = 1\right) + \sum_{i=1}^{m} \mathbb{P}(E_i | W = 1)$$

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

$$\begin{aligned} \epsilon_{\max} &\leq \frac{\epsilon}{2} + \sum_{i=2}^{m} 2^{-n(\mathrm{I}(X;Y)-3\epsilon)} \\ &= \frac{\epsilon}{2} + m 2^{-n(\mathrm{I}(X;Y)-3\epsilon)} \\ &= \frac{\epsilon}{2} + 2^{-3n\epsilon} 2^{-n(\mathrm{I}(X;Y)-R)+1} \\ &\leq \epsilon \end{aligned}$$

for large enough n and $R < \mathrm{I}(X;Y)$. □

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 $|\mathcal{X}|^{mn}$ codewords and 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: 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: are more complicated since they use dependicies 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 digitized speech, the obvious approach is to first apply symbol coding for compression, Theorem 2.7, 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 digitized signal into channel coding without an extra compression layer before. Secondly, the channel input will not be an iid sequence. This is a case that needs discussion even without

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 answer 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

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

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

A special case is a stochastic process with X_i iid 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

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

whenever this limit exists.

Obviously, if X_i are iid then the entropy rate exists and $\mathcal{H}(X) = \lim \frac{1}{n}(H(X_1) + \cdots + H(X_n)) = H(X_1)$. However, already for the case when the X_i are independent but not identically distributed the above limit does not necessarily exists (e.g. binary variables X_i with $\mathbb{P}(X_i = 1) = 0.5$ if $\log\log i \in (2k, 2k+1]$ and$\mathbb{P}(X_i = 1) = 0)$ if $\log\log i \in (2k+1, 2k+2]$ for $k = 0, 1, 2, \dots$ 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 stationary stochastic processes X, the entropy rate exists and $\mathcal{H}(X) = \lim_{n\to\infty} H(X_n | X_{n-1}, \dots X_1)$.*

We prepare the proof with two Lemmas

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

Proof. $H(X_{n+1}|X_n,\dots,X_1) \le H(X_{n+1}|X_n,\dots,X_2) = H(X_n|X_{n-1},\dots,X_1)$ where we used that conditioning reduces entropy and for the second equality the stationarity. Since $H(X_n|X_{n-1},\dots,X_1) \ge 0$ the limit exists. □

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

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

$$\left|\frac{1}{n}\sum_{i=1}^n a_i - a\right| \le \frac{1}{n}\sum_{i=1}^n |a_i - a| \le \frac{1}{n}\sum_{i=1}^{n_0} |a_i - a| + \frac{n-n_0}{n}\epsilon \le \frac{1}{n}\sum_{i=1}^{n_0} |a_i - a| + \epsilon$$

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

We now can give a proof of Theorem 4.21.

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

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

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

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

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

for all n and all $x_1,\dots,x_n \in \mathcal{X}$. A Markov chain is time-invariant if

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

for all n and all $a, b \in \mathcal{X}$. A time-invariant Markov chain with state space $\mathcal{X} = \{x_1,\dots,x_m\}$ is characterized by its initial state X_1 and its probability transition matrix (P_{ij}) where $P_{ij} := \mathbb{P}(X_1 = x_j|X_2 = x_i)$. In this case, the pmf of X_{n+1} is given as $p_{X_{n+1}}(x_j) = \sum_i p_{X_n}(x_i)P_{ij}$.

Given a time-invariant Markov process X, a distribution on $\mathcal{X}$ such that the distribution X_{n+1} equals the same as the distribution of X_n for all n is called stationary distribution of X. Hence, a pmf μ on $\mathcal{X}$ is a stationary distribution, if $\mu_j = \sum_{i=1}^m \mu_i P_{ij}$ for all j where $\mu_i = \mu(x_i)$, or in matrix notation

$$\mu P = \mu.$$

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

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

Using the definition of conditional entropy this becomes

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

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 $\mu(a) = \frac{\beta}{\alpha+\beta}$, $\mu(b) = \frac{\alpha}{\alpha+\beta}$. 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[8] graph (V, E) with vertices $V = \{1, \ldots, m\}$. Associate with the edge connecting node i and j a weight $w_{ij} = w_{ji} \geq 0$ (if there's no edge, set $w_{ij} = 0$). Define a Markov chain on the set of vertices V by

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

(Choose the next vertex at random from the neigboring 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

$$w_i = \sum_j w_{ij} \text{ and the sum of weight of all edges } w = \sum_{i,j:j>i} w_{ij}$$

then $\sum_i w_i = 2w$ and we expect that $\mu_i = \frac{w_i}{2w}$. Indeed, we can directly verify $\mu P = \mu$,

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

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

$$\begin{aligned} \mathcal{H}(X) = H(X_2|X_1) &= -\sum_i \mu_i \sum_j P_{ij} \log P_{ij} \\ &= -\sum_i \frac{w_i}{2w} \sum_j \frac{w_{ij}}{w_i} \log \frac{w_{ij}}{w_i} \end{aligned}$$

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

$$
\begin{aligned}
&= -\sum_i \sum_j \frac{w_{ij}}{2w} \log \frac{w_{ij}}{w_i} \\
&= -\sum_i \sum_j \frac{w_{ij}}{2w} \log \frac{w_{ij}}{w_i} \frac{2w}{2w} \\
&= -\sum_i \sum_j \frac{w_{ij}}{2w} \log \frac{w_{ij}}{2w} + \sum_i \sum_j \frac{w_{ij}}{2w} \log \frac{w_i}{2w} \\
&= -\sum_i \sum_j \frac{w_{ij}}{2w} \log \frac{w_{ij}}{2w} + \sum_i \frac{w_i}{2w} \log \frac{w_i}{2w} \\
&= H\left(\ldots, \frac{w_{ij}}{2w}, \ldots\right) - H\left(\ldots, \frac{w_i}{2w}, \ldots\right)
\end{aligned}
$$

In practice, one often is not directly interested in the Markov chain $X = (X_i)$ but to understand functions $Y_i = \varphi(X_i)$ of the Markov chain, resp. the process $Y = (Y_i)$; 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 (exercise prove that Y is Markov iff Φ is injective or constant). However, we know that $\mathcal{H}(Y)$ is well-defined since Y is stationary.

A first approach is to simply estimate $\mathcal{H}(Y)$ by the first n observations as $H(Y_n|Y_{n-1}, \ldots, Y_1)$. However, the convergence $\mathcal{H}(Y) = \lim_n H(Y_n|Y_{n-1}, \ldots, Y_1)$ 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 $H(Y_n|Y_{n-1}, \ldots, Y_1) - H(Y_n|Y_{n-1}, \ldots, Y_1, X_1)$ gives guarantees for the goodness of this estimate.

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

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

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

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

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

Proof. Using that $Y_1 = \varphi(X_1)$, the Markovianity of X, that $Y_i = \varphi(X_i)$ we get

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

Now using that conditioning reduces entropy we further estimate that for all k

$$\leq H(Y_n|Y_{n-1}, \ldots, Y_1, Y_0, \ldots, Y_{-k})$$

$$= H(Y_{n+k+1}|Y_{n+k}, \ldots, Y_1).$$

Hence,

$$H(Y_n|Y_{n-1}, \ldots, Y_2, X_1) \leq \lim_k H(Y_{n+k+1}|Y_{n+k}, \ldots, Y_1) = \mathcal{H}(Y).$$

□

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

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

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

exists. By the chain rule

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

so combining with the above we get

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

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

4.7. **Combining symbol and channel coding for DMCs [Section 4.7 is 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 $V^n := (V_1, \ldots, V_n)$ over a DMC. Therefore we use a coder $c : \mathcal{V}^n \to \mathcal{X}^n$ and recover V^n from the output sequence $\hat{V}^n$ by using a decoder $d : \mathcal{Y}^n \to \mathcal{V}^n$. We want to do this is such a way that $\mathbb{P}\left(V^n \neq \hat{V}^n\right)$ is small.

Theorem 4.30. *Let $(\mathcal{X}, M, \mathcal{Y})$ be 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 satisifies the AEP and*

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

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

Sketch of Proof. There exists a typical set $\mathcal{T}_\epsilon^{(n)}$ of size $\left|\mathcal{T}_\epsilon^{(n)}\right| \leq 2^{n(\mathcal{H}(V)+\epsilon)}$ such that $\mathbb{P}\left(V^n \in \mathcal{T}_\epsilon^{(n)}\right) \geq 1 - \epsilon$. Now consider a coder that only encodes elements in $\mathcal{T}_\epsilon^{(n)}$ and

elements in $\mathcal{V}^n \backslash \mathcal{T}_\epsilon^{(n)}$ are all encoded as the same codeword (representing error). We need at at most

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

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

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

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

$$\mathbb{P}\left(V^n \neq \hat{V}^n\right) \leq \mathbb{P}\left(V^n \notin \mathcal{T}_\epsilon^{(n)}\right) + \mathbb{P}\left(d(Y^n) \neq V^n | V^n \in \mathcal{T}_\epsilon^{(n)}\right) \leq \epsilon + \epsilon.$$

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

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

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

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

Now

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

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

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

□

We emphasize that above theorem makes no assumptions on the stochastic process V other than that the AEP holds; the sequence of random variables $(V_1, \ldots, V_n)$ can have very complciated 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 dapplying 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.

References

[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.

Appendix A. Probability theory

We briefly recall and introduce basic notation from probablity theory. We refer the reader to [3, 4] for an elementary introduction to probabilty 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 $\mathcal{A}$ 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 $(\bigcup_{n\in\mathbb{N}} A_n) \in \mathcal{A}$

Example A.1.

- $\mathcal{X} = \{a, b, c, d\}$ and $\mathcal{A} = \{\emptyset, \{a, b\}, \{c, d\}, \{a, b, c, d\}\}$.
- $\mathcal{X} = \mathbb{R}$ and $\mathcal{A}$ is the smallest σ-algebra that contains all open sets. ("Borel σ-algebra").

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

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

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, lim inf, lim sup, etc.

A.2. **Probabilty spaces.** A *probability space* $(\Omega, \mathcal{F}, \mathbb{P})$ is a measurable space $(\Omega, \mathcal{F})$ together with a map $\mathbb{P} : \mathcal{F} \to [0,1]$ such that

(1) $\mathbb{P}(\Omega) = 1$,
(2) **(σ-additivity)** $\mathbb{P}(\bigcup_n A_n) = \sum \mathbb{P}(A_n)$ for disjoint $(A_n) \subset \mathcal{F}$ (i.e. $A_i \cap A_j = \emptyset$ for $i \neq j$),

We refer to Ω as *sample space,* and to elements of $\mathcal{F}$ as *events*. A $(\mathcal{F}/\mathcal{A})$-measurable map $X : \Omega \to \mathcal{X}$ from Ω to another measurable space $\mathcal{X}$ with σ-algebra $\mathcal{A}$ is called a *random variable.*

Example A.2. Let $\Omega = \{H, T\}$, $\mathcal{F} = \{\emptyset, H, T, \{H, T\}\}$ and $X(\omega) = \begin{cases} 1 & \text{if } \omega = H \\ 0 & \text{if } \omega = T \end{cases}$ with state space $\mathcal{X} = \{H, T\}$. Given $p, q \in [0,1]$ can define two probablity measures $\mathbb{P}$,$\mathbb{Q}$ by setting $\mathbb{P}(X = H) = p$, $\mathbb{Q}(X = T) = q$. 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

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

Example A.3. Let $\Omega = \{H, T\}^{\mathbb{N}}$, let $\mathcal{F}$ be the smallest σ-algebra that contains the set $\{(\omega = (\omega_i) \in \Omega : \omega_n \in \{H, T\})\}$. Then $X_n(\omega) := \begin{cases} 1 & \text{if } \omega_n = H \\ 0 & \text{if } \omega_n = T \end{cases}$ 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 $\{X \in A\}$ and $\{Y \in B\}$ are independent for all measurable A, B. 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

$$X : \Omega \to \mathbb{R}$$

a *discrete rv*, 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 of X with $\mathcal{X}$. Given a probablity 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. Let $\Omega = \{H,T\}$, $\mathcal{F} = \{\emptyset, H, T, \{H,T\}\}$ and $X(\omega) = \begin{cases} 1 & \text{if } \omega = H \\ 0 & \text{if } \omega = T \end{cases}$ with state space $\mathcal{X} = \{H,T\}$. Given $p,q \in [0,1]$ can define two probablity measures $\mathbb{P}$,$\mathbb{Q}$ by setting $\mathbb{P}(X = H) = p$, $\mathbb{Q}(X = H) = q$.

We can regard two discrete rv X,Y with state spaces $\mathcal{X},\mathcal{Y}$ as one discrete rv (X,Y) with state 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,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* $\mathcal{X}$.

A.4. **Expectation.** Given a probablity space $(\Omega,\mathcal{F},\mathbb{P})$ and a discrete random variable $X : \Omega \to \mathcal{X} \subset \mathbb{R}$, 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 rv defined on $(\Omega,\mathcal{F},\mathbb{P})$ then $X \perp\!\!\!\perp Y$ 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. We call

$$Var[X] := \mathbb{E}\left[(X - \mathbb{E}[X])^2\right]$$

the *variance of X* (if this expectation exists) and

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

the *covariance* of X and Y.

A.5. **Conditional Probabilties and conditional Expectations.** Given a probablity space $(\Omega,\mathcal{F},\mathbb{P})$ and $A \in \mathcal{F}$ with $\mathbb{P}(A) > 0$, we define the *conditional probability* $\mathbb{P}(\cdot|A) : \mathcal{F} \to [0,1]$ 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 where $\mathbb{Q}_A(\cdot) := \mathbb{P}(\cdot|A)$. Given two discrete rv X, Y we call

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

the *conditional pmf of Y given X*.

If $A \in \mathcal{F}$ with $\mathbb{P}(A) \geq 0$ and X is a discrete random variable, then define the *conditional expectation of X given A* as

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

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 $f : \mathbb{R} \to \mathbb{R}$ 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 a real-valued random variable such that $\mathbb{E}[X]$ exists. If $\varphi : \mathbb{R} \to \mathbb{R}$ is a convex function such that $\mathbb{E}[|\varphi(X)|] < \infty$, then*

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

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