Information Theory Lecture Notes 2023

Brian M. Kurkoski

2023 May 27

.

Information Theory Lecture Notes

Information Theory Lecture Notes
2023
Brian M. Kurkoski
kurkoski@jaist.ac.jp

Contents

Introduction

This is the lecture notes for I232E Information Theory, taught at the Japan Advanced Institute of Science and Technology. The goal of these lecture notes is to provide material that students need to succeed in the course. When I began writing these lecture notes, I used *Elements of Information Theory* by Cover and Thomas as a starting point, and I recommend you use this book during the course. The JAIST library has a large number of copies for checkout.

Major topics, including compression, channel coding, and rate-distortion theory, follow a consistent format: first a concrete coding scheme is given as motivation, followed by a theorem on optimality, followed by proof of the theorem. Presenting information theory in a systematic structure takes precedence over expressing the most general results possible. Emphasis is placed on developing students understanding of mathematical tools in parallel with understanding of information theory — for example, the law of large numbers and single-variable asymptotic equipartition property receives substantial attention. Consistency and clarity of mathematical notation is important, and the precise meaning of notation is explained.

Programming exercises introduce students to numerical aspects of information theory. By including programming exercises, students can more readily grasp the abstract nature of information theory. Source code examples are included, to introduce programming technique and style to students. The examples are in Matlab, although recently some examples from Python have been added.

An online component of the course is the self-study quiz (SSQ). SSQs are simple, online exercises you can take to test your understanding. Many of the self-study quizzes are in the body of the text, and you should go to the LMS to see the answer and solution. In addition, SSQs support a "flipped classroom" — students are expected to read the first section of the lecture notes, and take a corresponding self-study quiz before coming to class.

In 2022, we made a printed, bound version of Chapters 1–6 for the students at the beginning of the course. Chapters 7 and beyond are being revised at this time, and will be printed for the students later.

These lecture notes are constantly being improved. If you find errors, please post a note in the on-line Discussion Forum. I hope you enjoy information theory!

Notation

An attempt has been made to use consistent notation throughout these lecture notes, some of which is summarized in the table below.

$\mathbf{x}$	vectors are indicated by lower case bold face, for example: $$\mathbf{x} = [x_1, x_2]$$
$[\cdot]^{\mathrm{t}}$	vector and matrix transpose, for example: $$\mathbf{x}^{\mathrm{t}} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$
$\mathbf{x} + \mathbf{y}$	vector addition
$\mathbf{G}$	matrices are indicated by upper case bold face, for example: $$\mathbf{G} = \begin{bmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{bmatrix}$$
$\mathbf{I}_m$	m-by-m identity matrix, for example: $$\mathbf{I}_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$
$\det(\mathbf{G})$	the determinant of a square matrix $\mathbf{G}$.
$\mathcal{A}$	sets are indicated by calligraphic script, for example $\mathcal{A} = \{0, 1, 2, 3, 4\}$
$\lvert\mathcal{A}\rvert$	cardinality of a set, for example $\lvert\mathcal{A}\rvert = 5$
$\mathcal{A} \setminus a$	set subtraction, for example $\mathcal{A} \setminus \{0, 2\} = \{1, 3, 4\}$
$\mathbb{Z}$	the set of integers $\{\ldots, -2, -1, 0, 1, 2, \ldots\}$
$\mathbb{Z}^n$	the set of all integer vectors in n-dimensions, for example $(-4, 0, 1, 3) \in \mathbb{Z}^4$
$\mathbb{R}$	the set of real numbers
$\mathbb{R}^n$	the n-dimensional Euclidean space
$\lvert\lvert\mathbf{x}\rvert\rvert^2$	squared length of vector $\mathbf{x}$, $\sum_{i=1}^{n} x_i^2$
$\lvert\lvert\mathbf{x} - \mathbf{y}\rvert\rvert^2$	squared Euclidean distance between $\mathbf{x}$ and $\mathbf{y}$
$\binom{n}{k}$	n choose k, $\binom{n}{k} = \frac{n!}{k!(n-k)!}$

Chapter 1

Measuring Information: Entropy

There are three important ways to measure information: entropy, mutual information and the Kullback-Leibler divergence (KL divergence). The phrase "measuring information" is meant in an informal sense, since entropy, mutual information and KL divergence are not measures or metrics, in the mathematical sense. This chapter introduces entropy.

1.1 What is Information?

Imagine a race of eight horses. We want to send a message to another person indicating which horse won the race, and we'll consider the average number of bits needed for the message. If we do not know the probability of winning, we could assign a 3-bit message to each horse, and transmit that. Assign the 3-bit message 000 to horse A, the message 001 to horse B, etc. The average message length is 3 bits.

But, we can assign shorter messages to the horses more likely to win, longer messages to horses less likely to win. One such assignment, using a variable-length message, is shown in Table 1.1. Horse A is most likely to win, and has the shortest 1-bit message 0; Horse B is next-most-likely and has a 2-bit message 10, etc. Using the messages in the table, the average message length is computed as:

$$1 \cdot \frac{1}{2} + 2 \cdot \frac{1}{4} + 3 \cdot \frac{1}{8} + 4 \cdot \frac{1}{16} + 4\Big(6 \cdot \frac{1}{64}\Big) \quad = \quad 2 \text{ bits} \tag{1.1}$$

That is, it is possible to describe the winner with 2 bits, rather than 3 bits. This is connected with the idea of entropy, which is the minimum number of bits to describe a random variable.

Additional reading: Cover and Thomas, Sections 2.1, 2.2, 2.5.

horse name	probability of win	3-bit message	variable-length message
Adios	$\frac{1}{2}$	000	0
Big Brown	$\frac{1}{4}$	001	10
Cigar	$\frac{1}{8}$	010	110
Deep Impact	$\frac{1}{16}$	011	1110
Easy Goer	$\frac{1}{64}$	100	111100
Funny Cide	$\frac{1}{64}$	101	111101
Go Man Go	$\frac{1}{64}$	110	111110
Hyperion	$\frac{1}{64}$	111	111111

Table 1.1: A list of horses and their probability of a win. A 3-bit message requires 3 bits on average two transmit the identity of the winning horse. A variable-length message requires only 2 bits on average to transmit.

1.2 Entropy

Entropy is a measure of the uncertainty in the random variable.

Definition 1.1. The *entropy* $H(\mathsf{X})$ of a discrete random variable X with probability distribution $p_{\mathsf{X}}(x)$ is defined by:

$$H(\mathsf{X}) = -\sum_{x \in \mathcal{X}} p_{\mathsf{X}}(x) \log p_{\mathsf{X}}(x) \tag{1.2}$$

where we take: $0 \log 0 = 0$. Unless otherwise noted, the logarithm is base 2, which corresponds to measuring entropy in bits[1].

Entropy can be described as the smallest number of bits required to describe the outcome of an experiment, as the following examples show. In the case of the horse race, the minimum number of bits is two. In the case of a fair coin flip, the number of bits is one. And in the case of rolling a fair die, the number of bits is not an integer.

Example 1.1. *Entropy of a fair coin flip* Let $\mathcal{X} = \{\text{heads}, \text{tails}\}$ and $p_{\mathsf{X}}(\text{heads}) = p_{\mathsf{X}}(\text{tails}) = \frac{1}{2}$, and $H(\mathsf{X})$ is the entropy of a fair coin flip:

$$H(\mathsf{X}) = -\frac{1}{2}\log\frac{1}{2} - \frac{1}{2}\log\frac{1}{2} = 1 \text{ bit} \tag{1.3}$$

This is an important idea: one bit is the amount of information contained in a fair coin flip.

Example 1.2. Find the entropy $H(\mathsf{X})$ of the result X of rolling a six-sided die, where $\mathcal{X} = \{1, 2, 3, 4, 5, 6\}$ and $p_{\mathsf{X}}(x) = \frac{1}{6}$:

$$H(\mathsf{X}) = -6 \cdot \Big(\frac{1}{6}\log\frac{1}{6}\Big) = \log 6 \approx 2.584 \text{ bits} \tag{1.4}$$

[1] If the base is e, the measure is not bits, but a curious unit called "nats".

Example 1.3. In the example of the horse race, the random variable X indicates the winner of the race and has distribution:

$$p_{\mathsf{X}}(x) = [\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \frac{1}{64}, \frac{1}{64}, \frac{1}{64}, \frac{1}{64}]. \tag{1.5}$$

The entropy is:

$$H(\mathsf{X}) = 2 \text{ bits}. \tag{1.6}$$

The entropy is equal to the expected length of the variable length code in Table 1.1. So two bits is the minimum number of bits (on average) to describe the outcome of the horse race. This is discussed in Chapter 4.

SSQ 1.1. Let X be a ternary random variable with $\mathsf{X} = \{1, 2, 3\}$ with probability distribution $p_{\mathsf{X}}(x)$:

$$p_{\mathsf{X}}(x) = \begin{cases} \frac{1}{2} & \text{if } x = 1 \\ \frac{1}{4} & \text{if } x = 2 \\ \frac{1}{4} & \text{if } x = 3 \end{cases} \tag{1.7}$$

Calculate $H(\mathsf{X})$. *Go to the course website for solutions and more SSQs.*

The entropy of a constant is 0. Since a constant is not random, it has no uncertainty and the entropy is 0. The distribution for a constant a is $p_{\mathsf{X}}(a) = 1$, and $\mathcal{X} = \{a\}$, for some a (so that $\mathsf{X} = a$). Then $H(\mathsf{X}) = -1 \log 1 = 0$ bits .

The entropy of a uniform distribution is $\log |\mathcal{X}|$ For a uniform distribution over $\mathcal{X}$ values, $p_{\mathsf{X}}(x) = \frac{1}{|\mathcal{X}|}$. Then, $H(\mathsf{X}) = -\sum_{x \in \mathcal{X}} \frac{1}{|\mathcal{X}|} \log \frac{1}{|\mathcal{X}|} = \log |\mathcal{X}|$.

Proposition 1.1. *Uniform distribution maximizes entropy.* Let X take on values from $\mathcal{X}$. Then, $H(\mathsf{X}) \leq \log |\mathcal{X}|$, with equality if and only if X has a uniform distribution over $\mathcal{X}$.

This is proved on page 63.

Entropy is non-negative. The lower bound $H(\mathsf{X}) \geq 0$ can be shown by noting that $p_{\mathsf{X}}(x) \leq 1$ means $\log \frac{1}{p_{\mathsf{X}}(x)} \geq 0$.

Thus, entropy satisfies the following inequality:

$$0 \leq H(\mathsf{X}) \leq \log |\mathcal{X}|. \tag{1.8}$$

The upper bound is achieved with equality when X is uniformly distributed. The lower bound is achieved with equality when X is a constant.

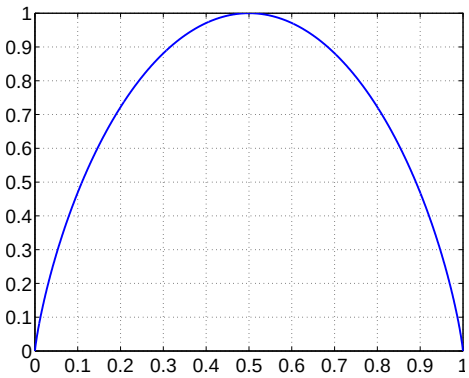


Figure 1.1: Binary entropy function.

1.2.1 Binary Entropy Function

An important random variable is the binary random variable. Let X with $\mathcal{X} = \{0, 1\}$ have distribution:

$$p_{\mathsf{X}}(x) = \begin{cases} p & \text{if } x = 0 \\ 1 - p & \text{if } x = 1. \end{cases} \tag{1.9}$$

Then, $H(\mathsf{X}) = -p \log p - (1 - p) \log(1 - p)$. Clearly $H(\mathsf{X})$ is a function of p, and this is the called the *binary entropy function*:

$$h(p) = -p \log p - (1 - p) \log(1 - p). \tag{1.10}$$

The binary entropy function is plotted in Fig. 1.1. Binary entropy is 0 when $p = 0$ or $p = 1$. In these cases, X is a constant, and the entropy of a constant is 0. The binary entropy function has the maximum value of 1 when $p = \frac{1}{2}$. Like flipping a coin with heads and tails equally likely, the uncertainty is maximum when $p = \frac{1}{2}$.

1.2.2 Joint Entropy

The joint entropy is the entropy of two or more random variables. Look at *Understanding Joint Distributions* on page 37 for some intuition on joint random variables.

Definition 1.2. The *joint entropy* $H(\mathsf{X}, \mathsf{Y})$ of discrete random variables X and Y jointly distributed as $p_{\mathsf{X},\mathsf{Y}}(x, y)$ is:

$$H(\mathsf{X}, \mathsf{Y}) = -\sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p_{\mathsf{X},\mathsf{Y}}(x, y) \log p_{\mathsf{X},\mathsf{Y}}(x, y) \tag{1.11}$$

Note that $H(\mathsf{X}, \mathsf{Y})$ does not take negative values.

Example 1.4. Let $\mathcal{X} = \{\text{apple}, \text{banana}\}$ and $\mathcal{Y} = \{\text{yellow}, \text{green}\}$. Find the joint entropy $H(\mathsf{X}, \mathsf{Y})$ when X and Y are jointly distributed according to:

$p_{\mathsf{XY}}(x, y)$	$y = \text{Y}$	$y = \text{G}$
$x = \text{A}$	$\frac{1}{8}$	$\frac{1}{4}$
$x = \text{B}$	$\frac{1}{2}$	$\frac{1}{8}$

The joint entropy is:

$$H(\mathsf{X}, \mathsf{Y}) = -\big(\frac{1}{2}\log\frac{1}{2} + \frac{1}{4}\log\frac{1}{4} + \frac{1}{8}\log\frac{1}{8} + \frac{1}{8}\log\frac{1}{8}\big) \tag{1.12}$$

$$= \frac{7}{4} \text{ bits} . \tag{1.13}$$

SSQ 1.2. Let X, Y be jointly distributed random variables with joint distribution $p_{\mathsf{XY}}(x, y)$:

$p_{\mathsf{XY}}(x, y)$	$x = 0$	$y = 1$
$y = 0$	$\frac{1}{4}$	$\frac{1}{4}$
$y = 1$	$\frac{1}{4}$	$\frac{1}{4}$

Find the joint entropy $H(\mathsf{X}, \mathsf{Y})$.

1.3 Conditional Entropy and Its Properties

1.3.1 Definition

Conditional entropy $H(\mathsf{Y}|\mathsf{X})$ is the uncertainty of Y given that X is known. If X and Y are dependent, then knowledge of X can reduce the uncertainty of Y. Conditional entropy is one of the most important concepts in information theory. There are two types of conditional entropy, $H(\mathsf{Y}|\mathsf{X} = x)$ and $H(\mathsf{Y}|\mathsf{X})$.

Definition 1.3. The *conditional entropy* $H(\mathsf{Y}|\mathsf{X} = x)$ is given by:

$$H(\mathsf{Y}|\mathsf{X} = x) = -\sum_{y \in \mathcal{Y}} p_{\mathsf{Y}|\mathsf{X}}(y|x) \log p_{\mathsf{Y}|\mathsf{X}}(y|x) \tag{1.14}$$

Definition 1.4. The *conditional entropy* $H(\mathsf{Y}|\mathsf{X})$ of discrete random variables

X and Y jointly distributed as $p_{\mathsf{X},\mathsf{Y}}(x,y)$ is:

$$H(\mathsf{Y}|\mathsf{X}) = \sum_{x\in\mathcal{X}} p_{\mathsf{X}}(x) H(\mathsf{Y}|\mathsf{X}=x) \tag{1.15}$$

$$= -\sum_{x\in\mathcal{X}} p_{\mathsf{X}}(x) \sum_{y\in\mathcal{Y}} p_{\mathsf{Y}|\mathsf{X}}(y|x) \log p_{\mathsf{Y}|\mathsf{X}}(y|x) \tag{1.16}$$

$$= -\sum_{x\in\mathcal{X}} \sum_{y\in\mathcal{Y}} p_{\mathsf{X}}(x) p_{\mathsf{Y}|\mathsf{X}}(y|x) \log p_{\mathsf{Y}|\mathsf{X}}(y|x) \tag{1.17}$$

$$= -\sum_{x\in\mathcal{X}} \sum_{y\in\mathcal{Y}} p_{\mathsf{X}\mathsf{Y}}(x,y) \log p_{\mathsf{Y}|\mathsf{X}}(y|x). \tag{1.18}$$

Note that $H(\mathsf{Y}|\mathsf{X})$ is a number, while $H(\mathsf{Y}|\mathsf{X}=x)$ is a function of x.

Example 1.5. Let Y be a random variable that indicates how Steve goes to work, $\mathcal{Y} = \{\text{Bicycle}, \text{Train}\}$. Let X be a random variable that indicates the weather, $\mathcal{X} = \{\text{Sunny}, \text{Rainy}\}$. Let the conditional probability $p_{\mathsf{Y}|\mathsf{X}}(y|x)$ be given by:

$p_{\mathsf{Y}\|\mathsf{X}}(y\|x)$	$y = \text{B}$	$y = \text{T}$
$x = \text{S}$	$\frac{1}{2}$	$\frac{1}{2}$
$x = \text{R}$	0	1

and let $p_{\mathsf{X}}(\text{S}) = \frac{2}{3}$ and $p_{\mathsf{X}}(\text{R}) = \frac{1}{3}$. Find $H(\mathsf{Y}|\mathsf{X}=x)$ and $H(\mathsf{Y}|\mathsf{X})$.

Solution:

$$H(\mathsf{Y}|\mathsf{X}=\text{S}) = -\sum_{y\in\mathcal{Y}} p_{\mathsf{Y}|\mathsf{X}}(y|\text{S}) \log p_{\mathsf{Y}|\mathsf{X}}(y|\text{S}) = -\frac{1}{2}\log\frac{1}{2} - \frac{1}{2}\log\frac{1}{2} \tag{1.19}$$

$$= 1 \tag{1.20}$$

$$H(\mathsf{Y}|\mathsf{X}=\text{R}) = -\sum_{y\in\mathcal{Y}} p_{\mathsf{Y}|\mathsf{X}}(y|\text{R}) \log p_{\mathsf{Y}|\mathsf{X}}(y|\text{R}) = -1\log 1 + 0\log 0 \tag{1.21}$$

$$= 0 \tag{1.22}$$

Next,

$$H(\mathsf{Y}|\mathsf{X}) = \sum_{x\in\mathcal{X}} p_{\mathsf{X}}(x) H(\mathsf{Y}|\mathsf{X}=x) \tag{1.23}$$

$$= p_{\mathsf{X}}(\text{S}) H(\mathsf{Y}|\mathsf{X}=\text{S}) + p_{\mathsf{X}}(\text{R}) H(\mathsf{Y}|\mathsf{X}=\text{R}) \tag{1.24}$$

$$= \frac{2}{3}. \tag{1.25}$$

SSQ 1.3. Let X and Y have conditional distribution given by:

$p_{\mathsf{X}\|\mathsf{Y}}(x\|y)$	$x=0$	$x=1$	$x=2$	$x=3$
$y=0$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$
$y=1$	$\frac{1}{2}$	$\frac{1}{2}$	0	0

(1.26)

(a) What is $H(\mathsf{X}|\mathsf{Y}=0)$? What is $H(\mathsf{X}|\mathsf{Y}=1)$?
(b) Let $p_{\mathsf{Y}}(0)=\frac{1}{4}, p_{\mathsf{Y}}(1)=\frac{3}{4}$. What is $H(\mathsf{X}|\mathsf{Y})$?

1.3.2 Properties of Conditional Entropy

Intuitively, entropy is the measure of uncertainty in bits. Conditional entropy can reduce the uncertainty. If X and Y are dependent random variables, then knowing something about Y will reduce your uncertainty about X. This is an important proposition:

Proposition 1.2. *Conditioning reduces entropy*: $H(\mathsf{X}|\mathsf{Y}) \leq H(\mathsf{X})$, with equality if and only if X and Y are independent.

This is proved in Section 3.2.1 on page 63. Note that $H(\mathsf{Y}|\mathsf{X}) = H(\mathsf{Y})$ if and only if X and Y are independent.

As an example, let X be tomorrow's weather with $\mathcal{X} = \{\text{Sunny, Rainy, Cloudy, Snowy}\}$, and let $\widehat{\mathsf{X}}$ be today's forecast of tomorrow's weather. Clearly, we hope that the forecast will reduce our uncertainty of tomorrow weather — otherwise the forecast would be useless. We can express this idea using entropy as $H(\mathsf{X}|\widehat{\mathsf{X}}) < H(\mathsf{X})$.

In some cases, conditional entropy can be found from the logic of the problem, instead of performing computations. Continuing the weather forecast, suppose a genie tells us if the weather forecast is correct or erroneous. The error is $\mathsf{E}=0$ if the weather forecast is correct, and $\mathsf{E}=1$ if the forecast is incorrect. It should be fairly easy to see that $H(\mathsf{X}|\widehat{\mathsf{X}}, \mathsf{E}=0) = 0$ because if the weather forecast is correct, then there is no uncertainty about tomorrow's weather. On the other hand, $H(\mathsf{X}|\widehat{\mathsf{X}}, \mathsf{E}=1)$ is not 0.

Finally, suppose that now it is tomorrow and we can observe the weather. There is no uncertainty about the weather, since we can observe it directly. We can express this idea as $H(\mathsf{X}|\mathsf{X}) = 0$.

SSQ 1.4. Eve is watching a video of her favorite soccer team play a match, whose outcome X is one of $\{\text{Win}, \text{Tie}, \text{Lose}\}$. Eve makes the following statement: "I think my favorite team will win." But the video was recorded yesterday, and her friend Cheryl knows the outcome. Cheryl replies with $\mathsf{E} \in \{\text{correct}, \text{incorrect}\}$, indicating whether Eve's statement is correct or incorrect. From Eve's point of view, which of the following are equal to 0? From Cheryl's point of view, which of the following are equal 0?

1. $H(\mathsf{X}|\mathsf{E} = \text{correct})$
2. $H(\mathsf{X}|\mathsf{E} = \text{incorrect})$

Example 1.6. Let X be the result of rolling a six-sided die, where $\mathcal{X} = \{1, 2, 3, 4, 5, 6\}$ and $p_{\mathsf{X}}(x) = \frac{1}{6}$. Let Y indicate whether X is odd or even, so $\mathcal{Y} = \{\text{odd}, \text{even}\}$. Find $H(\mathsf{Y}|\mathsf{X})$ and $H(\mathsf{X}|\mathsf{Y})$.

Clearly if you know X, then you know Y, so there is no uncertainty:

$$H(\mathsf{Y}|\mathsf{X}) = 0 \text{ bits} . \tag{1.29}$$

On the other hand if you only know $\mathsf{Y} = \text{even}$, then X could be 2, 4 or 6, each with probability $\frac{1}{3}$:

$$H(\mathsf{X}|\mathsf{Y} = \text{even}) = \log 3 \tag{1.30}$$

and similarly $H(\mathsf{X}|\mathsf{Y} = \text{odd}) = \log 3$. Since $p_{\mathsf{Y}}(\text{odd}) = p_{\mathsf{Y}}(\text{even}) = \frac{1}{2}$, we have:

$$H(\mathsf{X}|\mathsf{Y}) = \frac{1}{2}\log 3 + \frac{1}{2}\log 3 = \log 3 \approx 1.585 \text{ bits} \tag{1.31}$$

From Example 1.2, $H(\mathsf{X}) \approx 2.584$ while $H(\mathsf{X}|\mathsf{Y}) \approx 1.585$, showing that conditioning reduces entropy.

Proposition 1.3. *Conditional entropy of functions* For any[2] function $g(\cdot)$, $H(g(\mathsf{X})|\mathsf{X}) = 0$.

This proposition says that if you know a random variable, then you also know a function of that random variable. A special case is $g(x) = x$, so that $H(\mathsf{X}|\mathsf{X}) = 0$ — if you know a variable, then there is no uncertainty about it.

These ideas can be formalized by using a function g, as follows.

Proposition 1.4. *Entropy conditioned on a bijective function* Let $g(x)$ be a bijective function, that is, $g(x)$ and $g^{-1}(x)$ have one unique value for all $x \in \mathcal{X}$. Then, $H(\mathsf{X}|g(\mathsf{X})) = 0$.

Note that if $g(\cdot)$ is not bijective, then the entropy may not be zero, which is illustrated in the following example.

[2]The function g is deterministic, that is, not random. A student once asked "what if g is random?" We usually do not think of functions as being random, but if it were, then yes, it could increase entropy.

Example 1.7. Let X be defined on $\mathcal{X} = \{-1, 0, 1\}$ with $p_{\mathsf{X}}(0) = \frac{1}{2}, p_{\mathsf{X}}(-1) = p_{\mathsf{X}}(1) = \frac{1}{4}$. Let $g(x) = x^2$ and let $\mathsf{Y} = g(\mathsf{X})$, so that $\mathcal{Y} = \{0, 1\}$ and $p_{\mathsf{Y}}(0) = \frac{1}{2}$ and $p_{\mathsf{Y}}(1) = \frac{1}{2}$. Compute $H(\mathsf{Y}|\mathsf{X})$ and $H(\mathsf{X}|\mathsf{Y})$.

Solution. First compute $H(\mathsf{Y}|\mathsf{X})$:

$$H(\mathsf{Y}|\mathsf{X}) = p_{\mathsf{X}}(-1)H(\mathsf{Y}|\mathsf{X} = -1) + p_{\mathsf{X}}(0)H(\mathsf{Y}|\mathsf{X} = 0) + p_{\mathsf{X}}(1)H(\mathsf{Y}|\mathsf{X} = 1)$$

Given X is known, there is no uncertainty about Y, that is $H(\mathsf{Y}|\mathsf{X} = x) = 0$, so $H(\mathsf{Y}|\mathsf{X}) = 0$.

On the other hand, to compute $H(\mathsf{X}|\mathsf{Y})$, note that x^2 is not bijective, since $-1^2 = 1^2$. The conditional distribution $p_{\mathsf{X}|\mathsf{Y}}(x|y)$ is:

$p_{\mathsf{X}\|\mathsf{Y}}(x\|y)$	$x=-1$	$x=0$	$x=1$
$y=0$	0	1	0
$y=1$	$\frac{1}{2}$	0	$\frac{1}{2}$

and $p_{\mathsf{Y}}(y) = [\frac{1}{2}, \frac{1}{2}]$. In particular, $H(\mathsf{X}|\mathsf{Y} = 0) = 0$ since if $y = 0$ we could only have $x = 0$. But $H(\mathsf{X}|\mathsf{Y} = 1)$ is not 0, because x could be -1 or 1.

$$\begin{aligned} H(\mathsf{X}|\mathsf{Y} = 1) &= \sum_{x \in \mathcal{X}} p_{\mathsf{X}|\mathsf{Y}}(x|1) \log p_{\mathsf{X}|\mathsf{Y}}(x|1) && (1.32)\\ &= -\big(\frac{1}{2}\log\frac{1}{2} + 0\log 0 + \frac{1}{2}\log\frac{1}{2}\big) && (1.33)\\ &= 1 && (1.34) \end{aligned}$$

So that:

$$\begin{aligned} H(\mathsf{X}|\mathsf{Y}) &= H(\mathsf{X}|\mathsf{Y} = 0)p_{\mathsf{Y}}(0) + H(\mathsf{X}|\mathsf{Y} = 1)p_{\mathsf{Y}}(1) && (1.35)\\ &= 0 \cdot \frac{1}{2} + 1 \cdot \frac{1}{2} && (1.36)\\ &= \frac{1}{2}. && (1.37) \end{aligned}$$

1.3.3 Chain Rules for Entropy

Proposition 1.5. *Chain Rule for Entropy* For random variables X and Y:

$$H(\mathsf{X}, \mathsf{Y}) = H(\mathsf{X}) + H(\mathsf{Y}|\mathsf{X}) \tag{1.38}$$

Proof:

$$\begin{aligned} H(\mathsf{X},\mathsf{Y}) &= -\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p_{\mathsf{XY}}(x,y)\log p_{\mathsf{XY}}(x,y) && (1.39)\\ &= -\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p_{\mathsf{XY}}(x,y)\log p_{\mathsf{X}}(x)p_{\mathsf{Y}|\mathsf{X}}(y|x) && (1.40)\\ &= -\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p_{\mathsf{XY}}(x,y)\log p_{\mathsf{X}}(x) - \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p_{\mathsf{XY}}(x,y)\log p_{\mathsf{Y}|\mathsf{X}}(y|x)\\ &= -\sum_{x\in\mathcal{X}} p_{\mathsf{X}}(x)\log p_{\mathsf{X}}(x) - \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p_{\mathsf{XY}}(x,y)\log p_{\mathsf{Y}|\mathsf{X}}(y|x) && (1.41)\\ &= H(\mathsf{X}) + H(\mathsf{Y}|\mathsf{X}) \quad \square && (1.42) \end{aligned}$$

Note that in general $H(\mathsf{X}|\mathsf{Y}) \neq H(\mathsf{Y}|\mathsf{X})$. However, the following does hold:

$$H(\mathsf{X}) - H(\mathsf{X}|\mathsf{Y}) = H(\mathsf{Y}) - H(\mathsf{Y}|\mathsf{X}) \tag{1.43}$$

Proposition 1.6. *Generalized Chain Rule for Entropy.* Let $\mathsf{X}_1, \mathsf{X}_2, \ldots, \mathsf{X}_n$ be jointly distributed as $p_{\mathsf{X}_1,\mathsf{X}_n}(x_1,\ldots,x_n)$ Then,

$$H(\mathsf{X}_1,\mathsf{X}_2,\ldots,\mathsf{X}_n) = \sum_{i=1}^{n} H(\mathsf{X}_i|\mathsf{X}_{i-1}\ldots,\mathsf{X}_1) \tag{1.44}$$

Proof:

$$\begin{aligned} H(\mathsf{X}_1,\mathsf{X}_2) &= H(\mathsf{X}_1) + H(\mathsf{X}_2|\mathsf{X}_1) && (1.45)\\ H(\mathsf{X}_1,\mathsf{X}_2,\mathsf{X}_3) &= H(\mathsf{X}_1) + H(\mathsf{X}_2,\mathsf{X}_3|\mathsf{X}_1) && (1.46)\\ &= H(\mathsf{X}_1) + H(\mathsf{X}_2|\mathsf{X}_1) + H(\mathsf{X}_3|\mathsf{X}_2,\mathsf{X}_1) && (1.47) \end{aligned}$$

Continuing this process iteratively, we have:

$$\begin{aligned} H(\mathsf{X}_1,\mathsf{X}_2,\ldots,\mathsf{X}_n) &= H(\mathsf{X}_1) + H(\mathsf{X}_2|\mathsf{X}_1) + \cdots + H(\mathsf{X}_n|\mathsf{X}_{n-1},\cdots,\mathsf{X}_2,\mathsf{X}_1)\\ &= \sum_{i=1}^{n} H(\mathsf{X}_i|\mathsf{X}_{i-1},\ldots,\mathsf{X}_1) \quad \square \end{aligned}$$

Proposition 1.7. *Independence bound on entropy.* Let $\mathsf{X}_1, \mathsf{X}_2, \ldots, \mathsf{X}_n$ random variables jointly distributed as $p_{\mathsf{X}}(x_1,\ldots,x_n)$. Then,

$$H(\mathsf{X}_1,\mathsf{X}_2,\ldots,\mathsf{X}_n) \leq \sum_{i=1}^{n} H(\mathsf{X}_i) \tag{1.48}$$

Proof:

$$H(\mathsf{X}_1,\mathsf{X}_2,\ldots,\mathsf{X}_n) = \sum_{i=1}^{n} H(\mathsf{X}_i|\mathsf{X}_{i-1},\ldots,\mathsf{X}_1) \leq \sum_{i=1}^{n} H(\mathsf{X}_i) \quad \square \tag{1.49}$$

SSQ 1.5. What is the correct relationship, $=$, $\geq$, $\leq$ or "unknown" for each?

(a) $H(2\mathsf{X})$_____ $H(\mathsf{X})$

(b) $H(2\mathsf{X})$_____ $H(\mathsf{X}^2)$

(c) $H(\mathsf{X}^2|\mathsf{X})$_____ $H(\mathsf{X}|\mathsf{X}^2)$

(d) $H(\mathsf{X},\mathsf{Y},\mathsf{Z})$_____ $H(\mathsf{X})+H(\mathsf{Y})+H(\mathsf{Z})$

Example 1.8. Find the entropies $H(\mathsf{X},\mathsf{Y})$, $H(\mathsf{X}|\mathsf{Y})$ and $H(\mathsf{Y}|\mathsf{X})$ for random variables X,Y with $\mathsf{X}=\mathcal{Y}=\{1,2,3,4\}$ jointly distributed as:

$$p_{\mathsf{XY}}(x,y) \quad = \quad \begin{bmatrix} \frac{1}{8} & \frac{1}{16} & \frac{1}{32} & \frac{1}{32} \\ \frac{1}{16} & \frac{1}{8} & \frac{1}{32} & \frac{1}{32} \\ \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} \\ \frac{1}{4} & 0 & 0 & 0 \end{bmatrix} \tag{1.50}$$

(x is in the columns, y is in the rows). Then, the marginal distribution of X and Y are obtained by summing rows and columns of $p_{\mathsf{XY}}(x,y)$ respectively:

$$p_{\mathsf{X}}(x) \quad = \quad \begin{bmatrix} \frac{1}{2} & \frac{1}{4} & \frac{1}{8} & \frac{1}{8} \end{bmatrix} \text{ and} \tag{1.51}$$

$$p_{\mathsf{Y}}(y) \quad = \quad \begin{bmatrix} \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \end{bmatrix} \tag{1.52}$$

Hence, $H(\mathsf{X})=7/4$ bits and $H(\mathsf{Y})=2$ bits. The joint entropy $H(\mathsf{X},\mathsf{Y})$ is:

$$H(\mathsf{X},\mathsf{Y}) \quad = \quad -\sum_{x=1}^{4}\sum_{y=1}^{4} p_{\mathsf{XY}}(x,y)\log p_{\mathsf{XY}}(x,y) = \frac{27}{8} \text{ bits} \tag{1.53}$$

And the conditional entropies are:

$$H(\mathsf{X}|\mathsf{Y}) = H(\mathsf{X},\mathsf{Y}) - H(\mathsf{Y}) = \frac{27}{8} - 2 = \frac{11}{8} \text{ bits, and} \tag{1.54}$$

$$H(\mathsf{Y}|\mathsf{X}) = H(\mathsf{X},\mathsf{Y}) - H(\mathsf{X}) = \frac{27}{8} - \frac{7}{4} = \frac{13}{8} \text{ bits.} \tag{1.55}$$

1.4 Matlab Source Code

1.4.1 Matlab basics

Command line computations:

```
1 >> log2(3)
2
3 ans =
```

```
4
5       1.5850
6
7   >> px = [1/2 1/4 1/4];
8   >> -sum( px .* log2(px) )
9
10  ans =
11
12      1.5000
```

Learn about new commands:

```
1   >> doc linspace
2   >> help linspace
3    linspace Linearly spaced vector.
4       linspace(X1, X2) generates a row vector of 100 linearly
5       equally spaced points between X1 and X2.
```

1.4.2 Custom Function: Binary Entropy Function

Create a custom function `binaryEntropyFunction.m`. First, create a new file in the text editor (open if it already exists):

```
1   >> edit binaryEntropyFunction.m
```

Then type source code below into the text editor, which computes the binary entropy function $h(p)$:

```
1   function h = binaryEntropyFunction(p)
2
3   p(find(p<1E-10)) = 1E-10;
4
5   h = - p .* log2(p) - (1-p) .* log2(1-p);
```

Run a command, then display a variable:

```
1   >> H = binaryEntropyFunction(0.11);
2   >> H
3
4   H =
5
6       0.4999
```

Run a command and show the output (no semicolon):

```
1   >>  binaryEntropyFunction(0.11)
2
3   ans =
4
5       0.4999
```

Make a plot of the binary entropy function:

```
>> p = linspace(0,1,101);
>> hp = binaryEntropyFunction(p);
>> plot(p,hp);
>> xlabel('source probability p')
>> ylabel('entropy h(p)')
```

1.4.3 Compute Entropy

Given an arbitrary source distribution $p_{\mathsf{X}}(x)$, find the entropy $H(\mathsf{X})$.

```
>> px = rand(1,100);         %100 random numbers between 0 and 1
>> px = px / sum(px);        %force px to sum to 1.
>> H = computeEntropy(px)    %compute the entropy
```

```
function H = computeEntropy(px)

assert( all(px) >= 0,'px are not all non-negative');
assert( abs( 1 - sum(px) ) < 1E-10, 'px does not sum to 1');

%change 0 to 1E-10 avoids px*log2(px) = NaN
px(find(px < 1E-10)) = 1E-10;

H = -sum( px .* log2(px) );
```

Given $p_{\mathsf{Y}|\mathsf{X}}(y|x)$ and $p_{\mathsf{X}}(x)$, compute the conditional entropy $H(\mathsf{Y}|\mathsf{X})$.

```
function H = computeConditionalEntropy(pygx,px)

assert( all(px) >= 0,'px are not all non-negative');
assert( abs( 1 - sum(px) ) < 1E-10, 'px does not sum to 1');

[X,Y] = size(pygx);
assert(X == length(px),'number X elements in px and pygx disagree')

t = zeros(1,X);  %t(x) is H(Y | X = x)
for x = 1:X;
    t(x) = computeEntropy(pygx(x,:));
end

H = t * px(:);   %H(Y|X) = sum_x p(x)*t(x)
```

1.5 Python Source Code

Python is an open-source language that is widely available, it is pre-installed on some systems. Unfortunately, we only have a few Python examples at this time.

Command line computations:

```
$ python

>>> import math
>>> math.log(3,2)
1.5849625007211563
```

Various methods to compute entropy.

```
>>> import math
>>> px = [0.25, 0.25, 0.5]

>>> #method 1: use numpy, matrix and array library
>>> import numpy as np
>>> h = -(np.log2(px) * px).sum()
>>> print h
1.5

>>> #method 2: for loop
>>> h  = 0
>>> for p in px:
...     h = h - p * math.log(p,2)
...
>>> print h
1.5

>>> #method 3: List comprehensions
>>> h = sum([ -p * math.log(p,2) for p in px])
>>> print h
1.5
```

Create a custom function. Put the following text into a file binaryEntropyFunction.py:

```
import math

def binaryEntropyFunction(p) :
   return -p * math.log(p,2) - (1-p) * math.log(1-p,2)
```

Run a command and show the output. Use numpy to vectorize the inputs and outputs

```
>>> from binaryEntropyFunction import *
>>> import numpy as np
>>> p = np.array([0.1, 0.2, 0.3])
>>> h = np.vectorize(binaryEntropyFunction)
>>> h(p)
array([ 0.46899559,  0.72192809,  0.8812909 ])
```

1.6 Exercises

1.1 Assume that Bob does not know the result of the horse race in Section 1.1 (where the horses win with probability $\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \ldots,$), but Alice does know. How many "yes-no" questions, on average, does Bob need to ask Alice in order to identify the winning horse?

1.2 You have 9 gold ingots, but one of them is a counterfeit. The counterfeit is lighter than the others; you cannot otherwise distinguish the counterfeit. A balance with two cups is to be used; the balance will show "left cup is heavier," "right cup is heavier," or "equal."

(a) What is the minimum number of uses of the balance that is guaranteed to always determine the counterfeit? Describe a method to determine the counterfeit.

(b) Assume you have n ingots. Find a lower bound on t, the number of uses of the balance to determine the counterfeit. Use an information theory perspective.

1.3 Let X and Y be jointly distributed random variables with $\mathcal{X} = \{1, 2, 3, 4\}$ and $\mathcal{Y} = \{1, 2, 3\}$. Let the conditional probability $p_{\mathsf{X}|\mathsf{Y}}(x|y)$ be given by:

$p_{\mathsf{X}\|\mathsf{Y}}(x\|y)$	$x=1$	$x=2$	$x=3$	$x=4$
$y=1$	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{8}$	$\frac{1}{4}$
$y=2$	0	$\frac{1}{2}$	$\frac{1}{2}$	0
$y=3$	0	1	0	0

(a) Compute $H(\mathsf{X}|\mathsf{Y}=1)$, $H(\mathsf{X}|\mathsf{Y}=2)$ and $H(\mathsf{X}|\mathsf{Y}=3)$.

Now let:

$$p_{\mathsf{Y}}(y) = \begin{cases} \frac{4}{7} & \text{if } y = 1 \\ \frac{2}{7} & \text{if } y = 2 \\ \frac{1}{7} & \text{if } y = 3 \end{cases}$$

(b) Compute $H(\mathsf{X}|\mathsf{Y})$.

1.4 *Temperature vs. weather* Consider two sample spaces:

$$\mathcal{T} = \{0°\text{C}, 10°\text{C}, 20°\text{C}\}$$
$$\mathcal{W} = \{\text{snowy}, \text{rainy}, \text{sunny}\}$$

In some city, the temperature T and weather W are jointly distributed as:

$p_{\mathsf{WT}}(w,t)$	snowy	rainy	cloudy
0°C	$\frac{1}{6}$	$\frac{1}{12}$	$\frac{1}{12}$
10°C	$\frac{1}{12}$	$\frac{1}{6}$	$\frac{1}{12}$
20°C	$\frac{1}{12}$	$\frac{1}{12}$	$\frac{1}{6}$

(a) Find $H(\mathsf{T})$ and $H(\mathsf{W})$.
(b) Find $H(\mathsf{T}, \mathsf{W})$.
(c) Find $H(\mathsf{W}|\mathsf{T})$.

1.5 Let X and Z be independent random variables where X is a $(0, 1)$ binary random variable distributed as $[\frac{1}{2}, \frac{1}{2}]$, and Z be distributed as:

$$p_{\mathsf{Z}}(z) = \begin{cases} 1-p & z = 0 \\ p & z = 1 \end{cases}. \tag{1.75}$$

Now, let $\mathsf{Y} = \mathsf{X} \oplus \mathsf{Z}$, where $\oplus$ is the exclusive-or operation $0 \oplus 0 = 1 \oplus 1 = 0$ and $1 \oplus 0 = 0 \oplus 1 = 1$. Recall the binary entropy function is $h(p)$, with $H(\mathsf{Z}) = h(p)$ and $H(\mathsf{X}) = h(\frac{1}{2})$.

(a) Find $p_{\mathsf{Y}}(y)$
(b) Find $p_{\mathsf{Y}|\mathsf{X}}(y|x)$
(c) Find $H(\mathsf{Y}|\mathsf{X})$
(d) Find $H(\mathsf{Y} \oplus \mathsf{X}|\mathsf{X})$

1.6 Find the derivative $h'(p)$ of the binary entropy function $h(p)$.

1.7 Let X be a binary variable be distributed as $p_{\mathsf{X}}(0) = p_{\mathsf{X}}(1) = \frac{1}{2}$. Let Y be a binary variable distributed as $p_{\mathsf{Y}}(0) = 1 - p$ and $p_{\mathsf{Y}}(1) = p$; the variables X and Y are independent. Consider a new random variable $\mathsf{Z} = \mathsf{X} + \mathsf{Y}$ so that $\mathcal{Z} = \{0, 1, 2\}$.

(a) Find $H(\mathsf{X})$ and $H(\mathsf{Y})$
(b) Find $H(\mathsf{Z}|\mathsf{X}, \mathsf{Y})$
(c) Find $H(\mathsf{Z}|\mathsf{X} = 0)$ and $H(\mathsf{Z}|\mathsf{X} = 1)$

1.8 Let X be a discrete random variable. For any function g, show that the entropy of a $g(\mathsf{X})$ is less than or equal to the entropy of X by justifying the following steps.

$$\begin{aligned} H(\mathsf{X}, g(\mathsf{X})) &= H(\mathsf{X}) + H(g(\mathsf{X})|\mathsf{X}) && (1.94)\\ &= H(\mathsf{X}), && (1.95)\\ H(\mathsf{X}, g(\mathsf{X})) &= H(g(\mathsf{X})) + H(\mathsf{X}|g(\mathsf{X})) && (1.96)\\ &\geq H(g(\mathsf{X})). && (1.97) \end{aligned}$$

Thus, $H(g(\mathsf{X})) \leq H(\mathsf{X})$.

1.9 For jointly distributed X, Y and Z, prove $H(\mathsf{X}, \mathsf{Y}|\mathsf{Z}) = H(\mathsf{X}|\mathsf{Z}) + H(\mathsf{Y}|\mathsf{X}, \mathsf{Z})$.

1.10 You should write a program to complete this exercise. Refer to the lecture notes for an example.

Let X with $|\mathcal{X}| = 32$ be distributed as $p_{\mathsf{X}}(x)$ given as:

```
px = [ 1/256 13/256 6/256 11/256 1/256 15/256 7/256 5/256 ...
       4/256 7/256 1/256 11/256 14/256 10/256 13/256 14/256 ...
       5/256 14/256 5/256 3/256 12/256 8/256 9/256 7/256 ...
       16/256 2/256 15/256 3/256 8/256 4/256 3/256 9/256 ]
```

(a) Compute the entropy $H(\mathsf{X})$.
(b) If X was instead uniformly distributed, what is the maximum value of $H(\mathsf{X})$?

Let X, Y be jointly distributed with $|\mathcal{X}| = 4$ and $|\mathcal{Y}| = 6$, with conditional distribution $p_{\mathsf{Y}|\mathsf{X}}(y|x)$:

```
pygx = [ 3/19 3/19 2/19 3/19 4/19 4/19 ...
         4/18 3/18 4/18 3/18 3/18 1/18 ...
         1/15 4/15 1/15 4/15 2/15 3/15 ...
         4/15 2/15 4/15 1/15 1/15 3/15 ]
```

and $p_{\mathsf{X}}(x) = [\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}]$,
(c) Compute the conditional entropy $H(\mathsf{Y}|\mathsf{X})$
(d) If instead $p_{\mathsf{Y}|\mathsf{X}}(y|x)$ is conditionally uniform, what is the value of $H(\mathsf{Y}|\mathsf{X})$?

Chapter 2

Tour of Probability Theory

Probability theory is fundamental to information theory, and this chapter describes some of the basic probability theory that will be used.

2.1 Random Variables

2.1.1 Single Random Variables

A *random variable* is a variable that takes on a value that represents the outcome of a probabilistic experiment. In this chapter, only discrete random variables are considered; later continuous random variables are used. For a random variable X, probability is defined using a set $\mathcal{X}$ and a function $p_{\mathsf{X}}(x)$. The set $\mathcal{X}$ is called the *sample space* and the function $p_{\mathsf{X}}(x)$ is called the *probability distribution* or *probability mass function*. The function $p_{\mathsf{X}}(x)$ must satisfy two conditions:

$$0 \leq p_{\mathsf{X}}(x) \leq 1 \tag{2.6}$$

for all $x \in \mathcal{X}$, and,

$$\sum_{x \in \mathcal{X}} p_{\mathsf{X}}(x) = 1. \tag{2.7}$$

Sanserif type X is used for random variables, distinct from X. Calligraphic script $\mathcal{X}$ is used for sets, and the sample space is a set. Sometimes $\mathsf{X} \sim p_{\mathsf{X}}(x)$ is written to mean "X is distributed as $p_{\mathsf{X}}(x)$."

For example, let X be a random variable denoting the outcome of rolling a die with 6 sides. Then the sample space is $\mathcal{X} = \{1, 2, 3, 4, 5, 6\}$, and the probability distribution is:

$$p_{\mathsf{X}}(x) = \begin{cases} \frac{1}{6} & \text{for } x \in \{1, 2, 3, 4, 5, 6\} \\ 0 & \text{otherwise} \end{cases} \tag{2.8}$$

- Random variable: A random variable X has a probability distribution $p_{\mathsf{X}}(x)$ for x in the sample space $\mathcal{X}$, where:

$$0 \leq p_{\mathsf{X}}(x) \leq 1 \text{ and } \sum_{x \in \mathcal{X}} p_{\mathsf{X}}(x) = 1. \tag{2.1}$$

  Here, $p_{\mathsf{X}}(x)$ is $\Pr[\mathsf{X} = x]$, that is "the probability the random variable X is equal to x."

- Jointly distributed random variables X, Y have a joint probability distribution p_{XY} for $x \in \mathcal{X}$ and $y \in \mathcal{Y}$.
- Independence of X and Y means $p_{\mathsf{X},\mathsf{Y}}(x, y) = p_{\mathsf{X}}(x)p_{\mathsf{Y}}(y)$.
- Definition of conditional probability:

$$p_{\mathsf{X}|\mathsf{Y}}(x|y) = \frac{p_{\mathsf{XY}}(x, y)}{p_{\mathsf{Y}}(y)}. \tag{2.2}$$

- Bayes rule:

$$p_{\mathsf{X}|\mathsf{Y}}(x|y) = \frac{p_{\mathsf{Y}|\mathsf{X}}(y|x)p_{\mathsf{X}}(x)}{p_{\mathsf{Y}}(y)} \tag{2.3}$$

- Expected value

$$E[\mathsf{X}] = \sum_{x \in \mathcal{X}} x p_{\mathsf{X}}(x). \tag{2.4}$$

- Expected value of a function $g(x)$:

$$E[g(\mathsf{X})] = \sum_{x \in \mathcal{X}} g(x) p_{\mathsf{X}}(x). \tag{2.5}$$

- Marginalization: $p_{\mathsf{X}}(x) = \sum_{y \in \mathcal{Y}} p_{\mathsf{XY}}(x, y)$
- Theorem of total probability: $p_{\mathsf{X}}(x) = \sum_{y \in \mathcal{Y}} p_{\mathsf{X}|\mathsf{Y}}(x|y)p_{\mathsf{Y}}(y)$
- Independent and identically distributed, i.i.d.: for a sequence of n i.i.d. random variables $\mathsf{X}_1, \mathsf{X}_2, \ldots, \mathsf{X}_n$ have distribution $p_{\mathsf{X}1}(x_1)p_{\mathsf{X}2}(x_2) \cdots p_{\mathsf{X}n}(x_n)$.

Figure 2.1: Summary of important random variable relationships.

Another example is if Z represents the outcome of flipping a coin, which has heads and tails, each with probability $\frac{1}{2}$. Then $\mathcal{Z} = \{\text{heads}, \text{tails}\}$ and $p_{\mathsf{Z}}(\text{heads}) = p_{\mathsf{Z}}(\text{tails}) = \frac{1}{2}$.

A discrete random variable X has real, discrete values, and so the sample space $\mathcal{X}$ is likewise discrete, and often $\mathcal{X}$ is a subset of the integers. The probability distribution can be written is various ways:

$$\Pr(\mathsf{X} = x) \text{ or } \; p(x) \text{ or } \; p_{\mathsf{X}}(x) \tag{2.9}$$

for $x \in \mathcal{X}$. Note that X is a random variable, while x is a constant. The term $\Pr(\mathsf{X} = x)$ and $p_{\mathsf{X}}(x)$ show both X and x, but sometimes we use $p(x)$ to save space. Writing $\Pr(\mathsf{X})$ is undesirable (without the "$= x$"), because it does not explicitly show the dependence on x.

Often $p_{\mathsf{X}}(x)$ is written as a vector, for example:

$$p_{\mathsf{X}}(x) = [\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{8}], \tag{2.10}$$

which is understood to mean,

$$p_{\mathsf{X}}(x) = \begin{cases} \frac{1}{2} & x = 1 \\ \frac{1}{4} & x = 2 \\ \frac{1}{8} & x = 3 \\ \frac{1}{8} & x = 4 \end{cases} \tag{2.11}$$

when $\mathcal{X}$ is are integers $\{1, 2, 3, 4\}$ or $\mathcal{X}$ is otherwise not important.

Example 2.1. A *binary random variable*[1] X, with parameter p, has a sample space with two values $\mathcal{X} = \{0, 1\}$ and has value 1 with probability p. The probability distribution is:

$$\begin{aligned} p_{\mathsf{X}}(0) &= \Pr(\mathsf{X} = 0) = 1 - p \\ p_{\mathsf{X}}(1) &= \Pr(\mathsf{X} = 1) = p. \end{aligned} \tag{2.12}$$

Alternatively, $p_{\mathsf{X}}(x) = [1 - p, p]$ is sometimes used.

2.1.2 Jointly Distributed Random Variables

Jointly distributed random variables X and Y are defined for sample spaces $\mathcal{X}$ and $\mathcal{Y}$ respectively, with a joint probability distribution:

$$p_{\mathsf{XY}}(x, y) = \Pr(\mathsf{X} = x, \mathsf{Y} = y). \tag{2.13}$$

Again, we almost always write $p_{\mathsf{XY}}(x, y)$ or $p(x, y)$ instead of $\Pr(\mathsf{X} = x, \mathsf{Y} = y)$. The joint probability distribution satisfies

$$0 \leq p_{\mathsf{XY}}(x, y) \leq 1, \tag{2.14}$$

[1] often called a Bernoulli random variable

Matrix representation of probability functions and other functions Joint probability distributions and other functions can be represented as matrices. In general, the matrix rows represent the first variable of the probability distribution, and the matrix columns represent the second variable. For example,

$$p_{\mathsf{XZ}}(x,z) = \begin{bmatrix} \frac{1}{21} & \frac{2}{21} & \frac{3}{21} \\ \frac{4}{21} & \frac{5}{21} & \frac{6}{21} \end{bmatrix} \tag{2.18}$$

means that X and Z are jointly distributed, that $p_{\mathsf{XZ}}(2,1) = \Pr(\mathsf{X}=2,\mathsf{Z}=1) = \frac{4}{21}$, and that $|\mathcal{X}| = 2$ and $|\mathcal{Z}| = 3$. Usually, you can assume that $\mathcal{X} = \{1,2\}$ and $\mathcal{Z} = \{1,2,3\}$.

The exception is conditional probability distributions. Conditional probability distributions are written so the rows sum to one:

$$p_{\mathsf{Y}|\mathsf{X}}(y|x) = \begin{bmatrix} \frac{1}{3} & \frac{2}{3} \\ \frac{1}{4} & \frac{3}{4} \\ \frac{1}{5} & \frac{4}{5} \end{bmatrix} \tag{2.19}$$

means that $p_{\mathsf{Y}|\mathsf{X}}(2|3) = \Pr(\mathsf{Y}=2|\mathsf{X}=3) = \frac{4}{5}$, and that $|\mathcal{X}| = 3$ and $|\mathcal{Y}| = 2$. See also Example 2.5 for probability computation using matrices.

for all $x \in \mathcal{X}$ and all $y \in \mathcal{Y}$ and:

$$\sum_{x\in\mathcal{X}} \sum_{y\in\mathcal{Y}} p_{\mathsf{XY}}(x,y) = 1 \tag{2.15}$$

The *marginal distribution* of X is:

$$p_{\mathsf{X}}(x) = \sum_{y\in\mathcal{Y}} p_{\mathsf{XY}}(x,y) \tag{2.16}$$

and the marginal distribution of Y is:

$$p_{\mathsf{Y}}(y) = \sum_{x\in\mathcal{X}} p_{\mathsf{XY}}(x,y) \tag{2.17}$$

Example 2.2. Consider the jointly distributed random variables X and Y, with $\mathcal{X} = \{1,2,3\}$, $\mathcal{Y} = \{1,2,3\}$ and probability distribution given in matrix form:

$$p_{\mathsf{XY}}(x,y) = \begin{bmatrix} \frac{1}{2} & 0 & 0 \\ \frac{1}{4} & \frac{1}{16} & 0 \\ 0 & \frac{1}{8} & \frac{1}{16} \end{bmatrix}, \tag{2.20}$$

where the rows correspond to values of x and columns correspond to values of y, so $p_{\mathsf{XY}}(2,1) = \frac{1}{4}$. Then the marginal distribution $p_{\mathsf{X}}(x)$ is:

$$p_{\mathsf{X}}(x) = \sum_{y=1}^{3} p_{\mathsf{XY}}(x,y), \tag{2.21}$$

Understanding joint distributions The following is a perspective on joint distributions. Consider $\mathcal{X} = \{\text{apple}, \text{banana}\}$ and $\mathcal{Y} = \{3, 5\}$ and the joint distribution $p_{\mathsf{XY}}(x, y)$ is:

$p_{\mathsf{XY}}(x,y)$	3	5
apple	$\frac{2}{14}$	$\frac{3}{14}$
banana	$\frac{4}{14}$	$\frac{5}{14}$

A joint distribution is the probability that two random variables have a particular pair of outcomes. For example, the probability that $\mathsf{X} = \text{banana}$ and $\mathsf{Y} = 3$ *at the same time* is 4/14. It is sometimes convenient to think of the pair (X, Y) as a single random variable Z on the sample space:

$$\mathcal{Z} = \{(\text{apple}, 3), (\text{apple}, 5), (\text{banana}, 3), (\text{banana}, 5)\}, \tag{2.22}$$

where $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ is the Cartesian product. The probability distribution on Z is:

$$p_{\mathsf{Z}}(z) = \begin{cases} \frac{2}{14} & \text{if } z = (\text{apple, } 3) \\ \frac{3}{14} & \text{if } z = (\text{apple, } 5) \\ \frac{4}{14} & \text{if } z = (\text{banana, } 3) \\ \frac{5}{14} & \text{if } z = (\text{banana, } 5) \end{cases} \tag{2.23}$$

The random variable Z and the random variable pair (X, Y) are connected in this way.

Above, the example used a *Cartesian product* $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$, sometimes called a *product set*. As another example, the Cartesian product $\mathcal{X}^n$ of $\mathcal{X} = \{0, 1\}$ is the set of all binary sequences of length n, for example $\mathcal{X}^3 = \{000, 001, 010, 011, 100, 101, 110, 111\}$. Wikipedia: Cartesian product

which has values $\frac{1}{2}, \frac{5}{16}, \frac{3}{16}$. Similarly, the marginal distribution $p_{\mathsf{Y}}(y)$ has values $\frac{3}{4}, \frac{3}{16}, \frac{1}{16}$. □

2.1.3 Conditional Probability Distributions

For two jointly distributed random variables X and Y, the *conditional distribution of* X *given* Y is defined as:

$$p_{\mathsf{X}|\mathsf{Y}}(x|y) = \Pr(\mathsf{X} = x|\mathsf{Y} = y) = \frac{\Pr(\mathsf{X} = x, \mathsf{Y} = y)}{\Pr(\mathsf{Y} = y)}. \tag{2.24}$$

The term $p_{\mathsf{X}|\mathsf{Y}}(x|y)$ or $\Pr(\mathsf{X} = x|\mathsf{Y} = y)$ is read as "the probability X is equal to x, given Y is equal to y."

A property that we often use is that for any value of y,

$$\sum_{x \in \mathcal{X}} p_{\mathsf{X}|\mathsf{Y}}(x|y) = 1 \tag{2.25}$$

must hold.

Example 2.3. Consider again the joint distribution in (2.20). Compute $p_{\mathsf{Y}|\mathsf{X}}(2|3)$, that is, $\Pr(\mathsf{Y}=2|\mathsf{X}=3)$:

$$p_{\mathsf{Y}|\mathsf{X}}(2|3) = \frac{p_{\mathsf{X},\mathsf{Y}}(3,2)}{p_{\mathsf{X}}(3)} = \frac{1/8}{3/16} = \frac{2}{3} \tag{2.26}$$

The conditional distribution $p_{\mathsf{Y}|\mathsf{X}}(y|x)$ can be written in matrix form:

$$p_{\mathsf{Y}|\mathsf{X}}(y|x) = \begin{bmatrix} 1 & 0 & 0 \\ \frac{4}{5} & \frac{1}{5} & 0 \\ 0 & \frac{2}{3} & \frac{1}{3} \end{bmatrix}. \tag{2.27}$$

Conditional probability distributions matrix is usually written so the rows sum to 1 (rather than the columns summing to 1).

2.1.4 Bayes' Rule, Total Probability, All-Knowing Joint Distribution

Bayes' rule (or Bayes' theorem) states:

$$p_{\mathsf{X}|\mathsf{Y}}(x|y) = \frac{p_{\mathsf{Y}|\mathsf{X}}(y|x)p_{\mathsf{X}}(x)}{p_{\mathsf{Y}}(y)}, \tag{2.28}$$

and is easily proved by observing that $p_{\mathsf{XY}}(x,y)$ can be written two ways $p_{\mathsf{XY}}(x,y) = p_{\mathsf{X}|\mathsf{Y}}(x|y)p_{\mathsf{Y}}(y)$ and $p_{\mathsf{XY}}(x,y) = p_{\mathsf{Y}|\mathsf{X}}(y|x)p_{\mathsf{X}}(x)$.

The *theorem of total probability* or *law of total probability* is related to marginalization (2.16), (2.17). For X,Y jointly distributed as $p_{\mathsf{XY}}(x,y)$:

$$p_{\mathsf{X}}(x) = \sum_{y\in\mathcal{Y}} p_{\mathsf{X}|\mathsf{Y}}(x|y)p_{\mathsf{Y}}(y), \tag{2.29}$$

obtained using the definition of conditional probability.

The theorem of total probability has various forms. For example, if X,Y and Z are jointly distributed, then:

$$p_{\mathsf{Y}|\mathsf{X}}(y|x) = \sum_{z\in\mathcal{Z}} p_{\mathsf{Y}|\mathsf{XZ}}(y|x,z)p_{\mathsf{Z}|\mathsf{X}}(z|x) \tag{2.30}$$

is also a useful form of the theorem of total probability.

For two random variables, the joint distribution $p_{\mathsf{XY}}(x,y)$ is the "all-knowing distribution." This means that given $p_{\mathsf{XY}}(x,y)$, it is possible to compute $p_{\mathsf{X}}(x)$, $p_{\mathsf{Y}}(y)$, $p_{\mathsf{X}|\mathsf{Y}}(x|y)$ and $p_{\mathsf{Y}|\mathsf{X}}(y|x)$. If you are given only $p_{\mathsf{Y}|\mathsf{X}}(y|x)$, then you additionally need $p_{\mathsf{X}}(x)$ to find the joint distribution using $p_{\mathsf{XY}}(x,y) = p_{\mathsf{Y}|\mathsf{X}}(y|x)p_{\mathsf{X}}(x)$. In fact, sometimes $p_{\mathsf{Y}|\mathsf{X}}(y|x)p_{\mathsf{X}}(x)$ is called the "joint distribution."

The phrase "X,Y are jointly distributed as $p_{\mathsf{XY}}(x,y)$" simply means that X and Y are random variables that have a known joint distribution, and they are not necessarily independent.

2.1.5 Discrete Memoryless Channel

An important example of a conditional probability distribution is a discrete memoryless channel, or DMC. A discrete memoryless channel is a model of a communications channel.

Definition 2.1. A *discrete memoryless channel* (DMC) consists of an input alphabet $\mathcal{X}$, an output alphabet $\mathcal{Y}$ and a conditional probability distribution $p_{\mathsf{Y}|\mathsf{X}}(y|x)$.

As a model of a communications channel, when a symbol x from an alphabet $\mathcal{X}$ is transmitted, another symbol y from an alphabet $\mathcal{Y}$ is received. Given that x was transmitted, the probability that y was received is given by the conditional distribution $p_{\mathsf{Y}|\mathsf{X}}(y|x)$ which specifies the DMC; x is called the channel input, y is called the channel output.

Example 2.4. The binary symmetric channel (BSC) is an important example of a communications channel. The BSC has input $\mathcal{X} = \{0, 1\}$ and output $\mathcal{Y} = \{0, 1\}$. The channel transition probabilities $p_{\mathsf{Y}|\mathsf{X}}(y|x)$ are:

$$p_{\mathsf{Y}|\mathsf{X}}(y|x) = \begin{bmatrix} 0.8 & 0.2 \\ 0.2 & 0.8 \end{bmatrix}. \tag{2.31}$$

This is an example of the *binary symmetric channel* (BSC). If the transmitter sends $x = 0$, then $y = 0$ is correctly received with probability 0.8. An error occurs if $y = 1$ is received; this occurs with probability 0.2. The probabilities on y are reversed if $x = 1$ is transmitted. The following is an example of some transmitted x and received y:

$x =$	1	1	1	1	1	0	1	0	0	0	1	0	1	1	0	1	0	0	0	1
$y =$	1	1	1	1	0	1	1	1	0	0	1	0	1	0	1	1	0	0	0	1
					E	E		E						E	E					

The errors are marked by "E."

Example 2.5. Consider a discrete memoryless channel given by conditional probability distribution:

$$p_{\mathsf{Y}|\mathsf{X}}(y|x) = \begin{bmatrix} \frac{2}{3} & \frac{1}{3} & 0 \\ 0 & \frac{2}{3} & \frac{1}{3} \\ \frac{1}{3} & 0 & \frac{2}{3} \end{bmatrix}, \tag{2.32}$$

where the rows are x and the columns are y. Given an input distribution $p_{\mathsf{X}}(x) = [\frac{1}{2}, \frac{1}{4}, \frac{1}{4}]$, compute the output distribution $p_{\mathsf{Y}}(y)$.

We have

$$p_{\mathsf{Y}}(y) = \sum_{x \in \mathcal{X}} p_{\mathsf{XY}}(x, y) = \sum_{x \in \mathcal{X}} p_{\mathsf{Y}|\mathsf{X}}(y|x) p_{\mathsf{X}}(x) \tag{2.33}$$

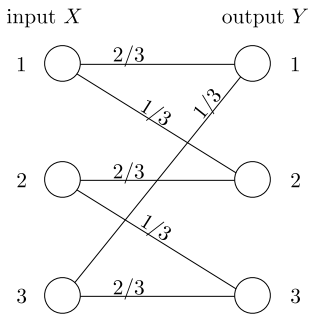


Figure 2.2: Example of a discrete memoryless channel, or DMC, with input X and output Y.

From this, find:

$$p_{\mathsf{Y}}(1) = \frac{2}{3}\cdot\frac{1}{2} + \quad 0 \quad + \frac{1}{3}\cdot\frac{1}{4} = \frac{5}{12} \tag{2.34}$$

$$p_{\mathsf{Y}}(2) = \frac{1}{3}\cdot\frac{1}{2} + \frac{2}{3}\cdot\frac{1}{4} + \quad 0 \quad = \frac{4}{12} \tag{2.35}$$

$$p_{\mathsf{Y}}(3) = \quad 0 \quad + \frac{1}{3}\cdot\frac{1}{4} + \frac{2}{3}\cdot\frac{1}{4} = \frac{3}{12} \tag{2.36}$$

The computation can also be performed in matrix form:

$$\begin{bmatrix}\frac{1}{2} & \frac{1}{4} & \frac{1}{4}\end{bmatrix}\cdot\begin{bmatrix}\frac{2}{3} & \frac{1}{3} & 0\\ 0 & \frac{2}{3} & \frac{1}{3}\\ \frac{1}{3} & 0 & \frac{2}{3}\end{bmatrix} = \begin{bmatrix}\frac{5}{12} & \frac{4}{12} & \frac{3}{12}\end{bmatrix}. \tag{2.37}$$

SSQ 2.1. Consider a discrete memoryless channel with input X and output Y. The input alphabet is $\mathcal{X} = \{1, 2\}$, output alphabet is $\mathcal{Y} = \{1, 2, 3, 4, 5\}$, with $p_{\mathsf{X}}(1) = \frac{1}{4}$ and $p_{\mathsf{X}}(2) = \frac{3}{4}$, and $p_{\mathsf{Y}|\mathsf{X}}(y|x)$:

$$p_{\mathsf{Y}|\mathsf{X}}(y|x) = \begin{bmatrix}1/2 & 1/4 & 0 & 1/8 & 1/8\\ 0 & 1/2 & 1/4 & 0 & 1/4\end{bmatrix} \tag{2.38}$$

Compute the output distribution, $p_{\mathsf{Y}}(y)$ for $y = 1, 2, 3, 4, 5$.

2.2 Independence, Expected Value and Variance

2.2.1 Independence and Conditional Independence

Let two random variables X and Y have a joint distribution $p_{\mathsf{XY}}(x, y)$.

Definition 2.2. Two random variables X and Y are *independent* if and only if:

$$p_{\mathsf{XY}}(x, y) = p_{\mathsf{X}}(x)p_{\mathsf{Y}}(y). \tag{2.39}$$

Also, X and Y are independent if an only if

$$p_{\mathsf{X}|\mathsf{Y}}(x|y) = p_{\mathsf{X}}(x), \tag{2.40}$$

for all $x \in \mathcal{X}, y \in \mathcal{Y}$. This is easily obtained using $p_{\mathsf{XY}}(x,y) = p_{\mathsf{X}|\mathsf{Y}}(x|y)p_{\mathsf{Y}}(y)$.

If $p_{\mathsf{X}}(x) = p_{\mathsf{Y}}(x)$ for all $x \in \mathcal{X}$, then we say X and Y are *independent and identically distributed*, often abbreviated iid.

Definition 2.3. Let three variables X, Y and Z be jointly distributed. Then, X and Y are *conditionally independent given* Z, if the following holds:

$$p_{\mathsf{XY}|\mathsf{Z}}(x,y|z) = p_{\mathsf{X}|\mathsf{Z}}(x|z) \cdot p_{\mathsf{Y}|\mathsf{Z}}(y|z) \tag{2.41}$$

for all $x \in \mathcal{X}$, for all $y \in \mathcal{Y}$ and all $z \in \mathcal{Z}$.

Even if X and Y are not independent, it is possible that X and Y are conditionally independent.

2.2.2 Expected Value

Definition 2.4. The *expected value* $E[\mathsf{X}]$, of a random variable X with probability distribution $p_{\mathsf{X}}(x)$ is:

$$E[\mathsf{X}] = \sum_{x \in \mathcal{X}} x p_{\mathsf{X}}(x) \tag{2.42}$$

The expected value $E[\mathsf{X}]$ is sometimes called the mean.

Example 2.6. The expected value $E[\mathsf{X}]$ of the binary random variable X given in Example 2.1 on page 35 (2.12) is:

$$E[\mathsf{X}] = 0 \cdot (1-p) + 1 \cdot p = p. \tag{2.43}$$

Example 2.7. If X is rolling a die, then $E[\mathsf{X}]$ is given by:

$$\begin{aligned} E[\mathsf{X}] &= \sum_{x=1}^{6} p_{\mathsf{X}}(x)x && (2.44)\\ &= \frac{1}{6}\cdot 1 + \frac{1}{6}\cdot 2 + \frac{1}{6}\cdot 3 + \frac{1}{6}\cdot 4 + \frac{1}{6}\cdot 5 + \frac{1}{6}\cdot 6 && (2.45)\\ &= 3.5 && (2.46) \end{aligned}$$

The expectation of a deterministic function g of a random variable X is:

$$E[g(\mathsf{X})] = \sum_{x \in \mathcal{X}} g(x) p_{\mathsf{X}}(x) \tag{2.47}$$

Example 2.8. Find $E[\mathsf{X}^2]$ when X is the outcome of a die roll:

$$E[\mathsf{X}^2] = \sum_{x=1}^{6} p_{\mathsf{X}}(x)x^2 \tag{2.48}$$

$$= \frac{1}{6}\cdot 1^2 + \frac{1}{6}\cdot 2^2 + \frac{1}{6}\cdot 3^2 + \frac{1}{6}\cdot 4^2 + \frac{1}{6}\cdot 5^2 + \frac{1}{6}\cdot 6^2 \tag{2.49}$$

$$= \frac{91}{6} \tag{2.50}$$

Example 2.9. Let $\mathsf{X} \sim p_{\mathsf{X}}(x) = [\frac{1}{2}, \frac{1}{4}, \frac{1}{4}]$, and let $g(x) = -\log p_{\mathsf{X}}(x)$. Find $E[g(\mathsf{X})]$:

$$E[g(\mathsf{X})] = -\sum_{x=1}^{3} p_{\mathsf{X}}(x)\log p_{\mathsf{X}}(x) \tag{2.51}$$

$$= -\frac{1}{2}\log\frac{1}{2} - 2\Big(\frac{1}{4}\log\frac{1}{4}\Big) = \frac{3}{2}. \tag{2.52}$$

Definition 2.5. The *variance* of a random variable X, denoted $\mathrm{Var}[\mathsf{X}]$, is:

$$\mathrm{Var}[\mathsf{X}] = E[\mathsf{X}^2] - (E[\mathsf{X}])^2. \tag{2.53}$$

Example 2.10. The variance $\mathrm{Var}[\mathsf{X}]$ of the die roll random variable X is:

$$\mathrm{Var}[\mathsf{X}] = E[\mathsf{X}^2] - (E[\mathsf{X}])^2 \tag{2.54}$$

$$= \frac{91}{6} - \left(\frac{7}{2}\right)^2 = \frac{35}{12} \tag{2.55}$$

For two jointly distributed random variables X and Y, the *conditional expectation* $E[\mathsf{X}|\mathsf{Y}=y]$ is:

$$E[\mathsf{X}|\mathsf{Y}=y] = \sum_{x\in\mathcal{X}} x\, p_{\mathsf{X}|\mathsf{Y}}(x|y) \tag{2.56}$$

The conditional expectation $E[\mathsf{X}|\mathsf{Y}=y]$ is a function of y, for example, $f(y) = E[\mathsf{X}|\mathsf{Y}=y]$. Note that $E[\mathsf{X}|\mathsf{Y}]$ is distinct from $E[\mathsf{X}|\mathsf{Y}=y]$, in particular $E[\mathsf{X}|\mathsf{Y}]$ is a random variable equal to $f(\mathsf{Y})$. Since $E[\mathsf{X}|\mathsf{Y}]$ is a random variable, it has an expectation. In fact, it is equal to $E[\mathsf{X}]$:

Proposition 2.1. *Law of Total Expectation* Let X and Y be jointly distributed random variables. Then:

$$E\big[E[\mathsf{X}|\mathsf{Y}]\big] = E[\mathsf{X}]. \tag{2.57}$$

Expectation and Variance of Sums of Random Variables For any $\mathsf{X}_1, \mathsf{X}_2, \ldots, \mathsf{X}_n$ and constants $a_1, a_2, \ldots, a_n$,

$$E[a_1\mathsf{X}_1 + a_2\mathsf{X}_2 + \cdots a_n\mathsf{X}_n] = a_1E[\mathsf{X}_1] + a_2E[\mathsf{X}_2] + \cdots a_nE[\mathsf{X}_n] \tag{2.58}$$

even if the X_i are not independent.

For any independent $\mathsf{X}_1, \mathsf{X}_2, \ldots, \mathsf{X}_n$ and constants $a_1, a_2, \ldots, a_n$:

$$\mathrm{Var}[a_1\mathsf{X}_1 + a_2\mathsf{X}_2 + \cdots a_n\mathsf{X}_n] = a_1^2\mathrm{Var}[\mathsf{X}_1] + a_2^2\mathrm{Var}[\mathsf{X}_2] + \cdots a_n^2\mathrm{Var}[\mathsf{X}_n] \tag{2.59}$$

2.2.3 Events and Their Union Bound

In addition to probability distributions, we also deal with probability of events. An event is a a binary random variable, which is either true or false. An event is an outcome from an experiment, to which a probability may be assigned. If E is an event that can either occur or not occur, then $\Pr(\mathsf{E})$ is "the probability that the event E occurs." Examples of events are "a fair coin toss is heads" and "a die roll shows 5 or 6". The probability of these two events are 0.5 and $\frac{1}{3}$, respectively. Wikipedia: Event

Let $\mathsf{E}_1, \mathsf{E}_2, \ldots, \mathsf{E}_n$ be events. The union $\mathsf{E}_1 \cup \mathsf{E}_2 \cup \cdots \cup \mathsf{E}_n$ is the event that at least one of $\mathsf{E}_1, \mathsf{E}_2, \ldots, \mathsf{E}_n$ occurs. The union bound is an upper bound on at least one of the E_i occurring:

$$\Pr(\bigcup_{i=1}^{n} \mathsf{E}_i) \leq \sum_{i=1}^{n} \Pr(\mathsf{E}_i) \tag{2.60}$$

For $n = 2$, we have:

$$\Pr(\mathsf{E}_1 \cup \mathsf{E}_2) \leq \Pr(\mathsf{E}_1) + \Pr(\mathsf{E}_2) \tag{2.61}$$

since $\Pr(\mathsf{E}_1 \cup \mathsf{E}_2) = \Pr(\mathsf{E}_1) + \Pr(\mathsf{E}_2) - \Pr(\mathsf{E}_1 \cap \mathsf{E}_2)$.

Example 2.11. For example, is some communication system, let E_1 be the event that an error of one type occurs with $\Pr(\mathsf{E}_1) = 0.1$. Let E_2 be the event that another type of error occurs with $\Pr(\mathsf{E}_2) = 0.05$. If either E_1or E_2 occurs, then the system fails, indicated by an event E, that is $\mathsf{E} = \mathsf{E}_1 \cup \mathsf{E}_2$. Find an upper bound on $\Pr(\mathsf{E})$ using the union bound:

$$\begin{aligned} \Pr(\mathsf{E}) &= \Pr(\mathsf{E}_1 \cup \mathsf{E}_2) && (2.62)\\ &\leq \Pr(\mathsf{E}_1) + \Pr(\mathsf{E}_2) && (2.63)\\ \Pr(\mathsf{E}) &\leq 0.15 && (2.64) \end{aligned}$$

The union bound is useful, because we do not need to know the correlation between E_1 and E_2 to obtain an upper bound on $\Pr(\mathsf{E})$.

Probability of exclusive events. Let E_1 and E_2 be two exclusive events. Then:

$$\Pr(\mathsf{E}_1 \cup \mathsf{E}_2) = \Pr(\mathsf{E}_1) + \Pr(\mathsf{E}_2), \tag{2.65}$$

or more generally if all $\mathsf{E}_1, \mathsf{E}_2, \ldots, \mathsf{E}_n$ are exclusive:

$$\Pr(\bigcup_{i=1}^{n} \mathsf{E}_i) = \sum_{i=1}^{n} \Pr(\mathsf{E}_i) \tag{2.66}$$

Since E_1 and E_2 are exclusive, $\Pr(\mathsf{E}_1 \cap \mathsf{E}_2) = 0$.

For example, if X is the outcome of rolling a fair die, and E_i is the event the die shows i, that is $\mathsf{X} = i$, then the probability that X is odd is:

$$\begin{aligned} \Pr(\mathsf{X} \text{ is odd}) &= \Pr(\mathsf{E}_1 \cup \mathsf{E}_3 \cup \mathsf{E}_5) && (2.67)\\ &= \Pr(\mathsf{E}_1) + \Pr(\mathsf{E}_3) + \Pr(\mathsf{E}_5) && (2.68)\\ &= \frac{1}{6} + \frac{1}{6} + \frac{1}{6} = \frac{1}{2} && (2.69) \end{aligned}$$

That is, the probability of exclusive events can be calculated exactly.

2.3 Random Vectors

2.3.1 Random Vectors

A random variable X_i has distribution $p_{\mathsf{X}}(x)$ on sample space $\mathcal{X}$. Then, a *random vector* $\mathbf{X}$ is a sequence of n random variables:

$$\mathbf{X} = \big(\mathsf{X}_1, \mathsf{X}_2, \mathsf{X}_3, \ldots, \mathsf{X}_n\big). \tag{2.70}$$

The sample space of $\mathbf{X}$ is the Cartesian product $\mathcal{X}^n$. Usually, the sequence of random variables are independent and identically distributed.

Let $p_{\mathbf{X}}(\mathbf{x})$ be the joint distribution of the vector $\mathbf{X}$. Assuming the X_i are independent, the joint distribution is:

$$p_{\mathbf{X}}(\mathbf{x}) = p_{\mathsf{X}_1}(x_1) p_{\mathsf{X}_2}(x_2) \cdots p_{\mathsf{X}_n}(x_n) \tag{2.71}$$

and since the X_i are identically distributed,

$$p_{\mathbf{X}}(\mathbf{x}) = \prod_{i=1}^{n} p_{\mathsf{X}}(x), \tag{2.72}$$

where $p_{\mathsf{X}}(x)$ is the distribution of the random variable X.

Definition 2.6. Let $\mathbf{X} = \mathsf{X}_1 \mathsf{X}_2 \cdots, \mathsf{X}_n$ be a random vector of n random variables, independent and identically distributed. The *sample mean* $\overline{\mathsf{X}}_n$ is:

$$\overline{\mathsf{X}}_n = \frac{1}{n} \sum_{i=1}^{n} \mathsf{X}_i \tag{2.73}$$

The sample mean $\overline{\mathsf{X}}_n$ is also a random variable. In order to contrast with the sample mean, we call $E[\mathsf{X}_i]$ the *true mean*. The sample mean $\overline{\mathsf{X}}_n$ may be

close to the true mean $E[\mathsf{X}_i]$, but is not necessarily the same. Note that $E[\overline{\mathsf{X}}_n]$ is equal to $E[\mathsf{X}_i]$, easily shown using (2.58).

Binary random vectors If X_i is a binary random variable $\mathcal{X} = \{0, 1\}$, with probability of a one equal to $0 \leq p \leq 1$, as given in (2.12), then we say $\mathbf{X}$ is a *binary random vector*. The binary random vector $\mathbf{X}$ has parameters n and p, and is the outcome of n of a sequence of n yes/no experiments. The sample space is $\mathcal{X}^n = \{0, 1\}^n$.

The expected value $E[\overline{\mathsf{X}}_n]$ of the binary random vector is:

$$E[\overline{\mathsf{X}}_n] = p, \tag{2.74}$$

which is the same as the expected value for any element $E[\mathsf{X}_i] = p$.

Example 2.12. Let $\mathbf{X}$ be a binary random vector with $n = 15$ and $p = \frac{1}{3}$.The mean and variance of $\overline{\mathsf{X}}_n$ are:

$$E[\overline{\mathsf{X}}_n] = \frac{1}{3} \tag{2.75}$$

$$\text{Var}[\overline{\mathsf{X}}_n] = \frac{p(1-p)}{n} = 0.0148. \tag{2.76}$$

What is the average number of one's you expect in any realization of $\mathbf{X}$? You should expect this value to be $\frac{1}{3}$, but if you take an actual realization of $\mathbf{X}$, for example:

$$\mathbf{x} = 0\ 0\ 0\ 0\ 0\ 1\ 1\ 1\ 1\ 1\ 0\ 1\ 0\ 0\ 0 \tag{2.77}$$

there are 6 ones, so the average number of ones is $\frac{6}{15} = 0.4$, which is close to, but not equal to $E[\mathsf{X}] = \frac{1}{3}$.

As shown in this example, the sample mean is calculated for a specific sequence. On the other hand, the true mean in calculated using $p_{\mathsf{X}}(x)$.

The binary random vector $\mathbf{X}$ has z ones and $n - z$ zeros. Let Z be the sum of X:

$$\mathsf{Z} = \sum_{i=1}^{n} \mathsf{X}_i \tag{2.78}$$

so that Z is a random variable expressing the number of ones in $\mathbf{X}$. This is the *binomial random variable*. Wikipedia: Binomial random variable

Definition 2.7. A *binomial random variable* Z with parameters n and p has sample space $\mathcal{Z} = \{0, 1, \ldots, n\}$ and probability distribution:

$$p_{\mathsf{Z}}(z) = \binom{n}{z} p^z (1-p)^{n-z} \quad \text{for } z = 0, 1, \ldots, n, \tag{2.79}$$

where $\binom{n}{z} = \frac{n!}{z!(n-z)!}$. Here $\binom{n}{z}$ is the number of distinct ways to place z ones into n positions.

2.3.2 Binary Random Vector Example

Consider an example of a binary random vector with $n = 4$ and $p = \frac{1}{4}$. The sample space is:

$$\mathcal{X}^4 = \{0000, 0001, 0010, \ldots, 1111\}. \tag{2.80}$$

Now we answer questions (a) what is the probability of $\mathbf{X} = 0110$? (b) What is the probability that $\mathbf{X}$ has two 1's and two 0's. (c) What is the probability that $\mathbf{X}$ has one or two 1's?

(a) The probability of $\mathbf{x} = 0110$ is $\Pr[\mathbf{X} = 0110] = p_{\mathbf{X}}(0110)$, computed using (2.72):

$$\begin{aligned} p_{\mathbf{X}}(0110) &= p_{\mathsf{X}}(0)p_{\mathsf{X}}(1)p_{\mathsf{X}}(1)p_{\mathsf{X}}(0) && (2.81)\\ &= (1-p)\cdot p \cdot p \cdot (1-p) = p^2(1-p)^2 && (2.82)\\ &= \frac{9}{256} \approx 0.03515 && (2.83) \end{aligned}$$

Any other sequence with two ones and two zeros, for example 1010, will also occur with probability $\frac{9}{256}$.

(b) The sequences with two ones and two zeros are:

$$\{1100, 1010, 1001, 0110, 0101, 0011\} \tag{2.84}$$

There are $\binom{4}{2} = 6$ sequences, and so the probability of any of them occurring is $6 \cdot \frac{9}{256} = \frac{27}{128}$. This is the same as the binomial random variable with $\mathsf{Z} = 2$, so using (2.79):

$$p_{\mathsf{Z}}(2) = \binom{4}{2} p^2(1-p)^2 = \frac{27}{128}. \tag{2.85}$$

The event "$\mathbf{X}$ is any sequence with z ones" is the same as the event "$\mathsf{Z} = z$."

(c) The event $\mathbf{X}$ has one 1 and the event $\mathbf{X}$ has two 1's are mutually exclusive, so:

$$\begin{aligned} \Pr(\mathbf{X} \text{ has one } 1 \cup \mathbf{X} \text{ has two 1's}) &= \Pr(\mathbf{X} \text{ has one } 1) + \Pr(\mathbf{X} \text{ has two 1's})\\ &= p_{\mathsf{Z}}(1) + p_{\mathsf{Z}}(2)\\ &= \sum_{z=1}^{2} \binom{n}{z} p^z (1-p)^{n-z}\\ &= \frac{27}{64} + \frac{27}{128} = \frac{81}{128}. \end{aligned}$$

2.4 Law of Large Numbers

A central result of information theory, Shannon's channel coding theorem (Chapter 8), is a clever application of the law of large numbers. This section reviews the law of large numbers.

2.4.1 Markov inequality

For a single random variable, the Markov inequality is an upper bound on the probability that a random variable is greater than or equal to some positive constant. Markov's inequality relates probabilities to expectations, and provides bounds for the cumulative distribution function[2] of a random variable. The Markov inequality is often not a tight bound, but nonetheless useful.

Proposition 2.2. *Markov Inequality* If X is any nonnegative random variable and $a > 0$, then:

$$\Pr(\mathsf{X} \geq a) \leq \frac{E[\mathsf{X}]}{a} \tag{2.86}$$

An example of an application of Markov's inequality is the fact that (assuming incomes are non-negative) no more than 1/5 of the population can have more than 5 times the average income. Wikipedia: Markov Inequality

2.4.2 Chebyshev inequality

What is the probability that the realization of a random variable X is close to its mean $E[\mathsf{X}]$? The Chebyshev inequality gives a bound on the probability that X is within ϵ of its mean:

Proposition 2.3. *Chebyshev inequality* Let X be a random variable with finite expected value $E[\mathsf{X}]$ and finite non-zero variance $\mathrm{Var}[\mathsf{X}]$. Then for $\epsilon > 0$:

$$\Pr(|\mathsf{X} - E[\mathsf{X}]| < \epsilon) \geq 1 - \frac{\mathrm{Var}[\mathsf{X}]}{\epsilon^2}. \tag{2.87}$$

Some textbooks may write the inequality as:

$$\Pr(|\mathsf{X} - E[\mathsf{X}]| \geq \epsilon) \leq \frac{\mathrm{Var}[\mathsf{X}]}{\epsilon^2}. \tag{2.88}$$

Note that this is only interesting when $\mathrm{Var}[\mathsf{X}] \leq \epsilon^2$.

Consider a special case. Let X have expected value μ and variance σ^2. Choose $\epsilon^2 = 2\sigma^2$. Using (2.87), the probability that X is in the interval:

$$(\mu - \epsilon, \mu + \epsilon) = (\mu - 2\sigma^2, \mu + 2\sigma^2) \tag{2.89}$$

is $1 - \frac{\sigma^2}{\epsilon^2} = 0.5$ or greater.

2.4.3 Random Vectors: How Close Is the Sample Mean to the True Mean?

If we take a realization $\mathsf{X}_1, \mathsf{X}_2, \ldots, \mathsf{X}_n$, its sample mean will generally differ from the expected mean, as was shown in Example 2.12. We ask the question — how

[2] If $p_{\mathsf{X}}(x) = \Pr[\mathsf{X} = x]$ is the probability density function for X, then the cumulative distribution function is $F_{\mathsf{X}}(x) = \Pr[\mathsf{X} \leq x]$ for X.

close is the sample mean of random vector to its expected mean. In simple cases, the probability the sample mean is close to the expected mean can be computed exactly. Instead of an exact value, the Chebyshev inequality is used to give a lower bound on this probability.

We are interested sequences $\mathbf{X}$ for which the sample mean $\overline{\mathsf{X}}_n$ is "epsilon close" to its expected mean $E[\overline{\mathsf{X}}_n]$:

$$|\overline{\mathsf{X}}_n - E[\overline{\mathsf{X}}_n]| \leq \epsilon, \tag{2.90}$$

where $\epsilon > 0$ is some small constant value. Since $\overline{\mathsf{X}}_n$ is a random variable, we are interested in the probability:

$$\Pr(|\overline{\mathsf{X}}_n - E[\mathsf{X}]| \leq \epsilon), \tag{2.91}$$

where $E[\overline{\mathsf{X}}_n] = E[\mathsf{X}]$. Computing this probability directly can be difficult, instead a lower bound q:

$$\Pr\left(|\overline{\mathsf{X}}_n - E[\mathsf{X}]| \leq \epsilon\right) \geq q, \tag{2.92}$$

will be found using the Chebyshev inequality.

Example 2.13. Continue Example 2.12 to find an upper bound on the probability that $\overline{\mathsf{X}}_n$ is within $\epsilon = \frac{2}{15}$ of its mean, that is, what is the probability that the sample mean $\overline{\mathsf{X}}_n$ takes on values in $\frac{1}{3} \pm \frac{2}{15}$ or $(\frac{3}{15}, \frac{7}{15})$. An equivalent condition is $\mathbf{X}$ has 3, 4, 5, 6 or 7 ones. The numerical mean and variance are $E[\overline{\mathsf{X}}_n] = \frac{1}{3}$ and $\mathrm{Var}[\overline{\mathsf{X}}_n] = \frac{2}{9n} \approx 0.0148$. Then, using[3] (2.87):

$$\Pr(|\overline{\mathsf{X}}_n - E[\mathsf{X}]| < \frac{2}{15}) \geq 1 - \frac{2/135}{4/225} = \frac{1}{6}, \tag{2.93}$$

That is, the Chebyshev inequality states that the probability that $|\overline{\mathsf{X}}_n - E[\mathsf{X}]|$ is less than $\epsilon = \frac{2}{15}$ is greater than $\frac{1}{6}$. The exact value is

$$\sum_{i=3}^{7} \binom{15}{i} p^i (1-p)^{15-i} \approx 0.8324. \quad \square \tag{2.94}$$

We are interested in the following case: for a fixed ϵ and fixed q, what value of n is needed to satisfy (2.92)? When n is small, this n can be found by direct computation. But for large n and arbitrary distribution, we can form a lower bound using the Chebyshev inequality. If n is increased, then the lower bound q provided by the Chebyshev bound will increase towards 1. The table below shows the values of n required such that the Chebyshev bound will satisfy the specified value of q.

q	Chebyshev bound on n	exact value of n
$\frac{1}{6}$	15	—
0.9	125	30
0.99	1250	80
0.999	12500	134

[3] Note that even though a strict inequality is inside the probability, if $\frac{2}{15}$ is replaced with $\frac{2}{15}$ plus a very small number, then all the values in range $\frac{1}{3} \pm \frac{2}{15}$ will be included.

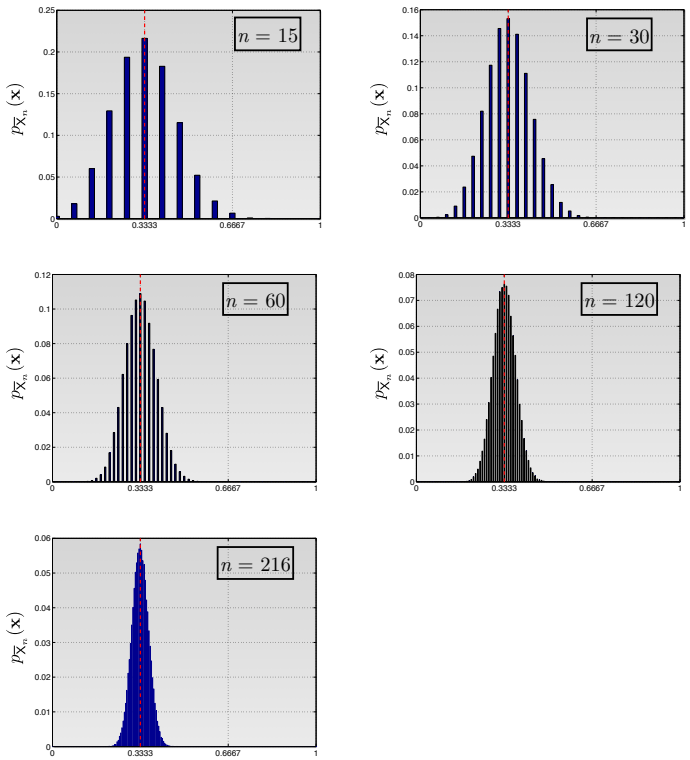


Figure 2.3: Example of the sample mean of a binary random vector with $p = \frac{1}{3}$ for various values of n.

In addition, Fig. 2.3 shows the probability distribution on the random variable $\overline{\mathsf{X}}_n$, for various values of n, where it is clear that the distribution tends to concentrate around the mean as n becomes larger. Thus, any desired probability q can be achieved by making n sufficiently large, for any fixed ϵ. The law of large numbers lets n go to infinity, and shows that q can go to 1.

2.4.4 Law of Large Numbers

The law of large numbers plays a central role in information theory. The law of large numbers shows that the sample mean of a sequence of random variables approaches the expected mean, as the number of random variables goes to infinity.

The *weak law of large numbers* states that the sample mean converges in probability towards the expected value. Let $\mathsf{X}_1, \mathsf{X}_2, \ldots, \mathsf{X}_n$ be a sequence of independent and identically distributed random variables with mean $E[\mathsf{X}]$. Then, for any positive number ε

$$\lim_{n\to\infty} \Pr\big(|\overline{\mathsf{X}}_n - E[\mathsf{X}]| < \varepsilon \big) = 1. \tag{2.95}$$

The proof is left as an exercise.

Interpreting this result, the law of large numbers[4] states that for any nonzero margin ϵ, no matter how small, with a sufficiently large sample size there will be a very high probability that the average of the observations will be close to the expected value; that is, within the margin.

The related central limit theorem states that the average of a large number of samples will asymptotically approach a Gaussian distribution, when the average is appropriately scaled. The central limit theorem is a powerful result, but is not used in this course.

2.5 Matlab Source Code

2.5.1 Basic Probability Operations

Compute $p_{\mathsf{Y}}(y) = \sum_{x \in \mathcal{X}} p_{\mathsf{X}}(x) \cdot p_{\mathsf{Y}|\mathsf{X}}(y|x)$ (`pygx` is "probability of Y given X").

```
>> pygx = [1 0 0 ; 4/5 1/5 0 ; 0 2/3 1/3]

pygx =

    1.0000         0         0
    0.8000    0.2000         0
         0    0.6667    0.3333

>> px = [1/2 1/4 1/4];
>> py = px * pygx

py =

    0.7000    0.2167    0.0833
```

Marginalization: $p_{\mathsf{Y}}(y) = \sum_{y \in \mathcal{Y}} p_{\mathsf{XY}}(x, y)$ and $p_{\mathsf{X}}(x) = \sum_{x \in \mathcal{X}} p_{\mathsf{XY}}(x, y)$.

```
>> pxy = [1/2 0 0 ; 1/4 1/16 0 ; 0 1/8 1/16]

pxy =

    0.5000         0         0
    0.2500    0.0625         0
         0    0.1250    0.0625

>> py = sum(pxy,1)

py =

    0.7500    0.1875    0.0625
```

[4] The lecture notes use the *weak law of large numbers*. The *strong* law of large numbers states that $\Pr\left(\lim_{n\to\infty} \overline{\mathsf{X}}_n = E\mathsf{X}\right) = 1$, which is a more powerful result, but more difficult to prove.

```
>> px = sum(pxy,2)' %transpose

px =

    0.5000    0.3125    0.1875

>> rats(px)         %display as rational numbers

ans =

      1/2           5/16          3/16
```

Compute joint distribution $p_{\mathsf{XY}}(x,y) = p_{\mathsf{X}}(x)p_{\mathsf{Y}|\mathsf{X}}(y|x)$.

```
>> [X,Y] = size(pygx);
>> pxy = repmat(px(:),1,Y) .* pygx;
```

Compute conditional distribution $p_{\mathsf{X}|\mathsf{Y}}(x|y) = \frac{p_{\mathsf{XY}}(x,y)}{p_{\mathsf{Y}}(y)}$ and $p_{\mathsf{Y}|\mathsf{X}}(y|x) = \frac{p_{\mathsf{XY}}(x,y)}{p_{\mathsf{Y}}(y)}$:

```
>> [X,Y] = size(pxy);
>> py = sum(pxy,1);
>> px = sum(pxy,2);
>> pxgy = pxy' ./ repmat(py,X,1)'; %transpose: rows sum to 1
>> pygx = pxy  ./ repmat(px,1,Y);
```

2.5.2 Random Variable Generation

Generate n samples of a random variable according to a distribution $p_{\mathsf{X}}(x)$.

```
function X = randomSamples(px,n)

if nargin < 2
    n = 1;  %default n
end

Fx = cumsum([0 px(1:end-1)] );      %cumulative distribution
X=zeros(1,n);                       %pre-allocate
for ii = 1:n;
    X(ii)=find(Fx < rand,1,'last'); %generate sample
end
```

Generate $n = 10$ samples from the distribution $p_{\mathsf{X}}(x) = [\frac{1}{4}, \frac{1}{2}, \frac{1}{4}]$

```
>> randomSamples([1/4 1/2 1/4],10)

ans =

     2     3     3     2     2     1     2     2     2     2
```

2.5.3 Sample Mean Experiments

Conduct a large number of experiments to find the number of samples within epsilon ϵ of the expected mean. Then use this to compute the probability of being within ϵ of the expected mean.

2.6 Python Source Code

Python version of Subsection 2.5.3, to conduct a large number of experiments.

```
from __future__ import division
import numpy
import math

X       = [1, 2, 3]
px      = [1/4, 1/2, 1/4]
n       = 50
epsilon = 0.1
numberOfExperiments = 1000

trueMean = 0;
for i in range(len(px)):
    trueMean = trueMean + X[i] * px[i]

sampleMeans = [0] * numberOfExperiments;
for i in range(numberOfExperiments):
    x = numpy.random.choice(X,n,p=px)
    sampleMeans[i] = numpy.mean(x)

v = [t for t in sampleMeans if abs(t-trueMean) < epsilon]
numberWithinEpsilon = len(v)

probability = numberWithinEpsilon / numberOfExperiments
print probability
```

2.7 Exercises

2.1 Let $\mathsf{X}_1, \mathsf{X}_2, \mathsf{X}_3$ be independently and identically distributed binary random variables with sample space $\mathcal{X} = \{0, 1\}$ and $p_{\mathsf{X}}(x) = [1 - p, p]$.

(a) What is the sample space of $\mathbf{X}$, where $\mathbf{X} = \mathsf{X}_1, \mathsf{X}_2, \mathsf{X}_3$?

(b) Find the joint distribution $p_{\mathbf{X}}(\mathbf{x})$.

2.2 Consider an $n = 25$ binary random vector $\mathbf{X} = \mathsf{X}_1\mathsf{X}_2 \cdots \mathsf{X}_{25}$. Let each X_i be independently and identically distributed as $p_{\mathsf{X}}(x) = [0.4, 0.6]$, so $p = 0.6$. Compute the following probabilities.

(a) $\Pr(\mathbf{X} = 1111111111110000000000000)$

(b) Pr($\mathbf{X}$ has 12 ones)

(c) Pr($\mathbf{X}$ has 13 ones)

(d) Pr($\mathbf{X}$ has 12, 13, 14 or 15 ones)

(e) Pr($\mathbf{X}$ has 12 or fewer ones)

(f) Pr($\mathbf{X}$ has 14 or more ones)

2.3 Let X have sample space $\mathcal{X} = \{0, 1\}$ and be distributed as $p_{\mathsf{X}}(x) = [\frac{2}{3}, \frac{1}{3}]$. Let Z have sample space $\mathcal{Z} = \{0, 2\}$ and be distributed as $p_{\mathsf{Z}}(0) = \frac{1}{2}$ and $p_{\mathsf{Z}}(2) = \frac{1}{2}$. X and Z are independent.

Now, let $\mathsf{Y} = \mathsf{X} \cdot \mathsf{Z}$, where $\cdot$ is usual multiplication.

(a) What is the joint distribution $p_{\mathsf{XZ}}(x, z)$?
(b) What is the sample space $\mathcal{Y}$?
(c) What is the probability distribution $p_{\mathsf{Y}}(y)$?
(d) What are the probability distributions $p_{\mathsf{Y}|\mathsf{Z}}(y|0)$ and $p_{\mathsf{Y}|\mathsf{Z}}(y|2)$.
(e) What is the entropy $H(\mathsf{X})$ and $H(\mathsf{Z})$?
(f) What is the entropy $H(\mathsf{Y}|\mathsf{Z})$?

2.4 Let X and Z be independent random variables where X is a $\mathcal{X} = \{0, 1\}$ binary random variable distributed as $p_{\mathsf{X}}(x) = [\frac{1}{2}, \frac{1}{2}]$, and Z is distributed as:

$$p_{\mathsf{Z}}(z) = \begin{cases} 1 - p & z = 0 \\ p & z = 1 \end{cases}. \tag{2.123}$$

Now, let $\mathsf{Y} = \mathsf{X} + \mathsf{Z}$, where $+$ is usual addition.

(a) What is the joint distribution $p_{\mathsf{XZ}}(x, z)$?
(b) What is the sample space $\mathcal{Y}$?
(c) What is the probability distribution $p_{\mathsf{Y}}(y)$?
(d) What is the joint distribution $p_{\mathsf{XYZ}}(x, y, z)$?
(e) What is the joint distribution $p_{\mathsf{XY}}(x, y)$?
(f) What is the conditional distribution $p_{\mathsf{Y}|\mathsf{X}}(y|x)$?
(g) What is the entropy $H(\mathsf{X})$ and $H(\mathsf{Z})$? Use the binary entropy function.
(h) What is the conditional entropy $H(\mathsf{Y}|\mathsf{X})$? Use the binary entropy function if necessary.

2.5 Let $\mathbf{Z}$ be a binary random vector:

$$\mathbf{Z} = (\mathsf{Z}_1, \mathsf{Z}_2, \ldots, \mathsf{Z}_n) \tag{2.150}$$

with $p_{\mathsf{Z}}(z) = [(1 - p), p]$.

Think of Z_i as an error in position i. An error occurs with a small probability p. $\mathbf{Z}$ is an "error vector", indicating position of errors in a sequence of length n.

Assume $p = 0.1$ and $n = 8$.

(a) What is the probability of 0 errors?

(b) What is the probability of 1 error?
(c) What is the probability of 0, 1 or 2 errors?

2.6 Let the random variable X be distributed as:

$$p_{\mathsf{X}}(x) = \begin{cases} \frac{1}{4} & \text{if } x = 1 \\ \frac{1}{2} & \text{if } x = 2 \\ \frac{1}{4} & \text{if } x = 3 \end{cases}. \tag{2.156}$$

Consider the sample mean:

$$\overline{\mathsf{X}}_n = \frac{1}{n}\sum_{i=1}^{n} \mathsf{X}_n. \tag{2.157}$$

(a) Find $E[\mathsf{X}]$
(b) Find Var$[\mathsf{X}]$
(c) Using the Chebyshev inequality, what value of n is needed to guarantee that the probability that $\overline{\mathsf{X}}_n$ is within $\epsilon = 0.1$ of its mean is greater than 0.999?
(d) Next, it is expected that the mean should be even closer to its mean, within $\epsilon = 0.01$. Now what value of n is needed to guarantee this?

You should write a program to for the following parts. Refer to the Information Theory Lecture Notes for an example. For the random variable $\mathbf{X}$, write a program that randomly generates n samples from the distribution $p_{\mathsf{X}}(x)$, and computes $\overline{\mathsf{X}}_n$.

(e) If $n = 50$, perform Monte Carlo experiments to estimate the probability that $\overline{\mathsf{X}}_n$ is within $\varepsilon = 0.1$ of the true mean. You should perform 1000 Monte Carlo experiments, and count the number of times $|\overline{\mathsf{X}}_n - EX| \leq \varepsilon$ is satisfied.
(f) Repeat for $n = 100$.
(g) Repeat for $n = 200$.

2.7 Let X, Y and Z be jointly distributed random variables as $p_{\mathsf{XYZ}}(x, y, z)$. Prove that:

$$p_{\mathsf{Y}|\mathsf{X}}(y|x) = \sum_{z \in \mathcal{Z}} p_{\mathsf{Y}|\mathsf{XZ}}(y|xz) p_{\mathsf{Z}|\mathsf{X}}(z|x) \tag{2.160}$$

This is a variation on the theorem of total probability.

2.8 Prove the union bound. Let $A_1, A_2, \ldots, A_n$ be events. Prove the union bound:

$$\Pr(\bigcup_{i=1}^{n} A_i) \leq \sum_{i=1}^{n} \Pr(A_i) \tag{2.165}$$

2.9 Prove the Chebyshev Inequality, for a random variable X with mean μ and variance σ^2:

$$\Pr[|\mathsf{X} - \mu| \geq k\sigma] \leq \frac{1}{k^2} \tag{2.171}$$

using the Markov inequality.

2.10 Let $\mathsf{X}_1, \mathsf{X}_2, \ldots \mathsf{X}_n$ be a sequence of n independent and identically distributed random variables with expected value $E[\mathsf{X}]$ and variance $\mathrm{Var}[\mathsf{X}]$. The sample mean is:

$$\overline{\mathsf{X}}_n = \frac{1}{n} \sum_{i=1}^{n} \mathsf{X}_i. \tag{2.175}$$

The law of large numbers states that the sample mean converges in probability towards the expected value:

$$\lim_{n \to \infty} \Pr\left(|\overline{\mathsf{X}}_n - E[\mathsf{X}]| < \epsilon\right) = 1 \tag{2.176}$$

Give a proof of the law of large numbers. Use the Chebyshev inequality in your proof.

Chapter 3

Mutual Information and KL divergence

There are three key quantities in information theory. Entropy was given in Chapter 1. This chapter gives the other two: mutual information and Kullback-Leibler divergence. Informally, the mutual information $I(\mathsf{X};\mathsf{Y})$ between two random variables X and Y is how much X tells you about Y (and, how much Y tells you about X). The Kullback-Leiber divergence $D(p||q)$ is a kind of "distance" between a true probability distribution $p(x)$ and an approximate distribution $q(x)$.

3.1 Mutual Information

The mutual information $I(\mathsf{X},\mathsf{Y})$ is the reduction in the uncertainty of X by knowing Y. Wikipedia: Mutual Information

Definition 3.1. Let X and Y be jointly distributed random variables. Then the *mutual information* $I(\mathsf{X};\mathsf{Y})$ between X and Y is:

$$I(\mathsf{X};\mathsf{Y}) = H(\mathsf{X}) - H(\mathsf{X}|\mathsf{Y}) \tag{3.1}$$

Alternatively, mutual information may be defined as follows. Consider random variables X and Y with a joint probability distribution function $p_{\mathsf{X},\mathsf{Y}}(x,y)$ and marginal distributions $p_{\mathsf{X}}(x)$ and $p_Y(y)$. Then $I(\mathsf{X};\mathsf{Y})$ is given by:

$$I(\mathsf{X};\mathsf{Y}) = \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p_{\mathsf{X},\mathsf{Y}}(x,y)\log\frac{p_{\mathsf{X},\mathsf{Y}}(x,y)}{p_{\mathsf{X}}(x)p_{\mathsf{Y}}(y)}. \tag{3.2}$$

We use (3.1) so often that we use it as the "definition of mutual information." Many textbooks take (3.2) as the definition, but it is straightforward to show they are equivalent.

Additional reading: Cover and Thomas, Sections 2.3 to 2.5, 2.8, 2.10.

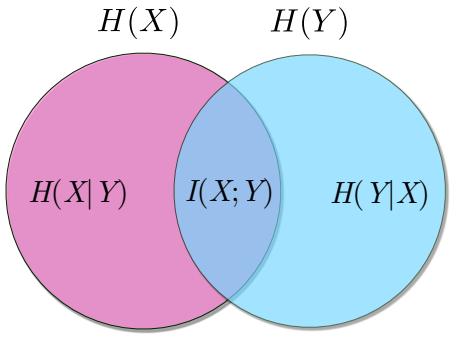


Figure 3.1: Venn diagram expressing relationships

Some intuition about mutual information: if the mutual information is high, then X tells you a lot about Y (and Y tells you a lot about X). If the mutual information is low, then X does not tell you much about Y (and vice verse). If the mutual information is zero, then X and Y are independent.

3.1.1 Properties of Mutual Information

This section describes key properties of mutual information.

Proposition 3.1. For jointly distributed random variables X, Y:

$$I(\mathsf{X};\mathsf{Y}) = H(\mathsf{X}) + H(\mathsf{Y}) - H(\mathsf{X},\mathsf{Y}) \tag{3.3}$$

This can be easily shown by applying the chain rule (1.38) $H(\mathsf{Y}|\mathsf{X}) = H(\mathsf{X}) - H(\mathsf{X},\mathsf{Y})$ to the definition of mutual information $I(\mathsf{X};\mathsf{Y}) = H(\mathsf{Y}) - H(\mathsf{Y}|\mathsf{X})$.

Symmetry of Mutual Information From (3.2) or (3.3), it is easy to see that mutual information is symmetric in its two variables:

$$I(\mathsf{X};\mathsf{Y}) = I(\mathsf{Y};\mathsf{X}). \tag{3.4}$$

The mutual information between X and itself, $I(\mathsf{X};\mathsf{X})$ is called the *self information*, and,

$$I(\mathsf{X};\mathsf{X}) = H(\mathsf{X}). \tag{3.5}$$

The relationship between mutual information and entropy can be expressed by a Venn diagram in Fig. 3.1. The entropy $H(\mathsf{X})$ and $H(\mathsf{Y})$ are represented by circles, and the joint entropy $H(\mathsf{X},\mathsf{Y})$ is represented by the union of the two circles. The mutual information $I(\mathsf{X};\mathsf{Y})$ is the intersection of the two circles. The left circle $H(\mathsf{X})$ has two parts, so that $H(\mathsf{X}) = I(\mathsf{X};\mathsf{Y}) + H(\mathsf{X}|\mathsf{Y})$, from the definition of mutual information.

Non-Negativity of Mutual Information Mutual information is non-negative:

$$I(\mathsf{X};\mathsf{Y}) \geq 0, \tag{3.6}$$

with equality if and only if X and Y are independent. This is shown in the next section.

Mutual information upper bound Since $I(\mathsf{X};\mathsf{Y}) = H(\mathsf{X}) - H(\mathsf{X}|\mathsf{Y})$, mutual information is upper bounded by:

$$I(\mathsf{X};\mathsf{Y}) \leq H(\mathsf{X}) \text{ and} \tag{3.7}$$
$$I(\mathsf{X};\mathsf{Y}) \leq H(\mathsf{Y}). \tag{3.8}$$

Random variables X and Y take values from the alphabets $\mathcal{X}$ and $\mathcal{Y}$, respectively, and $H(\mathsf{X}) \leq \log|\mathcal{X}|$. Mutual information is upper bounded as:

$$I(\mathsf{X};\mathsf{Y}) \leq \log|\mathcal{X}| \tag{3.9}$$
$$I(\mathsf{X};\mathsf{Y}) \leq \log|\mathcal{Y}| \tag{3.10}$$

The first bound can be shown by writing $I(\mathsf{X};\mathsf{Y}) = H(\mathsf{X}) - H(\mathsf{X}|\mathsf{Y})$ and using $H(\mathsf{X}) \leq \log|\mathcal{X}|$ and $H(\mathsf{X}|\mathsf{Y}) \geq 0$.

3.1.2 Conditional Mutual Information and Chain Rules

Just as there is conditional entropy $H(\mathsf{X}|\mathsf{Z})$, there is also conditional mutual information $I(\mathsf{X};\mathsf{Y}|\mathsf{Z})$.

Definition 3.2. The *conditional mutual information* of random variable X and Y given Z is:

$$I(\mathsf{X};\mathsf{Y}|\mathsf{Z}) = H(\mathsf{X}|\mathsf{Z}) - H(\mathsf{X}|\mathsf{Y},\mathsf{Z}) \tag{3.11}$$

Proposition 3.2. Let X,Y and Z be jointly distributed random variables. Then the *chain rule for mutual information* is

$$I(\mathsf{X};\mathsf{YZ}) = I(\mathsf{X};\mathsf{Y}|\mathsf{Z}) + I(\mathsf{X};\mathsf{Z}) \tag{3.12}$$

This chain rule can be generalized to an arbitrary number of variables, as follows.

Proposition 3.3. Let $\mathsf{X}_1,\mathsf{X}_2,\ldots,\mathsf{X}_n$ and Y be jointly distributed. Then the following holds:

$$I(\mathsf{X}_1,\mathsf{X}_2,\ldots\mathsf{X}_n;\mathsf{Y}) = \sum_{i=1}^{n} I(\mathsf{X}_i;\mathsf{Y}|\mathsf{X}_{i-1},\mathsf{X}_{i-2},\ldots,\mathsf{X}_1) \tag{3.13}$$

Proof:

$$I(\mathsf{X}_1,\mathsf{X}_2,\ldots\mathsf{X}_n;\mathsf{Y}) = H(\mathsf{X}_1,\mathsf{X}_2,\ldots,\mathsf{X}_n) - H(\mathsf{X}_1,\mathsf{X}_2,\ldots,\mathsf{X}_n|\mathsf{Y}) \tag{3.14}$$
$$= \sum_{i=1}^{n} H(\mathsf{X}_i|\mathsf{X}_{i-1},\ldots,\mathsf{X}_1) - \sum_{i=1}^{n} H(\mathsf{X}_i|\mathsf{X}_{i-1},\ldots,\mathsf{X}_1,\mathsf{Y}) \tag{3.15}$$
$$= \sum_{i=1}^{n} I(\mathsf{X}_i;\mathsf{Y}|\mathsf{X}_1,\mathsf{X}_2,\ldots,\mathsf{X}_{i-1}) \quad \square \tag{3.16}$$

3.1.3 Numerical Example

A discrete memoryless channel has input $\mathcal{X} \in \{0,1\}$ and output $\mathcal{Y} \in \{0,1\}$. Given a conditional distribution $p_{\mathsf{Y}|\mathsf{X}}(y|x)$ and an input distribution $p_{\mathsf{X}}(x)$, it is convenient to write mutual information $I(\mathsf{X};\mathsf{Y})$ by modifying (3.2) as:

$$I(\mathsf{X};\mathsf{Y}) = \sum_{x\in\mathcal{X}} \sum_{y\in\mathcal{Y}} p_{\mathsf{Y}|\mathsf{X}}(y|x) p_{\mathsf{X}}(x) \log \frac{p_{\mathsf{Y}|\mathsf{X}}(y|x)}{p_{\mathsf{Y}}(y)}. \tag{3.17}$$

In this example, find the mutual information $I(\mathsf{X};\mathsf{Y})$ for the DMC given by:

$$p_{\mathsf{Y}|\mathsf{X}}(y|x) = \begin{bmatrix} 0.711 & 0.289 \\ 0.289 & 0.711 \end{bmatrix} \tag{3.18}$$

(rows correspond to $x = 0, 1$) with input distribution $p_{\mathsf{X}}(x) = [\frac{1}{4}, \frac{3}{4}]$.

To apply (3.17), first find $p_{\mathsf{Y}}(y)$:

$$p_{\mathsf{Y}}(y) = \sum_{x\in\mathcal{X}} p_{\mathsf{Y}|\mathsf{X}}(y|x) p_{\mathsf{X}}(x) = \begin{bmatrix} \frac{1}{4} & \frac{3}{4} \end{bmatrix} \cdot \begin{bmatrix} 0.711 & 0.289 \\ 0.289 & 0.711 \end{bmatrix} = \begin{bmatrix} 0.3945 & 0.6055 \end{bmatrix} \tag{3.19}$$

Then,

$$\begin{aligned} I(\mathsf{X};\mathsf{Y}) = \quad & 0.711 \cdot \frac{1}{4} \log \frac{0.711}{0.3945} \\ & + 0.289 \cdot \frac{1}{4} \log \frac{0.289}{0.6055} \\ & + 0.289 \cdot \frac{3}{4} \log \frac{0.289}{0.3945} \\ & + 0.711 \cdot \frac{3}{4} \log \frac{0.711}{0.6055} \\ & = 0.1002 \end{aligned}$$

SSQ 3.1. Let X and Y be random variables with conditional probability distribution:

$$p_{\mathsf{Y}|\mathsf{X}}(y|x) = \begin{bmatrix} 0.89 & 0.11 \\ 0.11 & 0.89 \end{bmatrix}$$

where the rows are $\mathcal{X} = \{0, 1\}$ and the columns are $\mathcal{Y} = \{0, 1\}$. This is a model of a communications channel where the probability of error is 0.11. The input distribution is $p_{\mathsf{X}}(x) = [0.5, 0.5]$. Compute $I(\mathsf{X}; \mathsf{Y})$.

Another way to measure the dependence between two variables is the correlation coefficient. The (Pearson) correlation coefficient is a measure of the linear correlation between two variables. Wikipedia: Correlation Coefficient

Definition 3.3. The *covariance* between jointly distributed random variables X, Y is:

$$\mathrm{Cov}(\mathsf{X}, \mathsf{Y}) = E[\mathsf{XY}] - E[\mathsf{X}]E[\mathsf{Y}]. \tag{3.20}$$

If X, Y are independent then $E[\mathsf{XY}] = E[\mathsf{X}]E[\mathsf{Y}]$ and $\mathrm{Cov}(\mathsf{X}, \mathsf{Y}) = 0$. Wikipedia: Covariance

Definition 3.4. The *correlation coefficient* ρ between two random variables X and Y is:

$$\rho = \frac{\mathrm{Cov}(\mathsf{X}, \mathsf{Y})}{\sqrt{\mathrm{Var}[\mathsf{X}]\mathrm{Var}[\mathsf{Y}]}} \tag{3.21}$$

The correlation coefficient satisfies $-1 \leq \rho \leq 1$. $\rho = 0$ implies X and Y have no linear correlation.

Correlation measures only the *linear* relationship between X and Y. If X and Y are dependent but with no linear correlation, then $\rho = 0$, and the dependence is not clear. On the other hand, mutual information expresses dependence between X and Y. In this sense, mutual information is superior because it can reveal dependence which correlation coefficient cannot.

3.2 Kullback-Leiber Divergence

The KL divergence $D\big(p(x)||q(x)\big)$ is a measure of a "distance" between two distributions $p(x)$ and $q(x)$. It is helpful to think of $p(x)$ as a true distribution, and $q(x)$ as an approximation distribution. Then $D(p||q)$ is the penalty of using q to approximate p. KL divergence can also be seen as a measure of the information lost when q is used to approximate p — this is discussed as the cost of miscoding in Subsection 4.4.4 on page 87. The KL divergence is sometimes called the information divergence or relative entropy. Wikipedia: Kullback–Leibler divergence

Definition 3.5. The *KL divergence* $D\big(p(x)||q(x)\big)$ or $D(p||q)$ between the two probability distribution functions $p(x)$ and $q(x)$ is:

$$D(p||q) = \sum_{x\in\mathcal{X}} p(x) \log \frac{p(x)}{q(x)} \tag{3.22}$$

with $0 \log \frac{0}{q} = 0$ and $p \log \frac{p}{0} = \infty$.

The KL divergence is of fundamental importance. Its properties will be used to prove several properties of entropy and mutual information.

KL divergence is non-negative:

Proposition 3.4. For probabiity distributions $p(x)$ and $q(x)$ defined on $\mathcal{X}$:

$$D(p||q) \geq 0 \tag{3.23}$$

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

This will be proved in a Proposition 13.3 on page 212.

KL divergence is not symmetric in p and q, that is:

$$D(p||q) \neq D(q||p), \tag{3.24}$$

in general. Also, KL divergence does not satisfy the triangle inequality, that is, for three distributions $p(x)$, $q(x)$ and $r(x)$:

$$D(p||q) \not\leq D(p||r) + D(r||q). \tag{3.25}$$

Because KL divergence is not symmetric, and because KL divergence does not satisfy the triangle inequality, it is not a *metric*, in the mathematical sense. Wikipedia: Metric

The mutual information $I(\mathsf{X};\mathsf{Y})$ is the KL divergence between the joint distribution $p_{\mathsf{XY}}(x,y)$ and the product distribution $p_{\mathsf{X}}(x)p_{\mathsf{Y}}(y)$:

$$I(\mathsf{X};\mathsf{Y}) = \sum_{x\in\mathcal{X}} \sum_{y\in\mathcal{Y}} p_{\mathsf{XY}}(x,y) \log \frac{p_{\mathsf{XY}}(x,y)}{p_{\mathsf{X}}(x)p_{\mathsf{Y}}(y)} \tag{3.26}$$

$$= D\big(p_{\mathsf{XY}}(x,y) \,||\, p_{\mathsf{X}}(x)p_{\mathsf{Y}}(y)\big) \tag{3.27}$$

SSQ 3.2. Let $\mathcal{X} = \{a, b, c\}$ and consider two distributions $p(x)$ and $q(x)$, given by:

	$p(x)$	$q(x)$
a	$\frac{1}{2}$	$\frac{1}{3}$
b	$\frac{1}{4}$	$\frac{1}{3}$
c	$\frac{1}{4}$	$\frac{1}{3}$

Calculate $D(p||q)$ and $D(q||p)$. Verify that in this case, $D(p||q) \neq D(q||p)$

3.2.1 Consequences of Non-Negativity of KL divergence

The non-negativity of KL divergence allows proving a three results: (1) non-negativity of mutual information (2) conditioning reduces entropy and (3) the uniform distribution maximizes entropy.

(1) The non-negativity of mutual information follows from the non-negativity of KL divergence

$$\begin{aligned} I(\mathsf{X};\mathsf{Y}) &= D(p_{\mathsf{XY}}(x,y)||p_{\mathsf{X}}(x)p_{\mathsf{Y}}(y)) \\ &\geq 0 && \text{non-negativity of KL divergence} \end{aligned}$$

(2) Conditioning reduces entropy; this was given in Proposition 1.2 on page 21:

Proof of Proposition 1.2

$$\begin{aligned} I(\mathsf{X};\mathsf{Y}) &\geq 0 && \text{non-negativity of mutual information} \\ H(\mathsf{X}) - H(\mathsf{X}|\mathsf{Y}) &\geq 0 && \text{definition of mutual information} \\ H(\mathsf{X}) &\geq H(\mathsf{X}|\mathsf{Y}). \quad \square \end{aligned}$$

Proposition 3.5. For a random variable X with sample set $\mathcal{X}$, let Q be the uniform random variable on the same sample set $\mathcal{X}$ (that is, $\mathcal{Q} = \mathcal{X}$). Then:

$$H(\mathsf{X}) = \log|\mathcal{X}| - D(p_{\mathsf{X}}||p_{\mathsf{Q}}). \tag{3.28}$$

Proof

$$\begin{aligned} D(p_{\mathsf{X}}||p_{\mathsf{Q}}) &= \sum_{x\in\mathcal{X}} p_{\mathsf{X}}(x)\log\frac{p_{\mathsf{X}}(x)}{p_{\mathsf{Q}}(x)} && (3.29) \\ &= \sum_{x\in\mathcal{X}} p_{\mathsf{X}}(x)\log p_{\mathsf{X}}(x) + \sum_{x\in\mathcal{X}} p_{\mathsf{X}}(x)\log|\mathcal{X}| && (3.30) \\ D(p_{\mathsf{X}}||p_{\mathsf{Q}}) &= \log|\mathcal{X}| - H(\mathsf{X}). \quad \square && (3.31) \end{aligned}$$

(3) The uniform distribution maximizes entropy; this was given in Proposition 1.1 on page 17.

Proof of Proposition 1.1 Let Q be the uniform random variable as in Proposition 3.5. Then:

$$\begin{aligned} D\big(p_{\mathsf{X}}||p_{\mathsf{Q}}\big) &= \log|\mathcal{X}| - H(\mathsf{X}) && \text{Proposition 3.5} && (3.32) \\ \log|\mathcal{X}| - H(\mathsf{X}) &\geq 0 && \text{non-negativity of KL divergence} && (3.33) \\ H(\mathsf{X}) &\leq \log|\mathcal{X}| && && (3.34) \end{aligned}$$

Equality is achieved in (3.34) when $D(p_{\mathsf{X}}(x)||p_{\mathsf{Q}}) = 0$. Since $D(p_{\mathsf{X}}(x)||p_{\mathsf{Q}}) = 0$ if and only if $p_{\mathsf{X}} = p_{\mathsf{Q}}$, we have $H(\mathsf{X}) = \log|\mathcal{X}|$ if and only if X is uniformly distributed. $\square$

3.3 Data Processing Inequality and Markov Chains

The data processing inequality expresses the idea *processing cannot not increase information*. First, a brief introduction to Markov chains is given.

3.3.1 Markov Chains

Here a three-variable Markov chain is described (general Markov chains that use a large sequence of random variables also exist). Let X, Y and Z be jointly distributed random variables. These random variables form a *Markov chain*, written

$$\mathsf{X} \to \mathsf{Y} \to \mathsf{Z} \tag{3.35}$$

if the conditional probability $p_{\mathsf{Z}|\mathsf{XY}}(z|x,y)$ does not change if X is dropped:

$$\Pr(\mathsf{Z}=z|\mathsf{X}=x,\mathsf{Y}=y) = \Pr(\mathsf{Z}=z|\mathsf{Y}=y) \text{ or} \tag{3.36}$$

$$p_{\mathsf{Z}|\mathsf{XY}}(z|x,y) = p_{\mathsf{Z}|\mathsf{Y}}(z|y). \tag{3.37}$$

The idea of Markovity is expressed by "the future (Z) depends on the present (Y) and not the past (X) ."

The random variables X, Y and Z are jointly distributed as $p_{\mathsf{XYZ}}(x,y,z)$. By the definition of conditional probability:

$$p_{\mathsf{XYZ}}(x,y,z) = p_{\mathsf{Z}|\mathsf{XY}}(z|x,y) \cdot p_{\mathsf{XY}}(x,y) \tag{3.38}$$

$$= p_{\mathsf{Z}|\mathsf{XY}}(z|x,y) \cdot p_{\mathsf{Y}|\mathsf{X}}(y|x) \cdot p_{\mathsf{X}}(x) \tag{3.39}$$

If $\mathsf{X} \to \mathsf{Y} \to \mathsf{Z}$ forms a Markov chain, then an equivalent condition to form a Markov chain is that the joint probability distribution factors as:

$$p_{\mathsf{XYZ}}(x,y,z) \quad = \quad p_{\mathsf{Z}|\mathsf{Y}}(z|y) p_{\mathsf{Y}|\mathsf{X}}(y|x) p_{\mathsf{X}}(x). \tag{3.40}$$

Here are four properties of Markov chains:

(1) $\mathsf{X} \to \mathsf{Y} \to \mathsf{Z}$ if and only if X and Z are conditionally independent given Y, that is:

$$p_{\mathsf{XZ}|\mathsf{Y}}(x,z|y) = \frac{p_{\mathsf{XYZ}}(x,y,z)}{p_{\mathsf{Y}}(y)} = \frac{p_{\mathsf{Z}|\mathsf{XY}}(z|xy) p_{\mathsf{XY}}(x,y)}{p_{\mathsf{Y}}(y)} = p_{\mathsf{X}|\mathsf{Y}}(x|y) p_{\mathsf{Z}|\mathsf{Y}}(z|y)$$

(2) $\mathsf{X} \to \mathsf{Y} \to \mathsf{Z}$ implies $\mathsf{Z} \to \mathsf{Y} \to \mathsf{X}$ also forms a Markov chain.

(3) If g is a function, then $\mathsf{X} \to \mathsf{Y} \to g(\mathsf{Y})$ forms a Markov chain.

(4) If $\mathsf{X} \to \mathsf{Y} \to \mathsf{Z}$ then $I(\mathsf{X};\mathsf{Z}|\mathsf{Y}) = 0$

3.3.2 Data Processing Inequality

"Processing cannot increase information" is as follows. If Z is some function of Y, then $I(\mathsf{X};\mathsf{Z})$ cannot be greater than $I(\mathsf{X};\mathsf{Y})$.

Proposition 3.6. *Data Processing Inequality.* If $\mathsf{X} \to \mathsf{Y} \to \mathsf{Z}$ is a Markov chain then

$$I(\mathsf{X};\mathsf{Y}) \geq I(\mathsf{X};\mathsf{Z}). \tag{3.41}$$

Proof: By the chain rule for mutual information, write mutual information $I(\mathsf{X};\mathsf{Y},\mathsf{Z})$ two different ways:

$$I(\mathsf{X};\mathsf{Y},\mathsf{Z}) = I(\mathsf{X};\mathsf{Z}) + \underbrace{I(\mathsf{X};\mathsf{Y}|\mathsf{Z})}_{\geq 0} = I(\mathsf{X};\mathsf{Y}) + \underbrace{I(\mathsf{X};\mathsf{Z}|\mathsf{Y})}_{0} \tag{3.42}$$

$$I(\mathsf{X};\mathsf{Z}) \leq I(\mathsf{X};\mathsf{Y}), \tag{3.43}$$

which is the data processing inequality. □

For a function g, $\mathsf{X} \to \mathsf{Y} \to g(\mathsf{Y})$ is a Markov chain. The data processing inequality says that a deterministic function g cannot increase information, that is $I(\mathsf{X};\mathsf{Y}) \geq I(\mathsf{X};g(\mathsf{Y}))$. Note also that if $\mathsf{X} \to \mathsf{Y} \to \mathsf{Z}$ then $I(\mathsf{X};\mathsf{Y}|\mathsf{Z}) \leq I(\mathsf{X};\mathsf{Y})$ — observation reduces dependence of random variables. Since $\mathsf{Z} \to \mathsf{Y} \to \mathsf{X}$ also forms a Markov chain, $I(\mathsf{Y};\mathsf{Z}) \geq I(\mathsf{X};\mathsf{Z})$ also holds.

We have the following for conditional entropy:

Proposition 3.7. If $\mathsf{X} \to \mathsf{Y} \to \mathsf{Z}$ then $H(\mathsf{X}|\mathsf{Z}) \geq H(\mathsf{X}|\mathsf{Y})$.

3.4 Fano's Inequality

Fano's inequality is a lower bound on probability of estimation error P_e. For a system that has $P_e \to 0$, the lower bound must also go to zero. This is used to prove the converse part of the channel coding theorem, and appears often in information theory.

Let X and Y be jointly distributed. We know Y and want to estimate X.

Definition 3.6. Let $\widehat{\mathsf{X}} = g(\mathsf{Y})$ be an *estimate* of X. The function g is called the *estimator* of X, where g is a deterministic function, and $\widehat{\mathcal{X}} = \mathcal{X}$.

Since $\widehat{\mathsf{X}} = g(\mathsf{Y})$, a Markov chain is formed: $\mathsf{X} \to \mathsf{Y} \to \widehat{\mathsf{X}}$. The probability of error P_e for the estimator g is:

$$P_e = \Pr[\widehat{\mathsf{X}} \neq \mathsf{X}]. \tag{3.44}$$

Fano's inequality gives a lower bound on P_e.

Proposition 3.8. *Fano's Inequality* For any estimator $\widehat{\mathsf{X}}$ such that $\mathsf{X} \to \mathsf{Y} \to \widehat{\mathsf{X}}$, we have:

$$h(P_e) + P_e \log |\mathcal{X}| \geq H(\mathsf{X}|\widehat{\mathsf{X}}) \tag{3.45}$$

Proof of Fano's inequality Let E be an error random variable:

$$\mathsf{E} = \begin{cases} 1 & \text{if } \widehat{\mathsf{X}} \neq \mathsf{X} \\ 0 & \text{if } \widehat{\mathsf{X}} = \mathsf{X} \end{cases} \tag{3.46}$$

From this definition, note the following as preparation:

- $P_e = \Pr(\widehat{\mathsf{X}} \neq \mathsf{X})$ so $P_e = \Pr(\mathsf{E} = 1)$
- $H(\mathsf{E}) = h(P_e)$, where h is the binary entropy function.
- $H(\mathsf{E}|\mathsf{X}, \widehat{\mathsf{X}}) = 0$
- $H(\mathsf{E}|\widehat{\mathsf{X}}) \leq H(\mathsf{E}) = h(P_e)$ (conditioning reduces entropy)
- and:

$$\begin{aligned} H(\mathsf{X}|\mathsf{E}, \widehat{\mathsf{X}}) &= Pr(\mathsf{E} = 0)H(\mathsf{X}|\widehat{\mathsf{X}}, \mathsf{E} = 0) + Pr(\mathsf{E} = 1)H(\mathsf{X}|\widehat{\mathsf{X}}, \mathsf{E} = 1) && (3.47)\\ &\leq (1 - P_e)0 + P_e \log |\mathcal{X}| && (3.48)\\ &\leq P_e \log |\mathcal{X}| && (3.49) \end{aligned}$$

Now we write the proof by writing $H(\mathsf{E}, \mathsf{X}|\widehat{\mathsf{X}})$ two ways:

$$\begin{aligned} H(\mathsf{E}, \mathsf{X}|\widehat{\mathsf{X}}) &= H(\mathsf{E}, \mathsf{X}|\widehat{\mathsf{X}}) && (3.50)\\ H(\mathsf{X}|\widehat{\mathsf{X}}) + H(\mathsf{E}|\mathsf{X}, \widehat{\mathsf{X}}) &= H(\mathsf{E}|\widehat{\mathsf{X}}) + H(\mathsf{X}|\mathsf{E}, \widehat{\mathsf{X}}) && (3.51)\\ H(\mathsf{X}|\widehat{\mathsf{X}}) + \underbrace{H(\mathsf{E}|\mathsf{X}, \widehat{\mathsf{X}})}_{=0} &= \underbrace{H(\mathsf{E}|\widehat{\mathsf{X}})}_{\leq h(P_e)} + \underbrace{H(\mathsf{X}|\mathsf{E}, \widehat{\mathsf{X}})}_{\leq P_e \log |\mathcal{X}|} && (3.52)\\ H(\mathsf{X}|\widehat{\mathsf{X}}) &\leq h(P_e) + P_e \log |\mathcal{X}|. \quad \square && (3.53) \end{aligned}$$

There are some variations to Fano's inequality. Since $H(\mathsf{X}|\widehat{\mathsf{X}}) \geq H(\mathsf{X}|\mathsf{Y})$ by the data processing inequality for conditional entropy, the following also holds:

$$h(P_e) + P_e \log |\mathcal{X}| \geq H(\mathsf{X}|\mathsf{Y}) \tag{3.54}$$

Fano's Inequality can be further weakened using $h(P_e) \leq 1$ to give:

$$\begin{aligned} P_e &\geq \frac{H(\mathsf{X}|\mathsf{Y}) - 1}{\log |\mathcal{X}|} \text{ or} && (3.55)\\ P_e &\geq \frac{H(\mathsf{X}|\widehat{\mathsf{X}}) - 1}{\log |\mathcal{X}|} && (3.56) \end{aligned}$$

3.5 Descriptions Using Expectation

Entropy, mutual information and KL divergence can be described using expectation. As shown in Subsection 2.2.2, for a random variable X with distribution $p_{\mathsf{X}}(x)$ and a function g, the expectation of $g(\mathsf{X})$ is given by:

$$E[g(\mathsf{X})] = \sum_{x \in \mathcal{X}} p_{\mathsf{X}}(x) g(x). \tag{3.57}$$

If we take the function $g(x) = -\log p_{\mathsf{X}}(x)$, then:

$$E[g(\mathsf{X})] = -\sum_{x \in \mathcal{X}} p_{\mathsf{X}}(x) \log p_{\mathsf{X}}(x), \tag{3.58}$$

which is the entropy $H(\mathsf{X})$, that is:

$$H(\mathsf{X}) = -E\big[\log p_{\mathsf{X}}(\mathsf{X})\big] \tag{3.59}$$

Take a close look at (3.59). The expectation is taken with respect to X, and the expectation contains $p_{\mathsf{X}}(\mathsf{X})$ and not $p_{\mathsf{X}}(x)$. Cover and Thomas called this expression "eerily self-referential," and it certainly is!

Similarly, the joint entropy $H(\mathsf{X},\mathsf{Y})$ of discrete random variables X and Y is expressed by:

$$H(\mathsf{X},\mathsf{Y}) = -E[\log p_{\mathsf{XY}}(\mathsf{X},\mathsf{Y})] \tag{3.60}$$

$$= -\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p_{\mathsf{X},\mathsf{Y}}(x,y)\log p_{\mathsf{X},\mathsf{Y}}(x,y) \tag{3.61}$$

and conditional entropy:

$$H(\mathsf{Y}|\mathsf{X}) = -\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p_{\mathsf{XY}}(x,y)\log p_{\mathsf{Y}|\mathsf{X}}(y|x) \tag{3.62}$$

$$= -E[\log p_{\mathsf{Y}|\mathsf{X}}(\mathsf{Y}|\mathsf{X})] \tag{3.63}$$

The mutual information $I(\mathsf{X};\mathsf{Y})$ can be expressed as:

$$I(\mathsf{X};\mathsf{Y}) = E[\log \frac{p_{\mathsf{X},\mathsf{Y}}(\mathsf{X},\mathsf{Y})}{p_{\mathsf{X}}(\mathsf{X})p_{\mathsf{Y}}(\mathsf{Y})}] \tag{3.64}$$

Let X be the random variable distributed according to $p(x)$. Then the KL divergence between $p(x)$ and a distribution $q(x)$ is:

$$D\big(p(x)||q(x)\big) = E[\log \frac{p(\mathsf{X})}{q(\mathsf{X})}] \tag{3.65}$$

$$= \sum_{x\in\mathcal{X}} p(x)\log\frac{p(x)}{q(x)}, \tag{3.66}$$

Conditional mutual information is expressed as:

$$I(\mathsf{X};\mathsf{Y}|\mathsf{Z}) = H(\mathsf{X}|\mathsf{Z}) - H(\mathsf{X}|\mathsf{Y},\mathsf{Z}) \tag{3.67}$$

$$= E[\log \frac{p_{\mathsf{XY}|\mathsf{Z}}(\mathsf{X},\mathsf{Y}|\mathsf{Z})}{p_{\mathsf{X}|\mathsf{Z}}(\mathsf{X}|\mathsf{Z})p_{\mathsf{Y}|\mathsf{Z}}(\mathsf{Y}|\mathsf{Z})}] \tag{3.68}$$

3.6 Matlab Source Code

3.6.1 Compute Mutual Information

Given an arbitrary conditional probability distribution $p_{\mathsf{Y}|\mathsf{X}}(y|x)$ and marginal distribution $p_{\mathsf{X}}(x)$, find the mutual information $I(\mathsf{X};\mathsf{Y})$.

```
1 function I = computeMutualInformation(pygx,px)
2
```

```
%compute pxgy using inputs pygx and px
%first, find joint distribution pxy:
[X,Y] = size(pygx);
pxy   = repmat(px(:),1,Y) .* pygx;
%second, find conditional distribution pxgy:
py    = sum(pxy,1);
pxgy  = pxy' ./ repmat(py,X,1)'; %transpose: rows sum to 1

%compute mutual information I = H(X) - H(X|Y)
HX    = computeEntropy(px);
HXgY  = computeConditionalEntropy(pxgy,py);
I     = HX - HXgY;
```

3.7 Exercises

3.1 What is the correct relationship, $=$, $\geq$, $\leq$ or ? (for unknown) for each pair below. Give reason with equations or phrase like "conditioning reduces entropy."

(a) $I(\mathsf{X};\mathsf{Y})$ _____ 0.
(b) $H(\mathsf{X},\mathsf{Y})$_____ $H(\mathsf{X})+H(\mathsf{Y})$
(c) $I(\mathsf{X};\mathsf{Y})+H(\mathsf{X}|\mathsf{Y})$ _____ $H(\mathsf{X})$.
(d) $I(\mathsf{X};\mathsf{X})$_____ $H(\mathsf{X})$.
(e) $I(\mathsf{X};\mathsf{Y})$ _____ $H(\mathsf{X})-H(g(\mathsf{Y})|\mathsf{Y})$.
(f) $H(\mathsf{X}|\mathsf{Y})$_____ $H(\mathsf{X})+H(\mathsf{Y})$
(g) $H(2\mathsf{X})$ _____ $H(\mathsf{X})$
(h) $H(\mathsf{X}_2|\mathsf{X}_1)$_____ $H(\mathsf{X}_2|\mathsf{X}_1,\mathsf{X}_0)$

3.2 Let $p_{\mathsf{X},\mathsf{Y}}$ be given by:

$$P_{\mathsf{X},\mathsf{Y}} = \begin{bmatrix} \frac{1}{3} & \frac{1}{3} \\ 0 & \frac{1}{3} \end{bmatrix}, \tag{3.69}$$

where the rows are $\mathcal{X}=\{0,1\}$ and the columns are $\mathcal{Y}=\{0,1\}$. Find:

(a) $H(\mathsf{X})$, $H(\mathsf{Y})$
(b) $H(\mathsf{X}|\mathsf{Y})$, $H(\mathsf{Y}|\mathsf{X})$
(c) $H(\mathsf{X},\mathsf{Y})$
(d) $H(\mathsf{Y})-H(\mathsf{Y}|\mathsf{X})$
(e) $I(\mathsf{X};\mathsf{Y})$.

3.3 Consider a DMC with input X and output Y. The input distribution is $p_X(x)=[\frac{1}{2},\frac{1}{2},0]$ and the channel transition probabilities are:

$$p_{Y|X}(y|x) = \begin{bmatrix} \frac{2}{3} & \frac{1}{3} & 0 \\ 0 & \frac{2}{3} & \frac{1}{3} \\ \frac{1}{3} & 0 & \frac{2}{3} \end{bmatrix}, \tag{3.70}$$

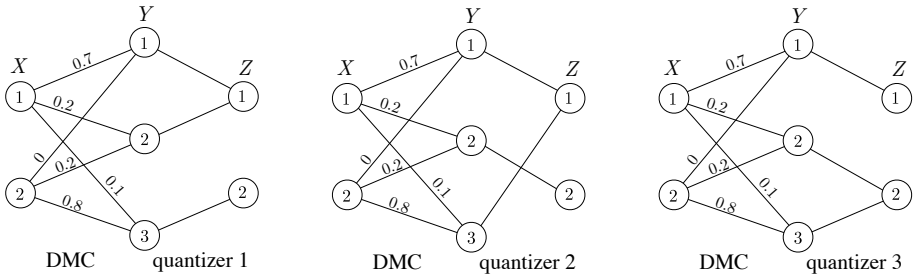


Figure 3.2: DMC for Problem 3.5.

Find $p_Y(y)$.

Find $I(X;Y)$.

3.4 Consider jointly distributed X and Y with $\mathcal{X} = \{0,1\}$ and $\mathcal{Y} = \{0,1\}$ with joint distribution given by:

$p_{\mathsf{XY}}(x,y)$	$y=0$	$y=1$
$x=0$	$\frac{1}{3}$	$\frac{1}{6}$
$x=1$	$\frac{1}{6}$	$\frac{1}{3}$

Let Z be a new random variable $\mathsf{Z} = \mathsf{X} + \mathsf{Y}$. Here "+" means real addition so $\mathcal{Z} = \{0,1,2\}$.

(a) Find $H(\mathsf{X})$, $H(\mathsf{Y})$ and $H(\mathsf{X},\mathsf{Y})$
(b) Find the joint distribution $p_{\mathsf{XYZ}}(x,y,z)$.
(c) Find $I(\mathsf{X},\mathsf{Y};\mathsf{Z})$.
(d) Find $I(\mathsf{X};\mathsf{Z})$.

3.5 Consider a discrete memoryless channel with 2 inputs X and 3 outputs Y:

$$p_{\mathsf{Y}|\mathsf{X}} = \begin{bmatrix} 0.7 & 0.2 & 0.1 \\ 0 & 0.2 & 0.8 \end{bmatrix},$$

as shown in the figure. The input distribution is $p_{\mathsf{X}}(1) = p_{\mathsf{X}}(2) = \frac{1}{2}$. A new binary output Z is created "quantizing" Y. There are three candidate quantizers with $\mathcal{Z} \in \{1,2\}$, corresponding to $\mathsf{Z}_1, \mathsf{Z}_2$ and Z_3, as shown in the figure. The conditional probability distribution $p_{\mathsf{Z}|\mathsf{X}}$ can be found, for example for quantizer 1:

$$p_{\mathsf{Z1}|\mathsf{X}}(1|1) = \sum_y p_{\mathsf{Y}|\mathsf{X}}(y|1) p_{\mathsf{Z1}|\mathsf{Y}}(1|y) \quad = 0.7 \cdot 1 + 0.2 \cdot 1 + 0.1 \cdot 0 = 0.9, \tag{3.75}$$

since for Quantizer 1, $p_{\mathsf{Z1}|\mathsf{Y}}(1|1) = 1$, $p_{\mathsf{Z1}|\mathsf{Y}}(1|2) = 1$, $p_{\mathsf{Z1}|\mathsf{Y}}(1|3) = 0$.

(a) Compute $I(\mathsf{X};\mathsf{Y})$.
(b) Compute the mutual information for Quantizer 1: $I(\mathsf{X};\mathsf{Z}_1)$.

(c) Compute the mutual information for Quantizer 2: $I(\mathsf{X};\mathsf{Z}_2)$.

(d) Compute the mutual information for Quantizer 3: $I(\mathsf{X};\mathsf{Z}_3)$.

(e) Of the three quantizers, which has the greatest mutual information?

(f) $\mathsf{X} \to \mathsf{Y} \to \mathsf{Z}$ forms a Markov chain. Is the data processing inequality satisfied?

3.6 Consider two distributions $p(x)$ and $q(x)$ on $\mathcal{X} = \{1, 2, 3, 4\}$ and given by:

x	$p(x)$	$q(x)$
1	$\frac{1}{2}$	$\frac{1}{2}$
2	$\frac{1}{4}$	$\frac{1}{8}$
3	$\frac{1}{8}$	$\frac{1}{8}$
4	$\frac{1}{8}$	$\frac{1}{4}$

(a) Calculate $D(p||q)$.

(b) Find a distribution $q'(x)$ such that $D(p||q') = 0$.

Verify that in this case, $D(p||q) \neq D(q||p)$.

3.7 For jointly distributed X, Y and Z, prove the following.

(a) $H(\mathsf{XY}|\mathsf{Z}) = H(\mathsf{X}|\mathsf{Z}) + H(\mathsf{Y}|\mathsf{XZ})$

(b) $I(\mathsf{X};\mathsf{Z}) + I(\mathsf{Y};\mathsf{Z}|\mathsf{X}) = I(\mathsf{XY};\mathsf{Z})$

Give a justification for each step of your proof.

3.8 Let X, Y and Z be jointly distributed variables such that $\mathsf{X} \to \mathsf{Y} \to \mathsf{Z}$ forms a Markov chain. Prove $H(\mathsf{X}|\mathsf{Z}) \geq H(\mathsf{X}|\mathsf{Y})$.

3.9 *Fano's inequality for the weather* Consider two sample spaces:

$$\mathcal{X} = \{\text{snowy}, \text{rainy}, \text{sunny}\}$$
$$\mathcal{Y} = \{0°\text{C}, 10°\text{C}, 20°\text{C}\}$$

In some city the temperature Y and weather X are jointly distributed as:

$$p_{\mathsf{XY}}(x, y) = \begin{bmatrix} \frac{1}{6} & \frac{1}{12} & \frac{1}{12} \\ \frac{1}{12} & \frac{1}{6} & \frac{1}{12} \\ \frac{1}{12} & \frac{1}{12} & \frac{1}{6} \end{bmatrix} \tag{3.81}$$

where rows are x and columns are y. Let the estimator g be given by:

$$g(y) = \begin{cases} \text{snowy} & y = 0°\text{C} \\ \text{rainy} & y = 10°\text{C} \\ \text{sunny} & y = 20°\text{C} \end{cases}. \tag{3.82}$$

(a) With $\widehat{\mathsf{X}} = g(\mathsf{Y})$, find in matrix form $p_{\mathsf{X}|\widehat{X}}(x|\widehat{x})$.

(b) Compute $H(\mathsf{X}|\widehat{\mathsf{X}})$.

(c) Using your answer to (b), use Fano's inequality to find a P_e lower bound.
(d) Compute P_e exactly.

3.10 Consider a binary symmetric channel with input $\mathsf{X} \in \{0, 1\}$ and output $\mathsf{W} \in \{0, 1\}$ with error probability q, so that $p_{\mathsf{W}|\mathsf{X}}(1|0) = p_{\mathsf{W}|\mathsf{X}}(0|1) = q$. The input X is distributed as $p_{\mathsf{X}}(x) = [1 - p, p]$.

(a) Find the joint distribution $p_{\mathsf{X},\mathsf{W}}(x, w)$ using p and q.
(b) Find $H(\mathsf{X}, \mathsf{W})$, $H(\mathsf{W})$ and $H(\mathsf{X}|\mathsf{W})$.

Now consider a new random variable $\mathsf{Y} \in \{0, 1\}$ with $p_{\mathsf{Y}}(y) = [1/2, 1/2]$, which is independent of X and W. Let Z be:

$$\mathsf{Z} = \mathsf{W} + \mathsf{Y}$$

(c) Find $I(\mathsf{X}; \mathsf{Z}|\mathsf{Y})$ and $I(\mathsf{Y}; \mathsf{Z}|\mathsf{X})$
(d) Fix $q = 0.1$. Make a plot of $I(\mathsf{X}; \mathsf{Z}|\mathsf{Y})$ versus p for $0 < p < 1$.

For parts (b) and (c), express your answer as a function of p and q using the binary entropy function $h(\cdot)$.

3.11 *Information Bottleneck* Let $\mathsf{X} \to \mathsf{Y} \to \mathsf{Z}$ form a Markov chain. The Markov chain starts in one of J states, reduces to M states, then increases back to K states, where $K > M$. That is $|\mathcal{X}| = J$, $|\mathcal{Y}| = M$ and $|\mathcal{Z}| = K$.

(a) Show that $I(\mathsf{X}; \mathsf{Z}) \leq \log M$.
(b) For $M = 2$ and $M = 1$, what is the maximum amount of information that can pass through this bottleneck?

Chapter 4

Source Coding for a Single Source

This chapter introduces source coding, also called compression. The case of compressing a single random variable X, distributed as $p_{\mathsf{X}}(x)$ is considered. In the example of the horse race given in Section 1.1, the single variable is the outcome of the horse race. A code which describes which describes the winner of a horse race with a low average length was given in Table 1.1. This chapter shows how to construct these codes for any source. Topics covered include:

- Describe D-ary codes, particularly prefix codes.
- The Kraft inequality is an inequality satisfied by the codeword length of any prefix code.
- Describe Huffman codes, for a source X with a probability distribution $p_{\mathsf{X}}(x)$.
- Optimal codes for an independent source X. For an optical code C^*, the expected length $L(C^*)$ satisfies :

$$H(\mathsf{X}) \leq L^* \leq H(X) + 1. \tag{4.1}$$

4.1 Source Code Strings

4.1.1 Non-Singular Codes and Uniquely Decodable Codes

This section considers the encoding of source symbols to strings, and the decoding of a string to source symbols. Source codes are strings over a D-ary alphabet $\mathcal{D} = \{0, 1, \ldots, D-1\}$. Using the alphabet $\mathcal{D}$, the set of finite-length strings is $\mathcal{D}^*$. For example, with $D = 3$, $\mathcal{D}^* = \{0, 1, 2, 00, 01, 02, 10, 11, 12, \ldots\}$.

Additional reading: Cover and Thomas, Sections 5.1 to 5.6.

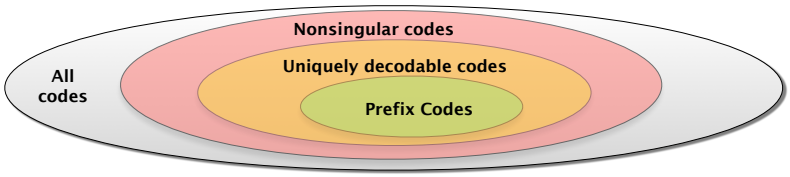


Figure 4.1: Classes of codes.

Definition 4.1. A *source code* C is a mapping from an alphabet $\mathcal{X}$ (the alphabet of the source) to $\mathcal{D}^*$ a set of finite-length strings of symbols from a D-ary alphabet.

$$C : \mathcal{X} \to \mathcal{D}^* \tag{4.2}$$

The source has m symbols, $|\mathcal{X}| = m$, and the code also has m strings.

Let $C(x)$ denote the codeword corresponding to x and let $\ell(x)$ or ℓ_i denote the length of $C(x)$. The maximum length is:

$$\ell_{\max} = \max_{x \in \mathcal{X}} \ell(x). \tag{4.3}$$

Example 4.1. Consider a source with $\mathcal{X} = \{\text{Red, Blue, Yellow, Green}\}$. If $D = 2$, then an example of a source code is:

$x \in C$	$C(x)$	$\ell(x)$
Red	$C(\text{Red}) = 0$	$\ell(\text{Red}) = 1$
Blue	$C(\text{Blue}) = 10$	$\ell(\text{Blue}) = 2$
Yellow	$C(\text{Yellow}) = 110$	$\ell(\text{Yellow}) = 3$
Green	$C(\text{Green}) = 1110$	$\ell(\text{Green}) = 4$

which has $\ell_{\max} = 4$. If $D = 3$, then an example of a source code is:

$x \in C$	$C(x)$	$\ell(x)$
Red	$C(\text{Red}) = 0$	$\ell(\text{Red}) = 1$
Blue	$C(\text{Blue}) = 1$	$\ell(\text{Blue}) = 1$
Yellow	$C(\text{Yellow}) = 21$	$\ell(\text{Yellow}) = 2$
Green	$C(\text{Green}) = 22$	$\ell(\text{Green}) = 2$

which has $\ell_{\max} = 2$.

Three types of codes are explained: nonsingular codes, uniquely decodable codes and prefix (or instantaneous) codes. These codes are hierarchically organized:

prefix codes $\subseteq$ uniquely decodable codes $\subseteq$ nonsingular codes $\subseteq$ all codes

as shown in Fig. 4.1. That is, all prefix codes are uniquely decodable codes, etc.

Definition 4.2. A source code C is a *nonsingular code*, if all distinct source symbol x_i and x_j have the distinct codewords $C(x_i)$ and $C(x_j)$, that is:

$$x_i \neq x_j \implies C(x_i) \neq C(x_j). \tag{4.4}$$

These codes are easily understood from non-examples.

Example 4.2. This is a non-example, the following is *not* a nonsingular code:

$$\begin{aligned} C(\text{Red}) &= 0 \\ C(\text{Blue}) &= 10 \\ C(\text{Yellow}) &= 11 \\ C(\text{Green}) &= 0. \end{aligned}$$

It is not nonsingular because $C(\text{Red}) = C(\text{Green}) = 0$.

Definition 4.3. A concatenation of n codewords in C is called *an extension code* C^+ of the code C. C^+ is a mapping from finite-length strings of $\mathcal{X}$ to finite-length strings of $\mathcal{D}$:

$$C^+(x_1x_2\cdots x_n) = \underbrace{C(x_1)C(x_2)\cdots C(x_n)}_{\text{indicates concatenation}} \tag{4.5}$$

Example 4.3. If $C(x_1) = 00$ and $C(x_2) = 11$ then $C^+(x_1x_2) = 0011$.

Definition 4.4. A code is called *uniquely decodable* if its extension is nonsingular.

Example 4.4. The following is *not* an example of a uniquely decodable code, but it is a nonsingular code:

$$\begin{aligned} C(\text{Red}) &= 0 \\ C(\text{Blue}) &= 10 \\ C(\text{Yellow}) &= 11 \\ C(\text{Green}) &= 01, \end{aligned}$$

This code is not uniquely decodable because 0110 can be interpreted as either $C(\text{Red})C(\text{Yellow})C(\text{Red})$ or $C(\text{Green})C(\text{Blue})$.

4.1.2 Prefix Codes

For the rest of this chapter, only prefix codes are considered, which are described in this subsection.

Definition 4.5. A source code C is a *prefix code* (or *instantaneous code*), if no codeword has a prefix of another codeword.

Example 4.5. The following is *not* a prefix code, but it is a uniquely decodable:

$$\begin{aligned} C(\text{Red}) &= 0 \\ C(\text{Blue}) &= 01 \\ C(\text{Yellow}) &= 011 \\ C(\text{Green}) &= 111, \end{aligned}$$

This code is uniquely decodable because for example the sequence 0111111 is correctly decoded as $C(\text{Red})C(\text{Green})C(\text{Green})$. But this code is not a prefix code, because the correct sequence cannot be determined until all symbols are received. For example, the sequence could also be partially interpreted as $C(\text{Blue})C(\text{Green}) + 11$ or $C(\text{Yellow})C(\text{Green}) + 1$. In addition, $C(\text{Yellow})$ has $C(\text{Blue})$ as a prefix, so this is not a prefix code.

Example 4.6. The following is a prefix code:

$$\begin{aligned} C(\text{Red}) &= 0 \\ C(\text{Blue}) &= 10 \\ C(\text{Yellow}) &= 110 \\ C(\text{Green}) &= 111. \end{aligned}$$

It can be seen that no codeword is the prefix of any other codeword. Then, each codeword can be decoded uniquely without having to look at the following symbols.

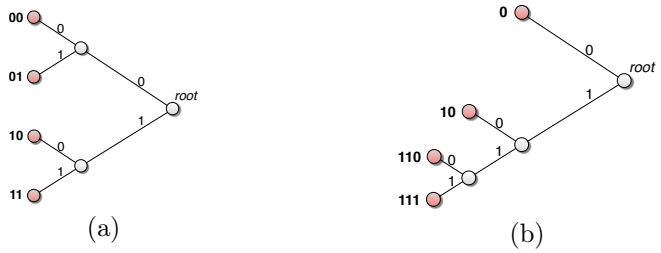
SSQ 4.1. For each source code, is it non-singular? Uniquely decodable? A prefix code?

- $C_1 = \{000, 10, 00, 11\}$.
- $C_2 = \{00, 02, 1, 22, 123\}$.
- $C_3 = \{0, 01\}$.

Prefix codes can be represented by a code tree

Definition 4.6. A *code tree* is a D-ary tree representing a D-ary prefix code. The children of the root node correspond to the first codeword symbol. Successive children correspond to successive codeword symbols. The tree's leaves represent the codewords.

Example 4.7. Consider a code with four codewords of equal length: 00, 01, 10, 11. The code tree for this code is shown in the figure (a) below. The four codewords are red leaves of the trees. Figure (b) below shows the code tree for Example 4.6.

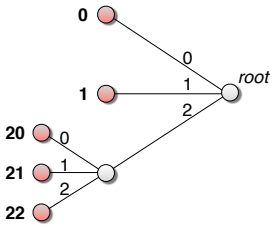


D-ary trees can be constructed for D-ary codes as well.

Example 4.8. Consider a ternary $D = 3$ code with codewords

$$0, 1, 20, 21, 22.$$

The corresponding code tree is a ternary tree, shown in this figure:



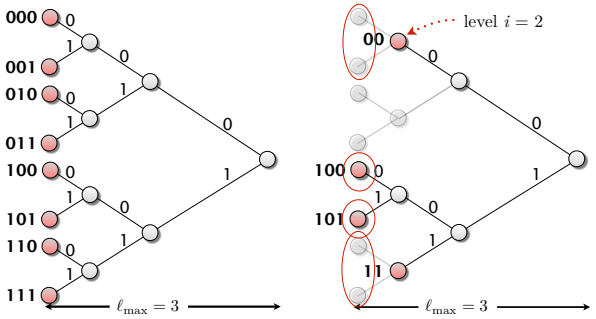
4.2 Kraft Inequality

Recall that the source X produces one symbol from the alphabet $\mathcal{X} = \{1, 2, \ldots, m\}$. Each symbol x is mapped to a codeword $C(x)$, from a prefix code. The length of codeword $C(x)$ is $\ell(x)$. In this section, the probability distribution on X is not used.

Proposition 4.1. *Kraft Inequality.* For any prefix code over an alphabet of size D, the codeword lengths $\ell(1), \ell(2), \ldots, \ell(m)$ must satisfy the inequality:

$$\sum_{x \in \mathcal{X}} D^{-\ell(x)} \leq 1 \tag{4.6}$$

Conversely, given $\ell(x)$ that satisfy this inequality, there exists a prefix code with these words lengths.



(a) (b)

Figure 4.2: Proving Kraft Inequality. (a) Equal length codewords. (b) Non-equal length codewords.

The Kraft inequality also holds for uniquely decodable codes, but we concentrate on prefix codes.

Example 4.9. Verify the Kraft inequality for the binary and non binary codes in the table below.

$D = 2$		$D = 3$	
$C(x)$	$\ell(x)$	$C(x)$	$\ell(x)$
$C(\text{Red}) = 0$	$\ell(\text{Red}) = 1$	$C(\text{Red}) = 0$	$\ell(\text{Red}) = 1$
$C(\text{Blue}) = 10$	$\ell(\text{Blue}) = 2$	$C(\text{Blue}) = 1$	$\ell(\text{Blue}) = 1$
$C(\text{Yellow}) = 110$	$\ell(\text{Yellow}) = 3$	$C(\text{Yellow}) = 21$	$\ell(\text{Yellow}) = 2$
$C(\text{Green}) = 1110$	$\ell(\text{Green}) = 4$	$C(\text{Green}) = 22$	$\ell(\text{Green}) = 2$

For the $D = 2$ case, we see $2^{-1} + 2^{-2} + 2^{-3} + 2^{-4} = \frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \frac{1}{16} = \frac{15}{16} \leq 1$.

For the $D = 3$ case, we have $\frac{1}{3} + \frac{1}{3} + \frac{1}{9} + \frac{1}{9} = \frac{8}{9} \leq 1$.

Example 4.10. Consider the $D = 2$ binary code in the table below.

C(Red)=0
C(Blue)=10
C(Yellow)=11
C(Green)=01

In this case, the Kraft in equality $\frac{1}{2} + \frac{1}{4} + \frac{1}{4} + \frac{1}{4} = \frac{5}{4} \not\leq 1$ is not satisfied. This is because the code is not uniquely decodable — the Kraft inequality is valid only for prefix codes and uniquely decodable codes.

Proof of Kraft Inequality The Kraft inequality is proved for $D = 2$, and pictures are used to illustrate the ideas. First consider an equal-length code, each $\ell(x) = \ell_{\max}$. Fig. 4.2-(a) shows a binary tree, where a codeword is a path

from root to leaf. Because of the prefix condition, no codeword ancestor is a codeword. There are $m \leq 2^{\ell_{\max}}$ codewords (equality if the codebook is all binary sequences of length $\ell_{\max}$). Clearly the Kraft inequality is satisfied in this case:

$$\ell(x) = \ell_{\max} \tag{4.7}$$

$$2^{-\ell(x)} = 2^{-\ell_{\max}} \tag{4.8}$$

$$\sum_{x\in\mathcal{X}} 2^{-\ell(x)} = m2^{-\ell_{\max}} \qquad \text{Number of codewords is } m \tag{4.9}$$

$$\leq 2^{\ell_{\max}} 2^{-\ell_{\max}} \tag{4.10}$$

$$\sum_{x\in\mathcal{X}} 2^{-\ell(x)} \leq 1, \tag{4.11}$$

which is the Kraft inequality for $D = 2$.

Now, consider a non equal length code, illustrated in Fig. 4.2 (b). The maximum length is $\ell_{\max}$. Short codewords are internal nodes. Consider a codeword node at level i. It has descendants at level $\ell_{\max}$ (these decedents are not codewords, but nodes in the tree). The number of descendants of $C(x)$ is $2^{\ell_{\max}-\ell(x)}$. Sum of the codeword length of all decedents is,

$$\sum_{x\in\mathcal{X}} 2^{\ell_{\max}-\ell(x)} \tag{4.12}$$

which cannot be greater than the total number of leaf nodes in the tree $2^{\ell_{\max}}$:

$$\sum 2^{\ell_{\max}-\ell(x)} \leq 2^{\ell_{\max}} \tag{4.13}$$

$$\sum 2^{-\ell(x)} \leq 1 \tag{4.14}$$

This shows that given a prefix code, it must satisfy the Kraft inequality. Conversely, given $\ell(1), \ell(2), \ldots$ that satisfy $\sum 2^{-\ell(x)} \leq 1$, show that a code can be constructed. Since we can always construct a tree, it means we can always construct a code. In addition the proof is readily generalized from $D = 2$ to arbitrary D. □

4.3 Huffman Codes

The previous section did not consider the distribution on the source X. Now, this section considers a source X with probability distribution $p_{\mathsf{X}}(x)$ over $\mathcal{X}$. Huffman coding coding, an optimal source code construction technique, is described. Huffman codes are an important type of entropy coding Wikipedia: Entropy Coding

4.3.1 Expected Length of Codes

Definition 4.7. The *expected length* $L(C)$ of a source code C for a random variable X with probability mass function $p_{\mathsf{X}}(x), x \in \mathcal{X}$ is given by:

$$L(C) \quad = \quad \sum_{x\in\mathcal{X}} p_{\mathsf{X}}(x)\ell(x) \tag{4.15}$$

For binary codes, the units of $L(C)$ are bits/source symbol.

Example 4.11. The table below shows a source X with probability distribution $p_{\mathsf{X}}(x)$, a binary code for this source, and a ternary code for this source.

$C_2(x), D = 2$	$C_3(x), D = 3$	
$C_2(\text{Red}) = 0$	$C_3(\text{Red}) = 0$	$p_{\mathsf{X}}(\text{Red}) = 1/2$
$C_2(\text{Blue}) = 10$	$C_3(\text{Blue}) = 1$	$p_{\mathsf{X}}(\text{Blue}) = 1/4$
$C_2(\text{Yellow}) = 110$	$C_3(\text{Yellow}) = 21$	$p_{\mathsf{X}}(\text{Yellow}) = 1/8$
$C_2(\text{Green}) = 1110$	$C_3(\text{Green}) = 22$	$p_{\mathsf{X}}(\text{Green}) = 1/8$

For the $D = 2$ code, the expected length is:

$$L(C_2) = 1\frac{1}{2} + 2\frac{1}{4} + 3\frac{1}{8} + 4\frac{1}{8} = \frac{15}{8} \text{ bits/source symbol.} \tag{4.16}$$

For the $D = 3$ code, the expected length is:

$$L(C_3) = 1\frac{1}{2} + 1\frac{1}{4} + 2\frac{1}{8} + 2\frac{1}{8} = \frac{5}{4}. \tag{4.17}$$

SSQ 4.2. Find $L(C)$ for the following source code:

$\Pr(\text{Apple}) = 1/3$	$C(\text{Apple}) = 00$
$\Pr(\text{Banana}) = 1/3$	$C(\text{Banana}) = 01$
$\Pr(\text{Cherry}) = 1/9$	$C(\text{Cherry}) = 10$
$\Pr(\text{Durian}) = 1/9$	$C(\text{Durian}) = 110$
$\Pr(\text{Elderberry}) = 1/9$	$C(\text{Elderberry}) = 111$

4.3.2 Huffman Codes

Huffman codes are optimal prefix codes. Huffman codes are constructed using tree, then by labeling the tree to obtain the codewords. Construction begins by combining the *two least likely symbols*, to obtain a set of with $m - 1$ source symbols. Then proceed recursively, until only one symbol with probability 1 remains. In the algorithm below, we use $p_x = p_{\mathsf{X}}(x)$ as an abbreviation.

Construction of binary Huffman codes:

1. Input is

$$\mathcal{X} = \{1, 2, \ldots, m\} \tag{4.18}$$
$$p_1 \geq p_2 \geq \cdots \geq p_{m-1} \geq p_m \tag{4.19}$$

2. Combine the two least likely symbols:

$$(m-1, m) \to m'$$
$$p_{m-1} + p_m \to p'$$

3. Repeat step 2 on:

$$\mathcal{X}' = 1, 2, 3, ..., m-2, m-1 \quad \left(|\mathcal{X}'| = |\mathcal{X}| - 1\right)$$
$$p_1 \geq p_2 \geq \cdots \geq p_{m-2} \geq p_{m-2} \geq p_{m-1}$$

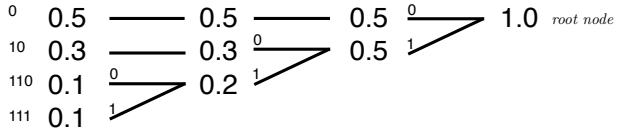
where p' replaces one of the $p_1, \ldots, p_{m-1}$ values. Repeat until only one node with probability 1.0 remains.

Codebook construction:

4. Label the branch of each node with (0,1)
5. Each codeword is a sequence of labels for that leaf node.

Example 4.12. Construct a binary Huffman code for a source with $p_{\mathsf{X}}(x) = (0.5, 0.3, 0.1, 0.1)$.

The construction procedure is shown this figure.



Begin by writing the probabilities in descending order. Combine the smallest two, 0.1 and 0.1 to form a new node with probability 0.2. Now the three probabilities are $0.5, 0.3$ and 0.2. Continuing, combine the smallest two, 0.3 and 0.2 to form a new node with probability of 0.5. The two remaining nodes are combined to form the root node with probability 1.0.

Now, label each separating branch with 0 and 1. The path from the root node to the leaf node is the codeword:

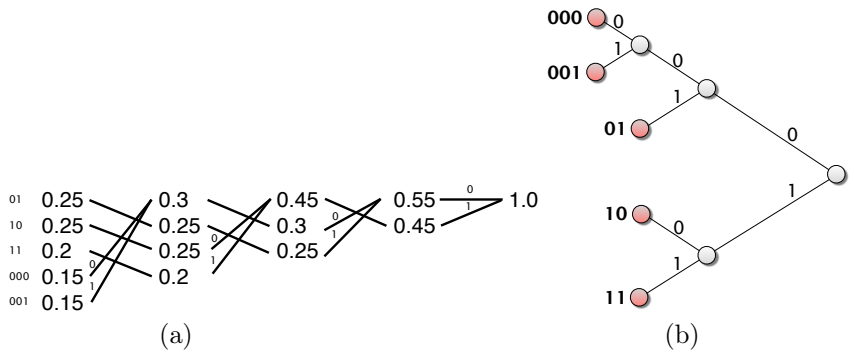
$p_{\mathsf{X}}(x)$	codeword	$\ell(x)$
0.5	0	1
0.3	10	2
0.1	110	3
0.1	110	3

The expected length is:

$$L(C) = \sum_{x \in \mathcal{X}} p_{\mathsf{X}}(x)\ell(x) = 0.5 \cdot 1 + 0.3 \cdot 2 + 0.1 \cdot 3 + 0.1 \cdot 3$$
$$= 1.7 \text{ bits/source symbol}$$

Example 4.13. Consider a random variable X with distribution $p_{\mathsf{X}}(x) = (0.25, 0.25, 0.2, 0.15, 0.15)$. Construct a Huffman code and find the expected length.

The first step is to combine symbols 4 and 5 into a single symbol with probability 0.3. The construction proceeds recursively. Figure (a) below shows the resulting tree and a possible labeling. Figure (b) more clearly expresses the code as a tree.



The resulting codebook is given by

x	$p_{\mathsf{X}}(x)$	$C(x)$
1	0.25	01
2	0.25	10
3	0.2	11
4	0.15	000
5	0.15	001

The expected length of this code is:

$$L(C) = \sum_{x \in \mathcal{X}} p_{\mathsf{X}}(x)\ell(x) \tag{4.20}$$

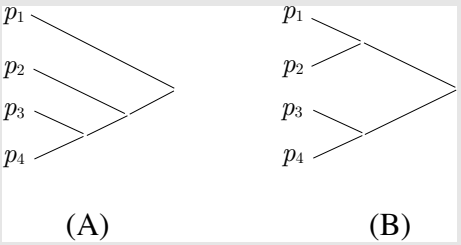
$$= 2 \cdot (0.25 \cdot 2) + 0.2 \cdot 2 + 2 \cdot (0.15 \cdot 3) \tag{4.21}$$

$$= 2.3 \text{ bits/source symbol} \tag{4.22}$$

Some properties of Huffman codes:

- Binary Huffman codes are optimal in the sense of minimizing $L(C)$.
- A Huffman code is not unique. If all the bits in C are inverted, then another optimal code with the same $L(C)$ is obtained. Or, two codewords of the same length can be exchanged without changing $L(C)$.
- The codewords for symbols m and $m-1$ are the same length, and are the longest codewords; other symbols may also have this same longest length. See codewords for 4 and 5 in Example 4.12.

SSQ 4.3. Distinct Huffman codes for $|\mathcal{X}| = 4$. Consider two possible trees:



1. Construct a binary Huffman code for

$$\{p_1, p_2, p_3, p_4\} = \{0.6, 0.17, 0.15, 0.08\}. \tag{4.23}$$

Which tree above corresponds to the Huffman code? Compute the expected number of bits per symbol, $L(C)$.

2. Construct a binary Huffman code for

$$\{p_1, p_2, p_3, p_4\} = \{0.3, 0.26, 0.22, 0.22\}. \tag{4.24}$$

Which tree above corresponds to the Huffman code? Compute the expected number of bits per symbol, $L(C)$.

4.3.3 Non-binary Huffman Codes

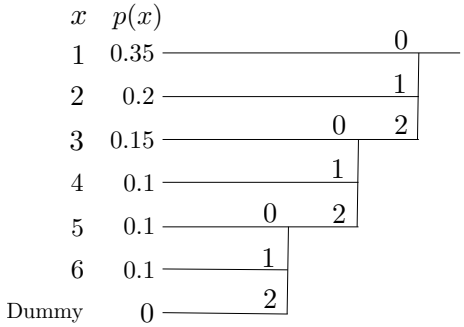
Non-binary Huffman codes with can be constructed as well using a D-ary tree. When constructing a non-binary $D \geq 3$ code, we want to combine D symbols at each step, particularly the last step. In that case, we need to start with with $k(D-1)+1$ symbols, adding some dummy symbols of probability 0. The number of dummy symbols to add is

$$(1 - |\mathcal{X}|) \bmod (D-1) \tag{4.25}$$

or equivalently $(D-1) - \big((|\mathcal{X}| - D) \bmod (D-1)\big)$.

Example 4.14. Construct[1] a ternary Huffman code for the input distribution $p_{\mathsf{X}}(x) = \{0.35, 0.2, 0.15, 0.1, 0.1, 0.1\}$. Since $D = 3$ and $|\mathcal{X}| = 6$, the number of dummy symbols is $-5 \bmod 2 = 1$. We follow the binary Huffman procedure, except that we combine three symbols in each step. This figure shows the dummy symbol and the tree:

[1] Example provided by former TA Javier Cuadros.



The resulting code is:

x	$C(x)$
1	0
2	1
3	20
4	21
5	220
6	221

This code has expected length:

$$L(C) = 1 \cdot (0.35 + 0.2) + 2 \cdot (0.15 + 0.1) + 3 \cdot (0.1 + 0.1) \tag{4.26}$$

$$= 1.65 \text{ ternary symbols/source symbol} \tag{4.27}$$

4.4 Bounds on length of optimal source codes

This section gives a lower bound on the expected length of any code. The expected length of a code is no less than the entropy of the source. An upper bound on the expected length of the optimal code is also given.

4.4.1 Entropy bound on single-variable compression

Definition 4.8. Given a source X distributed as $p_{\mathsf{X}}(x)$, the *base-D entropy* $H_D(\mathsf{X})$ is:

$$H_D(\mathsf{X}) = -\sum_{x \in \mathcal{X}} p_{\mathsf{X}}(x) \log_D p_{\mathsf{X}}(x) \tag{4.28}$$

Definition 4.9. A probability distribution is called *D-adic* if each probability is equal to D^{-n} for some non-negative integer n.

An example of a 2-adic distribution is $\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{8}$. An example of a 3-adic distribution is $\frac{1}{3}, \frac{1}{3}, \frac{1}{9}, \frac{1}{9}, \frac{1}{9}$.

We are interested in an optimal code C^* in the sense that the expected length $L(C^*)$ is as small as possible.

Definition 4.10. A code C^* with lengths $\ell^*(1), \ell^*(2), \ldots$ and probabilities $p_{\mathsf{X}}(1), p_{\mathsf{X}}(2), \ldots$ is an *optimal code* if:

$$L(C^*) = \sum_{x \in \mathcal{X}} p_{\mathsf{X}}(x) \ell^*(x) \tag{4.29}$$

is minimal.

Now we have the main result:

Proposition 4.2. *Entropy bound on single-variable compression* Let $\ell^*(1)$, $\ell^*(2), \ldots, \ell^*(m)$ be optimal codewords lengths for source X distributed as $p_{\mathsf{X}}(x)$ and a D-ary alphabet, and let $L(C^*)$ be the expected length. Then $L(C^*)$ satisfies:

$$H_D(\mathsf{X}) \leq L(C^*) \leq H_D(\mathsf{X}) + 1. \tag{4.30}$$

The lower bounds holds with equality if and only if $p_{\mathsf{X}}(x)$ is D-adic.

Since an optimal code C^* satisfies $H_D(\mathsf{X}) \leq L(C^*)$, then any other code C' with $L(C^*) \leq L(C')$ must also satisfy this inequality. The upper bound and lower bound are proved separately in the following subsections.

4.4.2 Proof of lower bound

Here we prove $H_D(\mathsf{X}) \leq L$, where L is the expected length of any code.

Proof Here we take $p_i = p_{\mathsf{X}}(i)$ and $\ell_i = \ell(i)$. First, define c and r_i:

$$c = \sum_{j \in \mathcal{X}} D^{-\ell_j} \tag{4.31}$$

$$r_i = \frac{D^{-\ell_i}}{\sum_{j \in \mathcal{X}} D^{-\ell_j}} = \frac{D^{-\ell_i}}{c} \tag{4.32}$$

Note $c \leq 1$ by the Kraft inequality, Proposition 4.1.

Prove $H(\mathsf{X}) \leq L$, by writing $L - H_D(\mathsf{X})$ as:

$$\begin{aligned}
L - H_D(\mathsf{X}) &= \sum_{i \in \mathcal{X}} p_i \ell_i + \sum_{i \in \mathcal{X}} p_i \log_D p_i && (4.33)\\
&= \sum_{i \in \mathcal{X}} p_i \log_D D^{\ell_i} + \sum_{i \in \mathcal{X}} p_i \log_D p_i && (4.34)\\
&= \sum_{i \in \mathcal{X}} p_i \log_D \left(\frac{p_i}{D^{-\ell_i}} \cdot \frac{c}{c} \right) && (4.35)\\
&= \sum_{i \in \mathcal{X}} p_i \log_D \frac{p_i}{r_i} + \sum_{i \in \mathcal{X}} p_i \log_D \frac{1}{c} && (4.36)\\
&= D(p||r) + \log \frac{1}{c} && (4.37)\\
&\geq 0 && (4.38)
\end{aligned}$$

since the KL divergence is non-negative and $c \leq 1$ by the Kraft inequality. Equality holds if and only if p_i is D-adic, that is $p_i = D^{-\ell_i}$:

$$c = \sum_{j \in \mathcal{X}} D^{-\ell_j} = \sum_i p_i = 1 \tag{4.39}$$

$$\log \frac{1}{c} = 0 \text{ and} \tag{4.40}$$

$$r_i = \frac{D^{-\ell_i}}{c} = p_i \tag{4.41}$$

Since $r_i = p_i$ for all i, $D(p||r) = 0$. Since $D(p||r)$ and $\log \frac{1}{c}$ are both 0, equality holds.

4.4.3 Proof of upper bound

Here $L(C^*) \leq H_D(\mathsf{X}) + 1$ is proved, by showing that a "good" code C^{good} achieves this bound.

Consider a good code C^{good} for which the lengths are given by:

$$l(x) = \lceil -\log p(x) \rceil. \tag{4.42}$$

where $\lceil t \rceil$ rounds t up to the nearest integer, that is the least integer greater than or equal to t. This C^{good} is a good code, but may not be optimal.To show that C^{good} with lengths (4.42) exists, we show that these lengths satisfy the Kraft inequality:

$$\sum_{x \in \mathcal{X}} D^{-\lceil \log \frac{1}{p(x)} \rceil} \leq \sum_{x \in \mathcal{X}} D^{-\log \frac{1}{p(x)}} = \sum_{x \in \mathcal{X}} p(x) = 1 \tag{4.43}$$

The converse of the Kraft inequality states that given $\ell(x)$ satisfying the inequality, that there must exist a code with these lengths. Thus, there exists a code with lengths given by (4.42).

For any t, the inequality $\lceil t \rceil < t + 1$ holds. Using this:

$$\ell(x) < \log \frac{1}{p(x)} + 1 \tag{4.44}$$

$$\sum_{x \in \mathcal{X}} p(x)\ell(x) < \sum_{x \in \mathcal{X}} p(x) \log \frac{1}{p(x)} + \sum_{x \in \mathcal{X}} p(x) \tag{4.45}$$

$$L(C^{\text{good}}) < H(\mathsf{X}) + 1 \tag{4.46}$$

Finally, if C^{good} satisfies this condition, then an optimal code C^* must also satisfy it, that is $L(C^*) \leq L(C^{\text{good}})$ implies $L(C^*) < H(\mathsf{X}) + 1$. □

The upper bound is important for the following reason. For a source, what is the lowest possible expected length for compression? In general, we would like to achieve $H_D(\mathsf{X})$, but we cannot unless the source is D-adic. But, it is always possible to achieve at least $H_D(\mathsf{X}) + 1$. In other words, the overhead beyond $H_D(\mathsf{X})$ is one bit. This overhead can be eliminated by spreading the code over many symbols $\mathsf{X}_1, \mathsf{X}_2, \ldots, \mathsf{X}_n$, which is the subject of Chapter 5.

4.4.4 KL is the Cost of Miscoding

In Section 3.2, KL divergence was described as a measure between a true distribution and an approximate distribution. In addition, the KL divergence is a measure of the information lost when q is used to approximate p, as the following cost of miscoding example shows.

To construct an optimal Huffman source code, we need to know the source distribution p. Suppose p is not known, and instead we construct a source code C using another source q. Since C is not optimized for the source p, the expected length may increase. Interestingly, the KL divergence can be used to describe this increase. $H(p)+D(p||q)$ bits on average are required to describe the random variable following p, when using a code for q. In other words, a code designed for a source q has an expected length of $H(p)+D(p||q)$ when used to compress a source with distribution p. The KL divergence is the expected number of extra bits required for a source code sampled from p, when using a code designed for q (rather than using a code designed for p).

4.5 Exercises

4.1 Consider the following random variable distribution:

$$p_{\mathsf{X}}(x) = (\frac{1}{21}, \frac{1}{21}, \frac{2}{21}, \frac{4}{21}, \frac{6}{21}, \frac{7}{21}). \tag{4.47}$$

(a) Find a binary Huffman code
(b) Find a ternary Huffman code
(c) Calculate $L = \sum_i p_{\mathsf{X}}(x)\ell_i$ for each case.

4.2 The source coding theorem shows that the optimal code for a random variable X has an expected length less than $H(\mathsf{X}) + 1$. Give an example of a random variable for which the expected length of the optimal code is close to $H(\mathsf{X}) + 1$ (that is, for any $\epsilon > 0$, construct a distribution for which the optimal code has $L > H(\mathsf{X}) + 1 - \epsilon$).

4.3 KL divergence $D(p||q)$ is the cost of miscoding. Consider a source X distributed as p. The optimal code has expected length $H(\mathsf{X})$. If instead of the optimal code, we used the code optimal for q, then the expected length increases to $H(\mathsf{X})+D(p||q)$. Thus, $D(p||q)$ is the cost of miscoding.

Let the random variable X have $\mathcal{X} = \{1, 2, 3, 4, 5\}$. Consider two distributions p_i and q_i on X:

Symbol	p_i	q_i
1	$\frac{1}{2}$	$\frac{1}{2}$
2	$\frac{1}{4}$	$\frac{1}{8}$
3	$\frac{1}{8}$	$\frac{1}{8}$
4	$\frac{1}{16}$	$\frac{1}{8}$
5	$\frac{1}{16}$	$\frac{1}{8}$

(a) Calculate $h(p)$, $h(q)$, $D(p||q)$ and $D(q||p)$.

(b) Find a Huffman code C_1 and its expected length $L(C_1)$ for source p.
(c) Find a Huffman code C_2 and its expected length $L(C_2)$ for source q.
(d) Show that $L(C_1)$ and $L(C_2)$ from the previous step satisfy the entropy bound.
(e) Now assume that we use code C_2 when the distribution is p. What is the average length of the codeword? By how much does it exceed the entropy p?
(f) What is the loss if we use code C_1 when the distribution is q?

4.4 Let source X be distributed according p_x for $x = 1, 2, \ldots, m$. C is a D-ary code for X with lengths ℓ_x for $x = 1, 2, \ldots, m$. Minimize $L(C)$ using Lagrange multipliers. That is, find $\ell_1, \ldots, \ell_m$ that minimize:

$$\min_{\ell_1,\ldots,\ell_m} \sum_{x=1}^{m} p_x \ell_x \tag{4.49}$$

subject to the restriction that the code satisfies the Kraft inequality:

$$\sum_{x=1}^{m} D^{-\ell_x} \leq 1. \tag{4.50}$$

To use Lagrange multipliers, ignore the restriction that ℓ_x are integers and assume ℓ_x are non-negative real numbers.

4.5 A random variable X takes on m values and has entropy $H_3(\mathsf{X})$. A ternary code C is found for this source, which has average length:

$$L(C) = H_3(\mathsf{X}) \tag{4.60}$$

(a) What can you say about the distribution on X? Give an example of a distribution for which $L(C) = H_3(\mathsf{X})$ holds.
(b) Show that m is odd.

4.6 More Huffman Codes

(a) Construct a *binary* Huffman code for $\{p_1, p_2, p_3, p_4, p_5, p_6\} = \{0.35, 0.25, 0.15, 0.13, 0.11, 0.$ Write the Huffman tree and write the code in a table.
(b) For the same probability distribution as part ((a)), construct a *ternary* (3-level symbol) Huffman code. Write the Huffman tree and write the code in a table.

4.7 Which of the following cannot be Huffman codes? If not, write the reason.

(a) {1,01,11}
(b) {00,10,01,110}
(c) {01,10}

Chapter 5

Source Coding for Memoryless Sources

This chapter considers source coding, also called compression, of n random sources:

$$\mathbf{X} = (\mathsf{X}_1, \mathsf{X}_2, \ldots, \mathsf{X}_n) \tag{5.1}$$

where the X_i are independent and identically distributed, that is, the source is memoryless. This can be called a *vector source* of n symbols. All the symbols are compressed together, that is the vector $\mathbf{x}$ is compressed to a codeword denoted $C(\mathbf{x})$. This is lossless source coding, which means that given the codeword $C(\mathbf{x})$, we can recover $\mathbf{x}$ exactly.

The central result is that there exists a source code C with code rate R, such that:

$$H(\mathsf{X}) \leq R \leq H(\mathsf{X}) + \epsilon', \tag{5.2}$$

where $\epsilon' > 0$ can be made as small as we want. The code rate R is expected length of the codeword, divided by the number of source symbols n.

This improves the result for single $n = 1$ sources from Chapter 4. That is $H(\mathsf{X}) \leq R \leq H(\mathsf{X}) + 1$, that is $+1$ is replaced by a small number ϵ'. Instead of possibly losing one bit with single-variable compression, it is possible to achieve a compression rate which is close to the entropy. However, in order to achieve this, we must let n get large, and in particular we let $n \to \infty$.

To achieve this, this lecture covers the following:

- the code rate of vector source coding and the Vector Source Coding Theorem,
- sample entropy of a sequence and typical sets, which are the high-probability sequences

Additional reading: Cover and Thomas, Chapter 3.

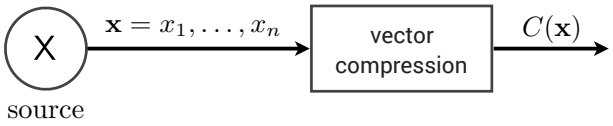


Figure 5.1: Vector compression of a source. The source produces n symbols $\mathbf{x} = x_1, x_2, \ldots, x_n$. These are jointly compressed to a codeword denoted $C(\mathbf{x})$.

- the asymptotic equipartition property (AEP) and its properties
- describe a "vector compression scheme" and a theorem that its code rate satisfies (5.2) prove the theorem using the AEP.

The importance of this chapter is that it shows how to prove a typical theorem in information theory. We study the AEP for the simplest case, single variable compression. More advanced proofs, such as the channel coding theorem, follow the same basic principles.

5.1 Vector Source Coding

Chapter 4 studied source coding for a single source. The codebook mapped symbols from this source to D-ary codewords of variable length. This chapter considers source coding for a vector source. The codebook will map symbols from a vector source to codewords of variable length.

The vector source is

$$\mathbf{X} = (\mathsf{X}_1, \mathsf{X}_2, \ldots, \mathsf{X}_n), \tag{5.3}$$

from the alphabet $\mathcal{X}^n$, with iid $\mathsf{X}_i \sim p_{\mathsf{X}}(x)$ having entropy $H(\mathsf{X})$. The source X_i is assumed to be independent and identically distributed, or memoryless, that is the joint distribution is:

$$p_{\mathbf{X}}(\mathbf{x}) = \prod_{i=1}^{n} p_{\mathsf{X}}(x_i) \tag{5.4}$$

For any input $\mathbf{x} = x_1, x_2, \ldots, x_n$, vector compression produces a codeword $C(\mathbf{x})$, as shown in Fig. 5.1. The length of codeword $C(\mathbf{x})$ is $\ell(\mathbf{x})$.

Definition 5.1. For length-n vectors, a *vector source code* C is a mapping from an alphabet $\mathcal{X}^n$ (the alphabet of the source) to $\mathcal{D}^*$ a set of finite-length strings of symbols from a D-ary alphabet.

$$C : \mathcal{X}^n \to \mathcal{D}^* \tag{5.5}$$

This definition is analogous to Definition 4.1 for single sources.

Example 5.1. Consider a vector binary source with $n = 3$ and $p_{\mathsf{X}}(0) = 0.9$. An example of a vector source code is given in the table below, along with $p_{\mathbf{X}}(\mathbf{x})$.

$\mathbf{x}$	$p_{\mathbf{X}}(\mathbf{x})$	$C(\mathbf{x})$
0 0 0	0.729	0
1 0 0	0.081	100
0 1 0	0.081	101
0 0 1	0.081	110
1 1 0	0.009	11100
1 0 1	0.009	11101
0 1 1	0.009	11110
1 1 1	0.001	11111

There is of course a decompression function which maps a codeword $C(\mathbf{x})$ back to the original sequence, this is also called reconstruction. This is lossless source coding, since we reconstruct $\mathbf{x}$ exactly. In the example above, if we get the codeword 0, then we can reconstruct the source sequence 000. We do not study reconstruction — it is assumed that it is possible.

"Source coding" means the same thing as compression. We are interested in the compression rate, which is called the code rate. The code rate is the codeword length per source symbol, on average. For a binary code, a rate less than one indicates there is compression. In the example, $\mathbf{x} = 000$ is compressed to $C(\mathbf{x}) = 0$ — since 3 bits are reduced to 1 bit, this sequence contributes $1/3$ to the code rate. On the other hand, $\mathbf{x} = 111$ is "compressed" to $C(\mathbf{x}) = 11111$; this is not compression, and this sequence contributes $5/3$ to the code rate.

Definition 5.2. The *code rate* R for a vector source code C is:

$$R = \frac{1}{n} \sum_{\mathbf{x} \in \mathcal{X}^n} p_{\mathbf{X}}(\mathbf{x}) \ell(\mathbf{x}) \tag{5.6}$$

This definition is an extension of the expected length for a single-source code given in Definition 4.7. Since we are coding over n source symbols, the rate is the expected length divided by the number of source symbols n.

Example 5.2. The rate R of the vector source code in Example 5.1 is given by:

$$\begin{aligned} R &= \frac{1}{3}(1 \cdot 0.729 + 3 \cdot 3 \cdot 0.081 + 3 \cdot 5 \cdot 0.009 + 5 \cdot 0.001) \\ &= 0.5327. \end{aligned}$$

Without a source code, or equivalently $C(\mathbf{x}) = \mathbf{x}$, then the rate is $R = 1$. But by using a source code, the rate is lower $R = 0.5327$. This indicates the amount of compression obtained. Lower rates mean more compression.

Now we are ready to state the main result of this chapter.

Proposition 5.1. *Vector Source Coding Theorem* Let $\mathbf{X} = (\mathsf{X}_1, \ldots, \mathsf{X}_n)$ be n independent and identically distributed random variables X_i, with $\mathsf{X}_i \sim p_{\mathsf{X}}(x)$

having entropy $H(\mathsf{X})$. Let $\epsilon' > 0$. Then, there exists a vector source code with rate R that satisfies:

$$R \leq H(\mathsf{X}) + \epsilon', \tag{5.7}$$

for n sufficiently large.

The proof is given in Section 5.4.2. Here ϵ' is a positive constant that can be made arbitrarily small. It is related to ϵ of the typical set.

Proposition 5.1 says that it is possible to find a code with coding rate (compression rate) R which is very close to $H(\mathsf{X})$. The larger n becomes, the closer we can get to this optimal value. On the other hand, the best possible rate is $H(\mathsf{X})$, and so there exists a code with rate R that satisfies:

$$H(\mathsf{X}) \leq R \leq H(\mathsf{X}) + \epsilon'. \tag{5.8}$$

Example 5.3. The entropy of the source in Example 5.1 is $H(\mathsf{X}) = h(0.9) \approx 0.4690$. The code with $n = 3$ has $R = 0.5327$. But if we let n get larger, then this rate can be reduced towards the ideal 0.4690. The larger n is, the closer we get to the ideal value.

SSQ 5.1. Consider a vector binary source with $n = 3$ and $p_{\mathsf{X}}(0) = 0.8$. For each sequence $\mathbf{x}$, the codeword $C(\mathbf{x})$ is given as:

$\mathbf{x}$	$p_{\mathsf{X}}(\mathbf{x})$	$C(\mathbf{x})$
0 0 0	0.512	0
1 0 0	0.128	100
0 1 0	0.128	101
0 0 1	0.128	110
1 1 0	0.032	11100
1 0 1	0.032	11101
0 1 1	0.032	11110
1 1 1	0.008	11111

What is the rate R for this code? For this source, what is the lowest possible compression rate?

5.2 Sample Entropy and Typical Sets

5.2.1 Sample Entropy

Chapter 2 for random vectors $\mathbf{X} = \mathsf{X}_1\mathsf{X}_2\cdots\mathsf{X}_n$ compared the true mean $E[\mathsf{X}_i]$ with the sample mean of realizations $x_1x_2\cdots x_n$:

$$\overline{x}_n = \frac{1}{n}\sum_{i=1}^{n} x_i. \tag{5.9}$$

As n gets larger, the distribution of the sample mean tends to concentrate around the true mean. For example, if we roll a fair die 6 times and obtain the outcome $6, 4, 4, 2, 3, 4, 5$, the sample mean is 4. But if we roll the die 100 times, the sample mean might be a number like 3.69, which is closer to the true mean of 3.5.

In the same way, there is a sample entropy for a realization $\mathbf{x} = x_1, x_2, \ldots, x_n$ distributed as $p_{\mathbf{X}}(\mathbf{x})$, and the sample entropies will tend to concentrate around the true entropy $H(\mathsf{X})$ as n gets large.

Definition 5.3. For any fixed sequence $\mathbf{x}$, jointly distributed according to $p_{\mathbf{X}}(\mathbf{x})$, the *sample entropy* is defined as:

$$-\frac{1}{n}\log p_{\mathbf{X}}(\mathbf{x}). \tag{5.10}$$

When $\mathsf{X}_1\mathsf{X}_2\ldots\mathsf{X}_n$ are iid with marginal distribution $p_{\mathsf{X}}(x_i)$, the sample entropy of the sequence $\mathbf{x} = x_1x_2\ldots x_n$ is:

$$-\frac{1}{n}\sum_{i=1}^{n}\log p_{\mathsf{X}}(x_i) \tag{5.11}$$

since $p_{\mathbf{X}}(\mathbf{x}) = p_{\mathsf{X}}(x_1)p_{\mathsf{X}}(x_2)\cdots p_{\mathsf{X}}(x_n)$. Note that (5.9) is not unlike (5.11), but an important difference is that to compute the sample entropy, knowledge of $p_{\mathbf{X}}(\mathbf{x})$ is needed.

Example 5.4. Consider $n = 4$ iid binary random variables $\mathsf{X}_1\mathsf{X}_2\mathsf{X}_3\mathsf{X}_4$ with $p_{\mathsf{X}}(x) = [\frac{3}{4}, \frac{1}{4}]$. Find the sample entropy of all possible sequences.

Start with $\mathbf{x} = 0000$. Using (5.11), the sample entropy of $\mathbf{x} = 0000$ is:

$$-\frac{1}{4}\sum_{i=1}^{4}\log p_{\mathsf{X}}(x_i) = -\frac{1}{4}\Big(4\log\frac{3}{4}\Big) \approx 0.4150. \tag{5.12}$$

The sample entropy of $\mathbf{x} = 1000$ is:

$$-\frac{1}{4}\sum_{i=1}^{4}\log p_{\mathsf{X}}(x_i) = -\frac{1}{4}\Big(3\log\frac{3}{4} + \log\frac{1}{4}\Big) = 2 - \frac{3}{4}\log 3 \approx 0.8113. \tag{5.13}$$

Any sequence with 3 zeros and 1 one has the same sample entropy of 0.8113.

The possible values for $p_{\mathbf{X}}(\mathbf{x})$ are:

$$\{0.31641, 0.10547, 0.03516, 0.01172, 0.00391\}, \tag{5.14}$$

which correspond to sequences with 0, 1, 2, 3 and 4 ones, respectively. The corresponding sample entropies are:

$$\{0.4150, 0.8113, 1.2075, 1.6038, 2.0\}. \tag{5.15}$$

A list of all $n = 4$ sequences, their probability $p_{\mathbf{X}}(\mathbf{x})$, and their sample entropy are shown in Table 5.1. □

SSQ 5.2. Consider a sequence of $n = 5$ independent and identically distributed binary random variables, $\mathbf{X} = \mathsf{X}_1, \mathsf{X}_2, \mathsf{X}_3, \mathsf{X}_4, \mathsf{X}_5$ with $p_{\mathsf{X}_i}(0) = 0.8$ and $p_{\mathsf{X}_i}(1) = 0.2$.

Compute $H(\mathsf{X}_i)$. Compute the sample entropy of the sequence $\mathbf{x} = 01001$.

5.2.2 Typical Sets and Typical Sequences

Typical sets and typical sequences are central to proofs in information theory. A sequence $\mathbf{x} = (x_1, x_2, \ldots x_n)$ is an ϵ-typical sequence if its sample entropy (5.11) is within ϵ of the true entropy, $H(\mathsf{X})$. The set of all typical sequences is called the typical set. For a given n and ϵ, the typical set is denoted $\mathcal{T}_\epsilon^{(n)}$.

Let $\mathbf{X} = (\mathsf{X}_1, \mathsf{X}_2, \ldots, \mathsf{X}_n)$ be distributed as $p_{\mathbf{X}}(\mathbf{x})$. The entropy of one symbol is $H(\mathsf{X})$. For a parameter $\epsilon \geq 0$, a sequence $\mathbf{x} \in \mathcal{X}^n$ is called a *typical sequence* if:

$$H(\mathsf{X}) - \epsilon \;\leq\; -\frac{1}{n} \log p_{\mathbf{X}}(\mathbf{x}) \;\leq\; H(\mathsf{X}) + \epsilon. \tag{5.16}$$

That is, $\mathbf{x}$ is a typical sequence if $\mathbf{x}$ has sample entropy which is ϵ-close to the true entropy.

Definition 5.4. The *typical set* $\mathcal{T}_\epsilon^{(n)}$ is the set of sequences $\mathbf{x} \in \mathcal{X}^n$ with sample entropy ϵ-close to the true entropies:

$$\mathcal{T}_\epsilon^{(n)} = \left\{ \mathbf{x} \in \mathcal{X}^n : \left| -\frac{1}{n} \log p_{\mathbf{X}}(\mathbf{x}) - H(\mathsf{X}) \right| < \epsilon \right\} \tag{5.17}$$

The size of the typical set is $|\mathcal{T}_\epsilon^{(n)}|$.

A *typical sequence* is a member of the typical set, for a given value of n and ϵ.

An important questions is: given a sequence drawn at random, what is the probability it is in the typical set? An element drawn at random will be in the typical set with probability $\Pr(\mathbf{X} \in \mathcal{T}_\epsilon^{(n)})$:

$$\Pr(\mathbf{X} \in \mathcal{T}_\epsilon^{(n)}) = \sum_{\mathbf{x} \in \mathcal{T}_\epsilon^{(n)}} p_{\mathbf{X}}(\mathbf{x}). \tag{5.18}$$

It is possible to upper bound and lower bound this probability by first writing (5.16) as:

$$2^{-n(H(\mathsf{X})+\epsilon)} \leq p_{\mathbf{X}}(\mathbf{x}) \leq\; 2^{-n(H(\mathsf{X})-\epsilon)}, \tag{5.19}$$

and then by summing over $\mathbf{x} \in \mathcal{T}_\epsilon^{(n)}$,

$$|\mathcal{T}_\epsilon^{(n)}| 2^{-n(H(\mathsf{X})+\epsilon)} \;\leq\; \Pr(\mathbf{X} \in \mathcal{T}_\epsilon^{(n)}) \;\leq\; |\mathcal{T}_\epsilon^{(n)}| 2^{-n(H(\mathsf{X})-\epsilon)}, \tag{5.20}$$

since $\sum_{\mathbf{x} \in \mathcal{T}_\epsilon^{(n)}} 1 = |\mathcal{T}_\epsilon^{(n)}|$.

Table 5.1: List of all sequences for a binary random variable sequence X^4 with $p = \frac{1}{4}$. * indicates sequences in $\mathcal{T}_{0.01}^{(4)}$. † indicates sequences in $\mathcal{T}_{0.45}^{(4)}$.

$x_1\ x_2\ x_3\ x_4$	$p(x_1, x_2, x_3, x_4)$	$-\frac{1}{n}\log p(x_1, x_2, x_3, x_4)$
0 0 0 0 †	0.31641	0.41504
1 0 0 0 † *	0.10547	0.81128
0 1 0 0 † *	0.10547	0.81128
0 0 1 0 † *	0.10547	0.81128
0 0 0 1 † *	0.10547	0.81128
1 1 0 0 †	0.03516	1.20752
1 0 1 0 †	0.03516	1.20752
0 1 1 0 †	0.03516	1.20752
1 0 0 1 †	0.03516	1.20752
0 1 0 1 †	0.03516	1.20752
0 0 1 1 †	0.03516	1.20752
1 1 1 0	0.01172	1.60376
1 1 0 1	0.01172	1.60376
1 0 1 1	0.01172	1.60376
0 1 1 1	0.01172	1.60376
1 1 1 1	0.00391	2.00000

For binary random vectors, the typical set consists of sequences with $a, a+1, \ldots, d$ ones (where a and d are suitably choosen), and the size of the typical set will be:

$$|\mathcal{T}_\epsilon^{(n)}| = \sum_{i=a}^{d} \binom{n}{i}. \tag{5.21}$$

Example 5.5. Continue Example 5.4 of the binary random vector with $p_{\mathsf{X}}(x) = [\frac{3}{4}, \frac{1}{4}]$ and $n = 4$. Find the typical set, its size, and its probability when (a) $\epsilon = 0.01$ and (b) $\epsilon = 0.45$.

The true entropy $H(\mathsf{X}_i)$ is 0.81128. In general, the typical set $\mathcal{T}_\epsilon^{(n)}$ are those sequences that have sample entropy $-\frac{1}{n}\log p_{\mathbf{X}}(\mathbf{x})$ within ϵ of $H(\mathsf{X})$. (a) For $\epsilon = 0.01$, then the typical set is:

$$\mathcal{T}_{0.01}^{(4)} = \{1000, 0100, 0010, 0001\}, \tag{5.22}$$

because the sample entropy of a sequences with 1 one is 0.88128 as shown in (5.15). The size of the typical set is $|\mathcal{T}_\epsilon^{(n)}| = 4$, and the probability of being in the typical set is:

$$\Pr(\mathbf{X} \in \mathcal{T}_\epsilon^{(n)}) = \sum_{i=1}^{1} \binom{4}{i} (\frac{3}{4})^{4-i} (\frac{1}{4})^i = 0.4219. \tag{5.23}$$

(b) For $\epsilon = 0.45$ the typical set is expanded to include:

$$\mathcal{T}_{0.45}^{(4)} = \{\text{all sequences with 0, 1 or 2 ones}\}. \tag{5.24}$$

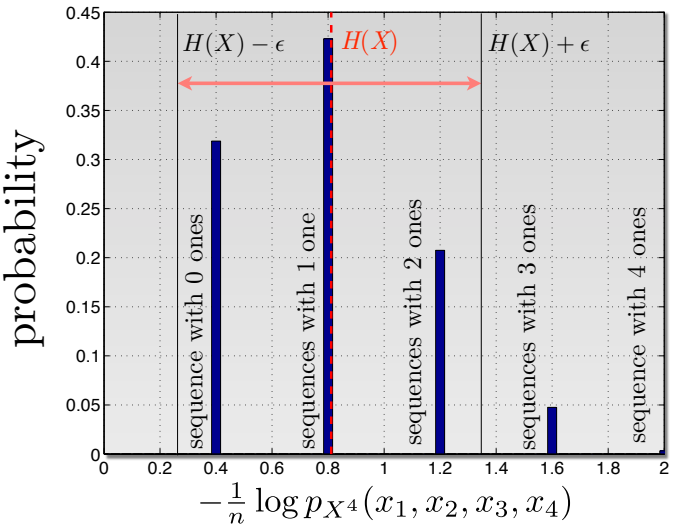


Figure 5.2: Probability distribution versus sample entropy $-\frac{1}{n}\log p_{\mathsf{X}^4}(x_1,x_2,x_3,x_4)$, for sequences of 0, 1, 2, 3 and 4 ones.

because the sample entropies $\{0.41504, 0.81128, 1.20752\}$ are within $\epsilon = 0.45$ of $H(\mathsf{X}) = 0.81128$. The size of the typical set is $|\mathcal{T}^{(4)}_{0.45}| = 11$, and the probability of being in the typical set is:

$$\Pr(\mathbf{X} \in \mathcal{T}^{(n)}_{\epsilon}) = \sum_{i=0}^{3} \binom{4}{i} (\frac{3}{4})^{4-i} (\frac{1}{4})^{i} = 0.9492. \tag{5.25}$$

The probability of each sequence with a given sample entropy is shown in Fig. 5.2.

The set $\mathcal{X}^n$ can be partitioned into two sets, the typical set $\mathcal{T}^{(n)}_{\epsilon}$ and its compliment $\overline{\mathcal{T}}^{(n)}_{\epsilon}$. For example with $n = 4$, $\epsilon = 0.01$, the partition is:

$$\begin{aligned}\mathcal{X}^n &= \mathcal{T}^{(n)}_{\epsilon} \cup \overline{\mathcal{T}}^{(n)}_{\epsilon} \\ &= \Big\{0000, 0001, 0010, 0100, 1000, \\ &\qquad 0011, 0101, 1001, 0110, 1010, 1100, 0111, 1011, 1101, 1110, 1111\Big\}\end{aligned}$$

where red indicate elements of the typical set, and blue indicates elements of its complement.

SSQ 5.3. Consider a sequence of $n = 5$ independent and identically distributed binary random variables, $\mathbf{X} = \mathsf{X}_1, \mathsf{X}_2, \mathsf{X}_3, \mathsf{X}_4, \mathsf{X}_5$ with $p_{\mathsf{X}_i}(0) = 0.8$ and $p_{\mathsf{X}_i}(1) = 0.2$.

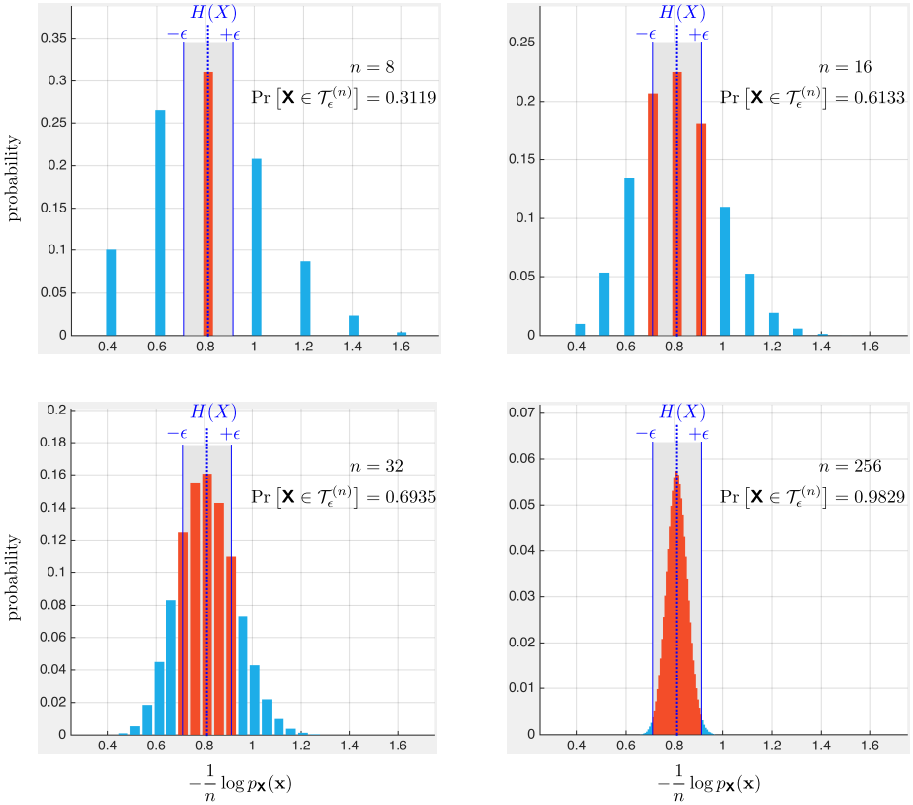


Figure 5.3: Sample entropy of $\mathbf{x}$ versus probability of $\mathbf{x}$, for a $p = \frac{1}{4}$ binary random variable for $n = 8, 16, 32$ and 256. For $\epsilon = 0.1$, those sample entropies in the typical set are shown in red.

Which sequences are in the typical set with $\epsilon = 0.01$? With $\epsilon = 0.45$?

5.3 Asymptotic Equipartition Property

As preparation for proving Proposition 5.1, typical sequences when $n \to \infty$ are considered. The asymptotic equipartition property or AEP, expresses the idea that a randomly drawn sequence will be one of the likely sequences with increasingly high probability:

$$\Pr[\text{likely sequences}] \to 1 \text{ as } n \to \infty. \tag{5.26}$$

A random sequence $\mathbf{X} = \mathsf{X}_1, \mathsf{X}_2, \ldots, \mathsf{X}_n$ will be in the typical set $\mathcal{T}_\epsilon^{(n)}$ with probability that approaches 1 as n becomes large:

$$\lim_{n\to\infty} \Pr[\mathbf{X} \in \mathcal{T}_\epsilon^{(n)}] = 1 \tag{5.27}$$

When we say *asymptotic* we mean "as n becomes asymptotically large."

For a fixed value of ϵ, as n gets larger, the typical sequences will concentrate near the entropy, as the example in Fig. 5.3 shows. As $n \to \infty$ we can make the probability that a randomly drawn sequence is in the typical set approach 1. This will be proven using the law of large numbers in Proposition 5.3

In addition, the size of the typical set can be relatively small, in particular $|\mathcal{T}_\epsilon^{(n)}| \to 2^{nH(\mathsf{X})}$ as $n \to \infty$. Combining these two ideas, we summarize this as "most of the probability is contained in a small set."

The asymptotic equipartition property is the application of the law of large numbers to entropy. Given in Subsection 2.4.4, the law of large numbers states that the sample mean $\overline{\mathsf{X}}_n = \frac{1}{n}\sum_i \mathsf{X}_i$ approaches the true mean $E[\mathsf{X}]$ of the random variable:

$$\lim_{n\to\infty} \Pr\left[|\overline{\mathsf{X}}_n - E[\mathsf{X}]| < \epsilon\right] = 1. \tag{5.28}$$

In the same way, the AEP states that the sample entropy $-\frac{1}{n}\log p_{\mathbf{X}}(\mathsf{X}_1, \mathsf{X}_2, \ldots, \mathsf{X}_n)$ approaches the entropy $H(\mathsf{X})$ of the random variable.

Proposition 5.2. *Asymptotic Equipartition Property* If $\mathsf{X}_1, \mathsf{X}_2, \ldots, \mathsf{X}_n$ are a sequence of n independent and identically distributed random variables with probability distribution $p_{\mathsf{X}}(x)$ then:

$$\lim_{n\to\infty} \Pr[\mathbf{X} \in \mathcal{T}_\epsilon^{(n)}] = 1. \tag{5.29}$$

Proof Begin by defining Y_i and $\overline{\mathsf{Y}}$ as:

$$\mathsf{Y}_i = -\log p_{\mathsf{X}}(\mathsf{X}_i) \text{ and} \tag{5.30}$$

$$\overline{\mathsf{Y}} = \frac{1}{n}\sum_{i=1}^{n} \mathsf{Y}_i. \tag{5.31}$$

First, note that $\overline{\mathsf{Y}}$ can be written as:

$$\overline{\mathsf{Y}} = -\frac{1}{n}\sum_{i=1}^{n}\log p_{\mathsf{X}}(\mathsf{X}_i) \qquad \text{definition of } \mathsf{Y}_i \text{ and } \overline{\mathsf{Y}} \tag{5.32}$$

$$= -\frac{1}{n}\log p_{\mathbf{X}}(\mathsf{X}_1, \mathsf{X}_2, \ldots, \mathsf{X}_n) \quad \text{the } \mathsf{X}_i \text{ are iid .} \tag{5.33}$$

Let $H(\mathsf{X})$ be the entropy of X_i and find $E[\overline{\mathsf{Y}}]$ as:

$$\begin{aligned} E[\overline{\mathsf{Y}}] &= \frac{1}{n}\sum_{i=1}^{n} E[\mathsf{Y}_i] && \text{expectation of a sum} \\ &= -\frac{1}{n}\sum_{i=1}^{n} E[\log p_{\mathsf{X}}(\mathsf{X}_i)] && \text{definition of } \mathsf{Y}_i \\ &= \frac{1}{n}\sum_{i=1}^{n} H(\mathsf{X}_i) && \text{expectation form, eqn. (3.59) on page 67} \\ &= H(\mathsf{X}) && \text{i.i.d. } \mathsf{X}_i \text{ have same } H. \end{aligned} \tag{5.34}$$

By applying the law of large numbers to $\overline{\mathsf{Y}}$, the following statements hold:

$$\begin{aligned} \lim_{n\to\infty} \Pr[|\overline{\mathsf{Y}} - E[\overline{\mathsf{Y}}]| < \epsilon] = 1 \quad & \text{LLN for } \overline{\mathsf{Y}} \\ \lim_{n\to\infty} \Pr\left[|-\frac{1}{n}\log p_{\mathbf{X}}(\mathsf{X}_1, \mathsf{X}_2, \ldots, \mathsf{X}_n) - H(\mathsf{X})| < \epsilon\right] = 1 \quad & \text{by (5.33) and (5.34)} \\ \lim_{n\to\infty} \Pr[\mathbf{X} \in \mathcal{T}_\epsilon^{(n)}] = 1 \quad & \text{Definition 5.4} \end{aligned}$$

which completes the proof. □

Now two properties of the typical set and the AEP are given. In the following, ϵ is a small positive constant that can be made as small as desired by letting n become large.

Proposition 5.3. *Most sequences are typical* For a random vector $\mathbf{X}$, most sequences are typical, that is:

$$\Pr\left[\mathbf{X} \in \mathcal{T}_\epsilon^{(n)}\right] > 1 - \epsilon, \tag{5.35}$$

for n sufficiently large. For the complementary event, $\Pr[\mathbf{X} \in \overline{\mathcal{T}}_\epsilon^{(n)}] \leq \epsilon$.

Proof By the AEP, as $n \to \infty$, $\Pr[\mathbf{X} \in \mathcal{T}_\epsilon^{(n)}] = 1$. So for any small $\delta > 0$, there must be a finite n_0 such that for all $n \geq n_0$:

$$\Pr\left[\mathbf{X} \in \mathcal{T}_\epsilon^{(n)}\right] > 1 - \delta \tag{5.36}$$

holds. Because of this, the restriciton *for n sufficiently large* is required. In the words of Cover and Thomas "Setting $\delta = \epsilon$, we obtain the result. The identification of $\delta = \epsilon$ will conveniently simplify notation later." □

The following proposition says that the number of elements in the typical set is nearly $2^{nH(\mathsf{X})}$.

Proposition 5.4. The size of the typical set is bounded as:

$$(1-\epsilon)2^{n(H(\mathsf{X})-\epsilon)} \leq |\mathcal{T}_\epsilon^{(n)}| \leq 2^{n(H(\mathsf{X})+\epsilon)}. \tag{5.37}$$

The lower bound holds for n sufficiently large.

Proof First, the upper bound is shown as:

$$1 = \sum_{\mathbf{x}\in\mathcal{X}^n} p_{\mathbf{X}}(\mathbf{x}) \geq \sum_{\mathbf{x}\in\mathcal{T}} p_{\mathbf{X}}(\mathbf{x}) \overset{\text{(a)}}{\geq} \sum_{\mathbf{x}\in\mathcal{T}} 2^{-n(H(\mathsf{X})+\epsilon)} = |\mathcal{T}_\epsilon^{(n)}| 2^{-n(H(\mathsf{X})+\epsilon)},$$

where inequality (a) follows from (5.19). The lower bound is shown as:

$$\begin{aligned} 1-\epsilon < \Pr(\mathbf{X}\in\mathcal{T}_\epsilon^{(n)}) & \qquad \text{most sequences are typical} \\ \leq |\mathcal{T}_\epsilon^{(n)}| 2^{-n(H(\mathsf{X})-\epsilon)} & \qquad \text{by eqn (5.20)} \\ (1-\epsilon)2^{n(H(\mathsf{X})-\epsilon)} \leq |\mathcal{T}_\epsilon^{(n)}|, & \end{aligned}$$

which completes the proof. □

5.4 Vector Source Coding

This section describes a specific Vector Compression Scheme, and gives the proof of Proposition 5.1.

5.4.1 Vector Compression Scheme

The Vector Compression Scheme is given as follows. For each $\mathbf{x} \in \mathcal{X}^n$, this scheme forms a codeword $C(\mathbf{x})$. The set of all sequences $\mathcal{X}^n$ can be partitioned into the typical set $\mathcal{T}_\epsilon^{(n)}$ or its complement $\overline{\mathcal{T}}_\epsilon^{(n)}$:

$$\mathcal{X}^n = \mathcal{T}_\epsilon^{(n)} \cup \overline{\mathcal{T}}_\epsilon^{(n)}. \tag{5.38}$$

Those sequences that are in the typical set $\mathcal{T}_\epsilon^{(n)}$ are compressed one way, and those sequences in the complementary set $\overline{\mathcal{T}}_\epsilon^{(n)}$ are compressed another way:

- If $\mathbf{x} \in \mathcal{T}_\epsilon^{(n)}$, encode as follows. The size of the typical set is $|\mathcal{T}_\epsilon^{(n)}|$, and so we can assign each element of $\mathcal{T}_\epsilon^{(n)}$ an index from the set:

  $$\{1, 2, 3, \ldots, |\mathcal{T}_\epsilon^{(n)}|\} \tag{5.39}$$

  (Alternatively, assign an index from the set $\{1, 2, 3, \ldots, 2^{nr}\}$ where $r = \frac{1}{n}\log|\mathcal{T}_\epsilon^{(n)}|$.) Form a codeword $C(\mathbf{x})$ as 0 followed by the binary index, requiring $\log|\mathcal{T}_\epsilon^{(n)}|$.

  The codeword length is

  $$\ell(\mathbf{x}) = \lceil \log|\mathcal{T}_\epsilon^{(n)}| \rceil + 1 \leq \log|\mathcal{T}_\epsilon^{(n)}| + 2, \tag{5.40}$$

  since $|\mathcal{T}_\epsilon^{(n)}|$ may not be a power of 2.

Table 5.2: Example of compression scheme with $n = 4$.

$\mathbf{x}$	$p_{\mathbf{X}}(\mathbf{x})$	in typical set?	$C(\mathbf{x})$	$\ell(\mathbf{x})$
0 0 0 0	0.3164		1 0 0 0 0	5
1 0 0 0	0.1055	*	0 0 0	3
0 1 0 0	0.1055	*	0 1 0	3
1 1 0 0	0.0352		1 1 1 0 0	5
0 0 1 0	0.1055	*	0 0 1	3
1 0 1 0	0.0352		1 1 0 1 0	5
0 1 1 0	0.0352		1 0 1 1 0	5
1 1 1 0	0.0117		1 1 1 1 0	5
0 0 0 1	0.1055	*	0 1 1	3
1 0 0 1	0.0352		1 1 0 0 1	5
0 1 0 1	0.0352		1 0 1 0 1	5
1 1 0 1	0.0117		1 1 1 0 1	5
0 0 1 1	0.0352		1 0 0 1 1	5
1 0 1 1	0.0117		1 1 0 1 1	5
0 1 1 1	0.0117		1 0 1 1 1	5
1 1 1 1	0.0039		1 1 1 1 1	5

- If $\mathbf{x} \in \overline{\mathcal{T}}_{\epsilon}^{(n)}$, no compression is performed: the codeword $C(\mathbf{x})$ is 1 followed by $\mathbf{x}$ itself.

  The codeword length is

$$\ell(x) = \lceil \log |\mathcal{X}^n| \rceil + 1 \leq n \log |\mathcal{X}| + 2. \tag{5.41}$$

This Vector Compression Scheme is a prefix code.

Example 5.6. An example of the code, for a binary source with $n = 4$, $p_{\mathsf{X}}(1) = \frac{1}{4}$ and $\epsilon = 0.01$ is given in Table 5.2. This code has expected length:

$$3 \cdot \underbrace{4 \cdot 0.1055}_{=0.4220} + 5 \cdot (1 - 4 \cdot 0.1055) = 4.1562 \tag{5.42}$$

so the code rate is:

$$R = 1.039, \tag{5.43}$$

which greater than the source rate of 1. Because n is small, the scheme is worse than uncompressed. But the scheme becomes optimal as n increases, which is illustrated by the following.

5.4.2 Proof of Vector Source Coding Theorem

This subsection proves Proposition 5.1 Vector Source Coding Theorem by using properties of typical sequences to upper bound the code rate R, where

$$R = \frac{1}{n} \sum_{\mathbf{x} \in \mathcal{X}^n} p_{\mathbf{X}}(\mathbf{x}) \ell(\mathbf{x}). \tag{5.44}$$

as given in Definition 5.1.

For $\mathbf{x} \in \mathcal{T}_\epsilon^{(n)}$, the length $\ell(\mathbf{x})$ of the codeword is upper bounded as:

$$\begin{aligned} \ell(\mathbf{x}) &\leq \log |\mathcal{T}_\epsilon^{(n)}| + 2 && \text{eqn. (5.40)} && (5.45)\\ &\leq \log 2^{n(H(\mathsf{X})+\epsilon)} + 2 && \text{Proposition 5.4} && (5.46)\\ &= n(H(\mathsf{X}) + \epsilon) + 2 && && (5.47) \end{aligned}$$

For $\mathbf{x} \in \overline{\mathcal{T}}_\epsilon^{(n)}$, recall from eqn.(5.41) that the length $\ell(\mathbf{x})$ is upper bounded by as:

$$\ell(x) \leq n \log |\mathcal{X}| + 2 \tag{5.48}$$

Now, the rate R is bounded as:

$$\begin{aligned} nR &= \sum_{\mathbf{x} \in \mathcal{X}^n} p_{\mathbf{X}}(\mathbf{x}) \ell(\mathbf{x}) && && (5.49)\\ &= \sum_{\mathbf{x} \in \mathcal{T}_\epsilon^{(n)}} p_{\mathbf{X}}(\mathbf{x}) \ell(\mathbf{x}) + \sum_{\mathbf{x} \in \overline{\mathcal{T}}_\epsilon^{(n)}} p_{\mathbf{X}}(\mathbf{x}) \ell(\mathbf{x}) && && (5.50)\\ &\leq (n(H(\mathsf{X}) + \epsilon) + 2) \sum_{\mathbf{x} \in \mathcal{T}_\epsilon^{(n)}} p_{\mathbf{X}}(\mathbf{x}) + (n \log |\mathcal{X}| + 2) \sum_{\mathbf{x} \in \overline{\mathcal{T}}_\epsilon^{(n)}} p_{\mathbf{X}}(\mathbf{x}) && && (5.51)\\ &= (n(H(\mathsf{X}) + \epsilon) + 2) \underbrace{\Pr[\mathbf{X} \in \mathcal{T}_\epsilon^{(n)}]}_{\leq 1} + (n \log |\mathcal{X}| + 2) \underbrace{\Pr[\mathbf{X} \in \overline{\mathcal{T}}_\epsilon^{(n)}]}_{\leq \epsilon \text{ by Prop 5.3}} && && (5.52)\\ &\leq (n(H(\mathsf{X}) + \epsilon) + 2) \cdot 1 + (n \log |\mathcal{X}| + 2)\epsilon && && (5.53)\\ R &\leq (H(\mathsf{X}) + \epsilon) + \frac{2}{n} + (\log |\mathcal{X}| + \frac{2}{n})\epsilon && && (5.54)\\ &= H(\mathsf{X}) + \epsilon', && && (5.55) \end{aligned}$$

where

$$\epsilon' = \epsilon + \epsilon \log |\mathcal{X}| + \epsilon \frac{2}{n} + \frac{2}{n} \tag{5.56}$$

can be made arbitrarily small by a choice of ϵ followed by a choice of n.

5.4.3 "Super-Alphabet" Perspective

The vector compression scheme is not a practical technique because indexing typical sets is difficult for large n. It was introduced to demonstrate the importance of typical sets and the asymptotic equipartition property. These concepts

are fundamental to information theory, and are easiest studied in this single variable case. In Chapter 8, the channel coding theorem requires two variables. There, fewer details will be given because we have already described the basic ideas in the one variable case of this chapter.

As with many important mathematical results, there is more than one way to prove Proposition 5.1. An optimal coding scheme (such as Huffman coding) can be applied to a super-alphabet of the vector source. For a single source Z over $\mathcal{Z}$, the expected length L of an optimal single-source code satisfies $H(\mathsf{Z}) \leq L \leq H(\mathsf{Z}) + 1$.

Suppose we have an independent and identically distributed vector source $\mathsf{X}_1, \mathsf{X}_2, \ldots, \mathsf{X}_n$ over $\mathcal{X}^n$, and want to form a vector source code with rate R. Think of this vector source as a single-variable random variable $\mathsf{Z} = \mathsf{X}_1, \ldots, \mathsf{X}_n$ which takes values from a super-alphabet $\mathcal{Z} = \mathcal{X}^n$. The relationship between the expected length L and the code rate R is:

$$R = \frac{1}{n} L. \tag{5.57}$$

By Proposition 4.2, the entropy bound on single-variable compression can be written as:

$$H(\mathsf{Z}) \leq L \quad \leq H(\mathsf{Z}) + 1 \tag{5.58}$$

$$H(\mathsf{X}_1, \ldots, \mathsf{X}_n) \leq nR \ \leq H(\mathsf{X}_1, \ldots, \mathsf{X}_n) + 1 \tag{5.59}$$

$$\frac{1}{n} H(\mathsf{X}_1, \ldots, \mathsf{X}_n) \leq R \quad \leq \frac{1}{n} H(\mathsf{X}_1, \ldots, \mathsf{X}_n) + \frac{1}{n} \tag{5.60}$$

Since X_i are independent and identically distributed, we have $H(\mathsf{X}_1, \ldots, \mathsf{X}_n) = nH(\mathsf{X})$, so:

$$H(\mathsf{X}) \leq R \leq H(\mathsf{X}) + \frac{1}{n}. \tag{5.61}$$

In other words, there exists a vector source code for $\mathsf{X}_1, \ldots, \mathsf{X}_n$ for which the rate R is no worse than $H(\mathsf{X}) + \frac{1}{n}$. This can be made as close as desired to $H(\mathsf{X})$ by using long blocks with large n.

5.5 Exercises

5.1 Let $\mathbf{X} = \mathsf{X}_1, \mathsf{X}_2, \ldots, \mathsf{X}_n$ be iid with probability distribution $p_{\mathbf{X}}(x_1, \ldots, x_n)$. Find:

$$\lim_{n \to \infty} \left(p_{\mathbf{X}}(x_1, \ldots, x_n) \right)^{1/n}$$

5.2 *Huffman codes for a single source versus vector source.*

(a) Consider a source code for a single random variable X which takes values from $\{1, 2, 3\}$:

$$p_{\mathsf{X}}(x) = \begin{cases} \frac{1}{3} & x = 1 \\ \frac{1}{3} & x = 2 \\ \frac{1}{3} & x = 3 \end{cases} \tag{5.64}$$

Find a binary Huffman code for this source X. What is the expected length?

(b) Now consider an $n = 3$ vector random variable, $\mathbf{X} = \mathsf{X}_1\mathsf{X}_2\mathsf{X}_3$, which takes on values from $\{111, 112, \ldots, 333\}$. Assuming the X_i are independent, find the joint distribution:

$$p_{\mathbf{X}}(x_1, x_2, x_3) = p_{\mathsf{X}}(x_1)p_{\mathsf{X}}(x_2)p_{\mathsf{X}}(x_3). \tag{5.65}$$

(c) Find a binary Huffman code for this vector source $\mathbf{X}$. What is the expected length? What is the expected length per source symbol (that is, the expected length divided by n).

(d) Which is better, single-symbol compression or vector compression? If we allow n to become large, what is the best possible compression in bits/symbol for this source?

5.3 Consider a sequence of iid binary random variables, $\mathbf{X} = \mathsf{X}_1, \mathsf{X}_2, \ldots, \mathsf{X}_n$, where $p_{\mathsf{X}}(0) = 0.9$ and $p_{\mathsf{X}}(1) = 0.1$.

(a) With $n = 25$ and $\epsilon = 0.2$, which sequences fall into the typical set $\mathcal{T}_{0.2}^{(25)}$? What is the probability of this typical set? What is the size of this typical set? (Writing a program is strongly recommended. To find $\binom{n}{k}$, the Matlab function is `nchoosek(n,k)` The Python function `>>> from scipy.misc import comb >>> comb(n,k,exact=True)` seems to work.)

(b) Assume that the Vector Compression Scheme in Subsection 5.4.1 is used to compress this source, where sequence $\mathbf{x}$ is compressed to codeword $C(\mathbf{x})$ with length $\ell(\mathbf{x})$. Again use $n = 25$ and $\epsilon = 0.2$. If $\mathbf{x} \in \mathcal{T}_{0.2}^{(25)}$, what is the length $\ell(\mathbf{x})$? If $\mathbf{x} \notin \mathcal{T}_{0.2}^{(25)}$, what is the length $\ell(\mathbf{x})$? Find the code rate R.

5.4 Consider a sequence of iid binary random variables, $\mathsf{X}_1, \mathsf{X}_2, \ldots, \mathsf{X}_n$, where $p_{\mathsf{X}}(0) = 0.4$ and $p_{\mathsf{X}}(1) = 0.6$.

(a) Calculate $H(\mathsf{X})$

(b) With $n = 25$ and $\epsilon = 0.1$, which sequences are in the typical set $\mathcal{T}_{0.1}^{(25)}$?

(c) What is the probability of the typical set? What is the size of the typical set?

(d) With $n = 25$, what is the smallest value of ϵ such that $\Pr(\mathbf{X} \in \mathcal{T}_{\epsilon}^{(n)}) \geq 0.99$?

Chapter 6

Source Coding for Markov Sources

From Chapter 5, we know that a memoryless source X which generates a sequence such as:

$$0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 1\ 0\ 1\ 0\ 0\ 0\ 0\ 1\ 0\ 1\ 1\ 0\ 1\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0$$

can be compressed at a rate not less than $H(\mathsf{X})$, the entropy of the source. The intuitive idea of this chapter is that a source with memory which generates a sequence such as:

$$0\ 0\ 0\ 0\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 1\ 1\ 1$$

can be compressed at a lower rate than if the memory had been ignored.

Source with memory include stationary sources and Markov sources. In a Markov process, the probability of the future X_{t+1} depends on the present X_t, but does not depend on the past X_{t-1}. Here, the index t can be thought of as time.

6.1 Markov Chains

Markov chains were introduced in Section 3.3. This section generalizes Markov chains, which are a special case of a stochastic process.

6.1.1 Stochastic Processes

A *stochastic process* $\mathsf{X}_1, \mathsf{X}_2, \ldots, \mathsf{X}_n$ is an indexed sequence of random variables (stochastic means random). The random variable X_t is called the *state* at time

Additional reading: Cover and Thomas, Sections 4.1, 4.2 and 5.4.

t. We will refer to t as time, but it is more general, it might refer to distance or some other quantity. The stochastic process has a joint probability distribution:

$$\Pr[\mathsf{X}_1 = x_1, \mathsf{X}_2 = x_2, \cdots, \mathsf{X}_n = x_n] = p_{\mathbf{X}}(x_1, x_2, \ldots, x_n) \tag{6.1}$$

Let X_t take values from the set of size m, $\mathcal{X} = \{1, 2, \ldots, m\}$. Each $x \in \mathcal{X}$ is called a *state* and $\mathcal{X}$ is called the *state space*. Let $s_t(x) = \Pr[\mathsf{X}_t = x]$ be the probability that the state is x at time $t = 1, 2, 3, \ldots$. The *state vector* is $\mathbf{s}$:

$$\mathbf{s}_t = [s_t(1), s_t(2), \ldots, s_t(m)]. \tag{6.2}$$

The state vector at time $t = 1$ is the *initial state vector*:

$$\mathbf{s}_1 = [s_1(1), s_1(2), \ldots, s_1(m)]. \tag{6.3}$$

It is much easier to study stochastic processes that are stationary. For a stationary distribution, the distribution of r adjacent random variables are the same anywhere in the time sequence.

Definition 6.1. A stochastic process is *stationary* if the distribution is shift-invariant:

$$\Pr[\mathsf{X}_1 = x_1, \mathsf{X}_2 = x_2, \cdots, \mathsf{X}_n = x_r] = \Pr[\mathsf{X}_{1+\ell} = x_1, \mathsf{X}_{2+\ell} = x_2, \cdots, \mathsf{X}_{n+\ell} = x_r]$$

for every r, every ℓ and all $x_1, \ldots, x_n$.

6.1.2 Markov Chain

A Markov chain is a kind of stochastic process.

Definition 6.2. A discrete, stationary, stochastic process $\mathsf{X}_1, \mathsf{X}_2, \ldots$ is a first-order *Markov chain* or *Markov process* if:

$$\Pr[\mathsf{X}_{t+1} = x_{t+1} | \mathsf{X}_t = x_t, \mathsf{X}_{t-1} = x_{t-1}, \cdots, \mathsf{X}_1 = x_1] = \Pr[\mathsf{X}_{t+1} = x_{t+1} | \mathsf{X}_t = x_t]$$

for all t. The idea of Markovity is expressed by "the future depends on the present and not the past." A Markov chain can be written as $\mathsf{X}_1 \to \mathsf{X}_2 \to \mathsf{X}_3 \to \cdots$.

The joint distribution $p_{\mathbf{X}}(x_1, \cdots, x_n)$ of a Markov chain is:

$$p_{\mathbf{X}}(x_1, x_2, \cdots, x_n) = p_{\mathsf{X}_1}(x_1) p_{\mathsf{X}_2|\mathsf{X}_1}(x_2|x_1) \cdots p_{\mathsf{X}n|\mathsf{X}n-1}(x_n|x_{n-1}) \tag{6.4}$$

Definition 6.3. A *time-invariant Markov chain*[1] satisfies:

$$\Pr[\mathsf{X}_{t+1} = j | \mathsf{X}_t = i] = \Pr[\mathsf{X}_2 = j | \mathsf{X}_1 = i] \tag{6.5}$$

for all i, j, and t. That is, $\Pr[\mathsf{X}_{t+1} = j | \mathsf{X}_t = i]$ does not depend upon t.

[1] Also called a homogenous Markov process.

For Markov chains, we say "X_n is the state at time t" If $\mathsf{X}_t = a$ then we say "The state value at time t is a."

A time-invariant Markov chain is characterized by *transition probabilities* $p_{i,j}$:

$$p_{i,j} = \Pr[\mathsf{X}_{t+1} = j | \mathsf{X}_t = i], \tag{6.6}$$

which does not depend on t due to the time-invariant property. The transition probabilities can be written using a m-by-m *state-transition matrix* $\mathbf{P}$:

$$\mathbf{P} = \begin{bmatrix} p_{1,1} & p_{1,2} & \cdots & p_{1,m} \\ p_{2,1} & p_{2,2} & \cdots & p_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ p_{m,1} & p_{m,2} & \cdots & p_{m,m} \end{bmatrix} \tag{6.7}$$

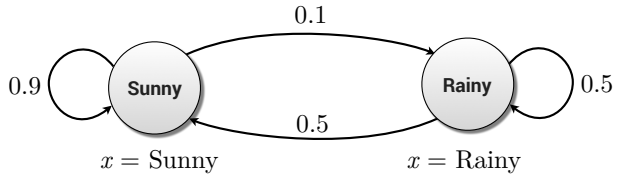
Note that $\mathbf{P}$ is a *right stochastic matrix* meaning each row is a probability distribution (that is, each row sums to one and the elements are nonnegative).

A time-invariant Markov chain can be represented by a state diagram with m nodes representing the m states. The state diagram has an edge from i to j labeled with the probability $p_{i,j}$.

Example 6.1. A concrete example of a Markov source is a simple model of the weather with $\mathcal{X} = \{\text{Sunny, Rainy}\}$. Given today is Sunny, the probability that tomorrow is Rainy is $\Pr[\mathsf{X}_{n+1} = \text{Rainy} | \mathsf{X}_n = \text{Sunny}] = 0.1$. All the possible transitions are given by a transition matrix:

$$\mathbf{P} = \begin{bmatrix} 0.9 & 0.1 \\ 0.5 & 0.5 \end{bmatrix}. \tag{6.8}$$

The state-transition diagram is shown below.



Given state vector $\mathbf{s}_t$ at time t, what is the state vector $\mathbf{s}_{t+1}$ at time $t+1$?. Consider the probability that the Markov chain is in state x at time $t+1$. Using the theorem of total probability we can write:

$$\Pr[\mathsf{X}_{t+1} = x] = \sum_{x' \in \mathcal{X}} \Pr[\mathsf{X}_{t+1} = x | \mathsf{X}_t = x'] \Pr[\mathsf{X}_t = x'] \tag{6.9}$$

The state vector $\mathbf{s}_{t+1}$ at time $t+1$ can be conveniently expressed in matrix form:

$$\mathbf{s}_{t+1} = \mathbf{s}_t \cdot \mathbf{P}. \tag{6.10}$$

The matrix $\mathbf{P}$ is called right stochastic because it appears on the right side of $\mathbf{s}_t$.

This generalizes as follows. If you know the state vector at time t is $\mathbf{s}_t$, then the state vector at time $t+k$ is:

$$\mathbf{s}_{t+k} = \mathbf{s}_t \cdot \mathbf{P}^k \tag{6.11}$$

That is, the k-step transition matrix is $\mathbf{P}^k$.

Example 6.2. Continuing Example 6.1, find $\mathbf{s}_2, \mathbf{s}_3$ and $\mathbf{s}_4$ when the initial state vector is:

$$\mathbf{s}_1 = \begin{bmatrix} 0 & 1 \end{bmatrix}.$$

Then, $\mathbf{s}_2$ can be found as:

$$\begin{aligned} \mathbf{s}_2 &= \mathbf{s}_1\mathbf{P} \\ &= \begin{bmatrix} 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0.9 & 0.1 \\ 0.5 & 0.5 \end{bmatrix} \\ &= \begin{bmatrix} 0.5 & 0.5 \end{bmatrix} \end{aligned}$$

Similarly, $\mathbf{s}_3$ and $\mathbf{s}_4$ are:

$$\begin{aligned} \mathbf{s}_3 &= \mathbf{s}_2 \cdot \mathbf{P} \\ &= \mathbf{s}_1 \cdot \mathbf{P}^2 \\ &= \begin{bmatrix} 0.7 & 0.3 \end{bmatrix} \\ \mathbf{s}_4 &= \mathbf{s}_1\mathbf{P}^3 \\ &= \begin{bmatrix} 0.78 & 0.22 \end{bmatrix} \end{aligned}$$

The blue curve in Fig. 6.1 shows the state vectors $\mathbf{s}_1, \mathbf{s}_2, \mathbf{s}_3, \ldots, \mathbf{s}_{10}$.

Fig. 6.1 also shows the evolution of the state vectors when the initial state is $[0.5, 0.5]$ in green, and $[1, 1]$ in red.

SSQ 6.1. Consider the three-state Markov chain with state transition matrix $\mathbf{P}$:

$$\mathbf{P} = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 1/2 \\ 0 & 0 & 1 \end{bmatrix}. \tag{6.12}$$

and initial state vectors$\mathbf{s}_1 = [1, 0, 0]$. (a) What is the state vector at time $t = 2$? (b) What is the state vector at time $t = 4$?

6.1.3 Steady-State Distribution

What happens to the state vector $\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_t, \ldots$ if we let the Markov chain go through arbitrarily many steps, $t \to \infty$? A *steady-state distribution* (stationary

distribution[2]) $\mathbf{z}$ is a state vector such that:

$$\mathbf{z} = \mathbf{z}\mathbf{P}. \tag{6.13}$$

That is, the state vector of the steady-state distribution does not change after taking one or more steps in the Markov chain. The steady-state distribution can be seen in Fig. 6.1 — as time progresses, all initial state distributions evolve toward the same steady-state distribution. Importantly, the steady-state distribution is independent of the initial state distribution.

It is possible to find the steady-state distribution $\mathbf{z}$ by observing that the steady-state distribution

$$\mathbf{z} = \mathbf{z}\mathbf{P}$$

can be written as:

$$\mathbf{z}\underbrace{(\mathbf{P} - \mathbf{I}_m)}_{=\mathbf{Q}} = \mathbf{0} \tag{6.14}$$

where $\mathbf{I}_m$ is the $m \times m$ identity matrix. To avoid the trivial solution $\mathbf{z} = \mathbf{0}$, it is necessary to add the restriction $z_1 + z_2 + \cdots z_m = 1$. To do this replace the first column[3] of $\mathbf{Q}$ with the all-ones column, call this matrix $\widetilde{\mathbf{Q}}$. Then, (6.14) is re-written as:

$$\mathbf{z}\widetilde{\mathbf{Q}} = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \end{bmatrix}. \tag{6.15}$$

Then, the steady-state distribution is found by inverting $\widetilde{\mathbf{Q}}$:

$$\mathbf{z} = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \end{bmatrix} \cdot \widetilde{\mathbf{Q}}^{-1} \tag{6.16}$$

Example 6.3. Find the steady-state distribution for Example 6.1, where the state transition matrix is:

$$\mathbf{P} = \begin{bmatrix} 0.9 & 0.1 \\ 0.5 & 0.5 \end{bmatrix}.$$

The matrix $\mathbf{Q}$ and $\widetilde{\mathbf{Q}}$ are:

$$\mathbf{Q} = \begin{bmatrix} -0.1 & 0.1 \\ 0.5 & -0.5 \end{bmatrix} \text{ and } \widetilde{\mathbf{Q}} = \begin{bmatrix} 1 & 0.1 \\ 1 & 0.5 \end{bmatrix}$$

the steady-state distribution is given by:

$$\mathbf{z} = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0.1 \\ 1 & 0.5 \end{bmatrix}^{-1} = \begin{bmatrix} \frac{5}{6} & \frac{1}{6} \end{bmatrix}$$

Fig. 6.1 illustrates how the probability distribution evolves towards the steady-state value.

---

[2]"Steady-state distribution" is preferred over "stationary distribution" to avoid confusion with stationary processes.

[3]The choice of first column is arbitrary — any column can be selected.

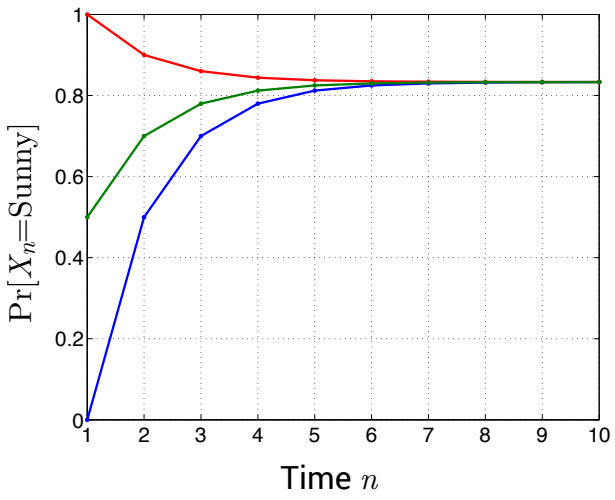


Figure 6.1: Evolution of state vector when the initial state distribution is $[0, 1]$ (blue), or $[0.5, 0.5]$ (green) or $[1, 0]$ (red). All three evolve towards the same steady-state distribution $[5/6, 1/6]$.

SSQ 6.2. Find the steady-state distribution for the two-state Markov chain with probability transition matrix:

$$\mathbf{P} = \begin{bmatrix} \frac{2}{3} & \frac{1}{3} \\ \frac{1}{4} & \frac{3}{4} \end{bmatrix} \tag{6.17}$$

Example 6.4. Find the steady-state distribution of the Markov chain given by:

$$\mathbf{P} = \begin{bmatrix} 0.8 & 0.1 & 0.1 \\ 0.5 & 0.5 & 0 \\ 0.7 & 0.2 & 0.1 \end{bmatrix}. \tag{6.18}$$

The matrix $\mathbf{Q}$ and $\widetilde{\mathbf{Q}}$ are:

$$\mathbf{Q} = \begin{bmatrix} 0.8-1 & 0.1 & 0.1 \\ 0.5 & 0.5-1 & 0 \\ 0.7 & 0.2 & 0.1-1 \end{bmatrix} = \begin{bmatrix} -0.2 & 0.1 & 0.1 \\ 0.5 & -0.5 & 0 \\ 0.7 & 0.2 & -0.9 \end{bmatrix} \tag{6.19}$$

$$\widetilde{\mathbf{Q}} = \begin{bmatrix} 1 & 0.1 & 0.1 \\ 1 & -0.5 & 0 \\ 1 & 0.2 & -0.9 \end{bmatrix} \tag{6.20}$$

The matrix inverse is:

$$\widetilde{\mathbf{Q}}^{-1} = \begin{bmatrix} 0.7377 & 0.18033 & 0.081967 \\ 1.4754 & -1.6393 & 0.16393 \\ 1.1475 & -0.16393 & -0.98361 \end{bmatrix} \tag{6.21}$$

Referring to (6.16), the steady-state distribution is the top row:

$$\mathbf{z} = \begin{bmatrix} 0.7377 & 0.18033 & 0.081967 \end{bmatrix}.$$

Note that $\mathbf{z}$ is a probability vector (non-negative and sums to one), and that it satisfies $\mathbf{z} = \mathbf{zP}$.

6.2 Entropy Rate

This section first defines the entropy rate of any stochastic process. Next, we restrict to stationary for processes and show that the conditional entropy rate is equal to the entropy rate. Then, it is shown that a stationary source can be compressed at the entropy rate. Finally, we restrict to stationary Markov sources and give the entropy rate, and thus the compression limit, of a stationary Markov source.

6.2.1 For Stochastic Processes

Consider a vector source $\mathsf{X}_1, \mathsf{X}_2, \ldots, \mathsf{X}_n$. From Chapter 5, if the X_i are independent and identically distributed, each with entropy $H(\mathsf{X})$, then there is a code C with rate R which is close to $H(\mathsf{X})$.

What if the sequence is not independent and identically distributed? We expect that there is a code with rate close to:

$$\frac{1}{n} H(\mathsf{X}_1, \mathsf{X}_2, \ldots, \mathsf{X}_n). \tag{6.22}$$

For a stochastic process, n is not fixed, but could be arbitrarily large. This leads to the idea of entropy rate.

Definition 6.4. The *entropy rate* of a stochastic process $\mathsf{X}_1, \mathsf{X}_2, \ldots$ is defined by:

$$H(\mathcal{X}) = \lim_{n \to \infty} \frac{1}{n} H(\mathsf{X}_1, \mathsf{X}_2, \ldots, \mathsf{X}_n) \tag{6.23}$$

when the limit exists.

Examples:

1. *Keyboard with m symbols* Someone randomly hits an m-symbol keyboard. Each symbol is equally likely. There are m^n sequences of length n, all equally likely. Then:

$$H(\mathsf{X}_1, \mathsf{X}_2, \ldots, \mathsf{X}_n) = \log m^n \tag{6.24}$$

$$H(\mathcal{X}) = \log m \text{ bits per symbol} \tag{6.25}$$

2. $\mathsf{X}_1, \mathsf{X}_2, \ldots$ are i.i.d. random variables:

$$H(\mathcal{X}) \quad = \quad \lim_{n\to\infty} \frac{1}{n} H(\mathsf{X}_1, \ldots, \mathsf{X}_n) = \lim \frac{1}{n} n H(\mathsf{X}_1) = H(\mathsf{X}_1) \quad (6.26)$$

3. Sequence of independent but not identically distributed X_i:

$$H(\mathsf{X}_1, \mathsf{X}_2, \ldots, \mathsf{X}_n) = \sum_{i=1}^{n} H(\mathsf{X}_i) \quad (6.27)$$

Definition 6.5. The *conditional entropy rate* is:

$$H'(\mathcal{X}) = \lim_{n\to\infty} H(\mathsf{X}_n | \mathsf{X}_{n-1}, \ldots, \mathsf{X}_2, \mathsf{X}_1) \quad (6.28)$$

Compare $H(\mathcal{X})$ and $H'(\mathcal{X})$. $H(\mathcal{X})$ is the entropy per symbol of n random variables. $H'(\mathcal{X})$ is the conditional entropy of the last random variable, given the past.

6.2.2 For Stationary Stochastic Processes

Proposition 6.1. For a stationary stochastic process, $H(\mathsf{X}_n | \mathsf{X}_{n-1}, \ldots, \mathsf{X}_1)$ is non-increasing in n and has a limit $H'(\mathcal{X})$.

Proof Conditioning reduces entropy:

$$H(\mathsf{X}_{n+1} | \mathsf{X}_n, \ldots, \mathsf{X}_1) \quad \leq \quad H(\mathsf{X}_{n+1} | \mathsf{X}_n, \ldots, \mathsf{X}_2) \quad (6.29)$$

By stationarity:

$$H(\mathsf{X}_{n+1} | \mathsf{X}_n, \ldots, \mathsf{X}_2) \quad = \quad H(\mathsf{X}_n | \mathsf{X}_{n-1}, \ldots, \mathsf{X}_1) \quad (6.30)$$

Then, $H(\mathsf{X}_n | \mathsf{X}_{n-1}, \ldots, \mathsf{X}_1)$ is a decreasing sequence of non-negative numbers. Thus, this sequence has a limit, which is $H'(\mathcal{X})$.

Proposition 6.2. For a stationary stochastic process:

$$H(\mathcal{X}) \quad = \quad H'(\mathcal{X}). \quad (6.31)$$

Proof By chain rule:

$$\begin{aligned}
H(\mathcal{X}) &= \lim_{n\to\infty} \frac{1}{n} H(\mathsf{X}_1, \ldots, \mathsf{X}_n) && \text{definition of entropy rate} \\
&= \lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} H(\mathsf{X}_i | \mathsf{X}_{i-1}, \ldots, \mathsf{X}_1) && \text{entropy chain rule} \\
&= \lim_{n\to\infty} H(\mathsf{X}_n | \mathsf{X}_{n-1}, \ldots, \mathsf{X}_1) && \text{long term average dominates} \\
&= H'(\mathcal{X}) && \text{Proposition 6.1} \quad \square
\end{aligned}$$

Now consider a *source code* for a stationary stochastic process $\mathsf{X}_1, \mathsf{X}_2, \ldots, \mathsf{X}_n$ which takes values from $\mathcal{X}^n$. We want to form a vector source code with code rate R, as given in Definition 5.2. Let R_n^* be the rate of an optimal code, the code having the lowest possible rate. Then we have the following:

Proposition 6.3. If $\mathsf{X}_1, \mathsf{X}_2, ...\mathsf{X}_n$ is a stationary stochastic process,

$$\lim_{n\to\infty} R_n^* = H(\mathcal{X}) \tag{6.32}$$

where $H(\mathcal{X})$ is the entropy rate of the process.

Proof Use the same super-alphabet approach of Subsection 5.4.3, that is, define a new variable Z with alphabet $\mathcal{Z}$:

$$\mathsf{Z} = (\mathsf{X}_1, \mathsf{X}_2, \ldots \mathsf{X}_n) \tag{6.33}$$
$$\mathcal{Z} = \mathcal{X}^n \tag{6.34}$$

The source is a single-variable random variable Z which takes values from $\mathcal{X}^n$, where optimal compression satisfies $H(\mathsf{Z}) \leq L_n \leq H(\mathsf{Z}) + 1$. The relationship between the expected length L_n for the source for Z, and the code rate R_n is

$$R_n = \frac{1}{n} L_n. \tag{6.35}$$

Then,

$$H(\mathsf{Z}) \leq L \quad \leq H(\mathsf{Z}) + 1 \tag{6.36}$$
$$H(\mathsf{X}_1, \mathsf{X}_2, \ldots, \mathsf{X}_n) \leq nR_n \leq H(\mathsf{X}_1, \mathsf{X}_2, \ldots, \mathsf{X}_n) + 1 \tag{6.37}$$
$$\underbrace{\frac{1}{n} H(\mathsf{X}_1, \mathsf{X}_2, \ldots, \mathsf{X}_n)}_{\to H(\mathcal{X})} \leq R_n \leq \underbrace{\frac{1}{n} H(\mathsf{X}_1, \mathsf{X}_2, \ldots, \mathsf{X}_n)}_{\to H(\mathcal{X})} + \underbrace{\frac{1}{n}}_{\to 0} \tag{6.38}$$

Take the limit as $n \to \infty$ in (6.38). By the definition of entropy rate (6.23), R_n is bounded between $H(\mathcal{X})$ and $H(\mathcal{X}) + \frac{1}{n}$ and as $n \to \infty$ the proposition holds. □

In other words, there exists a vector source code for $\mathsf{X}_1, \ldots, \mathsf{X}_n$ for which the rate R approaches $H(\mathcal{X})$ as $n \to \infty$.

6.2.3 For Stationary Markov Chains

Proposition 6.3 shows that any stationary source can be compressed at the entropy rate of the process. In this section, we find the entropy rate of a stationary Markov process. Note that for a stationary Markov chain, the initial state distribution is equal to the steady-state distribution, and thus $H(\mathsf{X}_{n+1}|\mathsf{X}_n) = H(\mathsf{X}_2|\mathsf{X}_1)$.

Proposition 6.4. For a stationary Markov chain, the entropy rate is given by:

$$H(\mathcal{X}) = H(\mathsf{X}_2|\mathsf{X}_1) \tag{6.39}$$

Note that the stationary distribution $\mathbf{z} = z_1, \ldots, z_m$ satisfies:

$$z_j \quad = \quad \sum_{i=1}^{m} z_i p_{i,j}. \tag{6.40}$$

Given a stationary Markov chain with state transitions $\mathbf{P}$ and steady-state distribution $\mathbf{z}$, we can find $H(\mathcal{X})$

Proposition 6.5. Let $\mathsf{X}_1, \mathsf{X}_2, \ldots,$ be a stationary Markov chain with steady-state distribution $\mathbf{z}$ and transition matrix $\mathbf{P}$. Let $\mathsf{X}_1 \sim \mathbf{z}$. Then the entropy rate is:

$$H(\mathcal{X}) = -\sum_{i=1}^{m}\sum_{j=1}^{m} z_i p_{i,j} \log p_{i,j} \tag{6.41}$$

The Markov chain is stationary, and the right-hand side of (6.41) is the computation of $H(\mathsf{X}_2|\mathsf{X}_1)$.

Example 6.5. Find the optimal coding rate for the Markov chain from Example 6.3. Compare this with the optimal coding rate if the Markov structure is ignored.

The state transition matrix and steady-state distribution for this Markov chain are:

$$\mathbf{P} = \begin{bmatrix} 0.9 & 0.1 \\ 0.5 & 0.5 \end{bmatrix} \text{ and } \mathbf{z} = \begin{bmatrix} \frac{5}{6} & \frac{1}{6} \end{bmatrix}$$

Applying (6.41) to these,

$$\begin{aligned} H(\mathcal{X}) &= -\frac{5}{6}\big(0.9\log 0.9 + 0.1\log 0.1\big) - \frac{1}{6}\big(0.5\log 0.5 + 0.5\log 0.5\big) \\ &= 0.5575 \text{ bits/source symbol} \end{aligned}$$

Thus, the optimal coding rate is $R^* = 0.5575$.

If the Markovity is ignored, then we have an independent and identically distributed random vector source distributed as $\begin{bmatrix} \frac{5}{6} & \frac{1}{6} \end{bmatrix}$. Using the results of Chapter 5, this can be compressed at rate $R = H(\mathsf{X}_1) = h(\frac{5}{6}) = 0.65$ bits/source symbol. By comparing $H(\mathcal{X})$ and $H(\mathsf{X}_1)$, it is clear that lower compression rates are obtained by considering the Markovity of the source.

6.3 Exercises

6.1 Consider a two-state Markov chain $\mathsf{X}_1, \mathsf{X}_2, \mathsf{X}_3, \ldots$ with probability transition matrix:

$$\mathbf{P} = \begin{bmatrix} \frac{6}{7} & \frac{1}{7} \\ \frac{2}{7} & \frac{5}{7} \end{bmatrix}$$

(a) What is the stationary distribution?

(b) What is the entropy rate $H(\mathcal{X})$?

(c) What is the single-variable entropy, $H(\mathsf{X}_1)$?

(d) Which has lower compression rate, compression using the Markov property, or single-variable compression?

6.2 Give a proof of Proposition 6.4.

6.3 A Markov chain has two states $\{0, 1\}$. The probability of going from state 0 to 1 is p. The probability of going from state 1 to 0 is $(1-p)/2$, for $0 \leq p \leq 1$. Find the entropy rate of this Markov chain.

6.4 DNA, which encodes genetic instructions in all living things, uses a code over a quaternary (4-ary) alphabet. This code is abbreviated $\mathcal{X} = \{$ A, C, G, T $\}$. DNA consists of long strings from this alphabet: $\mathsf{X}_1, \mathsf{X}_2, \mathsf{X}_3, \ldots$. A possible first-order Markov model is given by (`https://bit.ly/2yxMzCf`):

	X_{n+1}			
X_n	A	C	G	T
A	0.180	0.274	0.426	0.120
C	0.171	0.367	0.274	0.188
G	0.161	0.339	0.375	0.125
T	0.079	0.355	0.384	0.182

(a) Find the steady-state distribution of this Markov chain.

(b) Find the entropy of X.

(c) Find the entropy rate $H(\mathcal{X})$ of this process.

6.5 *Conditional Huffman Coding for a Markov Chain* Write a program for binary conditional Huffman coding and decoding as follows.

- Inputs: $n, m, \mathcal{X} = \{1, 2, \ldots, m\}, \mathbf{p}_{X_1} = [p_1, \ldots, p_m]$ and $\mathbf{P} = [p(X_n = i | X_{n-1} = j)]_{m \times m}$.
- Step 1: Generate a Markov chain: $\mathbf{x} = [x_1, x_2, \ldots, x_n]$ with $X_i \in \mathcal{X}$ based on $\mathbf{p}_{X_1}$ and $\mathbf{P}$.
- Step 2: **Encode** $\mathbf{x}$ using binary conditional Huffman coding, and **output** codeword $\mathbf{c}(\mathbf{x})$ and rate R per symbol.
- Step 3: **Decode** with $\mathbf{c}(\mathbf{x})$ using conditional Huffman decoding, and **output** the decoded sequence $\hat{\mathbf{x}}$.

You should submit a program file (e.g. Matlab, Python, or C) for this exercise. Your program should work for any values of n and m, any probability vector $\mathbf{p}_{X_1} = [p_1, \ldots, p_m]$, and any probability transition matrix $\mathbf{P} = [p(X_n = i | X_{n-1} = j)]_{m \times m}$. We will test your program using some examples (e.g. the example in the slides of Lecture 6). Sample Matlab source code is given.

6.6 Under what conditions is a Markov chain a stationary process?

```
function main()
  clc; clear; close all;
  space = 1:4;
  px1 = [1/2 1/4 1/8 1/8];
  P = [1/2 1/4 1/8 1/8; 1/4 1/8 1/8 1/2; 1/8 1/8 1/2 1/4; 1/8 1/2 1/4
  1/8];
  n = 10;
  %% generate a Markov chain x with length n based on p_x1 and P
  x = Markov_chain(space, px1, P, n);


  %% encode x using conditional Huffman coding, and output codeword
  cx, rate R, entropy rate Hx, stationary distribution z...
  [cx R, Hx, z] = Conditional_Huffman_Encoding(x, space, px1, P);


  %% decode with cx using conditional Huffman decoding, and output
  decoded sequence x_dec
  x_dec = Conditional_Huffman_Decoding(cx, px1, P);


  %% check that the decoding is correct
  error = sum( abs(x_dec - x) );
end


%% Source generation function
function [x] = Markov_chain(space, px1, P, n)
    % write your code here
end


%% Encoding process function
function[cx,R,Hx,z] = Conditional_Huffman_Encoding(x, space, px1, P)
    % write your code here
end


%% Decoding process function
function [x_dec] = Conditional_Huffman_Decoding(cx, px1, P)
    % write your code here
end
```

Chapter 7

Channel Coding and Channel Capacity

Real communication channels are complex, and so information theory introduces channel models, which are understood to be mathematical representations of communications models. This chapter introduces the communications systems model that is used. Then, the definition of channel capacity is given, followed by the capacity of a few discrete memoryless channels.

7.1 Communication System Model

This subsection describes a mathematical model of a communications system. A communication system consists of (a) a code and its encoder (b) channel and (c) decoder, as shown in Fig. 7.1.

7.1.1 Code and Its Encoder

The encoder maps messages to codewords. The terms message, code, encoder and rate are defined as follows.

Definition 7.1. A *message* W is random variable representing one of M information symbols:

$$\mathcal{W} = \{1, 2, \ldots, M\}, \tag{7.1}$$

where W is uniformly distributed, that is $p_{\mathsf{W}}(w) = \frac{1}{M}$ for $w \in \mathcal{W}$.

Additional reading: Cover and Thomas, Sections 7.1 to 7.5, 7.11.

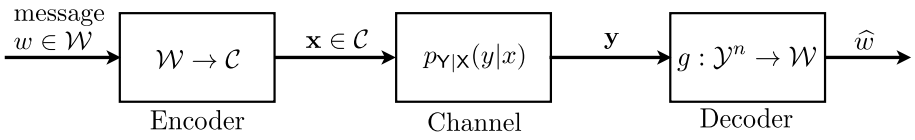


Figure 7.1: Model of a communications system.

Definition 7.2. An (M, n) *code* having a *codebook* $\mathcal{C}$ consists of M vectors:

$$\mathcal{C} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_M \end{bmatrix} \tag{7.2}$$

where each *codeword* $\mathbf{x}_i$ consists of n symbols:

$$\mathbf{x} = (x_1, x_2, \ldots, x_n), \tag{7.3}$$

with $x_i \in \mathcal{X}$.

The codebook alphabet is $\mathcal{X}$. For a binary code, $\mathcal{X} = \{0, 1\}$. However, non-binary codes are also used, for example $\mathcal{X} = \{0, 1, 2\}$ is a ternary alphabet which forms a ternary code.

Definition 7.3. The *rate* R of an (M, n) code is:

$$R = \frac{1}{n} \log M. \tag{7.4}$$

As usual, if we take log base 2, then the units of R is bits per transmission.

Equivalently, $M = 2^{nR}$, so the message set is also written $\mathcal{W} = \{1, 2, \ldots, 2^{nR}\}$.

The rate R measures how much information a code can carry for each channel use. For a code with n symbols, the channel is used n times. The code carries $\log M$ bits of information in total — in other words, we need $\log M$ bits to select one of the codewords. Then, the rate $\frac{1}{n} \log M$ is the average number of bits per channel use.

The higher the code rate, the more information can be carried. The codebook is a set that satisfies $\mathcal{C} \subseteq \mathcal{X}^n$, and the number of codewords $M = |\mathcal{C}|$ satisfies $M \leq |\mathcal{X}|^n$, so the rate is bounded by:

$$0 \leq R \leq \log |\mathcal{X}|. \tag{7.5}$$

For example, for binary codes $|\mathcal{X}| = 2$, and so $0 \leq R \leq 1$.

In general, a high rate code carries a large amount of information, but can only be used on more reliable channels. A low rate code carries only a small amount of information, but can be used on less reliable channels. Sometimes the number of bits of information is written as $k = \log M$ bits.

Definition 7.4. An *encoder* is a mapping from the M messages of $\mathcal{W}$ to the M codewords of $\mathcal{C}$:

$$\mathcal{W} \to \mathcal{C} \tag{7.6}$$

Since $|\mathcal{W}| = |\mathcal{C}|$, the encoding mapping is bijective, that is, there is a one-to-one correspondence between messages and codewords.

Example 7.1. The table below gives an encoding mapping for a ternary code with $\mathcal{X} = \{0, 1, 2\}$:

message w	codeword $\mathbf{x}$
1	2 1 0 2 1 0
2	2 0 0 0 1 1
3	2 2 0 2 2 1
4	1 0 2 0 2 1

For this code, the number of messages is $M = 4$. This means that the code carries $\log M = 2$ bits of information. The length of the code is $n = 6$ and the code rate is $R = 1/3$.

SSQ 7.1. What is (a) the number of codewords (b) the block length n and (c) the rate R for an error correcting code with the following codebook?

$$\mathcal{C} = \{1101001, 0011010, 0111100, 1100110\} \tag{7.7}$$

7.1.2 Channel

Recall the discrete memoryless channel (DMC) given in Definition 2.1 on page 39. The DMC has input alphabet $\mathcal{X}$, output alphabet $\mathcal{Y}$ and transition probability $p_{\mathsf{Y}|\mathsf{X}}(y|x)$. The channel model is n uses of the DMC, where the channel input sequence is $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ and the channel output sequence is $\mathbf{y} = (y_1, y_2, \ldots, y_n)$.

The channel is described by joint conditional distribution $p_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x})$. Because the channel is memoryless, we have:

$$\begin{aligned} p_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x}) &= p_{\mathsf{Y}|\mathsf{X}}(y_1|x_1)p_{\mathsf{Y}|\mathsf{X}}(y_2|x_2)\cdots p_{\mathsf{Y}|\mathsf{X}}(y_n|x_n) \\ &= \prod_{i=1}^{n} p_{\mathsf{Y}|\mathsf{X}}(y_i|x_i). \end{aligned}$$

Note that $\mathcal{X}$ is both the codebook alphabet and the channel input alphabet.

Binary Symmetric Channel The binary symmetric channel (BSC) is a discrete memoryless channel where an error occurs with probability α. It has

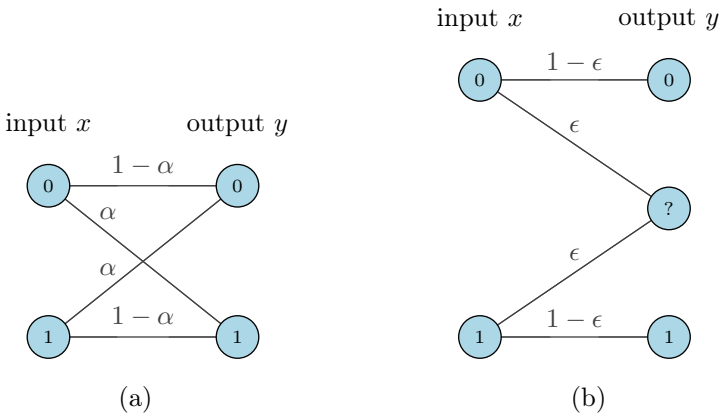


Figure 7.2: (a) Binary symmetric channel (BSC) with error probability α. (b) Binary erasure channel with erasure probability ϵ.

binary inputs $\mathcal{X} = \{0, 1\}$, binary outputs $\mathcal{Y} = \{0, 1\}$. For a parameter α, the probability transition matrix $p_{\mathsf{Y}|\mathsf{X}}(y|x)$ is:

$$p_{\mathsf{Y}|\mathsf{X}}(y|x) = \begin{bmatrix} 1-\alpha & \alpha \\ \alpha & 1-\alpha \end{bmatrix}, \tag{7.8}$$

where $0 \leq \alpha \leq 1$. For the BSC, an error occurs if an input 0 is changed to a 1, and this happens with probability α. Similarly, an input 1 changes to a 0 with probability p. There is no error with probability $1 - \alpha$.

Binary Erasure Channel The binary erasure channel (BEC) is a discrete memoryless channel also has binary inputs $\mathcal{Y} = \{0, 1\}$. The outputs are $\mathcal{Y} = \{0, ?, 1\}$ where ? is an erasure symbol. For a parameter ϵ, the probability transition matrix $p_{\mathsf{Y}|\mathsf{X}}(y|x)$ is:

$$p_{\mathsf{Y}|\mathsf{X}}(y|x) = \begin{bmatrix} 1-\epsilon & \epsilon & 0 \\ 0 & \epsilon & 1-\epsilon \end{bmatrix}, \tag{7.9}$$

where $0 \leq \epsilon \leq 1$. For the BEC, an input symbol 0 or 1 is erased with probability ϵ, and received correctly with probability $1 - \epsilon$. It is not possible for an input 0 to become an output 1 (or 1 to become a 0).

7.1.3 Decoder

Definition 7.5. A *decoding function* is a rule g which maps received sequences $\mathbf{y}$ to messages:

$$g : \mathcal{Y}^n \to \mathcal{W}, \tag{7.10}$$

where $\mathcal{W} = \{1, 2, \ldots, M\}$.

The output of the decoding function $\widehat{w} = g(\mathbf{y})$ is an estimate of the transmitted message. If $\widehat{w} = w$, then no error occurred, and there was a reliable

transmission. If $\widehat{w} \neq w$, then an error occurred. The decoder can also declare "failure" which means no codeword was selected, which is also an error. Three types of error probabilities are defined.

The *conditional probability of error* λ_w is conditioned on the event that message w was sent:

$$\lambda_w = \Pr\left(g(\mathbf{Y}) \neq w | \mathbf{X} = \mathbf{x}(w)\right)$$

The *average probability of error* is P_e:

$$P_e = \frac{1}{M} \sum_{w=1}^{M} \lambda_w \tag{7.11}$$

averaged over all codewords.

The *maximum probability of error* $\lambda^{(n)}$ is

$$\lambda^{(n)} = \max_{w \in \mathcal{W}} \lambda_i \tag{7.12}$$

7.2 Example Using Repeat Code

In order to motivate the ideas of a channel code, this section gives an example of a communication system model using a repeat code. The binary repeat code is a code with $M = 2$ codewords, $0000 \cdots 0$ and $1111 \cdots 1$. Tthis section gives an example of a communication system model using an $n = 5$ repeat code on the binary symmetric channel with error probability $p = 0.1$. The probabilities of error λ_0, λ_1 and P_e are computed.

7.2.1 Encoder for Repeat Code

The $n = 5$ binary repeat code has 2 codewords,

$$\mathcal{C} = \{00000, 11111\}, \tag{7.13}$$

so there are two messages $\mathcal{W} = \{0, 1\}$ which are equally likely, $p_{\mathsf{W}}(0) = p_{\mathsf{W}}(1) = \frac{1}{2}$. If the message is $w = 0$, then the encoder transmits $\mathbf{x}_0 = 00000$ and if the message is $w = 1$ then the encoder transmits $\mathbf{x}_1 = 11111$.

7.2.2 Binary Symmetric Channel

The binary symmetric channel (BSC) is a simple model of communication errors. It has binary inputs $\mathcal{X} = \{0, 1\}$ and binary outputs $\mathcal{Y} = \{0, 1\}$. A transmitted bit is either received correctly with probability $1 - \alpha$, or received in error with probability α. The conditional probability distribution is:

$$p_{\mathsf{Y}|\mathsf{Z}}(y|z) = \begin{bmatrix} 1-\alpha & \alpha \\ \alpha & 1-\alpha \end{bmatrix} \tag{7.14}$$

An equivalent model is to define a binary random variable Z with probability distribution

$$p_{\mathsf{Z}}(z) = \begin{cases} 1-\alpha & \text{if } z = 0 \\ \alpha & \text{if } z = 1 \end{cases} \tag{7.15}$$

here $z = 0$ indicates "no error" and $z = 1$ indicates "error". Then, the channel output can be written as:

$$\mathsf{Y} = \mathsf{X} \oplus \mathsf{Z}, \tag{7.16}$$

where $\oplus$ indicates addition modulo 2. The noise Z is independent of the input X, that is $p_{\mathsf{Z}|\mathsf{X}}(z|x) = p_{\mathsf{Z}}(z)$.

An error vector $\mathbf{z} = z_1 z_2 \dots z_n$ is generated by n uses of the channel. The probability of an error vector with m ones is:

$$\Pr(\mathbf{z} \text{ has } m \text{ ones}) = \binom{n}{m} \alpha^m (1-\alpha)^{n-m} \tag{7.17}$$

For example, when $\alpha = 0.1$ the probability of any sequence with 2 ones and 3 zeros:

$$\Pr(\mathbf{z} \text{ has 2 ones}) = \binom{5}{2} 0.9^3 0.1^2 = 0.0729. \tag{7.18}$$

7.2.3 Decoder: Majority Vote

The decoder for the repeat code uses the "majority vote" decoding rule. It maps the channel output $\mathbf{y}$ to an estimated message $\widehat{w}$ as follows:

- If the sequence $\mathbf{y}$ has $(n-1)/2$ or more zeros, then the estimated message is $\widehat{w} = 0$.
- If the sequence $\mathbf{y}$ has $(n-1)/2$ or more ones, then the estimated message $\widehat{w} = 1$.

In other words, the symbol with the most "votes" wins. A decoding error occurs if $w \neq \widehat{w}$.

For $n = 5$, if the received sequence $\mathbf{y}$ has 0,1 or 2 ones, then the majority vote rule selects the codeword 00000 and message $\widehat{w} = 0$. If the received sequence has 3, 4 or 5 ones, then the majority vote rule selects the codeword 11111 and message $\widehat{w} = 1$. The table below summarizes the decoding rule.

$\mathbf{y}$ has ...	example $\mathbf{y}$	estimated codeword $\widehat{\mathbf{x}}$	estimated message $\widehat{w}$
0 ones	00000	00000	0
1 one	00010	00000	0
2 ones	10010	00000	0
3 ones	01110	11111	1
4 ones	10111	11111	1
5 ones	11111	11111	1

Suppose $w = 0$ is the message. The encoder selects $\mathbf{x} = 00000$. If the error vector is $\mathbf{z} = 11001$, then the channel output is $\mathbf{y} = 11001$. The decoder, seeing that there are more 1's than 0's, selects the codeword 11111 and estimated message $\widehat{w} = 1$. Since $\widehat{w} \neq \widehat{w}$, a decoding error occurred.

Suppose $w = 1$ is the message. The encoder selects $\mathbf{x} = 11111$. If the error vector is $\mathbf{z} = 10010$, then the channel output is $\mathbf{y} = 01101$. The decoder, seeing that there are more 1's than 0's, selects the codeword 11111, which corresponds to the message $w = 1$. Since $w = \widehat{w}$, there are no decoding errors.

7.2.4 Probability of Decoder Error

Now, compute the probability of decoder error λ_1, λ_2 and P_e, when the channel error probability is $p = 0.1$. Consider the case $w = 1$, or $\mathbf{x} = 00000$ was transmitted. If $\mathbf{z}$ has 3, 4 or 5 ones, then $\mathbf{y}$ also will have 3, 4 or 5 ones, and the decoder will make an error. The probability of decoder error is the same as the probability that $\mathbf{z}$ has more than 2 ones:

$$\begin{aligned}
\lambda_1 &= \Pr[\widehat{\mathsf{W}} = 1 | \mathsf{W} = 0] && (7.19)\\
&= \Pr[\widehat{\mathsf{X}} = 11111 | \mathsf{X} = 00000] && (7.20)\\
&= \Pr[\mathsf{Z} \text{ has 3, 4 or 5 ones} | \mathsf{X} = 00000] && (7.21)\\
&= \sum_{m=3}^{5} \binom{5}{m} p_{\mathsf{Y}|\mathsf{X}}(1|0)^m p_{\mathsf{Y}|\mathsf{X}}(0|0)^{5-m} && (7.22)\\
&= \sum_{m=3}^{5} \binom{5}{m} 0.1^m 0.9^{5-m} \approx 0.00856 && (7.23)
\end{aligned}$$

If the message is $w = 2$, then by symmetry, the error probability is likewise:

$$\lambda_2 = \Pr[\widehat{\mathsf{W}} = 1 | \mathsf{W} = 2] \approx 0.00856 \quad (7.24)$$

The average probability of error is also:

$$\begin{aligned}
P_e &= \frac{1}{2}\left(\Pr[\widehat{\mathsf{W}} = 2 | \mathsf{W} = 1] + \Pr[\widehat{\mathsf{W}} = 1 | \mathsf{W} = 2]\right) && (7.25)\\
&\approx 0.00856. && (7.26)
\end{aligned}$$

Because the repeat code is simple, the probability of error can be computed, but for most codes codes, it is difficult to compute the probability of error exactly. When proving the achievability of the channel coding theorem, only an upper bound on the probability of decoder error is computed.

7.3 Channel Capacity

7.3.1 Motivation for Channel Capacity

How can we use the channel input most efficiently? Answering this question leads us to the capacity of the channel. First, we motivate this question by showing how to select the inputs of two example channels.

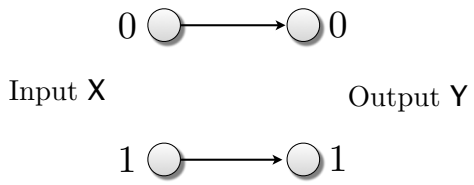


Figure 7.3: A noiseless channel, with capacity 1.

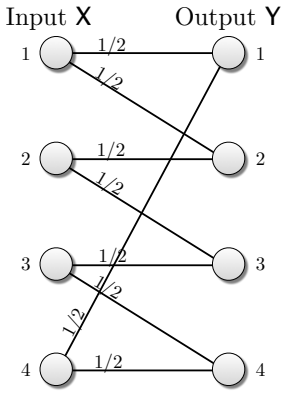


Figure 7.4: A simplified noisy typewriter channel, with four inputs and four outputs.

The discrete memoryless channel, or DMC, is a simple model for communications. The DMC was introduced in Subsection 2.1.5 on page 39. It consists of an input X, an output Y, and probability transitions $p_{\mathsf{Y}|\mathsf{X}}(y|x)$.

On a perfect communication channel, there is no noise. For example, consider a channel with two inputs $\mathcal{X} = \{0, 1\}$ and two outputs $\mathcal{Y} = \{0, 1\}$ and probability transitions:

$$p_{\mathsf{Y}|\mathsf{X}}(y|x) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \tag{7.27}$$

shown in Fig. 7.3. This channel can carry one bit of information each time the channel is used, or one bit per channel use.

How we use the channel inputs plays an important role in how much information the channel can carry. Consider the following channel:

$$p_{\mathsf{Y}|\mathsf{X}}(y|x) = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ 0 & \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & 0 & 0 & \frac{1}{2} \end{bmatrix}, \tag{7.28}$$

with alphabets $\mathcal{X} = \mathcal{Y} = \{1, 2, 3, 4\}$, shown in Fig. 7.4. Suppose we allow using all inputs $\{1, 2, 3, 4\}$ — but if $y = 2$ is received, then one cannot immediately distinguish between $x = 1$ and $x = 2$. Instead, consider we allow using only the inputs $\{1, 3\}$. If $y = 1$ or 2 is received, then $x = 1$ was transmitted and

no errors occurred. Similarly if $y = 3$ or 4, then only $x = 3$ could have been transmitted. Clearly, this channel can carry at least one bit.

7.3.2 Channel Capacity Definition

In the example of the noisy typewriter channel, how we use the inputs has an influence on the amount of information the channel can carry. More generally, the choice of the channel's input distribution $p_{\mathsf{X}}(x)$ affects how much information a channel can carry. In particular, the information capacity is the maximization over all possible input distributions. Later, we will show that the information capacity is the maximum amount of information a channel can carry.

Definition 7.6. For a discrete memoryless channel $p_{\mathsf{Y}|\mathsf{X}}(y|x)$, the *"information" capacity* C of a discrete memoryless channel is:

$$C = \max_{p_{\mathsf{X}}(x)} I(\mathsf{X};\mathsf{Y}). \tag{7.29}$$

Definition 7.7. An optimal $p^*_{\mathsf{X}}(x)$ is called the *capacity-achieving input distribution*:

$$p^*_{\mathsf{X}}(x) = \arg\max_{p_{\mathsf{X}}(x)} I(\mathsf{X};\mathsf{Y}). \tag{7.30}$$

The capacity-achieving input distribution $p^*_{\mathsf{X}}(x)$ is not necessarily unique, that is, there could be multiple input distributions that achieve the capacity.

Five properties related to channel capacity are given:

1. $C \geq 0$.
2. $C \leq \log |\mathcal{X}|$,
3. $C \leq \log |\mathcal{Y}|$.
4. $I(\mathsf{X};\mathsf{Y})$ is a continuous function of $p_{\mathsf{X}}(x)$.
5. $I(\mathsf{X};\mathsf{Y})$ is a concave function of $p_{\mathsf{X}}(x)$.

Properties 1–3 are easily shown using properties of mutual information: $I(\mathsf{X};\mathsf{Y}) \geq 0$, $I(\mathsf{X};\mathsf{Y}) \leq \log |\mathcal{X}|$ and $I(\mathsf{X};\mathsf{Y}) \leq \log |\mathcal{Y}|$. Property 4 is easily observed from (3.2) on page 57. For example, if $|\mathcal{X}| = 3$, then $p_{\mathsf{X}}(x) = [p_1, p_2, 1 - p_1 - p_2]$ and mutual information is a continuous function of p_1 and p_2.

Property 5 is shown in Chapter 13. Since maximization of a concave function is equivalent to minimization of a convex function, finding C is a convex optimization problem. In general, it is not easy to perform the maximization in (7.29). However, for certain DMCs, it is possible to find the capacity explicitly.

7.3.3 Capacity of the Zero-Error Channel

Recall from Subsection 1.2.1 on 18 that the entropy of the binary random variable $p_{\mathsf{X}}(x) = [1-p, p]$ is the binary entropy function $h(p)$:

$$h(p) = -\alpha \log \alpha - (1-p)\log(1-p), \tag{7.31}$$

and $h(p) = h(1-p)$.

In the *zero-error channel*, the input is reproduced exactly at the output, see Fig. 7.3 Probability transition matrix is

$$p_{\mathsf{Y}|\mathsf{X}}(y|x) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{7.32}$$

It should be clear enough that the capacity is 1. More concretely, recognize that $\mathsf{X} = \mathsf{Y}$, so $H(\mathsf{Y}|\mathsf{X}) = 0$ and $H(\mathsf{Y}) = H(\mathsf{X})$. With $p_{\mathsf{X}}(x) = [1-p, p]$:

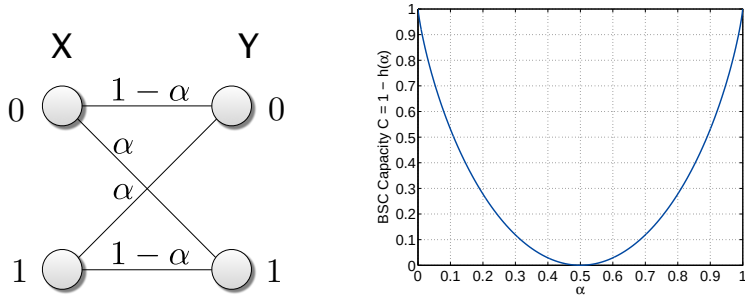
$$\begin{aligned} I(\mathsf{X};\mathsf{Y}) &= H(\mathsf{Y}) - H(\mathsf{Y}|\mathsf{X}) && \text{Definition of mutual information} \\ &= H(\mathsf{X}) \\ &= h(p) && \text{the binary entropy function} \\ C &= \max_{\alpha} I(\mathsf{X};\mathsf{Y}) \\ &= \max_{\alpha} h(p) \\ &= 1 && \text{see Fig. 1.1} \end{aligned}$$

and the capacity-achieving input distribution is $p^*_{\mathsf{X}}(x) = [\frac{1}{2}, \frac{1}{2}]$.

7.3.4 Capacity of the Binary Symmetric Channel

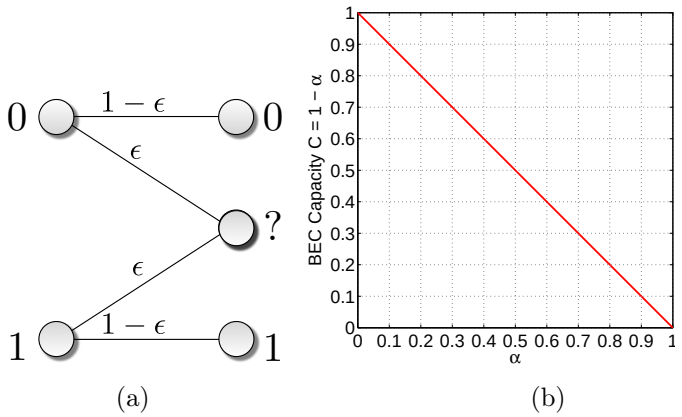
Recall the binary symmetric channel given in Subsection 7.2.2 has probability transition matrix:

$$p_{\mathsf{Y}|\mathsf{X}}(y|x) = \begin{bmatrix} 1-\alpha & \alpha \\ \alpha & 1-\alpha \end{bmatrix}, \tag{7.33}$$



(a) (b)

Figure 7.5: (a) Binary symmetric channel with error probability α. (b) Capacity of BSC is $C = 1 - h(\alpha)$.



(a) (b)

Figure 7.6: (a) Binary erasure channel (BEC) with error probability ϵ. (b) Capacity of BEC is $C = 1 - \epsilon$.

and is shown in Fig. 7.5-(a).

Proposition 7.1. The capacity of the binary symmetric channel (BSC) with error probability α is:

$$C = 1 - h(\alpha) \tag{7.34}$$

with capacity-achieving input distribution $p^*_{\mathsf{X}}(x) = [\frac{1}{2}, \frac{1}{2}]$.

Proof To find the capacity, upper bound the mutual information as:

$$\begin{aligned}
I(\mathsf{X};\mathsf{Y}) &= H(\mathsf{Y}) - H(\mathsf{Y}|\mathsf{X}) && \text{def. of mutual information} \\
&= H(\mathsf{Y}) - \sum_{x\in\mathcal{X}} p_{\mathsf{X}}(x) H(\mathsf{Y}|\mathsf{X}=x) \\
&= H(\mathsf{Y}) - \sum_{x\in\mathcal{X}} p_{\mathsf{X}}(x) h(\alpha) && \text{Given } x\text{, uncertainty is } \alpha \\
&= H(\mathsf{Y}) - h(\alpha) \\
&\leq 1 - h(\alpha) && \text{entropy upper bound}
\end{aligned}$$

Choosing $p_{\mathsf{X}}(x) = [\frac{1}{2}, \frac{1}{2}]$ will achieve the upper bound with equality ($h(\alpha)$ is a constant with respect to the maximization.) In this case, Y is a uniform random variable and $H(\mathsf{Y})$ is maximum. □

The capacity is plotted in Fig. 7.5-(b). Observe that if $\alpha = 0$, then the capacity $C = 1$. If $\alpha = 0.5$, then $C = 0$. That is, if the probability of error is 0.5, then channel has 0 capacity. Another important value is $\alpha \approx 0.11003$, which corresponds to $C = \frac{1}{2}$.

7.3.5 Capacity of the Binary Erasure Channel

In a binary erasure channel, some bits are lost rather than in error. It has two inputs $\mathcal{X} = \{0, 1\}$ and three outputs, $\mathcal{Y} = \{0, ?, 1\}$, where "?" represents an

erasure. The channel probability transition matrix is:

$$P_{\mathsf{Y}|\mathsf{X}} = \begin{bmatrix} 1-\epsilon & \epsilon & 0 \\ 0 & \epsilon & 1-\epsilon \end{bmatrix}, \tag{7.35}$$

and is shown in Fig. 7.6-(a).

Proposition 7.2. The capacity of the binary erasure channel (BEC) with erasure probability ϵ is:

$$C = 1 - \epsilon \tag{7.36}$$

with capacity-achieving input distribution $p^*_{\mathsf{X}}(x) = [\frac{1}{2}, \frac{1}{2}]$.

The capacity is plotted as a function of ϵ in Fig. 7.6-(b). Even if ϵ is close to 1, the channel still has non-zero capacity.

7.4 Matlab Source Code

7.4.1 Capacity of a Binary Input Channel

Consider the following binary-input, binary-output channel:

$$p_{\mathsf{X}|\mathsf{Y}}(x|y) = \begin{bmatrix} 1 & 0 \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix}. \tag{7.37}$$

The source code below makes a plot of the input distribution $p = p_{\mathsf{X}}(1)$ versus mutual information $I(\mathsf{X};\mathsf{Y})$, and finds the capacity C and capacity-achieving input distribution.

```
clear all
close all

pxgy = [ ...
    1     0 ; ...
    1/2   1/2];

%compute mutual information for all input distributions
p = linspace(eps, 1-eps,1001);
IXY = zeros(size(p));
for ii = 1:length(p)
    px      =[ 1-p(ii) ; p(ii) ] ;
    IXY(ii) = computeMutualInformation(pxgy, px );
end

%Capacity is maximum of mutual information
[C,ind] = max(IXY);
pstar   = p(ind);
fprintf('Capacity C = %g\n',C);
fprintf('Capacity-achieving input Cdistribution p* = %g\n',pstar);
```

```
21
22 plot(p,IXY)
23 hold on
24 plot(pstar,C,'bo')
25 text(pstar,C,'Capacity')
26 xlabel('Input distribution p');
27 ylabel('Mutual Information I(X;Y)');
28 grid on
```

Exercises

7.1 Consider the binary repeat code with n even on the binary symmetric channel with probability of error $\alpha = 0.05$. Since n is even, if $n/2$ ones are received, then the decoder should declare an error.

(a) Assuming the "majority vote" decoder, find an expression for the probability of error P_e with respect to n and α
(b) What is P_e when $n = 8$ and $\alpha = 0.05$?

7.2 Consider the repeat code with $n = 5$ on an *asymmetric* channel, with inputs $x \in \mathcal{X} = \{0, 1\}$ and outputs $y \in \mathcal{Y} = \{0, 1\}$:

$$p_{\mathsf{Y}|\mathsf{X}} = \begin{bmatrix} 0.8 & 0.2 \\ 0.3 & 0.7 \end{bmatrix} \tag{7.38}$$

where the rows are x and columns are y, for example $p_{\mathsf{Y}|\mathsf{X}}(1|0) = 0.2$. The decoder is the "majority vote" decoder. Compute the following:

(a) λ_1, the error probability given $\mathsf{X}^5 = 00000$,
(b) λ_2, the error probability given $\mathsf{X}^5 = 11111$,
(c) P_e the average error probability,
(d) $\lambda^{(5)}$ the maximum error probability:

$$\lambda^{(5)} = \max_{w \in \mathcal{W}} \lambda_w \tag{7.39}$$

7.3 Consider a commander of an army besieged in a fort for whom the only means of communication to his allies is a set of carrier pigeons. Assume that each carrier pigeon can carry 8 bits of information (i.e. one byte), that pigeons are released once every 5 minutes and that each pigeon takes 3 minutes to reach its destination.

(a) Assume that all pigeons reach his allies safely. What is the capacity of this link in bits/hour?
(b) Now assume that the enemy tries to shoot down the pigeon, and they manage to hit a fraction α of them. Since the pigeons are sent at a constant rate, the allies know when the pigeons are missing. What is the capacity of this link in bits/hour?

(c) Now assume that every time the enemy shoots down a pigeon, they send out a dummy pigeon with a uniform random 8-bit sequence. What is the capacity of this link, in bits/hour?

7.4 *Capacity of a BSC with correlated noise* Consider an independent and identically distributed noise sequence $\mathsf{W}_1, \mathsf{W}_2, \ldots, \mathsf{W}_n$ where $p_{\mathsf{W}}(w) = [1-p, p]$. For a BSC with output $\mathsf{U}_i = \mathsf{X}_i \oplus \mathsf{W}_i$, the capacity is $C_{\text{bsc}} = 1 - h(p)$. Now consider a *correlated* noise sequence $\mathbf{Z} = \mathsf{Z}_1, \mathsf{Z}_2, \ldots, \mathsf{Z}_n$ where each Z_i has the same distribution as each W_i (the Z_i are identically distributed, but not independent). The channel output is $\mathsf{Y}_i = \mathsf{X}_i \oplus \mathsf{Z}_i$. The X_i are independent and identically distributed, all variables are binary {0,1} and $\oplus$ is mod-2 addition.

(a) Prove that $H(\mathbf{Z}) \leq nh(p)$. Under what condition does equality hold?
(b) Show that $I(\mathbf{X}; \mathbf{Y}) \geq nC_{\text{bsc}}$.
(c) From part ((b)), conclude that correlated noise *increases* capacity.

7.5 Consider the channel with input $\mathcal{X} = \{1, 2, 3\}$, output $\mathcal{Y} = \{1, 2, 3\}$ and:

$$p_{\mathsf{Y}|\mathsf{X}}(y|x) = \begin{bmatrix} 0.3 & 0.2 & 0.5 \\ 0.5 & 0.3 & 0.2 \\ 0.2 & 0.5 & 0.3 \end{bmatrix} \tag{7.55}$$

where the rows correspond to x.

(a) Find $H(\mathsf{Y}|\mathsf{X} = x)$ for $x = 1, 2, 3$. Show this is equal to $H(\mathsf{Y}|\mathsf{X})$, independent of $p_{\mathsf{X}}(x)$.
(b) Show that if X is uniformly distributed, then Y is uniformly distributed.
(c) Show that the capacity is $\log 3 - h_3(0.5, 0.3, 0.2)$. Here $h_3(\cdot)$ is the ternary entropy function, the natural extension of the binary entropy function $h(\cdot)$.

7.6 *Capacity of binary adder channel* Consider a channel with input $\mathsf{X} \in \{0, 1\}$, with $p_{\mathsf{X}}(0) = 1 - p$ and $\Pr(\mathsf{X} = 1) = p$. Let the noise $\mathsf{Z} \in \{0, 1\}$ be distributed as $p_{\mathsf{Z}}(0) = p_{\mathsf{Z}}(1) = \frac{1}{2}$; X and Z are independent. The channel output is $\mathsf{Y} = \mathsf{X} + \mathsf{Z}$ with real addition, so that $\mathsf{Y} \in \{0, 1, 2\}$.

(a) Find the channel transition probability $\Pr[\mathsf{Y} = y|\mathsf{X} = x]$, joint distribution $\Pr[\mathsf{X} = x, \mathsf{Y} = y]$ and marginal distribution $\Pr[\mathsf{Y} = y]$ for all values of x and y.
(b) Find the mutual information $I(\mathsf{X}; \mathsf{Y})$.
(c) What is the capacity of this channel? What is the capacity-achieving input distribution?

7.7 *Errors and Erasures Channel.* Consider a 2-input, 3-output DMC, with $\mathcal{X} = \{0, 1\}$ and $\mathcal{Y} = \{0, ?, 1\}$. Let the probability of error be α and let the probability of erasure be ϵ, so the channel conditional probabilities are:

$$p_{\mathsf{Y}|\mathsf{X}}(y|x) = \begin{bmatrix} 1-\alpha-\epsilon & \epsilon & \alpha \\ \alpha & \epsilon & 1-\alpha-\epsilon \end{bmatrix} \tag{7.72}$$

Assume that the capacity achieving input distribution is $p_{\mathsf{X}}(0) = p_{\mathsf{X}}(1) = \frac{1}{2}$.

(a) Find the capacity of the errors and erasures channel.
(b) Verify that if $\epsilon = 0$, the capacity of the BSC is obtained.
(c) Verify that if $\alpha = 0$, the capacity of the BEC is obtained.

7.8 Consider a DMC with input X from $\mathcal{X} = \{0, 1, 2\}$. The noise Z is from $\mathcal{Z} = \{0, 1, 2\}$ with $p_{\mathsf{Z}}(z) = 1/3$. The channel output is $\mathsf{Y} = \mathsf{X} + \mathsf{Z}$. Find the capacity and the capacity-achieving input distribution.

7.9 Find the capacity and the capacity-achieving input distribution of the following channels.

(a)

$$p_{\mathsf{Y}|\mathsf{X}}(y|x) = \begin{bmatrix} \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{bmatrix} \tag{7.91}$$

(b)

$$p_{\mathsf{Y}|\mathsf{X}}(y|x) = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & 0 & \frac{1}{2} \end{bmatrix} \tag{7.92}$$

(c)

$$p_{\mathsf{Y}|\mathsf{X}}(y|x) = \begin{bmatrix} p & 1-p & 0 & 0 \\ 1-p & p & 0 & 0 \\ 0 & 0 & q & 1-q \\ 0 & 0 & 1-q & q \end{bmatrix} \tag{7.93}$$

7.10 Consider a DMC with inputs $|\mathcal{X}| = 2$ and outputs $|\mathcal{Y}| = 5$ and

$$p_{\mathsf{Y}|\mathsf{X}}(y|x) = \begin{bmatrix} \frac{5}{15} & \frac{4}{15} & \frac{3}{15} & \frac{2}{15} & \frac{1}{15} \\ 0 & 0 & \frac{10}{20} & \frac{6}{20} & \frac{4}{20} \end{bmatrix}. \tag{7.113}$$

(a) Let $p_{\mathsf{X}}(x) = [1-p, p]$. Make a plot of p versus $I(\mathsf{X};\mathsf{Y})$.

(b) Numerically find the channel capacity C.

(c) Numerically find the capacity-achieving value of p^*.

This problem is suitably solved using a computer.

Chapter 8

Channel Coding Theorem

Shannon's channel coding theorem is perhaps the most celebrated result in information theory. The previous chapter introduced both error-correcting codes and channel capacity. The channel coding theorem ties them together: If the code rate R is less than capacity C, then it is possible to make the probability of decoding error to be as small as you want. Conversely, if $R > C$, then reliable decoding is not possible.

It is presented in the following steps:

- Section 8.1 gives the two-variable AEP need to prove the chanel coding theorem. The two-variable AEP is the extension of the one-variable AEP used for source coding. Properties of the AEP are given.
- Section 8.2 states the channel coding theorem. The encoder and decoder used to proved the channel coding theorem is given.
- Section 8.3: The proof of the channel coding theory has two parts. The direct part is proved using the AEP.
- Section 8.4: The converse part is proved using Fano's inequality.

8.1 Joint Typicality and Joint AEP

8.1.1 Jointly Typical Sequences

Typical sequences and the AEP for one variable was given in Sections 5.2 and 5.3. This section introduces *joint typicality*, which extends the idea of typical sequences from one sequence $\mathbf{x}$, to two sequences $(\mathbf{x}, \mathbf{y})$. Given any pair of sequences, joint typicality answers the question, "given two sequences $\mathbf{x}$ and $\mathbf{y}$, can we say that $\mathbf{x}$ caused $\mathbf{y}$?" If $\mathbf{x}$ is the input to a DMC, and $\mathbf{y}$ is corresponding

Additional reading: Cover and Thomas, Sections 7.6 to 7.9.

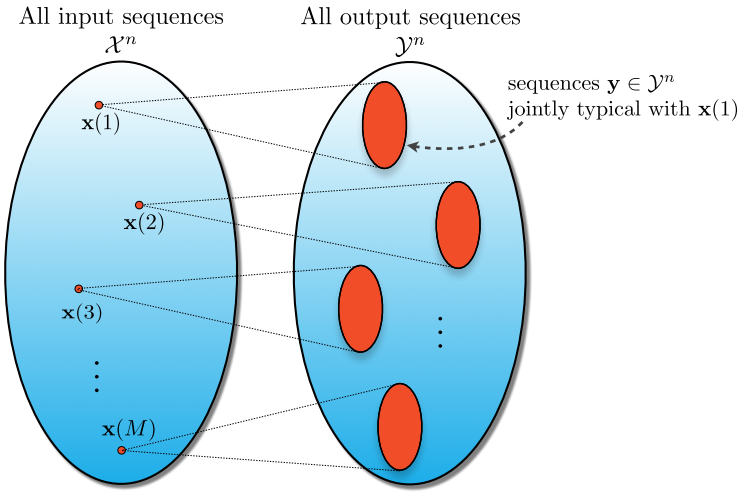


Figure 8.1: Illustration of jointly typical sequences.

output, then the answer is "yes," with high probability. But of $\mathbf{x}$ and $\mathbf{y}$ are independent sequences, then the answer is "no," with high probability.

Fig. 8.1 illustrates the idea of jointly typical sequences. If $\mathbf{x}(1)$ is a channel input, randomly drawn sequence from $\mathcal{X}^n$, then the group of channel output sequences $\mathbf{y}$ which are highly likely are jointly typical, and are shown in red.

Let $\mathbf{X}, \mathbf{Y}$ be jointly distributed as $p_{\mathbf{XY}}(\mathbf{x}, \mathbf{y})$. The individual entropies are $H(\mathsf{X})$ and $H(\mathsf{Y})$, and the joint entropy is $H(\mathsf{X}, \mathsf{Y})$. Sequences are written as:

$$\mathbf{x} = (x_1, x_2, \ldots, x_n) \tag{8.1}$$

$$\mathbf{y} = (y_1, y_2, \ldots, y_n) \tag{8.2}$$

$$(\mathbf{x}, \mathbf{y}) = \Big((x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\Big). \tag{8.3}$$

The case of interest is $\mathbf{X}$, an independent and identically distributed sequence of random variables, input to a DMC $p_{\mathsf{Y}|\mathsf{X}}(y|x)$. $\mathbf{Y}$ is the output of the DMC, with:

$$p_{\mathbf{XY}}(\mathbf{x}, \mathbf{y}) = \prod_{i=1}^{n} p_{\mathsf{XY}}(x_i, y_i) = \prod_{i=1}^{n} p_{\mathsf{Y}|\mathsf{X}}(y|x) p_{\mathsf{X}}(x). \tag{8.4}$$

Definition 8.1. The set $\mathcal{T}_\epsilon^{(n)}$ of *jointly typical* sequences is the set of sequences $\mathbf{x}, \mathbf{y} \in \mathcal{X}^n \times \mathcal{Y}^n$ with sample entropy ϵ-close to the true entropies:

$$\begin{aligned} \mathcal{T}_\epsilon^{(n)} = \Big\{ & (\mathbf{x}, \mathbf{y}) \in \mathcal{X}^n \times \mathcal{Y}^n : \\ & \Big| -\frac{1}{n} \log p_{\mathbf{X}}(\mathbf{x}) - H(\mathsf{X}) \Big| < \epsilon, \\ & \Big| -\frac{1}{n} \log p_{\mathbf{Y}}(\mathbf{y}) - H(\mathsf{Y}) \Big| < \epsilon, \\ & \Big| -\frac{1}{n} \log p_{\mathbf{XY}}(\mathbf{x}, \mathbf{y}) - H(\mathsf{X}, \mathsf{Y}) \Big| < \epsilon \Big\}. \end{aligned} \tag{8.5}$$

If $\mathbf{x}$ is typical only or if $\mathbf{y}$ is typical only, then $(\mathbf{x}, \mathbf{y})$ is not jointly typical. All three conditions must be satisfied for $(\mathbf{x}, \mathbf{y})$ to be jointly typical.

8.1.2 Numerical Example

Consider a binary input X with $p_{\mathsf{X}}(0) = 1 - t$ and $p_{\mathsf{X}}(1) = t$ on a BSC with error probability p. The channel conditional distribution is:

$$p_{\mathsf{Y}|\mathsf{X}}(y|x) = \begin{bmatrix} 1-p & p \\ p & 1-p \end{bmatrix} \tag{8.6}$$

and the joint distribution is:

$$p_{\mathsf{XY}}(x, y) = \begin{bmatrix} (1-p)(1-t) & p(1-t) \\ p \cdot t & (1-p)t \end{bmatrix}, \tag{8.7}$$

where rows are x, columns are y.

Consider the specific numerical case of $t = 0.25$, and $p = 0.11$. The joint distribution is:

$$p_{\mathsf{XY}}(x, y) = \begin{bmatrix} 0.6675 & 0.0825 \\ 0.0275 & 0.2225 \end{bmatrix}, \tag{8.8}$$

so $H(\mathsf{X}, \mathsf{Y}) = 1.311$. And $p_{\mathsf{Y}}(y) = [0.695, 0.305]$, so $H(\mathsf{Y}) = 0.8873$. Also, $H(\mathsf{X}) = 0.8113$.

For $n = 4$ and $\epsilon = 0.35$, is the pair $(\mathbf{x}, \mathbf{y}) = (0100, 1000)$ jointly typical? Test the three conditions:

- First, compute the sample entropy of $\mathbf{x} = 0100$:

$$-\frac{1}{n} \log p_{\mathbf{X}}(\mathbf{x}) = -\frac{1}{4} \log \left((\frac{3}{4})^3 \cdot (\frac{1}{4})^1 \right) \tag{8.9}$$

$$\approx 0.8113 \tag{8.10}$$

  Since $| -\frac{1}{n} \log p_{\mathbf{X}}(\mathbf{x}) - H(\mathsf{X})| = |0.8113 - 0.8113| = 0$ is less than $\epsilon = 0.35$, this sequence 0100 is typical in $\mathbf{X}$.

- Next, compute the sample entropy of $\mathbf{y} = 1000$:

$$-\frac{1}{n} \log p_{\mathbf{Y}}(\mathbf{y}) = -\frac{1}{4} \log \left((0.305) \cdot (0.695)^3 \right) \tag{8.11}$$

$$\approx 0.8220 \tag{8.12}$$

  Since $| -\frac{1}{n} \log p_{\mathbf{Y}}(\mathbf{y}) - H(Y)| = |0.8220 - 0.8133| = 0.0087$ is less than $\epsilon = 0.35$, this sequence is ϵ-close to its entropy.

- Finally, compute $-\frac{1}{n} \log p_{\mathbf{XY}}(\mathbf{x}, \mathbf{y})$. Note that the pair $\mathbf{x} = 0100, \mathbf{y} = 1000$

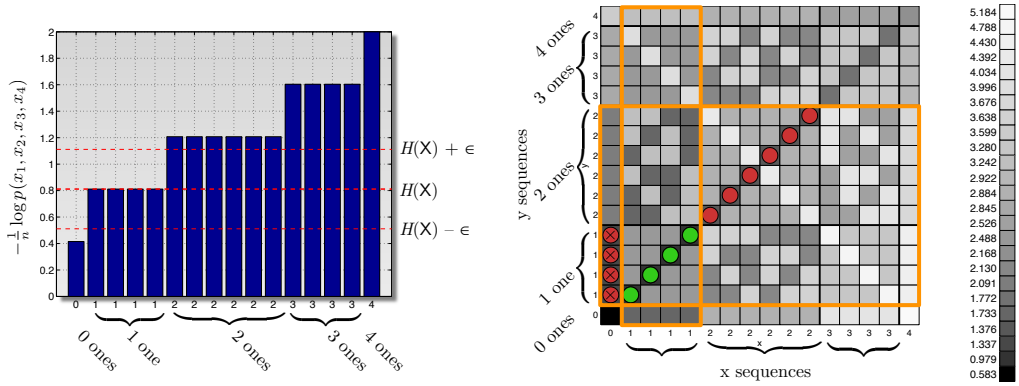


Figure 8.2: Joint typical sequences for $n = 4, \epsilon = 0.35$. (a) Empirical entropies for $\mathbf{x}$. (b) Joint sample entropies for $(\mathbf{x}, \mathbf{y})$. (c) Values of the joint sample entropies. Sequences which are ϵ-close to $H(\mathsf{X}, \mathsf{Y})$ are shown with dots; green dots indicate jointly typical sequences.

has two errors, so $p_{\mathsf{Y}|\mathsf{X}}(0|1) = p_{\mathsf{Y}|\mathsf{X}}(1|0) = 0.11$ and $p_{\mathsf{Y}|\mathsf{X}}(0|0) = 0.89$.

$$-\frac{1}{n}\log p_{\mathbf{XY}}(\mathbf{x}, \mathbf{y}) = -\frac{1}{n}\log p_{\mathbf{X}}(\mathbf{x}) - \frac{1}{n}\log p_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x}) \tag{8.13}$$

$$= -\frac{1}{n}\log \prod_{i=1}^{n} p_{\mathsf{X}}(x_i) - \frac{1}{n}\log \prod_{i=1}^{n} p_{\mathsf{Y}|\mathsf{X}}(y_i|x_i) \tag{8.14}$$

$$\approx 0.8113 - \underbrace{\frac{1}{4}\log(0.11)^2(0.89)^2}_{1.6763} \tag{8.15}$$

$$\approx 2.4876 \tag{8.16}$$

The difference between the sample entropy and $H(\mathsf{X}, \mathsf{Y})$ $|2.4876 - 1.311| \approx 1.1764$ is greater than $\epsilon = 0.35$ of the joint entropy, and so this $(\mathbf{x}, \mathbf{y})$ are not ϵ-close.

Only the first two conditions are satisfied. Since all three conditions are not satisfied, the pair $(\mathbf{x}, \mathbf{y}) = (0000, 1000)$ is not jointly typical.

This example can be used to visualize joint typicality. Fig. 8.2 expresses the sample entropies for all sequences. Fig. 8.2-(a) shows the sample entropies for X alone; so the typical set for $\mathbf{x}$ consists of sequences with 1 one. For $\mathbf{y}$, the typical set consists of sequences with 1 or 2 ones (figure not shown for $\mathbf{y}$).

Fig. 8.2-(b) shows the joint sample entropies for all sequences $\mathbf{x}$ and $\mathbf{y}$ in a 16×16 grid, with $\mathbf{x}$ sequences on the horizontal axis and $\mathbf{y}$ sequences on the vertical axis. The value of the joint sample entropy is shown using a light-dark scale, indicated in Fig. 8.2-(c), where light indicates higher values of sample entropy. Red dots indicate sequences which have sample entropy ϵ-close to the true entropy for $H(\mathsf{X})$ or $H(\mathsf{Y})$ but are not jointly typical. The four sequences

satisfying all three conditions for joint typicality are shown with green dots. The numerical example of $\mathbf{x} = 0000$ and $\mathbf{y} = 1000$ is shown using $\times$. Sequences which are typical in $\mathbf{x}$ (1 one) are shown with a heavy orange rectangle, and sequences which are typical in $\mathbf{y}$ (1 or 2 ones) are also shown with a heavy orange rectangle. These four sequences are:

$$\begin{aligned}\mathcal{T}_\epsilon^{(n)} = \{&(1000, 1000),\\ &(0100, 0100),\\ &(0010, 0010),\\ &(0001, 0001)\}.\end{aligned}$$

That is, sequences having 1 one in both $\mathbf{x}$ and $\mathbf{y}$ are jointly typical.

For this example, it can be seen that the channel input 1000 causes the output 1000, and only this output is **caused** in the sense of being jointly typical . Of course if n is larger, we expect that a given input should cause a variety of outputs.

8.1.3 Joint AEP

This section extends propositions for the single-variable AEP in Subsection 5.2.2 to the joint AEP. The first three propositions are the two-variable analogs of single-variable propositions.

Proposition 8.1. *Joint Asymptotic Equipartition Property.* Let $(\mathbf{x}, \mathbf{y})$ be sequences of length n drawn i.i.d. from $p_{\mathbf{XY}}(\mathbf{x}, \mathbf{y})$. Then:

$$\lim_{n\to\infty} \Pr\left[(\mathbf{X}, \mathbf{Y}) \in \mathcal{T}_\epsilon^{(n)}\right] = 1. \tag{8.17}$$

That is, the probability that a randomly drawn $(\mathbf{x}, \mathbf{y})$ are jointly typical approaches one as $n \to \infty$. As with single variable AEP, this is the law of large numbers applied to the sample entropy.

Next, three propositions for the AEP are given for sequences $(\mathbf{x}, \mathbf{y})$ of length n drawn i.i.d. from $p_{\mathbf{XY}}(\mathbf{x}, \mathbf{y})$.

Proposition 8.2. *Most sequences are typical*

$$\Pr\left[(\mathbf{X}, \mathbf{Y}) \in \mathcal{T}_\epsilon^{(n)}\right] \geq 1 - \epsilon, \tag{8.18}$$

for sufficiently large n. For the complementary event, $\Pr\left[(\mathbf{X}, \mathbf{Y}) \notin \mathcal{T}_\epsilon^{(n)}\right] < \epsilon$.

That is, a sequence drawn randomly from $p_{\mathsf{XY}}(x, y)$ typical with high probability. "Most sequences are jointly typical".

Proposition 8.3. *Upper bound on size of typical set* The size of the typical set is upper bounded as:

$$|\mathcal{T}_\epsilon^{(n)}| \leq 2^{n(H(\mathsf{X},\mathsf{Y})+\epsilon)}. \tag{8.19}$$

The following proposition deals with two random variables and does not have a single-variable analog. Consider that X, Y are jointly distributed. We let $\widetilde{\mathsf{X}}$ and $\widetilde{\mathsf{Y}}$ be *independent* random variables with the same distribution as X and Y. That is $(\widetilde{\mathsf{X}}, \widetilde{\mathsf{Y}}) \sim p_{\mathsf{X}}(x)p_{\mathsf{Y}}(y)$ and $\mathsf{X} \sim p_{\mathsf{X}}(\mathbf{x})$ and $\mathsf{Y} \sim p_{\mathsf{Y}}(\mathbf{y})$.

Proposition 8.4. *Independent sequences are not jointly typical* If $\widetilde{\mathsf{X}}$ and $\widetilde{\mathsf{Y}}$ are independent random variables with the same distribution as X and Y, then:

$$\Pr\left((\widetilde{\mathbf{X}}, \widetilde{\mathbf{Y}}) \in \mathcal{T}_\epsilon^{(n)}\right) \leq 2^{-n(I(\mathsf{X};\mathsf{Y})-3\epsilon)} \tag{8.20}$$

Note that $I(\mathsf{X}, \mathsf{Y}) \geq 0$, so $2^{-nI(\mathsf{X};\mathsf{Y})}$ is small for large n. Proposition 8.4 says that independent $\widetilde{\mathbf{X}}$ and $\widetilde{\mathbf{Y}}$ will be jointly typical with low probability. If $\mathbf{x}$ does not "explain" $\mathbf{y}$, then the probability they are jointly typical is small.

8.2 Channel Coding Theorem

This section first gives the channel coding theorem. Then the random codebook construction and jointly typical decoding to be used in the proof are given.

8.2.1 Channel Coding Theorem

The channel coding theorem is one of the central results of information theory. For a given DMC, is it possible to have reliable communications? In information theory, *reliable communications* means the maximum probability of decoding error $\lambda^{(n)}$ ((7.12) on page 121) can be made as small as wanted, by making the length of the code n as large as necessary. The lower the desired probability of decoding error, the longer the code length should be. As n goes to infinity, the probability of error goes to zero. The probability of error is non-zero for any finite n. If someone asks "For some DMC, does there exist a code with maximum error probability $\lambda^{(n)}$?" Then the answer is: yes, if $R < C$, and you allow n to be big enough.

The channel coding theorem states that reliable communications is possible if and only if the code rate R satisfies $R < C$, where C is the channel capacity.

Proposition 8.5. *Channel Coding Theorem* For every rate $R < C$, there exists a sequence of $(2^{nR}, n)$ codes with

$$\lim_{n\to\infty} \lambda^{(n)} = 0 \tag{8.21}$$

(direct part, or achievability). Conversely, any sequence of $(2^{nR}, n)$ codes with $\lim_{n\to\infty} \lambda^{(n)} = 0$ must have $R \leq C$. (converse part)

We often refer to an achievable rate to mean that it is known how to make the probability of error go to zero.

Definition 8.2. A rate R is an *achievable rate* for a sequence of $(2^{nR}, n)$ codes if the maximal probability of error $\lambda^{(n)}$ goes to zero as $n \to \infty$.

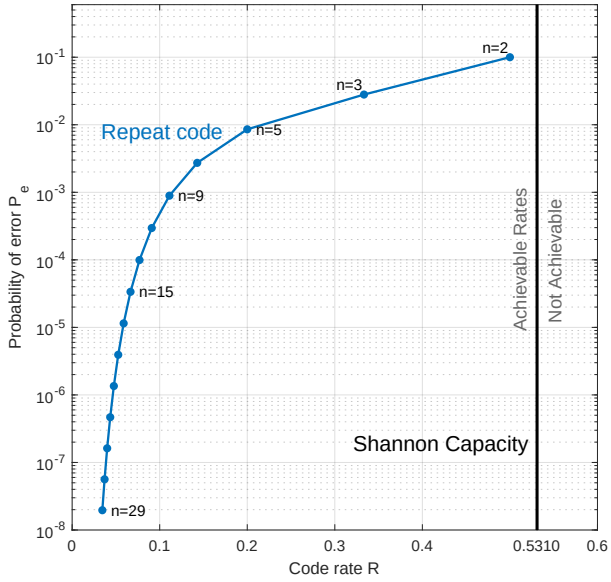


Figure 8.3: For the binary symmetric channel with error probability $\alpha = 0.1$, the capacity is $C = 0.531$. Only codes with rates $R < 0.531$ can achieve low probability of error. The probability of error for the repeat code with various block lengths n are also shown.

With this definition, the channel coding theorem can be stated as: all code rates below capacity C are achievable, and code rates above capacity are not achievable.

Comments on the channel coding theorem The channel coding theorem is an "if and only if" statement:

1. The "if" part, or direct part, says that for rates $R < C$, there exists a code for which reliable decoding is possible.

2. The "only if" part, or converse, says that if you have a code for which reliable decoding is possible, then it must be that $R < C$. In other words, it is not possible to have reliable communications when $R > C$.

The channel coding theorem is proven using random codebooks. The proof shows the existence of at least one good code. However, the proof does not give a technique to construct practical codes. Random codes and joint typical decoding is not efficient. There are however, practical codes such as low-density parity-check (LDPC) codes and polar codes, which have error-rate performance quite close to the channel capacity, when n is large enough.

Consider the example of the binary symmetric channel (BSC) with error probability $\alpha = 0.1$. For this channel, the capacity is $C = 0.531$. By the channel coding theorem, only codes with rates $R < 0.531$ can achieve low probability of error. One such error correcting code is the repeat code, and the probability of error is shown in Fig. 8.3. It can be seen that for $n \to \infty$, the probability

of error for the repeat code get small, but the rate R also gets small. At a probability of error such as $P_e = 10^{-7}$, there is a large gap between the repeat code and the capacity — this indicates that the repeat code is poor code for this channel. Indeed, there are much better codes, but such codes are not covered in this course, as information theory deals with the fundamental limits of communication.

8.2.2 Encoder and Channel

Generate an (M, n) code by randomly generating M codewords $\mathbf{x}$ of length n from a distribution $p_{\mathsf{X}}(x)$. The codebook $\mathcal{C}$ is:

$$\mathcal{C} = \begin{bmatrix} x_1(1) & x_2(1) & \cdots & x_n(1) \\ x_1(2) & x_2(2) & \cdots & x_n(2) \\ \vdots & & & \\ x_1(w) & x_2(w) & \cdots & x_n(w) \\ \vdots & & & \\ x_1(M) & x_2(M) & \cdots & x_n(M) \end{bmatrix} \tag{8.22}$$

Each entry $x_i(w)$ generated independent and identically distributed according to $p_{\mathsf{X}}(x)$. The code rate is $R = \frac{1}{n} \log M$, or $M = 2^{nR}$, and so the message w is from the set $\{1, 2, \ldots, 2^{nR}\}$. The transmitted codeword is $\mathbf{x}(w)$, one row of the above matrix.

The message w is chosen uniformly at random from $\{1, 2, \ldots, 2^{nR}\}$. The message random variable W has entropy $H(\mathsf{W}) = \log 2^{nR} = nR$. The codeword $\mathbf{x}(w)$ is transmitted over a DMC $p_{\mathsf{Y}|\mathsf{X}}(y|x)$, and received as $\mathbf{y}$. Note that the codebook is generated according to $p_{\mathsf{X}}(x)$ from $\mathcal{X}$. So the codebook alphabet $\mathcal{X}$ is the same as the channel input alphabet. Also, the code symbol distribution $p_{\mathsf{X}}(x)$ is the same as the channel input distribution.

8.2.3 Decoding

The randomly-generated codebook $\mathcal{C}$ is known to both the encoder and decoder. Using the channel output $\mathbf{y}$, jointly typical decoding is performed, as follows. The decoder estimates $\widehat{w}$ was sent if:

1. $(\mathbf{x}(\widehat{w}), \mathbf{y})$ are jointly typical, and
2. There is no other index w' such that $(\mathbf{x}(w'), \mathbf{y})$ are jointly typical, where $w' \neq \widehat{w}$.

Define the event E_w as:

$$\mathsf{E}_w = \text{Event that } (\mathbf{x}(w), \mathbf{y}) \text{ are jointly typical.} \tag{8.23}$$

$$\overline{\mathsf{E}}_w = \text{Event that } (\mathbf{x}(w), \mathbf{y}) \text{ are not jointly typical.} \tag{8.24}$$

Assuming that $w = 1$ is transmitted, a decoding error occurs if $\overline{\mathsf{E}}_1$ or any of $\mathsf{E}_2, \mathsf{E}_3, \ldots, \mathsf{E}_{2^{nR}}$ occurs. Let E be the decoder event error, then

$$\mathsf{E} = \overline{\mathsf{E}}_1 \cup \mathsf{E}_2 \cup \mathsf{E}_3 \cup \cdots \cup \mathsf{E}_{2^{nR}}, \tag{8.25}$$

and $2^{nR} = M$ is the number of messages.

8.3 Proof of Channel Coding Theorem — Direct Part

To prove the Channel Coding Theorem, we assume the communication system in the previous section: the codebook is randomly constructed, the channel is a DMC, and the decoder uses jointly-typical decoding.

8.3.1 Probability of Decoder Error

We distinguish between the probability of error $P_e(\mathcal{C})$ for a specific code $\mathcal{C}$:

$$P_e(\mathcal{C}) = \frac{1}{M} \sum_{w=1}^{M} \lambda_w(\mathcal{C}), \tag{8.26}$$

and the probability of error averaged over all codes, called $\Pr(\mathsf{E})$:

$$\Pr(\mathsf{E}) = \sum_{\mathcal{C}} P_e(\mathcal{C}) \Pr(\mathcal{C}) \tag{8.27}$$

For a fixed codebook $\mathcal{C}'$, the probability of error $P_e(\mathcal{C}')$ is very difficult to compute. But, the average probability of error $\Pr(\mathsf{E})$, averaged over all codes, it is possible to show the existence of at least one good code.

By the symmetry of the code construction, the probability of error averaged over all codes does not depend on which message w was transmitted. That is $\sum_{\mathcal{C}} \Pr(\mathcal{C})\lambda_w$ does not depend on w. Thus, analyzing the error probability under the assumption the message $w = 1$ was transmitted is sufficient:

$$\Pr(\mathsf{E}) = \Pr(\mathsf{E}|\mathsf{W} = 1) \tag{8.28}$$

Without loss of generality, assume $w = 1$ is the message, so $\mathbf{x}(1)$ is transmitted.

8.3.2 Proof of Direct Part of Proposition 8.5

First, the proof upper bounds the average probability of error $\Pr(\mathsf{E})$. Then the upper bound is extended to $\lambda^{(n)}$.

If $\mathbf{x}(1)$ is the input to the channel and $\mathbf{y}$ is the output, then the event E_1 is the same as $\mathbf{X}$ and $\mathbf{Y}$ being jointly typical. By Proposition 8.2:

$$\Pr(\overline{\mathsf{E}}_1) \leq \epsilon. \tag{8.29}$$

Event $\overline{\mathsf{E}}_1$ means $\mathbf{x}(1)$ and $\mathbf{y}$ are not jointly typical.

Due to the random code construction, $\mathbf{x}(1)$ and $\mathbf{x}(w)$ are independent for $w \neq 1$. Since $\mathbf{x}(1)$ is the channel input and $\mathbf{y}$ is the channel output, it must be that $\mathbf{x}(w)$ and $\mathbf{y}$ are independent, for $w \neq 1$. By Proposition 8.4:

$$\Pr(\mathsf{E}_i) \leq 2^{-n(I(\mathsf{X};\mathsf{Y})-3\epsilon)} \tag{8.30}$$

for $i = 2, 3, \ldots, 2^{nR}$.

A decoding error event E occurs if any of those events occur, that is, for the case of $w = 1$, a decoding error is the union of events:

$$\mathsf{E} = \overline{\mathsf{E}}_1 \cup \mathsf{E}_2 \cup \mathsf{E}_3 \cup \cdots \cup \mathsf{E}_{2^{nR}}. \tag{8.31}$$

Now upper bound the probability of error $\Pr(\mathsf{E})$ as:

$$\begin{aligned}
\Pr(\mathsf{E}) &= \Pr(\overline{\mathsf{E}}_1 \cup \mathsf{E}_2 \cup \cdots \cup \mathsf{E}_{2^{nR}}) \\
&\leq \Pr(\overline{\mathsf{E}}_1) + \sum_{i=2}^{2^{nR}} \Pr(\mathsf{E}_i) && \text{union bound} \\
&\leq \epsilon + \sum_{i=2}^{2^{nR}} 2^{-n(I(\mathsf{X};\mathsf{Y})-3\epsilon)} && \text{by (8.29) and (8.30)} \\
&= \epsilon + (2^{nR}-1)2^{-n(I(\mathsf{X};\mathsf{Y})-3\epsilon)} \\
&\leq \epsilon + 2^{nR}2^{-n(I(\mathsf{X};\mathsf{Y})-3\epsilon)} \\
&= \epsilon + 2^{-n(I(\mathsf{X};\mathsf{Y})-3\epsilon-R)} \\
&\leq 2\epsilon. && \text{Given } \epsilon, \text{ choose } n \text{ large enough, see below}
\end{aligned}$$

If $R < I(\mathsf{X};\mathsf{Y}) - 3\epsilon$, then it is possible to make n large enough that $2^{-n(I(\mathsf{X};\mathsf{Y})-3\epsilon-R)} < \epsilon$, since ϵ is a constant. Recall capacity $C = \max_{p_\mathsf{X}(x)} I(\mathsf{X};\mathsf{Y})$, and choose $p_\mathsf{X}(x)$ to be equal to a capacity-maximizing input distribution $p^*_\mathsf{X}(x)$. Then, replace $I(\mathsf{X};\mathsf{Y})$ with C. Thus,

$$\Pr(\mathsf{E}) \leq 2\epsilon \text{ if } R < C - 3\epsilon, \tag{8.32}$$

is the upper bound on the average probability of error.

Since the average error of probability is no more than 2ϵ, there must be at least one codebook $\mathcal{C}^*$ with error probability:

$$P_e(\mathcal{C}^*) \leq 2\epsilon. \tag{8.33}$$

Finally, we want to upper bound the maximum probability of error. Some codewords have conditional error rates higher than 2ϵ, but no more than half have conditional error rate greater than 4ϵ. These codewords with high conditional error rate are expurgated, or removed so that the maximum probability of error is:

$$\lambda^{(n)} \leq 4\epsilon. \tag{8.34}$$

Expurgating half the codewords reduces the number of bits in the message alphabet $\log|\mathcal{W}|$ by one bit, so the rate of the new code is $R - \frac{1}{n}$.

Thus, the maximal probability of error is $\lambda^{(n)} \leq 4\epsilon$ if $R - \frac{1}{n} < I(\mathsf{X};\mathsf{Y}) - 3\epsilon$. This completes the proof. □

8.4 Proof of Channel Coding Theorem — Converse Part

8.4.1 Modification of Fano's Inequality

Recall Fano's inequality (Proposition 3.8 on page 65) for the Markov chain $\mathsf{X} \to \mathsf{Y} \to \widehat{\mathsf{X}}$, where $\widehat{\mathsf{X}}$ is an estimator of X, and the probability of error $P_e = \Pr(\mathsf{X} \neq \widehat{\mathsf{X}})$. Then, $h(P_e) + P_e \log |\mathcal{X}| \geq H(\mathsf{X}|\widehat{\mathsf{X}}) \geq H(\mathsf{X}|\mathsf{Y})$ holds. Now make a slight modification of Fano's inequality for channel coding.

Proposition 8.6. *Fano's inequality for channel coding* For a DMC, a codebook $\mathcal{C}$ of length n and rate R, and input message W:

$$H(\mathsf{W}|\widehat{\mathsf{W}}) \leq 1 + P_e^{(n)} nR. \tag{8.35}$$

Proof For the communication system:

$$w \to \mathbf{x}(w) \to \mathbf{y} \to \widehat{w} \tag{8.36}$$

forms a Markov chain. Since $h(P_e) \leq 1$, take $h(P_e) = 1$. Since W is uniformly distributed on $\mathcal{W} = \{1, 2, \ldots, 2^{nR}\}$ we have $H(\mathsf{W}) = \log |\mathcal{W}| = \log 2^{nR} = nR$.

So Fano's inequality for $\mathsf{W}, \widehat{\mathsf{W}}$ (taking the place of $\mathsf{X}, \widehat{\mathsf{X}}$) is:

$$\begin{aligned} H(\mathsf{W}|\widehat{\mathsf{W}}) &\leq h(P_e) + P_e \log |\mathcal{W}| && (8.37)\\ &\leq h(P_e) + P_e nR, && (8.38)\\ &\leq 1 + P_e nR. \qquad \square && (8.39) \end{aligned}$$

8.4.2 Channel reuse does not increase capacity

Capacity is not increased by using the channel many times.

Proposition 8.7. *Channel Reuse* Let $\mathbf{Y}$ be the result of passing $\mathbf{X}$ through a DMC of capacity C. Then:

$$I(\mathbf{X}; \mathbf{Y}) \quad \leq \quad nC \quad \text{for all } p_{\mathsf{X}}(x) \tag{8.40}$$

Proof

$$\begin{aligned} I(\mathbf{X}; \mathbf{Y}) &= H(\mathbf{Y}) - H(\mathbf{Y}|\mathbf{X}) \\ &= H(\mathbf{Y}) - \sum_{i=1}^{n} H(\mathsf{Y}_i|\mathsf{Y}_1, \ldots, \mathsf{Y}_{i-1}, \mathbf{X}) \\ &= H(\mathbf{Y}) - \sum_{i=1}^{n} H(\mathsf{Y}_i|\mathsf{X}_i) \\ &\leq \sum_{i=1}^{n} H(\mathsf{Y}_i) - \sum_{i=1}^{n} H(\mathsf{Y}_i|\mathsf{X}_i) \\ &= \sum_{i=1}^{n} I(\mathsf{X}_i; \mathsf{Y}_i) \\ &\leq nC \end{aligned}$$

8.4.3 Proof of Converse

Proof of Converse of Proposition 8.5 Show that any sequence of codes $(2^{nR}, n)$ codes with $\lambda^{(n)} \to 0$ must have $R \leq C$. Note that because $\lambda^{(n)} \geq P_e^{(n)}$ holds, $\lambda^{(n)} \to 0$ implies $P_e^{(n)} \to 0$, as $n \to \infty$.

$$\begin{aligned}
nR &= H(\mathsf{W}) \\
&= H(\mathsf{W}|\widehat{\mathsf{W}}) + I(\mathsf{W};\widehat{\mathsf{W}}) \\
&\leq 1 + P_e^{(n)} nR + I(\mathsf{W};\widehat{\mathsf{W}}) \\
&\leq 1 + P_e^{(n)} nR + I(\mathbf{X};\mathbf{Y}) \\
&\leq 1 + P_e^{(n)} nR + nC \\
R &\leq \frac{1}{n} + P_e^{(n)} R + C \\
R &\leq \lim_{n\to\infty} \frac{1}{n} + P_e^{(n)} R + C \\
R &\leq C
\end{aligned}$$

That is, for any code that has probability of error going to 0, it must be true that $R \leq C$. □

Also,

$$P_e^{(n)} \geq 1 - \frac{C}{R} - \frac{1}{nR} \tag{8.41}$$

So even if $n \to \infty$, $P_e^{(n)}$ is bounded away from 0 for $R > C$ This is the weak converse: error rates above capacity grow as $\frac{1}{n}$.

Chapter 9

Differential Entropy and the Gaussian Channel

In this chapter, we move from the world of discrete random variables, to continuous random variables. Entropy, called differential entropy in the continuous case, the KL divergence, mutual information and the AEP all have analogs in the continuous domain. Most familiar properties, such as entropy chain rules, still hold. Two important exceptions are that differential entropy can be negative (always non-negative in the discrete case) and the entropy-maximizing distribution is the Gaussian distribution (is the uniform distribution in the discrete case).

This chapter introduces the Gaussian channel, a continuous-input, continuous-output channel. The capacity of the Gaussian channel is $C = \frac{1}{2}\log(1 + \text{SNR})$, where SNR is the signal-to-noise ratio of the channel. Shannon's channel coding theorem states that reliable communication is possible if and only if the rate is less than capacity.

9.1 Differential Entropy

9.1.1 Continuous Random Variables

This section briefly reviews probability theory for continuous random variables.

Until now we assumed discrete random variables, for example X had a sample space $\mathcal{X} = \{1, 2, \ldots, |\mathcal{X}|\}$. In this chapter, *continuous* random variables with a continuous sample space, for example $\mathcal{X} = \mathbb{R}$ are considered.

Definition 9.1. A *continuous random variable* X has a probability distribution

Additional reading: Cover and Thomas, Sections 8.1 to 8.6, 9.1, 9.4.

$p_{\mathsf{X}}(x)$ or $f(x)$ which satisfies:

$$p_{\mathsf{X}}(x) \geq 0 \text{ for all } x \in \mathbb{R}, \text{ and} \tag{9.1}$$

$$\int_{-\infty}^{\infty} p_{\mathsf{X}}(x)dx = 1. \tag{9.2}$$

The continuous random variable X has a cumulative distribution function $F_{\mathsf{X}}(x)$:

$$F_{\mathsf{X}}(x) \quad = \quad \Pr(\mathsf{X} \leq x) = \int_{-\infty}^{x} p_{\mathsf{X}}(u)du \tag{9.3}$$

The set where $p_{\mathsf{X}}(x) > 0$ is called the *support set* $\mathcal{X}$:

$$\mathcal{X} \quad = \quad \{x \in \mathbb{R} \mid p_{\mathsf{X}}(x) > 0\} \tag{9.4}$$

From this, for any $a < b$:

$$\Pr(a \leq \mathsf{X} \leq b) = F_{\mathsf{X}}(b) - F_{\mathsf{X}}(a) = \int_{a}^{b} p_{\mathsf{X}}(u)du. \tag{9.5}$$

Example 9.1. The distribution $p_{\mathsf{X}}(x)$ uniform on $(0, a)$ is:

$$p_{\mathsf{X}}(x) \quad = \quad \begin{cases} 1/a & \text{if } 0 \leq x < a \\ 0 & \text{otherwise} \end{cases} . \tag{9.6}$$

The Gaussian (or normal) distribution $p_{\mathsf{X}}(x)$ with mean m, variance σ^2 is:

$$p_{\mathsf{X}}(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-m)^2}{2\sigma^2}} \text{ for } x \in \mathbb{R} \tag{9.7}$$

We often abbreviate the Gaussian distribution as $\mathsf{X} \sim \mathcal{N}(m, \sigma^2)$.

The *mean* of a continuous-valued random variable X is

$$E[\mathsf{X}] = \int_{-\infty}^{\infty} x p_{\mathsf{X}}(x)dx. \tag{9.8}$$

Given a function g, the expectation of a function g of X is $E[g(\mathsf{X})] = \int_{-\infty}^{\infty} g(x)p_{\mathsf{X}}(x)dx$.

The *variance* of X is $\mathrm{Var}[\mathsf{X}]$ is:

$$\mathrm{Var}[\mathsf{X}] = E[\mathsf{X}^2] - (E[\mathsf{X}])^2 \tag{9.9}$$

$$= \int_{-\infty}^{\infty} x^2 p_{\mathsf{X}}(x)dx - \left(\int_{-\infty}^{\infty} x p_{\mathsf{X}}(x)dx \right)^2 . \tag{9.10}$$

The probability distribution $p_{\mathsf{Z}}(z)$ of the sum $\mathsf{Z} = \mathsf{X} + \mathsf{Y}$ of two independent random variables X and Y is found by convolution of $p_{\mathsf{X}}(x)$ and $p_{\mathsf{Y}}(y)$. Let independent X, Y have probability distributions $p_{\mathsf{X}}(x), p_{\mathsf{Y}}(y)$ respectively. Then $p_{\mathsf{Z}}(z)$ is given by:

$$p_{\mathsf{Z}}(z) = (p_{\mathsf{X}} * p_{\mathsf{Y}})(z) = \int_{-\infty}^{\infty} p_{\mathsf{X}}(z-w)p_{\mathsf{Y}}(w)dw = \int_{-\infty}^{\infty} p_{\mathsf{X}}(w)p_{\mathsf{Y}}(z-w)dw \tag{9.11}$$

Example 9.2. Let $p_{\mathsf{X}}(x) = e^{-x}$ for $x \geq 0$ and $p_{\mathsf{Y}}(y) = e^{-y}$ for $y \geq 0$. Then the distribution of $\mathsf{Z} = \mathsf{X} + \mathsf{Y}$ is:

$$p_{\mathsf{Z}}(z) = \int_0^z e^{-(z-w)} e^{-w} dw \tag{9.12}$$

$$= z e^{-z}. \quad \square \tag{9.13}$$

9.1.2 Single-Variable Differential Entropy

Now we consider the entropy, mutual information and KL divergence for continuous random variables.

Definition 9.2. The *differential entropy*[1] $H(\mathsf{X})$ of a continuous random variable X with probability distribution $p_{\mathsf{X}}(x)$ is:

$$H(\mathsf{X}) = -\int_{\mathcal{X}} p_{\mathsf{X}}(x) \log p_{\mathsf{X}}(x) dx \quad \text{in bits} \tag{9.14}$$

$$H(\mathsf{X}) = -\int_{\mathcal{X}} p_{\mathsf{X}}(x) \ln p_{\mathsf{X}}(x) dx \quad \text{in nats} \tag{9.15}$$

Example 9.3. Find the entropy $H(\mathsf{X})$ of the uniform distribution given by (9.6):

$$H(\mathsf{X}) = -\int_0^a \frac{1}{a} \log \frac{1}{a} dx = \log a \tag{9.16}$$

Note: for $a < 1, \log a < 0$ so differential entropy is negative.

Example 9.4. Find $H(\mathsf{X})$ for the 0 mean Gaussian distribution $\mathcal{N}(0, \sigma^2)$:

$$H(\mathsf{X}) = -\int p_{\mathsf{X}}(x) \ln p_{\mathsf{X}}(x) dx \tag{9.17}$$

$$= -\int p_{\mathsf{X}}(x) \Big(-\frac{x^2}{2\sigma^2} - \ln \sqrt{2\pi\sigma^2} \Big) \tag{9.18}$$

$$= \frac{EX^2}{2\sigma^2} + \frac{1}{2} \ln 2\pi\sigma^2 \tag{9.19}$$

$$= \frac{1}{2} + \frac{1}{2} \ln 2\pi\sigma^2 \tag{9.20}$$

$$= \frac{1}{2} \ln e + \frac{1}{2} \ln 2\pi\sigma^2 \tag{9.21}$$

$$= \frac{1}{2} \ln 2\pi e\sigma^2 \text{ in nats} \tag{9.22}$$

Or,

$$H(\mathsf{X}) = \frac{1}{2} \log 2\pi e\sigma^2 \text{ in bits} \tag{9.23}$$

[1] Many authors write $h(\mathsf{X})$ for the differential entropy. Here, $H(\mathsf{X})$ is used to avoid confusion with the binary entropy function. Alternatively, sometimes $H(f)$ is written, if the probability distribution is $f(x)$.

Proposition 9.1. Translation does not change differential entropy. For a constant c:

$$H(\mathsf{X}+c) = H(\mathsf{X}) \tag{9.24}$$

Proposition 9.2. For a non-zero constant a, $H(a\mathsf{X}) = H(\mathsf{X}) + \log|a|$.

Note that while Proposition 9.1 holds for discrete random variables, Proposition 9.2 does not hold for discrete random variables.

SSQ 9.1. Find the differential entropy $-\int p_{\mathsf{X}}(x)\ln p_{\mathsf{X}}(x)dx$ of

1. the exponential density $\lambda e^{-\lambda x}$, $x \geq 0$,
2. the Laplace desnity $\frac{1}{2}\lambda e^{-\lambda|x|}$.

9.1.3 Multivariable Differential Entropy

This section covers multivariable differential entropy, KL divergence and mutual information.

Definition 9.3. The *differential entropy* of $\mathsf{X}_1, \mathsf{X}_2, \ldots, \mathsf{X}_n$ with joint density $p_{\mathbf{X}}(\mathbf{x})$

$$H(\mathsf{X}_1, \ldots, \mathsf{X}_n) = -\int \cdots \int p_{\mathbf{X}}(\mathbf{x}) \log p_{\mathbf{X}}(\mathbf{x}) d\mathbf{x} \tag{9.25}$$

With $\mathbf{X} = \mathsf{X}_1, \mathsf{X}_2, \ldots, X_n$, this is also written $H(\mathbf{X})$.

Definition 9.4. If X, Y are jointly distributed as $p_{\mathsf{XY}}(x, y)$, then *conditional differential entropy* is:

$$H(\mathsf{X}|\mathsf{Y}) = -\int \int p_{\mathsf{XY}}(x, y) \log p_{\mathsf{X}|\mathsf{Y}}(x|y) dx dy \tag{9.26}$$

Let $\mathbf{A}$ be an n-by-n square matrix. Since $\mathbf{X}$ is vector then $\mathbf{AX}$ is also a vector. Its differential entropy is given by the following proposition. This generalizes Proposition 9.2 to jointly distributed random variables.

Proposition 9.3. For $\mathbf{X} = \mathsf{X}_1, \ldots, \mathsf{X}_n$ and an n-by-n square matrix $\mathbf{A}$ the entropy of $\mathbf{AX}$ is:

$$H(\mathbf{AX}) \quad = \quad H(\mathbf{X}) + \log|\det(\mathbf{A})| \tag{9.27}$$

When $n = 1$, the entropy is $H(a\mathsf{X}) = H(\mathsf{X}) + \log|a|$.

Just as in the discrete case, the following holds:

$$H(\mathsf{X}|\mathsf{Y}) = H(\mathsf{X}, \mathsf{Y}) - H(\mathsf{Y}), \tag{9.28}$$

as well as the following three propositions.

Proposition 9.4. *Conditioning reduces entropy* For jointly distributed X, Y:

$$H(\mathsf{X}|\mathsf{Y}) \quad \leq \quad H(\mathsf{X}) \tag{9.29}$$

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

Proposition 9.5. *Chain rule for differential entropy* For jointly distributed $\mathsf{X}_1, \mathsf{X}_2, \ldots, \mathsf{X}_n$:

$$H(\mathsf{X}_1, \ldots, \mathsf{X}_n) \quad = \quad \sum_{i=1}^{n} H(\mathsf{X}_i|\mathsf{X}_1, \ldots, \mathsf{X}_{i-1}) \tag{9.30}$$

Proposition 9.6. *Independence bound on entropy* For jointly distributed $\mathsf{X}_1, \ldots, \mathsf{X}_n$:

$$H(\mathsf{X}_1, \ldots, \mathsf{X}_n) \leq \sum_{i=1}^{n} H(\mathsf{X}_i). \tag{9.31}$$

SSQ 9.2. Find the joint differential entropy $H(\mathbf{X})$ when $\mathbf{X} = \mathsf{X}_1\mathsf{X}_2$ is uniformly distributed on the rectangle $[0, 3] \times [0, 5]$:

$$p_{\mathbf{X}}(x_1, x_2) = \begin{cases} \frac{1}{15} & 0 \leq x_1 \leq 3, 0 \leq x_2, \leq 5 \\ 0 & \text{otherwise} \end{cases}.$$

9.1.4 KL Divergence and Mutual Information

Definition 9.5. The *Kullback-Leibler divergence* $D\big(f(x)||g(x)\big)$ between distributions $f(x)$ and $g(x)$ is given by:

$$D\big(f(x)||g(x)\big) = \int f(x) \log \frac{f(x)}{g(x)} dx \tag{9.32}$$

The relative entropy is a "distance" between the distance between a true density $f(x)$ and another density $g(x)$.

Proposition 9.7. $D(f||g) \geq 0$ with equality if and only if $f = g$.

Proof For a convex function ϕ, Jensen's inequality states:

$$E[\phi(\mathsf{X})] \geq \phi(E[\mathsf{X}]) \tag{9.33}$$

$$\int_{\mathcal{X}} p_{\mathsf{X}}(x)\phi(x)dx \geq \phi\Big(\int_{\mathcal{X}} x p_{\mathsf{X}}(x)\Big) \tag{9.34}$$

(see Proposition 13.1 on page 211). Then, taking $\phi(x) = \log x$, where log is a

convex function and writing KL divergence as

$$-D(f(x)||g(x)) = \int_{\mathcal{X}} f(x) \log \frac{g(x)}{f(x)} \tag{9.35}$$

$$\leq \log \int_{\mathcal{X}} f(x) \frac{g(x)}{f(x)} \quad \text{Jensen's inequality} \tag{9.36}$$

$$= \log \int_{\mathcal{X}} g(x) \tag{9.37}$$

$$\leq \log 1 = 0 \tag{9.38}$$

$$D(f(x)||g(x)) \geq 0 \tag{9.39}$$

Note Jensen's inequality was used in the form $E[\phi(g(\mathsf{X}))] \geq \phi(E[g(\mathsf{X})])$. □

Definition 9.6. Let X and Y be continuous jointly distributed random variables. Then the *mutual information* $I(\mathsf{X};\mathsf{Y})$ between X and Y is:

$$I(\mathsf{X};\mathsf{Y}) = H(\mathsf{X}) - H(\mathsf{X}|\mathsf{Y}) \tag{9.40}$$

From the definition, then the mutual information $I(\mathsf{X};\mathsf{Y})$ can be expressed as:

$$I(\mathsf{X};\mathsf{Y}) = \int \int p_{\mathsf{XY}}(x,y) \log \frac{p_{\mathsf{XY}}(x,y)}{p_{\mathsf{X}}(x)p_{\mathsf{Y}}(y)} dxdy \tag{9.41}$$

where $p_{\mathsf{XY}}(x,y)$ is the joint distribution of X and Y.

The following relationships from the discrete case hold for the continuous case as well:

$$I(\mathsf{X};\mathsf{Y}) = H(\mathsf{X}) - H(\mathsf{X}|\mathsf{Y}) \tag{9.42}$$

$$I(\mathsf{X};\mathsf{Y}) = H(\mathsf{Y}) - H(\mathsf{Y}|\mathsf{X}) \tag{9.43}$$

$$I(\mathsf{X};\mathsf{Y}) = H(\mathsf{X}) + H(\mathsf{Y}) - H(\mathsf{X},\mathsf{Y}) \text{ and} \tag{9.44}$$

$$I(\mathsf{X};\mathsf{Y}) = D\big(p_{\mathsf{XY}}(x,y)||p_{\mathsf{X}}(x)p_{\mathsf{Y}}(y)\big) \tag{9.45}$$

Proposition 9.8. $I(\mathsf{X};\mathsf{Y}) \geq 0$ with equality if and only if X and Y are independent.

SSQ 9.3. Are the following statements true for discrete random variables? For continuous random variables?

1. $I(\mathsf{X};\mathsf{Y}) \leq 0$ is possible.
2. $H(\mathsf{X}) \leq 0$ is possible.
3. For a constant c, $H(c\mathsf{X}) = H(\mathsf{X})$.

9.2 Differential Entropy of Gaussians

Because of the importance random variables with a Gaussian distribution, we study the differential entropy of such Gaussians, particularly multivariate Gaussians.

Discrete versus continuous Below the definitions of discrete and continuous entropy, mutual information and KL divergence are compared.

	discrete	continuous
$H(\mathsf{X})$	$-\sum_{x\in\mathcal{X}} p_{\mathsf{X}}(x)\log p_{\mathsf{X}}(x)$	$-\int_{\mathcal{X}} p_{\mathsf{X}}(x)\log p_{\mathsf{X}}(x)dx$
$H(\mathsf{X}\|\mathsf{Y})$	$-\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p_{\mathsf{XY}}(x,y)\log p_{\mathsf{X}\|\mathsf{Y}}(x\|y)$	$-\int_{\mathcal{X}}\int_{\mathcal{Y}} p_{\mathsf{XY}}(x,y)\log p_{\mathsf{X}\|\mathsf{Y}}(x\|y)dydx$
$D\big(f(x)\|\|g(x)\big)$	$\sum_{x\in\mathcal{X}} f(x)\log\frac{f(x)}{g(x)}$	$\int_{\mathcal{X}} f(x)\log\frac{f(x)}{g(x)}dx$
$I(\mathsf{X};\mathsf{Y})$	$\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p_{\mathsf{XY}}(x,y)\log\frac{p_{\mathsf{XY}}(x,y)}{p_{\mathsf{X}}(x)p_{\mathsf{Y}}(y)}$	$\int_{\mathcal{X}}\int_{\mathcal{Y}} p_{\mathsf{XY}}(x,y)\log\frac{p_{\mathsf{XY}}(x,y)}{p_{\mathsf{X}}(x)p_{\mathsf{Y}}(y)}dxdy$

Below, some properties are compared.

	discrete	continuous
Non-negativity of $H(\mathsf{X})$?	Yes: $H(\mathsf{X})\geq 0$	No: $H(\mathsf{X})<0$ possible
Maximum entropy	$H(\mathsf{X})\leq\log\|\mathcal{X}\|$	$H(\mathsf{X})\leq\log 2\pi e\sigma^2$
Entropy maximizing $p_{\mathsf{X}}(x)$	uniform $p_{\mathsf{X}}(x)=\frac{1}{\|\mathcal{X}\|}$	Gaussian $\frac{1}{\sqrt{2\pi\sigma^2}}\exp(-\frac{(x-m)^2}{2\sigma^2})$
Shift by constant c	$H(\mathsf{X}+c)=H(\mathsf{X})$	
Multiply by constant a	$H(a\mathsf{X})=H(\mathsf{X})$	$H(a\mathsf{X})=H(\mathsf{X})+\log\|a\|$
Cond. reduces entropy	$H(\mathsf{X}\|\mathsf{Y})\leq H(\mathsf{X})$	
Chain rule	$H(\mathsf{X},\mathsf{Y})=H(\mathsf{X}\|\mathsf{Y})+H(\mathsf{Y})$	
Mutual Information	$I(\mathsf{X};\mathsf{Y})=D\big(p_{\mathsf{XY}}(x,y)\|\|p_{\mathsf{X}}(x)p_{\mathsf{Y}}(y)\big)$	
Non-negativity	$D\big(p_{\mathsf{XY}}(x,y)\|\|p_{\mathsf{X}}(x)p_{\mathsf{Y}}(y)\big)\geq 0\Rightarrow I(\mathsf{X};\mathsf{Y})\geq 0$	

9.2.1 Single Gaussians

Proposition 9.9. Among random variables with mean 0 and variance σ^2, the Gaussian random variable $\mathcal{N}(0, \sigma^2)$ maximizes differential entropy.

Proof Let $\mathsf{Z} \sim \phi(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp(-\frac{x^2}{2\sigma^2})$ and let $\mathsf{X} \sim p_{\mathsf{X}}(x)$ be any zero-mean random variable with the same variance, that is, $\int x^2 \phi(x) dx = \int x^2 p_{\mathsf{X}}(x) dx = \sigma^2$. Both Z and X have zero mean. Then the KL divergence $D(p_{\mathsf{X}}(x)||\phi(x))$ is written as:

$$
\begin{aligned}
D\big(p_{\mathsf{X}}(x)||\phi(x)\big) &= \int_{\mathcal{X}} p_{\mathsf{X}}(x) \ln \frac{p_{\mathsf{X}}(x)}{\phi(x)} && (9.46)\\
&= \int_{\mathcal{X}} p_{\mathsf{X}}(x) \ln p_{\mathsf{X}}(x) dx - \int_{\mathcal{X}} p_{\mathsf{X}}(x) \ln \phi(x) dx && (9.47)\\
&= \int_{\mathcal{X}} p_{\mathsf{X}}(x) \ln p_{\mathsf{X}}(x) dx - \int_{\mathcal{X}} \phi(x) \ln \phi(x) dx && (9.48)\\
&= -H(\mathsf{X}) + H(\mathsf{Z}) && (9.49)
\end{aligned}
$$

where the next-to-last line holds because:

$$
\begin{aligned}
\int_{\mathcal{X}} p_{\mathsf{X}}(x) \ln \phi(x) dx &= \int_{\mathcal{X}} \phi(x) \ln \phi(x) dx && (9.50)\\
\int_{\mathcal{X}} p_{\mathsf{X}}(x) \Big(\frac{x^2}{2\sigma^2} + \ln \sqrt{2\pi\sigma^2} \Big) dx &= \int_{\mathcal{X}} \phi(x) \Big(\frac{x^2}{2\sigma^2} + \ln \sqrt{2\pi\sigma^2} \Big) dx && (9.51)
\end{aligned}
$$

can be seen equal, due to the equal variance assumption. Hence, by the non-negativity of KL divergence:

$$
0 \leq D\big(p_{\mathsf{X}}(x)||\phi(x)\big) = -H(\mathsf{X}) + H(\mathsf{Z}) \tag{9.52}
$$

that is $H(\mathsf{X}) \leq H(\mathsf{Z})$. The Gaussian distribution with variance σ^2 maximizes differential entropy among all random variables with variance σ^2.

Example 9.5. The binary-input additive white Gaussian noise (AWGN) channel has input $\mathsf{X} \in \{+1, -1\}$ with uniform input distribution $p_{\mathsf{X}}(x) = [\frac{1}{2}, \frac{1}{2}]$, AWGN Z distributed as $\mathcal{N}(0, \sigma^2)$ and channel output is Y:

$$
\mathsf{Y} = \mathsf{Z} + \mathsf{X}. \tag{9.53}
$$

Find $I(\mathsf{X}; \mathsf{Y})$ for the binary-input AWGN channel.

Due to the noise, $p_{\mathsf{Y}|\mathsf{X}}(y|x)$ is a Gaussian, with mean of -1 or $+1$:

$$
p_{\mathsf{Y}|\mathsf{X}}(y|x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(y-x)^2/2\sigma^2} \text{ for } x \in \{-1, +1\} \tag{9.54}
$$

Noting that X is discrete and Y is continuous, compute $I(\mathsf{X};\mathsf{Y})$ directly as

$$I(\mathsf{X};\mathsf{Y}) = \sum_{x\in\{\pm 1\}} \int_{-\infty}^{\infty} p_{\mathsf{X}}(x) p_{\mathsf{Y}|\mathsf{X}}(y|x) \log_2 \frac{p_{\mathsf{Y}|\mathsf{X}}(y|x)}{p_{\mathsf{Y}}(y)} dy \tag{9.55}$$

$$= \sum_{x\in\{\pm 1\}} \int_{-\infty}^{\infty} p_{\mathsf{X}}(x) p_{\mathsf{Y}|\mathsf{X}}(y|x) \log_2 \frac{p_{\mathsf{Y}|\mathsf{X}}(y|x)}{\sum_{x'} p_{\mathsf{X}}(x') p_{\mathsf{Y}|\mathsf{X}}(y|x')} dy \tag{9.56}$$

$$= \sum_{x\in\{\pm 1\}} \int_{-\infty}^{\infty} p_{\mathsf{X}}(x) p_{\mathsf{Y}|\mathsf{X}}(y|x) \log_2 \frac{2p_{\mathsf{Y}|\mathsf{X}}(y|x)}{p_{\mathsf{Y}|\mathsf{X}}(y|-1) + p_{\mathsf{Y}|\mathsf{X}}(y|+1)} dy \tag{9.57}$$

$$= \frac{1}{2} \sum_{x\in\{\pm 1\}} \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(y-x)^2/2\sigma^2} \log_2 \frac{2e^{-(y-x)^2/2\sigma^2}}{e^{-(y+1)^2/2\sigma^2} + e^{-(y-1)^2/2\sigma^2}} dy \tag{9.58}$$

$$= \frac{1}{2\sqrt{2\pi\sigma^2}} \sum_{x\in\{\pm 1\}} \int_{-\infty}^{\infty} e^{-(y-x)^2/2\sigma^2} \log_2 \frac{2e^{yx/\sigma^2}}{e^{y/\sigma^2} + e^{-y/\sigma^2}} dy \tag{9.59}$$

$$= \frac{1}{\sqrt{2\pi\sigma^2}} \int_{-\infty}^{\infty} e^{-(y-1)^2/2\sigma^2} \log_2 \frac{2e^{yx/\sigma^2}}{e^{y/\sigma^2} + e^{-y/\sigma^2}} dy \tag{9.60}$$

$$= \frac{1}{\sqrt{2\pi\sigma^2}} \int_{-\infty}^{\infty} e^{-(y-1)^2/2\sigma^2} \log_2 \frac{2}{1 + e^{-2y/\sigma^2}} dy \tag{9.61}$$

Above, $x^2 = 1$ with $x = \pm 1$, was used. Also, the value of the integral is independent of x, so $x = 1$ was chosen arbitrarily. This can not be found in closed form, and must be evaluated numerically. For example, if Var $= 0.9578$, $I(\mathsf{X};\mathsf{Y}) = 0.5$.

9.2.2 Multivariate Gaussian

The *multivariate Gausssian distribution*[2] of a n-dimensional random vector

$$\mathbf{X} = \mathsf{X}_1, \mathsf{X}_2, \ldots, \mathsf{X}_n$$

is:

$$p_{\mathbf{X}}(\mathbf{x}) = \frac{1}{(2\pi)^{\frac{n}{2}} |\det \mathbf{K}|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\mathbf{x}-\mathbf{m})^t \mathbf{K}^{-1} (\mathbf{x}-\mathbf{m})\right)$$

with mean vector $\mathbf{m}$:

$$\mathbf{m} \quad = \quad E[\mathsf{X}_1], \ldots, E[\mathsf{X}_n]$$

and covariance matrix $\mathbf{K}$:

$$\mathbf{K} = E[(\mathbf{X}-\mathbf{m})^t (\mathbf{X}-\mathbf{m})] \tag{9.62}$$

which has value:

$$k_{i,j} = \mathrm{Cov}(\mathsf{X}_i, \mathsf{X}_j) = E[(\mathsf{X}_i - m_i)^t (\mathsf{X}_j - m_j)] \tag{9.63}$$

[2]The Gaussian distribution is also called the normal distribution.

in row i, column j. We sometimes write $\mathbf{X} \sim \mathcal{N}(\mathbf{m}, \mathbf{K})$. When $n = 1$ the multivariate distribution reduces the single-variable Gaussian distribution (9.7). Wikipedia: Multivariate Gaussian distribution

If $\mathsf{X} \sim \mathcal{N}(m_1, v_1)$ and $\mathsf{Y} \sim \mathcal{N}(m_2, v_2)$ are independent Gaussian random variables, then $\mathsf{X}+\mathsf{Y} \sim \mathcal{N}(m_1+m_1, v_1+v_2)$. That is, the sum of two independent Gaussian random variables is also Gaussian, where the means and variances add. Even if X and Y are not independent, $\mathsf{X}+\mathsf{Y}$ is still Gaussian. Wikipedia: Sum of normally distributed random variables

9.2.3 Entropy of Multivariate Gaussian Distribution

Proposition 9.10. Let $\mathsf{X}_1, \ldots, \mathsf{X}_n \sim \mathcal{N}(\mathbf{m}, \mathbf{K})$ have a multivariate Gaussian distribution. Then:

$$H(\mathsf{X}_1, \ldots, \mathsf{X}_n) = \frac{1}{2} \log(2\pi e)^n |\det(\mathbf{K})|, \tag{9.64}$$

where det denotes determinant.[3]

The following states that a Gaussian distribution maximize entropy, among all distributions with a covariance matrix equal to $\det \mathbf{K}$.

Proposition 9.11. Let the random vector $\mathbf{X}$ have zero mean and covariance $K = E[\mathbf{X}\mathbf{X}^t]$ (or $K(i,j) = E[\mathsf{X}_i, \mathsf{X}_j]$). Then:

$$H(\mathbf{X}) \leq \frac{1}{2} \log(2\pi e)^n |\det \mathbf{K}| \tag{9.65}$$

with equality if and only if $\mathbf{X} \sim \mathcal{N}(0, \mathbf{K})$.

Example 9.6. Let $(\mathsf{X}, \mathsf{Y}) \sim \mathcal{N}(0, \mathbf{K})$ where:

$$\mathbf{K} = \begin{bmatrix} \sigma^2 & \rho\sigma^2 \\ \rho\sigma^2 & \sigma^2 \end{bmatrix}$$

Find $h(\mathsf{X}), h(\mathsf{Y}), h(\mathsf{X}, \mathsf{Y})$ and $I(\mathsf{X}; \mathsf{Y})$.

X and Y are Gaussians. From the covariance matrix $\mathbf{K}$, variance of X and Y are both σ^2, so:

$$H(\mathsf{X}) = H(\mathsf{Y}) = \frac{1}{2} \log 2\pi e\sigma^2. \tag{9.66}$$

The determinant $\det \mathbf{K} = \sigma^4 - \rho^2\sigma^4$, so:

$$H(\mathsf{X}, \mathsf{Y}) = \frac{1}{2} \log(2\pi e)^2 \sigma^4 (1 - \rho^2) \tag{9.67}$$

We can write:

$$I(\mathsf{X}; \mathsf{Y}) = H(\mathsf{X}) + H(\mathsf{Y}) - H(\mathsf{X}, \mathsf{Y}), \tag{9.68}$$

[3] Both Cover and Thomas and Wikipedia write $|\mathbf{K}|$ to mean $|\det \mathbf{K}|$.

Thus:

$$I(\mathsf{X};\mathsf{Y}) = \log 2\pi e\sigma^2 - \frac{1}{2}\log(2\pi e)^2\sigma^4(1-\rho^2) \tag{9.69}$$

$$= \log 2\pi e\sigma^2 - \log(2\pi e)\sigma^2 - \frac{1}{2}\log(1-\rho^2) \tag{9.70}$$

$$= -\frac{1}{2}\log(1-\rho^2) \tag{9.71}$$

If $\rho = 0$, then X and Y are independent and $I(\mathsf{X};\mathsf{Y}) = 0$. If $\rho = \pm 1$, then X and Y are completely correlated and $I(\mathsf{X};\mathsf{Y})$ is infinite.

9.3 Coding for the Gaussian Channel

The Gaussian channel model is a good model of various communications systems.

9.3.1 Gaussian Channel Model

The *additive white Gaussian channel* (AWGN) channel model has n inputs $x_1, x_2, \ldots, x_n$ satisfying a power constraint P:

$$\frac{1}{n}\sum_{i=1}^{n} x_i^2 \leq P. \tag{9.72}$$

The x_i are real numbers. The noise sequence $z_1, z_2, \ldots, z_n$ is independent and identically distributed Gaussians,

$$\mathbf{z}_i \sim \mathcal{N}(0, N), \tag{9.73}$$

with mean 0 and variance N. The channel output is y_i:

$$y_i = x_i + z_i, \tag{9.74}$$

for $i = 1, 2, \ldots, n$.

The *signal-to-noise ratio* or SNR is:

$$\text{SNR} = \frac{P}{N} \tag{9.75}$$

Without the power constraint (9.72), we could obtain a zero error rate by choosing codewords to be infinitely far apart. Real systems of course have power constraints.

Note that power of a signal is the square of its value. Thus, the noise power is

$$N = \int x^2 p_{\mathsf{Z}}(z)dz = E[\mathsf{Z}^2]. \tag{9.76}$$

The input X_i is assumed to be zero mean $E[\mathsf{X}] = 0$, so Y is also zero mean: $E\mathsf{Y}_i] = E[\mathsf{X}_i + \mathsf{Z}_i] = E[\mathsf{X}_i] + E[\mathsf{Z}_i] = 0$.

Gaussian channel model is shown in Fig. 9.1.

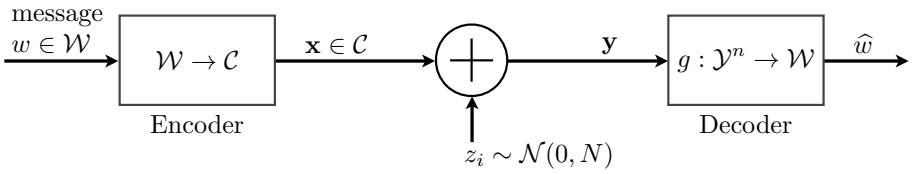


Figure 9.1: Communication system model consisting of an encoder, channel and decoder for the AWGN channel.

9.3.2 Gaussian Channel Code

Definition 9.7. An (M, n) *Gaussian channel code* $\mathcal{C}$ with power constraint P consists of a message index set $\{1, 2, \ldots, M\}$, and codebook

$$\mathcal{C} = \{\mathbf{x}(1), \mathbf{x}(2), \ldots, \mathbf{x}(M)\} \tag{9.77}$$

each with n elements: $\mathbf{x} = (x_1, x_2, \ldots, x_n)$, with each $x_i \in \mathbb{R}$ a real number, where each codeword satisfies the power constraint P:

$$\frac{1}{n}\sum_{i=1}^{n} x_i^2(w) \leq P. \tag{9.78}$$

An encoding function maps messages to codewords:

$$\{1, \ldots, M\} \to \mathcal{C} \tag{9.79}$$

The code rate R is:

$$R = \frac{1}{n}\log M \tag{9.80}$$

A codeword $\mathbf{x}$ its transmitted over the channel and received as $\mathbf{y} \in \mathcal{Y}^n$. The decoding function g is a mapping:

$$g : \mathcal{Y}^n \to \{1, 2, \ldots, M\} \tag{9.81}$$

Define probabilities of decoding error:

$$\lambda_i = \Pr(g(\mathbf{Y}) \neq w | \mathsf{X} = \mathbf{x}(w)) \tag{9.82}$$

$$\lambda = \max_i \lambda_i \tag{9.83}$$

$$P_e = \frac{1}{2^{nR}}\sum \lambda_i. \tag{9.84}$$

Example 9.7. Consider the $n = 1$ code with $\mathcal{C} = \{-3, -1, +1, +3\}$. Find the rate R. What power constraint does this code satisfy?

The code rate R is

$$R = \frac{1}{n}\log M = \log 4 = 2 \text{ bits}\ .$$

All the codewords must satisfy the power constraint. The power for codewords $(-3, -1, 1, 3)$ are $(9, 1, 1, 9)$ respectively. Since 9 is the greatest, the code satisfies any power constraint $P \geq 9$.

Example 9.8. Consider the code with $\mathcal{C} = \{-1, +1\}$. This is transmitted over an AWGN channel with noise power N. The decoder decides -1 was transmitted if the received symbol y is less than 0, and decides $+1$ was transmitted if the received symbol $y > 0$. What is the probability of decoder error?

Assume -1 was transmitted. An error occurs if the noise $z > 1$:

$$\lambda_{-1} = \Pr[\widehat{x} = +1 | x = -1] = \Pr[y > 0 | x = -1] \tag{9.85}$$
$$= \Pr[z > 1] \tag{9.86}$$
$$= \frac{1}{\sqrt{2\pi N}} \int_1^\infty e^{-\frac{z^2}{2N}} dz \tag{9.87}$$
$$= \frac{1}{2}\text{erfc}(\frac{1}{\sqrt{2N}}), \tag{9.88}$$

where erfc is the complementary error function.[4] By symmetry, we have:

$$P_e = \frac{1}{2}\text{erfc}(\frac{1}{\sqrt{2N}}) \tag{9.89}$$

For small noise like $N = 0.1$, $P_e = 7.810^{-4}$. For large noise like $N = 1$, $P_e = 0.1587$

9.4 Capacity of the Gaussian Channel

Definition 9.8. A rate R is *achievable* if there exists a sequence of $(2^{nR}, n)$ codes with codewords satisfying the power constraint P, such that the maximal probability of error λ goes to zero. The *capacity* is the supremum of achievable rates.

Definition 9.9. The *information capacity* of the Gaussian channel with transmit power constraint P is:

$$C = \max_{p_{\mathsf{X}}(x): E\mathsf{X}^2 \le P} I(\mathsf{X}; \mathsf{Y}), \tag{9.90}$$

where $p_{\mathsf{X}}(x)$ is the input distribution.

Proposition 9.12. *Gaussian Channel Coding* The capacity (the supremum of achievable rates) of a Gaussian channel with power constraint P and noise variance N is:

$$C = \frac{1}{2}\log\left(1 + \frac{P}{N}\right) \text{ bits per transmission} \tag{9.91}$$

[4] If X is a Gaussian random variable with mean m and variance v, then:

$$\Pr(\mathsf{X} > x) = \frac{1}{\sqrt{2\pi v}} \int_x^\infty e^{-\frac{(u-m)^2}{2v}} du = \frac{1}{2}\text{erfc}(\frac{x-m}{\sqrt{2v}})$$

where erfc is the complementary error function: $\text{erfc}(x) = \frac{2}{\sqrt{\pi}} \int_x^\infty e^{-u^2} du$. Many programming languages have functions which give numerical values for erfc. Wikipedia: Error function.

The capacity-achieving input distribution is $p^*_{\mathsf{X}}(x)$ is a zero-mean Gaussian with variance P

In other words, reliable communication is possible if and only if[5] code rates R satisfy $R < C$.

Now, an upper bound on $I(\mathsf{X};\mathsf{Y})$ is given, and we show that it is achieved with a Gaussian input distribution. By assumption, $E[\mathsf{Z}] = 0$ and the noise power is $E[\mathsf{Z}^2] = N$. The noise entropy is $H(\mathsf{Z}) = \frac{1}{2}\log 2\pi eN$; X and Z are independent so $E[\mathsf{XZ}] = 0$. The variance, or noise power, of Y is:

$$E[(\mathsf{Y} - E\mathsf{Y})^2] = E[\mathsf{Y}^2] = E[(\mathsf{X}+\mathsf{Z})^2] = E[\mathsf{X}^2] + 2E[\mathsf{XZ}] + E[\mathsf{Z}^2] = P + N, \tag{9.92}$$

since X, Z uncorrelated means $E[\mathsf{XZ}] = 0$.

Since $E[\mathsf{Y}^2] = P + N$, by Proposition 9.9.

$$H(\mathsf{Y}) \leq \frac{1}{2}\log 2\pi e(P+N) \tag{9.93}$$

with equality if X is Gaussian. If X is Gaussian, then Y will be Gaussian (since the sum of two Gaussians is a Gaussian).

We can write:

$$\begin{aligned} I(\mathsf{X};\mathsf{Y}) &= H(\mathsf{Y}) - H(\mathsf{Y}|\mathsf{X}) \\ &= H(\mathsf{Y}) - H(\mathsf{X}+\mathsf{Z}|\mathsf{X}) \\ &= H(\mathsf{Y}) - H(\mathsf{Z}|\mathsf{X}) \\ &= H(\mathsf{Y}) - H(\mathsf{Z}) \\ &\leq \frac{1}{2}\log 2\pi e(P+N) - \frac{1}{2}\log 2\pi eN \\ &= \frac{1}{2}\log(1+\frac{P}{N}) \end{aligned}$$

The capacity of the Gaussian channel is upper bounded by:

$$C = \max_{p_{\mathsf{X}}(x):E\mathsf{X}^2\leq P} I(\mathsf{X};\mathsf{Y}) \leq \frac{1}{2}\log\big(1+\frac{P}{N}\big). \tag{9.94}$$

This is obtained with equality when $\mathsf{X} \sim \mathcal{N}(0,P)$, since a Gaussian distribution maximizes entropy.

9.5 Parallel Gaussian Channels

Consider k parallel and independent Gaussian channels, modeled as:

$$\mathsf{Y}_j \quad = \quad \mathsf{X}_j + \mathsf{Z}_j \quad \text{for } j = 1,\ldots,k. \tag{9.95}$$

[5] see Cover and Thomas, Sec. 9.1 and 9.2 for proofs.

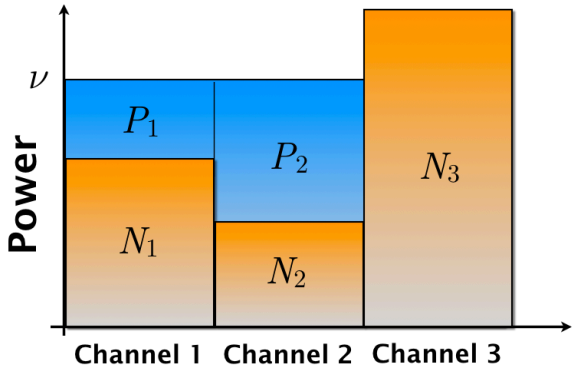


Figure 9.2: Waterfilling for three parallel channels

Each channel j has a transmit power constraint $P_j = E[\mathsf{X}_j^2]$ and noise power N_j, so that $\mathsf{Z}_j \sim \mathcal{N}(0, N_j)$. In addition, there is a global constraint on the total power:

$$E\mathsf{X}_1^2 + E\mathsf{X}_2^2 + \cdots E\mathsf{X}_j^2 = \sum_i P_i \leq P \tag{9.96}$$

Definition 9.10. The information capacity of the parallel Gaussian channel is:

$$C = \max_{p_{\mathbf{X}}(\mathbf{x}):\sum E\mathsf{X}_i^2 \leq P} I(\mathsf{X}_1, \ldots, \mathsf{X}_k; \mathsf{Y}_1, \ldots, \mathsf{Y}_k) \tag{9.97}$$

where $p_{\mathbf{X}}(\mathbf{x}) = p_{\mathsf{X}_1 \cdots \mathsf{X}_k}(x_1, \ldots, x_k)$ is the joint input distribution.

The goal is to allocate the transmit power among the channels to maximize the parallel channels' information capacity. The optimal allocation for channel j is denoted P_j^*, and clearly $P_j^* \leq P_j$ and $\sum_j P_j^* \leq P$ must hold. Define the $(\cdot)^+$ function:

$$(x)^+ = \max(0, x). \tag{9.98}$$

Proposition 9.13. *Waterfilling parallel Gaussian channels* For parallel Gaussian channels, the information capacity is achieved by $P_1^*, \ldots, P_k^*$ satisfying

$$P_i^* = (\nu - N_i)^+ \tag{9.99}$$

and $\sum_i P_i^* = P$

The waterfilling idea is expressed for three parallel channels in Fig. 9.2. Channels 1 and 2 have low noise power. Power P_1 and P_2 is allocated to these channels such that $P_1 + N_1 = P_2 + N_2$ — this is waterfilling, since we add power to each channel to bring the sum to the same level. However, Channel 3 has a large amount of noise and is not used at all.

Proof To find the optimal power allocation $P_1^*, \ldots, P_k^*$; Lagrange multipliers will be used. Since Z_j are independent:

$$\begin{aligned} I(\mathsf{X}_1, \ldots, \mathsf{X}_k; \mathsf{Y}_1, \ldots, \mathsf{Y}_k) &= H(\mathsf{Y}_1, \ldots, \mathsf{Y}_k) - H(\mathsf{Y}_1, \ldots, \mathsf{Y}_k | \mathsf{X}_1, \ldots, \mathsf{X}_k) \\ &= H(\mathsf{Y}_1, \ldots, \mathsf{Y}_k) - H(\mathsf{Z}_1, \ldots, \mathsf{Z}_k | \mathsf{X}_1, \ldots, \mathsf{X}_k) \\ &= H(\mathsf{Y}_1, \ldots, \mathsf{Y}_k) - H(\mathsf{Z}_1, \ldots, \mathsf{Z}_k) \\ &= H(\mathsf{Y}_1, \ldots, \mathsf{Y}_k) - \sum_i H(\mathsf{Z}_i) \qquad \text{independence of } \mathsf{Z}_i \\ &\leq \sum_i H(\mathsf{Y}_i) - \sum_i H(\mathsf{Z}_i) \\ &\leq \sum_i \frac{1}{2} \log\left(1 + \frac{P_i}{N_i}\right) \end{aligned}$$

Equality is achieved by:

$$(\mathsf{X}_1, \ldots, \mathsf{X}_k) \sim \mathcal{N}(0, \begin{bmatrix} P_1 & 0 & \cdots & 0 \\ 0 & P_2 & \cdots & 0 \\ & & \vdots & \\ 0 & 0 & \cdots & P_k \end{bmatrix}) \tag{9.100}$$

Achieve the goal by solving an optimization problem:

$$\max_{P_1, \ldots, P_k} \left(\frac{1}{2} \log(1 + \frac{P_1}{N_1}) + \cdots + \frac{1}{2} \log(1 + \frac{P_k}{N_k}) \right) \quad \text{such that} \quad \sum P_i = P$$

Using Lagrange multipliers, the Largrangian J is:

$$J(P_1, \ldots, P_k) = \sum_j \frac{1}{2} \log(1 + \frac{P_j}{N_j}) + \lambda \sum_j P_j \tag{9.101}$$

and λ is the Lagrange multiplier. For $i = 1, \ldots, k$:

$$\frac{\partial}{\partial P_i} J(P_1, \ldots, P_k) = \frac{1}{2} \frac{1}{P_i + N_i} + \lambda = 0 \tag{9.102}$$

$$P_i = \underbrace{\left(-\frac{1}{2\lambda}\right)}_{\stackrel{\text{def}}{=} \nu} - N_i \tag{9.103}$$

$$P_i = \nu - N_i \tag{9.104}$$

Alternatively, note that $\lambda = -\frac{1}{2} \frac{1}{P_i + N_i}$ so another solution is:

$$-\frac{1}{2} \frac{1}{P_1 + N_1} = -\frac{1}{2} \frac{1}{P_2 + N_2} = \cdots = -\frac{1}{2} \frac{1}{P_k + N_k}, \tag{9.105}$$

that is, $P_i + N_i$ should be a constant, defined to be ν.

But, P_i must be non-negative. Let P_i^* be the optimal input powers. Use the Kuhn-Tucker conditions to verify the solution is:

$$P_i^* = (\nu - N_i)^+ . \quad \square \tag{9.106}$$

9.6 Source Code

This Matlab source code numerically computes an integral to find $I(\mathsf{X};\mathsf{Y})$ for the binary-input AWGN channel with variance `var`, as given by (9.61).

```
>> var = 0.9578;
>> fun = @(y) (1/sqrt(2*pi*var)) .* exp( - (y-1).^2 / (2*var) ) .*
    log2( 2  ./ ( 1 + exp( - 2*y / var))  );
>> IXY = integral(fun,-10,10)

IXY =

    0.5000
```

9.7 Exercises

9.1 Let X be a Gaussian-distributed random variable with mean 0 and variance v. Show that $E[\mathsf{X}] = 0$ and $\mathrm{Var}[X] = v$. Hint: $\int xe^{cx}\,\mathrm{d}x = \frac{e^{cx}}{c^2}(cx-1)$ and $\int_{-\infty}^{\infty} x^2 e^{-ax^2}\,\mathrm{d}x = \frac{1}{2}\sqrt{\frac{\pi}{a^3}} \quad (a>0)$.

9.2 *Packing spheres*

(a) In two dimensions, consider packing circles of radius $N = 1$ inside a large circle of radius $S = 10$. Find an upper bound on M, the number of circles, by dividing the area of the large circle, by the area of small circle.

(b) Now consider n-dimensional spheres. The volume of a sphere with radius r in n dimensions is $\mathrm{Vol}(n, r)$:

$$\mathrm{Vol}(n,r) = \frac{\pi^{n/2}}{\Gamma(\frac{n}{2}+1)} r^n, \tag{9.107}$$

where Γ is the gamma function. Let the radius of the large sphere be $\sqrt{P+N}$, and let the radius of the small sphere be $\sqrt{N}$. As before, find an upper bound on the number of spheres M.

(c) Now let $R = \frac{1}{n}\log M$. Using your answer to part (b), what is an upper bound on R?

9.3 Let $\mathsf{X}_1, \mathsf{X}_2$ be jointly distribution zero-mean Gaussian random variables with covariance matrix

$$\mathbf{K} = \begin{bmatrix} N & \rho N \\ \rho N & N \end{bmatrix} \tag{9.116}$$

What is the variance of $\mathsf{X} = \mathsf{X}_1 + \mathsf{X}_2$? What is the differential entropy $H(\mathsf{X})$?

9.4 Let X be distributed as the Dirac delta function $\delta(x)$ where

$$\delta(x) = \begin{cases} 0 & x \neq 0 \\ \text{undefined} & x = 0 \end{cases} \tag{9.121}$$

and $\int_{-\infty}^{\infty} \delta(x)dx = 1$. Show $H(\mathsf{X})$ is unbounded.

9.5 *Triangular probability distribution* Consider two uniformly distributed random variables X and Y:

$$p_{\mathsf{X}}(x) = \begin{cases} \frac{1}{2} & -1 \le x \le 1 \\ 0 & \text{otherwise} \end{cases} \quad \text{and} \quad p_{\mathsf{Y}}(y) = \begin{cases} \frac{1}{2} & -1 \le y \le 1 \\ 0 & \text{otherwise} \end{cases} \tag{9.123}$$

(a) Find the probability distribution of $\mathsf{X} + \mathsf{Y}$, using convolution.

(b) Let Z be distributed as a triangular distribution:

$$p_{\mathsf{Z}}(z) = \begin{cases} c + c^2 z & -\frac{1}{c} \le z \le 0 \\ c - c^2 z & 0 < z \le \frac{1}{c} \\ 0 & \text{otherwise} \end{cases} \tag{9.128}$$

for $c > 0$. Find differential entropy of Z in nats, using natural log ln.

(c) What is the differential entropy of $\mathsf{X} + \mathsf{Y}$, in nats?

9.6 *Discrete input, uniform output channel* Consider a binary-input, continuous output channel. Let $p_{\mathsf{X}}(0) = p, p_{\mathsf{X}}(1) = 1 - p$. and let $\mathsf{Y} = \mathsf{X} + \mathsf{Z}$, where Z is uniform over the interval $[0, a]$, $a > 1$, and Z is independent of X.

(a) Calculate $I(\mathsf{X};\mathsf{Y}) = H(\mathsf{X}) - H(\mathsf{X}|\mathsf{Y})$, using discrete entropies.

(b) Now calculate $I(\mathsf{X};Y)$ the other way $H(\mathsf{Y}) - H(\mathsf{Y}|\mathsf{X})$, using differential entropies.

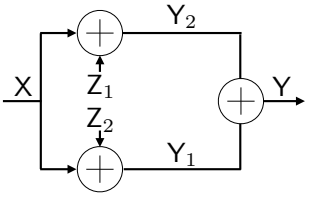
(c) Calculate the capacity of this channel by maximizing over p.

9.7 *Capacity of the binary-input AWGN channel* The binary-input AWGN channel achieves capacity with $p^*_{\mathsf{X}}(x) = [\frac{1}{2}, \frac{1}{2}]$. The signal-to-noise ratio SNR in decibels (dB) is:

$$\text{SNR in dB} = 10 \log_{10} \frac{1}{\sigma^2}. \tag{9.146}$$

(a) Make a plot of capacity C versus SNR dB for the binary-input AWGN channel. (b) On the same graph, plot the capacity of the AWGN channel with Gaussian input distribution. (c) What happens to achievable rates as SNR $\to 0$? What happens to achievable rates as SNR $\to \infty$? What can you conclude from this?

9.8 *Multipath Gaussian Channel* Consider a Gaussian-noise channel where the signal X with power constraint P takes two paths to the receiver, and each path experiences noise Z_i:



(a) Find the capacity of this channel when Z_1 and Z_2 are jointly-distributed Gaussian random variables with covariance matrix:

$$\mathbf{K} = \begin{bmatrix} N & \rho N \\ \rho N & N \end{bmatrix} \tag{9.147}$$

(b) What is the capacity for $\rho = 0$, $\rho = 1$ and $\rho = -1$?
(c) For SNR $= 4$, make a plot of capacity vs. ρ.

9.9 *Quantization that maximizes mutual information* An arbitrary, continuous random variable X has probability distribution $f_{\mathsf{X}}(x)$ and cumulative distribution $F(x)$:

$$F(x) = \int_{-\infty}^{x} p_{\mathsf{X}}(u)du.$$

Let $F^{-1}(y) = x$ be the inverse function of $F(x) = y$.

Consider quantization of continuous X to a discrete $\widehat{\mathsf{X}}$ with M values $\{1, 2, \ldots, M\}$. Let $a_1 < a_2 < \cdots < a_{M+1}$ be $M + 1$ quantization boundaries, and let $g(x)$ be the quantizing function:

$$g(x) = i \text{ if } a_i \leq x < a_{i+1}, \text{ for } i = 1, 2, \ldots, M$$

so that $\widehat{\mathsf{X}} = g(\mathsf{X})$. Assume that $a_1 \to -\infty$ so that $F(a_1) = 0$. Assume that $a_{M+1} \to \infty$ so that $F(a_{M+1}) = 1$. Then, the following theorem holds:

Theorem If the a_i are chosen such that:

$$a_{i+1} = F^{-1}\big(\frac{i}{M}\big), \text{ for } i = 1, 2, \ldots M - 1, \tag{A}$$

then mutual information $I(\mathsf{X}; \widehat{\mathsf{X}})$ is maximized, with maximization over $a_2, \ldots, a_M$.

Prove this theorem in 3 steps.

(a) Prove that if $\widehat{\mathsf{X}}$ is uniformly distributed, then $I(\mathsf{X}; \widehat{\mathsf{X}})$ is maximized.
(b) Prove that if $\widehat{\mathsf{X}}$ is uniformly distributed, then the a_i satisfy:

$$F(a_{i+1}) - F(a_i) = \frac{1}{M}, \text{ for } i = 1, 2, \ldots, M. \tag{B}$$

(c) Prove that (A) and (B) are equivalent.
(d) Let X be Gaussian distributed with mean 0, variance 1. Quantize X to $\widehat{\mathsf{X}}$ using $M = 4$ levels. Find numerical values of a_2, a_3, a_4 such that $I(\mathsf{X}; \widehat{\mathsf{X}})$ is maximized.

Chapter 10

Rate-Distortion Theory

Lossless compression was studied in Chapters 4–5, where it was always possible to reconstruct the original source from the compressed version. For compression of a random vector $\mathbf{X}$, the single-letter entropy $H(\mathsf{X})$ is a lower bound on the compression rate R, that is $H(\mathsf{X}) \leq R$.

The subject of this chapter is *lossy* compression. In lossy compression we can achieve rates less than $H(\mathsf{X})$. However, the reconstructed sequence (or "decompressed" sequence) is not the same as the original sequence. The original and reconstructed sequences should be as similar as possible, so a way of measuring similarity, called the distortion measure, is introduced. There is a tradeoff between rate and distortion, and this tradeoff is the subject of rate-distortion theory.

10.1 Rate-Distortion Code, Expected Distortion

10.1.1 Rate-Distortion Code

In source coding, the goal is to represent a source vector by a message with as few bits as possible. The source vector $\mathbf{x}$ is compressed by an encoder f, which produces one of M messages, from the set $\{1, 2, \ldots, M\}$. This is the compressed message which can be stored or transmitted. To recover, the message is decompressed to a vector $\widehat{\mathbf{x}}$. In *lossless* compression, $\widehat{\mathbf{x}}$ is identical to $\mathbf{x}$. But in *lossy* compression, $\widehat{\mathbf{x}}$ is similar to, but not always identical to, $\mathbf{x}$.

Lossy source coding is described Fig. 10.1, which shows a source $\mathbf{x}$ with elements from $\mathcal{X}^n$:

$$\mathbf{x} = (x_1, x_2, \ldots, x_n) \tag{10.1}$$

Additional reading: Cover and Thomas, Sections 10.1 to 10.3; MacKay *Information Theory, Inference and Learning Algorithms* Chapter 20.

$$\mathbf{X} \to \boxed{\text{encoder } f} \to f(\mathbf{X}) \to \boxed{\text{decoder } g} \to \widehat{\mathbf{X}} = g\big(f(\mathbf{X})\big)$$

Figure 10.1: Block diagram of source encoder and decoder.

A source encoder f maps $\mathbf{x}$ to a message m, so that $f(\mathbf{x}) = m$. A decoder g maps a codeword to a sequence $\widehat{\mathbf{x}}$ with elements from $\widehat{\mathcal{X}}^n$:

$$\widehat{\mathbf{x}} = (\widehat{x}_1, \widehat{x}_2, \ldots, \widehat{x}_n). \tag{10.2}$$

The set $\mathcal{X}$ is called the *source alphabet* and the set $\widehat{\mathcal{X}}$ is called the *reconstruction alphabet*. When $\mathcal{X}$ is discrete, we often use $\widehat{\mathcal{X}} = \mathcal{X}$.

A rate-distortion code uses a codebook $\mathcal{C}$ with codewords from $\mathcal{X}^n$. A codebook $\mathcal{C}$ has M codewords of n symbols each and the code rate R is:

$$R = \frac{1}{n} \log M \tag{10.3}$$

in bits per source symbol. The codewords correspond to one of $M = 2^{nR}$ messages.

Example 10.1. For a binary source with $n = 6$, consider a codebook $\mathcal{C}$:

message m	codebook $\mathcal{C}$
1	000000
2	000111
3	111000
4	111111

It can be seen that there are 4 codewords, requiring 2 bits of storage to represent the codeword. The code rate is $R = \frac{1}{6} \log 4 = \frac{1}{3}$.

The compression and decompression process can be described as follows. If $\mathbf{x} \in \mathcal{X}^n$ is a source sequence, then $f(\mathbf{x})$ is the message, from the set $\{1, 2, \ldots, M\}$. The message $m \in \{1, 2, \ldots, M\}$ requires $\frac{1}{n} \log M$ bits per source simple. Given m, the decompressed sequence (or reconstructed sequence) is $\widehat{\mathbf{x}} = g(m)$. This decompressed $\widehat{\mathbf{x}}$ is generally not the same as $\mathbf{x}$, but similar to it. In particular, the encoding function should choose the message $f(\mathbf{x})$ so to minimize the distortion of the reconstruction $g\big(f(\mathbf{x})\big)$.

The following is the formal definition of a rate-distortion code. This definition does not explicitly use $\mathcal{C}$, but it is implied by the f and g mappings.

Definition 10.1. A $(2^{nR}, n)$ *rate distortion code* consists of an encoding function f_n:

$$f_n : \mathcal{X}^n \to \{1, 2, \ldots, 2^{nR}\} \tag{10.4}$$

and a decoding function g_n:

$$g_n : \{1, 2, \ldots, 2^{nR}\} \to \widehat{\mathcal{X}}^n \tag{10.5}$$

10.1.2 Distortion Measure and Encoding

In lossless compression, the input is recovered exactly: $g(f(\mathbf{x})) = \mathbf{x}$. But in lossy compression, the decoder output $g(f(\mathbf{x})) = \widehat{\mathbf{x}}$ might be different, $\mathbf{x} \neq \widehat{\mathbf{x}}$.

We need a measurement of "different," and that measurement is called distortion. A distortion function is $d(\mathbf{x}, \widehat{\mathbf{x}})$ for sequences or $d(x_i, \widehat{x}_i)$ for elements of the sequence. We want $\mathbf{x}$ and $\widehat{\mathbf{x}}$ to be as close as possible, that is, the distortion should be as small as possible, and zero distortion $d(\mathbf{x}, \widehat{\mathbf{x}}) = 0$ means $\mathbf{x} = \widehat{\mathbf{x}}$.

For discrete random variables including binary random variables, an important distortion function is *Hamming distortion* :

$$d(\mathbf{x}, \widehat{\mathbf{x}}) = \frac{1}{n} \sum_{i=1}^{n} d(x_i, \widehat{x}_i), \tag{10.6}$$

where

$$d(x_i, \widehat{x}_i) = \begin{cases} 0 & \text{if } x_i = \widehat{x}_i \\ 1 & \text{if } x_i \neq \widehat{x}_i \end{cases}. \tag{10.7}$$

For discrete random variable sources, the distortion measure can be written as a $|\mathcal{X}| \times |\widehat{\mathcal{X}}|$ matrix. The binary Hamming distortion function given by (10.7) in matrix form is:

$$d(x, \widehat{x}) = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}. \tag{10.8}$$

where the rows correspond to x and columns correspond to $\widehat{x}$.

Example 10.2. Continuing Example 10.1, compress the sequence $\mathbf{x} = 111100$ to a message to minimize Hamming distortion.

First, compute the Hamming distortion between $\mathbf{x} = 111100$ and each codeword in $\mathcal{C}$:

$$\begin{aligned} d(111100, 000000) &= 4 \\ d(111100, 000111) &= 5 \\ d(111100, 111000) &= 1 \\ d(111100, 111111) &= 2. \end{aligned}$$

Clearly, 111000 is the codeword which has the smallest Hamming distortion, so the encoder choses $f(111100) = 3$. The source sequence $\mathbf{x}$ is compressed to the message $m = 3$. Thus a six-bit source sequence $\mathbf{x}$ can be represented by a two-bit message m.

In addition, the decompressed sequence is $g(3) = 111000$.

The example shows the *reconstruction process*. The source sequence $\mathbf{x}$ is compressed to a codeword $m \in \{1, 2, \ldots, 2^{nR}\}$. Then this is decompressed to a codeword from a codebook:

$$\mathbf{x} \xrightarrow{f} m \xrightarrow{g} \widehat{\mathbf{x}}. \tag{10.9}$$

For studying the rate-distortion tradeoff, we are interested in the mapping from $\mathbf{x}$ to $\widehat{\mathbf{x}}$:

$$\mathbf{x} \xrightarrow{g \circ f} \widehat{\mathbf{x}}. \tag{10.10}$$

For Example 10.2, this process is $1111000 \to 3 \to 111000$, but the intermediate step of finding m is omitted:

$$1111, \xrightarrow{g \circ f} 111000. \tag{10.11}$$

The reconstruction process shows how the codeword $\widehat{\mathbf{x}} = g\big(f(\mathbf{x})\big)$ is an approximation of source $\mathbf{x}$.

SSQ 10.1. For the source code in Example 10.1, which codeword has the lowest Hamming distortion with $\mathbf{x} = 100101$? With $\mathbf{x} = 111111$?

10.1.3 Expected Distortion

Expected distortion is the distortion averaged over all source sequences.

Definition 10.2. The *expected distortion* is for a $(2^{nR}, n)$ code is D:

$$D = E\Big[d\Big(\mathbf{X}, g\big(f(\mathbf{X})\big)\Big)\Big] \tag{10.12}$$

$$= \sum_{\mathbf{x} \in \mathcal{X}^n} p_{\mathbf{X}}(\mathbf{x}) d\Big(\mathbf{x}, g\big(f(\mathbf{x})\big)\Big) \tag{10.13}$$

Example 10.3. Continuing Example 10.2 with $n = 6$, assume the source is distributed as $p_{\mathsf{X}}(x) = [1-p, p]$ where $p = \frac{1}{2}$. What is the expected distortion?

First find $p_{\mathbf{X}}(\mathbf{x})$. If $\mathbf{x}$ has k ones, then:

$$p_{\mathbf{X}}(\mathbf{x}) = p^k (1-p)^{n-k} = \left(\frac{1}{2}\right)^k \left(\frac{1}{2}\right)^{n-k} = \left(\frac{1}{2}\right)^6 = \frac{1}{64}. \tag{10.14}$$

Table 10.1 lists all sequences, their corresponding codewords and distortion. There are 4 sequences with distortion 0, 24 sequences with distortion $\frac{1}{6}$ and 36 sequences with distortion $\frac{2}{6}$. Then, the expected distortion is

$$E[\mathsf{X}, g\big(f(\mathsf{X})\big)] = \sum_{x \in \mathcal{X}} p_{\mathsf{X}}(x) d(\mathbf{x}, g\big(f(\mathbf{x})\big)) \tag{10.15}$$

$$= 4\left(\frac{1}{64} \cdot 0\right) + 24\left(\frac{1}{64} \cdot \frac{1}{6}\right) + 36\left(\frac{1}{64} \cdot \frac{2}{6}\right) \tag{10.16}$$

$$= \frac{1}{4} \tag{10.17}$$

As shown in Example 10.1, this code has rate $R = \frac{1}{3}$. Thus, the example is a rate-distortion code with pair $(R, D) = (\frac{1}{3}, \frac{1}{4})$.

The following example illustrates a code with low rate, but high distortion

Table 10.1: All sequences $\mathcal{X}^6$, and the corresponding codeword $g\big(f(\mathbf{x})\big)$ using the example code.

Source $\mathbf{x}$	codeword $\widehat{\mathbf{x}} = g\big(f(\mathbf{x})\big)$	distortion $d(\mathbf{x}, \widehat{\mathbf{x}})$
0 0 0 0 0 0	0 0 0 0 0 0	0
1 0 0 0 0 0	0 0 0 0 0 0	1/6
0 1 0 0 0 0	0 0 0 0 0 0	1/6
1 1 0 0 0 0	1 1 1 0 0 0	1/6
0 0 1 0 0 0	0 0 0 0 0 0	1/6
1 0 1 0 0 0	1 1 1 0 0 0	1/6
0 1 1 0 0 0	1 1 1 0 0 0	1/6
1 1 1 0 0 0	1 1 1 0 0 0	0
0 0 0 1 0 0	0 0 0 0 0 0	1/6
1 0 0 1 0 0	0 0 0 0 0 0	2/6
0 1 0 1 0 0	0 0 0 0 0 0	2/6
1 1 0 1 0 0	1 1 1 0 0 0	2/6
0 0 1 1 0 0	0 0 0 0 0 0	2/6
1 0 1 1 0 0	1 1 1 0 0 0	2/6
0 1 1 1 0 0	1 1 1 0 0 0	2/6
1 1 1 1 0 0	1 1 1 0 0 0	1/6
0 0 0 0 1 0	0 0 0 0 0 0	1/6
1 0 0 0 1 0	0 0 0 0 0 0	2/6
0 1 0 0 1 0	0 0 0 0 0 0	2/6
1 1 0 0 1 0	1 1 1 0 0 0	2/6
0 0 1 0 1 0	0 0 0 0 0 0	2/6
1 0 1 0 1 0	1 1 1 0 0 0	2/6
0 1 1 0 1 0	1 1 1 0 0 0	2/6
1 1 1 0 1 0	1 1 1 0 0 0	1/6
0 0 0 1 1 0	0 0 0 1 1 1	1/6
1 0 0 1 1 0	0 0 0 1 1 1	2/6
0 1 0 1 1 0	0 0 0 1 1 1	2/6
1 1 0 1 1 0	1 1 1 1 1 1	2/6
0 0 1 1 1 0	0 0 0 1 1 1	2/6
1 0 1 1 1 0	1 1 1 1 1 1	2/6
0 1 1 1 1 0	1 1 1 1 1 1	2/6
1 1 1 1 1 0	1 1 1 1 1 1	1/6
0 0 0 0 0 1	0 0 0 0 0 0	1/6
1 0 0 0 0 1	0 0 0 0 0 0	2/6
0 1 0 0 0 1	0 0 0 0 0 0	2/6
1 1 0 0 0 1	1 1 1 0 0 0	2/6
0 0 1 0 0 1	0 0 0 0 0 0	2/6
1 0 1 0 0 1	1 1 1 0 0 0	2/6
0 1 1 0 0 1	1 1 1 0 0 0	2/6
1 1 1 0 0 1	1 1 1 0 0 0	1/6
0 0 0 1 0 1	0 0 0 1 1 1	1/6
1 0 0 1 0 1	0 0 0 1 1 1	2/6
0 1 0 1 0 1	0 0 0 1 1 1	2/6
1 1 0 1 0 1	1 1 1 1 1 1	2/6
0 0 1 1 0 1	0 0 0 1 1 1	2/6
1 0 1 1 0 1	1 1 1 1 1 1	2/6
0 1 1 1 0 1	1 1 1 1 1 1	2/6
1 1 1 1 0 1	1 1 1 1 1 1	1/6
0 0 0 0 1 1	0 0 0 1 1 1	1/6
1 0 0 0 1 1	0 0 0 1 1 1	2/6
0 1 0 0 1 1	0 0 0 1 1 1	2/6
1 1 0 0 1 1	1 1 1 1 1 1	2/6
0 0 1 0 1 1	0 0 0 1 1 1	2/6
1 0 1 0 1 1	1 1 1 1 1 1	2/6
0 1 1 0 1 1	1 1 1 1 1 1	2/6
1 1 1 0 1 1	1 1 1 1 1 1	1/6
0 0 0 1 1 1	0 0 0 1 1 1	0
1 0 0 1 1 1	0 0 0 1 1 1	1/6
0 1 0 1 1 1	0 0 0 1 1 1	1/6
1 1 0 1 1 1	1 1 1 1 1 1	1/6
0 0 1 1 1 1	0 0 0 1 1 1	1/6
1 0 1 1 1 1	1 1 1 1 1 1	1/6
0 1 1 1 1 1	1 1 1 1 1 1	1/6
1 1 1 1 1 1	1 1 1 1 1 1	0

Example 10.4. Consider the same source as Example 10.3, but the codebook only has one codeword:

$$\mathcal{C} = \{000000\}. \tag{10.18}$$

This code has lowest possible rate $R = 0$. However, the distortion is much higher.

SSQ 10.2. For a source code with $\mathcal{C} = \{000, 111\}$, find the expected distortion for an $n = 3$ binary source with $p_{\mathbf{X}}(\mathbf{x}) = \frac{1}{8}$.

10.2 Rate-Distortion Theorem

Various codes will have various value for R and D, together these two numbers are the *rate-distortion pair* (R, D). Given a fixed rate, we want codes with distortion as low as possible. Likewise, for a fixed distortion, we want the rate to be as low as possible. We want both the rate and distortion to be as low as possible, but clearly there is a tradeoff. The goal is to find that tradeoff, and to find the fundamental limits of rate-distortion.

This section describes the fundamental limits on the tradeoff between rate and distortion. Example 10.3 used $n = 6$ to achieve a specific rate-distortion pair $(\frac{1}{3}, \frac{1}{4})$. Fundamental limits can be achieved by considering random codebooks with length $n \to \infty$.

Section 10.1 used fixed-length codes $\mathcal{C}$ to show the reconstruction process. Now, the reconstruction process is described using a probability distribution $p_{\widehat{\mathsf{X}}|\mathsf{X}}(\widehat{x}|x)$. The $n = 6$ example has such a distribution:

$$p_{\widehat{\mathsf{X}}|\mathsf{X}}(\widehat{x}|x) = \begin{bmatrix} \frac{3}{4} & \frac{1}{4} \\ \frac{1}{4} & \frac{3}{4} \end{bmatrix}, \tag{10.19}$$

obtained from Table 10.1.

10.2.1 Rate-Distortion Function

The mutual information $I(\mathsf{X};\mathsf{Y})$ is the "rate of flow" of bits between X and Y. In the channel capacity case, the goal was to *maximize* this flow of information across a channel. In rate-distortion theory, the goal is to *minimize* this flow of information, without exceeding some distortion. The idea of minimization of information flow rate can be thought of as representing $\mathbf{x}$ by an approximation $\widehat{\mathbf{x}}$, where $\widehat{\mathbf{x}}$ has as few bits as possible.

The optimized tradeoff between rate and distortion is characterized by the rate-distortion function. The code in Example 10.3 has the rate-distortion pair $(R, D) = (\frac{1}{3}, \frac{1}{4})$. We expect that as the allowed distortion D increases, the code rate R will decrease. For example, by allowing the decompressed image or audio to have lower quality, the source can be more highly compressed.

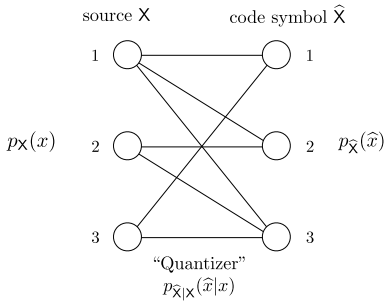


Figure 10.2: An example of a rate-distortion problem with $|\mathcal{X}| = 3$ and $|\widehat{\mathcal{X}}| = 3$. The source distribution $p_{\mathsf{X}}(x)$ is fixed, while $p_{\widehat{\mathsf{X}}|\mathsf{X}}(\widehat{x}|x)$ is to be optimized.

Definition 10.3. A rate-distortion pair (R, D) is *achievable* if there exists a sequence of $(2^{nR}, n)$ codes with encoder and decoder f_n, g_n with:

$$\lim_{n\to\infty} E\Big[d\Big(\mathbf{X}, g_n\big(f_n(\mathbf{X})\big)\Big)\Big] \leq D \tag{10.20}$$

Definition 10.4. A *rate-distortion region* for a source is the closure of the set of achievable rate distortion pairs (R, D).

Definition 10.5. The *rate-distortion function* $R(D)$ is the infimum of rates R such that (R, D) is in the rate distortion region of the source of a given distortion D.

Definition 10.6. The *information-rate distortion function* $R_{\mathrm{I}}(D)$ for a source X is:

$$R_{\mathrm{I}}(D) \quad = \quad \min_{p_{\widehat{\mathsf{X}}|\mathsf{X}}(\widehat{x}|x):Ed(\mathsf{X},\widehat{\mathsf{X}})\leq D} I(\mathsf{X};\widehat{\mathsf{X}})$$

where the minimum is taken over all $p_{\widehat{\mathsf{X}}|\mathsf{X}}(\widehat{x}|x)$ that satisfy $E[d(\mathsf{X},\widehat{\mathsf{X}})] \leq D$ where:

$$E[d(\mathsf{X},\widehat{\mathsf{X}})] = \sum_{x\in\mathcal{X}}\sum_{\widehat{x}\in\widehat{\mathcal{X}}} p_{\mathsf{X},\widehat{\mathsf{X}}}(x,\widehat{x})d(x,\widehat{x}) \tag{10.21}$$

10.2.2 Rate-Distortion Theorem

The goal of rate-distortion theory is to find good $p_{\widehat{\mathsf{X}}|\mathsf{X}}(\widehat{x}|x)$. The source distribution $p_{\mathsf{X}}(x)$ is fixed, while the $p_{\widehat{\mathsf{X}}|\mathsf{X}}(\widehat{x}|x)$ is to be optimized. Since the input distribution $p_{\mathsf{X}}(x)$ is fixed, this is equivalent to finding good $p_{\mathsf{X},\widehat{\mathsf{X}}}(x,\widehat{x})$. A good code $\mathcal{C}$ will satisfy this $p_{\widehat{\mathsf{X}}|\mathsf{X}}(\widehat{x}|x)$. With those definitions, we can now give the rate-distortion theorem:

Proposition 10.1. *Rate-Distortion Theorem* The rate distortion function for an iid source X with distribution $p_{\mathsf{X}}(x)$ and distortion function $d(\cdot,\cdot)$ is equal to

the information rate distortion function. That is:

$$R(D) = R_{\mathrm{I}}(D), \tag{10.22}$$

$$R(D) = \min_{p_{\widehat{\mathsf{X}}|\mathsf{X}}(\widehat{x}|x):Ed(\mathsf{X},\widehat{\mathsf{X}})\leq D} I(\mathsf{X};\widehat{\mathsf{X}}) \tag{10.23}$$

The rate distortion function $R(D)$, given by Definition 10.5, concerns the construction of codes. The information rate distortion function $R_{\mathrm{I}}(D)$, given by Definition 10.6, concerns minimization of mutual information. Proposition 10.1 says that these two functions are equal. The proposition gives a necessary and sufficient condition on the existence of a rate distortion code.

Compare this with the channel coding theorem, Proposition 8.5. In channel coding, capacity is the maximization of mutual information — in rate-distortion theory, the rate-distortion function is the minimization of mutual information. In channel coding, the channel $p_{\mathsf{Y}|\mathsf{X}}(y|x)$ is fixed and the input distribution is optimized — in rate-distortion theory, the source distribution $p_{\mathsf{X}}(x)$ is fixed, and the conditional distribution $p_{\widehat{\mathsf{X}}|\mathsf{X}}(\widehat{x}|x)$ is optimized.

10.2.3 Comments on the Proof

Let $p_{\mathsf{X}}(x)$ be the source distribution, and let $p_{\widehat{\mathsf{X}}}(x)$ be the distribution of the reconstruction alphabet $\widehat{\mathcal{X}}$. The codebook of M words each of length n is obtained by drawing randomly from $p_{\widehat{\mathsf{X}}}(x)$ to obtain entries $\widehat{x}_i(m)$:

$$\mathcal{C} = \begin{bmatrix} \widehat{x}_1(1) & \widehat{x}_2(1) & \cdots & \widehat{x}_n(1) \\ \widehat{x}_1(2) & \widehat{x}_2(2) & \cdots & \widehat{x}_n(2) \\ \vdots & & & \\ \widehat{x}_1(M) & \widehat{x}_2(M) & \cdots & \widehat{x}_n(M) \end{bmatrix}, \tag{10.24}$$

where $M = 2^{nR}$ for a code of rate R.

In Cover and Thomas, section 10.4 proves the converse to the Rate-Distortion Theorem. Section 10.5 proves the achievability. The proofs are similar to the channel coding case, with the addition of the distortion constraint to typical sets.

10.3 $R(D)$ for Discrete Sources

Given a source $p_{\mathsf{X}}(x)$ and a distortion measure $d(x,\widehat{x})$, the rate-distortion function $R(D)$ can be found analytically in a few cases[1]. This section describes the test channel, a procedure for finding the test channel, and then applies it to the binary source, where $R(D)$ can be found in closed form.

[1] The Arimoto-Blahut algorithm, the subject of a future chapter, is a numerical technique for finding solutions in general.

$$\widehat{\mathsf{X}} \rightarrow \boxed{p_{\mathsf{X}|\widehat{\mathsf{X}}}(x|\widehat{x})} \rightarrow \mathsf{X}.$$

Figure 10.3: Test channel for rate-distortion theory.

10.3.1 Test Channel

To find the optimal $p_{\widehat{\mathsf{X}}|\mathsf{X}}(\widehat{x}|x)$ it is often convenient to consider the *test channel* $p_{\mathsf{X}|\widehat{\mathsf{X}}}(x|\widehat{x})$, illustrated in Fig. 10.3. If we write mutual information as $I(\mathsf{X};\widehat{\mathsf{X}}) = H(\mathsf{X}) - H(\mathsf{X}|\widehat{\mathsf{X}})$, then the test channel is useful for computing $H(\mathsf{X}|\widehat{\mathsf{X}})$.

Since $p_{\mathsf{X}}(x)$ is fixed, the joint distribution $p_{\mathsf{X},\widehat{\mathsf{X}}}(x,\widehat{x})$ is readily obtained from $p_{\mathsf{X}|\widehat{\mathsf{X}}}(x|\widehat{x})$. The joint distribution $p_{\mathsf{X},\widehat{\mathsf{X}}}(x,\widehat{x})$ must satisfy two conditions:

1. The expected distortion given by (10.21)

$$D = \sum_{x\in\mathcal{X}} \sum_{\widehat{x}\in\widehat{\mathcal{X}}} p_{\mathsf{X},\widehat{\mathsf{X}}}(x,\widehat{x}) d(x,\widehat{x}) \tag{10.25}$$

   must be satisfied with equality, and

2. The marginal distribution must satisfy the given $p_{\mathsf{X}}(x)$, that is,

$$\sum_{\widehat{x}\in\widehat{\mathcal{X}}} p_{\mathsf{X},\widehat{\mathsf{X}}}(x,\widehat{x}) = p_{\mathsf{X}}(x) \tag{10.26}$$

Generally speaking, there is no systematic way to find the test channel. A procedure that works in some cases is:

1. Parameterize the test channel, considering the distortion metric. If $d(x,\widehat{x}) = \infty$, then the corresponding $p_{\mathsf{X}|\widehat{\mathsf{X}}}(x|\widehat{x})$ must be 0. Exploit any symmetries.

2. Parameterize the test channel input distribution.

3. Find the unknown parameters to satisfy (10.25) and (10.26). It may not be possible to find all parameters.

4. Compute $H(\mathsf{X}|\widehat{\mathsf{X}})$ for the test channel. If all parameters can be found in Step 3, write $R(D) = H(\mathsf{X}) - H(\mathsf{X}|\widehat{\mathsf{X}})$, using the parameters. If not all parameters can be found, maximize $H(\mathsf{X}|\widehat{\mathsf{X}})$ in the unknown parameters, since:

$$\min I(\mathsf{X};\widehat{\mathsf{X}}) = H(\mathsf{X}) - \max H(\mathsf{X}|\widehat{\mathsf{X}}), \tag{10.27}$$

   and write $I(\mathsf{X};\widehat{\mathsf{X}})$ using the maximized parameters and $\max H(\mathsf{X}|\widehat{\mathsf{X}})$ over the parameters. $R(D)$ should *not* contain the unknown parameters.

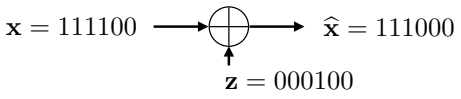
10.3.2 $R(D)$ for the Binary Source

The rate-distortion function $R(D)$ for a binary source, $\mathsf{X} \in \{0,1\}$ with $p_{\mathsf{X}}(x) = [1-p, p]$ and Hamming distortion metric (10.7) is derived. We use $\widehat{\mathcal{X}} = \{0,1\}$.

Proposition 10.2. *Rate-distortion function for binary source* The rate-distortion function $R(D)$ for a binary source X, with Hamming distortion is given by:

$$R(D) = \begin{cases} h(p) - h(D), & 0 \le D \le \min(p, 1-p) \\ 0, & \min(p, 1-p) < D \end{cases} \tag{10.28}$$

Before the proof, consider the reconstruction $\widehat{\mathbf{x}} = g\big(f(\mathbf{x})\big)$ as a noise process where $\widehat{\mathbf{x}} = \mathbf{x} \oplus \mathbf{z}$. Referring to Example10.2:



The key observation is that $\mathbf{z}$ is a distortion vector. In general, if $p_{\mathsf{Z}}(z) \sim [1-D, D]$, then the expected distortion will be D. Also, note that the Hamming distortion is:

	$\widehat{x}$	
$d(x,\widehat{x})$	0	1
$x=0$	0	1
$x=1$	1	0

Proof Assume that $p < \frac{1}{2}$. We want to calculate

$$R(D) = \min I(\mathsf{X}; \widehat{\mathsf{X}}) \tag{10.29}$$

First write a lower bound on $I(\mathsf{X}; \widehat{\mathsf{X}})$:

$$\begin{aligned} I(\mathsf{X}; \widehat{\mathsf{X}}) &= H(\mathsf{X}) - H(\mathsf{X}|\widehat{\mathsf{X}}) && (10.30) \\ &= h(p) - H(\mathsf{X} \oplus \widehat{\mathsf{X}}|\widehat{\mathsf{X}}) && (10.31) \\ &\ge h(p) - H(\mathsf{X} \oplus \widehat{\mathsf{X}}) && (10.32) \\ &\ge h(p) - h(D) && (10.33) \end{aligned}$$

Now, we show that $H(\mathsf{X}|\widehat{\mathsf{X}}) = h(D)$ so that the bound can be achieved with equality. Choose $E[d] = D$ with equality, so the following will be the minimization of mutual information.

1. The test channel has input $\widehat{\mathsf{X}}$, and output X, both are binary. Parameterize $p_{\mathsf{X}|\widehat{\mathsf{X}}}(1|0) = s$, so $p_{\mathsf{X}|\widehat{\mathsf{X}}}(0|0) = 1-s$. By the symmetry of the distortion metric, $p_{\mathsf{X}|\widehat{\mathsf{X}}}(0|1) = s$, so the test channel is:

$$p_{\mathsf{X}|\widehat{\mathsf{X}}}(x|\widehat{x}) = \begin{bmatrix} 1-s & s \\ s & 1-s \end{bmatrix}. \tag{10.34}$$

2. Parametrize the test channel input as $p_{\widehat{\mathsf{X}}}(1) = r$ and $p_{\widehat{\mathsf{X}}}(0) = 1 - r$.

3. Find the unknown parameters. The joint distribution is:

$$p_{\mathsf{X},\widehat{\mathsf{X}}}(x,\widehat{x}) = p_{\mathsf{X}|\widehat{\mathsf{X}}}(x|\widehat{x}) \cdot p_{\widehat{\mathsf{X}}}(x) \tag{10.35}$$

$$= \begin{bmatrix} (1-s)(1-r) & sr \\ s(1-r) & (1-s)r \end{bmatrix} \tag{10.36}$$

where rows are $\widehat{\mathcal{X}}$. Referring (10.25), we must have $s(1-r)+sr = D$, or $s = D$. Next, find r such that $p_{\mathsf{X}}(x) = [1-p, p]$. That means:

$$\sum_{\widehat{x}\in\widehat{\mathcal{X}}} p_{\widehat{\mathsf{X}}}(x) \cdot p_{\mathsf{X}|\widehat{\mathsf{X}}}(x|\widehat{x}) = p_{\mathsf{X}}(x)$$

$$\begin{bmatrix} 1-r & r \end{bmatrix} \cdot \begin{bmatrix} 1-D & D \\ D & 1-D \end{bmatrix} = \begin{bmatrix} 1-p & p \end{bmatrix}, \quad \text{See Example 2.2}$$

$$(1-r)D + r(1-D) = p$$

$$r = \frac{p-D}{1-2D}.$$

4. Find $H(\mathsf{X}|\widehat{\mathsf{X}})$:

$$H(\mathsf{X}|\widehat{\mathsf{X}}) = p_{\widehat{\mathsf{X}}}(1)H(\mathsf{X}|\widehat{\mathsf{X}}=1) + p_{\widehat{\mathsf{X}}}(0)H(\mathsf{X}|\widehat{\mathsf{X}}=0) \tag{10.37}$$

$$= rh(D) + (1-r)h(D) \tag{10.38}$$

$$= h(D), \tag{10.39}$$

and find the rate-distortion function:

$$R(D) = I(\mathsf{X};\widehat{\mathsf{X}}) \tag{10.40}$$

$$= H(\mathsf{X}) - H(\mathsf{X}|\widehat{\mathsf{X}}) \tag{10.41}$$

$$= h(p) - h(D) \tag{10.42}$$

That is, we can achieve $I(\mathsf{X};\widehat{\mathsf{X}}) = h(p) - h(D)$ exactly, and moreover the distortion is $\Pr(\mathsf{X} \neq \widehat{\mathsf{X}}) = D$. Note that if $D = p$, then $R(D) = 0$.

If $p \leq D \leq \frac{1}{2}$, we can achieve $R(D) = 0$ by letting $\widehat{\mathsf{X}} = 0$. Then $I(\mathsf{X};\widehat{\mathsf{X}}) = 0$ and $D = p$. □

Fig. 10.4 shows the rate-distortion curve given in (10.28). The case of $D = 0$ corresponds to the lossless case — $h(p) \leq R$. The case of $R = 0$ can be achieved by a codebook with one codeword, the all-zeros codeword.

10.4 Quantization of Continuous-Valued Sources

Until now, we only handled discrete random variables, for example $\mathcal{X} = \{1, 2, 3, \ldots, M\}$. In this section we introduce quantization of continuous random variables, for example $\mathcal{X} = \mathbb{R}$, the set of real numbers. Chapter 9 introduced entropy and mutual information for continuous random variable.

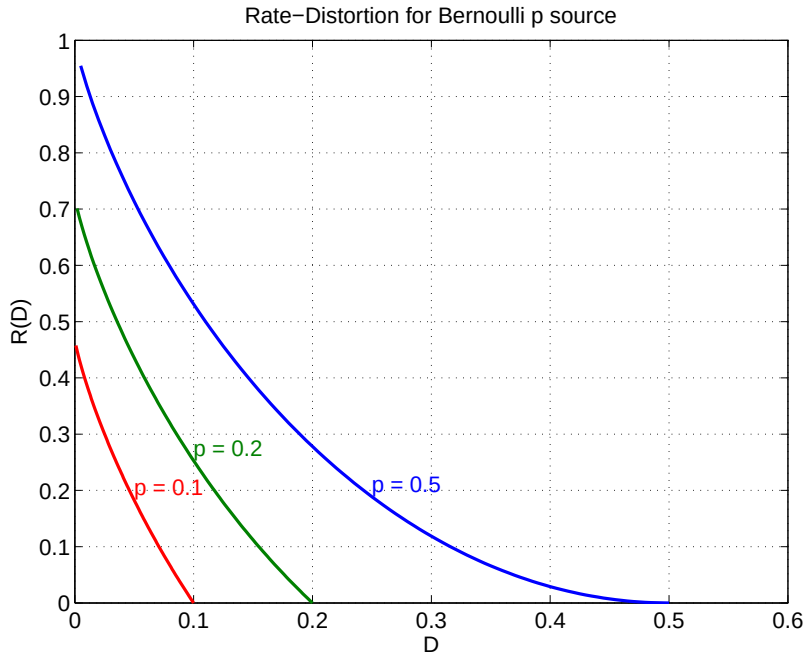


Figure 10.4: Rate-distortion function for binary random variable with $p_{\mathsf{X}}(x) = 0.1$, 0.2 and 0.5.

10.4.1 Background

Consider an n-dimensional source $\mathbf{X} = (\mathsf{X}_1, \mathsf{X}_2, \ldots, \mathsf{X}_n)$ and the sample space $\mathcal{X}^n \subseteq \mathbb{R}^n$.

For continuous sources, a common distortion metric is *squared-error distortion* or Euclidean distance. The squared-error distortion between a sequence $\mathbf{x} = (x_1, \ldots, x_n)$ and $\widehat{\mathbf{x}} = (\widehat{x}_1, \ldots, \widehat{x}_n)$ is:

$$d(\mathbf{x}, \widehat{\mathbf{x}}) = \frac{1}{n} \sum_{i=1}^{n} d(x_i, \widehat{x}_i), \tag{10.43}$$

where

$$d(x_i, \widehat{x}_i) = (x_i - \widehat{x}_i)^2. \tag{10.44}$$

From Definition 10.2, the expected distortion for a $(2^{nR}, n)$ distortion code is:

$$D = E[(\mathsf{X} - \widehat{\mathsf{X}})^2]. \tag{10.45}$$

The codebook consists of K *reconstruction points*[2]:

$$\mathcal{C} = \{\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_K\} \tag{10.46}$$

[2]The codebook size is denoted by K instead of M, and the reconstruction points are $\mathbf{m}_i$ instead of $\widehat{\mathbf{x}}_i$. This is for compatibility with the usual notation of the K-means algorithm.

For each $\mathbf{m}_i$, the *reconstruction region* $\mathcal{R}_i \subset \mathbb{R}^n$ are the points are closest to $\mathbf{m}_i$:

$$\mathcal{R}_i = \{\mathbf{x} | d(\mathbf{x}, \mathbf{m}_i) \leq d(\mathbf{x}, \mathbf{m}_j), j \neq i\} \tag{10.47}$$

Example 10.5. For $n = 1$, consider 1-bit quantization of a Gaussian source. Let X be distributed as a zero-mean, variance σ^2 Gaussian. To quantize X to one bit, clearly we should place the decision at $x = 0$. The minimum distortion is given by the reconstruction points $\widehat{\mathcal{X}} = \{-\sqrt{\frac{2}{\pi}}\sigma, \sqrt{\frac{2}{\pi}}\sigma\}$, that is:

$$\widehat{x} = \begin{cases} -\sqrt{\frac{2}{\pi}}\sigma & \text{if } x < 0 \\ \sqrt{\frac{2}{\pi}}\sigma & \text{if } x \geq 0 \end{cases}. \tag{10.48}$$

Suppose a $\sigma^2 = 1$ source produces:

$$\mathbf{x} = (1.03, 0.73, -0.30, 0.29, -0.79),$$

then the messages are binary:

$$m = (1, 1, 0, 1, 0)$$

and the reconstructed points are:

$$\widehat{\mathbf{x}} = \sqrt{\frac{2}{\pi}}, \sqrt{\frac{2}{\pi}}, -\sqrt{\frac{2}{\pi}}, \sqrt{\frac{2}{\pi}}, -\sqrt{\frac{2}{\pi}}, \tag{10.49}$$

10.4.2 Rate-Distortion for Gaussian Sources

Consider a Gaussian source with $\mathsf{X} \sim \mathcal{N}(0, \sigma^2)$. Rate-distortion function for lossy compression of this source, with a squared-error distortion D is the best possible scheme as $n \to \infty$.

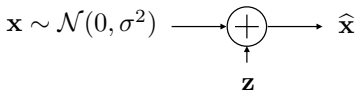
For the lossy source coding problem, the rate-distortion theorem, Proposition 10.1, was stated for discrete sources. However, the theorem is also valid for well-behaved continuous sources.

Proposition 10.3. *Rate-distortion function for Gaussian source* The rate-distortion function $R(D)$ for a Gaussian source X with squared-error distortion is:

$$R(D) = \begin{cases} \frac{1}{2}\log\frac{\sigma^2}{D}, & 0 \leq D \leq \sigma^2 \\ 0 & D > \sigma^2 \end{cases} \tag{10.50}$$

The rate-distortion curve $R(D)$ is illustrated for several values of σ^2 in Fig. 10.5.

As in the binary source case, first consider the reconstruction $\widehat{\mathbf{x}} = g\big(f(\mathbf{x})\big)$ as a noise process:



Again the key observation is to connect the quantization noise Z with the distortion:

$$\mathsf{Z} = \widehat{\mathsf{X}} - \mathsf{X} \tag{10.51}$$

$$\mathrm{Var}[\mathsf{Z}] = \mathrm{Var}[\widehat{\mathsf{X}} - \mathsf{X}] \tag{10.52}$$

$$= E[(\widehat{\mathsf{X}} - \mathsf{X})^2] \quad \text{Assume 0-mean Gaussians} \tag{10.53}$$

$$= D \quad \text{definition of expected distortion} \tag{10.54}$$

Proof Extending the rate-distortion theorem for continuous alphabets:

$$R(D) = \min_{p_{\widehat{\mathsf{X}}|\mathsf{X}}(\widehat{x}|x): E(\mathsf{X}-\widehat{\mathsf{X}})^2 \le D} I(\mathsf{X}; \widehat{\mathsf{X}}). \tag{10.55}$$

First a lower bound on $I(\mathsf{X}; \widehat{\mathsf{X}})$, then use this to construct a test channel; the test channel will be used to show the bound can be achieved with equality. The lower bound is:

$$\begin{aligned}
I(\mathsf{X}; \widehat{\mathsf{X}}) &= H(\mathsf{X}) - H(\mathsf{X}|\widehat{\mathsf{X}}) \\
&= \frac{1}{2}\log 2\pi e\sigma^2 - H(\mathsf{X} - \widehat{\mathsf{X}}|\widehat{\mathsf{X}}) \\
&\ge \frac{1}{2}\log 2\pi e\sigma^2 - H(\mathsf{X} - \widehat{\mathsf{X}}) \\
&\ge \frac{1}{2}\log 2\pi e\sigma^2 - H\Big(\mathcal{N}\big(0, E[(\mathsf{X} - \widehat{\mathsf{X}})^2]\big)\Big) \\
&= \frac{1}{2}\log 2\pi e\sigma^2 - \frac{1}{2}\log 2\pi e E(\mathsf{X} - \widehat{\mathsf{X}})^2 \\
&\ge \frac{1}{2}\log 2\pi e\sigma^2 - \frac{1}{2}\log 2\pi e D \\
&= \frac{1}{2}\log \frac{\sigma^2}{D},
\end{aligned} \tag{10.56}$$

To find the rate-distortion function, the test channel is:

$$\mathsf{X} = \widehat{\mathsf{X}} + \mathsf{Z} \tag{10.57}$$

From the above argument, to achieve the lower bound with equality, we should choose $\mathsf{X} - \widehat{\mathsf{X}}$ to be Gaussian. Since X is Gaussian, Z is also Gaussian, let $\mathsf{Z} \sim \mathcal{N}(0, s)$ for some parameter s. Using the relationship at (10.53),

$$D = E[(\mathsf{X} - \widehat{\mathsf{X}})^2] = E[\mathsf{Z}^2], \tag{10.58}$$

it is clear we should choose $s = D$. From (10.57), $E[\widehat{\mathsf{X}}^2] = D - \sigma^2$. In summary, if $D \le \sigma^2$, we choose:

$$\widehat{\mathsf{X}} \sim \mathcal{N}(0, \sigma^2 - D), \mathsf{Z} \sim \mathcal{N}(0, D) \tag{10.59}$$

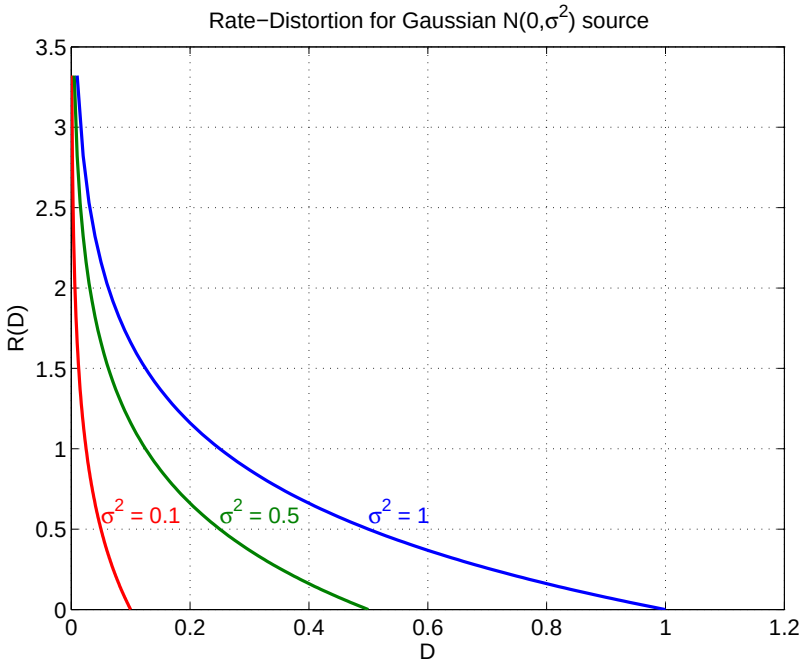


Figure 10.5: Rate-Distortion function for a Gaussian random variable with $\sigma^2 = 0.1, 0.5, 1$.

For this joint distribution, we can find:

$$I(\mathsf{X};\widehat{\mathsf{X}}) \quad = \quad \frac{1}{2}\log\frac{\sigma^2}{D}, \tag{10.60}$$

achieving the bound (10.56) with equality when $0 \leq D \leq \sigma^2$. If $D > \sigma^2$ then choose $\widehat{\mathsf{X}} = 0$ and $R = 0$.

10.4.3 K-Means Algorithm

Example 10.5 showed how to quantize a single Gaussian distribution to 1 bit. However, given 2 bits, it is not immediately clear how to find the optimal reconstruction points, even for an $n = 1$ Gaussian source. And for a multivariate $n \geq 2$ Gaussian source, optimal quantization is particularly difficult.

However, there are two properties of the reconstruction points and their regions.

- Given the reconstruction points $\widehat{\mathcal{X}}$, the reconstruction region should minimize the average distortion. With Euclidean distance metric, any point x should be quantized to the closest point $\widehat{x}$.
- Within some region $\mathcal{R}_i$, its reconstruction point should be chosen to minimize the expected distortion for that region.

Algorithm 10.1 K-Means Algorithm

Require: L points $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_L\}$. Integer K satisfying $K \leq L$.
Ensure: K means $\widehat{\mathcal{X}} = \{\mathbf{m}_1, \ldots, \mathbf{m}_K\}$ which attempt to minimize the expected mean-squared error with $\mathcal{X}$.
Initalization Step:
Construct $\widehat{\mathcal{X}}$ by randomly selecting K of the L points.
Assignment Step:
for $k = 1, 2, \ldots, K$ **do**

$$\mathcal{R}_i = \{\mathbf{x} \in \mathcal{X} | d(\mathbf{x}, \mathbf{m}_i) \leq d(\mathbf{x}, \mathbf{m}_j), \text{for all } j \neq i\} \tag{10.61}$$

end for
If any set $\mathcal{R}_i$ is empty, then update $\mathbf{m}_i$ by a random selection from $\mathcal{X}$.
Update Step (Centroid Step):
for $k = 1, 2, \ldots, K$ **do**

$$\mathbf{m}_i = \frac{1}{|\mathcal{R}_i|} \sum_{\mathbf{x}' \in \mathcal{R}_i} \mathbf{x}' \tag{10.62}$$

end for
Repeat: the Assignment Step and Update Step until the assignment $\widehat{\mathcal{X}}$ does not change.

Instead of a distribution $p_{\mathsf{X}}(x)$, assume that we are given a sequence of L points $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_L\}$ which are sampled from the distribution $p_{\mathsf{X}}(x)$.

The two observations suggest a technique to find a good quantizer. Start with a set of reconstruction points, and assign each $\mathbf{x}_\ell$ to one reconstruction point, the one which is closest. This defines the reconstruction regions. Then, for each region $\mathcal{R}_i$ update the mean $\mathbf{m}_i$ to minimize the distortion within the region. But updating the means will change the optimal assignments, so return to the previous step.

Alternating between these two steps is the core idea of the K-means algorithm, shown in Algorithm 10.1. The assignment step is shown in (10.61), this requires computing the distance between all pairs $\mathbf{x}_\ell$ and $\mathbf{m}_i$. Note that it is possible that some set is empty. If this happens, then update the corresponding mean $\mathbf{m}_i$ with a random selection from $\mathcal{X}$. The update step is shown in (10.62). For each region $\mathcal{R}_i$, the new mean is the average, or centroid, of all the points in that region.

While not optimal, the K-means algorithm is highly effective and widely used. It is often used for clustering, so is usually called K-means clustering. Here, it is being used for quantization. Wikipedia: K-means clustering

10.5 Exercises

10.1 *Rate-Distortion with Erasures* Find the rate-distortion function $R(D)$ for the following problem. Consider a binary source X with $p_{\mathsf{X}}(1) = p$ from $\mathcal{X} = \{0, 1\}$ This is to be coded to a ternary code with $\widehat{\mathcal{X}} = \{0, ?, 1\}$ using the distortion metric:

$$d(x, \widehat{x}) = \begin{cases} 0 & \text{if } \widehat{x} = x \\ 1 & \text{if } \widehat{x} =? \\ \infty & \text{otherwise} \end{cases} . \tag{10.63}$$

10.2 *1-bit quantization of Gaussian* Consider 1-bit quantization of a Gaussian source $\mathsf{X} \sim \mathcal{N}(0, \sigma^2)$.

(a) Show that the reconstruction points that minimize the expected mean-squared error are:

$$\widehat{\mathcal{X}} = \{-\sqrt{\frac{2}{\pi}}\sigma, \sqrt{\frac{2}{\pi}}\sigma\}. \tag{10.77}$$

(b) Find the expected distortion, using the reconstruction set from part (a).

10.3 *K-means for Gaussian Quantization* Perform quantization of a Gaussian source, by implementing the K-means algorithm in your favorite programming language. Generate M random samples from a zero-mean, unit variance Gaussian source, $x_1, x_2, \ldots, x_M$ (use a large value such as $M = 1000$ or $M = 10000$). Apply the K-means algorithm to obtain K reconstruction points, $m_1, m_2, \ldots, m_K$ (This is a one-dimensional K-means algorithm). The distortion function is the average mean-squared error, computed as:

$$MSE = \frac{1}{M} \sum_{k=1}^{K} \sum_{x' \in \widehat{\mathcal{X}}_i} (x' - m_i)^2 \tag{10.78}$$

For a fixed data set, you may repeat the K-means algorithm several times and take the best MSE value. You should write the source code yourself and not use a library.

(a) Plot the rate-distortion function for this source.
(b) On the same plot, show the theoretical R-D pair for $K = 2$.
(c) On the same plot, show the R-D pairs obtained using your K-means algorithm, for $K = 2, 4$ and 8.
(d) Submit the source code you wrote.

Chapter 11

Network Information Theory

Network information theory deals with communication scenarios where there is more than one transmitter and one receiver. In distributed source coding, there are two data sources which must be compressed independently, with the goal of reconstructing both data sources losslessly at the decoder. In the multiple access channel, two users each have a message they want to transmit to one receiver using a shared medium. For both cases, the two users are not allowed to communicate with each other. It is shown that by using coding, the compression rates and communication rates are higher than naive approaches.

11.1 Distributed Source Coding

11.1.1 Motivation

Lossless distributed source coding deals with the lossless compression of two or more *correlated* data streams. Each of the correlated streams is encoded separately and the compressed data from all these encoders are jointly decoded by a single decoder. Since this is lossless compression, the source outputs can be constructed from the compression version with arbitrary small error probability. The following demonstrates the usefulness of distributed source coding.

> Suppose the weather in two neighboring towns, Janestown and Thomasville, is correlated, and that a resident of one town, Jane, wants to send the weather history for a year to a resident of the other town, Thomas. Each day the weather in each town is equally likely to be either "good" or "bad" independently of the past weather, but that the weather in the two towns each day is different with probability p.

Additional reading: Cover and Thomas, Section 15.3 and 15.4. and J. K. Wolf and B. M. Kurkoski, "Slepian-Wolf coding," *Scholarpedia*, vol. 3, no. 11, p. 6789, 2008. Z. Xiong, A. D. Liveris, S. Cheng, "Distributed Source Coding for Sensor Networks," *IEEE Signal Processing Magazine*, September 2004.

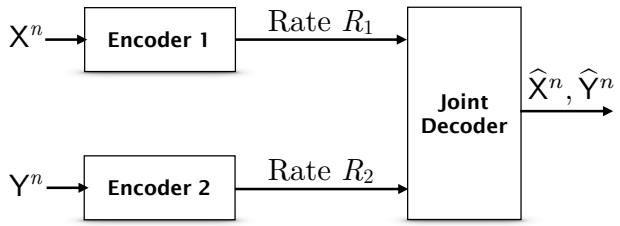


Figure 11.1: Block diagram for distributed source coding.

Assume that Jane records the daily weather in Janestown for a year. Similarly, Thomas does the same for the weather in Thomasville. At the end of the year, Jane wants to send a message to Thomas that will allow Thomas to determine the weather for the year in Janestown. Jane wants to send the shortest message possible. It is clear that all that Jane needs to transmit is the difference between the two daily weather sequences, that is, a "1" if the weather is different, and a "0" if the weather is the same. This difference sequence has 365 digits, of which $365p$ are "1" s and $365(1-p)$ are "0", on average. If Jane knew the weather in both towns, she could losslessly compress this sequence to a length of approximately:

$$365 \cdot h(p) \text{ bits} \tag{11.1}$$

where $h(\cdot)$ is the binary entropy function. What is surprising is that Jane can use this same length message even if she doesn't know the weather in Thomasville and hence doesn't know the difference sequence.

Compared to an encoder that ignores the correlation in the data streams, the separate encoders can achieve better compression rates by exploiting the fact that the data streams are correlated. The surprising result is that Slepian-Wolf coding can in fact achieve the same compression rate as the optimal single encoder that has all correlated data streams as inputs.

This idea has various practical applications, particularly where correlated data streams are physically separated, or where the encoder has limited computational ability. It can be applied to sensor networks such as those for monitoring temperature or seismic activity where wireless transmitters, distributed over some environment, collect data and transmit it to a central location. Two transmitters that are near each other sense similar values and thus produce correlated outputs. Because transmitter resources such as battery power are limited, transmitting at higher compression rates improves the system's performance. The Slepian-Wolf theorem has practical application even when the encoder has access to the multiple correlated data streams. For example, to reduce complexity for video compression, adjacent video frames are correlated. Each frame could be encoded separately, reducing the computational complexity.

11.1.2 Distributed Source Coding

Recall vector source coding for a single source as in Chapter 5, which connects the expected length and rate of an optimal code. Let $\mathbf{x} = x_1, x_2, \ldots, x_n$ be a vector sequence where x_i are independent and identically distributed:

$$p_{\mathbf{X}}(\mathbf{x}) = \prod_{i=1}^{n} p_{\mathsf{X}}(x_i). \tag{11.2}$$

An important result is that the rate R of an optimal code C^* satisfies:

$$H(\mathsf{X}) \leq R(C^*) \leq H(\mathsf{X}) + \epsilon' \tag{11.3}$$

Here ϵ' is a parameter that can be made as small as desired as $n \to \infty$.

In this chapter, source coding of two sources $\mathbf{x}$ and $\mathbf{y}$:

$$\mathbf{x} = x_1, x_2, \ldots, x_n \tag{11.4}$$
$$\mathbf{y} = y_1, y_2, \ldots, y_n \tag{11.5}$$

are considered, as shown in Fig. 11.1. The source is correlated, that is, x_i, y_i are jointly distributed, but pairs (x_i, y_i) are independent and identically distributed:

$$p_{\mathbf{XY}}(\mathbf{x}, \mathbf{y}) = \prod_{i=1}^{n} p_{\mathsf{XY}}(x_i, y_i) \tag{11.6}$$

When $\mathbf{x}$ and $\mathbf{y}$ are binary, it is convenient to represent the source correlation as the number of positions where $\mathbf{x}$ and $\mathbf{y}$ differ. Define $\mathbf{z}$ as:

$$\mathbf{z} = \mathbf{x} \oplus \mathbf{y}, \tag{11.7}$$

so that $z_i = 0$ if the two source sequences agree in position i. If they disagree, then $z_i = 1$.

In distributed source coding, each source is encoded independently. That is, Encoder 1 sees source $\mathbf{x}$ only and compresses this source at rate R_1. Likewise, Encoder 2 sees source $\mathbf{y}$ only and compresses this source at rate R_2. Using what we already know from Chapter 5, each encoder can compress its source at rates

$$R_1 \geq H(\mathsf{X}) \text{ and} \tag{11.8}$$
$$R_2 \geq H(\mathsf{Y}). \tag{11.9}$$

This is a naive scheme because although the $(\mathsf{X}_i, \mathsf{Y}_i)$ are correlated, Encoder 1 and Encoder 2 have ignored this correlation.

Now, temporarily assume that Encoder 2 knows Encoder 1's sequence. Then, it is not hard to show that:

$$R_1 \geq H(\mathsf{X}) \tag{11.10}$$
$$R_2 \geq H(\mathsf{Y}|\mathsf{X}). \tag{11.11}$$

Since $H(\mathsf{Y}|\mathsf{X}) \leq H(\mathsf{Y})$, this means that Encoder 2 can transmit with a lower rate than in the naive scheme.

11.1.3 Slepian-Wolf Theorem

The Slepian-Wolf theorem states something surprising: the rates in (11.10) and (11.11) can be achieved even if Encoder 2 does *not* know Encoder 1's sequence.

The efficiency of the system is measured by the rates in encoded bits per source symbol of the compressed data streams that are output by the encoders. The Slepian-Wolf Theorem specifies the set of rates that allow the decoder to reconstruct these correlated data streams with arbitrarily small error probability.

Although the theorem holds for much more general classes of inputs, the special case stated below is for two correlated data streams, $\mathbf{X}$ and $\mathbf{Y}$ which are formed from making n independent drawings from a joint probability distribution $p_{\mathsf{XY}}(x, y) = \Pr(\mathsf{X} = x, \mathsf{Y} = y)$.

Encoder 1, observes $\mathbf{X}$ and then sends a message to the decoder which is a number from the set $\{1, 2, \cdots, 2^{nR_1}\}$. Similarly, encoder 2 observes $\mathbf{Y}$ and then sends a message to the decoder which is a number from the set $\{1, 2, \cdots, 2^{nR_2}\}$. The outputs from the two encoders are the inputs to the single decoder. The decoder, upon receiving these two inputs, outputs two n-vectors $\widehat{\mathbf{X}}$ and $\widehat{\mathbf{Y}}$ which are estimates of $\mathbf{X}$ and $\mathbf{Y}$, respectively. Refer to Fig. 11.1

Definition 11.1. A rate pair (R_1, R_2) is an *achievable rate pair* if there exists a sequence of $\big((2^{nR_1}, 2^{nR_2}), n\big)$ distributed source codes such that:

$$\Pr(\widehat{\mathbf{X}} \neq \mathbf{X}, \widehat{\mathbf{Y}} \neq \mathbf{Y}) \tag{11.12}$$

can be made as small as desired by choosing n sufficiently large.

Definition 11.2. The closure of the set of all achievable rate pairs is called the *achievable rate region.*

The following entropies can be calculated for the pair of random variables X, Y with joint probability distribution $p_{\mathsf{XY}}(x, y)$:

$$H(\mathsf{X}, \mathsf{Y}), H(\mathsf{X}|\mathsf{Y}), H(\mathsf{Y}|\mathsf{X}), H(\mathsf{X}) \text{ and } H(\mathsf{Y}) \tag{11.13}$$

When calculating entropies, all logarithms will be taken to the base 2.

Proposition 11.1. *Slepian-Wolf Theorem* The achievable rate region for the pair of rates, (R_1, R_2) is the set of points that satisfy the three inequalities:

$$R_1 \geq H(\mathsf{X}|\mathsf{Y}), \tag{11.14}$$

$$R_2 \geq H(\mathsf{Y}|\mathsf{X}), \tag{11.15}$$

$$R_1 + R_2 \geq H(\mathsf{X}, \mathsf{Y}). \tag{11.16}$$

An example of an achievable rate region is shown in Fig. 11.2.

The significance of the Slepian-Wolf theorem is seen by comparing it with the entropy bound for compression of single sources. Separate encoders which ignore the source correlation can achieve rates only $R_1 + R_2 \geq H(\mathsf{X}) + H(\mathsf{Y})$. However, for Slepian-Wolf coding, the separate encoders exploit their knowledge of the correlation to achieve the same rates as an optimal joint encoder, namely, $R_1 + R_2 \geq H(\mathsf{X}, \mathsf{Y})$.

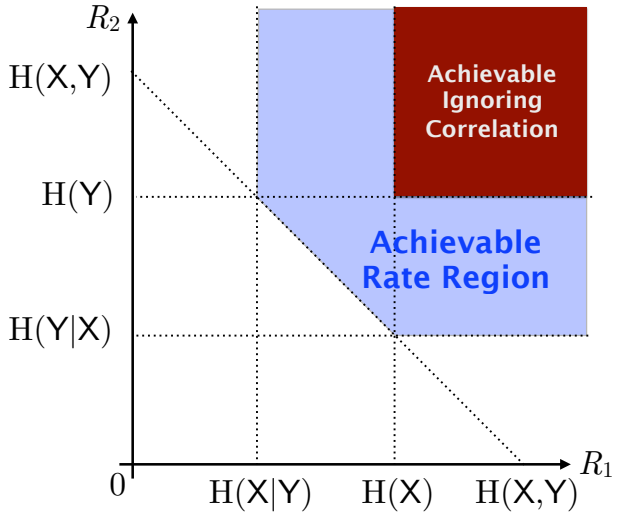


Figure 11.2: Achievable rate region for distributed source coding. The smaller region is achievable if the correlation is ignored.

11.2 Finite-Length Example

The Slepian-Wolf Theorem is valid for asymptotically long sequences with $n \to \infty$. To give a sense of distributed source coding, this section gives a concrete example of Slepian-Wolf coding using finite-length sequences. Instead of assuming that X_i and Y_i are correlated random variables, it is assumed that the two length-n sequences $\mathbf{x}$ and $\mathbf{y}$ are binary sequences that differ in a small number of positions. Also, the sources are uniformly-distributed binary sequences so that all binary source sequences are equally likely.

11.2.1 Slepian-Wolf Using Binning

In a concrete scheme, Encoder 1 transmits binary sequence $\mathbf{x}$ of length n to the destination without compression and thus has rate $R_1 = 1$. Encoder 2 has a codebook where all 2^n binary sequences are separated into 2^r bins. Each bin has an r-bit label, where $r \leq n$. Encoder 2 finds the bin which contains $\mathbf{y}$, then transmits that bin's r-bit label, called $\mathbf{s}$. This compression scheme has rate $R = \frac{r}{n} \leq 1$.

The decoder receives $\mathbf{x}$ and $\mathbf{s}$. To reconstruct $\mathbf{y}$ from $\mathbf{x}$ and $\mathbf{s}$, then decoder looks at the bin for $\mathbf{s}$. This bin contains multiple codewords, and it selects the codeword which differs from $\mathbf{x}$ in the fewest number of positions.

This scheme works when when the difference between $\mathbf{x}$ and $\mathbf{y}$ is not too large. "Not too large" depends on the construction of the codebook. In the language of coding theory, if the minimum distance of the code in each bin is d, then then Hamming distance between $\mathbf{x}$ and $\mathbf{y}$ should be $(d-1)/2$ or less, where d is an odd number.

Example 11.1. Consider Slepian-Wolf compression of binary sources $\mathbf{x}$ and $\mathbf{y}$ with length $n = 3$ using a codebook with four bins, given by the following table.

$\mathbf{s} = 00$	$\mathbf{s} = 01$	$\mathbf{s} = 10$	$\mathbf{s} = 11$
000	001	010	011
111	110	101	100

Here $r = 2$ and there are $2^r = 4$ bins. Each bin has two codewords.

Suppose $\mathbf{x} = 010$ and $\mathbf{y} = 110$, which differ in one position. Encoder 1 sends $\mathbf{x} = 010$ without compression, with $R_1 = 1$. Encoder 2 selects $\mathbf{s} = 01$, because this bin contains $\mathbf{y}$, with $R_2 = 2/3$.

The decoder receives $(\mathbf{x}, \mathbf{s}) = (010, 01)$. To find $\mathbf{y}$, in bin $\mathbf{s} = 01$, the decoder sees that 001 differs from $\mathbf{x}$ in two positions, but 110 differs in only one position. Thus the decoder selects $\widehat{\mathbf{y}} = 110$ as the reconstructed output, which is correct.

In the above example, if the encoders ignored the correlation, then both would need to transmit with $R_1 = 1$ and $R_2 = 1$, with 0 and 1 equally likely. By using a codebook, the rate pair became $(R_1, R_2) = (1, 2/3)$. In the language of coding theory, bin $\mathbf{s} = 00$ is the repeat code with minimum distance $d = 3$. The other 3 bins are cosets of the repeat code, which also have minimum distance of 3. This scheme will succeed when $\mathbf{x}$ and $\mathbf{y}$ differ in one or zero positions.

11.2.2 Slepian-Wolf Using Linear Codes

It is possible to implement Slepian-Wolf coding using the linear block codes, as follows. The code is specified by an r-by-n *parity-check matrix*[1] $\mathbf{H}$. The *syndrome* $\mathbf{s}$ corresponding to a source sequence $\mathbf{y}$ is:

$$\mathbf{s} = \mathbf{y}\mathbf{H}^{\mathrm{t}} \tag{11.17}$$

Note that all operations are performed modulo-2, so that $\mathbf{s}$ is a binary sequence.

A *syndrome table* ϕ is a mapping from syndromes $\mathbf{s}$ to source sequences with t or fewer ones (and $n - t$ or more zeros): That is, ϕ maps from sequences of length r to sequences of length n. The syndrome is a many-to-one mapping. The syndrome table is the inverse operation, but since there are many possible sequences, the source sequences with the smallest number of ones is chosen. The syndrome table ϕ can be found from $\mathbf{H}$ — details are omitted and ϕ will be provided.

Now consider Slepian-Wolf coding of two length-n sources $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ and $\mathbf{y} = (y_1, y_2, \ldots, y_n)$. The binary sequences $\mathbf{x}$ and $\mathbf{y}$ differ in at most t positions. Encoder 1 performs no compression, that is the decoder knows $\mathbf{x}$, which is $R_1 = 1$. However, Encoder 2 performs the following operation. It computes

[1] We are using a linear error-correcting code. The dimension of the code is $k = n - r$ and the error-correcting capability is t. Linear error-correcting codes are beyond the scope of this lecture, but enough coding theory is introduced to explain Slepian-Wolf coding.

$\mathbf{s}_2 = \mathbf{H}\mathbf{y}^{\mathrm{t}}$ and sends $\mathbf{s}_2$ to the decoder. Since $\mathbf{s}_2$ has length r, the rate is $R_2 = r/n$.

Now the decoder wants to recover $\mathbf{x}$ and $\mathbf{y}$ exactly. It already has $\mathbf{x}$. The decoder recovers $\mathbf{y}$ from $\mathbf{s}_2$ and $\mathbf{x}$ as follows. Using the syndrome table, compute:

$$\mathbf{c} = \phi(\mathbf{s}_2), \tag{11.18}$$

which is the lowest-weight vector in the bin. Then, the following operations find $\widehat{\mathbf{y}}$, the vector in the bin closest to $\mathbf{x}$:

$$\mathbf{s}_1 = (\mathbf{x} \oplus \mathbf{c})\mathbf{H}^{\mathrm{t}} \tag{11.19}$$

$$\widehat{\mathbf{z}} = \phi(\mathbf{s}_1) \tag{11.20}$$

$$\widehat{\mathbf{y}} = \mathbf{x} \oplus \widehat{\mathbf{z}}, \tag{11.21}$$

where all operations are modulo-2. If the linear error-correcting code defined by $\mathbf{H}$ can correct t errors, then the Slepian-Wolf scheme will succeed when $\mathbf{x}$ and $\mathbf{y}$ differ in at most t positions.

Example 11.2. Repeat Example 11.1 using the parity check matrix $\mathbf{H}$:

$$\mathbf{H} = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}, \tag{11.22}$$

which has syndrome table:

$\mathbf{s}$	$\phi(\mathbf{s})$
0 0	0 0 0
0 1	0 0 1
1 0	0 1 0
1 1	1 0 0

This code generates the same bins as in Example 11.1.

As before, the source sequences are $\mathbf{x} = 010$ and $\mathbf{y} = 110$. Encoder 1 transmits $\mathbf{x}$ uncompressed. Encoder 2 finds $\mathbf{s} = \mathbf{y}\mathbf{H}^{\mathrm{t}}$ as:

$$\mathbf{s} = \begin{bmatrix} 1 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}^{\mathrm{t}} = \begin{bmatrix} 0 & 1 \end{bmatrix},$$

and transmits this.

The decoder receives $(\mathbf{x}, \mathbf{s}) = (010, 01)$. First, the decoder finds $\mathbf{c}$ in the syndrome table:

$$\mathbf{c} = \phi(\mathbf{s}) = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \tag{11.23}$$

Then, the following operations find $\widehat{\mathbf{y}}$, the vector in the bin closest to $\mathbf{x}$:

$$\mathbf{s}_1 = (\mathbf{x} - \mathbf{c})\mathbf{H}^{\mathrm{t}} = \begin{bmatrix} 1 & 1 \end{bmatrix}$$
$$\widehat{\mathbf{z}} = \phi(\mathbf{s}_1) = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}$$
$$\widehat{\mathbf{y}} = \mathbf{x} - \widehat{\mathbf{z}} = \begin{bmatrix} 1 & 1 & 0 \end{bmatrix},$$

which is the correct answer.

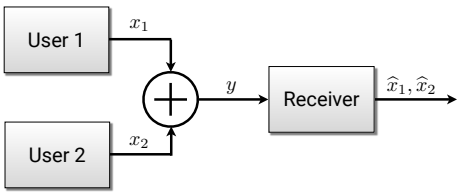


Figure 11.3: A single-use MAC channel with two users. In this example, the receiver can reliably determine $\widehat{x}_1, \widehat{x}_2$ from y.

11.3 Multiple Access Channel: Motivation

In a multiple access channel, two or more users transmit messages to one receiver. They must share a noisy communications channel, for example, two mobile phones transmitting wirelessly to a base station. There are now two impairments: (1) two users transmitting at the same time will interfere with each other (2) the channel itself may be noisy. Each of two users has its own rate, and usually there is a tradeoff between the two rates. The MAC capacity maximizes the achievable rates, while discovering the tradeoff between the two rates. Only two-user MAC is studied here; this can be generalized to multiple users.

11.3.1 Single-Use MAC Channel with No Noise

Consider the following simple example of a MAC channel. In the single-use MAC channel, two transmitters transmit one symbol each, and the receiver sees the sum, shown in Fig. 11.3. User 1 can select a symbol x_1 from alphabet $\mathcal{X}_1$ and User 2 can select a symbol x_2 from alphabet $\mathcal{X}_2$. The receiver sees y:

$$y = x_1 + x_2, \tag{11.24}$$

that is, there is no noise. The receiver should be able to estimate $\widehat{x}_1, \widehat{x}_2$ with no errors.

Suppose User 1 has available alphabet $\mathcal{X}_1 = \{0, 1, 2\}$ and User 2 has available transmit alphabet $\mathcal{X}_2 = \{0, 1, 2, 3\}$. The two users can agree on a scheme to use. How can the users communicate to the receiver?

In one approach, User 1 uses the full available alphabet $\{0, 1, 2, 3\}$ with rate $R_1 = \log 4 = 2$, while User 2 does not transmit, choosing $x_2 = 0$. This achieves the rate pair $(R_1, R_2) = (2, 0)$. At the receiver $y = x_1$, so the message has no errors. On the other hand, User 2 could transmit the full available set $\{0, 1, 2\}$ with rate $R_2 = \log 3$, while User 1 does not transmit, choosing $x_1 = 0$. This achieves rate $(R_1, R_2) = (0, \log 3)$.

In another approach, User 1 and User 2 transmit at the same time and the receiver performs joint decoding. The following table expresses all values of $y = x_1 + x_2$:

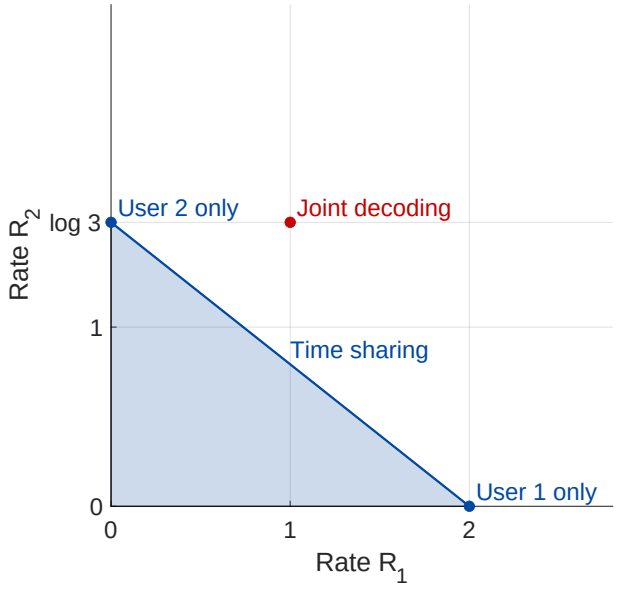


Figure 11.4: Joint decoding gives higher rates than time sharing.

	x_2		
x_1	0	1	2
0	0	1	2
1	1	2	3
2	2	3	4
3	3	4	5

Suppose User 1 chooses a subset of the possible alphabet, specifically $x_1 \in \{0, 3\}$, which gives rate $R_1 = 1$. User 2 uses the full alphabet $\mathbf{x}_2 \in \{0, 1, 2\}$, giving rate $R_2 = \log 3$. In this case, the receiver uses a decoding table to obtained the estimates of the transmitted values $\widehat{x}_1, \widehat{x}_2$:

y	$(\widehat{x}_1, \widehat{x}_2)$
0	(0,0)
1	(0,1)
2	(0,2)
3	(3,0)
4	(3,1)
5	(3,2)

Receive performs joint decoding to uniquely determine x_1, x_2 from y without error. The rate pair for this scheme is $(R_1, R_2) = (1, \log 2)$, which is shown in Fig. 11.3.

For the MAC channel, a key metric is the *sum rate* $R_1 + R_2$. When User 1 only transmits the sum rate is 2 bits, and when User 2 only transmits the sum rate is $\log 3 \approx 1.58$ bits. On the other hand, the joint decoding approach can

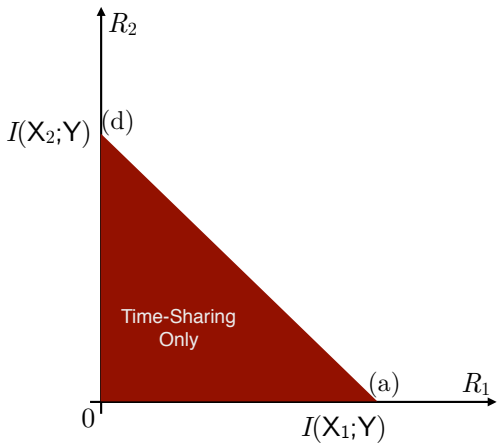


Figure 11.5: Time-sharing is a naive approach for two users to share the multiple-access channel.

achieve sum rate $R_1 + R_2 = 1 + \log 3 \approx 2.58$ bits, which is the greatest of the three cases.

If multiple uses of the channel are used, then the users can perform time sharing. User 1 transmits some fraction α of the time, with $0 \leq \alpha \leq 1$, and User 2 transmits fraction $1 - \alpha$ of the time. Then, any rate pair:

$$(R_1, R_2) = (2\alpha, \alpha \log 3) \tag{11.25}$$

is achievable. Fig. 11.4 shows possible achievable rates for time sharing, including User 1 only, User 2 only, and time sharing, achieved by the line connecting the two points for the two users. The figure also shows the joint decoding achievable rates, which are higher than achieved by time sharing.

SSQ 11.1. Consider a single-use MAC channel where the channel output is the mod-6 sum of its inputs: $y = x_1 + x_2 \bmod 6$. User 1 uses $\mathcal{X}_1 = \{0, 3\}$. User 2 has available $\{0, 1, 2, 3, 4, 5\}$. Which values should User 2 choose to maximize R_2?

11.4 Multiple Access Channel and Its Capacity

In the single-use MAC, $n = 1$, meaning the channel could only be used once and we chose $\mathcal{X}_1, \mathcal{X}_2$ to avoid errors due to interference. Now we consider codes with n large.

Recall channel coding as in Chapters 7–8. There was one transmitter and one receiver. For a channel with probability distribution $p_{\mathsf{Y}|\mathsf{X}}(y|x)$, the input is $\mathbf{x} = x_1, x_2, \ldots, x_n$ and the output is $\mathbf{y} = y_1, y_2, \ldots, y_n$. Then, the highest possible communication rate R is:

$$R < C = \max_{p_{\mathsf{X}}(x)} I(\mathsf{X};\mathsf{Y}), \tag{11.26}$$

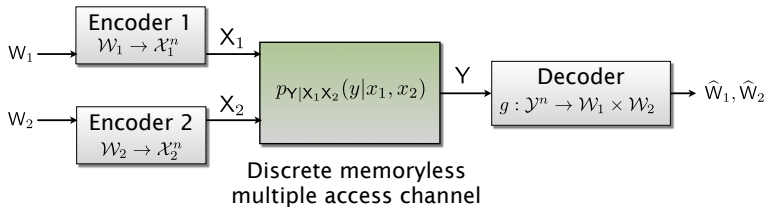


Figure 11.6: Multiple access channel: encoders, discrete memoryless channel model, and decoder.

where the x_i are distributed as $p_{\mathsf{X}}(x)$ and the y_i are distributed as $p_{\mathsf{Y}}(y)$. We now extend this to the idea of *two* transmitters, while keeping one receiver.

11.4.1 Definition and Achievable Rates

Now, the formal definition of the MAC channel is given, and illustrated in Fig. 11.6.

Definition 11.3. A *multiple access channel* is a discrete memoryless multiple access channel, with two input alphabets $\mathcal{X}_1$ and $\mathcal{X}_1$, and output alphabet $\mathcal{Y}$, and conditional probability distribution:

$$p_{\mathsf{Y}|\mathsf{X}_1\mathsf{X}_2}(y|x_1,x_2). \tag{11.27}$$

For a codewords of length n, the channel will be used n times. The channel is independent and identically distributed as follows:

$$p_{\mathbf{Y}|\mathbf{X}_1\mathbf{X}_2}(\mathbf{y}|\mathbf{x}_1,\mathbf{x}_2) = \prod_{i=1}^{n} p_{\mathsf{Y}|\mathsf{X}_1,\mathsf{X}_2}(y_i|x_{1,i},x_{2,i}) \tag{11.28}$$

The two users have messages w_1 and w_2. Encoding is performed independently for the two users:

$$w_1 \in \mathcal{W}_1 = \{1,2,\ldots,2^{nR_1}\} \text{ and}$$
$$w_2 \in \mathcal{W}_2 = \{1,2,\ldots,2^{nR_2}\},$$

where User 1 has rate R_1 and User 2 has rate R_2. Each user independently encodes its message to a length n sequence:

$$\mathcal{W}_1 \to \mathcal{X}_1^n \tag{11.29}$$
$$\mathcal{W}_2 \to \mathcal{X}_2^n, \tag{11.30}$$

Because of independence, $H(\mathsf{X}_1|\mathsf{X}_2) = H(\mathsf{X}_1)$. The two codewords $\mathbf{x}_1$ and $\mathbf{x}_2$ are inputs to the multiple-access channel (11.28). The channel output is $\mathbf{y}$.

The decoding function is:

$$g : \mathcal{Y}^n \to \mathcal{W}_1 \times \mathcal{W}_2. \tag{11.31}$$

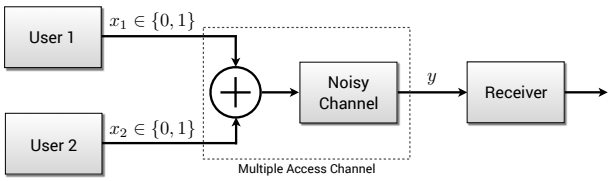


Figure 11.7

so that the decoder produces two estimates $\widehat{w}_1$ and $\widehat{w}_2$.

The average probability of decoder error is:

$$P_e = \frac{1}{2^{n(R_1+R_2)}} \sum_{w_1,w_2} \Pr(\widehat{w}_1 \neq w_1 \text{ or } \widehat{w}_2 \neq w_2 | w_1, w_2 \text{ sent}). \tag{11.32}$$

Both estimates $\widehat{w}_1$ and $\widehat{w}_2$ must be correct.

Definition 11.4. A rate pair (R_1, R_2) is *achievable* for the discrete-memoryless MAC channel if there exists a sequence of codes with $P_e \to 0$ as $n \to \infty$. The *capacity region* of the multiple-access channel is the closure of the set of achievable rate pairs (R_1, R_2).

11.4.2 MAC Channel Naive Approach

A naive approach for MAC coding is to allow only one user to transmit. Let User 2 transmit at rate $R_2 = 0$ by choosing a fixed symbol x_2 from its transmit alphabet $\mathcal{X}_2$. User 1 transmits at full rate, using a capacity-achieving code for the point-to-point channel. Then the rate pair:

$$(R_1, R_2) = (\max I(\mathsf{X}_1; \mathsf{Y} | \mathsf{X}_2 = x_2), 0) \tag{11.33}$$

is possible, shown as point (a) in Fig. 11.5. User 2 chooses the x_2 from $\mathcal{X}_2$ which maximizes the rate for User 1. Likewise, by reversing the roles of User 1 and User 2, then the rate pair:

$$(R_1, R_2) = (0, \max I(\mathsf{X}_2; \mathsf{Y} | \mathsf{X}_1 = x_1)) \tag{11.34}$$

is achievable, shown as point (b). By using time-sharing arguments, and point along the line connecting (a) and (d) is also possible.

Now consider a MAC channel with noise shown in Fig. 11.7. In this channel, the sum of the two users signals is passed through a noisy channel. Since this channel has noise, that each user has needs a long $n \to \infty$ capacity-achieving code for reliable communications.

11.4.3 MAC Channel Capacity

What is surprising, is that the MAC channel capacity is higher than this naive scheme, as the following proposition shows.

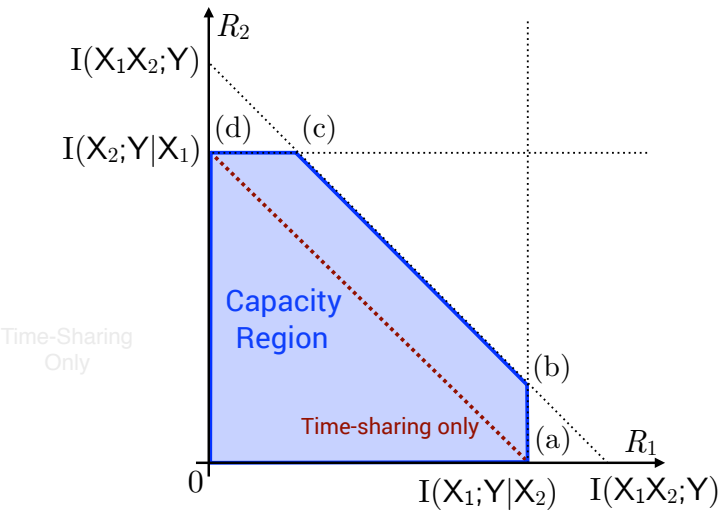


Figure 11.8: Pentagonal outer bound on the achievable rate region for the multiple-access channel.

Proposition 11.2. *Multiple-Access Channel (MAC) Capacity* The capacity of the multiple access channel is the closure of the rates convex hull of all (R_1, R_2) satisfying:

$$R_1 < I(\mathsf{X}_1;\mathsf{Y}|\mathsf{X}_2) \tag{11.35}$$
$$R_2 < I(\mathsf{X}_2;\mathsf{Y}|\mathsf{X}_1) \tag{11.36}$$
$$R_1 + R_2 < I(\mathsf{X}_1\mathsf{X}_2;\mathsf{Y}) \tag{11.37}$$

for some choice of the product distribution $p_{\mathsf{X}_1}(x_1)p_{\mathsf{X}_2}(x_2)$

The general form of the MAC capacity region is a pentagon, and is illustrated in Fig. 11.8. For some channels, particularly noiseless channels, this figure is exact. But for some channels, the capacity region is a subset of the pentagon, because there is no joint distribution $p_{\mathsf{X}_1}(x_1)p_{\mathsf{X}_2}(x_2)$ which achieves all the points of the capacity region, particularly near (b) and (c).

What is surprising is that User 1 may be able to transmit at full rate while User 2 also begins transmitting, as shown by point (b). Let User 2 transmit at a non-zero rate, while User 1 continues to transmit at maximum rate. User 2 sees User 1 as noise, and transmit as a point-to-point channel using a capacity-achieving code:

$$R_2 = I(\mathsf{X}_2;\mathsf{Y}) \tag{11.38}$$

Then, User 1 can transmit at rate

$$R_2 = I(\mathsf{X}_1;\mathsf{Y}|\mathsf{X}_2) \tag{11.39}$$

Points (c) can be obtained by repeating the above procedure and switching the roles of X_1 and X_2.

In addition, note that

$$I(\mathsf{X}_1, \mathsf{X}_2; \mathsf{Y}) = I(\mathsf{X}_1; \mathsf{Y}|\mathsf{X}_2) + I(\mathsf{X}_2; \mathsf{Y}) \tag{11.40}$$

and so the point (b) is the pair $\big(R_1, R_2\big) = \big(I(\mathsf{X}_1; \mathsf{Y}|\mathsf{X}_2), I(\mathsf{X}_2; \mathsf{Y})\big)$.

The capacity region is the convex hull of the achievable rate pairs, over all input distributions $p_{\mathsf{X}_1}(x_1)p_{\mathsf{X}_2}(x_2)$. In general, the input distribution that achieves the rate pair at one point of the convex hull may not be optimal at another point of the input distribution.

11.5 Multiple Access Channel Examples

11.5.1 Example — Independent BSCs

Suppose x_1 is the input to a BSC with error probability p_1 with output y_1. Likewise, x_2 is the input to a BSC with error probability p_2 with output y_2. The receiver can independently observe $y = (y_1, y_2)$. Find the MAC rate region.

$$\begin{aligned}
I(\mathsf{X}_1; \mathsf{Y}|\mathsf{X}_2) &= H(\mathsf{X}_1|\mathsf{X}_2) - H(\mathsf{X}_1|\mathsf{X}_2\mathsf{Y}_1, \mathsf{Y}_2) \\
&= H(\mathsf{X}_1) - H(\mathsf{X}_1|\mathsf{Y}_1, \mathsf{Y}_2) && \mathsf{X}_1, \mathsf{X}_2 \text{ are independent} \\
&= H(\mathsf{X}_1) - H(\mathsf{X}_1|\mathsf{Y}_1) && \mathsf{Y}_1 \text{ is independent of } \mathsf{X}_1, \mathsf{Y}_1 \\
&= I(\mathsf{X}_1; \mathsf{Y}_1) \\
&= 1 - h(p_1).
\end{aligned}$$

And then:

$$\begin{aligned}
I(\mathsf{X}_1\mathsf{X}_2; \mathsf{Y}_1, \mathsf{Y}_2) &= H(\mathsf{X}_1\mathsf{X}_2) - H(\mathsf{X}_1\mathsf{X}_2|\mathsf{Y}_1\mathsf{Y}_2) \\
&= H(\mathsf{X}_1) + H(\mathsf{X}_2) - H(\mathsf{X}_1|\mathsf{Y}_1) - H(\mathsf{X}_2|\mathsf{Y}_2) \\
&= I(\mathsf{X}_1; \mathsf{Y}_1) + I(\mathsf{X}_2; \mathsf{Y}_2)
\end{aligned}$$

11.5.2 Example — Binary erasure MAC

The *binary erasure multiple-access channel* has inputs $\mathcal{X}_1 = \mathcal{X}_2 = \{0, 1\}$. The channel output is the real addition:

$$\mathsf{Y} = \mathsf{X}_1 + \mathsf{X}_2, \tag{11.41}$$

so $\mathcal{Y} = \{0, 1, 2\}$.

Point (a) If we set $\mathsf{X}_2 = 0$, then $\mathsf{X}_1 = \mathsf{Y}$. With $p_{\mathsf{X}_1}(x) = [\frac{1}{2}, \frac{1}{2}]$ we have $I(\mathsf{X}_1; \mathsf{Y}|\mathsf{X}_2) = 1$, so $R_1 = 1$, and $(R_1, R_2) = (1, 0)$ is achievable. **Point (d)** By symmetry, $(R_1, R_2) = (0, 1)$ is also achievable.

Point (b) Assume User 1 transmits uncoded data with input distribution $[\frac{1}{2}, \frac{1}{2}]$. For User 2 this X_1 is noise and User 2 should maximize $I(\mathsf{X}_2; \mathsf{Y})$. What channel does User 2 see? From the distribution $p_{\mathsf{Y}|\mathsf{X}_1\mathsf{X}_2}(y|x_1, x_2)$, we can find $p_{\mathsf{Y}|\mathsf{X}_2}(y|x_2)$:

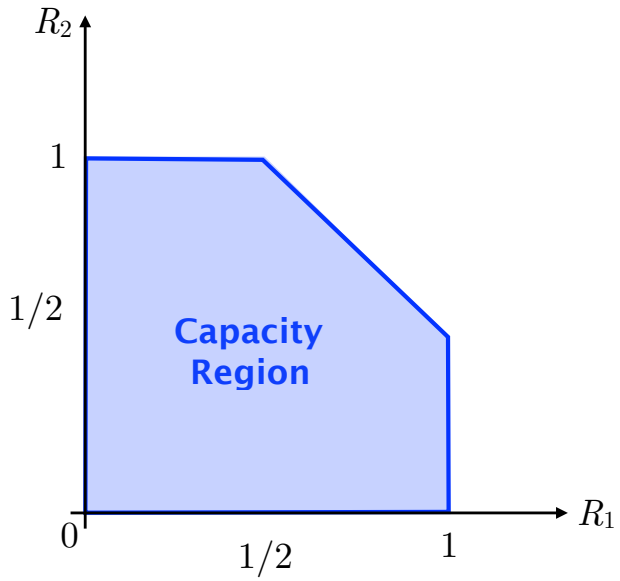


Figure 11.9: Capacity region the binary erasure multiple access channel.

	$p_{\mathsf{Y}\|\mathsf{X}_2}(y\|x_2)$		
x_2	0	1	2
0	$1/2$	$1/2$	0
1	0	$1/2$	$1/2$

For X_2, the channel looks like a binary erasure channel with erasure probably $1/2$. (Given X_2 transmits a 0, Y will be 0 with probability $1/2$. Given $\mathsf{X}_2 = 1$, $\mathsf{Y} = 2$ with probability $1/2$. Otherwise, $\mathsf{Y} = 1$ with probability $1/2$, which acts as the erasure symbol). The capacity of the binary erasure channel is $1 - \epsilon$, and here $\epsilon = \frac{1}{2}$ that means User 2 can achieve $R_2 = 1/2$. The decoder uses successive cancellation decoding to remove the interference from User 2; in this way User 1 has an error-free channel and can communicate at a rate of $R_1 = 1$. Thus, $(R_1, R_2) = (1, 1/2)$ is achievable.

The decoding for this case can be achieved using successive cancellation decoding, illustrated in Fig. 11.10. The receiver first decodes the $R_1 = 1/2$ code used by User 1. Having obtained the correct codeword $\widehat{\mathbf{x}}_1$, this is subtracted from $\mathbf{y}$ to obtain the sequence for User 2 $\widehat{\mathbf{x}}_2 = \mathbf{y} - \widehat{x}_1$.

Point (c) User 2 can transmit at full rate $R_2 = 1$ with uncoded data, and User 1 can use a code for the binary erasure channel. By symmetry, this achieves the rate pair $(R_1, R_2) = (1/2, 1)$.

This achievable rate region for the binary erasure MAC is shown in Fig. 11.9.

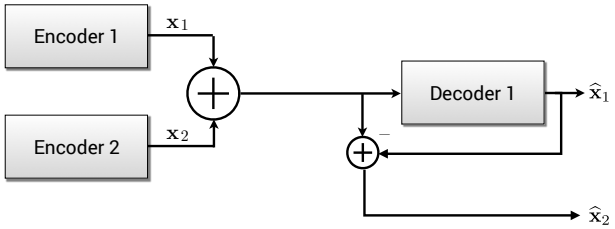


Figure 11.10: Successive cancelation decoder for the binary erasure channel.

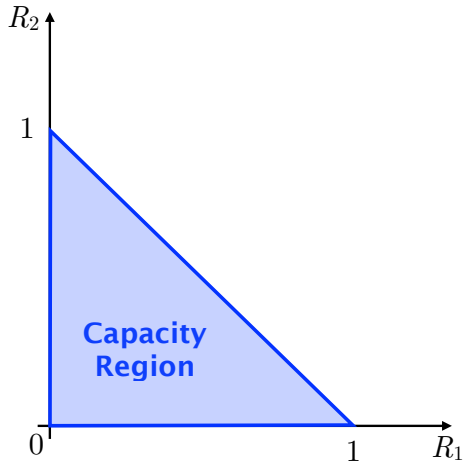


Figure 11.11: Capacity region of binary multiplier channel, with $\mathsf{Y} = \mathsf{X}_1 \cdot \mathsf{X}_2$.

11.5.3 MAC Example — Binary Multiplier Channel

The *binary multiplier multiple-access channel* has inputs $\mathcal{X}_1 = \mathcal{X}_2 = \{0, 1\}$. The channel output is real multiplication:

$$\mathsf{Y} = \mathsf{X}_1 \cdot \mathsf{X}_2, \tag{11.42}$$

so $\mathcal{Y} = \{0, 1\}$.

If we set $\mathsf{X}_2 = 1$, then $\mathsf{X}_1 = \mathsf{Y}$. With $p_{\mathsf{X}_1}(x) = [\frac{1}{2}, \frac{1}{2}]$ we have $I(\mathsf{X}_1; \mathsf{Y}|\mathsf{X}_2) = 1$, so $R_1 = 1$, and $(R_1, R_2) = (1, 0)$ is achievable. By symmetry, $(R_1, R_2) = (0, 1)$ is also achievable.

Since Y is binary, we have $I(\mathsf{X}_1\mathsf{X}_2; \mathsf{Y}) \leq \log |\mathcal{Y}| = 1$. So we have $R_1 + R_2 < 1$. The capacity region is shown in Fig. 11.11. Points along $R_1 + R_2 = 1$ can be achieved using time-sharing between $(0, 1)$ and $(1, 0)$. Since time-sharing is a naive approach, achievable rates on the multiplier channel cannot be improved using clever coding techniques.

11.6 Exercises

11.1 Let X and Y be jointly distributed as follows:

$p_{\mathsf{XY}}(x,y)$	$x=1$	$x=2$	$x=3$	$x=4$	$x=5$
$y=1$	0	0.05	0.05	0.05	0.05
$y=2$	0.05	0	0.05	0.05	0.05
$y=3$	0.05	0.05	0	0.05	0.05
$y=4$	0.05	0.05	0.05	0	0.05
$y=5$	0.05	0.05	0.05	0.05	0

Assume that an encoder knows both x and y, and a decoder already knows y. The encoder wants to transmit the value of x to the decoder. What is the most efficient binary code for the encoder? What operation should the decoder perform? What is the expected length of this code?

11.2 *Slepian-Wolf 1* Let X_i and Z_i be independent random variables with $p_{\mathsf{X}}(x) = [1-p, p]$ and $p_{\mathsf{Z}}(z) = [1-r, r]$ for $0 \leq p, r \leq 1$. Let $\mathsf{Y}_i = \mathsf{X}_i \oplus \mathsf{Z}_i$ where $\oplus$ denotes addition modulo 2. Let the source vector $\mathbf{X} = (\mathsf{X}_1, \ldots, \mathsf{X}_n)$ be encoded at rate R_1 and let $\mathbf{Y} = (\mathsf{Y}_1, \ldots, \mathsf{Y}_n)$ be encoded at rate R_2.

(a) What is $p_{\mathsf{Y}}(y)$?
(b) What region of rates allows recovering of $\mathbf{X}, \mathbf{Y}$ with probability of error tending to zero as $n \to \infty$? Draw a pentagonal region and label the key points.
(c) On the same figure, draw the region of achievable rates, assuming the correlation between X and Y is ignored.

11.3 *Multiple access channel 1* Consider a multiple access channel with inputs $\mathcal{X}_1 = \{0, 1, 2, 3\}$ and $\mathcal{X}_2 = \{0, 1\}$ The channel is given by:

$$\mathsf{Y} = \mathsf{X}_1 + \mathsf{X}_2 \bmod 4 \tag{11.54}$$

Find the capacity region for this channel.

11.4 *Mod-3 multiple access channel* Consider a two-user multiple access channel with inputs $\mathcal{X}_1 = \mathcal{X}_2 = \{0, 1, 2\}$. The user rates are R_1 and R_2. The channel is given by:

$$\mathsf{Y} = \mathsf{X}_1 + \mathsf{X}_2 \bmod 3. \tag{11.66}$$

Find the capacity region for this multiple access channel. Draw the capacity region, and show numerical values for key points.

11.5 Let W and Z be independent random variables with:

$$p_{\mathsf{W}}(w) = [1-p, p] \text{ and } p_{\mathsf{Z}}(z) = [1-p, p] \tag{11.79}$$

Let two new random variables be:

$$\mathsf{X} = \mathsf{WZ} \text{ and} \tag{11.80}$$
$$\mathsf{Y} = \mathsf{W} + \mathsf{Z}. \tag{11.81}$$

This induces a distribution on X and Y. Let (x_i, y_i) be a sequence of random variables from this distribution. User 1 sends $\mathbf{x} = (x_1, \ldots, x_n)$ with rate R_1. User 2 sends $\mathbf{y} = (y_1, \ldots, y_n)$ with rate R_2.

(a) Draw the pentagonal Slepian-Wolf rate region for recovering $\mathbf{x}$ and $\mathbf{y}$ at the decoder.
(b) Draw the rate region, and label all the intersections.
(c) After successfully decoding $\widehat{\mathbf{x}}$ and $\widehat{\mathbf{y}}$, what uncertainty does the decoder have about (W, Z)?

11.6 Consider Slepian-Wolf coding for length $n = 7$ sources $\mathbf{x}$ and $\mathbf{y}$ which differ in at most $t = 1$ positions. User 1 sends $\mathbf{x}$ uncompressed and User 2 compresses $\mathbf{y}$ using the following parity-check matrix:

$$\mathbf{H} = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}. \tag{11.97}$$

The corresponding syndrome table is:

$\mathbf{s}$	$\mathbf{e} = \phi(\mathbf{s})$
000	0000000
111	1000000
110	0100000
101	0010000
011	0001000
100	0000100
010	0000010
001	0000001

(a) Suppose the source sequences are $\mathbf{x} = (1, 1, 1, 1, 1, 1, 1)$ and $\mathbf{y} = (1, 1, 1, 1, 1, 1, 0)$. What are the transmitted messages, assuming Encoder 1 does not compress and Encoder 2 compresses?
(b) Suppose the decoder receives (1010011) and $\mathbf{s}_2 = (010)$. What are the decoded messages $\widehat{\mathbf{x}}$ and $\widehat{\mathbf{y}}$?
(c) What is the rate pair (R_1, R_2) for this code?
(d) Assume instead that x, y are jointly distributed as:

$$p_{\mathsf{XY}}(x, y) = \begin{bmatrix} \frac{6}{14} & \frac{1}{14} \\ \frac{1}{14} & \frac{6}{14} \end{bmatrix} \tag{11.98}$$

which corresponds to x and y differing with probability $\frac{1}{7}$. Plot the Slepian-Wolf rate region for this source. On the same graph, plot your answer from part ((c)). You will find the rate pair for the $n = 7$ code is outside of the Slepian-Wolf region. Why?

Chapter 12

Network Information Theory, Part 2

In the networks studied thus far, each node in the network was either a transmitter or a receiver, all were "single hop" networks. In this section, nodes are allowed to both transit and receive, forming a "multi hop" network. Some nodes are information sources, meaning they have information. Some nodes are information destinations, meaning the want information.

For single-hop networks, the information capacity can be derived, and we did so in many cases. But for multi hop networks, quite often the best we can do is find either an upper bound on the capacity, or find an achievable rate, which is a lower bound on the capacity.

12.1 Graphical Networks

A network consists of nodes and connections between those nodes. Source nodes have messages. Destination nodes want those messages.

We start by studying graphical networks, where nodes are connected by a noiseless link of some capacity. There is no noise, and no interference from the other nodes. Later, we study general multitermal networks, where nodes are connected by a conditional probability distribution, which allow for noise and interference.

A graphical network consists of:

- a set of nodes $\mathcal{N}$,
- a set of directed edges $\mathcal{E}$ that connect pairs of nodes: $\mathcal{E} \subseteq \mathcal{N} \times \mathcal{N}$,
- a capacity $C_{i \to j}$ for the directed edge from node i to node j,

Additional reading: El Gamal and Kim, Chapter 15. Cover and Thomas, Section 15.10.

- one or more source nodes that have information, and
- one or more destination nodes that want information; the set of destination nodes is $\mathcal{D} \subset \mathcal{N}$.

A graphical network is expressed using a directed acyclic graph (DAG) with capacities assigned to each edge. Wikipedia: Directed Acyclic Graph. A node which has both inputs and outputs is called a *relay node*.

The network is *delay free*, which means that all transmissions happen instantaneously and simultaneously. The network is error-free, the received symbol is always equal to transmitted symbol.

A source node produces a bit stream which is divided into vectors of m bits. Each vector is represented by an element from a finite field with $q = 2^m$ elements:

$$\mathcal{W} = \{0, 1, 2, \ldots, 2^m - 1\} \tag{12.1}$$

This finite field is denoted $\mathbb{F}_q$. So we view the source as producing a sequence of finite-field symbols, and we can work on the finite field. Because it is a field, we can perform addition, subtraction, multiplication and division on elements $w \in \mathcal{W}$. The source may send multiple messages $w_1, w_2, \ldots$. A link with capacity $1, 2, 3, \ldots$ can carry $1, 2, 3, \ldots$, messages from this alphabet.

We consider two types of operations that nodes can perform:

- In *routing*, the node selects from its inputs and send these as outputs, without changing its inputs. For example, for a node has two input edges and one output edge, all with capacity one, if the inputs are w_1 and w_2, then the output may be w_1 (or it may be w_2)
- In *network coding* the relay forms output messages from combinations of inputs messages. Using the same example, the output may be w_1, or w_2, or some linear combination such as $w_1 \oplus w_2$, where $\oplus$ is addition in the finite field $\mathbb{F}_q$.

There is set of source nodes which have information. There is a set of destination nodes denoted $\mathcal{D}$. Networks differ depending on the number of source nodes and number of destination nodes:

- *Unicast network* A unicast network has one source node, assumed to be node 1, and one destination node $d \in \mathcal{N}$. All the other nodes are called relays which can aid in transmission. Nodes in the network that have both inputs and outputs are called relays. First routing is considered — a relay has one or more inputs, and the relay selects one of its inputs to transmit on each of its outputs. The capacity of a unicast network is the maximum of the number messages that can be sent from the source to all of the destinations.
- *Multicast network* We consider multicast networks which have one source node and multiple destination nodes. This can be generalized to multiple source nodes.

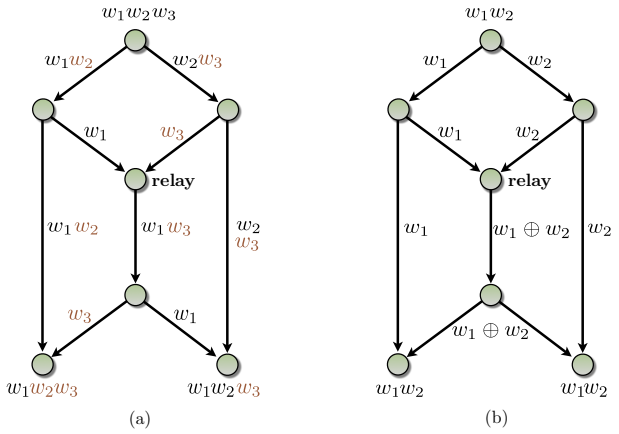


Figure 12.1: (a) Butterfly network with routing, capacity is 3/2. (b) Butterfly network with network coding, capacity is 2.

12.2 Capacity of Unicast Network

The capacity of the unicast network can be found by the max-flow, min-cut theorem, and the capacity can be achieved by routing.

Definition 12.1. For a source and destination, a *cut* $(\mathcal{S}, \mathcal{S}^c)$ partitions the nodes $\mathcal{N}$ so that the source is in $\mathcal{S}$ and the destination is in $\mathcal{S}^c$.

The edges crossing the cut are

$$\{(k, \ell) \in \mathcal{E}, |k \in \mathcal{S}, \ell \in \mathcal{S}^c\} \tag{12.2}$$

Definition 12.2. The *capacity of a cut* is:

$$C(\mathcal{S}) = \sum_{(k,\ell)\in\mathcal{E}, k\in\mathcal{S}, \ell\in\mathcal{S}^c} C_{k\to\ell}$$

Proposition 12.1. *Max-Flow Min-Cut* The capacity C of a network with one source (node $1 \in \mathcal{S}$) and one destination (node $d \in \mathcal{S}^c$) is:

$$C = \min_{\mathcal{S}\subset\mathcal{N}, 1\in\mathcal{S}, d\in\mathcal{S}^c} C(\mathcal{S})$$

This capacity can be achieved by routing.

12.3 Capacity of Multicast Networks

A multicast network has one source and multiple destinations. The destination set is $\mathcal{D} \subseteq \mathcal{N}$. Destinations want the same message. The following gives an upper bound on the capacity.

Proposition 12.2. *Cut-Set Bound* The capacity C of a network with one source (node $1 \in \mathcal{S}$) and destination set $\mathcal{D}$ is upper bounded as:

$$C \leq \min_{j \in \mathcal{D}} \min_{\mathcal{S} \subset \mathcal{N}, 1 \in \mathcal{S}, j \in \mathcal{S}^c} C(\mathcal{S})$$

The capacity from the source to one destination j is given by max-flow min-cut. Then, the capacity for the multicast network is the capacity of the worst link — the minimum over all the max-flow min-cuts.

The cut-set bound is an upper bound on the capacity. In general, the capacity cannot be achieved by routing, as is shown in the following example.

The example of the butterfly network with routing is shown in Fig. 12.1-(a). For this network, the cut-set bound is 2, but the highest rate achievable by routing is 3/2.

Fig. 12.1-(b) also shows the butterfly network using network coding, where the relay node is allowed to compute the sum, over a finite field, of its two inputs. The destination receives w_1 and $w_1 \oplus w_2$. The destination can recover the original messages by computing $w_2 = (w_1 \oplus w_2) \ominus w_1$. Assuming that the binary field is used, this operation can be written in matrix form:

$$\begin{bmatrix} w_1 \\ w_1 \oplus w_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \tag{12.3}$$

Because the matrix is invertible over the binary field, the solution is easily obtained:

$$\begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}^{-1} \cdot \begin{bmatrix} w_1 \\ w_1 \oplus w_2 \end{bmatrix} = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \tag{12.4}$$

The following theorem states that the cut set bound is indeed the capacity of the multicast network. Routing does not achieve capacity, but network coding can.

Proposition 12.3. *Network Coding Theorem* The capacity C of the graphical multicast network with destination set $\mathcal{D}$ is:

$$C = \min_{j \in \mathcal{D}} \min_{\mathcal{S} \subset \mathcal{N}, 1 \in \mathcal{S}, j \in \mathcal{S}^c} C(\mathcal{S})$$

To illustrate the principles of network coding, consider the network shown in Fig. 12.2. The figure shows the linear functions computed by each node. The messages are from a finite field with 16 elements $\{0, 1, \alpha, \alpha^2, \ldots, \alpha^{14}\}$. m_{ij} is the message from node i to node j.

A linear network code with $R = 2$ induces the linear transformation: The first stage is:

$$\begin{bmatrix} w_{1\to2} \\ w_{1\to3} \end{bmatrix} = \begin{bmatrix} a_1 & a_2 \\ a_3 & a_4 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \tag{12.5}$$

The second stage is:

$$\begin{bmatrix} w_{2\to4} \\ w_{3\to4} \end{bmatrix} = \begin{bmatrix} a_6 & 0 \\ a_5 a_8 & a_7 \end{bmatrix} \begin{bmatrix} w_{1\to2} \\ w_{1\to3} \end{bmatrix} \tag{12.6}$$

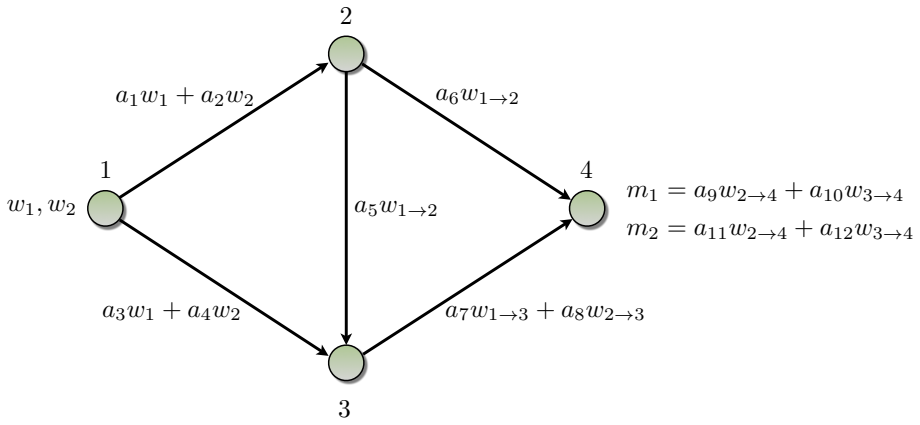


Figure 12.2: Linear network code for example.

Taken altogether:

$$\begin{bmatrix} s_1 \\ s_2 \end{bmatrix} = \begin{bmatrix} \alpha_9 & \alpha_{10} \\ \alpha_{11} & \alpha_{12} \end{bmatrix} \cdot \begin{bmatrix} \alpha_6 & 0 \\ \alpha_5\alpha_8 & \alpha_7 \end{bmatrix} \cdot \begin{bmatrix} \alpha_1 & \alpha_2 \\ \alpha_3 & \alpha_4 \end{bmatrix} \cdot \begin{bmatrix} M_1 \\ M_2 \end{bmatrix} \tag{12.7}$$

$$= A \cdot \begin{bmatrix} M_1 \\ M_2 \end{bmatrix} \tag{12.8}$$

If A is invertible in the field, then:

$$A^{-1} \cdot \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} = \begin{bmatrix} M_1 \\ M_2 \end{bmatrix} \tag{12.9}$$

the original messages can be recovered.

12.4 General Networks

12.4.1 Generalized Cut-Set Bound

Now we consider a general information network of N nodes $\mathcal{N}$. At each time index, node i has an input x_i and output y_i. The channel is represented by the conditional probability distribution:

$$p_{\mathbf{Y}|\mathbf{X}}(y^{(1)}, \ldots, y^{(N)}|x^{(1)}, \ldots, x^{(N)}), \tag{12.10}$$

and the channel uses $1, 2, \ldots, n$ are iid. This function allows for noise and interference in the network.

The message sent from node i to node j is $w^{(ij)}$ and the corresponding information rate is $R^{(ij)}$, so the message is from the set $\{1, 2, \ldots, 2^{nR(i,j)}\}$ The channel input at node i is $x^{(i)}$ which depends on both all its message $w^{(i1)}, \ldots, w^{(iN)}$, and all the past values of the received sequences at $y^{(i)}$. Node i transmits this sequence $x_1^{(i)}, x_2^{(i)}, \ldots, x_n^{(i)}$. The input distribution is $p_{\mathbf{X}}(\mathbf{x})$.

For decoding, node j decodes the message received from node i, by mapping the received sequences $(y_1^{(j)}, \ldots, y_n^{(j)})$ to the estimated message $\widehat{w}^{(ij)}$.

The probability of error for the message $w^{(ij)}$ is:

$$P_e^{(ij)} = \Pr(w^{(ij)} \neq \widehat{w}^{(ij)}) \tag{12.11}$$

A set of rates $R^{(ij)}$ is said to be achievable if there exist encoders and decoder such that $P_e^{(ij)} \to 0$ and $n \to \infty$ for all $i, j \in \mathcal{N}$.

Now we form a cut, $(\mathcal{S}, \mathcal{S}^c)$, and make a statement about the information flow across the cut.

Proposition 12.4. *Cut-Set Bound on Information Flow* If the information rates $R^{(ij)}$ are achievable, then there exists some joint probability distribution $p_{\mathbf{X}}(\mathbf{x})$ such that

$$\sum_{i \in \mathcal{S}, j \in \mathcal{S}^c} R^{(ij)} \leq I\big(\mathsf{X}^{(\mathcal{S})}; \mathsf{Y}^{(\mathcal{S}^c)} | \mathsf{X}^{(\mathcal{S}^c)}\big) \tag{12.12}$$

for all $\mathcal{S} \subset \{1, 2, \ldots, m\}$. Thus, the total rate of flow of information across cut sets is bounded by the conditional mutual information.

The proof is similar to the converse of the multiple access channel, in particular uses Fano's inequality.

Example 12.1. Illustrate the cut-set bound on information flow using the example of the MAC channel. The two transmitters are nodes 1 and 2, and the receiver is node 3.

Apply the theorem for three cut sets: $\mathcal{S}_1 = \{1\}$, $\mathcal{S}_2 = \{2\}$ and $\mathcal{S}_3 = \{1, 2\}$. Then, the inequalities of the theorem for each case are:

$$R^{(12)} + R^{(13)} \leq I(\mathsf{X}_1; \mathsf{Y}_2, \mathsf{Y}_3 | \mathsf{X}_2, \mathsf{X}_3) \tag{12.13}$$

$$R^{(21)} + R^{(23)} \leq I(\mathsf{X}_2; \mathsf{Y}_1, \mathsf{Y}_3 | \mathsf{X}_1, \mathsf{X}_3) \tag{12.14}$$

$$R^{(13)} + R^{(23)} \leq I(\mathsf{X}_1 \mathsf{X}_2; \mathsf{Y}_3 | \mathsf{X}_3) \tag{12.15}$$

Two messages of interest are $w^{(13)}$ and $w^{(23)}$, so rates $R^{(13)}$ and $R^{(23)}$ may be non-zero. But the two users do not communicate with each other, so $R^{(12)} = R^{(21)} = 0$. Similarly, nodes 1 and 2 have nothing to receive and node 3 has nothing to transmit, so y_1, y_2 and x_3 are all zeros and can be ignored. Then the inequalities become:

$$R^{(13)} \leq I(\mathsf{X}_1; \mathsf{Y}_3 | \mathsf{X}_2) \tag{12.16}$$

$$R^{(23)} \leq I(\mathsf{X}_2; \mathsf{Y}_3 | \mathsf{X}_1) \tag{12.17}$$

$$R^{(13)} + R^{(23)} \leq I(\mathsf{X}_1 \mathsf{X}_2; \mathsf{Y}_3), \tag{12.18}$$

which are identical to the inequalities of the MAC capacity region.

In this case, the cut set bound on information flow gave the capacity exactly. But in general, this bound does not give the capacity, and may not be tight.

12.4.2 Relay Channel

The relay channel has one transmitter with a message w to send to a receiver. In addition, an intermediate node called a relay receives the signal from the transmitter, and can help with communication. The transmitter, relay and receiver are node 1,2 and 3, respectively. The transmitter has an output x_1, the relay has input y_2 and output x_2 and the receiver has input y_3. The memoryless channel is characterized by the conditional probability distribution

$$p_{\mathsf{Y}_2\mathsf{Y}_3|\mathsf{X}_1\mathsf{X}_2}(y_2y_3|x_1x_2). \tag{12.19}$$

Except in some special cases, the capacity of this channel is not known. An upper bound on the capacity can be computed using the cut-set bound for information flow. In addition, the three nodes can use various encoding and decoding strategies, which lead to differing achievable rates. Here only a couple of strategies are mentioned.

Achievable rates — No relay The most basic strategy is to not use the relay at all, in which case the relay channel becomes a point-to-point channel. This forms a lower bound on the performance of the relay.

Achievable rates — Transmitter as noise The receiver can decode the signal from the relay, treating the signal from the transmitter as noise.

Upper bound on achievable rate We use max-flow, min-cut analysis to form an upper bound on the relay channel capacity.

$$R^{(12)} + R^{(13)} \leq I(\mathsf{X}_1; \mathsf{Y}_2, \mathsf{Y}_3|\mathsf{X}_2, \mathsf{X}_3) \qquad \text{for } \mathcal{S}_1 \tag{12.20}$$

$$R^{(23)} + R^{(13)} \leq I(\mathsf{X}_1\mathsf{X}_2; \mathsf{Y}_3|\mathsf{X}_3) \qquad \text{for } \mathcal{S}_2. \tag{12.21}$$

Node 3 has nothing to transmit, so we can drop X_3. We are only interested in $R^{(13)}$ so can take $R^{(12)} = R^{(23)} = 0$.

$$R^{(12)} + R^{(13)} \leq I(\mathsf{X}_1; \mathsf{Y}_2, \mathsf{Y}_3|\mathsf{X}_2) \qquad \text{for } \mathcal{S}_1 \tag{12.22}$$

$$R^{(23)} + R^{(13)} \leq I(\mathsf{X}_1\mathsf{X}_2; \mathsf{Y}_3) \qquad \text{for } \mathcal{S}_2. \tag{12.23}$$

Chapter 13

Optimization in Information Theory

.

Previous chapters showed that the channel capacity and rate-distortion function are found by:

$$C = \max_{p_{\mathsf{X}}(x)} I(\mathsf{X};\mathsf{Y}) \text{ and} \tag{13.1}$$

$$R(D) = \min_{p_{\widehat{\mathsf{X}}|\mathsf{X}}(\widehat{x}|x):Ed\leq D} I(\mathsf{X};\widehat{\mathsf{X}}), \tag{13.2}$$

respectively. In a few special cases C and $R(D)$ can be represented in simple analytic form, such as the binary symmetric channel with error probability p, where $C = 1 - h(p)$. But given an arbitrary channel or source, numerical methods must be used in order to find C or $R(D)$.

The above two problems are convex optimization problems. This chapter describes a numerical technique to compute the capacity C of a discrete memoryless channel, and the related problem of finding the rate-distortion function $R(D)$ of a discrete memoryless source. This technique, sometimes called the Arimoto-Blahut algorithm, is an efficient means to find the numerical value of the channel capacity and the rate-distortion function.

13.1 Convexity of Information Measures

Maximization and minimization of the information measures of entropy, mutual information, and KL divergence play a central role in information theory. This section presents an overview of convexity and concavity, and the convexity of entropy, etc;

Additional reading: Cover and Thomas, Sections 2.6, 2.7 and 10.8. R. Yeung, Sections 10.1 and 10.2. Tishby, Pereira and Bialek "The Information Bottleneck Method," Allerton 1999.

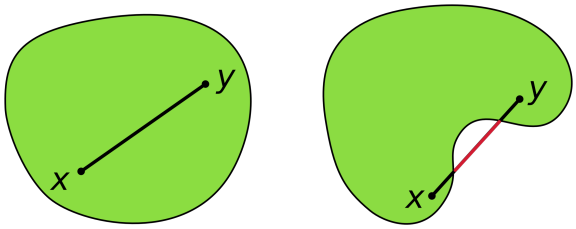


Figure 13.1: Left: a convex set. Right: a non-convex set. Image credit: Wikipedia/-Convex set.

- There are two important senses of convexity, a *convex set* and a *convex function*.
- that the KL divergence $D(\mathbf{p}||\mathbf{q})$ is a convex function of $\mathbf{p}, \mathbf{q}$,
- entropy $H(\mathsf{X})$ is a concave function of $p_{\mathsf{X}}(x)$,
- mutual information $I(\mathsf{X};\mathsf{Y})$ is a concave function of $p_{\mathsf{X}}(x)$, for fixed $p_{\mathsf{Y}|\mathsf{X}}(y|x)$ and
- mutual information $I(\mathsf{X};\mathsf{Y})$ is a convex function of $p_{\mathsf{Y}|\mathsf{X}}(y|x)$, for fixed $p_{\mathsf{X}}(x)$.

13.1.1 Convex Sets and Convex Functions

Definition 13.1. Let $\mathcal{D}$ be a subset of $\mathbb{R}^n$. Then $\mathcal{D}$ is a *convex set* if for all $\mathbf{x}, \mathbf{y} \in \mathcal{D}$:

$$\lambda\mathbf{x} + (1-\lambda)\mathbf{y} \in \mathcal{D}, \tag{13.3}$$

for all $0 \leq \lambda \leq 1$.

A set which is not convex is called non-convex. Fig. 13.1 shows examples of a convex set and a non-convex set. Wikipedia: Convex Set

It is important to distinguish between convex sets and convex functions, and the rest of the chapter concentrates on convex functions.

A real-valued function $f(\cdot)$ defined on an interval is called convex (or convex downward or lower convex) if the line segment between any two points $f(\mathbf{p})$ and $f(\mathbf{q})$ on the graph of the function lies above the graph. Examples of convex functions are the quadratic function $f(x) = x^2$ and the exponential function $f(x) = e^x$, for $x \in \mathbb{R}$. A convex function is illustrated in Fig. 13.2. Wikipedia: Convex Function

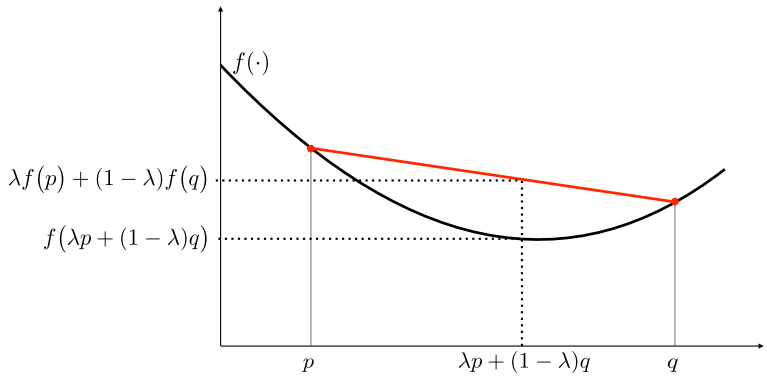


Figure 13.2: A convex function.

Definition 13.2. Let f be a real-valued function with domain $\mathcal{D} \subseteq \mathbb{R}^n$. The function f is a *convex function* if for any, $\mathbf{p}, \mathbf{q} \in \mathcal{D}$:

$$f\big(\lambda \mathbf{p} + (1-\lambda)\mathbf{q}\big) \leq \lambda f(\mathbf{p}) + (1-\lambda) f(\mathbf{q}), \tag{13.4}$$

for $0 \leq \lambda \leq 1$.

The function is called *strictly convex* if

$$f\big(\lambda \mathbf{p} + (1-\lambda)\mathbf{q}\big) < \lambda f(\mathbf{p}) + (1-\lambda) f(\mathbf{q}), \tag{13.5}$$

for any $0 \leq \lambda \leq 1$ and $\mathbf{p} \neq \mathbf{q}$. A function f is said to be (strictly) concave if $-f$ is (strictly) convex.

Proposition 13.1. *Jensen's Inequality* If $f(x)$ is a convex function, and X is a random variable, then:

$$E\big[f(\mathsf{X})\big] \geq f\big(E[\mathsf{X}]\big) \tag{13.6}$$

Wikipedia: Jensen's Inequality

The random variable X is distributed as $p_{\mathsf{X}}(x)$, so Jensen's inequality can be written as:

Alternatively:

$$\sum_{x \in \mathcal{X}} f(x) p_{\mathsf{X}}(x) \geq f\Big(\sum_{x \in \mathcal{X}} x p_{\mathsf{X}}(x) \Big).$$

Equality holds if and only if f is linear. The inequality is switched if f is a concave function.

Two tests for convexity of a function are described. A function f is convex if it can be shown to satisfy (13.4). Additionally, the second-derivative of f can be tested. A function $f(x)$ is convex on $\mathcal{D}$ if the following condition holds:

$$\frac{d^2}{dx^2} f(x) \geq 0 \text{ for all } \; x \in \mathcal{D}, \tag{13.7}$$

then $f(x)$ is convex on $\mathcal{D}$. The second derivative is also written $f''(x)$.

Example 13.1. Let $f(x) = x^3$. Here, $f''(x) = 6x$, and so f is convex on the interval $x \geq 0$.

A useful property of convexity is that affine transformations preserve convexity.

Proposition 13.2. Let $f(\mathbf{x})$ be convex in domain $\mathcal{D}_f \subseteq \mathbb{R}^m$, and define

$$g(\mathbf{x}) = f(\mathbf{A}\mathbf{x} + \mathbf{b}), \tag{13.8}$$

where $\mathbf{A}$ is an $n \times m$ matrix and $\mathbf{b}$ is an $m \times 1$ vector, with domain $\mathcal{D}_g \subseteq \mathbb{R}^n$. Then, $g(\mathbf{x})$ is also convex.

Proposition 13.3. *Non-negativity of KL divergence* For any two distributions $p(x)$ and $q(x)$ on a common set $\mathcal{X}$, the KL divergence satisfies:

$$D\big(p(x)||q(x)\big) \geq 0 \tag{13.9}$$

Proof Let $\mathcal{A} = \{x \mid p(x) > 0\}$. If $q(x') > 0$ and $p(x') = 0$, then x' is in $\mathcal{X}$ but not in $\mathcal{A}$.

$$\begin{aligned}
-D\big(p(x)||q(q)\big) &= \sum_{x\in\mathcal{A}} p(x) \log \frac{q(x)}{p(x)} && \text{Definition of KL divergence} && (13.10)\\
&\leq \log \sum_{x\in\mathcal{A}} p(x) \frac{q(x)}{p(x)} && \text{Jensen's equality} && (13.11)\\
&= \log \sum_{x\in\mathcal{A}} q(x) && && (13.12)\\
&\leq \log \sum_{x\in\mathcal{X}} q(x) && \text{not all } x \text{ are in } \mathcal{A} && (13.13)\\
&\leq \log 1 = 0 && && (13.14)\\
D\big(p(x)||q(q)\big) &\geq 0. && && \square
\end{aligned}$$

13.1.2 KL divergence Is Convex

The KL divergence $D(\mathbf{p}||\mathbf{q})$ is expressed using vectors:

$$\mathbf{p} = (p_1, p_2, \ldots, p_n) \tag{13.15}$$
$$\mathbf{q} = (q_1, q_2, \ldots, p_n). \tag{13.16}$$

Then KL divergence is convex in the following sense.

Proposition 13.4. *Convexity of KL divergence.* $D(\mathbf{p}||\mathbf{q})$ is a convex in the pair of distributions $(\mathbf{p}, \mathbf{q})$.

The proof is given in Subsection 13.2.2.

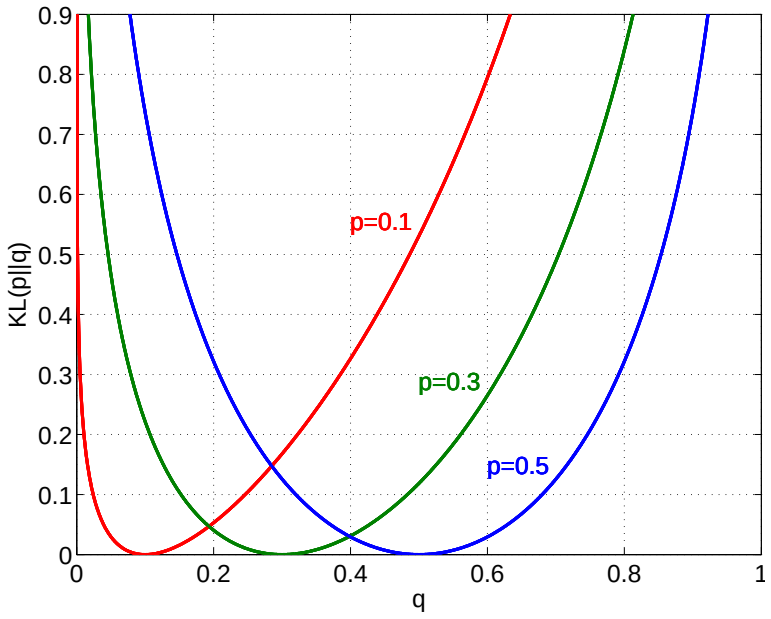


Figure 13.3: Convexity of KL divergence.

Example 13.2. The convexity of KL divergence is illustrated for $n = 2$ using parameters p and q. Let $\mathbf{p} = [1-p, p]$ and $\mathbf{q} = [1-q, q]$. The KL divergence is:

$$D(\mathbf{p}||\mathbf{q}) = p \log \frac{p}{q} + (1-p) \log \frac{1-p}{1-q}, \tag{13.17}$$

and this function of p and q is shown in Fig. 13.3. The KL divergence is seen to be convex in q, and is also convex in p. Note also that the KL divergence goes to zero when $p = q$.

13.1.3 Entropy is Concave

Entropy is a concave function of the probability distribution. Recall the probability distribution can be expressed as a vector $p_{\mathsf{X}}(x) = [p_1, p_2, \ldots, p_n]$.

Proposition 13.5. *Concavity of Entropy.* $H(\mathsf{X})$ is a concave function of $p_{\mathsf{X}}(x)$.

Proof Let p_{Q} be the uniform distribution with the same cardinality as X,that is $p_{\mathsf{Q}}(q) = \frac{1}{|\mathcal{X}|}$ for $q = \{1, 2, \ldots, |\mathcal{X}|\}$. By Proposition 3.5 on page 63:

$$H(\mathsf{X}) = \log |\mathcal{X}| - D(p_{\mathsf{X}}||p_{\mathsf{Q}}). \tag{13.18}$$

Since $D(p_{\mathsf{X}}||p_{\mathsf{Q}})$ is convex in $p_{\mathsf{X}}(x)$, $-D(p_{\mathsf{X}}||p_{\mathsf{Q}})$ is concave in $p_{\mathsf{X}}(x)$ and thus $H(\mathsf{X})$ is also concave in $p_{\mathsf{X}}(x)$. □

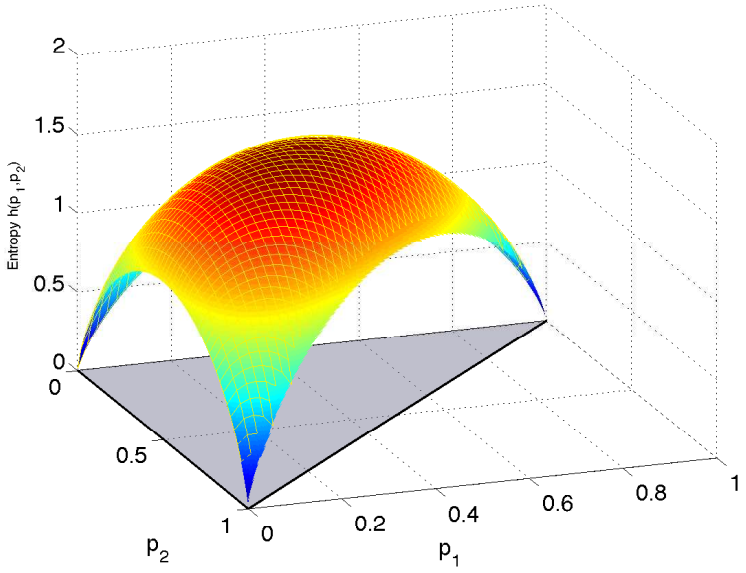


Figure 13.4: Concavity of entropy of a ternary variable.

When X is a binary random variable, concavity can be observed in the binary entropy function in Fig. 1.1 on page 18. Consider a ternary random variable X with $\mathcal{X} = \{0, 1, 2\}$ and,

$$p_{\mathsf{X}}(x) = \begin{cases} p_1 & x = 1 \\ p_2 & x = 2 \\ 1 - p_1 - p_2 & x = 3 \end{cases}.$$

Entropy $H(\mathsf{X})$ as a function of p_1 and p_2, as shown in Fig. 13.4, which is seen to be concave. Note that $p_1 + p_2 + p_3 = 1$, so $p_1 + p_2 \leq 1$. Since p_3 is dependent on p_1 and p_2, it is not shown.

13.1.4 Mutual Information is Concave in p_{X} and Convex in $p_{\mathsf{Y}|\mathsf{X}}$

Consider random variables X and Y. If the joint distribution is written as $p_{\mathsf{Y}|\mathsf{X}}(y|x)p_{\mathsf{X}}(x)$, then mutual information $I(\mathsf{X};\mathsf{Y})$ is concave in $p_{\mathsf{X}}(x)$ and convex in $p_{\mathsf{Y}|\mathsf{X}}(y|x)$.

Proposition 13.6. *Mutual Information is concave in* p_{X}. Let X and Y be random variables having a joint distribution $p_{\mathsf{XY}}(x, y) = p_{\mathsf{X}}(x)p_{\mathsf{Y}|\mathsf{X}}(y|x)$. Mutual information $I(\mathsf{X};\mathsf{Y})$ is a concave function of $p_{\mathsf{X}}(x)$ for fixed $p_{\mathsf{Y}|\mathsf{X}}(y|x)$.

Proof is given as an exercise.

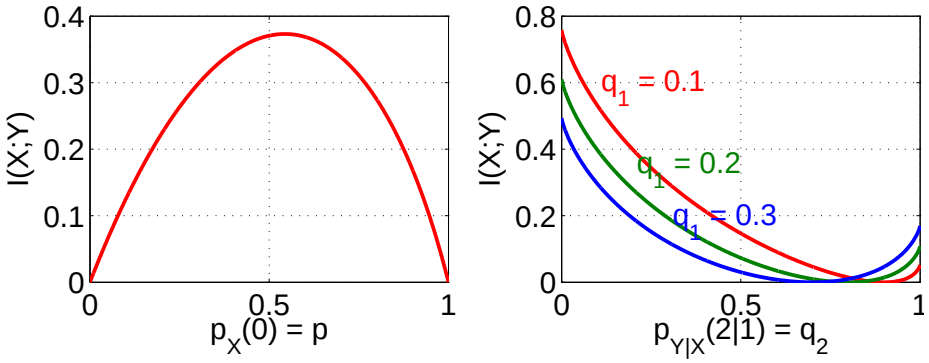


Figure 13.5: Mutual information is concave in p_{X} (left) and convex in $p_{\mathsf{Y}|\mathsf{X}}$ (right).

Proposition 13.7. *Mutual Information is convex in* $p_{\mathsf{Y}|\mathsf{X}}$. Let X and Y be random variables having a joint distribution $p_{\mathsf{XY}}(x,y) = p_{\mathsf{X}}(x)p_{\mathsf{Y}|\mathsf{X}}(y|x)$. Mutual information $I(\mathsf{X};\mathsf{Y})$ is a convex function of $p_{\mathsf{Y}|\mathsf{X}}(y|x)$ for fixed $p_{\mathsf{X}}(x)$.

Proof is given as an exercise.

Example 13.3. Consider a channel with input X and output Y. The input X has probability distribution:

$$p_{\mathsf{X}}(x) = \begin{cases} p & x = 0 \\ 1-p & x = 1 \end{cases}. \tag{13.19}$$

The channel transition probabilities $p_{\mathsf{Y}|\mathsf{X}}(x|y)$ are given by:

$$p_{\mathsf{Y}|\mathsf{X}}(y|x) = \begin{bmatrix} 1-q_1 & q_1 \\ q_2 & 1-q_2 \end{bmatrix} \tag{13.20}$$

Mutual information $I(\mathsf{X};\mathsf{Y})$ is a function of p, q_1 and q_2.

Mutual information is concave in p, for fixed q_1, q_2. This is shown in Fig. 13.5, for $q_1 = 0.05$ and $q_2 = 0.3$.

Mutual information is convex in q_1 and q_2, for fixed p. Convexity in q_2 is clear in Fig. 13.5, where $p = 0.5$ (mutual information is also convex in q_1, but this cannot be seen as easily in the figure).

13.2 Convexity of KL Divergence

This section gives the log-sum inequality, which is then used to prove the convexity of KL divergence.

13.2.1 Log-Sum Inequality

Proposition 13.8. *Log-Sum Inequality* For non-negative numbers, $a_1, a_2, \ldots, a_n$ and $b_1, b_2, \ldots, b_n$:

$$\sum_{i=1}^{n} a_i \log \frac{a_i}{b_i} \geq \left(\sum_{i=1}^{n} a_i\right) \log \frac{\sum_{i=1}^{n} a_i}{\sum_{i=1}^{n} b_i} \tag{13.21}$$

with equality if and only if a_i/b_i are equal, for all i. Wikipedia: Log-sum inequality

Proof: for $t > 0$, make the following definitions:

$$f(t) = t \log t \tag{13.22}$$

$$\alpha_i = \frac{b_i}{\sum_{j=1}^{n} b_j} \text{ and} \tag{13.23}$$

$$t_i = \frac{a_i}{b_i} \tag{13.24}$$

Note that f is strictly convex, since $f''(t) = (1/t) \log e > 0$ for all $t > 0$. We have the sequence of inequalities:

$$\begin{aligned}
\sum_{i=1}^{n} \alpha_i f(t_i) &\geq f\left(\sum_{i=1}^{n} \alpha_i t_i\right) \\
\sum_i \alpha_i t_i \log t_i &\geq \left(\sum_i \alpha_i t_i\right) \log \left(\sum_i \alpha_i t_i\right) \\
\sum_i \frac{a_i}{\sum_j b_j} \log \frac{a_i}{b_i} &\geq \left(\sum_i \frac{a_i}{\sum_j b_j}\right) \log \left(\sum_i \frac{a_i}{\sum_j b_j}\right), \\
\frac{1}{\sum_j b_j} \sum_i a_i \log \frac{a_i}{b_i} &\geq \left(\frac{1}{\sum_j b_j} \sum_i a_i\right) \log \frac{\sum_i a_i}{\sum_j b_j},
\end{aligned}$$

which gives the log-sum inequality by canceling the colored terms.

13.2.2 Proof of Convexity of KL divergence

Proof of Proposition 13.4 The proof technique is to choose an two arbitrary distinct distribution pairs $(\mathbf{p}_1, \mathbf{q}_1)$ and $(\mathbf{p}_2, \mathbf{q}_2)$ (see Fig. 13.6), and using the log-sum inequality, show that KL divergence satisfies the definition of convex functions (13.4).

Let $0 \leq \lambda \leq 1$. Expand $\lambda D(\mathbf{p}_1||\mathbf{q}_1) + (1-\lambda) D(\mathbf{p}_2||\mathbf{q}_2)$:

$$\lambda D(\mathbf{p}_1||\mathbf{q}_1) + (1-\lambda) D(\mathbf{p}_2||\mathbf{q}_2) = \sum_{x \in \mathcal{X}} \lambda p_{1,x} \log \frac{p_{1,x}}{q_{1,x}} + \sum_{x \in \mathcal{X}} (1-\lambda) p_{2,x} \log \frac{p_{2,x}}{q_{2,x}} \tag{13.25}$$

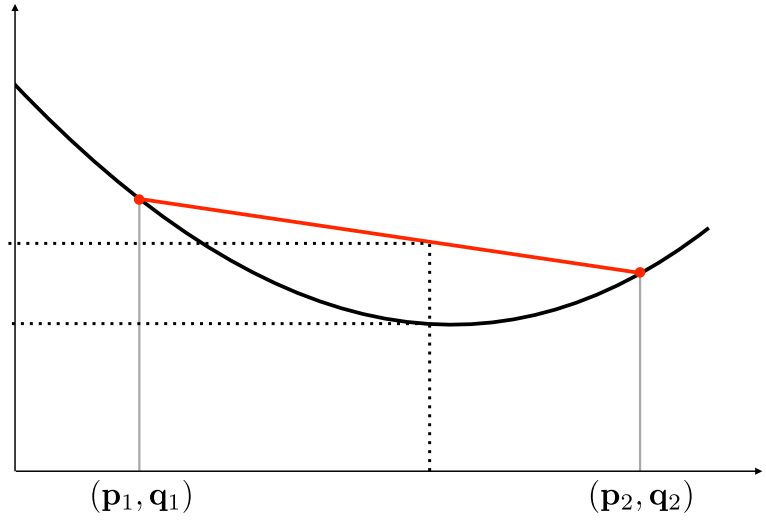


Figure 13.6: KL divergence is convex in the pair $(\mathbf{p}, \mathbf{q})$.

One term of the sum is:

$$\lambda p_{1,x} \log \frac{\lambda p_{1,x}}{\lambda q_{1,x}} + \quad (1-\lambda) p_{2,x} \log \frac{(1-\lambda) p_{2,x}}{(1-\lambda) q_{2,x}} \tag{13.26}$$

Apply the log-sum inequality with $n = 2$ ($a_1 = \lambda p_{1,x}, a_2 = (1-\lambda)p_{2,x}, b_1 = \lambda q_{1,x}, b_2 = (1-\lambda)q_{2,x}$) to just *one* term on the right-hand side of (13.26):

$$\lambda p_{1,x} \log \frac{\lambda p_{1,x}}{\lambda q_{1,x}} + \quad (1-\lambda) p_{2,x} \log \frac{(1-\lambda) p_{2,x}}{(1-\lambda) q_{2,x}} \geq \quad \big(\lambda p_{1,x} + (1-\lambda) p_{2,x}\big) \log \frac{\big(\lambda p_{1,x} + (1-\lambda) p_{2,x}\big)}{\big(\lambda q_{1,x} + (1-\lambda) q_{2,x}\big)}$$

Then sum over all $\mathcal{X}$:

$$\underbrace{\sum_{x \in \mathcal{X}} \lambda p_{1,x} \log \frac{\lambda p_{1,x}}{\lambda q_{1,x}}}_{\lambda D(\mathbf{p}_1 || \mathbf{q}_1)} + \underbrace{\sum_{x \in \mathcal{X}} (1-\lambda) p_{2,x} \log \frac{(1-\lambda) p_{2,x}}{(1-\lambda) q_{2,x}}}_{(1-\lambda) D(\mathbf{p}_2 || \mathbf{q}_2)} \geq \underbrace{\sum_{x \in \mathcal{X}} \big(\lambda p_{1,x} + (1-\lambda) p_{2,x}\big) \log \frac{\big(\lambda p_{1,x} + (1-\lambda) p_{2,x}\big)}{\big(\lambda q_{1,x} + (1-\lambda) q_{2,x}\big)}}_{D(\lambda \mathbf{p}_1 + (1-\lambda)\mathbf{p}_2 || \lambda \mathbf{q}_1 + (1-\lambda)\mathbf{q}_2)}$$

$$\lambda D(\mathbf{p}_1 || \mathbf{q}_1) + (1-\lambda) D(\mathbf{p}_2 || \mathbf{q}_2) \geq D(\lambda \mathbf{p}_1 + (1-\lambda)\mathbf{p}_2 || \lambda \mathbf{q}_1 + (1-\lambda)\mathbf{q}_2) \tag{13.27}$$

Since (13.27) is the definition of a convex function, the proof is completed. □

13.3 Computation of Channel Capacity

13.3.1 Alternating Maximization

Given given a discrete memoryless channel $p_{\mathsf{X}|\mathsf{Y}}(x|y)$, the capacity C is the maximization of mutual information $I(\mathsf{X};\mathsf{Y})$:

$$C = \max_{p_{\mathsf{X}}(x)} I(\mathsf{X};\mathsf{Y}), \tag{13.28}$$

over all $p_{\mathsf{X}}(x)$. If a capacity-achieving input distribution $p^*_{\mathsf{X}}(x)$ is known, then the capacity can be found. However, there is no analytic solution to this optimization problem, except in a few special cases, and only numerical solutions are possible.

Define the following for convenience:

$$r(x) = p_{\mathsf{X}}(x) \tag{13.29}$$

$$p(y|x) = p_{\mathsf{Y}|\mathsf{X}}(y|x) \tag{13.30}$$

$$q(x|y) = p_{\mathsf{X}|\mathsf{Y}}(x|y) \tag{13.31}$$

The "backwards channel" is $q(x|y) = \Pr(\mathsf{X} = x|\mathsf{Y} = y)$. Clearly, $r(x)$ depends on $q(x|y)$ and likewise $q(x|y)$ depends on $r(x)$. Using the above notation, the mutual information can be written as:

$$I(\mathsf{X};\mathsf{Y}) = \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} r(x)p(y|x)\log\frac{q(x|y)}{r(x)} \tag{13.32}$$

In order to obtain a numerical solution, the key point is to write the problem as an alternating maximization problem:

$$C = \max_{q(x|y)}\max_{r(x)} I(\mathsf{X};\mathsf{Y}) \tag{13.33}$$

The Arimoto-Blahut algorithm uses alternating maximization, where first $r(x)$ is fixed, and $q(x|y)$ is maximized. Then, $q(x|y)$ is fixed and $r(x)$ is maximized. This proceeds iteratively, until a stable solution is found. Explicitly, the optimization problem is:

$$C = \max_{q(x|y)}\max_{r(x)}\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} r(x)p(y|x)\log\frac{q(x|y)}{r(x)}. \tag{13.34}$$

The first step is to fix $r(x)$, and find $q^*(x|y)$ which maximizes

$$q^*(x|y) = \arg\max_{q(x|y)}\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} r(x)p(y|x)\log\frac{q(x|y)}{r(x)}, \tag{13.35}$$

which is given by

$$q^*(x|y) = \frac{r(x)p(y|x)}{\sum_{x'\in\mathcal{X}} r(x')p(y|x')}. \tag{13.36}$$

The details of how to find (13.36) is given in Subsection 13.5.1

The second step is to fix $q(x|y)$ and find $r^*(x)$ which maximizes

$$r^*(x) = \arg\max_{r(x)}\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} r(x)p(y|x)\log\frac{q(x|y)}{r(x)} \tag{13.37}$$

which is given by

$$r^*(x) = \frac{\prod_y \left(q(x|y)\right)^{p(y|x)}}{\sum_{x'\in\mathcal{X}}\prod_y \left(q(x'|y)\right)^{p(y|x')}}. \tag{13.38}$$

Algorithm 13.1 Computation of Channel Capacity

Require: A discrete memoryless channel $p(y|x)$.
Ensure: Channel capacity C, capacity-achieving input distribution $p^*_{\mathsf{X}}(x)$
(a) Initialize $r(x)$ with a random distribution
(b) Fix $r(x)$, maximize over $q(x|y)$. For all $x \in \mathcal{X}$, $y \in \mathcal{Y}$:

$$q(x|y) = \frac{r(x)p(y|x)}{\sum_{x' \in \mathcal{X}} r(x')p(y|x')}$$

(c) Fix $q(x|y)$, maximize over $r(x)$. For all $\mathbf{x} \in \mathcal{X}$:

$$r(x) = \frac{\prod_y \left(q(x|y)\right)^{p(y|x)}}{\sum_{x' \in \mathcal{X}} \prod_y \left(q(x'|y)\right)^{p(y|x')}}$$

(d) Go to step (b) until the solution $r(x)$ stabilizes.
(e) Capacity C is $I(\mathsf{X};\mathsf{Y})$ computed using using $r(x)$ and $p(y|x)$

The details of how to find (13.38) is given in Subsection 13.5.2.

The above procedure is summarized in Algorithm 13.1, which iterates between the two steps until convergence is achieved.

13.3.2 Numerical Example

This subsection gives a numerical example of the Arimoto-Blahut algorithm to find the capacity-achieving input distribution and the capacity C. Consider a three-input, three-output discrete memoryless channel with $p(y|x)$ given by:

$$p(y|x) = \begin{bmatrix} 0.75 & 0.25 & 0 \\ 0 & 1 & 0 \\ 0 & 0.5 & 0.5 \end{bmatrix}, \tag{13.39}$$

so $\mathcal{X} = \{1, 2, 3\}$ and $\mathcal{Y} = \{1, 2, 3\}$. A plot of $I(\mathsf{X};\mathsf{Y})$ is shown in Fig. 13.7. The contour lines illustrate the mutual information as a function of the input distribution $r(1), r(2)$. Since $r(3)$ is dependent on $r(1)$ and $r(2)$, it is not shown.

First, choose the initial vector $r(x)$ arbitrarily:

$$r(x) = \begin{bmatrix} 0.1 & 0.1 & 0.8 \end{bmatrix} \tag{13.40}$$

which is shown as a red point "0" in Fig. 13.7.

In Step (b), compute $q(x|y)$ using (13.36). For $x = 3, y = 2$:

$$q(3|2) = \frac{0.8 \cdot 0.5}{0.1 \cdot 0.25 + 0.1 \cdot 1 + 0.8 \cdot 0.5} \approx 0.7619 \tag{13.41}$$

Computing all the values in this way gives:

$$q(x|y) = \begin{bmatrix} 1.0000 & 0.0000 & 0.0000 \\ 0.0476 & 0.1905 & 0.7619 \\ 0.0000 & 0.0000 & 1.0000 \end{bmatrix}. \tag{13.42}$$

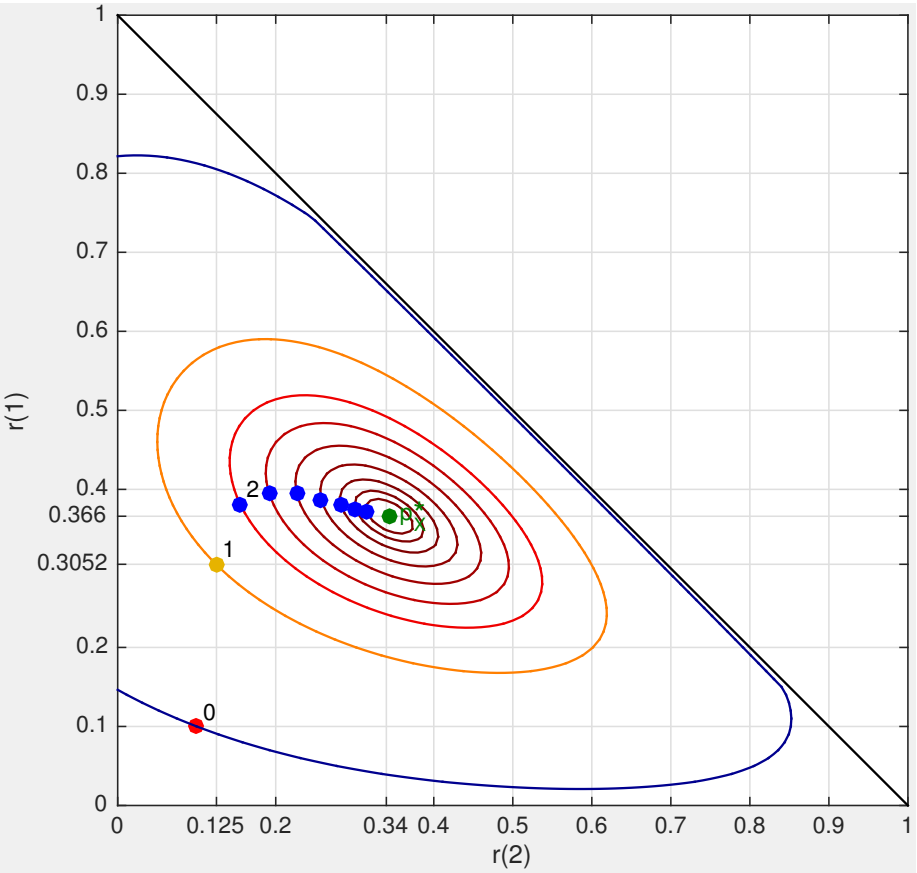


Figure 13.7: Example of capacity computation. Mutual information $I(\mathsf{X};\mathsf{Y})$ versus input distribution $r(1), r(2)$ is shown using contour lines. The Arimoto-Blahut algorithm is initialized with the red point denoted "0". The result of the first iteration is the orange point "1". Iterations progress until convergence at the green point "p_{X}^*" is obtained.

In step (c), compute $r(x)$. The numerator of $r(1), r(2)$ and $r(3)$ in (13.38) are:

$$q(1|1)^{p(1|1)} \cdot q(1|2)^{p(2|1)} \cdot q(1|3)^{p(3|1)} = 1^{0.75} \cdot 0.0476^{0.25} \cdot 0^0 \approx 0.4617 \quad (13.43)$$

$$q(2|1)^{p(1|2)} \cdot q(2|2)^{p(2|2)} \cdot q(2|3)^{p(3|2)} = 0^0 \cdot 0.1905^1 \cdot 0^0 \approx 0.1905 \quad (13.44)$$

$$q(3|1)^{p(1|3)} \cdot q(3|2)^{p(2|3)} \cdot q(3|3)^{p(3|3)} = 0^0 \cdot 0.7619^{0.5} \cdot 1^{0.5} \approx 0.8729. \quad (13.45)$$

Using these to compute the denominator, $r(x)$ is:

$$r(x) = [0.3052, 0.1245, 0.5703]. \quad (13.46)$$

This value is shown as an orange dot labeled as "1" in Fig. 13.7.

In step (d), the solution $r(x)$ changed significantly from the previous iteration, so go to step (b). The next step is shown by "2" in Fig. 13.7, and the progress are shown by blue dots. The iterations will continue until $r(x)$ stabilizes, this is a capacity-achieving input distribution. For this channel, the capacity-achieving input distribution is:

$$p^*_{\mathsf{X}}(x) = [0.3657, 0.3440, 0.2903], \quad (13.47)$$

which is indicated by a green point labeled as "$p^*_{\mathsf{X}}(x)$" in Fig. 13.7. This $p^*_{\mathsf{X}}(x)$ is used to compute the capacity:

$$C = 0.7845 \text{ bits/channel use.} \quad (13.48)$$

An example of source code, written in Matlab, which implements this algorithm is given in Sec. 13.7.

13.4 Computation of the Rate-Distortion Function

13.4.1 Alternating Minimization

Given a source X distributed as $p_{\mathsf{X}}(x)$, and a distortion metric $d(x, \widehat{x})$, the rate-distortion function $R(D)$ is the minimization of mutual information $I(\mathsf{X}; \widehat{\mathsf{X}})$:

$$R(D) = \min_{p_{\widehat{\mathsf{X}}|\mathsf{X}}(\widehat{x}|x): Ed < D} I(\mathsf{X}; \widehat{\mathsf{X}}), \quad (13.49)$$

over all $p_{\widehat{\mathsf{X}}|\mathsf{X}}(\widehat{x}|x)$, subject to the distortion constraint $E[d(\mathsf{X}, \widehat{\mathsf{X}})] \leq D$ (the term $Ed < D$ is shorthand to save space). If an optimal $p_{\widehat{\mathsf{X}}|\mathsf{X}}(\widehat{x}|x)$ is known, then the rate-distortion function can be found. However, as with the channel capacity computation, we must find solutions numerically in all but a few special cases.

Define the following for convenience:

$$p(x) = p_{\mathsf{X}}(x) \quad (13.50)$$

$$q(x|y) = p_{\widehat{\mathsf{X}}|\mathsf{X}}(\widehat{x}|x) \quad (13.51)$$

$$r(x) = p_{\widehat{\mathsf{X}}}(x) \quad (13.52)$$

Using the above notation, mutual information can be written two ways:

$$I(\mathsf{X};\widehat{\mathsf{X}}) = \sum_{x\in\mathcal{X}}\sum_{\widehat{x}\in\widehat{\mathcal{X}}} q(\widehat{x}|x)p(x)\log\frac{q(\widehat{x}|x)}{r(\widehat{x})} \text{ and} \tag{13.53}$$

$$I(\mathsf{X};\widehat{\mathsf{X}}) = D\big(q(\widehat{x}|x)p(x)||r(\widehat{x})p(x)\big). \tag{13.54}$$

In order to obtain a numerical solution, the key point is to write the problem as a alternating minimization problem:

$$R(D) = \min_{r(\widehat{x})}\min_{q(\widehat{x}|x)} I(\mathsf{X};\widehat{\mathsf{X}}). \tag{13.55}$$

The Arimoto-Blahut algorithm maximizes by alternating between two step. In the first step, $r(\widehat{x})$ is fixed, and $q(\widehat{x}|x)$ is minimized. In the second step, $q(\widehat{x}|x)$ is fixed and $r(\widehat{x})$ is minimized. This proceeds iteratively, until a stable solution is found.

The first step is to fix $r(\widehat{x})$ and to find $q^*(\widehat{x}|x)$ which minimizes the mutual information in (13.53)

$$q^*(\widehat{x}|x) = \arg\min_{q(\widehat{x}|x)}\sum_{x\in\mathcal{X}}\sum_{\widehat{x}\in\widehat{\mathcal{X}}} q(\widehat{x}|x)p(x)\log\frac{q(\widehat{x}|x)}{r(\widehat{x})} \tag{13.56}$$

which is given by

$$q^*(\widehat{x}|x) = \frac{r(\widehat{x})e^{-\lambda d(x,\widehat{x})}}{\sum_{\widehat{x}'\in\widehat{\mathcal{X}}} r(\widehat{x}')e^{-1}e^{-\lambda d(x,\widehat{x}')}} \tag{13.57}$$

The second step is to fix $q(\widehat{x}|x)$ and find the $r^*(\widehat{x})$ which minimizes the mutual information in (13.54):

$$r^*(\widehat{x}) = \arg\min_{r(\widehat{x})} D\big(q(\widehat{x}|x)p(x)||r(\widehat{x})p(x)\big). \tag{13.58}$$

which is given by:

$$r^*(\widehat{x}) = \sum_{x\in\mathcal{X}} p(x)q(\widehat{x}|x) \tag{13.59}$$

The above procedure is summarized in Algorithm 13.2, which iterates between the two steps until convergence is achieved. The details of how to find (13.62) is given in Subsection 13.5.3 how to find (13.63) is given in Subsection 13.5.4.

13.5 Optimization Details

This section gives details on the optimization problems used by the Arimoto-Blahut algorithms.

Algorithm 13.2 Computation of Rate-Distortion Function

Require: A discrete probability distribution $p(x)$, an output alphabet $\widehat{\mathcal{X}}$, a distortion measure $d(x, \widehat{x})$, a parameter λ.

Ensure: Using optimized $q^*(\widehat{x}|x)$ and $r^*(\widehat{x})$ from the last iteration, output R and D given by:

$$R = \sum_{x\in\mathcal{X}} \sum_{\widehat{x}\in\widehat{\mathcal{X}}} q^*(\widehat{x}|x)p(x) \log \frac{q^*(\widehat{x}|x)}{r^*(\widehat{x})} \text{ and} \tag{13.60}$$

$$D = \sum_{x\in\mathcal{X}} \sum_{\widehat{x}\in\widehat{\mathcal{X}}} q^*(\widehat{x}|x)p(x)d(x, \widehat{x}) \tag{13.61}$$

(a) Initialize with any choice of $r(\widehat{x})$, for example a random distribution.
(b) Fix $r(\widehat{x})$, minimize over $q(\widehat{x}|x)$:

$$q(\widehat{x}|x) = \frac{r(\widehat{x})e^{-\lambda d(x,\widehat{x})}}{\sum_{\widehat{x}'\in\widehat{\mathcal{X}}} r(\widehat{x}')e^{-\lambda d(x,\widehat{x}')}} \tag{13.62}$$

(c) Fix $q(\widehat{x}|x)$, minimize over $r(\widehat{x})$:

$$r(\widehat{x}) = \sum_{x\in\mathcal{X}} p(x)q(\widehat{x}|x) \tag{13.63}$$

(d) Go to step (b) until the solution stabilizes.

13.5.1 Optimization details: Fix $r(x)$, maximize $q(x|y)$

The first step of the capacity computation is to fix $r(x)$, and find $q^*(x|y)$ which maximizes

$$q^*(x|y) = \arg\max_{q(x|y)} \sum_{x\in\mathcal{X}} \sum_{y\in\mathcal{Y}} r(x)p(y|x) \log \frac{q(x|y)}{r(x)}, \tag{13.64}$$

which is given by

$$q^*(x|y) = \frac{r(x)p(y|x)}{\sum_{x'\in\mathcal{X}} r(x')p(y|x')}. \tag{13.65}$$

Proof Let $w(y)$ be given by:

$$w(y) = \sum_{x'\in\mathcal{X}} r(\widehat{x}')p(y|x') \tag{13.66}$$

so that $q^*(x|y)w(y) = p(y|x)$.

$$
\begin{aligned}
&\sum_{y\in\mathcal{Y}}\sum_{x\in\mathcal{X}} p(y|x)\log\frac{q^*(x|y)}{r(x')} - \sum_{y\in\mathcal{Y}}\sum_{x\in\mathcal{X}} p(y|x)\log\frac{q(\widehat{x}|x)}{r(x')} \\
&= \sum_{y\in\mathcal{Y}}\sum_{x\in\mathcal{X}} p(y|x)\log\frac{q^*(x|y)}{q(\widehat{x}|x)} \\
&= \sum_{y\in\mathcal{Y}}\sum_{x\in\mathcal{X}} q^*(x|y)w(y)\log\frac{q^*(x|y)}{q(\widehat{x}|x)} \\
&= \sum_{y\in\mathcal{Y}} w(y)D(q^*(x|y)||q(\widehat{x}|x)) \\
&\geq 0
\end{aligned}
$$

The above technique can also be used to show:

$$r^*(y) = \arg\min D(p(y|x)p(x)||p(x)r(y)) \tag{13.67}$$

is given by

$$r^*(y) = \sum_{x\in\mathcal{X}} p(y|x)p(x) \tag{13.68}$$

For related problem, see Exercise 13.5

See Dhillon et al. [KDD 2002; Lemma 2], Cover and Thomas page 333, Yeung page 219

13.5.2 Optimization details: Fix $q(x|y)$, maximize $r(x)$

The second step is to fix $q(x|y)$ and find the $r^*(x)$ which maximizes

$$r^*(x) = \arg\max_{r(x)} \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} r(x)p(y|x)\log\frac{q(x|y)}{r(x)} \tag{13.69}$$

subject to the constrtaint $\sum_{x\in\mathcal{X}} r(x) = 1$.

Use Lagrange multipliers. The functional J is:

$$J = \Big(\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} r(x)p(y|x)\log\frac{q(x|y)}{r(x)}\Big) + \lambda\Big(\sum_{x\in\mathcal{X}} r(x) - 1\Big) \tag{13.70}$$

Take the derivative with respect to $r(x)$ for all x:

$$\frac{\partial J}{\partial r(x)} = \sum_{y\in\mathcal{Y}} p(y|x)\log q(x|y) - \sum_{y\in\mathcal{Y}} p(y|x)\log r(x) - \sum_{y\in\mathcal{Y}} p(y|x) + \lambda \tag{13.71}$$

$$= \sum_{y\in\mathcal{Y}} p(y|x)\log q(x|y) - \log r(x) - 1 + \lambda \tag{13.72}$$

Now set $J = 0$ and solve for $r(x)$:

$$\log r(x) = \sum_{y\in\mathcal{Y}} p(y|x)\log q(x|y) - 1 + \lambda \tag{13.73}$$

$$r(x) = 2^{\lambda-1}\prod_{y\in\mathcal{Y}} q(x|y)^{p(y|x)} \tag{13.74}$$

Lagrange multiplier did not account for the scalar factor, and we need $\sum r(x) = 1$, so:

$$r^*(x) = \frac{\prod_{y\in\mathcal{Y}} q(x|y)^{p(y|x)}}{\sum_{x'\in\mathcal{X}}\prod_{y\in\mathcal{Y}} q(\widehat{x}'|x)^{p(y|x')}}, \tag{13.75}$$

which is the optimal value. Refer also to Yeung, page 221.

13.5.3 Optimization Details: Fix $r(\widehat{x})$, Minimize Over $q(\widehat{x}|x)$

In this case, find the minimum of $I(\mathsf{X},\widehat{\mathsf{X}})$ using (13.56). The minimization problem is find:

$$\min_{q(\widehat{x}|x)} \sum_{x\in\mathcal{X}}\sum_{\widehat{x}\in\widehat{\mathcal{X}}} q(\widehat{x}|x)p(x)\log\frac{q(\widehat{x}|x)}{r(\widehat{x})} \tag{13.83}$$

such that:

$$\sum_{x\in\mathcal{X}} p(x)q(\widehat{x}|x)d(x,\widehat{x}) = D \text{ and} \tag{13.84}$$

$$\sum_{\widehat{x}\in\widehat{\mathcal{X}}} q(\widehat{x}|x) = 1. \tag{13.85}$$

Using the Lagrange multiplier[1] λ, The functional is:

$$J = \Big(\sum_{x\in\mathcal{X}}\sum_{\widehat{x}\in\widehat{\mathcal{X}}} q(\widehat{x}|x)p(x)\log\frac{q(\widehat{x}|x)}{r(\widehat{x})}\Big) + \lambda\Big(\sum_{x\in\mathcal{X}}\sum_{\widehat{x}\in\widehat{\mathcal{X}}} p(x)q(\widehat{x}|x)d(x,\widehat{x}) - D\Big)$$

Take the derivative with respect to $q(\widehat{x}|x)$ for all x and $\widehat{x}$:

$$\begin{aligned}\frac{\partial J}{\partial q(\widehat{x}|x)} &= \frac{\partial}{\partial q(\widehat{x}|x)}\Big(\sum_{x\in\mathcal{X}}\sum_{\widehat{x}\in\widehat{\mathcal{X}}} q(\widehat{x}|x)p(x)\log\frac{q(\widehat{x}|x)}{r(\widehat{x})}\Big) + \lambda\Big(\sum_{x\in\mathcal{X}}\sum_{\widehat{x}\in\widehat{\mathcal{X}}} p(x)q(\widehat{x}|x)d(x,\widehat{x}) - D\Big)\\ &= p(x)\log\frac{q(\widehat{x}|x)}{r(\widehat{x})} + q(\widehat{x}|x)p(x)\Big(\frac{r(\widehat{x})}{q(\widehat{x}|x)}\cdot\frac{1}{r(\widehat{x})}\Big) + \lambda\cdot\Big(p(x)d(x,\widehat{x})\Big)\\ &= p(x)\log\frac{q(\widehat{x}|x)}{r(\widehat{x})} + p(x) + \lambda\cdot\Big(p(x)d(x,\widehat{x})\Big)\end{aligned}$$

Now set equal to 0, solve for $q(\widehat{x}|x)$:

$$\log\frac{q(\widehat{x}|x)}{r(\widehat{x})} + 1 + \lambda d(x,\widehat{x}) = 0 \tag{13.86}$$

$$q(\widehat{x}|x) = r(\widehat{x})e^{-1}e^{-\lambda d(x,\widehat{x})} \tag{13.87}$$

[1] We write $+\lambda$ in (13.86) rather than $-\lambda$ because later we restrict to $\lambda > 0$

The method of Lagrange multipliers can be used to find local maxima and minima of a function $f(\mathbf{x})$ subject to equality constraints $g(\mathbf{x}) = 0$. The idea is to form a Lagrangian function L:

$$L = f(\mathbf{x}) - \lambda g(\mathbf{x}), \tag{13.76}$$

where λ is a parameter. We solve $\frac{\partial L}{\partial x_i} = 0$ to find optimal $\mathbf{x}^* = (x_1^*, x_2^*, \ldots, x_n^*)$.
Wikipedia: Lagrange multiplier
The method is best described by an example. Solve

$$\max_{p_i} \sum_{i=1}^{n} a_i \log p_i \text{ such that } \sum_{i=1}^{n} p_i = 1 \tag{13.77}$$

This can be shown by forming Lagrangian L with Lagrange multiplier λ:

$$L = \sum_{i=1}^{n} a_i \log p_i - \lambda \sum_{i=1}^{n} p_i \tag{13.78}$$

$$\frac{\partial L}{\partial p_i} = \frac{a_i}{p_i} - \lambda = 0 \tag{13.79}$$

$$p_i^* = \frac{a_i}{\lambda} \tag{13.80}$$

Find λ by applying (13.80) to the constraint:

$$\sum_{i=1}^{n} p_i = 1 \;\Rightarrow\; \sum_{i=1}^{n} \frac{a_i}{\lambda} = 1 \;\Rightarrow\; \lambda = \sum_{i=1}^{n} a_i \tag{13.81}$$

Applying (13.81) to (13.80), the solution:

$$p_i^* = \frac{a_i}{a_1 + a_2 + \cdots + a_n}. \tag{13.82}$$

is obtained.

Figure 13.8: The Method of Lagrange multipliers.

Lagrange multiplier did not account for a scalar factor, and we need $\sum_{\widehat{x}\in\widehat{\mathcal{X}}} q(\widehat{x}|x) = 1$, so:

$$q(\widehat{x}|x) = \frac{r(\widehat{x})e^{-\lambda d(x,\widehat{x})}}{\sum_{\widehat{x}\in\widehat{\mathcal{X}}} r(\widehat{x})e^{-\lambda d(x,\widehat{x})}} \tag{13.88}$$

Note that λ should be fixed. But as we change λ, it sweeps out the rate-distortion curve.

13.5.4 Optimization Details: Fix $q(\widehat{x}|x)$, Minimize Over $r(\widehat{x})$

The second step of the rate-distortion functiom computation is to fix $r(\widehat{x})$, and find $q^*(x|y)$ which minimizes:

$$r^*(\widehat{x}) = \arg\min_{r(\widehat{x})} D(p(x)q(x|y)||p(x)r(\widehat{x})) \tag{13.89}$$

which is given by

$$r^*(\widehat{x}) = \sum_{x\in\mathcal{X}} p(x)q(x|y) \tag{13.90}$$

The proof is similar to the optimization problem in Subsection 13.5.1.

13.6 Information Bottleneck and Its Method

13.6.1 Introduction

Consider a Markov chain $\mathsf{X} \to \mathsf{Y} \to \mathsf{Z}$, where the joint distribution $p_{\mathsf{Y}|\mathsf{X}}(y|x)p_{\mathsf{X}}(x)$ is known. Let $\mathcal{X} = \{1, \ldots, J\}$ be the sample space for X, let $\mathcal{Y} = \{1, \ldots, M\}$ be the sample space for Y and let $\mathcal{Z} = \{1, \ldots, K\}$ be the sample space for Z.

The *information bottleneck* means that information should be passed from X to Z through a random variable Y. Consider the objective function:

$$\min_{p_{\mathsf{Z}|\mathsf{Y}}(z|y)} I(\mathsf{Y};\mathsf{Z}) - \beta I(\mathsf{X};\mathsf{Z}), \tag{13.91}$$

where $\beta \geq 0$ is a fixed parameter. The optimization goal is to find $p_{\mathsf{Z}|\mathsf{Y}}(z|y)$, which is a mapping from the observation Y to the estimate Z. Sometimes we write $Q(y) = z$ instead of $p_{\mathsf{Z}|\mathsf{Y}}(z|y)$.

If we take the Information Bottleneck problem with $\beta \to \infty$ it reduces to:

$$\max_{p_{\mathsf{Z}|\mathsf{Y}}(z|y)} I(\mathsf{X};\mathsf{Z}) \tag{13.92}$$

The optimal solution satisfies $p_{\mathsf{Z}|\mathsf{Y}}(z|y) = 0$ or 1, that is the mapping is deterministic. But for any finite β, $p_{\mathsf{Z}|\mathsf{Y}}(z|y)$ is probabilistic. For $\beta = 0$ the constraint term is ignored, an the minimum of $I(\mathsf{Y};\mathsf{Z})$ is 0. This is achieved by a $p_{\mathsf{Z}|\mathsf{Y}}(z|y)$ which is uniform in some sense.

From a rate-distortion point of view, $y \to z$ is a mapping that requires a certain number of R-D codewords for a fixed amount of distortion. If β increases, the number of $R-D$ codewords may increase (at the same time the "distortion" $I(\mathsf{X};\mathsf{Z})$ decreases).

13.6.2 Alternative Representation of the IB Problem

The conditional *input* distribution $p_{\mathsf{X}|\mathsf{Y}}(x|y)$ is readily found from the joint distribution $p_{\mathsf{Y}|\mathsf{X}}(y|x)p_{\mathsf{X}}(x)$:

$$p_{\mathsf{X}|\mathsf{Y}}(x|y) = \frac{p_{\mathsf{Y}|\mathsf{X}}(y|x)p_{\mathsf{X}}(x)}{p_{\mathsf{Y}}(y)} \tag{13.93}$$

$$= \frac{p_{\mathsf{Y}|\mathsf{X}}(y|x)p_{\mathsf{X}}(x)}{\sum_{x\in\mathcal{X}} p_{\mathsf{Y}|\mathsf{X}}(y|x)p_{\mathsf{X}}(x)}. \tag{13.94}$$

A vector representation is useful. For each channel output $y \in \mathcal{Y}$, define a vector of channel inputs:

$$\mathbf{u}_y = \big(p_{\mathsf{X}|\mathsf{Y}}(1|y), \cdots, p_{\mathsf{X}|\mathsf{Y}}(J|y)\big). \tag{13.95}$$

The points $\mathbf{u}_y$ are in the J-dimensional probability simplex, $\mathbf{u}_y \in \mathcal{U} \subseteq [0,1]^J$. Besides conditioning X on Y, we can also condition X on Z. For each $z \in \mathcal{Z}$, define a vector of channel inputs conditioned on z as:

$$\mathbf{v}_z = \big(p_{\mathsf{X}|\mathsf{Z}}(1|z), \cdots, p_{\mathsf{X}|\mathsf{Z}}(J|z)\big), \tag{13.96}$$

which are in the same probability simplex, $\mathbf{v}_z \in \mathcal{V} \subseteq [0,1]^J$.

Using this vector space representation, the M channel outputs are represented by $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_M$, and the K quantizer outputs are represented by $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_M$. The information (Kullback-Leiber) divergence between $\mathbf{u}$ and $\mathbf{v}$ is:

$$D(\mathbf{u}||\mathbf{v}) = \sum_{i=1}^{J} u_i \log \frac{u_i}{v_i}, \tag{13.97}$$

where $\mathbf{u}_y$ and $\mathbf{v}_z$ are probability distributions in vector form.

The points $\mathbf{u}_y$ are in the J-dimensional probability simplex, $\mathbf{u}_y \in \mathcal{U} \subseteq [0,1]^J$. Let $\mathbf{U}$ be the random vector given by:

$$\mathbf{U} = \big(p_{\mathsf{X}|\mathsf{Y}}(1|\mathsf{Y}), \ldots, p_{\mathsf{X}|\mathsf{Y}}(J|\mathsf{Y})\big). \tag{13.98}$$

Let $\mathbf{V}$ be the random vector given by:

$$\mathbf{V} = \big(p_{\mathsf{X}|\mathsf{Z}}(1|\mathsf{Z}), \ldots, p_{\mathsf{X}|\mathsf{Z}}(J|\mathsf{Z})\big), \tag{13.99}$$

which can be seen as a random variable on $\mathcal{Z}$.

Using this vector space representation, the M channel outputs are represented by $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_M$, and the K quantizer outputs are represented by

Algorithm 13.3 $\beta \to \infty$ Information Bottleneck Method

Input: Discrete memoryless channel $p_{\mathsf{Y}|\mathsf{X}}(y|x)$ input distribution $p_{\mathsf{X}}(x)$, number of outputs K

Initialization: Calculate M data points $\mathcal{U} = \{\mathbf{u}_1, \ldots, \mathbf{u}_M\}$:

$$\mathbf{u}_y = \frac{p_{\mathsf{Y}|\mathsf{X}}(y|x) \cdot p_{\mathsf{X}}(x)}{\sum_{x' \in \mathcal{X}} p_{\mathsf{Y}|\mathsf{X}}(y|x') p_{\mathsf{X}}(x')} \tag{13.102}$$

so that:

$$\mathbf{u}_y = \big(p_{\mathsf{X}|\mathsf{Y}}(1|y), \cdots, p_{\mathsf{X}|\mathsf{Y}}(J|y)\big). \tag{13.103}$$

Randomly choose K of the M data points as the initial means $\mathbf{v}_1, \ldots, \mathbf{v}_K$.

Step 1 (Assignment Step): Group points with the same nearest mean to the same cluster:

$$\mathcal{U}_k = \{\mathbf{u} | D(\mathbf{u}||\mathbf{v}_k) \leq D(\mathbf{u}||\mathbf{v}_l), k \neq l\} \tag{13.104}$$

Step 2 (Update Step): Update mean for each cluster:

$$\mathbf{v}_k = \frac{1}{|\mathcal{U}_k|} \sum_{\mathbf{u} \in \mathcal{U}_k} \mathbf{u} \tag{13.105}$$

Step 3: Go to Step 1 until the clusters stop changing, or a maximum number of iterations have been reached.

Output: Use clusters $\mathcal{U}_k$ to generate the quantizer Q.

$\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_M$. The information (Kullback-Leiber) divergence between $\mathbf{u}$ and $\mathbf{v}$ is:

$$D(\mathbf{u}||\mathbf{v}) = \sum_{i=1}^{J} u_i \log \frac{u_i}{v_i}, \tag{13.100}$$

where $\mathbf{u}_y$ and $\mathbf{v}_z$ are probability distributions in vector form.

Proposition 13.9. Let $\mathsf{X} \to \mathsf{Y} \to \mathsf{Z}$ form a Markov chain. Let $\mathbf{U}$ and $\mathbf{V}$ be defined as above. Then:

$$E[D(\mathbf{U}||\mathbf{V})] = H(\mathsf{X}|\mathsf{Z}) - H(\mathsf{X}|\mathsf{Y}), \tag{13.101}$$

where E denotes expectation.

13.6.3 The $\beta \to \infty$ Information Bottleneck

Show that maximizing the mutual information is equivalent to minimizing the KL divergence:

$$\begin{aligned}
\arg\max I(\mathsf{X};\mathsf{Z}) &= \arg\min I(\mathsf{X};\mathsf{Y}) - I(\mathsf{X};\mathsf{Z}) && \text{constant does not change arg} \\
&= \arg\min H(\mathsf{X}|\mathsf{Z}) - H(\mathsf{X}|\mathsf{Y}) && \text{definition of mutual information} \\
&= \arg\min E[D(\mathbf{U}||\mathbf{V})] && \text{Proposition 13.9}
\end{aligned}$$

Use arg min because the minimum changes, but $p_{\mathsf{Z}|\mathsf{Y}}(z|y)$ which achieves this minimum does not change.

Thus, maximizing $I(\mathsf{X};\mathsf{Z})$ is equivalent to minimizing $E[D(\mathbf{U}||\mathbf{V})]$ and:

$$\widetilde{Q}^* = \arg\min_{\widetilde{Q}} E[D(\mathbf{U}||\mathbf{V})]. \tag{13.106}$$

The main point here is that minimization of KL divergence is a suitable metric for quantizer design, and is the metric used by the $\beta \to \infty$ information bottleneck method. When $\beta \to \infty$ the information bottleneck method reduces to a modified K-means algorithm. The modification is that the Euclidean distance metric is replaced with the KL divergence metric. The $\beta \to \infty$ information bottleneck method is shown in Algorithm 13.3

13.6.4 Channel Quantization

The motivation for studying the information bottleneck method is both classification and the quantization of channels. See the lecture slides for more details. Here basics of channel quantization are given.

If X is a Gaussian random variable with mean m and variance v, then:

$$\Pr(\mathsf{X} > x) = Q(\frac{x-m}{\sqrt{v}}) = \frac{1}{2}\text{erfc}(\frac{x-m}{\sqrt{2v}}) \tag{13.107}$$

Suppose an AWGN channel output is y, and this is quantized to one of K levels, $z \in \{1, 2, \ldots, K\}$. The quantization boundaries are $q_1, q_2, \ldots, q_{M+1}$. Then y is quantized to z if $q_z < y < q_{z-1}$. If we take $q_1 = -\infty$ and $q_{M+1} = \infty$, the quantization function Q can be represented by:

$$Q(y) = \begin{cases} 1 & \text{if } \infty < y \leq q_2 \\ 2 & \text{if } q_2 < y \leq q_3 \\ \vdots & \\ M & \text{if } q_M < y \leq \infty \end{cases} \tag{13.108}$$

If the channel is a binary-input AWGN channel with $x = \pm 1$, then the conditional output distribution is:

$$p_{\mathsf{Z}|\mathsf{X}}(z|x) = \frac{1}{2}\text{erfc}(\frac{q_z - x}{\sqrt{2v}}) - \frac{1}{2}\text{erfc}(\frac{q_{z-1} - x}{\sqrt{2v}}). \tag{13.109}$$

The channel from X to Z is a discrete memoryless channel with joint distribution $p_{\mathsf{Z}|\mathsf{X}}(z|x)p_{\mathsf{X}}(x)$ and mutual information $I(\mathsf{X};\mathsf{Z})$.

13.7 Source Code

This Matlab source code implements the Arimoto-Blahut algorithm for computing the capacity of a DMC given by $p_{\mathsf{Y}|\mathsf{X}}(y|x)$.

```
function [C,pxstar] = capacityComputation(pygx)

[X Y] = size(pygx);

%initial random input distribution
px    = rand(1,X);
px    = px / sum(px);

%0 -> 1E-6 for numerical stability
pygx(find(pygx==0)) = 1E-6;

r = px;
p = pygx;

for i = 1:200
    %Fix r(x), maximize over q(x|y):
    for x = 1:X
        for y = 1:Y
            q(x,y) = p(x,y) * r(x) / sum(r(:) .* p(:,y));
        end
    end

    %Fix q(x|y), maximize over r(x):
    for x = 1:X
        r(x) = prod(q(x,:).^p(x,:));
    end
    r = r ./ sum(r);
end

pxstar = r;
pxy    = repmat(pxstar',1,Y) .* pygx;   %joint distribution pxy
py     = sum(pxy,1);                    %marginal distribution py

%Capacity is mutual information using pxstar
C      = sum(sum(pxy .* log2( pxy ./ (pxstar' * py)))) ;
```

13.8 Exercises

13.1 Show that $f(x) = x^2$ is a convex function. For $y > 0$, show that $g(y) = y \log y$ is a convex function.

13.2 For what values of t is the function $g(t) = (t-1)^3$ convex?

13.3 Maximize $f(x, y) = x + y$ subject to the constraint $x^2 + y^2 = 1$ using Lagrange multipliers.

13.4 (a) Prove that mutual information $I(\mathsf{X};\mathsf{Y})$ is concave in $p_{\mathsf{X}}(x)$ for fixed $p_{\mathsf{Y}|\mathsf{X}}(y|x)$ by writing $I(\mathsf{X};\mathsf{Y}) = H(\mathsf{Y}) - H(\mathsf{Y}|\mathsf{X})$. (b) Prove that mutual

information is convex is $p_{\mathsf{Y}|\mathsf{X}}(y|x)$ for a fixed $p_{\mathsf{X}}(x)$ by writing $I(\mathsf{X};\mathsf{Y}) = D(p_{\mathsf{X}}(x)p_{\mathsf{Y}|\mathsf{X}}(y|x)||p_{\mathsf{X}}(x)p_{\mathsf{Y}}(y))$.

13.5 Consider a collection of M probability distributions $\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_M$ on a J-ary alphabet, so that each $\mathbf{y}$ is a vector:

$$\mathbf{y} = (y_1, y_2, \ldots, y_J) \tag{13.125}$$

with $\sum_{i=1}^{J} y_i = 1$. Find the probability distribution $\mathbf{z}^*$ that minimizes the total KL divergence

$$\mathbf{z}^* = \arg\min_{\mathbf{z}} \sum_{i=1}^{M} D(\mathbf{y}_i||\mathbf{z}), \text{ such that } \sum_{j=1}^{J} z_j = 1 \tag{13.126}$$

where $\mathbf{z} = (z_1, z_2, \ldots, z_J)$. (A possible approach is to use Lagrange multipliers.)

13.6 *Optimal input distribution is not necessarily unique.* Consider the the noisy keyboard channel with four inputs and four outputs, given by:

$$p_{Y|X} = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ 0 & \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & 0 & 0 & \frac{1}{2} \end{bmatrix}. \tag{13.141}$$

The capacity of this channel is $C = 1$ bit/channel use. This may be achieved by the input distribution $p_{\mathsf{X}}(x) = [\frac{1}{2}, 0\frac{1}{2}, 0]$ It may also be achieved by the input distribution $p_{\mathsf{X}}(x) = [0, \frac{1}{2}, 0, \frac{1}{2}]$. Give a proof that any distribution $[b, \frac{1}{2} - b, b, \frac{1}{2} - b]$ for $0 \leq b \leq \frac{1}{2}$ is a capacity-achieving distribution.

13.7 *Arimoto-Blahut algorithm for channel capacity* Implement Arimoto-Blahut algorithm for computing the capacity of a DMC in any programming language you like. The Information Theory Lecture Notes contains a partial example in Matlab — note that this implementation does not check for convergence. Use your implementation to compute the capacity C and capacity-achieving input distribution p_X^* of the following channel:

$$p_{\mathsf{Y}|\mathsf{X}}(y|x)y|x = \frac{1}{45}\begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 5 & 2 & 8 & 1 & 3 & 6 & 4 & 7 & 9 \\ 5 & 8 & 2 & 9 & 7 & 4 & 6 & 3 & 1 \end{bmatrix}. \tag{13.145}$$

13.8 *Arimoto-Blahut algorithm for rate-distortion computation* Modify your source code of the previous problem to implement the Arimoto-Blahut algorithm for computing the $R(D)$ curve for a source. Your implementation should have three input arguments: the source distribution $p_{\mathsf{X}}(x)x$, the distortion metric as a $|\mathcal{X}|$-by-$|\widehat{\mathcal{X}}|$ matrix d, and a parameter λ. The output should be $R(D)$ and D. If you are using Matlab, use a function declaration like:

```
function [RD,D] = rateDistortionComputation(p,d,lambda)
```

(a) Use your implementation to find the $R(D)$ curve for the binary source with $p_{\mathsf{X}}(x)x = [0.2, 0.8]$ and Hamming distortion metric. By changing λ, obtain pairs $R(D), D$ sweep an $R(D)$ curve. On the same graph, plot $R(D) = h(p) - h(D)$ to validate your algorithm.

(b) Use your implementation to plot the $R(D)$ curve for a ternary source with $p_{\mathsf{X}}(x)x = [\frac{1}{3}, \frac{1}{3}, \frac{1}{3}]$ and distortion metric in matrix form:

$$d(x, \widehat{x}) = \begin{bmatrix} 0 & 1 & 4 \\ 1 & 0 & 1 \\ 4 & 1 & 0 \end{bmatrix}. \tag{13.146}$$

What is the value of $R(0)$? How can you find $R(0)$ exactly? For what minimum value of D^* do you obtain $R(D^*) = 0$? What scheme can achieve $R(D^*) = 0$ for this value of D?

13.9 *Information bottleneck method for BI-AWGN channel* The output Y of the binary-input AWGN channel is $\mathsf{Y} = \mathsf{X} + \mathsf{Z}$, where X is uniform on $\mathcal{X} = \{-1, +1\}$ and $\mathsf{Z} \sim \mathcal{N}(0, \sigma^2)$ is Gaussian with variance σ^2.

(a) Find $p_{\mathsf{Y}|\mathsf{X}}(y|x)$ and $p_{\mathsf{X}|\mathsf{Y}}(x|y)$ for this channel.
(b) Perform channel quantization of the binary-input AWGN channel by implementing the information bottleneck algorithm in your favorite programming language. Generate $M/2$ random samples from $\mathcal{N}(-1, \sigma^2)$ and $M/2$ random samples from $\mathcal{N}(1, \sigma^2)$, called $y_1, y_2, \ldots, y_M$. For each one, compute $p_{\mathsf{X}|\mathsf{Y}}(x|y)$ using part (a). Apply the information bottleneck method to $p_{\mathsf{X}|\mathsf{Y}}(x|y)$ with $K = 4$ to obtain four clusters. Use $M = 1000$ and $\sigma^2 = 0.1$. Plot $p_{\mathsf{Y}|\mathsf{X}}(y| - 1), p_{\mathsf{Y}|\mathsf{X}}(y| + 1)$ and the four means m_1, m_2, m_3, m_4 you obtained.

Note Referring to the previous homework problem "K-means for Gaussian Quantization", the information bottleneck method can be implemented by changing the assignment step from the Euclidean distance metric to the KL divergence metric. No change is needed for the update step.