Lecture Notes: Information Theory and Statistics

Caution: Very Rough Draft

October 19, 2005

Contents

Chapter 1

Entropy and Codes

1.1 Prelude: entropy's physics origin

The idea of entropy was invented in 1850 by the Prussian theoretical physicist Rudolf Julius Emmanuel Clausius (1822- 1888) who played an important role in establishing theoretical physics as a discipline. As many physicists of the time, such as Laplace, Poisson, Sadi Carnot and Clapeyron, Clausisus was into the theory of the heat, called the caloric theory at the time, which was based on two axioms: 1. the heat in the universe is conserved and 2. the heat in a substance is a function of the state of the substance. Clausius' most famous paper was read in 1850 to Berlin Academy and published in Annalen der Physik in in the same year, laying the foundation of modern thermodynamics. In this paper, he argued that the two axioms are wrong and gave the first and second laws of thermodynamics in place of the two axioms. The first thermodynamics law stated the equivalence of heat and work and it was well supported by experimental data of Joule. The acceptance of the first law refuted both axioms in the caloric theory.

For the second law of thermodynamics, Clausius set up an equation (in modern notations):

$$\bar{d}Q = dU + \bar{d}W,$$

where $\bar{d}Q$ was the change in the heat, dU the energy change in the system, and $\bar{d}W$ the change in the external work done. d stands for "true differential" because U (and entropy S to be introduced later), as we know now, is a function of the state of the system, while $\bar{d}$ is a differential which depends on how a system is brought from its initial state to its final state.

The introduction of the energy of the system, U, was of great significance and U was later name intrinsic energy by another physicist William Thomson. In the same 1850 paper, Clausius also recognized entropy as the quantity that remains invariant during changes of volume and temperature in a Carnot cycle (which transmits heat between two heat reservoirs at different temperatures and at the same time converts heat into work). He did not name the important entropy concept at that time, however. In the fifteen years to follow, Clausius continued to refine the two laws of themodynamics. In 1865, Clausius gave the two laws of thermodynamics in the following form:

1. The energy of the universe is constant.

2. The entropy of the universe tends to a maximum.

And the paper contained the equation:

$$dS = \bar{d}Q/T,$$

where S is the entropy, Q is the internal energy or heat, and T the temeprature. The amazing property of entropy is that, although the integration of $\bar{d}Q$ depends on the detailed path, but the integration of $\bar{Q}/T = dS$ does not!

Clausius' most important contribution to physics is undoubtedly his idea of the irreversible increase in entropy, and yet there seems no indication of interest from him in Boltzmann's views on thermodynamics and probability or Josiah Willard Gibbs' work on chemical equilibrium, both of which were utterly dependent on his idea. It is strange that he himself showed no inclination to seek a molecular understanding of irreversible entropy or to find further applications of the idea; it is stranger yet, and even tragic, that he expressed no concern for the work of his contemporaries who were accomplishing those very tasks.

Ludwig Boltzmann (1844-1906), a theoretical phycist at Vienna (and Graz), became famous because of his invention of statistical mechanics. This he did independently of Josiah Willard Gibbs. Their theories connected the properties and behaviour of atoms and molecules with the large scale properties and behaviour of the substances of which they were the building blocks. In 1877, Boltzmann quantifies entropy of an equilibrium thermodynamic system as

$$S = K \log W,$$

S - entropy, K - Boltzman constant, W - number of microstates in the system. This formula is also carved onto Boltzman's tombstone even though it has been said that Planck was the first who wrote this down.

In the United States, J. W. Gibbs (1839-1903), a Europe-trained mathematical physicist at Yale College, advanced a branch of physics called statistical mechanics to describe microscopic order and disorder. Statistical mechanics describes the behavior of a substance in terms of the statistical behavior of the atoms and molecules contained in it. His work on statistical mechanics provided a mathematical framework for quantum theory and for Maxwell's theories. His last publication, Elementary Principles in Statistical Mechanics, beautifully lays a firm foundation for statistical mechanics.

In the 1870s, Gibbs introduced another expression to describe entropy such that if all the microstates in a system have equal probability, his term reduces to k log W. This formula, often simply called Boltzmann-Gibbs entropy, has been a workhorse in physics and thermodynamics for 120 years.

$$S = -\sum_j p_j \log p_j,$$

where p_j is the probability that the system is at microstate j. Mathematically, it is easy to see that if $p_j = 1/W$, Gibbs' entropy formula agrees with Boltzman's entropy formula.

1.2 Shannon's entropy and codes

Claude Elwood Shannon was born in Gaylord, Michigan, on April 30, 1916 and passed away on Feb. 26, 2001. He is considered as the founding father of electronic communications age. His work on technical and engineering problems within the communications industry laid the groundwork for both the computer industry and telecommunications.

> *The fundamental problem of communication is that of reproducing at one point either exactly or approximately a message selected at another point.*
>
> Claude Shannon
> *A Mathematical Theory of Communication*

In Shannon's information theory, a message is a random draw from a probability distribution on messages and entropy gives the data compression (source coding) limit. Shannon's entropy measures "information" content in a message, but this "information" is not the meaningful information. It is simply the uncertainty in the message just as Boltzmann-Gibbs entropy measures the disorder in a thermodynamic system.

Shannon's information theory concerns with point-to-point communications as in telephony, and characterizes the limits of communication. Abstractly, we work with *messages* or sequences of symbols from a discrete *alphabet* that are generated by some *source*. In the rest of the chapter, we consider the problem of *encoding* the sequence of symbols for storage or (noiseless) transmission. In the literature of information theory, this general problem is referred to as *source coding*. How compactly can we represent messages emanating from a discrete source? In his original paper in 1948, Shannon assumed sources generated messages one symbol at a time according to a probability distribution; each new symbol might depend on the preceding symbols as well as the alphabet. Therefore, Shannon defined a source to be a discrete stochastic process.

In more familiar terms, this chapter concerns *data compression.* The framework studied by Shannon can be applied to e-mail messages, Web pages, Java programs, and any data stored on your hard drive. How small can we compress these files? The source coding tools introduced in this chapter help us address this question. While Shannon's probabilistic view of a source is not valid for a fixed data file, we can still apply the concepts of his theory and gain useful insights into the basic properties of compression algorithms.

Throughout this chapter, the reader will find a number of connections with the field of statistics. We might expect a certain overlap given Shannon's stochastic characterization of a source.

1.3 Examples of Code

A *code* $\mathcal{C}$ on a discrete *alphabet* $\mathcal{X}$ is simply a mapping from *symbols* in $\mathcal{X}$ to a set of *codewords*. With this mapping, we encode *messages* or sequences of symbols from $\mathcal{X}$. Throughout this chapter, we consider codes that are *lossless* in the sense that messages can be *decoded* exactly, without any loss of information.

Example 1.1 (Simple binary codes). Let our alphabet consist of just three symbols, $\mathcal{X} = \{a, b, c\}$. A *binary* code is a mapping from $\mathcal{X}$ to strings of 0's and 1's. Here is one such code:

$$\begin{array}{ccc} a & \rightarrow & 00 \\ b & \rightarrow & 01 \\ c & \rightarrow & 10 \end{array} \tag{1.1}$$

Here, we encode each symbol with two binary digits or *bits*[1]; each symbol is assigned a number 0,1,2 and the code is just a binary representation of that number. The *length* of each codeword is a fixed 2 bits, making it a *fixed-length* code. With this code, the 10-symbol message *aabacbcbaa* becomes 00000100100110010000 and the 10-symbol message *bcccbabcca* is 01101010010001101000, each requiring 20 bits. Formally, we encode messages with the *extension* of $\mathcal{C}$ that concatenates the codewords for each symbol in the message. Decoding involves splitting the encoded string into pairs of 0'1 and 1's, determining the integer associated with each pair (0, 1 or 2) and then performing a table lookup to see which symbol is associated with each integer.

Here is another binary code for the same alphabet:

$$\begin{array}{ccc} a & \rightarrow & 0 \\ b & \rightarrow & 10 \\ c & \rightarrow & 11 \end{array} \tag{1.2}$$

Notice that each codeword now involves a different number of bits so that this code is a *variable length* code. Applying the extension of this code, the 10-symbol message *aabacbcbaa* becomes 001001110111000 and the 10-symbol message *bcccbabcca* is 1011111111001011110. In the previous example, decoding involved processing pairs of bits. In this case, we notice that the codewords form a so-called *prefix* code; that is, no codeword is the prefix of another. This property means that encoded messages are *uniquely decodable*, even if we don't include special separating markers between the codewords.

[1] Bits was suggested to Shannon by his statistician colleague J. W. Tukey at Bell Labs.

Notice that the 10-symbol message *aabacbcbaa* requires 15 bits to encode; and the 10-symbol message *bcccbabcca* needs 18 bits. Given the short codeword for a, we expect this code to do better with the first 10-symbol message. In both cases, however, this code improves on the fixed-length scheme (1.1) requiring 20 bits per 10-symbol message. In the next section we will illustrate how the mapping (1.2) was constructed and see how our assumptions about messages guide code design.

Example 1.2 (ASCII and Unicode). Originally, the American Standard Code for Information Interchange (ASCII) was a 7 bit coded character set for English letters, digits, mathematical symbols and punctuation. It was widely used for storing and transmitting basic English language documents. Each symbol was mapped to a digit between 0 and 127. It was common to include an $8th$ bit referred to as the *parity bit* to check that the symbol has been transmitted correctly; here the $8th$ bit might be 1 if the number of the symbol being sent is odd and zero if its even. Newer operating systems work with legitimate 8-bit extensions to ASCII, encoding larger character sets that include more mathematical symbols, graphics symbols and some non-English characters. Unicode is a 16-bit code that assigns a unique number to every character or symbol in use, from Bengali to Braille. With 16 bits, there is room for over 65K characters in the code set. Both ASCII and Unicode are fixed-length encoding schemes like (1.1).

Example 1.3 (Morse Code). This encoding is named for Samuel Morse, originally a professor of arts and design at New York University. The alphabet for Morse's original code consisted of numbers that mapped to a fixed collection of words. The codewords consisted of of dots, dashes and pauses.

The more familiar version of Morse code was developed by Alfred Vail. Here, the alphabet consists of English letters, numbers and punctuation. The codewords consist of strings made from a set of five symbols; dot, dash, short gap (between each letter), medium gap (between words) and long gap (between sentences). In designing the codewords, Morse and Vail adopted a compression strategy: the letter "e" is a single dot, and "t" is a single dash. These letters appear more commonly in standard written English than letters assigned longer strings of dots and dashes. Two-symbol codewords were assigned to "a," "i," "m" and "n." Morse and Vail did derive their coding scheme by counting letters in samples of text, but instead counted the individual pieces of type in each section of a printer's type box. (In frequency counts of characters taken from modern texts, "o" appears more frequently than "n," and "m" often doesn't score among the top 10.)

Even with this compression, codes were built on top of Morse code. Telegraph companies charged based on the length of the message sent. Codes emerged that encoded complete phrases in five-letter groups that were sent as single words. Examples: BYOXO ("Are you trying to crawl out of it?"), LIOUY ("Why do you not answer my question?"), and AYYLU ("Not clearly coded, repeat more clearly."). The letters of these five-letter code words were sent individually using Morse code.

With this example, we see how we can reduce the size of a data set by capitalizing on regularities. Here the regularities are in the form of frequencies, or rather some ordering of the frequencies of letters in common English transmissions. As we will see, the same principle guided the design of (1.2), making it better suited for messages with more a's than b's or c's.

Example 1.4 (Braille). Braille was developed by a blind Frenchman named Louis Braille in 1829. Braille is based on a 6-bit encoding scheme, which allows a maximum of 63 possible codes. Since only 26 of this codes are required for encoding the letters of the alphabet, the remainder of the codes are used to encode common words (and, for, of, the, with) and common two-letter combinations (ch, gh, sh, th, wh, ed, er, ou, ow). In 1992 there was an attempt to unite separate Braille codes for mathematics, scientific notation and computer symbols into one Unified English Braille Code (UEBC).

In Braille we see even more direct use of frequent structures in English. By directly encoding common words and word-fragments, we achieve even more compression.

We now collect some of the definitions introduced in the examples. A *code* or *source code* $\mathcal{C}$ is a mapping from an *alphabet* $\mathcal{X}$ to a set of codewords. In *binary code* the codewords are strings of 0's and 1's. A *non-singular code* maps each symbol in $\mathcal{X}$ into a different codeword. The *extension* $\mathcal{C}^*$ of a code $\mathcal{C}$ is the mapping from finite length strings of symbols of $\mathcal{X}$ to finite length binary strings, defined by $\mathcal{C}^*(x_1, \dots, x_n) = \mathcal{C}(x_1) \cdots \mathcal{C}(x_n)$, the concatenation of the codewords $\mathcal{C}(x_1)$, ..., $\mathcal{C}(x_n)$. If the extension of a code is non-singular, the code is called *uniquely decodable*. Codes with the *prefix* property are examples of uniquely decodable mappings. We say that a code is *prefix* or *instataneous* if no codeword is the prefix of any other codeword. Such codes allows decoding as soon as a codeword is finished or a leave node in the binary code tree is reached (hence the name "instataneous"). Uniquely decodable but not prefix codes need to look ahead for decoding. For example, on our three letter alphabet $\mathcal{X} = \{a, b, c\}$, the code

$$a \to 0,\ b \to 01,\ c \to 11$$

is uniquely decodable, but not prefix because a corresponds to an internal node 0. Nevertheless, the strings of 0's and 1's which come from encoding using this code can be uniquely decoded. For instance, 001011 is uniquely decoded to *abac*.

Exercise Set 1

1. Design a uniquely decodable, but not prefix code on an alphabet of size 3.

2. Write a meaningful email to a friend without using letter "e".

1.4 Codes and Probability Distributions

Given a binary code $\mathcal{C}$ on $\mathcal{X}$, the *length function* L maps symbols in $\mathcal{X}$ to the length of their codeword in bits. Using the code in (1.2), we have $L(a) = 1$ and $L(b) = L(c) = 2$. In general, there is a correspondence between the length function of a prefix code and the quantity $-\log_2 Q$ for a probability distribution Q defined on $\mathcal{X}$. To make this precise, we first introduce the Kraft inequality.

Theorem 1.1 (Kraft inequality). *For any binary prefix code, the code length function L must satisfy the inequality*

$$\sum_{x\in\mathcal{X}} 2^{-L(x)} \leq 1\,. \tag{1.3}$$

Conversely, given a set of codeword lengths that satisfy this inequality, there exists a prefix binary code with these code lengths.

Proof. Given a binary prefix code, its codewords correspond to only leave nodes of the binary code tree, because of the prefix property (no codewords can be a prefix of another codeword so no internal nodes are codewords). Then the branch lengths of the leave codeword nodes are the lengths of the codewords. Complete the tree by adding single-leave nodes which don't correspond to any codewords to result in an expanded alphabet $\mathcal{X}'$ on all the leave nodes of a complete tree. Obviously

$$\sum_{x'\in\mathcal{X}'} 2^{-L(x')} = 1,$$

which implies that

$$\sum_{xin\mathcal{X}} 2^{-L(x)} \leq \sum_{x'\in\mathcal{X}'} 2^{-L(x')} = 1.$$

For the other direction, given any set of code word lengths $L(x), x \in \mathcal{X} = \{1, 2, ..., k\}$ which satisfy the Kraft inequality. Pick the first node in a binary tree from left to right of depth l_1 as the codeword for 1 and take out its offsprings from the tree so to make it a leave node on the code tree. Then pick the first remaining node of depth l_2 as the codeword for 2, etc. The Kraft's inequality ensures that we have enough nodes to go around to give everyone a codeword as a leave node of the code tree. □

Using the Kraft inequality, we can take any length function L and construct a distribution as follows

$$Q(a) = \frac{2^{-L(a)}}{\sum_{a\in\mathcal{X}} 2^{-L(a)}} \quad \text{for any } a \in \mathcal{X}. \tag{1.4}$$

Conversely, for any distribution Q on $\mathcal{X}$ and any $a \in \mathcal{X}$, we can find a prefix code with length function $L(a) = \lceil -\log_2 Q(a)\rceil$, the smallest integer greater than or equal to $-\log_2 Q(a)$. This is because

$$L(x) \geq -\log Q(x), \text{ hence } -L(x) \leq \log Q(x),$$

and it follows that

$$\sum 2^{-L(x)} \leq \sum 2^{\log Q(x)} = \sum Q(x) = 1.$$

Now, consider a stochastic *source* of symbols. That is, suppose our messages are constructed by randomly selecting elements of $\mathcal{X}$ according to a distribution P. Then, the *expected length* of a code $\mathcal{C}$ is given by

$$L_{\mathcal{C}} = \sum_{x \in \mathcal{X}} P(x)L(x)\,. \tag{1.5}$$

The following theorem characterizes the shortest expected code length given a source with distribution P.

Theorem 1.2 (Information inequality). *Given two distribution Q and P,*

$$E_P \log P(X)/Q(X) \geq 0,$$

and the equality holds if and only if $P = Q$.

Proof.

$$E_P \log \frac{P(X)}{Q(X)} = -E_P \log \frac{Q(X)}{P(X)} \geq^* -\log E_P \frac{Q(X)}{P(X)} = -\log 1 = 0.$$

The inequality above holds due to Jensen's inequality applied to the convex function $-\log$. That is, for $Y = Q(X)/P(X)$

$$E[-\log Y] \geq -\log EY,$$

which is the same as

$$-(E \log Y) \geq -\log EY.$$

□

Theorem 1.3 (Shannon's source coding theorem). *Suppose the elements of $\mathcal{X}$ are generated according to a probability distribution P. For any prefix code $\mathcal{C}$ on $\mathcal{X}$ with length function $L(\cdot)$, the expected code length $L_{\mathcal{C}}$ is bounded below*

$$L_{\mathcal{C}} \geq -\sum_{x \in \mathcal{X}} P(x) \log_2 P(x) = H(P) \tag{1.6}$$

where quality holds if and only if $L = -\log_2 P$.

Proof. By Kraft's inequality,

$$C_L = \sum_x 2^{-L(x)} \leq 1.$$

Then

$$Q(x) = 2^{-L(x)}/C_L.$$

is a probability distribution.

Since $E_P L = E_P[-\log(Q(X)) - \log C_L] \geq E_P[-\log(Q(X)),$

$$EP_L - H(P) = -E_P \log \frac{Q(X)}{P(X)} \geq 0,$$

□

Note that implicit in this result is that we know the distribution P that generates messages we wish to encode. To a statistician this seems like an impossible luxury. Instead, it is more realistic to consider one or more distributions Q that approximate P in some sense. In coding problems, we can evaluate different models based on their ability to compress the data. We will formalize these notions in later chapters. For now, we illustrate ties between codes and probability distributions by describing several well-known encoding schemes.

1.5 Coding algorithms based on a known distribution

We now consider several coding schemes and evaluate them based on their ability to compress a corpus of text. We took as our test case 175 stories classified by the online news service from Google as having to do with the power outage that hit the Northeastern United States on Thursday, August 14, 2003 (stories collected on August 15, 2003). There were 1,022,574 characters in this sample, which means a simple ASCII encoding would require 8,180,592 bits (or 998.6 Kb)

Example 1.1 (Shannon). Suppose we are given an alphabet $\mathcal{X} = \{x_1, \ldots, x_n\}$ with probability function P. Now, consider a length function of the form $L^*(x) = \lceil -\log P(x) \rceil$, where $\lceil y \rceil$ denotes the smallest integer greater or equal to y. These lengths satisfy Kraft's inequality since

$$\sum_{i=1}^{n} 2^{-\lceil -\log P(x_i) \rceil} \leq \sum_{i=1}^{n} 2^{\log P(x_i)} = \sum_{i=1}^{n} P(x_i) = 1\,. \tag{1.7}$$

Therefore, by Kraft's inequality we can find a code with this length function. Since the ceiling operator introduces an error of at most one bit, we have that

$$H(P) \leq EL^* \leq H(P) + 1 \tag{1.8}$$

from Information inequality.

Shannon proposed a simple scheme that creates the code with length function L^*. Suppose that the symbols in our alphabet are ordered so that $P(x_1) \geq P(x_2) \geq \cdots \geq P(x_n)$. Define $F_i = \sum_{j=1}^{i-1} P(x_i)$, the sum of the probabilities of symbols 1 through $i-1$. The codeword for x_i is then taken to be F_i rounded to $\lceil -\log P_i \rceil$ bits. The resulting code is prefix, and as indicated above has an expected code length within one bit of the entropy.

Now let us construct the Shannon code for a distribution P on $\{a, b, c\}$ with probabilities 11/20, 1/4, 1/5, respectively. Its entropy can be easily calculated as $H(P) = 1.439$ bits. Moreover,

$$F_1 = 0, F_2 = 11/20, F_3 = 4/5,$$

$$\lceil -\log P_1 \rceil = \lceil 0.86 \rceil = 1; \lceil -\log P_2 \rceil = \lceil 2 \rceil = 2; \lceil -\log P_3 \rceil = \lceil 2.3 \rceil = 3$$

Rounding F_1, F_2, F_3 to 1, 2 and 3 bits, we obtain the codewords for a, b, c as

$$C(F_1) = 0, C(F_2) = 10, C(F_3) = 110,$$

because

$$F_1 = 0, F_2 = 11/20 = 1/2 + 1/20; F_3 = 4/5 = 1/2 + 1/4 + 1/20.$$

This Shannon code has an expected code length

$$L = 11/20 + 2 \times 1/4 + 3 \times 1/5 = 8/5 = 1.6 \text{ bits}$$

which is quite close to the entropy 1.439 bits.

In Figure 1.1 we show the Shannon code for the characters in the collection of documents. Using this code, we need 5,534,865 bits (or 675.6 Kb) to code the articles, a 32% reduction. The entropy of the character distribution is 4.77, and the expected code length of the Shannon code is 5.41 [2].

We now consider two coding methods that are directly based on binary trees. Recall from our proof of the Kraft inequality that the leaves of a binary tree can be used to represent a prefix code. The next two constructions recursively construct codes in a top-down (Shannon-Fano coding) and bottom-up (Huffman coding) fashion.

Example 1.2 (Shannon-Fano). This is a top-down method for forming a binary tree that will characterize a prefix code.

1. List all the possible messages, with their probabilities, in decreasing probability order
2. Divide the list into two parts of (roughly) equal probability
3. Start the code for those messages in the first part with a 0 bit and for those in the second part with a 1
4. Continue recursively until each subdivision contains just one message

[2] Strictly speaking, we should have also accounted for the bits needed to encode the frequencies in Figure 1.1 since they are estimated from data. We will elaborate on this point in the MDL sections later.

		Shannon Code		Huffman Code	
x	$P(x)$	bits	codeword	bits	codeword
space	0.173	3	000	3	111
e	0.085	4	0010	4	1101
t	0.062	5	01000	4	1010
a	0.060	5	01010	4	1001
o	0.057	5	01100	4	0111
r	0.054	5	01101	4	0100
s	0.048	5	01111	4	0010
i	0.047	5	10001	4	0001
n	0.047	5	10010	4	0000
l	0.028	6	101000	5	01101
d	0.027	6	101010	5	01011
h	0.027	6	101100	5	01010
u	0.024	6	101101	6	110011
c	0.022	6	101111	6	110010
g	0.015	7	1100001	6	101100
w	0.015	7	1100011	6	100010
p	0.014	7	1100101	6	100000
y	0.014	7	1100110	6	011001
m	0.014	7	1101000	6	011000
f	0.013	7	1101010	6	001110
b	0.010	7	1101011	7	1100010
.	0.009	7	1101101	7	1100000
k	0.008	7	1101110	7	1011011
,	0.008	7	1101111	7	1011010
S	0.007	8	11100000	7	1000010

Figure 1.1: Characters from 175 news articles related to the massive blackout in the Northeastern part of the United States. We present the alphabet (in this case, letters, numbers and some punctuation), the frequency of each symbol in the corpus, and the codewords for the associated Shannon code (we will explain the last two columns related to the Huffman code shortly).

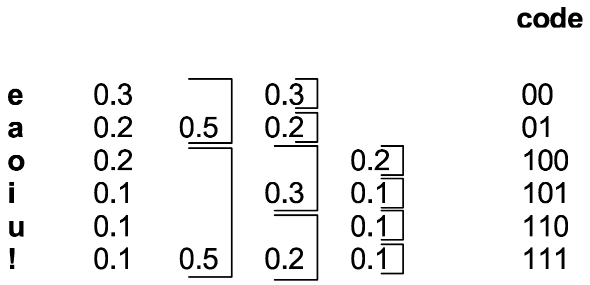


Figure 1.2: Constructing the Shannon-Fano code on a 6-letter alphabet, $\mathcal{A} = \{a, e, i, o, u, !\}$.

It is possible to show that for the Shannon-Fano code,

$$H(P) \leq L \leq H(P) + 2. \tag{1.9}$$

To illustrate the process, consider a six-symbol alphabet $\mathcal{X} = \{a, e, i, o, u, !\}$. We give the probability function in Figure 1.2; note that we have sorted the symbols in terms of their frequency. The entropy of this distribution is 2.45. Note that the expected code length of the resulting Shannon-Fano code is 2.5 bits. In Figure 1.2, we compare this code length to that of the Shannon code built from the same frequency table. Here, the Shannon code has an expected code length of 3 bits.

Shannon-Fano code (or any prefix code) can be re-expressed in terms of the game of 20 questions. Suppose we want to find a sequence of yes-no questions to determine an object from a class of objects with a known probability distribution. We first group the objects into two subsets of (roughly) equal probabilities and ask which subset contains the object, and so on. This is exactly how the Shnnon-Fano code was constructed. For any prefix code, there is a binary tree with leaf nodes as the code words for the objects. We start from the root of the tree, and group the objects into two subsets according to whether the objects are the descendants of the left branch or the right branch and ask the question whether the object is in the left subset or the right subset, and so on. In terms of the expected number of questions asked, the optimal sequence of questions corresponds to the Huffman code which we will now begin to introduce.

In his 1948 masterpiece, Shannon solved the source coding problem when the message gets longer and longer (the entropy rate is achieved in the limit) and this is the celebrated Shannon's source coding theorem which we will take up later. However, he left open the optimal coding question for a fixed size alphabet. That is, among all prefix codes, which code gives the shortest expected code length with respect to a message generating distribution P. Even though we know the Shannon code gets close to the lower bound entropy within one bit, in the finite case, the entropy might not be achievable. Huffman (1952) solved this finite sample optimality problem by showing that the Huffman code obtains the shortest average code length among all prefix codes. This is a surprising result since Huffman code is a greedy algorithm, but it nevertheless achieves the global optimality.

Example 1.3 (Huffman Coding). This is a bottom-up method for forming a binary tree that will characterize a prefix code. We recursively combine small trees to form one large binary tree. Start with as many trees as there are symbols in the alphabet. While there is more than one tree:

1. Find the two trees with the smallest total probability.
2. Combine the trees into one, setting one as the left child and the other as the right.
3. Now the tree contains all the symbols. A ’0’ represents following the left child; a ’1’ represents following the right child.

It is possible to demonstrate that the Huffman code is optimal among all prefix codes in the sense that it produces the shortest expected code length. In fact,

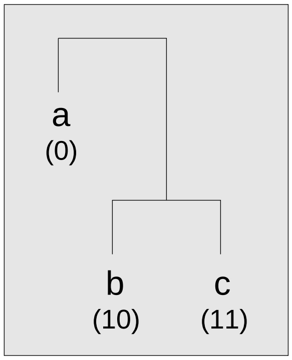
$$H(P) \leq L \leq H(P) + 1. \tag{1.10}$$

We now illustrate the coding procedure on a couple examples.

We continue with the simple example used for illustrating the Shannon code. Let $\mathcal{X} = \{a, b, c\}$ and let P denote a probability distribution on $\mathcal{X}$ with $P(a) = 11/20$ and $P(b) = 1/4$ and $P(c) = 1/5$. We can construct a code for $\mathcal{X}$ by growing a binary tree from the end-nodes $\{a, b, c\}$. This procedure is similar to the greedy algorithm used in agglomerative, hierarchical clustering (Jobson, 1992). First, we choose the two elements with the smallest probabilities, b and c, and connect them with leaves 0 and 1, assigned arbitrarily, to form the intermediate node bc having node probability $1/4 + 1/5 = 9/20$. We then iterate the process with the new set of nodes $\{a, bc\}$. Since there are only two nodes left, we connect a and bc with leaves 0 and 1, again assigned arbitrarily, and reach the tree’s root. The tree obtained through this construction as well as the resulting code are given explicitly in Figure 1. Let L be the code length function associated with this code so that $L(a) = L(0) = 1$, $L(b) = L(10) = 2$,

		Shannon Code		S-F Code		Huffman Code	
x	$P(x)$	bits	codeword	bits	codeword	bits	codeword
e	0.3	2	00	2	00	2	00
a	0.2	3	010	2	01	2	10
o	0.2	3	011	3	100	3	010
u	0.1	4	1011	3	101	3	111
i	0.1	4	1100	3	110	3	011
!	0.1	4	1110	3	111	3	110

Figure 1.3: Several coding schemes applied to a small, 6-character alphabet.



$$\begin{aligned} \mathcal{C}:\mathcal{X} &\rightarrow \{0,1\}^* = \text{ strings of 0's and 1's} \\ a &\rightarrow 0 \\ b &\rightarrow 10 \\ c &\rightarrow 11 \end{aligned}$$

Figure 1.4: Constructing a Huffman code in Example 1: At the left is the binary tree e on which the code is based, and on the right is an explicit description of the final mapping.

and $L(c) = L(11) = 2$. It is easy to see that in this case, our code length is given exactly by $L(x) = -\log_2 P(x)$ for all $x \in \mathcal{X}$.

This small example is slightly unrealistic in that the probabilities are all a power of 2. To further illustrate the coding process, we reconsider our 6-letter alphabet from the Shannon-Fano example. The codewords are given in Figure 1.2. Notice that while the codewords are a bit different than those from the Shannon-Fano code, the code lengths are the same. This means that the expected code length is also 2.5 (the source having an entropy of 2.45).

We now consider the Huffman code for the character distribution of the news articles relating to the power outage. In Figure 1.1, we exhibit the codewords for the Huffman table. Note that they are not longer than those of the Shannon code. In fact, the expected code length for this code is 4.81. Recall that the entropy of the distribution is 4.77 and the Shannon code had an expected length of 5.41.

As a final example, we apply the scheme to an alphabet consisting of the words in the corpus of 175 news stories. For the moment, we ignored punctuation and reduced the stream to a series of words. To make things even easier, we regularized the words in the sense that we removed all the capitalization. In all, there are 10,269 words. The distribution of counts is quite skewed, and seems to obey Zipf's law; see Figure 1.3. From this large alphabet, we then build both the Shannon code and the Huffman code. In Figure 1.3 we present both a sample of the frequency distribution as well as the codewords. Again, the Huffman table tends to have shorter codewords. The entropy of the distribution is 10.02, and the expected code lengths are 10.43 for Shannon and 10.05 for Huffman. In terms of actual compression, this means that we can encode the words with 1,665,567 bits (208,196 bytes or 203Kb) for Shannon and 1,605,460 bits (200,683 bits or 196Kb) for Huffman. This represents a tremendous savings over the character-based codes we've considered so far. Naturally, by reducing the data to lower-case words, we have simplified the stream and have made the

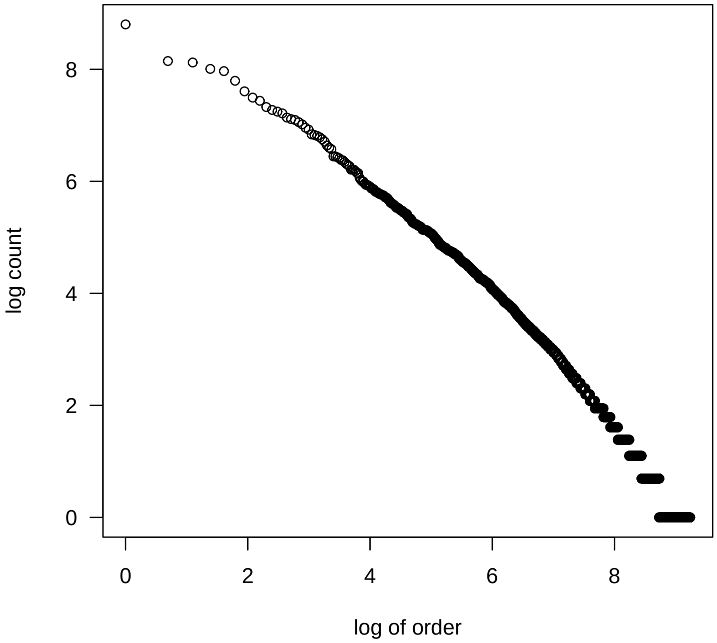


Figure 1.5: A frequency plot based on the words from the news stories corpus. The appearance of a straight line in this plot is said to indicate the existence of a power law, in this case, Zipf's law.

job easier. Still, if we were to have coded a file consisting of the stream of words (inserting a space between each) in ASCII, it would require 945,513 bytes or 923Kb.

A modification of Shannon code removes the sorting step by encoding the middle points in the jumps of the CDF and this gives the Shannon-Fano-Elias code which is particularly convenient for block coding.

Example 1.4 (Shannon-Fano-Elias). Let $\mathcal{X} = \{1, 2, \ldots, m\}$ and $Q(x) > 0$ for $x \in \mathcal{X}$. Define the cumulative distribution function $F(x)$ and the so-called modified distribution function $\bar{F}(x)$ to be

$$F(x) = \sum_{a \leq x} Q(a) \quad \text{and} \quad \bar{F}(x) = \sum_{a < x} Q(a) + \frac{1}{2} Q(x) \tag{1.11}$$

Since $Q(x)$ is positive on $\mathcal{X}$, the cumulative distribution function has the property that $F(a) \neq F(a')$ if $a \neq a'$. Looking at F, it is clear that we can map $F(x)$ back to x and hence $\bar{F}(x)$ can be used to encode x. In general, as a set of codewords, $\bar{F}(x)$ can be quite complicated, and might even require an infinite number of bits to describe. Instead, we build a code from a truncated expansion of $\bar{F}(x)$. We round to $l(x)$ bits, meaning

$$\bar{F}(x) - \lfloor \bar{F}(x) \rfloor_{l(x)} < \frac{1}{2^{l(x)}} \tag{1.12}$$

Now, if we set $l(x) = \lceil -\log Q(x) \rceil + 1$, then

$$\frac{1}{2^{l(x)}} \leq \frac{Q(x)}{2} = \bar{F}(x) - F(x-1) \tag{1.13}$$

so that the truncated value $\lfloor \bar{F}(x) \rfloor_{l(x)}$ can be mapped back to x using the cumulative distribution function. Like Huffman's algorithm, this construction also produces a prefix code (using $F(x)$ would not guarantee this). Also, this code also uses shorter code words for less frequently observed symbols.

In Figure 1.3 we exhibit the codewords for this construction. The main difference between this code and the Shannon code is that the symbols are not sorted according to their probability before code construction, so that some extra effort has to be expended to guarantee the prefix property of the resulting code.

So far, we have discussed four coding schemes, Shannon, Shannon-Fano, Huffman, and Shannon-Fano-Elias. We have seen that in some cases, we are close to the entropy bound, in others not so close. In the case of Huffman code, it is common to consider an alphabet formed by blocks of length n from the source. If the blocks are too small and the alphabet is small, then coding cannot provide much gains. For example, if we try to encode a string of 0's and 1's, taking blocks of size 1 will not produce any compression gains. No matter how we code, we will always be forced to communicate one bit for each input bit. This holds for every prefix code and when $H(P)$ is small, this is far away from the entropy lower bound given by the Information inequality.

		Shannon Code		Huffman Code		SFE Code	
x	$P(x)$	bits	codeword	bits	codeword	bits	codeword
the	0.042	5	00000	5	11100	6	000001
to	0.022	6	000010	6	111110	7	0000110
and	0.021	6	000100	6	111011	7	0001001
in	0.019	6	000101	6	110001	7	0001100
of	0.018	6	000110	6	101110	7	0001110
a	0.015	7	0001111	6	100000	8	00100001
power	0.013	7	0010001	6	001100	8	00100101
by	0.011	7	0010011	6	000010	8	00101000
new	0.011	7	0010100	7	1111001	8	00101011
on	0.010	7	0010101	7	1100110	8	00101101
august	0.009	7	0010111	7	1011011	8	00110000
said	0.009	7	0011001	7	1010011	8	00110010
york	0.008	7	0011010	7	1000111	8	00110100
thursday	0.008	8	00110110	7	1000011	9	001101100
score	0.008	8	00111000	7	0111110	9	001110000
pm	0.007	8	00111010	7	0111010	9	001110100
was	0.007	8	00111100	7	0110011	9	001111000
were	0.007	8	00111110	7	0101011	9	001111011
for	0.006	8	01000000	7	0011100	9	001111111
it	0.006	8	01000001	7	0001101	9	010000010
that	0.006	8	01000100	7	0000111	9	010000101
news	0.006	8	01000110	8	11111101	9	010001000
as	0.005	8	01000111	8	11110100	9	010001011
⋮	⋮	⋮	⋮	⋮	⋮		
parts	0.001	10	0111111101	10	1010100011	11	01111110000
still	0.001	10	1000000000	10	1010010101	11	01111110010
cause	0.001	10	1000000001	10	1010010110	11	01111111000
beneath	0.001	10	1000000010	10	1010010111	11	01111111011
now	0.001	10	1000000011	10	1010100000	11	01111110110
services	0.001	10	1000000100	10	1010100001	11	01111110100
threshold	0.001	10	1000000101	10	1010001101	11	01111111101
across	0.001	10	1000000110	10	1010010100	11	01111111111
million	0.001	10	1000000111	10	1010000011	11	10000000001
⋮	⋮	⋮	⋮	⋮	⋮		

Figure 1.6: Shannon, Huffman and Shannon-Fano-Elias encodings of the 175 news stories. Here, the “symbols” consist of 10269 words that appear in the corpus.

In general, we can improve the behavior of these schemes by encoding larger blocks of data. That is, rather than work with a single symbol at a time, we consider strings of length n. To see why there might be an advantage to doing this, let $P^n(x_1, \ldots, x_n) = P(x_1) \cdots P(x_n)$. Then,

$$H(P^n) \leq EL^n \leq H(P^n) + 1 \tag{1.14}$$

Since we have an iid sequence of symbols, the entropy can be written

$$H(P^n) = \sum H(P) = nH(P) \tag{1.15}$$

so that per symbol we have

$$H(P) \leq L < H(P) + \frac{1}{n} \tag{1.16}$$

This means we can get arbitrarily close to the entropy limit by considering longer and longer blocks. This is the celebrated Shannon's source coding theorem. Now let us prepare ourselves for its proof by studying the entropy function a bit in depth.

Exercise Set 2

1. *Compressing letters in ”Sons and Lovers” by D. H. Lawrence*

   Letter percentages (%) from Sons and Lovers are given in a decreasing order:

e	t	a	h	o	i	s	n	r
12.95	8.59	7.84	7.59	7.21	6.67	6.52	6.44	5.59

d	l	w	u	m	y	g	f	c
4.75	4.47	2.85	2.72	2.68	2.17	2.13	1.99	1.94

b	p	k	v	x	j	q	z
1.39	1.39	1.07	0.76	0.10	0.08	0.08	0.03

   Design the Shannon Code from these percentages, and display the code in a binary tree. Calculate the average code length per letter and compare with the estimated entropy rate.

2. *Optimality of Huffman code*

   Prove that Huffman code gives the shortest average code length among all prefix codes.

3. *Compare Huffman code on symbols with Huffman code on 3-tuples (blocks)*

   Given an iid binary message source with probability $0.1, 0.9$, design the Huffman code on $\{0, 1\}$ and the Huffman code on $\{0, 1\}^3$. Display the codes in binary trees, and Calculate the average code lengths and compare with the entropy rate.

1.6 Shannon's source coding theorem

Even though Shannon's source coding theorem holds for ergodic sequences, we cover only the iid case in this section. A sequence of iid symbols $X_1, ..., X_n$ from $\mathcal{X}$, each with entropy $H(P)$, can be compressed into more than $nH(P)$ bits with negligible loss of information as $n \to \infty$; conversely, information inequality ensures the entropy $nH(P)$ for the product measure as a lower bound.

Theorem 1.4 (Asymptotic Equipartition Property (AEP)). *If $X_1, ..., X_n$ are iid with a distribution P,*

$$-\frac{1}{n} \log_2 P(X_1, ..., X_n) \to H(P) \tag{1.17}$$

in probability as n tends to infinity.

Proof. The result follows from the weak law of large numbers. □

Corollary 1.1. *Under the assumptions of the above theorem, the average code lengths per symbol of the Shannon and Fano-Shannon-Elias codes on $\mathcal{X}^n$ tend to the entropy $H(P)$ as the sequence gets longer and longer.*

One can interpret or re-write the above result using the terminology of a typical set.

Definition 1 (Typical Set). For a given $\epsilon > 0$, the typical set is defined

$$A_\epsilon^{(n)} = \{x^n \in \mathcal{X} : \ 2^{-n(H(P)+\epsilon)} \leq P(x^n) \leq 2^{-n(H(P)-\epsilon)}\} \in \mathcal{X}^n.$$

The set $A_\epsilon^{(n)}$ is typical in the sense that the strings $x^n \in A_\epsilon^{(n)}$ account for most of the probability in the product space, $\mathcal{X}^n$, and the probability of each string $x^n \in A_\epsilon^{(n)}$ is close to uniform; that is, the cardiality of $A_\epsilon^{(n)}$ is approximately $2^{nH(P)}$, and the probability of each string $x^n \in A_\epsilon^{(n)}$ is about $2^{-nH(P)}$. When P is the uniform distribution on $\mathcal{X}$, we have seen that the entropy is $\log |\mathcal{X}|$. For the typical set, we have $\log_2 A_\epsilon^{(n)} \approx \log_2 2^{nH(P)} = nH(P)$. Hence, through the notion of a typical set, the uniform distribution emerges as an important tool for understanding the entropy of general P. These statements are made precise by the following theorem:

Theorem 1.5. *Given a probabiity distribution P on a set $\mathcal{X}$, and iid observations $X_1, \ldots, X_n$ from P, we have the following results about the typical set $A_\epsilon^{(n)}$ for $\epsilon > 0$*

1. *If $x^n \in A_\epsilon^{(n)}$,*
$$\left| -\frac{1}{n} \log_2 P(x^n) - H(P) \right| < \epsilon \,, \tag{1.18}$$

2. *For large n,*
$$P\left(A_\epsilon^{(n)}\right) > 1 - \epsilon \,, \tag{1.19}$$

3. *$|A_\epsilon^{(n)}| < 2^{n(H(P)+\epsilon)}$, and*

4. *For large n,*
$$\left|A_\epsilon^{(n)}\right| > (1-\epsilon)\, 2^{n(H(P)-\epsilon)} \tag{1.20}$$

Proof. The proof of (3) follows from the chain of (in)equalities

$$\begin{aligned} 1 &= \sum_{x^n \in \mathcal{X}^n} P(x^n) \\ &\geq \sum_{x^n \in A_\epsilon^{(n)}} P(x^n) \\ &\geq \left|A_\epsilon^{(n)}\right| 2^{-n(H(P)+\epsilon)}. \end{aligned}$$

□

An entropy-achieving block coding scheme can be devised based on the corollary. For a given $\epsilon > 0$, first we use one bit to indicate whether a sequence $x^n \in \mathcal{X}^n$ is in $A_\epsilon^{(n)}$ or not; we enumerate the sequences in $A_\epsilon^{(n)}$ in lexicographic order to give each sequence an integer index which is the code for this sequence. This takes less than $n(H(P)+\epsilon)+2$ bits. For the sequences in the compliment of $A_\epsilon^{(n)}$, it takes at most $n \log_2 |\mathcal{X}| + 2$ bits to encode. As a result, this coding scheme has average code length per symbol approximates the entropy rate on $A_\epsilon^{(n)}$.

All the codes introduced earlier when applied to the block alphabet $\mathcal{X}^n$ lead to the entropy rate in the limit. But some are easier to implement on the block than others. In particular, Huffman and Shannon codes need sorting which is a demanding computational task for a large alphabet $\mathcal{X}^n$ when n is not small. It is particularly hard to move from one block size to the next for these codes. The Shannon-Fano-Elias code, however, is easily updated when the block size changes and has acquired a new name, *Arithmetic Code*, when applied to blocks.

Example 1.1 (Arithmetic Code). We end this set of examples with an encoding scheme that builds on the the Shannon-Fano-Elias code. Assume we have a model Q that we want to use to compress strings from a particular source. The distribution Q does not have to correspond to the true data-generating distribution. Suppose we have a string $x_1, x_2, \ldots, x_n$ that we want to compress. In the simplest case, our model Q might assume that each symbol appears in the string independently. We might also consider a Markov model in which $Q(x_i) = Q(x_i|x_{i-1})$. No matter how we specify the model, it is important that we can compute the probability of x_i given the previous elements of the sequence.

Formally, we consider mapping symbols or sequences of symbols onto subintervals of $[0,1)$. Here is an outline of the method

1. We begin with a "current interval" $[L,H)$ initialized to $[0,1)$.

2. For each symbol in the string we want to compress

   (a) We subdivide the current interval into subintervals, one for each possible symbol. The size of a symbol's subinterval is proportional to the probability that the symbol will be the next to appear in the string, according to the model of the source.

   (b) We select the subinterval corresponding to the next symbol that is actually observed and make it the new current interval.

3. We output enough bits to distinguish the final current interval from all other possible final intervals.

The length of the final subinterval is equal to the product of the probabilities of the symbols; that is, $Q(x_1, \ldots, x_n) = Q(x_1)Q(x_2|x_1)\cdots Q(x_n|x_1, \ldots, x_{n-1})$. For and independent model this is just $Q(x_1)\cdots Q(x_n)$. The final step of this process requires $\lfloor -\log Q \rfloor + 1$ bits as in Shannon-Fano-Elias code. When the

Next symbol	L	H
	0.0	1.0
a	0.0	0.9
a	0.0	0.81
a	0.0	0.729
a	0.0	0.6561
a	0.0	0.59049
a	0.0	0.531441
a	0.0	0.4782969

Figure 1.7: Arithmetic coder for simple sequence.

true data generating distribution P is not known, we can use estimates of $P(x_i|x_1, ..., x_{i-1})$ based on $x_1, ..., x_{i-1}$ as $Q(x_i|x_1, ..., x_{i-1})$.

To see the benefit of this kind of process, consider a case when the probabilities of observing a symbol are slightly skewed. Consider a case in which the probability of seeing the symbol a is 0.9. We set up our probability table so that the letter a occupies the range 0.0 to 0.9. For message "aaaaaaa", the encoding process then looks like this:

Now we know what the range of low and high values are, all that remains is to use the middle-point in the interval [0, 0.4782969) to encode this message. Truncating the middle-point 0.2391485 into $\lceil -\log 0.4782969 \rceil + 1 = 3$ bits, we get the codeword 001 for the message "aaaaaaa".

1.7 Entropy

As we know, the entropy formula appeared as early as in Gibbs' works in the late 1800s and has been shown to be a lower bound on the average code length in the information inequality. Now let us formally introduce it in this section with its properties.

Given a probability function P defined on a discrete alphabet $\mathcal{X}$, we define the entropy $H(P)$ to be

$$H(P) = -\sum_{x \in \mathcal{X}} P(x) \log P(x)\,. \tag{1.21}$$

The logarithm in this expression is usually in base 2, and the units of entropy are referred to as *bits*. The entropy function enjoys the following properties which are easy to prove:

Theorem 1.6. *Let X be a random variable on $\mathcal{X}$ with distribution P. Then*

$$0 \leq H(X) \leq \log_2 |\mathcal{X}| < |\mathcal{X}|.$$

That is, the entropy of a random variable X is non-negative and is bounded by the cardinality of $\mathcal{X}$, $|\mathcal{X}|$. Equality holds if and only if X is uniformly distributed on $\mathcal{X}$. Moreover, $H(X) = H(P)$ is concave in P.

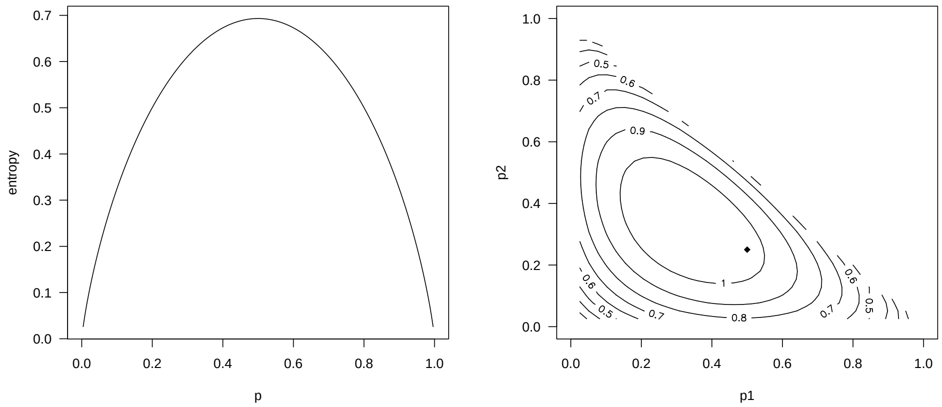


Figure 1.8: Left: Entropy of a coin toss, plotted as a function of p. Right: Entropy of a multinomial with 3 values plotted as a function of p_1 and p_2.

Proof. The entropy is non-negative because $-\log P(x) \geq 0$.

By using Lagrange multiplier, it is easily seen that $H(P)$ is maximized when $P(x) = 1/|\mathcal{X}|$ under the constraint

$$\sum_x P(x) = 1.$$

The concavity of $H(P)$ follows from the concavity of log. □

Example 1.1 (Entropy of a coin toss). Let $\mathcal{X} = \{0, 1\}$ and and $P(1) = p$. $H(p) = -p \log_2 p - (1-p)\log_2(1-p)$ with a graph. Then $H(0.5) = 1$. The function $H(p)$ is given on the left in Figure 1.8. It is easily calculated that it reaches its maximum at $p = 1/2$ and is symmetric around $p = 1/2$. The function is rather flat around its maximum as well, but is quite steep at the ends of [0,1].

Example 1.2 (Multinomial distribution). Given an alphabet with A symbols, taking probabilities $b' = p_1, \ldots, p_A$, the entropy is

$$H(b') = \sum p_i \log p_i$$

For 3 values, $b' = (p_1, p_2, p_3)$. The function $H(b')$ is given on the right in Figure 1.8, plotted as a function of p_1 and p_2. The point marked in this figure (0.5,0.25) indicates the entropy value for our toy example given above, where the the alphabet $\mathcal{X} = \{a, b, c\}$.

The entropy function can also be derived from a set of axioms as demonstrated in Shannon (1948). Below is a list of requirements that seem reasonable for a measure of information. Various authors have shown that collections of these properties are in fact sufficient to prove that entropy must take the form defined above.

Continuity $H(p_1, ..., p_k)$ is a continuous function of the vector p. This makes sense because we would not want small changes in p to yield large differences in information.

Monotonicity For $p_j = 1/m$, the entropy $H(1/m, \ldots, 1/m)$ should be an increasing function of m. When dealing with choices between equally likely events, there is more choice or uncertainty when there are more possible events.

Conditioning $H_m(p_1, \ldots, p_m) = H_{m-1}(p_1+p_2, p_2, \ldots, p_m)+(p_1+p_2)H_2(\frac{p_1}{p_1+p_2}, \frac{p_2}{p_1+p_2})$

Theorem 1.7. *If a function satisfies continuity, monotonicity, and the conditioning conditions, then*

$$H_m(p_1, \ldots, p_m) = -\sum_{i=1}^{m} p_i \log p_i \quad \text{for } m = 2, 3, \ldots$$

For a pair of random variables (X, Y), putting its joint distribution in the entropy formula gives the joint entropy of the pair which we denote by $H(X, Y)$.

Definition 2 (Joint Entropy). The joint entropy $H(X, Y)$ of a pair of discrete random variables (X, Y) with a joint distribution $P(x, y)$ is

$$H(X, Y) = -\sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} P(x, y) \log P(x, y) = -E \log P(X, Y) \tag{1.22}$$

When side information X, which is dependent on Y, is available at no cost or little cost, it is better to use X for the compression of the variable of interest Y. This is the case in distributed compression and predictive coding. For the next definition, we let $P(y|x)$ denote the conditional distribution of Y given $X = x$.

Definition 3 (Conditional Entropy). Let (X, Y) have the joint distribution function $P(x, y)$. Then the conditional entropy is defined to be

$$\begin{aligned}
H(Y|X) &= \sum_{x \in \mathcal{X}} P(x) H(Y|X = x) && (1.23)\\
&= -\sum_{x \in \mathcal{X}} P(x) \sum_{y \in \mathcal{Y}} P(y|x) \log P(y|x) && (1.24)\\
&= -\sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} P(x, y) \log P(y|x) && (1.25)\\
&= -E \log P(Y|X) && (1.26)
\end{aligned}$$

Theorem 1.8 (Chain rule for entropy).

$$H(X, Y) = H(X) + H(Y|X) \tag{1.27}$$

Proof.

$$H(X,Y) = -\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} P(x,y)\log P(x,y) \tag{1.28}$$

$$= -\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} P(x,y)\log P(x)P(y|x) \tag{1.29}$$

$$= -\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} P(x,y)\log P(x) - \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} P(x,y)\log P(y|x) \tag{1.30}$$

$$= -\sum_{x\in\mathcal{X}} P(x)\log P(x) - \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} P(x,y)\log P(y|x) \tag{1.31}$$

$$= H(X) + H(Y|X) \tag{1.32}$$

□

Example 1.3 (Experiments with News Stories). For this example, we consider all the stories from `nytimes.com` on September 15, 2003. All the entropy calculations are done in the unit of nat. This collection consists of 76,018 words

- We then ran the so-called Brill's part of speech tagger over the corpus, assigning each word one of 35 labels (noun, verb, adjective, etc.). If X is a word and Y is the part of speech, then $H(X) = 7.2$, $H(Y) = 2.8$ and $H(X,Y) = 7.4$.

- Next, if for each sentence, we let X and Y denote consecutive words in the sentence, $H(X) = 7.1$, $H(Y) = 7.3$ and $H(X,Y) = 10.1$. (Note that the entropy of X is slightly less than the previous example; we have left off the last X in each sentence.)

- Finally, suppose we let X represent words again and Y an indicator of which story the word belonged to. Our test sample has 116 stories. Then, $H(X) = 7.2$, $H(Y) = 4.7$ and $H(X,Y) = 10.0$. This means that certain words appear only in certain stories, reducing the entropy more than knowing the part of speech or the previous word.

Corollary 1.2.

$$H(X,Y|Z) = H(X|Z) + H(Y|X,Z) \tag{1.33}$$

Example 1.4. Suppose X and Y are both binary and X is uniform and the input variable into a symmetric binary channel with switching probability p and Y is the output of the channel. Their joint distribution is given in the table below, and their marginal distributions are given in the margins.

	y		
x	0	1	
0	0.5 (1-p)	0.5 p	0.5
1	0.5 p	0.5 (1- p)	0.5
	0.5	0.5	

Then $H(Y|X) = H(p)$ and $H(X,Y) = H(X) + H(Y|X) = 1 + H(p)$. When p is small, the channel noise is low so Y is very much like X and the joint entropy is close to the entropy of X.

Theorem 1.9. *$H(X|Y) \leq H(X)$ with equality if and only if X and Y are independent.*

Similarly we can define the joint entropy for any random variable vector $(X_1, ..., X_n)$:

$$H(X_1, ..., X_n) = \sum_{x_1,...,x_n} P(x_1, ..., x_n)[-\log P(x_1, ..., x_n)].$$

It follows that if $X_1, ..., X_n$ are iid, then

$$H(X_1, ..., X_n) = nH(X_1),$$

with the special case that for a binomial random variable X with a success probability p:

$$H(X) = nH(p).$$

In general, the entropy chain rule takes the form

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

because

$$P(x_1, \ldots, x_n) = \prod_{i=1}^{n} P(x_i|x_{i-1}, \ldots, x_1).$$

Theorem 1.10. *$H(X_1, ..., X_n) \leq \sum_i H(X_i)$ with equality if and only $X_1, ..., X_n$ are independent.*

If the message process is not iid, then the entropy of the marginal distribution does not reflect the coding limit since the depedences between the symbols in the process are not taken into account. However, we can still consider coding on blocks of sylmbos and this leads to the follwoing deifinition of

Definition 4 (Entropy Rate of a Stochastic Process). The entropy rate of a stochastic process $\underline{X} = (X_1, ..., X_n, ...)$ is defined as

$$H(\underline{X}) = \lim_{n\to\infty} H(X_1, ..., X_n).$$

If we take the conditional view of the process or think about predictive coding, then we arrive at an alternative definition for the entropy rate

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

For stationary processes, these two definitions are equivalent fortunately

Theorem 1.11. *For a stationary stochastic process on $\mathcal{X}$, $H(\underline{X})$ exists and equals $H'(\underline{X})$.*

Proof. Since conditioning reduces entropy,

$$\begin{aligned} & H(X_{n+1}|X_n, X_{n-1}, ..., X_2, X_1) \\ \leq \ & H(X_{n+1}|X_n, X_{n-1}, ..., X_2) \\ = \ & H(X_n|X_{n-1}, X_{n-2}, ..., X_1). \end{aligned}$$

The last equality holds because of stationarity. Hence the sequence

$$H(X_{n+1}|X_n, X_{n-1}, ..., X_2)$$

is non-negative and non-increasing so has a limit. By the chain rule on the joint entropy $H(X_1, ..., X_n)$, the averged joint entropy or $H(\underline{X})$ is the average of the above sequence and hence also shares the same limit. □

Example 1.5 (Entropy rate of a stationary Markov process). If $\underline{X}$ is a stationary Markov process of order 1,

$$H(X_n|X_{n-1}, ..., X_1) = H(X_n|X_{n-1}) = H(X_2|X_1),$$

and the entropy rate

$$H(\underline{X}) = H(X_2|X_1).$$

For a stationary and ergodic Markov chain with stationary distribution π_i and transition matrix p_{ij}, The entropy rate is

$$\sum_i \pi_i \sum_j [-p_{ij} \log p_{ij}].$$

For a two-state Markov chain with a transition Matrix

$$\begin{pmatrix} 1-p_1 & p_1 \\ p_2 & 1-p_2 \end{pmatrix}$$

and stationary distribution

$$\pi_1 = \frac{p_2}{p_1+p_2}; \pi_2 = \frac{p_1}{p_1+p_2}.$$

The entropy of X_n is

$$H(X_n) = H(\pi_1) = H(\frac{p_2}{p_1+p_2}).$$

However, the entropy rate of the sequence is LOWER due to the dependence and it is

$$H(X_2|X_1) = \frac{p_2}{p_1+p_2} H(p_1) + \frac{p_1}{p_1+p_2} H(p_2)$$

For low flip rates $p_1 = 0.01, p_2 = 0.02$, the marginal entropy is

$$H(\pi_1) = H(2/3) = 0.92(bits),$$

while the entropy rate of the sequence is

$$2/3 \times H(0.01) + 1/3 \times H(0.02) = 2 \times 0.08/3 + 0.14/3 = 0.1(bits).$$

Recall that the Clausius invented concept of entropy in Physics specifically in the statement of Second Law of Thermodynamics. We have now acquired enough basics on entropy to actually prove a version of Second Law of Thermodynamics as follows.

Theorem 1.12. *For a stationary Markov sequence $X_1, ..., X_n, ...$, the conditional entropy $H(X_n|X_1)$ is non-decreasing, while the marginal entropy $H(X_n)$ is fixed.*

Proof.

$$\begin{aligned} H(X_n|X_1) &\geq H(X_n|X_1, X_2) \ \ (property\ (3)) \\ &= H(X_n|X_2) \ \ (Markov\ property) \\ &= H(X_{n-1}|X_1) \ \ (Stationarity) \end{aligned}$$

□

Entropy is easily extendible to the continuous case by replacing summation with an integral operator. (If we try to apply the summation definition to a continuous distribution, we end up with a value of infinity.) But one may argue that for a fixed precision, the essential information about the distribution is captured by the so-called *differential entropy.*

Definition 5 (Differential Entropy). Given a continuous distribution with a positive density function f on R^d, its differential entropy is defined as

$$H(f) = -\int_{R^d} f(x) \log_2 f(x) dx, \tag{1.34}$$

with the convention that $0 \log_2 0 = 0$.

For simplicity, suppose f is non-zero only on the unit cube of R^d, then discretizing the unit cube by a small cube of size δ^d leads to a discretized distribution P_d with cell probabilities $p_\delta(x) \approx f(x)\delta^d$. This distribution has entropy

$$H(P_\delta) \approx H(f) - d \log_2 \delta. \tag{1.35}$$

When the precision increases or $\delta \to 0$, $H(P_\delta)$ tends to infinity, but the increasing part is the same for all densities if we use the same precision and hence not reflecting on the density under consideration.

Example 1.6 (Uniform Distribution). For a uniform density on $[\alpha, \beta]$, the differential entropy is

$$H_e = \ln(\beta - \alpha). \tag{1.36}$$

Example 1.7 (Gaussian Distribution). For a Gaussian density f with mean μ and variance σ^2,

$$H_e(f) = 1/2 + \ln(2\pi\sigma^2)/2. \tag{1.37}$$

Both calculations confirm that the differential entropy captures the randomness or variability in the random variable, but differential entropy can be negative, unlike entropy in the discrete case.

1.8 Estimation of Entropy

We begin with a bioinformatics example where entropy estimates are used to represent biological information.

Motifs are chromosome regions with specific biological structural significance or function. They usually short, about 6-20 base pairs. Examples include splice sites, transcription factor binding sites, translation initiation sites, enhancers, and silencers. The table below is a weight matrix learned from 15,155 mamalian donor sites (exon and intron junctions) from the SpliceDB database. Entries are frequencies of bases at each position.

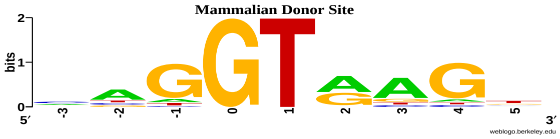
Base	-3	-2	-1	0	+1	+2	+3	+4	+5
A	33	61	10	0	0	53	71	7	16
C	37	13	3	0	0	3	8	6	16
G	18	12	80	100	0	42	12	81	22
T	12	14	7	0	100	2	9	6	46

Sequence logo is a graphical method to display patterns in a set of alligned sequences:

- Height of stack at each position is the "information" content from the frequencies:

  max. entropy - estimated entropy = 2- estimated entropy

- Letters A, T, G, C are arranged in decreasing order of frequency whose heights are proportional to the frequencies.



The entropy estimate above is the plug-in estimate.[3] In general, given an iid sequence $X_1, ..., X_n$ with probabilities $p_1, ..., p_k$ on $\{1, ..., k\}$, the plug-in estimate of entropy is based on empirical counts as follows. Let $N_j = \sum_i I(X_i = j)$ for $j = 1, ..., k$ and they are multinomial. The MLE of p's are

$$\hat{p_j} = N_j/n.$$

Then the plug-in MLE of $H(X)$ is

$$\hat{H} = H(\hat{p_1}, ..., \hat{p_k}) = -\sum_{j=1}^{k} \frac{N_j}{n} \log \frac{N_j}{n}.$$

Miller (1954) showed that the plug-in estimate of entropy has a downward bias. That is,

$$H - \hat{H} = \sum_j \frac{N_j}{n} \log \frac{N_j}{np_j} + \sum_j \{\frac{N_j}{n} - p_j\} \log p_j$$

Tbe expected bias

$$E(H - \hat{H}) = E(\sum_j \frac{N_j}{n} \log \frac{N_j}{np_j}),$$

because the second term has expectation zero. Because $2\sum_j N_j \log \frac{N_j}{np_j}$ has an approximate χ^2_{k-1} distribution,

$$E(H - \hat{H}) \approx (k-1)/(2n) + O(1/n^2).$$

The $1/n^2$ term is actually

$$(\sum \frac{1}{p_j} - 1)/(12n^2).$$

From Miller's expansion, we can easily see that when X is NOT uniform,

$$\sqrt{n}(\hat{H} - H) \to N(0, \sigma_H^2),$$

where

$$\sigma_H^2 = -\sum_{j \neq j'} p_j p_{j'} \log p_j \log p_{j'} + \sum_j p_j(1 - p_j)(\log p_j)^2.$$

When X is uniform, a faster convergence rate holds:

$$n(\hat{H} - H) \to \frac{1}{2}\chi^2_{k-1}.$$

In neuroscience, the plug-in entropy estimate is used over blocks of spike train data to arrive at entropy estimates. This method was proposed by Strong

[3] Thanks to Xiaoyue Zhao and Terry Speed for providing the data.

et al and we describe it now. For a window size T, take non-overlapping windows and estimate the joint probabilities of T-tuples and plug in these empirical joint probabilities to get entropy estimate $\hat{H}_T$ for window size T. Stationarity is implicitly assumed.

For a sequence of size n with enough mixing, one could generalize Miller's result to show that the bias of Strong et al's estimate is of order

$$O(2^T/n),$$

which follows that when $T = O(\log n)$ the bias is of order $O(1)$.

In the next example we apply Strong et al's estimate to some songbird spike train data corresponding to natural and synthetic stimuli[4]. Spike trains (0-1 sequence) are recorded on a single neuron in the auditory pathway of a song bird while sound stimuli are played to the bird. We apply Strong et al's method to two spike train sequences. The first one corresponds to natural stimulus or bird song and its length is 865 and the second corresponds to a synthetic song which is more similar to a bird song than white noise to make the neuron respond, but it is not a real bird song. The estimated entropies are given in Figure1.8 for $T = 1, ..., 10$ since $2^1 0 = 1024$ which is already larger than the lengths of both sequences. Obviously the entropy curve for the natural stimulus is higher than the one for the synthetic stimulus, but it is hard to judge whether their difference is significant or not without a measure of uncertainty for the entropy estimate. Moreover, it is even harder to interpret this difference as biologically meaningful or evidence that the neuron cell responds to natural stimulus better than to synthetic stimulus.

There is a similar downward bias of MLE plug in for estimating differential entropy. Suppose $f(x^n, \theta)$ is a parametric n-tuple density function of a stationary and ergodic sequence, $\hat{\theta}$ is the MLE of θ.

Then the MLE plug-in estimate of differential entropy rate

$$\hat{H}_n(f) = -\log f(x^n, \hat{\theta})/n,$$

underestimates $H(f_n) = E[-\log f(X^n, \theta)/n]$ under regularity conditions and the expected bias is

$$\frac{1}{2}d/n,$$

where d is the dimension of θ, because

$$-\log f(x^n, \theta) + \log f(x^n, \hat{\theta}) \approx \frac{1}{2}\chi^2_d.$$

Exercise Set 3

1. For a Poisson random variable X with mean parameter μ, prove that its entropy tends to infinity as $\mu \to \infty$.

[4] Data are kindly provided by F. Theunissen Lab, UC Berkeley.

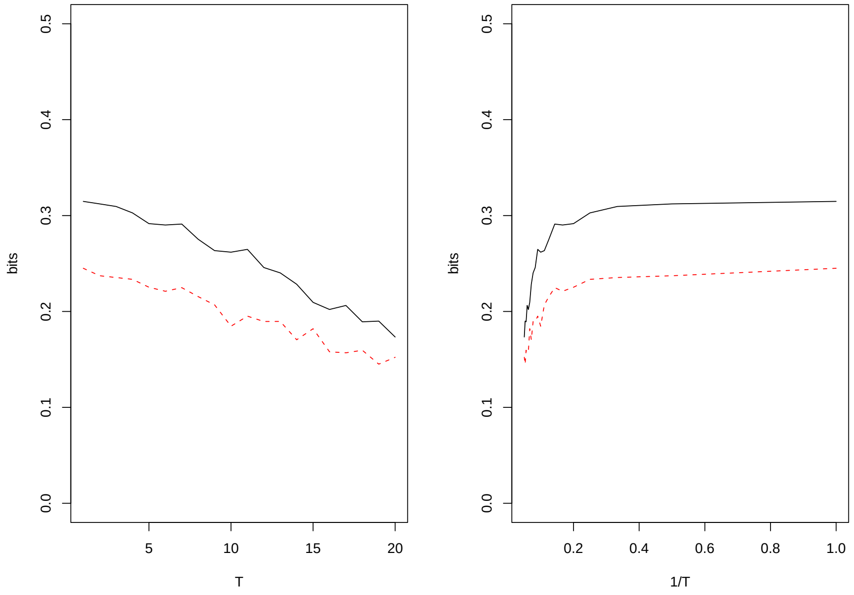


Figure 1.9: In the left panel, the upper curve gives the entropy estimates for T=1,...10 for the natural stimulus sequence and the lower curve is for the synthetci stimulus sequence. The right panel contains the same information, but it is plotted against 1/T as in Strong et al.

2. For a "nice" parametric family of dimension d, show that the expected downward bias of the plug-in differential entropy estimate based on n iid samples has a leading term $\frac{1}{2}d/n$ as $n \to \infty$.

3. Simulate iid Bernouli sequences of various sizes and various success probabilities to find out how Strong et al's method works for entropy estimation.

1.9 Maximum Entropy (Maxent) Principle: the first visit

In his famous 1957 paper ("information theory and statistical mechanics"), Ed. T. Jaynes wrote:

> Information theory provides a constructive criterion for setting up probability distributions on the basis of partial knowledge, and leads to a type of statistical inference which is called the maximum entropy estimate. It is least biased estimate possible on the given information; i.e., it is maximally noncommittal with regard to missing information.

That is to say, when characterizing some unknown events with a statistical model, we should always choose the one that has Maximum Entropy. Maximum entropy principle has been applied to a variety of fields such as computer vision, spatial statistics, and natural language processing.

Below are some well-known examples of Maxent distributions

Example 1.1 (Gaussian). If X is continuous and has known first and second moments α_i for $i = 1, 2$ and $\alpha_2 - \alpha_1^2 > 0$, then the maxent distribution is $N(\mu, \sigma^2)$ with

$$\mu = \alpha_1, \sigma^2 = \alpha_2 - \alpha_1^2.$$

Example 1.2 (Exponential). If X is positive and continuous and has a known first moment α_1, then X is exponential with mean α_1.

Example 1.3 (uniform). Uniform on a finite set $\{1, ..., k\}$ is the maxent distribution with no moment constraints.

Example 1.4 (Boltzmann). The maxent distribution on a finite set $\{1, ..., k\}$ with a first moment constraint $\alpha_1 > 0$ is

$$p_j = e^{\lambda j} / \sum_{j=1}^{k} e^{\lambda j}.$$

That is, the most probable "macrostate" (probability distribution) is $(p_1, ..., p_k)$ as above, provided that

$$\sum_j j p_j = \alpha_1$$

is fixed.

Actually maxent distributions (when they exist) are in the exponential family as stated in the next theorem.

Theorem 1.13. *The maxent distribution is the distribution that maximizes the entropy of f over all probability density functions such that*

- $f(x) \geq 0$
- $\int f(x)dx = 1$
- $\int f(x)T_i(x)dx = \alpha_i$ *for* $i = 1, ..., m$

And the maxent solution takes the form

$$f(x) = e^{\lambda_0 - 1 + \sum_{i=1}^m \lambda_i T_i(x)},$$

where the λ's are chosen to satisfy the constraints.

Proof. Here is a derivation through calculus

Let

$$J(f) = -\int f \log f + \lambda_0 \int f + \sum_i \lambda_i \int f T_i.$$

Differentiate with respect to $f(x)$:

$$\frac{\partial J}{\partial f(x)} = -\log f(x) - 1 + \lambda_0 + \sum_i \lambda_i T_i(x).$$

Setting this to zero, we get

$$f(x) = e^{\lambda_0 - 1 + \sum_{i=1}^m \lambda_i T_i(x)},$$

where the λ's are chosen to satisfy the constraints.

We can now verify such an f does have the maximum entropy. If suffices to show that for any g satisfying all the constraints:

$$H(f) - H(g) \geq 0.$$

$$\begin{aligned}
H(f) - H(g) &= -\int f(x) \log f(x) dx + \int g(x) \log g(x) dx \\
&= -\int f(x)(\lambda_0 - 1 + \sum_{i=1}^m \lambda_i T_i(x)) dx + \int g(x) \log g(x) dx \\
&= -\int g(x)(\lambda_0 - 1 + \sum_{i=1}^m \lambda_i T_i(x)) dx + \int g(x) \log g(x) dx \\
&= -\int g(x) \log f(x) dx + \int g(x) \log g(x) dx \\
&= \int g(x) \log[g(x)/f(x)] dx
\end{aligned}$$

The last expression is non-negative due to information inequality.

□

Chapter 2

Relative Entropy or Kullback Leibler Divergence

Now let us go back to coding. We have seen in the previous chapter that when we know the message generating distribution P, we know how to do optimal coding or near-optimal coding via Huffman, Shannon, Shannon-Fano-Elias codes. But what happens if we don't know P and use a wrong code?

We consider again the creation of codes, this time with an eye toward their performance. Suppose we create a code based on the frequency of words found in the New York Times in 2004, and then use it to encode stories from this year. We collected all the articles from the issues appearing April 20, 2004 and March 9, 2005 In the two issues of the paper, we collected over 17,213 words (including numbers and abbreviations), with 7,457 appearing only once Let $P(x)$ and $Q(x)$ denote the frequency of word x from the 2004 and 2005 issues, respectively The entropies are given by

$$\sum_x Q(x) \log \frac{1}{Q(x)} = 12.77 \quad \text{and} \quad \sum_x P(x) \log \frac{1}{P(x)} = 12.94$$

The most frequent words appearing in these texts do not carry content; they are pronouns, articles and prepositions We might expect that many of these non-content words appear in roughly the same proportions in 2004 as in 2005

word	n.04	freq.04	bits.04	n.05	freq.05	bits.05
the	6375	0.0626	4	5783	0.0622	5
to	2777	0.0273	6	2543	0.0274	6
of	2708	0.0266	6	2365	0.0254	6
a	2557	0.0251	6	2497	0.0269	6
and	2338	0.0230	6	2137	0.0230	6
in	2248	0.0221	6	2107	0.0227	6
that	1377	0.0135	7	1315	0.0142	7

said	972	0.0096	7	1027	0.0111	7
for	952	0.0094	7	893	0.0096	7
he	901	0.0089	7	741	0.0090	7

While many of the non-content words seem to have similar distributions between 2004 and 2005, what about the people and places that make up the news? The who, what, where, when and why of the daily news certainly changes from year to year

The next table lists the words that gained popularity from 2004 to 2005.

word	n.04	freq.04	bits.04	n.05	freq.05	bits.05	diff
lebanon	1	1.96e-05	16	49	5.38e-04	11	5
lebanese	1	1.96e-05	16	47	5.16e-04	11	5
arts	0	9.82e-06	17	34	3.76e-04	12	5
bolton	0	9.82e-06	17	28	3.12e-04	12	5
hezbollah	1	1.96e-05	16	28	3.12e-04	12	4
march	30	3.04e-04	12	103	1.12e-03	10	2
prison	10	1.08e-04	14	27	3.01e-04	12	2
syria	9	9.82e-05	14	30	3.33e-04	12	2

The second table gives words that dropped in popularity from 2004 to 2005.

word	n.04	freq.04	bits.04	n.05	freq.05	bits.05	diff
saatchi	41	4.12e-04	12	0	1.08e-05	17	-5
dvd	32	3.24e-04	12	0	1.08e-05	17	-5
cantalupo	32	3.24e-04	12	0	1.08e-05	17	-5
april	111	1.10e-03	10	15	1.72e-04	13	-3
bonds	57	5.69e-04	11	8	9.68e-05	14	-3
kerry	43	4.32e-04	12	3	4.30e-05	15	-3
tax	32	3.24e-04	12	8	9.68e-05	14	-2
campaign	58	5.79e-04	11	26	2.90e-04	12	-1

We are ready to quantify the difference of the two codes. Depending which year we apply the codes, the expected difference is taking over a different distribution. Let $Q(x)$ and $P(x)$ denote the probabilities of word x in 2004 and 2005, respectively We can then write the difference in code lengths as

$$\log\frac{1}{Q(x)} - \log\frac{1}{P(x)} = \log\frac{P(x)}{Q(x)}$$

If the code from 2004 gives a shorter codeword, this quantity is negative; when the code from 2004 assigns a longer codeword, it is positive. Averaging over the distribution of words from the 2005 paper, the expected difference in code lengths is given by

$$\sum_x P(x)\log\frac{P(x)}{Q(x)}$$

Can this quantity ever be negative? More to the point, by using the 2004 distribution an we ever achieve (on average) a shorter encoding of the paper in 2005?

In fact, we know from the Information Inequality that the difference in average code lengths must be non-negative That is, we know that

$$\sum_x P(x)\log\frac{1}{Q(x)} - \sum_x P(x)\log\frac{1}{P(x)} = \sum_x P(x)\log\frac{P(x)}{Q(x)} \geq 0$$

If we use our code from 2004, we can encode the 2005 paper with a average code length of

$$\sum_x P(x)\log\frac{1}{Q(x)} = 13.29$$

Instead, if we build a code using the frequencies from 2005, we have an average code length equal to the entropy, or

$$\sum_x P(x)\log\frac{1}{P(x)} = 12.94$$

The difference is 13.29 - 12.94 = 0.35

Before we leave this example, it is worth questioning why we might consider words rather than characters as you have done for your lab. Users of Morse code, for example, eventually introduced another layer of compression onto the code; Five-letter codewords like LIOUY and AYYLU were introduced to represent complete phrases (in this case “Why do you not answer my question” and “Not clearly coded, repeat more clearly”). Shannon also toyed with representing English by character- and then word-based codes.

What is not mentioned here is the effort required to decode a message; word-based models require large codebooks In the next section of this workshop, we compare different codes or “models” for a data set and explicitly balance the compression achievable with a given code against the cost of representing its codebook This will yield the principle of minimal description length; but before we get ahead of ourselves...

2.1 Kullback-Leibler Divergence

In this section we formally define the Kullback-Leibler divergence between two probability distributions P and Q which measures the coding loss or redundancy when a wrong code book is used. We have in fact seen this quantity in Chapter 1 in information inequality and

Definition 6. Kullback-Leibler Divergence

$$D(P\|Q) = \sum_{x\in\mathcal{X}} P(x)\log\frac{P(x)}{Q(x)}$$

where $0\log 0/q = 0$ and $p\log p/0 = \infty$

As to be seen, in many respects KL divergence acts as a measure of dissimilarity or "distance" between distributions, but it is not symmetric and does not satisfy the trianglular inequality. It nevertheless has some desirable properities as shown below.

Non-negativity (information inequality): $D(P\|Q) \geq 0$ with equality if and only if $P(x) = Q(x)$, $x \in \mathcal{X}$.

Convexity: Given pairs of distributions P_1, Q_1 and P_2, Q_2 ,

$$\lambda D(P_1\|Q_1) + (1-\lambda)D(P_2\|Q_2) \geq D(\lambda P_1 + (1-\lambda)P_2\|\lambda Q_1 + (1-\lambda)Q_2)$$

for any $\lambda \in [0, 1]$.

Obvious from the coding exercise of the news paper stories from 2004 and 2005, Kullback-Leibler divergence represents a coding a loss when the wrong code book or wrong distribution is used. It is called redundancy in source coding. That is, returning to our coding example, suppose we create code lengths

$$L(x) = \left\lceil \log \frac{1}{Q(x)} \right\rceil$$

for some distribution Q. Assuming the true distribution of the source is P, we can easily show that expected code length satisfies

$$H(P) + D(P\|Q) \leq EL < H(P) + D(P\|Q) + 1$$

In their original paper, Kullback and Leibler cast their divergence measure as a tool for distinguishing between statistical populations In fact, they refer to the quantity

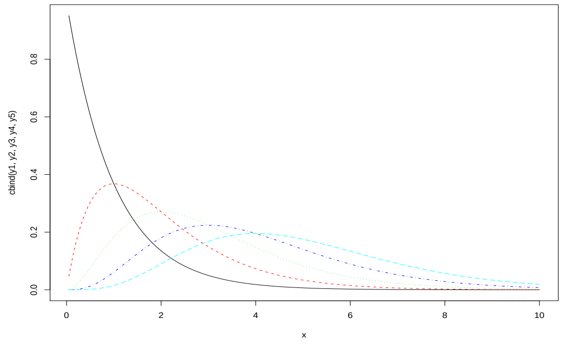
$$\log \frac{P(x)}{Q(x)}$$

as "the information in x for discrimination between" the distributions P and Q (we recognize this as the logarithm of likelihood ratio, or Bayes factor assuming a uniform prior on the two distributions). Their divergence is then the mean information for discrimination per observation from P.

Example 2.1 (the gamma family). In Figure 2.1 we have plotted the densities for five gamma distributions. The scale parameter for each is just 1, and the shape parameters are 1, 2, 3, 4 and 5 (tracking the modes from left to right).

We present the pairwise KL divergences in the form of a (dis)similarity matrix. Within each row, the largest distances occur in the first column; as expected the exponential distribution looks quite distinct. Notice the asymmetry in the table, it is also the most extreme for entries involving the first column/row.

```
          shape parameter
        1     2     3     4     5
```



	1	0.00	0.54	1.78	3.42	5.34
shape	2	0.42	0.00	0.27	0.95	1.91
parameter	3	1.15	0.23	0.00	0.18	0.64
	4	1.98	0.72	0.16	0.00	0.13
	5	2.85	1.34	0.53	0.12	0.00

A good property of KL is its scale-invariance which is not enjoyed by the L^2 distance for example. In the gamma example, we are essentially computing the distance between the distributions for a single observation. If we instead consider n independent observations from P and Q, $x = (x_1, \ldots, x_n) \in \mathcal{X}^n$, we have

$$D(P^n \| Q^n) = \sum_{x \in \mathcal{X}^n} P^n(x) \log \frac{P^n(x)}{Q^n(x)} = nD(P\|Q)$$

In KL divergenece, we can clearly see the accumulation of information to distinguish two distributions when the sample size n increases. The KL divergence of the product distributions is proportional to the sample size.

In many language applications, documents are treated naively as bags of words. The frequency of words becomes a clue to their content; documents that use words in roughly in the same proportion are said to be similar. What problems might we encounter applying KL divergence directly to this kind of data?

The first problem with KL divergence is its asymmetry. The symmetry issue is often addressed by defining a new measure that is the sum

$$D(P\|Q) + D(Q\|P)$$

This is often called the J-divergence and was originally studied by Jeffreys in the context of identifying invariant prior distributions Jeffreys wrote the J-divergence in the form

$$D(P\|Q) + D(Q\|P) = \sum_x [P(x) - Q(x)] \log \frac{P(x)}{Q(x)}$$

The second problem with KL divergence is when the two distributions do not share the same support, the KL divergence is technically infinity. While this may be sensible in some applications, it is a bit extreme in some situations like the document example. To soften the blow, Lin (1991) proposed the Jensen-Shannon divergence

$$JS(P,Q) = \frac{1}{2}D\left(P\|\frac{P+Q}{2}\right) + \frac{1}{2}D\left(Q\|\frac{P+Q}{2}\right)$$

It is easy to check that

$$JS(P,Q) = \frac{1}{2}H\left(\frac{P+Q}{2}\right) - \frac{1}{2}H(P) - \frac{1}{2}H(Q)$$

Moreover, JS-divergence is at least twice as large as the J-divergence. It is also a convenient dissimilarity measure between documents which are summarized by word frequencies or distributions.

Theorem 2.1. *For any distributions P and Q,*

$$J(P,Q) \leq \frac{1}{2}JS(P,Q).$$

Proof. Enough to show $D(P||\frac{P+Q}{2}) \leq \frac{1}{2}D(P||Q)$.

□

Example 2.2 (News stories). Let's consider stories taken from a single day of the New York Times, January 2, 2004. We applied the JS divergence to pairs of stories to compute a (symmetric) dissimilarity matrix. A simple visualization device was used to check the output for clusters, categorizing stories solely on their section of the paper.

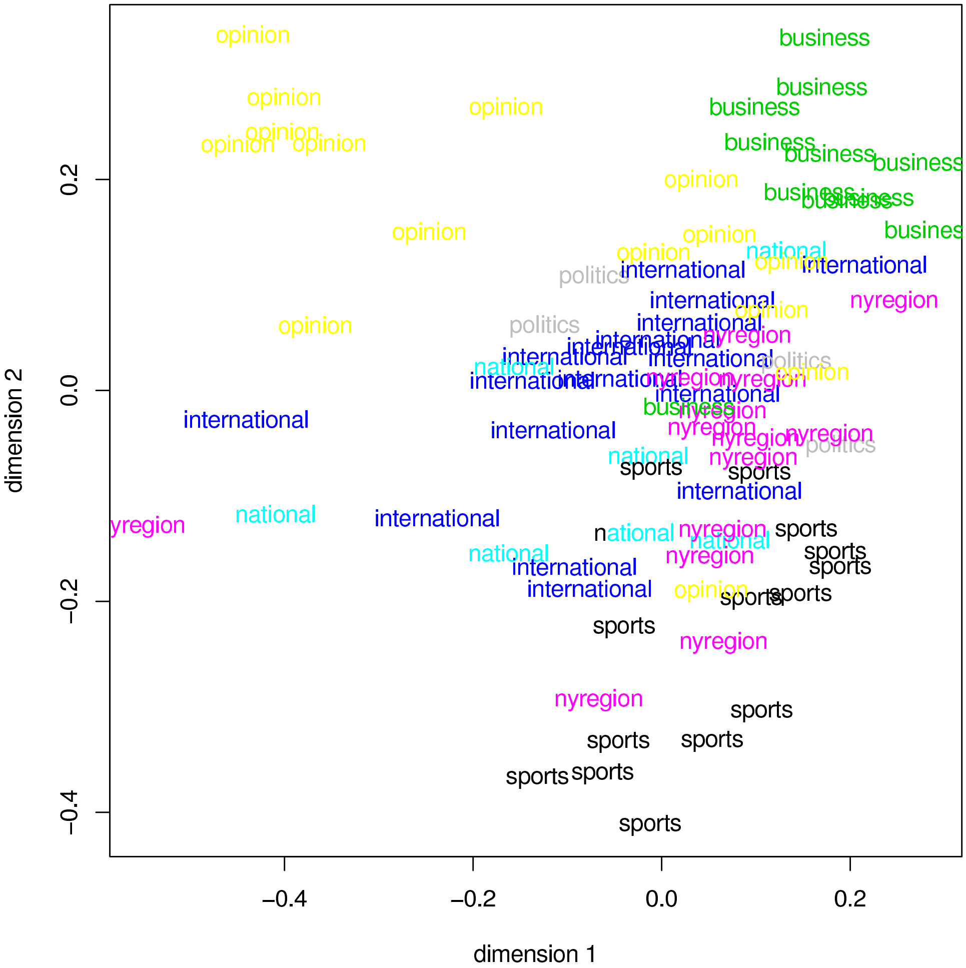
Figure 2.1 shows the result of simple multidimensional scaling (MDS) applied to the JS dissimilarity matrix. MDS is widely used in Psychology, Social Science and Economy. In general, MDS maps the units (documents in our case) to points in a low dimensional Euclidean space to minimize some stress measure such as

$$\sum_{i,j}(d_{ij} - JS(document_i||document_j)^2,$$

where d_{ij} is the reproduced distance in R^2 of the mapped points. For our example, we used the R function cmdscale (classical multidimensional scaling). The cmdscale algorithm is based on Gower, J. C. (1966) which uses the principle coordinate system for the mapping.

Notice the isolation of certain categories like business and sports in Figure 2.1. It also makes sense to see the cagetory "nyregion" has overlaps witih other categories like sports and international.

Relative entropy or KL divergence is easily extendible to the continuous case by replacing summation with an integral operator. we try to apply the summation definition to a continuous distribution, we end up with a value of infinity.



That is, for two density functions f and g, this replacement of summation by integration in D works because the precision terms from discretizing f and g cancel each other out. Hence we have

Definition 7 (Differential KL Divergence). Given two probability densities f and g,

$$D(f||g) = \int f(x) \log[f(x)/g(x)]dx.$$

For mathematical convenience, we denote by $K(f, g)$ the Kullback-Leibler divergence when log is taken with respect to base e:

$$K(f,g) = \int f(x)\ln[f(x)/g(x)]dx = \ln 2D(f,g).$$

Recall that differential entropy which could be negative and is not the limit of the discrete entropy when precison tends to zero. The differential KL divergence is a well-defined limit of the discretized version of the KL divergence becase the precision terms from discretizing f and g cancel each other out in the ratio.

2.2 KL as a dissimilarity measure on distributions

While the relative entropy is used as a kind of distance measure between distributions, it does not satisfy some important properties. The most obvious difficulty is that it is not symmetric. It also does not satisfy the so-called triangle inequality. Hence it is not a distance on the space of probability measures, but it nevertheless measures the "dis-similarity" of two probability measures and arises naturally in many statistical contexts. It is actually acts like a "distance" in some sense. In particular, it is closely related to other proper distances such as the L_1 and L_2 norms and the Hellinger distance as revealed in the inequalities in the following theorem. For simplicity, we assume the two probability measures share the same probability sample space and have density functions f and g with respect to the same dominating measure μ (e.g., the counting measure in the discrete case and the Lebesgue measure in the continuous case).

Definition 8 (L_1 norm).

$$L_1(f,g) = \int |f(x) - g(x)| d\mu(x).$$

Definition 9 (L_2 norm).

$$L_2^2(f,g) = \int |f(x) - g(x)|^2 d\mu(x).$$

Definition 10 (Hellinger distance).

$$H^2(f,g) = \int |\sqrt{f(x)} - \sqrt{g(x)}|^2 d\mu(x).$$

As L_1 and Hellinger distances, KL-divergence is scale-invariant, but L_2 is not.

Theorem 2.2. *The following inequalities hold on the various"distances" between probability measures:*

1. *Obviously,*
$$L_1(f,g) \le L_2(f,g).$$

2.

$$H^2(f,g) \le L_1(f,g) \le H(f,g)\sqrt{4-H^2(f,g)} \le 2H(f,g).$$

3.

$$2 - L_1(f,g) \le \{1 - H^2(f,g)/2\}^2.$$

4. Pinker's inequality:

$$L_1(f,g) \le \sqrt{2K(f||g)}.$$

5. Bretagnolle-Huber inequalities:

$$L_1(f,g) \le 2\sqrt{1-e^{-K(f||g)}} \le 2 - e^{-K(f||g)}.$$

Proof. The first inequality follows from Cauchy-Schwartz inequality.

□

Hence D is a stronger "distance" measure than L_1^2, and L_2 and H are stronger than L_1 too. In some sense, D behaves as a squared distance. More importantly, it arises as the natural measure of dissimilarity in many different contexts, for example, MLE for misspecified models, Large deviations, and hypothesis testing.

Much theory has been carried out for density estimation in terms of L_1, L_2 or H. We will see later in this course how using D simplifies and unifies the upper and lower bounding in minimax density estimation over smooth classes of functions.

2.3 KL and MLE

2.3.1 Basic Classical Maximum Likelihood Theory

Assume we have a parametric family of probability models $g \in \mathcal{G}$ where $g = g(x;\theta)$ and the data generating distribution belongs to this family. The maximum likelihood estimator $\hat{\theta} = \arg\max_\theta g_\theta(x_1, \ldots, x_n)$ is optimal in the sense that it achieves equality in the Cramer-Rao information bound if the true distribution belongs to the parametric model: for any unbiased estimator $\hat{\theta}_n$

$$E(\hat{\theta}_n - \theta)(\hat{\theta}_n - \theta)' \ge J_n(\theta)^{-1}, \tag{2.1}$$

where $J_n(\theta)$ is the Fisher information matrix,

$$J_n(\theta) = E_\theta \left(\frac{\partial}{\partial\theta} \ln g_\theta(X_1, \ldots, X_n)\right)\left(\frac{\partial}{\partial\theta} \ln g_\theta(X_1, \ldots, X_n)\right)'. \tag{2.2}$$

If $X_1, ..., X_n$ are iid, then

$$J_n = nJ_1,$$

a property similar to Shannon's entropy. Also similar to Shannon's entropy, Fisher information characterize the limit of possible for parameter estimation if the true distribution is contained in the parametric family.

There is an equivalent expression for the Fisher information. We now introduce some notation. Let the log-likelihood function be

$$l_n(\theta) = \ln g_\theta(X_1, ..., X_n),$$

and the score vector

$$U_n = (U^1, ..., U^d)^T = (\partial l_n/\partial\theta_1, ..., \partial l_n/\partial\theta_d)^T,$$

with

$$E_\theta U_n = (0, ..., 0)^T.$$

Then

$$J_n = [-E_\theta \frac{\partial^2 l_n}{\partial\theta_i \partial\theta_j}]_{d\times d}.$$

The proof is straightforward from the following facts

$$E_\theta \frac{\partial l_n}{\partial\theta_i} = 0.$$

$$\partial l_n/\partial\theta_i = \frac{\partial g_\theta}{\partial\theta_i}/g_\theta.$$

$$\frac{\partial^2 l_n}{\partial\theta_i\partial\theta_j} = \frac{\partial^2 g_\theta(X^n)}{\partial\theta_i\partial\theta_j}/g_\theta - \frac{\partial g_\theta(X^n)}{\partial\theta_i}\frac{\partial g_\theta(X^n)}{\partial\theta_j}/g_\theta^2.$$

Expanding $0 = \frac{\partial l_n}{\partial\theta_i}(\hat\theta)$ around the true parameter θ, we get

$$\sqrt{n}(\hat\theta - \theta) \approx [J_n(\theta)]^{-1} \times \sqrt{n}U_n.$$

The asymptotic normality of the maximum likelihood estimator follows:

$$\sqrt{n}(\hat\theta - \theta) \to_d N(0, J_1^{-1}),$$

provided that the observations are iid and the parametric family is "nice".

All the information we gather from an iid sample is about the underlying probability distribution or density and in a parametric family this information has to be mapped into the parameter space for parameter estimation. If we measure the distance between the distribution by the KL divergence K, the following equation confirms the pivotal role that the Fisher information plays in parameter estimation:

$$\lim_{\Delta\theta\to 0} \frac{1}{(\Delta\theta)^2} K(g_\theta \| g_{\theta+\Delta\theta}) = \frac{1}{2}J_1(\theta), \tag{2.3}$$

which says that when two members of the parametric family are close, the "distance" in terms of KL divergence between them are quadratic in terms of the the Euclidean distance between the two parameters and the curvature is determined by the Fisher information up to a universal constant.

Before we close this section, it is worth going through the proof of Cramer-Rao bound in the 1-dim case to highlight once more the central role of the score statistic.

Proof. For any estimator $T(X)$ of θ, we look at the projection of the centered $T(X)$ which can be upper bounded by Cauchy-Schwartz inequality:

Because $E_\theta U(X) = 0$,

$$E_\theta UT = E_\theta[(U - E_\theta U)(T - E_\theta T(X))] \leq E_\theta(U - E_\theta U)^2 E_\theta(T - E_\theta T)^2 = J(\theta)V_\theta(T).$$

So what matters in the centered $T(X)$ is its projection onto the direction of the score statistic which is the direction that MLE follows, asymptotically, providing the most efficient estimator.

On the other hand,

$$\begin{aligned} E_\theta UT &= \int f_\theta(x) f'_\theta(x)/f_\theta(x)T(x)dx \\ &= \int f'_\theta(x)T(x)dx \\ &= d/d\theta \int f\theta(x)T(x)dx \\ &= d/d\theta\theta \\ &= 1. \end{aligned}$$

That is,

$$J(\theta)V_\theta(T) \geq 1.$$

□

2.3.2 Limit of MLE when the model is misspecified

When the true density f is not in the parametric family. Let θ^* be such that the K-L divergence satisfies

$$E_f \log f(X)/g(X;\theta^*) < E_f \log f(X)/g(X;\theta) \tag{2.4}$$

for all $\theta \neq \theta^*$. Then, we have the following lemma.

Lemma 2.1. *Assume that $g(x;\theta)$ are distinct for all θ; that the $g(x;\theta)$ and f have common support; and that $X_1, \ldots, X_n$ are iid according to f. Then, as $n \to \infty$ we have that*

$$P_f\left(g(X_1;\theta^*)\cdots g(X_n;\theta^*) > g(X_1;\theta)\cdots g(X_n;\theta)\right) \to 1\,. \tag{2.5}$$

Proof. The inequality is equivalent to

$$\frac{1}{n}\sum_i \log g(X_i;\theta)/g(X_i;\theta^*) < 0\,. \tag{2.6}$$

By the LLN, the left side tends in probability toward

$$E_f \log g(X;\theta)/g(X;\theta^*)\,. \tag{2.7}$$

But, rewriting

$$\begin{align} E_f \log g(X;\theta)/g(X;\theta^*) &= E_f \log g(X;\theta) - E_f \log g(X;\theta^*) \tag{2.8}\\ &= -E_f \log f(X)/g(X;\theta) + E_f \log f(X)/g(X;\theta^*) \tag{2.9}\\ &< 0\,. \tag{2.10} \end{align}$$

□

Theorem 2.3. *In addition to the assumptions of the lemma, let the parameter space for θ contain an open interval around θ^*. Assume that $g(\cdot;\theta)$ is differentiable with respect to θ in the open interval – which we denote g'. Let $l_n(\theta;x) = \sum \log g(x_i;\theta)$ denote the log-likelihood. Then with probability $\to 1$ as $n \to \infty$,*

$$l_n'(x;\theta) = \sum_i \frac{g'(x;\theta)}{g(x;\theta)} = 0 \tag{2.11}$$

has a root $\hat{\theta}_n$ such that $\hat{\theta}_n \to \theta^*$ in probability.

Proof. Let a be small enough that $(\theta^* - a, \theta^* + a)$ is in the open interval containing θ^*. Let

$$S_n = \{x : l_n(\theta^*;x) > l_n(\theta^* - a, x) \text{ and } l_n(\theta^*;x) > l_n(\theta^* + a, x)\}\,. \tag{2.12}$$

By the above lemma, $P_f(S_n) \to 1$. For any $x \in S_n$ there exists a value $\theta^* - a < \hat{\theta}_n < \theta^* + a$ at which $l_n(\theta)$ has a local maximum. That means $l_n'(\hat{\theta}_n) = 0$. Therefore for any $a > 0$ sufficiently small, there exists a sequence $\hat{\theta}_n$ that depends on a such that

$$P_f(|\hat{\theta}_n - \theta^*| < a) \to 1\,. \tag{2.13}$$

In the case of multiple roots, we consider $\tilde{\theta}_n$ which is the root closest to θ^*. Then, $P_f(|\tilde{\theta}_n - \theta^*| < a) \to 1$. □

Now we have shown the limit of MLE when the model is misspecified, we could almost repeat the proof for the asymptotic normality of the MLE when the model is correct, but not quite. For simplicity, let's assume $d = 1$, then we still have

$$\hat{\theta} - \theta^* \approx -l_n'(\theta^*)/l_n''(\theta^*) \tag{2.14}$$

For iid data from distribution f, the two definitions of Fisher information part their ways. The first definition gives

$$J_1(\theta^*) = Var_f(U_1(\theta^*) = E_f U_1^2(\theta^*),$$

because

$$E_f U_1(\theta^*) = E_f[\frac{\log g(X,\theta^*)}{d\theta}] = 0$$

due to the fact that $E_f[-\log g_\theta(X)]$ is minimized at $\theta = \theta^*$. It is interesting to note that the score statistics U_1 still has mean zero even though the model doesn't contain the true distribution.

The second definition of Fisher information is

$$J_2(\theta^*) = E_f[-l_1''(\theta^*)].$$

Properly normalizing the numerator and the denomerator in (2.14) to obtain the asymptotic limits:

$$n^{-1/2} l_n'(\theta^*) \to_d N(0, J_1(\theta^*),$$

and

$$\frac{1}{n} l_n''(\theta^*) \to_P J_2(\theta^*),$$

Hence the asymptotic distribution of MLE is

$$N(\theta^*, J_1(\theta^*)/(n \times J_2^2(\theta^*)).$$

When f belongs to the parametric family $f = g_{\theta_0}$, we recover the classical asymptotic normality result for the MLE because

$$J_1 = J_2.$$

Exercise Set 4

1. Prove the convexity of KL divergence. That is, Given pairs of distributions P_1, Q_1 and P_2, Q_2 ,

   $$\lambda D(P_1 \| Q_1) + (1-\lambda) D(P_2 \| Q_2) \geq D(\lambda P_1 + (1-\lambda) P_2 \| \lambda Q_1 + (1-\lambda) Q_2)$$

   for any $\lambda \in [0, 1]$.

2. Professor Mark Hansen at UCLA obtained the documents on Senators' contributions to the Roberts' confirmation hearing and put them at

   qual.stat.ucla.edu/202/roberts

   which contains files like

   transcript1.txt (transcripts)

   KENNEDY.cnt (word counts of Kennedy's transcript)

The JS distance matrix between the documents (or the frequencies of the words in the documents) is at

www.stat.berkeley.edu/ binyu/212A/hearing.txt

Use the R function cmdscale on the distance matrix to find a 2-dim representation of the documents and comment on the plot.

Moreover, read the following paper on the algorithm used in cmdscale and write a concise and clear description of the algorithm (if neccessary, read other papers cited in the Gower paper):

Gower, J. C. (1966) Some distance properties of latent root and vector methods used in multivariate analysis. *Biometrika*, **53**, 325-328.

3. Calculate the distance matrices based on L_1, L_2 and Hellinger distances between documents based on the data at the website, and then do an MDS for the three cases and plot the results.

2.4 Mutual Information

For a pair of random variables or a vector of them, an important question is on how much information they contain about each other; or how depedent they are. Mutual information to be defined below answers this question naturally and is the key ingredient in channel coding.

Let X, Y have a joint distribution $P(x, y)$, and have marginal distributions $P(x)$ and $P(y)$. The mutual information $I(X; Y)$ is the relative entropy or KL divergence between the joint distribution of X, Y and the product of their marginals; or rather, how far the joint distribution $P(x, y)$ is from independence. It therefore measures the dependence between the two variables – the larger the mutual information, the more dependence they are.

Definition 11 (Mutual Information).

$$I(X;Y) = \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} P(x,y)\log\frac{P(x,y)}{P(x)P(y)}, = KL(P(x,y)||P(x)P(y). \qquad (2.15)$$

$I(X, Y) = 0$ if and only if X and Y are independent. The definition of $I(X, Y)$ can easily be extended to the continuous case by replacing the summation by an integration.

Using the definitions of mutual information and conditional entropy, we find the following expression.

$$\begin{aligned} I(X;Y) &= \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} P(x,y)\log\frac{P(x,y)}{P(x)}P(y) & (2.16)\\ &= \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} P(x,y)\log\frac{P(y|x)}{P(y)} & (2.17)\\ &= \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} P(x,y)\log P(y|x) - \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} P(x,y)\log P(y) & (2.18)\\ &= \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} P(x,y)\log P(y|x) - \sum_{y\in\mathcal{Y}} P(y)\log P(y) & (2.19)\\ &= H(Y) - H(Y|X) & (2.20) \end{aligned}$$

Also, recall that $H(Y) - H(Y|X) = H(X) - H(X|Y)$. Therefore, the mutual information records the reduction in uncertainty of X by knowing Y and, equivalently, the reduction in uncertainty in Y after knowing X. This is also the amount of information that X or Y contains about the other.

Example 2.1 (Bivariate Gaussian). Assume (X, Y) are normal with mean zero and variance 1 and correlaction coefficient ρ. It is straghtforward to calculate that

$$I(X,Y) = -\frac{1}{2}\ln(1-\rho^2).$$

This is a monotonic transformation of the correlation coefficient which is the only measure of dependence in the Gaussian case. The closer $|\rho|$ to 1, the larger the mutual information. The mutual information does not differetiate a positive correlation with a negative one, however.

Example 2.2 (Mutual information and document similarity). Here we consider the use of mutual information in a text mining capacity.

It is easy to verify the chain rule for mutual information based on the chain rule for entropy and the entropy difference expression of mutual information:

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

Inherited from properties of K, I has the following concavity and convexity:

$I(X,Y)$ is concave in $P(x)$ for fixed $P(y|x)$.

$I(X,Y)$ is convex in $P(y|x)$ for fixed $P(x)$.

2.4.1 Sufficiency

In the age of massive data collection, it is crucial to understand that manipulation of data doesn't increase the information. This is stated most concisely in the data processing inequality.

Theorem 2.4 (Data Processing Inequality). *If $X \to Y \to Z$ are Markov, then*

$$I(X,Y) \geq I(X,Z). \tag{2.21}$$

Proof. Using the chain rule on mutual information,

$$I(X;Z,Y) = I(X,Z) + I(X,Y|Z) \geq 0.$$

Similarly,

$$I(X;Y,Z) = I(X,Y) + I(X,Z|Y) = I(X,Y)$$

since X and Z are independent given Y. Therefore

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

□

Corollary 2.1. *For any function g, $I(X,Y) \geq I(X,g(Y))$.*

Let $f_\theta(x)$ be a parametric family. Our goal is to make inferences about θ from sample $x_1,\ldots,x_n \sim f_\theta(x_1,...,x_n)$. A statistic $T(x_1,\ldots,x_n)$ is sufficient if given $T(X_1,...,X_n)$, θ and X are independent or the distribution of $X_1,...,X_n$ given $T(X_1,...,X_n)$ is not dependent on θ.

One can find a sufficient statistic in a straightforward manner by capitalizing the factorization theorem:

Theorem 2.5 (Factorization Theorem). *T is a sufficient statistic for f_θ if and only*

$$f_\theta(x_1,\ldots,x_n) = g_\theta(T(x_1,\ldots,x_n))h(x_1,\ldots,x_n|T(x_1,\ldots,x_n)) \tag{2.22}$$

where g and h are two nonnegative functions and $h(x_1,\ldots,x_n|T(x_1,\ldots,x_n))$ does not depend on θ.

For example, if $x_1,\ldots,x_n$ are i.i.d. Gaussian with mean θ and variance 1, $T(\underline{x}) = \bar{x}$ is sufficient.

Theorem 2.6 (Mutual Information and Sufficiency). *T is a sufficient statistic for $f_\theta(X)$ iff*

$$I(\Theta,X) = I(\Theta,T(X).$$

Proof. Since T is a mapping from X, we have

$$\underline{X} \to T(\underline{X}),$$

which implies, by the data processing inequality, that

$$I(\Theta,T(\underline{X})) \leq I(\Theta,\underline{X}). \tag{2.23}$$

On the other hand, T is a sufficient statistic,

$$\Theta \to T(\underline{x}) \to \underline{x},$$

so

$$I(\Theta, T(\underline{X})) \geq I(\Theta, \underline{X}). \tag{2.24}$$

Combining two inequalities gives $I(\Theta, T(\underline{X})) = I(\Theta, \underline{X})$, so the mutual information of the parameter and the data is the same as mutual information of the parameter and the sufficient statistic. To prove the converse, we provide another proof of the theorem without using the Data Processing Inequality in the following lemma. □

Lemma 2.2. *Suppose X has density $f_\theta(x)$, $T(X)$ has density $g_\theta(T(x))$, and m_g and m_f are the marginal densities of $T(X)$ and X respectively:*

$$m_g(T(x)) = \int \pi(\theta) g_\theta(T(x)) d\theta.$$

$$m_f(T(x)) = \int \pi(\theta) f_\theta(x) d\theta.$$

Then

$$Q(\theta, x) = \pi(\theta) g_\theta(T(x))/m_g(T(x)) \times m_f(x)$$

is a joint density of θ and x, that is

$$\int \int Q(\theta, x) d\theta dx = 1.$$

And

$$I(\Theta, \underline{X}) - I(\Theta, T(\underline{X})) = K(\pi(\theta) * f_\theta(x), Q(\theta, x)).$$

Proof.

$$\begin{aligned}
\int \int Q(\theta, x) d\theta dx &= \int \int \pi(\theta) g_\theta(T(x))/m_g(T(x)) \times m_f(x) d\theta dx \\
&= \int [\int \pi(\theta) g_\theta(T(x))] d\theta / m_g(T(x)) \times m_f(x) dx \\
&= \int [\int \pi(\theta) g_\theta(T(x)) d\theta] / m_g(T(x))] \times m_f(x) dx \\
&= \int [m_g(T(x))]/m_g(T(x))] \times m_f(x) dx \\
&= \int m_f(x) dx \\
&= 1.
\end{aligned}$$

□

It follows from the non-negativity of Kullback-Leibler divergence that $I(\Theta, X) \geq I(\Theta, T(X))$. And we can give a complete proof of the theorem using this expression of the difference of the mutual informations.

$$K(\pi(\theta) * f_\theta(x), Q(\theta, x)) = 0$$

iff

$$\pi(\theta) * f_\theta(x) = Q(\theta, x))$$

iff

$$f_\theta(x) = g_\theta(T(x)) m_f(x)/m_g(T(x))$$

iff $T(x)$ is a sufficient statistic of X relative to $f_\theta(x)$.

Example 2.3 (Sufficient statistics).

- $X_1, \ldots, X_n$ i.i.d. Bernoulli(p). $T(\underline{X}) = \sum_{i=1}^n X_i$.
- $X_1, \ldots, X_n$ i.i.d. $N(\theta, \sigma^2)$. $T(\underline{X}) = (\sum X_i, \sum X_i^2)$, or equivalently $(\bar{x}, s^2)$.
- $X_1, \ldots, X_n$ i.i.d. Uniform(θ, θ+1). $T(\underline{X}) = (\min(X_1, \ldots, X_n), \max(X_1, \ldots, X_n))$.

2.4.2 Fano's inequality

Suppose X is the unknown and Y is our data. When $H(X|Y)$ is large, there is much entropy or variability remaining about the unknown X even after we collect the data Y and we can not hope to guess or estimate well X from Y. Fano's inequality quantifies this observation, and it has two important applications to be covered in the later lectures: proving the converse to Shannon's channel coding theorem and giving lower bounds in minimax density estimation.

Theorem 2.7 (Fano's Inequality). *Suppose $X \to P(Y|X) \to Y \to \hat{X}$, that is, $\hat{X}$ is an estimate of X based on Y. Let $P_e = P(X \neq \hat{X})$ be the probability of error. Then*

$$H(P_e) + P_e \log(|\mathcal{X}| - 1) \geq H(X|Y) \tag{2.25}$$

where $H(P_e) = -P_e \log P_e - (1 - P_e)\log(1 - P_e)$. Note that if $P_e = 0$, then $H(X|Y) = 0$.

Proof. Let $E = \mathbf{1}(\hat{X} \neq X)$. Then

$$H(E, X|Y) = H(X|Y) + H(E|X, Y) = H(X|Y)$$

(if you know X and Y, you know E, so $H(E|X, Y) = 0$).

On the other hand, one can also write

$$H(E, X|Y) = H(E|Y) + H(X|E, Y). \tag{2.26}$$

$$H(X|E, Y) = P(E = 1)H(X|E = 1, Y) + P(E = 0)H(X|E = 0, Y).$$

If $E = 0$ and we know Y, we also know X, so $H(X|E = 0, Y) = 0$. Hence

$$H(E, X|Y) = H(E|Y) + H(X|E, Y) = H(E|Y) + P(E = 1)H(X|E = 1, Y).$$

Putting the two identities together gives

$$H(X|Y) = H(E|Y) + P(E = 1)H(X|E = 1, Y).$$

If $E = 1$, there are $|\mathcal{X}| - 1$ possible values for $\hat{X}$, so $H(X|E, Y) \leq H(X) \leq \log(|\mathcal{X}| - 1)$. Or we could bound it by $H(X)$. Combining all the inequalities we get

$$H(X|Y) \leq H(E|Y) + P(E = 1)H(X|E = 1, y) \leq H(P_e) + P_e \log(|\mathcal{X}| - 1), \tag{2.27}$$

or

$$H(X|Y) \leq H(E|Y) + P(E = 1)H(X|E = 1, y) \leq H(P_e) + P_e H(X). \tag{2.28}$$

□

Note: Fano's inequality is sharp: let $X \in \{1, \ldots, m\}$, $Y \equiv 1 = \hat{X}$, $P_e = P(\hat{X} \neq X)$. Let the distribution of X be $p_1 = 1 - P_e$, $p_2 = \cdots = p_m = \frac{P_e}{m-1}$. Check that this achieves equality.

Corollary 2.2.

$$P_e \geq 1 - \frac{I(X, Y) + 1}{H(X)} \tag{2.29}$$

Proof. $1 + P_e H(X) \geq H(P_e) + P_e H(X) \geq H(X|Y) = H(X|Y) - H(X) + H(X) = H(X) - I(X, Y)$. Rearrange and get $H(X)(P_e - 1) \geq -I(X, Y) - 1$.

When X is uniform, $H(X) = \log(|\mathcal{X}|)$, and $P_e \geq 1 - \frac{I(X,Y)+1}{\log(|\mathcal{X}|)}$. □

In our document example where X and Y are consecutive words, $H(X) = 7.1$, $H(X|Y) = 2.7$, Fano's inequality gives:

$$1 + 7.1P_e \geq 2.7 \text{ or } P_e \geq 1.7/7.1 = 0.239,$$

which means that one cannot guess the next word from the previous word with an accuracy better than 23.9%, or the error rate is at least 23.9%, no matter now sophiscated the method is.

Note that this lower bound depends on only two summary quantities of the joint distribution which in this document example is a huge two way table:

$$H(X), H(X|Y).$$

For a symmetric binary channel with cross-error probability $\epsilon < 1/2$, let X be an input variable with distribution $1/2, 1/2$,

$$H(Y|X) = h(\epsilon) = H(X|Y),\ H(Y) = H(X) = 1.$$

Fano's is

$$H(P_e) + P_e \geq H(\epsilon).$$

The maximum likelihood (and the Bayesian maximum posterior) decoder is $\hat{X} = Y$, and it has an error ϵ. When ϵ is small, the LHS of the inequality is dominated by $H(P_e)$ which matches the RHS. In this sense, the Fano's inequality is "almost" tight.

Exercise Set 5

1. Prove Pinsker's inequality.
2. For a nice parametric family $\{g(\cdot, \theta\}$, prove that

$$K(g_\theta, g_{\theta+\Delta\theta}) \approx \frac{1}{2}(\Delta\theta)^2 J(\theta),$$

   where $J(\theta)$ is the Fisher information.
3. Suppose $X_1, ..., X_n$ are iid from uniform $[0, \theta]$. Prove that $max(X_1, ..., X_n)$ is a sufficient statistic for θ.
4. For a symmetric binary channel with cross-error probability $\epsilon < 1/2$, let X be an input variable with distribution $p, 1-p$, find the quantities involved in the Fano's inequality, interpret the inequality, and compare the inequality with the error rates of the maximum likelihood decoder and the maximum posterior decoder.

2.4.3 Non-parametric minimax density estimation

After a few decades of successful parametric modeling, the need to leave particular families of distributions was increased enough in the 60's that a systematic study began of estimation procedures which do not rely on a parametric form of distributions. Presumably, this is because more data were collected but the science or social science discipline iteself was not ready to suggest a parametric form either because of the lack of subject knowledge or because it was impossible due to the high noise to signal ratios in other fields rather than physical sciences.

Given iid samples $x_1, ..., x_n$ from a density f, here are two of the common nonparametric density estimators:

- histogram density estimator
- kernel density estimator:

$$\hat{f}(x) = \frac{1}{nh} \sum_{i=1}^{n} \phi \left(\frac{x - x_i}{h} \right)$$

The first estimator has as long a history as the tabulation of data and the second can at least be traced back to the 60's.

When there is no parametric form for the underlying density, Fisher's Maximum Likelihood theory cannot provide the fundamental limit on estimation efficiency. Other frameworks become necessary. It is known that if no constraints are put on the underlying distribution, it is futile to study the question of what is the best estimator since for any given rate α_n going to zero as n tends to infinity and any density estimator, there is always a density such that the estimation error under this density is slower than α_n.

One successful formulation of nonparametric density estimation has been the minimax approach with regularity conditions imposed on all the densities in a particular class to which the true density is assumed to belong. More precisely, let $\mathcal{M}$ be the class and we seek a procedure that gives the best performance for the worst density in the class:

$$\hat{f}^* = \arg\min_{\hat{f}} \max_{f\in\mathcal{M}} E_f d(f, \hat{f})$$

where d is a distance measure between densities.

For example $\mathcal{M}$ could be the following Lipschitz class:

$$\mathcal{M} = \{f : f \text{ is on } [0,1],\ 0 < c_0 \le f \le c_1, |f'| < c'\}\,.$$

For smooth density function classes, the minimax risk usually converges to zero at a rate $n^{-\alpha}$ for some $0 < \alpha < 1/2$, which is slower than the usual parametric rate of $n^{-1/2}$. To be more precise, if the logarithm of the density has p continuous derivatives the rate is $\alpha = 2p/(2p+1)$.

Historically, the minimax optimal rates were obtained in two separate stages. For the lower bound, an intricate subclass was constructed and explicit risk calculations were possible based on inequalities such as Assouad's, Le Cam's or Fano's (cf. Yu, 1996). One popular trick was to build an subclass indexed by a hypercube with expanding dimensions by having local pertubations of a uniform density function within the constraints of the class. Depending on the pertubation is up or down, we get the hypercube subclass. For the upper bound, one tries different known density estimators such as histogram or kernel estimators to match the rates in the lower bounds. There was no systematic way of finding minimax optimal estimators.

Yang and Barron (1999) unified the calculations in the lower and upper bounds by introducing the use of KL divergence and associated inequalities.

Suppose we are given a density function class $\mathcal{M}$ with some smoothness properties. Recall that we have seen the Hellinger distance and the KL-divergence between any two densities f and g (with respect to the Lebesque measure):

$$H(f,g) = (\int (\sqrt{f} - \sqrt{g})^2 dx)^{1/2}.$$

$$D(f\|g) = \int f \log \frac{f}{g} dx.$$

Assumption: $D(f,g) \asymp H^2(f,g)$ when f and g are "close", i.e., there exist constants A and A' s.t.

$$AH^2(f,g) \le D(f\|g) \le A'H^2(f,g) \text{ when } H^2(f,g) \le \epsilon.$$

D is very convenient to use in analysis because there are many information theory inequalities related to it, but it is not a distance. H, on the other hand, is a distance, so if the assumption holds, one can switch back and forth between D and H and take advantage of both.

The fundamental limit of estimation over $\mathcal{M}$ is determined by the metric entropy of the class which is the counterpart of dimension in the finite dimensional case. For a given distance d, there are two ways to define such an entropy depending on whether or not we are considering the covering entropy or the packing entropy. For the covering entropy, we use the KL-divergence for technical convenience; and for the packing entropy, we use the Hellinger distance to take advantage of its triangle inequality.

Definition 12 (Minimal covering number). of a set $\mathcal{M}$, $N_d(\epsilon_n)$, is the minimum number of balls of radius (in terms of $\sqrt{D}$) ϵ_n needed to cover $\mathcal{M}$. That is, given an ϵ_n-cover $T_D(\epsilon_n) = \{f_1, ..., f_K\}$ with cardinality $K = |T_D(\epsilon_n)|$ such that for any $f \in \mathcal{M}$, there is an $f_i \in T_D(\epsilon_n)$ with the property that $D(f, f_i) \leq \epsilon_n^2$, we have $N_D(\epsilon_n) \leq K$. $V_D(\epsilon_n) = \log(N_D(\epsilon_n))$ is called the *metric entropy.*

Definition 13 (Maximum packing number). of a set $\mathcal{M}$ is the maximum number of balls of radius (in terms of H) ϵ_n that can be packed inside $\mathcal{M}$so that their centers are in $\mathcal{M}$ and the balls do not overlap.

Lemma 2.3 (Mixture of Product Measures as an Estimator). *Suppose $G_D(\epsilon_n)$ is a minimal cover of $\mathcal{M}$ at radius ϵ_n, let p be the mixture of the product measures of the centers in the cover:*

$$p(x^n) = \frac{1}{N_D(\epsilon_n)} \sum_i^{N_D(\epsilon_n)} f_i(x^n).$$

Then for any $f \in \mathcal{M}$,

$$D(f(x^n)||p(x^n) \leq V_D(\epsilon_n) + n\epsilon_n^2.$$

Moreover, the mixture density $p(x^n)$ is the Bayes estimator of $f(x^n) \in G(\epsilon_n)$ with respect to the uniform prior on $G(\epsilon_n)$ and risk or loss function D. That is,

$$p(x^n) = argmin_{Qonx^n} \frac{1}{N_D} \sum_i^{N_D} D(f_i(x^n)||Q(x^n)).$$

The same conclusion still holds when the prior is not uniform and G is any arbitrary density set.

Proof. For a given $f \in \mathcal{M}$, there is a f_j in $G_D(\epsilon_n)$ such that $D(f||f_i) \leq \epsilon_n^2$, obviously

$$\frac{1}{N_D(\epsilon_n)} \sum_i^{N_D(\epsilon_n)} f_i(x^n) \leq \frac{1}{N_D(\epsilon_n)} f_j(x^n).$$

Hence

$$\begin{aligned} D(f(x^n)||p(x^n)) &= E_{f(x^n)}\{\log f(x^n)/[\frac{1}{N_D(\epsilon_n)}\sum_i^{N_D(\epsilon_n)} f_i(x^n)]\} \\ &\leq E_{f(x^n)}\log f(x^n)/\{\frac{1}{N_D(\epsilon_n)}f_j(x^n)\} \\ &= \log|G_D(\epsilon_n)| + D(f(x^n)||f_j(x^n) \\ &= V_D(\epsilon_n) + nD(f||f_j) \\ &= V_D(\epsilon_n) + n\epsilon_n^2. \end{aligned}$$

For the second conclusion, let us prove it in the general case.

For a set of densities $f_\theta(x)$ indexed by θ and a prior density $\pi(\theta)$ on the set, we want to show that

$$m(x) = \int f_\theta(x)\pi(\theta)d\theta = \text{ argmin}_Q \int D(f_\theta||Q)\pi(\theta)d\theta.$$

That is, for any density Q, it suffices to show that

$$I = \int D(f_\theta||Q)\pi(\theta)d\theta - \int D(f_\theta||m)\pi(\theta)d\theta \geq 0.$$

Notice that

$$\begin{aligned} I &= \int\int f_\theta(x)\log[m(x)/Q(x)]dx\pi(\theta)d\theta \\ &= \int \log[m(x)/Q(x)][\int f_\theta(x)\pi(\theta)d\theta]dx \\ &= \int m(x)\log[m(x)/Q(x)]dx \\ &= D(m||Q) \geq 0 \end{aligned}$$

.

□

Theorem 2.8 (Upper bound). *Using the notations in the previous lemma, construct a marginal density using the mixture density on the n-tuples:*

$$\hat{f}(x) = \frac{1}{n}\sum_{i=0}^{n-1} p(x_{i+1} = x|x^i) = \frac{1}{n}\sum_{i=0}^{n-1}\hat{f}_i,$$

where $\hat{f}_i = p(x_{i+1} = x|x^i)$*. Then for any* $f \in \mathcal{M}$*,*

$$E_{f(x^n)}D(f||\hat{f}) \leq \frac{1}{n}V(\epsilon_n) + \epsilon_n^2.$$

Proof. For any $f \in \mathcal{M}$, since $D(p||q)$ is convex in the pair (p, q), and $\hat{f}$ is a convex combination,

$$\begin{aligned}
E_{f(x^n)} D(f||\hat{f}) &\leq E_{f(x^n)} \frac{1}{n} \sum_{i=0}^{n-1} D(f||\hat{f}_i) \\
&= \frac{1}{n} \sum_{i=0}^{n-1} E_{f(x^n)} D(f||\hat{f}_i) \\
&= \frac{1}{n} \sum_{i=0}^{n-1} E_{f(x^n)} \log f(x_{i+1}/p(x_{i+1}|x^i) \\
&= \frac{1}{n} E_{f(x^n)} \sum_{i=0}^{n-1} \log f(x_{i+1}/p(x_{i+1}|x^i) \\
&= \frac{1}{n} E_{f(x^n)} \log(f(x^n)/p(x^n)) \\
&= \frac{1}{n} D(f(x^n)||p(x^n)) \\
&\leq V_D(\epsilon_n)/n + \epsilon_n^2, \quad \text{by Lemma 2.3 .}
\end{aligned}$$

□

Theorem 2.9 (Lower bound). *Let $\hat{f}$ be an estimate based on an i.i.d. sample $X_1, \ldots, X_n$ from $f \in \mathcal{M}$. Then*

$$\min_{\hat{f}} \max_{f \in \mathcal{M}} E_f H^2(f, \hat{f}) \asymp \min_{\hat{f}} \max_{f \in \mathcal{M}} E_f D(f\|\hat{f}) \geq \epsilon_n^2/8$$

The proof follows from the following two lemmas and the assumption that $D(f||g)$ and $H^2(f, g)$ are equivalent when f and g are close.

Lemma 2.4. *Given a maximal packing net $G(\epsilon_n)$, for any estimator $\hat{f}$,*

$$\max_{f \in \mathcal{M}} E_f H^2(f, \hat{f}) \geq (\epsilon_n/2)^2 \max_{f \in G(\epsilon_n)} P_f\{f \neq \tilde{f}\}$$

where $\tilde{f} = arg \min_{f \in G(\epsilon_n)} H^2(\hat{f}, f)$ is the projection of $\hat{f}$ on the maximal packing net.

Proof. First let us show that if $f \in G(\epsilon_n)$, then

$$\{f \neq \tilde{f}\} \subset \{H(f, \hat{f}) \geq \epsilon_n/2\}.$$

By contradiction: if $H(f, \hat{f}) < \epsilon_n/2$, but $f \neq \tilde{f}$, then there exists $f' \in G(\epsilon_n)$ such that $H(\hat{f}, f') < H(\hat{f}, f) < \epsilon_n/2$ (recall that $\tilde{f}$ is projection of $\hat{f}$ onto the net G, and so since f is not equal $\tilde{f}$, there must be another function in the net that is closer to $\hat{f}$). But then by triangular inequality

$$H(f, f') \leq H(\hat{f}, f') + H(\hat{f}, f) < \epsilon_n,$$

which contradicts the packing property of the net, since both f and f' belong to $G(\epsilon_n)$.

From the above and from Markov inequality we get

$$P_f\{f \neq \tilde{f}\} \leq P_f\{H(f,\hat{f}) \geq \epsilon_n/2\} \leq \frac{E_f H^2(f,\hat{f})}{(\epsilon_n/2)^2},$$

and therefore

$$E_f H^2(f,\hat{f}) \geq (\epsilon_n/2)^2 P_f\{f \neq \tilde{f}\}.$$

Since $G(\epsilon_n) \subset \mathcal{M}$, we get

$$\max_{f\in\mathcal{M}} E_f H^2(f,\hat{f}) \geq \max_{f\in G(\epsilon_n)} E_f H^2(f,\hat{f}) \geq (\epsilon_n/2)^2 P_f\{f \neq \tilde{f}\}.$$

□

Lemma 2.5.

$$\max_{f\in G(\epsilon_n)} P_f\{f \neq \tilde{f}\} \geq 1/2.$$

Proof. Let Θ be uniform on $G(\epsilon_n) = \{f_i : i = 1,\ldots,N(\epsilon_n)\}$. The entropy $H(\Theta) = \log(N(\epsilon_n)) = V(\epsilon_n)$. From the corollary to Fano's inequality, Note that $\Theta \to f_i \to X^n$, and

$$\begin{aligned} I(\Theta, X^n) &= \sum_{i=1}^{N(\epsilon_n)} \frac{1}{N(\epsilon_n)} \int f_i(x^n) \log \frac{f_i(x^n)}{f_{\epsilon_n}(x^n)} dx^n \\ &\leq \sum_{i=1}^{N(\epsilon_n)} \frac{1}{N(\epsilon_n)} \int f_i(x^n) \log \frac{f_i(x^n)}{Q(x^n)} dx^n \end{aligned}$$

where

$$f_{\epsilon_n}(x^n) = \sum_{i=1}^{N(\epsilon_n)} \frac{1}{N(\epsilon_n)} f_i(x^n)$$

is the Bayes mixture density which minimizes the Kullback-Leibler divergence $D(f\|Q)$ for any other density $Q(x^n)$, hence the last inequality.

Take another net $\tilde{G}(\tilde{\epsilon}_n)$ such that for any $f \in \mathcal{M}$ there exists $\bar{f} \in \tilde{G}(\tilde{\epsilon}_n)$ such that $D(f\|\bar{f}) \leq \tilde{\epsilon}_n^2$. Define

$$Q(x^n) = \frac{1}{N(\tilde{\epsilon}_n)} \sum_{\tilde{G}(\tilde{\epsilon}_n)} f_i(x^n)$$

If we replace the last sum by just one term from $\tilde{G}(\tilde{\epsilon}_n)$, we get

$$\log \frac{f_i(x^n)}{Q(x^n)} \leq \log \frac{f_i(x^n)}{\frac{1}{N(\tilde{\epsilon}_n)} \bar{f}(x^n)}$$

Take expectations:

$$E_f \log \frac{f_i(x^n)}{\frac{1}{N(\tilde{\epsilon}_n)} \bar{f}(x^n)} = \log N(\tilde{\epsilon}_n) + D(f_i(x^n) \| \bar{f}(x^n)) \leq V(\tilde{\epsilon}_n) + n\tilde{\epsilon}_n^2$$

Plugging into Fano's inequality and choosing $\tilde{\epsilon}_n$ such that $V(\tilde{\epsilon}_n) + n\tilde{\epsilon}_n^2 + 1 = V(\epsilon_n)/2$ we get

$$\max_{f \in G(\epsilon_n)} P_f\{f \neq \tilde{f}\} \geq 1 - \frac{V(\tilde{\epsilon}_n) + n\tilde{\epsilon}_n^2 + 1}{V(\epsilon_n)} = \frac{1}{2}.$$

□

If we choose $\epsilon_n = \alpha_n$ such that

$$V(\alpha_n) = n\alpha_n^2,$$

then we obtain the optimal minimax rate α_n^2 since the rates in the lower and upper bounds would match with such a choice.

Example 2.4 (Lipschitz Class). For the Lipschitz class defined earlier, it can be shown that

$$V(\epsilon) = 1/\epsilon.$$

It follows that the optimal rate is $n^{-2/3}$ after verifyting that H^2 and D are equivalent for all memebers of the class.

2.5 The Method of Types

When the source distribution is unknown, a *universal* coding scheme would first transmit the parameter in the model and then use the estimated probability model (with one of the above entropy codes) to transmit the data. If we assume our source is iid P on a finite alphabet $\mathcal{X} = \{1, ..., m\}$, for an observed sequence $X_1, ..., X_n$, the counts of them in the m categories form a multinomial distribution and we can send the estimated parameters in this model first, or equivalently, the counts $N_1, ..., N_m$ or frequencies $P_{\boldsymbol{x}} = (N_1, ..., N_m)/n$ in the m categories among the X's, which is the empirical distribution of $X_1, ..., X_n$.

Definition 14 (Type $P_{\boldsymbol{x}}$). The sequence $X_1, ..., X_n$ is said to have a type $P_{\boldsymbol{x}}$ if its empirical distribution is $P_{\boldsymbol{x}}$.

Definition 15 (Type Class Q).

$$T(Q) = \{\boldsymbol{x} = (x_1, ..., x_n) \in \mathcal{X}^n : P_{\boldsymbol{x}} = Q\}.$$

It is easy to see that there are $n+1$ type classes based on sequences of size n in the binary case $m = 2$. In the general case, the number of type classes is less than $(n+1)^m$.

Stirling's approximation gives the number of sequences in a type class $T(Q)$ as:

Theorem 2.10 (Size of a type class $T(P)$).

$$\frac{1}{(n+1)^m} 2^{nH(P)} \leq |T(P)| \leq 2^{nH(P)}.$$

Proof. Here we provide an alternative proof (Cover and Thomas, p. 282-284):

The upper bound is trivial. For the lower bound, use the fact that the type class $T(P)$ has the highest probability under distribution P^n. It follows that

$$\begin{aligned}
1 &= \sum_{Q \in \mathcal{P}_n} P^n(T(Q)) \\
&\leq \sum_{Q \in \mathcal{P}_n} \max_Q P^n(T(Q)) \\
&= \sum_{Q \in \mathcal{P}_n} P^n(T(P)) \\
&\leq (n+1)^m P(T(P)) \\
&= (n+1)^m T(P) 2^{-nH(P)}.
\end{aligned}$$

□

Apparently, in the binary case, we know the number of type classes is $n+1$ rather than the bound $(n+1)^2$ used in the general case proof, so we can sharpen the lower bound to

$$\frac{1}{n+1} 2^{nH(k/n)}.$$

Theorem 2.11 (Probability of a type class $T(Q)$). *For any type P and any distribution Q, the probability of the type class $T(P)$ under Q^n is $2^{-nD(P||Q)}$ to first order in the exponent. More precisely,*

$$\frac{1}{(n+1)^m} 2^{-nD(P||Q)} \leq |Q^n(T(P))| \leq 2^{-nD(P||Q)}.$$

Proof. It follows from the previous theorem and the fact that, for any x^n with type P,

$$Q^n(x^n) = 2^{-n(H(P)+D(P||Q))}.$$

The fact holds by writing out the product probability $Q^n(x^n)$ and verifying that it is the right hand side of the above equation.

□

Exercise Set 6

1. Prove LLN in terms of D using the above theorems. That is, $D(P_{x^n}||P) \to 0$ with probability 1.

2. CT, pp. 333 #1

3. CT, pp. 334, #5.

Example 2.1 (Universal Source Coding). For a given sequence x^n, it takes less than $m\log(n+1)$ bits to send over the type class information and if we use a uniform code over this type class (all the sequences have the same probability in a type class under an iid assumption on the source distribution), then it takes $\log|T(P_{x^n})| \leq nH(P_{x^n})$ bits to transmit the membership of the observed x^n in the type class. Hence the average code length of resulted two-part code $L(x^n)/n$ tends to $H(P)$ almost surely as $n \to \infty$ because $H(P_{x^n}) \to H(P)$ almost surely.

Read p. 279-291 of CT.

2.6 Large Deviations

Definition 16 (Exponent). A real number a is said to be the exponenet for a sequence a_n, $n \geq 1$, where $a_n \geq 0$ and $a_n \to 0$ if

$$\lim_{n\to\infty}\left(-\frac{1}{n}\log a_n\right) = -a\,.$$

Sanov's Theorem

The Central Limit Theorem provides good approximations to events of "moderate deviations" from the typical, but fails to provide accurate estimates of "rare events" or large deviations from the typical. These "rare event" calculations arise in applications from hypothesis testing (which we will cover) to finance (estimating probability of large portfolio losses). The classical LD results can be derived based on the method of types bounds we have just seen and we end with a discussion of the general case. This rare probability often turns out to be exponential with a negative size factor (n in the iid case) and an exponent which is describing how close the events to the typical in terms of the KL-divergence.

Theorem 2.12 (Sanov's theorem). *Let $X_1, ..., X_n$ be iid Q. Let E be a set of probability distributions on a finite alphabet $\mathcal{X}$. Then*

$$Q^n(E) = Q^n(E \cap \mathcal{P}_n) \leq (n+1)^m 2^{-nD(P^*||Q)},$$

where

$$P^* = argmin_{P\in E} D(P||Q),$$

is the distribution in E that is closest to P in KL divergence. If, in addition, the set E is the closure of its interior, then

$$\frac{1}{n}P^n(E) \to -D(P^*||Q).$$

Proof. The upper bound is straightforward: the moral is that in the log scale and to the first order, the maximum probability in a summation of probabilities gives the right answer. Here we need E to be nice so that we can find a distribution $P \in E$ which is close to P^*. Under the assumption on E, since $\cup_n \mathcal{P}_n$ is dense in the set of all distributions on $\mathcal{X}^n$, we can find a sequence of distributions P_n so that for $n \leq n_0$, $P_n \in E \cap \mathcal{P}_n$, and $D(P_n||Q) \to D(P^*||Q)$. For $n \leq n_0$,

$$\begin{aligned} Q^n(E) &= \sum_{P \in E \cap \mathcal{P}_n} Q^n(T(P)) \\ &\geq Q^n(T(P_n)) \\ &\geq \frac{1}{(n+1)^m} 2^{-nD(P_n||Q)}, \end{aligned}$$

which implies the lower bound. □

In this characterization of a large deviations result, the set of probabilities E are used to describe the event of interest. We now give an example based on sample averages.

Example 2.1 (Sample averages). We now consider the probability of events based on the sample mean, the simplest case being that the sample mean is larger than some threshold τ. Let $X_1, \ldots, X_n$ be iid according to Q, with each $X_i \in \mathcal{X} = \{0, \ldots, m\}$. To compute the probability

$$Pr(\frac{1}{n} \sum_i X_i > \tau)$$

We consider the set E

$$E = \{P : \sum_{j=1}^{n} jP(j) > \tau\} \tag{2.30}$$

We now want to minimize the KL divergence between elements in E and the data generating distribution Q. Introducing Lagrange multipliers λ and ν we want to minimize the following expression

$$\sum_j P(j) \log \frac{P(j)}{Q(j)} - \lambda \sum_j P(j) j + \nu \sum_j P(j) \,.$$

The first multiplier λ is associated with the constraint on the sample mean (2.30), while the second ν ensures that the function P is a probability. The solution is then of the form

$$P^*(j) = \frac{2^{j\lambda} Q(j)}{\sum_{j'} 2^{j'\lambda} Q(j')}$$

where we select λ to satisfy the constraint that $\sum_j jP(j) = \tau$. The distribution P^* is called a twisted distribution and we'll see other examples of this in connection with Stein's theorem. In Figure 2.1 we consider tossing a fair dice.

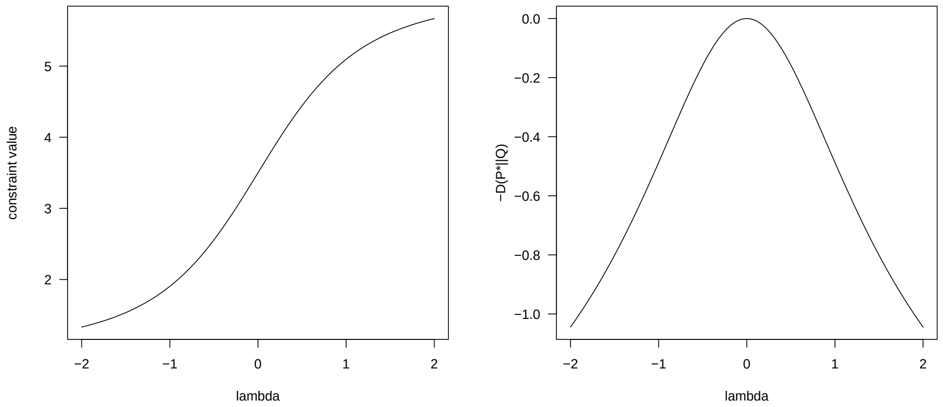


Figure 2.1:

Therefore, $j = 6$ in the above notation. In the figure, we plot first the relationship between λ and τ, and then the minimized KL divergence as a function of λ. Note that $\lambda = 0$ corresponds to $\tau = 3.5$, the mean of the true data-generating distribution. This yields an exponent of zero as we would expect.

If we rewrite this expression in terms of the natural logarithm, the we find that $D_e(P^*\|Q)$ reduces to

$$\begin{aligned} D(P^*\|Q) &= \sum_j \frac{e^{j\lambda}Q(j)}{\sum_{j'} e^{j'\lambda}} \log \frac{e^{j\lambda}Q(j)}{\sum_{j'} e^{j'\lambda}} \frac{1}{Q(j)} \\ &= \lambda\tau - \ln \sum_j e^{j\lambda}Q(j) \\ &= \lambda\tau - \ln M(\lambda) \end{aligned}$$

We recognize the last term in this expression as the natural logarithm of the moment generating function $M(\lambda)$ for Q. We will see this form for the exponent again in Cramer's theorem.

The general case

From Theorem 2.12, in the iid case, to the first order

$$-\log Q(T(P)) = nD(P||Q),$$

where Q is the true generating probability distribution and P is the type under consideration. Because

$$\begin{aligned}\max[P(A), P(B)] &\leq P(A \cup B) \\ &\leq [P(A) + P(B)] \\ &\leq 2\max[P(A), P(B)],\end{aligned}$$

on a log scale, we have for all practical purposes (since for rare events, the log probabilities are negatively large and factor 2 becomes negligible on the log scale),

$$\log P(A \cup B) \approx \max[\log P(A), \log P(B)];$$

or

$$-\log P(A \cup B) \approx \min[-\log P(A), -\log P(B)];$$

If the rare event E of interest is a union of type classes, then

$$-\log Q_n(E) \approx n \mathrm{min}_{P \in E} D(P||Q),$$

where Q_n is the product measure based on Q.

This can be interpreted as follows. If we want to calculate the probability of a "rare" event E in a "large" experiment under a model Q, then all we have to do is to find an appropriate model P under which the event is "natural" or normal and then

$$-\log Q_n(E) \approx \text{ “size factor” } D(P||Q).$$

In the iid case, the size factor is n as we have seen. Sanov's theorem

So the art of Large Deviations lies in the choice of the alternate model P. The choice is by no means unique and a wrong choice can lead to a wrong answer. In our iid case, the good choice is obviously the distribution which matches the type in the observed sequence. In other situations, things are not always clear. This is the same idea behind importance sampling for estimating the probability of a rare event through Monte Carlo simulations.

Example 2.2 (A Markov Alternate Model). Suppose we toss a fair coin n times independently of each other, and let $X_1, \ldots, X_n$ denote the outcome. We would like to compute the probability $q(n, k)$ that the number k of occurrences of two consecutive heads is roughly ns, for some given $0 < s < 1$. Here, we would count the run HHH as two occurences of the event. We need to build some memory into the alternate model P_n so that consecutive occurrences of H is natural under P_n. Toward this end, we consider the class of Markov models with transition matrix

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

The stationary distribution π of this chain is given by $\pi(H) = \alpha/(\alpha+\beta)$, and $\pi(T) = \beta/(\alpha+\beta)$. (Recall that the stationary distrubution of $\mathbb{P}$ satisfies

$\mathbb{P}\pi = \pi$, so that a chain started in this distribution remains there in the sense that $P(X_n = H) = \pi(H)$ for all n.)

Therefore, the expected ratio of the number of consecutive heads to the total run n is then

$$\alpha/(\alpha + \beta) \times (1 - \beta).$$

So we should choose a model with

$$s = \frac{\alpha(1-\beta)}{\alpha+\beta},$$

to match the expected number of HH as in the rare event. If we calculate the KL divergence of this model with the true model with $\alpha = \beta = 1/2$, we get

$$\begin{aligned} F(\alpha, \beta) &= 1 + \frac{\alpha(1-\beta)}{\alpha+\beta} \log(1-\beta) + \frac{\alpha\beta}{\alpha+\beta} \log \alpha + \\ &\quad \frac{\alpha\beta}{\alpha+\beta} \log \beta + \frac{\beta(1-\alpha)}{\alpha+\beta} \log(1-\alpha). \end{aligned}$$

Following the recipe given above, we find the member of the class of Markov chains that is closest to the true iid coin tossing model in thesense of KL divergence, subject to the constraint that the proportion of consecutive heads is s; that is, $\alpha(1-\beta)/(\alpha+\beta)$ is equal to s. Set

$$g(s) = \min_{(\alpha,\beta):s=\frac{\alpha(1-\beta)}{\alpha+\beta}} F(\alpha, \beta)$$

If we apply the constraint, we can set $\beta =$ and then search numerically for α.

In Figure 2.2, we plot F as a function of s. Notice that the function attains a maximum of zero at $1/4$, corresponding to the fact that under the data generating distribution, seeing about $s = 1/4$ is natural. For each fixed value of s, we read the "exponent" $g(s)$ from the graph so that

$$\frac{1}{n} \log q(n, k) \approx g(s)$$

Models with a lot of dependence (generating many more HH pairs or many more TT pairs) have a faster decline in the probability.

The fact that this recipe works in more complex settings than those covered by Sanov's theorem can be explained somewhat intuitively.

$$Q(E) = \int_E dQ = \int_E dQ/dP \; dP.$$

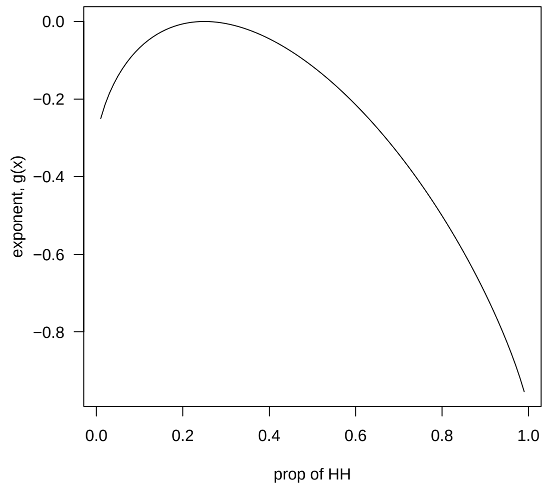


Figure 2.2:

If we pick P such that $P(E) \approx 1$, then

$$\begin{aligned} & \log Q(E) - \log P(E) \\ = \; & \log[\frac{1}{P(E)} \int_E dQ/dP dP] \\ = \; & \log\{\frac{1}{P(E)} \int_E \exp[-\log dP/dQ] dP\} \\ \geq \; & -\frac{1}{P(E)} \int_E \log[dP/dQ] dP \\ \approx \; & -D(P||Q) \end{aligned}$$

Every P with $P(E) \approx 1$ gives a lower bound and the particular $P(\cdot) = Q(\cdot \cap E)/Q(E)$ gives the exact answer! But we are back to square 1. The trick or art is to choose a manageable class of alternate models so that the best lower bound over the class matches the upper bound or gives the correct answer. This is problem-dependent as we have demonstrated through the Markov example.

Historically, the first LD result was not proved in terms of the KL divergence. Rather, moment generating functions were employed by Cramér (1938) to show

Theorem 2.13 (Cramér's theorem). *Let $X_1, .., X_n$ be iid random variables. Assume that the moment generating function*

$$M(\theta) = E[\exp(\theta X])]$$

is finite for all $\theta \in R$. Let $\phi(a)$ be the Legendre transform of $\psi(\theta) = \log M(\theta)$:

$$\phi(a) = \sup_\theta [a\theta - \psi(\theta)] = \sup_\theta [a\theta - \log M(\theta)].$$

$\phi(a) \geq 0$ and $\phi(a) = 0$ if and only if $a = E[X]$. Then

$$P\{\frac{X_1 + ... + X - n}{n} \sim a] \approx \exp[-n\phi(a)]$$

for $a \neg E(X)$.

As one might have guessed, the Legendre transform representation of the exponent can be equivalently expressed in terms of the KL divergence.

2.7 Stein's Lemma in Hypothesis Testing

Hypothesis testing is one of the two major inference frameworks (the other being estimation) in classifical statistical inference. Its most celebrated result is the Neyman-Pearson Lemma obtained in the ??? which establishes the central role that the likelihood ratio was going to and is playing in hypothesis testing. The framework mimics the decision process to force a yes/no answer to the hypothesis being tested. Testing procedures are widely used in industry, especially mandated by FDA regulations in pharmaceutical industry to report clinical trial results. It is often used in health studies and the reports conform to the jargons such as statistical significant or highly statistically significant, which are often taken by the general public as "significant" in the common sense, but "scientifically proven" because of the appearence of the word "statistically". This is very unfortunately since all the evidence in terms of the probability of obtaining something more extreme than what has been observed or the p-value is calculated assuming the dominate hypothsis is correct (which is often not the case) and the testing is done allowing itself to committ two types of errors, which are in the ideal case small but still non-zero.

Here we concentrate on the simplest case of telling two hypothesis or distributions apart.

Definition 17. Statistical Simple Hypothesis Testing

Let $X_1, ..., X_n$ be iid with distribution Q and we test two simple hypotheses:

- $H_1: \; Q = P_1$ (null hypothesis) vs
- $H_2: \; Q = P_2$ (alternative hypothesis).

There are two typos of errors: false positive or type I error (H_1 is wrong, but accepted) and false negative or type II error (H_2 is true, but rejected).

Theorem 2.14 (Neyman-Pearson lemma). *The optimal test given a type I error not larger than α is in the form of the likelihood ratio test, that is, accept H_1 when*

$$A_n(T) = \{\frac{P_1(x^n)}{P_2(x^n)} > T\}$$

where the cut-off value T is chosen such that

$$P_1(A_n^c(T)) \leq \alpha.$$

Given a data string, the observed p-value of the test is the probability under the null hypothesis that we would have observed some test statistics value which is more extreme than the observed test statistics (e.g. likelihood ratio). p-values are comparable across different tests, while it is not always the case with test statistics.

The optimality is defined in the sense that no other tests at this level with a smaller type II error or a smaller probability

$$P_2(A_n(T));$$

equivalently, there are not other tests with a larger power or a higher probability

$$P_2(A_n^c(T)).$$

Example 2.1 (Gaussian location family).

The likelihood ratio test can be re-written in terms of KL divergences. That is,

$$\frac{P_1(x^n)}{P_2(x^n)} > T$$

is equivalent to

$$D(P_{x^n}||P_2) - D(P_{x^n}||P_1) > \frac{1}{n}\log T.$$

So the test is about how to choose type classes to form the acceptance region (or the rejection region). The boundary of the region is where the type classes are such that the differences between the KL divergences are a constant. LD can now be employed to choose the cut-off value T in a heuristic manner.

Next, we calculate the best error exponent when one of the two types of error goes to zero arbitrarily slowly.

Theorem 2.15 (Stein's lemma). *Assume $D(P_1||P_2) < \infty$. Let the two types of error are*

$$\alpha_n = P_1(A_n^c), \;\; \beta_n = P_2(A_n).$$

For $0 < \epsilon < 1/2$, define

$$\beta_n^\epsilon = min_{A_n \subset \mathcal{X}^n, \alpha_n < \epsilon} \beta_n.$$

Then

$$\lim_{\epsilon \to 0} \lim_{n \to \infty} \frac{1}{n} \log \beta_n^\epsilon = -D(P_1||P_2).$$

One would like to make both types of error go to zero, at least as the sample size n increases. The rates in Stein's Lemma tell us that, for this to happen, it is necessary to make the two hypotheses apart at least by $1/n$, or make

$D(P_1||P_2)n \to \infty$. In other words, we can not distinguish two distributions which are less than $1/\sqrt{n}$ apart in $\sqrt{D}$ which acts in the scale of a "distance". Here we do not assume any parametric models where it is well-known that we can not estimate an Euclidean parameter at a rate faster than $1/\sqrt{n}$ in Euclidean distance – which is the consequence of requiring the more essential distance between two distributions to be larger than $1/\sqrt{n}$.

2.8 Mutual Information

For a pair of random variables or a vector of them, an important question is on how much information they contain about each other; or how depedent they are. Mutual information to be defined below answers this question naturally and is the key ingredient in channel coding.

Definition 18 (Mutual Information). Let X, Y have a joint distribution $P(x, y)$, and have marginal distributions $P(x)$ and $P(y)$. The mutual information $I(X;Y)$ is the relative entropy between the joint distribution of X, Y and the product of their marginals; or rather, how far the joint distribution $P(x, y)$ is from independence. It therefore measures the dependence between the two variables – the larger the mutual information, the more dependence they are.

$$I(X;Y) = \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} P(x,y)\log\frac{P(x,y)}{P(x)P(y)}, \tag{2.31}$$

$I(X, Y) = 0$ if and only if X and Y are independent. The definition of $I(X, Y)$ can easily be extended to the continuous case by replacing the summation by an integration.

Using the definitions of mutual information and conditional entropy, we find the following expression.

$$\begin{aligned}
I(X;Y) &= \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} P(x,y)\log\frac{P(x,y)}{P(x)}P(y) && (2.32)\\
&= \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} P(x,y)\log\frac{P(y|x)}{P(y)} && (2.33)\\
&= \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} P(x,y)\log P(y|x) - \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} P(x,y)\log P(y) && (2.34)\\
&= \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} P(x,y)\log P(y|x) - \sum_{y\in\mathcal{Y}} P(y)\log P(y) && (2.35)\\
&= H(Y) - H(Y|X) && (2.36)
\end{aligned}$$

Also, recall that $H(Y) - H(Y|X) = H(X) - H(X|Y)$. Therefore, the mutual information records the reduction in uncertainty of X by knowing Y and, equivalently, the reduction in uncertainty in Y after knowing X. This is also the amount of information that X or Y contains about the other.

Example 2.1 (Bivariate Gaussian). Assume (X, Y) are normal with mean zero and variance 1 and correlaction coefficient ρ. It is straghtforward to calculate that

$$I(X, Y) = \log_2(1 - \rho^2).$$

This is a monotonic transformation of the correlation coefficient which is the only measure of dependence in the Gaussian case. The closer $|\rho|$ to 1, the larger the mutual information. The mutual information does not differetiate a positive correlation with a negative one, however.

Example 2.2 (Mutual information and document similarity). Here we consider the use of mutual information in a text mining capacity.

It is easy to verify the chain rule for mutual information based on the chain rule for entropy and the entropy difference expression of mutual information:

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

.

$D(p, q)$ is convex in pair (p, q)
$I(x, y)$ is concave in $p(x)$ for fixed $p(y|x)$
$I(x, y)$ is convex in $p(y|x)$ for fixed $p(x)$

2.8.1 Sufficiency

In the age of massive data collection, it is crucial to understand that manipulation of data doesn't increase the information. This is stated most concisely in the data processing inequality.

Theorem 2.16 (Data Processing Inequality). *If $X \to Y \to Z$ are Markov, then*

$$I(X, Y) \geq I(X, Z). \tag{2.37}$$

Proof. Using the chain rule on mutual information,

$$I(X; Z, Y) = I(X, Z) + I(X, Y|Z) \geq 0.$$

Similarly,

$$I(X; Y, Z) = I(X, Y) + I(X, Z|Y) = I(X, Y)$$

since X and Z are independent given Y. Therefore

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

□

Corollary 2.3. *For any function g, $I(X, Y) \geq I(X, g(Y))$.*

Let $f_\theta(x)$ be a parametric family. Our goal is to make inferences about θ from sample $x_1, \ldots, x_n \sim f_\theta(x_1, ..., x_n)$. A statistic $T(x_1, \ldots, x_n)$ is sufficient if given $T(X_1, ..., X_n)$, θ and X are independent or the distribution of $X_1, ..., X_n$ given $T(X_1, ..., X_n)$ is not dependent on θ.

One can find a sufficient statistic in a straightforward manner by capitalizing the factorization theorem:

Theorem 2.17 (Factorization Theorem). *T is a sufficient statistic for f_θ if and only*

$$f_\theta(x_1, \ldots, x_n) = g_\theta(T(x_1, \ldots, x_n))h(x_1, \ldots, x_n|T(x_1, \ldots, x_n)) \tag{2.38}$$

where g and h are two nonnegative functions and $h(x_1, \ldots, x_n|T(x_1, \ldots, x_n))$ does not depend on θ.

For example, if $x_1, \ldots, x_n$ are i.i.d. Gaussian with mean θ and variance 1, $T(\underline{x}) = \bar{x}$ is sufficient.

Theorem 2.18 (Mutual Information and Sufficiency). *T is a sufficient statistic for $f_\theta(X)$ iff*

$$I(\Theta, X) = I(\Theta, T(X).$$

Proof. Since T is a mapping from X, we have

$$\underline{X} \rightarrow T(\underline{X}),$$

which implies, by the data processing inequality, that

$$I(\Theta, T(\underline{X})) \leq I(\Theta, \underline{X}). \tag{2.39}$$

On the other hand, T is a sufficient statistic,

$$\Theta \rightarrow T(\underline{x}) \rightarrow \underline{x},$$

so

$$I(\Theta, T(\underline{X})) \geq I(\Theta, \underline{X}). \tag{2.40}$$

Combining two inequalities gives $I(\Theta, T(\underline{X})) = I(\Theta, \underline{X})$, so the mutual information of the parameter and the data is the same as mutual information of the parameter and the sufficient statistic. The converse??? □

Example 2.3 (Sufficient statistics).

- $X_1, \ldots, X_n$ i.i.d. Bernoulli(p). $T(\underline{X}) = \sum_{i=1}^n X_i$.
- $X_1, \ldots, X_n$ i.i.d. $N(\theta, \sigma^2)$. $T(\underline{X}) = (\sum X_i, \sum X_i^2)$, or equivalently $(\bar{x}, s^2)$.
- $X_1, \ldots, X_n$ i.i.d. Uniform(θ, θ+1). $T(\underline{X}) = (\min(X_1, \ldots, X_n), \max(X_1, \ldots, X_n))$.

2.8.2 Fano's inequality

Suppose X is the unknown and Y is our data. When $H(X|Y)$ is large, there is much entropy or variability remaining about the unknown X even after we collect the data Y and we can not hope to guess or estimate well X from Y. Fano's inequality quantifies this observation, and it has two important applications to be covered in the later lectures: proving the converse to Shannon's channel coding theorem and giving lower bounds in minimax density estimation.

Theorem 2.19 (Fano's Inequality). *Suppose $X \to P(Y|X) \to Y \to \hat{X}$, that is, $\hat{X}$ is an estimate of X based on Y. Let $P_e = P(X \neq \hat{X})$ be the probability of error. Then*

$$H(P_e) + P_e \log(|\mathcal{X}| - 1) \geq H(X|Y) \tag{2.41}$$

where $H(P_e) = -P_e \log P_e - (1 - P_e)\log(1 - P_e)$. Note that if $P_e = 0$, then $H(X|Y) = 0$.

Proof. Let $E = \mathbf{1}(\hat{X} \neq X)$. Then

$$H(E, X|Y) = H(X|Y) + H(E|X, Y) = H(X|Y)$$

(if you know X and Y, you know E, so $H(E|X, Y) = 0$). On the other hand, one can also write

$$H(E, X|Y) = H(E|Y) + H(X|E, Y). \tag{2.42}$$

Now, $H(E|Y) \leq H(E) = H(P_e)$, and

$$H(X|E, Y) = P(E = 1)H(X|E = 1, Y) + P(E = 0)H(X|E = 0, Y).$$

If $E = 0$ and we know Y, we also know X, so $H(X|E = 0, Y) = 0$. If $E = 1$, there are $|\mathcal{X}| - 1$ possible values for $\hat{X}$, so $H(X|E, Y) \leq H(X) \leq \log(|\mathcal{X}| - 1)$. Combining all the inequalities we get

$$H(X|Y) \leq H(E|Y) + P(E = 1)H(X|E = 1, y) \leq H(P_e) + P_e \log(|\mathcal{X}| - 1). \tag{2.43}$$

□

Note: Fano's inequality is sharp: let $X \in \{1, \dots, m\}$, $Y \equiv 1 = \hat{X}$, $P_e = P(\hat{X} \neq X)$. Let the distribution of X be $p_1 = 1 - P_e$, $p_2 = \cdots = p_m = \frac{P_e}{m-1}$. Check that this achieves equality.

Corollary 2.4.

$$P_e \geq 1 - \frac{I(X, Y) + 1}{H(X)} \tag{2.44}$$

Proof. $1 + P_e H(X) \geq H(P_e) + P_e H(X) \geq H(X|Y) = H(X|Y) - H(X) + H(X) = H(X) - I(X, Y)$. Rearrange and get $H(X)(P_e - 1) \geq -I(X, Y) - 1$.

When X is uniform, $H(X) = \log(|\mathcal{X}|)$, and $P_e \geq 1 - \frac{I(X,Y)+1}{\log(|\mathcal{X}|)}$. □

2.8.3 Channel Capacity

Most communication channels (between people, callers and receivers over wired telephone lines or wireless phones) are noisy. The question is how much information can be transmitted through a noisy channel. Before Shannon's 1948 work, the common belief was that errors had to occur with a positive chance for any noisy channel. Shannon (1948) proved it possible to transmit without errors in the limit when the message size gets large. However, he had to (1) allow an arbitrarily small but non zero probability of error; (2) use the channel many tims in succession (blocks) to the law of large numbers kicks in; (3) employ a random code book by looking at the average probility of error over randomly generated block books to prove the existence of at least one good code book.

Definition 19 (Discrete Channel). A discrete channel is characterized by, for any given input $x \in$ a finite alphabet $\mathcal{X}$, a conditional distribution $p(\cdot|x)$ on a finite alphabet $\mathcal{Y}$. The channel is said to be *memoryless* if

$$p(x_1, ..., x_n; y_1, ..., y_n) = \prod_{i=1}^{n} p(y_i|x_i)p(x_i).$$

A good channel's output or data y^n should contain much information about the input or unknown x^n. If we use mutual information to capture this observation, we should define the capacity of a channel to be the maximum information possible between a pair of input and output. That is,

Definition 20 (Channel Capacity). The information channel capacity of a discrete memoryless channel is

$$C = \max_{p(x)} I(X;Y),$$

where the maximum is taken over all possible input distribution p.

There is a duality between source coding and channel coding. In the former, we remove redundancy in the data to reduce the storage needed; in the latter, we add redundancy to the data to combat the channel noise. Channel decoding is basically an estimation problem, but the difference with the statistical estimatino problem lies in the fact that the unknowns or inputs to the channel can be manipulated to our advantage in channel coding but not in statistical estimation in general.

Example 2.4 (Binary symmetric channel with cross-over probability ϵ).

$$C = 1 - H(\epsilon).$$

Maximum capacity is 1 when $\epsilon = 0$ or the channel is noiseless.

Example 2.5 (Erasure channel erasure probability ϵ). Here the output alphabet $\mathcal{Y} = \{0, e, 1\}$ where e for erasure.

$$C = 1 - \epsilon.$$

Maximum capacity is 1 when $\epsilon = 0$ or the channel has no erasure.

Theorem 2.20 (Properties of C). *1. $C \geq 0$.*

2. $C \leq \log |\mathcal{X}|$.

3. $C \leq \log |\mathcal{Y}|$.

4. C is continuous in $p(x)$.

5. C is concave in $p(x)$.

The most important continuous channel is the Guassian additive noise channel and it is an accurate description of physical channels such as that in deep-space communication.

Example 2.6 (Gaussian Channel with a Power Constraint P). Given a continuous random input variable X, the output

$$Y = X + Z,$$

where Z is the channel noise which is independent of X and has a Gaussian distribution with mean 0 and variance N. Then

$$C = \max_{EX^2 \leq P} I(X, Y) = \frac{1}{2} \log(1 + P/N).$$

Hence the lower the channel noise and the higher the power, the larger the channel capacity.

Definition 21 ((M, n) Code). An (M, n) code for a channel consists of the following

1. An index set $\{1, ..., N\}$;
2. An encoding function X^n: $\{1, 2, ..., M\} \to \mathcal{X}^n$, yielding codewords $X^n(1), ..., X^n(M)$ which form the codebook.
3. A decoding function
$$g : \mathcal{Y}^n \to \{1, ..., M\}.$$

For a given (M, n) code, we use $\lambda^{(n)}$ and $P_e^{(n)}$ to denote the maximum probability of error and the average probability of error respectively. That is,

$$\lambda^{(n)} = \max_{i \in \{1, \ldots, M\}} \lambda_i$$

where

$$\lambda_i = P(g(Y^n) \neg i | X^n = X^n(i)).$$

$$P_e^{(n)} = \frac{1}{M} \sum_i \lambda_i.$$

We often denote an (M, n) code as a $(2^{nR}, n)$ code with $R = \log M/n$ (bits per transmission) being the rate of the code.

Theorem 2.21 (Shannon's channel coding theorem). *For a discrete memoryless channel, all rates below capacity C are achievable. Specifically, for every rate $R < C$, there exists a sequence of $(2^{nR}, n)$ codes with maximum probability of error $\lambda^{(n)}$ going to zero as $n \to \infty$. Conversely, any sequence of $(2^{nR}, n)$ codes with $\lambda^{(n)} \to 0$ must have $R \leq C$.*

Proof. Heuristics:

We work with typical sequences or typical joint sequences. In the world of typical sequences, things are roughly uniformly distributed. So counting does the job.

For any input distribution $p(x)$, there are only about $2^{nH(X)}$ typical sequences. For each of these typical sequences as an input, there are approximately $2^{nH(Y|X)}$ possible Y-sequences which are almost equally likely. We want to ensure that no two X-sequences produce the same Y output sequence.

The total number of possible (typical) Y-sequences is $\approx 2^{nH(Y)}$. This set has to be divided into sets of size $2^{nH(Y|X)}$ corresponding to the different input X sequences. Hence the total number of disjoint sets is less than or equal to

$$2^{n(H(Y)-nH(Y|X)} = 2^{nI(X,Y)}.$$

If the rate $R < C = \max I(X,Y)$, we could find an input distribution p^* such that $R < I(X^*, Y)$ and we could map the index set of the code into these disjoint sets through the X-typical sequences under p^*. Because R is strictly less than $I(X^*, Y)$ this translates into much "room" in the space of input sequences to cushion the rigorousness of the above argument which can be formalized via LLN for X and Y and (X, Y). The mapping of the code index into typical sequences under p^* is formally done via the famous random coding argument. Decoding can be done using maximum likelihood and through the joint typical sequences.

Converse:

Intuitively, it is easy to see that

$$I(X^n, Y^n) \leq nC,$$

which can be proved via the chain rules of entropy and conditional entropy. It says that the channel capacity can not be increased if we use the memoryless channel repeatedly.

Given any $(2^{nR}, n)$ code with its maximum prob. of error going to zero, its average prob. of error goes to zero too.

Let W be a uniform random variable on the index set of the code $\{1, ..., 2^{nR}\}$ and $\hat{W}$ be the decoded index. Then

$$P_e = P(\hat{W} \neq W) \to 0.$$

By Fano's inequality,

$$P_e \leq (H(W|Y^n) - 1)/(nR),$$

since $\log |2^{nR}| = nR$.

It follows that

$$nR = H(W) = H(W|Y^n)+I(W,Y^n) \le H(W|Y^n)+I(X^n(W),Y^n) \le nRP_e+1+nC,$$

which implies that

$$P_e > 1 - C/R - 1/(nR) > 0,$$

if $R > C$. This contradicts with $P_e \to 0$. Hence $R \le C$. □