Lecture Notes on Information Theory and Coding

Baoming Bai
State Key Lab. of ISN, Xidian University

April 21, 2015

Goals: The goal of this class is to establish an understanding of the instrinsic properties of transmission of information and the relation between coding and the foundamental limits of information transmission in the context of communication.

Course Outline:

- Entropy and Mutual information (Measure of information)
- Sourse coding
- Channel capacity
- The Gaussian channel
- Coding for a noisy channel (Block coding principles)
- Rate distortion theory

基本内容可以概括为：

$$IT \begin{cases} \text{通信的基本性能限} \\ \text{逼近性能限的方法} - \text{Coding: source coding, channel coding, network coding} \end{cases}$$

Textbook: T. M. Cover and J. A. Thomas, *Elements of Information Theory*, 2nd edition, New York, Wiley, 2006. (清华影印版，2003).

References:

1. C. E. Shannon, "A mathematical theory of communication", *Bell Syst. Tech. J.*, vol. 27, 1948.
2. R. G. Gallager, *Information Theory and Reliable Communication*, Wiley, 1968.
3. R. J. McEliece, *The Theory of Information and coding*, 1977. (This is a very readable small book)
4. Raymond W. Yeung, *Information Theory and Network Coding*, Springer, 2008. (中文版，2011)
5. J. Massey, *Digital Information Theory*, Course notes, ETH.
6. 王育民，李晖，梁传甲，*信息论与编码理论*，高等教育出版社，2005.
7. 仇佩亮，*信息论与编码*，高等教育出版社，2003.
8. 付祖芸，*信息论*，电子工业出版社.
9. R. G. Gallager, "Claude E. Shannon: A retrospective on his life, work, and impact," *IEEE Trans. Inform. Theory*, vol.47, no.7, pp. 2681-2695, Nov. 2001.

Chapter 1

Introduction

The fundamental problem of communication is that of reproducing at one point either exactly or approximately a message selected at another point. — Shannon

In 1948, Claude E. Shannon published a landmark paper entitled "A Mathematical Theory of Communication". This paper laid the groundwork of a entirely new scientific discipline, "Information Theory".

Information theory studies the transmission, processing and utilization of information.

1.1 Relationship of information theory and communication theory

Information theory answers two fundamental questions in communication theory:

1. What is the ultimate data compression? H
2. What is the ultimate transmission rate? C

Information theory also suggests means of achieving these ultimate limits of communication. (e.g. random coding, ML decoding)

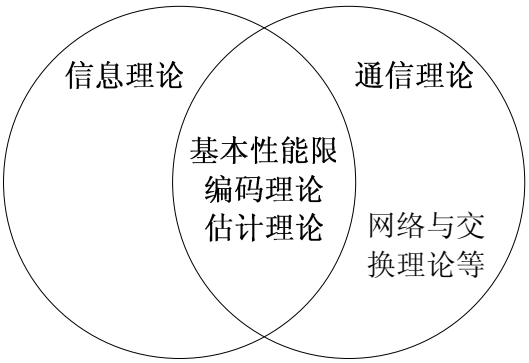


Figure 1.1: 信息理论和通信理论的关系示意图. Shannon理论主要研究基本限(fundamental limits)

通信系统传输的是信号；信号是消息的载体；消息中的不确定成分是信息。

- **狭义信息论(Shannon Theory)**
  Shannon在前人工作的基础上，用概率统计的方法研究通信系统。揭示了通信系统中传送的对象是信息；系统设计的中心问题是在干扰噪声中如何有效而可靠地传送信息。指出可以用编码方法实现这一目标；并在理论上证明了通信系统可达到的最佳性能限。
- 一般信息论：除Shannon理论外，还包括最佳接收理论（信号检测、估计与调制理论），噪声理论等。
- 广义信息论

信息论是通信与信息系统的基础理论，是现代通信发展的动力和源泉：

I have often remarked that the transistor and information theory, two Bell Laboratories breakthroughs within months of each other, have launched and powered the vehicle of modern digital communications. Solid state electronics provided the engine while information theory gave us the steering wheel with which to guide it. — Viterbi, IT News Lett., 1998.

- 信源编码定理→ 数据压缩技术→ 无线通信系统从1G变革到2G
- 信道编码定理→ 差错控制编码（Turbo,LDPC）→ 3G
- 数据处理定理→ 软判决译码
- 高斯噪声是最坏的加性噪声+ 多用户信息论→ CDMA、多用户检测
- MIMO容量理论→ 空时编码、预编码→ LTE、4G
- 多用户信息论→ 协作通信、网络编码→ 新一代无线系统

The recent work on the information-theoretic aspects of communication concentrated on: 1) Network information theory, and 2) MIMO systems.

1.2 What is information? (Measure of information)

For Shannon theory, information is what we receive when uncertainty is reduced.

How to measure:

- Amount of information should fulfill $I \geq 0$
- Amount of information should depend on probability $P(x)$
- For independent events: $P(X,Y) = P(X)P(Y) \rightarrow I = I(X) + I(Y)$

It should has the form of $\log \frac{1}{P_X(x)}$. (Self-information of the event $X = x$)

1.3 Applications

- Data compression: voice coder, MPEG, LZ algorithm.
- Modem

- Deep space communication (and coding was called a “marriage made in heaven”)
- CDMA, MIMO, 4G
- Physical layer security (Information-theoretic securiy)
- Quantum communication
- Stock market

1.4 Historical notes

- Sampling theorem: 1928 by Nyquist
- Hartley’s measure of information (1928)

$$I(X) = \log L,$$

  L=number of possible values of X.

- Information theory: 1948 by Shannon
  Investigate how to achieve the efficient and reliable communication
- Why using “entropy”?
  Shannon 与V. Neuman 讨论时, V. Neuman 建议用“熵”.

  1. 你的不确定函数在统计力学中已经被称为熵(entropy).
  2. 没有人知道熵到底是什么，所以有争论时你就永远立于不败之地.

- 在Shannon 1948年的原文中，既没有使用“mutual information”也没有用一个特殊符号来记它，而总是使用不确定性之差。术语“mutual information”及符号$I(X;Y)$是后来由Fano引入的.
- Shannon was born in Michigan, 1916. In 1936, he received B.S. degree in both electrical engineering and mathematics from the Univ. of Michigan. Received his M.S. and Ph.D. degree from MIT. In 1941, he joined Bell Lab. In 1958, he accepted a permanent appointment at MIT. 随后买了大房子, 房子里有很多玩具.
- Shannon的硕士论文是关于布尔代数与交换的，他基于此研究工作发表的第一篇论文won the 1940 Alfred Noble prize for the best paper in engineering published by an author under 30. It is widely recognized today as the foundation of the switching field and as one of the most important Master’s theses ever written. His Ph.D. dissertation, “An Algebra for Theoretical Genetics,” was completed in 1940. This thesis was never published.
- In 1961, Shannon published a pioneering paper ”Two-way Communication Channels”, which established the foundation for the discipline now known as ”multi-user information theory”; and later N. Abramson published his paper ”The Aloha System - Another Alternative for Computer Communications” in 1970 which introduced the concept of multiple access using a shared common channel. The information theoretic approach to multiaccess communication began in 1972 with a coding theorem developed by Ahlswede and Liao. In 1972, T. Cover published a paper “Broadcast channels”.

- 1995年，Gallager提出“Combining queueing theory with information theory for multiaccess”; 1998年，Ephremidus发表了论文“Information theory and communication networks: An unconsummated union”; 2000年，R. Alshwede, N. Cai, S.-Y. R. Li, and R. W. Yeung发表了著名论文“Network information flow”,提出了网络编码(Network coding)的思想; 2000年，P. Gupta和P. R. Kumar发表了论文“The Capacity of wireless networks”，提出了传送容量(Transport capacity)的概念；2003之后，研究正向着大规模网络的信息理论发展（Towards an IT of large networks）。

1.5 A model of digital communication systems

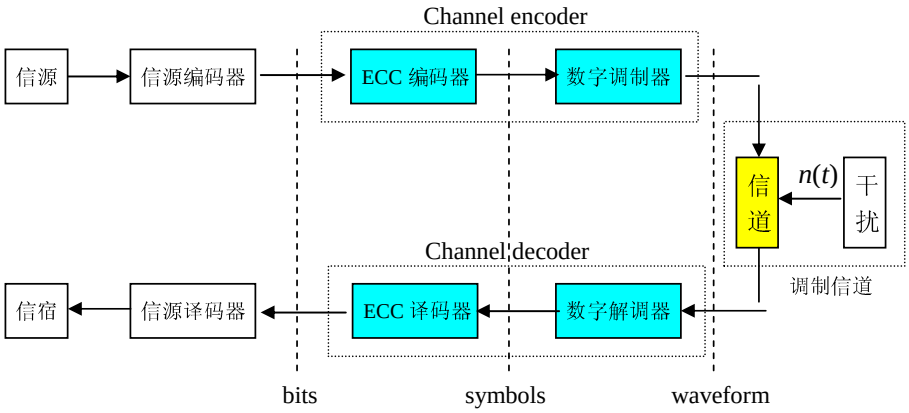


Figure 1.2: 数字通信系统示意图

- source: discrete/continuous; memoryless/with memory
- encoder: convert the messages into the signal which is suitable for transmission over physical channels.
- channel: wireless/cable, disk.
- interference.
- sink: destination.

1.6 Review of Probability

Bayes rule:

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

For a discrete random variable (r.v.),

- Probability Mass Function (PMF)

  $$P_X(x) = P(X = x)$$

  denotes the Prob. of the event that X takes on the value x.

For a continuous r.v.,

- Cumulative Distribution Function (CDF)

$$F_X(x) = P(X \leq x)$$

- Probability Density Function (PDF)

$$p_X(x) = \frac{\mathrm{d}}{\mathrm{dx}} F_X(x)$$

Chapter 2

Entropy & Mutual Information (Shannon's measure of information)

This chapter introduces basic concepts and definitions required for the discussion later. Mainly include: Entropy, Mutual information(互信息), and relative entropy(相对熵).

2.1 Entropy

Let X be a discrete variable with alphabet $\mathcal{X}$ and PMF $P_X(x) = \Pr(X = x), x \in \mathcal{X}$. For convenience, we will often write simply $P(x)$ for $P_X(x)$.

Definition 2.1.1. *The entropy of a discrete r.v. X is defined as*

$$\begin{aligned} H(X) &= \sum_{x\in\mathcal{X}} P(x) \log_b \frac{1}{P(x)} \\ &= -\sum_{x\in\mathcal{X}} P(x) \log_b P(x) \end{aligned} \tag{2.1}$$

when $b = 2$, the unit is called the *bit* (binary digit); when $b = e$, the unit is called the *nat* (natural unit). (Conversion is easy: $\log_b x = \log_b a \log_a x \Rightarrow H_b(x) = (\log_b a) H_a(x)$). Unless otherwise specified, we will take all logarithms to base 2, hence all entropies will be measured in bits.

In the above definition, we use the convention that $0 \log 0 = 0$. Note that equivalently, many books adopt the convention that the summation is taken over the corresponding support set. The support set of $P(X)$, denoted by $\mathcal{S}_X$, is the set of all $x \in \mathcal{X}$ such that $P(x) > 0$; i.e., $\mathcal{S}_X = supp(P_X) = \{x : P(x) > 0\}$.

The entropy $H(X)$ is also called the uncertainty of X, meaning that it is a measure of the randomness of X.

Note that the entropy $H(X)$ depends on the probabilities of different outcomes of X, but not on the names of the outcomes. For example,

$\mathcal{X} = \{Green, Blue, Red\}$	$\mathcal{Y} = \{Sunday, Monday, Friday\}$
$P(X) : 0.2, 0.3, 0.5$	$P(Y) : 0.2, 0.3, 0.5$

$$H(X) = H(Y)$$

Remark: The entropy of X can also be interpreted as the expected value of $\log \frac{1}{P(X)}$ (i.e., the average uncertainty):

$$H(X) = \mathsf{E} \log \frac{1}{P(X)} \tag{2.2}$$

where we define $\mathsf{E}[F(x)] = \sum_{x \in \mathcal{X}} P_X(x) F(x)$. Recall that $I(x) = \log \frac{1}{P_X(x)}$ is the self-information of the event $X = x$, so $H(X) = \mathsf{E}[I(x)]$ is also referred to as 平均自信息量。

A immediate consequence of the definition is that $H(X) \geq 0$.

Example 2.1.1. *Let*

$$X = \begin{cases} 1 & \textit{with probability } p \\ 0 & \textit{with probability 1-p} \end{cases}$$

Then

$$H(X) = -p \log p - (1-p) \log(1-p) \tag{2.3}$$

Equation (2.3) is often called the *binary entropy function*, and denoted by $H(p)$. Its graph is shown in Fig. 2.1. We can see that $H(X) = 1$ bit when $p = \frac{1}{2}$.

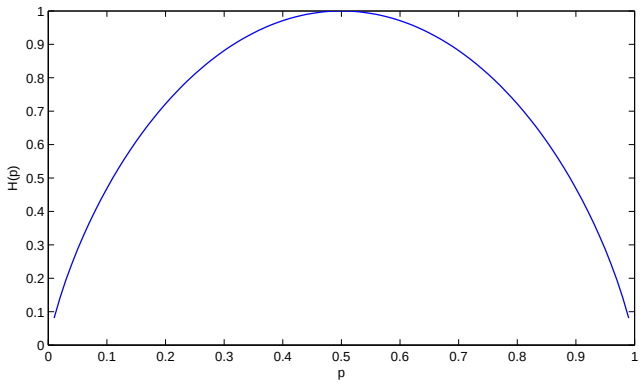


Figure 2.1: 二元随机变量的熵函数

Example 2.1.2. *Let*

$$X = \begin{cases} a & \textit{with probability } 1/2 \\ b & 1/4 \\ c & 1/8 \\ d & 1/8 \end{cases}$$

Then $H(X) = -\frac{1}{2}\log\frac{1}{2} - \frac{1}{4}\log\frac{1}{4} - \frac{1}{8}\log\frac{1}{8} - \frac{1}{8}\log\frac{1}{8} = \frac{7}{4}$ *bits.*

On the other hand, if X takes on values in $\mathcal{X} = \{a, b, c, d\}$ with equal probability, then we have $H(X) = -\frac{1}{4}\log\frac{1}{4} \times 4 = 2$ bits=$\log|\mathcal{X}|$.

We can see that the uniform distribution over the range $\mathcal{X}$ is the maximum entropy distribution over this range. (In other words, the entropy of X is maximized when its values are equally likely.)

2.2 Joint entropy and conditional entropy

We now extend the definition of the entropy of a single random variable to a pair of random variables.

(X, Y) can be considered to be a single vector-valued random variable.

Definition 2.2.1. *The joint entropy $H(XY)$ of a pair of discrete random variables (X, Y) with a joint distribution $P(x, y)$ is defined as*

$$\begin{aligned} H(XY) &\triangleq E[-\log P(X,Y)] \\ &= -\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} P(x,y)\log P(x,y) \\ &= -\sum_{(x,y)\in\mathcal{S}_{XY}} P(x,y)\log P(x,y) \end{aligned} \tag{2.4}$$

Definition 2.2.2. *The conditional entropy of the discrete random variable X, given that the event $Y = y$ occurs, is defined as*

$$\begin{aligned} H(X|Y=y) &= -\sum_{x\in\mathcal{X}} P(x|y)\log P(x|y) \\ &= E\left[-\log P(X|Y)|Y=y\right] \end{aligned} \tag{2.5}$$

Definition 2.2.3. *If $(X, Y) \sim P(x, y)$, then the conditional entropy of the discrete random variable X, given the discrete random variable Y, is defined as*

$$\begin{aligned} H(X|Y) &= \sum_{y\in\mathcal{Y}} P(y)H(X|Y=y) \\ &= -\sum_{y\in\mathcal{Y}} P(y)\sum_{x\in\mathcal{X}} P(x|y)\log P(x|y) \\ &= \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} P(x,y)\log P(x|y) \\ &= E[-\log P(X|Y)] \end{aligned} \tag{2.6}$$

Notice that $H(Y|X) \neq H(X|Y)$.

Theorem 2.2.1. $H(XY) = H(X) + H(Y|X) = H(Y) + H(X|Y)$

Proof.

$$\begin{aligned} H(XY) &= -\sum_x\sum_y P(x,y)\log P(x,y) \\ &= -\sum_x\sum_y P(x,y)\left[\log P(x) + \log P(y|x)\right] \\ &= -\sum_x P(x)\log P(x) - \sum_x\sum_y P(x,y)\log P(y|x) \\ &= H(X) + H(Y|X) \end{aligned} \tag{2.7}$$

□

Corollary 2.2.2. $H(XY|Z) = H(X|Z) + H(Y|XZ)$

We now generalize the above theorem to a more general case.

Theorem 2.2.3 (Chain rule for entropy)**.** *Let $X_1, X_2, \ldots, X_N$ be discrete random variables drawn according to $P(x_1, x_2, \ldots, x_N)$. Then*

$$H(X_1, X_2, \ldots, X_N) = \sum_{n=1}^{N} H(X_n|X_1, X_2, \ldots, X_{n-1})$$

Proof.

$$\begin{aligned} H(X_1, X_2, \cdots, X_N) &= E[-\log P(X_1, X_2, \cdots, X_N)] \\ &= E\left[-\log \prod_{n=1}^{N} P(X_n|X_1, \cdots, X_{n-1})\right] \\ &= \sum_{n=1}^{N} E[-\log P(X_n|X_1, \cdots, X_{n-1})] \\ &= \sum_{n=1}^{N} H(X_n|X_1, \cdots, X_{n-1}) \end{aligned} \tag{2.8}$$

□

If X_n are independent of each other, then

$$H(X_1, X_2, \cdots, X_N) = \sum_{n=1}^{N} H(X_n). \tag{2.9}$$

Similarly, we have

$$H(X_1, X_2, \cdots, X_N|Y) = \sum_{n=1}^{N} H(X_n|X_1, \cdots, X_{n-1}, Y).$$

2.3 Properties of the entropy function

Let X be a discrete random variable with alphabet $\mathcal{X} = \{x_k, k = 1, 2, \cdots, K\}$. Denote the pmf of X by $p_k = Pr(X = x_k), x_k \in \mathcal{X}$. Then the entropy $H(X)$ of X can be written as

$$H(\mathbf{p}) = -\sum_{k=1}^{K} p_k \log p_k$$

where $\mathbf{p} = (p_1, p_2, \cdots, p_K)$ is a K-vector of probabilities.

Property 2.3.1. $H(\mathbf{p}) \geq 0$ *(Non-negativity of entropy)*

Proof. Since $0 \leq p_k \leq 1$, we have $\log p_k \leq 0$. Hence, $H(\mathbf{p}) \geq 0$. □

Definition 2.3.1. *A function $f(x)$ is said to be convex-$\bigcup$ over an interval (a, b) if for every $x_1, x_2 \in (a, b)$ and $0 \leq \lambda \leq 1$,*

$$f(\lambda x_1 + (1 - \lambda)x_2) \leq \lambda f(x_1) + (1 - \lambda)f(x_2)$$

A function f is said to be strictly convex-$\bigcup$ if equality holds only when $\lambda = 0$ or $\lambda = 1$.

凸区域：若对于区域D中任意两点$\bar{\alpha}$ 和$\bar{\beta}$, $\bar{\alpha} \in D$, $\bar{\beta} \in D$, 有

$$\lambda\bar{\alpha} + (1-\lambda)\bar{\beta} \in D, \quad 0 \leq \forall\lambda \leq 1$$

则称D是凸区域。

凸函数：若定义在凸区域D上的函数$f(x)$满足

$$f(\lambda\bar{\alpha} + (1-\alpha)\bar{\beta}) \leq \lambda f(\bar{\alpha}) + (1-\lambda)f(\bar{\beta}), \qquad \forall\bar{\alpha}, \bar{\beta} \in D, 0 \leq \lambda \leq 1.$$

则称函数$f(x)$为凸-$\bigcap$ 函数.

Theorem 2.3.1. *If the function f has a second derivative which is non-negative (resp. positive) everywhere ($f''(x) \geq 0$), then the function is convex-$\bigcup$ (resp. strictly convex).*

Theorem 2.3.2 (Jensen's inequality)**.** *If f is a convex-$\bigcup$ function and X is a random variable, then*

$$E[f(x)] \geq f(E[X])$$

Proof. For a two mass point distribution, the inequality becomes

$$p_1 f(x_1) + p_2 f(x_2) \geq f(p_1 x_1 + p_2 x_2)$$

which follows directly from the definition of convex function.

Suppose that the theorem is true for distributions with $k-1$ mass points. Then writing $p_i' = \frac{p_i}{1-p_k}$ for $i = 1, 2, \ldots, k-1$, we have

$$\begin{aligned}
\sum_{i=1}^{k} p_i f(x_i) &= p_k f(x_k) + (1-p_k) \sum_{i=1}^{k-1} p_i' f(x_i) \\
&\geq p_k f(x_k) + (1-p_k) f(\sum_{i=1}^{k-1} p_i' x_i) \\
&\geq f\left(p_k x_k + (1-p_k) \sum_{i=1}^{k-1} p_i' x_i\right) \\
&= f(\sum_{i=1}^{k} p_i x_i)
\end{aligned} \tag{2.10}$$

□

Property 2.3.2. *$H(\mathbf{p})$ is the convex-$\bigcap$ function of $\mathbf{p}$.*

IT-inequality: For a positive real number r,

$$\log r \leq (r-1)\log e \tag{2.11}$$

with equality if and only if $r = 1$.

Proof. Let $f(r) = \ln r - (r-1)$. Then we have

$$f'(r) = \tfrac{1}{r} - 1 \text{ and } f''(r) = -\tfrac{1}{r^2} < 0.$$

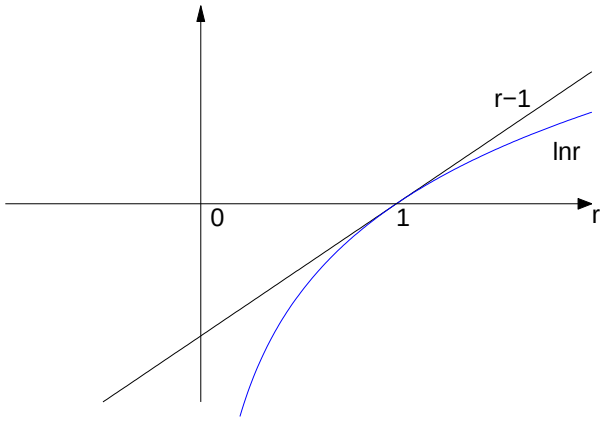


Figure 2.2: IT-inequality曲线

We can see that $f(r)$ is convex-$\bigcap$ in the interval $r > 0$. Moreover, the maximize value is zero, which is achieved with $r = 1$. Therefore,

$$\ln r \leq r - 1$$

with equality iff $r = 1$. Equation (2.11) follows immediately. □

Theorem 2.3.3. *If the discrete r.v. X has K possible values, then $H(X) \leq \log K$, with equality iff $P(x) = \frac{1}{K}$ for all x.*

Proof.

$$\begin{aligned} H(X) - \log K &= -\sum_{x \in \mathcal{S}_x} P(x) \log P(x) - \log K \\ &= \sum_x P(x)[\log \frac{1}{P(x)} - \log K] \\ &= \sum_x P(x) \log \frac{1}{KP(x)} \\ (By\ IT-equality) \quad &\leq \sum_x P(x)[\frac{1}{KP(x)} - 1] \log e \qquad (2.12) \\ &= [\sum_x \frac{1}{K} - \sum_x P(x)] \log e \\ &= [\sum_{x \in \mathcal{X}} \frac{1}{K} - 1] \log e \\ &= (1-1) \log e = 0 \end{aligned}$$

where equality holds in (2.12) iff $KP(x) = 1$ for all $x \in \mathcal{S}_x$. □

Theorem 2.3.4 (Conditioning reduces entropy)**.** *: For any two discrete r.v.'s X and Y,*

$$H(X|Y) \leq H(X)$$

with equality iff X and Y are independent.

(We can also use the relationship $I(X;Y) = H(X) - H(X|Y) \geq 0$ to obtain $H(X) \geq H(X|Y)$)

Proof.

$$\begin{aligned}
H(X|Y) - H(X) &= -\sum_{(x,y)\in\mathcal{S}_x y} P(x,y)\log P(x|y) + \sum_{x\in\mathcal{S}_x} P(x)\log P(x) \\
&= \sum_{x,y} P(x,y)\log\frac{P(x)}{P(x|y)} \\
&= \sum_{x,y} P(x,y)\log\frac{P(x)P(y)}{P(x,y)} \\
(By\ IT-equality) \quad &\leq \sum_{x,y} P(x,y)\left(\frac{P(x)P(y)}{P(x,y)} - 1\right)\log e \\
&= \left(\sum_{x,y} P(x)P(y) - \sum_{x,y} P(x,y)\right)\log e \\
&\leq \left(\sum_{x\in\mathcal{X},y\in\mathcal{Y}} P(x)P(y) - 1\right)\log e \\
&= (1-1)\log e \\
&= 0
\end{aligned}$$

□

Note that, however $H(X|Y=y)$ may exceed $H(X)$.

Theorem 2.3.5 (Independence bound on entropy)**.** *Let $X_1, \ldots, X_N$ be drawn according to $P(X_1, X_2, \ldots, X_N)$. Then*

$$H(X_1, X_2, \ldots, X_N) \leq \sum_{n=1}^{N} H(X_n)$$

with equality iff the X_n are independent.

Proof. By the chain rule for entropies,

$$\begin{aligned}
H(X_1, X_2, \ldots, X_N) &= \sum_{n=1}^{N} H(X_n|X_1, \ldots, X_n - 1) \\
&\leq \sum_{n=1}^{N} H(X_n) \qquad (2.13)
\end{aligned}$$

□

2.4 Relative entropy and mutual information

2.4.1 Relative entropy

The relative entropy is a measure of the "distance" between two distributions.

Definition 2.4.1. *The relative entropy (or Kullback Leiber distance) between two pmf $P(x)$ and $Q(x)$ is defined as*

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

It is sometimes called the information divergence/ cross-entropy/ Kullback entropy between $P(x)$ and $Q(x)$.

- In general, $D(P||Q) \neq D(Q||P)$, so that relative entropy does not have the symmetry required for a true "distance" measure.
- The divergence inequality: $D(P||Q) \geq 0$
  with equality iff $P(x) = Q(x)$ for all $x \in \mathcal{X}$.

Proof. Let $\mathcal{S}_X = \{x : P(x) > 0\}$ be the support set of $P(x)$. Then

$$\begin{aligned} -D(P||Q) &= \sum_{x\in\mathcal{S}_X} P(x)\log\frac{Q(x)}{P(x)} \qquad (2.14)\\ (By\ IT-inequality) \quad &\leq \sum_x P(x)\left[\frac{Q(x)}{P(x)} - 1\right]\log e \\ &= [\sum_x Q(x) - \sum_x P(x)]\log e \\ &\leq [\sum_{x\in\mathcal{X}} Q(x) - \sum_{x\in\mathcal{X}} P(x)]\log e \\ &= 0 \end{aligned}$$

□

2.4.2 Mutual information

Mutual information is a measure of the amount of information that one r.v. contains about another r.v. It is the reduction in the uncertainty of one r.v. due to the knowledge of the other.

Definition 2.4.2. *Consider two random variables X and Y with a joint pmf $P(x,y)$ and marginal pmf $P(x)$ and $P(y)$. The (average) mutual information $I(X;Y)$ between X and Y is the relative entropy between $P(x,y)$ and $P(x)P(y)$, i.e.,*

$$\begin{aligned} I(X;Y) &= D(P(x,y)||P(x)P(y)) \\ &= \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} P(x,y)\log\frac{P(x,y)}{P(x)P(y)} \\ &= E_{p(x,y)}\left[\log\frac{P(x,y)}{P(x)P(y)}\right] \\ &= \sum_x\sum_y P(x,y)\log\frac{P(x|y)}{P(x)} \qquad (2.15) \end{aligned}$$

Properties of mutual information:

1. Non-negativity of mutual information: $I(X;Y) \geq 0$
with equality iff X and Y are independent.

Proof. $I(X;Y) = D(P(x,y)||P(x)P(y)) \geq 0$, with equality iff $P(x,y) = P(x)P(y)$. □

2. Symmetry of mutual information: $I(X;Y) = I(Y;X)$

3.

$$\begin{aligned} I(X;Y) &= \sum_x \sum_y P(x,y) \log \frac{P(x|y)}{P(x)} \\ &= -\sum_{x,y} P(x,y) \log P(x) - (-\sum_{x,y} P(x,y) \log P(x|y)) \\ &= H(X) - H(X|Y) \end{aligned} \tag{2.16}$$

By symmetry, it also follows that $I(X;Y) = H(Y) - H(Y|X)$. Since $H(XY) = H(X) + H(Y|X)$, we have

$$I(X;Y) = H(X) + H(Y) - H(XY)$$

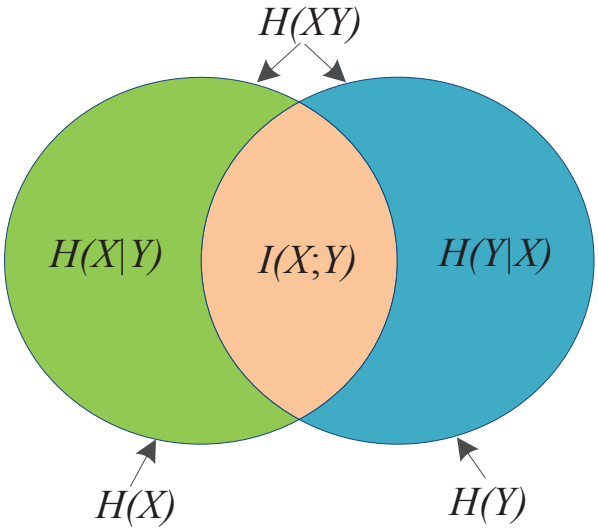


Figure 2.3: mutual information and conditional entropy

4. $I(X;X) = H(X) - H(X|X) = H(X)$. Hence, entropy is sometimes referred to as *average self-information*.

2.4.3 Conditional mutual information

Definition 2.4.3. *The conditional mutual information between the random variables X and Y, given the r.v. Z, is defined by*

$$\begin{aligned} I(X;Y|Z) &= H(X|Z) - H(X|YZ) \\ &= \sum_x \sum_y \sum_z P(x,y,z) \log \frac{P(x,y|z)}{P(x|z)P(y|z)} \\ &= E_{p(x,y,z)} \left[\log \frac{P(x|yz)}{P(x|z)} \right] \end{aligned} \tag{2.17}$$

It follows from the definition that

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

with equality iff conditional on each Z, X and Y are statistically independent; i.e., $P(x,y|z) = P(x|z)P(y|z)$ for each element in the joint sample space for which $P(z) > 0$.

We can visualize the situation in which $I(X;Y|Z) = 0$ as a pair of channels in cascade as shown in Fig.1. We assume that the output of the 2nd channel depends statistically only on the input to the 2nd channel,i.e. $p(y|z) = p(y|z,x)$, all x, y, z with $p(z,x) > 0$.

Mutiplying both sides by $P(x|z)$, we obtain $P(x,y|z) = P(x|z)P(y|z)$, so that $I(X;Y|Z) = 0$.

An important property:

$$\begin{aligned} I(X;YZ) &= I(YZ;X) \\ &= I(X;Y) + I(X;Z|Y) \\ &= I(X;Z) + I(X;Y|Z) \end{aligned} \tag{2.18}$$

Proof.

$$\begin{aligned} I(X;YZ) &= H(YZ) - H(YZ|X) \\ &= H(Y) + H(Z|Y) - [H(Y|X) + H(Z|XY)] \\ &= [H(Y) - H(Y|X)] + [H(Z|Y) - H(Z|XY)] \\ &= I(X;Y) + I(X;Z|Y) \end{aligned} \tag{2.19}$$

□

Theorem 2.4.1 (chain rule for mutual information)**.**

$$I(X_1, X_2, \ldots, X_N; Y) = \sum_{n=1}^{N} I(X_n; Y|X1, \ldots, X_{n-1})$$

Proof.

$$\begin{aligned} I(X_1, X_2, \ldots, X_N; Y) &= H(X_1, X_2, \ldots, X_N) - H(X_1, X_2, \ldots, X_N|Y) \\ &= \sum_{n=1}^{N} H(X_n|X_1, X_2, \ldots, X_{n-1}) - \sum_{n=1}^{N} H(X_n|X_1, X_2, \ldots, X_{n-1}, Y) \\ &= \sum_n I(X_n; Y|X_1, \ldots, X_{n-1}) \end{aligned} \tag{2.20}$$

□

2.4.4 Data processing inequality

The data processing inequality can be used to show that no clever manipulation of the data can improve the inferences that can be made from the data.

Definition 2.4.4. *Random variables X, Z, Y are said to form a Markov Chain in that order (denoted by $X \to Z \to Y$) if the conditional distribution of Y depends only on Z and is conditionally independent of X. Specifically,*

$$X \to Z \to Y \quad if P(x, z, y) = P(x)P(z|x)P(y|z)$$

Theorem 2.4.2 (Data processing inequality)**.** *If $X \to Z \to Y$, then*

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

Proof. By the chain rule, we obtain

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

Since X and Y are independent given Z, we have $I(X; Y|Z) = 0$. Thus,

$$I(X; Z) = I(X; Y) + I(X; Z|Y) \tag{2.21}$$

From the non-negativity of mutual information, $I(X; Z|Y) \geq 0$. Thus, (2.21) implies that

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

□

From the symmetry, it follows that

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

This theorem demonstrates that no processing of Z, deterministic or random, can increase the information that Z contains about X.

Corollary 2.4.3. *In particular, if $Y = f(Z)$, we have $I(X; Z) \geq I(X; f(Z))$.*

Proof. $X \to Z \to f(Z)$ forms a Markov Chain. □

Corollary 2.4.4. *If $X \to Z \to Y$, then $I(X; Z|Y) \leq I(X; Z)$.*

Proof. Using the fact $I(X; Y) \geq 0$, the corollary follows immediately from (2.21). □

Expressing the data processing inequality in terms of entropies, we have

$$\begin{aligned} &H(X) - H(X|Z) \geq H(X) - H(X|Y) \\ &\Rightarrow H(X|Z) \leq H(X|Y) \end{aligned} \tag{2.22}$$

The average uncertainty $H(X|Z)$ about the input of a channel given the output is called *the equivocation on the channel*, and thus the above inequality yields the intuitively satisfying result that this uncertainty or equivocation can never decrease as we go further from the input on a sequence of cascaded channels.

Example 2.4.1. *Suppose that the random vector $[X, Y, Z]$ is equally likely to take on values in $\{[000], [010], [100], [101]\}$. Then*

$$H(X) = h(\frac{1}{2}) = 1 \text{ bit}$$

Note that $P_{Y|X}(0|1) = 1$ so that

$$H(Y|X = 1) = 0$$

Similarly, $P_{Y|X}(0|0) = \frac{1}{2}$, so that

$$H(Y|X = 0) = h(\frac{1}{2}) = 1 \text{ bit}$$

Thus, $H(Y|X) = \frac{1}{2} \times 1 = \frac{1}{2}$ bit.

$$\begin{aligned} H(Z|XY) &= \sum_{x,y} P(x, y)H(Z|X = x, Y = y) \\ &= \frac{1}{4}(0) + \frac{1}{4}(0) + \frac{1}{2}(1) \\ &= \frac{1}{2} \text{ bit} \end{aligned}$$

and $H(XYZ) = H(X) + H(Y|X) + H(Z|XY) = 1 + \frac{1}{2} + \frac{1}{2} = 2$ bits.

Because $P_Y(1) = \frac{1}{4}, P_Y(0) = \frac{3}{4}$, we have

$$H(Y) = h(\frac{1}{4}) = 0.811 \quad \text{bits}$$

we see that $H(Y|X) = \frac{1}{2} < H(Y)$. However, $H(Y|X = 0) = 1 > H(Y)$.

Furthermore, $I(X; Y) = H(Y) - H(Y|X) = 0.811 - 0.5 = 0.311$ bits
In words, the first component of the random vector $[X, Y, Z]$ gives 0.311 bits of information about the 2nd component, and vice versa.

2.5 Sufficient Statistics

Suppose we have a family of pmfs $\{f_\theta(x)\}$ indexed by θ, and let X be a sample from a distribution in this family. Let $T(X)$ be any statistic (like the sample mean or variance). Then

$$\theta \to X \to T(X), \text{ and } I(\theta; T(X)) \leq I(\theta; X)$$

A statistic $T(X)$ is called sufficient for θ if it contains all the information in X about θ.

Definition 2.5.1. *A function $T(X)$ is said to be a sufficient statistic relative to the family $\{f_\theta(x)\}$ if X is independent of θ given $T(X)$; i.e., $\theta \to X \to T(X)$ forms a Markov Chain.*

It means that $I(\theta; X) = I(\theta; T(X))$ for all distributions on θ.

Example 2.5.1. *Let $X_1, \ldots, X_n, X_i \in \{0, 1\}$, be an i.i.d sequence of coin tosses of a coin with unknown parameter $\theta = Pr(X_i = 1)$. Given n, the number of 1's is a sufficient statistics for θ. Here, $T(X_1, \ldots, X_n) = \sum_{i=1}^n X_i$.*

Example 2.5.2. *If $X \sim N(0,1)$, i.e., if*

$$f_\theta(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{(x-\theta)^2}{2}} = N(\theta, 1)$$

and $X_1, \ldots, X_n$ are drawn independently according to this distribution, then $\bar{X}_n = T(X_1, \ldots, X_n) = \frac{1}{n}\sum_{i=1}^n X_i$ is a sufficient statistic for θ.

2.6 Fano's inequality

Suppose that we wish to estimate a r.v. X based on a correlated r.v. Y. Fano's inequality relates the probability of error in estimating X to its conditional entropy $H(X|Y)$.

Theorem 2.6.1. *For any r.v.'s X and Y, we try to guess X by $\hat{X} = g(Y)$. The error probability $P_e = P(\hat{X} \neq X)$ satisfies*

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

Proof. Applying chain rule on $H(X, E|Y)$, where error r.v.

$$E = \begin{cases} 1, & \text{if } \hat{X} \neq X \\ 0, & \text{if } \hat{X} = X \end{cases}$$

we have

$$\begin{aligned}
H(E, X|Y) &= H(X|Y) + H(E|XY) = H(X|Y) \\
H(E, X|Y) &= H(E|Y) + H(X|EY) \\
&\leq H(E) + H(X|EY) \\
&= H(P_e) + P(E=1)H(X|Y, E=1) + P(E=0)H(X|Y, E=0) \\
&= H(P_e) + P_e H(X|Y, E=1) + (1 - P_e)H(X|Y, E=0) \\
&= H(P_e) + P_e H(X|Y, E=1) + 0 \\
&\leq H(P_e) + P_e \log(|\mathcal{X}| - 1) \quad X\text{取除}\hat{X}\text{之外的其它}|\mathcal{X}| - 1\text{个值} \\
&\leq 1 + P_e \log |\mathcal{X}| \qquad (2.23)
\end{aligned}$$

□

物理解释：观察到Y（即已知Y）的条件下，对X还存在的不确定性可分为两部分：

- 判断猜测结果是否正确，其不确定性为$H(P_e)$；
- 如果判决是错的（with probability P_e），则这时X可能取除$\hat{X}$之外的其它$|\mathcal{X}| - 1$个值。为了确定是哪一个，所需的信息量$\leq \log(|\mathcal{X}| - 1)$.

2.7 Convex functions (互信息的凸性)

2.7.1 Concavity of entropy

Theorem 2.7.1. *Entropy $H(X)$ is concave in $P(x)$.*

That is, if X_1, X_2 are r.v.'s defined on $\mathcal{X}$ with distribution $P_1(x)$ and $P_2(x)$, respectively. For any $\theta \in [0,1]$, consider a r.v. X with

$$P_X(x) = \theta P_1(x) + (1-\theta)P_2(x) \quad \forall x$$

then

$$H(X) \geq \theta H(X_1) + (1-\theta)H(X_2)$$

Proof. Let Z be a binary r.v. with $P(Z=0) = \theta$. Let

$$X = \begin{cases} X_1 & \text{if } Z=0 \\ X_2 & \text{if } Z=1 \end{cases}$$

Then

$$\begin{aligned} H(X) &\geq H(X|Z) \\ &= \theta H(X|Z=0) + (1-\theta)H(X|Z=1) \\ &= \theta H(X_1) + (1-\theta)H(X_2) \end{aligned} \tag{2.24}$$

or

$$H(\theta P_1 + (1-\theta)P_2) \geq \theta H(P_1) + (1-\theta)H(P_2)$$

□

2.7.2 Concavity of mutual information

Theorem 2.7.2. *For a fixed transition probability $P(y|x)$, $I(X;Y)$ is a concave (convex-$\bigcap$) function of $P(x)$.*

Proof. Construct X_1,X_2, X, and Z as above. Consider

$$\begin{aligned} I(XZ;Y) &= I(X;Y) + I(Z;Y|X) \\ &= I(Z;Y) + I(X;Y|Z) \end{aligned} \tag{2.25}$$

Conditioned on X, r.v.'s Y and Z are independent, i.e., $P(y|x,z) = P(y|x)$. Using $I(Y;Z|X) = 0$, we have

$$\begin{aligned} I(X;Y) &\geq I(X;Y|Z) \\ &= \theta I(X;Y|Z=0) + (1-\theta)I(X;Y|Z=1) && (2.26) \\ &= \theta I(X_1;Y) + (1-\theta)I(X_2;Y) && (2.27) \end{aligned}$$

□

Theorem 2.7.3. *For a fixed input distribution $P(x)$, $I(X;Y)$ is convex-$\bigcup$ in $P(y|x)$.*

Proof. Consider a r.v. X and two channels with $P_1(y|x)$ and $P_2(y|x)$. When feed with X, the outputs of the two channels are denoted by Y_1 and Y_2.

Now let one channel be chosen randomly according to a binary r.v. Z that is independent of X, and denote the output by Y.

$$\begin{aligned} I(X;YZ) &= I(X;Z) + I(X;Y|Z) \\ &= I(X;Y) + I(X;Z|Y) \end{aligned} \tag{2.28}$$

Thus, $I(X;Y) < I(X;Y|Z) = \theta I(X;Y_1) + (1-\theta)I(X;Y_2)$. □

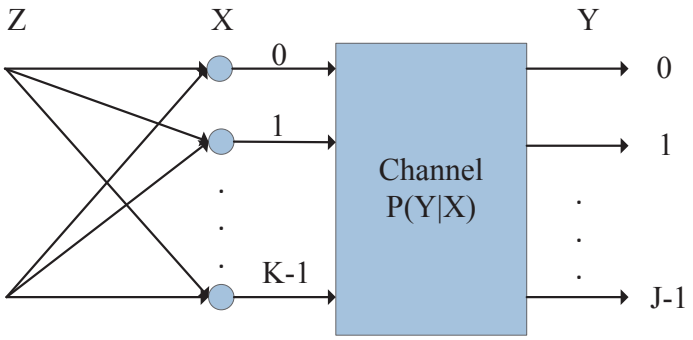


Figure 2.4: 给定信道下互信息的凸性

2.7.3 互信息的凸性—另一证明方法

Theorem 2.7.4. *Let $(X,Y) \sim P(x,y) = P(x)P(y|x)$. The mutual information $I(X;Y)$ is a convex-$\bigcap$ function of $P(x)$ for fixed $P(y|x)$ and a convex-$\bigcup$ function of $P(y|x)$ for fixed $P(x)$.*

Proof.

$$\begin{aligned} I(X;Y) &= H(Y) - H(Y|X) \\ &= H(Y) - \sum_x P(x)H(Y|X=x) \end{aligned} \tag{2.29}$$

$$\left\{ \begin{array}{l} \text{if } P(y|x) \text{ is fixed, then } P(y) \text{ is a linear function of } P(x) \\ H(Y) \text{ is a convex-}\bigcap \text{ function of } P(y) \end{array} \right.$$

$$\left. \begin{array}{l} \Rightarrow \ H(Y) \text{ is a convex-}\bigcap \text{ function of } P(x) \\ \text{The 2nd term in (2.29) is a linear function of } P(x) \end{array} \right\} \Rightarrow$$

The difference is a convex-$\bigcap$ function of $P(x)$.

To prove the 2nd part, we fix $P(x)$ and consider two different conditional distributions $P_1(y|x)$ and $P_2(y|x)$.

Let

$$P_\lambda(y|x) = \lambda P_1(y|x) + (1-\lambda)P_2(y|x)$$

Then

$$P_\lambda(x,y) = \lambda P_1(x,y) + (1-\lambda)P_2(x,y)$$

and

$$P_\lambda(y) = \lambda P_1(y) + (1-\lambda)P_2(y)$$

where $P_1(x,y) = P_1(y|x)P(x)$ and $P_2(x,y) = P_2(y|x)P(x)$.

If we let $Q_\lambda(x,y) = P(x)P_\lambda(y)$, then we have

$$\begin{aligned} Q_\lambda(x,y) &= P(x)[\lambda P_1(y) + (1-\lambda)P_2(y)] \\ &= \lambda Q_1(x,y) + (1-\lambda)Q_2(x,y) \end{aligned} \tag{2.30}$$

Since $I(X;Y) = D(P_\lambda||Q_\lambda)$ and $D(P||Q)$ is a convex function of (P,Q). it follows that $I(X;Y)$ is a convex-$\bigcup$ function of $P(y|x)$. □

Chapter 3

The Asymptotic Equipartition Property (渐进等同分割性)

3.1 Convergence of random variables

$$\lim_{n\to\infty} \frac{1}{n}\sum_{i=1}^{n} X_i \to \mathsf{E}[X] \qquad \text{for i.i.d. r.v.'s}$$

3.1.1 Type of convergence

- Almost sure convergence (also called convergence with Probability 1):

  $$P\left(\lim_{n\to\infty} Y_n(w) = Y(w)\right) = 1$$

  write $Y_n \underset{a.s.}{\longrightarrow} Y$

- Mean-square convergence:

  $$\lim_{n\to\infty} E[|Y_n - Y|^2] = 0$$

- Convergence in probability: $\forall \varepsilon > 0$

  $$\lim_{n\to\infty} P\left(|Y_n(w) - Y(w)| > \varepsilon\right) = 0$$

- Convergence in distribution: The CDF $F_n(y) = Pr(Y_n \leq y)$ satisfy:

  $$\lim_{n\to\infty} F_n(y) \to F_Y(y)$$

  at all y for which F is continuous.

3.1.2 Weak law of large numbers (WLLN)

$x_1, x_2, \cdots$, i.i.d., finite mean μ and variance σ^2.

$$S_n = \frac{x_1 + \cdots + x_n}{n}$$

Weak LLN:

$$P\left(|\frac{1}{n}\sum_{i=1}^{n} x_i - \mu| \geq \varepsilon\right) \leq \frac{\sigma^2}{n\varepsilon^2} \triangleq \delta$$

.

3.2 The AEP

Theorem 3.2.1 (AEP)**.** *If $X_1, X_2, \cdots, X_n$ are i.i.d. $\sim P(x)$, then*

$$-\frac{1}{n}\log P(x_1, x_2, \cdots, x_n) \to H(X) \qquad \textit{in probability}$$

Proof. Create r.v. $Y_i = \log P(X_i)$. Then apply the WLLN to Y:

$$\begin{aligned} -\frac{1}{n}\log P(x_1, x_2, \cdots, x_n) &= -\frac{1}{n}\sum_{i=1}^{n}\log P(X_i) \\ &= -\frac{1}{n}\sum_{i=1}^{n} Y_i \to -E[Y] \quad \text{in prob} \\ &= H(X) \end{aligned} \tag{3.1}$$

i.e., for all $\varepsilon > 0$,

$$\lim_{n\to\infty} P\left(\left|-\frac{1}{n}\log P(x_1, x_2, \cdots, x_n) - H(X)\right| \leq \varepsilon\right) = 1 (> 1 - \sigma)$$

□

Definition 3.2.1. *The typical set T_ε with respect to $P(x)$ is defined as*

$$T_\varepsilon = \left\{\mathbf{x} \triangleq (x_1, x_2, \cdots, x_n) \in \mathcal{X}^n : \left|-\frac{1}{n}\log P(\mathbf{x}) - H(X)\right| \leq \varepsilon\right\}$$

Thus, T_ε is the set of sequence $(x_1, x_2, \cdots, x_n) \in \mathcal{X}^n$ for which the sample average of the log pmf is within ε of its mean $H(X)$.

It can be rewritten as

$$T_\varepsilon = \left\{\mathbf{x} \in \mathcal{X}^n : 2^{-n[H(X)+\varepsilon]} \leq P(\mathbf{x}) \leq 2^{-n[H(X)-\varepsilon]}\right\}$$

which implies that the sequences in T_ε are approximately equiprobable.

As a consequence of AEP, we have

$$P(\mathbf{x} \in T_\varepsilon) > 1 - \delta \to 1 \qquad \text{(高概率集合)}$$

Theorem 3.2.2 (size of the typical set)**.** *The number of elements, $|T_\varepsilon|$, in the typical set T_ε satisfies*

$$(1-\varepsilon)2^{n[H(X)-\varepsilon]} \leq |T_\varepsilon| \leq 2^{n[H(X)+\varepsilon]}$$

Proof. Since $P(\mathbf{x}) \geq 2^{-n[H(X)+\varepsilon]}$ for each $\mathbf{x} \in T_\varepsilon$, we have

$$1 = \sum_{\mathbf{x}\in\mathcal{X}^n} P(\mathbf{x}) \geq \sum_{\mathbf{x}\in T_\varepsilon} P(\mathbf{x}) \geq \sum_{\varepsilon} 2^{-n[H(X)+\varepsilon]} = |T_\varepsilon| 2^{-n[H(X)+\varepsilon]}$$

This implies that $|T_\varepsilon| \leq 2^{n[H(X)+\varepsilon]}$.

Conversely, since $P_r(T_\varepsilon) > (1 - \frac{\sigma^2}{n\varepsilon}) = 1 - \delta = 1 - \varepsilon$, we have

$$1 - \varepsilon < \sum_{\mathbf{x}\in T_\varepsilon} P(\mathbf{x}) \leq \sum_{T_\varepsilon} 2^{-n[H(X)-\varepsilon]} = |T_\varepsilon| 2^{-n[H(X)-\varepsilon]}$$

which implies $|T_\varepsilon| > (1-\varepsilon)2^{n[H(X)-\varepsilon]}$ □

Compare to $|x^n| = 2^{n \log |x|}$:

Let

$$\alpha \triangleq \frac{|T_\varepsilon|}{|x^n|} \leq 2^{-n[\log|x| - H(x) - \varepsilon]} \xrightarrow{\text{as n}\uparrow} 0$$

即典型序列的数目远比非典型序列少.

Summary:

We conclude that for large n, the typical set T_ε has aggregate probability approximately 1 and contains approximately $2^{n[H(X)]}$ elements, each of which has probaility approximately $2^{-nH(X)}$. That is, asymptotically for very large n, the r.v. X^n resembles an equiprobable source with alphabet size $2^{nH(X)}$.

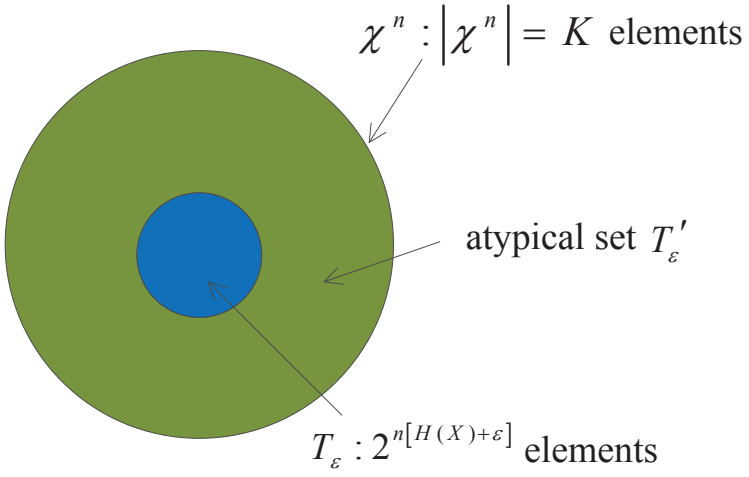


Figure 3.1: 典型序列集示意图

Example 3.2.1. *Consider binary r.v.'s X_i, i.i.d. with $P(X = 0) = p$ and $P(X = 1) = 1 - p$.*

A "typical" sequence of length n has roughly np 0's and $n(1-p)$ 1's, the probability for that to happen is

$$p^{np}(1-p)^{n(1-p)} = 2^{n[p\log p]+(1-p)\log(1-p)} = 2^{-nH(X)}$$

How many "typical" sequences are there?

$$\begin{aligned}\binom{n}{np} &= \frac{n!}{(np)!(n(1-p))!} \approx \frac{n^n e^{-n}}{(np)^{np}e^{-np}(n(1-p))^{n(1-p)}e^{-n(1-p)}} \\ &= \frac{1}{p^{np}(1-p)^{n(1-p)}} = 2^{nH(X)} \end{aligned} \tag{3.2}$$

(Note: Stirling formula: $n! \approx n^n e^{-n}\sqrt{2\pi n}$)

3.3 Using the typical set for data compression

Motivated by the AEP, we divided all sequence in $\mathcal{X}^n$ into two sets: T_ε and T_ε^c (its complement).

3.3.1 Source encoding method

- We order all elements in the T_ε and T_ε^c according to some order.
- Then we can represent each sequence in T_ε by giving the index of length $\leq n[H(X)+\varepsilon]+1$ bits (correction of 1 bit because of integrality)
- We prefix all these sequences by a $0 \Rightarrow$ total length $\leq n[H(X)+\varepsilon]+2$
- Similarly, we can index each sequence not in T_ε by using no more than $n\log|\mathcal{X}|+1$ bits. Prefix these indices by 1.
- Thus, we have a code for all the sequences in $\mathcal{X}^n$.

3.3.2 The average length of codeword

Let $l(\mathbf{x})$ = length of the binary codeword corresponding $\mathbf{x}$.

Then the expected length of the codeword is

$$\begin{aligned} E[l(\mathbf{x})] &= \sum_{\mathbf{x}\in T_\varepsilon} P(\mathbf{x})l(\mathbf{x}) + \sum_{\mathbf{x}\in T_\varepsilon^c} P(\mathbf{x})l(\mathbf{x}) \\ &\leq \sum_{T_\varepsilon} P(\mathbf{x})[n(H(X)+\varepsilon)+2] + \sum_{T_\varepsilon^c} P(\mathbf{x})[n\log|x|+2] \\ &\leq (1-\varepsilon)[n(H(X)+\varepsilon)+2] + \varepsilon(n\log|x|+2) \\ &= n(H(x)+\varepsilon) + 2 - \varepsilon[n(H(X)+\varepsilon)+2] + \varepsilon n\log|\mathcal{X}| + 2\varepsilon \\ &\leq n(H(X)+\varepsilon) + \varepsilon n\log|\mathcal{X}| + 2 \\ &= n(H(X)+\varepsilon'). \end{aligned} \tag{3.3}$$

where $\varepsilon' = \varepsilon + \varepsilon\log|\mathcal{X}| + \frac{2}{n}$.

So $E[\frac{1}{n}l(\mathbf{x})] \leq H(X)+\varepsilon$.

Chapter 4

Entropy Rates of a Stochastic Process

4.1 Stochastic processes

- A stochastic process is an indexed sequence of random variables $X_1, X_2, \ldots$, a map from Ω to $\mathcal{X}^\infty$.
- A stochastic process is characterized by the joint PMF:

$$P_{X_1 X_2 \ldots X_n}(x_1, x_2, \ldots, x_n)$$

- The entropy of a stochastic process

$$H(X_1, X_2, \ldots) = H(X_1) + H(X_2|X_1) + \cdots + H(X_i|X_1 \ldots X_{i-1}) + \ldots$$

- Difficulties:

$$\begin{cases} \text{Sum to infinity.} \\ \text{All terms are different in general.} \end{cases}$$

4.2 Entropy rate

- The entropy rate of a stochastic process $\{X_i\}$ is defined by

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

if it exists.

4.3 Entropy rate of stationary processes

- Chain rule:

$$\frac{1}{n} H(X_1, X_2, \ldots, X_n) = \frac{1}{n} \sum_{i=1}^{n} H(X_i|X_1 \ldots X_{i-1})$$

- For a stationary process,

$$\begin{aligned} H(X_{n+1}|{X_1}^n) &\leq H(X_{n+1}|{X_2}^n) \\ &= H(X_n|{X_1}^{n-1}). \end{aligned} \tag{4.1}$$

  Therefor, the sequence $H(X_n|{X_1}^{n-1})$ is non-increasing and non-negative, so limit exists.

- **Theorem 4.3.1.** *For a stationary process, the entropy rate*

$$\lim_{n\to\infty} \frac{1}{n} H({X_1}^n) = \lim_{n\to\infty} H(X_n|{X_1}^{n-1})$$

- Markov Chain: A discrete stochastic process is a Markov Chain if

$$P_{X_n|X_0\ldots X_{n-1}}(x_n|x_0,\ldots,x_{n-1}) = P_{X_n|X_{n-1}}(x_n|x_{n-1})$$

  for $n = 1, 2, \ldots$, and all $(x_0, \ldots, x_n) \in \mathcal{X}^{n+1}$

- Denote $p_{ij} = P(X_{n+1} = j|X_n = i)$

- The entropy rate of Markov Chain:

$$\lim_{n\to\infty} \frac{1}{n} H({X_1}^n) = \lim_{n\to\infty} \frac{1}{n} H(X_n|X_{n-1}) = -\sum_{i,j} \pi_i p_{ij} \log p_{ij}$$

- Significance of the entropy rate of a stochastic process arises from the AEP for a stationary ergodic process:

$$-\frac{1}{n} \log P(X_1, X_2, \ldots, X_n) \to H(\mathcal{X}) \text{ with probability 1.}$$

Chapter 5

Coding for Discrete Sources

5.1 Introduction

Three important classes of sources:

- Discrete sources (离散时间、离散信号取值集合)
  - The output of a discrete source is a sequence of letters from a given discrete alphabet $\mathcal{X}$.
  - A discrete alphabet is either a finite set of letters or a countably infinite set of letters.
- Analog sources (also called continuous-time, continuous-amplitude sources)
  - The output is an analog waveform.
- Discrete-time continuous-amplitude sources
  The output is a sequence of values which could be real number or multi-dimensional real numbers.
- 其它分类:

$$\left\{\begin{array}{l}\text{无记忆源}\\ \text{有记忆源}\end{array}\right. \qquad \left\{\begin{array}{l}\text{高斯源}\\ \text{Markov源}\end{array}\right. \qquad \left\{\begin{array}{l}\text{平稳源}\\ \text{各态历经源}\end{array}\right.$$

- *Discrete memoryless source* (DMS): DMS is a device whose output is a semi-infinite i.i.d. sequence of random variables $X_1, X_2, \ldots$, drawn from the finite set $\mathcal{X}$.
- An important parameter of a source is the rate R_s [source letters/s].

5.2 Coding a single random variable

- Definition: A source code $\mathcal{C}$ for a random variable X is a mapping from $\mathcal{X}$ to $\mathcal{D}^*$, the set of finite length strings of symbols from a D-ary alphabet.
- The same definition applies for sequence of r.v.'s, X^n.

- x (or x^n) – source symbol (string) $\in \mathcal{X}$.
  $\mathcal{D}$ – set of coded symbols.
  $C(x)$ – codeword corresponding to x.
  $\ell(x)$ – length of $C(x)$.
- For example, $\mathcal{X} = \{red, blue\}$, $C(red) = 00$, $C(blue) = 11$ is a source code with alphabet $\mathcal{D} = \{0, 1\}$.
- Without loss of generality, we will assume that $\mathcal{D} = \{0, 1 \ldots, D-1\}$.

Definition 5.2.1. *The expected length, $L(C)$, of a code is given by*

$$L(C) = \sum_{x \in \mathcal{X}} P_X(x)\ell(x) = \mathsf{E}[\ell(x)]$$

Goal: For a given source, find a code to minimize the expected length (per source symbol).

5.3 Fixed-length source codes (等长编码)

- Convert each source letter individually into a fixed-length block of ℓ D-ary symbols.
- The number of letters in the source alphabet, $K = |\mathcal{X}|$, satisfies $K \leq D^\ell$, then a different D-ary sequence of length ℓ may be assigned to each letter $x \in \mathcal{X}$. The resulting code is uniquely decodable.
- For example, for $\mathcal{X} = \{a, b, c, d, e, f, g\}$, $K = 7$, $\mathcal{D} = \{0, 1\}$, there exists an invertible mapping for $\mathcal{X}$ to binary 3-tuples:

$$a \to 000, b \to 001, \ldots, g \to 110$$

X	$C(x)$
a	000
b	001
⋮	⋮

Figure 5.1: $\ell = 3$的信源编码

- We can see that this coding method requires $\ell = \lceil \log |x| \rceil$ bits to encode each source letter.
- If we want to encode blocks of n source symbols at a time, the resulting source alphabet is the n-fold Cartesian product $\mathcal{X}^n = \mathcal{X} \times \mathcal{X} \times \cdots \times \mathcal{X}$, which has size $|\mathcal{X}^n| = K^n$.

- Using fixed-length source coding, we can encode each block of n source symbols into $\ell = \lceil \log_2 K^n \rceil$ bits. The rate $R = \frac{\ell}{n}$ of coded bits required per source symbol is then

$$R = \frac{\lceil \log_2 K^n \rceil}{n} \geq \frac{n \log_2 K}{n} = \log_2 K$$
$$R = \frac{\lceil \log_2 K^n \rceil}{n} < \frac{n \log_2 K + 1}{n} = \log_2 K + \frac{1}{n}$$

  If n is sufficiently large, then $R \to \log_2 K$.

5.3.1 编码定理

Goal: Minimize the average rate $R = \mathsf{E}[l(x^n)]/n$

- 对于等长编码，令M为待编码消息序列个数（或码字总数，codebook size），则$R = \frac{1}{n} \log_2 M$
- Encoder "compresses" x^n into an index $w \in \{1, 2, \cdots, 2^{nR}\}$. That is, the encoder sends nR bits for every source sequence x^n.
- 若采用D元等长编码，码长为L，则D元码字个数$M = D^L$，$R = \frac{L}{n} \log_2 D$

编码方法：Data compression by AEP

- Use $n \log |\mathcal{X}| + 1$ bits to describe (index) any sequence in $\mathcal{X}^n$.
- Since $|T_\varepsilon| \leq 2^{n(H+\varepsilon)}$, we use $n(H + \varepsilon) + 1$ bits to index all sequence in T_ε.
- Use an extra bit to indicate T_ε.
- $\mathsf{E}[\ell(x^n)] = \sum_{x^n} P(x^n)\ell(x^n) \leq n[H(X) + \varepsilon] \Rightarrow R = \frac{1}{n} E[\ell(x^n)] \leq H(X) + \varepsilon$.

另一方面，by Fano inequality,

$$\begin{aligned} nR &\geq H(\hat{X}^n) \quad (\because \text{最多有} 2^{nR} \text{个不同序列} \hat{x}^n) \\ &= H(\hat{X}^n) - H(\hat{X}^n | X^n) \\ &= I(X^n; \hat{X}^n) \\ &= H(X^n) - H(X^n | \hat{X}^n) \\ &= nH(X) - H(X^n | \hat{X}^n) \\ &\geq n \left[H(X) - \frac{H_2(P_e)}{n} - P_e \log_2 |\mathcal{X}| \right] \end{aligned} \tag{5.1}$$

(Fano不等式：$H_2(P_e) + P_e \log(|\mathcal{X}| - 1) \geq H(X^n | \hat{X}^n)$)

$\because$ 要求$P_e \to 0$ with n,

$\therefore R \geq H(X) \Longrightarrow \frac{L}{n} \geq \frac{H(X)}{\log_2 D}$.

Theorem 5.3.1 (Fixed-to-fixed source coding theorem, 无扰编码定理). *若$R > H(X)$，则R是可达的；若$R < H(X)$，则R不可达。*

- 编码效率：$\eta = \frac{H(X)}{R} \leq 1$.
- Example: 见中文教材的例3.2.3。

5.4 Variable-length source codes (不等长编码)

- A variable-length source code maps each source letter $x \in \mathcal{X}$ to a codeword $C(x)$ of length $\ell(x)$.
  For example,
  $$\mathcal{X} = \{a, b, c\}, \mathcal{D} = \{0, 1\}$$
  $$C(a) = 0, C(b) = 10, C(c) = 11$$

- The major property that we usually require for any variable-length code is that of *unique decodability*. This means that the input sequence of source letters can be reconstructed unambiguously from the encoded symbol sequence.

- Clearly, unique decodability requires that $C(x) \neq C(x')$ for $x \neq x'$.

- Definition: The extension of a code $\mathcal{C}$ is the code for finite strings of $\mathcal{X}$ given by the concatenation of the individual codewords: $C(x_1, x_2, \ldots, x_n) = C(x_1)C(x_2)\ldots C(x_n)$.
  For example,
  $$\left.\begin{array}{l} C(x_1) = 00 \\ C(x_2) = 11 \end{array}\right\} \Rightarrow C(x_1x_2) = 0011$$

  – A code is called non-singular if
  $$x_i \neq x_j \Rightarrow C(x_i) \neq C(x_j)$$

  – A code is called uniquely decodable if its extension is non-singular.

  For example, $C(a) = 0, C(b) = 10, C(c) = 11$ is prefix-free and uniquely decodable. However, the code $\mathcal{C}'$ defined by
  $$C'(a) = 0, C'(b) = 1, C'(c) = 01$$
  is not uniquely decodable.

5.4.1 Prefix-free codes

Checking whether a code is uniquely decodable can be quite complicated. However, there is a good class of uniquely decodable codes called *prefix-free codes*.

Definition 5.4.1. *A code is said to be prefix-free if no codeword is a prefix of any other codeword.*

Advantages $\left\{\begin{array}{l} \text{easy to check whether a code is prefix-free and therefore uniquely decodable.} \\ \text{can be decoded with no delay (instantaneous code).} \end{array}\right.$

- Any fixed-length code is prefix-free.

- Classes of codes:

X	Uniquely decodable, But not prefix-free	Prefix-free
a	10	0
b	00	10
c	11	110
d	110	111

Figure 5.2: 唯一可译码的分类及编码方式

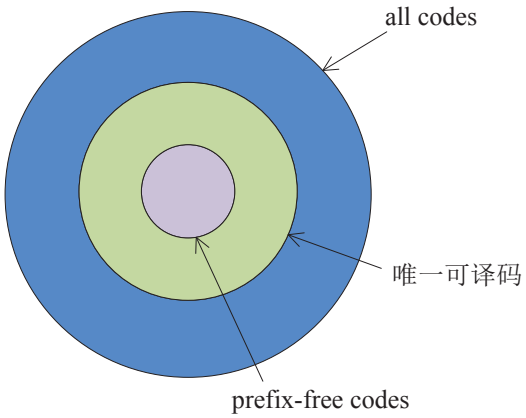


Figure 5.3: 码的分类

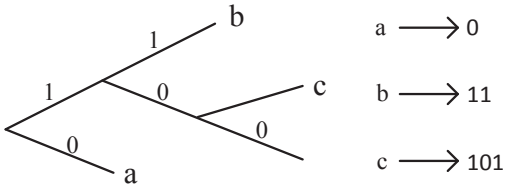


Figure 5.4: The binary code tree

5.4.2 Code tree

- The digits in the codewords are represented as the labels on the branches of a rooted tree.
- For prefix-free codes, each codeword corresponds to a leaf node.
- A prefix-free code will be called full if no new codeword can be added without destroying the prefix-free property.

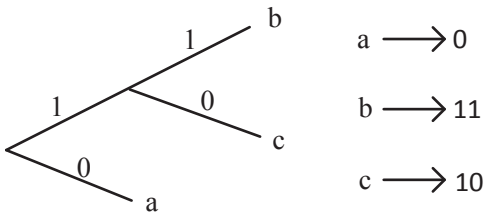


Figure 5.5: full tree

- D-ary tree: A D-ary tree is a finite rooted tree such that D branches stem outward from each (intermediate) node. D branches are labeled with the D different D-ary letters.

- Why the prefix-free condition guarantees unique decodability?

5.5 Kraft Inequality

The Kraft inequality tells us whether it it possible to conduct a prefix-free code for a given source alphabet $\mathcal{X}$ with a given set of codeword length $\{\ell(x)\}$.

Theorem 5.5.1 (Kraft Inequality)**.** *There exists a D-ary prefix-free code with codeword lengths $\ell_1, \ell_2, \ldots, \ell_k$ if and only if*

$$\sum_{i=1}^{k} D^{-\ell_i} \leq 1 \tag{5.2}$$

Every full prefix-free code satisfies (5.1) with equality.

Proof. First assume that $\mathcal{C}$ is a prefix-free code with codeword lengths $\{\ell_1, \ell_2, \ldots, \ell_k\}$. Let $\ell_{max} = \max_i \ell_i$. Consider constructing a D-ary tree for the code $\mathcal{C}$ by pruning the full D-ary tree of length ℓ_{max} at all nodes corresponding to codewords:

a) A codeword at depth ℓ_i has $D^{\ell_{max}-\ell_i}$ descendants(leaves) at depth ℓ_{max}.[Each of these descendant sets must be disjoint]

b) Begin with $i = 1$,we find the node X_i: corresponding to a codeword.

c) We prune the tree to make this node a leaf at depth ℓ_i.

d) By this process, we delete $D^{\ell_{max}-\ell_i}$ leaves from the tree. None of these leaves could have previously been deleted because of the prefix-free condition.

e) But there are only $D^{\ell_{max}}$ leaves that can be deleted. so we have

$$\sum_i D^{\ell_{max}-\ell_i} \leq D^{\ell_{max}}$$

or

$$\sum_i D^{-\ell_i} \leq 1$$

Next, conversely, suppose that we are given the set of codeword lengths $\ell_1, \ell_2, \ldots, \ell_k$ for which (5.1) is satisfied. □

- Without loss of generality, assume that we have ordered these lengths so that $\ell_1 \leq \ell_2 \leq \cdots \leq \ell_k$.
- Consider the following algorithm:
  a) Start with the full D-ary tree of length ℓ_{max},and $i \leftarrow 1$.
  b) Choose x_i as any surviving node at depth ℓ_i(not yet used as a codeword), and remove its descendants from the tree. Stop if there is no such surviving node.
  c) If $i = k$,stop.Otherwise $i = i + 1$ and goto b).

We now show that we can indeed choose x_i in step b) for all $i < k$. Suppose that $x_1, x_2, \ldots, x_{i-1}$ has been chosen. The number of surviving leaves at depth ℓ_{max} not stemming from any codeword is

$$D^{\ell_{max}} - (\sum_{j=1}^{i-1} D^{\ell_{max}-\ell_j}) = D^{\ell_{max}}(1 - \sum_{j=1}^{i-1} D^{-\ell_j}) > 0$$

with condition (3.1). There must be (unused) surviving nodes at depth $\ell_i < \ell_{max}$. Since $\ell_1 \leq \cdots \leq \ell_{i-1} \leq \ell_i$, no already chosen codeword can stem outward from such a surviving node and hence this surviving node may be chosen as x_i.
Example:
Construct a binary prefix-free code with lengths $\ell_1 = \ell_2 = \ell_3 = 2, \ell_4 = 3$ and $\ell_5 = 4$. Since $\sum_{i=1}^{5} 2^{-\ell_i} = \frac{15}{16} < 1$, such a prefix-free code exists.

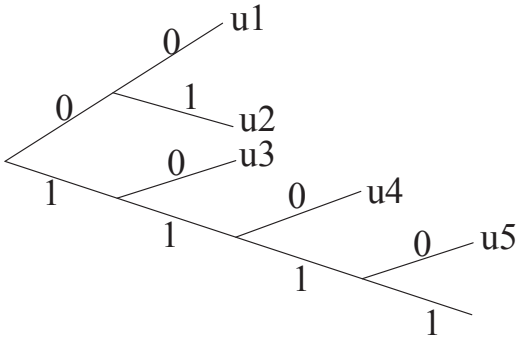


Figure 5.6: a binary prefix-free code tree

- Note: Just because a code has lengths that satisfy (5.1), it does not follow that the code is prefix-free, or even uniquely decodable.
- The same theorem holds for uniquely decodable codes. (这意味着对某一码字长度集合，存在唯一可译码和无前缀码。So why use any code other than a prefix-free code?)

Theorem 5.5.2 (Extended Kraft Inequality)**.** *For any prefix-free code over an alphabet of size D, the codeword length satisfy*

$$\sum_{i=1}^{\infty} D^{-\ell_i} \leq 1$$

Conversely, for any given set of codeword lengths that satisfy the inequality, we can construct a prefix-free code with these lengths.

Proof. Consider a codeword $y_1 y_2 \dots y_{\ell_i}$, where $y_j \in \{0, 1, \dots, D-1\} \triangleq \mathcal{D}$. Let $0.y_1 y_2 \dots y_{\ell_i} = \sum_{j=1}^{\ell_i} y_i D^{-j} \in [0, 1]$. This codeword corresponds to an interval

$$(0.y_1 y_2 \dots y_{\ell_i}, 0.y_1 y_2 \dots y_{\ell_i} + \frac{1}{D^{\ell_i}})$$

Prefix-free code implies the intervals are disjoint. Hence the sum of their lengths ≤ 1. □

5.6 Optimal codes

5.6.1 Problem formulation and Shannon codes

Let $\mathcal{X} = \{a_1, a_2, \dots, a_k\}$ be the source alphabet, and $P_i = P_X(X = a_i) > 0$. Suppose that we encode each source symbol into prefix-free codeword. Denote by $\mathcal{C}(a_i)$ the codeword for a_i and by ℓ_i the length of $\mathcal{C}(a_i)$.

- Optimal code is defined as code with smallest possible $L_(\mathcal{C})$ with respect to P_X.
- We now consider the problem of finding the prefix code with minimum expected length:$L = \sum_{i=1}^{k} P_i \ell_i = \sum_{x \in \mathcal{X}} P(x)\ell(x)$.
- Mathematically, this is a standard optimization problem:

$$\text{Minimize } L = \sum_i P_i \ell_i$$

$$\text{subject to } \sum_i D^{-\ell_i} \leq 1$$

  and $\ell_i = \ell(x)$ are integers.
- We first ignore the integer constraint on ℓ_i. With real variables, we may assume that $\sum_i D^{-\ell_i} = 1$.
- Using a Lagrange multiplier λ, we want to minimize

$$J = \sum_i P_i \ell_i + \lambda(\sum_i D^{-\ell_i} - 1)$$

Setting $\frac{\partial J}{\partial \ell_i} = 0$, we obtain

$$\frac{\partial J}{\partial \ell_i} = P_i - \lambda D^{-\ell_i} \ln D = 0 \Leftarrow (a^x)' = (a^x) \ln a$$

equivalently, $D^{-\ell_i} = \frac{P_i}{\lambda \ln D}$

Since $\sum_i P_i = 1$ and $D^{-\ell_i} = 1$, we have $\lambda = \frac{1}{\ln D}$ and hence

$$P_i = D^{-\ell_i}$$

This yields optimal lengths $\ell_i^* = -\log_D P_i$
The expected codeword length

$$\begin{aligned} L_{min}(non-integer) &= -\sum_i P_i \log_D P_i \\ &= \sum_i P_i \log_2 P_i \diagup \log_2 D \\ &= H(X) \diagup \log_2 D \end{aligned}$$

Theorem 5.6.1 (Entropy bounds for prefix-free codes)**.** *Let L_{min} be the minimum expected codeword length over all D-ary prefix-free code. Then*

$$\frac{H(X)}{\log_2 D} \leq L_{min} < \frac{H(X)}{\log_2 D} + 1$$

Proof. ① Let $\ell_1 \dots \ell_k$ be the codeword lengths of an arbitrary prefix-free code.

$$\begin{aligned} \frac{H(X)}{\log D} - L &= \frac{1}{\log D} \sum_i P_i \log \frac{1}{P_i} - \sum_i P_i \ell_i \\ &= \frac{1}{\log D} \sum_i P_i \log \frac{1}{P_i} - \sum_i P_i \log_D D^{-\ell_i} \\ &= \frac{1}{\log D} [\sum_i P_i \log \frac{D^{-\ell_i}}{P_i}] \\ \text{IT Inequality} &\leq \frac{1}{\log D} [\log_e \sum_i P_i (\frac{D^{-\ell_i}}{P_i} - 1)] \qquad (*) \\ &= \frac{\log e}{\log D} (\sum_i D^{-\ell_i} - \sum_i P_i) \\ \text{Kraft Inequality} &\leq 0 \end{aligned} \tag{5.3}$$

(*) is satisfied with equality iff $\frac{D^{-\ell_i}}{P_i} = 1$; i.e.,$P_i = D^{-\ell_i}$.

② We now show that there exists a prefix-free code with $L(\mathcal{C}) < \frac{H(X)}{\log D} + 1$.

- Let us choose the codeword length to be $\ell_i = \lceil -\log_D P_i \rceil$. Then

$$\begin{aligned} &\underbrace{-\log_D P_i \leq \ell_i}_{\Downarrow} < -\log_D P_i + 1 \\ &D^{-\ell_i} \leq P_i \\ &\sum D^{-\ell_i} \leq \sum P_i = 1 \Rightarrow kraft \text{inequality is satisfied.} \end{aligned} \tag{**}$$

Thus, a prefix-free code exists with the above length.

- From the RHS of (**),

$$L = \sum_i P_i \ell_i < \sum P_i(-\log_D P_i + 1) = \frac{H(X)}{\log_2 D} + 1$$

③ $\therefore \frac{H(X)}{\log_2 D} \leq L_{min} \leq L < \frac{H(X)}{\log_2 D} + 1$

□

Summary [Shannon code]:

- Ideal codeword length $\ell_i = -\log_D P_i$.
  This is optimal when $-\log_D P_i$ is an integer for any i.
- For general distribution, set
$$\ell_i = \lceil -\log_D P_i \rceil$$
- Bounds for the codeword length:
$$-\log_D P_i \leq \ell_i < -\log_D P_i + 1$$
  Average codeword length
$$H_D(X) \leq L < H_D(X) + 1$$
- Example: $P_X(x) = \{\frac{1}{3}, \frac{1}{3}, \frac{1}{4}, \frac{1}{12}\}$.
  Then
$$\begin{aligned} H(X) &= 1.8554 \\ \ell_i &= \lceil -\log_D P_i \rceil = (2,2,2,4) \\ E[\ell(x)] &= \frac{13}{6} = 2.1667 \end{aligned}$$

  Comparing to the obvious codeword length assignment (2,2,2,2,) loss 0.1667 bit per source symbol.

5.6.2 Improvement

Coding over multiple i.i.d. source symbols: View $(X_1, X_2, \ldots, X_n)$ as one super-symbol from $\mathcal{X}^n$.

Apply the bounds derived above,

$$H(X_1 \ldots X_n) \leq E[\ell(X_1^n)] < H(X_1 \ldots X_n) + 1$$

Since $X_1 \ldots X_n$ are i.i.d, $H(X_1^n) = \sum_i H(X_i) = nH(X)$ implies

$$H(X) \leq \frac{1}{n} E[\ell(X_1^n)] < H(X) + \frac{1}{n}$$

Theorem 5.6.2 (Prefix-free source coding theorem)**.** *For any DMS with entropy $H(X)$, there exists a D-ary prefix-free coding of source blocks of length n such that the expected codeword length per source symbol L_n satisfies*

$$\frac{H(X)}{\log D} \leq L_n < \frac{H(X)}{\log D} + \frac{1}{n}$$

From this theorem, $H(X)$ is the minimum expected codeword length per source symbol required to describe the source.

Usually, $R = L_n \log_2 D$[bits/source symbol] is called the rate of a prefix-free code.

5.6.3 Unknown distribution

If assign the codeword length as

$$\ell_i = \lceil -\log q(x) \rceil$$

and the true distribution of X is $P_X(i) = P_i$, then

$$H(P) + D(p||q) \leq E_p[\ell(X)] < H(P) + D(p||q) + 1$$

Proof.

$$\begin{aligned} E_p[\ell(X)] &= \sum_x P(x) \lceil \log \frac{1}{q(x)} \rceil \\ &< \sum_x P(x) [\log \frac{1}{q(x)} + 1] \\ &= \sum_x P(x) \log (\frac{p(x)}{q(x)} \frac{1}{p(x)}) + 1 \\ &= \sum_x P(x) \log \frac{p(x)}{q(x)} + \sum_x P(x) \log \frac{1}{p(x)}) + 1 \\ &= D(p||q) + H(P) + 1 \end{aligned}$$

□

- Penalty of $D(p||q)$ bits per source symbol due to the wrong distribution.
- Discussion
  - For any n, any code over i.i.d sequence X_1^n,$\frac{1}{n} E[\ell(X_1^n)] \geq H(X)$.
  - We can achieve this when $n \to \infty$, AEP code, Shannon code.

$$\lim_{n \to \infty} \frac{1}{n} E[\ell(X_1^n)] = H(X)$$

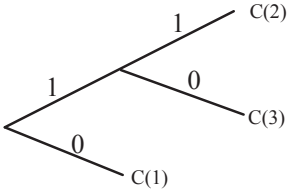
  - True or False: for finite n,
    - Shannon code is "optimal"?
    - A code with codeword length $\ell_i = -\log P_X(i)$,$\forall i$ is optimal
    - Any prefix code must satisfy $\ell_i \geq -\log P_i$, $\forall i$
    - The optimal code must satisfy $\ell_i \leq \lceil -\log P_i \rceil$, $\forall i$

5.7 Huffman codes

The optimal prefix code(in the sense of minimal L) for a given distribution can be constructed by a simple algorithm discovered by Huffman in 1950(as a term paper in Fano's IT class at MIT).

Huffman's trick, in today's jargon, was to "think outside the box". He ignored the kraft inequality, and looked at the binary code tree to establish properties that an optimal prefix-free code should have.

- Example:
  A simple optimal code



P	X	c(X)	ℓ (X)
0.6	1	0	1
0.3	2	11	2
0.1	3	10	2

Figure 5.7: a simple optimal code

Lemma 5.7.1. *Optimal codes have the property that if $\ell_i > \ell_j$, then $P_i \leq P_j$.*

Proof. Let $\mathcal{C}$ be the optimal code. Consider a code $\mathcal{C}'$, with the codewords i and j of $\mathcal{C}$ interchanged. Then

$$L(\mathcal{C}') - L(\mathcal{C}) = \sum_k P_k \ell'_k - \sum_k P_k \ell_k$$
$$= [P_i \ell_j + P_j \ell_i] - [P_i \ell_i + P_j \ell_j] \qquad = (P_j - P_i)(\ell_i - \ell_j)$$

Note that $\ell_i - \ell_j > 0$, and since $\mathcal{C}$ is optimal, $L(\mathcal{C}') - L(\mathcal{C}) \geq 0$. Hence we must have $P_j \geq P_i$. □

Lemma 5.7.2. *Optimal prefix-free codes have the property that the two longest codewords have the same length.*

Proof. Otherwise, one can delete the last bit of the longer one, preserving the prefix-free property and achieving lower codeword length. □

Lemma 5.7.3. *The two longest codewords differ only in the last bit and correspond to the two least likely symbols.*

Proof. By Lemma(5.7.1), the longest codewords must belong to the least probable source symbols.
If there is a maximal length codeword without a sibling, then we can delete the last bit of codeword and still satisfy the prefix-free property. This reduces the average codeword length. □

- The Huffman algorithm chooses an optimal code tree by starting at leaves for the least likely symbols and working in.
  Example 1:

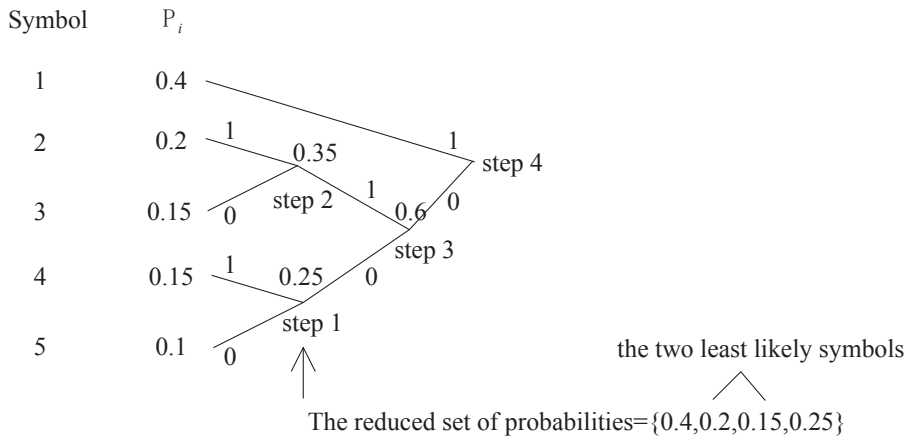


Figure 5.8: Huffman tree

Optimal length:

$$-\log P_i = \{1.32, 2.32, 2.74, 2.74, 3.32\}$$
$$\approx \text{the length}\{1,3,3,3,3\}\text{of the optimal code}$$

$$\begin{cases} H(X) = 2.15\text{bits/symbol} \\ L(\mathcal{C}) = 2.2\text{bits/symbol} \end{cases}$$

Example 2:

Symbol	P_i	codeword
1	0.35	11
2	0.2	01
3	0.2	00
4	0.15	101
5	0.1	100

- Let X be a r.v. over $\mathcal{X}$ with $\bar{P}$, and X' be a r.v. with reduced set of probabilities $\bar{P}'$.

- Let L' be the expected codeword length of code for X'. Then

$$L = L' + (P_k + P_(k-1)) \qquad (\because L = \sum_{i=1}^{k-2} P_i\ell_i + P_{k-1}(\ell'_{k-1}+1) + P_k(\ell'_{k-1}+1))$$

$L_{min} = L'_{min} + P_k + P_{k-1} \Rightarrow$ finding the optimal reduced code, yields the optimal final code.

- Note that there are many optimal codes. The Huffman algorithm produce one such optimal code.

- D-ary Huffman code:$|\mathcal{X}| = (D-1)\theta + D \quad K = |\mathcal{X}|$

$$\begin{aligned} \text{第一次合并的消息个数} &= [(k-2) mod (D-1)] + 2 \\ &= R_{D-1}(k-2) + 2 \end{aligned}$$

5.8 Shannon-Fano-Elias Coding

- Modified CDF:

$$\bar{F}(x) = \sum_{a<x} P(a) + \frac{1}{2}P(x)$$

It is the midpoint of the step corresponding to x.

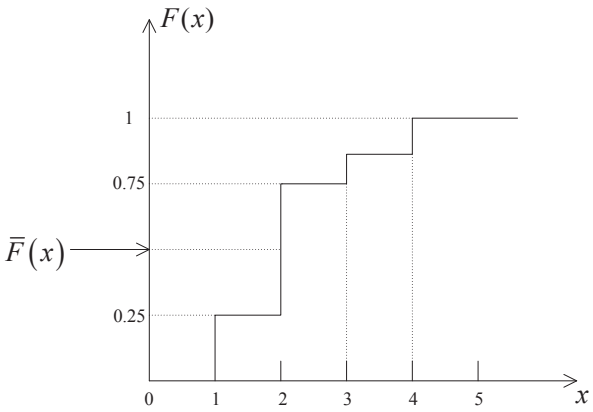


Figure 5.9: $P(x) = \{0.25, 0.5, 0.125, 0.125\}$

- Using the value of $\bar{F}(x)$ as a code for x:
(the first $\ell(x)$ bits of $\bar{F}(x)$) $\bar{F}(x)$ in binary $\rightarrow$ codeword

$$\begin{aligned} \text{then,} \bar{F}(x) - \lfloor \bar{F}(x) \rfloor_{\ell(x)} &< \frac{1}{2^{\ell(x)}} \\ &< \frac{P(x)}{2} \quad (\because \ell(x) = \lceil \log \frac{1}{P(x)} \rceil + 1 > \log_2 \frac{1}{P(x)} + 1) \\ &= \bar{F}(x) - F(X-1) \end{aligned}$$

That is, $\lfloor \bar{F}(x) \rfloor_{\ell(x)}$ lies within the step corresponding to x. Thus $\ell(x)$ bits suffice to describe x.

- Each codeword $z_1 z_2 \ldots z_\ell$ is considered to represent the interval $[0.z_1 z_2 \ldots z_\ell, 0.z_1 z_2 \ldots z_\ell + \frac{1}{2^\ell}]$.$\rightarrow$ prefix-free

- The expected length of this code is

$$L + \sum_x P(x)\ell(x) = \sum_x P(x)[\lceil \log \frac{1}{P(x)} \rceil + 1] < \sum_x [\log \frac{1}{P(x)} + 2] = H(X) + 2$$

- Example:

x	$P(x)$	$\bar{F}(x)$	$\bar{F}(x)$ in binary	$\ell(x) = \lceil \log \frac{1}{P(x)} + 1 \rceil$	$\mathcal{C}(x)$
1	0.25	0.125	0.001	3	001
2	0.5	0.5	0.10	2	10
3	0.125	0.8125	0.1101	4	1101
4	0.125	0.9375	0.1111	4	1111

$$\begin{cases} L(\mathcal{C}) = 2.75 \text{ bits} \\ H(X) = 1.75 \text{ bits} \end{cases}$$

code efficiency $\eta \triangleq \frac{H(X)}{R} = \frac{H(X)}{L} \leq 1$

5.9 Arithmetic Coding

- Calculate the pmf $P(x^n)$ and the CDF $F(x^n)$ for sequence x^n.
- Using Elias coding, we can use a number in $[(F(x^n) - P(x^n)), F(x^n)]$ as the code for x^n.
- Example:对二进制序列0100进行算术编码。
  假定$P_X(0) = P_0 = \frac{3}{4}, P_X(1) = P_1 = \frac{1}{4}$

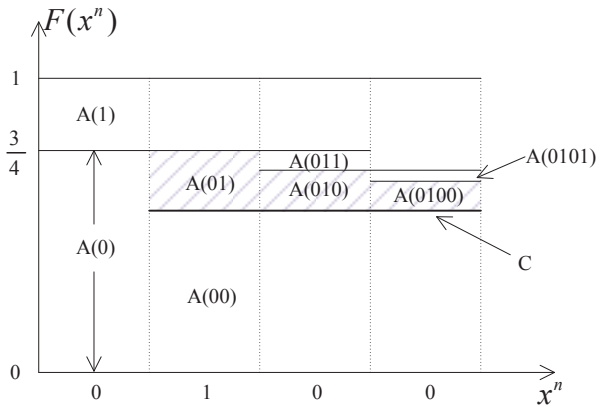


Figure 5.10: 算术编码

– 输入区间$A = [0, 1]$
– 输入第一位“0”后，区间A按概率分割为两个区间：$A(0) = A \cdot P_0, A(1) = [P_0, 1] = AP_1$（长度）并取$A(0)$为下次编码区间，即新的$A + A(0)$。
– 输入第二位“1”后，区间$A = A(0)$分割为：

$$A(00) = A \cdot P_0, A(01) = AP_1$$

并取$A = A(01)$为下次分割区间。
– 可以看出，A区间在不断变小，编码区间底线C要么不变，要么上升。
– 编码过程：
① 初始化：$C = 0, A = 1$
② 输入高概率符号“0”，$C = 0$,$A = AP_0 = \frac{3}{4}$
③ 输入低概率符号“1”，$C = C + AP_0 = \frac{9}{16}$,$A = AP_1 = \frac{3}{16}$

④ 输入高概率符号“0”，$C = \frac{9}{16}$,$A = \frac{3}{16} \times \frac{3}{4} = \frac{9}{64}$
⑤ 输入低概率符号“0”，

$$C = \frac{9}{16} \qquad A = \frac{9}{64} \times \frac{3}{4} = \frac{27}{256} = (\frac{16}{256}) + \frac{11}{256}$$
$$= 0.1001 \qquad = 0.00011011 \quad \text{(binary)}$$

最后码字区间为：$0.1001 \leq \mathcal{C}(x) < 0.10101011$
codeword:101

5.10 Lempel-Ziv universal source coding

已知信源的概率分布，我们能用Huffman算法构造最佳码。对于未知概率分布的源，it is desirable to have a *one-pass* (or online) algorithm to compress the data that “learns” the probability distribution of the data and uses it to compress the incoming symbols.

- Universally optimal: asymptotic compression rate approached the entropy rate of the source for any stationary ergodic source.
- Lempel-Ziv (LZ) algorithms do not require prior knowledge of the source statistics.
- LZ77: sliding window LZ algorithm, it uses string-matching on a sliding window. (它的实现晚一点，典型实现：MS Windows)
- LZ78: tree-structured LZ algorithm, it uses an adaptive dictionary. (典型实现：UNIX的compress)

The key idea of LZ algorithm is to parse the string into phrases and to replace phrases by pointers to where the string has occurred in the past.

- If a match is not found in the window，the next character is sent uncompressed. To distinguish between these two cases, a flag bit is needed.
- Phrase types: (flag f, 匹配位置 u（反向计数）, 匹配长度 n) or (f, c (未压缩字符))
- 然后就可对这些(f, u, n)进行等长编码。

5.10.1 LZ77算法（Gallager’08）

5.10.2 LZ78算法