Introduction to Information Theory

Wireless Information Transmission System Lab.
Institute of Communications Engineering
National Sun Yat-sen University

Table of Contents

- Mathematical Models for Information Sources
- A Logarithmic Measure of Information
  - Average Mutual Information and Entropy
  - Information Measures for Continuous Random Variables
- Coding for Discrete Sources
  - Coding for Discrete Memoryless Sources
  - Discrete Stationary Sources
  - The Lempel-Ziv Algorithm
- Channel Models and Channel Capacity
  - Channel Models
  - Channel Capacity

Introduction

- Two types of sources: analog source and digital source.
- Whether a source is analog or discrete, a digital communication system is designed to transmit information in digital form.
- The output of the source must be converted to a format that can be transmitted digitally.
- This conversion of the source output to a digital form is generally performed by the *source encoder*, whose output may be assumed to be a sequence of binary digits.
- In this chapter, we treat source encoding based on mathematical models of information sources and provide a quantitative measure of the information emitted by a source.

Mathematical Models for Information Sources

- The output of any information source is random.
- The source output is characterized in statistical terms.
- To construct a mathematical model for a discrete source, we assume that each letter in the alphabet $\{x_1, x_2, \cdots, x_L\}$ has a given probability p_k of occurrence.

$$p_k = P(X = x_k), \qquad 1 \le k \le L$$

$$\sum_{k=1}^{L} p_k = 1$$

Mathematical Models for Information Sources

- Two mathematical models of discrete sources:
  - If the output sequence from the source is statistically independent, i.e. the current output letter is statistically independent from all past and future outputs, then the source is said to be *memoryless*. Such a source is called a *discrete memoryless source* (DMS).
  - If the discrete source output is statistically dependent, we may construct a mathematical model based on statistical stationarity. By definition, a discrete source is said to be stationary if the joint probabilities of two sequences of length n, say $a_1, a_2, \cdots, a_n$ and $a_{1+m}, a_{2+m}, \cdots, a_{n+m}$, are identical for all $n \geq 1$ and for all shifts m. In other words, the joint probabilities for any arbitrary length sequence of source outputs are invariant under a shift in the time origin.

A Logarithmic Measure of Information

- Consider two discrete random variables with possible outcomes x_i, i=1,2,⋯, n, and y_i, i=1,2,⋯,m.
- When X and Y are statistically independent, the occurrence of $Y=y_j$ provides no information about the occurrence of $X=x_i$.
- When X and Y are fully dependent such that the occurrence of $Y=y_j$ determines the occurrence of $X=x_i$, , the information content is simply that provided by the event $X=x_i$.
- *Mutual Information* between x_i and y_j: the information content provided by the occurrence of the event $Y=y_j$ about the event $X=x_i$, is defined as:

$$I\left(x_i;y_j\right)=\log\frac{P\left(x_i\mid y_j\right)}{P\left(x_i\right)}$$

A Logarithmic Measure of Information

- The units of $I(x_i,y_j)$ are determined by the base of the logarithm, which is usually selected as either 2 or e.
- When the base of the logarithm is 2, the units of $I(x_i,y_j)$ are *bits*.
- When the base is e, the units of $I(x_i,y_j)$ are called *nats* (natural units).
- The information measured in nats is equal to ln2 times the information measured in bits since:

$$\log_a b = \frac{\log_c b}{\log_c a}$$

$$\ln a = \ln 2 \log_2 a = 0.69315 \log_2 a$$

- When X and Y are statistically independent, $p(x_i|y_j)=p(x_i)$, $I(x_i,y_j)=0$.
- When X and Y are fully dependent, $P(x_i|y_j)=1$, and hence $I(x_i,y_j)=-\log P(x_i)$.

Information of the event $X=x_j$.

A Logarithmic Measure of Information

- *Self-information* of the event X=x_i is defined as $I(x_i)$=-log$P(x_i)$≥0.
- Note that a high-probability event conveys less information than a low-probability event.
- If there is only a single event x with probability $P(x)$=1, then $I(x)$=0.

Information = Amount of Uncertainty

- Example: A discrete information source that emits a binary digit with equal probability.
  - The information content of each output is:

$$I(x_i) = -\log_2 P(x_i) = -\log_2 \frac{1}{2} = 1 \text{ bit}, \qquad x_i = 0,1$$

  - For a block of k binary digits, if the source is memoryless, there are M=2^k possible k-bit blocks. The self-information is:

$$I(x_i') = -\log_2 2^{-k} = k \text{ bits}$$

A Logarithmic Measure of Information

- The information provided by the occurrence of the event $Y=y_j$ about the event $X=x_i$ is identical to the information provided by the occurrence of the event $X=x_i$ about the event $Y=y_j$ since:

$$I(x_i;y_j)=\log\frac{P(x_i\,|\,y_j)}{P(x_i)}=\log\frac{P(x_i\,|\,y_j)P(y_j)}{P(x_i)P(y_j)}=\log\frac{P(x_i,y_j)}{P(x_i)P(y_j)}$$

$$=\log\frac{P(y_j\,|\,x_i)P(x_i)}{P(x_i)P(y_j)}=\log\frac{P(y_j\,|\,x_i)}{P(y_j)}=I(y_j;x_i)$$

- Example : X and Y are binary-valued {0,1} random variables that represent the input and output of a binary channel.
  - The input symbols are equally likely.

A Logarithmic Measure of Information

- Example (cont.):
  - The output symbols depend on the input according to the conditional probability:

$$P(Y=0\,|\,X=0)=1-p_0 \qquad P(Y=1\,|\,X=0)=p_0$$

$$P(Y=1\,|\,X=1)=1-p_1 \qquad P(Y=0\,|\,X=1)=p_1$$

  - Mutual information about *X*=0 and *X*=1, given that *Y*=0:

$$P(Y=0)=P(Y=0\,|\,X=0)P(X=0)$$

$$+P(Y=0\,|\,X=1)P(X=1)=\frac{1}{2}(1-p_0+p_1)$$

$$P(Y=1)=P(Y=1\,|\,X=0)P(X=0)$$

$$+P(Y=1\,|\,X=1)P(X=1)=\frac{1}{2}(1-p_1+p_0)$$

A Logarithmic Measure of Information

- Example (cont.)

  The mutual information about *X*=0 given that *Y*=0 is:

$$I\left(x_1;y_1\right)=I\left(0;0\right)=\log_2\frac{P\left(Y=0\mid X=0\right)}{P\left(Y=0\right)}=\log_2\frac{2\left(1-p_0\right)}{1-p_0+p_1}$$

  The mutual information about *X*=1 given that *Y*=0 is:

$$I\left(\mathrm{x}_2;y_1\right)\equiv I\left(1;0\right)=\log_2\frac{2p_1}{1-p_0+p_1}$$

  - If the channel is *noiseless*, p_0=p_1=0:

$$I\left(0;0\right)=\log_2 2=1\ \text{ bit}$$

  - If the channel is *useless*, p_0=p_1=0.5:

$$I\left(0;0\right)=\log_2 1=0\ \text{ bit}$$

A Logarithmic Measure of Information

- *Conditional self-information* is defined as:

$$I\left(x_i \mid y_j\right) = \log \frac{1}{P\left(x_i \mid y_j\right)} = -\log P\left(x_i \mid y_j\right) \geq 0$$

$$I\left(x_i ; y_j\right) = \log P\left(x_i \mid y_j\right) - \log P\left(x_i\right) = I\left(x_i\right) - I\left(x_i \mid y_j\right)$$

- We interpret $I(x_i|y_j)$ as the self-information about the event $X=x_i$ after having observed the event $Y=y_j$.
- The mutual information between a pair of events can be either <u>positive or negative, or zero</u> since both $I(x_i|y_j)$ and $I(x_i)$ are greater than or equal to zero.

Average Mutual Information and Entropy

- *Average mutual information* between X and Y:

$$I(X;Y)=\sum_{i=1}^{n}\sum_{j=1}^{m}P(x_i,y_j)I(x_i;y_j)$$

$$=\sum_{i=1}^{n}\sum_{j=1}^{m}P(x_i,y_j)\log\frac{P(x_i,y_j)}{P(x_i)P(y_j)}$$

  - $I(X;Y)=0$ when X and Y are statistically independent.
  - $I(X;Y)\geq 0$.
- *Average self-information* $H(X)$:

$$H(X)=\sum_{i=1}^{n}P(x_i)I(x_i)$$

  - When X represents the alphabet of possible output letters from a source, $H(X)$ represents the average self-information per source letter, and it is called the *entropy*.

Average Mutual Information and Entropy

- In the special case, in which the letter from the source are equally probable, $P(x_i)=1/n$, we have:

$$H(X) = -\sum_{i=1}^{n} \frac{1}{n} \log \frac{1}{n} = \log n$$

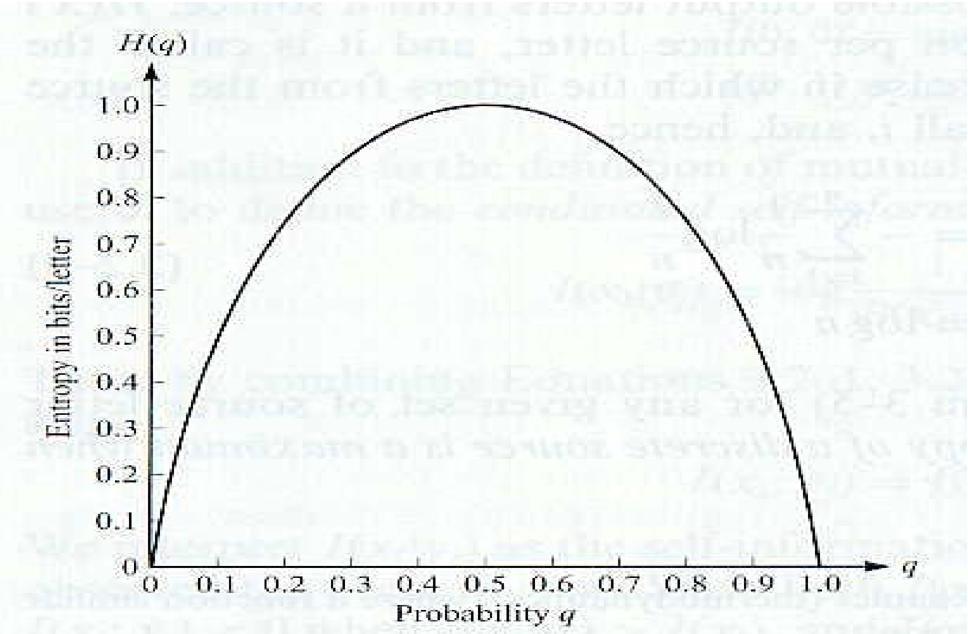
- In general, $H(X) \le \log n$ for any given set of source letter probabilities.
- In other words, the entropy of a discrete source is a maximum when the output letters are equally probable.

Average Mutual Information and Entropy

- Example : Consider a source that emits a sequence of statistically independent letters, where each output letter is either 0 with probability *q* or 1 with probability 1-*q*.
  - The entropy of this source is:

$$H(X) \equiv H(q) = -q \log q - (1-q) \log (1-q)$$

  - Maximum value of the entropy function occurs at *q*=0.5 where *H*(0.5)=1.



Average Mutual Information and Entropy

- The *average conditional self-information* is called the *conditional entropy* and is defined as:

$$H\left(X\,|\,Y\right)=\sum_{i=1}^{n}\sum_{j=1}^{m}P\left(x_i,y_j\right)\log\frac{1}{P\left(x_i\,|\,y_j\right)}$$

- $H(X|Y)$ is the information or uncertainty in X after Y is observed.

$$I\left(X;Y\right)=\sum_{i=1}^{n}\sum_{j=1}^{m}P\left(x_i,y_j\right)\log\frac{P\left(x_i,y_j\right)}{P\left(x_i\right)P\left(y_j\right)}$$

$$=\sum_{i=1}^{n}\sum_{j=1}^{m}P\left(x_i,y_j\right)\left[\log\left\{P\left(x_i\,|\,y_j\right)P\left(y_j\right)\right\}-\log P\left(x_i\right)-\log P\left(y_j\right)\right]$$

$$=\sum_{i=1}^{n}\sum_{j=1}^{m}P\left(x_i,y_j\right)\left[\log P\left(x_i\,|\,y_j\right)-\log P\left(x_i\right)\right]$$

$$=-\sum_{i=1}^{n}P\left(x_i\right)\log P\left(x_i\right)+\sum_{i=1}^{n}\sum_{j=1}^{m}P\left(x_i,y_j\right)\left[\log P\left(x_i\,|\,y_j\right)\right]$$

$$=H\left(X\right)-H\left(X\,|\,Y\right)$$

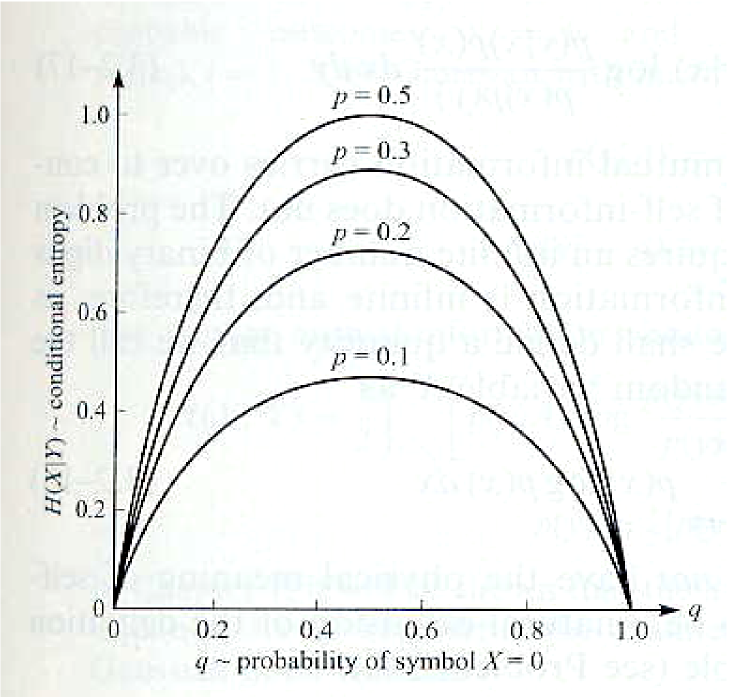
Average Mutual Information and Entropy

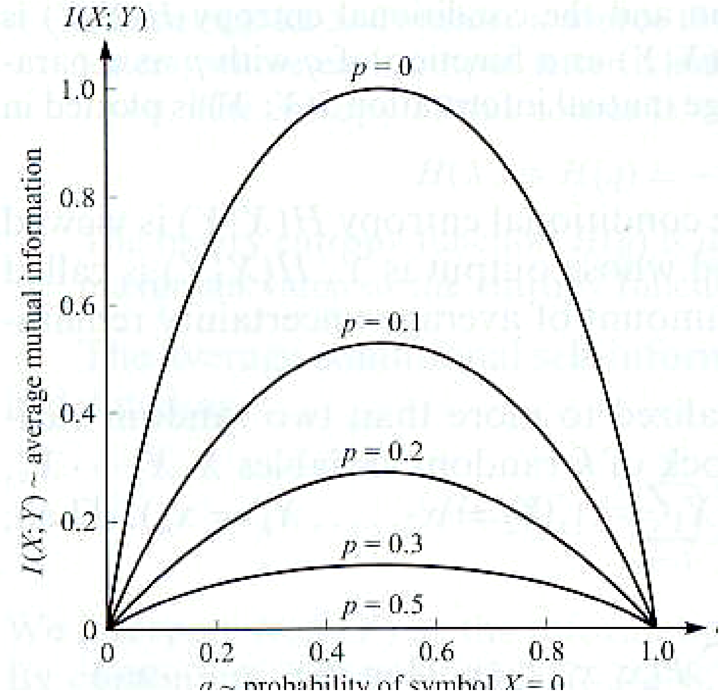
- Since $I(X;Y) \geq 0$, it follows that $H(X) \geq H(X|Y)$, with equality if and only if X and Y are statistically independent.
- $H(X|Y)$ can be interpreted as the average amount of (conditional self-information) uncertainty in X after we observe Y.
- $H(X)$ can be interpreted as the average amount of uncertainty (self-information) prior to the observation.
- $I(X;Y)$ is the average amount of (mutual information) uncertainty provided about the set X by the observation of the set Y.
- Since $H(X) \geq H(X|Y)$, it is clear that conditioning on the observation Y does not increase the entropy.

Average Mutual Information and Entropy

- Example: Consider the case of $p_0=p_1=p$. Let $P(X=0)=q$ and $P(X=1)=1\text{-}q$.
  - The entropy is:

$$H(X) \equiv H(q) = -q\log q - (1-q)\log(1-q)$$





Average Mutual Information and Entropy

- As in the proceeding example, when the conditional entropy $H(X|Y)$ is viewed in terms of a channel whose input is X and whose output is Y, $H(X|Y)$ is called the *equivocation* and is interpreted as the amount of average uncertainty remaining in X after observation of Y.

用模稜兩可的話;含糊其辭

Average Mutual Information and Entropy

- Entropy for two or more random variables:

$$H\left(X_1X_2\ldots X_K\right)=-\sum_{j_1=1}^{n_1}\sum_{j_2=1}^{n_2}\cdots\sum_{j_k=1}^{n_k}P\left(x_{j_1}x_{j_2}\ldots x_{j_k}\right)\log P\left(x_{j_1}x_{j_2}\ldots x_{j_k}\right)$$

since $P\left(x_1x_2\ldots x_k\right)=P\left(x_1\right)P\left(x_2\mid x_1\right)P\left(x_3\mid x_1x_2\right)\ldots P\left(x_k\mid x_1x_2\ldots x_{k-1}\right)$

$$\begin{aligned}H\left(X_1X_2X_3\ldots X_k\right)&=H\left(X_1\right)+H\left(X_2\mid X_1\right)+H\left(X_3\mid X_1X_2\right)\\&\quad+\ldots+H\left(X_k\mid X_1X_2\ldots X_{k-1}\right)\\&=\sum_{i=1}^{k}H\left(X_i\mid X_1X_2\ldots X_{i-1}\right)\qquad(1)\end{aligned}$$

Since $H\left(X\right)\geq H\left(X\mid Y\right)\ \Rightarrow\ H\left(X_1X_2\ldots X_k\right)\leq\sum_{m=1}^{k}H\left(X_m\right)$

where $X=X_m$ and $Y=X_1X_2\ldots X_{m-1}$

Information Measures for Continuous Random Variables

- If X and Y are random variables with joint PDF $p(x,y)$ and marginal PDFs $p(\mathrm{x})$ and $p(y)$, the average mutual information between X and Y is defined as:

$$I(X;Y)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} p(x)p(y\,|\,x)\ \log\frac{p(y\,|\,x)\,p(x)}{p(x)\,p(y)}\,dx\,dy$$

- Although the definition of the average mutual information carriers over to continuous random variables, the concept of self-information does not.
- The problem is that a continuous random variable requires an infinite number of binary digits to represent it exactly. Hence, its self-information is infinite and, therefore, its entropy is also infinite.

Information Measures for Continuous Random Variables

- *Differential entropy* of the continuous random variables X is defined as:

$$H(X) = -\int_{-\infty}^{\infty} p(x) \log p(x) dx$$

Note that this quantity does not have the physical meaning of self-information, although it may appear to be a natural extension of the definition of entropy for a discrete random variable.

- *Average conditional entropy* of X given Y is defined as:

$$H(X|Y) = -\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} p(x,y) \log p(x|y) dxdy$$

- *Average mutual information* may be expressed as:

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

Information Measures for Continuous Random Variables

- Suppose X is discrete and has possible outcomes x_i, $i=1,2,\cdots,n$, and Y is continuous and is described by its marginal PDF $p(y)$.
  - When X and Y are statistically dependent, we may express $p(y)$ as:

$$p(y)=\sum_{i=1}^{n} p(y\,|\,x_i)P(x_i)$$

  - The *mutual information* provided about the event $X= x_i$ by the occurrence of the event $Y=y$ is:

$$I(x_i;y)=\frac{p(y\,|\,x_i)P(x_i)}{p(y)P(x_i)}=\log\frac{p(y\,|\,x_i)}{p(y)}$$

  - The *average mutual information* between X and Y is:

$$I(X;Y)=\sum_{i=1}^{n}\int_{-\infty}^{\infty} p(y\,|\,x_i)P(x_i)\log\frac{p(y\,|\,x_i)}{p(y)}\,dy$$

Information Measures for Continuous Random Variables

- Example: Let X be a discrete random variable with two equally probable outcomes $x_1=A$ and $x_2=-A$.
  - Let the conditional PDFs $p(y|x_i)$, $i=1,2$, be Gaussian with mean x_i and variance σ^2.

$$p(y\,|\,A)=\frac{1}{\sqrt{2\pi}\sigma}e^{-(y-A)^2/2\sigma^2} \qquad p(y\,|\,-A)=\frac{1}{\sqrt{2\pi}\sigma}e^{-(y+A)^2/2\sigma^2}$$

  - The average mutual information is:

$$I(X;Y)=\frac{1}{2}\int_{-\infty}^{\infty}\left[p(y\,|\,A)\log\frac{p(y\,|\,A)}{p(y)}+p(y\,|\,-A)\log\frac{p(y\,|\,-A)}{p(y)}\right]dy$$

where $\quad p(y)=\frac{1}{2}\left[p(y\,|\,A)+p(y\,|\,-A)\right]$

Coding for Discrete Memoryless Sources

- Consider the process of encoding the output of a source, i.e., the process of representing the source output by a sequence of binary digits.
- A measure of the efficiency of a source-encoding method can be obtained by comparing the average number of binary digits per output letter from the source to the entropy $H(X)$.
- The *discrete memoryless source* (DMS) is by far the simplest model that can be devised for a physical model. Few physical sources closely fit this idealized mathematical model.
- It is always more efficient to encode blocks of symbols instead of encoding each symbol separately.
- By making the block size sufficiently large, the average number of binary digits per output letter from the source can be made arbitrarily close to the entropy of the source.

Coding for Discrete Memoryless Sources

- Suppose that a DMS produces an output letter or symbol every τ_s seconds.
- Each symbol is selected from a finite alphabet of symbols x_i, $i=1,2,\cdots,L$, occurring with probabilities $P(x_i)$, $i=1,2,\cdots,L$.
- The entropy of the DMS in bits per source symbol is:

$$H(X) = -\sum_{i=1}^{L} P(x_i)\log_2 P(x_i) \le \log_2 L$$

- The equality holds when the symbols are equally probable.
- The average number of bits per source symbol is $H(X)$.
- The source rate in bits/s is defined as $H(X)/\ \tau_s$.

Coding for Discrete Memoryless Sources

- Fixed-length code words
  - Consider a block encoding scheme that assigns a unique set of *R* binary digits to each symbol.
  - Since there are *L* possible symbols, the number of binary digits per symbol required for unique encoding is:

$$R = \log_2 L \qquad \text{when } L \text{ is a power of 2.}$$

$$R = \lfloor \log_2 L \rfloor + 1 \quad \text{when } L \text{ is not a power of 2.}$$

$\lfloor x \rfloor$ denotes the largest integer less than *x*.

  - The code rate *R* in bits per symbol is *R*.
  - Since $H(X) \leq \log_2 L$, it follows that $R \geq H(X)$.

Coding for Discrete Memoryless Sources

- Fixed-length code words
  - The *efficiency* of the encoding for the DMS is defined as the ratio $H(X)/R \leq 1$.
  - When L is a power of 2 and the source letters are equally probable, $R=H(X)$.
  - If L is not a power of 2 , but the source symbols are equally probable, R differs from $H(X)$ by at most 1 bit per symbol.
  - When $\log_2 L >> 1$, the efficiency of this encoding scheme is high.
  - When L is small, the efficiency of the fixed-length code can be increased by encoding a sequence of J symbols at a time.
  - To achieve this, we need L^J unique code words.

Coding for Discrete Memoryless Sources

- Fixed-length code words

$$\boxed{2^N \geq L^J}$$

  - By using sequences of N binary digits, we have 2^N possible code words. $N \geq J \log_2 L$. The minimum integer value of N required is $N = \lfloor J \log_2 L \rfloor + 1$.
  - The average number of bits per source symbol is $N/J=R$ and the inefficiency has been reduced by approximately a factor of $1/J$ relative to the symbol-by-symbol encoding.
  - By making J sufficiently large, the *efficiency* of the encoding procedure, measured by the ratio $H(X)/R=JH(X)/N$, can be made as close to unity as desired.
  - The above mentioned methods introduce no distortion since the encoding of source symbols or block of symbols into code words is unique. This is called *noiseless*.

Coding for Discrete Memoryless Sources

- *Block coding failure* (or distortion), with probability of P_e, occurs when the encoding process is not unique.
- Source coding theorem I: (by Shannon)
  - Let X be the ensemble of letters from a DMS with finite entropy $H(X)$.
  - Blocks of J symbols from the source are encoded into code words of length N from a binary alphabet.
  - For anyε>0, the probability P_e of a block decoding failure can be made arbitrarily small if J is sufficiently large and

$$R \equiv \frac{N}{J} \geq H\left(X\right) + \varepsilon$$

  - Conversely, if $R \leq H(X) - \varepsilon$, P_e becomes arbitrarily close to 1 as J is sufficiently large.

Coding for Discrete Memoryless Sources

- Variable-length code words
  - When the source symbols are not equally probable, a more efficient encoding method is to use variable-length code words.
  - In the Morse code, the letters that occur more frequently are assigned short code words and those that occur infrequently are assigned long code words.
  - *Entropy coding* devises a method for selecting and assigning the code words to source letters.

Coding for Discrete Memoryless Sources

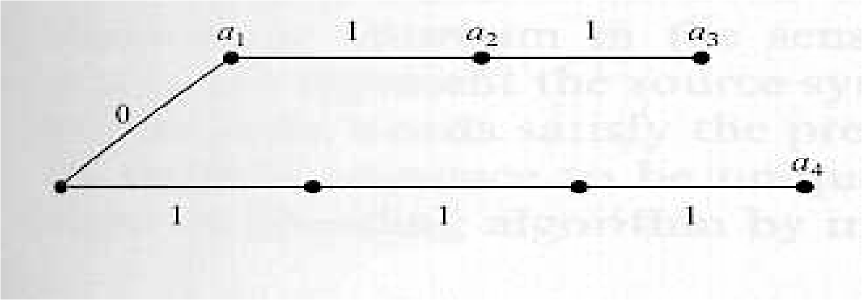
- Variable-length code words

Letter	$P(a_k)$	Code I	Code II	Code III
a_1	1/2	1	0	0
a_2	1/4	00	10	01
a_3	1/8	01	110	011
a_4	1/8	10	111	111

  - Code I is not uniquely decodable. (Eg: 1001:1,00,1;10,01)
  - Code II is *uniquely decodable* and *instantaneously decodable*.
    - Digit 0 indicates the end of a code word and no code word is longer than three binary digits.
    - *Prefix condition*: no code word of length $l<k$ that is identical to the first l binary digits of another code word of length $k>l$.

Coding for Discrete Memoryless Sources

- Variable-length code words
  - Code III has a tree structure:



    - The code is uniquely decodable.
    - The code is not instantaneously decodable.
    - This code does not satisfy the prefix condition.
  - Objective: devise a systematic procedure for constructing uniquely decodable variable-length codes that minimizes:

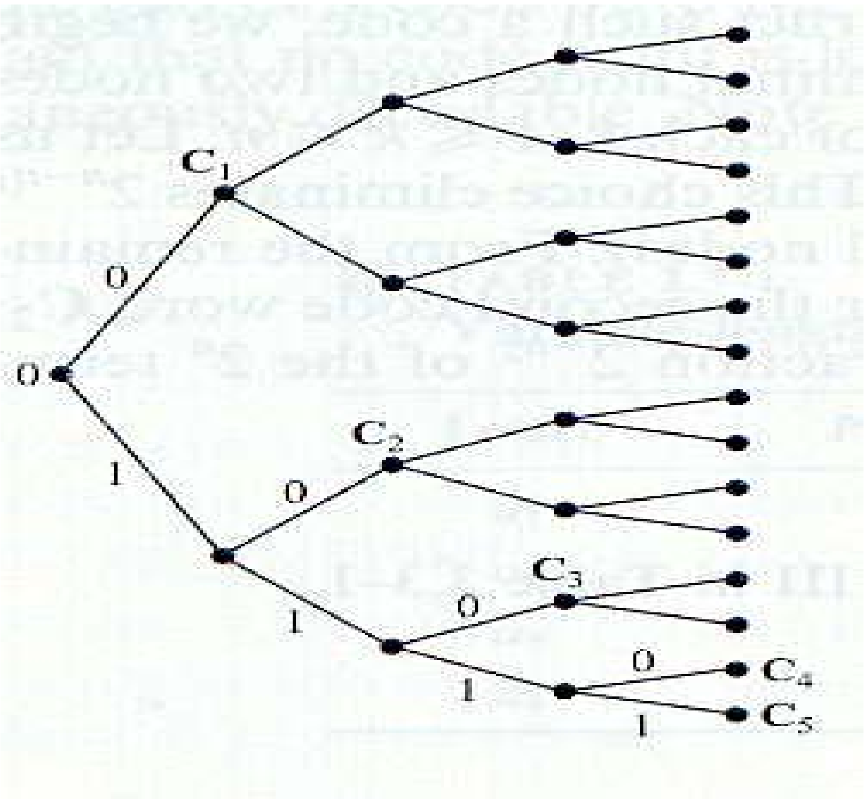
$$\overline{R} = \sum_{k=1}^{L} n_k P(a_k)$$

Coding for Discrete Memoryless Sources

- Kraft inequality
  - A necessary and sufficient condition for the existence of a binary code with code words having lengths $n_1 \le n_2 \le \cdots \le n_L$ that satisfy the prefix condition is

$$\sum_{k=1}^{L} 2^{-n_k} \le 1$$

  - Proof of sufficient condition:
    - Consider a code tree that is embedded in the full tree of 2^n ($n = n_L$) nodes.



Coding for Discrete Memoryless Sources

- Kraft inequality
  - Proof of sufficient condition (cont.)
    - Let's select any node of order n_1 as the first code word $\mathbf{C}_1$. This choice eliminates 2^{n-n_1} terminal nodes (or the fraction 2^{-n_1} of the 2^n terminal nodes).
    - From the remaining available nodes of order n_2, we select one node for the second code word $\mathbf{C}_2$. This choice eliminates 2^{n-n_2} terminal nodes.
    - This process continues until the last code word is assigned at terminal node L.
    - At the node $j<L$, the fraction of the number of terminal nodes eliminated is:

$$\sum_{k=1}^{j} 2^{-n_k} < \sum_{k=1}^{L} 2^{-n_k} \leq 1$$

    - At node $j<L$, there is always a node $k>j$ available to be assigned to the next code word. Thus, we have constructed a code tree that is embedded in the full tree. Q.E.D.

Coding for Discrete Memoryless Sources

- Kraft inequality
  - Proof of necessary condition
    - In code tree of order $n=n_L$, the number of terminal nodes eliminated from the total number of 2^n terminal nodes is:

$$\sum_{k=1}^{L} 2^{n-n_k} \le 2^n \quad \Rightarrow \quad \sum_{k=1}^{L} 2^{-n_k} \le 1$$

- Source coding theorem II
  - Let X be the ensemble of letters from a DMS with finite entropy $H(X)$ and output letters x_k, $1 \le k \le L$, with corresponding probabilities of occurrence p_k, $1 \le k \le L$. It is possible to construct a code that satisfies the prefix condition and has an average length $\bar{R}$ that satisfies the inequalities:

$$H(X) \le \bar{R} < H(X) + 1$$

Coding for Discrete Memoryless Sources

- Source coding theorem II (cont.)
  - Proof of lower bound:

$$H(X)-\overline{R}=\sum_{k=1}^{L}p_k\log_2\frac{1}{p_k}-\sum_{k=1}^{L}p_k n_k=\sum_{k=1}^{L}p_k\log_2\frac{2^{-n_k}}{p_k}$$

$$=\sum_{k=1}^{L}p_k\left(\ln\frac{2^{-n_k}}{p_k}\Big/\ln 2\right)=\sum_{k=1}^{L}p_k\left(\ln\frac{2^{-n_k}}{p_k}\cdot\log_2 e\right)$$

since $\ln x\le x-1$, we have:

Kraft inequality.

$$H(X)-\overline{R}\le(\log_2 e)\sum_{k=1}^{L}p_k\left(\frac{2^{-n_k}}{p_k}-1\right)\le(\log_2 e)\left(\sum_{k=1}^{L}2^{-n_k}-1\right)\le 0$$

Equality holds if and only if $p_k=2^{-n_k}$ for $1\le k\le L$.

Note that $\log_a{}^b=\log b/\log a$.

Coding for Discrete Memoryless Sources

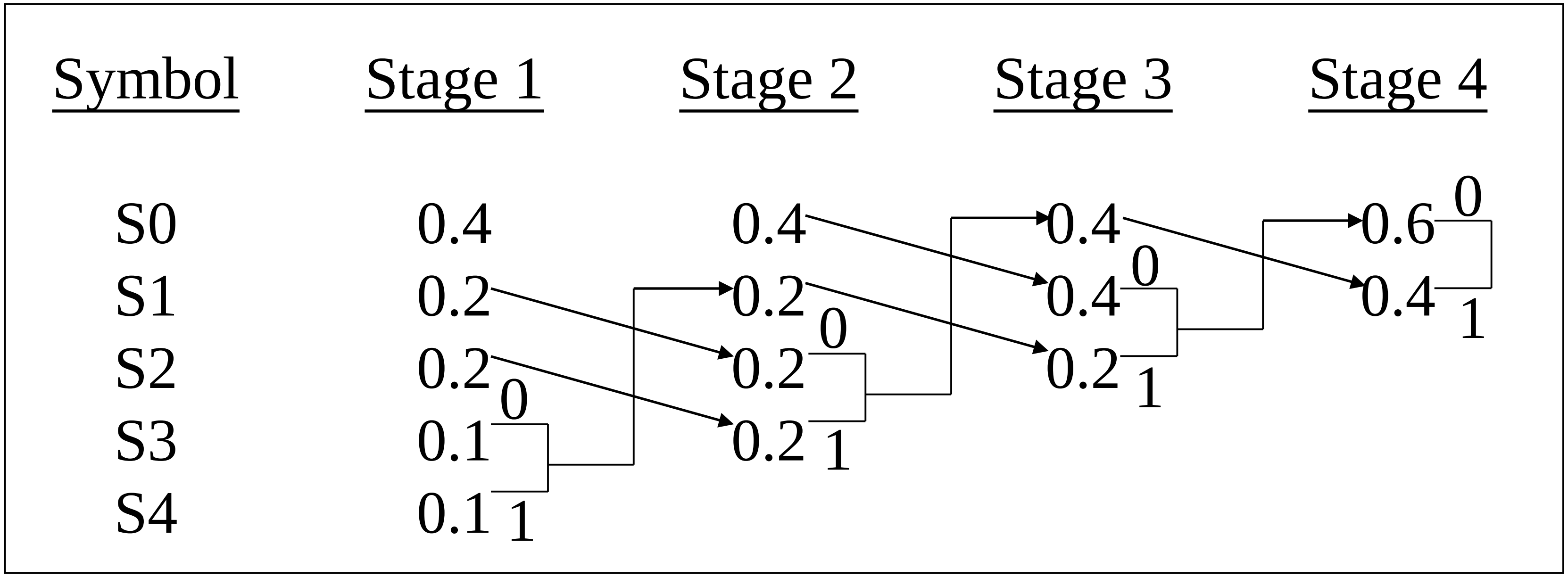
- Source coding theorem II (cont.)
  - Proof of upper bound:
    - The upper bound may be established under the constraint that n_k, $1\leq k \leq L$, are integers, by selecting the $\{n_k\}$ such that $2^{-n_k} \leq p_k < 2^{-n_k+1}$.
    - If the terms $p_k \geq 2^{-n_k}$ are summed over $1\leq k \leq L$, we obtain the Kraft inequality, for which we have demonstrated that there exists a code that satisfies the prefix condition.
    - On the other hand, if we take the logarithm of $p_k < 2^{-n_k+1}$, we obtain $\log_2 p_k < -n_k + 1$ or $n_k < 1 - \log_2 p_k$.
    - If we multiply both sides by p_k and sum over $1\leq k \leq L$, we obtain the desired upper bound.

Coding for Discrete Memoryless Sources

- Huffman coding algorithm
  1. The source symbols are listed in order of decreasing probability. The two source symbols of lowest probability are assigned a 0 and a 1.
  2. These two source symbols are regarded as being combined into a new source symbol with probability equal to the sum of the two original probabilities. The probability of the new symbol is placed in the list in accordance with its value.
  3. The procedure is repeated until we are left with a final list of source statistics of only two for which a 0 and a 1 are assigned.
  4. The code for each (original) source symbol is found by working backward and tracing the sequence of 0s and 1s assigned to that symbol as well as its successors.

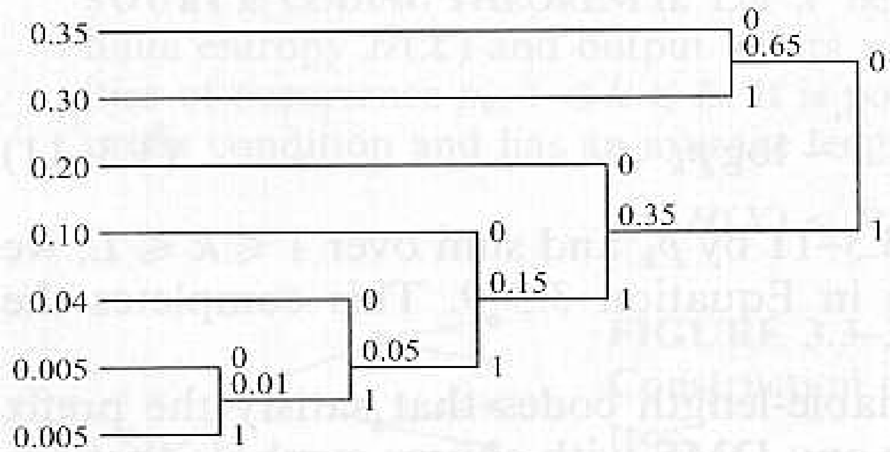
Coding for Discrete Memoryless Sources

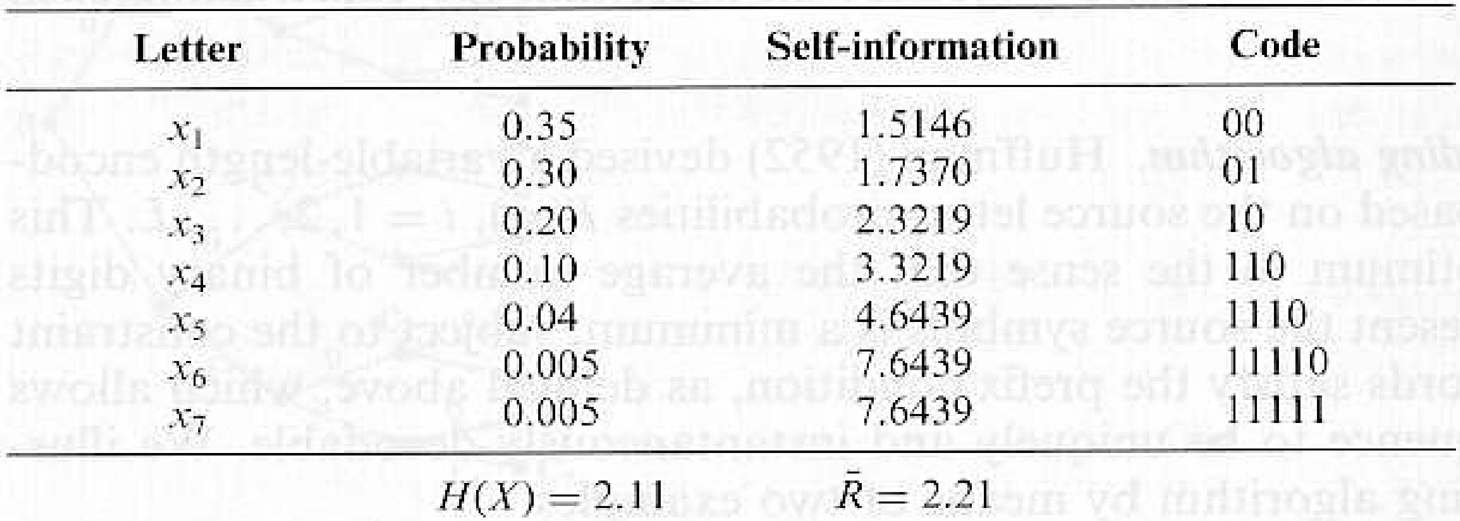
Symbol	Probability	Code Word
S0	0.4	00
S1	0.2	10
S2	0.2	11
S3	0.1	010
S4	0.1	011



Coding for Discrete Memoryless Sources

- Huffman coding algorithm
  - Example



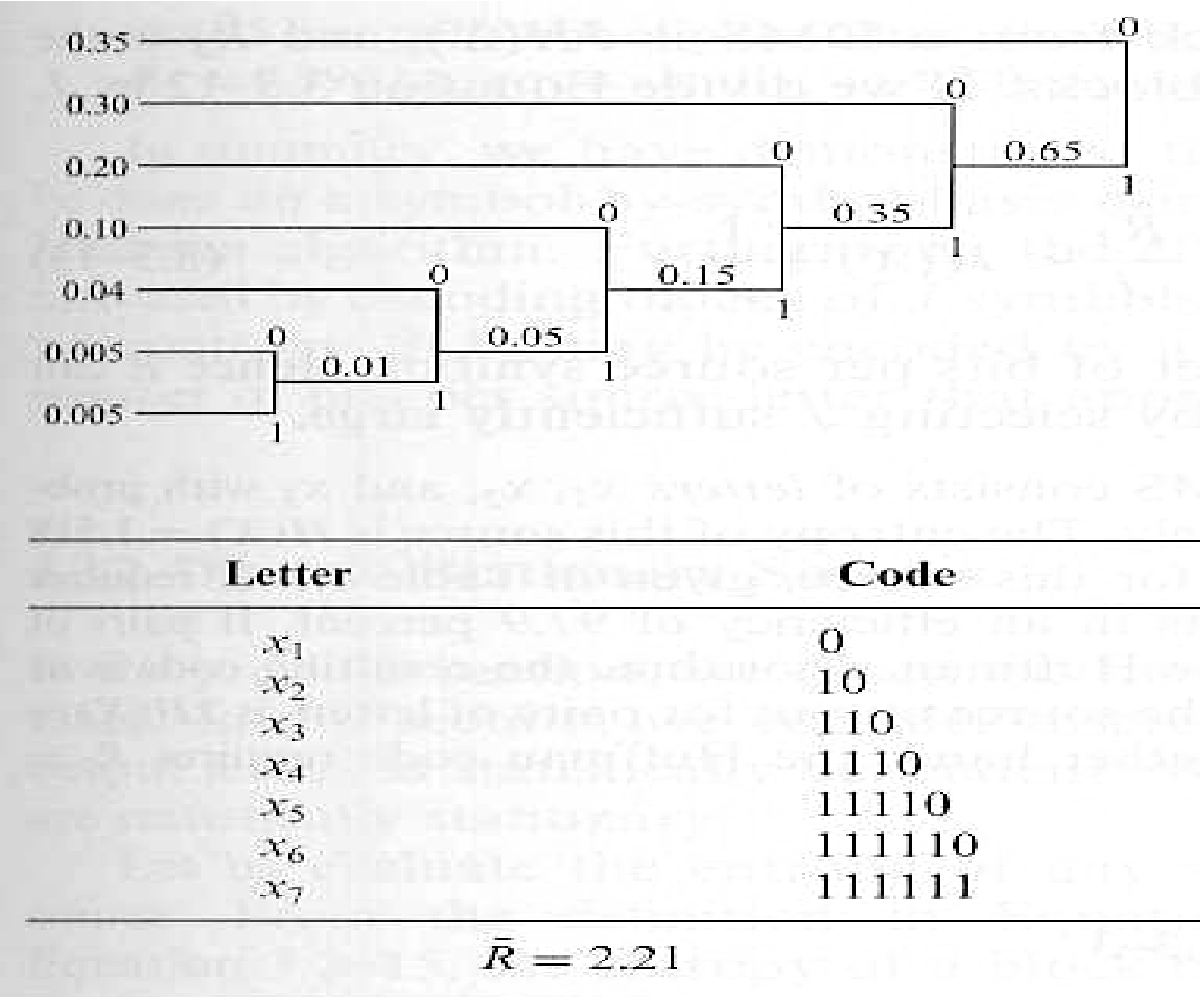


Letter	Probability	Self-information	Code
x_1	0.35	1.5146	00
x_2	0.30	1.7370	01
x_3	0.20	2.3219	10
x_4	0.10	3.3219	110
x_5	0.04	4.6439	1110
x_6	0.005	7.6439	11110
x_7	0.005	7.6439	11111

$H(X) = 2.11$ $\bar{R} = 2.21$

Coding for Discrete Memoryless Sources

- Huffman coding algorithm

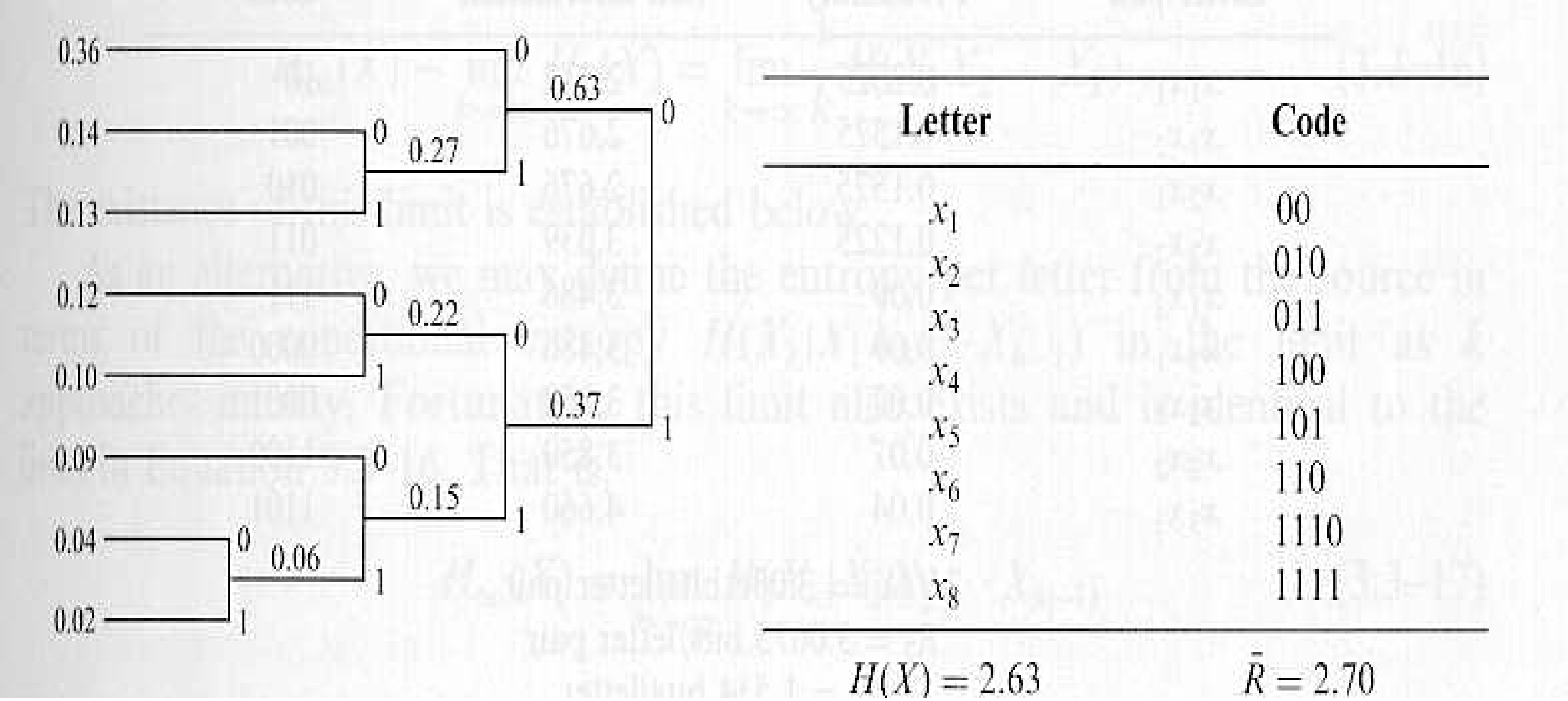


Letter	Code
x_1	0
x_2	10
x_3	110
x_4	1110
x_5	11110
x_6	111110
x_7	111111

$$\bar{R} = 2.21$$

Coding for Discrete Memoryless Sources

- Huffman coding algorithm
  - Example



Letter	Code
x_1	00
x_2	010
x_3	011
x_4	100
x_5	101
x_6	110
x_7	1110
x_8	1111

$H(X) = 2.63$ $\bar{R} = 2.70$

Coding for Discrete Memoryless Sources

- Huffman coding algorithm
  - This algorithm is optimum in the sense that the average number of binary digits required to represent the source symbols is a minimum, subject to the constraint that the code words satisfy the prefix condition, which allows the received sequence to be uniquely and instantaneously decodable.
  - Huffman encoding process is not unique.
  - Code words for different Huffman encoding process can have different lengths. However, the average code-word length is the same.
  - When a combined symbol is moved as high as possible, the resulting Huffman code has a significantly smaller variance than when it is moved as low as possible.

Coding for Discrete Memoryless Sources

- Huffman coding algorithm
  - The variable-length encoding (Huffman) algorithm described in the above mentioned examples generates a prefix code having an $\overline{R}$ that satisfies:

$$H(X) \le \overline{R} < H(X) + 1$$

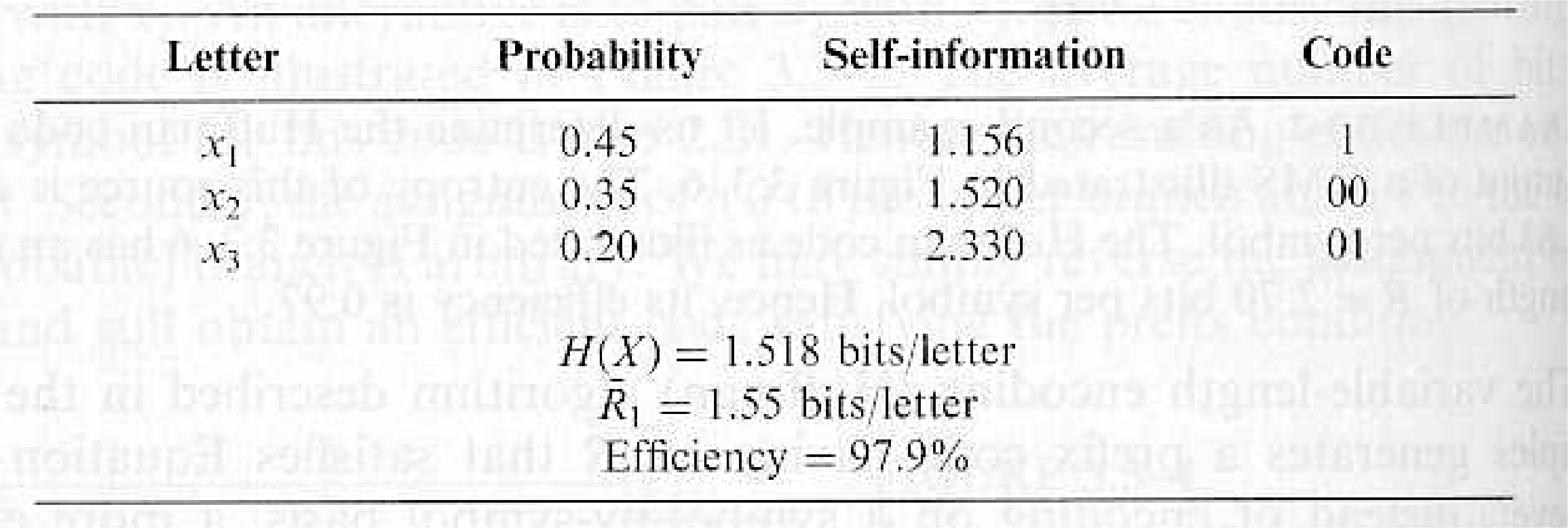
  - A more efficient procedure is to encode blocks of *J* symbols at a time. In such a case, the bounds of source coding theorem II become:

$$JH(X) \le \overline{R}_J < JH(X) + 1 \quad \Rightarrow \quad H(X) \le \frac{\overline{R}_J}{J} \equiv \overline{R} < H(X) + \frac{1}{J}$$

  - $\overline{R}$ can be made as close to *H*(*X*) as desired by selecting *J* sufficiently large.
  - To design a Huffman code for a DMS, we need to know the probabilities of occurrence of all the source letters.

Coding for Discrete Memoryless Sources

- Huffman coding algorithm
  - Example

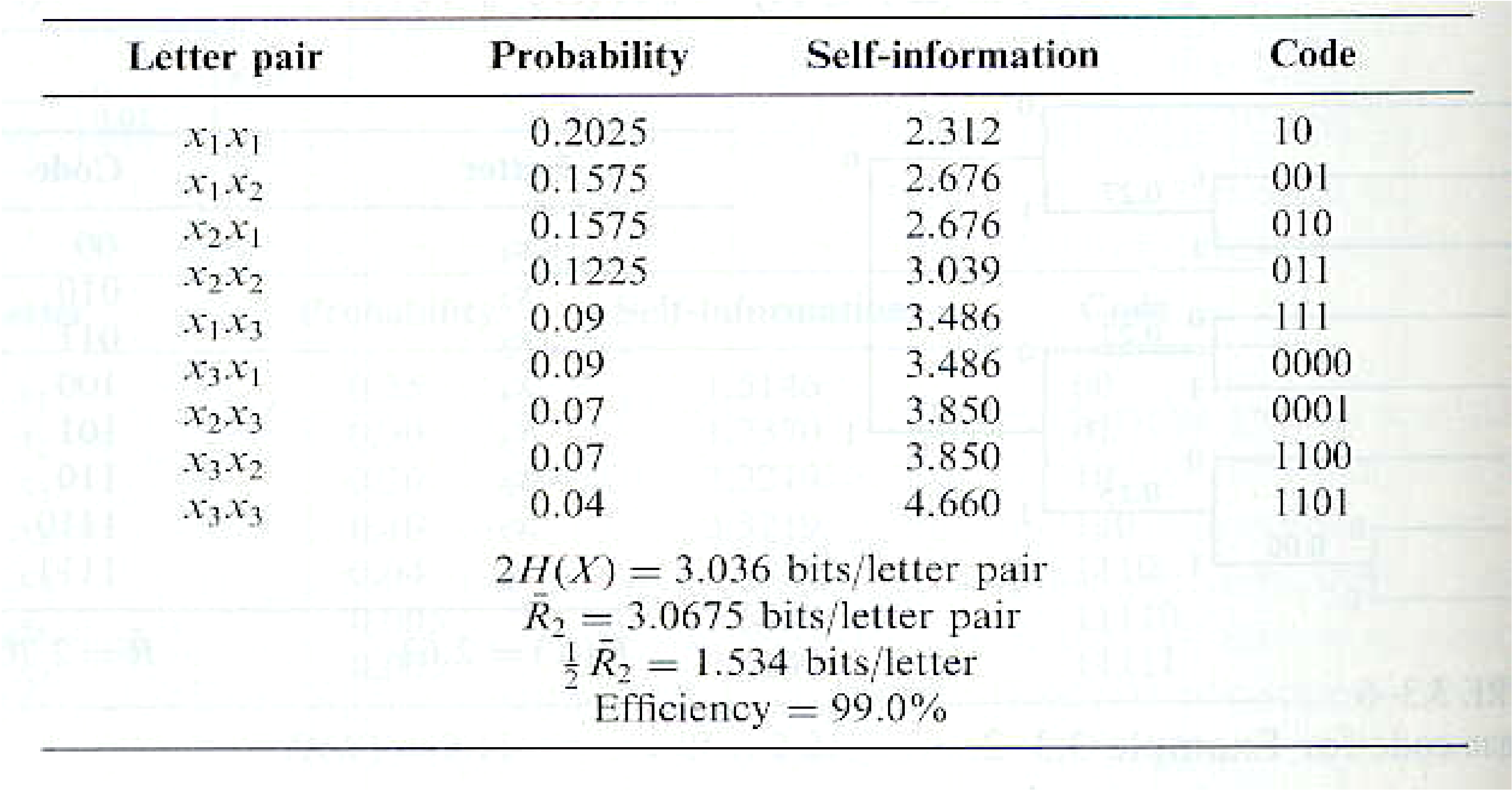


Letter	Probability	Self-information	Code
x_1	0.45	1.156	1
x_2	0.35	1.520	00
x_3	0.20	2.330	01

$$H(X) = 1.518 \text{ bits/letter}$$
$$\bar{R}_1 = 1.55 \text{ bits/letter}$$
$$\text{Efficiency} = 97.9\%$$

Coding for Discrete Memoryless Sources

- Huffman coding algorithm
  - Example



Letter pair	Probability	Self-information	Code
x_1x_1	0.2025	2.312	10
x_1x_2	0.1575	2.676	001
x_2x_1	0.1575	2.676	010
x_2x_2	0.1225	3.039	011
x_1x_3	0.09	3.486	111
x_3x_1	0.09	3.486	0000
x_2x_3	0.07	3.850	0001
x_3x_2	0.07	3.850	1100
x_3x_3	0.04	4.660	1101

$$2H(X) = 3.036 \text{ bits/letter pair}$$
$$\bar{R}_2 = 3.0675 \text{ bits/letter pair}$$
$$\tfrac{1}{2}\bar{R}_2 = 1.534 \text{ bits/letter}$$
$$\text{Efficiency} = 99.0\%$$

Discrete Stationary Sources

- We consider discrete sources for which the sequence of output letters is statistically dependent and statistically stationary.
- The *entropy* of a block of random variables $X_1X_2\cdots X_k$ is:

$$H\left(X_1X_2\ldots X_K\right)=\sum_{i=1}^{k}H\left(X_i\mid X_1X_2\ldots X_{i-1}\right)$$

Eq. (1), P. 20.

- $H(X_i|X_1X_2\cdots X_{i\text{-}1})$ is the *conditional entropy* of the *i*th symbol from the source given the previous *i*-1 symbols.
- The *entropy per letter* for the *k*-symbol block is defined as

$$H_k\left(X\right)=\frac{1}{k}H\left(X_1X_2\ldots X_k\right)$$

- *Information content* of a stationary source is defined as the entropy per letter in the limit as $k\rightarrow\infty$.

$$H_\infty\left(X\right)\equiv\lim_{k\rightarrow\infty}H_k\left(X\right)=\lim_{k\rightarrow\infty}\frac{1}{k}H\left(X_1X_2\ldots X_k\right)$$

Discrete Stationary Sources

- The *entropy per letter* from the source can be defined in terms of the conditional entropy $H(X_k|X_1X_2\cdots X_{k-1})$ in the limit as k approaches infinity.

$$H_\infty(X) = \lim_{k\to\infty} H(X_k \mid X_1X_2\ldots X_{k-1})$$

- For a discrete stationary source that emits J letters with $H_J(X)$ as the entropy per letter.

$$H(X_1\ldots X_J) \le \overline{R}_J < H(X_1\ldots X_J)+1$$

$$H_J(X) \le \overline{R} \equiv \frac{\overline{R}_J}{J} < H_J(X)+\frac{1}{J}$$

- In the limit as $J\to\infty$, we have:

$$H_\infty(X) \le \overline{R} < H_\infty(X)+\varepsilon$$

The Lempel-Ziv Algorithm

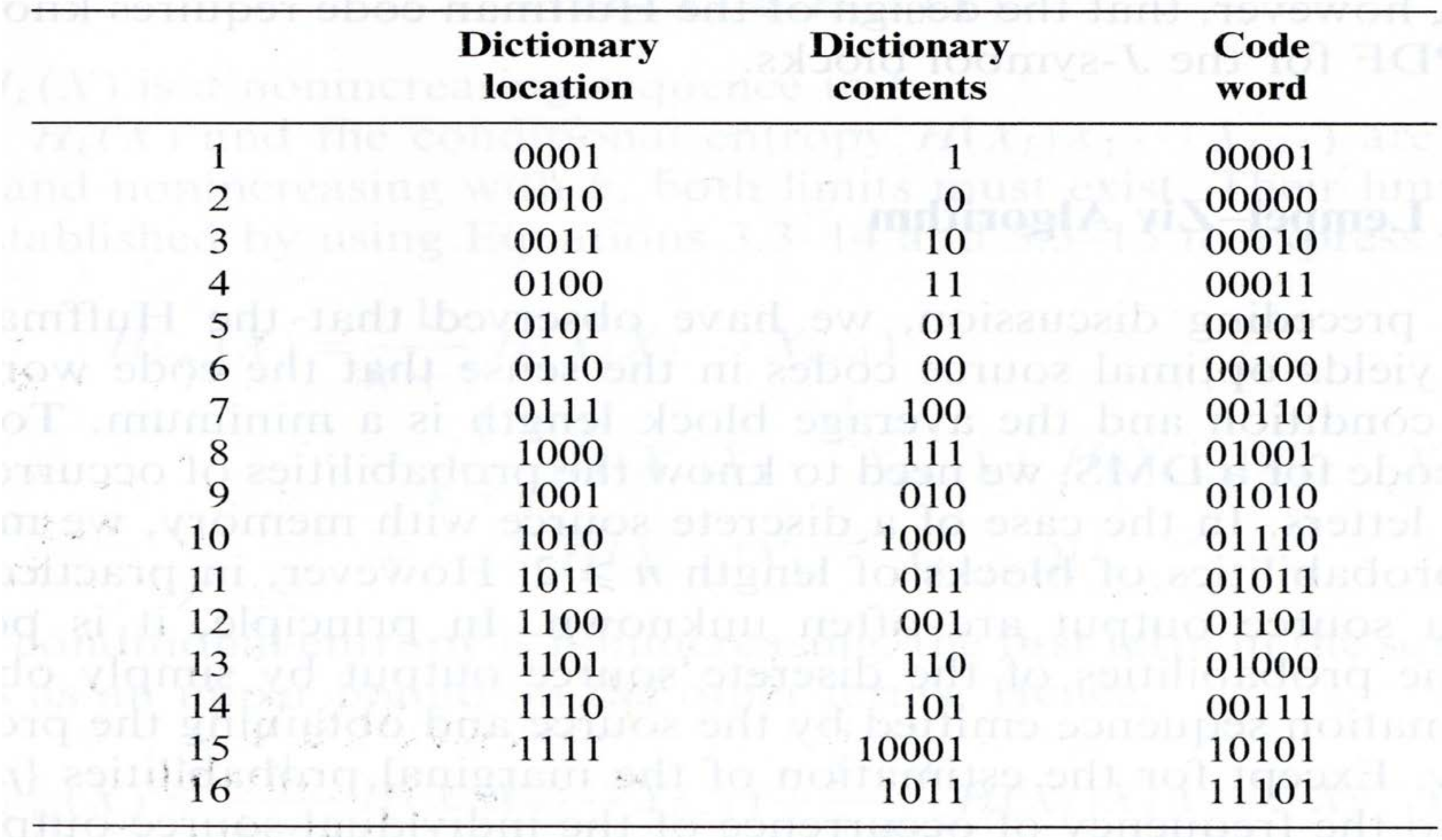
- For Huffman Coding, except for the estimation of the marginal probabilities $\{p_k\}$, corresponding to the frequency of occurrence of the individual source output letters, the computational complexity involved in estimating joint probabilities is extremely high.
- The application of the Huffman coding method to source coding for many real sources with memory is generally impractical.
- The *Lempel-Ziv source coding algorithm* is designed to be independent of the source statistics.
- It belongs to the class of *universal source coding algorithms*.
- It is a variable-to-fixed-length algorithm.

The Lempel-Ziv Algorithm

- Operation of Lempel-Ziv algorithm
  1. The sequence at the output of the discrete source is parsed into variable-length blocks, which are called *phrases*.
  2. A new phrase is introduced every time a block of letters from the source differs from some previous phrase in the last letter.
  3. The phrases are listed in a dictionary, which stores the location of the existing phrases.
  4. In encoding a new phrase, we simply specify the location of the existing phrase in the dictionary and append the new letter.
  - 10101101001001110101000011001110101100011011
  - 1,0,10,11,01,00,100,111,010,1000,011,001,110,101,10001,1011

The Lempel-Ziv Algorithm

- Operation of Lempel-Ziv algorithm (cont.)
  - To encode the phrases, we construct a dictionary:



	Dictionary location	Dictionary contents	Code word
1	0001	1	00001
2	0010	0	00000
3	0011	10	00010
4	0100	11	00011
5	0101	01	00101
6	0110	00	00100
7	0111	100	00110
8	1000	111	01001
9	1001	010	01010
10	1010	1000	01110
11	1011	011	01011
12	1100	001	01101
13	1101	110	01000
14	1110	101	00111
15	1111	10001	10101
16		1011	11101

The Lempel-Ziv Algorithm

- Operation of Lempel-Ziv algorithm (cont.)

5. The code words are determined by listing the dictionary location (in binary form) of the previous phrase that matches the new phrase in all but the last location.
6. The new output letter is appended to the dictionary location of the previous phrase.
7. The location 0000 is used to encode a phrase that has not appeared previously.
8. The source decoder for the code constructs an identical copy of the dictionary at the receiving end of the communication system and decodes the received sequence in step with the transmitted data sequence.

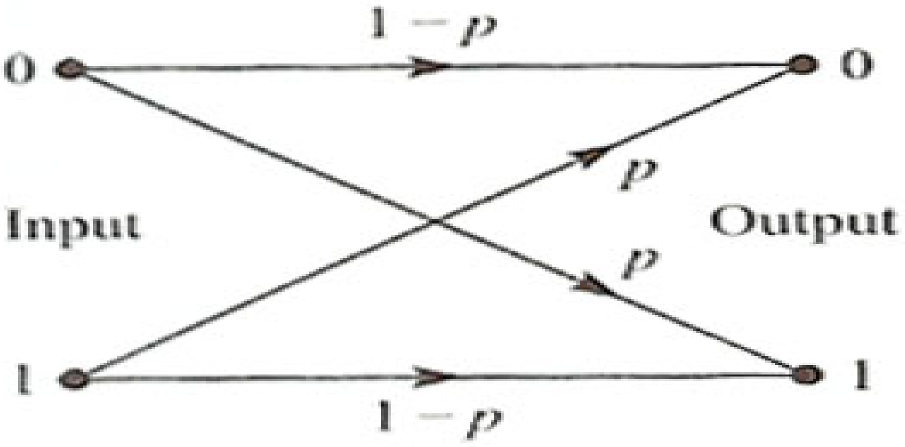
The Lempel-Ziv Algorithm

- Operation of Lempel-Ziv algorithm (cont.)
  - As the sequence is increased in length, the encoding procedure becomes more efficient and results in a compressed sequence at the output of the source.
  - No matter how large the table is, it will eventually overflow.
  - To solve the overflow problem, the source encoder and decoder must use an identical procedure to remove phrases from the dictionaries that are not useful and substitute new phrases in their place.
  - Lempel-Ziv algorithm is widely used in the compression of computer files.
    - E.g. “compress” and “uncompress” utilities under the UNIX© OS.

Channel Models

- Binary symmetric channel (BSC)
  - If the channel noise and other disturbances cause statistically independent errors in the transmitted binary sequence with average probability *p*, then

$$P(Y=0\,|\,X=1)=P(Y=1\,|\,X=0)=p$$
$$P(Y=1\,|\,X=1)=P(Y=0\,|\,X=0)=1-p$$



Channel Models

- Discrete memoryless channels (DMC)
  - BSC is a special case of a more general discrete-input, discrete-output channel.
  - Output symbols from the channel encoder are *q*-ary symbols, i.e., $X=\{x_0, x_1,\cdots, x_{q-1}\}$.
  - Output of the detector consists of *Q*-ary symbols, where $Q \geq M=2^q$.
  - If the channel and modulation are memoryless, we have a set of *qQ* conditional probabilities:

    $$P(Y=y_i \mid X=x_j) \equiv P(y_i \mid x_j)$$

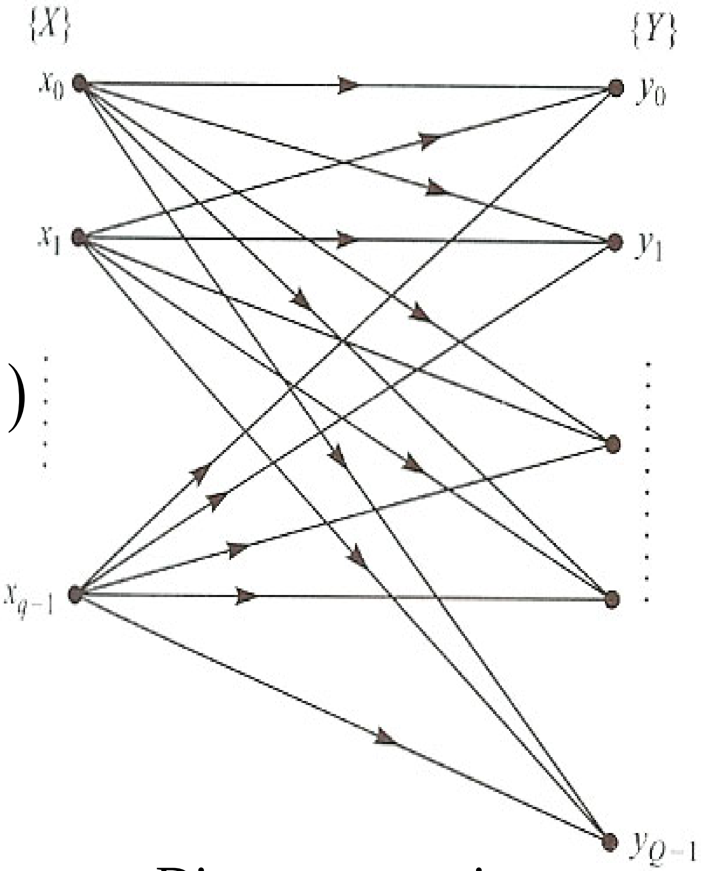
    where $i=0,1,\cdots,Q-1$ and $j=0,1,\cdots,q-1$.
  - Such a channel is called a *discrete memoryless channel* (DMC).

Channel Models

- Discrete memoryless channels (DMC)
  - ✓ Input $u_1, u_2, \cdots, u_n$
  - ✓ Output: $v_1, v_2, \cdots, v_\mathrm{n}$
  - ✓ The conditional probability is given by:

$$P(Y_1 = v_1,\ Y_2 = v_2, \ldots, Y_n = v_n | X = u_1, \ldots,\ X = u_n\ )$$
$$= \prod_{k=1}^{n} P(Y = v_k \mid X = u_k)$$

  - ✓ In general, the conditional probabilities $P(y_j|x_i)$ can be arranged in the matrix form $\mathbf{P}=[p_{ij}]$, called *probability transition matrix*.



Discrete *q*-ary input, *Q*-ary output channel

Channel Models

- Discrete-input, continuous-output channel
  - Discrete input alphabet $X=\{x_0,x_1,\cdots,x_{q\text{-}1}\}$.
  - Output of the detector is unquantized ($Q=\infty$).
  - The most important channel of this type is the additive white Gaussian noise (AWGN) channel, for which

    $$Y = X + G$$

    where G is a zeor-mean Gaussian random variable with variance σ^2 and $X=x_k$, $k=0,1,\cdots,q\text{-}1$.
  - $$p(y \mid X = x_k) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(y-x_k)^2 / 2\sigma^2}$$

$$p(y_1, y_2, \ldots, y_n \mid X_1 = u_1, X_2 = u_2, \ldots, X_n = u_n) = \prod_{i=1}^{n} p(y_i \mid X_i = u_i)$$

Channel Models

- Waveform channels
  - Assume that a channel has a given bandwidth *W*, with ideal frequency response *C*(*f*)=1 within the bandwidth *W*, and the signal at its output is corrupted by AWGN: *y*(*t*)=*x*(*t*)+*n*(*t*).
  - Expand *y*(*t*), *x*(*t*), and *n*(*t*) into a complete set of orthonormal functions:

$$y(t)=\sum_i y_i f_i(t), \quad x(t)=\sum_i x_i f_i(t), \quad n(t)=\sum_i n_i f_i(t).$$

$$y_i=\int_0^T y(t)\, f_i^*(t)\,dt=\int_0^T \left[x(t)+n(t)\right] f_i^*(t)\,dt=x_i+n_i$$

$$\int_0^T f_i(t) f_j^*(t)dt=\delta_{ij}=\begin{cases}1 & (i=j)\\ 0 & (i\neq j)\end{cases}$$

Channel Models

- Waveform channels
  - Since $y_i = x_i + n_i$, it follows that:

$$p(y_i \mid x_i) = \frac{1}{\sqrt{2\pi}\sigma_i} e^{-(y_i - x_i)^2 / 2\sigma_i^2}, \qquad i = 1, 2, \ldots$$

  - Since the functions $\{f_i(t)\}$ are orthonormal, it follows that the $\{n_i\}$ are uncorrelated.
  - Since they are Gaussian, they are also statistically independent:

$$p(y_1, y_2, \ldots, y_N \mid x_1, x_2, \ldots, x_N) = \prod_{i=1}^{N} p(y_i \mid x_i)$$

  - Samples of $x(t)$ and $y(t)$ may be taken at the Nyquist rate of $2W$ samples per second. Thus, in a time interval of length T, there are $N=2WT$ samples.

Channel Capacity

- Consider a DMC having an input alphabet $X=\{x_0,x_1,\cdots,x_{q\text{-}1}\}$, an output alphabet $Y=\{\mathrm{y}_0, \mathrm{y}_1, \cdots, \mathrm{y}_{Q\text{-}1}\}$, and the set of transition probabilities $P(y_i,x_j)$.
- The mutual information provided about the event $X=x_j$ by the occurrence of the event $Y=y_i$ is $\log[P(y_i|x_j)/P(y_i)]$, where

$$P(y_i) \equiv P(Y=y_i) = \sum_{k=0}^{q-1} P(x_k)P(y_i \mid x_k)$$

- Hence, the average mutual information provided by the output Y about the input X is:

$$I(X;\ Y) = \sum_{j=0}^{q-1}\sum_{i=0}^{Q-1} P(x_j)P(y_i \mid x_j)\log\frac{P(y_i \mid x_j)}{P(y_i)}$$

Channel Capacity

- The value of $I(X;Y)$ maximized over the set of input symbol probabilities $P(x_j)$ is a quantity that depends only on the characteristics of the DMC through the conditional probabilities $P(y_i|x_j)$. This quantity is called the *capacity* of the channel and is denoted by C:

$$C = \max_{P(x_j)} I(X;\ Y)$$

$$= \max_{P(x_j)} \sum_{j=0}^{q-1} \sum_{i=0}^{Q-1} P(x_j) P(y_i \mid x_j) \log \frac{P(y_i \mid x_j)}{P(y_i)}$$

- The maximization of $I(X;Y)$ is performed under the constraints that

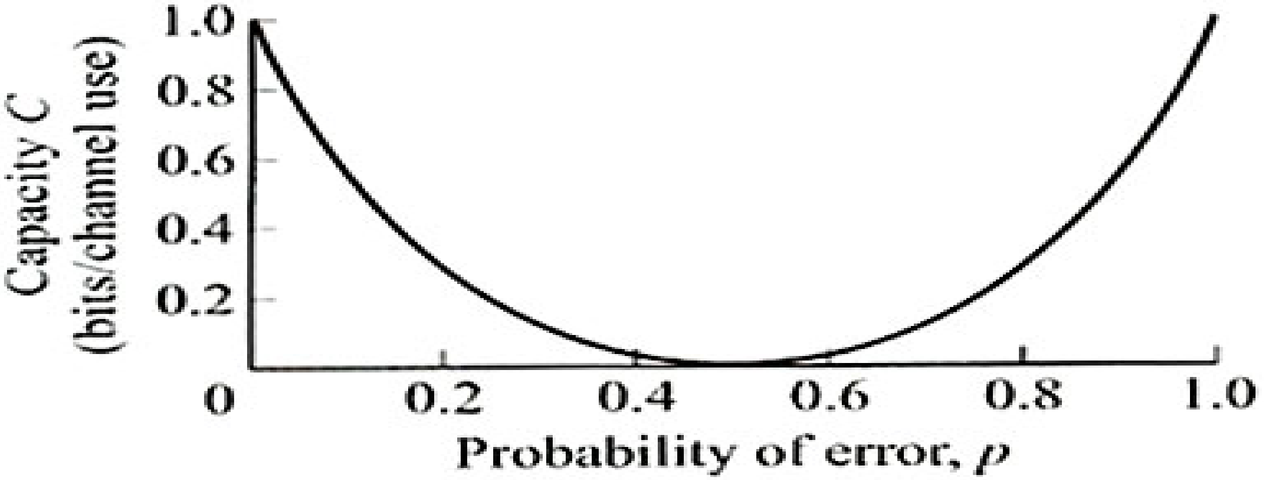
$$P(x_j) \geq 0 \text{ and } \sum_{j=0}^{q-1} P(x_j) = 1.$$

Channel Capacity

- Example: BSC with transition probabilities $P(0|1)=P(1|0)=p$.
  - The average mutual information is maximized when the input probabilities $P(0)=P(1)=½$.
  - The capacity of the BSC is

$$C = p\log 2p + (1-p)\log 2(1-p) = 1 - H(p)$$

where $H(p)$ is the binary entropy function.



Channel Capacity

- Consider the discrete-time AWGN memoryless channel described by

$$p\left(y \mid X = x_k\right) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\left(y-x_k\right)^2 / 2\sigma^2}$$

- The capacity of this channel in bits per channel use is the maximum average mutual information between the discrete input $X=\{x_0,x_1,\cdots,x_{q\text{-}1}\}$ and the output $Y=\{\infty,-\infty\}$:

$$C = \max_{P(x_i)} \sum_{i=0}^{q-1} \int_{-\infty}^{\infty} p\left(y \mid x_i\right) P\left(x_i\right) \log_2 \frac{P\left(y \mid x_i\right)}{P\left(y\right)} dy$$

where

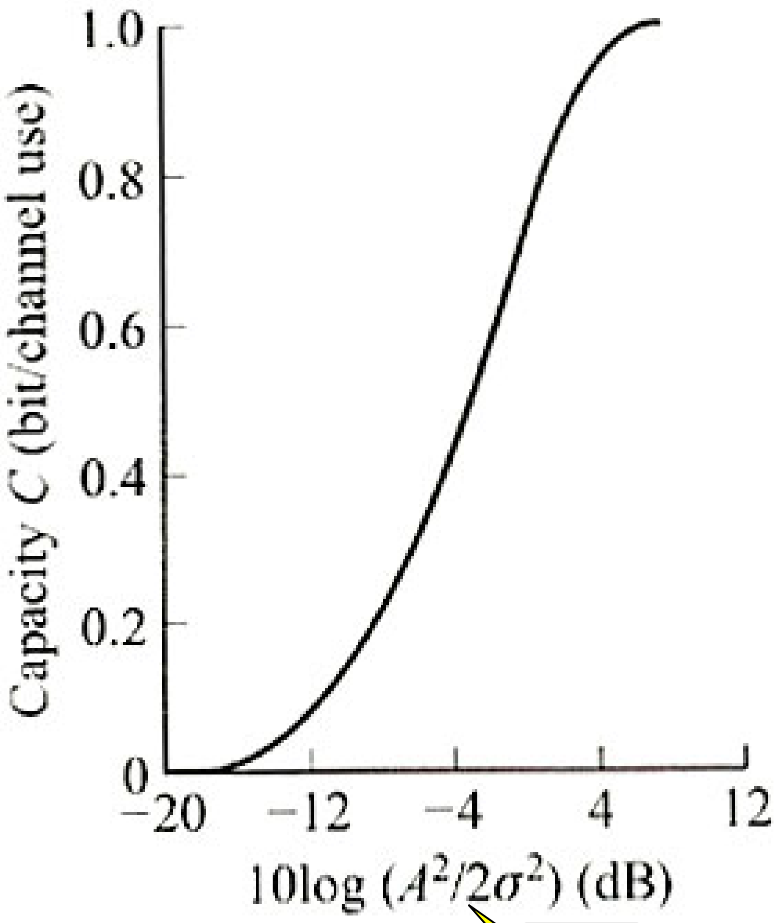
$$p\left(y\right) = \sum_{k=0}^{q-1} p\left(y \mid x_k\right) P\left(x_k\right)$$

Channel Capacity

- Example: Consider a binary-input AWGN memoryless channel with possible inputs $X=A$ and $X=-A$.
  - The average mutual information $I(X;Y)$ is maximized when the input probabilities are $P(X=A)=P(X=-A)=½$.

$$C=\frac{1}{2}\int_{-\infty}^{\infty}p(y\,|\,A)\log_2\frac{p(y\,|\,A)}{p(y)}dy$$

$$+\frac{1}{2}\int_{-\infty}^{\infty}p(y\,|-A)\log_2\frac{p(y\,|-A)}{p(y)}dy$$



SNR

Channel Capacity

- It is not always the case to obtain the channel capacity by assuming that the input symbols are equally probable.
- Nothing can be said in general about the input probability assignment that maximizes the average mutual information.
- It can be shown that the necessary and sufficient conditions for the set of input probabilities $\{P(x_j)\}$ to maximize $I(X;Y)$ and to achieve capacity on a DMC are:

$$
\begin{aligned}
I(x_j;Y) &= C \quad \text{for all } j \text{ with } P(x_j) > 0 \\
I(x_j;Y) &\le C \quad \text{for all } j \text{ with } P(x_j) = 0
\end{aligned}
$$

where C is the capacity of the channel and

$$
I(x_j;Y) = \sum_{i=0}^{Q-1} P(y_i \mid x_j) \log \frac{P(y_i \mid x_j)}{P(y_i)}
$$

Channel Capacity

- Consider a band-limited waveform channel with AWGN.
- The capacity of the channel per unit time has been defined by Shannon (1948) as

$$C = \lim_{T\to\infty} \max_{p(x)} \frac{1}{T} I(X;Y)$$

- Alternatively, we may use the samples or the coefficients $\{y_i\}$, $\{x_i\}$, and $\{n_i\}$ in the series expansions of $y(t)$, $x(t)$, and $n(t)$ to determine the average mutual information between $\mathbf{x}_N=[x_1\ x_2 \cdots x_N]$ and $\mathbf{y}_N=[y_1\ y_2 \cdots y_N]$, where $N=2WT$, $y_i = x_i + n_i$.

$$I(\mathbf{X}_N;\mathbf{Y}_N) = \int_{\mathbf{x}_N} \cdots \int\int_{\mathbf{y}_N} \cdots \int p(\mathbf{y}_N \,|\, \mathbf{x}_N)\, p(\mathbf{x}_N) \log \frac{p(\mathbf{y}_N \,|\, \mathbf{x}_N)}{p(\mathbf{y}_N)}\, d\mathbf{x}_N\, d\mathbf{y}_N$$

$$= \sum_{i=1}^{N} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} p(y_i \,|\, x_i)\, p(x_i) \log \frac{p(y_i|x_i)}{p(y_i)}\, dy_i\, dx_i \qquad (*)$$

Channel Capacity

where

$$p(y_i \mid x_i) = \frac{1}{\sqrt{\pi N_0}} e^{-(y_i - x_i)^2 / N_0}$$

- The maximum of $I(X;Y)$ over the input PDFs $p(x_i)$ is obtained when the $\{x_i\}$ are statistically independent zero-mean Gaussian random variables, i.e.,

$$p(x_i) = \frac{1}{\sqrt{2\pi}\sigma_x} e^{-x_i^2 / 2\sigma_x^2}$$

- From (*) in P.67

$$\max_{p(x)} I\left(\mathbf{X}_N;\ \mathbf{Y}_N\right) = \sum_{i=1}^{N} \frac{1}{2} \log\left(1 + \frac{2\sigma_x^2}{N_0}\right) = \frac{1}{2} N \log\left(1 + \frac{2\sigma_x^2}{N_0}\right)$$

$$= WT \log\left(1 + \frac{2\sigma_x^2}{N_0}\right)$$

Channel Capacity

- If we put a constraint on the average power in *x*(*t*), i.e.,

$$P_{av} = \frac{1}{T}\int_0^T E\left[x^2(t)\right]dt = \frac{1}{T}\sum_{i=1}^{N} E\left(x_i^2\right) = \frac{N\sigma_x^2}{T}$$

$$\sigma_x^2 = \frac{TP_{av}}{N} = \frac{P_{av}}{2W}$$

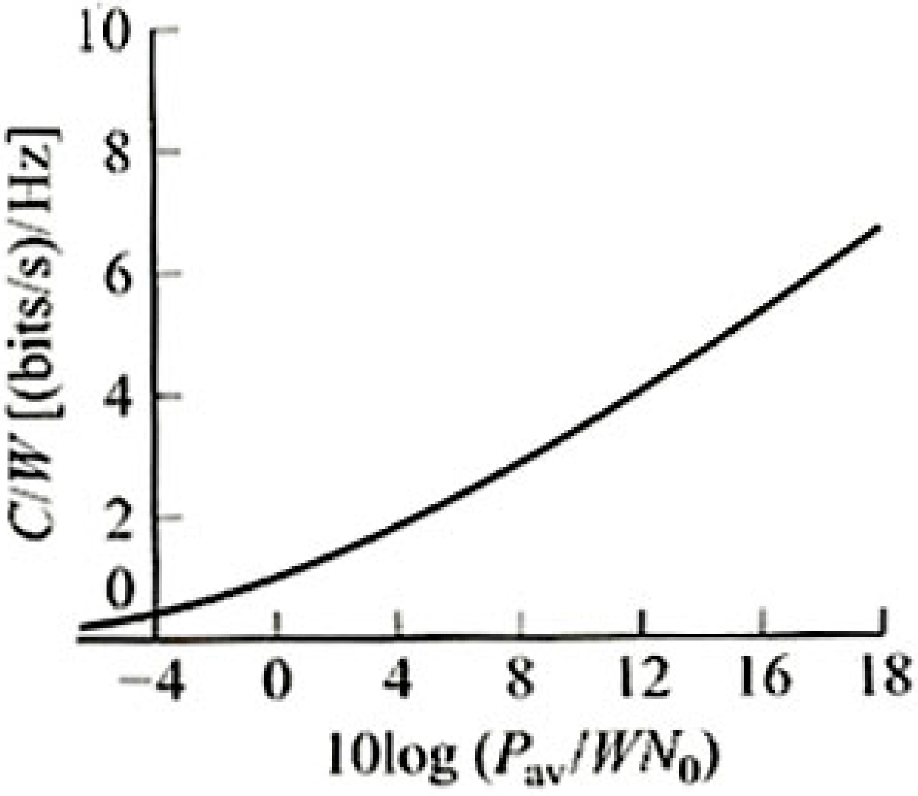
$$\max_{p(x)} I\left(\mathbf{X}_N;\mathbf{Y}_N\right) = WT\log\left(1+\frac{P_{av}}{WN_0}\right)$$

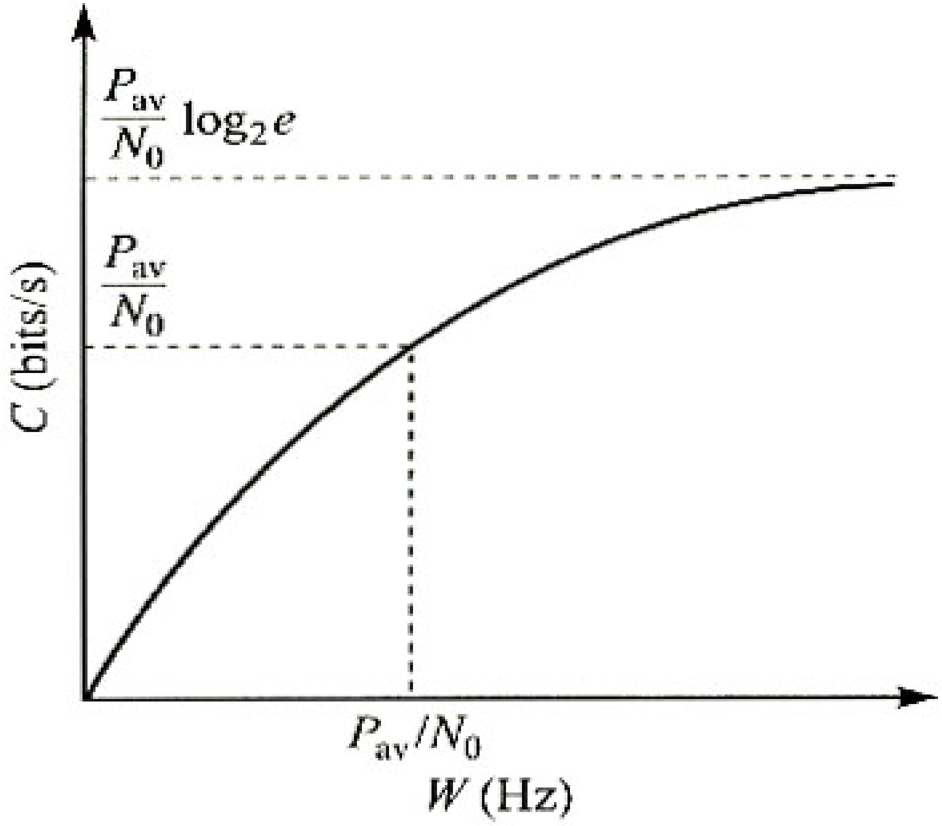
- Dividing both sides by *T* and we can obtain the capacity of the band-limited AWGN waveform channel with a band-limited and average power-limited input:

$$\boxed{C = W\log\left(1+\frac{P_{av}}{WN_0}\right)}$$

Channel Capacity

- Normalized channel capacity as a function of SNR for band-limited AWGN channel
- Channel capacity as a function of bandwidth with a fixed transmitted average power





Channel Capacity

- Note that as *W* approaches infinity, the capacity of the channel approaches the asymptotic value

$$C_{\infty} = \frac{P_{av}}{N_0}\log_2 e = \frac{P_{av}}{N_0 \ln 2} \quad \text{bits/s}$$

- Since P_{av} represents the average transmitted power and *C* is the rate in bits/s, it follows that

$$P_{av} = C\varepsilon_b$$

- Hence, we have

$$\frac{C}{W} = \log_2\left(1 + \frac{C}{W}\frac{\varepsilon_b}{N_0}\right)$$

- Consequently

$$\frac{\varepsilon_b}{N_0} = \frac{2^{C/W} - 1}{C/W}$$

Channel Capacity

- When C/W=1, ε_b/N_0=1 (0 dB).

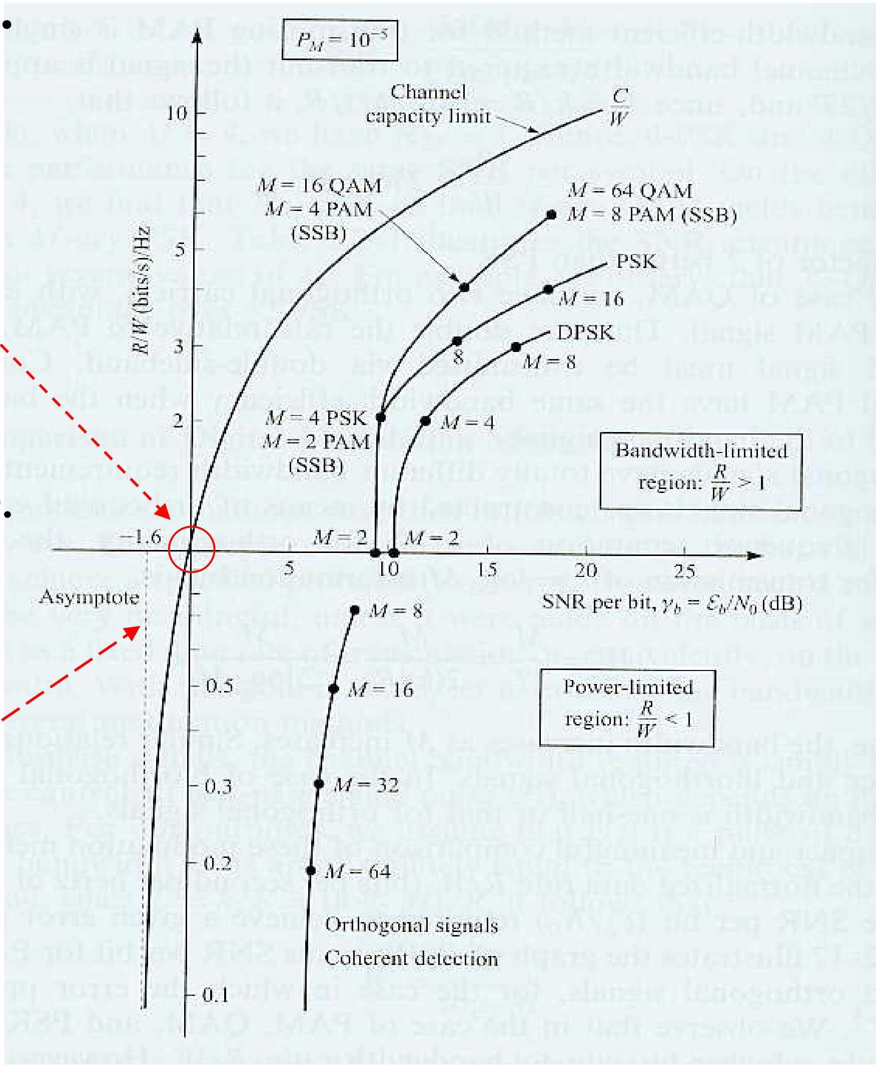
- When $C/W\to\infty$,

$$\frac{\varepsilon_b}{N_0} \approx \frac{2^{C/W}}{C/W} \approx \exp\left(\frac{C}{W}\ln 2 - \ln\frac{C}{W}\right)$$

$\frac{\varepsilon_b}{N_0}$ increases expontially as $C/W \to \infty$.

- When $C/W\to 0$

$$\frac{\varepsilon_b}{N_0} = \lim_{C/W\to 0}\frac{2^{C/W}-1}{C/W} = \ln 2$$



Channel Capacity

- The channel capacity formulas serve as upper limits on the transmission rate for reliable communication over a noisy channel.
- Noisy channel coding theorem by Shannon (1948)
  - There exist channel codes (and decoders) that make it possible to achieve reliable communication, with as small an error probability as desired, if the transmission rate $R<C$, where C is the channel capacity. If $R>C$, it is not possible to make the probability of error tend toward zero with any code.