

1. Introduction to Information Theory

The basic purpose of a **communication system** is to transmit signals generated by an **information source** through a **communication channel**.

Information Theory provides the mathematical framework to understand, model, and analyse the transmission of information over communication systems.

It mainly deals with the **mathematical modelling and analysis** of communication systems rather than the physical source and physical channel.

Why Study Information Theory?

Information theory is important because it:

1. Uses **probability theory** to model the behavior of information sources and communication channels.
2. Gives a precise meaning to the term **information**.
3. Provides the foundation for important concepts such as **source coding, channel coding, data compression, and reliable communication**.

Two Fundamental Questions

Information theory addresses two important questions:

1. **How much can a source be compressed?**
→ Answer: **Entropy of the source**
2. **What is the maximum rate at which information can be reliably transmitted through a noisy channel?**
→ Answer: **Channel Capacity**

Entropy and Capacity

- **Entropy** represents the average information content or the uncertainty associated with a source.
- **Channel Capacity** represents the intrinsic ability of a communication channel to convey information reliably and is related to the **noise characteristics** of the channel.

A key result of information theory is: $H(S) < C$

where, $H(S)$ = Entropy of the source C = Capacity of the channel

If the **source entropy is less than the channel capacity**, then, ideally, **error-free communication can be achieved over the noisy channel**.

Entropy tells us how much information the source produces

Channel capacity tells us how much information the channel can reliably carry.

2. Entropy and Information

- The entropy of a discrete random variable, representing the output of a source of information, is a measure of the average information content per source symbol.
- The entropy is represented by $H(S)$.
- In a digital communication system, the source produces information in the form of symbols. Since different symbols may occur with different probabilities, there is uncertainty about which symbol will be produced next.
- Information is closely related to **uncertainty and surprise**.
- If an event is certain, there is no surprise and hence no new information.
- If an event is unlikely, its occurrence gives us more information.
- Therefore, the amount of information is related to the inverse of the probability of occurrence.
- Consider a discrete source represented by a discrete random variable S . Its alphabet is

$$\mathcal{S} = \{s_0, s_1, \dots, s_{K-1}\}$$

where K is the number of symbols in the source alphabet.

- The probability of occurrence of symbol s_k is

$$P(S = s_k) = p_k, k = 0, 1, \dots, K - 1$$

- The probabilities satisfy the normalization condition

$$\sum_{k=0}^{K-1} p_k = 1, p_k \geq 0$$

- The source symbols emitted during successive signaling intervals are assumed to be statistically independent

2.1 Self-Information

The amount of information gained when the event $S = s_k$ occurs with probability p_k is called the self-information of the event.

It is defined as

$$I(s_k) = \log\left(\frac{1}{p_k}\right)$$

For binary signaling, the logarithm is normally taken to the base 2:

$$I(s_k) = \log_2\left(\frac{1}{p_k}\right) = -\log_2 p_k$$

The unit of information is then the bit

2.2 Important Properties of Self-Information

Property 1: Certain event

If $p_k = 1$ Then $I(s_k) = 0$

This means that a completely certain event provides **no new information**.

Property 2: Information is non-negative

For $0 < p_k \leq 1$

$$I(s_k) \geq 0$$

Thus, an event provides some information or no information, but not negative information.

Property 3: Less probable event gives more information

If $p_k < p_i$ then $I(s_k) > I(s_i)$

Therefore, an unlikely event carries more information than a highly probable event.

Property 4: Additive property

For statistically independent events s_k and s_l ,

$$I(s_k, s_l) = I(s_k) + I(s_l)$$

This additive property results from the logarithmic definition of self-information.

2.3 Unit of Information – Bit

When the logarithm is taken to base 2,

$$I(s_k) = \log_2 \left(\frac{1}{p_k} \right)$$

the resulting unit is called a **bit**.

Example: For two equally likely events,

$$p_k = \frac{1}{2}$$

therefore,

$$I(s_k) = \log_2 2 = 1 \text{ bit}$$

Hence:

One bit is the amount of information obtained when one of two equally likely events occurs.

2.4 Entropy

The self-information depends on which symbol is actually produced. Therefore, self-information itself can be considered a **discrete random variable**.

The average information produced by the source is obtained by taking the expectation of self-information.

The **entropy** of a discrete random variable S is defined as

$$H(S) = E[I(s_k)]$$

Therefore,

$$H(S) = \sum_{k=0}^{K-1} p_k I(s_k)$$

Substituting

$$I(s_k) = \log_2 \left(\frac{1}{p_k} \right)$$

we obtain

$$H(S) = \sum_{k=0}^{K-1} p_k \log_2 \left(\frac{1}{p_k} \right)$$

or equivalently,

$$H(S) = - \sum_{k=0}^{K-1} p_k \log_2 p_k$$

Definition

Entropy is the average information content per source symbol produced by a source.

Entropy depends only on the **probabilities of the source symbols**, not on the actual names or values of the symbols.

2.5 Properties of Entropy

or source having K possible symbols,

$$0 \leq H(S) \leq \log_2 K$$

i) Minimum Entropy

$$H(S) = 0$$

when one symbol has probability 1 and all other symbols have probability 0.

For example,

$$p_0 = 1, p_1 = p_2 = \dots = 0$$

There is no uncertainty because the source output is completely predictable.

ii) Maximum Entropy

$$H(S) = \log_2 K$$

when all K symbols are **equiprobable**:

$$p_k = \frac{1}{K}, k = 0, 1, \dots, K - 1$$

Maximum entropy therefore corresponds to **maximum uncertainty**.

Important Result

Condition	Entropy
Completely certain source	$H(S) = 0$
Maximum uncertainty	$H(S) = \log_2 K$
All symbols equally probable	Maximum entropy

Example1: Entropy of a Bernoulli Random Variable

A **Bernoulli random variable** has two possible symbols, 0 and 1.

Let $P(S = 0) = p_0$

and $P(S = 1) = p_1 = 1 - p_0$

The entropy is $H(S) = -p_0 \log_2 p_0 - p_1 \log_2 p_1$

Since $p_1 = 1 - p_0$,

$$H(S) = -p_0 \log_2 p_0 - (1 - p_0) \log_2 (1 - p_0)$$

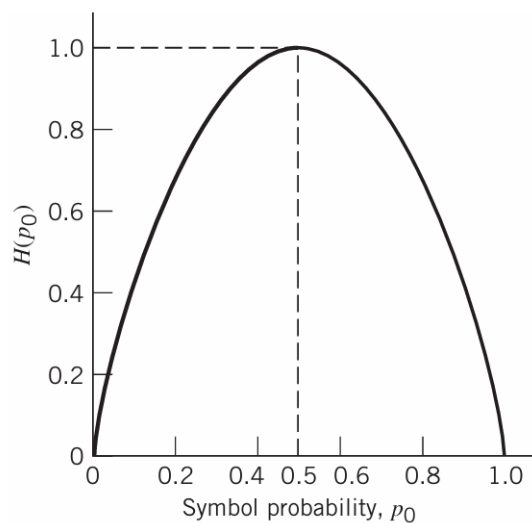
The corresponding **entropy function** is written as

$$H(p_0) = -p_0 \log_2 p_0 - (1 - p_0) \log_2 (1 - p_0)$$

Important Observations

Condition	Entropy
$p_0 = 0$	$H(S) = 0$
$p_0 = 1$	$H(S) = 0$
$p_0 = p_1 = \frac{1}{2}$	$H_{\max} = 1 \text{ bit}$

The entropy function is **symmetric about** $p_0 = \frac{1}{2}$



Thus, a binary source has maximum uncertainty when its two symbols are equally likely.

Example 2: A source emits one of four possible symbols during each signalling interval. The symbols occur with the probabilities

$$P_0 = 0.4, P_1 = 0.3, P_2 = 0.2 \text{ and } P_3 = 0.1$$

which sum to unity as they should. Find the amount of information gained by observing the source emitting each of these symbols, and also the entropy of the source.

Solution:

$$I(s_k) = \log_2 \left(\frac{1}{P_k} \right) \text{ bits.}$$

- for $s_0 \rightarrow I(s_0) = \log_2 \left(\frac{1}{P_0} \right) = \log_2 \left(\frac{1}{0.4} \right) = 1.322 \text{ bits.}$
- for $s_1 \rightarrow I(s_1) = \log_2 \left(\frac{1}{P_1} \right) = \log_2 \left(\frac{1}{0.3} \right) = 1.737 \text{ bits.}$
- for $s_2 \rightarrow I(s_2) = \log_2 \left(\frac{1}{P_2} \right) = \log_2 \left(\frac{1}{0.2} \right) = 2.322 \text{ bits.}$
- for $s_3 \rightarrow I(s_3) = \log_2 \left(\frac{1}{P_3} \right) = \log_2 \left(\frac{1}{0.1} \right) = 3.322 \text{ bits.}$

The entropy of the source is given by

$$H(s) = \sum_{k=0}^{K-1} P_k \log_2 \left(\frac{1}{P_k} \right)$$

$$H(s) = \sum_{k=0}^3 P_k \log_2 \left(\frac{1}{P_k} \right)$$

$$H(s) = P_0 \log_2 \left(\frac{1}{P_0} \right) + P_1 \log_2 \left(\frac{1}{P_1} \right) + P_2 \log_2 \left(\frac{1}{P_2} \right) + P_3 \log_2 \left(\frac{1}{P_3} \right)$$

$$H(s) = 0.4 \log_2 \left(\frac{1}{0.4} \right) + 0.3 \log_2 \left(\frac{1}{0.3} \right) + 0.2 \log_2 \left(\frac{1}{0.2} \right) + 0.1 \log_2 \left(\frac{1}{0.1} \right)$$

$$H(s) = 1.8465 \text{ bits/message symbol.}$$

2.7 Extension of a Discrete Memoryless Source

A source is called **memoryless** when the symbol emitted at any time is statistically independent of the symbols emitted at previous and future signaling intervals.

Instead of considering individual source symbols, we can consider a block of n successive source symbols.

If the original source alphabet contains K symbols, then the extended source has K^n possible blocks.

Since the source symbols are statistically independent, the probability of a particular block is equal to the **product of the probabilities** of the symbols forming that block.

The entropy of the extended source is $H(S^n) = nH(S)$

where:

- S = original source
- S^n = n -th order extension of the source
- $H(S)$ = entropy per source symbol
- $H(S^n)$ = entropy of the n -symbol block.

Example 3: Consider a discrete memoryless source with source alphabet $S = \{s_0, s_1, s_2\}$ and source statistics $\{1/4, 1/4, 1/2\}$.

- Calculate the entropy of the source.
- Calculate the entropy of the second-order extension of the source.

Solution:

a) Entropy of the source is

$$H(S) = P_0 \log_2 \left(\frac{1}{P_0} \right) + P_1 \log_2 \left(\frac{1}{P_1} \right) + P_2 \log_2 \left(\frac{1}{P_2} \right)$$

$$H(S) = \frac{1}{4} \log_2(4) + \frac{1}{4} \log_2(4) + \frac{1}{2} \log_2(2)$$

$$H(S) = \frac{3}{2} \text{ bits} = 1.5 \text{ bits}$$

b) Entropy of the second-order extension of the source:

Symbols:	$\sigma 0$	$\sigma 1$	$\sigma 2$	$\sigma 3$	$\sigma 4$	$\sigma 5$	$\sigma 6$	$\sigma 7$	$\sigma 8$
Sequences of symbols of S :	$s_0 s_0$	$s_0 s_1$	$s_0 s_2$	$s_1 s_0$	$s_1 s_1$	$s_1 s_2$	$s_2 s_0$	$s_2 s_1$	$s_2 s_2$
Probability:	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{1}{8}$	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{4}$

$$H(S^2) = \sum_{K=0}^{K-1} P_K \log_2 \left(\frac{1}{P_K} \right) = \sum_{K=0}^8 P_K \log_2 \left(\frac{1}{P_K} \right)$$

$$H(S^2) = \frac{1}{16} \log_2(16) + \frac{1}{16} \log_2(16) + \frac{1}{8} \log_2(8) + \frac{1}{16} \log_2(16) + \frac{1}{16} \log_2(16) + \frac{1}{8} \log_2(8) \\ + \frac{1}{8} \log_2(8) + \frac{1}{8} \log_2(8) + \frac{1}{4} \log_2(4)$$

$$H(S^2) = 3 \text{ bits.}$$

Note: From **a** and **b**, we see that:

- $H(S^2) = 3 \text{ bits}$
- $H(S) = 1.5 \text{ bits}$
- $H(S^2) = 2H(S)$.

3. SOURCE-CODING THEOREM

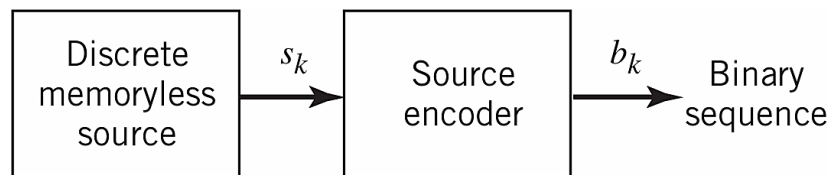
Source Encoding:

- It is the process of representing data generated by a discrete source in a more efficient form, usually as a sequence of binary digits (0's and 1's).
- The device that performs the representation is called a source encoder.
- When encoding symbols from a source, some symbols may occur more frequently than others.
- By assigning shorter codewords to more frequent symbols and longer codewords to less frequent symbols, we can achieve a more efficient encoding scheme.
- This method is known as variable-length coding.

Key Requirements for Source Codes:

- The codewords produced by the encoder are in Binary form.
- The source code is uniquely decodable, so that the original source sequence can be reconstructed perfectly from the encoded binary sequence.

Consider a discrete memoryless source whose output s_k is converted by the source encoder into a sequence of 0s and 1s, denoted by b_k .



- Let the binary codeword assigned to symbol s_k by the encoder have length l_k , measured in bits. The average codeword length ($\bar{L}$) of the source encoder is:

$$\bar{L} = \sum_{k=0}^{K-1} P_k l_k$$

- The parameter $\bar{L}$ represents the average number of bits per source symbol used in the source encoding process.

- The coding efficiency (η) is: $\eta = \frac{H(S)}{\bar{L}}$

Shannon's First Theorem (Source-Coding Theorem):

Statement: Given a discrete memoryless source whose output is denoted by the random variable S , the entropy $H(S)$ imposes the following bound on the average codeword length $\bar{L}$ for any source encoding scheme:

$$\bar{L} \geq H(S)$$

4. LOSSLESS DATA COMPRESSION ALGORITHMS

4.1 Lossless Data Compression:

- Lossless data compression is a process of removing redundancy from a signal without any loss of information.
- This process, typically applied to digital signals, results in an efficient representation that minimizes the average number of bits per symbol.

4.2 Prefix Coding:

- **Prefix Codes:** A prefix code is defined as a code in which no codeword is the prefix of any other codeword.
- It is a type of source code used for lossless data compression.
- These codes are uniquely decodable.
- Prefix codes are distinguished from other uniquely decodable codes by the fact that the end of a codeword is always recognizable.
- Hence, the decoding of a prefix can be accomplished as soon as the binary sequence representing a source symbol is fully received.
- For this reason, prefix codes are also referred to as **instantaneous codes**.

Illustrative Example of Prefix Coding:

To illustrate the meaning of a prefix code, consider the three source codes described in the table below:

Table 1: Illustrating the definition of a prefix code

Symbol source	Probability of occurrence	Code-I	Code-II	Code-III
S_0	0.5	0	0	0
S_1	0.25	1	10	01
S_2	0.125	00	110	011
S_3	0.125	11	111	0111

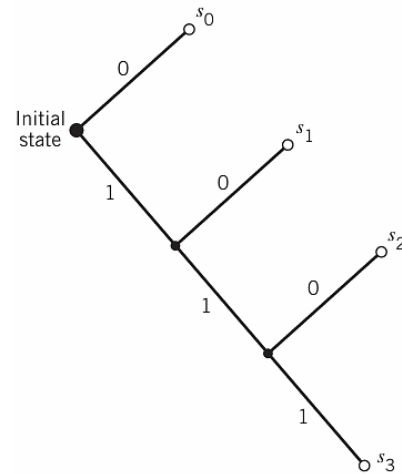
- **Code-I** is not a prefix code because the bit 0, the codeword for S_0 , is a prefix of 00, the codeword for S_2 . Likewise, the bit 1, the codeword for S_1 , is a prefix of 11, the codeword for S_3 .
- Similarly, we may show that **Code-III** is not a prefix code, but **Code-II** is a prefix code because no codeword is a prefix of another.

Decoding of a Prefix Code:

To decode a sequence of codewords from a prefix code, a decision tree is often used to represent the code.

Decision Tree for Code-II of Table-1:

- The decoding process involves traversing this tree based on the bits received in the encoded sequence.
- The decoder starts at the initial state and moves down the tree based on the bits it encounters, until it reaches a terminal state, which corresponds to one of the source symbols.
- After decoding a symbol, the decoder resets to the initial state for the next codeword.



Example: Consider the following encoded

sequence:

1011111000 ...

This sequence can be decoded using Code-II of Table-1 as follows:

$$\underbrace{10}_{s_1} \quad \underbrace{111}_{s_3} \quad \underbrace{110}_{s_2} \quad \underbrace{0}_{s_0} \quad \underbrace{0}_{s_0} \dots$$

4.3 KRAFT INEQUALITY:

Consider a discrete memoryless source with source alphabet $\{s_0, s_1, \dots, s_{K-1}\}$ and source probabilities $\{P_0, P_1, \dots, P_{K-1}\}$ with the codeword of symbol s_k having length l_k .

Where $k = 0, 1, \dots, K-1$. Then, according to the Kraft inequality, the codeword lengths always satisfy the following inequality:

$$\sum_{k=0}^{K-1} 2^{-l_k} \leq 1$$

where: l_k = length of the codeword for symbol s_k .

2 refers to the binary alphabet.

The Kraft inequality is a **necessary but not sufficient condition** for a source code to be a prefix code.

For example, referring to the three codes listed in Table-1, we see:

- **Code-I:**

$$\sum_{k=0}^3 2^{-l_k} = 2^{-1} + 2^{-1} + 2^{-2} + 2^{-2} = 1.5 \not\leq 1$$

Code-I violates the Kraft inequality; it cannot, therefore, be a prefix code.

- Similarly, the Kraft inequality is satisfied by both Codes II and III, but **only Code-II is a prefix code**.

- **Bounds on Average Codeword Length ($\bar{L}$):** For a discrete memoryless source with entropy $H(S)$, a prefix code can always be constructed such that:

$$H(S) \leq \bar{L} < H(S) + 1$$

- **Perfect Efficiency Condition (Dyadic Distribution):** The minimum possible length $\bar{L} = H(S)$ is achieved if and only if the symbol probability matches:

$$p_k = 2^{-l_k}$$

- l_k is the length of the codeword assigned to symbol s_k .
- A probability distribution following this form is called a **dyadic distribution**.

- **Codeword Length for Dyadic Source:** Under this condition, the ideal codeword length for symbol s_k is given by:

$$l_k = -\log_2 p_k$$

- **Average Length & Entropy Equality:** For a dyadic distribution, the total probability sums to 1 ($\sum_{k=0}^{K-1} 2^{-l_k} = \sum_{k=0}^{K-1} p_k = 1$), leading to:

$$\bar{L} = \sum_{k=0}^{K-1} \frac{l_k}{2^{l_k}}$$

$$H(S) = \sum_{k=0}^{K-1} \left(\frac{1}{2^{l_k}} \right) \log_2(2^{l_k}) = \sum_{k=0}^{K-1} \left(\frac{l_k}{2^{l_k}} \right)$$

- Hence, $\bar{L} = H(S)$, meaning the code is perfectly matched to the source.

• Extended Codes for Non-Dyadic Sources

- **Bounds for Extended Code ($\bar{L}_n$):** When symbol probabilities are not negative powers of 2, we group n symbols together using an n^{th} -order extended code with average length $\bar{L}_n$:

$$nH(S) \leq \bar{L}_n < nH(S) + 1$$

- **Per-Symbol Average Length:** Dividing the bounds by n yields:

$$H(S) \leq \frac{\bar{L}_n}{n} < H(S) + \frac{1}{n}$$

- **Limit as Extension Order Increases:** Taking the limit as $n \rightarrow \infty$:

$$\lim_{n \rightarrow \infty} \frac{1}{n} \bar{L}_n = H(S)$$

By choosing a sufficiently large extension order n , the average length per symbol ($\frac{\bar{L}_n}{n}$) can be made as close to the source entropy $H(S)$ as desired, though this increases decoding complexity.

4.4 HUFFMAN CODING

Huffman coding is an optimal prefix coding algorithm for lossless data compression. It minimizes average code length by assigning shorter binary codes to more frequent symbols and longer codes to less frequent ones.

Encoding Algorithm Steps:

1. **Sort & Split:** List source symbols in order of decreasing probability. Assign 0 and 1 to the two lowest-probability symbols (Splitting stage).
2. **Combine:** Add the two lowest probabilities together to form a new combined symbol and re-insert it into the sorted list.
3. **Repeat:** Continue combining the two lowest probabilities until only two symbols remain (assigned 0 and 1).
4. **Trace Codewords:** Work backward from the final stage, tracing the assigned 0s and 1s to construct each symbol's codeword.

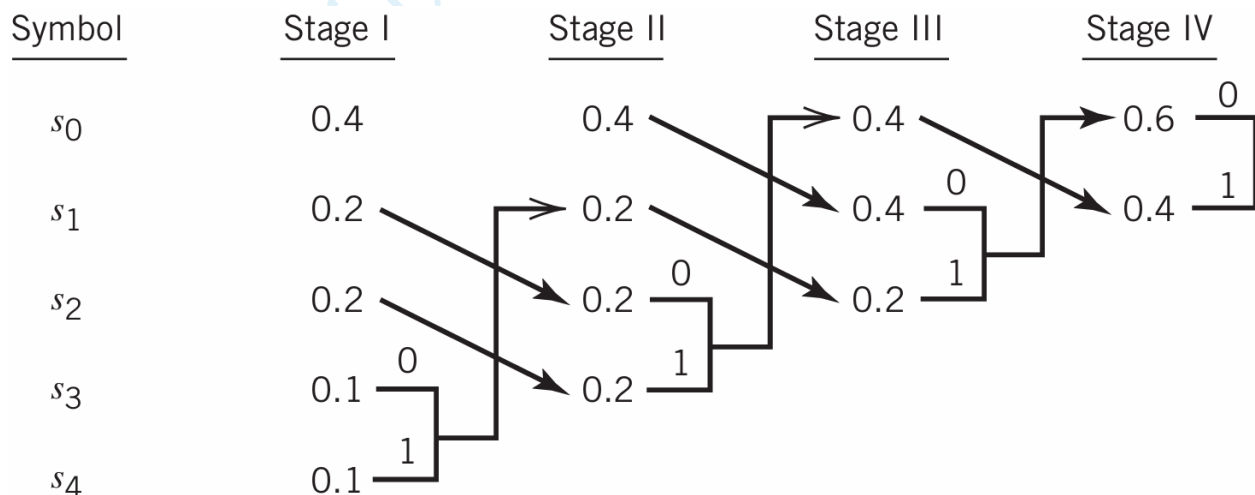
Example 4:

Given a discrete memoryless source with five symbols:

Symbol	s0	s1	s2	s3	s4
Probability	0.4	0.2	0.2	0.1	0.1

Compute the Huffman code (placing combined symbols as high as possible) and verify $\bar{L} \geq H(S)$.

Solution:



Resulting Source Code:

Symbol	Probability	Codeword	Length (l_k)
s_0	0.4	00	2
s_1	0.2	10	2
s_2	0.2	11	2
s_3	0.1	010	3
s_4	0.1	011	3

Average Codeword Length ($\bar{L}$):

$$\bar{L} = \sum_{k=0}^4 P_k l_k = 0.4(2) + 0.2(2) + 0.2(2) + 0.1(3) + 0.1(3) = \mathbf{2.2 \text{ binary symbols}}$$

Entropy $H(S)$:

$$\begin{aligned} H(S) &= 0.4 \log_2 \left(\frac{1}{0.4} \right) + 0.2 \log_2 \left(\frac{1}{0.2} \right) + 0.2 \log_2 \left(\frac{1}{0.2} \right) + 0.1 \log_2 \left(\frac{1}{0.1} \right) + 0.1 \log_2 \left(\frac{1}{0.1} \right) \\ &= \mathbf{2.121 \text{ bits}} \end{aligned}$$

Conclusion:

$$\bar{L} \geq H(S) \Rightarrow 2.2 \geq 2.121 \quad (\text{Verified})$$

Variance of Codeword Length:

The variance of average length $\bar{L}$ is defined as:

$$\sigma^2 = \sum_{k=0}^{K-1} P_k (l_k - \bar{L})^2$$

Moving combined symbols as high as possible during reduction minimizes variance (σ^2), producing more uniform codeword lengths across symbols.

Drawbacks of Huffman Coding:

1. **Requires Source Statistics:** Needs prior knowledge of the exact probability model of the source.
2. **Exponential Codebook Growth:** In text modeling, capturing higher-order relationships between words requires "super-symbols," causing the codebook size to grow exponentially.

Note: The **Lempel-Ziv algorithm** overcomes these limitations because it is adaptive and simpler to implement.

4.5 LEMPEL-ZIV CODING

- Lempel-Ziv coding is an adaptive, dictionary-based algorithm used for lossless data compression.
- The source data stream is parsed into segments that are the shortest subsequences not encountered previously.

Parsing Example:

Consider the binary sequence: 000101110010100101 ...

1. Initial Codebook: Assume symbols 0 and 1 are already stored at index positions 1 and 2.
 - Subsequences stored: 0, 1
 - Data to be parsed: 000101110010100101...
2. Step-by-Step Parsing:
 - 00: Shortest unseen subsequence → Add 00 to position 3.
Remaining stream: 01110010100101...
 - 01: Shortest unseen subsequence → Add 01 to position 4.
Remaining stream: 01110010100101...
 - 011: Shortest unseen subsequence → Add 011 to position 5.
Remaining stream: 10010100101...
 - Continue parsing until the sequence is fully parsed

Numerical positions	1	2	3	4	5	6	7	8	9
Subsequences	0	1	00	01	011	10	010	100	101
Numerical representations			11	12	42	21	41	61	62
Binary encoded blocks			0010	0011	1001	0100	1000	1100	1101

Numerical Representation Structure: Consists of a **[Pointer]** (index of the prefix subsequence already in the dictionary) followed by the **[Innovation Symbol]** (the new trailing bit).

- *Example:* For subsequence 01 at position 4:
- Prefix 0 is at position 1.
- Innovation symbol is 1.
- Numerical representation = 12.

Binary Encoded Block Construction: Combine the binary format of the pointer with the innovation symbol bit.

- *Example for 011 (Position 5):*
- Numerical representation is 42 (Prefix 01 at position 4 + innovation symbol 1).
- Binary representation of pointer 4 = 100.
- Innovation symbol = 1.
- Encoded Block = 1001.

Decoding Process:

The Lempel-Ziv decoder uses the binary pointer to identify the root subsequence and then appends the innovation symbol.

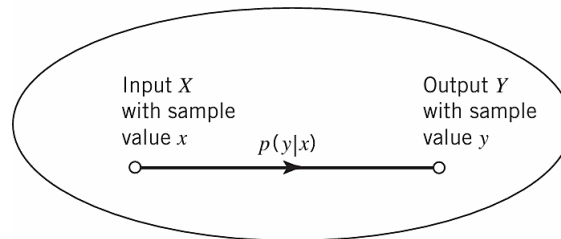
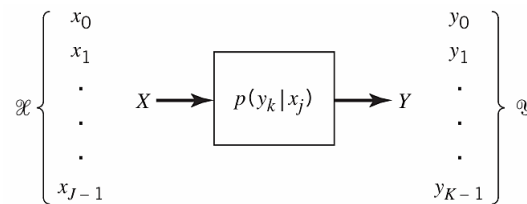
- **Decoding Example (Position 9 - Block 1101):**
 - The last bit 1 is the **innovation symbol**.
 - The remaining bits 110 represent pointer 6.
 - Pointer 6 refers to root subsequence 10.
 - Appending the innovation symbol 1 to root 10 yields the original subsequence **101**.

5. DISCRETE MEMORYLESS CHANNELS

- A discrete memoryless channel is a statistical model with an input $\mathbf{X}$ and an output $\mathbf{Y}$ that is a noisy version of $\mathbf{X}$; Both $\mathbf{X}$ & $\mathbf{Y}$ are random variables.
- Every unit of time, the channel accepts an input symbol $\mathbf{X}$ selected from an alphabet $\mathcal{X}$ and, in response, it emits an output symbol $\mathbf{Y}$ from an alphabet $\mathcal{Y}$.

$$\mathcal{X} = \{x_0, x_1, \dots, x_{J-1}\}$$

$$\mathcal{Y} = \{y_0, y_1, \dots, y_{K-1}\}$$



- The channel is said to be discrete when both of the alphabets $\mathcal{X}$ and $\mathcal{Y}$ have finite sizes.
- It is said to be "memoryless" when the current output symbol depends only on the current input symbol and not any previous or future symbol.
- The channel is characterized by a set of transition probabilities $p(y_k|x_j)$ for all j and k , for which according to probability theory:

$$0 \leq p(y_k|x_j) \leq 1 \quad \text{for all } j \text{ and } k$$

$$\sum_K p(y_k|x_j) = 1 \quad \text{for fixed } j$$

5.1 CHANNEL MATRIX & PROBABILITIES

- A convenient of describing a discrete memoryless channel is to arrange the various transition probabilities in the form of matrix:

$$\mathbf{P} = \begin{bmatrix} p(y_0|x_0) & p(y_1|x_0) & \dots & p(y_{K-1}|x_0) \\ p(y_0|x_1) & p(y_1|x_1) & \dots & p(y_{K-1}|x_1) \\ \vdots & \vdots & \ddots & \vdots \\ p(y_0|x_{J-1}) & p(y_1|x_{J-1}) & \dots & p(y_{K-1}|x_{J-1}) \end{bmatrix}$$

- The J -by- K matrix $\mathbf{P}$ is called the channel matrix, or stochastic matrix.
- Note that each row of the channel matrix $\mathbf{P}$ corresponds to a fixed channel input, whereas each column of the matrix corresponds to a fixed channel output.
- The probabilities of the input symbols is $p(x_j)$ for $j = 0, 1, \dots, J-1$.
- The joint probability distribution of the random variables $\mathbf{X}$ and $\mathbf{Y}$ is given by:

$$p(x_j, y_k) = p(y_k|x_j)p(x_j)$$
- The marginal probability distribution of the output random variable $\mathbf{Y}$ is obtained by averaging out the dependence of $p(x_j, y_k)$ on x_j , obtaining:

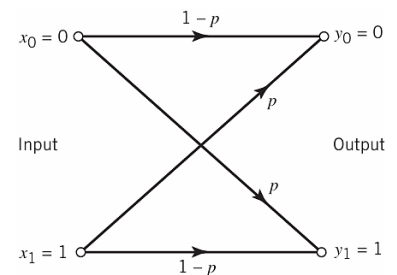
$$p(y_k) = \sum_{j=0}^{J-1} p(x_j, y_k)$$

$$p(y_k) = \sum_{j=0}^{J-1} p(y_k|x_j)p(x_j) \quad \text{for } k = 0, 1, \dots, K-1$$

5.2 EXAMPLE: BINARY SYMMETRIC CHANNEL

- The binary symmetric channel is a special case of the discrete memoryless channel with $J = K = 2$.
- The channel has two input symbols ($x_0 = 0, x_1 = 1$) and two output symbols ($y_0 = 0, y_1 = 1$).
- The channel is symmetric because the probability of receiving 1 if 0 is sent is same as the probability of receiving 0 if 1 is sent.
- This conditional probability of error is denoted by p .
- The stochastic / channel matrix is:

$$\mathbf{P} = \begin{bmatrix} 1-p & p \\ p & 1-p \end{bmatrix}$$



CONDITIONAL ENTROPY

- **Definition:** It is the average amount of uncertainty remaining about the channel input after the channel output has been observed.
- **Notation:** It is denoted by $H(X|Y)$.
- **Mathematical Formula (Conditional Probability Form):**

$$H(X|Y) = \sum_{k=0}^{K-1} \sum_{j=0}^{J-1} p(x_j|y_k) p(y_k) \log_2 \left(\frac{1}{p(x_j|y_k)} \right)$$

- **Mathematical Formula (Joint Probability Form):**

$$H(X|Y) = \sum_{k=0}^{K-1} \sum_{j=0}^{J-1} p(x_j, y_k) \log_2 \left(\frac{1}{p(x_j|y_k)} \right)$$

6. MUTUAL INFORMATION

- **Definition of $I(X; Y)$:** The mutual information $I(X; Y)$ is a measure of the uncertainty about the channel input, which is resolved by observing the output.

$$I(X; Y) = H(X) - H(X|Y) \rightarrow (2)$$

where,

- $H(X)$ – entropy of the channel input X .
 - $H(X|Y)$ – conditional entropy.
- **Definition of $I(Y; X)$:** The mutual information $I(Y; X)$ is a measure of the uncertainty about the channel output that is resolved by sending the channel input.

$$I(Y; X) = H(Y) - H(Y|X) \rightarrow (3)$$

6.1 PROPERTIES OF MUTUAL INFORMATION

Property 1: Symmetry

- **Statement:** The mutual information of a channel is symmetric in the sense that $I(X; Y) = I(Y; X)$.
- **Proof:** The entropy of the channel input X , can be written as:

$$\begin{aligned}
 H(X) &= \sum_{j=0}^{J-1} p(x_j) \log_2 \left(\frac{1}{p(x_j)} \right) \\
 H(X) &= \sum_{j=0}^{J-1} p(x_j) \log_2 \left(\frac{1}{p(x_j)} \right) \sum_{k=0}^{K-1} p(y_k | x_j) \\
 H(X) &= \sum_{j=0}^{J-1} \sum_{k=0}^{K-1} p(y_k | x_j) p(x_j) \log_2 \left(\frac{1}{p(x_j)} \right) \\
 H(X) &= \sum_{j=0}^{J-1} \sum_{k=0}^{K-1} p(x_j, y_k) \log_2 \left(\frac{1}{p(x_j)} \right) \rightarrow (4)
 \end{aligned}$$

- We can also write, by (1):

$$H(X|Y) = \sum_{j=0}^{J-1} \sum_{k=0}^{K-1} p(x_j, y_k) \log_2 \left(\frac{1}{p(x_j | y_k)} \right) \rightarrow (5)$$

- We know that:

$$I(X; Y) = H(X) - H(X|Y)$$

- Substituting (4) and (5) in above, we get:

$$I(X; Y) = \sum_{j=0}^{J-1} \sum_{k=0}^{K-1} p(x_j, y_k) \log_2 \left(\frac{p(x_j | y_k)}{p(x_j)} \right) \rightarrow (6)$$

- By Bayes' rule for conditional probabilities, we write:

$$\frac{p(x_j | y_k)}{p(x_j)} = \frac{p(y_k | x_j)}{p(y_k)} \rightarrow (7)$$

- By (7), (6) can be rewritten as:

$$I(X; Y) = \sum_{k=0}^{K-1} \sum_{j=0}^{J-1} p(x_j, y_k) \log_2 \left(\frac{p(y_k | x_j)}{p(y_k)} \right) \rightarrow (8)$$

$$I(X; Y) = I(Y; X)$$

which proves Property 1.

Property 2: Nonnegativity

- **Statement:** The mutual information is always non-negative, that is:

$$I(X; Y) \geq 0$$

- **Proof:**

- We know that:

$$p(x_j, y_k) = p(x_j|y_k)p(y_k)$$

we can rewrite:

$$p(x_j|y_k) = \frac{p(x_j, y_k)}{p(y_k)}$$

- Substituting above in (7), we get:

$$I(X; Y) = \sum_{j=0}^{J-1} \sum_{k=0}^{K-1} p(x_j, y_k) \log_2 \left(\frac{p(x_j, y_k)}{p(x_j)p(y_k)} \right) \rightarrow (9)$$

- For independent events:

$$p(x_j, y_k) = p(x_j)p(y_k) \rightarrow (10)$$

- Substituting (10) in (9), we get:

$$I(X; Y) = 0$$

Property 3: Expansion of the Mutual Information

- **Statement:** The mutual information of a channel is related to the joint entropy of the channel input and channel output by:

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

where the joint entropy $H(X, Y)$ is defined by:

$$H(X, Y) = \sum_{j=0}^{J-1} \sum_{k=0}^{K-1} p(x_j, y_k) \log_2 \left(\frac{1}{p(x_j, y_k)} \right) \rightarrow (11)$$

- **Proof:**

- The joint entropy, $H(X, Y)$ can be rewritten as:

$$H(X, Y) = \sum_{j=0}^{J-1} \sum_{k=0}^{K-1} p(x_j, y_k) \log_2 \left(\frac{p(x_j)p(y_k)}{p(x_j, y_k)} \right) + \sum_{j=0}^{J-1} \sum_{k=0}^{K-1} p(x_j, y_k) \log_2 \left(\frac{1}{p(x_j)p(y_k)} \right) \rightarrow (12)$$

- The first term on RHS of (12) is the negative of the mutual information of the channel, $I(X; Y)$.

- The second term on RHS of (12) can be re-written as:

$$\begin{aligned} \sum_{j=0}^{J-1} \sum_{k=0}^{K-1} p(x_j, y_k) \log_2 \left(\frac{1}{p(x_j)p(y_k)} \right) \\ = \sum_{j=0}^{J-1} \log_2 \left(\frac{1}{p(x_j)} \right) \sum_{k=0}^{K-1} p(x_j, y_k) + \sum_{k=0}^{K-1} \log_2 \left(\frac{1}{p(y_k)} \right) \sum_{j=0}^{J-1} p(x_j, y_k) \end{aligned}$$

- Where:

$$\sum_{j=0}^{J-1} p(x_j, y_k) = p(y_k) \quad \text{and} \quad \sum_{k=0}^{K-1} p(x_j, y_k) = p(x_j)$$

- We can rewrite:

$$\begin{aligned} \sum_{j=0}^{J-1} \sum_{k=0}^{K-1} p(x_j, y_k) \log_2 \left(\frac{1}{p(x_j)p(y_k)} \right) &= \sum_{j=0}^{J-1} p(x_j) \log_2 \left(\frac{1}{p(x_j)} \right) + \sum_{k=0}^{K-1} p(y_k) \log_2 \left(\frac{1}{p(y_k)} \right) \\ \sum_{j=0}^{J-1} \sum_{k=0}^{K-1} p(x_j, y_k) \log_2 \left(\frac{1}{p(x_j)p(y_k)} \right) &= H(X) + H(Y) \rightarrow (13) \end{aligned}$$

- Now, (12) becomes (Using (13)):

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

- Finally: $I(X; Y) = H(X) + H(Y) - H(X, Y)$

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

$H(X)$	$H(Y X)$
--------	----------

$H(X Y)$	$I(X; Y)$	$H(Y X)$
----------	-----------	----------

$H(X Y)$	$H(Y)$
----------	--------

7. CHANNEL CAPACITY

- **Definition:** The channel capacity of a discrete memoryless channel, commonly denoted by C , is defined as the maximum mutual information $I(X; Y)$ in any single use of the channel (i.e., signalling interval), where the maximization is over all possible input probability distribution $\{p(x_j)\}$ on X .

$$C = \max_{\{p(x_j)\}} I(X; Y) \text{ bits per channel use.} \rightarrow (14)$$

- **Constraints:** Equation (14) is subjected to two input probabilistic constraints:
 - $p(x_j) \geq 0$ for all j
 - $\sum_{j=0}^{J-1} p(x_j) = 1$

EXAMPLE 5

Problem Statement: Show that the capacity of the Binary Symmetric Channel is:

$$C = 1 - H(p)$$

Solution:

- Binary Symmetric Channel Model:

- Inputs: $x_0 = 0, x_1 = 1$
- Outputs: $y_0 = 0, y_1 = 1$
- Transition probabilities:
 - $p(y_1|x_0) = p(y_0|x_1) = p$
 - $p(y_0|x_0) = p(y_1|x_1) = 1 - p$

- Maximization Condition:

The mutual information $I(X; Y)$ is maximized when the channel input probabilities are

$$p(x_0) = p(x_1) = \frac{1}{2}.$$

$$C = I(X; Y)|_{p(x_0)=p(x_1)=1/2}$$

- Derivation:

$$C = \sum_{j=0}^{J-1} \sum_{k=0}^{K-1} p(x_j, y_k) \log_2 \left(\frac{p(y_k|x_j)}{p(y_k)} \right)$$

$$C = \sum_{j=0}^{J-1} \sum_{k=0}^{K-1} p(y_k|x_j) p(x_j) \log_2 \left(\frac{p(y_k|x_j)}{p(y_k)} \right)$$

Expanding the summation over all input/output pairs $(y_0, y_1$ and $x_0, x_1)$:

$$\begin{aligned} C = & p(y_0|x_0)p(x_0) \log_2 \left(\frac{p(y_0|x_0)}{p(y_0)} \right) + p(y_0|x_1)p(x_1) \log_2 \left(\frac{p(y_0|x_1)}{p(y_0)} \right) \\ & + p(y_1|x_0)p(x_0) \log_2 \left(\frac{p(y_1|x_0)}{p(y_1)} \right) + p(y_1|x_1)p(x_1) \log_2 \left(\frac{p(y_1|x_1)}{p(y_1)} \right) \end{aligned}$$

Substituting $p(x_0) = p(x_1) = \frac{1}{2}$ (which leads to $p(y_0) = p(y_1) = \frac{1}{2}$):

$$\begin{aligned} C = & (1-p) \left(\frac{1}{2} \right) \log_2 \left(\frac{1-p}{1/2} \right) + p \left(\frac{1}{2} \right) \log_2 \left(\frac{p}{1/2} \right) \\ & + p \left(\frac{1}{2} \right) \log_2 \left(\frac{p}{1/2} \right) + (1-p) \left(\frac{1}{2} \right) \log_2 \left(\frac{1-p}{1/2} \right) \end{aligned}$$

Combining like terms:

$$C = (1-p) \log_2(2(1-p)) + p \log_2(2p)$$

$$C = (1-p)[\log_2 2 + \log_2(1-p)] + p[\log_2 2 + \log_2 p]$$

$$C = (1-p)[1 + \log_2(1-p)] + p[1 + \log_2 p]$$

$$C = (1-p) + (1-p) \log_2(1-p) + p + p \log_2 p$$

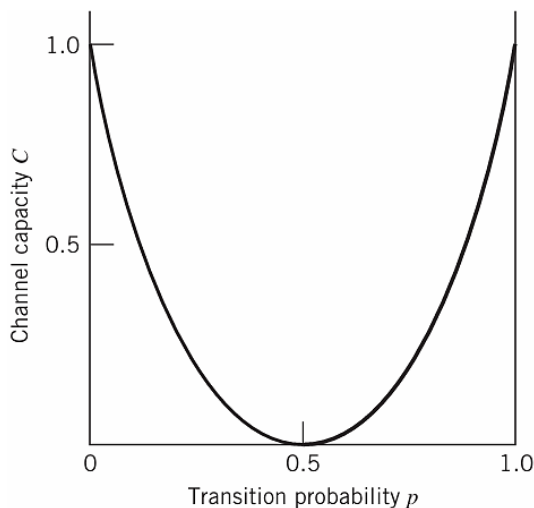
$$C = 1 + p \log_2 p + (1-p) \log_2(1-p)$$

Since $H(p) = -p \log_2 p - (1-p) \log_2(1-p)$,

we get: $C = 1 - H(p)$

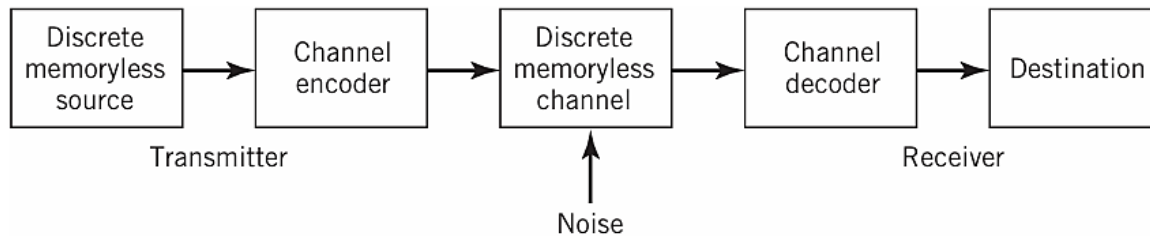
Key Observations & Graph:

- When the channel is noise-free ($p = 0$), the channel capacity C attains its maximum value of 1 bit per channel use, and $H(p)$ attains its minimum value of 0.
- Similarly, for $p = 1/2$, $C = 0$ and $H(p) = 1$.
- Capacity Curve: Plotting channel capacity C against transition probability p produces a symmetric V-like curve starting at $C = 1$ for $p = 0$, dropping to $C = 0$ at $p = 0.5$, and returning to $C = 1$ at $p = 1$.



8. CHANNEL CODING THEOREM

Shannon's second theorem, the channel-coding theorem, deals with improving communication reliability despite noise in the channel.



Function:

- Channel coding consists of mapping the incoming data sequence into a channel input sequence and inverse mapping the channel output sequence into an output data sequence such that the overall effect of noise on the system is minimized.
- The first mapping operation is performed in the transmitter by a channel encoder, whereas the inverse mapping operation is performed in the receiver by a channel decoder.
- In this process, redundancy is deliberately added to the transmitted data, which helps to recover the original data at the receiver even in the presence of errors.
- Note: Source coding reduces redundancy for efficiency, while channel coding increases redundancy for reliability.

8.1 STATEMENT OF SHANNON'S 2ND THEOREM (CHANNEL-CODING THEOREM)

It is in two parts as follows:

1. Direct Theorem:

Let a discrete memoryless source with an alphabet $\mathcal{S}$ have entropy $H(\mathcal{S})$ for random variable $\mathcal{S}$ and produce symbols once every T_s seconds. Let a discrete memoryless channel have capacity C and be used once every T_c seconds. Then, if:

$$\frac{H(\mathcal{S})}{T_s} \leq \frac{C}{T_c}$$

there exists a coding scheme for which the source output can be transmitted over the channel and be reconstructed with an arbitrarily small probability of error, where $\frac{C}{T_c}$ is called the critical rate.

2. Converse Theorem:

Conversely, if:

$$\frac{H(S)}{T_s} > \frac{C}{T_c}$$

it is not possible to transmit information over the channel and reconstruct it with an arbitrarily small probability of error.

APPLICATION OF THE CHANNEL-CODING THEOREM TO BINARY SYMMETRIC CHANNELS

- Consider a discrete memoryless source that emits equally likely binary symbols (**0s & 1s**) once every signalling interval, T_s seconds, with source entropy, $H(S)$ equal to unity ($H(S) = 1$).
- The channel-coding theorem implies that:

$$\frac{1}{T_s} \leq \frac{C}{T_c}$$

then the probability of error can be made low by the use of a suitable channel-encoding scheme.

- Defining code rate of the channel encoder as $r = \frac{T_c}{T_s}$, we may restate the condition simply as:

$$r \leq C$$

- That is, for $r \leq C$, there exists a code capable of achieving an arbitrarily low probability of error.

9. INFORMATION CAPACITY LAW (STATEMENT ONLY)

- Statement: The information capacity of a continuous channel of bandwidth B Hz, perturbed by AWGN (Additive White Gaussian Noise) of power spectral density $N_0/2$ and limited in bandwidth to B , is given by the formula:

$$C = B \log_2 \left(1 + \frac{P}{N_0 B} \right) \text{ bits/second}$$

where P is the average transmitted power.