

كلية الصفوة الجامعة قسم هندسة تقنيات الحاسوب

أسم المادة: نظرية المعلومات و الترميز المرحلة: الرابعة

أسم التدريسي: نور يحيى تسلسل المحاضرة: 1



الملاحظات:

Information Theory & Coding

Information theory provides a *quantitative measure of the information contained in message signals* and allows us to *determine the capacity of a communication system to transfer this information from source to destination*. Through the use of *coding*, a major topic of information theory, *redundancy can be reduced* from message signals so that channels can be used with improved efficiency. In addition, *systematic redundancy can be introduced* to the transmitted signal so that channels can be used with improved reliability.

Information theory attempts to analyse communication between a transmitter and a receiver through an unreliable channel, and in this approach performs, on the one hand, an analysis of information sources, especially the amount of information produced by a given source, and, on the other hand, states the conditions for performing reliable transmission through an unreliable channel. There are three main concepts in this theory:

1. The first one is the definition of a quantity that can be a valid measurement of information, which should be consistent with a physical understanding of its properties.
2. The second concept deals with the relationship between the information and the source that generates it. This concept will be referred to as source information. Well-known information theory techniques like compression and encryption are related to this concept.
3. The third concept deals with the relationship between the information and the unreliable channel through which it is going to be transmitted. This concept leads to the definition of a very important parameter called the channel capacity. A well-known information theory technique called error-correction coding is closely related to this concept.

Figure 1.1 illustrates the relationship of information theory to other fields. As the figure suggests, information theory intersects physics (statistical mechanics), mathematics (probability theory), electrical engineering (communication theory) and computer science (algorithmic complexity).

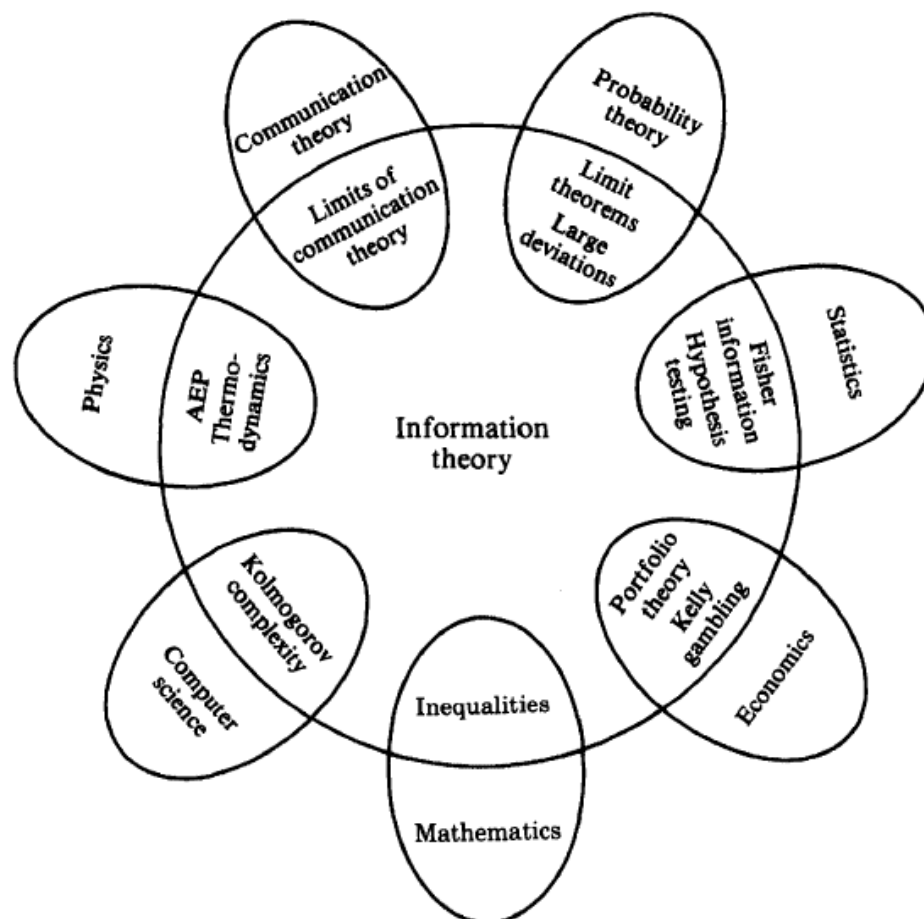
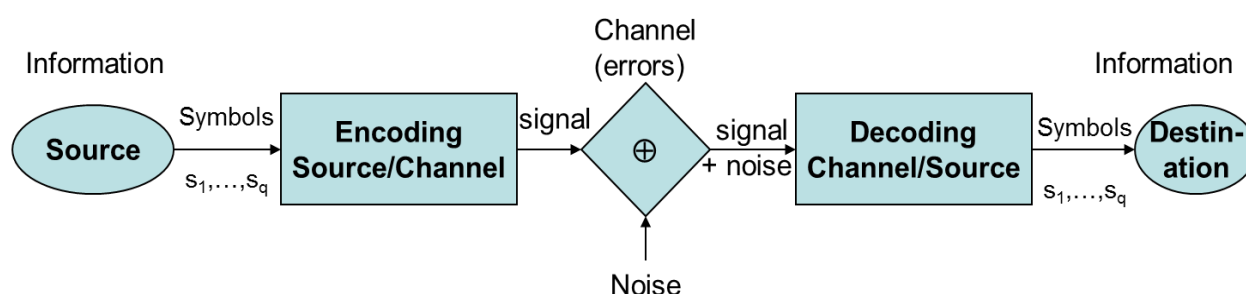


Figure 1.1. The relationship of information theory with other fields.

Digital Communications Model

In the transfer of digital information, the following framework is often used:



- The source is an object that produces an event, the outcome of which is selected at random according to a probability distribution. A

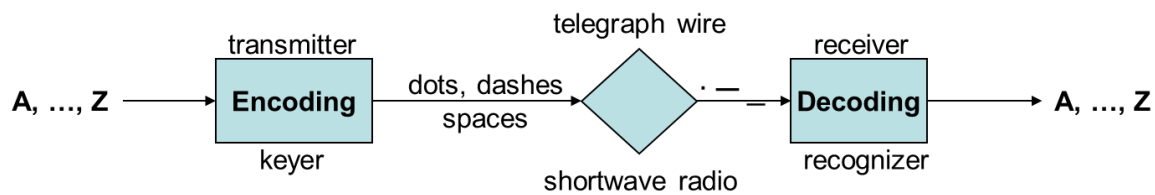
practical source in a communication system is a device that produces messages, and it can be either analog or discrete. A discrete information source is a source that has only a finite set of symbols as possible outputs. The set of source symbols is called the *source alphabet*, and the elements of the set are called *symbols* or *letters*. Information sources can be classified as having *memory* or being *memoryless*. A source with memory is one for which a current symbol depends on the previous symbols. A memoryless source is one for which each symbol produces is independent of the previous symbols. A *discrete memoryless source (DMS)* can be characterized by the *list of the symbols*, the *probability assignment to these symbols*, and the specification of the *rate of generating these symbols by the source*.

- The source encoder serves the purpose of removing as much redundancy as possible from the data. This is the *data compression* portion.
- The channel coder puts a modest amount of redundancy back in order to do error detection or correction.
- The channel is what the data passes through, possibly becoming corrupted along the way. There are a variety of channels of interest, including:
 - The magnetic recording channel
 - The telephone channel
 - Other band limited channels
 - The multi-user channel
 - Deep-space channels
 - Fading and/or jamming and/or interference channels
- The channel decoder performs error correction or detection
- The source decoder undoes what is necessary to get the data back.

There are also other possible blocks that could be inserted into this

model like *encryption/decryption* and *modulation/demodulation* block.

Example: Morse Code



Example: ASCII Code



كلية الصفوة الجامعة قسم هندسة تقنيات الحاسوب

أسم المادة: نظرية المعلومات و الترميز

المرحلة: الرابعة

أسم التدريسي: نور يحيى

تسلسل المحاضرة: 2



الملاحظات:

Probability

Probability: *How **likely** something is to happen.*

Many events can't be predicted with total certainty. The best we can say is how **likely** they are to happen, using the idea of probability.

Tossing a Coin

When a coin is tossed, there are two possible outcomes:

- heads (H) or
- tails (T)



We say that the probability of the coin landing **H** is $\frac{1}{2}$.

And the probability of the coin landing **T** is $\frac{1}{2}$

Throwing Dice

When a single [die](#) is thrown, there are six possible outcomes: 1, 2, 3, 4, 5, 6.



The probability of any one of them is $1/6$

[illegible]

Probability

In general:

$$\text{Probability of an event happening} = \frac{\text{Number of ways it can happen}}{\text{Total number of outcomes}}$$

Example: the chances of rolling a "4" with a die

Number of ways it can happen: 1 (there is only 1 face with a "4" on it)

Total number of outcomes: 6 (there are 6 faces altogether)

$$\text{So the probability} = \frac{1}{6}$$

Example: there are 5 marbles in a bag: 4 are blue, and 1 is red. What is the probability that a blue marble gets picked?

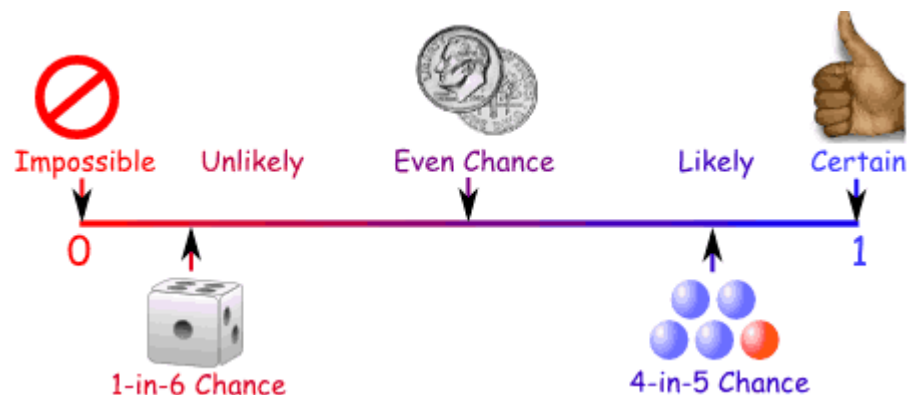
Number of ways it can happen: 4 (there are 4 blues)

Total number of outcomes: 5 (there are 5 marbles in total)

$$\text{So the probability} = \frac{4}{5} = 0.8$$

Probability Line

Probability is the **chance** that something will happen. It can be shown on a line.

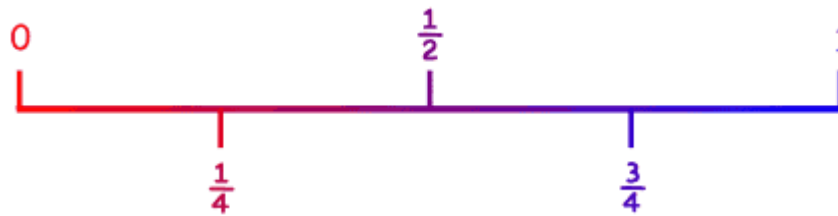


The probability of an event occurring is somewhere between impossible and certain.

As well as words we can use numbers (such as fractions or decimals) to show the probability of something happening:

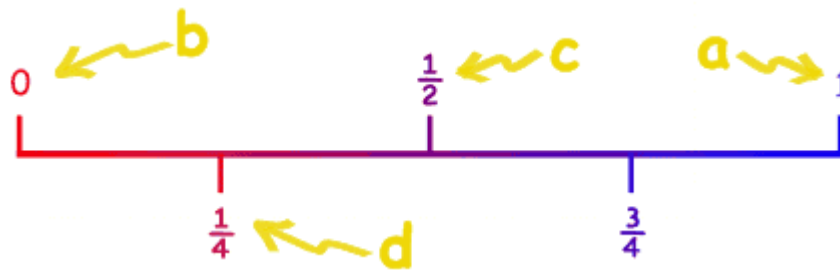
- Impossible is **zero**
- Certain is **one**.

Here are some fractions on the probability line:



We can also show the chance that something will happen:

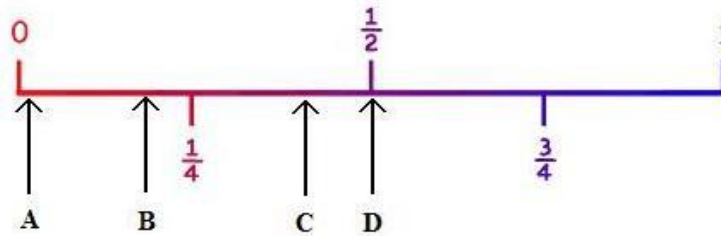
- The sun will rise tomorrow.
- I will not have to learn mathematics at school.
- If I flip a coin it will land heads up.
- Choosing a red ball from a sack with 1 red ball and 3 green balls



Between 0 and 1

- The probability of an event will **not** be less than 0.
This is because 0 is impossible (sure that something will not happen).
- The probability of an event will **not** be more than 1.
This is because 1 is certain that something will happen.

Questions??????



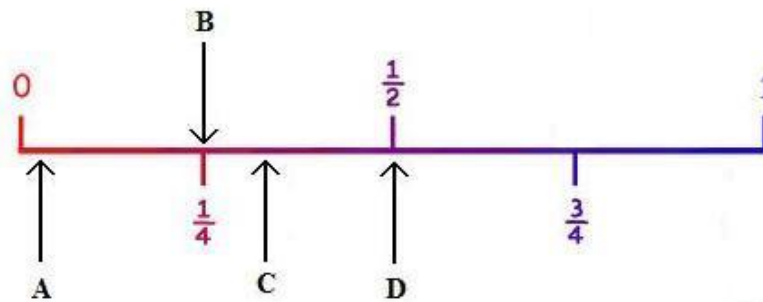
Which of the arrows A, B, C or D shows the best position on the probability line for the event 'Tomorrow it will snow in Karbala'?

A A

B B

C C

D D



A name is chosen at random from the telephone book. Which of the arrows A, B, C or D shows the best position on the probability line for the event 'The name begins with Z'?

A A

B B

C C

D D

Example: toss a coin 100 times, how many Heads will come up?

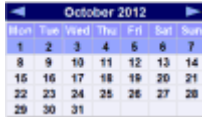
Probability says that heads have a $\frac{1}{2}$ chance, so we can **expect 50 Heads**.

But when we actually try it we might get 48 heads, or 55 heads ... or anything really, but in most cases it will be a number near 50.

Complement of an Event: All outcomes that are **NOT** the event.



When the event is **Heads**, the complement is **Tails**



When the event is {**Monday, Wednesday**} the complement is {**Tuesday, Thursday, Friday, Saturday, Sunday**}



When the event is {**Hearts**} the complement is {**Spades, Clubs, Diamonds, Jokers**}

So the Complement of an event is all the **other** outcomes (**not** the ones we want). And together the Event and its Complement make all possible outcomes.

The probability of an event is shown using "P":

P(A) means "Probability of Event A"

The complement is shown by a little mark after the letter such as **A'** (or sometimes **A^c** or **A**):

P(A') means "Probability of the complement of Event A"

The two probabilities always add to 1

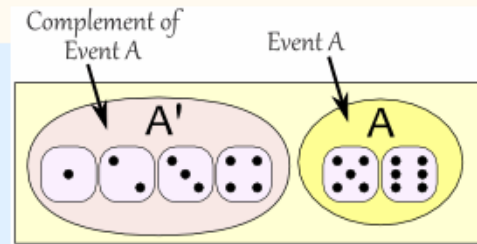
$$P(A) + P(A') = 1$$

Example: Rolling a "5" or "6"**Event A** is {5, 6}

Number of ways it can happen: 2

Total number of outcomes: 6

$$P(A) = \frac{2}{6} = \frac{1}{3}$$

The **Complement of Event A** is {1, 2, 3, 4}

Number of ways it can happen: 4

Total number of outcomes: 6

$$P(A') = \frac{4}{6} = \frac{2}{3}$$

Let us add them:

$$P(A) + P(A') = \frac{1}{3} + \frac{2}{3} = \frac{3}{3} = 1$$

Yep, that makes 1

It makes sense, right? **Event A** plus all outcomes that are **not Event A** make up all possible outcomes.

Why is the Complement Useful?

It is sometimes easier to work out the complement first.

Example. Throw two dice. What is the probability the two scores are different?Different scores are like getting a **2 and 3**, or a **6 and 1**. It is quite a long list:

$$A = \{ (1,2), (1,3), (1,4), (1,5), (1,6), (2,1), (2,3), (2,4), \dots \text{etc !} \}$$



But the complement (which is when the two scores are the same) is only **6 outcomes**:

$$A' = \{ (1,1), (2,2), (3,3), (4,4), (5,5), (6,6) \}$$

And its probability is:

$$P(A') = 6/36 = \mathbf{1/6}$$

Knowing that $P(A)$ and $P(A')$ together make 1, we can calculate:

$$\begin{aligned} P(A) &= 1 - P(A') \\ &= 1 - 1/6 \\ &= \mathbf{5/6} \end{aligned}$$

So in this case (and many others) it's easier to work out $P(A')$ first, then find $P(A)$

[illegible]

Probability: Types of Events

Life is full of random events!

You need to get a "feel" for them to be a smart and successful person.

The toss of a coin, throw of a dice and lottery draws are all examples of random events

Events : When we say "Event" we mean one (or more) outcomes.

Example Events:

- Getting a Tail when tossing a coin is an event
- Rolling a "5" is an event.

An event can include several outcomes:

- Choosing a "King" from a deck of cards (any of the 4 Kings) **is also** an event
- Rolling an "even number" (2, 4 or 6) is an event

Events can be:

- **Independent** (each event is **not** affected by other events),
- **Dependent** (also called "Conditional", where an event **is** affected by other events)
- **Mutually Exclusive** (events can't happen at the same time)

Let's look at each of those types.

Probability: Independent Events

Life is full of random events!

You need to get a "feel" for them to be a smart and successful person.

The toss of a coin, throwing dice and lottery draws are all examples of random events. Sometimes an event can affect the next event.

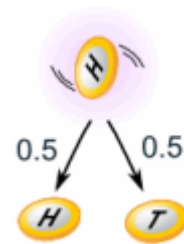
Example: taking colored marbles from a bag: as you take each marble there are less marbles left in the bag, so the probabilities change.

Independent Events are **not affected** by previous events.

This is an important idea!

A coin does not "know" it came up heads before.

And each toss of a coin is a perfect isolated thing.



Example: You toss a coin and it comes up "Heads" three times ... what is the chance that **the next toss** will also be a "Head"?

The chance is simply $\frac{1}{2}$ (or 0.5) just like **ANY** toss of the coin.

What it did in the past will not affect the current toss!

Some people think "it is overdue for a Tail", but *really truly* the next toss of the coin is totally independent of any previous tosses.

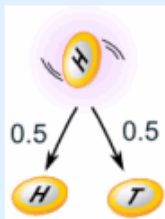
Saying "a Tail is due", or "just one more go, my luck is due" is called **The Gambler's Fallacy**

Of course your luck **may** change, because each toss of the coin has an equal chance.

Probability of Independent Events

"Probability" (or "Chance") is **how likely** something is to happen. So how do we calculate probability?

$$\text{Probability of an event happening} = \frac{\text{Number of ways it can happen}}{\text{Total number of outcomes}}$$



Example: what is the probability of getting a "Head" when tossing a coin?

Number of ways it can happen: 1 (Head)

Total number of outcomes: 2 (Head and Tail)

$$\text{So the probability} = \frac{1}{2} = 0.5$$

Example: what is the probability of getting a "4" or "6" when rolling a die?

Number of ways it can happen: 2 ("4" and "6")

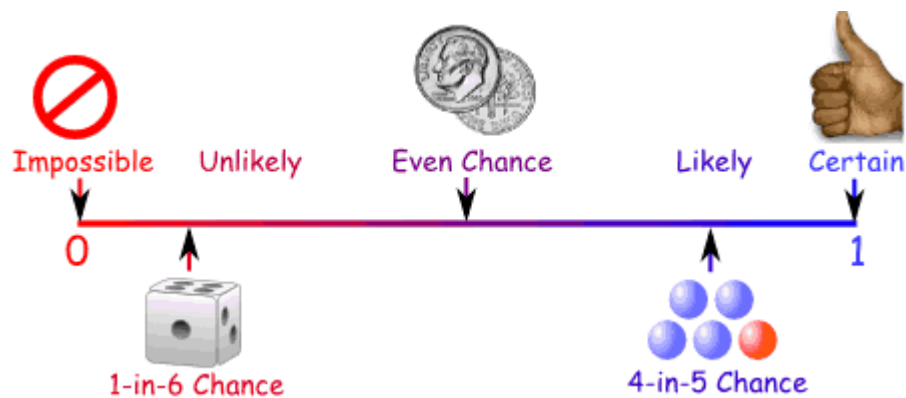
Total number of outcomes: 6 ("1", "2", "3", "4", "5" and "6")

$$\text{So the probability} = \frac{2}{6} = \frac{1}{3} = 0.333\dots$$



Ways of Showing Probability

Probability goes from **0** (impossible) to **1** (certain):



It is often shown as a **decimal** or **fraction**.

Example: the probability of getting a "Head" when tossing a coin:

- As a decimal: **0.5**
- As a fraction: **1/2**
- As a percentage: **50%**
- Or sometimes like this: **1-in-2**

Two or More Events

We can calculate the chances of two or more **independent** events by **multiplying** the chances

Example: Probability of 3 Heads in a Row

For each toss of a coin a "Head" has a probability of 0.5:

$$\begin{array}{c} \text{H} \\ 0.5 \end{array}$$

$$\begin{array}{c} \text{H} \quad \text{H} \\ 0.5 \times 0.5 = 0.25 \quad \left(\text{or } \frac{1}{2} \times \frac{1}{2} = \frac{1}{4}\right) \end{array}$$

$$\begin{array}{c} \text{H} \quad \text{H} \quad \text{H} \\ 0.5 \times 0.5 \times 0.5 = 0.125 \quad \left(\text{or } \frac{1}{2} \times \frac{1}{2} \times \frac{1}{2} = \frac{1}{8}\right) \end{array}$$

And so the chance of getting 3 Heads in a row is **0.125**

So each toss of a coin has a $\frac{1}{2}$ chance of being Heads, but **lots of Heads in a row** is unlikely.

Example: Why is it unlikely to get, say, 7 heads in a row, when *each* toss of a coin has a $\frac{1}{2}$ chance of being Heads?

Because we are asking two different questions:

Question 1: What is the probability of 7 heads in a row?

→ Answer: $\frac{1}{2} \times \frac{1}{2} \times \frac{1}{2} \times \frac{1}{2} \times \frac{1}{2} \times \frac{1}{2} \times \frac{1}{2} = 0.0078125$ (less than 1%).

Question 2: Given that **we have just got 6 heads** in a row, what is the probability that **the next toss** is also a head?

→ Answer: $\frac{1}{2}$, as the **previous** tosses don't affect the next toss.

Notation

We use "P" to mean "Probability Of",

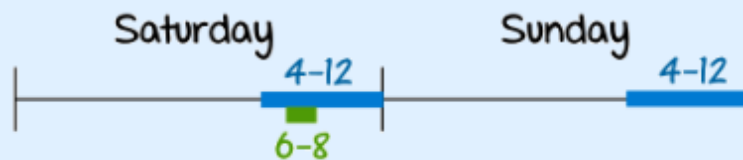
So, for Independent Events:

$$P(A \text{ and } B) = P(A) \times P(B)$$

Probability of A and B equals the probability of A times the probability of B

Example: your friend invites you to a movie, saying it starts some time on the weekend between 4 in the afternoon and midnight, but won't say more.

What are the chances it starts on Saturday between 6 and 8 at night?



Day: there are two days on the weekend, so **$P(\text{Saturday}) = 0.5$**

Time: between 4 and midnight is 8 hours, but you want between 6 and 8 which is only 2 hours:

$$P(\text{Your Time}) = 2/8 = 0.25$$

And:

$$\begin{aligned} P(\text{Saturday and Your Time}) &= P(\text{Saturday}) \times P(\text{Your Time}) \\ &= 0.5 \times 0.25 \\ &= \mathbf{0.125} \end{aligned}$$

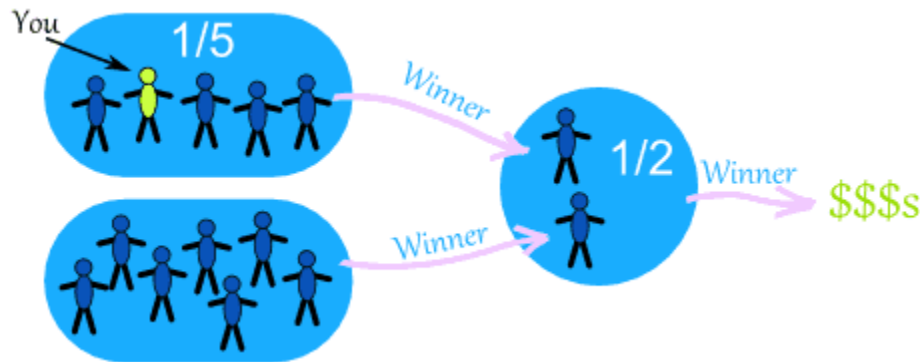
Or a 12.5% Chance

(Note: we could ALSO have worked out that you wanted 2 hours out of a total possible 16 hours, which is $2/16 = 0.125$. Both methods work here.)

Another Example

Imagine there are two groups:

- A member of each group gets randomly chosen for the winners circle,
- **then** one of those gets randomly chosen to get the big money prize:



What is your chance of winning the big prize?

- there is a **1/5 chance** of going to the winners circle
- and a **1/2 chance** of winning the big prize

So you have a 1/5 chance followed by a 1/2 chance ... which makes a 1/10 chance overall:

$$1/5 \times 1/2 = 1/10$$

Or we can calculate using decimals (1/5 is 0.2, and 1/2 is 0.5):

$$0.2 \times 0.5 = \mathbf{0.1}$$

So your chance of winning the big money is **0.1** (which is the same as 1/10).

كلية الصفوة الجامعة

قسم هندسة تقنيات الحاسوب

المرحلة: الرابعة

أسم المادة: نظرية المعلومات و الترميز

تسلسل المحاضرة: 3

أسم التدريسي: م.م. نور يحيى



الملاحظات:

Chapter One

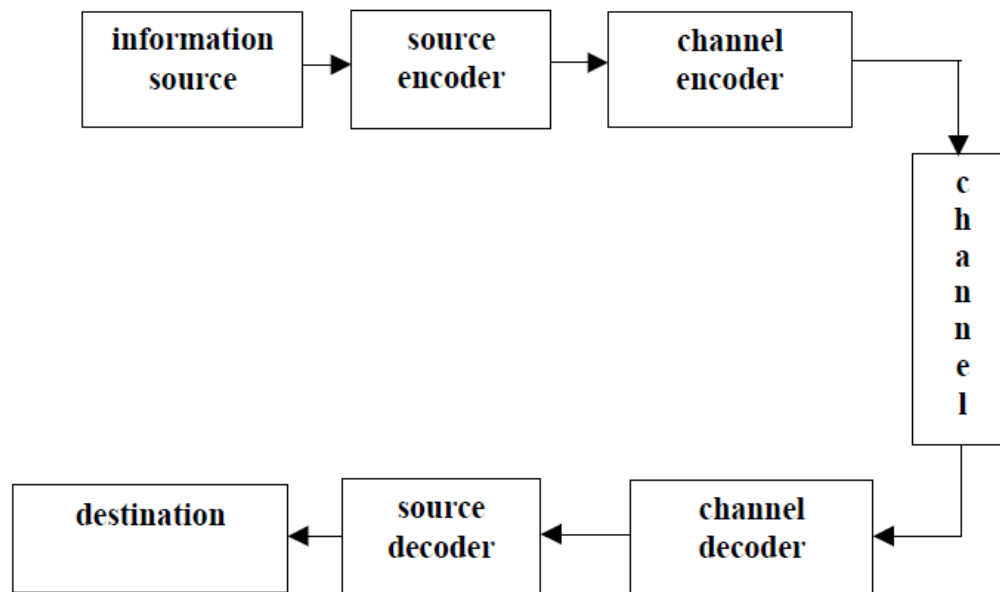
1. Introduction:

Most scientists agree that information theory began in 1948 with Shannon's famous article. In that paper, he provided answers to the following questions:

- What is “information” and how to measure it?
- What are the fundamental limits on the storage and the transmission of information?

Shannon Paradigm:

Transmitting a message from a transmitter to a receiver can be sketched as follows:



The components of information system as described by Shannon are:

1. An information source is a device which randomly delivers symbols from an alphabet. As an example, a PC (Personal Computer) connected to internet is an information source which produces binary digits from the binary alphabet $\{0, 1\}$.
2. A channel is a system which links a transmitter to a receiver. It includes signaling equipment and pair of copper wires or coaxial cable or optical fiber, among other possibilities.
3. A source encoder allows one to represent the data source more compactly by eliminating redundancy: it aims to reduce the data rate.

A channel encoder adds redundancy to protect the transmitted signal against transmission errors

2- Self- information:

In information theory, **self-information** is a measure of the information content associated with the *outcome* of a random variable. It is expressed in a unit of information, for example bits, nats, or hartleys, depending on the base of the logarithm used in its calculation.

A **bit** is the basic unit of information in computing and digital communications. A bit can have only one of two values, and may therefore be physically implemented with a two-state device. These values are most commonly represented as 0 and 1.

A **nat** is the **natural unit of information**, sometimes also **nit** or **nepit**, is a unit of information or entropy, based on natural logarithms and powers of e , rather than the powers of 2 and base 2 logarithms which define the bit. This unit is also known by its unit symbol, the nat.

The **hartley** (symbol **Hart**) is a unit of information defined by International Standard IEC 80000-13 of the International Electrotechnical Commission. One hartley is the information content of an event if the probability of that event occurring is $1/10$. It is therefore equal to the information contained in one decimal digit (or dit).

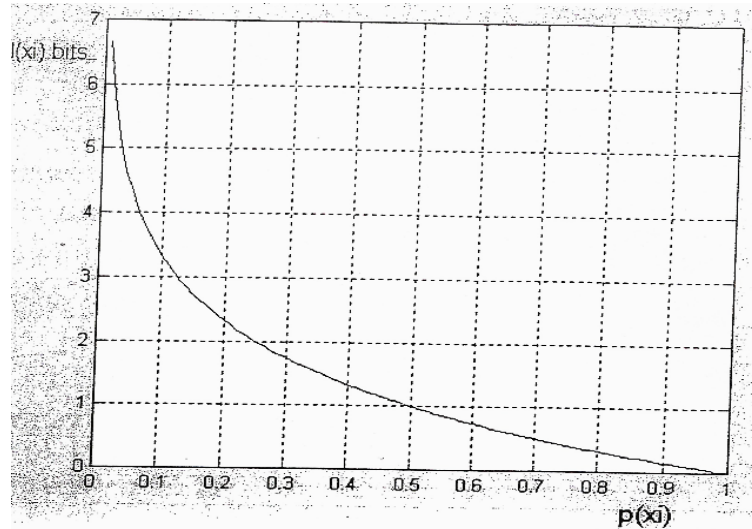
$$1 \text{ Hart} \approx 3.322 \text{ Sh} \approx 2.303 \text{ nat}.$$

The amount of self-information contained in a probabilistic event depends only on the probability of that event: the smaller its probability, the larger the self-information associated with receiving the information that the event indeed occurred.

Suppose that the source of information produces finite set of message x_1, x_2, \dots, x_n with prob. $p(x_1), p(x_2), \dots, P(x_n)$ and such that

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

- 1- Information is zero if $P(x_i) = 1$ (certain event)
- 2- Information increase as $P(x_i)$ decrease to zero
- 3- Information is a +ve quantity



The log function satisfies all previous three points hence:

$$I(x_i) = -\log_a P(x_i)$$

Where $I(x_i)$ is self information of (x_i) and if:

- i- If “a” = 2 , then $I(x_i)$ has the unit of bits
- ii- If “a” = e = 2.71828, then $I(x_i)$ has the unit of nats
- iii- If “a” = 10, then $I(x_i)$ has the unit of hartly

$$\text{Recall that } \log_a x = \frac{\ln x}{\ln a}$$

Example 1:

A fair die is thrown, find the amount of information gained if you are told that 4 will appear.

Solution:

$$P(1) = P(2) = \dots = P(6) = \frac{1}{6}$$

$$I(4) = -\log_2\left(\frac{1}{6}\right) = \frac{\ln\left(\frac{1}{6}\right)}{\ln 2} = 2.5849 \text{ bits}$$

Example 2:

A biased coin has $P(\text{Head})=0.3$. Find the amount of information gained if you are told that a tail will appear.

Solution:

$$P(\text{tail}) = 1 - P(\text{Head}) = 1 - 0.3 = 0.7$$

$$I(\text{tail}) = -\log_2(0.7) = -\frac{\ln 0.7}{\ln 2} = 0.5145 \text{ bits}$$

3.Probability: A probabilistic model is a mathematical description of an uncertain situation. A probability of an event A: If an experiment has A_1, A_2, \dots, A_n , outcomes, then:

$$Prob(A_i) = P(A_i) = \lim_{N \rightarrow \infty} \frac{n(A_i)}{N}$$

Where $n(A_i)$ = no. of times event (outcomes) (A_i) occurs

N = total number of trials.

Not that

$$1 \geq P(A_i) \geq 0, \quad \text{and}$$

$$\sum_{i=1}^n P(A_i) = 1$$

If $P(A_i) = 1$ then A_i is certain event

When the sample space Ω has a finite number of equally likely outcomes, so that the discrete uniform probability law applies. Then, the probability of any event A is given by

$$P(A) = \frac{\text{Number of elements of } A}{\text{Number of elements of } \Omega}$$

4- Independent and dependent Events

Events can be "**Independent**", meaning each event is **not affected** by any other events. For example tossing a coin each toss of a coin is a perfect isolated. But events can also be "dependent" ... which means they **can be affected by previous events**. For example: Marbles in a Bag 2 blue and 3 red marbles are in a bag. What are the chances of getting a blue marble? The chance is **2 in 5**. **But after taking one out** the chances change. So the next time, if we got a **red** marble before, then the chance of a blue marble next is **2 in 4**, if we got a **blue** marble before, then the chance of a blue marble next is **1 in 4**.

Example: What are the chances of drawing 2 blue marbles from a group of 2 blue and 3 red marbles?

Solution:

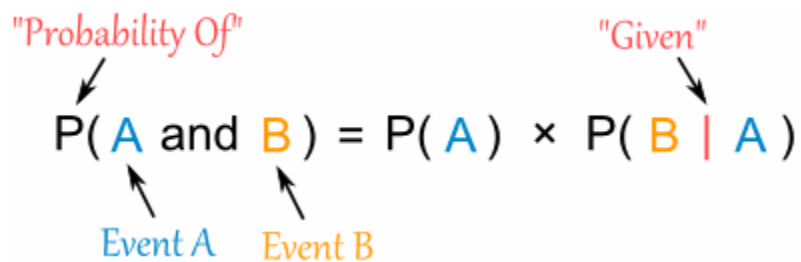
It is a **2/5 chance** followed by a **1/4 chance**:

$$\frac{2}{5} \times \frac{1}{4} = \frac{2}{20} = \frac{1}{10}$$

5- Conditional Probability

It is happened when there are dependent events. We have to use the symbol "|" to mean "given":

- $P(B|A)$ means "Event B **given** Event A has occurred".
- $P(B|A)$ is also called the "Conditional Probability" of B given A has occurred .
- And we write it as



$$P(\text{A and B}) = P(\text{A}) \times P(\text{B} | \text{A})$$

$$P(A | B) = \frac{\text{number of elements of A and B}}{\text{number of elements of B}}$$

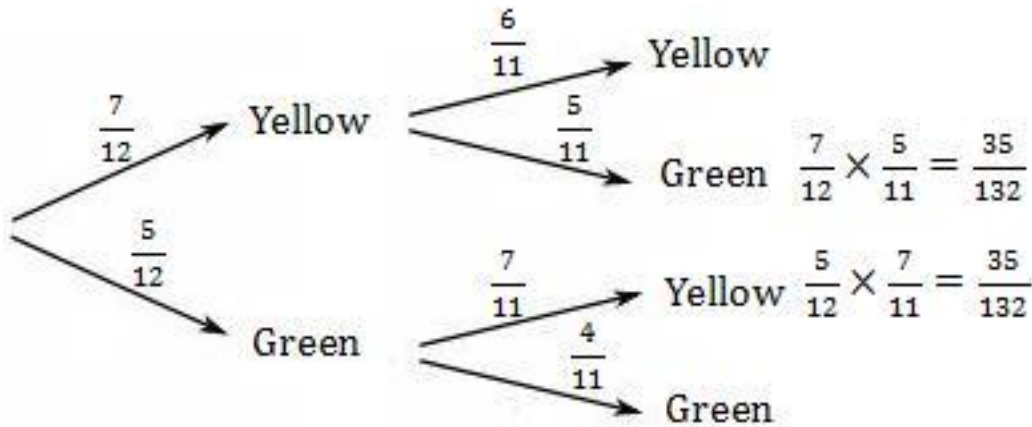
Or

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

Where $P(B) > 0$

Example: A box contains 5 green pencils and 7 yellow pencils. Two pencils are chosen at random from the box without replacement. What is the probability they are different colors?

Solution: Using a tree diagram:



Example: We toss a fair coin three successive times. We wish to find the conditional probability $P(A | B)$ when A and B are the events

$A = \{\text{more heads than tails come up}\}$, $B = \{\text{1st toss is a head}\}$.

The sample space consists of eight sequences,

$\Omega = \{HHH, HHT, HTH, HTT, THH, THT, TTH, TTT\}$,

$$P(B) = \frac{4}{8}$$

$$P(A \cap B) = \frac{3}{8}$$

$$P(A | B) = \frac{P(A \cap B)}{P(B)} = \frac{\frac{3}{8}}{\frac{4}{8}} = \frac{3}{4}$$

Bayes' Rule: Let A_1, A_2, \dots, A_n be disjoint events that form a partition of the sample space, and assume that $P(A_i) > 0$, for all i . Then, for any event B such that $P(B) > 0$, we have

$$P(A_i | B) = \frac{P(A_i)P(B|A_i)}{P(B)}$$

$$= \frac{P(A_i)P(B|A_i)}{P(A_1)P(B|A_1) + \dots + P(A_n)P(B|A_n)}$$

Entropy

In information theory, **entropy** is the average amount of information contained in each message received. Here, *message* stands for an event, sample or character drawn from a distribution or data stream. Entropy thus characterizes our uncertainty about our source of information.

Source Entropy:

If the source produces not equiprobable messages then $I(x_i), i = 1, 2, \dots, n$ are different. Then the statistical average of $I(x_i)$ over i will give the average amount of uncertainty associated with source X . This average is called source entropy and denoted by $H(X)$, given by:

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

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

Example:

Find the entropy of the source producing the following messages:

$$Px_1 = 0.25, \quad Px_2 = 0.1, \quad Px_3 = 0.15, \quad \text{and} \quad Px_4 = 0.5$$

Solution:

$$\begin{aligned} H(X) &= - \sum_{i=1}^n P(x_i) \log_a P(x_i) \\ &= - \frac{[0.25 \ln 0.25 + 0.1 \ln 0.1 + 0.15 \ln 0.15 + 0.5 \ln 0.5]}{\ln 2} \\ H(X) &= 1.7427 \frac{\text{bits}}{\text{symbol}} \end{aligned}$$

Example:

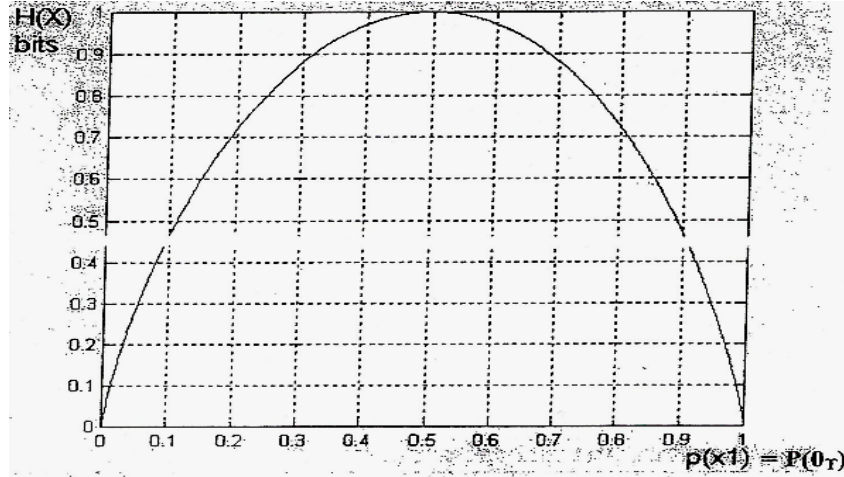
Find and plot the entropy of binary source.

$$\begin{aligned} P(0_T) + P(1_T) &= 1 \\ H(X) &= -[P(0_T) \log_2 P(0_T) + (1 \\ &\quad - P(0_T)) \log_2 (1 - P(0_T))] \text{ bits/symbol} \end{aligned}$$

If $P(0_T) = 0.2$, then $P(1_T) = 1 - 0.2 = 0.8$, and put in above equation,

$$H(X) = -[0.2 \log_2(0.2) + 0.8 \log_2(0.8)] = 0.7$$

Not that $H(X)$ is maximum equal to 1(bit) if: $P(0_T) = P(1_T) = 0.5$ as shown in figure.



If all messages are equiprobable, then $P(x_i) = 1/n$ so hat:

$$\begin{aligned}
 H(X) &= H(X)_{max} \\
 &= -\left[\frac{1}{n} \log_a \left(\frac{1}{n}\right)\right] \times n = -\log_a \left(\frac{1}{n}\right) = \log_a n \text{ bits/symbol}
 \end{aligned}$$

And $H(X) = 0$ if one of the message has the prob of a certain event.

Source Entropy Rate:

It is the average rate of amount of information produced per second.

$$R(X) = H(X) \times \text{rate of producing the symbols} = \frac{\text{bits}}{\text{sec}} = \text{bps}$$

The unit of $H(X)$ is bits/symbol and the rate of producing the symbols is symbol/sec, so that the unit of $R(X)$ is bits/sec.

$$\text{Sometimes } R(X) = \frac{H(X)}{\bar{\tau}},$$

$$\bar{\tau} = \sum_{i=1}^n \tau_i P(x_i)$$

$\bar{\tau}$ is the average time duration of symbols, τ_i is the time duration of the symbol x_i .

Example :

A source produces dots '.' And dashes '-' with $P(\text{dot})=0.65$. If the time duration of dot is 200ms and that for a dash is 800ms. Find the average source entropy rate.

Solution:

$$P(\text{dash}) = 1 - P(\text{dot}) = 1 - 0.65 = 0.35$$

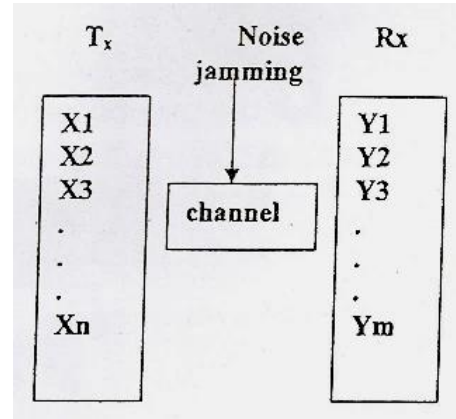
$$H(X) = -[0.65\log_2(0.65) + 0.35\log_2(0.35)] = 0.934 \text{ bits/symbol}$$

$$\bar{\tau} = 0.2 \times 0.65 + 0.8 \times 0.35 = 0.41 \text{ sec}$$

$$R(X) = \frac{H(X)}{\bar{\tau}} = \frac{0.934}{0.41} = 2.278 \text{ bps}$$

Mutual Information:

Consider the set of symbols x_1, x_2, \dots, x_n , the transmitter T_x may produce. The receiver R_x may receive y_1, y_2, \dots, y_m . Theoretically, if the noise and jamming is neglected, then the set $X = \text{set } Y$. However and due to noise and jamming, there will be a conditional probability $P(y_j | x_i)$:



- 1- $P(x_i)$ to be what is so called the apriori prob of the symbol x_i , which is the prob of selecting x_i for transmission.
- 2- $P(y_j | x_i)$ to be what is called the aposteriori prob of the symbol x_i after the reception of y_j .

The amount of information that y_j provides about x_i is called the mutual information between x_i and y_i . This is given by:

$$I(x_i, y_j) = \log_2 \left(\frac{\text{aposteriori prob}}{\text{apriori prob}} \right) = \log_2 \left(\frac{P(y_j | x_i)}{P(y_j)} \right)$$

Properties of $I(x_i, y_j)$:

- 1- It is symmetric, $I(x_i, y_j) = I(y_j, x_i)$.
- 2- $I(x_i, y_j) > 0$ if aposteriori prob > apriori prob, y_j provides +ve information about x_i .
- 3- $I(x_i, y_j) = 0$ if aposteriori prob = apriori prob, which is the case of statistical independence when y_j provides no information about x_i .
- 4- $I(x_i, y_j) < 0$ if aposteriori prob < apriori prob, y_j provides -ve information about x_i , or y_j adds ambiguity.

$$\text{Also } I(x_i, y_j) = \log_2 \left(\frac{P(x_i | y_j)}{P(x_i)} \right)$$

Example:

Show that $I(X, Y)$ is zero for extremely noisy channel.

Solution:

For extremely noisy channel, then y_j gives no information about x_i the receiver can't decide anything about x_i as if we transmit a deterministic signal x_i but the receiver receives noise like signal y_j that is completely has no correlation with x_i . Then x_i and y_j are statistically independent so that $P(x_i | y_j) = P(x_i)$ and $P(y_j | x_i) = P(y_j)$ for all i and j , then:

$$I(x_i, y_j) = \log_2 1 = 0 \text{ for all } i \text{ \& } j, \text{ then } I(X, Y) = 0$$

Transinformation (average mutual information):

It is the statistical average of all pair $I(x_i, y_j)$, $i = 1, 2, \dots, n$, $j = 1, 2, \dots, m$.

This is denoted by $I(X, Y)$ and is given by:

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

$$I(X, Y) = \sum_{i=1}^n \sum_{j=1}^m P(x_i, y_j) \log_2 \left(\frac{P(y_j | x_i)}{P(y_j)} \right) \frac{\text{bits}}{\text{symbol}}$$

or

$$I(X, Y) = \sum_{i=1}^n \sum_{j=1}^m P(x_i, y_j) \log_2 \left(\frac{P(x_i | y_j)}{P(x_i)} \right) \text{bits/symbol}$$

Expand above equation:

$$I(X, Y) = \sum_{i=1}^n \sum_{j=1}^m P(x_i, y_j) \log_2 (P(x_i | y_j)) - \sum_{i=1}^n \sum_{j=1}^m P(x_i, y_j) \log_2 (P(x_i))$$

And we have

$$\sum_{j=1}^m P(x_i, y_j) = p(x_i)$$

And by substituting:

$$I(X, Y) = \sum_{i=1}^n \sum_{j=1}^m P(x_i, y_j) \log_2 (P(x_i | y_j)) - \sum_{i=1}^n P(x_i) \log_2 (P(x_i))$$

$$\text{Or } I(X, Y) = H(X) - H(X | Y)$$

$$\text{Similarly } I(X, Y) = H(Y) - H(Y | X)$$

Marginal Entropies:

Marginal entropies is a term usually used to denote both source entropy

$H(X)$ defined as before and the receiver entropy $H(Y)$ given by:

$$H(Y) = - \sum_{j=1}^m P(y_j) \log_2 P(y_j) \quad \frac{\text{bits}}{\text{symbol}}$$

Joint entropy and conditional entropy:

The average information associated with the pair (x_i, y_j) is called joint or system entropy $H(X, Y)$:

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

$$= - \sum_{j=1}^m \sum_{i=1}^n P(x_i, y_j) \log_2 P(x_i, y_j) \quad \text{bits/symbol}$$

The average amount of information associated with the pairs $P(x_i | y_j)$ and $P(y_j | x_i)$ are called conditional entropies $H(Y | X)$ and $H(X | Y)$, and given by:

$$H(Y | X) = - \sum_{j=1}^m \sum_{i=1}^n P(x_i, y_j) \log_2 P(y_j | x_i) \quad \text{bits/symbol}$$

Return to first equation, we have: $P(x_i, y_j) = P(x_i)P(y_j | x_i)$, put inside log term

$$H(X, Y) = - \sum_{j=1}^m \sum_{i=1}^n P(x_i, y_j) \log_2 P(x_i)$$

$$- \sum_{j=1}^m \sum_{i=1}^n P(x_i, y_j) \log_2 P(y_j | x_i)$$

But

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

Put it in above equation yields:

$$H(X, Y) = - \sum_{i=1}^n P(x_i) \log_2 P(x_i) - \sum_{j=1}^m \sum_{i=1}^n P(x_i, y_j) \log_2 P(y_j | x_i)$$

So that $H(X, Y) = H(X) + H(Y | X)$

Example :

The joint probability of a system is given by:

$$P(X,Y) = \begin{matrix} x_1 \\ x_2 \\ x_3 \end{matrix} \begin{bmatrix} 0.5 & 0.25 \\ 0 & 0.125 \\ 0.0625 & 0.0625 \end{bmatrix}$$

Find:

- 1- Marginal entropies.
- 2- Joint entropy
- 3- Conditional entropies.
- 4- The mutual information between x_1 and y_2 .
- 5- The transinformation.
- 6- Draw the channel model.

$$1- P(X) = \begin{bmatrix} x_1 & x_2 & x_3 \\ 0.75 & 0.125 & 0.125 \end{bmatrix} \quad P(Y) = \begin{bmatrix} y_1 & y_2 \\ 0.5625 & 0.4375 \end{bmatrix}$$

$$H(X) = -[0.75 \ln(0.75) + 2 \times 0.125 \ln(0.125)]/\ln 2$$

$$= 1.06127 \text{ bits/symbol}$$

$$H(Y) = -[0.5625 \ln(0.5625) + 0.4375 \ln(0.4375)]/\ln 2$$

$$= 0.9887 \text{ bits/symbol}$$

2-

$$H(X,Y) = - \sum_{j=1}^m \sum_{i=1}^n P(x_i, y_j) \log_2 P(x_i, y_j)$$

$$H(X,Y)$$

$$= - \frac{[0.5 \ln(0.5) + 0.25 \ln(0.25) + 0.125 \ln(0.125) + 2 \times 0.0625 \ln(0.0625)]}{\ln 2}$$

$$= 1.875 \text{ bits/symbol}$$

$$3- H(Y | X) = H(X,Y) - H(X) = 1.875 - 1.06127 =$$

$$0.813 \frac{\text{bits}}{\text{symbol}}$$

$$H(X | Y) = H(X,Y) - H(Y) = 1.875 - 0.9887$$

$$= 0.886 \text{ bits/symbol}$$

4- $I(x_1, y_2) = \log_2 \left(\frac{P(x_1|y_2)}{P(x_1)} \right)$, but $P(x_1 | y_2) = P(x_1, y_2)/P(y_2)$

$$I(x_1, y_2) = \log_2 \left(\frac{P(x_1, y_2)}{P(x_1)P(y_2)} \right) = \log_2 \frac{0.25}{0.75 \times 0.4375} = -0.3923 \text{ bits}$$

That means y_2 gives ambiguity about x_1

5- $I(X, Y) = H(X) - H(X | Y) = 1.06127 - 0.8863 = 0.17497 \text{ bits/symbol.}$

6- To draw the channel model, must find $P(Y|X)$ matrix from $P(X, Y)$ matrix by dividing its rows by the corresponding $P(x_i)$:

$$P(X | Y) = \begin{matrix} x_1 \\ x_2 \\ x_3 \end{matrix} \begin{bmatrix} 0.5/0.75 & 0.25/0.75 \\ 0/0.125 & 0.125/0.125 \\ 0.0625/0.125 & 0.0625/0.125 \end{bmatrix}$$

$$= \begin{matrix} x_1 \\ x_2 \\ x_3 \end{matrix} \begin{bmatrix} 2/3 & 1/3 \\ 0 & 1 \\ 0.5 & 0.5 \end{bmatrix}$$

Example of joint entropy. Let $p(x, y)$ be given by

$X \backslash Y$	0	1
0	$\frac{1}{3}$	$\frac{1}{3}$
1	0	$\frac{1}{3}$

Find

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

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

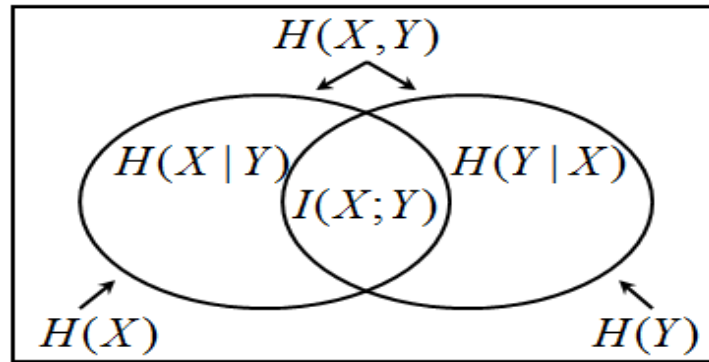
Solution:

(a) $H(X) = \frac{2}{3} \log \frac{3}{2} + \frac{1}{3} \log 3 = \log 3 - \frac{2}{3} = 0.918 \text{ bits} = H(Y)$.

(b) $H(X, Y) = 3 \times \frac{1}{3} \log 3 = \log 3 = 1.585 \text{ bits.}$

Venn diagrams:

The **Venn diagrams** is a helpful mean to understand the relations between mutual information and conditional entropies as shown below:



"Computing is not about computers any more. It is about living"

AlSafwa University College
Dep. Of Computer Techniques Engineering



Subject: Information Theory and Coding

Stage: Fourth

Academic Year: 2020-2021

Lecturer: Assist. Lec. Noor Yahya

Lecture: Chapter Two

Notes:

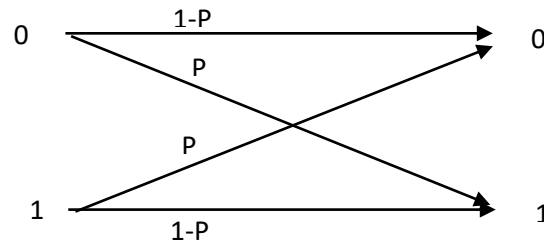
Chapter Two

2.1- Channel:

In telecommunications and computer networking, a communication channel or **channel**, refers either to a physical transmission medium such as a wire, or to a logical connection over a multiplexed medium such as a radio channel. A channel is used to convey an information signal, for example a digital bit stream, from one or several *senders* (or transmitters) to one or several *receivers*. A channel has a certain capacity for transmitting information, often measured by its bandwidth in Hz or its data rate in bits per second.

2.2- Binary symmetric channel (BSC)

It is a common communications channel model used in coding theory and information theory. In this model, a transmitter wishes to send a bit (a zero or a one), and the receiver receives a bit. It is assumed that the bit is *usually* transmitted correctly, but that it will be "flipped" with a small probability (the "crossover probability").



A **binary symmetric channel with crossover probability p** denoted by BSC_p , is a channel with binary input and binary output and probability of error p ; that is, if X is the transmitted random variable and Y the received variable, then the channel is characterized by the conditional probabilities:

$$\Pr(Y = 0 \mid X = 0) = 1 - P$$

$$\Pr(Y = 0 \mid X = 1) = P$$

$$\Pr(Y = 1 \mid X = 0) = P$$

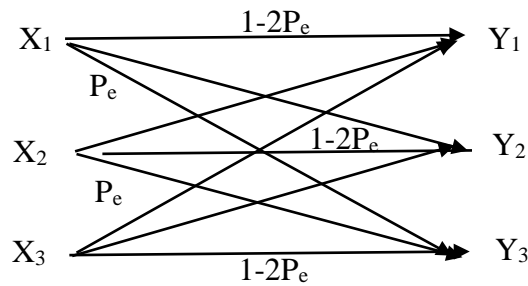
$$\Pr(Y = 1 \mid X = 1) = 1 - P$$

2.3- Ternary symmetric channel (TSC):

The transitional probability of TSC is:

$$P(Y | X) = \begin{matrix} x_1 \\ x_2 \\ x_3 \end{matrix} \begin{bmatrix} y_1 & y_2 & y_3 \\ 1-2P_e & P_e & P_e \\ P_e & 1-2P_e & P_e \\ P_e & P_e & 1-2P_e \end{bmatrix}$$

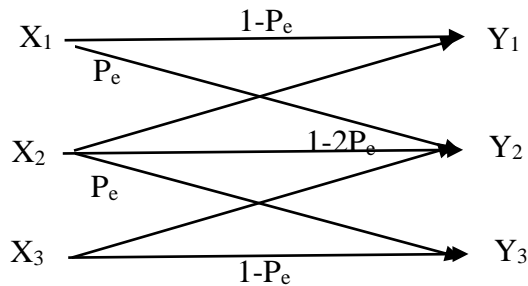
The TSC is symmetric but not very practical since practically x_1 and x_3 are not affected so much as x_2 . In fact the interference between x_1 and x_3 is much less than the interference between x_1 and x_2 or x_2 and x_3 .



Hence the more practice but nonsymmetric channel has the trans. prob.

$$P(Y | X) = \begin{matrix} x_1 \\ x_2 \\ x_3 \end{matrix} \begin{bmatrix} y_1 & y_2 & y_3 \\ 1-P_e & P_e & 0 \\ P_e & 1-2P_e & P_e \\ 0 & P_e & 1-P_e \end{bmatrix}$$

Where x_1 interfere with x_2 exactly the same as interference between x_2 and x_3 , but x_1 and x_3 are not interfere.



2.4- Special Channels:

- 1- Lossless channel: It has only one nonzero element in each column of the transitional matrix $P(Y|X)$.

$$P(Y|X) = \begin{matrix} & y_1 & y_2 & y_3 & y_4 & y_5 \\ \begin{matrix} x_1 \\ x_2 \\ x_3 \end{matrix} & \begin{bmatrix} 3/4 & 1/4 & 0 & 0 & 0 \\ 0 & 0 & 1/3 & 2/3 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \end{matrix}$$

This channel has $H(X|Y)=0$ and $I(X, Y)=H(X)$ with zero losses entropy.

- 2- Deterministic channel: It has only one nonzero element in each row, the transitional matrix $P(Y|X)$, as an example:

$$P(Y|X) = \begin{matrix} & y_1 & y_2 & y_3 \\ \begin{matrix} x_1 \\ x_2 \\ x_3 \end{matrix} & \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix} \end{matrix}$$

This channel has $H(Y|X)=0$ and $I(Y, X)=H(Y)$ with zero noisy entropy.

- 3- Noiseless channel: It has only one nonzero element in each row and column, the transitional matrix $P(Y|X)$, i.e. it is an identity matrix, as an example:

$$P(Y|X) = \begin{matrix} & y_1 & y_2 & y_3 \\ \begin{matrix} x_1 \\ x_2 \\ x_3 \end{matrix} & \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{matrix}$$

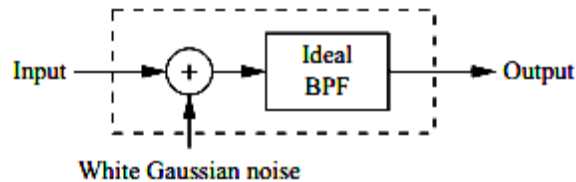
This channel has $H(Y|X)=H(X|Y)=0$ and $I(Y, X)=H(Y)=H(X)$.

2.5- Shannon's theorem:

- 1- A given communication system has a maximum rate of information C known as the channel capacity.
- 2- If the information rate R is less than C , then one can approach arbitrarily small error probabilities by using intelligent coding techniques.
- 3- To get lower error probabilities, the encoder has to work on longer blocks of signal data. This entails longer delays and higher computational requirements.

Thus, if $R \leq C$ then transmission may be accomplished without error in the presence of noise. The negation of this theorem is also true: if $R > C$, then errors cannot be avoided regardless of the coding technique used.

Consider a bandlimited Gaussian channel operating in the presence of additive Gaussian noise:



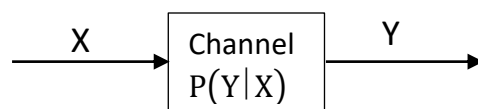
The Shannon-Hartley theorem states that the channel capacity is given by:

$$C = B \log_2 \left(1 + \frac{S}{N} \right)$$

Where C is the capacity in bits per second, B is the bandwidth of the channel in Hertz, and S/N is the signal-to-noise ratio.

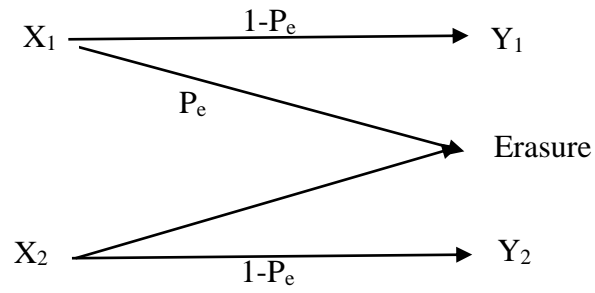
2.6- Discrete Memoryless Channel:

The Discrete Memoryless Channel (DMC) has an input X and an output Y . At any given time (t), the channel output $Y = y$ only depends on the input $X = x$ at that time (t) and it does not depend on the past history of the input. DMC is represented by the conditional probability of the output $Y = y$ given the input $X = x$, or $P(Y|X)$.



2.7 Binary Erasure Channel (BEC):

The Binary Erasure Channel (BEC) model are widely used to represent channels or links that “losses” data. Prime examples of such channels are Internet links and routes. A BEC channel has a binary input X and a ternary output Y .



Note that for the BEC, the probability of “bit error” is zero. In other words, the following conditional probabilities hold for any BEC model:

$$\Pr(Y = \text{"erasure"} \mid X = 0) = P$$

$$\Pr(Y = \text{"erasure"} \mid X = 1) = P$$

$$\Pr(Y = 0 \mid X = 0) = 1 - P$$

$$\Pr(Y = 1 \mid X = 1) = 1 - P$$

$$\Pr(Y = 0 \mid X = 1) = 0$$

$$\Pr(Y = 1 \mid X = 0) = 0$$

Channel Capacity (Discrete channel)

This is defined as the maximum of $I(X,Y)$:

$$C = \text{channel capacity} = \max[I(X,Y)] \quad \text{bits/symbol}$$

Physically it is the maximum amount of information each symbol can carry to the receiver. Sometimes this capacity is also expressed in bits/sec if related to the rate of producing symbols r :

$$R(X,Y) = r \times I(X,Y) \quad \text{bits/sec} \quad \text{or} \quad R(X,Y) = I(X,Y) / \bar{\tau}$$

1- Channel capacity of Symmetric channels:

The symmetric channel have the following condition:

- a- Equal number of symbol in X&Y, i.e. $P(Y|X)$ is a square matrix.
- b- Any row in $P(Y|X)$ matrix comes from some permutation of other rows.

For example the following conditional probability of various channel types as shown:

a- $P(Y | X) = \begin{bmatrix} 0.9 & 0.1 \\ 0.1 & 0.9 \end{bmatrix}$ is a BSC, because it is square matrix and 1st row is the permutation of 2nd row.

b- $P(Y | X) = \begin{bmatrix} 0.9 & 0.05 & 0.05 \\ 0.05 & 0.9 & 0.05 \\ 0.05 & 0.05 & 0.9 \end{bmatrix}$ is TSC, because it is square matrix and each row is a permutation of others.

c- $P(Y | X) = \begin{bmatrix} 0.8 & 0.1 & 0.1 \\ 0.1 & 0.8 & 0.1 \end{bmatrix}$ is a non-symmetric since it is not square although each row is permutation of others.

d- $P(Y | X) = \begin{bmatrix} 0.8 & 0.1 & 0.1 \\ 0.1 & 0.7 & 0.2 \\ 0.1 & 0.1 & 0.8 \end{bmatrix}$ is a non-symmetric although it is square since 2nd row is not permutation of other rows.

The channel capacity is defined as $\max[I(X,Y)]$:

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

$$I(X, Y) = H(Y) + \sum_{j=1}^m \sum_{i=1}^n P(x_i, y_j) \log_2 P(y_j | x_i)$$

But we have

$$P(x_i, y_j) = P(x_i)P(y_j | x_i) \quad \text{put in above equation yields:}$$

$$I(X, Y) = H(Y) + \sum_{j=1}^m \sum_{i=1}^n P(x_i)P(y_j | x_i) \log_2 P(y_j | x_i)$$

If the channel is symmetric the quantity:

$$\sum_{j=1}^m P(y_j | x_i) \log_2 P(y_j | x_i) = K$$

Where K is constant and independent of the row number i so that the equation becomes:

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

Hence $I(X, Y) = H(Y) + K$ for symmetric channels

$$\text{Max of } I(X, Y) = \max[H(Y) + K] = \max[H(Y)] + K$$

When Y has equiprobable symbols then $\max[H(Y)] = \log_2 m$

Then

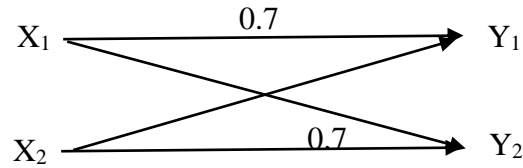
$$I(X, Y) = \log_2 m + K$$

Or

$$C = \log_2 m + K$$

Example 9:

For the BSC shown:



Find the channel capacity and efficiency if $I(x_1) = 2 \text{ bits}$

Solution:

$$P(Y | X) = \begin{bmatrix} 0.7 & 0.3 \\ 0.3 & 0.7 \end{bmatrix}$$

Since the channel is symmetric then

$$C = \log_2 m + K \quad \text{and } n = m$$

where n and m are number row and column respectively

$$K = 0.7 \log_2 0.7 + 0.3 \log_2 0.3 = -0.88129$$

$$C = 1 - 0.88129 = 0.1187 \text{ bits/symbol}$$

The channel efficiency $\eta = \frac{I(X,Y)}{C}$

$$I(x_1) = -\log_2 P(x_1) = 2$$

$$P(x_1) = 2^{-2} = 0.25 \quad \text{then } P(X) = [0.25 \quad 0.75]^T$$

And we have $P(x_i, y_j) = P(x_i)P(y_j | x_i)$ so that

$$P(X,Y) = \begin{bmatrix} 0.7 \times 0.25 & 0.3 \times 0.25 \\ 0.3 \times 0.75 & 0.7 \times 0.75 \end{bmatrix} = \begin{bmatrix} 0.175 & 0.075 \\ 0.225 & 0.525 \end{bmatrix}$$

$$P(Y) = [0.4 \quad 0.6] \rightarrow H(Y) = 0.97095 \text{ bits/symbol}$$

$$I(X,Y) = H(Y) + K = 0.97095 - 0.88129 = 0.0896 \text{ bits/symbol}$$

$$\text{Then } \eta = \frac{0.0896}{0.1187} = 75.6\%$$

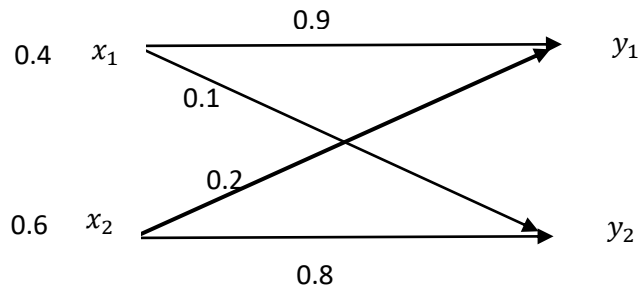
Review questions:

A binary source sending x_1 with a probability of 0.4 and x_2 with 0.6 probability through a channel with a probabilities of errors of 0.1 for x_1 and 0.2 for x_2 . Determine:

- 1- Source entropy.
- 2- Marginal entropy.
- 3- Joint entropy.
- 4- Conditional entropy $H(Y | X)$.
- 5- Losses entropy $H(X | Y)$.
- 6- Transinformation.

Solution:

- 1- The channel diagram:



$$\text{Or } P(Y | X) = \begin{bmatrix} 0.9 & 0.1 \\ 0.2 & 0.8 \end{bmatrix}$$

$$H(X) = - \sum_{i=1}^n p(x_i) \log_2 p(x_i)$$

$$H(X) = - \frac{[0.4 \ln(0.4) + 0.6 \ln(0.6)]}{\ln 2} = 0.971 \frac{\text{bits}}{\text{symbol}}$$

$$2- P(X, Y) = P(Y | X) \times P(X)$$

$$\therefore P(X, Y) = \begin{bmatrix} 0.9 \times 0.4 & 0.1 \times 0.4 \\ 0.2 \times 0.6 & 0.8 \times 0.6 \end{bmatrix} = \begin{bmatrix} 0.36 & 0.04 \\ 0.12 & 0.48 \end{bmatrix}$$
$$\therefore P(Y) = [0.48 \quad 0.52]$$

$$H(Y) = - \sum_{j=1}^m p(y_j) \log_2 p(y_j)$$

$$H(Y) = - \frac{[0.48 \ln(0.48) + 0.52 \ln(0.52)]}{\ln(2)} = 0.999 \text{ bits/symbol}$$

3- $H(X, Y)$

$$H(X, Y) = - \sum_{j=1}^m \sum_{i=1}^n P(x_i, y_j) \log_2 P(x_i, y_j)$$

$$H(X, Y) = - \frac{[0.36 \ln(0.36) + 0.04 \ln(0.04) + 0.12 \ln(0.12) + 0.48 \ln(0.48)]}{\ln(2)}$$

$$= 1.592 \text{ bits/symbol}$$

4- $H(Y | X)$

$$H(Y | X) = - \sum_{j=1}^m \sum_{i=1}^n P(x_i, y_j) \log_2 P(y_j | x_i)$$

$$H(Y | X) = - \frac{[0.36 \ln(0.9) + 0.12 \ln(0.2) + 0.04 \ln(0.1) + 0.48 \ln(0.8)]}{\ln(2)}$$

$$= 0.621 \frac{\text{bits}}{\text{symbol}}$$

Or $H(Y | X) = H(X, Y) - H(X) = 1.592 - 0.971 = 0.621 \frac{\text{bits}}{\text{symbol}}$

5- $H(X | Y) = H(X, Y) - H(Y) = 1.592 - 0.999 = 0.593 \text{ bits/symbol}$

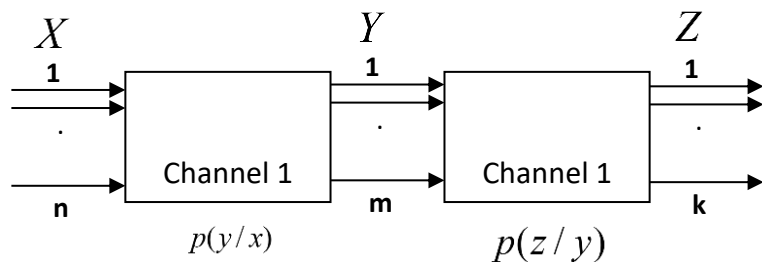
6- $I(X, Y) = H(X) - H(X | Y) = 0.971 - 0.593 = 0.378 \text{ bits/symbol}$

2- Cascading of Channels

If two channels are cascaded, then the overall transition matrix is the product of the two transition matrices.

$$p(z / x) = p(y / x) \cdot p(z / y)$$

$$\begin{matrix} (n \times k) & (n \times m) & (m \times k) \\ \text{matrix} & \text{matrix} & \text{matrix} \end{matrix}$$

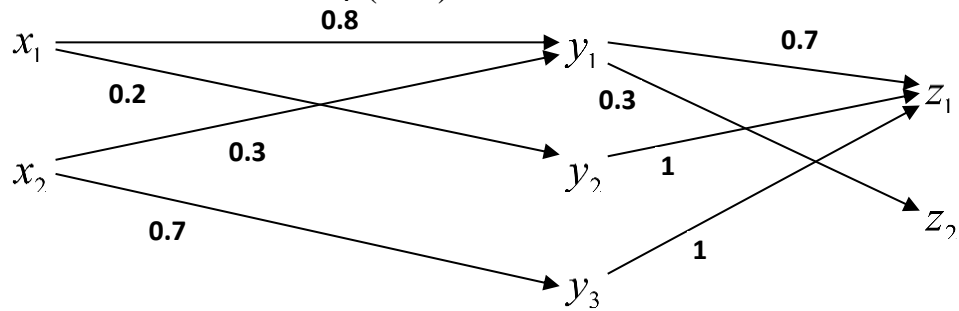


For the series information channel, the overall channel capacity is not exceed any of each channel individually.

$$I(X, Z) \leq I(X, Y) \quad \& \quad I(X, Z) \leq I(Y, Z)$$

Example:

Find the transition matrix $p(z/x)$ for the cascaded channel shown.



$$p(y/x) = \begin{bmatrix} 0.8 & 0.2 & 0 \\ 0.3 & 0 & 0.7 \end{bmatrix}, \quad p(z/y) = \begin{bmatrix} 0.7 & 0.3 \\ 1 & 0 \\ 1 & 0 \end{bmatrix}$$

$$p(z/x) = \begin{bmatrix} 0.8 & 0.2 & 0 \\ 0.3 & 0 & 0.7 \end{bmatrix} \begin{bmatrix} 0.7 & 0.3 \\ 1 & 0 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 0.76 & 0.24 \\ 0.91 & 0.09 \end{bmatrix}$$

كلية الصفوة الجامعة قسم هندسة تقنيات الحاسوب

أسم المادة: نظرية المعلومات و الترميز المرحلة: الرابعة



الملاحظات: ٢٠١٨/٢/١٨

CHAPTER THREE

Chapter Three

Source Coding

1- Sampling theorem:

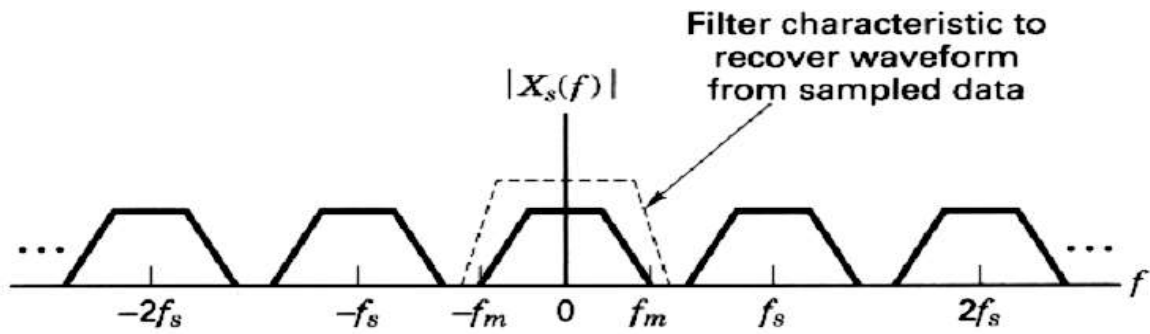
Sampling of the signals is the fundamental operation in digital communication. A continuous time signal is first converted to discrete time signal by sampling process. Also it should be possible to recover or reconstruct the signal completely from its samples.

The sampling theorem state that:

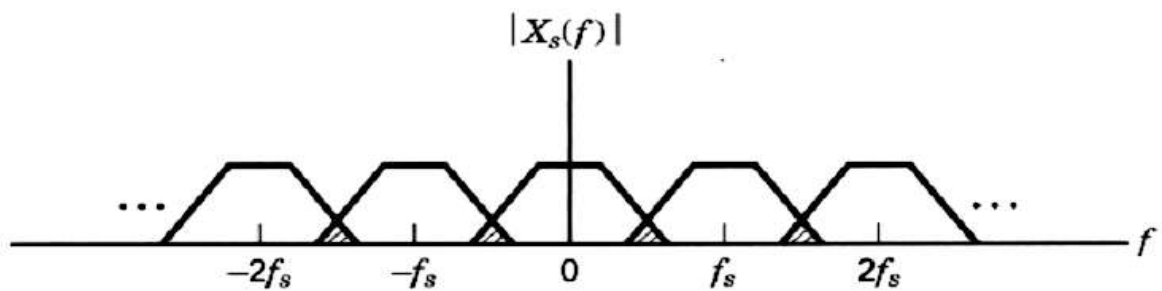
- i- A band limited signal of finite energy, which has no frequency components higher than W Hz, is completely described by specifying the values of the signal at instant of time separated by $1/2W$ second and*
- ii- A band limited signal of finite energy, which has no frequency components higher than W Hz, may be completely recovered from the knowledge of its samples taken at the rate of $2W$ samples per second.*

When the sampling rate is chosen $f_s = 2f_m$ each spectral replicate is separated from each of its neighbors by a frequency band exactly equal to f_s hertz, and the analog waveform can theoretically be completely recovered from the samples, by the use of filtering. It should be clear that if $f_s > 2f_m$, the replications will be move farther apart in frequency making it easier to perform the filtering operation.

When the sampling rate is reduced, such that $f_s < 2f_m$, the replications will overlap, as shown in figure below, and some information will be lost. This phenomenon is called aliasing.



Sampled spectrum $f_s > 2f_m$



Sampled spectrum $f_s < 2f_m$

A bandlimited signal having no spectral components above f_m hertz can be determined uniquely by values sampled at uniform intervals of $T_s \leq \frac{1}{2f_m} \text{ sec.}$

The sampling rate is $f_s = \frac{1}{T_s}$

So that $f_s \geq 2f_m$. The sampling rate $f_s = 2f_m$ is called Nyquist rate.

Example: Find the Nyquist rate and Nyquist interval for the following signals.

i- $m(t) = \frac{\sin(500\pi t)}{\pi t}$

ii- $m(t) = \frac{1}{2\pi} \cos(4000\pi t) \cos(1000\pi t)$

Solution:

i- $\omega t = 500\pi t \quad \therefore 2\pi f = 500\pi \quad \rightarrow f = 250\text{Hz}$

Nyquist interval = $\frac{1}{2f_{\max}} = \frac{1}{2 \times 250} = 2 \text{ msec.}$

$$\text{Nyquist rate} = 2f_{max} = 2 \times 250 = 500\text{Hz}$$

$$\begin{aligned} \text{ii- } m(t) &= \frac{1}{2\pi} \left[\frac{1}{2} \{ \cos(4000\pi t - 1000\pi t) + \cos(4000\pi t + 1000\pi t) \} \right] \\ &= \frac{1}{4\pi} \{ \cos(3000\pi t) + \cos(5000\pi t) \} \end{aligned}$$

Then the highest frequency is 2500Hz

$$\text{Nyquist interval} = \frac{1}{2f_{max}} = \frac{1}{2 \times 2500} = 0.2 \text{ msec.}$$

$$\text{Nyquist rate} = 2f_{max} = 2 \times 2500 = 5000\text{Hz}$$

H. W:

Find the Nyquist interval and Nyquist rate for the following:

$$\text{i- } \frac{1}{2\pi} \cos(400\pi t) \cdot \cos(200\pi t)$$

$$\text{ii- } \frac{1}{\pi} \sin \pi t$$

Example:

A waveform $[20 + 20\sin(500t + 30^\circ)]$ is to be sampled periodically and reproduced from these sample values. Find maximum allowable time interval between sample values, how many sample values are needed to be stored in order to reproduce 1 sec of this waveform?.

Solution:

$$x(t) = 20 + 20 \sin(500t + 30^\circ)$$

$$\omega = 500 \rightarrow 2\pi f = 500 \rightarrow f = 79.58 \text{ Hz}$$

Minimum sampling rate will be twice of the signal frequency:

$$f_{s(\min)} = 2 \times 79.58 = 159.15 \text{ Hz}$$

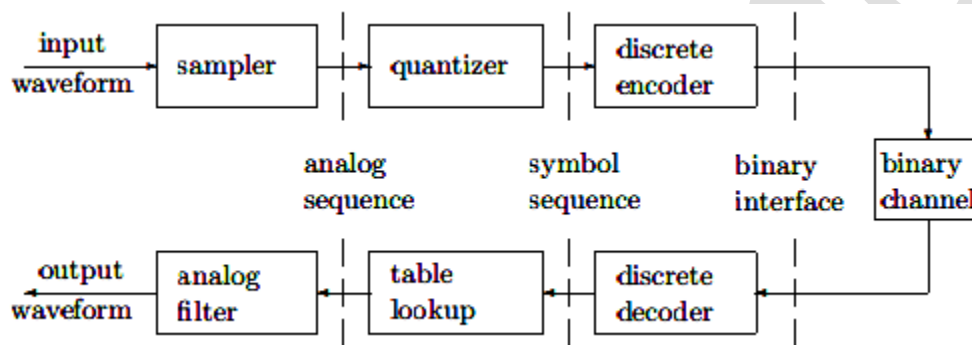
$$T_{s(\max)} = \frac{1}{f_{s(\min)}} = \frac{1}{159.15} = 6.283 \text{ msec.}$$

$$\text{Number of sample in } 1\text{sec} = \frac{1}{6.283\text{msec}} = 159.16 \approx 160 \text{ sample}$$

2- Source coding:

An important problem in communications is the efficient representation of data generated by a discrete source. The process by which this representation is accomplished is called source encoding. An efficient source encoder must satisfies two functional requirements:

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



The entropy for a source with statistically independent symbols:

$$H(Y) = - \sum_{j=1}^m P(y_j) \log_2 P(y_j)$$

We have:

$$\max[H(Y)] = \log_2 m$$

A code efficiency can therefore be defined as:

$$\eta = \frac{H(Y)}{\max[H(Y)]} \times 100$$

The overall code length, L, can be defined as the average code word length:

$$L = \sum_{j=1}^m P(x_j) l_j \quad \text{bits/symbol}$$

The code efficiency can be found by:

$$\eta = \frac{H(Y)}{L} \times 100$$

Not that $\max[H(Y)] \text{ bits/symbol} = L \text{ bits/codeword}$

i- Fixed- Length Code Words:

If the alphabet X consists of the 7 symbols {a, b, c, d, e, f, g}, then the following fixed-length code of block length $L = 3$ could be used.

C(a) = 000

C(b) = 001

C(c) = 010

C(d) = 011

C(e) = 100

C(f) = 101

C(g) = 110.

The encoded output contains L bits per source symbol. For the above example the source sequence *bad...* would be encoded into 001000011... . Note that the output bits are simply run together (or, more technically, concatenated). This method is nonprobabilistic; it takes no account of whether some symbols occur more frequently than others, and it works robustly regardless of the symbol frequencies.

This is used when the source produces almost equiprobable messages $p(x_1) \cong p(x_2) \cong p(x_3) \cong \dots \cong p(x_n)$, then $l_1 = l_2 = l_3 = \dots = l_n = L_c$ and for binary coding then:

1- $L_c = \log_2 n$ bit/message if $n = 2^r$ ($n = 2, 4, 8, 16, \dots$ and r is an integer) which gives $\eta = 100\%$

2- $L_c = \text{Int}[\log_2 n] + 1$ bits/message if $n \neq 2^r$ which gives less efficiency

Example

For ten equiprobable messages coded in a fixed length code then

$$p(x_i) = \frac{1}{10} \text{ and } L_c = \text{Int}[\log_2 10] + 1 = 4 \text{ bits}$$

$$\text{and } \eta = \frac{H(X)}{L_c} \times 100\% = \frac{\log_2 10}{4} \times 100\% = 83.048\%$$

Example: For eight equiprobable messages coded in a fixed length code then

$$p(x_i) = \frac{1}{8} \text{ and } L_c = \log_2 8 = 3 \text{ bits and } \eta = \frac{3}{3} \times 100\% = 100\%$$

Example: Find the efficiency of a fixed length code used to encode messages obtained from throwing a fair die (a) once, (b) twice, (c) 3 times.

Solution

- a- For a fair die, the messages obtained from it are equiprobable with a probability of $p(x_i) = \frac{1}{6}$ with $n = 6$.

$$L_c = \text{Int}[\log_2 6] + 1 = 3 \text{ bits/message}$$

$$\eta = \frac{H(X)}{L_c} \times 100\% = \frac{\log_2 6}{3} \times 100\% = 86.165\%$$

- b- For two throws then the possible messages are $n = 6 \times 6 = 36$ messages with equal probabilities

$$L_c = \text{Int}[\log_2 36] + 1 = 6 \text{ bits/message} = 6 \text{ bits/2-symbols}$$

$$\text{while } H(X) = \log_2 6 \text{ bits/symbol} \quad \eta = \frac{2 \times H(X)}{L_c} \times 100\% = 86.165\%$$

- c- For three throws then the possible messages are $n = 6 \times 6 \times 6 = 216$ with equal probabilities

$$L_c = \text{Int}[\log_2 216] + 1 = 8 \text{ bits/message} = 8 \text{ bits/3-symbols}$$

$$\text{while } H(X) = \log_2 6 \text{ bits/symbol} \quad \eta = \frac{3 \times H(X)}{L_c} \times 100\% = 96.936\%$$

ii- Variable-Length Code Words

When the source symbols are not equally probable, a more efficient encoding method is to use variable-length code words. For example, a variable-length code for the alphabet $X = \{a, b, c\}$ and its lengths might be given by

$$\begin{array}{ll} C(a) = 0 & l(a) = 1 \\ C(b) = 10 & l(b) = 2 \\ C(c) = 11 & l(c) = 2 \end{array}$$

The major property that is usually required from any variable-length code is that of unique decodability. For example, the above code C for the alphabet $X = \{a, b, c\}$ is soon shown to be uniquely decodable. However such code is not uniquely decodable, even though the codewords are all different. If the source decoder observes 01, it cannot determine whether the source emitted (a b) or (c).

Prefix-free codes: A **prefix code** is a type of code system (typically a variable-length code) distinguished by its possession of the "prefix property", which requires that there is no code word in the system that is a prefix (initial segment) of any other code word in the system. For example:

$$\{a = 0, b = 110, c = 10, d = 111\} \text{ is a prefix code.}$$

When message probabilities are not equal, then we use variable length codes. The following properties need to be considered when attempting to use variable length codes:

1) Unique decoding:

Example

Consider a 4 alphabet symbols with symbols represented by binary digits as follows:

$$A = 0$$

$$B = 01$$

$$C = 11$$

$$D = 00$$

If we receive the code word 0011 it is not known whether the transmission was DC or AAC . This example is not, therefore, uniquely decodable.

2) *Instantaneous decoding:*

Example

Consider a 4 alphabet symbols with symbols represented by binary digits as follows:

$$A = 0$$

$$B = 10$$

$$C = 110$$

$$D = 111$$

This code can be instantaneously decoded since no complete codeword is a prefix of a larger codeword. This is in contrast to the previous example where A is a prefix of both B and D . This example is also a ‘comma code’ as the symbol zero indicates the end of a codeword except for the all ones word whose length is known.

Example

Consider a 4 alphabet symbols with symbols represented by binary digits as follows:

$$A = 0$$

$$B = 01$$

$$C = 011$$

$$D = 111$$

The code is identical to the previous example but the bits are time reversed. It is still uniquely decodable but no longer instantaneous, since early codewords are now prefixes of later ones.

Shannon Code

For messages $x_1, x_2, x_3, \dots, x_n$ with probabilities $p(x_1), p(x_2), p(x_3), \dots, p(x_n)$ then:

$$1) l_i = -\log_2 p(x_i) \quad \text{if } p(x_i) = \left(\frac{1}{2}\right)^r \quad \left\{\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \dots\right\}$$

$$2) l_i = \text{Int}[-\log_2 p(x_i)] + 1 \quad \text{if } p(x_i) \neq \left(\frac{1}{2}\right)^r$$

Also define
$$F_i = \sum_{k=1}^{i-1} p(x_k) \quad 1 \geq \omega_i \geq 0$$

then the codeword of x_i is the binary equivalent of F_i consisting of l_i bits.

$$C_i = (F_i)_2^{l_i}$$

where C_i is the binary equivalent of F_i up to l_i bits. In encoding, messages must be arranged in a decreasing order of probabilities.

Example

Develop the Shannon code for the following set of messages,

$$p(x) = [0.3 \quad 0.2 \quad 0.15 \quad 0.12 \quad 0.1 \quad 0.08 \quad 0.05]$$

then find:

- (a) Code efficiency,
- (b) $p(0)$ at the encoder output.

Solution

x_i	$p(x_i)$	l_i	F_i	C_i	0_i
x_1	0.3	2	0	00	2
x_2	0.2	3	0.3	010	2
x_3	0.15	3	0.5	100	2
x_4	0.12	4	0.65	1010	2
x_5	0.10	4	0.77	1100	2
x_6	0.08	4	0.87	1101	1
x_7	0.05	5	0.95	11110	1

To find C_1

$$\begin{array}{rcl} 0 \times 2 = 0 & 0 & \downarrow \\ 0 \times 2 = 0 & 0 & \end{array}$$

To find C_2

$$\begin{array}{rcl} 0.3 \times 2 = 0.6 & 0 & \downarrow \\ 0.6 \times 2 = 1.2 & 1 & \downarrow \\ 0.2 \times 2 = 0.4 & 0 & \end{array}$$

To find C_3

$$\begin{array}{rcl} 0.5 \times 2 = 1.0 & 1 & \downarrow \\ 0.0 \times 2 = 0.0 & 0 & \downarrow \end{array}$$

To find C_4

$$\begin{array}{rcl} 0.65 \times 2 = 1.3 & 1 & \downarrow \\ 0.30 \times 2 = 0.6 & 0 & \downarrow \\ 0.60 \times 2 = 1.2 & 1 & \downarrow \\ 0.20 \times 2 = 0.4 & 0 & \downarrow \end{array}$$

To find C_5

$$\begin{array}{rcl} 0.77 \times 2 = 1.54 & 1 & \downarrow \\ 0.54 \times 2 = 1.08 & 1 & \downarrow \\ 0.08 \times 2 = 0.16 & 0 & \downarrow \\ 0.16 \times 2 = 0.32 & 0 & \downarrow \end{array}$$

(a) To find the code efficiency, we have

$$L_C = \sum_{i=1}^7 l_i p(x_i) = 3.1 \text{ bits/message.}$$

$$H(X) = -\sum_{i=1}^7 p(x_i) \log_2 p(x_i) = 2.6029 \text{ bits/message.}$$

$$\eta = \frac{H(X)}{L_C} \times 100\% = 83.965\%$$

(b) $p(0)$ at the encoder output is

$$p(0) = \frac{\sum_{i=1}^7 0_i p(x_i)}{L_C} = \frac{0.6 + 0.4 + 0.3 + 0.24 + 0.2 + 0.08 + 0.05}{3.1}$$

$$p(0) = 0.603$$

Example

Repeat the previous example using ternary coding.

Solution

$$1) l_i = -\log_3 p(x_i) \quad \text{if } p(x_i) = \left(\frac{1}{3}\right)^r \quad \left\{\frac{1}{3}, \frac{1}{9}, \frac{1}{27}, \dots\right\}$$

2) $l_i = \text{Int}[-\log_3 p(x_i)] + 1$ if $p(x_i) \neq \left(\frac{1}{3}\right)^r$ and $C_i = (F_i)_3^{l_i}$

x_i	$p(x_i)$	l_i	F_i	C_i	0_i
x_1	0.3	2	0	00	2
x_2	0.2	2	0.3	02	1
x_3	0.15	2	0.5	11	0
x_4	0.12	2	0.65	12	0
x_5	0.10	3	0.77	202	1
x_6	0.08	3	0.87	212	0
x_7	0.05	3	0.95	221	0

To find C_1

$$0 \times 3 = 0 \quad 0 \downarrow$$

To find C_2

$$0.3 \times 3 = 0.9 \quad 0 \downarrow$$

To find C_3

$$0.5 \times 3 = 1.5 \quad 1 \downarrow$$

To find C_4

$$0.65 \times 3 = 1.95 \quad 1 \downarrow$$

$$0.95 \times 3 = 2.85 \quad 2$$

To find C_5

$$0.77 \times 3 = 2.31 \quad 2 \downarrow$$

$$0.31 \times 3 = 0.93 \quad 0$$

(a) To find the code efficiency, we have

$$L_c = \sum_{i=1}^7 l_i p(x_i) = 2.23 \text{ ternary unit/message.}$$

$$H(X) = -\sum_{i=1}^7 p(x_i) \log_3 p(x_i) = 1.642 \text{ ternary unit/message.}$$

$$\eta = \frac{H(X)}{L_c} \times 100\% = 73.632\%$$

(b) $p(0)$ at the encoder output is

$$p(0) = \frac{\sum_{i=1}^7 0_i p(x_i)}{L_c} = \frac{0.6 + 0.2 + 0.1}{2.23}$$

$$p(0) = 0.404$$

Shannon- Fano Code:

In Shannon–Fano coding, the symbols are arranged in order from most probable to least probable, and then divided into two sets whose total probabilities are as close as possible to being equal. All symbols then have the first digits of their codes assigned; symbols in the first set receive "0" and symbols in the second set receive "1". As long as any sets with more than one member remain, the same process is repeated on those sets, to determine successive digits of their codes.

Example:

The five symbols which have the following frequency and probabilities, design suitable Shannon-Fano binary code. Calculate average code length, source entropy and efficiency.

Symbol	count	Probabilities	Binary codes	Length
A	15	0.385	00	2
B	7	0.1795	01	2
C	6	0.154	10	2
D	6	0.154	110	3
E	5	0.128	111	3

The average code word length:

$$L = \sum_{j=1}^m P(x_j) l_j$$

$$L = 2 \times 0.385 + 2 \times 0.1793 + 2 \times 0.154 + 3 \times 0.154 + 3 \times 0.128$$

$$= 2.28 \text{ bits/symbol}$$

The source entropy is:

$$H(Y) = - \sum_{j=1}^m P(y_j) \log_2 P(y_j)$$

$$H(Y) = -[0.385 \ln 0.385 + 0.1793 \ln 0.1793 + 2 \times 0.154 \ln 0.154 + 0.128 \ln 0.128] / \ln 2$$

$$H(Y) = 2.18567 \text{ bits/symbol}$$

The code efficiency:

$$\eta = \frac{H(Y)}{L} \times 100 = \frac{2.18567}{2.28} \times 100 = 95.86\%$$

Example

Develop the Shannon - Fano code for the following set of messages, $p(x) = [0.35 \ 0.2 \ 0.15 \ 0.12 \ 0.1 \ 0.08]$ then find the code efficiency.

Solution

x_i	$p(x_i)$	Code			l_i
x_1	0.35	0	0		2
x_2	0.2	0	1		2
x_3	0.15	1	0	0	3
x_4	0.12	1	0	1	3
x_5	0.10	1	1	0	3
x_6	0.08	1	1	1	3

$$L_C = \sum_{i=1}^6 l_i p(x_i) = 2.45 \text{ bits/symbol}$$

$$H(X) = -\sum_{i=1}^6 p(x_i) \log_2 p(x_i) = 2.396 \text{ bits/symbol}$$

$$\eta = \frac{H(X)}{L_C} \times 100\% = 97.796\%$$

Example

Repeat the previous example using with $r = 3$

Solution

x_i	$p(x_i)$	Code		l_i
x_1	0.35	0		1
x_2	0.2	1	0	2
x_3	0.15	1	1	2
x_4	0.12	2	0	2
x_5	0.10	2	1	2
x_6	0.08	2	2	2

$$L_C = \sum_{i=1}^6 l_i p(x_i) = 1.65 \text{ ternary unit/symbol}$$

$$H(X) = -\sum_{i=1}^6 p(x_i) \log_3 p(x_i) = 1.512 \text{ ternary unit/symbol}$$

$$\eta = \frac{H(X)}{L_C} \times 100\% = 91.636\%$$

Huffman Code

The Huffman coding algorithm comprises two steps, reduction and splitting. These steps can be summarized as follows:

1) Reduction

- List the symbols in descending order of probability.
- Reduce the r least probable symbols to one symbol with a probability equal to their combined probability.
- Reorder in descending order of probability at each stage.
- Repeat the reduction step until only two symbols remain.

2) Splitting

- Assign $0, 1, \dots, r$ to the r final symbols and work backwards.
- Expand or lengthen the code to cope with each successive split.

Example: Design Huffman codes for $A = \{a_1, a_2, \dots, a_5\}$, having the probabilities $\{0.2, 0.4, 0.2, 0.1, 0.1\}$.

Symbol	Step 1	Step 2	Step 3	Step 4	Codeword
a_2	0.4	0.4	0.4	0.6 0	1
a_1	0.2	0.2	0.4 } 0	0.4 1	01
a_3	0.2	0.2 } 0	0.2 } 1		000
a_4	0.1 } 0	0.2 } 1			0010
a_5	0.1 } 1				0011



Letter	Probability	Codeword
a_2	0.4	1
a_1	0.2	01
a_3	0.2	000
a_4	0.1	0010
a_5	0.1	0011

The average code word length:

$$L = 0.4 \times 1 + 0.2 \times 2 + 0.2 \times 3 + 0.1 \times 4 + 0.1 \times 4 = 2.2 \text{ bits/symbol}$$

The source entropy:

$$H(Y) = -[0.4 \ln 0.4 + 2 \times 0.2 \ln 0.2 + 2 \times 0.1 \ln 0.1] / \ln 2 = 2.12193 \text{ bits/symbol}$$

The code efficiency:

$$\eta = \frac{2.12193}{2.2} \times 100 = 96.45\%$$

It can be design Huffman codes with minimum variance:

Symbol	Step 1	Step 2	Step 3	Step 4	Codeword
a_2	0.4	0.4	0.4	0.6 0	00
a_1	0.2	0.2	0.4 } 0	0.4 1	10
a_3	0.2	0.2 } 0	0.2 } 1		11
a_4	0.1 } 0	0.2 } 1			010
a_5	0.1 } 1				011

The average code word length is still 2.2 bits/symbol. But variances are different!

Example

Develop the Huffman code for the following set of symbols

Symbol	A	B	C	D	E	F	G	H
Probability	0.1	0.18	0.4	0.05	0.06	0.1	0.07	0.04

Figure 10: A diagram illustrating the construction of a probability distribution \mathbf{p} from a sequence of distributions $\mathbf{p}^1, \mathbf{p}^2, \dots, \mathbf{p}^8$. The distributions are represented by horizontal bars of increasing length, with the total length of the sequence being 1.0. The distributions are labeled C, B, A, F, G, E, D, and H from top to bottom. The values for each distribution are given in the table below:

Distribution	Value
C	0.40
B	0.18
A	0.10
F	0.10
G	0.07
E	0.06
D	0.05
H	0.04

The diagram shows the cumulative distribution function (CDF) of the sequence, with the total length of the sequence being 1.0. The distributions are labeled C, B, A, F, G, E, D, and H from top to bottom. The values for each distribution are given in the table below:

Distribution	Value
C	0.40
B	0.18
A	0.10
F	0.10
G	0.07
E	0.06
D	0.05
H	0.04

Symbol	A	B	C	D	E	F	G	H
Probability	0.1	0.18	0.4	0.05	0.06	0.1	0.07	0.04
Codeword	011	001	1	00010	0101	0000	0100	00011
l_i	3	3	1	5	4	4	4	5

$$L_C = \sum_{i=1}^8 l_i p(x_i) = 2.61 \text{ bits/symbol}$$

$$\eta = \frac{H(X)}{L_c} \times 100\% = 97.778\%$$

Data Compression:

In computer science and information theory, data compression, source coding, or bit-rate reduction involves encoding information using fewer bits than the original representation. Compression can be either lossy or lossless.

Lossless data compression algorithms usually exploit statistical redundancy to represent data more concisely without losing information, so that the process is reversible. Lossless compression is possible because most real-world data has statistical redundancy. For example, an image may have areas of color that do not change over several pixels.

Lossy data compression is the converse of lossless data compression. In these schemes, some loss of information is acceptable. Dropping nonessential detail from the data source can save storage space. There is a corresponding trade-off between preserving information and reducing size.

Run-Length Encoding (RLE):

Run-Length Encoding is a very simple lossless data compression technique that replaces runs of two or more of the same character with a number which represents the length of the run, followed by the original character; single characters are coded as runs of 1. RLE is useful for highly-redundant data, indexed images with many pixels of the same color in a row.

Example:

Input: AAABBCCCCDEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
 AAAAAAAAAA

Output: 3A2B4C1D6E38A

The input message to RLE encoder is a variable while the output code word is fixed, unlike Huffman code where the input is fixed while the output is varied.

Example : Consider these repeated pixels values in an image ... 0 0 0 0 0 0 0 0 0 0 0 0
5 5 5 5 0 0 0 0 0 0 0 0 We could represent them more efficiently as (12, 0)(4, 5)(8, 0)

24 bytes reduced to 6 which gives a compression ratio of $24/6 = 4:1$.

Example :Original Sequence (1 Row): 11112223333311112222 can be encoded as:
(4,1),(3,2),(6,3),(4,1),(4,2). 21 bytes reduced to 10 gives a compression ratio of $21/10 = 2.1:1$.

Example : Original Sequence (1 Row): – HHHHHHHUFFFFFFFFFFFFFFF can be encoded as: (7,H),(1,U),(14,F) . 22 bytes reduced to 6 gives a compression ratio of $22/6 = 11:3$.

Savings Ratio : the savings ratio is related to the compression ratio and is a measure of the amount of redundancy between two representations (compressed and uncompressed). Let:

N_1 = the total number of bytes required to store an uncompressed (raw) source image.

N_2 = the total number of bytes required to store the compressed data.

The compression ratio C_r is then defined as:

$$C_r = \frac{N_1}{N_2}$$

- ▶ Larger compression ratios indicate more effective compression
- ▶ Smaller compression ratios indicate less effective compression
- ▶ Compression ratios less than one indicate that the uncompressed representation has high degree of irregularity.

The saving ratio S_r is then defined as :

$$S_r = \frac{(N_1 - N_2)}{N_1}$$

- ▶ Higher saving ratio indicate more effective compression while negative ratios are possible and indicate that the compressed image has larger memory size than the original.

Example: a 5 Megabyte image is compressed into a 1 Megabyte image, the savings ratio is defined as $(5-1)/5$ or $4/5$ or 80%.

This ratio indicates that 80% of the uncompressed data has been eliminated in the compressed encoding.

Chapter Four

Channel coding

1- Error detection and correction codes:

The idea of error detection and/or correction is to add extra bits to the digital message that enable the receiver to detect or correct errors with limited capabilities. These extra bits are called parity bits. If we have k bits, r parity bits are added, then the transmitted digits are:

$$n = r + k$$

Here n called code word denoted as (n, k) . The efficiency or code rate is equal to k/n .

Two basic approaches to error correction are available, which are:

a- Automatic-repeat-request (ARQ): Discard those frames in which errors are detected.

- For frames in which no error was detected, the receiver returns a positive acknowledgment to the sender.
- For the frame in which errors have been detected, the receiver returns negative acknowledgement to the sender.

b- Forward error correction (FEC):

Ideally, FEC codes can be used to generate encoding symbols that are transmitted in packets in such a way that each received packet is fully useful to a receiver to reassemble the object regardless of previous packet reception patterns. The most applications of FEC are:

Compact Disc (CD) applications, digital audio and video, Global System Mobile (GSM) and Mobile communications.

2- Basic definitions:

- **Systematic and nonsystematic codes:** If (a's) information bits are unchanged in their values and positions at the transmitted code word, then this code is said to be systematic (also called block code) where:

Input data $D = [a_1, a_2, a_3, \dots \dots \dots a_k]$, The output systematic code word (n, k) is:

$$C = [a_1, a_2, a_3, \dots \dots \dots a_k, c_1, c_2, c_3, \dots \dots \dots c_r]$$

However if the data bits are spread or changed at the output code word then, the code is said to be nonsystematic. The output of nonsystematic code word of (n, k):

$$C = [c_2, a_1, c_1, a_3, a_2, c_3, \dots \dots \dots]$$

- **Hamming Distance (HD):** It is important parameter to measure the ability of error detection. It the number of bits that differ between any two codewords C_i and C_j denoted by d_{ij} . For a binary (n, k) code with 2^k possible codewords, then minimum HD (d_{\min}) is $\min(d_{ij})$, where:

$$n \geq d_{ij} \geq 0$$

For any code word, the possible error detection is:

$$2t = d_{\min} - 1$$

For example, if $d_{\min} = 4$, then it is possible to detect 3 errors or less. The possible error correcting is:

$$t = \frac{d_{\min} - 1}{2}$$

So that for $d_{\min} = 4$, it is possible to correct only one bit.

Example (2): Find the minimum HD between the following codewords. Also determine the possible error detection and the number of error correction bits.

$$C_1 = [100110011], C_2 = [111101100] \text{ and } C_3 = [101100101]$$

Solution: Here $d_{12} = 6, d_{13} = 4$ and $d_{23} = 3$, hence $d_{min} = 3$.

The possible error detection $2t = d_{min} - 1 = 2$.

The possible error correction $t = \frac{d_{min}-1}{2} = 1$.

- **Hamming Weight:** It is the number of 1's in the non-zero codeword C_i , denoted by w_i . For example the codewords of $C_1 = [1011100]$, $C_2 = [1011001]$, $w_1 = 4$, and $w_2 = 3$ respectively. If we have two valid codewords- all ones and all zeros, in this case $HD = w_i$

3- Parity check codes (Error detection):

It is a linear block codes (systematic codes). In this code, an extra bit is added for each k information and hence the code rate (efficiency) is $k/(k + 1)$. At the receiver if the number of 1's is odd then the error is detected. The minimum Hamming distance for this category is $d_{min} = 2$, which means that the simple parity code is a single-bit error-detecting code; it cannot correct any error. There are two categories in this type: even parity (ensures that a code word has an even number of 1's) and odd parity (ensures that a code word has an odd number of 1's) in the code word.

Example: an even parity-check code of (5, 4) which mean that, $k = 4$ and $n = 5$.

Data word	Code word	Data word	Code word
0010	00101	0110	01100

1010	10100	1000	10001
------	-------	------	-------

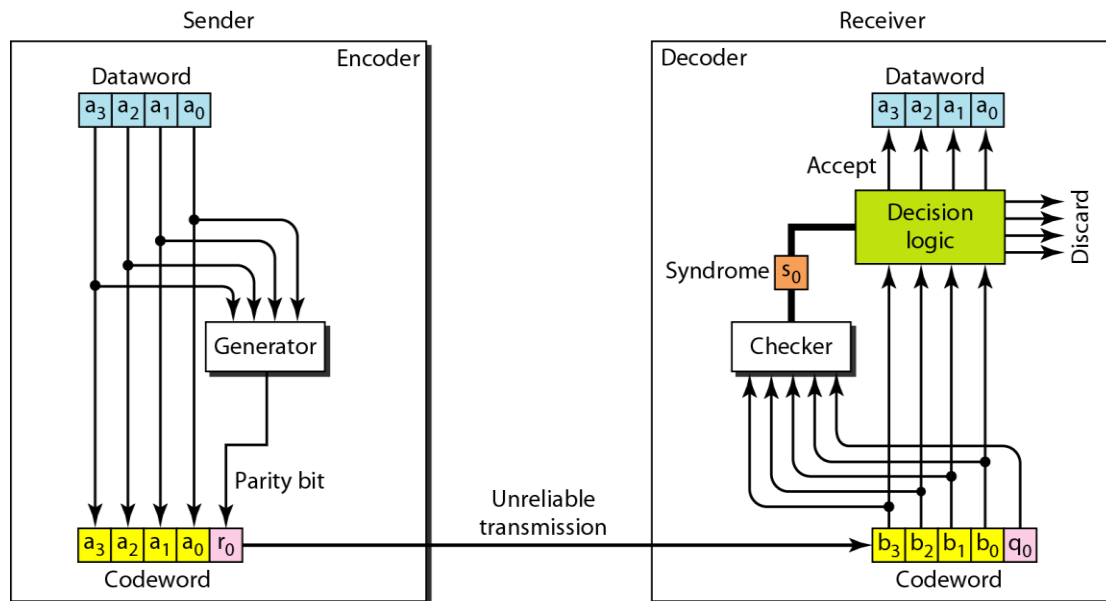
The above table can be repeated with odd parity-check code of (5, 4) as follow:

Data word	Code word	Data word	Code word
0010	00100	0110	01101
1010	10101	1000	10000

Note:

Error detection was used in early ARQ (Automatic Repeat on Request) systems. If the receiver detects an error, it asks the transmitter (through another backward channel) to retransmit.

The sender is calculate the parity bit to be added to the data word to form a code word. At the receiver, a syndrome is calculated. The syndrome is passed to the decision logic analyzer. If the syndrome is 0, there is no error in the received codeword; the data portion of the received codeword is accepted as the data word; if the syndrome is 1, the data portion of the received codeword is discarded. The data word is not created as shown in figure below.



4- Repetition codes:

The repetition code is one of the most basic error-correcting codes. The idea of the repetition code is to just repeat the message several times. The encoder is a simple device that repeats, r times.

For example, if we have a (3, 1) repetition code, then encoding the signal $m=101001$ yields a code $c=111000111000000111$.

Suppose we received a (3, 1) repetition code and we are decoding the signal $c=110001111$. The decoded message is $m=101$. For $(r, 1)$ repetition code an error correcting capacity of $r/2$ (i.e. it will correct up to $r/2$ errors in any code word). In other word the $d_{min} = r$, or increasing the correction capability depending on r value. Although this code is very simple, it also inefficient and wasteful because using only (2, 1) repetition code, that would mean we have to double the size of the bandwidth which means doubling the cost.

5- Linear Block Codes:

Linear block codes extend of parity check code by using a larger number of parity bits to either detect more than one error or correct for one or more errors. A block codes of an (n, k) binary block code can be selected a 2^k codewords from 2^n possibilities to form the code, such that each k bit information block is uniquely mapped to one of these 2^k codewords. In linear codes the sum of any two codewords is a codeword. The code is said to be *linear* if, and only if the sum of $V_i (+) V_j$ is also a code vector, where V_i & V_j are codeword vectors and $(+)$ represents modulo-2 addition.

5-1 Hamming Codes

Hamming codes are a family of linear error-correcting codes that generalize the Hamming (n, k) -code, and were invented by Richard Hamming in 1950. Hamming codes can detect up to two-bit errors or correct one-bit errors without detection of uncorrected errors. Hamming codes are perfect codes, that is, they achieve the highest possible rate for codes with their block length and minimum distance of three.

In the codeword, there are k data bits and $r = n - k$ redundant (check) bits, giving a total of n codeword bits. $n = k + r$

Hamming Code Algorithm:

General algorithm for hamming code is as follows:

1. r parity bits are added to an k - bit data word, forming a code word of n bits .
2. The bit positions are numbered in sequence from 1 to n .

3. Those positions are numbered with powers of two, reserved for the parity bits and the remaining bits are the data bits.
4. Parity bits are calculated by XOR operation of some combination of data bits. Combination of data bits are shown below following the rule.
5. It is characterized by $(n, k) = (2^m - 1, 2^m - 1 - m)$, where $m = 2, 3, 4 \dots$

5-2 Examl: Hamming(7,4)

This table describes which parity bits cover which transmitted bits in the encoded word. For example, p_2 provides an even parity for bits 2, 3, 6, and 7. It also details which transmitted bit is covered by which parity bit by reading the column. For example, d_1 is covered by p_1 and p_2 but not p_3 . This table will have a striking resemblance to the parity-check matrix (**H**).

Bit #	1	2	3	4	5	6	7
Transmitted bit	p_1	p_2	d_1	p_3	d_2	d_3	d_4
p_1	Yes	No	Yes	No	Yes	No	Yes
p_2	No	Yes	Yes	No	No	Yes	Yes
p_3	No	No	No	Yes	Yes	Yes	Yes

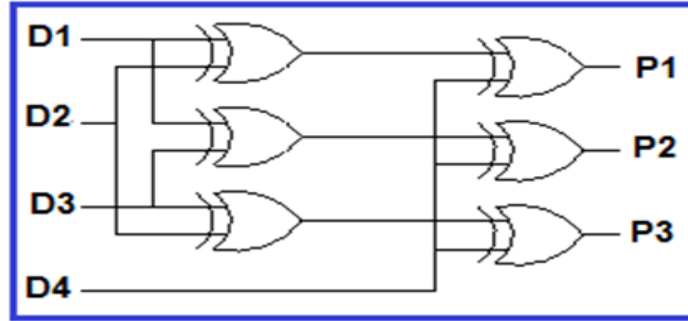
Or it can be calculate the parity bits from the following equations:

$$p_1 = d_1 \oplus d_2 \oplus d_4$$

$$p_2 = d_1 \oplus d_3 \oplus d_4$$

$$p_3 = d_2 \oplus d_3 \oplus d_4$$

The parity bits generating circuit is as following:



At the receiver, the first step in error correction, is to calculate the syndrome bits which indicate there is an error or no. Also, the value of syndrome determine the position detecting using syndrome bits = **CBA**. The equations for generating syndrome that will be used in the detecting the position of the error are given by:

$$A = p_1 \oplus d_1 \oplus d_2 \oplus d_4$$

$$B = p_2 \oplus d_1 \oplus d_3 \oplus d_4$$

$$C = p_3 \oplus d_2 \oplus d_3 \oplus d_4$$

Example:

Suppose we want to transmit the data 1011 over noisy communication channel. Determine the Hamming code word.

Solution:

The first step is to calculate the parity bit value as follow and put it in the corresponding position as follow:

$$p_1 = d_1 \oplus d_2 \oplus d_4 = 1 \oplus 0 \oplus 1 = 0$$

$$p_2 = d_1 \oplus d_3 \oplus d_4 = 1 \oplus 1 \oplus 1 = 1$$

$$p_3 = d_2 \oplus d_3 \oplus d_4 = 0 \oplus 1 \oplus 1 = 0$$

Bit position	1	2	3	4	5	6	7
--------------	---	---	---	---	---	---	---

Bit name	p_1	p_2	d_1	p_3	d_2	d_3	d_4
Received value	0	1	1	0	0	1	1

The the codeword is $c=0110011$

Suppose the following noise is added to the code word, then the received code becomes as:

The noise: $n = 0000100$

The received code word: $c_r = 0000100 \oplus 0110011 = 0110\textcolor{red}{1}11$

Now calculate the syndrome:

$$A = p_1 \oplus d_1 \oplus d_2 \oplus d_4 = 0 \oplus 1 \oplus 1 \oplus 1 = 1$$

$$B = p_2 \oplus d_1 \oplus d_3 \oplus d_4 = 1 \oplus 1 \oplus 1 \oplus 1 = 0$$

$$C = p_3 \oplus d_2 \oplus d_3 \oplus d_4 = 0 \oplus 1 \oplus 1 \oplus 1 = 1$$

So that $CBA = 101$ which indicate that an error in the fifth bit.

Hamming matrices:

Hamming codes can be computed in linear algebra terms through matrices because Hamming codes are linear codes. For the purposes of Hamming codes, two **Hamming matrices** can be defined: the **code generator matrix G** and the **parity-check matrix H**

	d_1	d_2	d_3	d_4
p_1	Yes	Yes	No	Yes
p_2	Yes	No	Yes	Yes
p_3	No	Yes	Yes	Yes

$$\mathbf{G} := \begin{pmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Bit #	1	2	3	4	5	6	7
Transmitted bit	p_1	p_2	d_1	p_3	d_2	d_3	d_4
p_1	Yes	No	Yes	No	Yes	No	Yes
p_2	No	Yes	Yes	No	No	Yes	Yes
p_3	No	No	No	Yes	Yes	Yes	Yes

$$\mathbf{H} := \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}$$

Example:

Suppose we want to transmit the data 1011 over noisy communication channel

$$\mathbf{x} = \mathbf{G}\mathbf{p} = \begin{pmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 2 \\ 3 \\ 1 \\ 2 \\ 0 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}$$

This mean that 0110011 would be transmitted instead of transmitting 1011. If no error occurs during transmission, then the received codeword \mathbf{r} is identical to the transmitted codeword \mathbf{x} : $\mathbf{r} = \mathbf{x}$. The receiver multiplies \mathbf{H} and \mathbf{r} to obtain the **syndrome** vector \mathbf{z} which indicates whether an error has occurred, and if so, for which codeword bit.

$$\mathbf{z} = \mathbf{H}\mathbf{r} = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 2 \\ 4 \\ 2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

suppose we have introduced a bit error on bit 5

$$\mathbf{r} = \mathbf{x} + \mathbf{e}_5 = \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

$$\mathbf{z} = \mathbf{H}\mathbf{r} = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 3 \\ 4 \\ 3 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}$$