Information Channels

So far, we have been considering an information source. Now, we shall start shifting our attention to information channels (also called communication channels). That is, we shall start moving from information generation to information transmission.

**Definition:** An information channel consists of an input alphabet $A = \{a_1, a_2, \ldots, a_n\}$, an output alphabet $B = \{b_1, b_2, \ldots, b_m\}$, and a set of conditional probabilities $P(b_j | a_i)$ for all $i, j$. $P(b_j | a_i)$ is the conditional probability that output symbol $b_j$ is received given that input symbol $a_i$ is sent.

The channel is said to be memoryless if the probability of a given output depends only on the input at that time, and is conditionally independent of previous inputs or outputs.

![Memoryless Channel Diagram](image)

Define $P_{ij} = P(Y = b_j | X = a_i) = P(b_j | a_i)$.

Then, we can form a matrix of transition (conditional) probabilities $P$:

$$
P = \begin{bmatrix}
P_{11} & P_{12} & \cdots & P_{1n} \\
P_{21} & P_{22} & \cdots & P_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
P_{n1} & P_{n2} & \cdots & P_{nn}
\end{bmatrix}
$$

The matrix $P$ is called a channel matrix. An information channel is completely specified by giving its channel matrix. If the rows are permutations of each other, we say that $P$ is input-uniform. If the columns are permutations of each other, $P$ is output-uniform.
The Binary Symmetric Channel (BSC)

Definition: A channel is symmetric if the rows of its channel matrix $P$ are permutations of each other, and the columns are also permutations of each other. Also, $P$ is square for a symmetric channel. E.g. the channel with the following channel matrix is symmetric:

$$
P = \begin{bmatrix}
0.3 & 0.2 & 0.5 \\
0.5 & 0.3 & 0.2 \\
0.2 & 0.5 & 0.3
\end{bmatrix}
$$

The binary symmetric channel is a symmetric channel with only two (binary) input symbols ($a_1 = 0, a_2 = 1$) and two output symbols ($b_1 = 0, b_2 = 1$). The input symbols are complemented with probability $p$, (i.e. the probability of error is $p$). The channel matrix for the BSC is given by:

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

Probability Relations in a Channel

It is possible for us to derive some useful relationships between the input and output probabilities using the channel matrix. Given the channel matrix $P$

$$
P = \begin{bmatrix}
p_{11} & p_{12} & \cdots & p_{1n} \\
p_{21} & p_{22} & \cdots & p_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
p_{n1} & p_{n2} & \cdots & p_{nn}
\end{bmatrix}
$$

We note that each row in the channel matrix corresponds to an input of the channel, while each column corresponds to an output. Therefore, for any given row, the sum must be equal to 1. That is:

$$
\sum_{i=1}^{n} p_{ii} = 1
$$

We observe that, if $a_1$ is sent, $b_1$ will occur with probability $p_{11}$, if $a_2$ is sent, $b_1$ will occur with probability $p_{21}$, etc. Thus,

$$
P(a_1)p_{11} + P(a_2)p_{21} + \cdots + P(a_n)p_{n1} = P(b_1)
$$

$$
P(a_1)p_{12} + P(a_2)p_{22} + \cdots + P(a_n)p_{n2} = P(b_2)
$$

$$
\vdots
$$

$$
P(a_1)p_{1n} + P(a_2)p_{2n} + \cdots + P(a_n)p_{nn} = P(b_n)
$$

More generally,

$$
P(b_i) = \sum_{a_j} P(a_j)p_{ji} = \sum_{a_j} P(a_j)P(b_i|a_j)
$$
More on Channel Probability Relations ...

We can also calculate some other important sets of probabilities associated with an information channel. By Baye’s Law, we can obtain the conditional probability of an input $a_i$ given that an output $b_j$ has been received:

$$P(a_i | b_j) = \frac{P(b_j | a_i)P(a_i)}{P(b_j)}$$

That is,

$$P(a_i | b_j) = \frac{P(b_j | a_i)P(a_i)}{\sum_{a_i} P(b_j | a_i)P(a_i)}$$

The numerator above is simply the joint probability of $a_i$ and $b_j$, i.e. $P(a_i, b_j)$. Usually, $P(b_j | a_i)$ - the entries in the channel matrix are called the forward probabilities, while $P(a_i | b_j)$ are called the backward probabilities.

Since the inputs $X$ and outputs $Y$ are probability events, we can also consider them using the concepts of entropy and mutual information.

---

Example - on probabilities associated with a channel

Assume we are given a binary channel with input probabilities $P(a = 0) = \frac{2}{3}$, and $P(a = 1) = \frac{1}{3}$, and the following channel matrix:

$$P = \begin{bmatrix}
\frac{2}{3} & \frac{1}{3} \\
\frac{1}{3} & \frac{2}{3}
\end{bmatrix}$$

We can get the following output probability:

$$P(b = 0) = (\frac{2}{3})(\frac{2}{3}) + (\frac{1}{3})(\frac{1}{3}) = \frac{5}{9}$$

Similarly,

$$P(b = 1) = \frac{4}{9}$$

The input conditional probabilities (i.e. backward probabilities) will be:

$$P(a = 0 | b = 0) = \frac{(\frac{2}{3})(\frac{2}{3})}{(\frac{5}{9})} = \frac{4}{5}$$

$$P(a = 1 | b = 1) = \frac{(\frac{1}{3})(\frac{1}{3})}{(\frac{4}{9})} = \frac{1}{4}$$

Verify that

$$P(a = 1 | b = 0) = \frac{1}{3}; \ P(a = 0 | b = 1) = \frac{2}{3}.$$  

We can also obtain the joint probabilities. For example,

$$P(a = 0, b = 0) = P(a = 0 | b = 0)P(b = 0) = (\frac{4}{5})(\frac{5}{9}) = \frac{2}{3}$$

Obtain the other joint probabilities.
Noiseless, Deterministic and Useless Channels

**Definition:** A channel whose channel matrix has *one and only one* non-zero element in each column is called a **noiseless channel**. Example

\[ P = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \]

For the noiseless channel, the output completely specifies the input. So, \( H(Y|X) = 0 \), or \( I(X;Y) = H(X) \).

**Definition:** A channel whose channel matrix has *one and only one* non-zero element in each row is called a **deterministic channel**.

\[ P = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \]

For the deterministic channel, the input completely specifies the output. That is, \( H(Y|X) = 0 \), or \( I(X;Y) = H(Y) \).

**Definition:** A channel whose channel matrix has identical rows is called a **useless channel**.

\[ P = \begin{bmatrix} 1 & p & p \\ 1 & p & p \end{bmatrix} \]

For the useless channel, knowledge of the output does not reduce the uncertainty in the input.

Hence, \( H(Y|X) = H(X) \); or \( I(X;Y) = 0 \).

A zero mutual information is expected for a channel that is useless:-)
More on Channel Capacity …

Recall that \( I(X; Y) = H(Y) - H(Y|X) \). Also \( I(X; Y) = H(X) - H(X|Y) \).

To compute the channel capacity, we use the former rather than the letter for convenience of computation.

Now \( H(Y|X) \) is given by

\[
H(Y|X) = - \sum_i \sum_j p_{ij} p_j \log p_j
\]

If the channel is input-uniform, the value of the second \( \Sigma \) will be equal for all \( i \), and hence independent of \( i \). For convenience, we replace the \( i \) by 1. Also, noting that \( \Sigma p_i = 1 \), we have

\[
H(Y|X) = - \sum_j p_j \log p_j
\]

Thus, we can write

\[
C = \max_p \{ H(Y) + \sum p_i \log p_i \}
\]

If this channel is both input-uniform and output-uniform, \( H(Y) \) will take its maximum value when \( q_1 = q_2 = \ldots = q_a = \frac{1}{a} \). That is, when \( p_1 = p_2 = \ldots = p_a = \frac{1}{a} \). The result will be:

\[
C = \log a + \sum p_i \log p_i
\]

With the above formulation, it is easy to observe that the capacity of a channel is independent of the input probabilities, but depends mainly on the channel matrix. That is, the capacity does not change by the way the channel is used.

Example - The Capacity of the Binary Symmetric Channel

The BSC is both input and output uniform.

Its channel capacity is given by

\[
C = 1 + p \log p + (1 - p) \log(1 - p)
\]

We can write this as:

\[
C = 1 - H(p)
\]

This capacity is depicted schematically below.

The maximum capacity is attained when \( p = 0 \) or \( p = 1 \), and minimum 0 when \( p = 1/2 \). If \( p = 0 \), we have no error, and the channel has its full capacity. When \( p = 1 \), the input symbol is completely inverted, from 0 to 1 and from 1 to 0. In this case the channel has its full capacity. At the output, we can just invert the symbol. When \( p = 1/2 \), the capacity is 0. That is, if input symbols change with probability 1/2, we can not guess anything about the input. The channel can not carry any information.
Error Probabilities and Decision Rules

The channel matrix gives us an idea of the error we can expect in transmitting information through the channel. Thus, given the channel matrix, we can talk of the probability of error. In practice, the error probability depends on how the receiver interprets the output symbols from the channel.

Suppose we are given the following channel matrix:

\[
P = \begin{bmatrix} 0.5 & 0.3 & 0.2 \\ 0.2 & 0.3 & 0.5 \\ 0.3 & 0.3 & 0.4 \end{bmatrix}
\]

We have three inputs \(\{a_1, a_2, a_3\}\) and three output symbols \(\{b_1, b_2, b_3\}\). The problem is, given that we have received some output symbol, which of the possible input symbols should we assign the output?

We need some rules to help us out …

Definition: Given an input alphabet \(A = \{a_1, a_2, ..., a_s\}\) and an output alphabet \(B = \{b_1, b_2, ..., b_r\}\), a \textit{decision rule} is a function \(f(b_i)\), that specifies a unique input symbol for each output symbol.

Using the channel matrix above, two possible decision rules could be:

\begin{itemize}
  \item \textbf{Decision Rule 1}: \(f(b_1) = a_1; f(b_2) = a_2; f(b_3) = a_3\);
  \item \textbf{Decision Rule 2}: \(f(b_1) = a_1; f(b_2) = a_2; f(b_3) = a_3\);
\end{itemize}

We observe that for a channel with \(s\)-inputs and \(r\)-outputs, we will have \(r^s\) different possible decision rules. The question then becomes which one of the \(r^s\) possible rules to select.

Decision Rules and Decoding

When we choose a certain decision rule, and apply the rule on a certain output symbol, we say that we have \textit{decoded} the output symbol. If the result from the decoding is not in the original input symbol set \(A\), we say that a \textit{decoding error} has occurred.

To choose the decision rule, we can safely assume that the objective is to minimize the channel error. Minimization of the channel error is thus equivalent to maximization of the probability of correct decoding.

Let \(P_e\) be the probability of channel error. We denote this as the average of \(P(E|b_i)\), the probability of error, given that the output symbol is \(b_i\). That is,

\[
P_e = P(E) = \sum_{b_i} P(E|b_i)
\]

Since the decision rule \(f(b_i)\) is independent of \(P(b_i)\), we can minimize \(P_e\), by selecting the \(f(b_i)\) that will minimize the individual conditional probability of error, \(P(E|b_i)\).

Suppose we choose some fixed decision rule: \(f(b_i) = a_i\). Then,

\[
P(E|b_i) = 1 - P(f(b_i) = a_i)
\]

Since the decision rule is fixed, \(P(f(b_i) = a_i)\) is simply the backward probability, \(P(a_i|b_i)\). Thus, to minimize the error as given in (**), we can choose \(f(b_i) = a_i^*\), where \(a_i^*\) is defined such that

\[
P(a_i^*|b_i) \geq P(a_i|b_i), \quad \forall i
\]

That is, the channel error probability is minimized if we use a decision rule that, for each output symbol, chooses the input symbol with the highest transition probability. This scheme is called the \textit{maximum-likelihood} decision rule.

Example. Using the previous example, one possible maximum-likelihood decision rule will be:

\textbf{Maximum-Likelihood Decision Rule}: \(f(b_1) = a_1; f(b_2) = a_2; f(b_3) = a_3\);

We also observe that this decision rule is not unique. We can still find other maximum-likelihood decision rules for the above channel.
Improving Message Reliability

One simple method we can use to increase the reliability of our messages when using the unreliable channel is by repeating the message several times. This is based on the fact that with repeated messages, since the input symbols are independent, the probability of channel error decreases with increasing length of the message.

For example, with the binary symmetric channel, with error probability \( p \), the probability of \( k \)-symbol errors in an \( n \)-symbol sequence, will be:

\[
P(\text{exactly } k \text{-symbol errors}) = \binom{n}{k} p^k (1-p)^{n-k}
\]

The example below repeats the message (0 or 1) 5 times using the BSC.

<table>
<thead>
<tr>
<th>Unused</th>
<th>Messages</th>
<th>Outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>00000</td>
<td>00000</td>
</tr>
<tr>
<td></td>
<td>00010</td>
<td>00010</td>
</tr>
<tr>
<td></td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td></td>
<td>11110</td>
<td>11110</td>
</tr>
<tr>
<td></td>
<td>11111</td>
<td>11111</td>
</tr>
</tbody>
</table>

Using maximum-likelihood decision rule, the probability of error will be

\[
P_e = P(5\text{-symbol errors}) + P(4\text{-symbol errors}) + P(3\text{-symbol errors})
\]

At \( p = 0.01 \), we have, \( P_e = 10^{-7} \).

Therefore we can reduce the error to as small as we wish by simply repeating the message.

Of course, there is always a price!

We have merely exchanged (information) message transmission rate for message reliability. A typical plot of this price will show that the message rate falls rapidly as the reliability increases.

So, can we do better? Shannon again has something to offer! An offer that is in fact very counter intuitive!

Before then, we look at one more notion - the Hamming distance.

The Hamming Distance

Definition. The **Hamming distance** \( D(x, y) \) of two code words \( x \) and \( y \) is the number of places where their binary values are different.

Example. For \( x = 1001 \) and \( y = 0100 \), \( D(x, y) = 3 \). That is, \( x \) and \( y \) are different at 3 bits.

The Hamming distance is a **metric**. That is, given \( x, y, \) and \( z \), we have the following properties for the Hamming distance:

1. **Always positive**:
   \[ D(x, y) \geq 0, \quad D(x, y) = 0 \iff x = y \]
2. **Symmetry**:
   \[ D(x, y) = D(y, x) \]
3. **Triangle Inequality**:
   \[ D(x, z) \leq D(x, y) + D(y, z) \]

Suppose we send 0 and 1 through an unreliable channel. Let us expand 0 and 1 to triple the size. That is, 0 is encoded into 000 and 1 is encoded into 111, and the encoded words are sent through the channel. If only one error occurs, 010 will be decoded to 0 and 011 will be decoded to 1, etc. See below.

\[
\begin{align*}
0 & \rightarrow 000 \quad \text{channel} \quad \rightarrow 010 \rightarrow 0 \\
1 & \rightarrow 111 \quad \text{channel} \quad \rightarrow 011 \rightarrow 1
\end{align*}
\]

In summary we have the following decoding scheme.

<table>
<thead>
<tr>
<th>Received words</th>
<th>Decoded symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td>{000, 001, 010, 100}</td>
<td>0</td>
</tr>
<tr>
<td>{111, 110, 101, 011}</td>
<td>1</td>
</tr>
</tbody>
</table>

We state that: a decision rule \( f(y) \) that assigns a received message \( y \) to a code word \( x \) if \( D(x, y) = \min[D(y, x)] \) is called a **nearest neighbour** decision rule, or **minimum (Hamming) distance** decision rule. This is also a maximum-likelihood decision rule.
More on Hamming Distance …

In general, a subset of \(2^n\) binary words of length \(n\) can be viewed as a code. To distinguish from the symbol C for capacity, let us use CODE to denote a code. The code words which are elements of CODE are sent through a communication channel and corrupted by some errors. After we receive a corrupted word, the question is whether we can recover the original code word, that is, decoding. In the above example, CODE = \{000, 111\} for \(n=3\), and a received word with one-bit error can be successfully decoded. The whole space S of received words is the set of all \(2^n\) words.

The set of words which can be decoded to a word \(w\) in CODE is called the decodable region of \(w\), denoted by \(D(w)\). Obviously \(D(v)\) and \(D(w)\) must be mutually disjoint for different \(v\) and \(w\) in CODE. If we allow \(k\) errors, \(D(w)\) is the set of words whose Hamming distance from \(w\) is \(k\). In the above example, we allow one bit error, and \(D(000) = \{000, 001, 010, 100\}\) and \(D(111) = \{111, 110, 101, 011\}\). Thus the whole space is classified into \(D(w)\) for some \(w\). The region that does not belong to any \(D(w)\) is called error detectable region, since in this region we can not know from where the received word was modified by the error and thus cannot detect that some error occurred. In the example above, the whole space is divided into decodable regions and there is no error detectable region.

Shannon’s Second Theorem (The Noisy Coding Theorem)

Now let us choose \(M\) words to encode source symbols \(a_1, ..., a_k\). If we assume that those source symbols appear with equal probability of \(\frac{k}{n}\) and there is no error in the channel, we can transmit information with the rate of

\[
R = \frac{\log M}{n} \text{ (bits/symbol)}.
\]

In this unit of (bits/symbol), symbol means a symbol in the channel. Note that \(R\) is between 0 and 1. If we assume that we can send one symbol per second through the channel, the meaning of \(R\) becomes the speed of information transmission. If we use the whole space for code words, \(R\) becomes 1, most efficient, but we can not allow any error. Note that \(M = 2^n\). By making \(R < 1\), we can send information through a noisy channel and recover the message at the receiver side.

Shannon’s second theorem relates information transmission rate \(R\), and the channel capacity, \(C\). The proof of the Theorem is somewhat involving. Here, we state it without proof, and offer some explanations.

**Theorem. (Shannon’s Second Theorem - The Noisy Channel Coding Theorem)**

Let a channel with capacity \(C\) be given. For a transmission rate \(R\), if \(R < C\), there exists a code whose transmission rate is \(R\) and error probability for decoding is less than any small number \(\epsilon > 0\). If \(R > C\), there is no such code.

The theorem says that there exists a code that can achieve nearly the channel capacity for transmission rate with almost 0 error (i.e. as little error as we may wish!). That is, we can achieve both efficiency and reliability at the same time!

Note that, once again, the theorem talks about the existence of such a code.

It does not provide any systematic method for constructing such a code. This theorem has thus generated a huge interest, and a lot of effort has been invested in finding such an efficient and reliable code, especially based on algebraic methods.

As we stand here today, we are still searching for this ultimate code!