EE 591: Introduction to Information Theory

NOTES-01:
INTRODUCTION & INFORMATION MEASURES
Course Schedule:
Tuesday and Thursday 9:15 – 10:45am, Venue: ESB 355

Course Coverage
♦ Measurement of Information
♦ Coding for Data Compression
♦ Coding for Error Protection
♦ Coding for Data Encryption

References:
Required

Recommended

Others
3. Materials to be handed in class

Contacts:
Don Adjeroh, don@csee.wcu.edu
Rm ESB 937, Tel: 293-0405 x 7869

Course Tutor: TBA

Assessment      Worth  |  Release Date
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>3 Assignments</td>
<td>10%</td>
</tr>
<tr>
<td></td>
<td>10%</td>
</tr>
<tr>
<td></td>
<td>15%</td>
</tr>
<tr>
<td>Project</td>
<td>30%</td>
</tr>
<tr>
<td>Final Test (open-book)</td>
<td>35%</td>
</tr>
</tbody>
</table>

Grading
<p>| |</p>
<table>
<thead>
<tr>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A ≥ 85</td>
</tr>
<tr>
<td>B 75-84</td>
</tr>
<tr>
<td>C 60-74</td>
</tr>
<tr>
<td>D 50-64</td>
</tr>
<tr>
<td>F &lt; 50</td>
</tr>
</tbody>
</table>

1 Although it has an EE code, it is also acceptable for CS graduate credit
Different views on Coding

- Coding for Efficiency - Data compression (information theory)
  Encoding such that the source data takes up a minimal storage space.
  Saves both transmission time, and storage space.

- Coding for Error Protection (coding theory)
  Encoding such that we can detect errors in the stored or transmitted data.
  Sometimes we may also want to correct the error.

- Coding for Data Security - encryption (cryptology)
  Encoding to protect the data from unauthorized use.
Information Theory and other fields
Introduction

Suppose we express message symbols from some data source using the binary symbols (1 or 0). The correspondence from message symbols to binary symbols is called a code. Consider the following example.

\[
\begin{array}{c|c}
  s_1 & 0 \\
  s_2 & 01 \\
  s_3 & 001 \\
  s_4 & 111 \\
\end{array}
\]

We say that symbol $s_2$ is encoded into codeword 01, etc. As long as we have a one-to-one correspondence, we can know what message symbol was encoded to a particular codeword.

Since we have to encode a sequence of message symbols, we have the following awkward situation with encoded sequence 111001. This sequence can be obtained from either $s_4s_3$ or $s_4s_1s_2$. In other words, this code is not uniquely decodable.
Consider the following two examples.

<table>
<thead>
<tr>
<th>(s_1)</th>
<th>0</th>
<th>(s_1)</th>
<th>00</th>
</tr>
</thead>
<tbody>
<tr>
<td>(s_2)</td>
<td>10</td>
<td>(s_2)</td>
<td>01</td>
</tr>
<tr>
<td>(s_3)</td>
<td>110</td>
<td>(s_3)</td>
<td>10</td>
</tr>
<tr>
<td>(s_4)</td>
<td>1110</td>
<td>(s_4)</td>
<td>11</td>
</tr>
</tbody>
</table>

Both codes are uniquely decodable. In the left example, 0 is the end of one code word. In the right example, we can divide the encoded sequence into subsequences of length 2, and can determine the message symbols. Hereafter we consider only uniquely decodable codes. We call a code like the one on the right an **equal-length** code.

Equal-length codes seem to serve our communication purposes. In reality, the frequencies of message symbols, or equivalently the probability of occurrence may not be equal over all message symbols.
Introduction …

How about the problem of sending information about the weather in San Francisco to New York using binary signals for a few days?

<table>
<thead>
<tr>
<th>Weather</th>
<th>Probability</th>
<th>CODE1</th>
<th>CODE2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sunny</td>
<td>1/4</td>
<td>00</td>
<td>10</td>
</tr>
<tr>
<td>Cloudy</td>
<td>1/4</td>
<td>01</td>
<td>110</td>
</tr>
<tr>
<td>Rainy</td>
<td>1/4</td>
<td>10</td>
<td>1110</td>
</tr>
<tr>
<td>Foggy</td>
<td>1/4</td>
<td>11</td>
<td>0</td>
</tr>
</tbody>
</table>

Average length 2 2.5

Then the minimum average code length is 2.

Next consider the weather in Los Angeles.

<table>
<thead>
<tr>
<th>Weather</th>
<th>Probability</th>
<th>CODE1</th>
<th>CODE2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sunny</td>
<td>1/4</td>
<td>00</td>
<td>10</td>
</tr>
<tr>
<td>Cloudy</td>
<td>1/8</td>
<td>01</td>
<td>110</td>
</tr>
<tr>
<td>Rainy</td>
<td>1/8</td>
<td>10</td>
<td>1110</td>
</tr>
<tr>
<td>Smoggy</td>
<td>1/2</td>
<td>11</td>
<td>0</td>
</tr>
</tbody>
</table>

Average length 2 1.875

From this example we see that we could use CODE1 or CODE2 depending on the probability distribution. In the Los Angeles example, suppose we send sunny, smoggy, smoggy, cloudy. Then under CODE1, we have 00111101, and under CODE2, 1000110.

The question here is, what is the most efficient code for the given data source, assuming that we can characterise the source by its probability distribution?
**Measurement of Information**

Given an event E, with probability of occurrence P(E), we denote the information content by I(E).

The information content is usually related to the element of surprise.

An event with high probability of occurrence has little information. That is, a statement that says the very obvious does not carry much information.

An event with a very rare occurrence has more information. A statement that shows something new has more information.

In general, I(E) should increase with decreasing P(E).
More on Information Measurement …

We make the following reasonable assumptions

Assumption 1
\( I(E) \geq 0 \), for all events

Assumption 2
Given that \( E_1 \) and \( E_2 \) are independent events,
\[
I(E_1 \cap E_2) \geq I(E_1) \\
I(E_1 \cap E_2) \geq I(E_2)
\]

Assumption 3
\[
I(E_1 \cap E_2) = I(E_1) + I(E_2)
\]

The problem of measuring information is then to find a function that meets all the above three assumptions.

We measure the amount of information \( I(E) \) of an event \( E \) with probability \( P(E) \) by
\[
I(E) = \log \left( \frac{1}{P(E)} \right) = -\log(P(E))
\]

The base of the logarithm can be 2, 10, \( e \), or \( r \), where \( e = 2.72 \ldots \) is the base for natural logarithm, \( r \) is an arbitrary integer. In each case, the amount is called \textbf{bits}, \textbf{Hartleys}, \textbf{nats}, or \textbf{r-ary units} respectively. We will mainly be concerned with \textbf{base 2}, and hence we shall use bits as the units for information measurement.

If need be, we can transform from one base to the other with the formula:
\[
\log_a(X) = \frac{\log_b(X)}{\log_b(a)}
\]
Example - on quantity of information

Is it really true that a picture is worth more than a thousand words? Let's find out …

Imagine that we have a high definition TV with a pixel resolution of 1024x1024, and images are stored with 256 grey levels.

Then, we have a total of \( 256^{1024 \times 1024} = (2^8)^{1024 \times 1024} = 2^{8 \times 1024 \times 1024} \) possible images. Assume that each image is equally probable. Then the probability of any image to occur is: \( \frac{1}{2^{8 \times 1024 \times 1024}} \)

The information content will be \( I = -\log \left( \frac{1}{2^{8 \times 1024 \times 1024}} \right) = 8,388,608 \) bits!

Now, consider the information contained in a random speech of 1000 words, for instance from a TV announcer. We can safely assume a 10000 word vocabulary for our announcer. The probability of any sequence of 1000 words will be: \( \frac{1}{10000^{1000}} \)

Then, the amount of information here will be: \( -\log \left( \frac{1}{10000^{1000}} \right) = 13,288 \) bits

So, a picture is indeed worth more than a thousand words :-)


Entropy

The entropy of an information source is the expectation on the information from the source.

Suppose we have an information source S, from which message symbols $s_1, ..., s_n$ are emitted with probabilities $P(s_1), ..., P(s_n)$. Then the entropy of the information source S is defined as:

$$H(S) = \sum_{i=1}^{n} P(s_i) I(s_i) = -\sum_{i=1}^{n} P(s_i) \log P(s_i)$$

Example.
The entropy of the weather in Los Angeles (using the previous tables) is given by

$$H(S) = \frac{1}{4} \log 4 + \frac{1}{8} \log 8 + \frac{1}{8} \log 8 + \frac{1}{2} \log 2 = \frac{1}{2} + \frac{1}{4} + \frac{1}{2} = 1.75 \text{ bits}$$
Properties of Entropy

\[ H(S) \geq 0 \]

\[ H(S) \leq \log n \]

Combining, we have \( 0 \leq H(S) \leq \log n \)

**Lemma 1.** For any real number \( x \),
\[ \ln(x) \leq x - 1 \]
where \( \ln \) denotes the natural logarithm. Equality holds if and only if \( x = 1 \).

**Lemma 2.** Let \( p_i, (i = 1, 2, \ldots, n) \) and \( q_i, (i = 1, 2, \ldots, n) \) be the two sets of probabilities. Then, \[ \sum_{i=1}^{n} p_i \log \frac{1}{p_i} \leq \sum_{i=1}^{n} p_i \log \frac{1}{q_i} \], with equality holding iff \( p_i = q_i \) for all \( i \).
Proof.

Consider \( \sum_{i=1}^{n} p_i \log \frac{q_i}{p_i} \).

Since the sums of \( p_i \) and \( q_i \) are both 1, we have from Lemma 1

\[
\sum_{i=1}^{n} p_i \log \frac{q_i}{p_i} \leq \frac{1}{\ln 2} \sum_{i=1}^{n} p_i \left( \frac{q_i}{p_i} - 1 \right) \quad (**)
\]

\[
= \frac{1}{\ln 2} \sum_{i=1}^{n} (q_i - p_i)
\]

\[
= \frac{1}{\ln 2} \left( \sum_{i=1}^{n} q_i - \sum_{i=1}^{n} p_i \right) = 0
\]

Thus, \( \sum_{i=1}^{n} p_i \log \frac{q_i}{p_i} \leq 0 \)

But, \( \log \frac{q_i}{p_i} = \log \left( \frac{1}{p_i} \right) - \log \left( \frac{1}{q_i} \right) \)

Thus, we can write:

\[
\sum_{i=1}^{n} p_i \log \frac{q_i}{p_i} = \sum_{i=1}^{n} p_i \log \left( \frac{1}{p_i} \right) - \sum_{i=1}^{n} p_i \log \left( \frac{1}{q_i} \right) \leq 0
\]

From line (***) above, we see that equality holds iff equality also holds in Lemma 1.

This happens iff \( \frac{q_i}{p_i} = 1 \), i.e. \( q_i = p_i \), \( \forall i \) and \( q_i \).
**Bounds on Quantity of Information**

We can then turn to the main result:

**Theorem 1 (Bounds on Entropy).**
For an information source S of size $n$, with symbols $s_i, (i = 1, 2, ..., n)$, and symbol probabilities $p_i, (i = 1, 2, ..., n)$, the entropy satisfies the following:

$$0 \leq H(S) \leq \log n.$$ 

The lower limit is attained iff one of the source symbols occurs with probability 1. The upper limit is achieved iff we have a source whose symbol probability is uniformly distributed.

**Proof.**
Let $q_i, (i = 1, 2, ..., n)$ be another set of probabilities, (that is, $\sum q_i = 1$), from a uniform distribution. That is, $q_i = \frac{1}{n}, (i = 1, 2, ..., n)$.

By Lemma 2, we have

$$H(S) = \sum_{i=1}^{n} p_i \log \frac{1}{p_i} \leq \sum_{i=1}^{n} p_i \log \frac{1}{q_i}$$

$$= \sum_{i=1}^{n} p_i \log \left( \frac{1}{1/n} \right)$$

$$= (\log n) \sum_{i=1}^{n} p_i$$

$$= \log n$$

How about the proof for the lower bound, $H(s) = 0$?
Binary Source

The simplest information source is the binary source. With probability \( p \) it emits symbol 1, while with probability \( 1-p \), it emits symbol 0. Then the entropy \( H(S) \) is given by

\[
H(S) = -(p \log p + (1-p) \log(1-p))
\]

This is called the entropy function, which is depicted below.
The maximum is at \( p = 1/2 \).

We define \( p \log p = 0 \) when \( p = 0 \). Thus the entropy function is 0 at \( p=0 \) and 1, maximum 1 at \( p = 1/2 \), and symmetric on opposite sides of \( p=1/2 \).

We notice that the max amount of information using \( n \) symbols increases slowly with \( n \). (in fact, as \( \log n \)). Thus, to double the maximum amount of information per symbol provided by a source of \( n \) symbols, we must increase the number of symbols to \( n^2 \).
Extensions of an Information Source

We say the given information source is of zero memory if message symbols $s_i$ are generated with probability $P(s_i)$, which are independent from one occurrence to the next. That is, the probability of a sequence of message symbols is given by the product of the probability of each symbol in the sequence.

The $k$-th extension $S^k$ of an information source $S$ is the set of message symbols taken $k$ at a time.

Definition 1.
Let $S$ be a zero-memory information source of size $n$, with symbols $s_i, (i=1,2,...,n)$, and symbol probabilities $P(s_i) = p_i, (i=1,2,...,n)$. Then, the $k$-th extension of $S$, denoted $S^k$, is another zero-memory source with $n^k$ symbols, $\{a_i, (i=1,2,...,n^k)\}$. Each $a_i$ corresponds to some sequence of $k$ symbols formed using the $s_i$'s. The probability of any symbol in $S^k$ is simply the probability of the corresponding sequence of $s_i$'s. That is, if $a_i$ corresponds to the sequence $(s_{i_{1}},s_{i_{2}},...,s_{i_{n}})$, then

$$P(a_i) = P(s_{i_{1}},s_{i_{2}},...,s_{i_{n}}) = P(s_{i_{1}})P(s_{i_{2}})...P(s_{i_{n}}) = p_{i_{1}}p_{i_{2}}...p_{i_{n}}.$$ 

Example. Let $S = \{s_1, s_2\}$ with probabilities $P(s_1) = \frac{1}{4}, P(s_2) = \frac{3}{4}$. Then the 2nd extension $S^2$ is given by $S^2 = \{s_1s_1, s_1s_2, s_2s_1, s_2s_2\}$ with probabilities $\{\frac{1}{16}, \frac{3}{16}, \frac{3}{16}, \frac{9}{16}\}$.

Note that the sum of the extended probabilities is 1.

Lemma 3. The sum of all the probabilities of the symbol sequences obtained by changing symbols in an extended symbol $\{x_1,x_2,...,x_k\}$ except for $x_i$ is equal to $P(x_i)$.

In the above example $P(s_1) = P(s_1s_1) + P(s_1s_2)$.
More on Extension of an Information Sources …

Theorem 2 (Entropy of an extended source).
Let $S$ be an information source, and let $S^k$ be its $k$-th extension. Then, $H(S^k) = kH(S)$.

Proof.

\[
H(S^k) = \sum_{i_1, i_2, \ldots, i_k} p_{i_1} p_{i_2} \ldots p_{i_k} \log \frac{1}{p_{i_1} p_{i_2} \ldots p_{i_k}}
\]

\[
= \sum_{i_1, i_2, \ldots, i_k} p_{i_1} p_{i_2} \ldots p_{i_k} \log \frac{1}{p_{i_1}} + \sum_{i_1, i_2, \ldots, i_k} p_{i_1} p_{i_2} \ldots p_{i_k} \log \frac{1}{p_{i_2}}
\]

\[+ \ldots + \sum_{i_1, i_2, \ldots, i_k} p_{i_1} p_{i_2} \ldots p_{i_k} \log \frac{1}{p_{i_k}} (**)
\]

The first summation gives:

\[
\sum_{i_1, i_2, \ldots, i_k} p_{i_1} p_{i_2} \ldots p_{i_k} \log \frac{1}{p_{i_1}} = \sum_{i_1=1}^{n} p_{i_1} \log \frac{1}{p_{i_1}} \left( \sum_{i_2=1}^{n} p_{i_2} \right) \ldots \left( \sum_{i_k=1}^{n} p_{i_k} \right) = H(S)
\]

Similarly, all the other $k-1$ summations in (**) will produce $H(S)$.

Considering all the $k$ sums, we have:

$H(S^k) = kH(S)$ as required.
More on Extension of a Source …

Example
Using the previous example, $S = \{s_1, s_2\}$ with probabilities $P(s_1) = \frac{1}{4}$, $P(s_2) = \frac{3}{4}$, and $S^2 = \{s_1s_1, s_1s_2, s_2s_1, s_2s_2\}$ with probabilities $\{\frac{1}{16}, \frac{3}{16}, \frac{3}{16}, \frac{9}{16}\}$.

We will have

$$H(S) = -\left(\frac{1}{4} \log \frac{1}{4} + \frac{3}{4} \log \frac{1}{4}\right) = 0.8112 \text{ bits}$$

$$H(S^2) = -\left(\frac{1}{16} \log \frac{1}{16} + \frac{3}{16} \log \frac{3}{16} + \frac{3}{16} \log \frac{3}{16} + \frac{9}{16} \log \frac{9}{16}\right) = 1.6225$$

What about $H(S^3)$?