ECE 566 - INFORMATION THEORY
Prof. Thinh Nguyen
LOGISTICS

- **Textbook:** Elements of Information Theory, Thomas Cover and Joy Thomas, Wiley, second edition

- **Class email list:** ece566-w20@engr.orst.edu
LOGISTICS

Grading Policy

- Quiz 1: 15%
- Quiz 2: 15%
- Quiz 3: 15%
- Quiz 4: 15%

Lowest score quiz will be dropped

- Homework: 15%
- Final: 40%
WHAT IS INFORMATION?

- Information theory deals with the concept of information, its measurement and its applications.
ORIGIN OF INFORMATION THEORY

Two schools of thoughts

- British
  - Semantic information: related to the meaning of messages
  - Pragmatic information: related to the usage and effect of messages

- American
  - Syntactic: related to the symbols from which messages are composed, and their interrelations
ORIGIN OF INFORMATION THEORY

Consider the following sentences
1. Jacqueline was awarded the gold medal by the judges at the national skating competition.
2. The judges awarded Jacqueline the gold medal at the national skating competition.
3. There is a traffic jam on I-5 between Corvallis and Eugene in Oregon
4. There is a traffic jam on I-5 in Oregon

(1) and (2): syntactically different but semantically and pragmatically identical.
(3) and (4): syntactically and semantically different, (3) gives more precise information than (4).
(3) and (4) are irrelevant for Californians.
ORIGIN OF INFORMATION THEORY

- The British tradition is closely related to philosophy, psychology and biology. Influenced scientists include MacKay, Carnap, Bar-Hillel, ...
  - Very hard if not impossible to quantify

- The American traditional deals with the syntactic aspects of information. Influenced scientists include Shannon, Renyi, Gallager and Csiszar, ...
  - Can be rigorously quantified by mathematics
ORIGIN OF INFORMATION THEORY

- Basic questions of information theory in the American tradition involve the measurement of syntactic information

- The fundamental limits on the amount of information which can be transmitted
- The fundamental limits on the compression of information which can be achieved
- How to build information processing systems approaching these limits?
ORIGIN OF INFORMATION THEORY

- H. Nyquist (1924) published an article wherein he discussed how messages (characters) could be sent over a telegraph channel with maximum possible speed, but without distortion.

- R. Hartley (1928) who first defined a measure of information as the logarithm of the number of distinguishable messages one can represent using \( n \) characters, each can take \( s \) possible letters.
For a message of length $n$, we have the Hartley’s measure of information as:

$$H_H(s^n) = \log(s^n) = n \log(s)$$

For a message of length of 1, we have the Hartley’s measure of information as:

$$H_H(s^1) = \log(s^1) = 1 \log(s)$$

This definition corresponds with the intuitive idea that a message consisting of $n$ symbols contains $n$ times as much information as a message consisting of only one symbol.
Note that any function $f(s^n) = nf(s)$ would satisfy our intuition. One can show that the only functions that satisfy such equation is of the form:

$$f(s) = \log_a(s)$$

The choice of $a$ is arbitrary, and is more a matter of normalization.
AND NOW..., THE SHANNON’S INFORMATION THEORY

(a.k.a communication theory, mathematical information theory, or in short as information theory)
“The fundamental problem of communication is that of reproducing at one point either exactly or approximately a message selected at another point.” (Claude Shannon 1948)

Channel Coding Theorem:
- It is possible to achieve near perfect communication of information over a noisy channel.

In this course we will:
- Define what we mean by information
- Show how we can compress the information in a source to its theoretically minimum value and show the tradeoff between data compression and distortion.
- Prove the Channel Coding Theorem and derive the information capacity of different channels
The little formula that starts it all

\[ H(X) = E[-\log_2(X)] = -\sum_{x \in A} p(x) \log_2 p(x) \]
SOME FUNDAMENTAL QUESTIONS
(that can be addressed by information theory)

What is the minimum number of bits that can be used to represent the following texts?

"There is a wisdom that is woe; but there is a woe that is madness. And there is a Catskill eagle in some souls that can alike dive down into the blackest gorges, and soar out of them again and become invisible in the sunny spaces. And even if he for ever flies within the gorge, that gorge is in the mountains; so that even in his lowest swoop the mountain eagle is still higher than other birds upon the plain, even though they soar."

- Moby Dick, Herman Melville
SOME FUNDAMENTAL QUESTIONS
(that can be addressed by information theory)

How fast can information be sent reliably over an error prone channel?

"There is a wisdom that is woe; but there is a woe that is madness. And there is a Catskill eagle in sme souls that can alike divi down into the blackst goages, and soar out of thea agein and become invisible in the sunny speces. And even if he for ever flies within the gorge, that gorge is in the mountains; so that even in his lotest swoap the mountin eagle is still highhr than other berds upon the plain, even though they soar."

- Moby Dick, Herman Melville
FROM QUESTIONS TO APPLICATIONS
SOME not so FUNDAMENTAL QUESTIONS
(that can be addressed by information theory)

How to make money? (Kelly’s Portfolio Theory)

What is the optimal dating strategy?
**Fundamental Assumption of Shannon’s Information Theory**

E.g., Newtonian universe is deterministic, all events can be predicted with 100% accuracy by Newton’s laws, given the initial conditions.

If there is no irreducible randomness, the description/history of our universe, therefore can be compressed into a few equations.
Fundamental Assumption of Shannon’s Information Theory

Probabilistic model

Some well-defined stochastic process that generates all the observations

Thus in certain sense, Shannon information theory is somewhat unsatisfactory since it is based on the “ignorant model.” Nevertheless, it is a good approximation, depending on how accurate the probabilistic model used to describe the phenomenon of interest.
"There is a wisdom that is woe; but there is a woe that is madness. And there is a Catskill eagle in some souls that can alike dive down into the blackest gorges, and soar out of them again and become invisible in the sunny spaces. And even if he for ever flies within the gorge, that gorge is in the mountains; so that even in his lowest swoop the mountain eagle is still higher than other birds upon the plain, even though they soar."

- *Moby Dick*, Herman Melville

We know the texts above is not random, but we can build a probabilistic model for it.

For example, a first order approximation could be to build a histogram of different letters in the text, then approximate the probability that a particular letter appears based on the histogram (assuming iid).
WHY SHANNON INFORMATION IS SOMEWHAT UNSATISFACTORY?

How much Shannon information does this picture contain?
The Kolmogorov complexity $K(x)$ of a sequence $x$ is the minimum size of the program and its inputs needed to generate $x$.

Example: If $x$ was a sequence of all ones, a highly compressible sequence, the program would simply be a print statement in a loop. On the other extreme, if $x$ were a random sequence with no structure, then the only program that could generate it would contain the sequence itself.
A discrete random variable $X$ takes a value $x$ from the alphabet $A$ with probability $p(x)$.
**Expected Value**

- If $g(x)$ is real valued and defined on $A$ then

\[
E_X[g(X)] = \sum_{x \in A} p(x)g(x)
\]
**SHANNON INFORMATION CONTENT**

- The Shannon Information Content of an outcome with probability $p$ is $-\log_2 p$
MINESWEEPER

- Where is the bomb?
- 16 possibilities - needs 4 bits to specify
**Entropy**

\[
H(X) = E[-\log_2(p(X))] = -\sum_{x \in A} p(x) \log_2(p(x))
\]
ENTROPY EXAMPLES

- Bernoulli Random Variable
**Entropy Examples**

- **Four Colored Shapes**
  
  \[ p(X) = \begin{bmatrix} \frac{1}{2} ; & \frac{1}{4} ; & \frac{1}{8} ; & \frac{1}{8} \end{bmatrix} \]

\[ A = [ \square ; \; \circ ; \; \triangle ; \; \star ] \]
DERIVATION OF SHANNON ENTROPY

\[ H(X) = \sum_{i=1}^{n} p_i \log_2 p_i \]

Intuitive Requirements:

1. We want \( H \) to be a continuous function of probabilities \( p_i \). That is, a small change in \( p_i \) should only cause a small change in \( H \)

2. If all events are equally likely, that is, \( p_i = 1/n \) for all \( i \), then \( H \) should be a monotonically increasing function of \( n \). The more possible outcomes there are, the more information should be contained in the occurrence of any particular outcome.

3. It does not matter how we divide the group of outcomes, the total of information should be the same.
DERIVATION OF SHANNON ENTROPY
DERIVATION OF SHANNON ENTROPY
DERIVATION OF SHANNON ENTROPY
DERIVATION OF SHANNON ENTROPY
LITTLE PUZZLE

Given twelve coins, eleven of which are identical, and one is either lighter or heavier than the rest. You have a scale that can determine whether what you put on the left pan is heavier, lighter, or the same as what you put on the right pan. What is the minimum number of weighing for you determine the odd coin, and whether it is heavier or lighter than the rest?
Joint Entropy $H(X, Y)$

$H(X, Y) = E[-\log p(X, Y)]$

<table>
<thead>
<tr>
<th>$p(X, Y)$</th>
<th>$Y = 0$</th>
<th>$Y = 1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X = 0$</td>
<td>1/2</td>
<td>1/4</td>
</tr>
<tr>
<td>$X = 1$</td>
<td>0</td>
<td>1/4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$p(X,Y)$</th>
<th>$Y = 0$</th>
<th>$Y = 1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X = 0$</td>
<td>1/4</td>
<td>1/4</td>
</tr>
<tr>
<td>$X = 1$</td>
<td>1/4</td>
<td>1/4</td>
</tr>
</tbody>
</table>
**Joint Entropy** $H(X,Y)$

Example: $Y = X^2$ \( X \in [-1, 1] \) \( p_X = [0.1, 0.9] \)

<table>
<thead>
<tr>
<th>$p(X,Y)$</th>
<th>$Y = 1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X = -1$</td>
<td>0.1</td>
</tr>
<tr>
<td>$X = 1$</td>
<td>0.9</td>
</tr>
</tbody>
</table>
**Conditional Entropy** $H(Y|X)$

$$H(Y|X) = E[-\log p(Y|X)]$$

| $p(Y|X)$ | $Y = 0$ | $Y = 1$ |
|----------|--------|--------|
| $X = 0$  | $2/3$  | $1/3$  |
| $X = 1$  | $0$    | $1$    |
**Conditional Entropy $H(Y|X)$**

Example: $Y = X^2$, $X \in [-1, 1]$, $p_X = [0.1, 0.9]$
INTERPRETATIONS OF CONDITIONAL ENTROPY $H(Y|X)$

- $H(Y|X)$: The average uncertainty/information in $Y$ when you know $X$

- $H(Y|X)$: The weighted average of row entropies

| $p(X,Y)$ | Y=0 | Y = 1 | $H(Y|X) = x$ | $p(X)$ |
|----------|-----|-------|--------------|--------|
| X=0      | 1/2 | 1/4   | $H(1/3)$    | 3/4    |
| X=1      | 0   | 1/4   | $H(1)$      | 1/4    |
Chain Rules

- For two RVs,

\[ H(X,Y) = H(X) + H(Y \mid X) = H(Y) + H(X \mid Y) \]

Proof:

- In general,

\[ H(X_1, X_2, \ldots, X_n) = \sum_{i=1}^{n} H(X_i \mid X_{i-1}, X_{i-2}, \ldots, X_1) \]

Proof:
**Graphical View of Entropy, Joint Entropy, and Conditional Entropy**

\[
H(X,Y) = H(X) + H(Y | X) = H(Y) + H(X | Y)
\]
The mutual information is the average amount of information that you get about $X$ from observing the value of $Y$

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

The mutual information is symmetrical

\[ I(X;Y) = I(Y;X) \]

Proof:
Mutual Information Example

- If you try to guess \( Y \), you have a 50% chance of being correct.
- However, what if you know \( X \)?
  - Best guess: choose \( Y = X \)
  - What is the overall probability of guessing correctly?

\[
\begin{array}{c|cc}
 p(X, Y) & Y = 0 & Y = 1 \\
\hline
 X = 0 & 1/2 & 1/4 \\
 X = 1 & 0 & 1/4 \\
\end{array}
\]

\[
\begin{align*}
H(X) &= 0.811 \\
H(Y) &= 1 \\
H(X|Y) &= 0.5 \\
H(Y|X) &= 0.689 \\
I(X;Y) &= 0.311
\end{align*}
\]


**CONDITIONAL MUTUAL INFORMATION**

\[
I(X;Y \mid Z) = H(X \mid Z) - H(X \mid Y,Z) = H(X \mid Z) + H(Y \mid Z) - H(X,Y \mid Z)
\]

Note: Z conditioning applies to both X and Y
**Chain Rule for Mutual Information**

\[
I(X_1, X_2, \ldots, X_n; Y) = \sum_{i=1}^{n} I(X_i; Y | X_{i-1}, X_{i-2}, \ldots, X_1)
\]

Proof:
Example of Using Chain Rule for Mutual Information

\[ X \xrightarrow{Z} + \xrightarrow{Y} \]

Find \( I(X,Z;Y) \).

<table>
<thead>
<tr>
<th>( p(X, Z) )</th>
<th>( Z = 0 )</th>
<th>( Z = 1 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( X = 0 )</td>
<td>1/4</td>
<td>1/4</td>
</tr>
<tr>
<td>( X = 1 )</td>
<td>1/4</td>
<td>1/4</td>
</tr>
</tbody>
</table>
CONCAVE AND CONVEX FUNCTIONS

$f(x)$ is strictly convex over $(a, b)$ if

$$f(\lambda u + (1 - \lambda)v) < \lambda f(u) + (1 - \lambda)f(v) \quad \forall u \neq v \in (a, b), \ 0 < \lambda < 1$$

Examples:

Technique to determine the convexity of a function:
JENSEN’S INEQUALITY

\[ f(X) \text{ convex } \quad \rightarrow \quad E[f(X)] \geq f(E[X]) \]

\[ f(X) \text{ strictly convex } \quad \rightarrow \quad E[f(X)] > f(E[X]) \]

Proof:
JENSEN’S INEQUALITY EXAMPLE

- Mnemonic example:

\[ f(x) = x^2 \]
RELATIVE ENTROPY

Relative entropy of Kullback-Leibler Divergence between two probability mass vectors (functions) \( p \) and \( q \) is defined as:

\[
D(p \| q) = \sum_{x \in A} p(x) \log \frac{p(x)}{q(x)} = E_p [\log \frac{p(x)}{q(x)}] = E_p [-\log q(x)] - H(X)
\]

Property of \( D(p \| q) \)
# Relative Entropy Example

<table>
<thead>
<tr>
<th></th>
<th>Rain</th>
<th>Cloudy</th>
<th>Sunny</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weather at Seattle</td>
<td>$p(x)$</td>
<td>1/4</td>
<td>1/2</td>
</tr>
<tr>
<td>Weather at Corvallis</td>
<td>$q(x)$</td>
<td>1/3</td>
<td>1/3</td>
</tr>
</tbody>
</table>

$$D(p \parallel q)$$

$$D(q \parallel p)$$
**Information Inequality**

\[ D(p \parallel q) \geq 0 \]

**Proof:**
INFORMATION INEQUALITY

- **Uniform distribution has highest entropy** \( H(X) \leq \log |A| \)
  
  **Proof:**

- **Mutual Information is non-negative** \( I(X;Y) \geq 0 \)
  
  **Proof:**
**Information Inequality**

- **Conditioning reduces entropy** \( H(X | Y) \leq H(X) \)
  
  **Proof:**

- **Independence bound** \( H(X_1, X_2, ..., X_n) \leq \sum_{i=1}^{n} H(X_i) \)
  
  **Proof:**
INFORMATION INEQUALITY

- Conditional independence bound

\[ H(X_1, X_2, ..., X_n \mid Y_1, Y_2, ..., Y_n) \leq \sum_{i=1}^{n} H(X_i \mid Y_i) \]

Proof:

- Mutual information independence bound

\[ I(X_1, X_2, ..., X_n; Y_1, Y_2, ..., Y_n) \geq \sum_{i=1}^{n} I(X_i; Y_i) \]

Proof:
Lecture 2: Symbol Code

- Symbol codes
  - Non-singular codes
  - Uniquely decodable codes
  - Prefix codes (instantaneous codes)
- Kraft Inequality
- Minimum Code Length
SYMBOL CODES

- Mapping of letters from one alphabet to another
  - ASCII codes: computers do not process English language directly. Instead, English letters are first mapped into bits, then processed by the computer.
  - Morse codes
  - Decimal to binary conversion

- Process of mapping from letters in one alphabet to letters in another is called coding.
Symbol Codes

- Symbol code is a simple mapping.
- Formally, symbol code $C$ is a mapping $X \rightarrow D^+$
  
  - $X =$ the source alphabet
  - $D =$ the destination alphabet
  - $D^+ =$ set of all finite strings from $D$

Example:

\[
\{X, Y, Z\} \rightarrow \{0,1\}^+, \quad C(X) = 0, \ C(Y) = 10, \ C(Z) = 1
\]

- Codeword: the result of a mapping, e.g., 0, 10, ...
- Code: a set of valid codewords.
**Symbol Codes**

- **Extension:** $C^+$ is a mapping $X^+ \rightarrow D^+$ formed by concatenating $C(x_i)$ without punctuation
  - Example: $C(XXYXZY) = 001001110$

- **Non-singular:** $x_1 \neq x_2 \rightarrow C(x_1) \neq C(x_2)$

- **Uniquely decodable:** $C^+$ is non-singular
  - That is $C^+(x^+)$ is unambiguous
Prefix Codes

- Instantaneous or Prefix code:
  - No codeword is a prefix of another
- Prefix → uniquely decodable → non-singular

Examples:

<table>
<thead>
<tr>
<th></th>
<th>C1</th>
<th>C2</th>
<th>C3</th>
<th>C4</th>
</tr>
</thead>
<tbody>
<tr>
<td>(W)</td>
<td>0</td>
<td>00</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>(X)</td>
<td>11</td>
<td>01</td>
<td>10</td>
<td>01</td>
</tr>
<tr>
<td>(Y)</td>
<td>00</td>
<td>10</td>
<td>110</td>
<td>011</td>
</tr>
<tr>
<td>(Z)</td>
<td>01</td>
<td>11</td>
<td>1110</td>
<td>0111</td>
</tr>
</tbody>
</table>
**Code Tree**

- Form a $D$-ary tree where $D = |D|$
  
  - $D$ branches at each node
  - Each branch denotes a symbol $d_i$ in $D$
  - Each leaf denotes a source symbol $x_i$ in $X$
  - $C(x_i)$ = concatenation of symbols $d_i$ along the path from the root to the leaf that represents $x_i$
  - Some leaves may be unused

$$C(WWYXZ) =$$
**Kraft Inequality For Prefix Codes**

- For any prefix code, we have:

\[ \sum_{i=1}^{X} D^{-l_i} \leq 1 \]

where \( l_i \) is the length of codeword \( i \) and \( D = |D| \).

**Proof:**
Kraft Inequality For Uniquely Decodable Codes

For any uniquely decodable code with codewords of lengths $l_1, l_2, \ldots, l_{|X|}$, then

$$\sum_{i=1}^{|X|} D^{-l_i} \leq 1$$

Proof:
Converse of Kraft Inequality

If \( \sum_{i=1}^{\left| X \right|} D^{-l_i} \leq 1 \), then there exists a prefix code with code lengths \( l_1, l_2, \ldots, l_{\left| X \right|} \).

Proof:
EXAMPLE OF CONVERSE OF KRAFT INEQUALITY
OPTIMAL CODE

- If \( l(C(x)) \) is the length of \( C(x) \), then \( C \) is optimal if \( L_C = E[l(C(X))] \) is as small as possible for a given \( p(X) \).
- \( C \) is a uniquely decodable code \( \rightarrow L_C \geq H(X)/\log_2 D \)

Proof:
SUMMARY

- Symbol code

- Kraft inequality for uniquely decodable code

- Lower bound for any uniquely decodable code
Lecture 3: Practical Symbol Codes

- Fano code
- Shannon code
- Huffman code
**FANO CODE**

- Put the probabilities in decreasing order
- Split as close to 50-50 as possible. Repeat with each half

<table>
<thead>
<tr>
<th></th>
<th>0.20</th>
<th>0.19</th>
<th>0.17</th>
<th>0.15</th>
<th>0.14</th>
<th>0.06</th>
<th>0.05</th>
<th>0.04</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>b</td>
<td>0.19</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>c</td>
<td>0.17</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>d</td>
<td>0.15</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>e</td>
<td>0.14</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>f</td>
<td>0.06</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>g</td>
<td>0.05</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>h</td>
<td>0.04</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- $H(p) = 2.81$ bits
- $L_{F} = 2.89$ bits
- Not optimal!
- $L_{opt} = 2.85$ bits
CONDITIONS FOR OPTIMAL PREFIX CODE (ASSUMING BINARY PREFIX CODE)

- An optimal prefix code must satisfy:
  - \( p(x_i) > p(x_j) \rightarrow l(x_i) \leq l(x_j) \) (else swap them)

- The two longest codewords must have the same length (else chop a bit off the codeword)

- In the tree corresponding to the optimum code, there must be two branches stemming from each intermediate node.
HUFFMAN CODE CONSTRUCTION

1. Take the two smallest $p(x_i)$ and assign each a different last bit. Then merge into a single symbol.

2. Repeat step 1 until only one symbol remains
HUFFMAN CODE EXAMPLE

\[X = [a, b, c, d, e], \ p_X = [0.25 \ 0.25 \ 0.2 \ 0.15 \ 0.15]\]

- a: 0.25 \[\rightarrow \ 0.25 \rightarrow \ 0.25 \rightarrow \ 0 \rightarrow \ 0.55 \rightarrow \ 0 \rightarrow \ 1.0\]
- b: 0.25 \[\rightarrow \ 0.25 \rightarrow \ 0 \rightarrow \ 0.45 \rightarrow \ 0.45 \rightarrow \ 1\]
- c: 0.2 \[\rightarrow \ 0.2 \rightarrow \ 1\]
- d: 0.15 \[\rightarrow \ 0 \rightarrow \ 0.3 \rightarrow \ 0.3 \rightarrow \ 1\]
- e: 0.15 \[\rightarrow \ 1\]
**Huffman Code Example (Cont)**

<table>
<thead>
<tr>
<th>Letter</th>
<th>Probability</th>
<th>Codeword</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1$</td>
<td>0.4</td>
<td></td>
</tr>
<tr>
<td>$x_2$</td>
<td>0.2</td>
<td></td>
</tr>
<tr>
<td>$x_3$</td>
<td>0.2</td>
<td></td>
</tr>
<tr>
<td>$x_4$</td>
<td>0.1</td>
<td></td>
</tr>
<tr>
<td>$x_5$</td>
<td>0.1</td>
<td></td>
</tr>
</tbody>
</table>
**Huffman Code is Optimal Prefix Code**

\[ p_2 = [0.55 \ 0.45], \quad c_2 = [0 \ 1], \quad L_2 = 1 \]

\[ p_3 = [0.25 \ 0.45 \ 0.3], \quad c_3 = [00 \ 1 \ 01], \quad L_3 = 1.55 \]

\[ p_4 = [0.25 \ 0.25 \ 0.2 \ 0.3], \quad c_4 = [00 \ 10 \ 11 \ 01], \quad L_4 = 2 \]

\[ p_5 = [0.25 \ 0.25 \ 0.2 \ 0.15 \ 0.15], \quad c_5 = [00 \ 10 \ 11 \ 010 \ 011], \quad L_5 = 2.3 \]

We want to show that all of these codes (include \( c_5 \)) is optimal
HUFFMAN OPTIMALITY PROOF
HOW SHORT ARE THE OPTIMAL CODE?

- If $l(x) = \text{length}(C(x))$ then $C$ is optimal if $L_C = E l(x)$ is as small as possible.

- We want to minimize

$$\sum_i p(x_i)l(x_i)$$

Subject to:

$$\sum_i D^{-l(x_i)} \leq 1$$

and

$l(x_i)$ are integers
**Optimal Codes (non-integer Length)**

- Minimize \( \sum_{i} p(x_i)l(x_i) \)

  subject to: \( \sum_{i} D^{-l(x_i)} \leq 1 \)

Use Lagrange multiplier method:
SHANNON CODE

- Round up optimal code lengths: \( l_i = \lceil -\log_D p(x_i) \rceil \)

- \( l_i \) satisfy the Kraft Inequality \( \rightarrow \) Prefix code exists. Construction for such code is done earlier.

- Average length:
  \[
  \frac{H(X)}{\log D} \leq L_s \leq \frac{H(X)}{\log D} + 1
  \]

Proof:
**SHANNON CODE EXAMPLES**

<table>
<thead>
<tr>
<th></th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
<th>$x_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p(x_i)$</td>
<td>1/2</td>
<td>1/4</td>
<td>1/8</td>
<td>1/8</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>$x_1$</th>
<th>$x_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p(x_i)$</td>
<td>0.99</td>
<td>0.01</td>
</tr>
</tbody>
</table>
Huffman Code vs. Shannon Code

<table>
<thead>
<tr>
<th></th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
<th>$x_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p(x_i)$</td>
<td>0.34</td>
<td>0.36</td>
<td>0.25</td>
<td>0.5</td>
</tr>
<tr>
<td>Shannon $[-\log p(x_i)]$</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>Huffman</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>

$H(X) = 1.78$ bits

$L_S = 2.15$ bits, $L_H = 1.94$ bits

```
\begin{array}{c|c|c|c|c}
\hline
   & x_1 & x_2 & x_3 & x_4 \\
\hline
p(x_i) & 0.34 & 0.36 & 0.25 & 0.5  \\
\hline
\hline
\text{Shannon} \left[-\log p(x_i)\right] & 2 & 2 & 2 & 5 \\
\hline
\text{Huffman} & 1 & 2 & 3 & 3 \\
\hline
\end{array}
```
**SHANNON COMPETITIVE OPTIMALITY**

- $l(x)$ is length of a uniquely decodable code, $l_S(x)$ is length of Shannon code then

$$p(l(X) < l_S(X) - c) \leq 2^{1-c}$$

Proof:
DYADIC COMPETITIVE OPTIMALITY

- If $p(x)$ is dyadic $\leftrightarrow \log(p(x_i))$ is integer for all $i$, then

$$p(l(X) < l_s(X)) \leq p(l_s(X) < l(X))$$

with equality iff $l(X) \equiv l_s(X)$

Proof:
SHANNON WITH WRONG DISTRIBUTION

- If the real distribution of $X$ is $p(x)$ but you assign Shannon lengths using the distribution $q(x)$ what is the penalty?

Answer: $D(p \parallel q)$

Proof:
SUMMARY
Lecture 4

- Stochastic Processes
- Entropy Rate
- Markov Processes
- Hidden Markov Processes
STOCHASTIC PROCESS

- Stochastic process \( \{X_i\} = X_1, X_2, X_3, \ldots \)
- Define entropy of \( \{X_i\} \) as
  \[
  H(\{X_i\}) = H(X_1, X_2, X_3, \ldots) = H(X_1) + H(X_2 | X_1) + H(X_3 | X_2, X_1) + \ldots.
  \]
- Define entropy rate of \( \{X_i\} \) as
  \[
  H(\bar{X}) = \lim_{n \to \infty} \frac{H(X_1, X_2, \ldots, X_n)}{n}
  \]
- Entropy rate estimates the additional entropy per new sample.
- Gives a lower bound on number of code bits per sample.
- If the \( X_i \) are not i.i.d. the entropy rate limit may not exist.
Examples
LIMIT OF CESARO MEAN

\[
\lim_{k \to \infty} a_k \to b \quad \Rightarrow \quad \lim_{n \to \infty} \frac{1}{n} \sum_{k=0}^{n} a_k \to b
\]

Proof:
A stochastic process \( \{X_i\} \) is stationary iff

\[ P(X_1 = a_1, X_2 = a_2, \ldots, X_n = a_n) = P(X_{k+1} = a_1, X_{k+2} = a_2, \ldots, X_{k+n} = a_n) \quad \forall k, n \]

If \( \{X_i\} \) is stationary, then \( H(\bar{X}) \) exists

Proof:
Block Coding is more efficient than single symbol coding
**Example of Block Code**

\[ X = [A; B], \ p_x = [0.9; 0.1] \]

\[ H(x_i) = 0.469 \]

- **n=1**
  
<table>
<thead>
<tr>
<th>sym</th>
<th>A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>prob</td>
<td>0.9</td>
<td>0.1</td>
</tr>
<tr>
<td>code</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
  
  \[ n^{-1}E \ l = 1 \]

- **n=2**
  
<table>
<thead>
<tr>
<th>sym</th>
<th>AA</th>
<th>AB</th>
<th>BA</th>
<th>BB</th>
</tr>
</thead>
<tbody>
<tr>
<td>prob</td>
<td>0.81</td>
<td>0.09</td>
<td>0.09</td>
<td>0.01</td>
</tr>
<tr>
<td>code</td>
<td>0</td>
<td>11</td>
<td>100</td>
<td>101</td>
</tr>
</tbody>
</table>
  
  \[ n^{-1}E \ l = 0.645 \]

- **n=3**
  
<table>
<thead>
<tr>
<th>sym</th>
<th>AAA</th>
<th>AAB</th>
<th>...</th>
<th>BBA</th>
<th>BBB</th>
</tr>
</thead>
<tbody>
<tr>
<td>prob</td>
<td>0.729</td>
<td>0.081</td>
<td>...</td>
<td>0.009</td>
<td>0.001</td>
</tr>
<tr>
<td>code</td>
<td>0</td>
<td>101</td>
<td>...</td>
<td>10010</td>
<td>10011</td>
</tr>
</tbody>
</table>
  
  \[ n^{-1}E \ l = 0.583 \]
Markov Process

- Discrete-valued Stochastic Process \{X_i\} is Markov iff
  \[ P(X_n \mid X_0, X_1, \ldots, X_{n-1}) = P(X_n \mid X_{n-1}) \]

- Time independent if
  \[ P(X_n = a \mid X_{n-1} = b) = P_{ab} \]
Finite state Markov process can be described using Markov chain diagram.
**Stationary Markov Process**

- As \( n \to +\infty \)

\[
P(n) \quad \text{approaches} \quad \begin{bmatrix} \varphi_1 & \varphi_2 & \cdots & \varphi_s \\ \varphi_1 & \varphi_2 & \cdots & \varphi_s \\ \varphi_1 & \varphi_2 & \cdots & \varphi_s \\ \vdots & \vdots & \ddots & \vdots \\ \varphi_1 & \varphi_2 & \cdots & \varphi_s \end{bmatrix}
\]

Each row is the stationary distribution

\[ (\varphi_1, \varphi_2, \ldots, \varphi_s) \]
STATIONARY MARKOV PROCESS EXAMPLE

\[ P = \begin{bmatrix} 0.6 & 0.1 & 0.2 & 0.1 \\ 0.1 & 0.7 & 0.1 & 0.1 \\ 0.2 & 0.2 & 0.5 & 0.1 \\ 0.1 & 0.3 & 0.1 & 0.5 \end{bmatrix} \]

\[ p_S = l^T = l^T P \]

\[ \begin{bmatrix} 0.241 & 0.385 & 0.207 & 0.167 \\ 0.241 & 0.385 & 0.207 & 0.167 \\ 0.241 & 0.385 & 0.207 & 0.167 \\ 0.241 & 0.385 & 0.207 & 0.167 \end{bmatrix} \]
CHESS BOARD

- $H(p_1)=0$, $H(p_1 | p_2)=0$
- $H(p_2)=1.58400$, $H(p_2 | p_1)=1.58400$
- $H(p_3)=3.10287$, $H(p_3 | p_2)=2.54795$
- $H(p_4)=2.99853$, $H(p_4 | p_3)=2.09299$
- $H(p_5)=3.111$, $H(p_5 | p_4)=2.36177$
- $H(p_6)=3.07129$, $H(p_6 | p_5)=2.20883$
- $H(p_7)=3.99141$, $H(p_7 | p_6)=2.24987$
- $H(p_8)=3.0827$, $H(p_8 | p_7)=2.23038$
EXAMPLE OF CODING FOR MARKOV SOURCE

Find $H(\tilde{X})$

a) Assuming the Markov model above with block code $n = 2$

b) Assuming i.i.d and symbol code

c) Assuming i.i.d and block code with $n = 2$.

d) Assuming i.i.d and block code with $n = 3$.

Which one of the above is minimum? Do you think when $n = 1000$, block code assuming i.i.d would be lower or higher than the answer in a)? Provide reason for your answer.
ALOHA WIRELESS EXAMPLE

- **M** users share wireless transmission channel
  - For each user independent in each time slot
    - If the queue is non-empty, transmit with the probability \( q \)
    - A new packet arrive for transmission with probability \( p \)
  - If two packets collide, they stay in the queue
  - At time \( t \), queue sizes are \( X_t = (n_1, \ldots, n_M) \)
    - \( \{X_t\} \) is Markov since \( P(X_t \mid X_{t-1}) \) depends only on \( X_{t-1} \)
  - Transmit vector \( Y_t \)
    - \( p(y_{i,t} = 1) = \begin{cases} 0 & x_{i,t} = 0 \\ q & x_{i,t} > 0 \end{cases} \)
  - \( \{Y_t\} \) is not a Markov since \( P(Y_t) \) depends on \( X_{t-1} \), not \( Y_{t-1} \).
  - \( \{Y_t\} \) is called *Hidden Markov Process*
**ALOHA Wireless Example**

<table>
<thead>
<tr>
<th>Waiting Packets</th>
<th>TX en</th>
</tr>
</thead>
<tbody>
<tr>
<td>6 5 4</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>x:</th>
<th>3 2 1 2 1 0 1 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 1 1 1 1 1 2 2 1</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TX enable, e:</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 0 0 1 1 1 1 1</td>
</tr>
<tr>
<td>1 1 0 0 1 1 0 0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>y:</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 0 0 1 1 0 1 1</td>
</tr>
<tr>
<td>0 1 0 0 1 1 0 0</td>
</tr>
</tbody>
</table>

\( y = (x > 0) e \) is a deterministic function of the Markov \([x; e]\)
If $X_i$ is a Markov Process, and $Y = f(X)$, then $Y_i$ is a stationary Hidden Markov Process.

What is entropy rate $H(\bar{Y})$?
HIDDEN MARKOV PROCESS
SUMMARY

- Entropy rate
  - Stationary
  - Stationary Markov
  - Hidden Markov
Lecture 5

- Stream codes
- Arithmetic code
- Lempel-Ziv code
Why not use Huffman code? (since it is optimal)

- **Good**
  - It is optimal, i.e., shortest possible code

- **Bad**
  - Bad for skewed probability
  - Employ block coding helps, i.e., use all $N$ symbols in a block. The redundancy is now $1/N$. However,
    - Must re-compute the entire table of block symbols if the probability of a single symbol changes.
    - For $N$ symbols, a table of $|X|^N$ must be pre-calculated to build the tree
    - Symbol decoding must wait until an entire block is received
ARITHMETIC CODE

- Developed by IBM (too bad, they didn’t patent it)
- Majority of current compression schemes use it

Good:
- No need for pre-calculated tables of big size
- Easy to adapt to change in probability
- Single symbol can be decoded immediately without waiting for the entire block of symbols to be received.
**Basic Idea in Arithmetic Coding**

- Sort the symbols in lexical order

- Represent each string $x$ of length $n$ by a unique interval $[F(x_{i-1}), F(x_i))$ in $[0,1)$. $F(x_i)$ is the cumulative distribution.

- $F(x_i) - F(x_{i-1})$ represents the probability of $x_i$ occurring.

- The interval $[F(x_{i-1}), F(x_i))$ can itself be represented by any number, called a tag, within the half open interval.

- The $l(x) = \left\lfloor \log \frac{1}{P(x)} \right\rfloor + 1$ significant bits of the tag $0.t_1t_2t_3...$ is the code of $x$. That is, $0.t_1t_2t_3...t_k000...$ is in the interval $[F(x_{i-1}), F(x_i))$
**Tag in arithmetic code**

- Choose the tag

\[
\bar{T}_X ([F(x_{i-1}), x_i]) = \sum_{k=1}^{i-1} P(X = x_k) + \frac{P(X = x_i)}{2} = F_X (x_{i-1}) + \frac{P(X = x_i)}{2} = \frac{F_X (x_{i-1}) + F_X (x)}{2}
\]
ARITHMETIC CODING EXAMPLE

- **a**: $\frac{1}{3}
- **b**: $\frac{2}{3}
- **bb**: lowercase
- **bba**: lowercase
- **15/27**: $0.100011100\ldots$
- **19/27**: $0.101101000\ldots$
- **tag = 17/27**: $0.101000010\ldots$
- **code = 1010**
**Arithmetic Coding Example**

- $P(a) = 1/3$, $P(b) = 2/3$.

<table>
<thead>
<tr>
<th>Character</th>
<th>Probability</th>
<th>Prefix Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>1/3</td>
<td>0000 aaa</td>
</tr>
<tr>
<td>ab</td>
<td>1/9</td>
<td>0001 aab</td>
</tr>
<tr>
<td>aba</td>
<td>2/27</td>
<td>001 aba</td>
</tr>
<tr>
<td>abb</td>
<td>1/9</td>
<td>0100 abb</td>
</tr>
<tr>
<td>bba</td>
<td>2/27</td>
<td>01011 baa</td>
</tr>
<tr>
<td>bbb</td>
<td>2/27</td>
<td>0111 bab</td>
</tr>
<tr>
<td>bab</td>
<td>1/9</td>
<td>101 bba</td>
</tr>
<tr>
<td>b</td>
<td>2/9</td>
<td>11 bbb</td>
</tr>
<tr>
<td>ba</td>
<td>1/9</td>
<td>11 bbb</td>
</tr>
<tr>
<td>ab</td>
<td>1/9</td>
<td>11 bbb</td>
</tr>
<tr>
<td>a</td>
<td>1/3</td>
<td>0000 aaa</td>
</tr>
</tbody>
</table>

Diagram:

- Nodes represent characters and their probabilities.
- Branches indicate the transition between characters.
- Prefix codes are associated with each node.
### Example of Arithmetic Coding

<table>
<thead>
<tr>
<th>Symbol</th>
<th>$F(x)$</th>
<th>$\overline{T_x(x)}$</th>
<th>In Binary</th>
<th>$\left\lceil \log \frac{1}{P(x)} \right\rceil + 1$</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>.5</td>
<td>.25</td>
<td>.010</td>
<td>2</td>
<td>01</td>
</tr>
<tr>
<td>2</td>
<td>.75</td>
<td>.625</td>
<td>.101</td>
<td>3</td>
<td>101</td>
</tr>
<tr>
<td>3</td>
<td>.875</td>
<td>.8125</td>
<td>.1101</td>
<td>4</td>
<td>1101</td>
</tr>
<tr>
<td>4</td>
<td>1.0</td>
<td>.9375</td>
<td>.1111</td>
<td>4</td>
<td>1111</td>
</tr>
</tbody>
</table>
## Arithmetic Code for Two-Symbol Sequences

<table>
<thead>
<tr>
<th>Message</th>
<th>$P(x)$</th>
<th>$\bar{T}_x(x)$</th>
<th>$\bar{T}_x(x)$ in Binary</th>
<th>$\left\lfloor \log \frac{1}{P(x)} \right\rfloor + 1$</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>.25</td>
<td>.125</td>
<td>.001</td>
<td>3</td>
<td>001</td>
</tr>
<tr>
<td>12</td>
<td>.125</td>
<td>.3125</td>
<td>.0101</td>
<td>4</td>
<td>0101</td>
</tr>
<tr>
<td>13</td>
<td>.0625</td>
<td>.40625</td>
<td>.01101</td>
<td>5</td>
<td>01101</td>
</tr>
<tr>
<td>14</td>
<td>.0625</td>
<td>.46875</td>
<td>.01111</td>
<td>5</td>
<td>01111</td>
</tr>
<tr>
<td>21</td>
<td>.125</td>
<td>.5625</td>
<td>.1001</td>
<td>4</td>
<td>1001</td>
</tr>
<tr>
<td>22</td>
<td>.0625</td>
<td>.65625</td>
<td>.10101</td>
<td>5</td>
<td>10101</td>
</tr>
<tr>
<td>23</td>
<td>.03125</td>
<td>.703125</td>
<td>.101101</td>
<td>6</td>
<td>101101</td>
</tr>
<tr>
<td>24</td>
<td>.03125</td>
<td>.734375</td>
<td>.101111</td>
<td>6</td>
<td>101111</td>
</tr>
<tr>
<td>31</td>
<td>.0625</td>
<td>.78125</td>
<td>.11001</td>
<td>5</td>
<td>11001</td>
</tr>
<tr>
<td>32</td>
<td>.03125</td>
<td>.828125</td>
<td>.110101</td>
<td>6</td>
<td>110101</td>
</tr>
<tr>
<td>33</td>
<td>.015625</td>
<td>.8515625</td>
<td>.1101101</td>
<td>7</td>
<td>1101101</td>
</tr>
<tr>
<td>34</td>
<td>.015625</td>
<td>.8671875</td>
<td>.1101111</td>
<td>7</td>
<td>1101111</td>
</tr>
<tr>
<td>41</td>
<td>.0625</td>
<td>.90625</td>
<td>.11101</td>
<td>5</td>
<td>11101</td>
</tr>
<tr>
<td>42</td>
<td>.03125</td>
<td>.953125</td>
<td>.111101</td>
<td>6</td>
<td>111101</td>
</tr>
<tr>
<td>43</td>
<td>.015625</td>
<td>.9765625</td>
<td>.1111011</td>
<td>7</td>
<td>111101</td>
</tr>
<tr>
<td>44</td>
<td>.015625</td>
<td>.984375</td>
<td>.1111111</td>
<td>7</td>
<td>111111</td>
</tr>
</tbody>
</table>
**Progressive Decoding**

Each additional bit received narrows down the possible interval.

$x = [a \ b], \ p = [0.6 \ 0.4]$
UNIQUENESS OF THE ARITHMETIC CODE

Idea of proving arithmetic is a uniquely decodable code

- Show that the truncation of the tag lies entirely within $[F(x_{i-1}, F(x_i))$, hence singular code
- Show that the truncation of the tag results in prefix code, hence uniquely decodable.

Proof:
UNIQUENESS OF THE ARITHMETIC CODE
Uniqueness of the arithmetic code
EFFICIENCY OF THE ARITHMETIC CODE

- Off at most by $2/N$

$$\frac{H(X_1, X_2, ..., X_N)}{N} \leq l_A < \frac{H(X_1, X_2, ..., X_N)}{N} + \frac{2}{N}$$

- Proof:
EFFICIENCY OF THE ARITHMETIC CODE
**ADAPTIVE ARITHMETIC CODE**

\[
p_n = \frac{1 + \text{count}(x_i = b)}{1 + n}
\]

\[p_1 = 0.5\]
\[p_2 = \frac{1}{3} \text{ or } \frac{2}{3}\]
\[p_3 = \frac{1}{4} \text{ or } \frac{1}{2} \text{ or } \frac{3}{4}\]
\[p_4 = \ldots\]

Coder and decoder only need to calculate the probabilities along the path that actually occurs.
**Dictionary Coding**

<table>
<thead>
<tr>
<th>index</th>
<th>pattern</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>a</td>
</tr>
<tr>
<td>2</td>
<td>b</td>
</tr>
<tr>
<td>3</td>
<td>ab</td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
<tr>
<td>n</td>
<td>abc</td>
</tr>
</tbody>
</table>

Both encoder and decoder are assumed to have the same dictionary (table).
**Ziv-Lempel Coding (ZL or LZ)**


- Adaptive dictionary technique.
  - Store previously coded symbols in a buffer.
  - Search for the current sequence of symbols to code.
  - If found, transmit buffer offset and length.
LZ77

Output triplet <offset, length, next>
Transmitted to decoder:  8 3 d 0 0 e 1 2 f

If the size of the search buffer is $N$ and the size of the alphabet is $M$ we need

$$\lfloor \log(N + 1) \rfloor + \lfloor \log(N + 1) \rfloor + \lfloor \log M \rfloor$$

bits to code a triplet.

**Variation:** Use a VLC to code the triplets!
**Drawbacks of LZ77**

- Repetetive patterns with a period longer than the search buffer size are not found.
- If the search buffer size is 4, the sequence
  \[a\ b\ c\ d\ e\ a\ b\ c\ d\ e\ a\ b\ c\ d\ e\ a\ b\ c\ d\ e\ \ldots\]
  will be expanded, not compressed.
SUMMARY

- Stream codes
  - Arithmetic
  - Lempel-Ziv
- Encoder and decoder operate sequentially
  no blocking of input symbols required
- Not forced to send ≥1 bit per input symbol
- Can achieve entropy rate even when $H(X)<1$
- Require a Perfect Channel
  - A single transmission error causes multiple wrong output symbols
  - Use finite length blocks to limit the damage
Lecture 6

- Markov chains
- Data Processing theorem
  - You can’t create information from nothing (no free lunch)
- Fano’s inequality
  - Lower bound for error in estimating $X$ from $Y$
Markov Chain

- Given 3 random variables $X,Y,Z$, they form a Markov chain denoted as $X \rightarrow Y \rightarrow Z$ if
  
  $$p(x, y, z) = p(x) p(y \mid x) p(z \mid y) \iff p(z \mid x, y) = p(z \mid y)$$

- A Markov chain $X \rightarrow Y \rightarrow Z$ means that
  - The only way that $X$ affects $Z$ is through the value of $Y$
  - $I(X;Z \mid Y) = 0 \iff H(Z \mid Y) = H(Z \mid X,Y)$
  - If you already know $Y$, then observing $X$ gives you no additional information about $Z$
  - If you know $Y$, then observing $Z$ gives you no additional information about $X$

- A common special case of a Markov chain is when $Z = f(Y)$
**Markov Chain Symmetry**

- $X \rightarrow Y \rightarrow Z \iff X$ and $Z$ are conditionally independent given $Y$
  
  \[ p(x, z \mid y) = p(x \mid y)p(z \mid y) \]

- $X \rightarrow Y \rightarrow Z \iff Z \rightarrow Y \rightarrow X$
DATA PROCESSING THEOREM

- If $X \rightarrow Y \rightarrow Z$ then $I(X ; Y) \geq I(X ; Z)$
  - Processing $Y$ cannot add new information about $X$
- If $X \rightarrow Y \rightarrow Z$ then $I(X ; Y) \geq I(X ; Y \mid Z)$
  - Knowing $Z$ can only decrease the amount $X$ tells you about $Y$

Proof:
Non-Markov: Conditioning increase mutual information

- Noisy channel  \( Z = X + Y \)
LONG MARKOV CHAIN

- $X_1 \to X_2 \to X_3 \to \ldots \to X_n$ then mutual information increases as you get closer and closer

$$I(X_1; X_2) \geq I(X_1; X_3) \geq I(X_1; X_4) \geq \ldots \geq I(X_1; X_n)$$
SUFFICIENT STATISTICS

- If pdf of $X$ depends on a parameter $\theta$ and you extract a statistic $T(X)$ from your observation, then $\theta \rightarrow X \rightarrow T(X) \Rightarrow I(\theta; T(X)) \leq I(\theta; X)$

- $T(x)$ is sufficient for $\theta$ if the stronger condition:

$$\theta \rightarrow X \rightarrow T(X) \rightarrow \theta \iff I(\theta; T(X)) = I(\theta; X)$$

$$\iff \theta \rightarrow T(X) \rightarrow X \rightarrow \theta$$

$$\iff p(X | T(X), \theta) = p(X | T(X))$$
EXAMPLES SUFFICIENT STATISTICS
**Fano’s Inequality**

- If we estimate $X$ from $Y$, what is $P(X \neq \hat{X})$?

  ![Diagram](image)

  $$P_e \equiv P(X \neq \hat{X}) \geq \frac{(H(X | Y) - H(P_e))}{\log(N - 1)} \geq \frac{(H(X | Y) - 1)}{\log(N - 1)}$$

  *$N$ is the size of the outcome set of $X$*

  **Proof:**
Fano’s Inequality Example

\[ X = \{1:5\}, \quad p_X = [0.35, 0.35, 0.1, 0.1, 0.1]^T \]

\[ Y = \{1:2\} \quad \text{if} \quad x \leq 2 \quad \text{then} \quad y = x \quad \text{with probability} \quad 6/7 \]

\[ \text{while if} \quad x > 2 \quad \text{then} \quad y = 1 \quad \text{or} \quad 2 \quad \text{with equal prob.} \]

Our best strategy is to guess \( \hat{x} = y \)

\[ p_{X|Y=1} = [0.6, 0.1, 0.1, 0.1, 0.1]^T \]

- actual error prob: \( p_e = 0.4 \)
SUMMARY

- Markov
- Data Processing Theorem
- Fano’s Inequality
Lecture 7

- Weak Law of Large Numbers
- The Typical Set
- Asymptotic Equipartition Principle
STRONG AND WEAK TYPICALITY

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>P(X)</td>
<td>0.5</td>
<td>0.25</td>
<td>0.25</td>
</tr>
</tbody>
</table>

- Strongly typical sequences: ABAACABCAABC
  - Correct proportion
  - $H(X) = -(0.5 \log(0.5) + 0.5 \log(0.25)) = 2.5$ bits

- Weak typical sequences: BBBBBBBBBBBBA
  - Incorrect proportion
CONVERGENCE OF RANDOM NUMBERS

- Almost Sure Convergence:
  \[ \Pr(\lim_{n \to \infty} X_n = X) = 1 \]

- Example:
Convergence of Random Numbers

- Convergence in Probability:

  \[ \lim_{n \to \infty} (\Pr(|X_n - X| \geq \varepsilon)) \to 0, \ \forall \varepsilon > 0 \]

- Example:
Weak Law of Large Numbers

Given i.i.d \{X_i\}, let
\[
n s_n = \frac{1}{n} \sum_{i=1}^{n} X_i
\]
then

\[
E[s_n] = E[X] = \mu, \quad Var(s_n) = \frac{1}{n} Var(X) = \frac{1}{n} \sigma^2
\]

As \(n\) increases, the variance of \(s_n\) decreases, i.e., values of \(s_n\) become clustered around the mean.

WLLN

\[
s_n \xrightarrow{\text{prob}} \mu, \quad P(\left| s_n - \mu \right| \geq \varepsilon) \xrightarrow{n \to \infty} 0
\]
Proof of Weak Law of Large Numbers

- Chebyshev’s Inequality

- WLLN
**Typical Set**

- $x^n$ is the i.i.d sequence of $\{X_i\}$ for $i = 1$ to $n$

\[ T^n_\varepsilon = \{ x^n \in X^n : \left| -n^{-1} \log p(x^n) - H(X) \right| < \varepsilon \} \]
Example of Typical Set

- Bernoulli with $p$
TYPICAL SETS
TYPICAL SET PROPERTIES
ASYMPTOTIC EQUIPARTITION PRINCIPLE

For any \( \varepsilon \), and \( n > N_\varepsilon \),
almost any event is almost equally surprising
SOURCE CODING AND DATA COMPRESSION
What $N_{\varepsilon}$, to ensure that $P(x^n \in T^n_{\varepsilon}) > 1 - \varepsilon$
Smallest High Probability Set
SUMMARY

•
Lecture 8

- Source and Channel Coding
- Discrete Memoryless Channels
  - Binary Symmetric Channel
  - Binary Erasure Channel
  - Asymmetric Channel
- Channel Capacity
SOURCE AND CHANNEL CODING

- Source Coding
- Channel Coding
**Discrete Memoryless Channel**

- Discrete input, discrete output
- Channel matrix \( Q \)
- Memoryless
**Binary Channels**

- Binary symmetric channel
- Binary erasure channel
- Z channel
Weakly Symmetric Channels

1. All columns of $Q$ have the same sum

2. Rows of $Q$ are permutation of each other
SYMMETRIC CHANNEL

1. Rows of $Q$ are permutations of each other

2. Columns of $Q$ are permutations of each other
CHANNEL CAPACITY

- Capacity of discrete memoryless channel

\[ C = \max_{p(x)} I(X;Y) \]

- Capacity of \( n \) uses of channel:

\[ C^n = \frac{1}{n} \max_{p(x)} I(X_1, X_2, X_3, \ldots, X_n; Y_1, Y_2, Y_3, \ldots, Y_n) \]
Mutual Information

Binary Symmetric Channel
Bernoulli Input

\[ I(X;Y) \]

Input Bernoulli Prob (p)
Channel Error Prob (f)
**Mutual Information is concave in** $p(x)$

- Mutual information is concaved in $p(x)$ for a fixed $p(y|x)$
- Proof:
Mutual Information is convex in $p(y \mid x)$

- Mutual information is convex in $p(y \mid x)$ given a fixed $p(x)$
- Proof:
USE CHANNEL CAPACITY
CAPACITY OF WEAKLY SYMMETRIC CHANNEL
Binary Erasure Channel
ASYMMETRIC CHANNEL CAPACITY
Lecture 9

- Jointly Typical Sets
- Joint AEP
- Channel Coding Theorem
SIGNIFICANCE OF MUTUAL INFORMATION
**Jointly Typical Sets**

\[ J^n_\varepsilon = \{(x^n, y^n) \in X Y^n : \left| -n^{-1} \log p(x^n) - H(X) \right| < \varepsilon, \]
\[ \left| -n^{-1} \log p(x^n) - H(X) \right| < \varepsilon, \]
\[ \left| -n^{-1} \log p((x^n, y^n)) - H(X, Y) \right| < \varepsilon \} \]
**Jointly Typical Example**

<table>
<thead>
<tr>
<th>$p(x,y)$</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.5</td>
<td>0.125</td>
</tr>
<tr>
<td>1</td>
<td>0.125</td>
<td>0.25</td>
</tr>
</tbody>
</table>
JOINTLY TYPICAL DIAGRAM
JOINTLY TYPICAL SET PROPERTIES
Joint AEP

If $p(x) = p(x')$, $p(y) = p(y')$, and $X', Y'$ are independent, then

$(1 - \varepsilon)2^{-n(I(X,Y)+3\varepsilon)} \leq p((x^n, y^n) \in J^n_{\varepsilon}) \leq 2^{-n(I(X,Y)-3\varepsilon)}$
CHANNEL CODE

\( w \in 1, 2, \ldots, M \)

Encoder \( x^n \) Noisy channel \( y^n \) Decoder \( \hat{w} = g(y^n) \)

\( \hat{w} \in 1, 2, \ldots, M \)
CHANNEL CODING THEOREM
Channel Coding Theorem

![Graph showing the relationship between bit error probability and rate R/C. The graph indicates that there is an achievable region for certain rates and a region where it is impossible to achieve certain error probabilities.]
Lecture 10

- Channel Coding Theorem
CHANNEL CODING: MAIN IDEA

\[ x^n \rightarrow \text{Noisy channel} \rightarrow y^n \]
RANDOM CODE
Random Coding Idea

$X^n \xrightarrow{\text{Noisy channel}} Y^n$

send $x_1$

$e_1 \cup e_3$ true – unlikely

$e_1$ true – very likely

$e_2$ true – unlikely
ERROR PROBABILITY OF RANDOM CODE
CODE SELECTION AND EXPURGATION
PROCEDURE SUMMARY
CONVERSE OF CHANNEL CODING THEOREM
LECTURE 11

- Minimum Bit Error Rate
- Channel with Feedback
- Joint Source Channel Coding
Minimum Bit Error Rate
CHANNEL WITH FEEDBACK
CAPACITY WITH FEEDBACK
Example: Binary Erasure Channel with Feedback
Joint Source Channel Coding
SOURCE CHANNEL PROOF (→)
SOURCE CHANNEL PROOF (←)
SEPARATION THEOREM