2/5/14 Class 7

What is a signal be? Can it be compressed?

9.2 Uncertainty, Information, and Entropy

Finite alphabet $\mathcal{S} = \{S_1, S_2, \ldots, S_k\}$

with probability $P(S = S_k) = P_k$, $\sum P_k = 1$

$H(S_k) = -\sum P_k \log_2 P_k$ (bit)

1. $I(S_k) = 0$, for $P_k = 1$
2. $I(S_k) > 0$, for $0 < P_k < 1$
3. $I(S_k) > I(S_i)$, $P_k < P_i$
4. $I(S_k S_i) = I(S_k) + I(S_i)$ if $S_k, S_i$, statistically independent

Entropy $H(\psi) = E[I(S_k)] = \sum P_k I(S_k) = \sum \frac{1}{P_k} \log_2 \frac{1}{P_k}$

average information content per source symbol

$k = 2$

properties of Entropy $0 \leq H(\psi) \leq \log_2 k$

Ex. 9.1: Entry of binary memoryless source $p_0$ $H(p_0) = p_0 \log_2 p_0 + (1-p_0) \log_2 (1-p_0)$

Extension of a discrete memoryless source in one symbol: $H(\psi^2) = n H(\psi)$

Ex. 9.2: Entropy of extended source

$H(\psi^2) = \sum_{S_i} P(S_i) \frac{1}{P(S_i)} \log_2 \frac{1}{P(S_i)}$

$= \frac{1}{16} \log_2 16 + \frac{1}{16} \log_2 16 = 3$ bits

Table 9.1
9.3 Source Coding Theorem

Requirements for code words are binary form
1. Unique and decodable, original source sequence can be reconstructed perfectly

Example: Morse code, from encoded binary sequence,

Average code length \( L = \frac{2}{0.4} \text{ bits} \)

Theorem: given a discrete memoryless source of entropy \( H(X) \)
the average code-word length \( L \) for any distortionless source encoding scheme is bounded by

\( L \geq H(X) \)

9.4 Data Compression: Question 1

How ZIP works

Prefix coding, Huffman coding, Lempel-Ziv coding

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Probability</th>
<th>Code I</th>
<th>Code II</th>
<th>Code III</th>
</tr>
</thead>
<tbody>
<tr>
<td>S0</td>
<td>0.5</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>S1</td>
<td>0.25</td>
<td>1</td>
<td>10</td>
<td>01</td>
</tr>
<tr>
<td>S2</td>
<td>0.125</td>
<td>00</td>
<td>110</td>
<td>011</td>
</tr>
<tr>
<td>S3</td>
<td>0.125</td>
<td>11</td>
<td>111</td>
<td>0111</td>
</tr>
</tbody>
</table>

Prefix code

Initial state: \( S_0 \)

Example: \( 101111000 \)

\( S_0 \rightarrow S_1 \rightarrow S_2 \rightarrow S_5 \)

Kraft-McMillian Inequality

\( \sum 2^{-L_k} \leq 1 \)
Given discrete memoryless source with entropy \( H(\psi) \)

a prefix code can be constructed with length \( \leq \frac{1}{n} H(\psi) + 1 \)

for extended code

\[ n H(\psi) = H(\psi^n) \leq \frac{1}{n} \sum_{i=1}^{n} H(\psi_i) + 1 = n H(\psi) + 1 \]

\[ \lim_{n \to \infty} \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{n} H(\psi_{ij}) = H(\psi) \]

Asymptotically achieving entropy.

Huffman coding: a special class of prefix codes.

Huffman encoding algorithm:

1. Source symbols are listed in order of decreasing probability. Two source symbols of lowest probability are assigned a 0 and 1. Splitting stage.

2. These two source symbols are regarded as being combined into a new source symbol with probability equal to the sum of the two original probabilities (the list of source symbols, and therefore source statistics, is thereby reduced in size by one).

The probability of the new symbol is placed in the list in accordance with its value.

3. The procedure is repeated until we are left with a final list of source statistics (symbols) of only two for which a 0 and 1 are assigned.
Example 9.3 Huffman Tree

Symbol | stage I | stage II | stage III | stage IV
--- | --- | --- | --- | ---
S0 | 0.4 | 0.4 | 0.4 | 0.6
S1 | 0.2 | 0.2 | 0.2 |
S2 | 0.2 | | |
S3 | | 0.1 | 0.1 |
S4 | | | 0.1 |

Symbol | probability | code word
--- | --- | ---
S0 | 0.4 | 00
S1 | 0.2 | 10
S2 | 0.2 | 11
S3 | 0.1 | 010
S4 | 0.1 | 011

\[ L = 0.4 \times 2 + 0.2 \times 2 + 0.2 \times 2 + 0.1 \times 3 = 2.2 \]

Drawback: need probability
Example: Morse code

Lempel-Ziv coding: adaptive and simpler

Parsing the source data stream into segments that are the shortest subsequences not encountered previously

Initial: Subsequence stored 0, 1

Data to be parsed: 00101110010100101

Shortest subsequence not stored

Step 1: Subsequence stored 0, 1, 0, 0

Data to be parsed: 01011100100100101

Step 2: Subsequence stored 0, 1, 0, 0, 0

Data to be parsed: 011100100101
Sequences: 0 1 00 01 011 10 010 100 101 101 010 100

Binary block: 001000111010010010110001001001100

Decoding: 1101, position 9, last bit 1, remaining 110 -> sequence 6 (10) so decoding is 101.

Fixed length codes to represent a variable number of source symbols.

\( n \) is the number of phrases in L-Z parsing. 
\[ \log(n) \] as the number of bits.

The total length of the compressed sequence:
\[ C(n) = \frac{n}{(k+1) \log n} \] as \( k \to 0, n \to \infty \).