Part II
Information Theory Concepts
Chapter 2 Source Models and Entropy

- Any information-generating process can be viewed as a source:
  - emitting a sequence of symbols
  - symbols from a finite alphabet
    - text: ASCII symbols
    - computer program in executed form: binary 0 and 1
    - n-bit image: 2^n symbols

Discrete Memoryless Sources (DMS)

- successive symbols statistically independent
- $S = \{s_1, s_2, \ldots, s_n\}$
- $\{p(s_1), p(s_2), \ldots, p(s_n)\}$
- $I(s_i)$, the information revealed by the occurrence of a certain source symbol, is defined as

$$I(s_i) = \log_2 \frac{1}{p(s_i)}$$

- Average Information per source symbol, entropy $H(s) = \Sigma\, p(s_i) I(s_i) = -\Sigma\, p(s_i) \log_2 p(s_i)$ bits/symbol

Extensions of a Discrete Memoryless Source

- DMS S with an alphabet of size n
- the output of the source grouped into blocks of N symbols
- S^N with an alphabet of size n^N: the Nth extension of the source S
- For a memoryless source, the probability of a symbol $\sigma_i = (s_{i_1}, s_{i_2}, \ldots, s_{i_N})$ from S^N is given by

$$p(\sigma_i) = p(s_{i_1})p(s_{i_2}) \ldots p(s_{i_N})$$

$$H(S^N) = NH(S)$$

Markov Sources

- DMS too restrictive
- In general, the previous part of a message influences the probabilities for the next symbol, the source has memory.
- In English text, the letter Q is almost always followed by the letter U.
- In digital images, the probability of a given pixel taking on a particular code value is dependent on the surrounding pixel values.

- Such a source can be modeled as a Markov source.
- An mth-order Markov source:

$$p(s_i|s_{j_1}, \ldots, s_{j_m})$$

$$\begin{array}{l} s_{j_1}, \ldots, s_{j_m} \text{ preceding to } s_i \\ i, j_k \quad (k = 1, 2, \ldots, m) = 1, 2, \ldots, n \end{array}$$

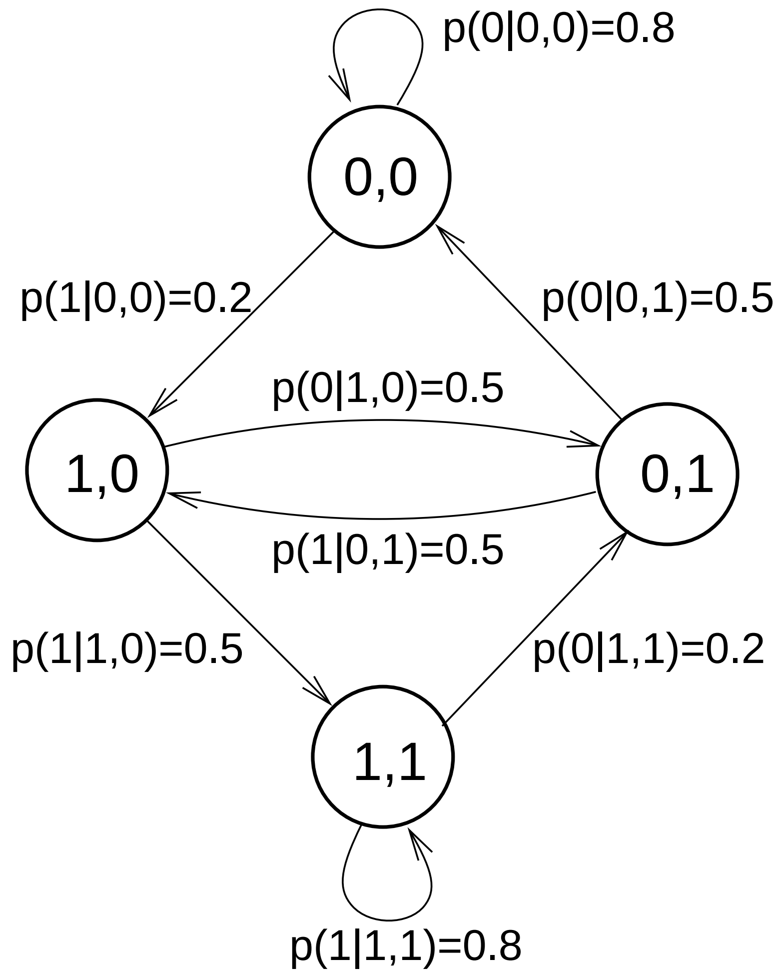
  $(s_{j_1}, \ldots, s_{j_m})$: a state for the mth-order Markov source, a total of n^m states

- For an ergodic Markov source, $\exists$ a unique probability distribution over the set of states: stationary or equilibrium distribution.

- $$H(S|s_{j_1}, \ldots, s_{j_m}) = -\Sigma_i \, p(s_i|s_{j_1}, \ldots, s_{j_m}) \cdot$$

$$\log p(s_i|s_{j_1}, \ldots, s_{j_m})$$

$$\begin{aligned} H(S) &= \Sigma_{S^m} H(S|s_{j_1}, \ldots, s_{j_m}) \cdot \\ &\quad p(s_{j_1}, \ldots, s_{j_m}) \\ &= -\Sigma_{S^{m+1}} \, p(s_{j_1}, \ldots, s_{j_m}) p(s_i|s_{j_1}, \ldots, s_{j_m}) \cdot \\ &\quad \log p(s_i|s_{j_1}, \ldots, s_{j_m}) \\ &= -\Sigma_{S^{m+1}} \, p(s_{j_1}, \ldots, s_{j_m}, s_i) \cdot \\ &\quad \log p(s_i|s_{j_1}, \ldots, s_{j_m}) \end{aligned}$$



- $p(0,0) = p(1,1) = \frac{5}{14}$, $p(0,1) = p(1,0) = \frac{2}{14}$

$H(S) = 0.801$ bit/symbol

Extensions of a Markov Source and Adjoint Sources

- The Nth extension of a Markov source, S^N, is a μth-order Markov source with symbols defined as blocks of N symbols from the original source, where $\mu = \lceil m/N \rceil$
- As in the case of a DMS,

$$H(S^N) = NH(S)$$

- The Nth extension of a Markov source, S^N, with source symbols $\{\sigma_1, \sigma_2, \ldots, \sigma_{n^N}\}$ and stationary probabilities $\{p(\sigma_1), p(\sigma_2), \cdots, p(\sigma_{n^N})\}$: a DMS with the same alphabet and the same symbol probabilities is called the adjoint source of S^N and denoted by $\bar{S^N}$.
  - The adjoint source ignores the conditional probabilities which describe the dependence between the extended symbols.
  - $H(\bar{S}^N) \geq H(S^N)$
  - $H_N(S) = \frac{H(\bar{S}^N)}{N} \rightarrow H(S)$

- The Noiseless Source Coding Theorem
  - S an ergodic source with an alphabet of size n and an entropy $H(S)$
  - encoding blocks of N source symbols at a time into binary codewords
  - For any $\delta > 0$, it is possible, by choosing N large enough, to construct a code so that the average number of bits per original source symbol, $\bar{L}$, satisfies

$$H(S) \leq \bar{L} \leq H(S) + \delta$$

Chapter 3 Variable-Length Codes

- Variable-length codes with source extensions to achieve the entropy of a source
  - a DMS $S = \{s_1, s_2, s_3, s_4;$ $p(s_1) = 0.60,\ p(s_2) = 0.30,\ p(s_3) = 0.05,\ p(s_4) = 0.05\}$
  - each codeword in the sequence is instantaneously decodeable without reference to the succeeding codewords iff no codeword be a prefix of some other codeword (called by a prefix condition code)

- entropy $H(S) = \Sigma_{i=1}^{n} p(s_i) I(s_i)$
  with $I(s_i) = -\log_2 p(s_i)$
- average codeword length or average length of the code, $\bar{L} = \Sigma_{i=1}^{n} p(s_i) L(s_i)$ with $L(s_i)$ being the length of the codeword for s_i
- To have $\bar{L} \approx H(S)$, we need $L(s_i) \approx -\log_2 p(s_i) = \log_2 \frac{1}{p(s_i)}$ bits
  or $L(s_i) = \lceil \log_2 \frac{1}{p(s_i)} \rceil$ (bits)
- Shannon-Fano coding

Symbol	Probability	Code I	Code II
s_1	0.60	00	0
s_2	0.30	01	10
s_3	0.05	10	110
s_4	0.05	11	111

$$H(S) = 1.40 \text{ bits/symbol}$$

$$\bar{L}_1 = 2.0 \text{ bits/symbol}$$

$$\bar{L}_2 = 1.5 \text{ bits/symbol}$$

- A code is compact (for a given source) if it has the smallest possible average codeword length.

- Code Efficiency and Source Extensions
  - Code II compact on S, its average codeword length is still far greater than $H(S)$
  - the code efficiency:

$$\eta = \frac{H(S)}{\bar{L}}$$

    * Code II: $\eta = \frac{1.4}{1.5} = 0.93$
  - Extension to S^2 of 16 symbols formed as pairs of symbols from S.
    * Table 3.2 shows a compact code of S^2, $\bar{L} = 2.86$ bits/extended symbol

$= 1.43$ bits/original source symbol

$\eta = \frac{1.40}{1.43} = 0.98$

- Huffman Codes for constructing compact codes
  - The Huffman code for a source $\{s_1, s_2\}$ has trivial codewords "0" and "1".
  - Consider $S = \{s_1, s_2, \ldots, s_n\}$ $(n > 2)$
    Let s_{n-1}, s_n be least probable symbols of this source.

Let Huffman code for

$$\{s_1 s_2, \cdots, s_{n-2}, \{s_{n-1}, s_n\}\}$$

be constructed and the codeword for $\{s_{n-1}, s_n\}$ be w. Then Huffman code for $\{s_1, \cdots, s_{n-1}, s_n\}$ will be Huffman code for $s_1, \cdots, s_{n-2}$ and $w\underline{\underline{0}}$ for s_{n-1}, $w\underline{\underline{1}}$ for s_n.

understanding Fig. 3

- Modified Huffman Codes
  - Frequently, most of symbols in a large symbol set have very small probabilities.
  - Lump the less probable symbols into a symbol called "Else" and design a Huffman code for the reduced symbol set: the modified Huffman code.
  - Whenever a symbol in the ELSE category needs to be encoded, the encoder transmits the codeword for ELSE followed

by extra bits needed to identify the actual message within the ELSE category.

* the loss in coding efficiency very small
* the storage requirements and the decoding complexity substantially reduced

- Group 3 international digital facsimile coding standards:
  - each binary image scan line:
    a sequence of alternating black and white runs which are encoded with separable variable-length code tables
  - A run is the number of times a particular value occurs consecutively along a scanline.

- 1728 pixels for each scanline
- each Huffman table should have 1728 entries
- greatly simplified by taking advantage of the fact that the longer runs are highly improbable
- The first 64 entries in each table represent the Huffman code for runs 0 to 63
- All other runs $64N + M$ $(1 \leq N \leq 27, 0 \leq M \leq 64)$:
  entries for 64 to 90 encode N
  entries for 0 to 63 encode M

- a run of 213: $N = 3$ and $M = 21$ its Huffman code
  the entry 67(64 + 3) for $N = 3$
  the entry 21 for $M = 21$
- simplifying the search for decoding

- Limitations of Huffman Coding
  - The ideal binary codeword length for a source symbol s_i from a DMS is $-\log_2 p(s_i)$, this condition is met only if $p(s_i) = \frac{1}{2^k}$.

– Otherwise, direct encoding of the individual source symbols may result in poor code efficiency.

* $p(s_1) \approx 1$, $p(s_2) = \ldots = p(s_n) \approx 0$

$$\begin{aligned} H(S) &= -p(s_1)\log_2 p(s_1) - \Sigma_{k\geq 2}\, p(s_k) \\ &\quad \log_2 p(s_k) \\ &\approx -(n-1)p(s_2)\log_2 p(s_2) \\ &= -(n-1)\tfrac{(1-p(s_1))}{(n-1)}\log_2 \tfrac{(1-p(s_1))}{(n-1)} \\ &= -(1-p(s_1))\log_2 \tfrac{(1-p(s_1))}{(n-1)} \\ &\longrightarrow 0 \quad \text{as } p(s_1) \to 1 \end{aligned}$$

* $\bar{L} \geq 1$ since the shortest codeword length for each individual symbol is one

- $S = \{0, 1\}$
  The Huffman codewords for "0" and "1" are "0" and "1", thus $\bar{L} = 1$, regardless of the symbol probabilities.
- Encoding an extended source may improve the coding efficiency, but convergence to the source entropy could be slow.
- The number of entries in the Huffman code table grows exponentially with the block size.

– For an mth order Markov source, the conditional probabilities $p(s_i|s_{i_1}, \ldots, s_{i_m})$ vary as the state $(s_{i_1}, \ldots, s_{i_m})$ changes. Thus, a separable Huffman table is needed for each state.

– The coding efficiency may still be low if the symbol conditional probabilities deviate from the ideal case.

– Using an extended source and encoding the adjoint, its entropy H_N may get close to the entropy H of the Markov source but the block size must be large.

– The Huffman coding cannot efficiently adapt to changing source statistics
– Arithmetic coding is more complex than Huffman coding, but it can overcome the limitations of Huffman coding