Review of source coding methods

Source coding

Quantization (a lossy data compression)

Analogue source is quantized into finite number of levels.

Introduces distortion - Data is lost and cannot be recovered.

Examples

- Pulse code modulation PCM
- Differential pulse code modulation DPCM
- Delta modulation DM
- Uniform quantization
- …

The fundamental limit on the performance is given by the rate-distortion bound.

Noiseless coding (lossless data compression)

Digital data (the results of quantization) are compressed. Compression attempts to represent data with as few bits as possible.

Examples

- Huffmann coding
- Lempel-Ziv coding
- Arithmetic coding

Waveform sources

- Voice coding
  - Waveform coders
    - PCM, ADPCM, DM
  - Vocoders
- Music coding
  - mp3, ogg
- Video coding
  - mpeg2, mpeg4
- Still picture coding
  - jpeg

Discrete sources

- Digital data compression
  - Constant length code words
  - Variable length code words
    - -Huffman encoding

Source coding theorem

- Foundation of source coding laid by Shannon
- Tells the minimum source encoded word length from a discrete information source
- A continuous information source must first be sampled, the sampling frequency should be equal to or larger than two times the signal bandwidth
- This usually means that the samples are correlated, the information contains redundancy
- Transmitting the redundancy part of the information means wasting of primary communication resources, transmit power and channel bandwidth
- The task of source encoding is to remove the redundancy, if necessary it can be added in the receiver

Self entropy

$$H(X) = -\sum_{i=1}^{n} P(x_i) \log P(x_i)$$

Entropy of a discrete source is maximum when the output symbols are equally probable

Conditional entropy

$$H(X \mid Y) = -\sum_{j=1}^{m} \sum_{i=1}^{n} P(x_i, y_j) \log P(x_i \mid y_j)$$

In a communication channel the conditional entropy is called also equivocation

Self information, conditional self information, mutual self information

$$l(x_i) = -\log P(x_i)$$

$$l(x_i \mid y_j) = -\log P(x_i \mid y_j)$$

$$l(x_i, y_j) = -\log P(x_i, y_j)$$

Mutual information

$$I(X,Y) = H(X) - H(X \mid Y) = H(Y) - H(Y \mid X)$$

Differential entropy

$$h(X) = -\int_{-\infty}^{\infty} p(x_i) \log p(x_i)\, dx$$

Entropy of a normally distributed variable

$$h\left(N\left(0, \sigma^2\right)\right) = \frac{1}{2} \log\left(2\pi e \sigma^2\right)$$

Example of the source encoding theorem

- For a discrete memoryless information source (subsequent symbols are statistically independent) the average source encoded code word length must fulfil the condition

$$\mathrm{E}\{L_{cwd}\} \geq H_s \qquad (3-1)$$

L_{cwd} is the length of a source coded code-word
H_s is the entropy of the source defined as

$$H_s = \sum_{k=1}^{K} p_k \log_2\left(\frac{1}{p_k}\right) \qquad (3-2)$$

p_k is the occurrence probability of each of the K source symbols

For an information source with memory (subsequent symbols are not statistically independent) the source entropy is defined as

$$H_S = \lim_{k\to\infty}\left(\frac{1}{k}\sum_{i=1}^{k} H_S\left(s_i \,|\, s_1, s_2, \ldots s_{i-1}\right)\right) \qquad (3-3)$$

$$H_S\left(s_i \,|\, s_1, s_2, \ldots s_{i-1}\right) = \sum_{s_1}\ldots\sum_{s_i} P\left\{s_1, \ldots, s_i\right\} \log_2\left(\frac{1}{P\left\{s_i \,|\, s_1, \ldots, s_{i-1}\right\}}\right) \qquad (3-4)$$

For a continuous signal source the average entropy is infinite because an infinite codeword length is needed to describe a signal amplitude sample. Instead a differential entropy is defined as

$$H = \int p(x)\log_2\left(\frac{1}{p(x)}\right)dx \qquad (3-5)$$

The efficiency of the source encoder is defined as

$$\eta = \frac{H_S}{\mathrm{E}\left\{L_{cwd}\right\}} \qquad (3-6)$$

Example 3.1

A discrete source generates the symbols 0,1,2,3,4,5,6,7,8,and 9 with equal probability. The source entropy is

$$H_s = \sum_{k=1}^{K} p_k \log_2\left(\frac{1}{p_k}\right) = 10 \cdot 0.1 \cdot \log_2\left(\frac{1}{0.1}\right) = 3.32 \text{ bit/symbol}$$

If 4 bits/symbol are used in the transmission the source encoder efficiency is

$$\eta = \frac{H_s}{\mathrm{E}\{L_{cwd}\}} = \frac{3.32}{4} = 0.83$$

Compression

Fundamental limit of the compression is given by the entropy of the source

If all the L source symbols are equally probable the average code length is

$$H(X) = -\sum_{i=1}^{n} P(x_i)\log_2 P(x_i) \le \log_2 L$$

The average code rate is

$R = \log_2 L$ if L is power of 2 or

$R = \lfloor \log_2 L \rfloor + 1$

We can improve (reduce) the code length by combining multiple symbols

$$\lfloor J \log_2 L \rfloor + 1 \Rightarrow R = \lfloor \log_2 L \rfloor + \frac{1}{J}$$

In case the symbols are not equally probable more likely symbols can be allocated shorter code words

Prefix code a code where end of each codeword can explicitly identified – the code is uniquely decodable and uniquely decodable

Letter	P(a_i)	Code I	Code II	Code III
a_1	1/2	1	0	0
a_2	1/4	00	10	01
a_3	1/8	01	110	011
a_4	1/8	10	111	111

Huffmann coding

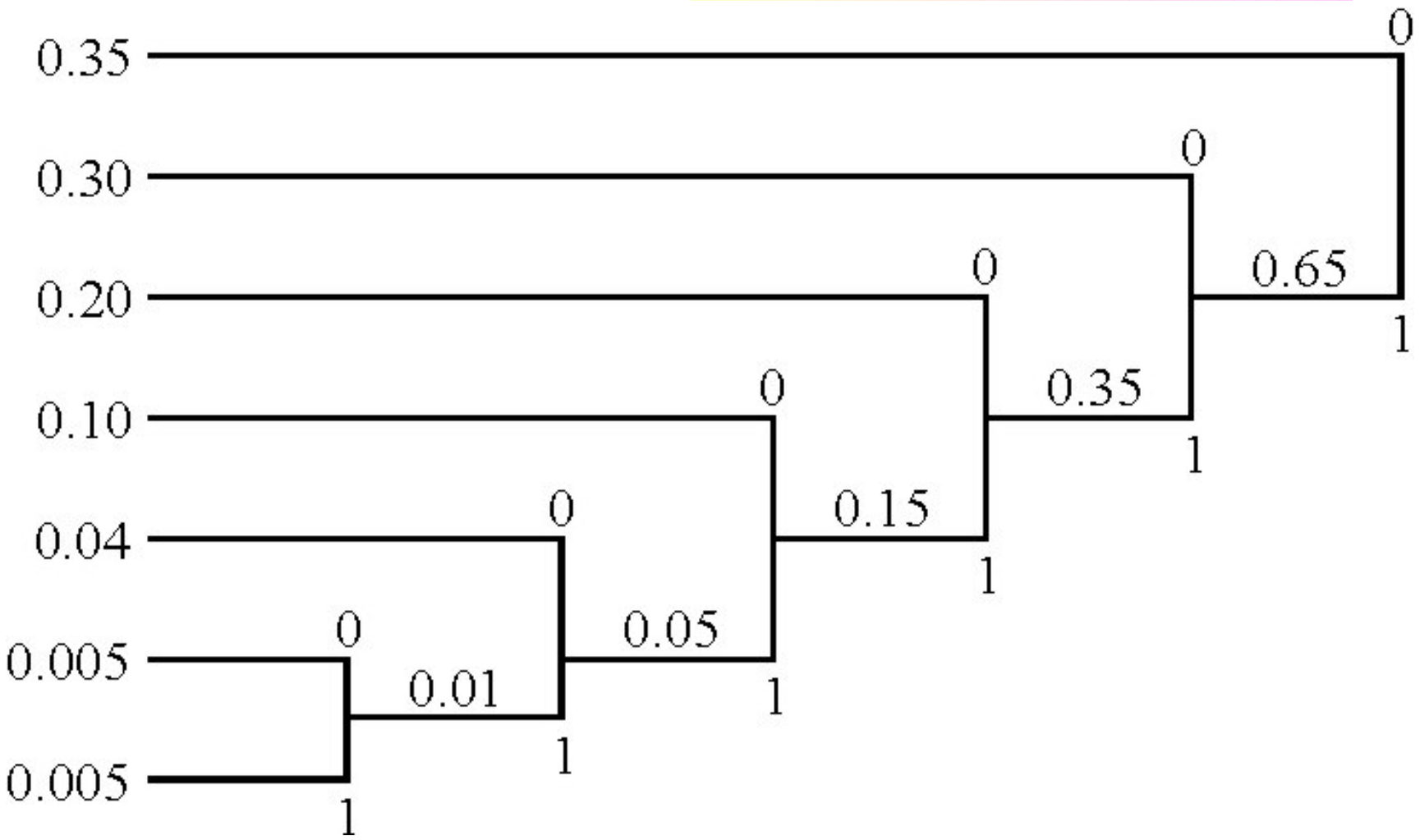
The algorithm minimizes the average number of digits required for describing the source.

Order the probabilities

$$p(x_1) \geq p(x_1) \geq \ldots \geq p(x_L)$$

Encoding algorithm

1. Select two least probable symbols and tie them together
2. Assign to the upper branch of the tied symbol 0 and to the lower branch 1
3. Replace the two lowest probability symbols with the new combined symbol. The combined symbol probability is the sum of the probabilities of the initial symbols
4. If there are any symbol left go back to step 1

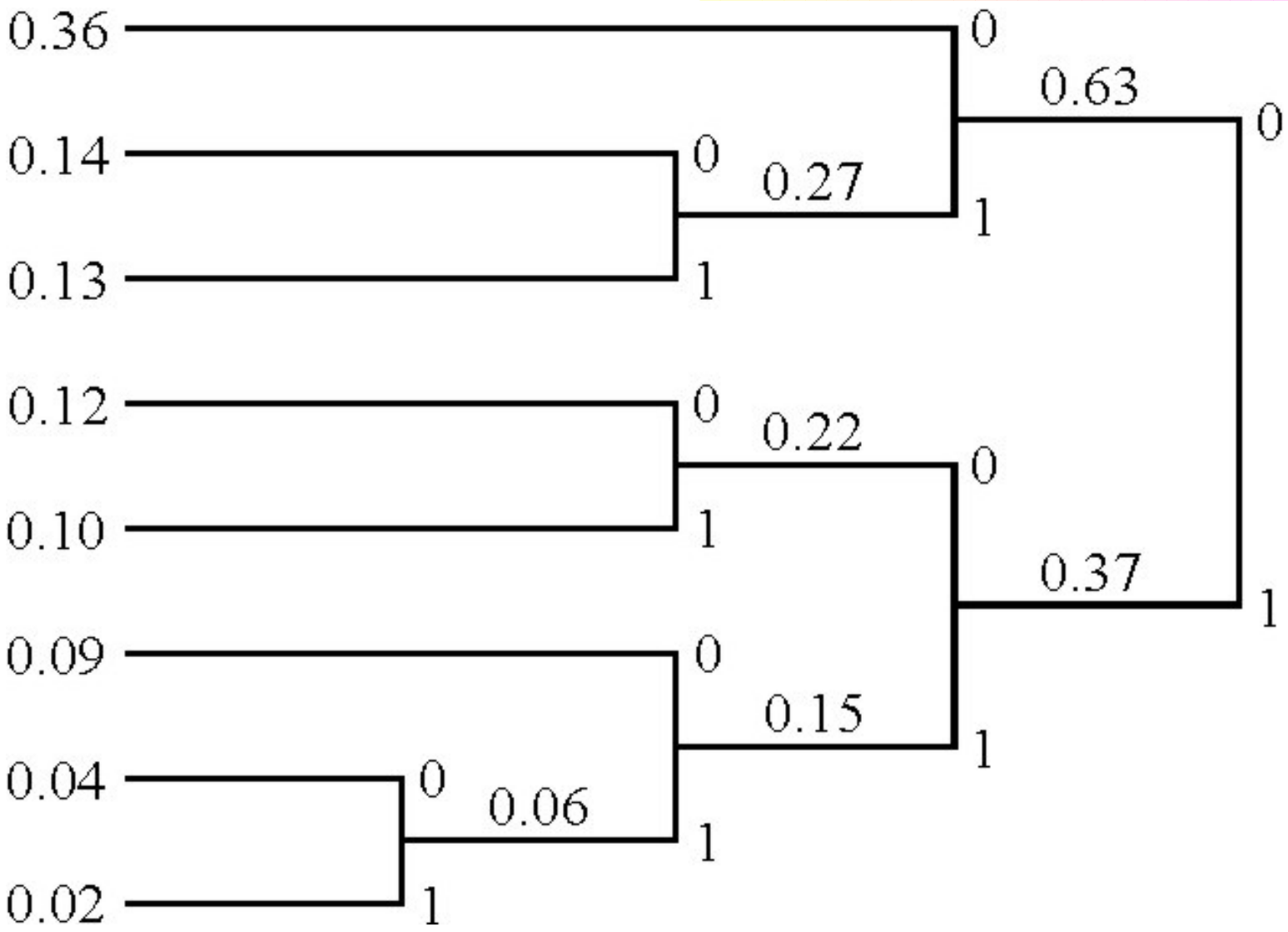


Example 1 Huffman coding

Input symbols	Pb	Self information $-\log_2\left(P(a_i)\right)$	Assigned bit sequence	Length of codeword l_i
a_1	0.35	1.51	0	1
a_2	0.30	1.74	01	2
a_3	0.20	2.32	011	3
a_4	0.10	3.32	0111	4
a_5	0.04	4.64	01111	5
a_6	0.005	7.64	011111	6
a_7	0.005	7.64	111111	6

Entropy $-\sum_{i=1}^{7} P(a_i)\log_2\left(P(a_i)\right) = 2.11$

Average codeword length $-\sum_{i=1}^{7} P(a_i) l_i = 2.21$



Example 2 Huffmann coding

Input symbols	Pb	Self information $-\log_2\left(P(a_i)\right)$	Assigned bit sequence	Length of codeword l_i
a_1	0.36	1.47	00	2
a_2	0.14	2.83	010	3
a_3	0.13	2.94	011	3
a_4	0.12	3.04	100	3
a_5	0.1	3.32	101	3
a_6	0.09	3.47	110	3
a_7	0.04	4.64	1110	4
a_8	0.02	5.64	1111	4

Entropy $-\sum_{i=1}^{7} P(a_i)\log_2\left(P(a_i)\right) = 2.62$

Average codeword length $-\sum_{i=1}^{7} P(a_i) l_i = 2.7$

Lempel Ziv coding

Adaptive code – From the input data is created a codebook by parsing trough the data vector.
Each element in the codebook is addressed as the address of the prefix of the bit sequence observed plus one new bit

Example
Input sequence
1010110100100111010100001100111010110001101 1
extracted phrases
1 0 10 11 01 00 100 111 010 1000 011 001 110 101 10001 1011

	Dict. location	Dict. content	Codeword
1	0001	1	00001
2	0010	0	00000
3	0011	10	00010
4	0100	11	00011
5	0101	01	00101
6	0110	00	00100
7	0111	100	00110
8	1000	111	01001
9	1001	010	01010
10	1010	1000	01110
11	1011	011	01011
12	1100	001	01101
13	1101	110	01000
14	1110	101	00111
15	1111	10001	10101
16		1011	11101

Rate distortion theory

In practical source coding there are constraints that make the process imperfect

This causes distortion which should be kept on a tolerable level

Therefore we talk about source encoding with a fidelity criterion

Rate distortion theory deals with this situation

With a limited code word length there is a finite set of code words and e.g. analogue information cannot be exactly represented $\rightarrow$ lossy data compression