Module 2

LOSSLESS IMAGE COMPRESSION SYSTEMS

Lesson 3

Lossless Compression: Huffman Coding

Version 2 ECE IIT, Kharagpur

Instructional Objectives

At the end of this lesson, the students should be able to:

1. Define and measure source entropy.
2. State Shannon's Coding theorem for noiseless channels.
3. Measure the coding efficiency of an encoding scheme.
4. State the basic principles of Huffman coding.
5. Assign Huffman codes to a set of symbols of known probabilities.
6. Encode a string of symbols into a Huffman coded bit stream.
7. Decode a Huffman coded bit stream.

3.0 Introduction

In lesson-2, we have learnt the basic difference between lossless and lossy image compression schemes. There are some applications, such as satellite image analysis, medical and business document archival, medical images for diagnosis etc, where loss may not be tolerable and lossless compression techniques are to be used. Some of the popular lossless image compression techniques being used are – (a) Huffman coding, (b) Arithmetic coding, (c) Ziv-Lempel coding, (d) Bit-plane coding, (e) Run-length coding etc.
In this lesson, we first begin with the information-theoretic basis of lossless coding, see how to measure the information content of a set of source symbols based on the probabilities of the symbols. We shall thereafter present Shannon's thorem which provides the theoretical lower bound of bit-rate achievable through lossless compression techniques. The rest of the lesson is devoted to the detailed treatment of Huffman coding, which is one of the most popular lossless compression techniques adopted in multimedia standards.

3.1 Source entropy- a measure of information content

Generation of information is generally modeled as a random process that has probability associated with it. If *P(E)* is the probability of an event, its information content *I(E),* also known as self information is measured as

$$I(E) = \log \frac{1}{P(E)} \quad \text{.....(3.1)}$$

If *P(E)=1,* that is, the event always occurs (like saying "*The sun rises in the east"*), then we obtain from above that *I(E)=0,* which means that there is no information associated with it. The base of the logarithm expresses the unit of information and if the base is 2, the unit is *bits.* For other values *m* of the base,

the information is expressed as *m-ary* units. Unless otherwise mentioned, we shall be using the base-2 system to measure information content.

Now, suppose that we have an alphabet of *n* symbols $\{a_i \mid i = 1,2,.....,n\}$ having probabilities of occurrences $P(a_1), P(a_2),....., P(a_n)$. If *k* is the number of source outputs generated, which is considered to be sufficiently large, then the average number of occurrences of symbol a_i is $kP(a_i)$ and the average self-information obtained from *k* outputs is given by

$$-k\sum_{i=1}^{n} P(a_i)\log P(a_i)$$

and the average information per source output for the source z is given by

$$H(\mathbf{z}) = -\sum_{i=1}^{n} P(a_i)\log P(a_i) \quad \text{.....(3.2)}$$

The above quantity is defined as the *entropy* of the source and measures the uncertainty of the source. The relationship between uncertainty and entropy can be illustrated by a simple example of two symbols a_1 and a_2, having probabilities $P(a_1)$ and $P(a_2)$ respectively. Since, the summation of probabilities is equal to 1,

$$P(a_2) = 1 - P(a_1)$$

and using equation (3.2), we obtain

$$H(\mathbf{z}) = -P(a_1)\log P(a_1) - (1 - P(a_1))\log(1 - P(a_1)) \quad \text{.....(3.3)}$$

If we plot *H*(z) versus $P(a_1)$, we obtain the graph shown in Fig.3.1.

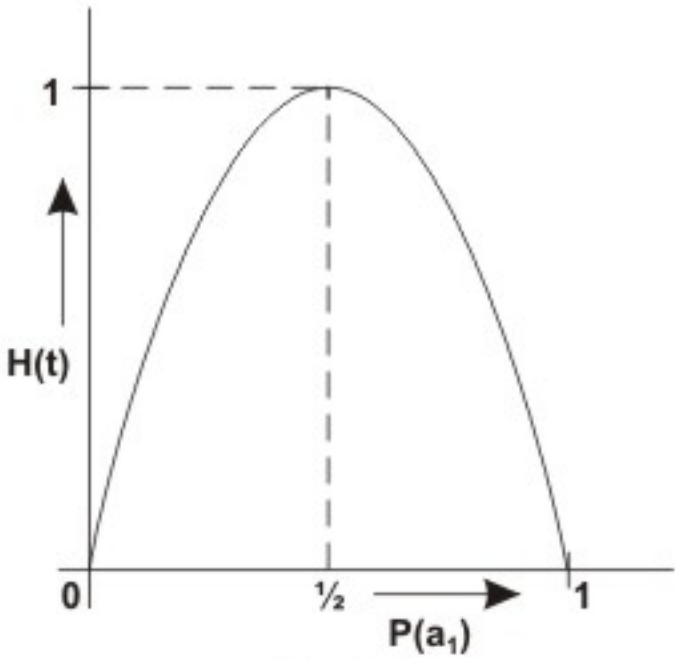

Fig. 3.1

It is interesting to note that the entropy is equal to zero for $P(a_1)=0$ and $P(a_1)=1$. These correspond to the cases where at least one of the two symbols is certain to be present. $H(z)$ assumes maximum value of 1-bit for $P(a_1)=1/2$. This corresponds to the most uncertain case, where both the symbols are equally probable.

3.1.1 Example: Measurement of source entropy

If the probabilities of the source symbols are known, the *source entropy* can be measured using equation (3.2). Say, we have five symbols $a_1, a_2, ..., a_5$ having the following probabilities:

$P(a_1)=0.2,\ P(a_2)=0.1,\ P(a_3)=0.05,\ P(a_4)=0.6,\ P(a_5)=0.05$

Using equation (3.2), the source entropy is given by

$H(\mathbf{z})=-0.2\log 0.2-0.1\log 0.1-0.05\log 0.05-0.6\log 0.6-0.05\log 0.05$ bits

= 1.67 bits

3.2 Shannon's Coding Theorem for noiseless channels

We are now going to present a very important theorem by Shannon, which expresses the lower limit of the average code word length of a source in terms of its entropy. Stated formally, the theorem states that *in any coding scheme, the average code word length of a source of symbols can at best be equal to the source entropy and can never be less than it.* The above theorem assumes the coding to be lossless and the channel to be noiseless.

If *m*(z) is the minimum of the average code word length obtained out of different uniquely decipherable coding schemes, then as per Shannon's theorem, we can state that

$$m(\mathbf{z}) \geq H(\mathbf{z}) \quad \text{.....(3.4)}$$

3.3 Coding efficiency

The coding efficiency (η) of an encoding scheme is expressed as the ratio of the source entropy H(z) *to the average code word length L*(z) *and is given by*

$$\eta = \frac{H(\mathbf{z})}{L(\mathbf{z})} \quad \text{.....(3.5)}$$

Since $L(\mathbf{z}) \geq H(\mathbf{z})$ according to Shannon's Coding theorem and both *L*(z) and *H*(z) are positive,

$$0 \leq \eta \leq 1 \quad \text{.....(3.6)}$$

3.4 Basic principles of Huffman Coding

Huffman coding is a popular lossless Variable Length Coding (VLC) (Section-2.4.3) scheme, based on the following principles:

(a) Shorter code words are assigned to more probable symbols and longer code words are assigned to less probable symbols.

(b) No code word of a symbol is a prefix of another code word. This makes Huffman coding uniquely decodable.

(c) Every source symbol must have a unique code word assigned to it.

In image compression systems (Section-2.4), Huffman coding is performed on the quantized symbols. Quite often, Huffman coding is used in conjunction with other lossless coding schemes, such as *run-length coding*, to be discussed in lesson-4. In terms of Shannon's noiseless coding theorem, Huffman coding is optimal for a fixed alphabet size, subject to the constraint that that the source symbols are coded one at a time.

3.5 Assigning Binary Huffman codes to a set of symbols

We shall now discuss how Huffman codes are assigned to a set of source symbols of known probability. If the probabilities are not known *a priori,* it should be estimated from a sufficiently large set of samples. The code assignment is based on a series of source reductions and we shall illustrate this with reference to the example shown in Section-3.1.1. The steps are as follows:

Step-1: Arrange the symbols in the decreasing order of their probabilities.

Symbol	Probability
a_4	0.6
a_1	0.2
a_2	0.1
a_3	0.05
a_5	0.05

Step-2: Combine the lowest probability symbols into a single compound symbol that replaces them in the next source reduction.

Symbol	Probability
a_4	$P(a_4)=0.6$
a_1	$P(a_1)=0.2$
a_2	$P(a_2)= 0.1$
$a_3 \, V \, a_5$	$P(a_3)+P(a_5)=0.1$

In this example, a_3 and a_5 are combined into a compound symbol of probability 0.1.

Step-3: Continue the source reductions of *Step-2,* until we are left with only two symbols.

Symbol	Probability
a_4	$P(a_4)=0.6$
a_1	$P(a_1)=0.2$
$a_2 \, V \, (a_3 \, V \, a_5)$	$P(a_2)+P(a_3)+P(a_5)=0.2$

Symbol	Probability
a_4	$P(a_4)=0.6$
$a_1 \, V \, (a_2 \, V \, (a_3 \, V \, a_5))$	$P(a_1)+P(a_2)+P(a_3)+P(a_5)=0.4$

The second symbol in this table indicates a compound symbol of probability 0.4. We are now in a position to assign codes to the symbols.

Step-4: Assign codes "0" and "1" to the last two symbols.

Symbol	Probability	Assigned Code
a_4	0.6	0
a_1	0.2	10
a_2	0.1	110
a_3 V a_5	0.1	111

In this case, "0" is assigned to the symbol a_4 and "1" is assigned to the compound symbol a_1 V (a_2 V (a_3 $V a_5$)). All the elements within this compound symbol will therefore have a prefix "1".

Step-5: Work backwards along the table to assign the codes to the elements of the compound symbols. Continue till codes are assigned to all the elementary symbols.

Symbol	Probability	Assigned Code
a_4	0.6	0
$a_1 V$ (a_2 V($a_3 V a_5$))	0.4	1

"11" is therefore going to be the prefix of a_2, a_3 and a_5, since this is the code assigned to the compound symbol of these three.

This completes the Huffman code assignment pertaining to this example. From this table, it is evident that shortest code word (length=1) is assigned to the most probable symbol a_4 and the longest code words (length=4) are assigned to the two least probable symbols a_3 and a_5. Also, each symbol has a unique code and no code word is a prefix of code word for another symbol. The coding has therefore fulfilled the basic requirements of Huffman coding, stated in Section-3.4.

Symbol	Probability	Assigned Code
a_4	0.6	0
a_1	0.2	10
a_2 V($a_3 V a_5$)	0.2	11

For this example, we can compute the average code word length. If $L(a_i)$ is the codeword length of symbol a_i, then the average codeword length is given by

Symbol	Probability	Assigned Code
a_4	0.6	0
a_1	0.2	10
a_2	0.1	110
a_3	0.05	1110
a_5	0.05	1111

$$L(\mathbf{z}) = \sum_{i=1}^{n} L(a_i)P(a_i)$$
$$= 0.6 \times 1 + 0.2 \times 2 + 0.1 \times 3 + 2 \times 0.05 \times 4$$
$$= 1.7 \quad \text{bits}$$

The coding efficiency is given by

$$\eta = \frac{H(\mathbf{z})}{L(\mathbf{z})} = 0.982$$

3.6 Encoding a string of symbols using Huffman codes

After obtaining the Huffman codes for each symbol, it is easy to construct the encoded bit stream for a string of symbols. For example, if we have to encode a string of symbols

$a_4 a_3 a_5 a_4 a_1 a_4 a_2$, we shall start from the left, taking one symbol at a time. The code corresponding to the first symbol a_4 is 0, the second symbol a_3 has a code 1110 and so on. Proceeding as above, we obtain the encoded bit stream as 0111011110100110.

In this example, 16 bits were used to encode the string of 7 symbols. A straight binary encoding of 7 symbols, chosen from an alphabet of 5 symbols would have required 21 bits (3 bits/symbol) and this encoding scheme therefore demonstrates substantial compression.

3.7 Decoding a Huffman coded bit stream

Since no codeword is a prefix of another codeword, Huffman codes are uniquely decodable. The decoding process is straightforward and can be summarized below:

Step-1: Examine the leftmost bit in the bit stream. If this corresponds to the codeword of an elementary symbol, add that symbol to the list of decoded symbols, remove the examined bit from the bit stream and go back to *step-1* until all the bits in the bit stream are considered. Else, follow *step-2.*

Step-2: Append the next bit from the left to the already examined bit(s) and examine if the group of bits correspond to the codeword of an elementary symbol. If yes, add that symbol to the list of decoded symbols, remove the examined bits from the bit stream and go back to *step-1* until all the bits in the bit stream are considered. Else, repeat *step-2* by appending more bits.

In the encoded bit stream example of Section-3.6, if we receive the bit stream 0111011110100110 and follow the steps described above, we shall first decode a_4 (“0”), then a_3 (“1110”), followed by a_5 (“1111”), a_4 (“0”),a_1 (“10”), a_4 (“0”) and a_2 (“110”). This is exactly what we had encoded.

3.8 Discussions and Further Reading

In this lesson, we have discussed *Huffman Coding,* which is one of the very popular lossless Variable Length Coding (VLC) schemes, named after the scientist who proposed it. The details of the coding scheme can be read from his original paper, listed in [1]. For a better understanding of the coding theory in the light of Shannon’s theorem, the reader is referred to Shannon’s original paper, listed in [2].

We have discussed how Huffman codes can be constructed from the probabilities of the symbols. The symbol probabilities can be obtained by making a relative frequency of occurrences of the symbols and these essentially make a *first order estimate* of the *source entropy*. Better source entropy estimates can be obtained if we examine the relative frequency of occurrences of a group of symbols, say by considering two consecutive symbols at a time. With reference to images, we can form pairs of gray level values, considering two consecutive pixels at a time and thus form a *second order estimate* of the *source entropy*. In this case, Huffman codes will be assigned to the pair of symbols, instead of individual symbols. Although *third, fourth and even higher order estimates* would make better approximations of the source entropy, the convergence will be slow and excessive computations are involved.

We have also seen that Huffman coding assignment is based on successive source reductions. For an n-symbol source, (n-2) source reductions must be performed. When n is large, like in the case of gray level values of the images for which n=256, we require 254 steps of reduction, which is excessively high. In such cases, Huffman coding is done only for few symbols of higher probability and for the remaining, a suitable prefix code, followed by a fixed length code is adopted. This scheme is referred to as *truncated Huffman coding.* It is somewhat less optimal as compared to *Huffman Coding*, but code assignment is much easier.

There are other variants of Huffman Coding. In one of the variants, the source symbols, arranged in order of decreasing probabilities are divided into a few blocks. Special shift up and/or shift down symbols are used to identify each block and symbols within the block are assigned Huffman codes. This encoding scheme is referred to as *Shift Huffman Coding.* The shift symbol is the most probable symbol and is assigned the shortest code word. Interested readers may refer to [3] for further discussions on Huffman Coding variants.

Questions

NOTE: The students are advised to thoroughly read this lesson first and then answer the following questions. Only after attempting all the questions, they should click to the solution button and verify their answers.

PART-A

A.1. Define the entropy of a source of symbols.

A.2. How is entropy related to uncertainty?

A.3. State Shannon's coding theorem on noiseless channels.

A.4. Define the coding efficiency of an encoding scheme.

A.5. State the basic principles of Huffman coding.

PART-B: Multiple Choice

In the following questions, click the best out of the four choices.

B.1 The entropy of a source of symbols is dependent upon

(A) The number of source outputs generated.

(B) The average codeword length.

(C) The probabilities of the source symbols.

(D) The order in which the source outputs are generated.

B.2 We have two sources of symbols to compare their entropies. Source-1 has three symbols a_1, a_2 and a_3 with probabilities $P(a_1) = 0.9, P(a_2) = P(a_3) = 0.05$. Source-2 also has three symbols a_1, a_2 and a_3, but with probabilities $P(a_1) = 0.4, P(a_2) = P(a_3) = 0.3$.

(A) Entropy of source-1 is higher than that of source-2.

(B) Entropy of source-1 is lower than that of source-2.

(C) Entropy of source-1 and source-2 are the same.

(D) It is not possible to compute the entropies from the given data.

B.3 Shannon's coding theorem on noiseless channels provides us with

(A) A lower bound on the average codeword length.

(B) An upper bound on the average codeword length

(C) A lower bound on the source entropy.

(D) An upper bound on the source entropy.

B.4 Which one of the following is not true for Huffman coding?

(A) No codeword of an elementary symbol is a prefix of another elementary symbol.

(B) Each symbol has a one-to-one mapping with its corresponding codeword.

(C) The symbols are encoded as a group, rather than encoding one symbol at a time.

(D) Shorter code words are assigned to more probable symbols.

B.5 A source of 4 symbols a_1, a_2, a_3, a_4 having probabilities $P(a_1) = 0.5, P(a_2) = 0.25, P(a_3) = P(a_4) = 0.125$ are encoded by four different encoding schemes and the corresponding codes are shown below. Which of the following gives us the best coding efficiency?

(A) $a_1 = 00, a_2 = 01, a_3 = 10, a_4 = 11$

(B) $a_1 = 0, a_2 = 10, a_3 = 110, a_4 = 111$

(C) $a_1 = 00, a_2 = 100, a_3 = 1100, a_4 = 1101$

(D) $a_1 = 111, a_2 = 110, a_3 = 10, a_4 = 0$

B.6 Which of the following must be ensured before assigning binary Huffman codes to a set of symbols?

(A) The channel is noiseless.

(B) There must be exactly 2^n symbols to encode.

(C) No two symbols should have the same probability.

(D) The probabilities of the symbols should be known *a priori*.

B.7 Refer to the Huffman code words assigned to five symbols $a_1, a_2, ..., a_5$ in the example shown in Section-3.5.The bit stream assigned to the sequence of symbols $a_1 a_4 a_5 a_4 a_1 a_4 a_2 a_4 a_4 a_1 a_4$ is

(A) 10011110100110000100

(B) 10011010110111010010

(C) 10011110111011000100

(D) 01100001011001110 11

B.8 A 4-symbol alphabet has the following probabilities $P(a_1) = 0.1, P(a_2) = 0.5, P(a_3) = 0.25, P(a_4) = 0.15$ and following codes are assigned to the symbols $a_1 = 110, a_2 = 0, a_3 = 10, a_4 = 111$. The average code word length for this source is

(A) 1.25 (B) 1.5

(C) 1.75 (D) 2.0

B.9 Decode a Huffman encoded bit stream 10011011110011001 10 which follows the codes assignment of the above problem. The sequence of symbols is

(A) $a_3 a_2 a_3 a_1 a_3 a_2 a_1 a_2 a_1$ (B) $a_3 a_2 a_1 a_4 a_3 a_2 a_1 a_3 a_1$

(C) $a_3 a_2 a_1 a_3 a_2 a_1 a_2 a_1$ (D) $a_3 a_2 a_1 a_4 a_3 a_2 a_1 a_2 a_1$

PART-C: Problems

C-1. A long sequence of symbols generated from a source is seen to have the following occurrences

Symbol	Occurrences
a_1	3003
a_2	996
a_3	2017
a_4	1487
a_5	2497

(a) Assign Huffman codes to the above symbols, following a convention that the group/symbol with higher probability is assigned a “0” and that with lower probability is assigned a “1”.

(b) Calculate the entropy of the source.

(c) Calculate the average code word length obtained from Huffman coding.

(d) Calculate the coding efficiency.

(e) Why is the coding efficiency less than 1?

SOLUTIONS

A.1 The entropy of a source of symbols is defined as the average information per source output. If we have an alphabet z of *n* symbols $\{a_i \mid i = 1,2,.....,n\}$ having probabilities of occurrences $P(a_1), P(a_2),....., P(a_n)$, the entropy of the source *H*(z) is given by

$$H(\mathbf{z}) = -\sum_{i=1}^{n} P(a_i)\log P(a_i)$$

The unit of entropy is dependent upon the base of the logarithm. For a base of 2, the unit is bits. In general, for a base *m*, the unit of entropy is *m-ary* units.

A.2 Entropy of a source is related to the uncertainty of the symbols associated with it. Greater the uncertainty, higher is the entropy. We can illustrate this by the two-symbol example discussed in Section-3.1

A.3 Shannon's coding theorem on lossless channels states that *in any coding scheme, the average code word length of a source of symbols can at best be equal to the source entropy and can never exceed it*.

If *m*(z) is the minimum of the average code word length obtained out of different uniquely decipherable coding schemes, then Shannon's theorem states that

$$m(\mathbf{z}) \geq H(\mathbf{z})$$

A.4 Refer to Section-3.3

A.5 Refer to Section-3.4

B.1 (C) B.2 (B) B.3 (A) B.4 (C) B.5 (B)

B.6 (D) B.7 (A) B.8 (C) B.9 (D).C.1

(a) Since the symbols are observed for a sufficiently long sequence, the probabilities can be estimated from their relative frequencies of occurrence.
$P(a_1) \approx 0.3, P(a_2) \approx 0.1, P(a_3) \approx 0.2, P(a_4) \approx 0.15, P(a_5) \approx 0.25$

Based on these probabilities, the source reductions can be done as follows:

Symbol	Prob.	Reduction-1		Reduction-2		Reduction -3	
		Symbol	Prob	Symbol	Prob	Symbol	Prob
a_1	0.3	a_1	0.3	$a_4+a_2+a_3$	0.45	a_1+a_5	0.55
a_5	0.25	a_5	0.25	a_1	0.3	$a_4+a_2+a_3$	0.45
a_3	0.2	a_4+a_2	0.25	a_5	0.25		
a_4	0.15	a_3	0.2				
a_2	0.1						

We can now work backwards to assign Huffman codes to the compound symbols and proceed to the elementary symbols

Reduction-3		Reduction-2		Reduction-1		Original	
Symbol	Code	Symbol	Code	Symbol	Code	Symbol	Code
a_1+a_5	0	a_1	00	a_1	00	a_1	00
$a_4+a_2+a_3$	1	a_5	01	a_5	01	a_5	01
		$a_4+a_2+a_3$	1	a_4+a_2	10	a_3	11
				a_3	11	a_4	100
						a_2	101

(b) The source entropy is given by

$$H(\mathbf{z}) = -\sum_{i=1}^{n} P(a_i)\log a_i$$

$$= -0.3\log 0.3 - 0.1\log 0.1 - 0.2\log 0.2 - 0.15\log 0.15 - 0.25\log 0.25 \quad \text{bits/ symbol}$$

$$= 2.227 \text{ bits/ symbol}$$

(c) The average code word length is given by

$$L(\mathbf{z}) = \sum_{i=1}^{n} P(a_i)L(a_i)$$

$$= 0.3\times 2 + 0.1\times 3 + 0.2\times 2 + 0.15\times 3 + 0.25\times 2 \quad \text{bits/ symbol}$$

$$= 2.25 \text{ bits/ symbol}$$

(d) The coding efficiency

$$\eta = \frac{H(\mathbf{z})}{L(\mathbf{z})} = \frac{2.227}{2.25} = 0.9897$$

(e) The student should think of this reason and check later.

References

1. Huffman, D.A., “A Method for the Construction of Minimum Redundancy Codes”, *Proc. IRE*, vol.40, no.10, pp.1098-1101, 1952.
2. Shannon, C.E., “A Mathematical Theory of Communications”, *The Bell Sys. Tech. J.,* vol. XXVII, no. 3, pp.379-423, 1948.
3.