CHAPTER 9

Information theory and source coding

9.1 Introduction

There may be a number of reasons for wishing to change the form of a digital signal as supplied by an information source prior to transmission. In the case of English language text, for example, we start with a data source consisting of about 40 distinct symbols (the letters of the alphabet, integers and punctuation). In principle we could transmit such text using a signal alphabet consisting of 40 distinct voltage waveforms. This would constitute an M-ary system where $M = 40$ unique signals. It may be, however, that for one or more of the following reasons this approach is inconvenient, difficult or impossible:

1. The transmission channel may be physically unsuited to carrying such a large number of distinct signals.
2. The relative frequencies with which different source symbols occur will vary widely. This will have the effect of making the transmission inefficient in terms of the time it takes and/or bandwidth it requires.
3. The data may need to be stored and/or processed in some way before transmission. This is most easily achieved using binary electronic devices as the storage and processing elements.

For all these reasons (and perhaps others) sources of digital information are almost always converted as soon as possible into binary form, i.e. each symbol is encoded as a binary word. After appropriate processing the binary words may then be transmitted directly, as either baseband or bandpass signals, or may be recoded into another multi-symbol alphabet. (In the latter case it is unlikely that the transmitted symbols map directly onto the original source symbols.) The body of knowledge, information theory, concerned with the representation of information by symbols gives theoretical bounds on the performance of communication systems and permits assessment of practical system efficiency. The landmarks in information theory were developed by Hartley and Nyquist in the 1920s and are summarised in [Shannon, 1948].

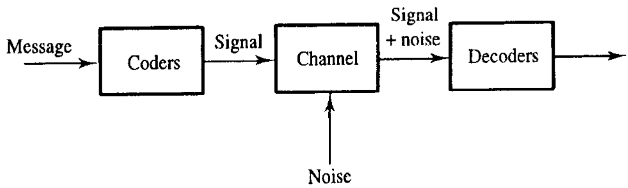

Figure 9.1 *The communications channel model.*

The simplified communications system shown in Figure 9.1, may include several encoders and decoders to implement one or more of the following processes:

- Formatting (which transforms information from its original, or natural, form to a well defined, and standard, digital form, e.g. PCM, see Chapter 6).
- Source coding (which reduces the average number of symbols required to transmit a given message).
- Encryption (which codes messages using a cipher to prevent unauthorised reception or transmission).
- Error control coding (which allows a receiver to detect, and sometimes correct, symbols which are received in error, see Chapter 10).
- Line coding/pulse shaping (which ensures the transmitted symbol waveforms are well suited to the characteristics of the channel, see Chapters 6 and 8).

It is the second of these processes, source coding, which is the principal concern of this chapter.

9.2 Information and entropy

9.2.1 The information measure

The concept of information content is related to predictability or scarcity value. That is, the more predictable or probable a particular message, the less information is conveyed by transmitting that message. For example, the football score Manchester United 7, Bradford Academicals 0, contains little information. The information content of Bradford Academicals 7, Manchester United 0, on the other hand is enormous![1]

In essence highly probable messages contain little information and we can write:

$$P(message) = 1 \text{ carries zero information}$$

$$P(message) = 0 \text{ carries infinite information}$$

The definition of information content for a symbol m should be such that it monotonically decreases with increasing message probability, $P(m)$, and it goes to zero

[1] Readers must be aware that Manchester United is a premier division team while Bradford Academicals is a non-league team and thus the chances are very remote for the second score to occur.

for a probability of unity. Another desirable property is that of additivity. If one were to communicate two (independent) messages in sequence, the total information content should be equal to the sum of the individual information contents of the two messages. We know that the total probability of the composite message is the product of the two individual, independent, probabilities. Therefore, the definition of information must be such that when probabilities are multiplied information is added.

The required properties of an information measure are summarised in Table 9.1.

Table 9.1 *Information measures.*

Message	*P (message)*	*Information content*
m_1	$P(m_1)$	I_1
m_2	$P(m_2)$	I_2
$(m_1 + m_2)$	$P(m_1)P(m_2)$	$I_1 + I_2$

The logarithm operation clearly satisfies these requirements. We thus *define* ($\underline{\underline{\Delta}}$) the information content, I_m, of a message, m, as:

$$I_m \underline{\underline{\Delta}} \log \frac{1}{P(m)} \equiv -\log P(m) \tag{9.1}$$

This definition satisfies the additivity requirement, the monotonicity requirement, and for $P(m) = 1$, $I_m = 0$. Note that this is true regardless of the base chosen for the logarithm. Base 2 is usually chosen, however, the resulting quantity of information being measured in bits:

$$I_1 + I_2 = -\log_2 P(m_1) - \log_2 P(m_2) = -\log_2 [P(m_1)\, P(m_2)] \quad \text{(bits)} \tag{9.2}$$

Table 9.2 *Word length and symbol probabilities.*

Vocabulary size	*No. binary digits*	*Symbol probability*
2	1	1/2
4	2	1/4
8	3	1/8
•	•	•
•	•	•
•	•	•
128	7	1/128

9.2.2 Multisymbol alphabets

Consider, initially, vocabularies of equiprobable message symbols represented by fixed length binary codewords. Thus a vocabulary size of four symbols is represented by the binary digit pairs 00, 01, 10, 11. The binary word length and symbol probabilities for other vocabulary sizes are shown in Table 9.2. The ASCII vocabulary used in teleprinters contains 128 symbols and therefore uses a $\log_2 128 = 7$ digit (fixed length) binary code word to represent each symbol. ASCII symbols are not equiprobable, however, and for this particular code each symbol does not, therefore, have a selection probability of 1/128.

9.2.3 Commonly confused entities

Several information-related quantities are sometimes confused. These are:

- Symbol — A member of a source alphabet. May or may not be binary, e.g. 2 symbol binary, 4 symbol PSK (see Chapter 11), 128 symbol ASCII.
- Baud — Rate of symbol transmission, i.e. 100 baud = 100 symbol/s.
- Bit — Quantity of information carried by a symbol with selection probability $P = 0.5$.
- Bit rate — Rate of information transmission (bit/s). (In the special, but common, case of signalling using independent, equiprobable, binary symbols, the bit rate equals the baud rate.)
- Message — A meaningful sequence of symbols. (Also often used to mean a source symbol.)

9.2.4 Entropy of a binary source

Entropy (H) is defined as the average amount of information conveyed per symbol. For an alphabet of size 2 and assuming that symbols are statistically independent:

$$H \underset{=}{\Delta} \sum_{m=1}^{2} P(m) \log_2 \frac{1}{P(m)} \quad \text{(bit/symbol)} \tag{9.3}$$

For the two symbol alphabet (0, 1) if we let $P(1) = p$ then $P(0) = 1 - p$ and:

$$H = p \log_2 \frac{1}{p} + (1-p) \log_2 \frac{1}{1-p} \quad \text{(bit/symbol)} \tag{9.4}$$

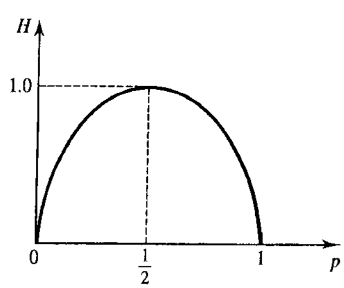

Figure 9.2 *Entropy for binary data transmission versus the selection probability, p, of a digital 1.*

The entropy is maximised when the symbols are equiprobable as shown in Figure 9.2. The entropy definition of equation (9.3) holds for all alphabet sizes. Note that, in the binary case, as either of the two messages becomes more likely, the entropy decreases. When either message has probability 1, the entropy goes to zero. This is reasonable since, at these points, the outcome of the transmission is certain. Thus, if $P(0) = 1$, we know the symbol 0 will be sent repeatedly. If $P(0) = 0$, we know the symbol 1 will be sent repeatedly. In these two cases, no information is conveyed by transmitting the symbols.

For the ASCII alphabet, source entropy would be given by $H = -\log_2 (1/128) = 7$ bit/symbol if all the symbols were both *equiprobable* and *statistically independent*. In practice H is less than this, i.e.:

$$H \triangleq \sum_{m=1}^{128} P(m) \log_2 \frac{1}{P(m)} < 7 \text{ bit/symbol} \tag{9.5}$$

since the symbols are neither, making the code less than 100% efficient. Entropy thus indicates the *minimum* number of binary digits required per symbol (averaged over a long sequence of symbols).

9.3 Conditional entropy and redundancy

For sources in which each symbol selected is not statistically independent from all previous symbols (i.e. sources with memory) equation (9.3) is insufficiently general to give the entropy correctly. In this case the joint and conditional statistics (section 3.2.1) of symbol sequences must be considered. A source with a memory of one symbol, for example, has an entropy given by:

$$H = \sum_i \sum_j P(j, i) \log_2 \frac{1}{P(j|i)} \quad \text{(bit/symbol)} \tag{9.6}$$

where $P(j, i)$ is the probability of the source selecting i and j and $P(j|i)$ is the probability that the source will select j given that it has previously selected i. Bayes's theorem, equation (3.3), can be used to re-express equation (9.6) as:

$$H = \sum_i P(i) \sum_j P(j|i) \log_2 \frac{1}{P(j|i)} \quad \text{(bit/symbol)} \tag{9.7}$$

(For independent symbols $P(j|i) = P(j)$ and equation (9.7) reduces to equation (9.3).) The effect of having dependency between symbols is to increase the probability of selecting some symbols at the expense of others *given a particular symbol history*. This reduces the average information conveyed by the symbols, which is reflected in a reduced entropy. The difference between the actual entropy of a source and the (maximum) entropy, H_{max}, the source could have if its symbols were independent and equiprobable is called the *redundancy* of the source. For an M symbol alphabet redundancy, R, is therefore given by:

$$R = H_{max} - H$$
$$= \log_2 (M) - H \quad \text{(bit/symbol)} \tag{9.8}$$

(It is easily shown that the quantity $R/(H^2 + RH)$ relates to the number of symbols per bit of information which are transmitted unnecessarily from an information theory point of view. There may well be good reasons, however, to transmit such redundant symbols, e.g. for error control purposes as discussed in Chapter 10.)

EXAMPLE 9.1

Find the entropy, redundancy and information rate of a four symbol source (A, B, C, D) with a baud rate of 1024 symbol/s and symbol selection probabilities of 0.5, 0.2, 0.2 and 0.1 under the following conditions:

(i) The source is memoryless (i.e. the symbols are statistically independent).

(ii) The source has a one symbol memory such that no two consecutively selected symbols can be the same. (The long term relative frequencies of the symbols remain unchanged, however.)

(i)

$$H = \sum_{m=1}^{M} p(m) \log_2 \frac{1}{p(m)}$$
$$= 0.5 \log_2 \left(\frac{1}{0.5} \right) + 2 \times 0.2 \log_2 \left(\frac{1}{0.2} \right) + 0.1 \log_2 \left(\frac{1}{0.1} \right)$$
$$= 0.5 \frac{\log_{10} 2}{\log_{10} 2} + 0.4 \frac{\log_{10} 5}{\log_{10} 2} + 0.1 \frac{\log_{10} 10}{\log_{10} 2}$$
$$= 1.761 \quad \text{bit/symbol}$$
$$R = H_{max} - H$$
$$= \log_2 M - H$$
$$= \log_2 4 - 1.761 = 2.0 - 1.761 = 0.239 \quad \text{bit/symbol}$$
$$R_i = R_s H = 1024 \times 1.761 = 1.803 \times 10^3 \quad \text{bit/s}$$

where R_i is the information rate and R_s is the symbol rate.

(ii) The appropriate formula to apply to find the entropy of a source with one symbol memory is equation (9.7). First, however, we must find the conditional probabilities which the formula contains. If no two consecutive symbols can be the same then:

$$P(A|A) = P(B|B) = P(C|C) = P(D|D) = 0$$

Since the (unconditional) probability of A is unchanged, $P(A) = 0.5$, then every, and only every, alternate symbol must be A, $P(\bar{A}|A) = P(A|\bar{A}) = 1.0$, where $\bar{A}$ represents not A. Furthermore if every alternate symbol is A then no two non-A symbols can occur consecutively, $P(\bar{A}|\bar{A}) = 0$. Writing the above three probabilities explicitly:

$$P(B|A) + P(C|A) + P(D|A) = 1.0$$

$$P(A|B) = P(A|C) = P(A|D) = 1.0$$

$$P(B|C) = P(B|D) = P(C|B) = P(C|D) = P(D|B) = P(D|C) = 0$$

Since the (unconditional) probabilities of B, C and D are to remain unchanged, i.e.:

$$P(B) = P(C) = 0.2 \text{ and } P(D) = 0.1$$

the conditional probability $P(B|A)$ must satisfy:

$$\begin{aligned} P(B) &= P(B|A)P(A) + P(B|B)P(B) + P(B|C)P(C) + P(B|D)P(D) \\ &= P(B|A)0.5 + 0 + 0 + 0 \\ \text{i.e. } P(B|A) &= \frac{0.2}{0.5} = 0.4 \end{aligned}$$

Similarly:

$$\begin{aligned} P(C) &= P(C|A)P(A) + P(C|B)P(B) + P(C|C)P(C) + P(C|D)P(D) \\ &= P(C|A)\, 0.5 + 0 + 0 + 0 \\ \text{i.e. } P(C|A) &= \frac{0.2}{0.5} = 0.4 \\ P(D) &= P(D|A)P(A) + P(D|B)P(B) + P(D|C)P(C) + P(D|D)P(D) \\ &= P(D|A)\, 0.5 + 0 + 0 + 0 \\ \text{i.e. } P(D|A) &= \frac{0.1}{0.5} = 0.2 \end{aligned}$$

We now have numerical values for all the conditional probabilities which can be substituted into equation (9.7):

$$H = \sum_i P(i) \sum_j P(j|i) \log_2 \frac{1}{P(j|i)} \quad \text{(bit/symbol)}$$

$$\begin{aligned} &= P(A)\left[P(A|A)\log_2\frac{1}{P(A|A)} + P(B|A)\log_2\frac{1}{P(B|A)} + P(C|A)\log_2\frac{1}{P(C|A)} + P(D|A)\log_2\frac{1}{P(D|A)}\right] \\ &+ P(B)\left[P(A|B)\log_2\frac{1}{P(A|B)} + P(B|B)\log_2\frac{1}{P(B|B)} + P(C|B)\log_2\frac{1}{P(C|B)} + P(D|B)\log_2\frac{1}{P(D|B)}\right] \\ &+ P(C)\left[P(A|C)\log_2\frac{1}{P(A|C)} + P(B|C)\log_2\frac{1}{P(B|C)} + P(C|C)\log_2\frac{1}{P(C|C)} + P(D|C)\log_2\frac{1}{P(D|C)}\right] \\ &+ P(D)\left[P(A|D)\log_2\frac{1}{P(A|D)} + P(B|D)\log_2\frac{1}{P(B|D)} + P(C|D)\log_2\frac{1}{P(C|D)} + P(D|D)\log_2\frac{1}{P(D|D)}\right] \\ &= 0.5\left[\, 0.0\log_2(1/0) + 0.4\log_2 2.5 + 0.4\log_2 2.5 + 0.2\log_2 5.0\,\right] \\ &+ 0.2\left[\, 1.0\log_2 1.0 + 0 + 0 + 0\,\right] + 0.2\left[\, 1.0\log_2 1.0 + 0 + 0 + 0\,\right] \end{aligned}$$

$$+ \; 0.1\left[\; 1.0\log_2 1.0 + 0 + 0 + 0 \;\right]$$

$$= 0.5\left[\; 0.4\log_{10} 2.5 + 0.4\log_{10} 2.5 + 0.2\log_{10} 5.0 \;\right]/\log_{10} 2 \;=\; 0.761 \;\; \text{bit/symbol}$$

$$R \;=\; H_{\max} - H$$

$$= \log_2 4 - 0.761 \;=\; 1.239 \;\; \text{bit/symbol}$$

(Notice that in this case the symbol A carries no information at all since its occurrences are entirely predictable.)

$$R_i \;=\; R_s H = 1024 \times 0.761 = 779 \quad \text{bit/s}$$

9.4 Information loss due to noise

The information transmitted by a memoryless source (i.e. a source in which the symbols selected are statistically independent) is related only to the source symbol probabilities by equation (9.1), i.e.:

$$I_{TX}(i_{TX}) \;=\; \log_2 \frac{1}{P(i_{TX})} \quad \text{(bits)} \tag{9.9}$$

where $I_{TX}(i_{TX})$ is the information transmitted when the ith source symbol is selected for transmission and $P(i_{TX})$ is the (a priori) probability that the ith symbol will be selected (see section 7.2 for the definition of a priori probability). The use of a subscript TX with i, although not really necessary, will help with what can be a confusing notation later in the chapter. For a noiseless channel there is no doubt on detecting a given received voltage v_{RX} corresponding to a given transmitted symbol, $(v_{RX}\!:i_{TX})$, which source symbol was selected for transmission. The quantity of information, in this case, gained by receiving v_{RX} is therefore identical to that transmitted, i.e.:

$$I_{RX}(v_{RX}\!:i_{TX}) \;=\; I_{TX}(i_{TX})$$

$$= \log_2 \frac{1}{P(i_{TX})} \quad \text{(bits)} \tag{9.10}$$

For a noisy channel, however, there is some uncertainty given a detected voltage, v_{RX}, about which symbol was actually selected at the source. This uncertainty is related to the (a posteriori) probability, $P(i_{TX}|v_{RX})$, that symbol i was transmitted given that voltage v_{RX} was detected (see again section 7.2 for the definition of an a posteriori probability). (For the noiseless channel $P(i_{TX}|v_{RX})$ is unity for one particular source symbol and zero for all others.) Intuition tells us that for a noisy channel the information received should be less than that transmitted by an amount related to the uncertainty in the decision process. In fact the received information is given by:

$$I_{RX}(v_{RX}\!:i_{TX}) \;=\; \log_2 \frac{P(i_{TX}|v_{RX})}{P(i_{TX})} \quad \text{(bits)} \tag{9.11}$$

Using $P(i_{TX}|v_{RX})$ in equation (9.11) corresponds to a soft decision process (see Figure 7.1). For hard decision processes the detected voltage v_{RX} is converted immediately to a received symbol j_{RX}. In this case equation (9.11) is rewritten as:

$$I_{RX}(j_{RX}) = \log_2 \frac{P(i_{TX}|j_{RX})}{P(i_{TX})} \quad \text{(bits)} \tag{9.12}$$

where $P(i_{TX}|j_{RX})$ can be found from (but is not an element of) the end-to-end symbol transition matrix, $P(i_{RX}|j_{TX})$, described in section 7.3. (Note the interchanged position of *TX* and *RX* quantities.) Care must be taken not to confuse the conditional probabilities here (which relate to channel induced uncertainties, see section 7.2) with those in section 9.3 (which relate to source memory).

The reduction in information content of each symbol due to noise means that the effective entropy, H_{eff}, of the symbols on reception is less than their transmission entropy, H. The effective (or reception) entropy of a symbol j_{RX} is:

$$H_{eff} = \sum_j P(j_{RX})\bar{I}_{RX}(j_{RX}) \quad \text{(bit/symbol)}$$

where $P(j_{RX})$ is the (unconditional) symbol reception probability, and the bar denotes averaging over all possible transmitted symbols:

$$\bar{I}_{RX}(j_{RX}) = \sum_j P(j_{RX}) \sum_i P(i_{TX}|j_{RX}) \log_2 \frac{P(i_{TX}|j_{RX})}{P(i_{TX})} \tag{9.13}$$

The difference between the transmission entropy and the effective entropy is called the equivocation, E, i.e.:

$$H_{eff} = H - E \quad \text{(bit/symbol)} \tag{9.14}$$

The equivocation essentially accounts for the uncertainty with which the transmitted symbols can be assumed to be the same as the received symbols. The information associated with this uncertainty and, therefore, the definition of equivocation is:

$$E = \sum_{i,j}\sum P(i_{TX}, j_{RX}) \log_2 \frac{1}{P(i_{TX}|j_{RX})} \quad \text{(bit/symbol)} \tag{9.15}$$

The similarity between equations (9.15) and (9.3) is notable, the essential difference being that $P(i_{TX}|j_{RX})$ relates to the probability of symbol error. Multiplying by the joint probability $P(i_{TX}, j_{RX})$ and summing over all i, j simply averages the resulting spurious 'information' over all types of error. Using $P(i_{TX}, j_{RX}) = P(j_{RX})P(i_{TX}|j_{RX})$ equation (9.15) can be rewritten as:

$$E = \sum_j P(j_{RX}) \sum_i P(i_{TX}|j_{RX}) \log_2 \frac{1}{P(i_{TX}|j_{RX})} \tag{9.16}$$

where $P(j_{RX})$ is an (unconditional) symbol reception probability. An alternative interpretation of equivocation is as negative information added by noise.

EXAMPLE 9.2

Consider a 3 symbol source A, B, C with the following transition matrix:

	A_{TX}	B_{TX}	C_{TX}
A_{RX}	0.6	0.5	0
B_{RX}	0.2	0.5	0.333
C_{RX}	0.2	0	0.667

(Note that the elements of the transition matrix are the conditional probabilities $P(i_{RX}|j_{TX})$ and that columns and rows are labelled i_{TX} and i_{RX} only for extra clarity.) For the specific a priori (i.e. transmission) probabilities: $P(A) = 0.5$; $P(B) = 0.2$; $P(C) = 0.3$, find: (i) the (unconditional) symbol reception probabilities $P(i_{RX})$ from the conditional and a priori probabilities; (ii) the equivocation; (iii) the source entropy; and (iv) the effective entropy.

(i) The symbol reception probabilities are found as follows:

$$\begin{aligned} P(A_{RX}) &= P(A_{TX})\,P(A_{RX}|A_{TX}) + P(B_{TX})\,P(A_{RX}|B_{TX}) + P(C_{TX})\,P(A_{RX}|C_{TX}) \\ &= 0.5\times0.6 + 0.2\times0.5 + 0.3\times0 = 0.4 \\ P(B_{RX}) &= 0.5\times0.2 + 0.2\times0.5 + 0.3\times0.333 = 0.3 \\ P(C_{RX}) &= 0.5\times0.2 + 0.2\times0 + 0.3\times0.667 = 0.3 \end{aligned}$$

(ii) To find the equivocation from equation (9.16) we require the probabilities $P(i_{TX}|j_{RX})$ which can be found in turn from $P(i_{RX}|j_{TX})$ and $P(i_{RX})$ using Bayes's rule, equation (3.3):

$$P(A_{TX}|A_{RX}) = \frac{P(A_{TX}, A_{RX})}{P(A_{RX})} = \frac{P(A_{RX}|A_{TX})\,P(A_{TX})}{P(A_{RX})} = \frac{0.6\times0.5}{0.4} = 0.75$$

$$P(B_{TX}|A_{RX}) = \frac{P(B_{TX}, A_{RX})}{P(A_{RX})} = \frac{P(A_{RX}|B_{TX})P(B_{TX})}{P(A_{RX})} = \frac{0.5\times0.2}{0.4} = 0.25$$

$$P(C_{TX}|A_{RX}) = \frac{P(A_{RX}|C_{TX})P(C_{TX})}{P(A_{RX})} = \frac{0\times0.3}{0.4} = 0$$

$$P(A_{TX}|B_{RX}) = \frac{P(B_{RX}|A_{TX})P(A_{TX})}{P(B_{RX})} = \frac{0.2\times0.5}{0.3} = 0.333$$

$$P(B_{TX}|B_{RX}) = \frac{P(B_{RX}|B_{TX})P(B_{TX})}{P(B_{RX})} = \frac{0.5\times0.2}{0.3} = 0.333$$

$$P(C_{TX}|B_{RX}) = \frac{P(B_{RX}|C_{TX})\,P(C_{TX})}{P(B_{RX})} = \frac{0.333\times0.3}{0.3} = 0.333$$

$$P(A_{TX}|C_{RX}) = \frac{P(C_{RX}|A_{TX})P(A_{TX})}{P(C_{RX})} = \frac{0.2\times0.5}{0.3} = 0.333$$

$$P(B_{TX}|C_{RX}) = \frac{P(C_{RX}|B_{TX})P(B_{TX})}{P(C_{RX})} = \frac{0\times0.2}{0.3} = 0$$

$$P(C_{TX}|C_{RX}) = \frac{P(C_{RX}|C_{TX})\,P(C_{TX})}{P(C_{RX})} = \frac{0.667\times0.3}{0.3} = 0.667$$

Equation (9.16) now gives the equivocation:

$$E = P(A_{RX})\left[P(A_{TX}|A_{RX})\log_2 \frac{1}{P(A_{TX}|A_{RX})} + P(B_{TX}|A_{RX})\log_2 \frac{1}{P(B_{TX}|A_{RX})}\right]$$

$$+ P(A_{RX})\left[P(C_{TX}|A_{RX})\log_2 \frac{1}{P(C_{TX}|A_{RX})}\right] + P(B_{RX})\left[P(A_{TX}|B_{RX})\log_2 \frac{1}{P(A_{TX}|B_{RX})}\right]$$

$$+ P(B_{RX})\left[P(B_{TX}|B_{RX})\log_2 \frac{1}{P(B_{TX}|B_{RX})} + P(C_{TX}|B_{RX})\log_2 \frac{1}{P(C_{TX}|B_{RX})}\right]$$

$$+ P(C_{RX})\left[P(A_{TX}|C_{RX})\log_2 \frac{1}{P(A_{TX}|C_{RX})} + P(B_{TX}|C_{RX})\log_2 \frac{1}{P(B_{TX}|C_{RX})}\right]$$

$$+ P(C_{RX})\left[P(C_{TX}|C_{RX})\log_2 \frac{1}{P(C_{TX}|C_{RX})}\right]$$

$$= 0.4\left[0.75\log_2\left(\frac{1}{0.75}\right) + 0.25\log_2\left(\frac{1}{0.25}\right) + 0\right]$$

$$+ 0.3\left[0.333\log_2\left(\frac{1}{0.333}\right) + 0.333\log_2\left(\frac{1}{0.333}\right) + 0.333\log_2\left(\frac{1}{0.333}\right)\right]$$

$$+ 0.3\left[0.333\log_2\left(\frac{1}{0.333}\right) + 0 + 0.667\log_2\left(\frac{1}{0.667}\right)\right]$$

$$= 0.4\,[0.311 + 0.5] + 0.3\,[0.528 + 0.528 + 0.528] + 0.3\,[0.528 + 0.390]$$

$$= 1.08 \text{ bit/symbol}$$

(iii) The source entropy H is:

$$H = \sum_i P(i_{TX})\log_2 \frac{1}{P(i_{TX})}$$

$$= 0.5\log_2\left(\frac{1}{0.5}\right) + 0.2\log_2\left(\frac{1}{0.2}\right) + 0.3\log_2\left(\frac{1}{0.3}\right)$$

$$= 1.48 \text{ bit/symbol}$$

(iv) The effective entropy, equation (9.14), is therefore:

$$H_{eff} = H - E$$

$$= 1.48 - 1.08 = 0.4 \text{ bit/symbol}$$

9.5 Source coding

Source coding does not change or alter the source entropy, i.e. the average number of information bits per source symbol. In this sense source entropy is a fundamental property of the source. Source coding does, however, alter (usually increase) the entropy of the source coded symbols. It may also reduce fluctuations in the information rate from the source and avoid symbol 'surges' which could overload the channel when the message sequence contains many high probability (i.e. frequently occurring, low entropy) symbols.

9.5.1 Code efficiency

Recall the definition of entropy (equation (9.3)) for a source with statistically independent symbols:

$$H = \sum_{m=1}^{M} P(m) \log_2 \frac{1}{P(m)} \quad \text{(bit/symbol)} \tag{9.17}$$

The maximum possible entropy, $H_{\max}$, of this source would be realised if all symbols were equiprobable, $P(m) = 1/M$, i.e.:

$$H_{\max} = \log_2 M \quad \text{(bit/symbol)} \tag{9.18}$$

A code efficiency can therefore be defined as:

$$\eta_{code} = \frac{H}{H_{\max}} \times 100\% \tag{9.19}$$

If source symbols are coded into another symbol set, Figure 9.3, then the new code efficiency is given by equation (9.19) where H and $H_{\max}$ are the entropy and maximum possible entropy of this new symbol set.

If source symbols are coded into binary words then there is a useful alternative interpretation of η_{code}. For a set of symbols represented by binary code words with lengths l_m (binary) digits, an overall code length, L, can be defined as the average codeword length, i.e.:

$$L = \sum_{m=1}^{M} P(m)\, l_m \quad \text{(binary digits/symbol)} \tag{9.20}$$

The code efficiency can then be found from:

$$\eta_{code} = \frac{H}{L} \quad \text{(bit/binary digit)} \times 100\% \tag{9.21}$$

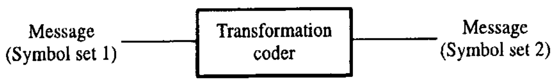

Figure 9.3 *Translating a message between different alphabets.*

Equations (9.21) and (9.19) are seen to be entirely consistent when it is remembered that the maximum information conveyable per L digit binary code word is given by:

$$H_{\max} \text{ (bit/symbol)} = L \text{ (bit/codeword)} \tag{9.22}$$

EXAMPLE 9.3
A scanner converts a black and white document, line-by-line into binary data for transmission. The scanner produces source data comprising symbols representing runs of up to six similar image pixel elements with the probabilities as shown below:

No. of consecutive pixels	1	2	3	4	5	6
Probability of occurrence	0.2	0.4	0.15	0.1	0.06	0.09

Determine the average length of a run (in pixels) and the corresponding effective information rate for this source when the scanner is traversing 1000 pixel/s.

$$H = \sum_{m=1}^{6} P(m) \log_2 \frac{1}{P(m)}$$

$$= 0.2 \times 2.32 + 0.4 \times 1.32 + 0.15 \times 2.74 + 0.1 \times 3.32 + 0.06 \times 4.06 + 0.09 \times 3.47$$

$$= 2.29 \quad \text{bit/symbol}$$

Average length:

$$L = \sum_{m=1}^{6} P(m)\, l_m = 0.2 + 0.8 + 0.45 + 0.4 + 0.3 + 0.54 = 2.69 \quad \text{pixels}$$

At 1000 pixel/s scan rate we generate 1000/2.69 = 372 symbol/s. Thus the source information rate is $2.29 \times 372 = 852$ bit/s.

We are generally interested in finding a more efficient code which represents the same information using fewer digits on average. This results in different lengths of codeword being used for different symbols. The problem with such variable length codes is in recognising the start and end of the symbols.

9.5.2 Decoding variable length codewords

The following properties need to be considered when attempting to decode variable length codewords:

(1) Unique decoding.
This is essential if the received message is to have only a single possible meaning. Consider an $M = 4$ symbol alphabet with symbols represented by binary digits as follows:

$A = 0$
$B = 01$
$C = 11$
$D = 00$

If we receive the code word 0011 it is not known whether the transmission was D, C or A, A, C. This example is not, therefore, uniquely decodable.

(2) Instantaneous decoding.

Consider now an $M = 4$ symbol alphabet, with the following binary representation:

$A = 0$
$B = 10$
$C = 110$
$D = 111$

This code can be instantaneously decoded using the decision tree shown in Figure 9.4 since no complete codeword is a prefix of a larger codeword. This is in contrast to the previous example where A is a prefix of both B and D. The latter example is also a 'comma code' as the symbol zero indicates the end of a codeword except for the all ones word whose length is known. Note that we are restricted in the number of available codewords with small numbers of bits to ensure we achieve the desired decoding properties.

Using the representation:

$A = 0$
$B = 01$
$C = 011$
$D = 111$

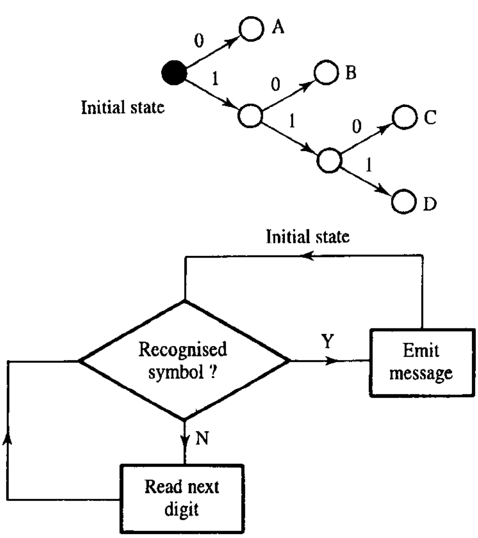

Figure 9.4 *Algorithm for decision tree decoding and example of practical code tree.*

the code is identical to the example just given but the bits are time reversed. It is thus still uniquely decodable but no longer instantaneous, since early codewords are now prefixes of later ones.

9.6 Variable length coding

Assume an $M = 8$ symbol source $A, \cdots, H$ having probabilities of symbol occurrence:

m	A	B	C	D	E	F	G	H
$P(m)$	0.1	0.18	0.4	0.05	0.06	0.1	0.07	0.04

The source entropy is given by:

$$H = \sum_m P(m) \log_2 \frac{1}{P(m)} = 2.55 \text{ bit/symbol} \tag{9.23}$$

and, at a symbol rate of 1 symbol/s, the information rate is 2.55 bit/s. The maximum entropy of an 8 symbol source is $\log_2 8 = 3$ bit/symbol and the source efficiency is therefore given by:

$$\eta_{source} = \frac{2.55}{3} \times 100\% = 85\% \tag{9.24}$$

If the symbols are each allocated 3 bits, comprising all the binary patterns between 000 and 111, the coding efficiency will remain unchanged at 85%.

Shannon-Fano coding [Blahut, 1987], in which we allocate the regularly used or highly probable messages fewer bits, as these are transmitted more often, is more efficient. The less probable messages can then be given the longer, less efficient bit patterns. This yields an improvement in efficiency compared with that before source coding was applied. The improvement is not as great, however, as that obtainable with another variable length coding scheme, namely Huffman coding, which is now described.

9.6.1 Huffman coding

The Huffman coding algorithm comprises two steps – reduction and splitting. These steps can be summarised by the following instructions:

(1) Reduction: List the symbols in descending order of probability. Reduce the two least probable symbols to one symbol with probability equal to their combined probability. Reorder in descending order of probability at each stage, Figure 9.5. Repeat the reduction step until only two symbols remain.

(2) Splitting: Assign 0 and 1 to the two final symbols and work backwards, Figure 9.6. Expand or lengthen the code to cope with each successive split and, at each stage, distinguish between the two split symbols by adding another 0 and 1 respectively to the codeword.

The result of Huffman encoding the symbols $A, \cdots, H$ in the previous example (Figures 9.5 and 9.6) is to allocate the symbols codewords as follows:

Symbol	*C*	*B*	*A*	*F*	*G*	*E*	*D*	*H*
Probability	0.40	0.18	0.10	0.10	0.07	0.06	0.05	0.04
Codeword	1	001	011	0000	0100	0101	00010	00011

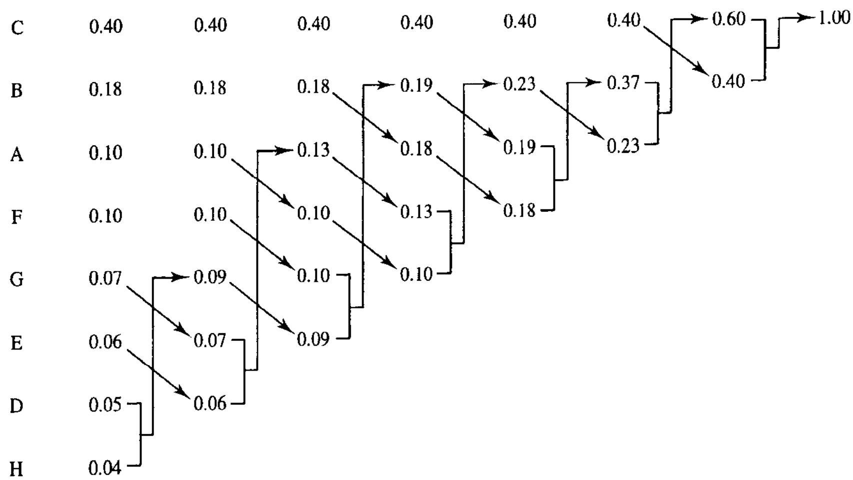

Figure 9.5 *Huffman coding of an 8-symbol alphabet – reduction step.*

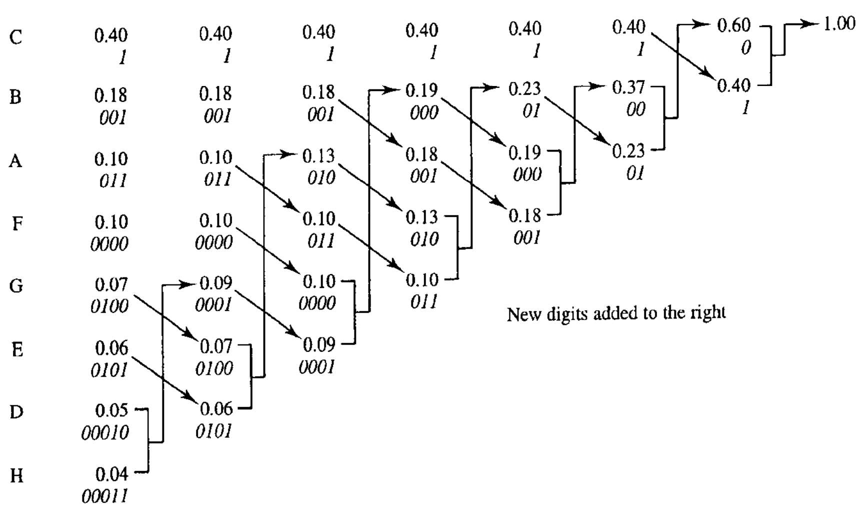

Figure 9.6 *Huffman coding – allocation of codewords to the 8 symbols.*

The code length is now given by equation (9.20) as:

$$L = 1(0.4) + 3(0.18 + 0.10) + 4(0.10 + 0.07 + 0.06) + 5(0.05 + 0.04) = 2.61 \quad (9.25)$$

and the code efficiency, given by equation (9.21), is:

$$\eta_{code} = \frac{H}{L} \times 100\% = \frac{2.55}{2.61} \times 100\% = 97.7\% \quad (9.26)$$

The 85% efficiency without coding would have been improved to 96.6% using Shannon-Fano coding but Huffman coding at 97.7% is even better. (The maximum efficiency is obtained when symbol probabilities are all negative, integer, powers of two, i.e. $1/2^n$.) Note that the Huffman codes are formulated to minimise the average code word length. They do not necessarily possess error detection properties but are uniquely, and instantaneously, decodable, as defined in section 9.5.2.

9.7 Source coding examples

An early example of source coding occurs in Morse where the most commonly occurring letter ‘e’ (dot) is allocated the shortest code (one bit) and less used consonants, such as ‘y’ (dash dot dash dash), are allocated 4 bits. A more recent example is facsimile (fax) transmission where an A4 page is scanned at 3.85 scan lines/mm in the vertical dimension with 1728 pixels across each scan line. If each pixel is then binary quantised into black or white this produces 2 Mbit of data per A4 page. If transmitted at 4.8 kbit/s over a telephone modem (see Chapter 11), this takes approximately 7 min/page for transmission.

In Group 3 fax runs of pixels of the same polarity are examined. (Certain run lengths are more common than others, Figure 9.7.) The black letters, or drawn lines, are generally not more than 10 pixels wide while large white areas are much more common. Huffman coding is employed in the ITU-T standard for Group 3 fax transmission, Figure 9.8, to allocate the shortest codes to the most common run lengths. The basic scheme is modified to include a unique end of line code for resynchronisation which determines

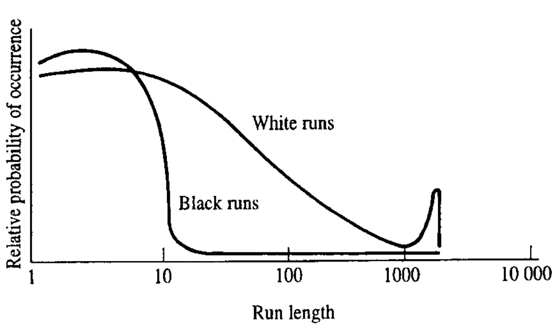

Figure 9.7 *Relative probability of occurrence of run lengths in scanned monochrome text.*

whether lines are received in error, as these should be always 1728 pixels apart.

In this application, Huffman coding improves the efficiency and reduces the time to transmit the page to less than 1 minute. Facsimile is now very significant with 25% of international telephone traffic representing fax transmissions between the 15 million terminals in use worldwide. Continued improvements in the Group 3 fax have made the transition to (the more advanced) Group 4 digital system unattractive. There is now an ITU-T V.17 data modem standard (see Section 11.6) which transmits fax signals at 14.4 kbit/s and which is very similar to the V.32 9.6 kbit/s standard used to achieve the Figure 11.55(c) result.

There is current interest in run length codes aimed at altering their properties to control the spectral shape of the coded signal. In particular it is desirable to introduce a null at DC, as discussed in section 6.4, and/or place nulls or minima at those locations where there is expected to be a null in the channel's frequency response. Such techniques are widely used to optimise the performance of optical and magnetic recording systems [Schouhammer-Immink]. Run length codes can also be modified to achieve some error control properties, as will be described in the next chapter.

9.7.1 Source coding for speech signals

Vocoders (voice coders) are simplified coding devices which extract, in an efficient way, the significant components in a speech waveform, exploiting speech redundancy, to achieve low bit rate (< 2.4 kbit/s) transmission. Speech basically comprises four *formants*, (Figure 9.9), and the vocoder analyses the input signal to find how the position, F_1, F_2, F_3 and F_4, and magnitudes, A_1, A_2, A_3 and A_4, of these formants vary with time. In this instance we fit a model to the input spectrum and transmit the model parameters rather than, for example, the ADPCM quantised error samples, $\varepsilon_q(kT_s)$, of

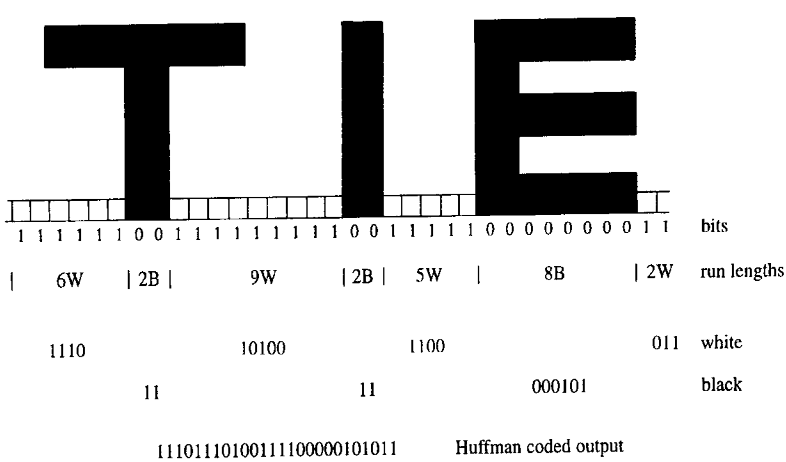

Figure 9.8 *Redundancy removal by source coding of printed character data (source: Pugh, 1991, reproduced with the permission of the IEE).*

section 5.8.3.

Two major vocoder designs exist at present, the channel vocoder and the linear predictive coder (LPC). The channel vocoder is basically a non-linearly spaced filterbank spectrum analyser with 19 distinct filters. The LPC vocoder is usually implemented as a cascade of linear prediction error filters which remove from the speech signal the components that can be predicted [Jayant and Noll, Gray and Markel] from its previous history by modelling the vocal tract as an all-pole filter. (See standard signals and systems texts such as [Jackson, Mulgrew and Grant] for a discussion of the properties of poles and zeros and their effect on system, and filter, frequency responses.) The LPC vocoder extends the DPCM system described in Chapter 5 to perform a full modelling of the speech production mechanism and removes the necessity for any quantised error sample transmissions.

In the LPC vocoder, Figure 9.10, the analyser and encoder normally process the signal in 20-ms frames and subsequently transmit the coarse spectral information via the filter coefficients. The residual error output from the parameter estimation operation is not transmitted. The error signal is used to provide an estimate of the input power level which is sent, along with the pitch information and a binary decision as to whether the input is voiced or unvoiced (Figure 9.10). The pitch information can be ascertained by using autocorrelation (see sections 2.6 and 3.3.3) or zero crossing techniques on either the input signal or the residual error signal, depending on the sophistication and cost of the vocoder implementation. The decoder and synthesiser apply the received filter coefficients to a synthesising filter which is excited with impulses at the pitch frequency if voiced, or by white noise if unvoiced. The excitation amplitude is controlled by the input power estimate information. This excitation with a synthetic signal reduces the transmission bit rate requirement but also reduces the speech quality. (Speech quality is assessed using the mean opinion score (MOS) scale (see section 5.7.3).)

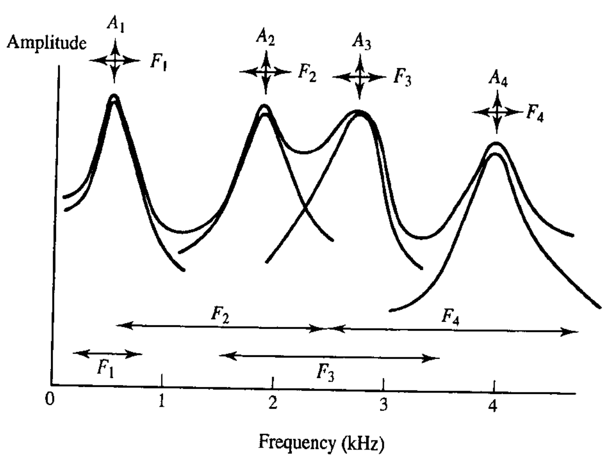

Figure 9.9 *Instantaneous spectral representation of speech as a set of four formant frequencies.*

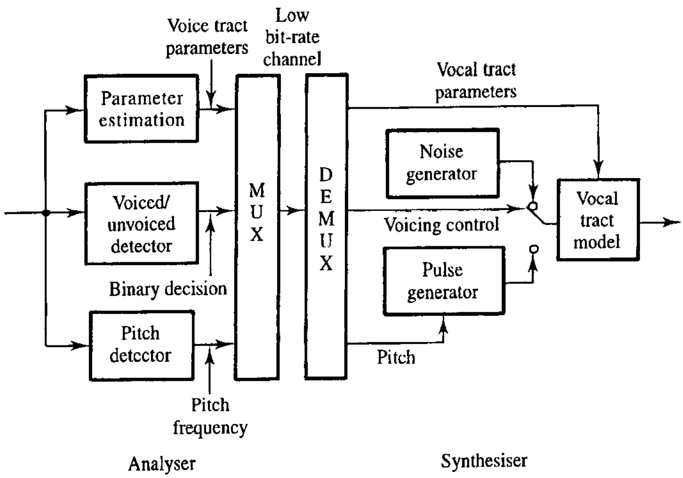

Figure 9.10 *Block diagram of the linear predictive vocoder.*

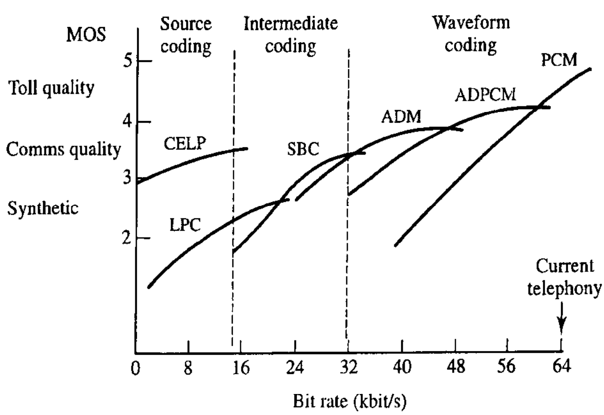

Figure 9.11 *Comparison of quality and transmission rate for various speech coding systems, using the mean opinion score (MOS) rating.*

With delays in the vocal tract of about 1 ms and typical speech sample rates of 8 to 10 kHz the number of predictor stages is normally in the range 8 to 12. 10 is the number adopted in the integrated NATO LPC vocoder standard (LPC-10, also US Federal Standard 1015) which transmits at a rate of 2.4 kbit/s and achieves a MOS of approximately 2, Figure 9.11 and Table 9.3, for a complexity which is 40 times that of PCM. The MOS score is low due to the synthesiser's excitation with noise or regenerated pitch information. This causes LPC to lose all background sounds and the speech to have a characteristic metallic sound. The basic spoken words, however, are still quite intelligible. These vocoders can be implemented as single chip DSP designs. In addition to vocoder applications, LPC synthesisers are used in several commercial speech systems, including the Texas Instruments speak and spell children's toy.

Much current research is in progress to improve the quality of vocoders. This includes studies of hybrid coder systems such as LPC excited from a codebook of possible signal vectors (CELP) or excited with multiple pulses (MPE) rather than simple pitch information as in Figure 9.10. US Federal Standard 1016 defines a 4.8 kbit/s CELP implementation for secure systems. In developing these systems low coder delay (not greater than 2 ms) is an important design goal.

CELP systems achieve a MOS of 3.2 at 4 kbit/s but they are 50 to 100 times as complex as the basic PCM coder. For higher quality transmission ITU-T G.728 defines a 16 kbit/s low delay CELP implementation. These two approaches, which also involve feedback to compare the synthesised speech with the original speech and minimise the difference, are being developed for mobile systems such as GSM (see section 15.4).

Table 9.3 *Rate, performance and complexity comparison for various speech coder designs.*

Class	*Technique*	*Bit rate (kbit/s)*	*MOS quality*	*Relative complexity*
Waveform coders	PCM G.711 ADPCM G.721 DM	64 32 16	4.3 4.1 3	1 10 0.3
Intermediate coder	SBC (2) G.722 (7 kHz)	64 – 48	4.3	30
Enhanced source coder	CELP/MPE for ½ rate GSM	4.8	3.2	50 – 100
Source coder	LPC-10 vocoder	2.4	2	40

9.7.2 High quality audio coders

Other audio coding techniques are based on frequency domain approaches where the 0.2 to 3.2 kHz speech bandwidth is split into 2 to 16 individual sub-bands by a filterbank or a discrete Fourier transform (DFT) (see section 13.5), Figure 9.12, to form a sub-band coder (SBC). Band splitting is used to exploit the fact that the individual bands do not all contain signals with the same energy. This permits the accuracy of the quantiser in the encoder to be reduced, in bands with low energy signals, saving on the coder transmission rate. While this was initially applied to achieve low bit rate intermediate quality speech at 16 kbit/s, the same technique is now used in the ITU-T G.722 0 to 7 kHz high quality audio coder which employs only two subbands to code this wideband signal at 64/56/48 kbit/s with a MOS between 3.7 and 4.3. This is a low complexity coder, implemented in one DSP microprocessor, which can operate at a P_e of 10^{-4} for ISDN teleconferencing and telephone loudspeaker applications. The MOS drops to 3.0 at a P_e of 10^{-3}. Compact disc player manufacturers are also investigating an extension of this technique with 32 channels within the 20 kHz hi-fi bandwidth. Simple 16-bit PCM at 44.1 ksample/s requires 700 kbit/s but the DFT based coder offers indistinguishable music quality at 88 kbit/s or only 2 bit/sample. This development of the ISO/MPEG standard is very important for the digital storage of broadcast quality signals, for example HDTV and digital audio (see Chapter 16).

Finally it has been shown that variants of these techniques can achieve high quality transmission at 64 kbit/s, which is also very significant for reducing the storage requirements of digital memory systems.

Similar coding techniques are applied to encode video signals but due to the large information content the bit rates are much higher. Video is usually encoded at 140 or 34 Mbit/s with the lower rate involving some of the prediction techniques discussed here. In video there is also redundancy in the vertical and horizontal dimensions, and hence spatial Fourier, and other, transformation techniques [Mulgrew and Grant] are attractive for reducing the bit rate requirements in, for example, video telephony applications (see Chapter 16).

9.7.3 String coding

Lempel-Ziv, or its common implementation, the LZW algorithm [Welch, 1984], trains on data to identify commonly occurring strings of input characters and builds up a table of these strings. Each string in the table is then allocated a unique 12 bit code. Commonly encountered word strings, e.g. 000, 0000, or 'High Street' experience compression when they are allocated their 12-bit codes, especially for long (> 10 character) strings. Thus the scheme relies on redundancy in the character occurrence, i.e. individual character string repetitions, but it does not exploit, in any way, positional redundancy.

LZW compression operates typically on blocks of 10,000 to 30,000 symbols and achieves little compression during the adaptive phase when the table is being constructed. It uses simplified logic, which operates at three clock cycles per symbol. After 2,000 to 5,000 words have been processed typical compression is 1.8:1 on text, 1.5:1 on object files and > 2:1 on data or program source code. There is no compression however on

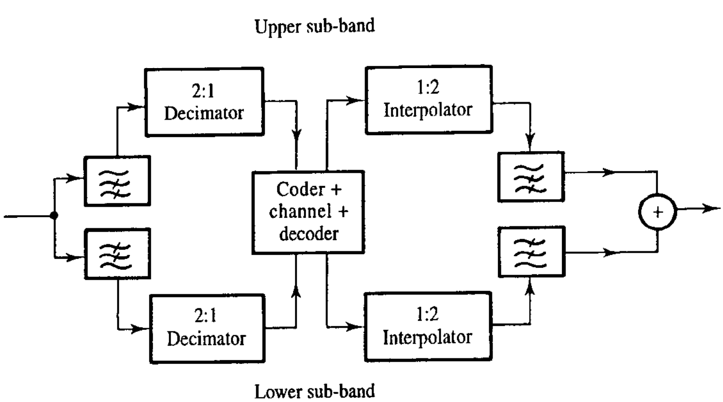

Figure 9.12 *Sub-band (speech) coder with two, equal width, sub-bands and separate encoding of sub-band signals after sample rate reduction.*

floating point arrays which present noise-like inputs.

9.8 Summary

The information content of a symbol is defined as the negative logarithm of the symbol's selection probability. If 2 is chosen as the base of the logarithm then the unit of information is the bit. The entropy of an information source is the average information per selected, or transmitted, symbol. The entropy of a binary source with statistically independent, equiprobable, symbols is 1 bit/binary digit. Statistical dependence of symbols, and unequal selection probabilities, both reduce the entropy of a source below its maximum value. Redundancy is the difference between a source's maximum possible entropy and its actual entropy. Noise at the receiver makes the decision process uncertain and thus reduces the information content of the received symbol stream. The information content of the symbols after detection is called the effective entropy and is equal to the entropy on transmission minus the equivocation introduced by the noise. In this sense noise may be interpreted as negative information added to the transmitted symbol stream.

Source coding removes redundancy, making transmission and/or storage of information more efficient. Source codes must, in general, be uniquely decodable. Source coding of discrete information sources is, normally, also loss-less, i.e. the coding algorithm, in the absence of symbol errors, is precisely reversible. Codes may, in addition, be instantaneously decodable, i.e. contain no code words which form a prefix of any other code words. The Huffman code, used for Group 3 fax transmission, is an example of a unique, loss-less, instantaneous code.

Vocoders use models of the physical mechanism of speech production to transmit intelligible, but low bit rate, speech signals. In essence it is the model parameters, in the form of adaptive filter coefficients, which are transmitted instead of the voice signals themselves. The adaptive filters are excited at the receiver by a pulse generator, of the correct frequency (pitch), for voiced sounds or white noise for unvoiced sounds.

Sub-band coding splits the transmitted signal into two or more frequency sub-bands using hardware or (DFT) software. This allows the quantisation accuracy for each sub-band to reflect the importance of that band as measured by the signal power it contains. Other transform coding techniques can be used to encode speech and images. Transform coding is particularly significant in two-dimensional image transmission as it minimises redundancy which takes the form of correlation in intensity or luminance values between closely spaced pixels. Vocoder and transform coding techniques are lossy in that there is an inevitable (but often small) loss in the information content of the encoded data.

String coding algorithms are particularly efficient at compressing character-based data but coding times are long since commonly occurring strings must be initially searched for in the data. String coding is used for compression of computer files in order to speed up transport between machines and save disc space.

The next chapter shows how the opposite technique to source coding, i.e. increasing redundancy, can be employed to detect, and sometimes correct, errors in data which may occur during its transmission or storage.

9.9 Problems

9.1. (a) Consider a source having an $M = 3$ symbol alphabet where $P(x_1) = ½$; $P(x_2) = P(x_3) = ¼$ and symbols are statistically independent. Calculate the information conveyed by the receipt of the symbol x_1. Repeat for x_2 and x_3. [$I_{x_1} = 1$ bit, $I_{x_2} = I_{x_3} = 2$ bits]
(b) Consider a source whose, statistically independent, symbols consist of all possible binary sequences of length k. Assume all symbols are equiprobable. How much information is conveyed on receipt of any symbol. [k bits]
(c) Determine the information conveyed by the specific message $x_1x_3x_2x_1$ when it emanates from each of the following, statistically independent, symbol sources: (i) $M = 4$; $P(x_1) = ½$, $P(x_2) = ¼$, $P(x_3) = P(x_4) = 1/8$, [7 bits] (ii) $M = 4$; $P(x_1) = P(x_2) = P(x_3) = P(x_4) = ¼$. [8 bits]

9.2. (a) Calculate the entropy of the source in Problem 9.1(a). [1½ bit/symbol]
(b) Calculate the entropy of the sources in Problem 9.1(c). [1¾ bit/symbol, 2 bit/symbol)]
(c) What is the maximum entropy of an 8 symbol source and under what conditions is this situation achieved? What are the entropy and redundancy if $P(x_1) = ½$, $P(x_i) = 1/8$ for $i = 2, 3, 4$ and $P(x_i) = 1/32$ for $i = 5, 6, 7, 8$? [3 bit/symbol, 2.25 bit/symbol]

9.3. Find the entropy, redundancy and code efficiency of a three symbol source A, B, C, if the following statistical dependence exists between symbols. There is a 20% chance of each symbol being succeeded by the next symbol in the *cyclical* sequence A B C and a 30% chance of each symbol being succeeded by the previous symbol in this sequence? [1.485 bit/symbol; 0.1 bit/symbol; 93.7%]

9.4. Show that the number of redundant symbols per bit of information transmitted by an M-symbol source with code efficiency η_{code} is given by $(1 - \eta_{\text{code}})/(\eta_{\text{code}} \log_2 M)$ symbol/bit.

9.5. Estimate the maximum information content of a black and white television picture with 625 lines and an aspect ratio of 4/3. Assume that 10 brightness values can be distinguished and that the picture resolution is the same along a horizontal line as along a vertical line. What maximum data rate does a picture rate of 25 picture/s correspond to and what, approximately, must be the bandwidth of the (uncoded and unmodulated) video signal if it is transmitted using binary symbols? (If necessary you should consult Chapter 16 to obtain TV scanning format information.) [3.322 bit/symbol and 1.73 Mbit/picture; 43.25 Mbit/s; 21.62 MHz]

9.6. Calculate the loss in information due to noise, per transmitted digit, if a random binary signal is transmitted through a channel, which adds zero mean Gaussian noise, with an average signal-to-noise ratio of: (a) 0 dB; (b) 5 dB; (c) 10 dB. [0.6311; 0.2307; 0.0094 bit/binit]

9.7. An information source contains 100 different, statistically independent, equiprobable symbols. Find the maximum code efficiency, if, for transmission, all the symbols are represented by binary code words of equal length. [7 bit words and 94.9%]

9.8. (a) Apply Huffman's algorithm to deduce an optimal code for transmitting the source defined in Problem 9.1(c)(i) over a binary channel. Is your code unique?
(b) Define the efficiency of a code and determine the efficiency of the code devised in part(a).
(c) Construct another code for the source of part (a) and assign equal length binary words irrespective of the occurrence probability of the symbols. Calculate the efficiency of this source. [(a) 0, 10, 110, 111, Yes; (b) 100%; (c) 87.5%]