UESTC4004
Digital Communications

Instructors: Prof. Muhammad Imran and Sajjad Hussain

This week: Sajjad Hussain sajjad.hussain@glasgow.ac.uk

Lecture Preview

- Understanding the concept of Coding
- Source Coding
  - Shanon-Fano Coding
  - Huffman Coding
- Channel Coding

Coding
Definitions

•Coding (or encoding) is a transformation procedure operating on the input signal prior to its entry into the communication channel. This procedure adapts the input signal to the communication system and improves its efficiency.

•In other words, coding is a procedure for associating words constructed from a finite alphabet of a language (e.g. a natural language) with given words of another language (encoding language) in one-to-one manner.

•Decoding is the inverse operation: restoration of words from the initial language.

Coding

Source Coding/Compression

•If the entropy of the source is less than the average word length of the original alphabet, this means that the original coding is redundant and that the original information may be compressed by an efficient source coding algorithm. The source coding efficiency is defined as the ratio of the source entropy to the average word length.

$$\text{Efficiency} = \frac{H(X)}{\bar{L}}$$

•Entropy: The average amount of information per message is referred to as the source entropy.

•The main idea behind the compression is to create such a code, for which the average length of the encoding vector (word) will not exceed the source entropy. In general, this means the codes that are used for compression are not uniform, i.e., the words have different number of bits.

Coding

Source Coding

•The most common source coding algorithms are the Shannon-Fano and Huffman used for discrete memoryless information sources.

•Shannon-Fano source coding is the first established and widely used coding method. It was invented simultaneously and independently of each other by C. Shannon and R. Fano in 1948.

•Huffman source coding constructs binary codes with minimum redundancy (i.e., maximum efficiency) for a set of discrete messages. This algorithm, which results in an optimum code, was invented by David A. Huffman in 1952, and it is still the main lossless compression method.

•Both coding algorithms belong to the category of lossless coding as the source data can be perfectly reconstructed from the compressed data.

•Lossy coding is when compression results a minimum tolerable distortion.

Coding

Shannon-Fano Source Coding

•We consider the original messages that are to be transmitted with their corresponding probabilities, i.e.,

$$[X] = [x_1, x_2, ..., x_n] \qquad [P] = [p_1, p_2, ..., p_n]$$

•Our task is to associate a sequence C_k of binary numbers of unspecified length n_k to each message x_k such that:

•No sequences can be obtained by other shorter sequences by just adding more binary digits to the shorter sequence (prefix property).

•The transmission of the coded message is “reasonably” efficient, that is, 1 and 0 appear independently and with “almost” equal probabilities. This ensures transmission of “almost” 1 bit of information per digit of the coded messages.

Coding

Shannon-Fano Source Coding

•A rule of thumb: more frequent messages are coded by a shorter sequence and less frequent messages are coded by a longer sequence.

•The messages of the information source must be arranged in order from most probable to least probable.

•Then the initial set of messages must be divided into two subsets whose total probabilities are as close as possible to being equal. All symbols then have the first digits of their codes assigned; symbols in the first set receive "0" and symbols in the second set receive "1".

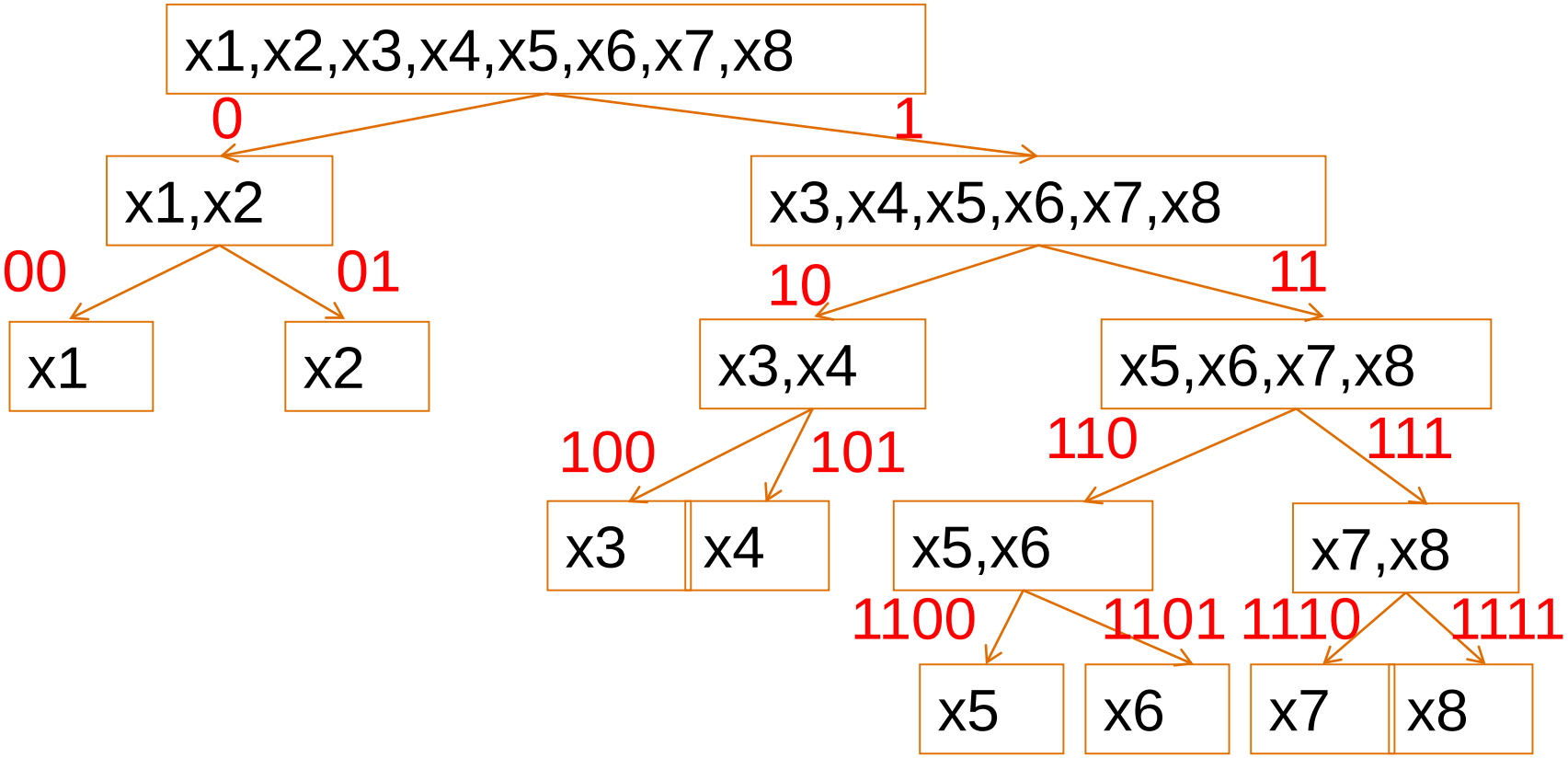
•The same process is repeated on those subsets, to determine the next digits of a symbol.

•When a subset has been reduced to one symbol, this means the symbol's code is complete.

Coding

Shannon-Fano Source Coding-Example

Message	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
Probability	0.25	0.25	0.125	0.125	0.0625	0.0625	0.0625	0.0625



Coding

Shannon-Fano Source Coding-Example

Message	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
Probability	0.25	0.25	0.125	0.125	0.0625	0.0625	0.0625	0.0625
Encoding sequence	00	01	100	101	1100	1101	1110	1111

Source entropy:

$$H(X) = -\sum P\{x_i\} \log P\{x_i\} = -\left[2 \cdot \left(\frac{1}{4}\log_2 \frac{1}{4}\right) + 2 \cdot \left(\frac{1}{8}\log_2 \frac{1}{8}\right) + 4 \cdot \left(\frac{1}{16}\log_2 \frac{1}{16}\right)\right] = 2.75$$

Average length of the encoding vector:

$$\bar{L} = \sum P\{x_i\} n_i = \left[2 \cdot \left(\frac{1}{4} \cdot 2\right) + 2 \cdot \left(\frac{1}{8} \cdot 3\right) + 4 \cdot \left(\frac{1}{16} \cdot 4\right)\right] = 2.75$$

- The efficiency for this example is 100%. We also see that the direct uniform plain binary encoding (3 bits/symbol) is redundant.

Coding

Shannon-Fano Source Coding-Properties

•The Shannon-Fano encoding is the most efficient when the probability of the occurrence of each message x_k is of the form $P\{x_k\} = 2^{-n_k}$.

•In this case, the efficiency is 100%, i.e.,

$$H(X) = -\sum_{k=1}^{N} P\{x_k\} \log_2 P\{x_k\} = \sum_{k=1}^{N} P\{x_k\} n_k = \bar{L}$$

•The prefix property always holds.

•The Shannon-Fano code is not unique because it depends on the partitioning of the input set of messages, which, in turn, is not unique.

•If it is not possible to partition the source messages with equal probabilities, the Shannon-Fano code may not be an optimum code, not leading to the lowest possible average length of the encoding sequence.

Coding

Huffman Source Coding

•For an optimum source coding, the longer encoding sequence should correspond to a message (letter) with lower probability:

$$P\left\{x_1\right\} \geq P\left\{x_2\right\} \geq \ldots \geq P\left\{x_N\right\} \leftrightarrow L\left\{x_1\right\} \leq L\left\{x_2\right\} \leq \ldots \leq L\left\{x_N\right\}$$

•For an optimum source coding, it is necessary that the length of the last two code words to be equal, otherwise the average length of the encoding vector will be unnecessarily increased. Thus,

$$L\left(x_{N-1}\right) = L\left(x_N\right)$$

•For an optimum source coding, it is necessary that the last two encoding vectors are identical except for the last digits.

Coding

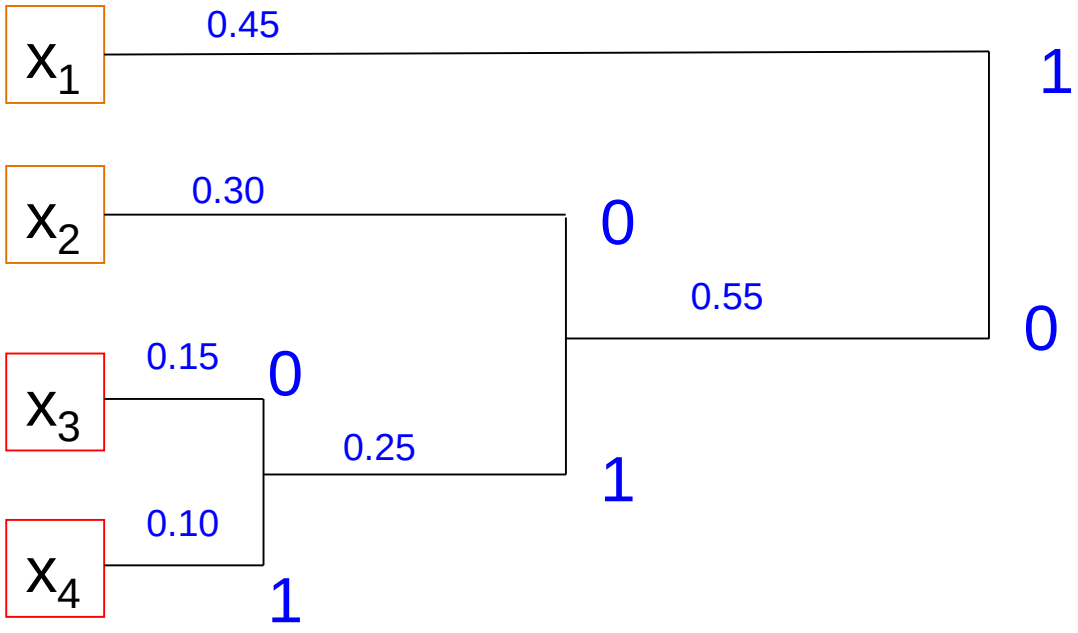
Huffman Source Coding

- The messages the information source must be arranged in order from most probable to least probable.
- Two least probable messages (the last two messages) are merged into the composite message with a probability equal to the sum of their probabilities. This new message must be inserted into the sequence of the original messages instead of its “parents”, carrying its probability.
- The previous step must be repeated until the last remaining two messages will compose a message.
- This process is utilized for constructing a binary tree, i.e., the Huffman tree.
- The encoding sequence for each message is obtained by crossing all possible paths in the tree.

Coding

Huffman Source Coding-Example

Message	x_1	x_2	x_3	x_4
Probability	0.45	0.30	0.15	0.10



Encoding sequences: $x_1 \rightarrow (1)$, $x_2 \rightarrow (00)$, $x_3 \rightarrow (010)$, $x_4 \rightarrow (011)$

Coding

Huffman Source Coding-Example

Message	x_1	x_2	x_3	x_4
Probability	0.45	0.30	0.15	0.10
Encoding sequence	1	00	010	011

Source entropy:

$$H(X) = -\sum P\{x_i\} \log P\{x_i\} =$$

$$-\left[0.45 \cdot \log 0.45 + 0.30 \cdot \log 0.30 + 0.15 \cdot \log 0.15 + 0.10 \cdot \log 0.10\right] = 1.782$$

Average length of the encoding vector:

$$\bar{L} = \sum P\{x_i\} n_i = 0.45 \cdot 1 + 0.30 \cdot 2 + 0.15 \cdot 3 + 0.10 \cdot 3 = 1.8$$

The efficiency for this example is 99%. We also see that the direct uniform plain binary encoding (2 bits/symbol) is redundant.

Coding

Huffman Source Coding-Properties

•Huffman source coding is the most efficient when the probability of the occurrence of each message does not follow any particular pattern, especially if the source messages cannot be partitioned with equal probabilities.

•Generally, Huffman code is the optimum lossless code with variable length.

•The prefix property always holds.

•Huffman code is not unique (as Shannon-Fano too) because in some cases they can arise equal probabilities for the Huffman tree branches giving several alternatives of selection.

Coding

Channel Coding

•Channel coding is process of adding some redundancy to the binary sequence under transmission (which has resulted from the source coding process), in order to improve its reliability and make it more robust against errors.

•By adding redundancy, it means to add additional bits to the binary sequence. This makes possible the error detection and error correction.

•An error is the replacement of one letter in a word by another one.

•Error detection means detection of the fact that the error has occurred without the exact identification of where.

•Error correction means the complete restoration of a word, which was originally sent, but was contaminated with errors during the transmission.

Coding

Channel Coding

- There are two different schemes for dealing with errors.
- A) Error detection and then retransmission, i.e., using automatic repeat request (ARQ) systems,
- B) Error detection and then error correction without retransmission, i.e., using forward error correction (FEC).
- The ARQ systems use the parity bits in order to detect errors. The parity bit is a single bit (1 or 0) appended to the end of the data word in order the number of 1s in the new word to be even (for even parity) or odd (for odd parity).
- The two main categories of FEC coding schemes are the block coding, i.e., a block of bits is processed as a whole to create the new sequence) and the convolutional coding, i.e, a real time process of the incoming sequence for creating another real time output sequence.

Coding
Channel Coding

•ARQ systems are the following:

•Stop and Wait ARQ: For each transmission, the transmitter waits for an acknowledgment of correct reception from the receiver. If the message received with errors a negative acknowledgment is sent and retransmission is ordered.

•Go and back ARQ: The transmitter transmits continuously until a negative acknowledgment is sent from the receiver in case of errors. Then all the messages are retransmitted starting from that when the error occurred.

•Selective ARQ: By increasing the complexity with providing buffers at both the transmitter and receiver, the receiver informs the transmitter about a specific message with errors and the transmitter retransmits this specific message. The receiver then stores the recovered message in the correct place in its buffer.

Coding

Channel Coding

•Block coding is characterized by the following metrics:

•Code rate *R* is the ratio of the number of input bits *k* to the output bits *n.* This is called a *(n, k)* code. Thus,

$$R = k / n$$

•Hamming distance *p* is the least minimum difference in bits between all pairs of coded words. The greater the Hamming distance, the more dissimilar the coded words, thus the better the chance to detect and correct errors.

•For example consider the two bit streams 0110100 and 1011100. The Hamming distance among them is *p=3.*

Coding

Channel Coding

•In block coding, the additional bits are selected in order to make the coded words as much as possible dissimilar.

•A block code with a Hamming distance *p* can detect up to *p-1* errors and correct *(p-1)/2* errors.

•. A very classical block code is the (4,7) Hamming code. This code considers 4 input bits and adds 3 more to create the output coded sequences. The Hamming distance for this code is *p=3.* Thus, it detects up to 2 errors and corrects 1 error.

Information Theory

Questions

1. Consider the example of slide 9. Compress with the Huffman source coding. Compare the results.
2. Consider the example of slide 14. Compress with the Shannon-Fano coding. Compare the results.
3. What is the Hamming distance for a block code that corrects 3 errors? How many errors can detect this code?
4. What are the types of automatic repeat request (ARQ) systems?
5. Determine the parity bits for the following data words assuming odd parity (*1001, 0110, 0011, 1100)*.
6. Determine the parity bits for the following data words assuming even parity (0111, 1110, 1000, 0000).