

Communication Background Notes

Xuechao Wei

June 5, 2014

1 Handout

1.1 MIT 6.450

1.1.1 Chapter-1 Introduction

Digital communication systems, are communication systems that use such a digital sequence as an interface between the source and the channel input (and similarly between the channel output and final destination).

The reasons why communication systems now usually contain a **binary interface** between source and channel:

- Digital hardware has become so cheap, reliable, and miniaturized, that digital interfaces are eminently practical.
- A standardized binary interface between source and channel simplifies implementation and understanding. Source coding/decoding can be done independently of the channel.
- A standardized binary interface between source and channel simplifies networking, which now reduces to sending binary sequences through the network.
- One of the most important of Shannon's information theoretic results is that if a source can be transmitted over a channel in any way at all, it can be transmitted using a binary interface between source and channel. This is known as the source/channel separation theorem.

Source coding: The most straightforward approach to analog source coding is called analog to digital (A/D) conversion.

Channel encoding: modulation refers to the process of combining a lowpass signal waveform with a high frequency sinusoid, then placing the signal waveform in a frequency band appropriate for transmission and regulatory requirements. An elementary waveform $p(t)$ is used to satisfy frequency constraints and reliably detect the binary digits from the received waveform in the presence of noise and intersymbol interference.

Error correction is usually separated from channel encoder as a individual layer because frequently the error probability incurred with simple modulation and demodulation techniques is too high.

1.1.2 Chapter-2 Coding for Discrete Sources

The motivation for using variable-length encoding on discrete sources is the intuition that data compression can be achieved by mapping more probable symbols into shorter bit sequences, and less likely symbols into longer bit sequences.

The Kraft inequality for prefix-free codes

Every prefix-free code for an alphabet $\chi = \{a_1, \dots, a_M\}$ with codeword lengths $\{l(a_j), 1 \leq j \leq M\}$ satisfies

$$\sum_{j=1}^M 2^{-l(a_j)} \leq 1. \quad (1.1)$$

Conversely, if 1.1 is satisfied, then a prefix-free code with lengths $\{l(a_j), 1 \leq j \leq M\}$ exists. Moreover, every full prefix-free code satisfies 1.1 with equality and every non-full prefix-free code satisfies it with strict inequality.

2 remarks on the Kraft inequality:

+ Just because a code has lengths that satisfy 1.1, it does not follow that the code is prefix-free, or even uniquely decodable.

+ The kraft inequality also holds for all uniquely-decodable codes—i.e., there exists a uniquely-decodable code with codeword lengths $\{l(a_j), 1 \leq j \leq M\}$ if and only if 1.1 holds. This will imply that if a uniquely-decodable code exists with a certain set of codeword lengths, then a prefix-free code exists with the same set of lengths. So why use any code other than a prefix-free code?

Minimum \bar{L} for prefix-free codes

Assume X_1, X_2, \dots are selected from χ using the same probability mass function (pmf) $\{p_X(a_1), \dots, p_X(a_M)\}$, and X_k is statically independent of the previous outputs X_1, \dots, X_{k-1} . This kind of source output is called *discrete memoryless source* (DMS) which is a toy source.

Suppose a set of lengths $l(a_1, \dots, l(a_M))$ (subject to the Kraft inequality) is chosen for encoding each symbol into a prefix-free codeword. Define $L(X)$ as a random variable representing the codeword length for the randomly selected source symbol. The expected value of L for the given code is then given by

$$\bar{L} = E[L] = \sum_{j=1}^M l(a_j) p_X(a_j).$$

We want to find \bar{L}_{min} , which is defined as the minimum value of \bar{L} over all sets of codeword lengths satisfying the Kraft inequality.

There are mainly two methods to achieve this: the Lagrange multiplier solution and the Huffman's algorithm.

1.2 MIT 6.451

References