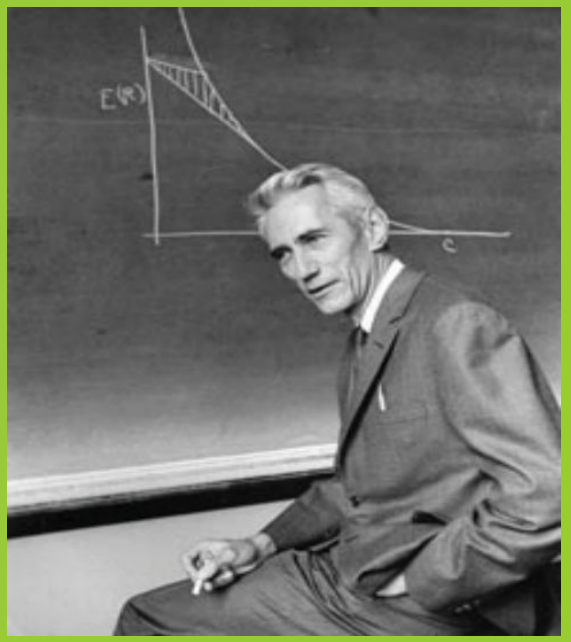
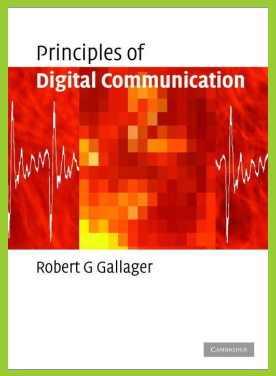
Source Coding
Brief Introduction



Ref:

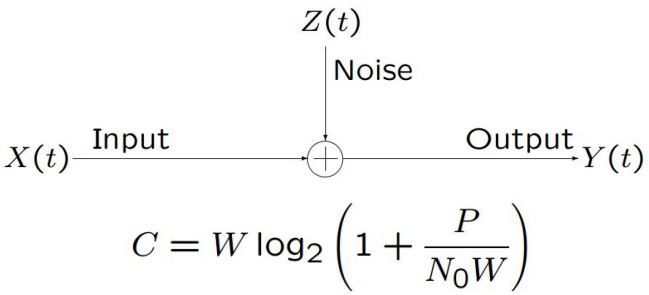
Principles of Digital Communication, Robert G. Gallager
MIT OpenCourse

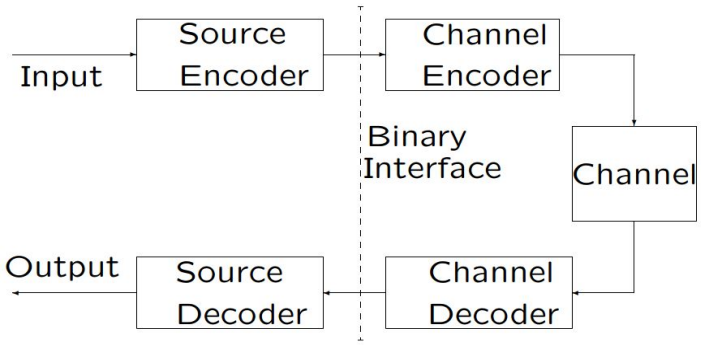


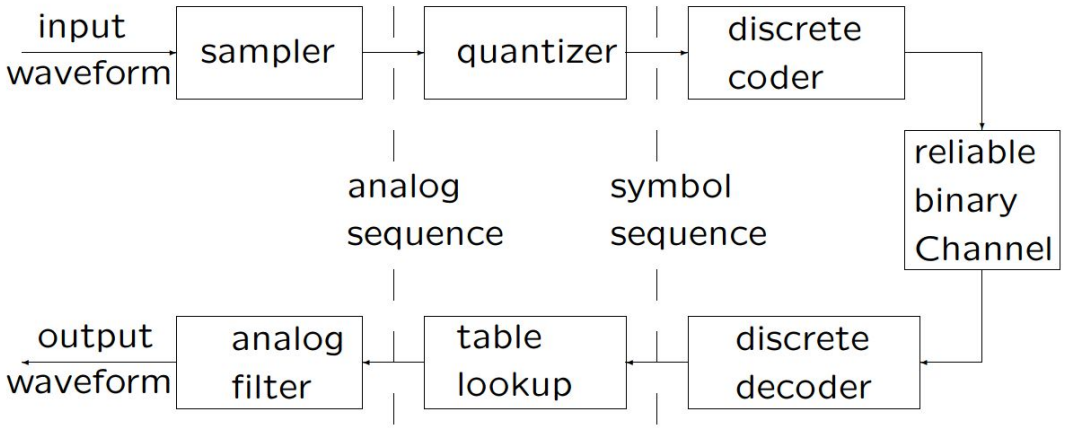
Written by Zhang Long
Long.A.Zhang@tieto.com
Dec 2nd, 2016

1. Discrete Sources Coding

1.1 Layering of source coding







Layering of source coding

1.2 What is Source Coding

Source Coding has 3 parts
- Analog waveform to analog sequence
- Quantizer (sequence to symbols)
- Symbols to bits

Binary coding: Mapping source **symbols** to **binary** digits (**alphabet**)

Unicode, ASCII, JPEG, GIF, AVI, MPEG, H.265, QuickTime (vector quantization) etc.

Random Symbol, Stochastic Process

- Standard Binary interface separates source and channel coding
- Multiplex data on high speed channels.
- Digital data can be “cleaned up” at each link in a network.
- Can separate problems of waveform sampling from quantization to discrete source coding.

1.3 Fixed-length Codes for discrete sources

For an alphabet size of M, require $2^L \geq M$.

To avoid wasting bits, choose L as smallest integer satisfying $2^L \geq M$, i.e.,,

$$\log_2 M \leq L < \log_2 M + 1; \quad L = \lceil \log_2 M \rceil$$

Segment source sequence into blocks of n; encode n symbols as a unit.

There are M^n n-tuples of source letters.

Fixed-length source coding on n-tuples requires

$$L = \lceil \log_2 M^n \rceil$$

Rate $\overline{L} = L/n$ bits per source symbol (bpss)

$$\log_2 M \leq \overline{L} < \log_2 M + \frac{1}{n}$$

1.4 Variable-length Code for discrete sources

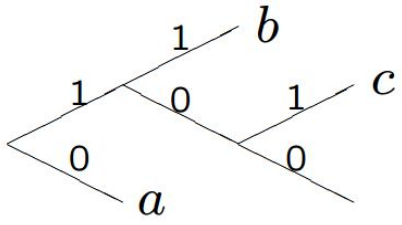
Motivation: Probable symbols should have shorter codewords than improbable to reduce bpss.

The major property that is usually required from any variable-length code:

- Unique Decodability
- Initial Synchronization
- Buffering

A code is ***prefix-free*** if no codeword is a prefix of any other codeword, are sometimes called ***instantaneous codes***.

Why prefix-free code?
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.



$a \rightarrow 0$
$b \rightarrow 11$
$c \rightarrow 101$

1.5 The Kraft Inequality

The Kraft inequality is a test on the existence of prefix-free codes with a given set of codeword lengths $\{l(x), x \in \mathcal{X}\}$.

Theorem (Kraft): Every prefix-free code for an alphabet $\mathcal{X}$ with codeword lengths $\{l(x), x \in \mathcal{X}\}$ satisfies

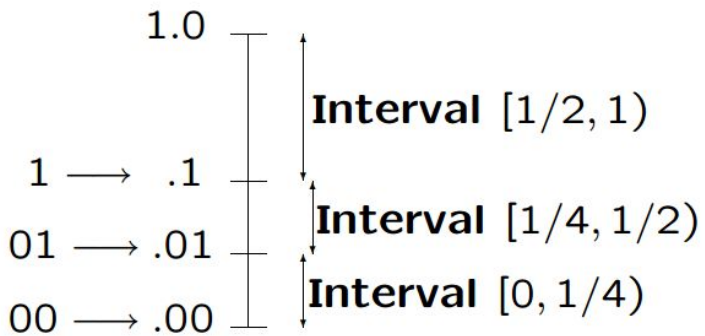
$$\sum_{x \in \mathcal{X}} 2^{-l(x)} \leq 1 \qquad (1)$$

Conversely, if (1), then a prefix-free code with lengths $\{l(x)\}$ exists.

Moreover, a prefix-free code is full iff (1) is satisfied with equality.

Represent binary codeword $y_1, y_2, \ldots, y_m$ **as**

$$.y_1 y_2 \cdots y_m = y_1/2 + y_2/4 + \cdots + y_m 2^{-m}$$



Here, in the same way, the base 2 expansion $.y_1y_2...y_n$ is viewed as 'covering' the interval

$$[\sum_{m=1}^{l} y_m 2^{-m},\ \sum_{m=1}^{l} y_m 2^{-m} + 2^{-l}).$$

A code has lengths that satisfy Kraft Inequality, it does not follow that the code is prefix-free, or even uniquely decodable.It determines which sets of codeword lengths are possible for prefix-free codes.

What set of codeword lengths can be used to *minimize* the expected length of a prefix-free code?

1.6 Discrete Memory Sources(DMS)

- The source output is an unending sequence $X_1, X_2, X_3, \ldots$, of randomly selected letters from a finite set X, called the source alphabet.
- Each source output $X_1, X_2, \ldots$ is selected from X using a common probability measure.
- Each source output X_k is statistically independent of the other source outputs $X_1, \ldots, X_{k-1}, X_{k+1}, \ldots$.

Let $l(x)$ be the length of the codeword for letter $x \in X$.
Then $L(X)$ is a random variable (rv) where $L(X) = l(x)$ for $X = x$.
Thus $L(X) = l(x)$ with probability $p_X(x)$

$$E(L) = \overline{L} = \sum_x p_X(x) l(x)$$

Thus•L is the number of encoder output bits per source symbol.

Choose integers $\{l(x)\}$ subject to Kraft to **minimize•L**

Let $\mathcal{X} = \{1, 2, \ldots, M\}$ with pmf $p_1, \ldots, p_M$.
Denote the unknown lengths by $l_1, \ldots, l_M$.

$$\overline{L}_{min} = \min_{l_1,\ldots,l_M : \sum 2^{-l_i} \le 1} \left\{ \sum_{i=1}^{M} p_i l_i \right\}$$

Note first that the minimum of $\sum_j l_j p_j$ subject to $\sum_j 2^{-l_j} \le 1$ must occur when the constraint is satisfied with equality, for otherwise, one of the l_j could be reduced, thus reducing $\sum_j p_j l_j$ without violating the constraint. Thus the problem is to minimize $\sum_j p_j l_j$ subject to $\sum_j 2^{-l_j} = 1$.

Minimize Lagrangian: $\sum_i (p_i l_i + \lambda 2^{-l_i})$.

$$\frac{\partial \sum_i (p_i l_i + \lambda 2^{-l_i})}{\partial l_i} = p_i - \lambda (\ln 2) 2^{-l_i} = 0$$

Thus $2^{-l_j} = p_j/(\lambda \ln 2)$. Since $\sum_j p_j = 1$, λ must be equal to $1/\ln 2$ in order to satisfy the constraint $\sum_j 2^{-l_j} = 1$. Then $2^{-l_j} = p_j$, or equivalently $l_j = -\log p_j$.

$$\overline{L}_{\min}(\text{noninteger}) = -\sum_{j=1}^{M} p_j \log p_j = \mathbf{H}(X)$$

1.7 Entropy Bounds

The entropy H(*X*) of the rv *X* is the minimum number of binary digits per symbol needed to represent the source.

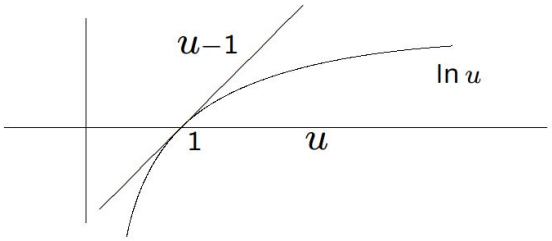
Let $\overline{L}_{min}$ be the minimum expected codeword length over all prefix-free codes for X. Then

$$H(X) \leq \overline{L}_{min} < H(X) + 1$$

$\overline{L}_{min} = H(X)$ iff each p_i is integer power of 2.

Let $l_1, \ldots, l_M$ be codeword lengths.

$$\begin{aligned} H(X) - \overline{L} &= \sum_i p_i \log \frac{1}{p_i} - \sum_i p_i l_i \\ &= \sum_i p_i \log \frac{2^{-l_i}}{p_i} \leq \sum_i p_i \left[\frac{2^{-l_i}}{p_i} - 1 \right] \log e \\ &= \sum_i \left[2^{-l_i} - p_i \right] \log e \leq 0 \end{aligned}$$



Concave(凹) function

Choose $l_i = \lceil -\log(p_i) \rceil$. Then

$l_i < -\log(p_i) + 1$ **so** $\overline{L}_{\min} \leq \overline{L} < \mathbf{H}(X) + 1$

$l_i \geq \log(p_i)$ **so** $\sum_i 2^{-l_i} \leq \sum_i p_i = 1$ (Kraft inequality)

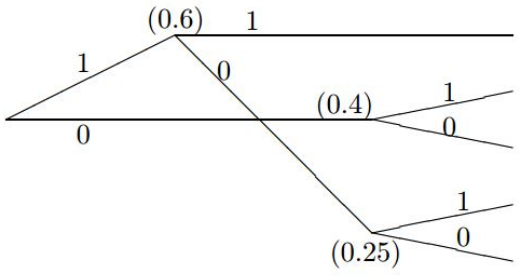
1.8 Huffman's algorithm for optimal source codes

Lemma *Optimal codes have the property that if $p_i > p_j$, then $l_i \leq l_j$.*

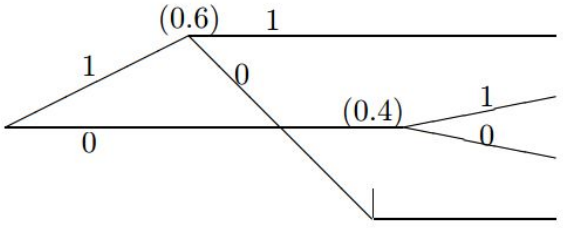
Lemma *Optimal prefix-free codes have the property that the associated code tree is full.*

Lemma *Optimal prefix-free codes have the property that, for each of the longest codewords in the code, the sibling of that codeword is another longest codeword.*

Lemma *Let X be a random symbol with a pmf satisfying $p_1 \geq p_2 \geq \cdots \geq p_M$. There is an optimal prefix-free code for X in which the codewords for $M-1$ and M are siblings and have maximal length within the code.*



p_j	symbol	codeword
0.35	1	11
0.2	2	01
0.2	3	00
0.15	4	101
0.1	5	100



p_j	symbol	codeword
0.35	1	11
0.2	2	01
0.2	3	00
0.25	4	10

$$\overline{L} = \overline{L}' + p_{M-1} + p_M$$

1.9 Review

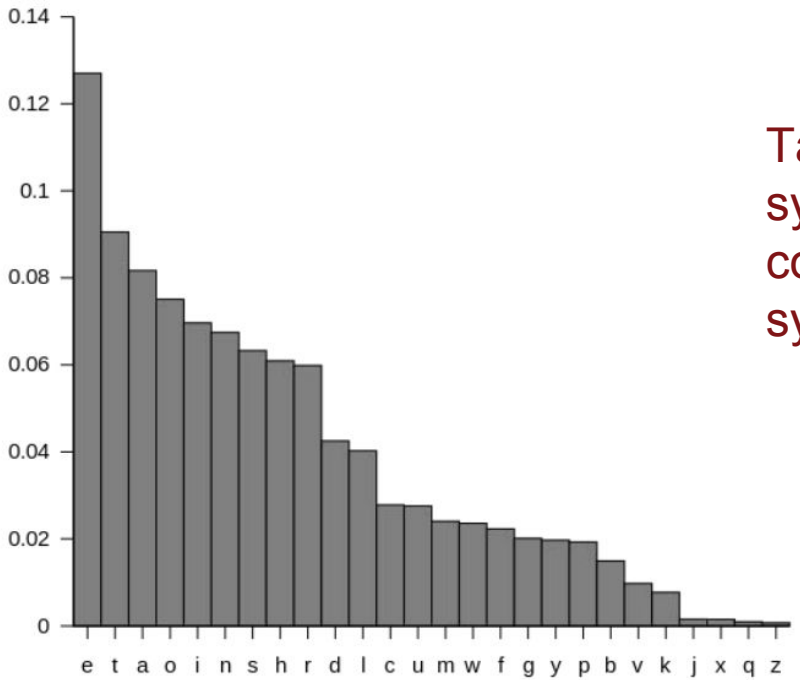
The Kraft inequality, $\sum_i 2^{-l_i} \leq 1$, is a necessary and sufficient condition on prefix-free code-word lengths.

Given a pmf, $p_1, \ldots, p_M$ on a set of symbols, the Huffman algorithm constructs a prefix-free code of minimum expected length, $\overline{L}_{\text{min}} = \sum_i p_i l_i$.

A discrete memoryless source (DMS) is a sequence of iid discrete chance variables $X_1, X_2, \ldots$. The entropy of a DMS is $\mathsf{H}(X) = \sum_i -p_i \log(p_i)$.

Theorem: $\mathsf{H}(X) \leq \overline{L}_{\text{min}} < \mathsf{H}(X) + 1$.

1.10 Relaxed Moment



Taking account of actual individual symbol probabilities, but not using context, entropy = 4.177 bits per symbol.

Shannon (1951) and others have found that the entropy of English text is a lot lower than 4.177

- Shannon estimated 0.6-1.3 bits/letter using human expts.
- More recent estimates: 1-1.5 bits/letter

The End

Thank You