Information and Coding Theory
NISER-M464-2012

Instructor: Deepak Kumar Dalai

28 December 2011

1 Introduction

What is a code ? Code is nothing but a mapping from a set of message words to a set of words, called code words.

Example 1. *1. ASCII code of alphabet, digits & symbols to understand by computers.*
2. Writing C-Program of an algorithm to understand by C-compilers.
3. Signature of a person to authenticate documents.

We want to communicate information, in terms of texts, from a sender, through a channel, to a receiver. There are many problems to send it through a channel. There are different necessities while sending the information.

1. Economy: In many situations it is desirable to use an alphabet smaller than that occur in natural language. This led to the techniques for codes for Data Compression. The study is called Information Theory.(Ex: Huffman code, jpeg, zip, tar.gz).

2. Reliability: Messages may be altered by noise in the process of transmission. Thus there is a need for codes that allow for Error detection and Error correction. The study is called Error correcting codes.

3.Security: Some messages are sent with the requirement that only the right person can understand them. This area of coding is known as Cryptography.

4.Suitability: The messages need to be coded such way that they are suitable for the channel for the electronic communication. For example ASCII codes.

References: 1. Coding and Information Theory by Steven Roman
2. Coding and Information Theory by Richard W. Hamming
3. Error correcting Codes by Mac William and Sloane

2 Information Theory

How to define information ?

Suppose you are reading a book, when do you say that you get some information ? Is it, when you will be suprised reading a text. If you are certain that which text you are going to read, then you are not gaining any information. If you are uncertain about the next text, then you will be surprised about occurence of the text and will gain some information.

The words "uncertainty", "surprise" and "information" are related. Before the event (experiment, reception of a message symbol, etc), there is the amount of uncertainty; when the event happens there is the amount of surprise; and after the event there is the gain of information. All these amounts are same.

Definition: Source: A source is an ordered pair $S = (X, p)$, where $X = \{x_1, x_2, \ldots, x_n\}$ is a finite set, known as a sourse alphabet, and P is a probability distribution on X. The probability of x_i by p_i.

When we receive a symbol x_i from X, how much information do we get ? How to quantify the information conveyed by the occurence of x_i of a source S. Let define a number $I(x_i)$ for each i, to measure how much informaion one gains on the occurence of x_i.

For example, if $p_1 = 1$ (and of course the other $p_i = 0$), then there is no surprise, no gain of information, since we were certain what the message must be.

On the other hand, if the probabilities are different, then when a symbol with low probability comes, we would fell more surprised, get more information than when a symbol of high probability came.

1. Thus, the amount of information is strictly decreasing function on the probability of occurence, with $I(x_i) = 0$ if $p_i = 1$.

We also feel that surprise is additive i.e., the information from two independent symbols is the sum of the information from each separately.

2. Thus $I(x_i x_j) = I(x_i) + I(x_j)$.

3. The function should be continous.

A function I satisfies 1, 2 and 3 is satisfied if and only if it has the form $I(x_i) = \log \frac{1}{p_i} = -\log p_i$.

Draw the graph of I vs p_i

The base is not very important, since any set of logs is proportional to another set of logs as $\log_a x = \log_a b \log_b x$. We usually take the base n, the number of symbols. In the binary case the we usually take base 2 and the units of information is called *bits*. We use base e (10), then the unit of information is called *nat* (*Hartley*).

Example 2. *Let S be an unbiased coin with x_1 and x_2 representing head and tail. Then $I(x_1) = I(x_2) = \log_2 2 = 1$. Thus 1 bit of information we get on the occurrence of head/tail from a toss of a unbiased coin.*

2.1 Entropy

The expectation function of getting information after sampling a symbol from S is

$$H_r(S) = \sum_{i=1}^{n} p_i I(x_i) = \sum_{i=1}^{n} p_i \log_r \frac{1}{p_i}$$

and $H_r(S)$ is called as *[r-ary] entropy* of the source S.

Example 3. $S = (X = \{x_1, x_2, x_3, x_4\}, \{p_1 = \frac{1}{2}, p_2 = \frac{1}{4}, p_3 = \frac{1}{8}, p_4 = \frac{1}{8}\})$.

$H_2(S) = 1\frac{3}{4}$ *bits of information.*

Example 4. *Consider binary source such as tossing of a coin with the probability of getting x_1(head) and x_2 (tail) are p and $1-p$. Then*

$$H_2(S) = p \log_2 \frac{1}{p} + (1-p) \log_2 \frac{1}{1-p}$$

Example 5. *If S is unbiased source i.e., $p_i = \frac{1}{n}$ for $1 \leq i \leq n$, then $H_r(S) = \log_r n$.*

The entropy of a balanced dice is $H_2(S) = \log_2 6 \approx 2.5849$ bits of information.

Example 6. *One gets an average of 4.07991 bits of information by sampling a single letter from English text.*

29 December 2011

2.2 Mathematical Properties of Entropy

Lemma 1. *[1] For all $x > 0$, $\log_e x \leq x - 1$. Equality holds iff $x = 1$*

Lemma 2. *[1, page 110] Let $P = \{p_1, \dots, p_n\}$ and $Q = \{q_1, \dots, q_n\}$ be two probality distributions. Then*

$$\sum_{i=1}^{n} p_i \log \frac{1}{p_i} \leq \sum_{i=1}^{n} p_i \log \frac{1}{q_i}$$

where $0. \log \frac{1}{0} = 0$ and $p. \log \frac{1}{0} = +\infty$ for $p > 0$. The equality holds iff $p_i = q_i, \forall i$

Theorem 1. *[2] Let X be a discrete random variable with range $\{x_1, \dots, x_n\}$. Then*

$$0 \leq H(X) \leq \log n.$$

Furthermore, $H(X) = \log n$ iff $p(x_i) = \frac{1}{n} \forall i$ and $H(X) = 0$ iff $p(x_i) = 1$ for some i.

Joint Entropy : How much [joint] information we expect by sampling two random variables.

Theorem 2. *[2] Let X and Y be discrete random variables. Then $H(X, Y) \leq H(X) + H(Y)$ with equality holding iff X and Y are independent.*

corollary 1. *[2] Let $X_1, \dots, X_n$ are discrete random variables. Then $H(X_1, \dots, X_n) \leq H(X_1) + \dots + H(X_n)$ with equality holding iff $X_1, \dots, x_n$ are independent.*

3 Noiseless Coding [2]

Source encoding for noiseless transmission. How to encode the text such that it contains least number of alphabet (i.e., most information per alphabet) to transmit in a noiseless channel ? Example of book: translating a book in English to Odia or Hindi. The transalated book will be thick. Is there any other language (encoding) which will be thinner ? What is the best language ?

3.1 Variable Length Encoding

Strings and Codes:
Let $\mathcal{A} = \{a_1, \dots, a_n\}$ be a finite set, refered as an *alphabet*. A *string*, or *word*, over the alphabet $\mathcal{A}$ is any sequence of elements of $\mathcal{A}$, which is written of the form $a = a_{i_1} a_{i_2} \dots a_{i_k}$. *empty string*($\theta$) is with no symbols. The length of a string a, *len*(a), is the number of alphabet symbols appearing in a. The set of all strings over $\mathcal{A}$ is denoted by $\mathcal{A}^*$.

[r**-ary**] **code**: Let $A = \{a - 1, \dots, a_r\}$ be a finite set, which is called as *Code alphabet.* An *[r-ary] code* is a nonempty subset C of the set A^*. The size of the code alphabet is called the *radix* of the code and the elements of C are called codewords. Example: binary code, ternary code.

Encoding scheme: Let $S = (X, P)$ be a source. An encoding scheme for S is an ordered pair (C, f), where C is a code and $f : X \mapsto C$ is an injective function, called an *encoding function.*

Average codeword length: The *average codeword length* of an encoding scheme (C, f) for a source $S = (X = \{x_1, \dots, x_n\}, P)$, is defined by

$$AveLen(C, f) = \sum_{i=1}^{n} P(x_i) len(f(x_i))$$

Example [2, Example 2.11].

Note that the average codeword length of an encoding scheme is not same as the average codeword length of a code. Our goal is to determine code of minimum average codeword length among all codes.

Fixed and Variable length codes
If all codewords in a code C have the same length, we say C is a *fixed length code* or, *block code*. If all codewords in a code C are of different lengths, we say C is a *variable length code*. Any coding scheme that uses fixed length code is refered as *fixed length encoding scheme* and similarly, for *variable length encoding scheme*.

If the probability distribution is not uniform, variable length encoding is more efficient than other one. Example: $S = \{s_1, \ldots, s_5\}$, $p(s_1) = 1 - \epsilon, P(\{s_2, s_3, s_4, s_5\}) = \epsilon$. Average codeword length $= 3$ (fixed length scheme), $= 1 + 2\epsilon < 3$ if $\epsilon < 1$ if we encode as $s_1 = 0, s_2 = 100, s_3 = 101, s_4 = 110$ and $s_5 = 111$.

Trouble in variable length coding: $S = \{a, b, c\}, C = \{0, 01, 001\}, f(a) = 0, f(b) = 01, f(c) = 001$. The codeword 001 can be decoded as ab or as c.

Uniquely decodeble/decipherable: A code C is uniquely decipherable/decodable if whenever $c_1, \ldots, c_k, d_1, \ldots, d_j$ are code words in C and

$$c_1 \ldots c_k = d_1 \ldots d_j$$

then $k = j$ and $c_i = d_i$ for all $i = 1, \ldots, k$.

Example: $S = \{a, b, c\}, C = \{1, 01, 001\}, f(a) = 0, f(b) = 01, f(c) = 001$. 1 acts as a codeword separator. 1001011 can be uniquely decodable as *acba*.

Instantaneous codes (Prefix property)
Another trouble in variable length coding: $S = \{a, b, c, d\}, C = \{0, 01, 011, 0111\}, f(a) = 0, f(b) = 01, f(c) = 011, f(d) = 0111$. If 0111 is transmitted, we can not decide which code word is transmitted unless we get all. After receiving 0, we cannot decide whether it is for a or, b, or c, or d. After receiving 01, we cannot decide whether it is for b, or c, or d.
Example: $S = \{a, b, c, d\}, C = \{0, 10, 110, 1110\}, f(a) = 0, f(b) = 10, f(c) = 110, f(d) = 1110$. In this case the codewords can be decodable as soon as they are received, since the presence of 0 indicates the end of a code word.
Instantaneous code: A code is said to be *instantaneous* if each code word in any string of codewords can be decoded as soon as it is received.

Instantaneous code and decision tree (finite automaton) [1].

This problem occurs iff a codeword is a prefix of another codeword.

Prefix property: A code is said to have *prefix property* if no codeword is a prefix of any other codeword, i.e., if when ever $c = x_1 \ldots x_k$ is a code word, then $x_1 \ldots x_l$ is not a codeword for $1 \le l < k$. Example: $C = \{1, 01, 001\}$.

Theorem 3. *A code C is instantaneous iff it has the prefix property.*

Example: Comma code, $C = \{0, 10, 110, \ldots, 1 \ldots 10, 11 \ldots 1\}$.

Construction of instantaneous code [1].

Theorem 4. Kraft's Theorem

1. *If C is an r-ary instantaneous code with codeword lengths $l_1, \ldots, l_n$, then these lengths must satisfy* **Kraft's inequality**
$$\sum_{i=1}^{r} \frac{1}{r^{l_i}} \le 1$$
2. *If the numbers $l_1, \ldots, l_n$ satisfy* **Kraft's inequality** *then there is an instantaneous r-ary code with codeword length $l_1, \ldots, l_n$.*

Example 2.1.6 [2] to construct an instantaneous code for a given lengths satisfying Kraft's inequality.

When does Kraft's equality satisfy? Examples of codes satisfying equality: Block code, comma code.

Kraft's enequality => existence of instantaneous code => existence of uniquely decodable code. Is uniquely decodable code => Kraft's inequality ?

Theorem 5. McMillan's Theorem
If $C = \{c_1, c_2, \ldots, c_n\}$ is a uniquely decipherable r-ary code, then it's codeword lengths $l_1, l_2, \ldots, l_n$ must satisfy Kraft's inequality

$$\sum_{i=1}^{n} \frac{1}{r^{l_i}} \leq 1.$$

Theorem 6. *If a uniquely decipherable code exists with codeword lengths $l_1, l_2, \ldots, l_n$, then an instantaneous code must exist with these same codeword lengths.*

corollary 2. *The minimum average codeword length, among all uniquely decipherable encoding scheme for a source S, is equal to the minimum average codeword length among all instantaneous encoding scheme for S.*

References

[1] R. W. Hamming. *Coding and Information Theory.* Printice Hall, 1980.

[2] S. Roman. *Coding and Information Theory.* Springer, 1992.