ISSN:0975-9646

Gaurav Sethi et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (4) , 2014, 5584-5586

Data Compression Techniques

Gaurav Sethi , Sweta Shaw ,Vinutha K , Chandrani Chakravorty

R.V. College of Engineering,
Bangalore.

Abstract : Data compression has important application in the field of file storage and distributed systems. It helps in reducing redundancy in stored or communicated data. This paper studies various compression techniques and analyzes the approaches used in data compression. Furthermore, information theory concepts that relates to aims and evaluation of data compression methods are briefly discussed. A framework for the evaluation and comparison of various compression algorithms is constructed and applied to the algorithms presented here.
This paper reports the theoretical and practical nature of compression algorithms. Moreover, it also discusses the future possibilities of research work in the field of data compression.

Keywords : Redundancy, Entropy, Optimality, Compression ratio, Probability coding

INTRODUCTION

Data compression is primarily a branch of information theory[1] which deals with techniques related to minimizing the amount of data to be transmitted and stored. The basic characteristic of data compression is to convert a string of characters into another set of characters which consists of same information but whose length is as small as possible.

With the extending use of computer in various disciplines, number of data processing applications are also increasing which requires processing and storage of large volumes of data. Simultaneously, proliferation of computer networks is encouraging the passive transmission of data over communication channels. In the above described scenario, it is best to compress the data in order to reduce the storage and communication costs. Reducing the size of a file to its half is equivalent to doubling the storage medium capacity.
On the downside, compressed data needs to be decompressed in order to view the data and this extra processing may prove detrimental to some applications. For instance, a scheme for compressing a video may require expensive hardware for decompressing the contents of the video. Hence the design of compression schemes involves trade-offs between various factors such as degree of compression, amount of distortion induced and computational resources involved in compressing and decompressing the data.

FUNDAMENTAL CONCEPTS

This section provides introduction to the information theory. It delivers the definitions and assumptions necessary for comprehensive study of compression algorithms.

INFORMATION THEORY

Information theory[2] is a branch of applied mathematics and computer science that deals with the signal process operations used for the purpose of compression and efficient storage and communication of data.

ENTROPY

Shannon derived the definition of entropy from statistical physics where entropy is understood as the randomness or disorder of a system. Entropy is mathematically defined as:
$H(S) = \sum p(s) \log2 (1/p)$
where S is the set of possible messages and p(s) is the probability of message s € S. Larger entropies represent larger average information thus the more random a set of messages the more information they contain on average.

PROBABILITY CODING

A code is a function that maps a source message into codeword. Source message is the basic units into which the string is broken. These basic units may be a single character or a set of characters. Codes can be classified as block-block, variable-variable, variable-block block-variable, or where block-block refers to mapping of source message into codeword of fixed length and variable-variable refers to mapping of source message to codeword of variable length.

The most commonly used codes, ASCII and EBCDI are the examples of block-block codes but they do not provide compression. Coding for achieving compression is basically taking probabilities for message and generating bit strings based on these probabilities. We use probability for the larger parts of message rather than the complete message.

CLASSIFICATION METHODS :

Data compression methods can also be categorized into static and dynamic compression methods. In static method, mapping from the set of messages to the set of codeword is fixed before transmission begins. Huffman coding [Huffman 1952] is the example of classic static defined word scheme. On the other hand, dynamic method changes the mapping of set of messages to the set of codeword over a period of time. For instance dynamic Huffman coding computes approximate probability of occurrence of a set of characters in a message.

FRAMEWORK FOR COMPARING DATA COMPRESSION MODELS :

In order to compare various data compression models, a framework must be established to decide about the relative efficiency of the compression techniques. Here we consider two basic dimensions along which the data compression schemes will be measured: algorithm complexity and

degree of compression achieved. In general, speed of the data compression is an important factor when it is used for the purpose of data transmission. Although the amount of compression is a primary concern in the case of storage applications, it is nonetheless necessary for the algorithm to be efficient enough to be applied practically for various schemes.

Several other measures that have been suggested are: redundancy, message length, compression ratio, decompression time, resources required for compression and decompression procedure.

Information theory assumes that all statistical parameters of a message source are known with perfect accuracy. It is also assumed that any cost associated with the code letters is uniform.

Redundancy: It is measured by finding the difference between the average codeword length and average information content. A code is said to be minimum redundancy code if it has minimum average codeword length for a discrete probability distribution.

Optimality: A code is said to be asymptotically optimal if for a given probability distribution the ratio between the average codeword length and entropy approaches 1 as the entropy tends to infinity.

Average Message Length: It is defined as SUM p(a(i)) l(i), where l(i) is the length of the codeword representing message a(i).This expression represents the length of codeword weighted by their probability of occurrence.

Compression Ratio: It is a comparison of the length of the coded message to the length of the original message. It is denoted by C and is given by the ratio of average message length to average codeword length.

STATIC ALGORITHMS COMPRESSION

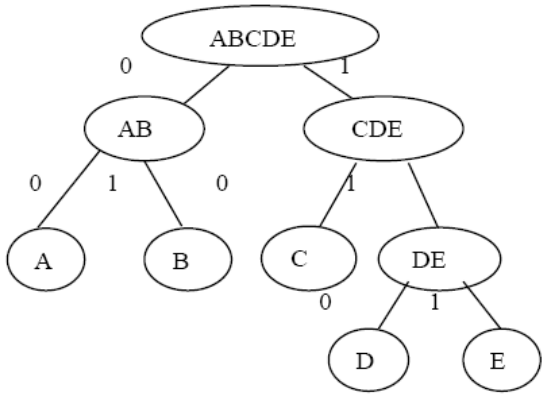
1. SHANNON – FANO CODING :

Shannon – Fano is very simple to implement but may not give the best compression ratio. Basically, it takes the source messages a(i) and their probabilities P(a(i)) and lists them in the order of decreasing probability.

This list is then divided into two groups with nearly equal total probabilities. The messages in first group are given 0 as prefix code and the messages in the second half of the list are given 1 as the prefix code. Each of the subgroups are iteratively divided according to the same criteria and the prefix codes are appended to them until each subset contains only one message. Thus, Shannon – Fano yields a minimal prefix code.

For Example: Let us consider following data related to a message ensemble.

Symbol	Frequency	Code	Total length
A	24	00	48
B	12	01	24
C	10	10	20
D	8	110	24
E	8	111	24

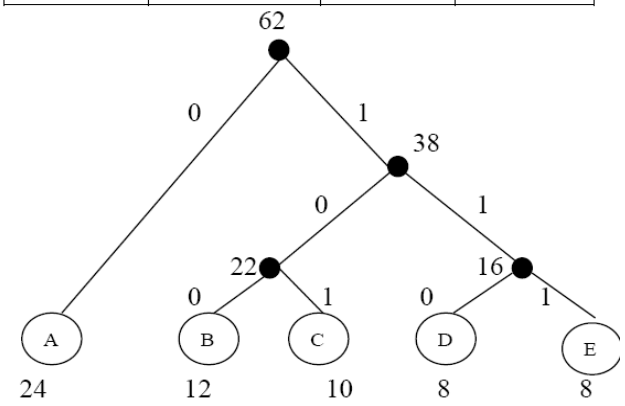


2. STATIC HUFFMAN CODING

Huffman algorithm inputs a list of non-negative weights [w = w1,w2, w3...wn] that represents the frequency or cost of the message symbol. It constructs a full binary labelled tree using weights. Initially for each weight in the list, a set of singleton tree is constructed. In each step two trees with the minimum weight (say w1 and w2) are merged to form a new tree. These two weights are then removed from the list and their sum, that is, (w1+w2) is added to the list. This process continues until there is only single value left in the weight list.

For Example: For the following table of data, Huffman code tree can be constructed as:

Symbol	Frequency	Code	Total length
A	24	0	24
B	12	100	36
C	10	101	30
D	8	110	24
E	8	111	24



COMPARISON OF STATIC HUFFMAN CODING AND SHANNON FANO CODING :

Although both Huffman coding [6] and Shannon – Fano coding generates a minimal prefix code for a message, it is noteworthy that Huffman coding scheme provides the optimal result, that is, minimum redundancy code. As proved by Gallager, Huffman codes have an upper bound on redundancy of P(n)+0.086, where P(n) is the probability of least occurring source message.

CONCLUSION

Data compression[8] is a topic of much importance and many applications. The Methods of data compression studied for almost four decades. This paper provided overview of the data compression methods of general utility. This Algorithms evaluated in terms of the amount of compression they provide efficiency of algorithm and the susceptibility to error.

Semantic dependent data compression techniques, are special- purpose methods designed to exploit local redundancy or context information. It should also noted that the algorithm BSTW is a general-purpose technique.

Susceptibility to error is the main drawback of each of the algorithms presented here. The channel errors are more devastating to adaptive algorithms than to static ones, it is possible for an error to propagate without limit even in the static case. Methods of limiting the effect of an error on the effectiveness of a data compression algorithm should be investigated.

REFERENCES

[1] Abramson, N. 1963 1963 Information Theory and Coding McGraw-Hill, New York

[2] Ash, R. B. 1965. Information Theory. Interscience Publishers, New York

[3] Connell, J. B. 1973. A Huffman-Shannon-Fano Code. Proc. IEEE 61, 7 (July), 1046-1047

[4] Cormack, G. V., and Horspool, R. N. 1984. Algorithms for Adaptive Huffman Codes. Inform

[5] Gallager, R. G. 1978. Variations on a Theme by Huffman. IEEE Trans. Inform. Theory 24, 6 (Nov.), 668-674

[6] Glassey, C. R., and Karp, R. M. 1976. On the Optimality of Huffman Trees. SIAM J. Appl. Math 31, 2 (Sept.), 368-378.

[7] Hester, J. H., and Hirschberg, D. S. 1985. Self-Organizing Linear Search. ACM Comput. Surv. 17, 3 (Sept.), 295-311.

[8] Horspool, R. N. and Cormack, G. V. 1987. A Locally Adaptive Data Compression schema Commun. ACM 16, 2 (Sept.), 792-794.

[9] Knuth, D. E. 1985. Dynamic Huffman Coding. J. Algorithms 6, 2 (June), 163-180.

[10] Reghbati, H. K. 1981. An Overview of Data Compression Techniques. Computer 14, 4 (Apr.), 71-75