Indian J.Sci.Res. 17(2): 336-339, 2018

ISSN: 0976-2876 (Print)
ISSN: 2250-0138(Online)

NOISELESS DATA COMPRESSION TECHNIQUE BY USING BURROWS-WHEELER TRANSFORM

[1]Karuna Khobragade, [2]Shruti Malame, [3]Pratiksha Kapse
[1,2,3] Department of Computer Science, Arts, Commerce and Science College,Tukum,Chandrapur, Maharashtra.

Abstract- It seems that there is no limit to the amount of data we need to store in our computers and also send these data to our friends and colleagues. For this purpose people tend to store a lot of files inside their storage. When the storage nears it's limit, then they try to reduce those files size to minimum by using data compression software's. Compression aims to represent an input data with least number of bits. In this paper we describe the Burrows-Wheeler Transform (BWT), a data compression technique which is the basis of some of the best compressor available today.

***Keywords*:** Data Compression, Noiseless, Noisy, BWT, Arithmetic Coding.

I. Introduction

Contemporary computers process and store huge Amount of data. The data may in the form of text,sound(audio, video), graphics(images).Thesedata are store and transmitted over (telecommunication satellite) networks.To store or transmit the relevant data, the data are preprocessed and/or postprocessed. These preprocessed or post processed form of data may include compression and decompression.

The data compression technique is reducing the amount of data required to represent a source of information for reducing the space required for data storage also reduces the time of data transmission over network. The process of reducing the size of a data file is popularly refered to as data compression.

There are Two Major categories of compression algorithm.

1) Noisy or Lossy
2) Noiseless or Lossless

Noisy: Noisy compression algorithm involves the reduction of a file size usually by removing small details.In this scheme some loss of information is acceptable. It is used for compressing the picture,videos,and sounds. Digital cameras and DVDs comes under this noisy compression.

Noiseless: With Noiseless compression, data is compressed without any loss of data. It means you want to get everything back that you put in.

Critical financial data files are examples where noiseless compression is required. It also used in the zip file format and in UNIX tool of gzip.

This type of compression used for storing database records, spreadsheet of word processing files. In these applications the loss of even a single bit could be catastrophic.

Noiseless data compression techniques are-

1. Run Length Encoding (RLE)
2. Burrows-Wheeler Transform (BWT)
3. Entropy Coding
   i) Shannon-Fano Coding
   ii) Huffman Coding
   iii) Arithmetic Coding

From all these noiseless data compression techniques we are going to discuss about the BWT.

This Burrow-Wheeler Transform was developed (invented) bythe Michael Burrows and David Wheeler in 1994.The Burrow-wheeler transform is a Block-Sorting Noiseless Data Compression Algorithm that works by applying a reversible transformation to a block of input data. BWT is a transformation algorithm that does not compress data but rearrange or change data to optimize input for next sequence of transformation or compression algorithm.

In short, the BWT itself does not perform any compression operations, i.e. it needs some other compression algorithm or technique for compressing that particular inputed data.It simply transform the input such that it can be more efficiently coded by Run-Length Encoder or Other Secondary Compression technique like Arithmetic coding.

The BWT is an algorithm that takes a block of data and rearranges it using a sorting algorithm. The resulting output block contain exactly the same data elements that it started with differencing only in their ordering. The transformation is reversible, meaning the original ordering of the data elements can be restored with no loss of fidelity. The BWT is performed on an entire block of data at once. Most of todaysfamiliar lossless compression algorithm operate in streaming mode, reading a single byte or a few bytes at a time. But with this new transform, we want to operate on the largest chunks of data possible.Since the BWT operates on data in memory, you

[1]**Corresponding Author**

may encounters files too big to processed in one fell swoop. In these cases the file must be split up and processed a block at a time.

II.Aim

The aim ofthis research paper is to study the Burrows-Wheeler Transform which is used in Data Compression. The main purpose behind this BWT is to compress data into the smallest space as possible, so it saves the storage space and provides an efficient format to allow faster data transmission via different networks. While Compressing the data into smallest storage space, the arithmetic coding can be used to improve the Compression ratio. The BWT based compression is close to the best known algorithm for text data nowadays, it could beuse to improve the compression performance of data.

III. Objectives:

The objective of this proposed work is -

1.To implement the BWT with the use of Arithmetic Coding of noiseless text compression algorithms for text transmissions with efficient utilization of communication bandwidth and for archival storage.

2.To develop new text compression techniques along with basic understanding of the interaction of encoding schemes and compression algorithms.

3.To achieve better data compression ratio to save storage space and better bandwidth.

4. To gain an understanding of (basic versions of major data compression algorithms, including Arithmetic coding and Burrows-Wheeler compression.

IV. Literature Review

Data Compression reduces the redundancy of the (text)data and also reduces the size of the (text)data. In the past, the lot of research work on lossless compression has been done. The objective is to reduce redundancy of data in order to able to store or transmit data in an efficient form. There has been extensive research in data compression algorithm and techniques [1,2,3,4,5&6].

The continuous attempts to obtain better efficiency of the lossless image compression lead to developing methods of increased implementation complexity.Some common compression technique include, Run Length Encoding[7], BWT[8,9], Ziv-Lampel[5], Huffman coding[11],arithmetic coding[10,15&16].The family of the block sorting algorithms based on the Burrows-Wheeler Transform(BWT) has grown over the past few years starting with the first implementation describedby Burrows and Wheeler [1994][8,9]. Several authors have presented improvements to theoriginal algorithm. Andersson and Nilsson have published several papers about RadixSort, which can be used as a first sorting step during the BWT [1994, 1996, 1998][17 &18]. In hisfinal BWT research report, Fenwick described some BWT sort improvements includingsorting long words instead of single bytes [1995][21&22]. Kurtz presented several papers aboutBWT sorting stages with suffix trees, which needed less space than other suffix treeimplementations and are linear in time [1998, 1999][19,20,23&24].Sadakane described a fast suffix array sorting scheme in 1997 and 2000[30,31]. In 1999, Larssonpresented an extended suffix array sorting scheme. Based on already sorted suffices[25],Seward developed in 2000 two fast suffix sorting algorithms called "copy" and "cache"[27].Itoh and Tanaka presented a fast sorting algorithm called the two stage suffix sort [1999][28].Kao improved the two stage suffix sort by some new techniques which are very fast forsequences of repeat symbols [1999][29]. Manzini and Ferragina published in 2002 someimproved suffix array sorting techniques based on the results of Seward and of Itoh andTanaka[26].

V.Methodology

Noiseless Data compression techniques use some methods for compressing the data for better transmission over networks and for saving the storage space. These techniques/ methods are –

1. Run Length Encoding (RLC)
2. Burrow-Wheeler Transform (BWT)
3. Entropy Encoding

i. Shannon-fano coding

ii. Huffman coding

iii Arithmetic coding

Burrow-Wheeler Transform (BWT) works in block mode while others mostly works in streaming mode. This algorithm classified into transformation algorithm because the main idea is to rearrange (by adding and sorting) and concentrate symbols. These concentrated symbols then can be used as input for another algorithm to achieve good compression ratios.

Since the BWT operates on data in memory, you may encounter files too big to process in one fell swoop. In these cases, the file must be split up and processed a block at a time[14](To speed up the sorting process, it is possible to do parallel sorting or using larger block of input if more memory available).

The algorithm for BWT is-

1. Create a string array
2. Generate all possible rotations of the inputed string, storing each in the array
3. Sort the array alphabetically

4. Return the last column of the array

The output of the BWT is then submitted to any other compression algorithm like Arithmetic Coding for further processing.

A. Arithmetic coding:

One of the most powerful technique is called Arithmetic coding. This converts the entire input data into a single floating point number.Arithmetic coding encodes the entire message into a single number, a fraction *n* where ($0.0 \leq n < 1.0$).

Arithmetic coding (ARI) is using statistical method to compress data. The method starts with a certain interval, it reads the input file symbol by symbol, and uses the probability of each symbol to narrow the interval. Specifying a narrower interval requires more bits, so the number constructed by the algorithm grows continuously. To achieve compression, the algorithm is designed such that a high-probability symbol narrows the interval less than a low-probability symbol, with the result that high-probability symbols contribute fewer bits to the output [12].

Arithmetic coding, is entropy coder widely used, the only problem is its speed, but compression tends to be better than Huffman (other statistical method algorithm) can achieve [13]. This technique is useful for final sequence of data compression combination algorithm and gives the most for compression ratio.

VI.Conclusion and Future Work

BWTNoiseless Data Compression has a wide range of applications andvariety of algorithms are developed for number of applications with maximum storage space and compression ratio. This research paper provide the better compression ratio by implementing the BWT and developing new techniques by using the arithmetic coding. Improvements in compressed size are obtained by alphabet reordering and selective reversal within column of the sorted matrix. This paper is also help to minimize the storage space.

References

[1] T.C.Bell,J.G.Cleary and I.H. Written, “Text Compression”, prentice Hall Publisher,1990.

[2] R.M.Gray, “Entropy and Information Theory”, Stringer-Verlag, 1990.

[3] R.M. Gray, “Source Coding Theory”, Kluwer Academic Publishers,1991.

[4] G.Held, “Data Compression”, John Wiley & Sons, 1991.

[5] J.Ziv and A.Lampel, “A Universal algorithm for sequential data DataCompression”, IEEE Trans.Information Theory, 23,337-343,1977.

[6] J.Ziv,“Coding Theorems for individual sequences”, IEEE Trans.Information Theory, 24,405-412,1978.

[7] S.W. Golomb, “Run-length encoding," IEEE Trans. Inform. Theory, vol. 2, no. 4, pp. 399-401, 1966.

[8] M.Burrows and D.J.Wheeler. A Block sorting Data CompressionAlgorithm.Technical Report, DIGITAL system Research Center,1994.

[9] M. Burrows and D.J. Wheeler, A Block-sorting Lossless DataCompressionAlgorithm. DigitalSystems Research Center Technical Report 124, Digital EquipmentCorporation, Palo Alto,CA, 1994.

[10] P. Howard & J. Vitter, “ Arithmetic Coding for Data Compression”, ProceedingsIEEE, vol.82,no.6,June 1995,pp.857-865.

[11] D.E. Knuth, “Dynamic Huffman coding," J. of Algorithms, vol. 6, pp. 163-180, June 1985.

[12] Salomon, D. 2004. Data Compression the Complete References ThirdEdition.Springer-Verlag New York, Inc.

[13] Campos, A. S. E. Basic arithmetic coding. Available:http://www.arturocampos.com/ac_arithmetic.html (last accessed July2012).

[14] Nelson, M. 1996. Data compression with Burrows-Wheeler Transform.Dr. Dobb's Journal.

[15] G.G. Langdon, “An introduction to arithmetic coding," IBM J. Res.Develop., vol. 28, no. 2,pp. 135-149, March 1984.

[16] I.H. Witten, R.M. Neal, and J.G. Cleary, “Arithmetic coding for datacompression,"Commun.ACM, vol. 30, no. 6, pp. 520-540, June 1987.

[17] ANDERSSON, A. AND NILSSON, S. 1994. A New Efficient Radix Sort.In *35th Symposium on Foundationsof Computer Science*, 714□721.

[18] ANDERSSON, A. AND NILSSON, S. 1998. Implementing Radixsort.*The ACMJournal of ExperimentalAlgorithmics*. Volume 3, Article 7.

[19] BALKENHOL, B. AND KURTZ, S. 1998. Universal DataCompression Based on the Burrows-WheelerTransformation: Theoryand Practice. *IEEE Transactions on Computers*, 49(10), 1043□1053.

[20] BALKENHOL, B., KURTZ, S. AND SHTARKOV, Y.M. 1999.Modifications of the Burrows and WheelerData Compression Algorithm.In *Proceedings of the IEEEData Compression Conference1999*, Snowbird.

[21] FENWICK, P. 1995. Improvements to the Block Sorting Text CompressionAlgorithm. Technical Report 120,University ofAuckland, New Zealand, Department of Computer Science.

[22] FENWICK, P. 1996. Block Sorting Text Compression - Final Report.*Technical Report 130*, University ofAuckland, New Zealand,Department of Computer Science.

[23] KURTZ, S. 1998. Reducing the Space Requirement of Suffix Trees.*Report 98□03*, TechnischeFakultat,Universitat Bielefeld.

[24] KURTZ, S. AND BALKENHOL, B. 1999. Space Efficient Linear Time Computation of the Burrows andWheeler-Transformation. ALTHÖFER, I. ET AL. Eds. *Numbers, Information and complexity*,Festschrift inhonour of Rudolf Ahlswede's 60th Birthday, 375□384

[25] LARSSON, N.J. 1999. *Structures of String Matching and DataCompression*.PhD thesis, Department ofComputer Science, Lund University, Sweden.

[26] MANZINI, G. AND FERRAGINA, P. 2002. Engineering aLightweight Suffix Array Construction Algorithm.*Lecture Notes in Computer Science*, Springer Verlag, Volume 2461, 698□710.

[27] SEWARD, J. 2000. On the performance of BWT sorting algorithms.In*Proceedingsof the IEEE DataCompression Conference 2000*,Snowbird, Utah, STORER, J.A.AND COHN, M. Eds. 173□182.

[28] ITOH, H. AND TANAKA, H. 1999. An Efficient Method forConstruction of Suffix Arrays.*IPSJ Transactionson Databases*, Abstract Vol.41, No.SIG01 – 004.

[29] KAO, T.-H. 2001. *Improving Suffix-Array Construction Algorithms with Applications*. Master's thesis,Department of Computer Science, Gunma University, Japan.

[30] SADAKANE, K. 1997. *Improvements of Speed and Performance of Data Compression Based on Dictionaryand Context Similarity*.Master's thesis, Department of Information Science, Faculty of Science, University ofTokyo, Japan.

[31] SADAKANE, K. 2000. *Unifying Text Search And Compression -Suffix Sorting, Block Sorting and Suffix Arrays*.PhD thesis, University of Tokyo, Japan.