**Lossless Text Compression using K-Means Clustering Technique**

**ABSTRACT**

**INTRODUCTION**

Data mining is defined as the process of discovering interesting knowledge from large amounts of data stored in databases, data warehouses, or other information repositories[1]. It is a widely developing field used primarily for Data analysis, interpretation, classification, etc. Although the conventional knowledge suggests usage of Data mining only for data analysis, Naren Ramakrishnan et.al. in his paper[2] has defined a whole new world which could be transformed by data mining. He has identified five different perspectives of Data Mining namely Compression, Induction, Approximation, Querying and Searching of which we have worked on exploiting machine learning algorithms used in Data Mining for improvising Text compression.

The machine learning algorithms used in Data Mining are broadly classified into supervised and unsupervised learning algorithms. Supervised Algorithms are those which have a predefined dataset based on which further outputs are obtained or rather predicted. These include algorithms like k-Nearest Neighbours, Naive Bayes and Bayesian Classification. Unsupervised learning algorithms are those that do not have any predefined dataset. The output is obtained by grouping similar data together. The future outcomes are predicted by learning from the present data and its outcomes. These algorithms are classified as Clustering, Factor Analysis and Latent Dirichlet Allocation. Among these, our work is based on Clustering algorithms.

Clustering may be defined as task of grouping the objects by their similarity. Objects with similar characteristics are grouped into the same cluster and dissimilar objects into different groups. In efficient clustering, the inter similarity of words in different clusters must be low whereas the intra similarity within a cluster must be high. Among the numerous clustering algorithms present we have taken the most seminal algorithm namely the K-means Clustering [6]. In this algorithm the given dataset is partitioned into clusters of a finite count K. K arbitrary objects are picked as centers of K clusters and every other object is put into a cluster with higher similarity. The similarity is physically measured by distance measures. A object is placed in a cluster whose center has the lowest distance from that object. The clusters are repeatedly checked for their mean values and objects are repositioned unless K clusters are properly formed.

Compression is the technique of representing data as a symbol to reduce space utilization and reduce the amount of data to be transmitted effectively. The compression techniques are of two types namely Lossless and Lossy compression. Lossless Compression techniques are primarily applicable for text data streams whereas Lossy compression techniques are majorly used for compression of multimedia data. In this paper we have used this clustering technique as a base to modify the existing Lossless text compression algorithms.In most lossless compression techniques the general methodology requires two steps. The first step involves generating code words for the data such that words with the high frequency have lesser code words and vice versa. The second step involves encoding the data based on the code words generated in the previous step. This type of compression procedure was initially proposed by Claude Shannon[4] and R.M. Fano[5] in the 1950’s, when they developed the Shannon-Fano compression algorithm. An improvement to this algorithm was proposed by D.A.Huffman [3] in 1952 which eliminated the drawbacks of Shannon Fano compression algorithm and provided a better compression ratio. \\

Both Shannon-Fano and Huffman schemes have the following similar properties:\\

\\1. Codes are obtained by variable length encoding.

\\2. Higher probable characters get shorter bit codes compared to less probable ones.

\\3. Obtained codes are uniquely decoded or satisfy the prefix property, i.e, no codeword of any character acts as prefix in any other codeword.

Among these properties we have exploited the second property in our work . When the text to be encoded gets larger the rarely occurring symbols get longer bit codes. Moreover for every word, its respective codeword becomes longer as the number of characters in it increases. We use this drawback to justify the need for another efficient compression technique. We bring in the concept of encoding words as a whole with their respective frequencies and length (as parameters), and also encode rarely occurring words, too with relatively shorter codewords. This concept has helped us obtain better results than the conventional huffman technique.

**RELATED WORK**

The earliest algorithm that formed the basis for compression was the Morse code. This is a method of transmitting words as a series of on-off tones, lights or clicks that can be understood by a skilled observer without special equipment [8]. The main characteristic of this code which enables fast transmission of messages is it’s property where in codes assigned to a letter are inversely proportional to its frequency i.e most occurring alphabet has least codeword length [8].

In 1948 , Shannon developed an lossless compression algorithm to generate prefix codes [4] for each letter based on their probabilities. However the performance of this algorithm was suboptimal. To eliminate the drawbacks of Shannon’s algorithm , Fano modified this and developed the Shannon-Fano Compression algorithm[5] in 1950. Even this algorithm did not generate the most optimal prefix codes for a letter.

By 1952, a revolution in the Lossless Compression forte was brought about by D.A.Huffman who introduced a new lossless prefix coding algorithm. It produced the most optimal prefix codes for letters based on their probabilities and has provided us with a very good compression ratio. Although Huffman Algorithm proved extremely efficient in encoding symbols , it may not be optimal compression algorithm for all types of data.

Following this [Abraham Lempel](https://en.wikipedia.org/wiki/Abraham_Lempel) and [Jacob Ziv](https://en.wikipedia.org/wiki/Jacob_Ziv) proposed two lossless compression algorithms called the LZ77 /cite {8} and LZ78 /cite {9} in the year 1977 and improvised those in 1978. During compression LZ77 algorithm uses a sliding window based dictionary approach wherein most recent data encoded are kept in track and each new sequence of data is represented as a references of their previous occurrence in uncompressed data streams. However in LZ78 algorithm an explicit dictionary is constructed for compression. During decompression LZ78 directly accesses the created explicit dictionary during compression. Whereas in LZ77 the decompression process starts from the beginning and continues sequentially. Both these algorithms proved to be more efficient if the number of repetitive sequences were more and the compression ratio was more than huffman algorithm.

**PROPOSED COMPRESSION ALGORITHM:**

The proposed compression algorithm has the following procedures.

**ENCODER**

One of the two main algorithms is the encoding module. The encoding module is split into four separate sections as follows.

**Unique Words Identification:**

Initially the input text document is scanned and the unique words in the document are identified along with their frequency and length. In our algorithm we allocate codes for the words directly as opposed to allocating codes for each letter in conventional Huffman. These unique words so obtained are sent as input to the k-means partitioning algorithm.

\subsubsection{K-Means Partitioning Algorithm}

The k-means partitioning algorithm is a clustering technique which groups together elements having more similarity into the same cluster wherein different clusters have minimum inter cluster similarity. In our algorithm we take the frequency and length of a word as a similarity measure for clustering. The words which are taken as input get clustered based on the Euclidean distance between the frequency and length of each word. Words with minimum Euclidean distance are sent into the same cluster.

The formula used for k-means clustering technique is

$$\sqrt{(f\_1-f\_2)^2 + (l\_1-l\_2)^2}$$

where $f\_1$ and $f\_2$ are frequencies of the two words to be compared and,

$l\_1$ and $l\_2$ are lengths of the two words to be compared.

**Huffman Encoding**

In our algorithm we have used 2 levels of Huffman encoding. The first level is the inner Huffman encoding in which codes are assigned for each word in a given cluster. The initial step of Huffman encoding involves arranging the words in a cluster in descending order of their frequencies. For two words with same frequency we use lexicographical ordering. After this a Huffman code tree is constructed for each cluster and codes are allocated for all the words. However when Huffman is applied individually for each cluster there is a possibility that the two words in different clusters may have the same codeword. To eliminate this controversy we go for appending cluster identification bits for each word based on the cluster to which it belongs to. Therefore a second level Huffman is applied for the clusters to allocate codeword for each cluster. These codes are appended to the beginning of each word's codeword to give the absolute code for each word. This final codeword set is taken as the final codeword table.

**Encoding The Input Text**

The final encoding is done by replacing each word in the input file with its corresponding code from the codeword table generated above. The compressed file so generated is found to have a better compression ratio when compared to conventional Huffman Encoding.

**DECODER**

The decoding process involves reading the encoded file character by character and comparing with the code table containing words and their codes. Since Huffman Encoding generates only prefix codes the tracing process for each word is done quite efficiently. The decoded file can thus be obtained by simple mapping of binary code values to their appropriate words.

What’s next? Does anyone have the format?

Should we explain why we are using Clustering techniques as opposed to others and why exactly Huffman among other variable length encoding?

I dont think this is necessary. we just chose seminal algorithms. other algorithms will be our future work.

Other compression techniques are Huffman(Canonical & Adaptive), Arithmetic coding, Context tree weighting, Burrows-Wheeler transform, LZ77, LZ78, LZW, LZF, PPMD, Run Length Encoding

Dictionary type : Byte pair encoding, DEFLATE, LZ77, LZ78, LZW, LZMA

write about these in related work.

PAPER FLOW

Topic

Author

Abstract

Introduction

Related Work/ Literature Summary - work on this first.

Problem Definition \*

Algorithm Design (Pseudocode)

Actual Algorithm and its trace

Experimental results

conclusion

references

**REFERENCES**

1. Jiwaei Han - Data Mining Concepts : Book
2. Naren Ramakrishnan : Data Mining from Serendipity to science

<http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=781632&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxpls%2Fabs_all.jsp%3Farnumber%3D781632>

3. Huffman - original

DAVID A. HUFFMAN, ASSOCIATE, IRE

*PROCEEDINGS OF THE I.R.E.*

A Method for the Construction of Minimum-Redundancy Codes\*

September, 1952

4. Shannon - original

C E Shannon, A mathematical theory of communication,*Bell Sys. Tech. J.*, Vol. 27, pp. 398–403, July 1948.

5. Fano - original

R M Fano,*The transmission of information*, Technical Report No. 65, Research Laboratory of Electronics, M.I.T., Cambridge, Mass., 1949.

6. Kmeans - original

MacQueen, J. B. (1967). [*Some Methods for classification and Analysis of Multivariate Observations*](http://projecteuclid.org/euclid.bsmsp/1200512992). Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability **1**. University of California Press. pp. 281–297. [MR](https://en.wikipedia.org/wiki/Mathematical_Reviews) [0214227](https://www.ams.org/mathscinet-getitem?mr=0214227). [Zbl](https://en.wikipedia.org/wiki/Zentralblatt_MATH) [0214.46201](https://zbmath.org/?format=complete&q=an:0214.46201). Retrieved 2009-04-07.

7. Adaptve huffman encoding algorithm 1

J. S. Vitter, "[Design and Analysis of Dynamic Huffman Codes](http://www.cs.duke.edu/~jsv/Papers/Vit87.jacmACMversion.pdf)", Journal of the ACM, 34(4), October 1987, pp 825–845.

8. LZ77

[Ziv, Jacob](https://en.wikipedia.org/wiki/Jacob_Ziv); [Lempel, Abraham](https://en.wikipedia.org/wiki/Abraham_Lempel) (May 1977). "A Universal Algorithm for Sequential Data Compression". [*IEEE Transactions on Information Theory*](https://en.wikipedia.org/wiki/IEEE_Transactions_on_Information_Theory) **23** (3): 337–343.[doi](https://en.wikipedia.org/wiki/Digital_object_identifier):[10.1109/TIT.1977.1055714](https://dx.doi.org/10.1109%2FTIT.1977.1055714). [CiteSeerX](https://en.wikipedia.org/wiki/CiteSeer#CiteSeerX): [10.1.1.118.8921](http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.118.8921).

9. LZ78

[Ziv, Jacob](https://en.wikipedia.org/wiki/Jacob_Ziv); [Lempel, Abraham](https://en.wikipedia.org/wiki/Abraham_Lempel) (September 1978). "Compression of Individual Sequences via Variable-Rate Coding". [*IEEE Transactions on Information Theory*](https://en.wikipedia.org/wiki/IEEE_Transactions_on_Information_Theory) **24** (5): 530–536.[doi](https://en.wikipedia.org/wiki/Digital_object_identifier):[10.1109/TIT.1978.1055934](https://dx.doi.org/10.1109%2FTIT.1978.1055934). [CiteSeerX](https://en.wikipedia.org/wiki/CiteSeer#CiteSeerX): [10.1.1.14.2892](http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.14.2892).

number your references following this and mark the place where they are used in the paper in [ ].

Forgy, E. W. (1965) Cluster analysis of multivariate data: efficiency vs interpretability of

classifications. Biometrics 21, 768–769.

Hartigan, J. A. and Wong, M. A. (1979). A K-means clustering algorithm.

Applied Statistics 28, 100–108.

Lloyd, S. P. (1957, 1982) Least squares quantization in PCM. Technical Note, Bell Laboratories.

Published in 1982 in IEEE Transactions on Information Theory 28, 128–137.

MacQueen, J. (1967) Some methods for classification and analysis of multivariate observations.

In Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability,

eds L. M. Le Cam & J. Neyman, 1, pp. 281–297. Berkeley, CA: University of California Press.

https://stat.ethz.ch/R-manual/R-devel/library/stats/html/kmeans.html