ICM Compression System Depending On Feature Extraction

[1.]Assist.Prof.Dr.AlaaKadhim F, Prof. Dr. Ghassan H. AbdulMajeed[2], RashaSubhi Ali [3]

[1] University of technology, Iraq

[2] Ministry of higher education, Iraq

[3] University of technology, Iraq

Abstract

The goals of data compression is the task of providing space on the hard drive, and reduce the use of bandwidth in the transmission network and transfer files quickly. In this paper intelligent techniques are used as ways of lossless data compression. These methods applied using clustering techniques. Clustering is one of the most important data mining techniques. The proposed system presents a new algorithm used to determine the best compression method. This algorithm, called the ideal compression method (ICM). In addition to ICM system there are three clustering algorithms were used compress the databases and also we proposed new decompression algorithm was used to recover the original databases. The compression algorithms are different in the method of selecting attributes (which parameters are used as centers of clusters). These algorithms are improved K-means, k- mean with the medium probability and k-mean with maximum gain ratio. ICM used to determine the best one of these three algorithms that can be used to compress the database file. The main objective of this research is to select optimal compression method for each database. The ICM algorithm depends on the property of min-max removal and also depends on a number of conditions that used to determine the best compression method. This method continued in removing min-max column until remain one column. The residua column data was used to specify best compression method. The standard k-means algorithm suffering from several drawbacks such as it was dealt with only numerical data types, number of clusters needed to be specified by the used and the centers of clusters selected randomly. The three compression algorithms are modification algorithms to the standard k-means algorithm. The modification was proposed in selecting the number of the clusters centers, specifying the number of the clusters and in dealing with the data types. Several experiments on different databases have been performed and the results are compared. The results shows that the maximum saving percentage is 98% minimum saving percentage is 61%, maximum decompression time is around 14 minutes, minimum decompression time is 6 seconds, maximum compression time is 17 minutes, minimum compression time is 5 seconds, maximum compression size is 1073 kilobytes and minimum compression size is 7 kilobytes. This research is organized as follow. Section one shows the introduction, Section two explains major clustering techniques, Section four shows the methodology of compression and decompression algorithms and system structure, Section five presents experiments and results and section six offers the conclusion.

Keywords:compression algorithm, ICM system, improved k-means algorithm, modified improved k-means algorithms, decompression algorithm.

1. Introduction

Data compression purposes are to reduce the number of bits used to store or transmit data [1]. Data is compressed by decreasing its redundancy, but this makes the data less reliable, more prone to errors. Data compression is popular for two reasons: (1) People like to collect data and hate to throw anything away. (2) People hate to wait a long time for data transfers [2]. This process may be useful if one wants to save the storage space. For example if one wants to store a 4MB file, it may be best to compress it to a smaller size to provide the storage space. Also compressed files are much more easily exchanged over the internet since they upload and download much faster [3]. Data Compression is essentially defined as a technique to reduce the size of data by applying different methods that can either be Lossy or Lossless [4]. A lossy data compression method is one where the retrieved data after decompression may not be exactly same as the original data, but is "close enough" to be useful in particular purpose. After one applies lossy data compression to a message, the message can never be recovered exactly as it was before it was compressed. Some data has been lost. Because lossy compression cannot be retrieved exactly the original message, it is not a good method of compression for critical data, such as textual data, data base. Lossless data compression is a technique that allows the use of data compression algorithms to compress the text data, database data and also allows the exact original data to be reconstructed from the compressed data. This is in inversion to the lossy data compression in which the exact original data cannot be reconstructed from the compressed data. Lossless compression is used when that it is important to the original data and the decompressed data to be exactly identical. The advantage of lossless methods over lossy methods is that Lossless compression results are in a nearer representation of the original inputted data. The performance of algorithms can be compared using the parameters such as Compression Ratio, decompression time, compression time and Saving Percentage. Lossless data compression works by finding frequent patterns in a message and encoding those patterns in an efficient manner. For this reason, lossless data compression is also referred to as redundancy reduction. Because redundancy reduction is dependent on patterns in the message, it does not work well on random messages. Lossless data compression is ideal for text [3].

2.Clustering Algorithms

Huge amounts of data are around us in our world, raw data that is mainly intractable for human or manual applications. So, the analysis of such data is now a necessity. The World Wide Web (WWW), business related services, and networks for science or engineering, among others, are continuously generating data in exponential growth since the development of powerful storage and connection tools. This enormous data growth does not easily allow to useful information to be extracted automatically. Data mining (DM) It is an important process where the methods are used to extract valid data patterns. Many people distinguish DM as synonym for the Knowledge Discovery in Databases (KDD) process, while others see DM as the main steps of KDD[5]. This section discusses the problems, assumptions and requirements of majoring clustering algorithms; this section focuses on k-mean clustering algorithm. Clustering technique is one of the most important data mining techniques.

Cluster analysis includes algorithms and methods aims for grouping or classifying objects. The aim of cluster analysis, or unsupervised classification, is to divide data (objects, instances, items) into groups (clusters) so that items belonging to the same group are more similar than items belonging to distinct groups [6]. Usually clustering techniques divided into hierarchical and partitioning and density based clustering. Hierarchical clustering is a method of cluster analysis which seeks to construct a hierarchy of clusters [7]. There are two approaches to creating hierarchical clustering models, both of which have very simple shapes:

Agglomerative clustering: A bottom-up approach which starts with many small clusters and iteratively merges selected clusters until a single root cluster is reached.

Divisive clustering: A top-down approach which starts with a single root cluster and iteratively partitions existing clusters into sub clusters [8].

Partitioning methods seek to get a single partition of the inputted data and dividing it into a fixed number of clusters [9]. The most popular partitioning algorithm is k-means, which will be presented in the next section [10]. The partitioning methods usually result in a set of M clusters, each object belonging to one cluster. Each cluster can be represented by a centroid of a cluster; this is some kind of summary description of all the objects contained in a cluster [7]. One of the most important clustering algorithm can be explained in the next section, because of we needed it in the research.

2.1 K-means Algorithm

K-means is one of the most commonly used clustering techniques due to its simplicity and speed. It partitions the data into k clusters by assigning each object to its closest cluster centroid (the mean value of the variables for all objects in that particular cluster) based on the distance measure used [11]. The algorithm provides a simple and understandable method for classifying data into a fixed a priori number of groups [10]. This algorithm is fast for large data sets, which are common in segmentation.

The basic algorithm for k-means works as follows:

1. Choose the number of clusters, k.
2. Select k cluster centroids (e.g., randomly chosen k objects from the data set).
3. Assign each object to the nearest cluster centroid.
4. Recompute the new cluster centroid.
5. Repeat step 3 and 4 until the convergence criterion is met (e.g., the assignment of objects to clusters no longer changes over multiple iterations) or maximum iteration is reached [11]. Usually, the K-means algorithm criterion function depends on square error criterion, which can be defined as:

$$E = \sum_{j=1}^{k} \sum_{\substack{i=1 \\ x_i \epsilon c_j}}^{n} \|x_i - m_j\|^2 \qquad (1)$$

In which, E is total square error of all the objects in the data cluster, xi is the vector of the i-th element of the dataset, mi is mean value of cluster Ci (x and m are both multi-dimensional). The function of this criterion is to make the generated cluster be as compacted and independent as possible [12].

A. Advantages of K-mean Clustering

K-means algorithm is considered one of the most important partitioning approaches. An advantage of such partitioning approaches is that it can undo previous clustering steps (by iterative relocation). K-mean clustering algorithm is simple, flexible, high-speed performance, measurable and efficient in large data collection.K-mean clustering algorithm is easy to understand and implements [13].

B.Weakness of K-mean Clustering

k-means algorithm in spite of its benefits it was suffered from several dis advantages such as it was applicable only to objects in a continuous n-dimensional space (numerical data type only),it was needing to specify k, the number of clustersin advance and Selecting optimal number of cluster for problem is difficult k.in addition , it was sensitive to noisy data and outliers, Not suitable to discover clusters with non-convex shapes [14].

3.Design and Methodology

The compression process is an important concept. It was used for the purposes of reducing the file size, providing memory space and providing fast file transfer. The proposed system consists of several phases. These phases comprise:

1. Input the database file
2. Apply the ICM algorithm
3. Compress the database file with best compression method
4. Returning th/e results(compressed file)

Phase 1: This phase includes selecting the database file is input to ICM and the best identified compression algorithm. The inputted database could contained any type of data (numerical data, text data,….etc).

Phase 2: This phase discusses the selection the optimal compression method. The ICM algorithm was used for this purpose. The ICM algorithm depends on extracted features. The feature was extracted from the inputted database by analyzing the inputted database. Also, this algorithm depends on several conditions used to specify best compression method (k-mean, k-mean with medium probability and k-mean with maximum gain ratio). Each one of these algorithms depends on specific calculations to specify the centers for the clusters.

Phase 3: The ideal compression method was determined in the previous phase. The best compression method can be applied to the inputted database in this phase.

Phase 4: The results of the compression algorithm are recorded at this phase. The final results represent compressed file contains database data in clusters form. The proposed system can be explained in the following architecture.

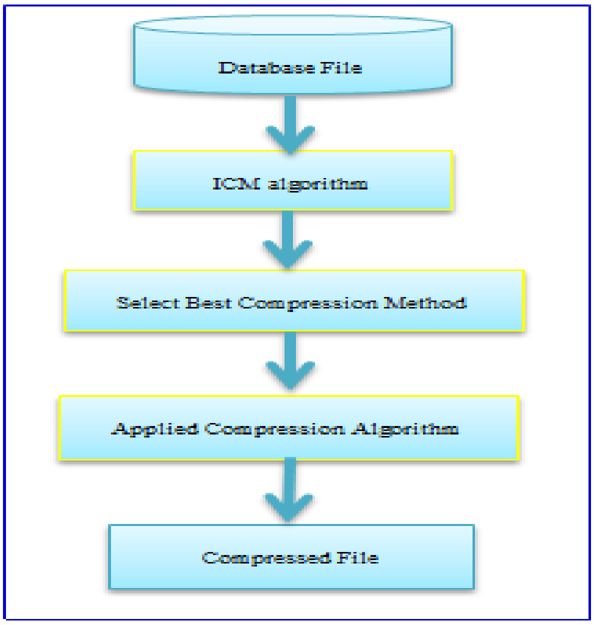

Figure 1: Proposed System Architecture

4.The Proposed System

The system consists of two parts. One of them for decision maker and this part is depended on analysis the database and features extraction and the second part for compression and decompression operation.

4.1 The ICM System Architecture

The new proposed ICM system is used to select best one of compression algorithms which can be used for the compression purpose. The structure of the ICM system starts by taking the database file that represents the input to the compression algorithm and then the ICM begins analysis to this database. In this system the database features have been extracted after analyzing the database. These features comprise number of columns, number of rows, number of clusters and total words length in each column. These features have been used to specify best compression method, which can be used to compress the database. There are several conditions also used in determining the optimal compression method.

The ICM system depends on min-max removal. This means remove the columns with minimum number of clusters then remove the columns with maximum number of clusters until remain one column. The information of this column used to specify the best compression method according to the following formulas:-

$$\text{Result}=\frac{n+l}{r} \quad \ldots\ldots (2)$$

N=number of clusters
L=words length in each column
R= number of rows in the database file
C= number of columns

$$Y=r*c \quad \ldots.. (3)$$

The results of this system were compared after doing manual execution for each one of the compression algorithm. The ideal compression method was specified according to the decompression time, compressed file size and the compression time sequentially.The ICM steps can be explained in the following algorithm:

```
Input: database file to be compressed
Output: best compression method
Begin:
Step1: feature extraction (find the number of columns, number of rows, number of clusters and total words length in each column).
Step2: while number of columns >1 do
Find the column with minimum number of clusters let it x
Find the column with maximum number of clusters let it z
Remove x
Remove z
Until still one column with its features
End while
Step3: apply equations number (2, 3)
Step4: evaluation conditions
If results<1 then k-mean with medium probability is best
Call k-mean with medium probability algorithm
Else if L<y then k-mean with maximum gain ratio is best
Call k-mean with maximum gain ratio algorithm
Else k-mean is best
Call k-mean algorithm
End if
End.
```

The decision maker was made during database analysis depending on the database features. This means decision maker was not made after doing the compression operation using the three clustering compressing algorithm. The ICM system can be explained in the following figure.

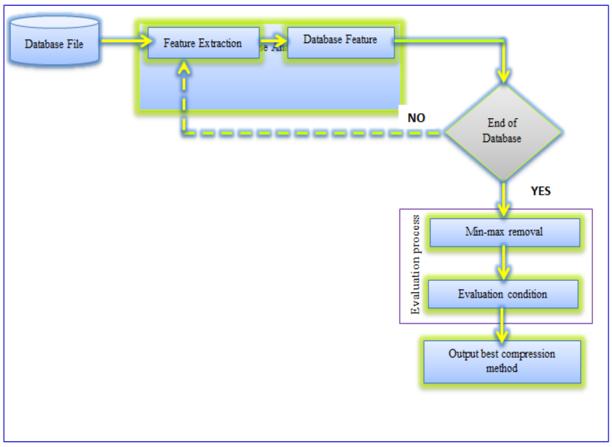

Figure 2: Structure Of The ICM Algorithm

4.2 Compression Algorithms

In this research there are three intelligent compression algorithms two of them new proposed compression algorithm and the other one from my previous practice. The general structure for the compression algorithm explained in the below figure.

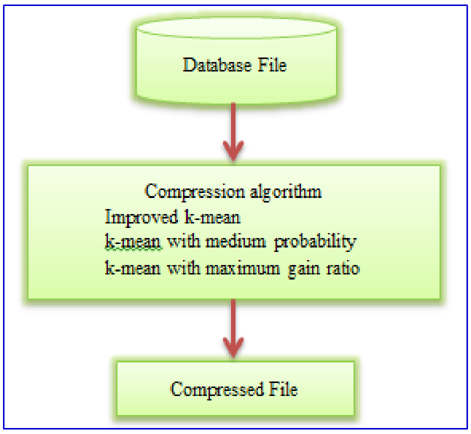

Figure 3: Compression Operation

The numbers of clusters in this research have been specified in dynamic form. The general structure for these algorithms is same in side of distributing data on the clusters, but the difference in specifying the centers of the clusters, and the number of the clusters.

The clusters centers are specified in each algorithm in different ways depending on special calculations for each algorithm. In these algorithms were solving the problems of conventional k-means algorithm. These algorithms are dealing with any type of data (text, numerical, categorical, date and etc.). The conventionalk-means algorithm was dealing with numerical data type only.The traditionally k-means algorithm has been updated by using three algorithms to improve the performance and to make it matching with inputted data. These are improved k-means, K-means with medium probability and K-means with maximum gain ratio.The K-means with medium probability produced better classification than the improved k-means algorithm. The clusters size may be balanced not very large and not very small in the same database data. The improved k-means can be produced very large and very small clusters in the same database data. For example using database of 100 records and 6 columns the result of improved k-means algorithm is clusters of size (8, 32, 1, 39, 19, and 1) records. While the K-means with medium probability can be produced clusters of size (17,,4 74,8 ,2 ,4236, 14 … etc.) records and the K-means with maximum gain ratio will be produced clusters of size (2, 1, 1, 5, 6, 1, 2, 2, ….etc.) records. The K-means with maximum gain ratio has been considered as a hybrid approach of clustering and classification method and it was resulted in more accurate classification and a large number of clusters. The improved k-means algorithm produced lesser number of clusters than the two proposed algorithm, while the K-means with medium probability introduced balanced number of clusters. According to the conducted experiments the K-means with maximum gain ratio algorithm has been taken more time for compression operation than the other two algorithms. The time is taken for gain ratio calculation. The modified improved k-means algorithm explained in the following sections.

4.2.1 Improved K-means Algorithm

This is the first compression algorithm used to compress the database. Improved k-means algorithm represents my previous practice. In this algorithm the attributes was selected according to the largest repetition attribute (minimum number of clusters), regardless of the words length that are available in these attributes.

The database was entered to the algorithm; the database was analyzing to extract the important features of that database. Next, repetition for each column is calculated. The results of this calculation were compared and columns with maximum repetition are selected. These attributes were used to generate the centers of clusters. The centers of the clusters and the numbers the clusters determined in dynamic way. Finally the data of remaining columns was distributed on the clusters according to the shared center. The number of records in each cluster is varying from one cluster to another.Improved K-means algorithm can be explained in the following steps:

Input:initially input the database file to be compressed.

Output: is the compressed file.

Begin:

Step1: features extraction (extract database features such as number of columns, number of rows, number of occurrences for each word in each column, and number of clusters).

Step2: selection of the columns with maximum repetition value.

Step3: generate clusters centers from these selected columns.

Step4: find number of the remaining columns (unselected columns) let it X.

Step5: assign each record in X to the cluster that belongs to it according to that clusters centers.

Step6: register the clusters centers and its data to the compressed file.

End.

The new proposed compression algorithms (K-means with medium probability and K-means with maximum gain ratio) were discussed in the next section. The K-means with medium probability and K-means with medium probability algorithms act as a modified algorithm to the traditional k-means and improved k-means algorithm.

4.2.2 K-means With Medium Probability

This algorithm is the second compression algorithm. It was used the maximum and medium values of repetition probability instead of using only maximum repetition values. This algorithm is same as to the previously discussed algorithm (k-means algorithm) in the parts of data analysis and data distribution. But it is different in the selecting the centers of the clusters; because of in this algorithm the medium probability with maximum probability was used determining the columns. Also, there is another similarity the centers of the clusters and the number of the clusters determined dynamically. The specified columns are used to generate the centers of the clusters. Next, the clusters centers and the clusters numbers are extracted which could be used to aggregate the database data. Finally the remaining columns data could be distributed on the extracted centers. The clusters size is not similar. The records (rows) are belonging to one cluster and not belonging to another cluster depending on special identifier.K-means with medium probability of occurrence algorithm illustrated in the following steps:

Input: initially input the database file to be compressed.
Output:compressed file.
Begin:
Step1: features extraction (extract database features such as number of columns, number of rows, number of occurrences for each word in each column, and number of clusters).
Step2: selection of the columns with maximum and medium repetition value.
Step3: generate clusters centers from these selected columns.
Step4: find number of the remaining columns (unselected columns) let it X.
Step5: assign each record in X to the cluster that belongs to it according to that clusters centers.
Step6: register the clusters centers and its data to the compressed file.
End.

4.2.3 K-means With Maximum Gain Ratio

This is the final compression method used in this research to compress the database file. K-means with maximum gain ratio algorithm is hybrid between classification and clustering techniques. It was using the maximum repetition value with maximum gain ratio in determining the columns that are used in clusters centers generation. This is the difference about the previously discussed algorithms. This algorithm similar to the previously discussed compression algorithms in the aspects of data distribution, data analysis and features extraction.

In first step of the extraction features the columns with maximum repetition extracted. The next step, the gain ratio is calculated for the remaining columns except the column that has largest repetition value (minimum number of clusters) and the column that has identifier (primary key).

The gain ratio can be calculated according to these equations:

Pi = number of class (words) occurrence / total number of rows

$$Entropy(S) = \sum i\ Pi \log2\ Pi \ldots\ldots\ldots\ldots\ldots (4)$$

Pi is the probability of class i

The entropy is 0 if all members of S belong to the same class. The range of entropy is 0 (best classification) to 1 (random classification).

$$I\ (S,\ A) = \sum_i \frac{|Si|}{|S|} \times \text{Entropy}\ (Si) \ldots\ldots\ldots (5)$$

Si is the number of rows that contain class i
S is the total number of rows

$$Gain(S,\ A) = Entropy\ (S) - I\ (S,\ A) \ldots (6)$$

K-means with maximum gain ratio can be explained in the following algorithm:

Input: initially input the database file to be compressed.
Output: compressed file.
Begin:
Step1: features extraction (extract database features such as number of columns, number of rows, number of occurrences for each word in each column, and number of clusters).
Step2: calculate gain ratio value for each column except identifier column and the maximum repetition value column (the decision attribute). The gain ratio calculated according to the following steps:
1-Attribute selection
2- Entropy calculated according to eq.4
3-Information Gain calculated according to eq.5
4-Gain Ratio calculated according to eq.6
Step3: selection of the columns with maximum and medium repetition value.
Step4: generate clusters centers from these selected columns.
Step5: find number of the remaining columns (unselected columns) let it X.
Step6: assign each record in X to the cluster that belongs to it according to that clusters centers.
Step7: register the clusters centers and its data to the compressed file.
End.

At this research the three compression algorithms are similar in the behavior of distributing data and analyzing data, but they are different in selecting the centers of the clusters. Also, the ICM algorithm was discussed. It was used for the purpose of determine best compression algorithm and from this the best centers are determined.

4.3 The proposal for Decompression Operation

In this research we proposed a way for storing data after compression the database file which is applicable to open by using one method. This provides easy to use for the beneficiary. Any file arrives to the beneficiary and this file was compressed using the improved k-means algorithm or the modified algorithms, this file can be opened in one behavior.

In the previous section the compression operation was explained. This section studies how to retrieve the original database file by using decompression algorithm. This research explains one decompression algorithm used to decompress the compressed files, which was resulted from applying any one of the three discussed compression algorithm. The decompression algorithm was done in reverse order of the compression operation. The lossless compression algorithm must be used, so decompressed file must be exactly similar to the original database file because of the database contain several data types (text, numerical,…etc).The structure of the decompression operation is described in the figure below.

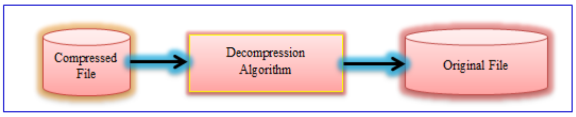

Figure 4: Decompression Operation

Decompression algorithm steps can be explained in the following figure.

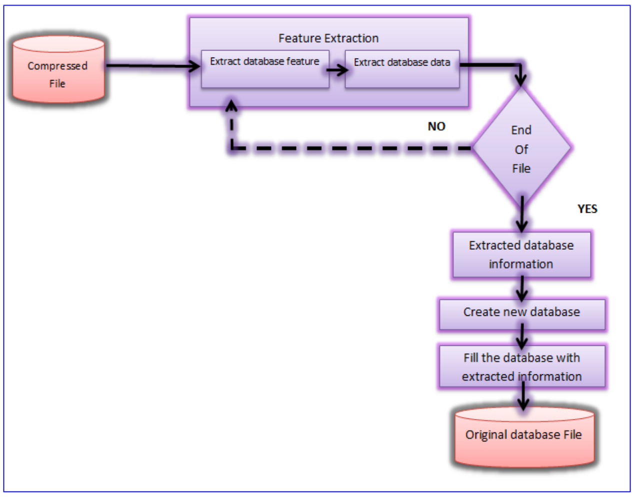

Figure 5: Decompression Algorithm Stages

In decompression operation, firstly the features of database were extracted. Next, the clusters data extracted. This operation continues until reaching the end of the compressed file. After that new database is generated and it is filled with this data. At this algorithm the original database was retrieved. The decompression time was computed and the results are compared.

Decompression algorithm can be explained in the following steps:

Input: initially input the final output from the compression algorithm to be decompressed.

Output: display the final retrieved database (exactly same as original database) to the user

Begin:

Step1: assign the input from step1 to the decompression algorithm (analysis data set) to extract database features and data (columns number, rows number, columns header name and database records).

Step2: create new database file.

Step3: create new table (tables) in the created database file.

Step4: alter table and add the extracted columns to it.

Step5: fill the database by adding the fields to the tables.

End.

5.Results and Discussion

This section discusses the results generated by the proposed system. The proposed system was applied on different data sets for different data types, and was conducted different experiments to specify the performance of the proposed system.

The proposed system performance was evaluated by using four parameters and these are:

1) **Decompression time:** decompression time is defined as the time that was consumed to retrieve the original database.

2) **Compression size:** compression size is defined as the file size after applying compression algorithm on the original file.

3**) Compression time:** compression time is defined as the time consumed to compress the database file.

4) **Saving percentage:** saving percentage is defined as the decreasing percentage of the original file. This computed using the following formulas:

$$\text{Saving percentage}=1-\left(\frac{\text{compressed file size}}{\text{original file size}}\right) * 100 \ ....(7)$$

The proposed system results show in the following tables. The following tables and graphs represent the comparison results for the proposed algorithms four parameters. Notes that the tables contain some terms like:

K-means with med prob=k-means with medium probability,

K-means with gain ratio= k-means with maximum gain ratio.

Table (1) shows the detailed information for the tested datasets such as number of columns and number of rows in each database.

Table 1: Dataset Information

Detail information for tested dataset		
databae name	number of columns	number of rows
dept	6	100
niaid	9	100
LMGIS	12	539
niaid2248	9	2248
hospital	20	498
DWC-job	8	3679
enwiki	12	3253
dept10000	6	10000
bacteria	6	4894

Table 2: Compressed File Size And Original File Size

Compressed fle size				
databae name	orginal DB size	k-means	k-means with med. prob.	k-means with gain ratio
Dept	356	7	7	7
Niaid	372	12	12	12
LMGIS	564	107	107	105
niaid2248	900	239	269	283
hospital	932	130	130	132
Enwiki	1236	197	202	218
DWC-job	1440	441	381	381
dept10000	2512	619	503	701
Bacteria	2748	979	1073	979

The above table explains a comparison for compressed file size results after applying compression algorithms and the original file size.

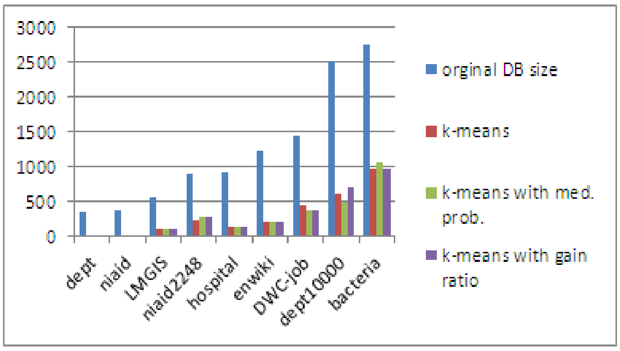

Figure 6: Compressed File Size Vs Original File Size

Table (3) presents the saving percentage when using the three compression algorithms separately.

Table 3: Resulted Saving Percentage Of The Proposed Algorithms

saving percentage			
databae name	k-means	k-means with med. prob.	k-means with gain ratio
dept	98.00%	98.00%	98.00%
niaid	97.00%	97.00%	97.00%
LMGIS	81.00%	81.00%	81.40%
niaid2248	73.00%	70.00%	69.00%
hospital	86.10%	86.10%	85.80%
enwiki	84.00%	83.60%	82.40%
DWC-job	69.40%	73.50%	73.50%
dept10000	75.40%	80.00%	72.10%
bacteria	64.00%	61.00%	64.00%

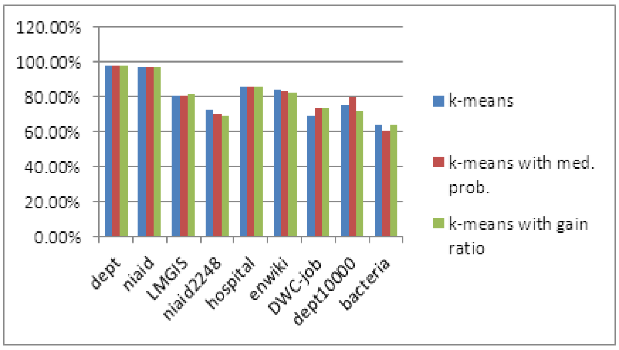

Figure 7: Saving Percentage Corresponding to the Original Database

In figure (7) and table (3) it is clear that the maximum saving percentage is 98% and minimum saving percentage is 61% for the databases that have been tested.

Table 4: Retrieved Database Size

Retrivedvs original DB size					
databae name	orginal DB size	retrived DB size	k-means	k-means with med. prob.	k-means with gain ratio
Dept	356	240	7	7	7
Niaid	372	240	12	12	12
LMGIS	564	485	107	107	105
niaid2248	900	768	239	269	283
hospital	932	768	130	130	132
enwiki	1236	1034	197	202	218
DWC-job	1440	1148	441	381	381
dept10000	2512	1696	619	503	701
bacteria	2748	2611	979	1073	979

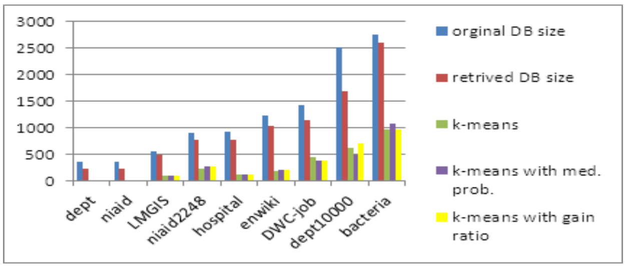

Figure 8: Retrieved and Compressed Database Size Vs Original Database Size

From table (4) and figure (8) we notice the recovered database size is smaller than the original database size because of the original database consists of storages area for deleted fields. When decompress the compressed files, the recovered database is exactly same as the original database but with no spaces for deleted fields. So the files size after decompressed it may be lesser than the original files size because of providing the deleted fields spaces.

Table (5) presents the time that has been possessed for compression operation by using the three compression algorithms.

Table 5: Compression Time

Compression time			
databae name	k-means	k-means with med. prob.	k-means with gain ratio
dept	5.3	5.8	13
niaid	8	7	25
LMGIS	7	7	145
niaid2248	10	8	820
hospital	7	6	180
enwiki	9	7	1020
DWC-job	11	5	600
dept10000	8	19	145
bacteria	7	160	2705

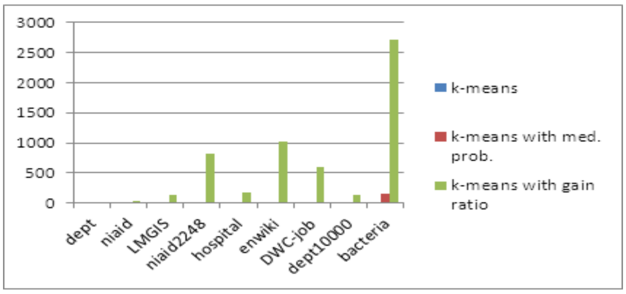

Figure 9: Illustration ofthe Compression Time forthe Three Compression Algorithm

Table 6: Decompression Time

Decompression time			
databae name	k-means	k-means with med. prob.	k-means with gain ratio
Dept	10	11.3	11.3
Niaid	14.2	14.1	14.4
LMGIS	10	6	7
niaid2248	240	228	256
Hospital	90	90	94
Enwiki	480	480	480
DWC-job	410	393	400
dept10000	840	820	478
Bacteria	352	353	352

The table above defines the decompression time consumed to decompress the compressed file. The time is measured in seconds for both compression and decompression operation.

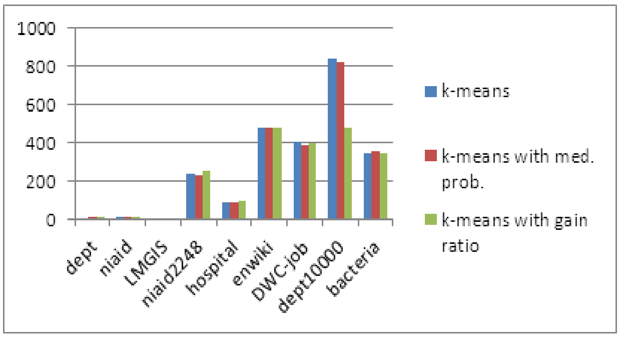

Figure 10: Decompression Time

The above graph is made on the basis of table (6). The above table and graph are discussed the comparison of decompression time which is consumed to recover the original file that was compressed by using the proposed algorithms.

Finally the ICM information and determined method is explained in the below table and figure corresponding to the database size.

Table 7: Best Selected Depends on the ICM System

7					
databae name	number of column	number of rows	orginal DB size	ICM time n seconds.millisecnds	ICM selection
dept	6	100	356	1.3	k-means
niaid	9	100	372	1.6	k-means with medium probability
LMGIS	12	539	564	1.7	k-means with maximum gain ratio
niaid2248	9	2248	900	1.8	k-means with medium probability
hospital	20	498	932	1.8	k-means with medium probability
enwiki	12	3253	1236	1.5	k-means with medium probability
DWC-job	8	3679	1440	4	k-means with medium probability
Dept10000	6	10000	2512	1.8	k-means with medium probability
bacteria	6	4894	2748	2	k-means

From the above table it is clear the best compression algorithm is achieved by applying the ICM system. Results have been compared manually by using the three compression algorithms. The ICM system has achieved dynamic selection for the best compression method.

6.Conclusion

1. The traditional k-means algorithm has dealt with numerical data type only and needing to specify number of the clusters from the user. These limitations of traditional k-means algorithm have been overcome by the proposed system. The proposed clustering algorithms deals with any type of data and do not need specifying the number of clusters from the user, instead of that, it is determined the number of clusters dynamically. The decompression algorithm operated in dynamic way.
2. The problem with conventional compression algorithm each compression algorithm was needed to a decompression algorithm (two compression algorithms needed two decompression algorithm). We proposed one decompression algorithm which is used to decompress all the files that were compressed in any one of the three compression algorithms. This feature provided facilities to the user which simplifying to them decompression operation. Various experiments have been made on different databases. These databases consist of different data types such as (number, text, date….etc.).
3. The ICM method can be solved the problem of selecting best compression method. Instead of applying each algorithm on the database and comparing the results for them and this would be taken much time. The ICM system takes a small time to select best compression algorithm depending on the database analysis and not on the applying the three compression algorithms. From the experiments ICM shows very good results in terms of dynamic selecting of optimal compression algorithm. The ICM system was based on several conditions to achieve optimal selection. The performance of compression algorithms is evaluated depending on some measurements such as decompression time, compressed file size, saving percentage and compression time sequentially.
4. The ICM method was solved the problem of selecting best centers for the database clusters by determining best compression clustering method.

Reference

[1]. Mark Nelson, “The Data Compression Book”, 2nd edition, Cambridge, MA 02139, IDG Books Worldwide, Inc.).

[2]. David Salomon,Giovanni Motta ,David Bryant, “Data Compression The Complete Reference”, 4th Edition, Springer-Verlag London, 2007.

[3]. Amandeep Singh Sidhu,Er. MeenakshiGarg, “Research Paper on Text Data Compression Algorithm using Hybrid Approach”, C.S.E. & Guru Kashi University, Talwandi Sabo, Bathinda, Punjab,

India, International Journal of Computer Science and Mobile Computing, IJCSMC, Vol. 3, Issue. 12, December 2014, pg.01 – 10.

[4]. R.S. Brar, B. Singh, “A survey on different compression techniques and bit reduction Algorithm for compression of text data”, International Journal of Advanced Research In Computer Science and Software Engineering (IJARCSSE) Volume 3, Issue 3, March 2013.

[5]. Salvador García ,JuliánLuengo, Francisco Herrera, “Data Preprocessing in Data Mining”, Intelligent Systems Reference Library Volume 72, Springer International Publishing Switzerland 2015.

[6]. Oleg Granichin ,Zeev (Vladimir) Volkovich,DvoraToledano-Kitai, “Randomized Algorithms in Automatic Control and Data Mining ABC”, Intelligent Systems Reference Library Volume 67, Springer-Verlag Berlin Heidelberg 2015.

[7]. PradeepRai,Shubha Singh, “A Survey of Clustering Techniques”, International Journal of Computer Applications (0975 – 8887) Volume 7– No.12, October 2010.

[8]. PawełCichosz,“ Data mining algorithms : explained using R”, Department of Electronics and Information Technology Warsaw University of Technology Poland, published 2015by John Wiley & Sons, Ltd.

[9]. Francisco de A.T. de Carvalho , Yves Lechevallier, Filipe M. de Melo, “Partitioning Hard Clustering Algorithms Based On Multiple Dissimilarity Matrices”, Elsevier, 6 September 2011.

[10]. VICTOR FIGUEROA,” Clustering Methods Applied To Wikipedia”, UniversitéLibre De Bruxelles, 2010 - 2011.

[11]. Jared Dean, “Big Data, Data Mining, and Machine Learning”, Value Creation for Business Leaders and Practitioners, Wiley & SAS Business Series, 2014.

[12]. Chunfei Zhang, Zhiyi Fang, “An Improved K-means Clustering Algorithm”, Journal of Information & Computational Science 10: 1 (2013) 193–199.

[13]. PriteshVora,BhaveshOza, “A Survey on K-mean Clustering and Particle Swarm Optimization”, International Journal of Science and Modern Engineering (IJISME) ISSN: 2319-6386, Volume-1, Issue-3, February 2013.

[14]. JiaweiHan,MichelineKamber ,Jian Pei, “Data Mining Concepts and Techniques”, 3rd edition, University of Illinois at Urbana-Champaign & Simon Fraser University, 2013.