International Journal of Applied Engineering Research ISSN 0973-4562 Volume 10, Number 14 (2015) pp 34945-34951

HB-K Means: An Algorithm for High Dimensional Data Clustering Using Bisecting K-Means

Aparna K
Associate Professor,
Department of Master of Computer Applications
BMS Institute of Technology & Management
Bengaluru, Karnataka
aparnak.bmsit@gmail.com

Mydhili K Nair
Associate Professor,
Department of Information Science and Engineering
M S Ramaiah Institute of Technology
Bengaluru, Karnataka
mydhili.nair@gmail.com

Abstract- In today's world, the volume of data is growing at a greater pace. Apparently, most of the data mining applications deal with these voluminous datasets. It is not easy to develop efficient clustering algorithms because of curse of dimensionality. The quality of clusters varies as the dataset tends to become larger. The most widely used clustering algorithm is the K-Means algorithm. But it comes with several drawbacks including time consumption and expensive computation. To overcome these limitations, a novel partitional clustering algorithm is proposed called HB-K Means (High Dimensional Bisecting K-Means) algorithm. In this algorithm, the high dimensional dataset is converted to Attribute Frequency Matrix and it is then clustered using the modified Bisecting K-Means algorithm. The experimentation is carried out on two large datasets of UCI machine learning repository and the proposed algorithm has achieved better clustering accuracy and lesser computation time compared to the traditional K-Means clustering algorithm.

Keywords: Data Mining, K-Means, Bisecting K-Means, High Dimensional Data Clustering, HB-K-Means (High dimensional Bisecting – K Means).

Introduction

Clustering is the process of grouping objects based on their similarities. A comprehensive study and analysis of the different clustering techniques is given in [1]. This similarity is measured based on some of the distance measures. Clustering is different from classification in that they do not possess any class labels [2, 3] and [4]. Cluster analysis is generally categorized as unsupervised learning unlike classification which is categorized as supervised learning. The aim of clustering is to organize the dataset in a proper way by forming appropriate clusters [5]. In other words, the goal is to partition the huge dataset into more organized and structured clusters which helps to mine useful information. Clustering has been an emerging area of research for many years now and has found extensive applications in wide and diverse areas [6–7]. There are several types of clustering algorithms that includes partitional clustering, hierarchical clustering, density based and grid based clustering etc. [8].

Traditional clustering algorithms become computationally expensive when the data set that needs to be clustered is voluminous. The reasons why the dataset could be large are: 1) the number of elements in the dataset may be huge, 2) each element may contain many attributes 3) the number of clusters to be formed may be more [9]. A distance metric such as Euclidean distance is generally used in many of the clustering algorithms in order to partition the dataset and consequently to form the clusters. Because of the huge accumulation of data in the recent years, most of the current datasets are of high dimensions. As a result, the similarity between objects is not very valid leading to inaccurate results [10]. Though high dimensional datasets provide an insight into useful patterns, they also come with lot of computational challenges [11]. This has given rise to feature selection/extraction problem [12] which is a very significant aspect of knowledge discovery and data mining.

Most of the major clustering algorithms suffer when applied on high dimensional datasets mainly because of the curse of dimensionality. As the number of dimensions in a dataset increases, the inaccuracy of the cluster formation also increases. This is because the distance measures do not tend to give valid results [13]. Also, in a high dimensional dataset, many of the dimensions are usually irrelevant leading to the presence of outliers in the clusters [14]. Recent literature on high dimensional data reveals techniques of forming clusters for a particular subset of dimensions [15]. It can be inferred from such techniques that each cluster is particular to a specific group of dimensions thus solving the problem of sparsity in high dimensional dataset to a greater extent [16].

One of the most widely used partitional clustering algorithms is the K-Means algorithm which takes the number of clusters as the input. The resultant clusters may not be precise most of the time because of the presence of outliers [17]. Moreover, the result is dependent on the initial seed values which can lead to non-discriminant outputs. The different approaches that can be adopted in order to improve this are, either choose the initial values arbitrarily or to pick up the first k data points [18]. Another approach could be to identify diverse sets of initial points and to select the most optimal set out of it. Moreover, as the dimensionality of the data increases, the number of distance calculations also increases exponentially leading to inaccurate results [19].

In this paper, we have developed a novel partitional-based clustering algorithm, known as HB K-Means (High dimensional Bisecting K-Means) for clustering high dimensional data. This algorithm consists of four major steps such as, attribute frequency matrix formation, binary matrix formation, outlier detection and cluster formation. Initially, attribute frequency matrix is constructed by finding the dense regions and their location in each dimension. Using the attribute frequency matrix, the outlier data points are detected and then the clustering is done by making use of Bisecting K-Means clustering which is an extension of the traditional K-

Means algorithm. At last, the experimentation is carried out using two different datasets to prove the efficiency of the proposed HB K-Means clustering algorithm.

The rest of the paper is organized as follows. Section 2 describes the related research in the field of high dimensional data clustering. Section 3 portrays the motivating algorithms for the proposed approach. Section 4 explains in detail the various steps we have adopted for our proposed approach and section 5 illustrates the experimentation with results and discussion of the proposed approach using various datasets. Finally, the conclusion is given in section 6.

Review of Related Works

A number of researches have been presented for clustering of data records using partitional clustering. Recently, the clustering of large datasets has received a great deal of attention among the data mining researchers. A brief review of some of the recent researches on clustering of data by using partitional methods is presented here.

The authors in [20] have come out with a new approach for formation of clusters. The paper deals with the continuous data sets. The dataset is initially sorted in a particular order. All the datasets that were closest to the given centroid was considered as part of a single cluster. Their experimental results have shown that the new approach performs better in terms of consuming less computational time. The authors also suggest that the approach can be extended for discrete datasets.

In [21], H. S Behera et al have proposed a new model for clustering called IHKMCA (Improved Hybridized K-Means Clustering Algorithm) in which the Canonical Variate Analysis is applied to the high dimensional dataset in order to reduce its dimensionality without affecting the original data. To this reduced low dimensional data set, the authors have applied the Genetic Algorithm in order to obtain the initial centroid values. K-Means algorithm is then applied to this modified reduced data set. The experiments have shown better results compared to the traditional algorithms in terms of time complexity.

In [22], the authors have chosen the technique of Principal Component Analysis for dimensionality reduction. They have applied PCA on the microarray gene expression data in order to reduce it to a low dimensional dataset. The experiments were carried out for both with and without reduction datasets by taking a valid number of Principal Components for analysis. The approach has shown promising results.

In [23 - 24] the authors have used a new algorithm in order to initialize the clusters before applying the K-Means algorithm. PCA technique is used initially for dimensionality reduction. From the reduced dataset, the initial seed values are selected which is a pre-requisite for the K-Means algorithm. These initial centroid values are then normalized using Z-score before giving them as input to the K-Means algorithm. The new algorithm is tested using some of the benchmark datasets and the results show better performance in terms of efficiency.

The authors in [25] have applied PCA for dimensionality reduction. Bisecting K-Means algorithm is then applied to the reduced data set since this requires no initialization of the seed values to start off with the cluster formation. This approach has shown significant improvement in time computation and accuracy of cluster formation. But it also shows that the efficiency tends to decrease as the size of the dataset increases.

Motivating Algorithms

Several algorithms were proposed in the literature for clustering of data. Here, we have presented two main partitional clustering algorithms such as K-Means and Bisecting K-Means that have motivated us to continue the research in this direction.

K-Means

K-Means clustering [26] is a well-known partitional algorithm that aims to partition N data points into *k* clusters, in which each data point is relevant to the cluster that has the nearest centroid value. Assume the data base, D_P has N data points $z_1, z_2, \ldots, z_N$ such that all data points exist inside the region, then the problem of computing the minimum variance clustering of the dataset into *k* clusters is none other than that of finding *k* centroid $\{c_i\}(i = 1, 2, \ldots, k)$ in R. The function to be minimized can be written as,

$$\frac{1}{N}\sum_{j=1}^{N}\left[\min_{i} d^2(z_j, c_i)\right]$$

where $d(z_j, c_i)$ is the Euclidean distance between z_j and c_i. The points $\{c_i\}$ $(i = 1, 2, \ldots, k)$ are named as cluster centroids. The problem in the above equation is to find out *k* cluster centroids, in which the average squared Euclidean distance (mean squared error, MSE) between a data point and its adjacent cluster centroid is minimized [27].

Steps:

- Initialize K centroids, one for each cluster.
- Calculate the distance $E_D(z_j, c_i)$ (given in definition 1) of every K centroids from data points z_j in D_P.
- Assign data point z_j to cluster C_i whose distance value is small compared to other clusters.
- Update the k centroids based on the memberships of the new clusters.
- Repeat step 2 to step 4, until there is no movement of the data points between the clusters.

Bisecting K-Means

K-Means is the most popular iterative centroid-based divisive algorithm and it is the most reputed and old method for clustering data points. In this section, we describe about an alternate of the K-Means algorithm, the Bisecting K-Means algorithm [7]. Most of the algorithms, which are similar to the K-Means algorithm, are concentrating on the centroid calculation for the clustering process. In this section, we use the Bisecting K-Means, which serves as an enhancement to the normal K-Means algorithm. The processing in the Bisecting K-Means algorithm can be given as follows:

Steps:

- Choose the centroid of a data 'M'. This can be given as, $w = \sum_{i=1}^{N} M_i$
- Randomly select a point c_l as one centroid
- Compute the second centroid, as $c_r = w - (c_l - w)$
- Divide the data *M* into two sub-clusters according to the following rule
- $x_i \in M_L, if \| x_i - c_l \| \leq \| x_i - c_r \|$
- $x_i \in M_R, if \| x_i - c_l \| > \| x_i - c_r \|$
- Re-compute the centroid by taking the mean value of the sub-clusters and repeat the above steps until, $w_l = c_l$ and $w_r = c_r$

Using the above procedure, two clusters are formed. Then, one of the clusters from those two clusters is chosen for splitting into two clusters based on the maximum number of data points. The same procedure is repeated until we obtain the desired number of clusters.

HB-K-Means: An Algorithm for High Dimensional Data Clustering using Bisecting K-Means

Most clustering algorithms do not work efficiently for data sets in high dimensional spaces because of the inherent sparsity of data. Consequently, a clustering algorithm is often preceded by feature selection whose goals are to find the particular dimensions in which points in the data set are correlated. Technology advances have made data collection easier and faster, resulting in larger, more complex datasets with many objects and dimensions [28]. In high dimensional data, many of the dimensions are often irrelevant and can confuse clustering algorithms by hiding clusters in noisy data. Bisecting K-Means is an efficient algorithm for the clustering of high dimension data. It starts with one large cluster of all the data points and divides the whole dataset into two clusters. K-means algorithms run multiple times to find a split that produces maximum intra cluster similarity. Then, the cluster with largest size is picked to split further. This cluster can be chosen based upon minimum intra cluster similarity also. This algorithm is run k-1 times to get k clusters. This algorithm performs better than regular K-Means because Bisecting K-Means produces almost uniform sized clusters, while in regular K-Means, there can be notable difference between sizes of the clusters. As small clusters tend to have high intra cluster similarity, large clusters have very low intra cluster similarity and overall intra cluster similarity decreases [29].

Due to efficiency of Bisecting K-Means algorithm in clustering, we develop a partition-based projection algorithm for data clustering of high dimensional data. The proposed clustering algorithm contains four major steps: 1) attribute frequency matrix formation, 2) binary matrix formation, 3) outlier detection and 4) discovering of clusters. The above processes are summarized in the following block diagram shown in Fig.1. From the figure, it can be understood that we take a high dimensional data as input and an attribute frequency matrix is created from it. Afterwards, a binary matrix is created from it and then, the outlier detection is done after which the clustering is done using modified Bisecting K-Means algorithm to get k-clusters as output.

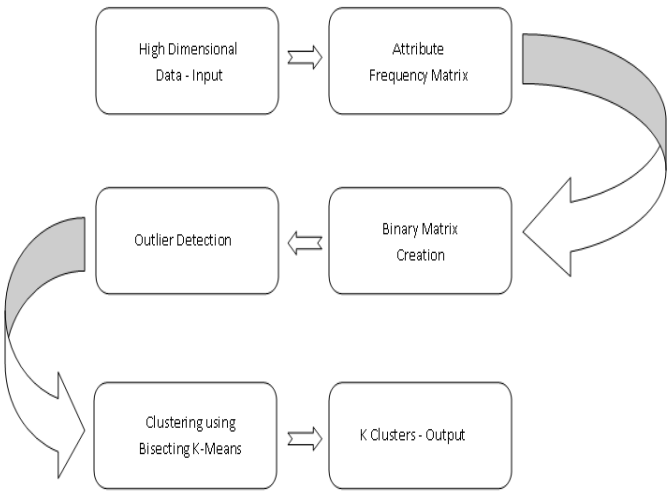

Fig. 1. Block Diagram of the proposed HB-K Means algorithm

Attribute Frequency Matrix Generation

Initially, the proposed algorithm obtains the high dimensional data set D as input that can be represented as D_{ij}. Here, 'i' belongs to the data points in the input data and 'j' belongs to the dimensions of the input data. Let us assume *n* is the number of data points and *m* is the number of attributes in the input dataset. The first step of the proposed algorithm is to find the attribute frequency matrix that is used to detect the outlier as well as forming the binary matrix. The following sequences of steps are used in finding the attribute frequency matrix that is represented as AFM_{ij}. The size of the attribute frequency matrix and the data matrix that is input to the proposed algorithm are equal.

Step 1: The mean value of every column of the data matrix is computed using the following formula.

$$M_j = \frac{1}{n}\sum_{i=1}^{n} D_{ij}$$

Step 2: The Standard Deviation (SD) of every column of the data matrix is also computed using the following formula.

$$S_j = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(D_{ij} - M_j)^2}$$

Step 3: The range is computed for every data value of data matrix d_{ij} based on the data value given in the data matrix and SD of every column of the data matrix.

$$R_{ij} = \{(d_{ij} - S_j), (d_{ij} + S_j)\}$$

Step 4: Then, the attribute frequency of each data element d_{ij} is found out using the following equation.

$$AFM_{i,j} = \frac{f_{i,j}(f_{i,j} - 1)}{n(n-1)}$$

where, *n* is the total number of data points and f_{ij} is the number of data points corresponding to the attribute 'j' that come under the range value R_{ij}.

Step 5: The calculated attribute frequencies for every data element are represented in a matrix known as the *attribute frequency matrix, AFM_{ij}*.

Binary Matrix Generation

The *Binary Matrix* is a new concept introduced in our paper to improve the efficiency of clustering. For finding the similarity between two data objects, we have used the binary matrix that contains 0 and 1. The binary matrix is exactly the same as that of input data matrix and also, it reflects the weights of every data element of input matrix. Based on the importance of every data element specified in the binary matrix, we calculate the distance value which is the empowered advantage of the proposed algorithm in clustering process. The procedure for constructing the binary matrix B_{ij} is given as follows:

1) The mean of all the attributes (columns) in the attribute frequency matrix AFM_{ij} is found out.

$$MA_j = \frac{1}{n}\sum_{i=1}^{n} AFM_{ij}$$

2) The obtained mean value of every column is now multiplied with the preset binary threshold value, ϕ.

$$K = MA_j * \phi$$

3) If the value of data element in the attribute frequency matrix is greater than the product obtained K, then the data element is replaced with one. Otherwise, the value is zero.

$$b_{ij} = \begin{cases} 1 & ; \ if \ \ AFM_{ij} > K \\ 0 & ; \ if \ AFM_{ij} \le K \end{cases}$$

Outlier Detection

The outlier detection is a process which removes the non-relevant data points from the input data matrix with the help of attribute frequency matrix. The outlier detection is done with a row wise operation of attribute frequency matrix. For finding the outlier data points, a *central tendency value* for every row is computed. The central tendency value CT_j can be calculated using the following formula.

$$CT_j = \frac{1}{m}\sum_{j=1}^{m} AFM_{ij}$$

Based on the above equation, we obtain *central tendency value, CT_j* for every row and those values are sorted in a descending order. Then, the top-L points are considered as outlier and they are removed from the data matrix. Once the outlier data points are removed, the number of data points is reduced from 'n' points to 'n-L' and the outlier removed data matrix represented as D_F, is subjected to the clustering process.

Modified Bisecting K-Means Clustering

In this step, the clustering process is done with the help of data matrix achieved from the previous step. Here, we make use of Bisecting K-Means clustering procedure to mine the clusters from the data matrix obtained from the outlier detection process. Here, at first, two clusters are mined from the final data matrix using the following procedure and then, the cluster that has more data points is given to clustering procedure again for mining two clusters from it. This procedure is repeated until we obtain the desired number of clusters.

Step 1. Randomly choose a data point, say $m_L \in R^P$ from the final data matrix D_F. This data point is used to calculate the mean value of the data points for splitting. We are considering this data point as a randomly selected point for calculating the Euclidean distance between the value and each and every data point.

Step 2. Calculate the mean $m_R \in R^P$ as $m_R = 2M - m_L$. The m_R value is calculated and is considered as a point for calculating the distance with other data points. The M value needed for calculating the mean is as follows.

$$M = \frac{1}{n-L}\sum_{i=1}^{n-L} D_{Fi}$$

Step 3. Split the data matrix $D_F = [D_{F1}, D_{F2}, \ldots. D_{F(n-L)}]$ into two sub-clusters D_L and D_R, in accordance with the following condition:

$$\begin{cases} D_{Fi} \in D_L & if \quad E_M(D_{Fi}, m_L) \le E_M(D_{Fi}, m_R) \\ D_{Fi} \in D_R & if \quad E_M(D_{Fi}, m_L) > E_M(D_{Fi}, m_R) \end{cases}$$

where

$$E_M(D_{Fi}, m_L) = \sqrt{\begin{matrix} D_{Fi}{}^{(1)}, m_L{}^{(1)})^2 * bi^{(1)} + (D_{Fi}{}^{(2)}, m_L{}^{(2)})^2 * bi^{(2)} + \\ \ldots\ldots + (D_{Fi}{}^{(n-L)}, m_L{}^{(n-L)})^2 * bi^{(n-L)} \end{matrix}}$$

$$E_M(D_{Fi}, m_R) = \sqrt{\sum_{j=1}^{m} (D_{Fi}{}^{j} - m_R{}^{j})^2 * bi^{j}}$$

Using the distance calculated using the above formula, we partition the data matrix into two clusters D_L and D_R.

Step 4. Compute the centroids of $D_L$and D_R using the following equation.

$$C_L = \frac{1}{N_L}\sum_{j=1}^{N_L} D_{L,j} \qquad C_R = \frac{1}{N_R}\sum_{j=1}^{N_R} D_{R,j}$$

The calculated centroid values of the resulting clusters are used for determining the stopping criteria for the proposed algorithm. The centroid values are calculated as the mean value of the points included in the respective clusters.

Step 5. If $m_L = c_L$ and $m_R = c_R$, stop, else, let $c_L = m_L$, $c_R = m_R$ and go to Step 3. If the centroid C_L matches with the randomly selected data point m_L and the centroid value C_R matches with the calculated mean value of data points m_R, the algorithm stops its iteration. Otherwise the algorithm will return to the step 3, and again split the newly formed clusters each into two clusters and continue the process until the stopping criteria is reached.

Results and Discussion

This section presents the experimental results of the proposed algorithm and the detailed discussion of the results obtained. Here, two different datasets are used for experimentation and the performance of the proposed algorithm is compared with the previous algorithms in terms of time and clustering accuracy.

Experimental Setup

The proposed approach of High Dimensional Data Clustering Using Bisecting K-Means is implemented in MATLAB. Here, we have tested our proposed approach using the Spambase and Pen-Based Recognition of Handwritten Digits Data sets. The testing was done on a computer with Intel Core 2 Duo CPU with clock speed 2.2GHz and 4 GB RAM.

Description of Datasets

Spambase Dataset (database 1): Spambase dataset [30] is a collection of spam and non-spam emails obtained from various users. The spam emails were obtained from the postmaster and users who marked their mail as spam. The non-spam emails were obtained from filed work and personal emails. Spambase dataset is a multivariate dataset which consists of 4601 instances with 57 attributes.

Pen-based recognition of handwritten digits dataset (database 2): Pen-based recognition of handwritten digits dataset [31] is a digit database formed by collecting 250 samples from 44 writers. The samples written by 30 writers are used for training, cross-validation and writer dependent testing, and the digits written by the other 14 are used for writer independent testing. This database is also available in the UNIPEN format. The dataset is a multivariate dataset which have 7494 instances with 16 attributes.

Comparative Analysis

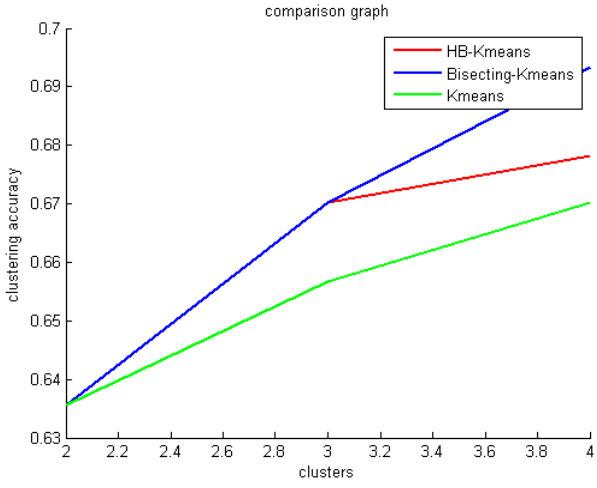

FIG. 2A. Comparative analysis of database 1 in terms of clustering accuracy

In this section, we compare the efficiency of different clustering algorithms namely K-Means, Bisecting K-Means and our proposed approach (HBK-Means) with respect to clustering accuracy and computation time. First, we consider the clustering accuracy against the number of clusters for different clustering approaches. Figure 2a shows a plot of clustering accuracy against the number of clusters for the spambase dataset. From the graph, it can be seen that HB K-Means and Bisecting K-Means have a higher clustering accuracy than K-Means clustering algorithm for any number of clusters. And the HB K-Means has higher accuracy than Bisecting K-Means for small number of clusters. Figure 2b portrays a plot of clustering accuracy against the number of clusters for the Pen-based recognition of handwritten digits dataset. From the graph, we can see that HB K-Means and Bisecting K-Means have a higher clustering accuracy than K-Means clustering algorithm for any number of clusters and the HB K-Means has higher accuracy than Bisecting K-Means for small number of clusters.

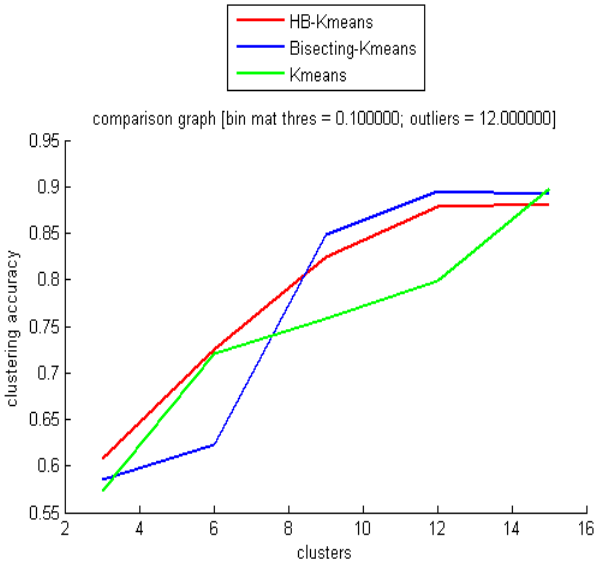

FIG. 2B. Comparative analysis of database 2 in terms of clustering accuracy

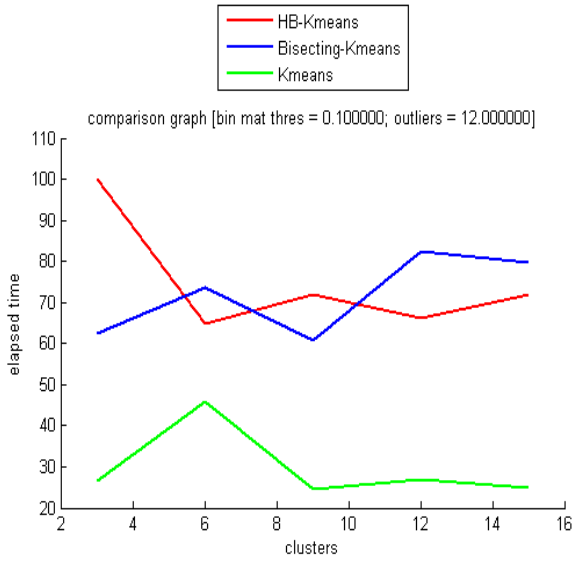

FIG. 3A. Comparative analysis of database 1 in terms of computation time

Next, we compare the computation time for various algorithms. Figure 3a shows the graph for the same for the

spambase dataset. It is seen that the computation time for K-Means algorithm is the same irrespective of the number of clusters. Also, it is the least. On the other hand, the computation time for Bisecting K-Means and HB K-Means algorithm decreases with increase in number of clusters. It can be seen that the computation time for Bisecting K-Means and HB K-Means is the same for small number of clusters. But it can be seen that as the number of clusters increases, the HB K-Means requires lesser computation time than the Bisecting K-Means algorithm and hence is an advantage.

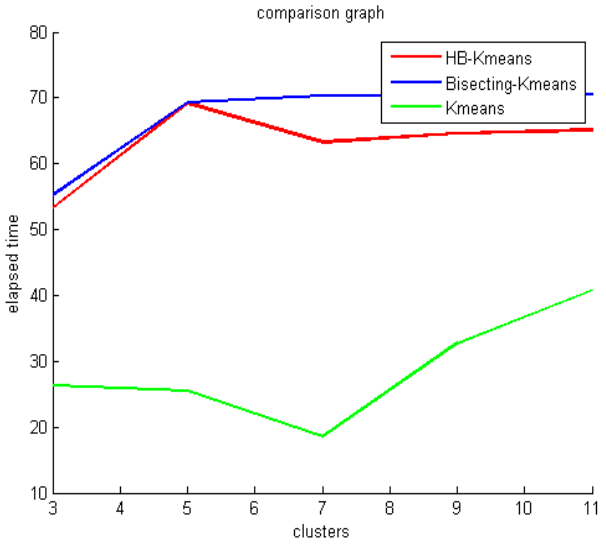

FIG. 3B. Comparative analysis of database 2 in terms of computation time

Figure 3b depicts the graph for the same for the Pen-based recognition of handwritten digits dataset. It is seen that the computation time for K-Means algorithm is the same irrespective of the number of clusters. And, it is the least. On the other hand, the computation time for Bisecting K-Means and HB K-Means algorithm decreases with increase in number of clusters. We could see that the computation time for Bisecting K-Means and HB K-Means is identical for small number of clusters. But it can be seen that as the number of clusters increases, the HB K-Means needs lesser computation time than the Bisecting K-Means algorithm and therefore is a benefit.

From the overall analysis, we can conclude that our proposed HB K-Means clustering approach is an efficient algorithm.

Conclusion

In this paper, we have proposed an efficient partitional clustering algorithm, known as HB-K Means for high dimensional data with the help of Bisecting K-Means algorithm. In the proposed method, the high dimensional dataset was converted to an attribute frequency matrix, which is then transformed to a binary matrix. Subsequently, the outliers are removed from the input data using the resulting attribute frequency matrix. The data which we got after the outlier removal process is then subjected to the modified Bisecting K-Means clustering to mine the cluster. The modified Bisecting K-Means is applied repeatedly until the required number of clusters is obtained. The algorithm is tested with various datasets such as Spambase and Pen-Based Recognition of Handwritten Digits Datasets and the results obtained shows that the algorithm provided better performance than the existing techniques. However, some constraints can be incorporated in order to obtain optimal results and hence improve the performance of the novel algorithm. This can be considered for future work.

References

[1] Aparna K and Mydhili K Nair, “Comprehensive Study and Analysis of Partitional Data Clustering Techniques”, International Journal of Business Analytics, Vol 2, Issue 1, pp. 23 – 38, January-March 2015.

[2] Xiyu LIU, Xinjiang XIE and Wenping WANG, "A Projection Clustering Technique Based on Projection", Journal of Service Science & Management, Vol. 2, pp. 362-367, 2009.

[3] Kiri Wagsta, Claire Cardie, Seth Rogers, Stefan Schroedgl, "Constrained K-means Clustering with Background Knowledge", In proceedings of the Eighteenth International Conference on Machine Learning, pp. 577-584, 2001.

[4] D. Sculley, "Web-Scale K-Means Clustering", In Proceedings of the 19th international conference on World Wide Web, pp. 1177-1178, 2010.

[5] Kevin Y. Yip, David W. Cheung and Michael K. Ng, "HARP: A Practical Projected Clustering Algorithm", IEEE Transactions On Knowledge And Data Engineering, Vol. 16, No. 11, November 2004.

[6] Chris Ding and Xiaofeng He, "Cluster merging and splitting in hierarchical clustering algorithms", In Proceedings of the IEEE International Conference on Data Mining, 2002.

[7] Sergio M. Savaresi and Daniel L. Boley, "On the performance of bisecting K-means and PDDP", In Proceedings of the First SIAM International Conference on Data Mining, 2001.

[8] Carlotta Domeniconi and Sheng Ma, "Subspace Clustering of High Dimensional Data", In Proceedings of International Conference on Data Mining, pp. 517-521, 2004.

[9] Andrew McCallum, Kamal Nigam and Lyle H. Ungar, "Efficient Clustering of High-Dimensional Data Sets with Application to Reference Matching", In Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 169 - 178, 2000.

[10] Mohamed Bouguessa and Shergrui Wang, "Mining Projected Clusters in High-Dimensional Spaces", IEEE Transactions on Knowledge and Data Engineering, Vol. 21, No. 4, pp. 507 - 522, 2008.

[11]John stone I.M, "Non-Linear dimensionality reduction by LLE", 2009.

[12]Jain .A.K, Murty.M.N and Flynn.P.J, "Data Clustering: A Review", ACM Computer Surveys, Vol.31, No.3, 1999.

[13]Michael Steinbach, George Karypis and Vipin Kumar, "A Comparison of Document Clustering Techniques", In Proceedings of KDD Workshop on Text Mining, 2000.

[14]Lance Parsons, Ehtesham Haque and Huan Liu, "Subspace Clustering for High Dimensional Data: A Review", Sigkdd Exploration, Vol. 6, No. 1, pp. 90-105, 2004.

[15]R. Agrawal, J. Gehrke, D. Gunopulos and P. Raghavan, "Automatic Subspace Clustering of High Dimensional Data for Data Mining Applications", ACM SIGMOD Conference, Vol. 27, No. 2, 1998.

[16]Charu C. Aggarwal, Jiawei Han, Jianyong Wang and Philip S.Yu, "A Framework for Projected Clustering of High Dimensional Data Streams", In Proceedings of the 30th VLDB Conference, pp. 852 - 863, 2004.

[17]Rajashree Dash, Debahuti Mishra, Amiya Kumar Rath and Milu Acharya, "A hybridized k-means clustering approach for high dimensional dataset", International Journal of Engineering, Science Technology, Vol. 2, No. 2, pp. 59-66, 2009.

[18]Xu R. and Wunsch D, "Survey of clustering algorithms", IEEE Transaction Neural Networks, Vol. 16, No. 3, pp. 645-678, 2005.

[19]Kohei Arai and Ali Ridho Barakbah, "Hierarchical K-means: an algorithm for centroids initialization for K-means", Reports of the Faculty of Science and Engineering of Saga University, Vol. 36, No.1, 2007.

[20]K. Prasanna, M. Sankara Prasanna Kumar and G. Surya Narayana, " A Novel Benchmark K-Means Clustering On Continuous Data", International Journal On Computer Science And Engineering (IJCSE), Vol. 3, No. 8, pp. 2974-2977, 2011.

[21]H.S Behera, Rosly Boy Lingdoh And Diptendra Kodamasingh, "An Improved Hybridized Kmeans Clustering Algorithm (IHKMCA) For Highdimensional Dataset & It's Performance Analysis", International Journal On Computer Science And Engineering (IJCSE), Vol. 3, No. 3, pp. 1183-1190, 2011.

[22]P. Valarmathie, Dr M V Srinath and K. Dinakaran, "An Increased Performance Of Clustering High Dimensional Data Through Dimensionality Reduction Technique", Journal Of Theoretical And Applied Information Technology, pp. 731-733, 2009.

[23]D.Napoleon And S.Pavalakodi, "New Method For Dimensionality Reduction Using K-Means Clustering Algorithm For High Dimensional Data Set", International Journal Of Computer Applications, Vol. 13, No.7, Pp. 41-46, 2011.

[24]P.Prabhu and N.Anbazhagan, "Improving The Performance Of K-Means Clustering For High Dimensional Data Set", International Journal On Computer Science And Engineering (IJCSE), Vol. 3, No. 6, Pp 2317-2322, 2011.

[25]R.Indhumathi And Dr.S.Sathiyabama, "Reducing And Clustering High Dimensional Data Through Principal Component Analysis", International Journal Of Computer Applications, Vol. 11, No. 8, Pp. 1–4, 2010.

[26]J. B. MacQueen, "Some Method for Classification and Analysis of Multivariate Observations", Proc. of Berkeley Symp. on Mathematical Statistics and Prob., Berkeley, U. of California Press, Vol. 1, pp. 281-297, 1967.

[27]Shai Ben-David, Dávid Pál and Hans Ulrich Simon, "Stability of K-Means Clustering", In Proceedings of the 20th annual conference on Learning theory, pp. 20-34, 2007.

[28]Dipti Patra," Integration of FCM, PCA and Neural Networks for Classification of ECG Arrhythmias", IAENG International Journal of Computer Science, Vol. 36, No.3, 2005.

[29]Ming-Chuan Hung, Jungpin Wu, Jin-Hua Chang and Don-Lin Yang, " An Efficient *k*-Means Clustering Algorithm Using Simple Partitioning", Journal Of Information Science And Engineering, Vol. 21, pp. 1157-1177, 2005.

[30]Spambase DataSet from "http://archive.ics.uci.edu/ml/datasets/Spambase"

[31]Pen-Based Recognition of Handwritten Digits Data Set from "http://archive.ics.uci.edu/ml/datasets/Pen-Based Recognition of Handwritten Digits".