International Journal of Technology (2016) 4: 691-700
ISSN 2086-9614

INCORPORATING STABILITY AND ERROR-BASED CONSTRAINTS FOR A NOVEL PARTITIONAL CLUSTERING ALGORITHM

K. Aparna[1*], Mydhili K. Nair [2]

[1] *Department of Computer Applications, BMS Institute of Technology & Management, Yelahanka, Bengaluru – 560064, Karnataka State, India*
[2] *Department of Information Science & Engineering, M S Ramaiah Institute of Technology, MSR Nagar, Mathikere, Bengaluru – 560054, Karnataka State, India*

(Received: August 2015 / Revised: November 2015 / Accepted: March 2016)

ABSTRACT

Data clustering is one of the major areas in data mining. The bisecting clustering algorithm is one of the most widely used for high dimensional dataset. But its performance degrades as the dimensionality increases. Also, the task of selection of a cluster for further bisection is a challenging one. To overcome these drawbacks, we developed a novel partitional clustering algorithm called a HB-K-Means algorithm (High dimensional Bisecting K-Means). In order to improve the performance of this algorithm, we incorporate two constraints, such as a stability-based measure and a Mean Square Error (MSE) resulting in CHB-K-Means (Constraint-based High dimensional Bisecting K-Means) algorithm. The CHB-K-Means algorithm generates two initial partitions. Subsequently, it calculates the stability and MSE for each partition generated. Inference techniques are applied on the stability and MSE values of the two partitions to select the next partition for the re-clustering process. This process is repeated until K number of clusters is obtained. From the experimental analysis, we infer that an average clustering accuracy of 75% has been achieved. The comparative analysis of the proposed approach with the other traditional algorithms shows an achievement of a higher clustering accuracy rate and an increase in computation time.

Keywords: Bisecting K-Means; Constraints; High dimensionality; Mean Square Error (MSE); Partitional clustering; Stability

1. INTRODUCTION

The process of grouping objects based on their similarities is commonly known as Clustering. A comprehensive study and analysis of the different clustering techniques is given in (Aparna & Nair, 2015(a)). Various distance measures are used as metrics in order to measure this similarity. Classification, also known as supervised learning, is different from clustering, since clustering does not possess any class labels (Liu et al., 2009; Wagsta et al., 2001; Sculley, 2010). The aim of clustering is to organize the dataset in a proper way by forming appropriate clusters (Yip et al., 2004). 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 (Ding & Xiaofeng, 2002; Savaresi & Daniel, 2001). There are several types of clustering algorithms that includes partitional clustering, hierarchical clustering, density-based and grid-based clustering, etc. (Domeniconi & Sheng, 2004).

[*] Corresponding author's email: aparnak.bmsit@gmail.com, Tel. +91-9986735823

Permalink/DOI: http://dx.doi.org/10.14716/ijtech.v7i4.1579

When the data set that needs to be clustered is voluminous, traditional clustering algorithms become computationally expensive. The dataset could become very large due to a number of reasons, such as the number of elements in the dataset may increase, the number of attributes in each element may be more and the number of clusters to be formed may also be more (McCallum et al., 2000). Euclidean distance is generally used as a distance metric in many of the clustering algorithms in order to partition the dataset and consequently to form the clusters. Most of the current datasets are of high dimensions because of the huge accumulation of data in the recent years. As a result, the similarity between objects is not very valid, leading to inaccurate results (Bouguessa & Shergrui, 2008).

In this paper, we have developed a novel partitional clustering algorithm called CHB-K Means (Constraint-based High dimensional Bisecting K-Means) to form clusters out of high dimensional data. This novel algorithm is an extension of the HB-K-Means algorithm (Aparna & Nair, 2015b). The HB-K-Means algorithm has an initial pre-processing step of forming a weighted frequency matrix, followed by the creation of a binary matrix. This binary matrix helps in detecting outliers in order to improve the performance of the traditional algorithms. This outlier free final data matrix is then given as input to the traditional Bisecting K-Means algorithm for clustering, thereby increasing the accuracy of the clusters. In order to further improve upon the performance of HB-K-Means algorithm, this paper deals with the incorporation of two important constraints, namely the Stability measure and the Mean Square Error resulting in the CHB-K-Means algorithm.

Prasanna et al., (2011) and Behera et al., (2011) have come out with a new approach for the formation of clusters. These two papers deal with the continuous data sets in which the Canonical Variate Analysis is applied, resulting in a new model. The Principal Component Analysis (PCA) technique is used for dimensionality reduction which is further normalized using Z-score before giving them as input to the K-Means algorithm. (Valarmathie et al., 2009; Napolean & Pavalakodi, 2011). Face clustering in videos has also been focused on. (Wu et al., 2013). Given the number of detected faces from real-world videos, all faces have been partitioned into K disjoint clusters. As a result, many pairwise constraints between faces were easily obtained from the temporal and spatial knowledge of the face tracks. These constraints were effectively incorporated into a generative clustering model based on the Hidden Markov Random Fields (HMRFs). Within the HMRF model, the pairwise constraints are augmented by label-level and constraint-level local smoothness to guide the clustering process. The parameters for both the unary and the pairwise potential functions are learnt by the simulated field algorithm, and the weights of constraints can be easily adjusted. Furthermore, an efficient clustering framework has been introduced, especially for face clustering in videos. Considering that faces in adjacent frames of the same face track are very similar, the framework is applicable to other clustering algorithms to significantly reduce the computational cost. Experiments on two face data sets from real-world videos demonstrate the significantly improved performance of the algorithm over state-of-the-art algorithms. Gu et al., (2013) have proposed a new semi-supervised spectral clustering method, i.e., SSNCut, for clustering over the LC similarities, with two types of constraints: must-link (ML) constraints on document pairs with high MS (or GC) similarities and cannot-link (CL) constraints on those with low similarities. They have empirically demonstrated the performance of SSNCut on MEDLINE document clustering, by using 100 data sets of MEDLINE records. Experimental results show that SSNCut outperformed a linear combination method and several well-known semi-supervised clustering methods, being statistically significant.

2. METHODOLOGY

2.1. HB-K Means: An Algorithm for High Dimensional Data Clustering using an Enhanced Bisecting K-Means algorithm

The steps involved in the proposed approach are summarized in the block diagram shown in Figure 1 below. In order to arrive at the Final Data Matrix, three main steps are applied to the high dimensional data as discussed below. The high dimensional data initially undergoes some preprocessing steps like text formatting. Thus, the formatted data is then used to arrive at the weighted frequency matrix. The weighted frequency matrix is calculated based on the mean, variance and standard deviation of the input matrix. Later the weighted matrix is used to create the binary matrix. The binary matrix is considered for a further process called outlier detection, resulting in the final data matrix which will be free of outliers. Finally, the Bisecting K-Means clustering is applied to the final data matrix in order to obtain the HB-K-Means algorithm (Aparna & Nair, 2015b).

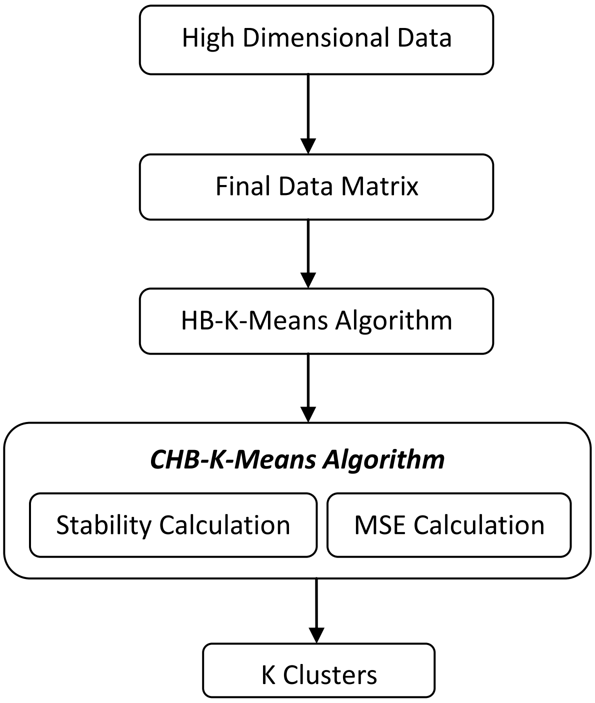

Figure 1 Block Diagram of the proposed algorithm

2.1.1. Initial input matrix generation

The initial processing phase of the proposed approach is the input matrix generation. This input matrix is the main input to the proposed HB-K-Means algorithm. This phase consists mainly of the construction of three matrices. The three matrices are the functional units of the input matrix generation. The key matrix among the three matrices is the matrix called the weighted attribute matrix. The other matrices are derived from the key weighted attribute matrix. The matrices generated are listed below:

a. Weighted Attribute matrix
b. Binary matrix
c. Final data matrix

(a) Weighted Attribute matrix

The process of generation of the weighted attribute matrix starts by selecting the dataset D of dimension n x m. 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 weighted frequency matrix that is used for detecting the outliers as well as for forming the binary matrix. The following sequences of steps are used in finding the attribute frequency matrix that is represented as WAM_{ij}. The size of the attribute frequency matrix and the size of input data matrix of the proposed algorithm are equal.

The initial process of the method is to find the mean value of every column of the data matrix and it is computed using the following formula:

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

Once the mean is calculated, the next step is to compute the standard deviation (SD). The SD of every column of the data matrix is also computed using the following formula:

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

Then, the range is computed as shown below for every value of the data matrix based on the data value given in the data matrix and SD of every column of the data matrix.

$$R_{ij} = \{(d_{ij} - S_j), (d_{ij} + S_j)\} \tag{3}$$

After that the attribute frequency, f_{ij} of each data element is calculated. The attribute frequency f_{ij} is then used for constructing the weighted frequency matrix, which is an important parameter in the proposed approach. The weighed frequency matrix is constructed with the help of following equation:

$$WAM_{ij} = \frac{f_{ij}(f_{ij} - 1)}{n(n-1)} \tag{4}$$

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. The calculated attribute frequency for every data element is stored in the weighted attribute matrix, WAM_{ij}.

(b) 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 also reflects the weights of every data element of the input matrix. Based on the importance of every data element specified in the binary matrix, we calculate the distance value that is the empowered advantage of the proposed algorithm in the clustering process. The procedure for constructing the binary matrix B_{ij} is that the mean of all the attributes (MA_j) in the weighted attribute matrix, WAM_{ij}, is found initially with the help of following equation. Later, the obtained mean value of every column is multiplied with the preset binary threshold value, $\emptyset$.

$$MA_j = \frac{1}{n}\sum_{i=1}^{n} WAM_{ij} \tag{5}$$

$$K = MA_j * \phi \tag{6}$$

If the value of the data element in the weighted frequency matrix is greater than the product obtained, then the data element is replaced with one (1). Otherwise, the value is zero (0).

$$b_{ij} = \left\{ \begin{matrix} 1 & ; \ if \ \ WAM_{ij} > K \\ 0 & ; \ if \ WAM_{ij} \leq K \end{matrix} \right\} \tag{7}$$

The generated binary matrix is then used for improving the clustering efficiency.

(c) Outlier detection
The outlier detection is a process which removes the non-relevant data points from the input data matrix with the help of the attribute frequency matrix. The outlier detection is done by a row wise operation on the attribute frequency matrix. The calculation of the outlier is executed using the value called the outlier detector (OD). This value is calculated through a row wise operation with the help of following equation:

$$OD = \frac{1}{m} \sum_{j=1}^{m} WAM_{ij} \tag{8}$$

According to the equation, all the rows are selected and the OD value is calculated. The OD values are arranged in ascending order. We assume a threshold called "L", which is considered as a limiting point for the outliers. Then, out of the total L points, which are expected to be outliers, the top L points are removed as outliers 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 is represented as D_F, then the final data matrix, is subjected to the clustering process.

Finally the clusters are formed by executing the Bisecting K-Means algorithm.

2.2. Constraint-based High dimensional Bisecting K-Means (CHB-K-Means) Clustering

The cluster centroid has to be optimized for getting better results, which is an add-on in Constraint-based HB-K-Means algorithm. There are two constraints added in the proposed methodology - the Stability and Mean Square Error (MSE). The stability is a measure to improve the quality of cluster. The proposed method deals with improving the stability of K-Means clustering by generating stable clusters according to K value. For instance, if the K value is set to 2, there will be two possibilities for cluster formation, i.e. either horizontal partitioning or vertical partitioning. So it can be accounted that either of the clusters will be with relevant number of data. In other cases, if the K value is set to 5, the number of data points in each cluster can vary, i.e. it can be either too low or too high, which will result in instability. So in order to rectify the problem, we initiate a stability score for each cluster centroid. The cluster centroid with highest stability will be considered and the rest is discarded.

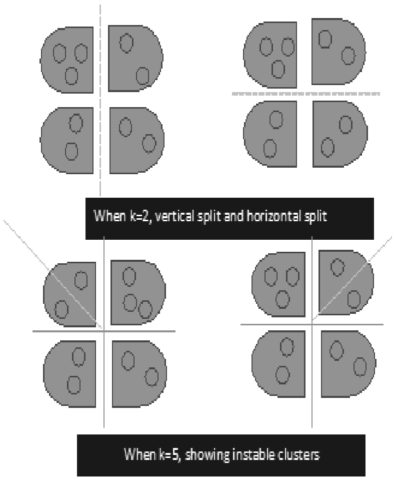
When k=2, vertical split and horizontal split
When k=5, showing instable clusters

Figure 2 Stability

Figure 2 demonstrates how instable clusters are formed. In the second set, when k = 5, the possible chances of formation of clusters are presented. Here the clusters are formed with a lower number of clusters or a higher number of clusters. The method designed in the proposed approach will enable us in selecting the most stable centroid for successive clustering and hence the clustering accuracy can be increased.

The next major constraint considered for the proposed approach is the Mean Square Error (MSE). The MSE value is calculated in order to efficiently identify the most related clusters in the data group. The MSE is a measure of closeness of the data points. The distance between the clusters should be minimized for stable clusters. So the aim of introducing MSE to the method is to get the best clusters from the high dimensional data. The MSE is calculated as the difference between the squares of the data points, which is considered to be a better value than the simple difference. In the proposed method, the better cluster that can be utilized for the further clustering process is identified using MSE. The cluster with the minimum MSE is considered as the relevant cluster for further processing. Basically, any amount at the center of a distribution should be associated with some measure of error. If we say that the centroid c_i is a good measure of cluster, then presumably we are saying that c_i represents the entire distribution in a better way than the other centroids. Hence, it can be inferred that the MSE (c), a centroid, is the measure of quality of that centroid, which can be qualified for an efficient clustering process. The MSE (c) can be calculated as follows:

$$MSE(c) = E(c - c_i)^2 \tag{9}$$

The calculation MSE of cluster c is represented in the above equation. Here c is the cluster centroid which is subjected to minimization and c_i is the other cluster centroid in total. So the value of c_i that minimizes the MSE is selected for further calculations. Now we move on to the total process of the proposed constraint-based clustering process. The procedure is to 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. The next step is to calculate the mean $m_R \in R^P$ as $m_R = 2M - m_L$. The M value needed for calculating the mean is as follows:

$$M = \frac{1}{n-L} \sum_{i=1}^{n-L} D_{F\,i} \tag{10}$$

The following step is to 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:

$$\begin{cases} D_{F\,\mathrm{i}} \in D_L & if \quad E_M(D_{F\,\mathrm{i}}, m_L) \le E_M(D_{F\,\mathrm{i}}, m_R) \\ D_{F\,\mathrm{i}} \in D_R & if \quad E_M(D_{F\,\mathrm{i}}, m_L) > E_M(D_{F\,\mathrm{i}}, m_R) \end{cases} \tag{11}$$

where

$$E_M(D_{F\,\mathrm{i}}, m_L) = \sqrt{\begin{matrix} D_{F\,\mathrm{i}}{}^{(1)}, m_L{}^{(1)})^2 * bi^{(1)} + (D_{F\,\mathrm{i}}{}^{(2)}, m_L{}^{(2)})^2 * bi^{(2)} + \\ \ldots\ldots. + (D_{F\,\mathrm{i}}{}^{(n-L)}, m_L{}^{(n-L)})^2 * bi^{(n-L)} \end{matrix}} \tag{12}$$

$$E_M(D_{F\,\mathrm{i}}, m_L) = \sqrt{\begin{matrix} D_{F\,\mathrm{i}}{}^{(1)}, m_L{}^{(1)})^2 * bi^{(1)} + (D_{F\,\mathrm{i}}{}^{(2)}, m_L{}^{(2)})^2 * bi^{(2)} + \\ \ldots\ldots. + (D_{F\,\mathrm{i}}{}^{(n-L)}, m_L{}^{(n-L)})^2 * bi^{(n-L)} \end{matrix}} \tag{13}$$

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

$$C_L = \frac{1}{N_L}\sum_{j=1}^{N_L} D_{L,j} \qquad C_R = \frac{1}{N_R}\sum_{j=1}^{N_R} D_{R,j} \tag{14}$$

Now, the calculated centroid values of the resulting clusters are used for determining the stopping criteria for the proposed algorithm. The generated clusters have to be revamped for stability. This cluster centroid can give either the best solution or a normal solution. The stability of the cluster has to be calculated for the best solutions. The stability of the cluster centroids gives a better solution in generating clusters. We have now calculated the C_L and C_R. For processing the clusters in terms of stability, another set of data points are selected and considered as centroids. Now calculate C_L' and C_R' for the newly selected points. Then, calculate the distance between the centroids, so as to find the instability of the clusters. The cluster with low instability has to be considered for further processing.

$$\begin{aligned} instable(C_L) &= dist(C_L, C_L^{'}) \\ instable(C_R) &= dist(C_R, C_R^{'}) \end{aligned} \tag{15}$$

The respective C value with less instability is considered for the further processing of the proposed method. The second constraint considered for the proposed approach is the MSE. The MSE of each cluster should be at the minimum for the clustering results. The MSE is calculated as the mean of the squared distance between each data point to the cluster centroid.

$$MSE(C_L) = mean(dist(d_i, C_L))^2 \tag{16}$$

$$MSE(C_R) = mean(dist(d_i, C_R))^2 \tag{17}$$

The cluster centroid for further processing is selected based on the following condition:

$$\begin{aligned} &if\ (instable(C_L)\ \&\ MSE(C_L) < C_L) \\ &\qquad C_L\ is\ selected\ \ for\ cluster\ \ formation \\ &if\ (instable(C_R)\ \&\ MSE(C_R) < C_R) \\ &\qquad C_R\ is\ selected\ \ for\ cluster\ \ formation \end{aligned}$$

The centroid values are calculated as the mean value of the points included in the respective clusters. Thus, the clusters are formed based on the constraints, stability and MSE, which will provide better clustering accuracy when compared to the conventional bisecting algorithms.

3. 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.

3.1. Experimental Setup

The proposed approach is implemented in MATLAB. Here, we have tested our proposed approach using the Spambase (Dataset 1) and Pen-Based Recognition of Handwritten Digits (Dataset 2) Datasets taken from UCI Machine Learning Repository. The testing was done using a computer with an Intel Core 2 Duo CPU with clock speed 2.2GHz and 4 GB RAM.

3.2. Performance Analysis

In this section, we plot the comparative analysis of the proposed approach with respect to the other conventional methods. The proposed method is compared with HB-K Means clustering, Bisecting K Means (Savaresi & Daniel, 2001) and the normal K Means algorithm (Dash et al., 2009). The proposed method turned out to be an enhancement to the HB-K Means algorithm (Aparna & Nair, 2015[2]). The performance all these algorithms are evaluated based on the clustering accuracy and computational time as discussed below.

3.2.1. Performance metrics

A number of metrics for comparing high dimensional data clustering algorithms were recently proposed in the literature. The performance metrics used in our paper are clustering accuracy and computation time.

Clustering Accuracy: Accuracy refers to the degree of closeness of measurement of a quantity to its actual value. Clustering Accuracy is the measure of closeness of the cluster formed as a result of the proposed algorithm to the required value which means how accurate the members of a cluster are. In our paper, the clustering accuracy is computed using the following formula:

$$CA = \frac{1}{N}\sum_{i=1}^{T} X_i \tag{18}$$

where, N is the number of data point and T is the number of class.

Computation Time: Computation time or time complexity quantifies the amount of time taken by an algorithm to run as a function and generate the required output. Also, it determines how the running time scales with the size and dimensionality of the dataset.

3.2.2. Performance analysis based on clustering accuracy

Figures 3 and 4 show the comparative analysis of different algorithms based on clustering accuracy. The analysis plotted here will test the algorithms as the number of clusters is increased from 2 to 7. The experiment is conducted by keeping the binary threshold and outlier threshold as a constant. It can be inferred from both the figures, that when the number of clusters is less, say, 2 and 3, the clustering accuracy is almost the same with respect to all four of the algorithms. As the number of clusters increases, it is observed that clustering accuracy of the proposed algorithm, CHB-K-Means, is better than the traditional K-Means and Bisecting K-Means algorithms. An average clustering accuracy of 80% is achieved for Dataset 1 and an average of 70% accuracy is obtained for Dataset 2.

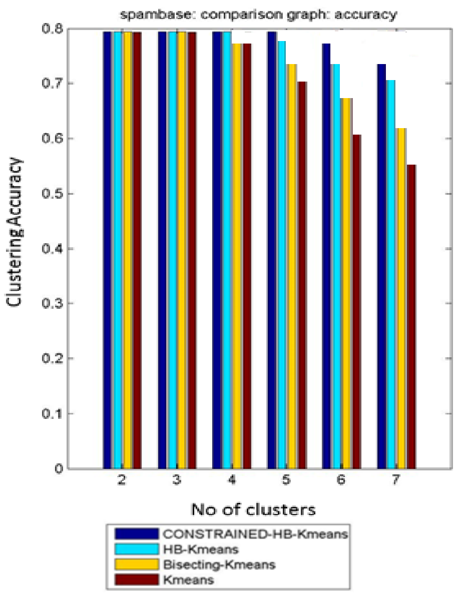

Figure 3 Comparative analysis of dataset 1 in terms of clustering accuracy

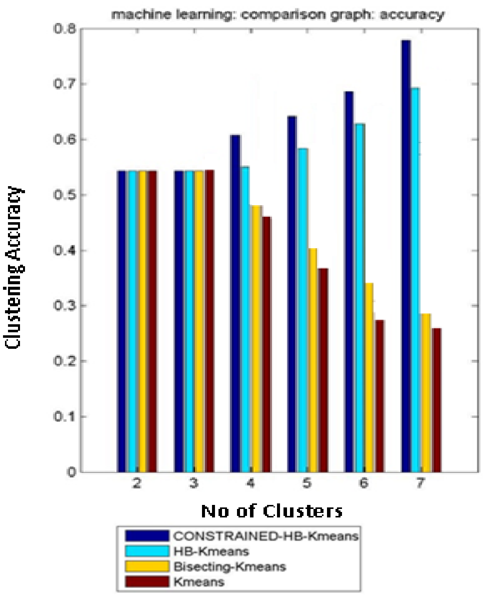

Figure 4 Comparative analysis of dataset 2 in terms of clustering accuracy

3.2.3. Performance analysis based on computational time

Figures 5 and 6 represent the comparative analysis of the different algorithms with respect to the computation time. It can be inferred from the figures that the computation time for the proposed algorithm, CHB-K-Means algorithm, increases as the number of clusters increases. From Figure 5, it is observed that the proposed algorithm takes 0.9 ms for generating 7 clusters, whereas K-Means algorithm takes only around 0.45 ms. Also, from Figure 6, it can be observed that the proposed algorithm takes 0.44 ms for generating 7 clusters whereas K-Means consumes only 0.22 ms. Hence, it can be inferred that the computation time is almost doubled up for the proposed algorithm when compared to the traditional K-Means and Bisecting K-Means algorithm. The time consumption is high for CHB-K Means algorithm because of the added constraints and that increase the loop in the processing phase, which results in high time consumption. These can be overcome by using effective optimization algorithms.

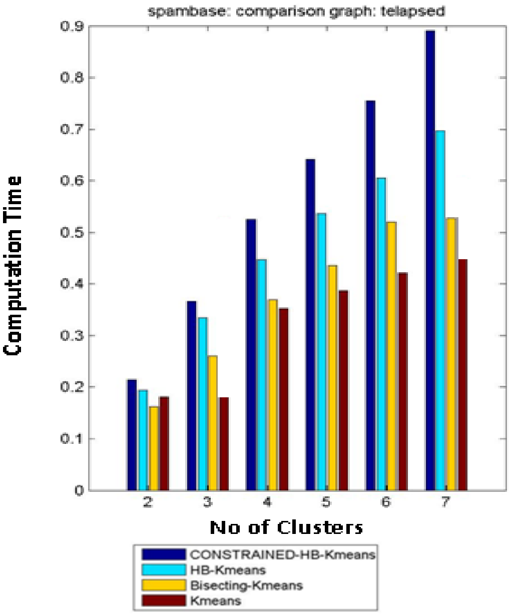

Figure 5 Comparative analysis of dataset 1 in terms of computation time

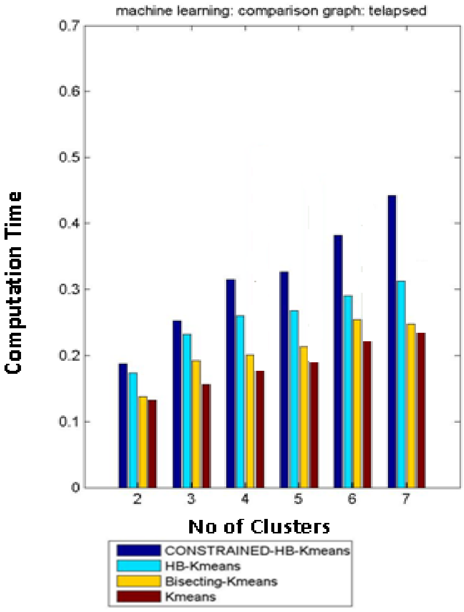

Figure 6 Comparative analysis of dataset 2 in terms of computation time

4. CONCLUSION

In this paper, a method called CHB-K Means clustering is utilized for high dimensional data with the help of constraints. In the proposed method, the high dimensional dataset was converted to a weighted attribute matrix, which is then transformed to a binary matrix. Hence, a data matrix which is free from outliers is obtained. This is given as input to the CHB-K Means clustering. The proposed method uses the constraints, such as Stability and Mean Square Error for improving the clustering efficiency. The Constraint-based Bisecting K-Means is applied repeatedly until the required number of clusters is obtained. The algorithm is tested with various datasets, such as Spam Base and Pen-Based Recognition of Handwritten Digits and the results obtained shows that the algorithm provided better performance than the existing techniques. The results showed that the proposed approach has achieved higher clustering accuracy over the other methods like HB-K Means, Bisecting K-Means and K-Means algorithms, which were used earlier for the high dimensional data clustering. The time elapsed by the proposed approach is high as compared to the other traditional methods, but it can be optimized by using some optimization techniques.

5. REFERENCES

Aparna, K., Nair, M.K., 2015a. Comprehensive Study and Analysis of Partitional Data Clustering Techniques. *International Journal of Business Analytics*, Volume 2(1), pp. 23–38

Aparna, K., Nair, M.K., 2015b. HB-K Means: An Algorithm for High Dimensional Data Clustering using Bisecting K-Means. *International Journal of Applied Engineering Research (IJAER),* Volume 10(14), pp. 34945–34951

Behera, H.S., Lingdoh, R.B., Kodamasingh, D., 2011. An Improved Hybridized K-Means Clustering Algorithm (IHKMCA) for High dimensional Dataset & Its Performance Analysis. *International Journal on Computer Science and Engineering (IJCSE),* Volume 3(3), pp. 1183–1190

Bouguessa, M., Wang, S., 2008. Mining Projected Clusters in High-Dimensional Spaces. *IEEE Transactions on Knowledge and Data Engineering*, Volume 21(4), pp. 507–522

Dash, R., Mishra, D., Rath, A.K., Acharya, M., 2009. A Hybridized K-means Clustering Approach for High Dimensional Dataset. *International Journal of Engineering, Science Technology*, Volume 2(2), pp. 59–66

Ding, C., He, X., 2002. Cluster Merging and Splitting in Hierarchical Clustering Algorithms. *In:* Proceedings of the IEEE International Conference on Data Mining, pp. 139–146

Domeniconi, C., Ma, S., 2004. Subspace Clustering of High Dimensional Data. *In:* Proceedings of International Conference on Data Mining, pp. 517–521

Gu, J.W.F., Feng, W., Zeng, J., Mamitsuka, H., 2013. Efficient Semi-supervised MEDLINE Document Clustering with MeSH-Semantic and Global-Content Constraints. *IEEE Transactions on Cybernetics*, Volume 43(4), pp. 1265–1276

Liu, X., Xie, X., Wang, W., 2009. A Projection Clustering Technique based on Projection. *Journal of Service Science & Management*, Volume 2, pp. 362–367

McCallum, A., Kamal, N., Ungar, L. H., 2000. 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

Napoleon, D., Pavalakodi, S., 2011. New Method for Dimensionality Reduction using K-Means Clustering Algorithm for High Dimensional Data Set. *International Journal of Computer Applications*, Volume 13(7), pp. 41–46

Prasanna, K.M., Kumar, S.P., Narayana, G.S., 2011. A Novel Benchmark K-Means Clustering on Continuous Data. *International Journal on Computer Science and Engineering (IJCSE)*, Volume 3(8), pp. 2974–2977

Savaresi, S.M., Boley, D.L., 2001. On the Performance of Bisecting K-means and PDDP. *In:* Proceedings of the First SIAM International Conference on Data Mining, pp. 1–14

Sculley, D., 2010. Web-scale K-Means Clustering. *In*: Proceedings of the 19th International Conference on World Wide Web, pp. 1177–1178

Valarmathie, P., Srinath, M.V., Dinakaran, K., 2009. An Increased Performance of Clustering High Dimensional Data through Dimensionality Reduction Technique. *Journal of Theoretical and Applied Information Technology*, pp. 731–733

Wagsta, K., Cardie, C., Rogers, S., Schroedgl, S., 2001. Constrained K-means Clustering with Background Knowledge. *In:* Proceedings of the Eighteenth International Conference on Machine Learning, pp. 577–584

Wu, B., Zhang, Y., Hu, B-G., Ji, Q., 2013. Constrained Clustering and Its Application to Face Clustering in Videos. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3507–3514

Yip, K.Y., Cheung, D.W., Ng, M.K., 2004. HARP: A Practical Projected Clustering Algorithm. *IEEE Transactions on Knowledge and Data Engineering*, Volume 16(11), pp.1387–1397