A NOVEL APPROACH FOR HIGH DIMENSIONAL DATA CLUSTERING

B.A Tidke[1], R.G Mehta[2], D.P Rana[3]

[1]*M.Tech Scholar, Computer Engineering Department, SVNIT, Gujarat, India,* ***p10co982@coed.svnit.ac.in***
[2]*Associate Professor, Computer Engineering Department, SVNIT, Gujarat, India,* ***rgm@coed.svnit.ac.in***
[3]*Assistant Professor, Computer Engineering Department, SVNIT, Gujarat, India,* ***dpr@coed.svnit.ac.in***

Abstract

High dimensional data clustering is the analysis of data with few to hundreds of dimensions. Large dimensions are not easy to handle and ***impossible in certain cases to visualize****. To improve the efficiency and accuracy of clustering on high dimensions, data reduction is required as pre-processing. A clustering ensemble is a paradigm that combines the outputs of several clustering algorithms to achieve a more accurate and stable final output. Clustering ensemble method based on a novel two-staged clustering algorithm is proposed in this paper. Existing clustering techniques normally merge small cluster with big ones results in removing the identity of those small clusters. The proposed algorithms work on split and merge technique to overcome this limitation. Experimental results of the proposed method on several data sets are compared with individual clustering results produced by well-known clustering algorithms.*

Index Terms: *Clustering, High Dimensional Data, Subspace, K-Means*

---------- *** ----------

1. INTRODUCTION

Data mining is the process of extracting potentially useful information from a data set [1]. Clustering is a popular but challenging data mining technique, which intended user to discover and understand the structure or grouping of the data in the set according to a certain similarity measure [2]. Clustering techniques need to specify the definition of a similarity measure between patterns, which is difficult to specify in the absence of any prior knowledge (unsupervised learning). Partition and hierarchical clustering methods are two main categories of algorithms in unsupervised learning. A partition algorithm partitions a data set into desired number of clusters. One of the most typical partition clustering algorithms is K-means, it is computationally efficient and does not require the user to specify many parameters [3]. Hierarchical clustering algorithm groups data objects to form a tree shaped structure [1]. It had broadly classified as agglomerative hierarchical clustering and divisive hierarchical clustering**.** In agglomerative approach, which is also called as bottom up approach, each data points are considered a separate cluster and on each iteration, clusters are merged, based on criteria. The merging can be done by using single link, complete link, centroid or wards method. In divisive approach, all data points are considered as a single cluster and they are splited into number of clusters based on certain criteria, and this is called as top down approach. Examples for this algorithms are BRICH [4] (Balance Iterative Reducing and Clustering using Hierarchies) and CURE (Cluster Using representatives) [5].

Traditional clustering have issues in clustering high dimensional data since in high dimensional data set for any point, its distance to its closest point and that to the farthest point tend to be similar which make clustering result useless [6]. Clustering algorithms become substantially inefficient if the required similarity measure is computed between data points in the full-dimensional space. To address this problem, a number of subspace and projected clustering algorithms have been proposed [3][7][8].Subspace clustering broadly divided into two categories bottom-up and top-down. The bottom-up search method takes advantage of the downward closure property of density to reduce the search space, using an APRIORI style approach to reduce the search space [9]. The top-down subspace clustering use a sampling technique to improve performance. Top-down algorithms create clusters that are partitions of the dataset, meaning each instance is assigned to only one cluster. Parameter tuning is necessary in order to get meaningful results. Often the most critical parameters for top-down algorithms is the number of clusters and the size of the subspaces, which are often very difficult to determine ahead of time. In addition, since subspace size is a parameter, top-down algorithms tend to find clusters in the same or similarly sized subspaces. However, most of them encounter difficulties when clusters hide in subspaces with very low dimensionality. These challenges motivate our effort to propose a two-step clustering.

Recent work has focused on the problem of how to interpret and how to combine different partitions produced by different clustering algorithms [10]. This framework, known as Combination of Clustering Ensembles or Ensemble methods, aims at obtaining better clustering results by combining information of different partitioning of the data. The input space for such a problem consists of a set of N data partitions, referred as a clustering ensemble. The rest of paper organized as follows. Related work has been discussed in Section 2. The generic research model and proposed method describes in Section 3. A methodology presents in Section 4 and finally Section 5 concludes the paper.

2. RELATED WORK

Recently lot of work has been done in the area of high dimensional data, which is explained briefly in K. Sim et al. [11], Kriegel et al. [12], G. Moise et al [13]. Some surveys have given overviews on some approaches. In the Well-known survey of Parsons et al. [19], the problem is introduced in a very illustrative way and some approaches are sketched.

2.1. Subspace clustering

C.C. Aggarwal et al. [15] proposed finding projected clusters in high dimensional data. PROCLUS finds the subspace dimensions for each cluster by examining the neighbouring locality of the space near it. The algorithm is run until the sum of intra-cluster distances ceases to change but encounter difficulties when clusters hide in subspaces with very low dimensionality. CLIQUE [16] was one of the first algorithms proposed that attempted to find clusters within subspaces of the dataset. The algorithm combines density, grid based clustering and uses an APRIORI style technique to find clusterable subspaces. Tuning the parameters for a specific dataset can be difficult. Both grid size and the density threshold are input parameters, which greatly affect the quality of the clustering results.

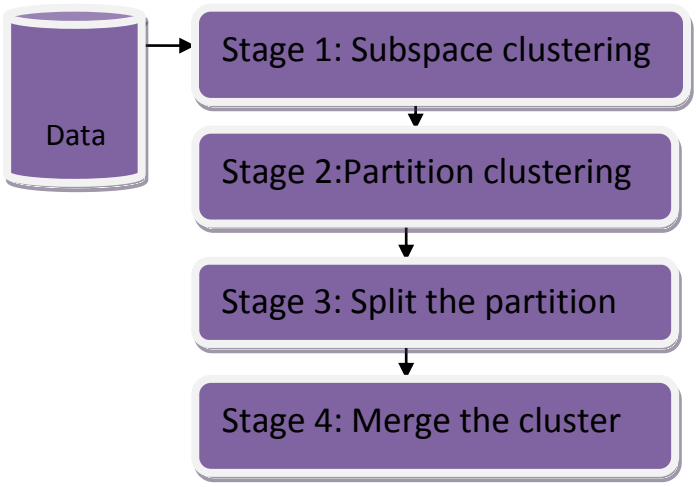
2.2. Two stages algorithm

M. Bouguessa et al.[17] propose a novel projected clustering algorithm, called Projected Clustering based on the K-Means Algorithm (PCKA). PCKA is composed of three phases: attribute relevance analysis, outlier handling, and discovery of projected clusters. Ali Alijamaat et al. [8] proposed method not to reduce dimension but to select subspaces by clustering and perform clustering based on these subspaces. R.Varshavsky et al. [18] proposed a clustering scheme, which included two main steps. For dimension reduction, it uses intrinsic properties of the data set after that several iteration of a clustering algorithms was applied, using different parameter. Base on BIC criterion the best result would be select. There are some weaknesses for this method e.g. since BIC fits a model to specific data distribution it cannot be used to compare models of different data sets. J. Ye et al. [19] proposed a framework, which integrates subspace selection and clustering. Equivalency between kernel K-Means clustering and iterative subspace selection has been shown.

2.3. Existing Clustering Ensembles

Strehl et al. [20] used a knowledge reuse framework and they have measured three different consensus functions for ensemble clustering. Time complexity would be issue for measuring consensus function. Fred et al. [21] proposed a technique after partition clusters co-occurrence matrix would calculated and then hierarchical clustering with a single-link method is applied on co-occurrence matrix. The method works when the number of clusters a priori known. S. Dudoit et al. [22], uses different clustering algorithms to produce partitions for combination by relabeling and voting. Y.Qian et al. [23] proposed sequential combination methods for data Clustering In improving clustering performance they proposed the use of more than one clustering method. They investigated the use of sequential combination clustering as opposed to simultaneous combination and found that sequential combination is less complex and here are improvements without the overhead cost of simultaneous clustering.

3. RESEARCH MODEL



***Stage1*:** PROCLUS samples the data, then selects a set of k-medoids and iteratively improves the clustering [15].The algorithm consists of three phase: initialization, iteration, and cluster refinement. Initialization works on greedy approach to select a set of potential medoids, which must present at some distance [9]. The iteration phase choose a random set of k-medoids from reduced dataset and replaces bad medoids with randomly chosen new medoids, and determines if clustering

has improved. Cluster quality is based on the average distance between instances and the nearest medoid. The total number of dimensions associated to medoids must be k*l, where l is an input parameter that selects the average dimensionality of cluster subspaces [14]. Once the subspaces have chosen for each medoid, average Manhattan segmental distance is used to assign points to medoids, forming clusters. The refinement phase calculates new dimensions for every medoid based on the clusters produced and reassigns points to medoids, removing outliers. While clusters may be found in different subspaces, the subspaces must be of similar sizes since the user must input the average number of dimensions for the clusters. Clusters are representing as sets of instances with associated medoids and subspaces and form non-overlapping partitions of the dataset with possible outliers. Due to the use of sampling, PROCLUS is somewhat faster than CLIQUE on large dataset.

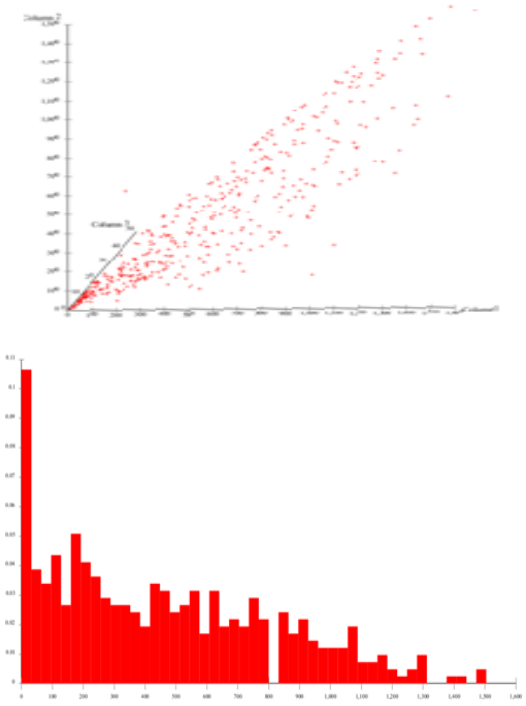


Fig [1]: Shows yeast real dataset from UCI repository with 1484 Instances and 8 dimension

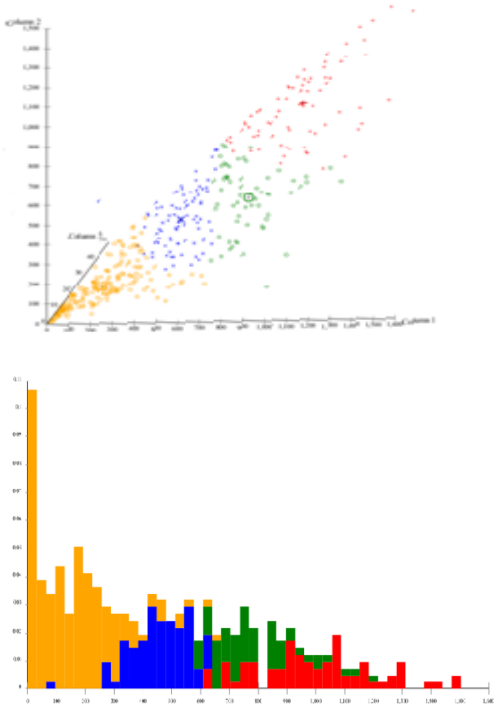


Fig [2]: Proclus algorithm k=4 on yeast data

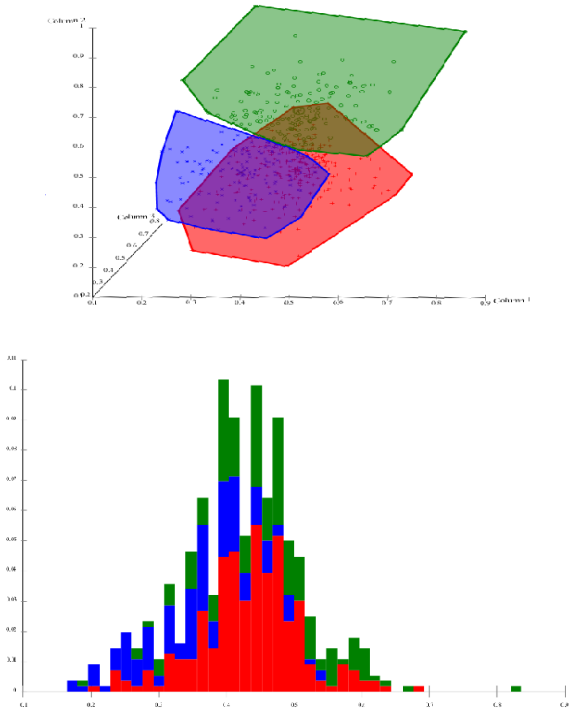
K	k-l	F1	Precision	Recall	Time
2	8	0.841	0.788	0.90	119ms
4	8	0.61	0.46	0.94	287ms
6	8	0.51	0.91	0.35	561ms

Table -1: Comparison of fi measure and time using Proclus algorithm on yeast data

As mentioned earlier, most of the research on subspace clustering is focused on defining the subspace clusters and how to efficiently mine them. The clusters are information extracted from the data, but not knowledge that is useful to the users. To convert information to knowledge, post-processing of the clusters is needed.

Stage 2: After getting subspaces now applied the partition-clustering algorithm k-means, which is computationally efficient on each subspace. Assume fixed number of k cluster so that k centroids assign, one for each cluster it works accurately if these centroids are placed far from each other. Now assign points to its nearest centroid if no points remain

move to next step again calculate centroid for new points this process continues in a loop till no changes occurred in centroids position. So now have clusters from each subspace here if suppose the number of k=4 cluster for each subspace so if there are 3 subspaces so there have been 12 different cluster after applying k-means. Fig [3] shows three clusters and their distribution over two dimensions on subspace 1 of yeast dataset.



Fig[3]: k-means algorithm k=3 on subspace_1 of yeast data dataset

K	Iteration	F1	Precision	Recall	Time
2	14	0.67	1.00	0.50	31ms
4	15	0.61	1.00	0.64	63ms
6	8	0.60	1.00	0.40	33ms

Table 2: Comparison of F1 measure and time using k-means algorithm on cluster 1 of yeast data

Stage 3: These stages follows a split and merge strategy, according to which natural clusters are split into smaller clusters in the partitions of the clustering ensemble, and then recovered during the ensemble combination phase, accomplishing a merging mechanism. Proposed split method takes n clusters which are produced by applying k-means algorithm on each subspaces. Each cluster are splited based on their size in terms of number of points present in each cluster. If size is bigger than a threshold which must be define prior depend on the dataset to be used then that cluster is splitted into two new cluster depending upon the distance between each point and the two centroids of two new cluster. Centroid is the mean of each cluster the distance between point and centroid is calculate using Euclidean distance. Similarly new cluster are also splitted if their size is bigger than threshold forming the hierarchy and this process is applied on each cluster. The algorithm steps are given below

Algorithm 1:
Input: k clusters and threshold T
Output: n cluster
1. Start with k clusters.
2. Check density of each cluster for given threshold T.
3. If density is more than threshold split the cluster into two based on the distance each point is assign to its closest centroid.

$$J=\sum_{j=1}^{k}\sum_{i-1}^{x} || x_i^{(j)} - c_j||^2 \quad (1)$$

Where $|| x_i^{(j)} - c_j||^2$ is a chosen distance measure between a data point $x_i^{(j)}$ and c_j the cluster centre, is an indicator of the distance of the *n* data points from their respective cluster centers.

4. Repeat it for each cluster till it reaches threshold value.
Now when hierarchy of cluster with similar in size formed by splitting phase merging is required to find out the closest cluster to be merge.

Stage 4: In standard hierarchical clustering, clusters are taken as they are in the clustering tree. However, one can improve the clusters by refining them according to a proper clustering objective function. In this paper, distance function is used to find out the closest cluster. Unlike hierarchical clustering two cluster which are nearby are merge. In proposed method, child cluster from any parent cluster can be merged, if there distance is smaller than other cluster in the hierarchy. Also check mean square error(MSE) of each merged cluster with the parent cluster if found to be larger, that cluster must be unmerged and available to be merge with some other cluster in the hierarchy this process repeats until all MSE of all possible combination of merged cluster is checked with its parent cluster. Finally the number of cluster merged and remain are the output cluster. The algorithm steps are given below:

Algorithm 2:
Input: hierarchy of cluster
Output: partition C1….Cn
1. Start with n node cluster.
2. Find the closest two cluster using Euclidean distance from the hierarchy and merge them
3. Calculate MSE of root cluster and new merge cluster

$$SSE = \sum_{j=1}^{k} \sum_{xi \in cj} || xi - \mu j ||^2 \quad (2)$$

Where, μj is the mean of cluster Cj and x is the data object belongs to Cj cluster. Formula to compute μj is shown in equation (3).

$$\mu_j = \frac{1}{nj} \sum x_i \in c_j x_i \quad (3)$$

In sum of squared error formula, the distance from the data object to its cluster centroid is squared and distances are minimized for each data object. Main objective of this formula is to generate compact and separate clusters as possible
4. If mse of new merge cluster is smaller than the cluster after splitting keep it otherwise unmerge them.
5. Repeat until all possible clusters are merge according to step 4.

4. METHODOLOGY

The first two stage of proposed method has been performed using ELKI clustering toolkit in which data mining algorithms and data management tasks are separated and allow for an independent evaluation. This separation makes ELKI unique among data mining frameworks like WEKA or YALE and frameworks for index structures like GIST. At the same time, ELKI is open to arbitrary data types, distance or similarity measures, or file formats [29]. Experiments tested on real world data sets. A simple illustration of each of these is given below.

Yeast: This data can be obtained from the UCI Machine Learning Repository (http://archive.ics.uci.edu/ml/). The set contains 1484 instances and 8 dimensions. Wine: This data also can be obtained from the UCI Machine Learning Repository (http://archive.ics.uci.edu/ml/). The set contains 178 instances and 13 dimensions. Another real data set is the NBA player career statistics. Statistics include number of games played, average number of minutes played per game and many more attributes contains 1548 instances with 15 dimensions.

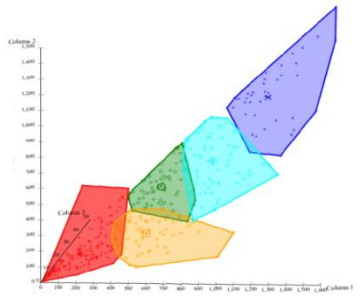
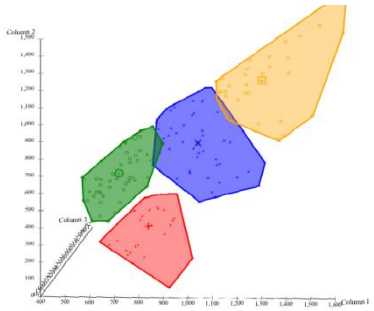
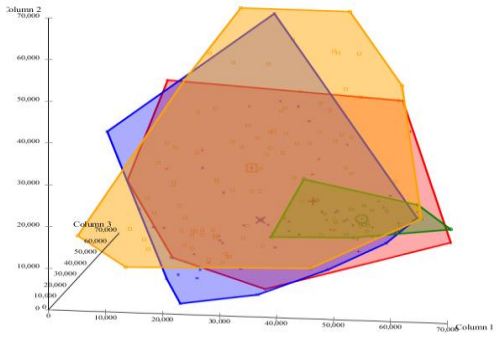


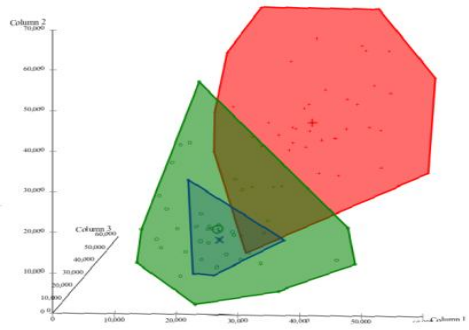
Fig 3 [a] Proclus algorithm on NBA player's dataset



[b] k-means on cluster_2 of Proclus output of NBA player's data set



[c] Proclus algorithm on wine dataset



[d] k-means on cluster_2 of Proclus output of wine data

5. CONCLUSION AND FUTURE WORK

Clustering ensembles have emerged as a prominent method for improving robustness, stability and accuracy of unsupervised classification solutions. This paper give explore of clustering high dimensional data and literature survey of methods proposed by many researchers to overcome the curse of dimensionality and proposed split and merge method for providing a clustering structure that dynamically selects its cluster number with an acceptable runtime and a favorable accuracy. Proposed approach can be highly effective to generate an initial clustering result with an automatically detected number of clusters, there are still many obvious directions to be explored in the future. Complexity of merging algorithm is high and needs to be make more efficient.

REFERENCES

[1]. A.K. Jain, M.N. Murty, and P.J. Flynn, "Data Clustering: A Review," ACM Computing Surveys, vol. 31, no. 3, pp. 264-323, Sept. 1999.

[2]. R. Xu and D. Wunsch, "Survey of clustering algorithms.," IEEE transactions on neural networks / a publication of the IEEE Neural Networks Council, vol. 16, no. 3, pp. 645-78, May. 2005.

[3]. A. K. Jain, "Data clustering : 50 years beyond K-means" PATTERN RECOGNITION LETTERS, 2009.

[4]. T. Zhang, R. Ramakrishnan, and M. Livny, "BIRCH: An efficient data clustering method for very large databases," in Proc. ACM SIGMOD Conf. Management of Data, 1996, pp. 103–114.

[5]. S. Guha, R. Rastogi, and K. Shim, "CURE: An efficient clustering algorithm for large databases," in Proc. ACM SIGMOD Int. Conf. Management of Data,, pp. 73–84, 1998.

[6]. C. Zang and B. Chen, "Automatic Estimation the Number of Clusters in Hierarchical Data Clustering," pp. 269-274.2011.

[7]. M. L. Yiu and N. Mamoulis,, "Iterative Projected Clustering by Subspace Mining,"IEEE Trans. Knowledge and Data Eng., vol. 17, no. 2, pp. 176-189, Feb. 2005.

[8]. Alijamaat, M. Khalilian, and N. Mustapha, "A Novel Approach for High Dimensional Data Clustering," 2010 Third International Conference on Knowledge Discovery and Data Mining, pp. 264-267, Jan. 2010.

[9]. A. Patrikainen and M. Meila, "Comparing Subspace Clusterings," IEEE Trans. Knowledge and Data Eng., vol. 18, no. 7, pp. 902-916, July 2006.

[10]. H. Luo, F. Jing and X. Xie, "Combining multiple clusterings using information theory based genetic algorithm," IEEE International Conference on Computational Intelligence and Security, vol. 1, pp. 84-89, 2006

[11]. K. Sim, V. Gopalkrishnan, A. Zimek, and G. Cong, "A survey on enhanced subspace clustering," Data Mining and Knowledge Discovery, 2012.

[12]. Kriegel HP, Kröger P, Zimek A, "Clustering high-dimensional data: a survey on subspace clustering, pattern-based clustering, and correlation clustering". ACM Trans Knowl Discov Data 3(1):1–58 2009.

[13]. Moise G, Zimek A, Kröger P, Kriegel HP, Sander J , Subspace and projected clustering: experimentalevaluation and analysis. Known Inf Syst 21(3):299–326, 2009.

[14]. L. Parsons, E. Haque, and H. Liu, "Subspace clustering for high dimensional data," ACM SIGKDD Explorations Newsletter, vol. 6, no. 1, pp. 90-105, Jun. 2004

[15]. Aggarwal CC, Wolf JL, Yu PS, Procopiuc C, Park JS "Fast algorithms for projected clustering." In Proceedings of the ACM international conference on management of data (SIGMOD), pp 61–72 1999

[16]. R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. Automatic subspace clustering of high dimensional data for data mining applications In Proceedings of the 1998 ACM SIGMOD international conference on Management of data, pages 94-105. ACM Press1998.

[17]. M. Bouguessa, S. Wang, and Q. Jiang "A K-Means-Based Algorithm for Projective Clustering," Proc. 18th IEEE Int'l Conf. Pattern Recognition (ICPR '06), pp. 888-891, 2006

[18]. R. Varshavsky, D. itorn and M. Linial, cluster algorithm optimizer: A framework for large datasets , ISBRA pp 85- 96,springer 2007.

[19]. J. Ye, Z.Zhao, M. Wu, and G. Tubingen, "Discriminative k-means for clustering," Advances in Neural Information Processing Systems, vol. 20, 2007.

[20]. A. Strehl and J. Ghosh "Cluster ensembles – A knowledge reuse framework for combining multiple partitions," Journal of Machine Learning Research, pp.583-617, Feb. 2002.
[21]. A. Fred and A. K. Jain. "Combining Multiple Clusterings Using Evidence Accumulation," Analysis, vol. 27, no. 6, pp. 835-850, 2005.
[22]. S. Dudoit and J. Fridlyand, "Bagging to improve the accuracy of a clustering procedure", Bioinformatics oxford university, vol. 19, no. 9, pp.1090-1099, Nov. 2003.
[23]. Yuntao Qian and Ching Y. Suen, Sequential combination methods for data clustering, Journal of computer science,2002.
[24]. J. Azimi, M. Mohammadi, A. Movaghar and M. Analoui, "Clustering ensembles using genetic algorithm" ,IEEE The international Workshop on computer Architecture for Machine perception and sensing, pp. 119-123, Sep. 2006.
[25]. Kriegel HP, Kröger P, Renz M, Wurst S "A generic framework for efficient subspace clustering of high-dimensional data". In: Proceedings of the 5th IEEE international conference on data mining (ICDM), pp 250–257 2005.
[26]. M. Bouguessa, S. Wang, and H. Sun, "An Objective Approach to Cluster Validation," Pattern Recognition Letters, vol. 27, no. 13, pp. 1419-1430, 2006
[27]. K.Y.L. Yip, D.W. Cheng, and M.K. Ng, "HARP: A Practical Projected Clustering Algorithm," IEEE Trans. Knowledge and Data Eng., vol. 16, no. 11, pp. 1387-1397, Nov. 2004.
[28]. J. Azimi, M. Abdoos and M. Analoui, "A new efficient approach in clustering ensembles," IDEAL LNCS, vol. 4881, pp. 395-405, 2007.
[29]. E. Achtert, H.-peter Kriegel, and A. Zimek, "ELKI : A Software System for Evaluation of Subspace Clustering Algorithms," Management, no. Ssdbm, pp. 580-585, 2008.

BIOGRAPHIES

B A Tidke is M.Tech scholar in Computer Engineering Department at SVNIT, Surat, Gujarat, India.

R G Mehta working as a Associate Professor in Computer Engineering Department, SVNIT, Surat, Gujarat, India.

D P Rana working as a Assistant Professor in Computer Engineering Department, SVNIT, Surat, Gujarat, India.