Analysis of Clustering with Multi- Viewpoint based Similarity Measure

G. Ramakrishna[1] and V. Sangeeta[2]

1& 2. Dept. of CSE, PYDAH College of Engineering and Technology, Gambheeram, AP, India.

Abstract: Clustering is one of the most interesting and important topics in data mining. The aim of clustering is to find intrinsic structures in data, and organize them into meaningful subgroups. The main concept is similarities/dissimilarities measure from multiple viewpoints. In this paper, we propose a Multi-Viewpoint based Similarity measuring method, named MVS. MVS is potentially more suitable for text documents than the popular cosine similarity. MVS, two criterion functions, *IR* and *IV*, and their respective clustering algorithms, MVSC-*IR* and MVSC-*IV*, have been introduced. Compared with other state-of-the-art clustering methods that use different types of similarity measure, on a large number of document datasets and under different evaluation metrics, the proposed algorithms show that they could provide significantly improved clustering performance.

Keywords*:* k-medoids, Euclidean distance, Jaccard coefficient, Clustering criterion.

I. Introduction

A cluster is a small group or bunch of something.Text mining is the analysis of data contained in natural language text. Text mining works by transposing words and phrases in unstructured data into numerical values which can then be linked with structured data in a database and analyzed with traditional datamining techniques. Document clustering (also referred to as Text clustering) is closely related to the concept of data clustering. Document clustering is a more specific technique for unsupervised document organization, automatic topic extraction and fast information retrieval or filtering. Nevertheless, according to a recent study [1], more than half a century after it was introduced, the simple algorithm *k*-means still remains as one of the top 10 data mining algorithms now a days. It is the most frequently used partitional clustering algorithm in practice. Another recent scientific discussion [2] states that *k*-means is the favorite algorithm that practitioners in the related fields choose to use. Needless to mention, *k*-means has more than a few basic drawbacks, such as sensitiveness to initialization and to cluster size, and its performance can be worse than other state-of-the-art algorithms in many domains. In spite of that, its simplicity, understandability and scalability are the reasons for its tremendous popularity. An algorithm with adequate performance and usability in most of application scenarios could be preferable to one with better performance in some cases but limited usage due to high complexity. While offering reasonable results, *k*meansis fast and easy to combine with other methods in larger systems.Hierarchical clustering algorithm that is based on some of the same premises as well-known partition- and density-based techniques. The time-complexity of k-medoids related algorithms is avoided in a systematic way and the influence of outliers is reduced. The hierarchical organization of data represents information at any desired level of granularity and relieves the user from the necessity of selecting parameters prior to clustering. Different levels in the hierarchy are efficiently calculated by using lower level solutions as starting points for the computation of higher level cluster centers. We use the P-tree data structure for efficient storage and access of data. Comparison with kmeans shows that we can achieve the benefits of improved outlier handling without sacrificing performance.

A common approach to the clustering problem is to treat it as an optimization process. An optimal

partition is found by optimizing a particular function of similarity (or distance) among data. Basically, there is an implicit assumption that the true intrinsic structure of data could be correctly described by the similarity formula defined and embedded in the clustering criterion function. Hence, effectiveness of clustering algorithms under this approach depends on the appropriateness of the similarity measure to the data at hand. For instance, the original *k*-means has sum-of-squared-error objective function that uses Euclidean distance. In a very sparse and high dimensional domain like text documents, spherical *k*means, which uses cosine similarity instead of Euclidean distance as the measure, is deemed to be more suitable[3], [4].

The work in this paper is motivated by investigations from the above and similar research findings. It appears to us that the nature of similarity measure plays a very important role in the success or failure of a clustering method. Our first objective is to derive a novel method for measuring similarity between data objects in sparse and high dimensional domain, particularly text documents. From the proposed similarity measure, we then formulate new clustering criterion functions and introduce their respective clustering algorithms, which are fast and scalable like *k*-means, but are also capable of providing high-quality and consistent performance.

II. EXISTING SYSTEM

There are many state-of-theart clustering approaches that do not employ any specific form of measurement, for instance, probabilistic modelbased method [5], non-negative matrix factorization [6], information theoretic co-clustering [7] and so on. Inthis paper, though, we primarily focus on methods that indeed do utilize a specific measure. In the literature, Euclidean distance is one of the most popular measures:

Dist(di, dj) = ||di – dj||(1)

It is used in the traditional *k*-means algorithm. The objective of *k*-means is to minimize the Euclidean distance between objects of a cluster and that cluster's centroid:

$$\min \sum_{r=1}^{k} \sum_{d_i \in S_r} \|d_i - C_r\|^2 \qquad (2)$$

However, for data in a sparse and high-dimensional space, such as that in document clustering, cosine similarity is more widely used. It is also a popular similarity score in text mining and information retrieval [8]. Particularly, similarity of two document vectors *di* and *dj*,*Sim*(*di, dj*), is defined as the cosine of the angle between them. For unit vectors, this equals to their inner product:

Sim(*di, dj*) = cos(*di, dj*) = $d^{i}_{i}dj$(3)

Cosine measure is used in a variant of *k*-means called spherical *k*-means [3]. While *k*-means aims to minimize Euclidean distance, spherical *k*-means intends to maximize the cosine similarity between documents in a cluster and that cluster's centroid

$$\max \sum_{r=1}^{k} \sum_{d_i \in S_r} \frac{d_i^t C_r}{\|C_r\|} \qquad (4)$$

The major difference between Euclidean distance and cosine similarity, and therefore between *k*-means and spherical *k*-means, is that the former focuses on vector magnitudes, while the latter emphasizes on vector directions. Besides direct application in spherical *k*-means, cosine of document vectors is also widely used in many other document clustering methods as a core similarity measurement. The min-max cut graph-based spectral method is an example [10]. In graph partitioning approach, document corpus is consider as a graph *G* =(*V,E*), where each document is a vertex in *V* and each edge in *E* has a weight equal to the similarity between a pair of vertices. Min-max cut algorithm tries to minimize the criterion function:

$$\min \sum_{r=1}^{k} \frac{Sim\,(S_r, S \setminus S_r)}{Sim\,(S_r, S_r)} \qquad (5)$$

$$\text{where } \underset{1 \le q,r \le k}{Sim\,(S_q, S_r)} = \sum_{d_i \in S_q, d_j \in S_r} Sim(d_i, d_j)$$

and when the cosine as in Eq. (3) is used, minimizingthe criterion in Eq. (5) is equivalent to:

$$\min \sum_{r=1}^{k} \frac{D_r^t D}{\|D_r\|^2} \qquad (6)$$

Another popular graph-based clustering technique is implemented in a software package called CLUTO [19]. This method first models the documents with a nearest neighbor graph, and then splits the graph into clusters using a min-cut algorithm. Besides cosine measure, the extended Jaccard coefficient can also be used in thismethod to represent similarity between nearest documents.Given non-unit document vectors ui, uj(di =ui/||ui||, dj= uj||uj||), their extended Jaccard coefficientis:

$$Sim_{eJacc}\,(u_i, u_j) = \frac{u_i^t u_j}{\|u_i\|^2 + \|u_j\|^2 - u_i^t u_j} \qquad (7)$$

Compared with Euclidean distance and cosine similarity, the extended Jaccard coefficient takes into account both the magnitude and the direction of the document vectors. If the documents are instead represented by their corresponding unit vectors, this measure has the same effect as cosine similarity. In [9], Strehl et al. compared four measures: Euclidean, cosine, Pearson correlation and extended Jaccard, and

concluded that cosine and extended Jaccard are the best ones on web documents.

III. PROPOSED SYSTEM

In this section, we present analytical study to show that the proposed MVS could be a very effective similarity measure for data clustering. In order to demonstrate its advantages, MVS is compared with cosine similarity (CS) on how well they reflect the true group structure in document collections.

The cosine similarity in Eq. (3) can be expressed in the following form without changing its meaning:

$$Sim(di, dj) = cos(di-0, dj-0) = (di-0)^t(dj-0) \quad (8)$$

Where 0 is vector 0 that represents the origin point. According to this formula, the measure takes 0 as one and only reference point. The similarity between two documents *di* and *dj*is determined w.r.t. the angle between the two points when looking from the origin. To construct a new concept of similarity, it is possible to use more than just one point of reference. We may have a more accurate assessment of how close or distant a pair of points are, if we look at them from many different viewpoints. From a third point *dh*, the directions and distances to *di* and *dj*are indicated respectively by the difference vectors (*di* − *dh*) and (*dj*− *dh*). By standing at various reference points *dh*to view *di*, *dj*and working on their difference vectors, we define similarity between the two documents as:

$$\underset{d_i,d_j \in S_r}{Sim(d_i,d_j)} = \frac{1}{n-n_r} \sum_{d_h \in S\backslash S_r} Sim(d_i-d_h, d_j-d_h) \quad (9)$$

As described by the above equation, similarity of two documents d_i and d_j- given that they are in the same cluster - is defined as the average of similarities measured relatively from the views of all other documents outside that cluster. We call this proposal the Multi-Viewpoint based Similarity, or MVS. From this point onwards, we will denote the proposed similarity measure between two document vectors *di* and *dj* by MVS(*di, dj*|*di, = dj Sr*), or occasionally MVS(*di, dj*) for short.

The final form of MVS in Eq. (9) depends on particular formulation of the individual similarities within the sum. If the relative similarity is defined by dot-product of the difference vectors, we have:

$$\begin{aligned} &\mathrm{MVS}(d_i,d_j|d_i,d_j \in S_r) \\ &= \frac{1}{n-n_r} \sum_{d_h \in S\backslash S_r} (d_i-d_h)^t(d_j-d_h) \\ &= \frac{1}{n-n_r} \sum_{d_h} \cos(d_i-d_h, d_j-d_h)\|d_i-d_h\|\|d_j-d_h\| \end{aligned} \quad (10)$$

The similarity between two points *di* and *dj*inside cluster *Sr*, viewed from a point *dh*outside this cluster, is equal to the product of the cosine of the angle between *di* and *dj*looking from *dh* and the Euclidean distances from *dh* to these two points. This definition is based on the assumption that *dh* is not in the same cluster with *di* and *dj*. The smaller the distances ||*di*−*dh*||and ||*dj*–*dh*||are, the higher the chance that *dh* is in fact in the same cluster with *di* and *dj*, and the similarity based on *dh* should also be small to reflect this potential.

MVS is compared with cosine similarity (CS) on how well they reflect the true group structure in document collections. Firstly, exploring Eq. (10), we have:

$$\begin{aligned} &\mathrm{MVS}(d_i,d_j|d_i,d_j \in S_r) \\ &= \frac{1}{n-n_r} \sum_{d_h \in S\backslash S_r} (d_i^t d_j - d_i^t d_h - d_j^t d_h + d_h^t d_h) \\ &= d_i^t d_j - \frac{1}{n-n_r} d_i^t \sum_{d_h} d_h - \frac{1}{n-n_r} d_j^t \sum_{d_h} d_h + 1, \ \|d_h\|=1 \\ &= d_i^t d_j - \frac{1}{n-n_r} d_i^t D_{S\backslash S_r} - \frac{1}{n-n_r} d_j^t D_{S\backslash S_r} + 1 \\ &= d_i^t d_j - d_i^t C_{S\backslash S_r} - d_j^t C_{S\backslash S_r} + 1 \end{aligned} \quad (11)$$

where $D_{S\backslash S_r} = \sum_{d_h \in S\backslash S_r} d_h$ is the composite vector of all the documents outside cluster r, called the outer composite w.r.t. cluster r, and $C_{S\backslash S_r} = D_{S\backslash S_r}/(n-n_r)$ the outer centroid w.r.t. cluster r, $\forall r = 1,\ldots,k$. From Eq. (11), when comparing two pairwise similarities $\mathrm{MVS}(d_i,d_j)$ and $\mathrm{MVS}(d_i,d_l)$, document d_j is more similar to document d_i than the other document d_l is, if and only if:

$$\begin{aligned} &d_i^t d_j - d_j^t C_{S\backslash S_r} > d_i^t d_l - d_l^t C_{S\backslash S_r} \\ \Leftrightarrow\ &\cos(d_i,d_j) - \cos(d_j, C_{S\backslash S_r})\|C_{S\backslash S_r}\| > \\ &\qquad \cos(d_i,d_l) - \cos(d_l, C_{S\backslash S_r})\|C_{S\backslash S_r}\| \end{aligned} \quad (12)$$

From this condition, it is seen that even when d_l is considered "closer" to d_i in terms of CS, i.e.$\cos(d_i, d_j) \leq \cos(d_i, d_l)$, d_l can still possibly be regardedas less similar to d_i based on MVS if, on the contrary,it is "closer" enough to the outer centroid $C_{S\backslash S_r}$than$d_j$is.

Multi-Viewpoint Based Clustering

Having defined out similarity measure, we now formulate out clustering criterion functions. The first function, class I_R, is the cluster size-weighted sum of average pairwise similarities of documents in the same cluster.Firstly, let us express this sum in a general form by function F:

$$F = \sum_{r=1}^{k} n_r \left[\frac{1}{n_r^2} \sum_{d_i,d_j \in S_r} Sim(d_i,d_j) \right] \quad (13)$$

In common practice, $\{\lambda r\}_1^k$ are often taken to be simplefunctions of the respective cluster sizes $\{nr\}_1^k$ [28]. Let us use a parameter α called the regulating

factor, whichhas some constant value (α [0, 1]), and let $\lambda r = n^{\alpha}_{r}$inEq. (15), the final form of our criterion function *IR* is:

$$I_R = \sum_{r=1}^{k} \frac{1}{n_r^{1-\alpha}} \left[\frac{n+n_r}{n-n_r} \|D_r\|^2 - \left(\frac{n+n_r}{n-n_r} - 1 \right) D_r^t D \right] \quad (16)$$

IV calculates the weighted difference between the twoterms: ||*Dr*||and *DtrD*/||*D*||*r*||, which again represent anintra-cluster similarity measure and an inter-cluster similaritymeasure, respectively. The first term is actuallyequivalent to an element of the sum in spherical *k*-meansobjective function in Eq. (4); the second one is similar toan element of the sum in min-max cut criterion in Eq.(6), but with ||*D*||*r* ||as scaling factor instead of _*Dr* _2.

$$I_V = \sum_{r=1}^{k} \left[\frac{n+\|D_r\|}{n-n_r} \|D_r\| - \left(\frac{n+\|D_r\|}{n-n_r} - 1 \right) \frac{D_r^t D}{\|D_r\|} \right] \quad (18)$$

IV. EXPERIMENTAL RESULTS

To demonstrate how well MVSCs can perform, we compare them with six other clustering methods on the twenty datasets in Table 1. In summary, the eight clustering algorithms are:

- MVSC-*IR*: MVSC using criterion function *IR*
- MVSC-*IV* : MVSC using criterion function *IV*
- k-means: standard k-means with Euclidean distance
- Sp*k*means: spherical *k*-means with CS
- graphCS: CLUTO's graph method with CS
- graphEJ: CLUTO's graph with extended Jaccard
- MMC: Spectral Min-Max Cut algorithm [10]
- HC:Hierarchical Clustering

Many clustering algorithms require parameter to be chosen to determine the granularity of the result. Partitioning methods such as the k-means and k-medoids algorithms require that the number of clusters, k, be specified. Density-based methods use input parameters that relate directly to cluster size rather than the number of clusters. Hierarchical methods avoid the need to specify either type of parameter and instead produce results in the form of tree structures that include all levels ofgranularity. When generalizing partitioning-based methods to hierarchical ones, the biggest challenge is the performance.

Hierarchical clustering as a search for equilibrium cluster centers requires us to have a fast method of finding data points based on their feature attribute values. Density-based algorithms such as DENCLUE achieve this goal by saving data in a special data structure that allows referring to neighbors.We use a data structure, namely a Peano Count Tree (or P-tree) [11, 12, 13, 14, 15] thatallows fast calculation of counts of data points based on their attribute values.

P-TREE

Many types of data show continuity in dimensions that are not themselves used as data mining attributes. Spatial data that is mined independently of location will consist of large areas of similar attribute values. Data streams and many types of multimedia data, such as videos, show a similar continuity in their temporal dimension. Peano Count Trees are constructed from the sequences of individual bits, i.e., 8 P-trees are constructed for byte-valued data. Compression is achieved by eliminating nodes that consist entirely of 0- or 1-values. Two and more dimensional data is traversed in Peano order, i.e., recursive raster order. This ensures that continuity in all dimensions benefits compression equally. Counts are maintained for every quadrant. The P-tree for an 8-row-8-column bit-band is shown in Figure 1

Hierarchical clustering algorithm that is based on some of the same premises as well-known partition- and density-based techniques. The time-complexity of k-medoids related algorithms is avoided in asystematic way and the influence of outliers isreduced. The hierarchical organization of datarepresents information at any desired level ofgranularity and relieves the user from the necessity of selecting parameters prior to clustering. Different levels in the hierarchy are efficiently calculated by using lower level solutions as starting points for the

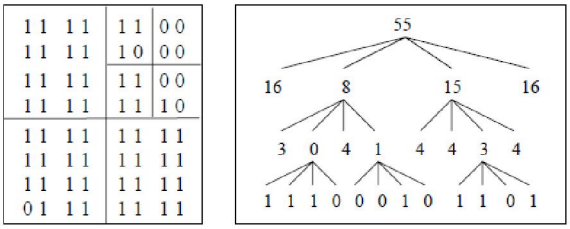

Fig 1: 8x8 image and P-Tree.

computation of higher level cluster centers. We use the P-tree data structure for efficient storage and access of data. Comparison with kmeans shows that we can achieve the benefits of improved outlier handling without sacrificing performance.

We tested the speed and effectiveness of Hierarchical clustering algorithm by comparing with the result of using k meansclustering.The data was generated with no assumptions on continuity in the structural dimension (e.g., location for spatial data, time for multimedia data).Such continuity would significantly benefit from the use of P-tree methods. The speed demonstrated in this section can therefore be seen as an upper bound to the time complexity. Speed

comparison was done on data with 2 attributes for a range of data set sizes.[15]

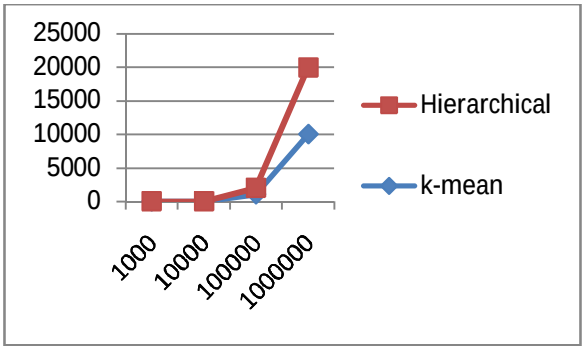

Fig 2: Speed Comparison, Hierarchical and K-means approach

The Table 1 compares the cluster centers of k-means for k = 5 with those found by Hierarchical algorithm. K-means results were significantly more influenced by the noise between the identifiable clusters than the results of our algorithm.

Table 1: Comparison of cluster center for the data set of Fig 2:

k-means (k=5)	X	11	4	35	4	23
	y	11	12	6	22	23
Hierarchical	X	9	27	24	4	18
	y	11	22	6	21	25

Clustering solution is evaluated by comparing the documents' assigned labels with their true labels provided by the corpus. Three types of external evaluation metric are used to assess clustering performance. They are the *FScore*, Normalized Mutual Information (*NMI*) and *Accuracy*. *FScore*is an equally weighted combination of the "precision" (*P*) and "recall" (*R*) values used in information retrieval. Given a clustering solution, *FScore*is determined as:

$$FScore = \sum_{i=1}^{k} \frac{n_i}{n} \max_j (F_{i,j})$$

$$\text{where } F_{i,j} = \frac{2 \times P_{i,j} \times R_{i,j}}{P_{i,j} + R_{i,j}}; P_{i,j} = \frac{n_{i,j}}{n_j}, R_{i,j} = \frac{n_{i,j}}{n_i}$$

Where *ni*denotes the number of documents in class *i*, *nj*the number of documents assigned to cluster *j*, and *ni,j*the number of documents shared by class *i* and cluster *j*. From another aspect, *NMI* measures the informationthe true class partition and the cluster assignment share.It measures how much knowing about the clusters helpsus know about the classes:

$$NMI = \frac{\sum_{i=1}^{k} \sum_{j=1}^{k} n_{i,j} \log\left(\frac{n \cdot n_{i,j}}{n_i n_j}\right)}{\sqrt{\left(\sum_{i=1}^{k} n_i \log \frac{n_i}{n}\right)\left(\sum_{j=1}^{k} n_j \log \frac{n_j}{n}\right)}}$$

Finally, *Accuracy* measures the fraction of documents that are correctly labels, assuming a one-to-one correspondence between true classes and assigned clusters. Let *q* denote any possible permutation of index set *{1, . . . , k},Accuracy* is calculated by:

$$Accuracy = \frac{1}{n} \max_q \sum_{i=1}^{k} n_{i,q(i)}$$

The best mapping *q* to determine *Accuracy* could be found by the Hungarian algorithm2. For all three metrics, their range is from 0 to 1, and a greater value indicates a better clustering solution.

It can be observed that MVSC-*IR* and MVSC-*IV* perform consistently well. In Fig. 1 19 out of 20 datasets, except *reviews*, either both or one of MVSC approaches are in the top two algorithms. The next consistent performer is Hierarchical Clustering. The other algorithms might work well on certain dataset. For example, graphEJ yields outstanding result on *classic*; graphCS and MMC are good on *reviews*.

The observation, which is also the main objectve of this empirical study, is that by applying MVSC to refine the output of spherical *k*-means, clustering are improved significantly. Both rMVSC-*IR* and rMVSC-*IV* lead to higher *NMIs* and *Accuracies* than Sp*k*means in all the cases. Interestingly, there are many circumstances where Sp*k*means' result is worse than that of NMF clustering methods, but after refined by MVSCs, it becomes better.

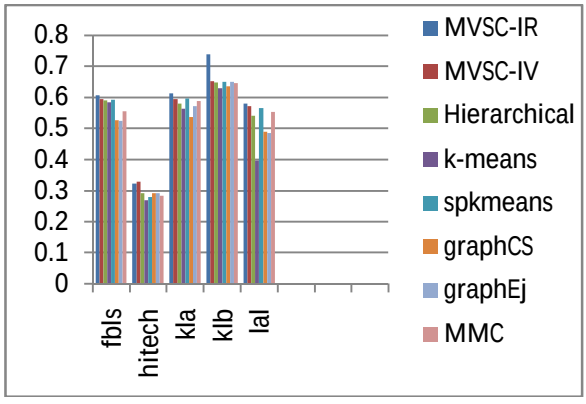

Fig2: Clustering results in FScore

But they do not fare very well on the rest of the collections

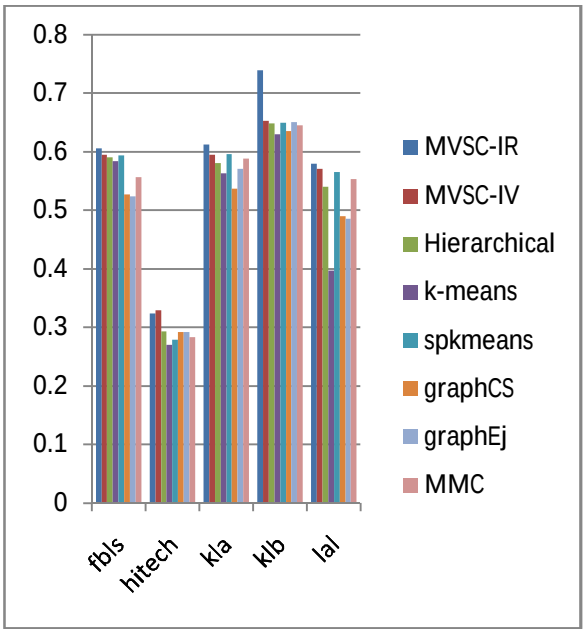

Fig 3:Clustering Results in NMI

V. CONCLUSION

In this paper, we propose a Multi-Viewpoint based Similarity measuring method, named MVS. MVS is potentially more suitable for text documents than the popular cosine similarity. Based on MVS, two criterion functions, I_R and I_V, and their respective clustering algorithms, MVSC-I_R and MVSC-I_V, have been introduced. Compared with other state-of-the-art clustering methods that use different types of similarity measure, on a large number of document datasets and under different evaluation metrics, the proposed algorithms show that they could provide significantly improved clustering performance. It is applied on the proposed criterion functions for hierarchical clustering algorithms. It is work on other types of sparse and high-dimensional data. Compared with other state-of-the-art clustering methods that use different types of similarity measure, on a large number of sparse and high-dimensional data datasets and under different evaluation metrics, the Hierarchical algorithms show that they could provide significantly improved clustering performance.

References

[1] X. Wu, V. Kumar, J. Ross Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. McLachlan, A. Ng, B. Liu, P. S. Yu, Z.-H. Zhou, M. Steinbach, D. J. Hand, and D. Steinberg, "Top 10 algorithms in data mining," *Knowl.Inf. Syst.*, vol. 14, no. 1, pp. 1–37, 2007.

[2] I. Guyon, U. von Luxburg, and R. C. Williamson, "Clustering: Science or Art?" *NIPS'09 Workshop on Clustering Theory*, 2009.

[3] I. Dhillon and D. Modha, "Concept decompositions for largesparse text data using clustering," *Mach. Learn.*, vol. 42, no. 1 2,pp. 143–175, Jan 2001.

[4] S. Zhong, "Efficient online spherical K-means clustering," in *IEEEIJCNN*, 2005, pp. 3180–3185.

[5] A. Banerjee, I. Dhillon, J. Ghosh, and S. Sra, "Clustering on the unit hypersphere using von Mises-Fisher distributions," *J. Mach.Learn. Res.*, vol. 6, pp. 1345–1382, Sep 2005.

[6] W. Xu, X. Liu, and Y. Gong, "Document clustering based on nonnegative matrix factorization," in *SIGIR*, 2003, pp. 267–273.

[7] I. S. Dhillon, S. Mallela, and D. S. Modha, "Information-theoretic co-clustering," in *KDD*, 2003, pp. 89–98.

[8]C. D. Manning, P. Raghavan, and H. Sch ¨ utze, *An Introduction to Information Retrieval*. Press, Cambridge U., 2009.

[9] A. Strehl, J. Ghosh, and R. Mooney, "Impact of similarity measures on web-page clustering," in *Proc. of the 17th National Conf. on Artif.Intell.: Workshop of Artif. Intell.for Web Search*. AAAI, Jul. 2000, pp. 58–64.

[10] C. Ding, X. He, H. Zha, M. Gu, and H. Simon, "A min-max cut algorithm for graph partitioning and data clustering," in *IEEEICDM*, 2001, pp. 107–114.

[11] Qin Ding, Maleq Khan, Amalendu Roy, and William Perrizo, "P-tree Algebra", ACM Symposium on AppliedComputing, Madrid, Spain, 2002.

[12] Maleq Khan, Qin Ding, William Perrizo, "K-Nearest Neighbor Classification of Spatial Data Streams using Ptrees",PAKDD-2002, Taipei, Taiwan, May 2002.

[13] Qin Ding, Qiang Ding, William Perrizo, "Association Rule Mining on Remotely Sensed Images using P-trees",PAKDD-2002, Taipei, Taiwan, 2002.

[14] Qin Ding, William Perrizo, Qiang Ding, "On Mining Satellite and other RSI Data", DMKD-2001, Santa Barbara, CA, 2001.

[15] A. Roy, "Implementation of Peano Count Tree and Fast P-tree Algebra", M. S. thesis, North Dakota State University, 2001.

V.Sangeeta received her B.Sc from Andhara University, Vishapatnam, Andharapradesh, India. MCA Degree from Pydah College of Engineering and Technology Visakhapatnam, Andharapradesh, India. M.Tech from Andhara University, Vishapatnam, Andharapradesh, India.and pursuing Ph.D from JNTU Kakinada.She is Associate Professor in Department of Compter Scince and Technoloty. She has 15 years of experience in teaching.

G.Ramakrishna completed his M.C.A. degree from Gayathry College of Science and Management affiliated to Andhra University, Visakhapatnam, Andhara Pradesh, India. Now he is pursuing M.Tech Degree from Pydah College of Engineering and Technology Visakhapatnam, Andharapradesh, India. He has intrest areas in Data Mining and Operating Syatems