

Soft TDCT: A Fuzzy Approach towards Triangle Density based Clustering

Hrishav Bakul Barua
IT services (BSNL Project)
Tata Consultancy Services
Kolkata, India

Sauravjyoti Sarmah
Computer Science and
Engineering Department
Jorhat Engineering College Jorhat

Mukul Biswas
IT Services
Tata Consultancy Services
Kolkata, India

ABSTRACT

Patterns and useful trends in large datasets has attracted considerable interest recently, and one of the most widely studied problems in this area is the identification and formation of clusters, or densely populated regions in a dataset. Prior work does not adequately address the problem of large datasets and minimization of I/O costs. The objective of this paper is to present a fuzzy logical approach towards clustering to refine the results obtained from the previous approach; Triangle-density based clustering technique (TDCT) [1], which was proposed in an earlier research paper in 2012. We hence name this algorithm as Soft Triangle Density Based Clustering Technique (STDCT). This algorithm incorporates soft clustering and is capable of identifying embedded clusters of arbitrary shapes as well as multi-density clusters over large spatial datasets with precision. Experimental results are reported to establish the superiority of the technique in terms of cluster quality and complexity. Fig. 1 depicts the formation of clusters of similar data.

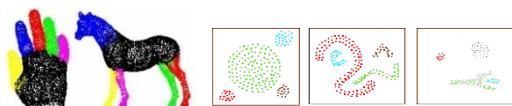


Figure 1- Formation of clusters

General Terms

Clustering Algorithms, Design, Experimentation, Performance

Keywords

Clustering; Density-based; Triangle Density; Polygon Approach; Fuzzy Clustering

1. INTRODUCTION

The technique of data clustering is a particular kind of data mining problem. The process of grouping a set of physical or abstract objects into classes of similar objects is called clustering. A cluster is a collection of data objects that are similar to one another within the same cluster and are dissimilar to the objects in other clusters [2]. Given a large set of data points (data objects); the data space is usually not uniformly occupied. Data Clustering identifies the sparse and the crowded places, and hence discovers the overall distribution patterns of the data set. Besides, the derived clusters can be visualized more efficiently and effectively than the original dataset. Mining knowledge from large amounts of spatial data is known as spatial data mining. It becomes a highly demanding field because huge amounts of spatial data have been collected in various applications ranging from geo-spatial data to bio-medical knowledge. The amount of spatial data being collected is increasing exponentially and has far

exceeded human's ability to analyze them. Recently, clustering has been recognized as a primary data mining method for knowledge discovery in spatial database. The development of clustering algorithms has received a lot of attention in the last few years and new clustering algorithms are proposed.

The density-based approach is famous for its capability of discovering arbitrary shaped clusters of good quality even in noisy datasets [3]. Density Based Spatial Clustering of Applications with Noise (DBSCAN) and Ordering Points to Identify the Clustering Structure (OPTICS) are two of the most popular density based clustering algorithms. In density-based clustering algorithms, a cluster is defined as a high-density region partitioned by low-density regions in data space. They can find out the clusters of different shapes and sizes from the large amount of data containing noise and outliers. Fig. 1 depicts the formation of clusters of similar data. TDCT [1] which is inspired by Density Based Clustering uses Polygon approach [1] to perform the clustering where the number of points inside a triangle (triangle density) of a polygon is calculated using barycentric formulae. This is because of the fact that partitioning of the data set can be performed more efficiently in triangular shape than in any other polygonal shape due to its smaller space dimension. The ratio of number of points between two triangles can be found out which forms the basis of nested clustering. In this paper, we present a fuzzy algorithm by using TDCT as the basis. This algorithm has two stages, firstly TDCT will give us the hard or crisp clusters and then we apply fuzzy logical techniques to refine the clusters (soft clustering) obtained from TDCT. The rest of the paper is organized as follows. Section 2 provides a selected literary review on density based, grid based and other multi-density as well as variable density data clustering techniques and fuzzy clustering techniques. Section 3 illustrates the background of the proposed work and gives the final proposed algorithm. In section 4, we present the experimental results and the performance analysis of the work. Section 5 presents the complexity analysis. Lastly, we conclude with a summary in section 6.

2. PREVIOUS WORKS IN THIS FIELD

Please this section portrays a selected literary review on some of the previous works in this field especially some relevant density based as well as grid based clustering techniques.

A. Density Based Approach: Most partitioning methods cluster objects based on the distance between the objects. Such methods can find only spherical-shaped clusters and encounter difficulty at discovering clusters of arbitrary shapes. Other clustering methods have been developed based on the notion of the density. Their general idea is to continue growing the given cluster as long as the density (number of

objects or data points) in the neighborhood exceeds some threshold; that is for each data point within a given cluster, the neighborhood of a given radius has to contain at least a minimum number of points. Such a method can be used to filter out noise (outliers) and discover clusters of arbitrary shape. To discover clusters with arbitrary shape, density based clustering methods have been developed. These typically regard clusters as dense regions of objects in the data space that are separated by regions of low density. DBSCAN [3] grows clusters according to a density-based connectivity analysis. OPTICS extends DBSCAN [3] to produce a cluster ordering obtained from a wide range of parameter settings. The idea behind density based clustering approach is that the density of points within a cluster is higher as compared to those outside of it. DBSCAN [3] is a density based clustering algorithm capable of discovering clusters of various shapes even in presence of noise. The key idea of DBSCAN is that for each point of a cluster, the neighborhood of a given radius (ϵ) has to contain at least a minimum number of points and the density in the neighborhood has to exceed some threshold. It is efficient for large spatial databases but, for massive datasets, it becomes very time consuming, even if the use of R* tree is made. Another drawback of DBSCAN is that due to the use of the global density parameters, it fails to detect embedded or nested clusters.

B. Grid Based Approach: Grid-based methods quantize the object space into a finite number of cells that form a grid structure. All of the clustering operations are performed on the grid structure. The main advantage of this approach is its fast-processing time, which is independent of the number of data objects and dependent only on the number of cells in each dimension in the quantized space. There is high probability that all data points that fall into the same grid cell belong to the same cluster. Therefore all data points belonging to the same cell can be aggregated and treated as one object [4]. It is due to this nature that grid-based clustering algorithms are computationally efficient which depends on the number of cells in each dimension in the quantized space. It has many advantages such as the total number of the grid cells is independent of the number of data points and is insensitive of the order of input data points. Some of the popular grid-based clustering techniques are STING [4], WaveCluster [5], CLIQUE [6], pMAFIA [7] etc. STING [4] uses a multi resolution approach to perform cluster analysis. The advantage of STING is that it is query-independent and easy to parallelize. However the shapes of clusters have horizontal or vertical boundaries but no diagonal boundary is detected. WaveCluster [5] also uses a multidimensional grid structure. It helps in detecting clusters of data at varying levels of accuracy. It automatically removes outliers and is very fast. However, it is not suitable for high dimensional data sets. CLIQUE [6] is a hybrid clustering method that combines the idea of both density-based and grid-based approaches. It automatically finds subspaces of the highest dimensionality and is insensitive to the order of input. Moreover, it has good scalability as the number of dimensions in the data increases. However, the accuracy of the clustering result may be degraded at the expense of simplicity of the method. pMAFIA[7] is an optimized and improved version of CLIQUE. It uses the concept of adaptive grids for detecting the clusters. It scales exponentially to the dimension of the cluster of the highest dimension in the data set.

C. Clustering Over Multi Density Data Space: One of the main applications of clustering spatial databases is to find clusters of spatial objects which are close to each other. Most traditional clustering algorithms try to discover clusters of

arbitrary densities, shapes and sizes. Very few clustering algorithms show preferable efficiency when clustering multi-density datasets. This is also because small clusters with small number of points in a local area are possible to be missed by a global density threshold. Some clustering algorithms that can cluster on multi-density datasets are Chameleon [8], SNN [9] (shared nearest neighbor), and the multi-stage density-isoline algorithm and so on. Chameleon [8] can handle multi-density datasets, but for large datasets the time complexity is too high. SNN [9] algorithm can find clusters of varying shapes, sizes and densities and can also handle multi-density dataset. The disadvantage of SNN is that the degree of precision is low on the multi density clustering and finding outliers. The multi-stage density-isoline algorithm [10] clusters datasets by the multi-stage way and the idea of density-isoline. The disadvantage of the algorithm is that each cluster cannot be separated efficiently. DGCL [11] is based on density-grid based clustering approach. But, since it uses a uniform density threshold it causes the low density clusters to be lost.

D. Clustering Over Variable Density Space: Most of the real life datasets have a skewed distribution and may also contain nested cluster structures the discovery of which is very difficult. Therefore, we discuss two density based approaches, OPTICS [12] and EnDBSCAN [13], which attempt to handle the datasets with variable density successfully. OPTICS can identify embedded clusters over varying density space. However, its execution time performance degrades in case of large datasets with variable density space and it cannot detect

nested cluster structures successfully over massive datasets. In EnDBSCAN [13], an attempt is made to detect embedded or nested clusters using an integrated approach. Based on our experimental analysis in light of very large synthetic datasets, it has been observed that EnDBSCAN can detect embedded clusters; however, with the increase in the volume of data, the performance of it also degrades. EnDBSCAN is highly sensitive to the parameters MinPts and ϵ . In addition to the above mentioned parameters, OPTICS requires an additional parameter i.e. ϵ'

E. Fuzzy Logic and Fuzzy Clustering:

Fuzzy logic is a type of logic that recognizes more than simple true and false values. With fuzzy logic, propositions can be represented with degrees of truthfulness and falsehood. Fuzzy logic has proved to be particularly useful in expert system and other artificial intelligence applications.

In hard clustering (crisp), data is divided into distinct clusters, where each data element belongs to exactly one cluster. In **fuzzy clustering** (also referred to as **soft clustering**), data elements can belong to more than one cluster, and associated with each element is a set of membership levels. These indicate the strength of the association between that data element and a particular cluster. Fuzzy clustering is a process of assigning these membership levels, and then using them to assign data elements to one or more clusters.

One of the most widely used fuzzy clustering algorithms is the Fuzzy C-Means (FCM) Algorithm[18]. The FCM algorithm attempts to partition a finite collection of n elements $X = \{x_1, \dots, x_n\}$ into a collection of c fuzzy clusters with respect to some given criterion. Given a finite set of data, the algorithm returns a list of c cluster centres $C = \{c_1, \dots, c_n\}$ and a partition matrix

$$W = w_{i,j} \in [0, 1], i=1, \dots, n, j=1, \dots, c,$$

where each element w_{ij} tells the degree to which element x_i belongs to cluster c_j . One of the other fuzzy based clustering algorithms is CDFKM [19].

3. THE PROPOSED WORK

Please The algorithm is divided into two main stages. The first stage incorporates the algorithm TDCT [1] which will provide us with rough or hard clusters. Then we take the crisp clusters formed by TDCT and apply a grid based fuzzy concept to get the refined and soft clusters.

A. Stage I: Implementation of TDCT

The execution of the TDCT algorithm includes the following steps:

1. Creating the octagon taking an arbitrary unclassified point as center and epsilon distance (ϵ) as radius.
2. Compute the number of points in each triangle of an octagon (density).
3. Identify the maximum dense triangle of the octagon.
4. Traverse the neighboring triangles starting from the dense triangle in both directions and finding the ratio between two triangles w.r.t points in them.
5. Merging two triangles if ratio is greater than a certain threshold (β) and mark them as classified.
6. Find the farthest points of the merged triangles and creating the next octagon out of them.
7. Repeat step 2 through 7 till we can classify no more into a cluster and assign *cluster_id*.
8. Repeat step 1 through 8 till whole dataset is classified and all clusters are formed.

The details of the theoretical background of TDCT and its execution steps can be found in [1]

B. Stage II: Fuzzification of TDCT

Phase I: Finding the **Global centroids** of the clusters obtained from TDCT and fuzzifying the clusters.

1. We identify the points in a particular cluster and average their X and Y co-ordinates to get their respective centroids.
2. The dataset is spatially divided into equal size. Initially, the data space is divided into $n \times n$ non-overlapping square grid cells, where n is a user input, and maps the data points to each cell.
3. We then proceed to identify the grid cells that contain the clusters we have received from TDCT. This can be done by identifying the points contained in a particular cluster and then finding out the grid cells containing those points. If we have points from more than one cluster contained in a grid cell, we assign the grid cell to all the respective clusters.
4. We then find the bordering grid cells for each cluster and hence the bordering points for each cluster are found.

The cluster centroids for all the clusters we have defined in step 1 are the cluster representatives C_l . The relationship between a data point and cluster representative is fuzzy. That is, a membership $u_{i,j}$ belongs to $\square [0, 1]$ is used to represent the degree of belongingness of data point \mathbf{X}_i and cluster centroids C_j . Denote the set of data points as $S = \{\mathbf{X}_i\}$.

With respect to the cluster representatives C_j and memberships $u_{i,j}$, where N is the number of data points (the bordering points of a the clusters); m is the fuzzifier parameter; k is the number of clusters (known in our case, as found by TDCT); and d_{ij} is the squared Euclidean distance between data point \mathbf{X}_i and cluster representative C_j . It is noted that $u_{i,j}$ should satisfy the following constraint:

$$\sum_{j=1}^k u_{i,j} = 1$$

, for $i=1$ to N

Now we can apply fuzzification in case of the bordering points of each cluster found in the following way:

1. Input the set of cluster centroids $SC0 = \{C_j(0)\}$ and Set $p = 1$.
2. Given the set of cluster centroids SCp , compute d_{ij} for $i = 1$ to N and $j = 1$ to k . Update memberships $u_{i,j}$ using the following equation:

$$u_{i,j} = \left((d_{if})^{\frac{1}{m}-1} \sum_{l=1}^k \left(\frac{1}{d_{il}} \right)^{\frac{1}{m}-1} \right)^{-1}$$

3. We obtain the memberships of the bordering points w.r.t the cluster centroids. Hence we can assign the bordering points to the clusters according to their level of memberships with the clusters.

But in some cases the membership we obtained w.r.t the cluster centroids may not be the correct membership of the data point. This happens especially in case of skewed patterns.

This needs special analysis. Hence, the concept of local centroid.

Phase II: **Local Centroid discovery** and accurate fuzzification.

1. The bordering grid cells obtained from **phase I** are revisited. We find the centroid of the points lying in a particular grid cell of the border. Hence we find the centroids of all the bordering grid cells of the clusters obtained from TDCT.

2. We then find the memberships of all the neighboring points of the centroids obtained. The grid cells with the centroids have four neighbors (the cells sharing the four sides of a grid cell are its neighbor). All these neighboring points contained in the neighboring cells are taken into account. The memberships are calculated as mentioned in **phase I**.

Finally, we obtain the memberships of the bordering points and assign them to the clusters according to their level of memberships.

4. RESULTS AND PERFORMANCE ANALYSIS

To evaluate the technique in terms of quality of clustering, the algorithm was also applied on the Chameleon t4.8k.dat, t5.8k.dat, t8.8k.dat and t7.10k.dat datasets [9]. Fig. 2(a),2(b),2(c) & 2(d) shows the result of clustering in Chameleon t4.8k.dat, t7.10k.dat, t8.8k.dat and t5.8k.dat datasets respectively.

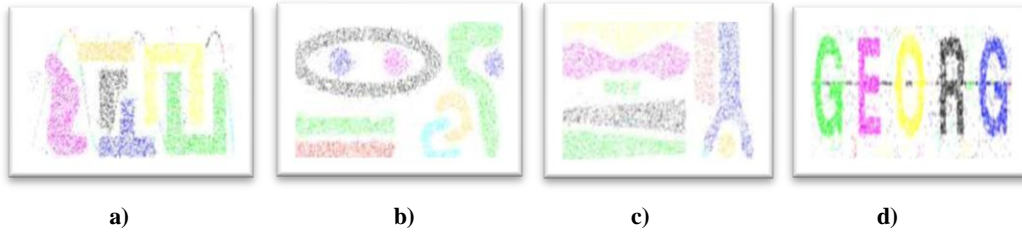


Figure 2(a),2(b),2(c) & 2(d)- Result of clustering in Chameleon datasets respectively

5. COMPLEXITY ANALYSIS

The complexity of TDCT is $O(n_c \times 2^*m*N)$. The partitioning of the dataset into $n \times n$ non overlapping cells results in a complexity of $O(N)$ where N is the total number of data points. $O(k)$ is the time complexity of finding the grid cells of

a cluster where k is the total number of grid cells containing a complete cluster. $O(l)$ is the time complexity of finding the bordering cells where l is the total number of bordering cells found. Thus the complexity of STDCT is $O(n_c \times 2^*m*N*k*l)$.

Table 1. Comparison of the Proposed Algorithm (STDCT) with its counterparts

Approach	Sr. No	Algorithm	No. Of Parameter	Optimized For	Structure	Multi-Density Cluster	Embedded Cluster	Complexity	Noise Handling
Partition Approach	1	K-means	No of Clusters	Separated Clusters	Spherical	No	No	$O(l^kN)$	No
	2	K-Medoids	No of Clusters	Separated Clusters, Large valued objects	Spherical	No	No	$O(k(N-k)^2)$	No
	3	K-modes	No. of Clusters	Separated Clusters, Large Datasets	Spherical	No	No	$O(l^k(N-k)^2)$	No
	4	FCM (Fuzzy C-means Clustering)	No. of Clusters	Separated Clusters	Non-convex shapes	No	No	$O(N)$	No
	5	PAM (Partition Around Medoids)	No. of Clusters	Separated Clusters, Large Datasets	Spherical	No	No	$O(l^k(N-k)^2)$	No
	6	CLARA (Clustering Large Applications)	No. of Clusters	Relatively Large Datasets	Spherical	No	No	$O(ksz^2+k(N-k))$	No

	7	CLARANS (A Clustering Algorithm based on Randomized Search)	No. of Clusters, Maximum no. of neighbors	Better than PAM & CLARA	Spherical	No	No	$O(kN^2)$	No
Hierarchical Approach	1	BIRCH (Balanced Iterative Reducing & Clustering using hierarchies)	Branching factor, Diameter, Threshold	Large Data	Spherical	No	No	$O(N)$	Yes
	2	CURE (Clustering Using Representatives)	No. of Clusters, No of representatives	Any Shaped Large data	Arbitrary	No	No	$O(N^2 \log N)$	Yes
	3	ROCK (Robust Clustering using links)	No. of Clusters	Small noisy data	Arbitrary	No	No	$O(N^2 + Nm_m m_a + N^2 \log N)$	Yes
	4	CHAMELEON	$3(k\text{-nearest neighbors, MIN-SIZE, } \alpha^c)$	Small datasets	Arbitrary	Yes	No	$O(N^2)$	Yes
Density Based Approach	1	DBSCAN (Density Based Spatial Clustering of Applications with Noise)	$2(\text{MinPts}, \epsilon)$	Large datasets	Arbitrary	No	No	$O(N \log N)$ using R^* tree	Yes
	2	OPTICS (Ordering Points To Identify the Clustering Structure)	$3(\text{MinPts}, \epsilon, \epsilon')$	Large datasets	Arbitrary	Yes	Yes	$O(N \log N)$ using R^* tree	Yes
	3	DENCLUE	$2(\text{MinPts}, \epsilon)$	Large datasets	Arbitrary	No	No	$O(N \log N)$ using R^* tree	Yes

	4	TDCT (Triangle-Density Clustering Technique)	$2(\epsilon, \beta)$	Large Spatial datasets	Arbitrary	Yes	Yes	$O(n_c \times 2^m * N)$	Yes
	5	3D-CATD (3-Dimensional Clustering Algorithm using Tetrahedron Density) [20]	$2(\epsilon, \beta)$	Large datasets, 3D datasets	Arbitrary	Yes	Yes	$O(n_c \times m * N)$	Yes
Grid-Based Approach	1	Wave Cluster	No. of cells for each dimension, No. of applications of transform	Any Shape, Large Data	Any	Yes	No	$O(N)$	Yes
	2	STING	No. of cells in lowest level, No. of objects in cell	Large spatial datasets	Vertical and horizontal boundary	No	No	$O(N)$	Yes
	3	CLIQUE	Size of the grid, minimum no. of points in each grid cell	High dimensional, Large datasets	Arbitrary	No	No	$O(N)$	Yes
	4	MAFIA	Size of the grid, minimum no. of points in each grid cell	High dimensional, Large datasets	Arbitrary	No	No	$O(c^k)$	Yes
Grid-Density Hybrid Approach	1	GDCT (Grid-Density Clustering Technique)	$2(n, \beta)$	Large datasets, 2D datasets	Arbitrary	Yes	Yes	$O(N/k+t)$	Yes
	2	GDCT Using Distributed Computin	$2(n, \beta)$	Large datasets, 2D datasets	Arbitrary	Yes	Yes	$O(N)$	Yes

		g							
	3	DisClus (Distributed Clustering)	2 (n,α)	High resolution multi-spectral Satellite Datasets	Arbitrar y	Yes	Ye s	$O(N)$	Yes
Graph Based Clustering	1	AUTOCLUST	NIL	Massive Data	Arbitrar y	No	No	$O(N \log N)$	Yes
Fuzzy Based Technique	1	STDCT(Soft Triangle Density Clustering Technique)	$2(\epsilon, \beta)$	Large Spatial datasets	Arbitrar y	Yes	Ye s	$O(n_c \times 2^*m^*N^*k^*l)$	Yes

Advantages of the proposed algorithm, STDCT:

- 1) Embedded cluster Detection
- 2) $O(n_c \times 2^*m^*N^*k^*l)$ complexity
- 3) Handling of huge datasets
- 4) Handling of single linkage problem
- 5) Lesser number of parameters than its counterparts
- 6) Soft clusters are formed by incorporating fuzzy membership functions

6. CONCLUSION

This paper presents a clustering technique for massive numeric datasets. The clustering algorithm is based on a Fuzzy Density-grid approach and can detect global as well as embedded clusters. Experimental results are reported to establish the superiority of the algorithm in light of several synthetic data sets.

In this paper we have only considered two-dimensional objects. But, spatial databases also contain extended objects such as polygons. Therefore, there is scope for scaling the proposed algorithm to detect clusters in such datasets with minor modifications, research of which is in progress. From a proper analysis of the designed technique, it can be safely concluded that the algorithm developed is working properly to a great extent

7. REFERENCES

- [1] Hrishav Bakul Barua, Dhiraj Kumar Das and Sauravjyoti Sarmah, "A Density Based Clustering Technique For Large Spatial Data Using Polygon Approach", TDCT, *IOSR Journal of Computer Engineering (IOSRJCE)* ISSN: 2278-0661 Volume 3, Issue 6 (July-Aug. 2012), PP 01-10.
- [2] J. Han and M. Kamber, *Data Mining: Concepts and Techniques*. India: Morgan Kaufmann Publishers, 2004.
- [3] M. Ester, H. P. Kriegel, J. Sander and X. Xu, "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise", in *International Conference on Knowledge Discovery in Databases and Data Mining (KDD-96)*, Portland, Oregon, 1996, pp. 226-231.
- [4] W. Wang, J. Yang, and R. R. Muntz, "STING: A Statistical Information Grid Approach to Spatial data Mining", in *Proc. 23rd International Conference on Very Large Databases, (VLDB)*, Athens, Greece, Morgan Kaufmann Publishers, 1997, pp. 186 - 195.
- [5] G. Sheikholeslami, S. Chatterjee and A. Zhang, "Wavecluster: A Multiresolution Clustering approach for very large spatial database", in *SIGMOD'98*, Seattle, 1998.
- [6] R. Agrawal, J. Gehrke, D. Gunopulos and P. Raghavan, "Automatic subspace clustering of high dimensional data for data mining applications", in *SIGMOD Record ACM Special Interest Group on Management of Data*, 1998, pp. 94–105.
- [7] H. S. Nagesh, S. Goil and A. N. Choudhary, "A scalable parallel subspace clustering algorithm for massive data sets", in *Proc. International Conference on Parallel Processing*, 2000, pp. 477.
- [8] L. Ertoz, M. Steinbach and V. Kumar, "Finding Clusters of Different Sizes, Shapes, and Densities in Noisy, High Dimensional Data", in *SIAM International Conference on Data Mining (SDM '03)*, 2003.
- [9] G. Karypis, Han and V. Kumar, "CHAMELEON: A hierarchical clustering algorithm using dynamic modeling", *IEEE Computer*, 32(8), pp 68-75, 1999.
- [10] Y. Zhao, S. Mei, X. Fan, S. Jun-de. 2003. Clustering Datasets Containing Clusters of Various Densities. *Journal of Beijing University of Posts and Telecommunications*, 26(2):42-47.
- [11] H. S. Kim, S. Gao, Y. Xia, G. B. Kim and H. Y. Bae, "DGCL: An Efficient Density and Grid Based Clustering Algorithm for Large Spatial Database", *Advances in*

- Web-Age Information Management (WAIM'06)*, pp. 362-371, 2006.
- [12] M. Ankerst, M. M. Breuing, H. P. Kriegel and J. Sander, "OPTICS: Ordering Points To Identify the Clustering Structure", in *ACMSIGMOD*, pp. 49-60, 1999.
- [13] S. Roy and D. K. Bhattacharyya, "An Approach to Find Embedded Clusters Using Density Based Techniques", in *Proc. ICDCIT*, LNCS 3816, pp. 523-535, 2005.
- [14] S. Sarmah, R. Das and D. K. Bhattacharyya, "Intrinsic Cluster Detection Using Adaptive Grids", in *Proc. ADCOM'07*, Guwahati, 2007.
- [15] S. Sarmah, R. Das and D.K. Bhattacharyya, "A Distributed Algorithm for Intrinsic Cluster Detection over Large Spatial Data" A grid-density based clustering Technique (GDCT), World Academy of Science, Engineering and Technology 45, pp. 856-866, 2008.
- [16] Rajib Mall, "Software Engineering".
- [17] Available: <http://steve.hollasch.net/cgindex/math/barycentric.html>
- [18] James C. Bezdek, Robert Ehrlich and William Full, "FCM: The Fuzzy *c*-Means Clustering Algorithm", *Computer & Geosciences*, Vol. 10, No 2-3, pp. 191-203, 1984, printed in USA.
- [19] CHIH-TANG CHANG, JIM Z. C. LAI AND MU-DER JENG, "A Fuzzy K-means Clustering Algorithm Using Cluster Center Displacement", *JOURNAL OF INFORMATION SCIENCE AND ENGINEERING* 27, 995-1009 (2011)
- [20] Hrishav Bakul Barua and Sauravjyoti Sarmah. Article: An Extended Density based Clustering Algorithm for Large Spatial 3D Data using Polyhedron Approach. *International Journal of Computer Applications* 58(2):4-15, November 2012. Published by Foundation of Computer Science, New York, USA (ISBN: 973-93-80871-32-3), (ISSN: 0975 – 8887)