

# Using an MST based Value for $\epsilon$ in DBSCAN Algorithm for Obtaining Better Result

**Nirmalya Chowdhury**

Department of Computer Science and Engineering, Jadavpur University, Kolkata - 700032, India  
Email: nirmalya\_chowdhury@yahoo.com

**Preetha Bhattacharjee**

Department of Computer Science and Engineering, Jadavpur University, Kolkata - 700032, India  
Email: nirmalya63@gmail.com

**Abstract**—In this paper, an objective function based on minimal spanning tree (MST) of data points is proposed for clustering and a density-based clustering technique has been used in an attempt to optimize the specified objective function in order to detect the “natural grouping” present in a given data set. A threshold based on MST of data points of each cluster thus found is used to remove noise (if any present in the data) from the final clustering.

A comparison of the experimental results obtained by DBSCAN (Density Based Spatial Clustering of Applications with Noise) algorithm and the proposed algorithm has also been incorporated. It is observed that our proposed algorithm performs better than DBSCAN algorithm. Several experiments on synthetic data set in  $\mathcal{R}^2$  and  $\mathcal{R}^3$  show the utility of the proposed method. The proposed method has also found to provide good results for two real life data sets considered for experimentation. Note that  $K$ -means is one of the most popular methods adopted to solve the clustering problem. This algorithm uses an objective function that is based on minimization of squared error criteria. Note that it may not always provide the “natural grouping” though it is useful in many applications.

**Index Terms**— Clustering, Natural Grouping, DBSCAN Algorithm

## I. INTRODUCTION

A lot of scientific effort has already been dedicated to cluster analysis problems, which attempts to extract the “natural grouping”, present in a given data set.

Hence, clustering is useful in that it can lead to the discovery of previously unknown groups within the data. The intuition behind the phrase “natural groups” is explained below in the context of data set in  $\mathcal{R}^2$ .

For a given data set  $S = \{X_1, X_2, \dots, X_n\} \subset \mathcal{R}^m$ .

what one perceives to be the groups present in  $S$  by viewing the scatter diagram of  $S$ , is termed as natural groups of  $S$ . Clustering techniques [1, 2, 3, 4, 5] aim to extract such natural groups present in a given data set and each such group is termed as a cluster. So we shall use the term “cluster” or “group” interchangeably in this paper. The existing clustering techniques may not always find the natural grouping.

In this paper, we have proposed an objective function for clustering based on MST of data points for each group

and a density-based clustering technique has been used in an attempt to optimize the specified objective function in order to detect the “natural grouping” present in a given data set. The method for obtaining the natural groups in  $\mathcal{R}^2$  can also be extended to  $\mathcal{R}^P$  ( $P > 2$ ). Most of the real life data sets exhibit overlapping clusters. Thus we have applied this method on synthetic data sets in  $\mathcal{R}^2$  and  $\mathcal{R}^3$ , and obtained good results. Note that the perception of natural groups in a data set is not possible for higher dimensional data. But the concept used for detecting the groups in  $\mathcal{R}^2$  may also be applicable to higher dimensional data to obtain a “meaningful” grouping.

In most of the real-life problems, sometimes a large variation in feature measurements are found and noise effect involved in the measuring instruments cannot be avoided. Quite often, an outlier is due to noise in the measurement process or error in data coding. Thus elimination of noise and outliers is one of the important tasks in clustering. Most of the widely used algorithms in partitional clustering such as  $K$ -means and basic Isodata cannot take care of this problem. Thus noise is retained in the final clustered output of such algorithms. The proposed clustering method using modified DBSCAN algorithm presented in this paper uses a parameter  $\epsilon$  based on the sum of edge weights of MST of data points to remove noise from final clustering.

The next section describes the original DBSCAN algorithm proposed by Martin Ester et al., along with all the proposed modification of the standard algorithm by different researcher.

The formulation of the problem is presented in the section III. Section IV describes DBSCAN algorithm. Our proposed method is presented in the form of an algorithm in section V. Experimental results can be found in section VI. Concluding remarks have been incorporated in section VII.

## II. A REVIEW OF STANDARD DBSCAN ALGORITHM AND ITS PROPOSED MODIFICATION

In the literature, one of the widely used density based clustering method is the DBSCAN (Density Based Spatial Clustering of Applications with Noise). In 1996

Martin Ester et al. [6] presented DBSCAN which is based on "...relying on a density-based notion of clusters which is designed to discover clusters of arbitrary shape". DBSCAN groups dense and nearby patterns together to form clusters. This approach can automatically find noisy outliers which are not achieved in most of the other clustering methods. In 1998 Martin Ester et al. [7] introduced Incremental DBSCAN algorithm which is based on the DBSCAN and is used for incremental updates of a clustering after insertion of a new object to the database and deletion of an existing object from the database. A. Hinneburg et al., 1998 introduced the DENCLUE [8] algorithm. It uses kernel density estimation. The result of density function gives the local density maxima value and this local density value is used to form the clusters. If the local density value is very small, the objects of clusters will be discarded as noise. M. Ankerst et al., 1999 presented OPTICS (Ordering Points to Identify the Clustering Structure) [9] was based on the concepts of DBSCAN algorithm and it can identify nested clusters and the structure of clusters. Sampling-based DBSCAN, SDBSCAN [10] introduced by A. Zhou et al., 2000, runs DBSCAN on a randomly selected subset of objects to form clusters to reduce execution time of DBSCAN by using a subset of data set, which is known as sampling. FDBSCAN [9] and IDBSCAN [11] reduce the execution time by utilizing smart core detection methods. SDBDC (Scalable Density-Based Distributed Clustering) [12] method also uses DBSCAN algorithm on both local sites and global site to cluster distributed objects. Partitioning the data set reduces the search space. This technique can improve execution time of DBSCAN. El-Sonbaty et al., 2004 [13] proposed CLARANS (Clustering Large Applications based on Randomized Search) can be used to partition a data set. SNN density-based clustering algorithm [14] is also based on DBSCAN and it is applicable to high-dimensional data consisting of time series data of atmospheric pressure at various points on the earth. KIDBSCAN [15] locates the high-density areas using k-means and introduces core points with respect to density rankings. ST-DBSCAN [16] is based on extensions to DBSCAN related with the identification of (i) core objects, (ii) noise objects, and (iii) adjacent clusters. In contrast to the existing density-based clustering algorithms, this algorithm has the ability of discovering clusters according to non-spatial, spatial and temporal values of the objects. Rough-DBSCAN [17] employs the well-known Leader clustering algorithm and rough set theory for sampling and then producing approximate clusters. Using low sampling rates for obtaining good execution time can degrade the accuracy substantially. SPARROW (Spatial Clustering Algorithm through Swarm Intelligence) [18] reduces the number of core points using a method that is inspired by the flocking mechanism of birds. The MEDBSCAN [19] algorithm has been proposed recently to improve the performance of DBSCAN algorithm, at the same time without losing the result accuracy. An Optimized Density Based Clustering Algorithm [20] has been used to overcome the

problem of the performance issue which exists in the density based clustering algorithms. The existing algorithms, a Fast DBSCAN Algorithm and Memory effect in DBSCAN algorithm, have been combined in the new solution to speed up the performance as well as improve the quality of the output. In this algorithm, number of Region-Query call has been reduced and Region-Query calls speed has been improved. Chen et al., 2011 [21] developed enhanced DBSCAN algorithm to detect anomalies. This algorithm can be used for mass data processing. DBSCAN\_BV [22] improve DBSCAN's execution time performance for binary data sets and Hamming distances. This algorithm achieves considerable speed gains by using a novel pruning technique, as well as bit vectors and binary operations. A new hybrid algorithm (PACA-DBSCAN) was proposed by Jiang et al., 2011 [23] based on partitioning-based DBSCAN and modified ant clustering algorithms. It can partition a database into  $N$  partitions according to the density of data, and then cluster each partition with DBSCAN. Superior to DBSCAN and PDBSCAN, the new hybrid algorithm reduces the sensitivity to the initial parameters, and it can deal with data of uneven density very well. GRPDBSCAN (Grid-based DBSCAN Algorithm with Referential Parameters) [24] was proposed by Darong et al., 2012, which combined the grid partition technique and multi-density based clustering algorithm. DBSCAN-GM [25] combines Gaussian-Means and DBSCAN algorithm. The idea of DBSCAN-GM is to cover the limitations of DBSCAN, by exploring the benefits of Gaussian-Means. It runs Gaussian-Means to generate small clusters with determined cluster centers, in order to estimate the values of DBSCAN's parameters. Edla et al., 2012 [26], proposed a DBSCAN method to cluster the gene expression data. This algorithm solves quadratic computational complexity of DBSCAN by using the prototypes produced from a squared error clustering method such as K-means. Then, the DBSCAN technique is applied efficiently using these prototypes. Tran et al. 2012 [27] proposed a revised DBSCAN algorithm to cluster data with dense adjacent clusters. It retains the key properties of the original DBSCAN algorithm and in addition it has the potential to improve clustering results by solving the issue of border objects. After the detection of all clusters, border objects are assigned to their closest core-density-reachable chains that represent a given cluster.

### III. DESCRIPTION OF THE PROBLEM

Clustering is an unsupervised technique used in discovering inherent structure present in the set of objects [1]. Clustering algorithms attempt to organize unlabeled pattern vectors into clusters or "natural groups" such that points within a cluster are more similar to each other than to points belonging to different clusters.

Let the set of patterns  $M$  be  $\{x_1, x_2, \dots, \dots, x_m\}$

Where  $x_i$  is the  $i_{th}$  pattern vector, Let the number of clusters be  $K$ . If the clusters are represented by  $C_1, C_2, \dots, \dots, C_K$  then

- P1.  $C_i \neq \phi$ , for  $i = 1, 2, \dots, K$   
 P2.  $C_i \cap C_j = \phi$ , for  $i \neq j$  and  
 P3.  $\cup_{i=1}^K C_i = M$ , where  $\phi$  represents null set.

Clustering techniques may broadly be divided into two categories: hierarchical and non-hierarchical [1]. The non-hierarchical or partitional clustering problem deals with obtaining an optimal partition of  $M$  into  $K$  subsets such that some clustering criterion is satisfied. Among the non-hierarchical clustering techniques, the  $K$ -means (or C-means or basic Isodata) algorithm has been one of the more widely used algorithms. DBSCAN algorithm makes use of the concept of density-reachability to extend a particular cluster. A cluster is formed by DBSCAN for all points that are density reachable.

In the clustering process, DBSCAN starts with an arbitrary point  $p$  and retrieves all points density-reachable from point  $p$  with respect to a given  $eps$  and  $MinPts$ . The said two parameters are given arbitrary values initially.

Because the process of expanding a cluster or merging clusters in DBSCAN relies on the density-reachability mechanism, some of the resulting clusters may be in non-convex or elongated shape. The proposed algorithm is

also able to form clusters in non-convex and elongated shapes, but it removes the problem of varying density in DBSCAN [2]. It was also reported that our proposed algorithm has outperformed DBSCAN in a number of data set [2].

A cluster is formed when all of these points are density-reachable, although some points are not directly density-reachable by some other points. For instance in Fig.1, the points  $p_1, p_2, \dots, p_{12}$ , and  $q$  is not directly density-reachable by the point  $p$ . Centering at the point  $p$ , a circle defined by a given  $eps$ -neighborhood containing at least two other points. The point  $p_1$  is directly density-reachable from  $p$ , and it is used as a center point to draw the next circle with the same  $eps$ -neighborhood. The same neighborhood-forming mechanism repeatedly applies to point  $p_2$  (which is directly density-reachable from  $p_1$ )  $p_3, \dots, p_{12}$  and  $q$  in sequence, as shown by the gray circles. However, in Web opinion clustering [30], it may create problems. Some of the Web opinions may be directly density-reachable to other Web opinions due to some connection in the weak concepts; that is, two clusters may be put together even though they are not as relevant to each other.

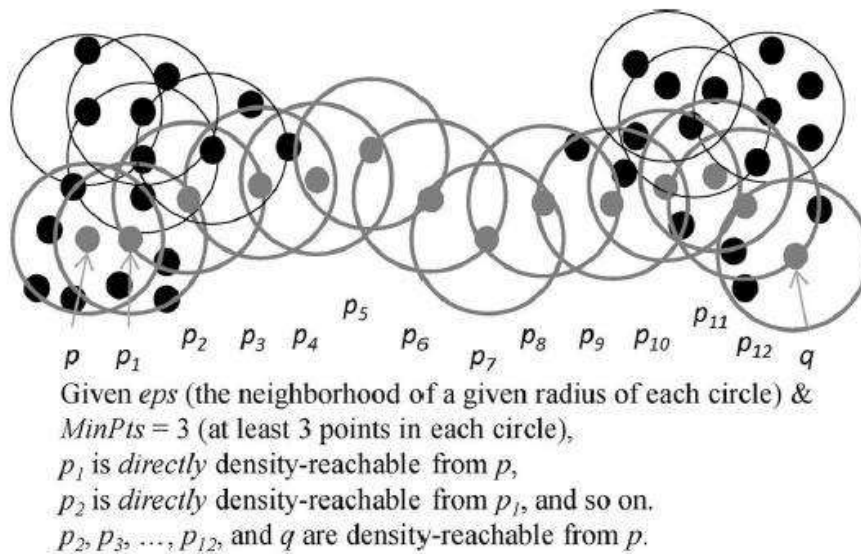


Fig. 1. Clustering weakly relevant concepts at both left and right end in an elongated shape of DBSCAN-identified cluster

The standard DBSCAN algorithm is presented in the next section.

#### IV. DBSCAN (DENSITY BASED SPATIAL CLUSTERING OF APPLICATIONS) ALGORITHM

The algorithm is designed to discover the clusters and the noise in a spatial database. To find a cluster, DBSCAN starts with an arbitrary point  $p$  and retrieves all points density-reachable from  $p$  with respect to  $eps$ -neighbourhood and  $MinPts$ .

**Algorithm: DBSCAN:** A density-based clustering algorithm.

#### Input:

$D$ : a data set containing  $n$  objects,  
 $eps$ -neighborhood: the radius parameter, and  
 $MinPts$ : the neighborhood density threshold.

**Output:** A set of density-based clusters.

#### Method:

- (1) mark all objects as unvisited;
- (2) **do**
- (3) randomly select an unvisited object  $p$ ;
- (4) mark  $p$  as visited;
- (5) **If** the  $eps$ -neighborhood of  $p$  has at least  $MinPts$  objects

```

(6) create a new cluster  $C$ , and add  $p$  to  $C$ ;
(7) let  $N$  be the set of objects in the  $eps$ -neighborhood of  $p$ ;
(8) for each point  $p'$  in  $N$ 
(9) if  $p'$  is unvisited
(10) mark  $p'$  as visited;
(11) if the  $eps$ -neighborhood of  $p'$  has at least  $MinPts$  points,
add those points to  $N$ ;
(12) if  $p'$  is not yet a member of any cluster,
add  $p'$  to  $C$ ;
(13) end for
(14) output  $C$ ;
(15) else mark  $p$  as noise;
(16) until no object is unvisited;

```

The next section describes the proposed modified version of the DBSCAN algorithm (MDBSCAN).

### V. THE PROPOSED ALGORITHM (MDBSCAN)

There are several ways in which a given data set can be clustered. In this paper, we have presented a modified version of DBSCAN algorithm where we have taken the value of  $eps$ -neighborhood to be  $(2.5*t)$ , where  $t = (L_N/N)^{1/p}$  where  $L_N$  = sum of edge weights of MST,  $N$  = number of data points and  $p$  is the dimensionality. We have taken the value of another important parameter namely  $MinPts$  to be 5% of  $N$ .

1. Let  $D = \{X_1, X_2, \dots, X_n\} \in \mathcal{R}^m (m > 2)$ , where initially all data objects are marked "unvisited".

Repeat:

2. We randomly select an unvisited object  $p$  and mark  $p$  as "visited" and check whether the  $eps$ -neighborhood of  $p$  contains at least  $Minpts$  objects, "visited" or "unvisited".
  - i. If not,  $p \in NS$ , where  $NS$  is a noise set.
  - ii. Otherwise, all objects in all  $\epsilon$ -neighborhood of  $p$  are marked as "visited" and
    - a. A new cluster  $C = \{y_1, y_2, \dots, y_k, p\} \in \mathcal{R}^m$ , is created for  $p$  if the objects in  $eps$ -neighborhood of  $p$  do not overlap with any clusters previously generated.
    - b. Else objects in  $eps$ -neighborhood of  $p$  are merged with the previously obtained cluster.
3. Objects in the noise set  $NS$  are excluded from the set if they belong in the  $eps$ -neighborhood of an "unvisited" core point.

Until all objects in the dataset  $D$  are "visited".

4. Objects that remain in the noise set are finally termed as noise points.

### VI. EXPERIMENTAL RESULT

The experimental results of the proposed method on various synthetic data sets are presented in this section.

Fig. 2(a) shows a data set of size 1200 where data points are generated from two clusters. One cluster is having the shape of a rectangle while the other is having the shape of the English letter 'P' enclosed within that rectangle. The clustering provided by the proposed method is as shown in Fig. 2(b). It can be seen from Fig. 2(b) that the proposed method has indeed detected the two natural groups present in the given data set. Note that the proposed method is also successful in removing noise from the given data.

Fig. 3(a) shows a data distribution of size 800 where data points are generated from two clusters. Both the clusters are having the shape the English letter 'C'. The clustering provided by the proposed method is as shown in Fig. 3(b). From Fig. 3(b) it is evident that the proposed method has successfully detected the two natural groups present in the given data set and the final clustering thus obtained does not contain the noise that was present in the input data.

Fig. 4(a) shows a data set of size 1000 where data points are generated from two clusters. One cluster is having the shape of the English letter 'C' while the other is having the shape of a circular disk. The clustering provided by the proposed method is as shown in Fig. 4(b). By viewing the grouping shown in Fig. 4(b) one can conclude that the proposed method is able to detect the two natural groups present in the given data set. Note that the proposed method is also successful in removing noise from the given data.

The proposed algorithm is also run on a synthetic data distribution in  $\mathcal{R}^3$  of size 450. Here data points are generated from three sets  $[0, 0.5] \times [0, 0.5] \times [0, 0.5]$ ,  $[0, 0.5] \times [0, 0.5] \times [1, 1.5]$  and  $[0, 0.5] \times [1, 1.5] \times [0, 0.5]$ . It is found that after applying the proposed method the desired cluster configuration has been obtained.

Experiments with two real life data sets have also been conducted using the proposed method. The proposed algorithm has been implemented on a set of 150 four dimensional pattern vectors, (Fishers' Iris Data) where the four dimensions represent the four features (Sepal Length, Sepal Width, Petal Length, Petal Width) of flowers originating from three classes [28]. Crude-oil data [29], having 56 data points, 5 features and 3 classes has also been chosen for experimentation. It may be noted that some of the clusters are overlapping in these real life data sets. However our proposed method could successfully group 92.5 % of the crude-oil data and 91.6 % of the iris data.

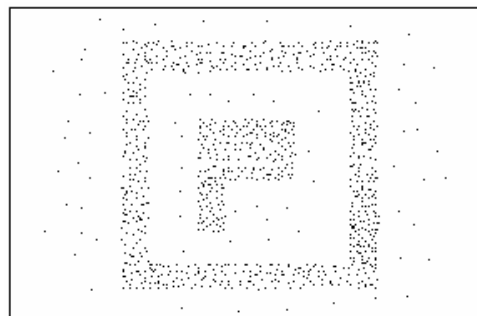


Fig. 2(a). Scatter Diagram with noise

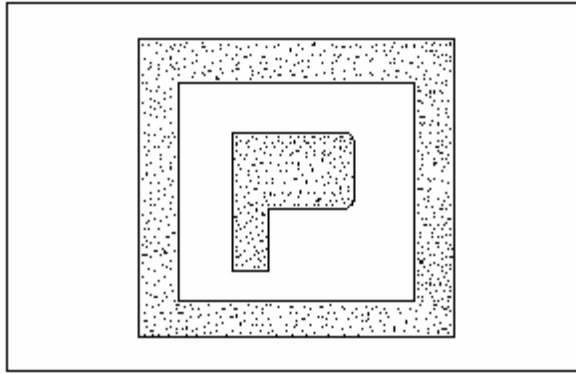


Fig. 2(b). Natural Grouping found by proposed method

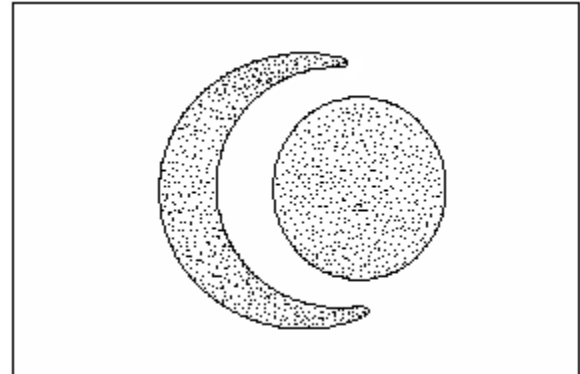


Fig. 4(b). Natural Grouping found by proposed method

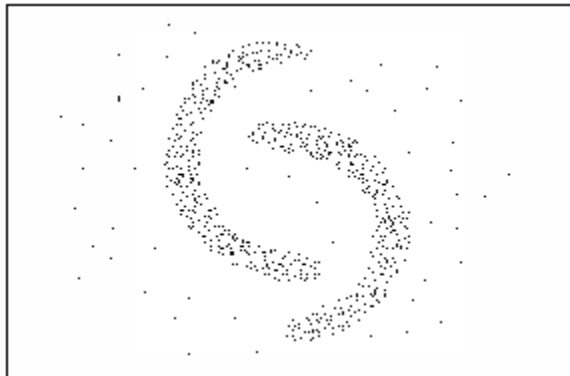


Fig. 3(a). Scatter Diagram with noise

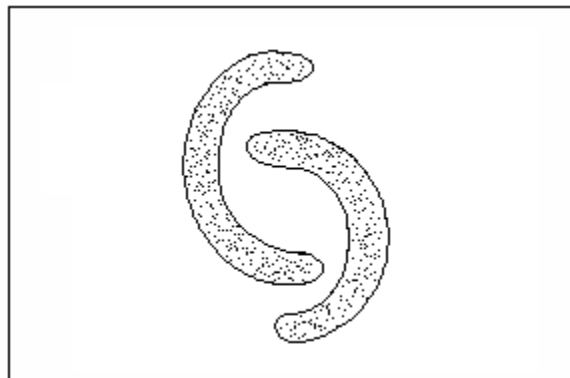


Fig. 3(b). Natural Grouping found by proposed method

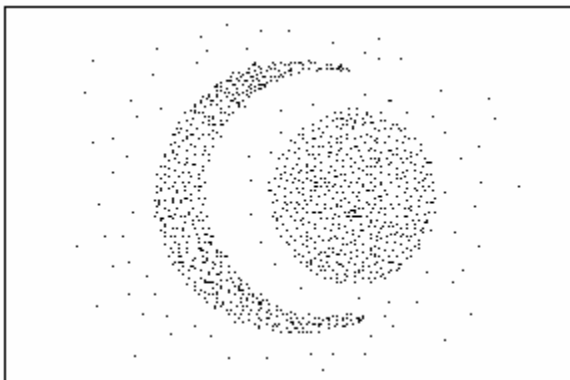


Fig. 4(a). Scatter Diagram with noise

## VII. CONCLUSION

The aim of this work is to observe whether the proposed values of *eps-neighbourhood* and *MinPts* based on the MST of data points in DBSCAN algorithm can lead to detection of natural groups present in a given data. Note that the proposed modified version of the DBSCAN algorithm has successfully provided the natural group present in all the synthetic data set considered for experimentation. The method has also provided good results for the two real life data set used to check the validity of the method. It can also be noted that the proposed method can also remove noise if any to provide the final clustering. So we can conclude that the proposed method has been found to be successful to provide good results for all the synthetic and real life data sets considered for experimentation.

## REFERENCES

- [1] Anderberg, M.R., *Cluster Analysis for Application*, Academic Press, Inc., NewYork, 1973.
- [2] Devijver, P.A. and J. Kittler., *Pattern Recognition: A statistical Approach*, Prentice-Hall International. HemelHemstead, Hertfordshire, UK, 1982.
- [3] Jain, A.K. and R.C. Dubes. *Algorithms for Clustering Data*. Prentice-Hall, Englewood, Cliffs NJ 1988.
- [4] Spath, H., *Cluster Analysis Algorithms*. Ellis Horwood, Chichester.UK, 1980.
- [5] Tou, T.J. and C.R. Gonzalez., *Pattern Recognition Principles*. Addison-Wesley Reading, MA, 1974.
- [6] Ester, M., Kriegel, H.P., Xu, X., 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proc. 2nd ACM SIGKDD, Portland, Oregon, pp. 226–231.
- [7] M. Ester, H. -P. Kriegel, J. Sander, M. Wimmer, and X. Xu, "Incremental clustering for mining in a Data Warehousing environment," Proc. 24th Int. Conf. on Very Large Databases (VLDB'98), New York, USA, 1998, pp. 323-333.
- [8] A. Hinneburg and D. Keim, "An efficient approach to clustering in large multimedia data sets with noise," in 4<sup>th</sup> International Conference on

- Knowledge Discovery and Data Mining, 1998, pp. 58–65.
- [9] M. Ankerst, M.M. Breunig, H.-P. Kriegel, J. Sander, OPTICS: Ordering points to identify the clustering structure, in: Proceedings of ACM SIGMOD International Conference on Management of Data, Philadelphia, PA, 1999, pp. 49–60.
- [10] Zhou, A., Zhou, S., Cao, J., Fan, Y., Hu, Y., 2000. Approaches for scaling DBSCAN algorithm to large spatial databases. *J. Comput. Sci. Technol.* 15 (6), 509–526.
- [11] Borah, B., Bhattacharyya, D., 2004. An improved sampling-based DBSCAN for large spatial databases. In: Proc. of Internat. Conf. on Intelligent Sensing and Information Processing, 2004, pp. 92–96.
- [12] E. Januzaj, H.-P. Kriegel, M. Pfeifle, Scalable density-based distributed clustering, in: Proceedings of PKDD, Pisa, Italy, Lectures Notes in Computer Science, 3202, Springer, 2004, pp. 231–244.
- [13] El-Sonbaty, Y., Ismail, M., Farouk, M., 2004. An efficient density based clustering algorithm for large databases. In: 16th IEEE Internat. Conf. on Tools with Artificial Intelligence, 2004, ICTAI, pp. 673–677.
- [14] P-N. Tan, M. Steinbach, V. Kumar, Introduction to Data Mining, Addison-Wesley, 2005.
- [15] Tsai, C., Liu, C., 2006. KIDBSCAN: A new efficient data clustering algorithm. In: Artificial Intelligence and Soft Computing – ICAISC 2006. Lecture Notes in Computer Science, 2150. Springer, Berlin/Heidelberg, pp. 02–711.
- [16] Birant, Derya, and Alp Kut. "ST-DBSCAN: An algorithm for clustering spatial-temporal data." *Data & Knowledge Engineering* 60.1 (2007): 208-221.
- [17] Viswanath, P., Babu, V.S., 2009. Rough-DBSCAN: A fast hybrid density based clustering method for large data sets. *Pattern Recognition Lett.* 30 (16), 1477–1488.
- [18] Folino, G., Forestiero, A., Spezzano, G., 2009. An adaptive flocking algorithm for performing approximate clustering. *Inform. Sci.* 179 (18), 3059–3078.
- [19] Li Jian; Yu Wei; Yan Bao-Ping; , "Memory effect in DBSCAN algorithm," *Computer Science & Education*, 2009.ICCSE '09. 4th International Conference on , vol., no., pp.31-36, 25-28 July 2009.
- [20] Peter, J. Hencil, and A. Antonysamy. "An Optimised Density Based Clustering Algorithm." *International Journal of Computer Applications* 6.9 (2010): 20-25.
- [21] Chen, Zhenguo, and Yong Fei Li. "Anomaly Detection Based on Enhanced DBScan Algorithm." *Procedia Engineering* 15 (2011): 178-182.
- [22] Mimaroglu, Selim, and Emin Aksehirli. "Improving DBSCAN's execution time by using a pruning technique on bit vectors." *Pattern Recognition Letters* 32.13 (2011): 1572-1580.
- [23] Jiang, Hua, et al. "A new hybrid method based on partitioning-based DBSCAN and ant clustering." *Expert Systems with Applications* 38.8 (2011): 9373-9381.
- [24] Darong, Huang, and Wang Peng. "Grid-based DBSCAN Algorithm with Referential Parameters." *Physics Procedia* 24 (2012): 1166-1170.
- [25] Smiti, Abir, and Zied Elouedi. "Dbscan-gm: An improved clustering method based on gaussian means and dbscan techniques." *Intelligent Engineering Systems (INES), 2012 IEEE 16th International Conference on.* IEEE, 2012.
- [26] Edla, Damodar Reddy, and Prasanta K. Jana. "A Prototype-Based Modified DBSCAN for Gene Clustering." *Procedia Technology* 6 (2012): 485-492.
- [27] Tran, Thanh N., Klaudia Drab, and Michal Daszykowski. "Revised DBSCAN algorithm to cluster data with dense adjacent clusters." *Chemometrics and Intelligent Laboratory Systems* (2012).
- [28] *IRS Data Users Handbook*, Document No. IRS/NRSA/NDC/HB-01/86, NRSA, Hyderabad, India, Sept. 1986.
- [29] Johnson, R. A., and Wichern, D. W. 1982. Applied Multivariate Statistical Analysis. Prentice-Hall, Englewood Cliffs, NJ, 1982.
- [30] Yang, Christopher C., and T. Dorbin Ng. "Analyzing and visualizing web opinion development and social interactions with density-based clustering." *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on* 41.6 (2011): 1144-1155.

#### Authors' Profiles

**Nirmalya Chowdhury** did his Ph.D. in Engineering from Jadavpur University, India in 1997. At present, he is working as Associate Professor in the department of Computer Science and Engineering, Jadavpur University, India. His fields of research include Pattern Recognition, Soft Computing, Natural Language Processing and Bioinformatics.

**Preetha Bhattacharjee** obtained his B.E. (Computer Sc. and Engg.) from Jadavpur University, India in 2013. Her areas of interests are Machine Learning, Data Mining.

**How to cite this paper:** Nirmalya Chowdhury, Preetha Bhattacharjee, "Using an MST based Value for  $\epsilon$  in DBSCAN Algorithm for Obtaining Better Result", *International Journal of Information Technology and Computer Science(IJITCS)*, vol.6, no.6, pp.55-60, 2014. DOI: 10.5815/ijitcs.2014.06.08