Asian Journal of Applied Science and Technology (AJAST)

(Quarterly International Journal) Volume 3, Issue 4, Pages 46-56, October-December 2019

Optics: A Density-Based Algorithm for Discovering Cluster in Large Databases with Noise

R.Nandhakumar[1] & Dr.Antony Selvadoss Thanamani[2]

[1]Assistant Professor, Department of Computer Science, Nallamuthu Gounder Mahalingam College, Pollachi-642001, India.
[2]Associate Professor & Head, Department of Computer Science, Nallamuthu Gounder Mahalingam College, Pollachi-642001, India.

Article Received: 13 April 2019 Article Accepted: 05 August 2019 Article Published: 16 October 2019

ABSTRACT

Cluster analysis is a primary method for database mining. It is either used as a stand-alone tool to get insight into the distribution of a data set, e.g. to focus further analysis and data processing, or as a preprocessing step for other algorithms operating on the detected clusters. Almost all of the well-known clustering algorithms require input parameters which are hard to determine but have a significant influence on the clustering result. Furthermore, for many real-data sets there does not even exist a global parameter setting for which the result of the clustering algorithm describes the intrinsic clustering structure accurately. We introduce a new algorithm for the purpose of cluster analysis which does not produce a clustering of a data set explicitly; but instead creates an augmented ordering of the database representing its density-based clustering structure. This cluster-ordering contains information which is equivalent to the density-based clustering's corresponding to a broad range of parameter settings. It is a versatile basis for both automatic and interactive cluster analysis. We show how to automatically and efficiently extract not only 'traditional' clustering information (e.g. representative points, arbitrary shaped clusters), but also the intrinsic clustering structure. For medium sized data sets, the cluster-ordering can be represented graphically and for very large data sets, we introduce an appropriate visualization technique. Both are suitable for inter- active exploration of the intrinsic clustering structure offering additional insights into the distribution and correlation of the data.

Keywords: *Cluster Analysis, OPTICS, DBSCAN, Reachability and Connectivity.*

1. INTRODUCTION

Larger and larger amounts of data are collected and stored in databases increasing the need for efficient and effective analysis methods to make use of the information contained implicitly in the data. One of the primary data analysis tasks is cluster analysis which is intended to help a user to understand the natural grouping or structure in a data set. Therefore, the development of improved clustering algorithms has received a lot of attention in the last few years. Roughly speaking, the goal of a clustering algorithm is to group the objects of a database into a set of meaningful subclasses. A clustering algorithm can be used either as a stand-alone tool to get insight into the distribution of a data set, e.g. in order to focus further analysis and data processing, or as a preprocessing step for other algorithms which operate on the detected clusters.

Applications of clustering are, for instance, the creation of thematic maps in geographic information systems by clustering feature spaces, the detection of clusters of objects in geographic information systems and to explain them by other objects in their neighborhood [17], or the clustering of a Web-log database to discover groups of similar access patterns which may correspond to different user profiles [7]. Most of the recent research related to the task of clustering has been directed towards efficiency. The more serious problem, however, is effectively, i.e. the quality or usefulness of the result. Although most traditional clustering algorithms do not scale well with the size and/or dimension of the data set, one way to overcome this problem is to use sampling in combination with a clustering algorithm (see e.g. [8]). This approach works well for many applications and clustering algorithms.

The idea is to apply a clustering algorithm A only to a subset of the whole database. From the result of A for the subset, we can then infer a clustering of the whole database which does not differ much from the result obtained by applying A to the whole data set. However, this does not ensure that the result of the clustering algorithm. A actually reflects the natural groupings in the data. There are three interconnected reasons why the affectivity of clustering algorithms is a problem. First, almost all clustering algorithms require values for input parameters which are hard to determine, especially for real-world data sets containing high- dimensional objects. Second, the algorithms are very sensible to these parameter values, often producing very different partitioning of the data set even for slightly different parameter settings. Third, high-dimensional real-data sets often have a much skewed distribution that cannot be revealed by a clustering algorithm using only one global parameter setting.

In this paper, we introduce a new algorithm for the purpose of cluster analysis which does not produce a clustering of a data set explicitly; but instead creates an augmented ordering of the database representing its density-based clustering structure. This cluster-ordering contains information which is equivalent to the density-based clustering corresponding to a broad range of parameter settings. It is a versatile basis for both automatic and interactive cluster analysis. We show how to automatically and efficiently extract not only 'traditional' clustering information (e.g. representative points, arbitrary shaped clusters), but also the intrinsic clustering structure. For medium sized data sets, the cluster-ordering can be represented graphically and for very large data sets, we introduce an appropriate visualization technique. Both are suitable for interactive exploration of the intrinsic clustering structure offering additional insights into the distribution and correlation of the data.

The rest of the paper is organized as follows. Related work on clustering is briefly discussed in section 2. In section 3, the basic notions of density-based clustering are defined and our new algorithm OPTICS to create an ordering of a data set with respect to its density-based clustering structure is presented. The application of this cluster-ordering for the purpose of cluster analysis is demonstrated in section 4. Both, automatic as well as interactive techniques are discussed. Section 5 concludes the paper with a summary and a short discussion of future research.

2. RELATED WORK

Existing clustering algorithms can be broadly classified into hierarchical and partitioning clustering algorithms (see e.g. [14]). Hierarchical algorithms decompose a database D of n objects into several levels of nested partitioning (clustering), represented by a dendrogram, i.e. a tree that iteratively splits D into smaller subsets until each subset consists of only one object. In such a hierarchy, each node of the tree represents a cluster of D. Partitioning algorithms construct a flat (single level) partition of a database D of n objects into a set of k clusters such that the objects in a cluster are more similar to each other than to objects in different clusters.

The Single-Link method is a commonly used hierarchical clustering method. Starting with the clustering obtained by placing every object in a unique cluster, in every step the two closest clusters in the current clustering are merged until all points are in one cluster. Other algorithms which in principle produce the same hierarchical structure have also been suggested (see e.g. [14], [12]).

Another approach to hierarchical clustering is based on the clustering properties of spatial index structures. The GRID and the BANG clustering apply the same basic algorithm to the data pages of different spatial index structures. A clustering is generated by a clever arrangement of the data pages with respect to their point density. This approach, however, is not well suited for high-dimensional data sets be- cause it is based on the effectively of these structures as spatial access methods. It is well-known that the performance i.e. the clustering properties of spatial index structures degenerate with increasing dimensionality of the data space (e.g. [3]). Recently, the hierarchical algorithm CURE has been proposed in [10]. This algorithm stops the creation of a cluster hierarchy if a level consists of k clusters where k is one of several in- put parameters. It utilizes multiple representative points to evaluate the distance between clusters, thereby adjusting well to arbitrary shaped clusters and avoiding the single-link effect.

This results in a very good clustering quality. To improve the scalability, random sampling and partitioning (pre-clustering) are used. The authors do provide a sensitivity analysis using one synthetic data set, showing that although some parameters can be varied without impacting the quality of the clustering. The parameter setting does have a profound influence on the result. Optimization based partitioning algorithms typically represent clusters by a prototype. Objects are assigned to the cluster rep- resented by the most similar (i.e. closest) prototype. An iterative control strategy is used to optimize the whole clustering such that, e.g., the average or squared distances of objects to its prototypes are minimized. Consequently, these clustering algorithms are effective in determining a good clustering if the clusters are of convex shape, similar size and density, and if their number k can be reasonably estimated.

Depending on the kind of prototypes, one can distinguish k-means, k-modes and k-medoid algorithms. For k-means algorithms (see e.g. [19]), the prototype is the mean value of all objects belonging to a cluster. The k-modes [13] algorithm extends the k-means paradigm to categorical domains. For k-medoid algorithms (see e.g. [18]), the prototype, called the medoid, is one of the objects located near the "center" of a cluster. The algorithm CLARANS introduced an improved k-medoid type algorithm restricting the huge search space by using two additional user-supplied parameters. It is significantly more efficient than the well-known k-medoid algorithms PAM and CLARA presented in [18], nonetheless producing a result of nearly the same quality.

Density-based approaches apply a local cluster criterion and are very popular for the purpose of database mining. Clusters are regarded as regions in the data space in which the objects are dense, and which are separated by regions of low object density (noise). These regions may have an arbitrary shape and the points inside a region may be arbitrarily distributed.

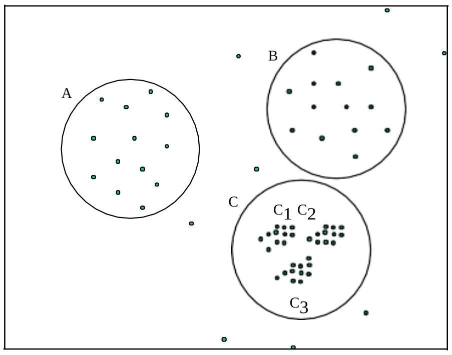
A common way to find regions of high-density in the dataspace is based on grid cell densities [14]. A histogram is constructed by partitioning the data space into a number of non-overlapping regions or cells. Cells containing a relatively large number of objects are potential cluster centers and the boundaries between clusters fall in the "valleys" of the histogram. The success of this method depends on the size of the cells which must be specified by the user. Cells of small volume will give a very "noisy" estimate of the density, whereas large cells tend to overly smooth the density estimate. In [6], a density-based clustering method is presented which is not grid-based. The basic idea for the algorithm DBSCAN is that for each point of a cluster the neighborhood of a given radius (Ɛ) has to contain at least a minimum number of points (MinPts) where Ɛ and MinPts are input parameters.

Another density-based approach is WaveCluster, which applies wavelet transform to the feature space. It can detect arbitrary shape clusters at different scales and has a time complexity of O(n). The algorithm is grid-based and only applicable to low-dimensional data. Input parameters include the number of grid cells for each dimension, the wavelet to use and the number of applications of the wavelet transform. In [11] the density-based algorithm DENCLUE is proposed. This algorithm uses a grid but is very efficient because it only keeps information about grid cells that do actually contain data points and manages these cells in a tree-based access structure. This algorithm generalizes some other clustering approaches which, however, results in a large number of input parameters. Also the density- and grid-based clustering technique CLIQUE [1] has been proposed for mining in high-dimensional data spaces. Input parameters are the size of the grid and a global density threshold for clusters. The major difference to all other clustering approaches is that this method also detects sub- spaces of the highest dimensionality such that high-density clusters exist in those subspaces.

3. ORDERING THE DATABASE WITH RESPECT TO THE CLUSTERING STRUCTURE

3.1 Motivation

An important property of many real-data sets is that their intrinsic cluster structure cannot be characterized by global density parameters. Very different local densities may be needed to re- veal clusters in different regions of the data space. For example, in the data set depicted in Figure 1, it is not possible to detect the clusters A, B, C_1, C_2, and C_3 simultaneously using one global density parameter. A global density-based decomposition would consist only of the clusters A, B, and C, or C_1, C_2, and C_3. In the second case, the objects from A and B are noise. The first alternative to detect and analyze such clustering structures is to use a hierarchical clustering algorithm, for instance the single-link method.



Definition: (Directly density-reachable)

Object *p* is *directly density-reachable* from object *q* wrt. Ɛ

and *MinPts* in a set of objects *D* if

1) *p* Ɛ *N (q)* (*N(q)* is the subset of *D* contained in the Ɛ -neighborhood of *q*.)

2) *Card*(*N (q)*) Ɛ *MinPts* (*Card*(*N*) denotes the cardinality of the set *N*)

The condition *Card*(*N(q)*) *MinPts* is called the “core object condition”. If this condition holds for an object *p*, then we call *p* a “core object”. Only from core objects, other objects can be directly density-reachable.

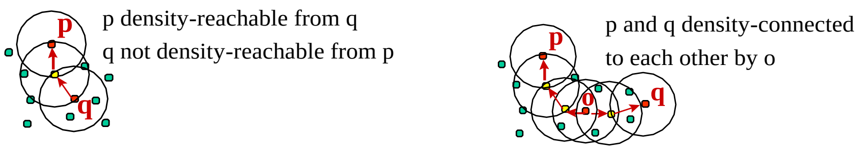

p
q
p density-reachable from q
q not density-reachable from p
p
o
q
p and q density-connected
to each other by o

Figure. Density-reachability and connectivity

Definition: (Density-reachable)

An object p is *density-reachable* from an object q wrt. Ɛ and *MinPts* in the set of objects D if there is a chain of objects p_1, .., p_n, $p_1 = q$, $p_n = p$ such that p_i Ɛ D and p_{i+1} is directly density-reachable from p_i wrt. Ɛ and *MinPts*.

This relation is not symmetric in general. Only core objects can be mutually density-reachable.

3.2 DENSITY-BASED CLUSTERING

The key idea of density-based clustering is that for each object of a cluster the neighborhood of a given radius (Ɛ has to contain at least a minimum number of objects (*MinPts*), i.e. the cardinality of the neighborhood has to exceed a threshold. The formal definitions for this notion of a clustering are shortly called “border objects” of the cluster - are, however, directly density-reachable from at least one core object of the cluster (in contrast to noise objects).

The algorithm DBSCAN [6], which discovers the clusters and the noise in a database according to the above definitions, is based on the fact that a cluster is equivalent to the set of all objects in D which are density-reachable from an arbitrary core object in the cluster (c.f. lemma 1 and 2 in [6]).

The retrieval of density-reachable objects is performed by iteratively collecting *directly* density-reachable objects. DBSCAN checks the Ɛ -neighborhood of each point in the database. If the Ɛ-neighborhood $N(p)$ of a point p has more than *MinPts* points, a new cluster C containing the objects in $N(p)$ is created.

Then, the Ɛ -neighborhood of all points q in C which have not yet been processed is checked. If $N(q)$ contains more than *MinPts* points, the neighbors of q which are not already contained in C are added to the cluster and their Ɛ -neighborhood is checked in the next step. This procedure is repeated until no new point can be added to the current cluster C.

3.2.1 Density-Based Cluster-Ordering

OrderedFile for writing and close this file after ending the loop. Each object from a database SetOfObjects is simply handed over to a procedure ExpandClusterOrder if the object is not yet processed. The pseudo-code for the procedure ExpandClusterOrder is depicted below. The procedure ExpandClusterOrder first retrieves the Ɛ -neighborhood of the object Object passed from the main loop OPTICS, sets its reachability-distance to UNDEFINED and determines its core-distance.

```
OPTICS (SetOfObjects, ε, MinPts, OrderedFile)
   OrderedFile.open();
   FOR i FROM 1 TO SetOfObjects.size
      DO Object := SetOfObjects.get(i);
      IF NOT Object.Processed THEN
         ExpandClusterOrder(SetOfObjects, Object,
         ε,MinPts, OrderedFile) OrderedFile.close();
END; // OPTICS
```

Algorithm OPTICS

Then, Object is writ- ten to OrderedFile. The IF-condition checks the core object property of Object and if it is not a core object at the generating distance Ɛ, the control is simply returned to the main loop OPTICS which selects the next unprocessed object of the data- base. Otherwise, if Object is a core object at a distance Ɛ, we iteratively collect directly density-reachable objects with respect to Ɛ and *MinPts*. Objects which are directly density-reachable from a current core object are inserted into the seed-list OrderSeeds for further expansion. The objects contained in OrderSeeds are sorted by their reachability-distance to the closest core object from which they have been directly density- reachable. In each step of the WHILE-loop, an object currentO- bject having the smallest reachability-distance in the seed-list is selected by the method OrderSeeds:next(). The Ɛ-neighbor- hood of this object and its core-distance are determined. Then, the object is simply written to the file OrderedFile with its core- distance and its current reachability-distance. If currentObject is a core object, further candidates for the expansion may be inserted into the seed-list OrderSeeds.

```
ExpandClusterOrder(SetOfObjects, Object, ε, MinPts,,OrderedFile);
   neighbors := SetOfObjects.neighbors(Object, ε);
   Object.Processed := TRUE; Object.reachability_distance
   := UNDEFINED; Object.setCoreDistance(neighbors, ε,
   MinPts); OrderedFile.write(Object);
   IF Object.core_distance <> UNDEFINED THEN
      OrderSeeds.update(neighbors, Object);
      WHILE NOT OrderSeeds.empty() DO currentObject :=
         OrderSeeds.next();
         neighbors:=SetOfObjects.neighbors(currentObject, ε);
         currentObject.Processed := TRUE;
         currentObject.setCoreDistance(neighbors, ε, MinPts);
         OrderedFile.write(currentObject);
         IF currentObject.core_distance<>UNDEFINED THEN
            OrderSeeds.update(neighbors, currentObject);
END; // ExpandClusterOrder
```

To retrieve the Ɛ -neighborhood of an object *o*, a *region query* with the center *o* and the radius Ɛ is used. Without any index support, to answer such a region query, a scan through the whole database has to be performed. In this case, the run-time of OPTICS would be $O(n^2)$. Having generated the augmented cluster-ordering of a database with respect to Ɛ and *MinPts*, we can extract any density-based clustering from this order with respect to *MinPts* and a clustering-distance. We first check whether the reachability-distance of the current object Object is larger than the clustering-distance Ɛ’. In this case, the object is not density-reachable with respect to Ɛ’ and *MinPts* from any of the objects which are located before the current object in the cluster-ordering. This is obvious, because if Object had been density-reachable with respect to Ɛ’ and *MinPts* from a preceding object in the order, it would have been assigned a reachability-distance of at most Ɛ'. Therefore, if the reachability-distance is larger than Ɛ', we look at the core-distance of Object and start a new cluster if Object is a core object with respect to Ɛ' and *MinPts*.

```
ExtractDBSCAN-Clustering (ClusterOrderedObjs,ε', MinPts)
// Precondition: ε' ≤ generating dist ε for ClusterOrderedObjs
   ClusterId := NOISE;
   FOR i FROM 1 TO ClusterOrderedObjs.size DO Object
      := ClusterOrderedObjs.get(i);
      IF Object.reachability_distance > ε' THEN
         // UNDEFINED > ε
         IF Object.core_distance ≤ ε' THEN
            ClusterId := nextId(ClusterId);
            Object.clusterId := ClusterId;
         ELSE
            Object.clusterId := NOISE;
      ELSE     // Object.reachability_distance ≤ ε'
         Object.clusterId := ClusterId;
END; // ExtractDBSCAN-Clustering
```

Algorithm ExtractDBSCAN-Clustering

NOISE (note that the reachability-distance of the first object in the cluster-ordering is always UNDEFINED and that we as- sume UNDEFINED to be greater than any defined distance). If the reachability-distance of the current object is smaller than Ɛ', we can simply assign this object to the current cluster because then it is density-reachable with respect to Ɛ' and *MinPts* from a preceding core object in the cluster-ordering. The clustering created from a cluster-ordered data set by Extract DBSCAN-Clustering is nearly indistinguishable from a clustering created by DBSCAN. Only some border objects may be missed when extracted by the algorithm ExtractDBSCAN- Clustering if they were processed by the algorithm OPTICS be- fore a core object of the corresponding cluster had been found. However, the fraction of such border objects is so small that we can omit a post processing (i.e. reassign those objects to a cluster) without much loss of information. To extract different density-based clustering from the cluster- ordering of a data set is not the intended application of the OPTICS algorithm. That an extraction is possible only demonstrates that the cluster-ordering of a data set actually contains the information about the intrinsic clustering structure of that data set (up to the generating distance Ɛ). This information can be analyzed much more effectively by using other techniques which are presented in the next section.

4. IDENTIFYING THE CLUSTERING STRUCTURE

The OPTICS algorithm generates the augmented cluster-ordering consisting of the ordering of the points, the reachability values and the core-values. However, for the following interactive and automatic analysis techniques only the ordering and the reachability-values are needed. To simplify the notation, we specify them formally:

REACHABILITY PLOTS AND PARAMETERS

The cluster-ordering of a data set can be represented and under- stood graphically. In principle, one can *see* the clustering structure of a data set if the reachability-distance values ***r*** are plotted for each object in the cluster-ordering ***o***, the reachability-plot for a very simple 2-dimensional data set. Note that the visualization of the cluster-order is independent from the dimension of the data set. For example, if the objects of a high-dimensional data set are distributed similar to the distribu- tion of the 2-dimensional data set (i.e. there are three "Gaussian bumps" in the data set), the "reachability- plot" would also look very similar.

A further advantage of cluster-ordering a data set compared to other clustering methods is that the reachability-plot is rather insensitive to the input parameters of the method, i.e. the generating distance Ɛ and the value for *MinPts*. Roughly speaking, the values have just to be "large" enough to yield a good result. The concrete values are not crucial because there is a broad range of possible values for which we always can see the clustering structure of a data set when looking at the corresponding reachability-plot

The generating distance Ɛ influences the number of clustering-levels which can be seen in the reachability-plot. The smaller we choose the value of Ɛ, the more objects may have an UNDEFINED reachability-distance. Therefore, we may not see clusters of lower density, i.e. clusters where the core objects are core objects only for distances larger than Ɛ.

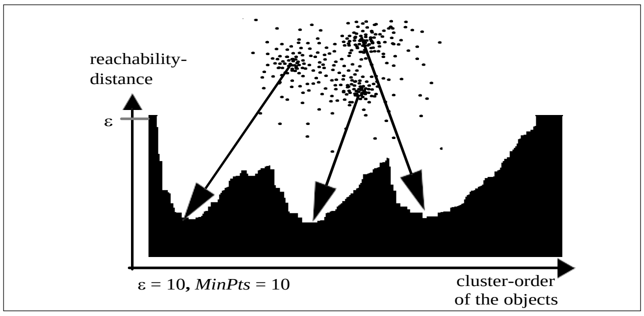
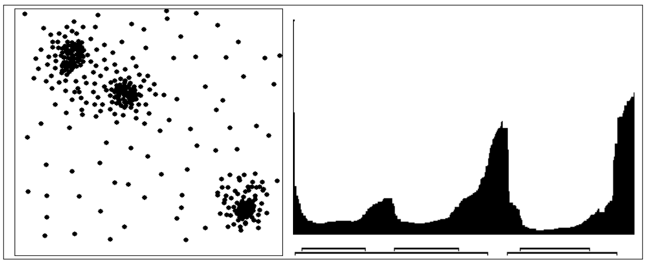


Figure for Illustration of the cluster-ordering

4.1. Experimental Evaluation

The clustering algorithm was implemented in MAT LAB. In section 4.1, we have identified clusters as "dents" in the reachability-plot. Here, we demonstrate that what we call a dent is, in fact, a Ɛ -cluster by showing synthetic, two-dimensional points as well as high-dimensional real-world example.



2-*d* synthetic data set (left), the reachability-plot (right).

In figure, we see an example of three equal size clusters, two of which are very close together, and some noise points. We see that the algorithm successfully identifies this hierarchical structure, i.e. it finds the two clusters and the higher-level cluster containing both of them. It also finds the third cluster, and even identifies an especially dense region within it, to extract the hierarchical cluster structure from the augmented cluster-ordering generated by OPTICS, both visually and automatically. The algorithm for the automatic extraction is highly efficient and of a very high quality. Once we have the set of points belonging to a cluster, we can easily compute traditional clustering information like representative points or shape descriptions.

5. Conclusions

In this paper, we proposed a cluster analysis method based on the OPTICS algorithm. OPTICS computes an *augmented cluster-ordering* of the database objects. The main advantage of our approach, when compared to the clustering algorithms pro- posed in the literature, is that we do not limit ourselves to one global parameter setting. Instead, the augmented cluster-ordering contains information which is equivalent to the density- based clustering corresponding to a broad range of parameter settings and thus is a versatile basis for both automatic and interactive cluster analysis. We demonstrated how to use it as a stand-alone tool to get in- sight into the distribution of a data set. Depending on the size of the database, we either represent the cluster-ordering graphically (for small data sets) or use an appropriate visualization technique (for large data sets). Both techniques are suitable for interactively exploring the clustering structure, offering additional insights into the distribution and correlation of the data. We also presented an efficient and effective algorithm to automatically extract not only 'traditional' clustering information but also the intrinsic, hierarchical clustering structure. There are several opportunities for *future research*. For very high-dimensional spaces, no index structures exist to efficiently support hyper sphere range queries needed by the OPTICS algorithm. Therefore it is infeasible to apply it in its current form to a database containing several million high-dimensional objects. Consequently, the most interesting question is whether we can modify OPTICS so that we can trade-off a limited amount of accuracy for a large gain in efficiency. Incrementally managing a cluster-ordering when updates on the database occur is another interesting challenge. Although there are techniques to update a 'flat' density-based decomposition [7] incrementally, it is not obvious how to extend these ideas to a density-based cluster-ordering of a data set.

REFERENCES

[1] Agrawal R., Gehrke J., Gunopulos D., Raghavan P.: "Automatic Subspace Clustering of High Dimensional Data for Data Mining Applications", Proc. ACM SIGMOD'98 Int. Conf. on Management of Data, Seattle, WA, 1998, pp. 94- 105.

[2] Ankerst M., Keim D. A., Kriegel H.-P.: "'Circle Seg- ments': A Technique for Visually Exploring Large Multidi- mensional Data Sets", Proc. Visualization'96, Hot Topic Session, San Francisco, CA, 1996.

[3] Berchthold S., Keim D., Kriegel H.-P.: "The X-Tree: An Index Structure for High-Dimensional Data", 22nd Conf. on Very Large Data Bases, Bombay, India, 1996, pp. 28-39.

[4] Beckmann N., Kriegel H.-P., Schneider R., Seeger B.: "The R*-tree: An Efficient and Robust Access Method for Points and Rectangles", Proc. ACM SIGMOD Int. Conf. on Management of Data, Atlantic City, NJ, ACM Press, New York, 1990, pp. 322-331.

[5] Ciaccia P., Patella M., Zezula P.: "M-tree: An Efficient Access Method for Similarity Search in Metric Spaces", Proc. 23rd Int. Conf. on Very Large Data Bases, Athens, Greece, 1997, pp. 426-435.

[6] Ester M., Kriegel H.-P., Sander J., Xu X.: "A Density- Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise", Proc. 2nd Int. Conf. on Knowledge Discovery and Data Mining, Portland, OR, AAAI Press, 1996, pp. 226-231.

[7] Ester M., Kriegel H.-P., Sander J., Wimmer M., Xu X.: "Incremental Clustering for Mining in a Data Warehousing Environment", Proc. 24th Int. Conf. on Very Large Data Bases, New York, NY, 1998, pp. 323-333.

[8] Ester M., Kriegel H.-P., Xu X.: "Knowledge Discovery in Large Spatial Databases: Focusing Techniques for Effi- cient Class Identification", Proc. 4th Int. Symp. on Large Spatial Databases, Portland, ME, 1995, in: Lecture Notes in Computer Science, Vol. 951, Springer, 1995, pp. 67-82.

[9] Grossman A., Morlet J.: "Decomposition of functions into wavelets of constant shapes and related transforms". Mathematics and Physics: Lectures on Recent Results, World Scientific, Singapore, 1985.

[10] Guha S., Rastogi R., Shim K.: "CURE: An Efficient Clustering Algorithms for Large Databases", Proc. ACM SIGMOD Int. Conf. on Management of Data, Seattle, WA, 1998, pp. 73-84.

[11] Hinneburg A., Keim D.: "An Efficient Approach to Clus- tering in Large Multimedia Databases with Noise", Proc. 4th Int. Conf. on Knowledge Discovery & Data Mining, New York City, NY, 1998.

[12] Hattori K., Torii Y.: "Effective algorithms for the nearest neighbor method in the clustering problem", Pattern Recog- nition, 1993, Vol. 26, No. 5, pp. 741-746.

[13] Huang Z.: "A Fast Clustering Algorithm to Cluster Very Large Categorical Data Sets in Data Mining", Proc. SIG- MOD Workshop on Research Issues on Data Mining and Knowledge Discovery, Tech. Report 97-07, UBC, Dept. of CS, 1997.

[14] Jain A. K., Dubes R. C.: "Algorithms for Clustering Da- ta," Prentice-Hall, Inc., 1988.

[15] Keim D. A.: "Pixel-oriented Database Visualizations", in: SIGMOD RECORD, Special Issue on Information Visu- alization, Dezember 1996.

[16] Keim D. A.: "Databases and Visualization", Proc. Tuto- rial ACM SIGMOD Int. Conf. on Management of Data, Montreal, Canada, 1996, p. 543.

[17] Knorr E. M., Ng R.T.: "Finding Aggregate Proximity Re- lationships and Commonalities in Spatial Data Mining," IEEE Trans. on Knowledge and Data Engineering, Vol. 8, No. 6, December 1996, pp. 884-897.

[18] Kaufman L., Rousseeuw P. J.: "Finding Groups in Data: An Introduction to Cluster Analysis", John Wiley & Sons, 1990.

[19] MacQueen, J.: "Some Methods for Classification and Analysis of Multivariate Observations", 5th Berkeley Symp. Math. Statist. Prob., Vol. 1, pp. 281-297.

AUTHORS PROFILE

R.NANDHAKUMAR	Dr. ANTONY SELVADOSS THANAMANI
 R.Nandhakumar, Assistant Professor in Computer Science, Nallamuthu Gounder Mahalingam College, undergoing his Ph.D in Data Mining. Published various papers under reputed journals. He is the coordinator in various activates in college like NAAC, ISO, etc.	 ***Dr. Antony Selvadoss Thanamani,*** Associate Professor and Head, Research Department of Computer Science, He is Research supervisor for Ph.D. degree in Computer Science in the Bharathiar University, Dravidian University, etc. He established Common Research centre, E-content studio, ISBN Nodal Agency at NGM College, Pollachi. He is Advisory Committee Member in National Conference on Advanced Computing. Editor in International Journal of Advanced Scientific Research, India. Editorial Board Member in International Journal of Advanced Research in Computer and Communication Engineering, India.