A Survey on Clustering Algorithms for Data in Spatial Database Management Systems

Dr.Chandra.E
Director
Department of Computer Science
DJ Academy for Managerial Excellence
Coimbatore, India

Anuradha.V.P
Research Scholar
Karpagam University
Coimbatore, India

ABSTRACT
Clustering is an essential task in data mining to group data into meaningful subsets to retrieve information from a given dataset of Spatial Data Base Management System (SDBMS). The information thus retrieved from the SDBMS helps to detect urban activity centers for consumer applications. Clustering algorithms group the data objects into clusters wherein the objects within a cluster are more similar to each other and are more dissimilar to objects in other clusters. Query processing is a data mining operation of SDBMS to retrieve the required information for consumer applications. There are several basic algorithms as well as advanced algorithms for clustering spatial data. The k-means algorithm is one of the basic clustering method in which an objective function has to be optimized. Extensions of k-means method are implemented for processing large datasets of a database. The clustering algorithms for grouping data in an SDBMS are based on such methods as partitioning methods, hierarchical clustering, and density based clustering. Hypergraphs and Delaunay triangulations are the enhanced features utilized in a spatial clustering algorithm. Each one of the clustering algorithm has advantages and limitations for processing multidimensional data and hence in spatial clustering process. This work makes an attempt at studying the feasibility of the algorithms for implementation in an SDBMS. Performance of the algorithms is studied with respect to various parameters.

General Terms
Your general terms must be any term which can be used for general classification of the submitted material such as Pattern Recognition, Security, Algorithms et. al.

Keywords
Spatial data mining, Clustering algorithms, Spatial data, Spatial clustering

1. INTRODUCTION
Spatial data mining is the discovery of interesting relationships and characteristics that may exist implicitly in spatial databases. Spatial data mining aims to automate such a knowledge discovery process [7]. It plays an important role in a) extracting interesting spatial patterns and features; b) capturing intrinsic relationships between spatial and non-spatial data; c) presenting data regularity concisely and at conceptual levels; and d) helping to reorganize spatial databases to accommodate data semantics, as well as to achieve better performance.

Spatial clustering is a major component of spatial data mining and is implemented as such to retrieve a pattern from the data objects distribution in a given data set. The clusters thus obtained would have to be interpreted to determine each one's significance in the context for which the clustering is carried out. The cluster analysis should also check the quality of the spatial clusters.

There are four methodologies that can be implemented for spatial clustering process. The Partitioning methods, Hierarchical methods, Density-based methods and Grid-based methods are the methodologies that are widely implemented for spatial clustering in SDBMS. In this work only Partitioning methods, Hierarchical methods, Density-based methods are considered for study.

Spatial clustering by itself is quite significant in that it is being implemented in a wide range of applications. Identifying urban activity centers, crime clusters, disease clusters and weather patterns are some of the applications of spatial clustering. Spatial clustering provides an insight into the distribution of data objects in the study region. Thus such a clustering process is a significant step towards the decision making process in such application areas as public safety measures, consumer related applications, ecological problems, public health measures and effective transportation facilities.

The spatial clustering algorithms under the three categories taken for study have their inherent merits and demerits. While the basic k-means algorithm produces only spherical shaped clusters, some of the other algorithms like OPTICS can provide elliptical shaped clusters. Though the criteria for deciding upon a particular algorithm depend on the specific application, some factors remain the same irrespective of the application. The general factors are stated as,

1) Data set size

2) Data dimensionality

3) Time complexity

The spatial clustering algorithm that is chosen should satisfy the requirements of the application for which the study is undertaken. Also the algorithm should be effective in processing data with noise and outliers in data as they are inherently present in geographical datasets.

Scope: The aim of the work is make a review study of the existing clustering algorithms that can process multidimensional data, i.e. spatial data. The feasibility of the algorithms to handle data with noise and outliers has been studied so as to confirm to the requirements of the specific application. Data set size, Data

dimensionality, Time complexity are the general factors with which the comparative study of the algorithms is performed.

Outline: Section 2 is a study of the Partitioning based clustering algorithms. Section 3 outlines the working of Hierarchical based clustering algorithms. Section 4 presents Density based clustering algorithms. Hypergraph based and clustering algorithms based on Similarity criterion are alsodealt in detail in Section 4. Section 5 presents a comparative analysis and discussion of the algorithms in brief. Section 7 presents conclusions of the work.

2. PARTITIONING ALGORITHMS

2.1 The K-means Method

The most commonly used heuristic for solving the k-means problem is based on a general iterative scheme for finding local optimum minimal solution. There are several different versions for k-means algorithm. One of the variants is the Lloyd's algorithm [14], which uses scalar data. It uses the principle that the optimal placement of a center is at the centroid of the associate cluster. Given any set of k centers Z, for each center $z \in Z$, let V (z) denote its neighbourhood, that is, the set of data points for which z is the nearest neighbour. At each level in the processing of the algorithm every center point z is moved to the centroid of V(z) and then updating of V(z) is done by re-computing the distance from each point to its nearest center. The process is repeated until the criterion function does not change after an iteration. As the Lloyd's algorithm was designed for scalar data, it can be implemented as a preprocessing technique while processing multidimensional data.

The k-means algorithm [13] identifies an initial set of cluster centers from the mean values of the objects in each cluster. By allocating each data object in a cluster to its nearest mean center, a new set of clusters are identified. The procedure is executed till the value of the objective function remains the same for successive iterations. A squared-error function that is utilized as the Objective Function in the k-means algorithm is stated as,

$$E = \sum_{i=1}^{k} \sum_{x \epsilon C_i} |x - m_i|^2 \qquad (2.1)$$

where x is the point in space representing the given object, and m_i is the mean of cluster C_i.

For datasets that are larger in size the continuous *k*-means algorithm [6] can be implemented since it is faster than the standard version. As a variation to the standard k-means algorithm, the continuous algorithm reference points are chosen as a random sample from the whole population of data points. In case of the sample being large, the distribution of these initial reference points reflects the distribution of points in the entire set.

A scalable framework [3] is possible for clustering data in large databases. The scalable framework extends the basic K-means algorithm for the clustering process. The scalable framework for clustering stores selectively important portions of the database and summarizes other portions. The size of an allowable pre-specified memory buffer determines the amount of summarizing and required internal bookkeeping. It works on the assumption that an interface to the database permits the algorithm to load a required number of data points. The means such as sequential scan, random sampling by which the data points are obtained is provided by the database engine.

The computational complexity of k-means algorithm in general is *O(n)*, where *n* is the total number of objects. The k-means algorithm is sensitive to noise and outlier data points since a small number of such data can substantially influence the mean value.

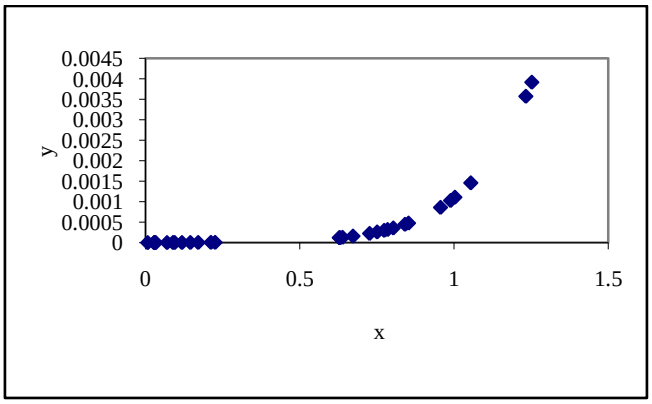

Figure.1 Distribution of Data Points

The standard k-means algorithm alternates between assigning the data-points to their closest centroid (the E-step) and moving each centroid to the mean of its assigned data-points (the M-step) and is binding to a local minimum. Quality of the results depends upon the initial centroids [16]. For better clustering the initial centroids have to be consistent with the distribution of data points. Figure. 1 illustrates the distribution of data points for a sample dataset (e.g. crime rate in a region) of a geographical database. Also such initial centroids can be obtained if certain centroids can be identified in which each centroid represents a group of similar objects. This distribution data points leads to the identification of cluster which when overlaid with thematic maps helps the law enforcement agencies to plan for crime prevention techniques.

A recent adaptation of k-means algorithm treats the clustering problem as an optimization problem [11]. The work is based on improved simulated annealing algorithm and is applied after the initial k clusters are formed. As the clustering problem is considered as the optimization problem, the problem is defined as a function that minimizes the sum of distances between each sample and the nearest cluster center. The function is stated as follows:

$$D = \sum_{i=1}^{k} \sum_{x \epsilon C_i} dist(C_i - x)^2 \qquad (2.2)$$

where, C_i is the i^{th} cluster, c_i is the i^{th} cluster center, x is the sample in C_i, dist represents the Euclidean distance between two data points. The algorithm implements a new data structure, sequence list to solve the clustering problem.

3. HIERARCHICAL ALGORITHMS

3.1 CURE

CURE is a hierarchical clustering algorithm, that employs the features of both the centroid based algorithms and the all point algorithms [8]. Basically CURE obtains a data sample from the given database. The algorithm CURE divides the data sample into groups and identifies some representative points from each group of the data sample. In the first phase, the algorithm considers a set of widely spaced points from the given datasets. In the next phase of the algorithm the selected dispersed points are moved towards the centre of the cluster by a specified value of a factor α. As a result of this process, some randomly shaped clusters are obtained from the datasets. In the process it identifies and eliminates outliers. In the next phase of the algorithm, the representative points of the clusters are checked for proximity with a threshold value and the clusters that are next to each other are grouped together to form the next set of clusters. In this hierarchical algorithm, the value of the factor α may vary between 0 and 1. The utilization of the shrinking factor α by CURE overcomes the limitations of the centroid-based and all-points approaches. As the representative points are moved through the clustering space, the ill effects of outliers are reduced by a greater extent. Thus the feasibility of CURE is enhanced by the shrinking factor α. The worst case time complexity of CURE is determined to be $O(n^2 \log n)$.

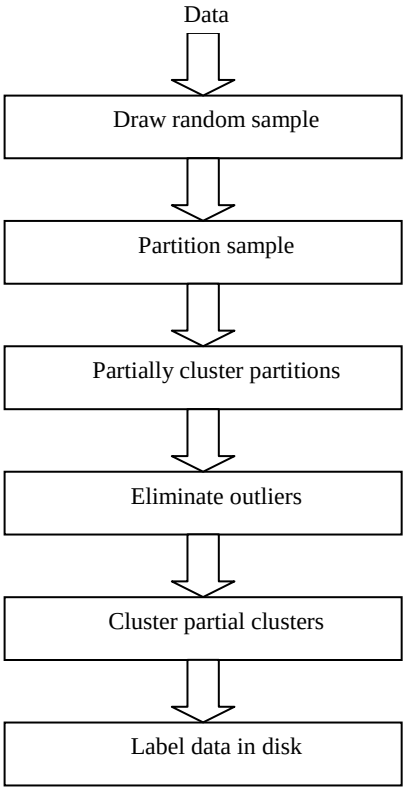

Figure. 2 Overview of CURE implementation

Figure 2 shows the overview of CURE implementation. A random sample of data objects is drawn from the given datasets. Partial clusters are obtained by partitioning the sample dataset and outliers are identified and removed in this stage. Final refined clusters are formed from the partial cluster set.

3.2 BIRCH

The clustering algorithm BIRCH is a main memory based algorithm, i.e., the clustering process is carried out with a memory constraint. BIRCH's incremental clustering is based on the concept of clustering feature and CF tree [17]. A clustering feature is a triple that contains the summary of the information of each cluster. Given N d-dimensional points or objects in a cluster: $\{\overline{X_i}\}$ where i=1,2, …, N, the Clustering feature (CF) as a vector of the cluster can be stated as,

$$CF = (N, \overline{LS}, SS) \qquad (3.1)$$

where N is the number of points in the cluster, $\overline{LS}$ is the linear sum on N points, i.e., $\sum_{i=1}^{N} \overline{X_i}$, and SS is the square sum of the data points. i.e.. $\sum_{i=1}^{N} \overline{X_i^2}$

A clustering feature tree (CF tree) contains the CFs that holds the summary of clusters. A CF tree is a height balanced tree that has two parameters namely, a branching factor, B, and threshold, T. The representation of a non-leaf node can be stated as$\{CF_{i,} child_i\}$:, where,

i = 1,2, …., B,
$child_i$- A pointer to its i[th] child node
$CF_{i,}$ - CF of the subcluster represented by the i[th] child

The non-leaf node is provides a representation for a cluster and the contents of the node represents all of the subclusters. In the same manner a leaf-node's contents represents all of its subclusters and has to confirm to a threshold value for T.

The BIRCH clustering algorithm is implemented in 4 phases. In phase1, the initial CF is built from the database based on the branching factor B and the threshold value T. Phase2 is an optional phase in which the initial CF tree would be reduced in size to obtain a smaller CF tree. Global clustering of the data points is performed in phase3 from either the initial CF tree or the smaller tree of phase2. As has been shown in the evaluation good clusters can be obtained from phase3 of the algorithm. If it is required to improve the quality of the clusters, phase4 of the algorithm would be needed in the clustering process.

The execution of Phase1 of BIRCH begins with a threshold value T. The procedure reads the entire set of data points in this phase, selects the data points based on a distance function. The selected data points are stored in the nodes of the CF tree. The data points that are closely spaced are considered to be clusters and are thus selected. The data points that are widely placed are considered to be outliers and thus are discarded from clustering. In this clustering process, if the threshold limit is exceeded before the complete scan of the database, the value is increased and a much smaller tree with all the chosen data points is built. An optimum value for threshold T is necessary in order to get

good quality clusters from the algorithm. If it is required to fine tune the quality of the clusters, further scans of the database is recommended through phase4 of the algorithm. The worst case time complexity of the algorithm is *O(n)*. The time needed for the execution of the algorithm varies linearly to the dataset size.

3.3. CHAMELEON

In agglomerative hierarchical approaches, the major disadvantage is that they are based on a static, user-specified inter-connectivity model, which either under estimates or over estimates the inter-connectivity of objects and clusters. This limitation is overcome by the algorithm CHAMELEON.

CHAMELEON makes use of a sparse graph, where the nodes represent data objects; weights in the edges represent similarities between the data objects [12]. CHAMELEON's sparse graph implementation lets it to scale to large databases in an effective manner. This implementation of sparse graph is based on the frequently used k-nearest neighbour graph representation.

CHAMELEON identifies the similarity between a pair of clusters namely, C_i and C_j by evaluating their relative inter-connectivity RI (C_i, C_j) and relative closeness RC (C_i, C_j). When the values of both RI (C_i, C_j) and RC (C_i, C_j) are high for any two clusters, CHAMELEON's agglomerative algorithm merges those two clusters.

The relative inter-connectivity between a pair of clusters C_i and C_j can be stated as:

$$RI(C_i, C_j) = \frac{\left|EC_{\{C_i,C_j\}}\right|}{\frac{\left|EC_{C_i}\right|+\left|EC_{C_j}\right|}{2}} \quad (3.2)$$

where,

$\left|EC_{\{C_i,C_j\}}\right|$- edge-cut of cluster containing both C_i and C_j

$\left|EC_{C_i}\right|$ - min-cut bisector indicating internal inter-connectivity of cluster C_i

$\left|EC_{C_j}\right|$ - min-cut bisector indicating internal inter-connectivity of cluster C_j

The relative closeness of two clusters C_i and C_j is stated as follows:

$$RC(C_i, C_j) = \frac{\bar{S}_{EC_{\{C_i,C_j\}}}}{\frac{|C_i|}{|C_i|+|C_j|}\bar{S}_{EC_{C_i}}+\frac{|C_j|}{|C_i|+|C_j|}\bar{S}_{EC_{C_j}}} \quad (3.3)$$

where,

$\bar{S}_{EC_{C_i}}$ - average edge weight of min-cut bisector of cluster C_i

$\bar{S}_{EC_{C_j}}$ - average edge weight of min-cut bisector of cluster C_j

$\bar{S}_{EC_{\{C_i,C_j\}}}$- average edge weight edges connecting vertices of cluster C_i with that of cluster C_j

CHAMELEON agglomerative hierarchical approach implements the algorithm in two separate phases. In the first phase, dynamic modeling of the data objects is done by clustering these objects into subclusters. In the second phase, a dynamic modeling framework is employed on the data objects to merge the subclusters in a hierarchical manner to get good quality cluster. The dynamic framework model can be implemented by two different methods. In the first method it is checked that the values of relative inter-connectivity and relative closeness between a pair of cluster cross a user-specified threshold value. For this purpose, these two parameters should satisfy the following conditions:

$$1) RI(C_i, C_j) \geq T_{RI}$$
$$2) RC(C_i, C_j) \geq T_{RC}$$

In the second method, CHAMELEON chooses a pair of clusters that maximizes a function that is given by,

$$RI(C_i, C_j) * RC(C_i, C_j)^{\alpha} \quad (3.4)$$

where α is user-specified parameter that takes the values between 0 and 1. The time complexity of CHAMELEON's two-phase algorithm is *O (nm+nlogn+m^2logm)*.

4. DENSITY BASED ALGORITHMS

4.1 DBSCAN: Density-Based Spatial Clustering Of Applications With Noise

DBSCAN is a density-based clustering algorithm [5] and an R*-Tree is implemented for the process [2]. The basic concept of this algorithm is that, in a given cluster within the specified radius of the neighbourhood of every point in a cluster, there must exist a minimum number of points. The density attributed to the points in the neighbourhood of a point in a cluster has to cross beyond a threshold value. Based on a distance function the shape of the neighbourhood is obtained and is expressed as dist(p, q) between points p and q.

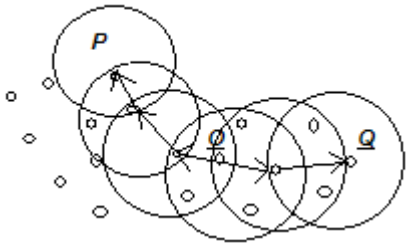

Figure.3 p and q are density-connected, connected by o

The DBSCAN algorithm identifies the clusters of data objects based on the density-reachability and density-connectivity of the core and border points present in a cluster. The primary operation of the algorithm can be stated as:

Given the parameters *Eps* and *MinPts*, a cluster can be identified by a two phase method. It is stated as, 1) select an arbitrary data point from the database that satisfies the core point condition as a seed; 2) fetch all the data points that are density-reachable from the seed point forming a cluster including the seed. Density-connectivity is depicted in Figure.3.

The algorithm requires the user to know the parameters *Eps* and *MinPts* for each cluster at least one point from the corresponding cluster. Since this is not feasible for every cluster, the algorithm uses global values for these two parameters. DBSCAN begins the clustering with an arbitrary data point p and retrieves the data points that are density-reachable from *p* with respect to *Eps* and *MinPts*. This approach leads to the following inferences, 1) if p is a core point this method results in a cluster that is relevant to *Eps* and *MinPts*, 2) if *p* is a border point, no points are density-reachable from p and the algorithm scans the next data point in the database.

4.2 OPTICS

OPTICS is a clustering algorithm that identifies the implicit clustering in a given dataset and is a density-based clustering approach [1]. Unlike the other density-based clustering algorithm DBSCAN which depends on a global parameter setting for cluster identification, OPTICS utilizes a multiple number of parameter settings. In that context the OPTICS is an extended work of DBSCAN algorithm. DBSCAN algorithm requires two parameters namely, ε the radius of the neighbourhood from a given representative data object and *MinPts* the threshold value for the occurrence of the number of data objects in the given neighbourhood.

OPTICS is implemented on the concept of Density-based Cluster Ordering which is an extension of DBSCAN algorithm. Density-based Cluster Ordering works on the principle that sparsely populated cluster for a higher ε value contains highly populated clusters for a lower value of ε. Multiple number of distance parameter ε have been utilized to process the data objects. OPTICS ensures good quality clustering by maintaining the order in which the data objects are processed, i.e., high density clusters are given priority over lower density clusters.

The cluster information in memory consists of two values for every processed object, i.e. the core-distance and reachability-distance.

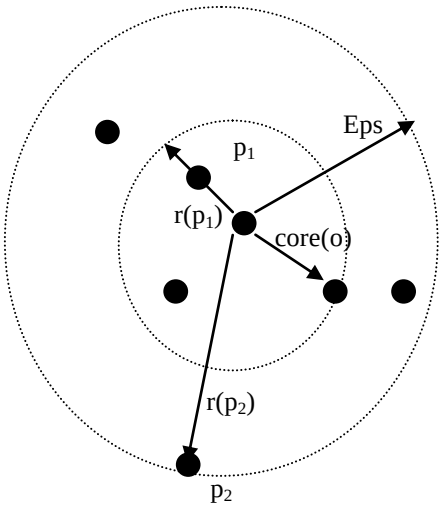

Figure 4. Core distance and reachability distances

Figure. 4 illustrates the concept of core distance and reachability distances. The reachability distances are $r(p_1)$ and $r(p_2)$ for data objects p_1 and p_2 respectively. The reachability distances are evaluated with respect to the *Eps* neighbouhood. The core-distance of a data object p is the shortest distance *Eps'* between p and a data object in its *Eps*-neighbourhood and so *p* is a core object with respect to *Eps'* if this neighbour is contained in $N_{Eps}(p)$. Also the reachability-distance of a data object *p* with respect to another data object o is the shortest distance such that *p* is directly density-reachable from *o* if *o* is a core object. Thus OPTICS produces an ordering of the given database. Along with ordering OPTICS also stores core-distance and reachability-distance of each data object, thereby resulting in better quality clusters.

The OPTICS clustering algorithm provides an efficient cluster-ordering with a set of ordering of the data objects with reachability-values and core-values. OPTICS implements pixel-oriented visualization techniques for large multidimensional data sets. Additionally, OPTICS utilizes automatic techniques to identify start and end of cluster structures to begin with and later groups them together to determine a set of nested clusters.

4.3 Hypergraph based clustering algorithms

The hypergraph [10] based clustering algorithm is a density-based algorithm and is executed in two phases. In the first phase of the procedure a Voronoi diagram is constructed from the set of spatial data objects S_D. A Delaunay triangle is built from the Voronoi diagram. Typically, a Delaunay triangulation [9][15] consists of three points that represents three data objects. If the combined length of a triangle is much greater than that of other triangles then there is low proximity among the three objects and thus it is eliminated. A hypergraph is constructed from the Delaunay triangulation and the edges of the hypergraph are derived from the sides of each one of the Delaunay triangle. Basically, a hypergraph represents the proximity and similarity among neighbouring data objects. A hypergraph is one that has evolved from the regular graph and is stated as such, i.e., for a hyper graph G (V,E), the set of vertices V represents data objects and the set of edges E represents hyperedges. A hyperedge would be a representation of the proximity and similarity of data objects of the data set. In the hypergraph any one vertex may have more than one edge connected to it depending upon the proximity value it has with other data objects in the cluster. The edge weights of the hypergraph state the degree of similarity among represented objects.

The edge weight in a hypergraph is determined from the total length of edges of its corresponding triangle. It is computed from the function,

$$W(e_H) = \frac{\sum_{e \epsilon T_H} length(e)}{\sum_{e \epsilon T_H} (length(e))^2} \quad (4.1)$$

From the function it should be noted that the total length of a triangle T_H should be small for a large edge weight in the corresponding hypergraph.

The clustering algorithm is implemented in two phases, once the hypergraph design is completed. The first phase, the Seeds phase in which a set of initial clusters are determined and are designated as seeds for the next phase, Hierarchical_clustering phase of the algorithm. In the second phase the seeds with high proximity values are grouped into clusters to get the next set of clusters. The procedure is repeated till a set of clusters with high intracluster similarity is derived. The computational complexity is *O(n).*

4.4 Clustering algorithm based on Similarity criterion

An SDBMS contains spatial objects that are of geometric in shape, i.e. lines and polygons [4]. A clustering algorithm is one that identifies similar objects and groups them together in to a cluster. Such a clustering procedure can be based on the geometric shape of the spatial objects under study. The algorithm utilizes the distance metric Euclidean distance as similarity criterion and clusters objects accordingly. The Euclidean distance for a line segment $\overline{XY}$ that rotates through an angle θ and translating through T about some origin is stated as,

$$d = \sqrt{(x'_{i1} - y'_{j1})^2 + (x'_{i2} - y'_{j2})^2} \tag{4.2}$$

where,

$$(x'_{i1} - y'_{j1})^2 + (x'_{i2} - y'_{j2})^2 = (x_{i1} - y_{j1})^2 + (x_{i2} - y_{j2})^2 \tag{4.3}$$

and,

$$x'_{i1} = x_{i1}cos\theta - x_{i2}sin\theta,\ x'_{i2} = x_{i1}sin\theta + x_{i2}cos\theta$$
$$y'_{j1} = y_{j1}cos\theta - y_{j2}sin\theta,\ y'_{j2} = y_{j1}sin\theta + y_{j2}cos\theta$$

And,

$$x_i = (x_{i1}\ x_{i2})^T \quad \text{and} \quad y_i = (y_{i1}\ y_{i2})^T$$

The basic procedure initially finds a core object, an object with at least two objects similar to it in the given dataset of spatial objects. The procedure then identifies other objects similar to the core object to group them into clusters. The spatial object as such is considered for checking similarities with that of the core object. It is essential that the objects need to be rotation invariant and transition invariant to be grouped into a cluster. The algorithm has a time complexity of *O* (n^2) and *O (n).*

5. COMPARATIVE ANALYSIS AND DISCUSSION

In this work on the study of spatial clustering algorithms basic clustering algorithms are also studied along with clustering algorithms for multidimensional data. Such a study would identify urban activity centers, which would be a key factor in the decision making process in consumer based applications. The k-means algorithm is a partitioning based clustering algorithm. Though there are several versions of k-means algorithm the Lloyd's algorithm is considered as the standard version of the algorithm. The basic k-means clustering algorithm is designed for clustering only scalar data. Hence it can be implemented as a preprocessing technique for clustering data of an SDBMS. It requires optimization of an Objective function to determine the clustering solution. As the method is versatile various adaptations have been presented over a period of time. A continuous k-means method can be adapted for large dataset sizes. The standard k-means procedure is binding to a local minimum and thus depends upon the initial centroids which are dependent on the distribution of data points. However, a scalable frame work of the k-means method is available for processing multidimensional data.

Clustering algorithms have evolved into hierarchical algorithms from partitioning based algorithms. CURE is one such hierarchical algorithm that depends upon a parameter, the shrinking factor α. The shrinking factor should have an optimum value for effective clustering of the data objects. Further, the shrinking factor reduces the ill effects of outliers considerably.

The hierarchical clustering algorithm BIRCH clusters the data objects by building CF trees for the data sets. The algorithm depends on a threshold value T with which the clustering is executed. Global as well as local clustering is carried out by BIRCH and is main memory based algorithm. Thus it has inherent memory constraint and is effective in handling outliers with an optimum threshold value. It has a linear processing time for clustering data objects.

CHAMELEON is an agglomerative hierarchical algorithm. The algorithm with inter-connectivity and relative closeness between clusters as parameters, which are checked against an optimum threshold value, produces good quality clusters.

DBSCAN is density-based clustering algorithm that effectively clusters multidimensional data with noise. It utilizes two global parameters *Eps* and *MinPts* for grouping data clusters. Density-reachability and density-connectivity of the data objects with respect to a threshold value are the basis for the clustering operation. A distance function is executed to determine neighbourhoods in a cluster. The limitation of the procedure is that the user should have knowledge of the two global parameters.

OPTICS is also a density-based clustering algorithm similar to DBSCAN. Unlike DBSCAN OPTICS utilizes multiple number of distance parameter ε to process and group data objects into clusters. With respect to a core object OPTICS produces an effective cluster ordering for a set of ordered objects with reachability values and core values. OPTICS processes multidimensional data with a pixel-oriented visualization technique and determines a set of nested clusters with an automatic technique.

The Hypergraph based clustering algorithm is a density related algorithm. The hypergraph design and the clustering of data object are executed in two phases. The hypergraph itself is constructed from a Delaunay triangle and its hyperedge represents the degree of similarity among neighbouring objects. The edge weight of the hyperedge is determined from a weights function. The clustering algorithm is executed in two phases to derive a set of clusters with high intracluster similarity value.

A more recent work groups spatial objects based on a similarity criterion. The spatial objects should satisfy the distance metric Euclidean distance to be grouped into a cluster. Additionally, it is required that the spatial objects should be rotation and transition invariant in nature.

Table I. illustrates the comparative study taken up in this work. Type of clustering, Dimensionality of data, Parameters used, Shape of clusters and Worst case time are factors considered for drawing the table of comparative study.

TABLE 1. SUMMARY OF COMPARATIVE STUDY

Algorithm	Type of Clustering	Dimensionality of Data	Parameters Used	Shape of the clusters	Worst Case Time
K-means method (Lloyd's Algorithm)	Local Clustering	Scalar data	m_i - mean of cluster C_i	Spherical	*O(n)*
K-means method (Scalable fame work)	Local Clustering	Multidimensional data, small datasets	Parameters of the individual cluster C_i	Spherical	*O(n)*
Continuous K-means	Local Clustering	Multidimensional data, small datasets	m_i - mean of cluster C_i	Spherical	*O(n)*
CURE	Global Clustering and Local Clustering	Multidimensional data, large datasets	shrinking factor α	Non-spherical	$O(n^2 \log n)$
BIRCH	Global Clustering and Local Clustering	Multidimensional data, large datasets	Branching Factor B, Threshold value T	Elliptical	*O(n)*
CHAMELEON	Global Clustering and Local Clustering	Multidimensional data, large datasets	relative inter-connectivity RI (C_i, C_j) , relative closeness RC (C_i, C_j)	Arbitrary shapes	*O (nm+nlogn+ m^2logm)*
DBSCAN	Global Clustering and Local Clustering	Multidimensional data, large datasets	Eps and MinPts	Arbitrary shapes	*O(n)*
OPTICS	Global Clustering and Local Clustering	Multidimensional data, large datasets	Multiple number of distance parameter ε	Arbitrary shapes	*O(n)*
Hypergraph based algorithm	Local Clustering	Multidimensional data, large datasets	Edge Weight, $W(e_H) = \frac{\sum_{e \epsilon T_H} length(e)}{\sum_{e \epsilon T_H} (length(e))^2}$	Arbitrary shapes	*O(n)*
Similarity Criterion	Local Clustering	Multidimensional data, large datasets	Euclidean Distance, $d = \sqrt{(x_{i1}^{'} - y_{j1}^{'})^2 + (x_{i2}^{'} - y_{j2}^{'})^2}$	Arbitrary shapes	*$O(n^2)$, O(n)*

6. CONCLUSION

Clustering is a significant data mining function as it plays a key role in information retrieval in database operations. The SDBMS is characterized with some unique features and so has specific requirements in a clustering algorithm. Specifically the geographic databases in the SDBMS have noisy data and outliers, which the algorithm should handle effectively. Partitioning methods, Hierarchical methods and Density-based methods are the methodologies studied in this paper. The basic K-means, K-means with scalable framework and Continuous K-means are the Partitioning methods. CURE, BIRCH, and CHAMELEON are the Hierarchical clustering algorithms. DBSCAN and OPTICS are density based clustering algorithms. A Hypergraph based clustering algorithm is also studied that is an enhancement to the basic Density-based algorithm and the hypergraph itself is derived from the Delaunay triangulation of the spatial datasets. A recent clustering algorithm groups datasets into clusters with the spatial object as such as the core object and it utilizes Euclidean distance for that purpose. The algorithms are studied with respect to key factors such as dimensionality of data, worst case time, etc. The algorithm that is efficient in processing geographic databases of the application would be implemented in the future work. The feasibility of the algorithm would be verified with synthetic and real datasets in the future work.

7. REFERENCES

[1] Ankerst, M., Breunig, M., Kreigel, H.-P. and Sander, J. 1999. OPTICS: Ordering Points To Identify the Clustering Structure. Proc. of ACM-SIGMOD International Conference on Management of Data, pp. 46-60.

[2] Beckmann N., Kriegel H.-P., Schneider R, and Seeger B. 1990. 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, pp. 322-331.

[3] Bradley, P. S., Fayyad, U. M., and Reina, C. A., 1998. Scaling Clustering Algorithms to Large Databases. Proc. Fourth Int'l Conf. Knowledge Discovery and Data Mining, pp. 9-15.

[4] Chen Guang-xue, Li Xiao-zhou, Chen Qi-feng and Li Xiao-zhou, 2010. Clustering Algorithms for Area Geographical Entities in Spatial Data Mining. Seventh International Conference on Fuzzy Systems and Knowledge Discovery, pp. 1630-1633.

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

[6] Faber, V., 1994. Clustering and Continuous k-means Algorithm. Los Almos Science. vol. 22, pp. 138-144.

[7] Gueting R.H., 1994. An Introduction to Spatial Database Systems. The VLDB Journal 3(4), pp. 357-399.

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

[9] In-Soon Kang, Tae-wan Kim and Ki-Joune Li, 1997. A Spatial Data Mining Method by Delaunay Triangulation. Proceedings of the 5th ACM International Workshop on Advances in Geographic Information Systems.

[10] Jong-Sheng Cheng and Mei-Jung Lo, 2001. A Hypergraph Based Clustering Algorithm for Spatial Data Sets. IEEE, pp. 83-90.

[11] Jinxin Dong, Minyong Qi, 2009. K-means Optimization algorithm for Solving Clustering Problem. 2nd International Workshop on Knowledge Discovery and Data Mining, pp.52-55.

[12] Karypis, G., Han, E-H., and Kumar, V., 1999. CHAMELEON: A Hierarchical Clustering Algorithm Using Dynamic Modeling. IEEE Trans. On COMPUTER. vol. 32, pp. 68-75.

[13] MacQueen.J, 1967. Some Methods for Classification and Analysis of Multivariate Observations. Proceedings of 5th Berkeley Symposium on Mathematical Statitics and Probablities. vol.1, pp.281-297.

[14] Tapas Kanungo, David M. Mount, Natha S. Netanyahu, Christine D. Paitko, Ruth Silverman and Angela Y. Wu, 2002. An Efficient k-Means Clustering Algorithm: Analysis and Implementation. IEEE Transactions on Pattern Analysis and Machine Interlligence. vol. 24, no. 7, pp. 881-892.

[15] Xiankun Yangand Weihong Cui, 2010. A Novel Spatial Clustering Algorithm Based on Delaunay Triangulation. Journal of Software Engineering and Applications. vol. 3, pp. 141-149.

[16] Yuan, F., Meng, Z. H., Zhang, H. X., Dong, C. R., 2004. A New Algorithm to Get The Initial Centroids. Proc. of the 3rd International Conference on Machine Learning and Cybernetics. pp. 1191-1193.

[17] Zhang, T., Ramakrishnan, R., Linvy, M., 1996. BIRCH: An Efficient Data Clustering Method for Very Large Databases. Proc. ACM SIGMOD Int. Conf. on Management of Data. ACM Press, New York, p.103-114.