Spatial Clustering Algorithms- An Overview

Article *in* Asian Journal of Computer Science and Information Technology · January 2014

CITATIONS
32

READS
12,322

3 authors:

Bindiya Varghese
Rajagiri College of Social Science
13 PUBLICATIONS **61** CITATIONS
SEE PROFILE

Avittathur Unnikrishnan
Defence Research and Development Organisation
62 PUBLICATIONS **326** CITATIONS
SEE PROFILE

K. Poulose Jacob
Cochin University of Science and Technology
116 PUBLICATIONS **712** CITATIONS
SEE PROFILE

Contents lists available at www.innovativejournal.in

Asian Journal of Computer Science and Information Technology

Journal homepage: http://www.innovativejournal.in/index.php/ajcsit

ISSN 2249 - 5126

SPATIAL CLUSTERING ALGORITHMS - AN OVERVIEW

Bindiya M Varghese[1], Unnikrishnan A[1], Poulose Jacob K[2]

NPOL Kochi, India.
CUSAT Kochi, India.

ARTICLE INFO

Corresponding Author:
Bindiya M Varghese
NPOL Kochi, India
bindiyabhi@gmail.com

ABSTRACT

An Overview of known spatial clustering algorithms The space of interest can be the two-dimensional abstraction of the surface of the earth or a man-made space like the layout of a VLSI design, a volume containing a model of the human brain, or another 3d-space representing the arrangement of chains of protein molecules. The data consists of geometric information and can be either discrete or continuous. The explicit location and extension of spatial objects define implicit relations of spatial neighborhood (such as topological, distance and direction relations) which are used by spatial data mining algorithms. Therefore, spatial data mining algorithms are required for spatial characterization and spatial trend analysis. Spatial data mining or knowledge discovery in spatial databases differs from regular data mining in analogous with the differences between non-spatial data and spatial data. The attributes of a spatial object stored in a database may be affected by the attributes of the spatial neighbors of that object. In addition, spatial location, and implicit information about the location of an object, may be exactly the information that can be extracted through spatial data mining.

INTRODUCTION

Spatial data means data related to space (Güting, 1994). The space of interest can be the two-dimensional abstraction of the surface of the earth or a man-made space like the layout of a VLSI design, a volume containing a model of the human brain, or another 3d-space representing the arrangement of chains of protein molecules. The data consists of geometric information and can be either discrete or continuous. The explicit location and extension of spatial objects define implicit relations of spatial neighborhood (such as topological, distance and direction relations) which are used by spatial data mining algorithms. Therefore, spatial data mining algorithms are required for spatial characterization and spatial trend analysis. Spatial data mining or knowledge discovery in spatial databases differs from regular data mining in analogous with the differences between non-spatial data and spatial data. The attributes of a spatial object stored in a database may be affected by the attributes of the spatial neighbors of that object. In addition, spatial location, and implicit information about the location of an object, may be exactly the information that can be extracted through spatial data mining (Usama Fayyad, Gregory Piatetsky-shapiro,Padhraic Smyth., 1996).

Spatial data

Spatial data consists of data that have a spatial component. Spatial objects can be made up of points, lines, regions, rectangles, surfaces, volumes, and even data of higher dimension which includes time. The spatial component is implemented with a specific location attribute such as address or implicitly done by partitioning the database based on location. Geographic Information systems (GIS), biomedical applications including medical imaging, agricultural science etc. produces large volume of spatial data.

Spatial Clustering

Clustering is a descriptive task that seeks to identify homogeneous groups of objects based on the values of their attributes (Ester, M., Frommelt, A., Kriegel, H.-P., and Sander, J, 1998). In spatial data sets, clustering permits a generalization of the spatial component like explicit location and extension of spatial objects which define implicit relations of spatial neighborhood. Current spatial clustering techniques can be broadly classified into three categories; partitional, hierarchical and locality-based algorithms.

Partition based algorithms

Given a set of objects and a clustering criterion, partitional clustering obtains a partition of objects into clusters such that the objects in a cluster is more similar to the objects inside the cluster than to objects in different clusters. Partitional clustering algorithms attempt to decompose the dataset directly into a set of *k* disjoint clusters, provided *k* is the number of initial clusters. An iterative optimization is done to emphasize the local structure of data, which involves minimizing some measure of dissimilarity in the objects within the cluster, while maximizing the

dissimilarity of different clusters. Partitional algorithms are generally iterative in nature and converge to some local optima. Given a set of data points $x_i \in \Re^d$, $i = 1,\ldots,N$, partitional clustering algorithms aim to organize them into K clusters $\{C_1, \ldots, C_K\}$ while maximizing or minimizing a pre-specified criterion function J.

K-mediod

K-medoids algorithms are partitional algorithm which attempt to minimize squared error, the distance between points labeled to be in a cluster and a point designated as the center of that cluster. A medoid can be defined as that object of a cluster, whose average dissimilarity to all the objects in the cluster is minimal i.e. it is a most centrally located point in the given data set. In contrast to the *k*-means algorithm *k*-medoids chooses data points as centers.

PAM

The Partitioning around medoid (PAM) algorithm represents a cluster by a medoid (Ng, Raymond T. and Jiawei Han., 1994). PAM is based on the search for k representative objects among the objects of the data set. These objects should represent various aspects of the structure of the data are often called centrotypes. In the PAM algorithm the representative objects are the so-called medoid of the clusters (Kaufman and Rousseeuw, 1987). After finding a set of k representative objects, the k clusters are constructed by assigning each object of the data set to the nearest representative object.

Initially, a random set of K items is taken to be the set of medoids. Then, at each step, all items other than the chosen medoids from the input sample set are examined one by one to see if they should be the new medoids. The algorithm chooses the new set of medoids which improves the overall quality of the clustering and replaces the old set of medoids with them.

Let K_i be the cluster represented by the medoid t_i. To swap with a non medoid t_h, the cost change of an item t_j associated with the of exchange of t_i with t_h, C_{jih} has to be computed. (M. Ester, H.-P. Kriegel, S. Jörg, and X. Xu, 1996). The total impact to quality by a medoid change TC_{jih} is given by $TC_{jih} = \sum_{j=1}^{n} Cjih$.

The k-medoid methods are very robust to the existence of outliers. Also, Clusters found by K-medoid methods do not depend on the order in which the objects are examined. They are invariant with respect to translations and orthogonal transformations of data points. PAM does not scale well to large datasets because of its computational complexity. For each iteration, the cost TC_{jih} has to be computed for $k(n-k)$ pair of objects. Thus the total complexity per iteration is $k(n-k)^2$, thereby making PAM not an alternative for large databases.

CLARA

CLARA (Clustering Large Applications) improves on the time complexity of PAM (Ng, Raymond T. and Jiawei Han., 1994). CLARA relies on sampling. PAM is applied to samples drawn from the large datasets. For better approximations, CLARA draws multiple samples and gives best clustering as the result. For accuracy, the quality of a clustering is measured based on the average dissimilarity of all objects in the entire data set, and not only of those objects in the samples.

The method used in CLARA, which was first described by Kaufman and Rousseeuw (1986), is based on the selection of five (or more) random samples of objects. The size of the samples depends on the number of clusters. For a clustering into k clusters, the size of the samples is given by $40 + 2k$. [2]. for CLARA, by applying PAM just to the samples, each iteration is of $O(k(40 + k)2 + k(n - k))$. This explains why CLARA is more efficient than PAM for large values of *n.*

CLARANS

CLARANS (Clustering Large Applications based on Randomized Search) improves on CLARA by using multiple different samples (Ng, Raymond T. and Jiawei Han., 1994). While CLARA draws a sample of nodes at the beginning of a search, CLARANS draws a sample of neighbors in each step of a search. This has the benefit of not confining a search to a localized area. In addition to the normal input to PAM, CLARANs uses two additional parameters; *numlocal* and *maxneighbor*. Numlocal indicates the number of samples to be taken. The *numlocal* also indicates the number of clustering to be made since a new clustering has to be done on every sample. *Maxneighbor* is the number of neighbors of a node to which any specific node can be compared. As *maxneighbor* increases, CLARANs resemble PAM, because all nodes are to be examined. J. Han et al shows the good choice for the parameters are *numlocal = 2* and *maxneighbor = max ((0.0125* x *k(n-k)),250)*. The disadvantage of CLARANS is that it assumes all data are in main memory.

SDCLARANS

Spatial dominant CLARANS assumes the data set to contain spatial and non-spatial components (Ng, Raymond T. and Jiawei Han., 1994). The general approach is to cluster spatial components using CLARANS and then examines the non-spatial within each cluster to derive a description of that cluster. For mining spatial attributes, a tool named DBLEARN is used (Jiawei Han , Yandong Cai , Nick Cercone, 1992). From a learning request, DBLEARN first extracts a set of relevant tuples via SQL queries. Then based on the generalization hierarchies of attributes, it ,iteratively generalizes the tuples. SDCLARANS is a combination of CLARANS and DBLEARN.

NSDCLARANS

Opposite to SDCLARANS, NSDCLARANS considers the non-spatial attributes in the first phase (Jiawei Han , Yandong Cai , Nick Cercone, 1992). DBLEARN is applied to the non-spatial attributes, until the final number of generalized tuples fall below a certain threshold. For each generalized tuple obtained above, the spatial components of the tuples represented by the current generalized tuple are collected, and CLARANS is applied.

K-Mean

K-means is one of the simplest unsupervised learning algorithms used for clustering. K-means partitions n observations into k clusters in which each observation belongs to the cluster with the nearest mean. This algorithm aims at minimizing an *objective function*, in this case a squared error function. The algorithm aims to minimize the objective function $J = \sum_{j=1}^{k} \sum_{i=1}^{n} \left\| x_i^j - c_j \right\|^2$ where $\left\| x_i^j - c_j \right\|^2$ is a chosen distance measure between a data point x_i^j and the cluster centre c_j, is an indicator of the distance of the n data points from their respective cluster centres.

DENCLUE

DENCLUE (Density basted Clustering) is a generalization of partitioning, locality-based and hierarchical or grid-based clustering approaches (A. Hinneburg and D. A. Keim, 1998). The influence of each data point can be modeled formally using a mathematical

function called influence function. This influence function is applied to each data point. The algorithm models the overall point density analytically using the sum of the influence functions of the points. An example influence function can be a Gaussian function $f_{Gauss}(x,y) = e^{-\frac{d(z,y)^2}{2\sigma^2}}$. The density function which results from a guassian influence function is $f^D_{Guass}(x) = \sum_{i=1}^{N} e^{-\frac{d(z,y)^2}{2\sigma^2}}$. Clusters can then be determined mathematically by identifying density attractors. Density attractors are local maxima of the overall density function. These can be either center-defined clusters, similar to k-means clusters, or multi-center-defined clusters, that is a series of center-defined clusters linked by a particular path which identify clusters of arbitrary shape. Clusters of arbitrary shape can also be defined mathematically. The mathematical model requires two parameters, α and ξ. α is a parameter which describes a threshold for the influence of a data point in the data space and ξ is a parameter which sets a threshold for determining whether a density-attractor is significant.

The three major advantages for this method of higher-dimensional clustering claimed by the authors are that the algorithm provides a firm mathematical base for finding arbitrary shaped clusters in high-dimensional datasets. Also, result show good clustering properties in data sets with large amounts of noise and significantly faster than existing algorithms.

Hierarchical

A sequence is said to be a hierarchical clustering if there exists 2 samples, c1 and c2, which belong in the same cluster at some level k and remain clustered together at all higher levels > k. The hierarchy is represented as a tree, called a dendrogram, with individual elements at one end and a single cluster containing every element at the other. Hierarchical clustering algorithms are either top-down or bottom-up. Bottom-up algorithms called hierarchical agglomerative clustering, treat each object as a singleton cluster at the outset and then successively merge (or agglomerate) pairs of clusters until all clusters have been merged into a single cluster. Top-down or divisive clustering proceeds by splitting clusters recursively until individual objects s are reached.

Agglomerative algorithms

CURE

CURE identifies clusters having non-spherical shapes and wide variances in size (Sudipto Guha , Rajeev Rastogi , Kyuseok Shim, 1998). CURE is a bottom-up hierarchical clustering algorithm, but instead of using a centroid-based approach or an all-points approach it employs a method that is based on choosing a well-formed group of points to identify the distance between clusters. CURE achieves this by representing each cluster by a certain fixed number of points that are generated by selecting well scattered points from the cluster. In fact, CURE begins by choosing a constant number, c of well scattered points from a cluster. These points are used to identify the shape and size of the cluster. The next step of the algorithm shrinks the selected points toward the centroid of the cluster using some pre- determined fraction α. These scattered points after shrinking are used as representatives of the cluster. The clusters with the closest pair of representative points are the clusters that are merged at each step of CURE's hierarchical clustering algorithm. CURE is less sensitive to outliers since shrinking the scattered points toward the mean reduces the adverse effects due to outliers since outliers are typically further away from the mean and are thus shifted a larger distance due to the shrinking.

The kinds of clusters identified by CURE can be tuned by varying α: between 0 and 1. CURE reduces to the centroid-based algorithm if $\alpha = 1$, while for $\alpha = 0$, it becomes similar to the all-points approach. CURE's hierarchical clustering algorithm have a space complexity linear to the input size n and has a worst-case time complexity of $O(n^2 \log n)$. For lower dimensions the complexity is further reduced to $O(n^2)$. The overview of CURE algorithm can be diagrammatically represented as [12]

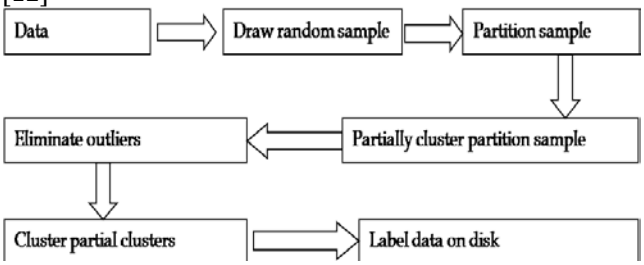

Figure 1 Overview of CURE

ROCK

ROCK (Robust Clustering using links) implements a new concept o links to measure the similarity/proximity between a pair of data points (Sudipto Guha, Rajeev Rastogi, Kyuseok Shim, 1999). A pair of data points are considered neighbors if their similarity exceeds a certain threshold. The number of links between a pair of points is then the common neighbors for the points. Points belonging to a single cluster will have a large number of common neighbors. Let $sim(p_i,p_j)$ be a similarity function that is normalized and captures the closeness between the pair of points p_i and p_j. The sim assumes values between 0 and 1. Given a threshold θ between 0 and 1, a pair of points (p_i, p_j) is defined to be neighbors if $sim(p_i,p_j) > \theta$. $Link(p_i,p_j)$, the number of common neighbors between the pair of points p_i and p_j. The criterion function is to maximize the sum of $link(p_q,p_r)$ for data pairs p_q, p_r belonging to a single cluster and at the same time, minimize the sum of $link(p_q,p_s)$ for p_q and p_s in different clusters. i.e. Maximize $\sum_{i=1}^{k} n_i * \sum_{p_q p_r \in C_i} \frac{link(p_q, p_r)}{n_i^{1+2f(\theta)}}$ where cluster C_i denotes cluster i of size n. The worst case time complexity of the algorithm is $O(n_2 + nm_m m_a + n^2 \log n)$, where m_m is the maximum number of neighbors, m_a is the average number of neighbors, and n is the number of data points. The space complexity is $O(\min\{n^2, nm_m m_a\})$

CHAMELEON

CHAMLEON measures the similarity based on a dynamic model (George Karypis , Eui-Hong (Sam) Han , Vipin Kumar, 1999). Two clusters are merged only if the inter-connectivity and closeness between two clusters are high relative to the internal inter-connectivity of the clusters and closeness of data points within the clusters. CHAMELEON operates on a sparse graph in which nodes represent data items, and weighted edges represent similarities among the data items. This sparse graph representation of the data set allows CHAMELEON to scale to large data sets. CHAMELEON finds the clusters in the data set by using a two phase algorithm. During the first phase, CHAMELEON uses a graph partitioning algorithm to cluster the data items into a large number of relatively small sub-clusters. During the second phase, it uses an agglomerative hierarchical clustering algorithm to find the

genuine clusters by repeatedly combining together these sub-clusters.

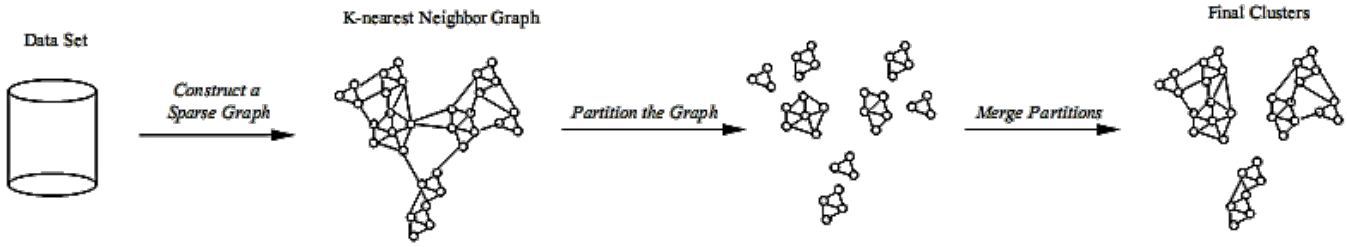

Figure 2 Overview of CHAMELEON algorithm

CHAMELEON's sparse graph representation of the data items is based on the k-nearest neighbor graph approach. Each vertex of the k-nearest neighbor graph represents a data item, and there exists an edge between two vertices, if data items corresponding to either of the nodes are among the k-most similar data points of the data point corresponding to the other node. CHAMELEON determines the similarity between each pair of clusters C_i and C_j by looking both at their relative inter-connectivity $RI(C_i, C_j)$ and their relative closeness $RC(C_i, C_j)$. CHAMELEON's hierarchical clustering algorithm selects to merge the pair of clusters for which both $RI(C_i, C_j)$ and $RC(C_i, C_j)$ are high; i.e., it selects to merge clusters that are well inter-connected as well as close together with respect to the internal inter-connectivity and closeness of the clusters. The relative inter-connectivity between a pair of clusters C_i and C_j is defined as the absolute inter-connectivity between C_i and C_j normalized with respect to the internal inter-connectivity of the two clusters C_i and C_j. The absolute inter-connectivity between a pair of clusters C_i and C_j is defined to be as the sum of the weight of the edges that connect vertices in C_i to vertices in C_j . This is essentially the edge-cut of the cluster, $EC_{\{Ci,Cj\}}$ containing both C_i and C_j such that the cluster is broken into C_i and C_j . The relative inter-connectivity between a pair of clusters Ci and Cj is given by $RI(C_i, C_j) = \frac{|EC\{C_i, C_j\}|}{\frac{|EC_{C_i}| + |EC_{C_j}|}{2}}$ which normalizes the absolute inter-connectivity with the average internal inter-connectivity of the two clusters. The relative closeness between a pair of clusters Ci and Cj is computed as,

$RC(C_i, C_j) = \frac{\bar{S}_{EC\{C_i, C_j\}}}{\frac{|C_i|}{|C_i|+|C_j|}\bar{S}_{EC_{C_i}} + \frac{|C_j|}{|C_i|+|C_j|}\bar{S}_{EC_{C_j}}}$, where $\bar{S}_{EC_{C_i}}$ and $\bar{S}_{EC_{C_j}}$ are the average weights of the edges that belong in the min-cut bisector of clusters C_i and C_j , respectively, and $\bar{S}_{EC\{C_i, C_j\}}$ is the average weight of the edges that connect vertices in C_i to vertices in C_j . The overall complexity of CHAMELEON's two-phase clustering algorithm is $O(nm + n \log n + m^2 \log m)$.

Divisive Algorithms

STING

Statistical Information Grid-based method exploits the clustering properties of index structures (Wei Wang , Jiong Yang , Richard R. Muntz, 1997). The spatial area is divided into rectangular cells which forms a hierarchical structure. Each cell at a high level is partitioned to form a number of cells of the next lower level. Statistical information of each cell is calculated and stored beforehand and is used to answer queries. For each cell, two types of parameters are considered; attribute-dependent and attribute-independent parameters. The attribute- independent parameter are the number of objects (points) in this cell, say n. Attribute-dependent parameters are

- m: mean of all values in this cell
- s: standard deviation of all values of the attribute in this cell
- min: the minimum value of the attribute in this cell
- max: the maximum value of the attribute in this cell
- distribution the type of distribution that the attribute value in this cell follows.

Clustering operations are performed using a top-down method, starting with the root. The relevant cells are determined using the statistical information and only the paths from those cells down the tree are followed. Once the leaf cells are reached, the clusters are formed using a breadth-first search, by merging cells based on their proximity and whether the average density of the area is greater than some specified threshold. The computational complexity is O(K), where K is the number of grid cells at the lowest level. Usually K << N, where N is the number of objects.

STING+

STING+ is an approach to active spatial data mining, which takes advantage of the rich research results of active database systems and the efficient algorithms in STING (Wei Wang , Jiong Yang , Richard R. Muntz, 1997)for passive spatial data mining (Wei Wang, Jiong Yang, Richard Muntz, 1999). A region in STING+ is defined as a set of adjacent leaf level cells. Also, object density and attribute conditions in STING+ are defined in terms of leaf level cells. The density of a leaf level cell is defined as the ratio of the number of objects in this cell divided by the area of this cell. A region is said to have a certain density *c* if and only if the density of every leaf level cell in this region is at least *c*. Conditions on attribute values are defined in a similar manner. Two kinds of conditions can be specified by the user. One condition is an absolute condition, i.e., the condition is satisfied when a certain state is reached. The other type of condition is a relative condition, i.e., the condition is satisfied when a certain degree of change has been detected. Therefore, four categories of triggers are supported by STING+;

1. Region-trigger: absolute condition on certain regions
2. Attribute-trigger: absolute condition on certain attributes
3. Region trigger: relative condition on certain regions
4. Attribute trigger: relative condition on certain attributes.

BIRCH

Balanced Iterative Reducing and Clustering using Hierarchies, is designed for clustering large amount of multidimensional metric data points (Tian Zhang , Raghu Ramakrishnan , Miron Livny, 1996). It requires only one scan of the entire database and uses only a limited memory. BIRCH uses a hierarchical data structure called a CF-tree, or Clustering-Feature-tree that captures the needed information. A clustering-feature vector CF is a triple that stores the information maintained about a cluster. The triple CF= $\{N, \overrightarrow{LS}, SS\}$ contains the number of data points in the cluster, N, and $\overrightarrow{LS}$, the linear sum of the N data points, i.e.$\sum_{i=1}^{N} \overrightarrow{X_i}$, and SS, the square-Sum of the N data points i.e. $\sum_{i=1}^{N} \overrightarrow{X_i}^2$. A CF-tree is a height balanced tree with a branching factor B. each internal node contains a CF triple for each of its children. Each leaf node also represents a cluster and contains a CF entry for each sub cluster in it. A sub cluster in a leaf node must have a diameter no greater than a given threshold value T.

In the pre-clustering phase, the entire database is scanned and an initial in-memory CF-tree is built, representing dense regions of points with compact summaries or sub-clusters in the leaf nodes. Phase 2, rescans the leaf nodes entries to build a smaller CF-tree. It can be used to remove outliers and make larger clusters from sub-clusters. Phase 3 attempts to compensate for the order-dependent input. It uses either an existing centroid based clustering algorithm, or a modification of an existing algorithm applied to the sub-clusters at the leaves as if these sub-clusters were single points. The pre-clustering algorithm is both incremental and approximate.

BIRCH is linear in both space and I/O time. The choice of threshold value is vital to an efficient execution of the algorithm. The worst case complexity of BIRCH can be $O(n^2)$.

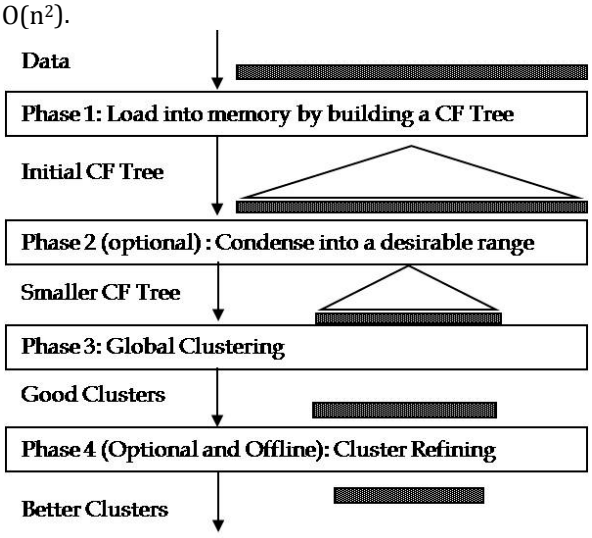

Figure 3 Overview of BIRCH

Grid Based

WAVE CLUSTER

WaveCluster is a clustering approach based on wavelet transforms (Gholamhosein Sheikholeslami , Surojit Chatterjee , Aidong Zhang, 2000). WaveCluster is based on the representation of spatial object as a feature vector where each element of the vector corresponds to one numerical attribute. These feature vectors of the spatial data can be represented in the spatial area, which is termed feature space, where each dimension of the feature space corresponds to one of the features. For an object with n numerical attributes, the feature vector will be one point in the n-dimensional feature space. The collection of objects in the feature space composes an n-dimensional signal. The high frequency parts of the signal correspond to the regions of the feature space where there is a rapid change in the distribution of objects, that is the boundaries of clusters. The low frequency parts of the n-dimensional signal which have high amplitude correspond to the areas of the feature space where the objects are concentrated, i.e., the clusters themselves.

The WaveCluster Algorithm is given below

1. Quantize feature space, and then assign objects to the units.
2. Apply wavelet transform on the feature space.
3. Find the connected components (clusters) in the sub bands of transformed feature space, at different levels.
4. Assign label to the units.
5. Make the lookup table.
6. Map the objects to the clusters.

The complexity of generating clusters is O(n) and is not impacted by Outliers. WaveCluster can find arbitrarily shaped clusters and does not need to know the desired number of clusters.

BANG

BANG structure adapts to the distribution of items so that the dense areas have larger number of smaller grids and less dense areas have a few large ones (Erich Schikuta , Martin Erhart, 1997). BANG organizes the value space containing the patterns. The patterns are treated as points in a k-dimensional value space and are inserted into the BANG-Structure (E. Schikuta, 1996). These points are stored accordingly to their pattern values preserving the topological distribution. The BANG-structure partitions the value space and administers the points by a set of surrounding rectangular shaped blocks. These blocks are then sorted based on their density, which is the number of items in the grid divided by its area. Based on the number of clusters needed, the grids with the highest density are treated as cluster centre.

CLIQUE

CLIQUE, named for Clustering In Quest, the data mining research project at IBM Almaden and is an grid-based approach for high dimensional data sets that provides "automatic sub-space clustering of high dimensional data" (Rakesh Agrawal , Johannes Gehrke , Dimitrios Gunopulos , Prabhakar Raghavan, 1998). CLIQUE identifies dense clusters in subspaces of maximum dimensionality. It generates cluster descriptions in the form of DNF expressions that are minimized for ease of comprehension. It produces identical results irrespective of the order in which input records are presented and does not presume any specific mathematical form for data distribution.

CLIQUE algorithm consists of the following steps;

1. Identification of subspaces that contain clusters.
2. Identification of clusters.
3. Generation of minimal description for the clusters.

The initial phase of the algorithm partitions the data space S into non-overlapping rectangular units, where $S = S = A_1 \; x \; A_2 \; x \; \ldots x \; A_d$ a d-dimensional numerical space.. The units are obtained by partitioning every dimension into ξ intervals of equal length, which is an input parameter. Each unit u is the intersection of one interval from each attribute. The *selectivity* of a unit

is defined to be the fraction of total data points contained in the unit. A unit u is dense, if *selectivity(u)* is greater than τ, where the density threshold τ is another input parameter. Similarly all the units in all subspaces of the original d-dimensional space are defined. A cluster is a maximal set of connected dense units in k-dimensions. Region in k dimensions is an axis-parallel rectangular k-dimensional set. A region can be expressed as a DNF expression on intervals of the domains A_i. A *region* R contained in a cluster C is said to be *maximal* if no proper superset of R is contained in C.A minimal description of a cluster is a non-redundant covering of the cluster with maximal regions.

The running time the algorithm is the exponential in the highest dimensionality of any dense unit. The algorithm makes *k* passes over the database. Thus the time complexity is $O(c^k + m\ k)$ for a constant *c* where *k* is the highest dimensionality of any dense unit and *m* is the number of input points. This algorithm can be improved by pruning the set of dense units to those that lie in "interesting" subspaces using a method called MDL-based pruning or minimal description length. Subspaces with large coverage of dense units are selected and the remainder is pruned.

MOSAIC

MOSAIC greedily merges neighboring clusters maximizing a given fitness function (Jiyeon Choo, Rachsuda Jiamthapthaksin , Chun-sheng Chen, Oner Ulvi Celepcikay, Christian Giusti and Christoph F. Eick, 2007). MOSAIC uses Gabriel graphs to determine which clusters are density- based neighboring and approximates non-convex shapes as the unions of small clusters that have been computed using a representative-based clustering algorithm. The Gabriel graph of a set of points S in the Euclidean plane expresses one notion of proximity or nearness of those points. MOSAIC constructs the Gabriel graph for a given set of representatives, and then uses the Gabriel graph to construct a Boolean merge-candidate relation that describes which of the initial clusters are neighboring. This merge candidate relation is then updated incrementally when clusters are merged.

Density Based Algorithms

DBSCAN

Density based Spatial Clustering of Applications with noise (DBSCAN) creates clusters with a minimum size and density (M. Ester, H.-P. Kriegel, S. Jörg, and X. Xu, 1996). Density is defined as the number of points within a certain distance of each other. The algorithm uses two parameters, *Eps* and *MinPts* to control the density of the cluster. *Minpts*, indicates the minimum number of points in any cluster.

Definitions: The *Eps-neighborhood* of a point is defined by $N_{Eps}(p) = \{ q \in D \mid dist(p, q) \leq Eps \}$. The distance function *dist(p, q)* determines the shape of the neighborhood. Algorithm DBSAN does not require the desired number of cluster as initial input. Two kinds of points in a cluster are specified in the algorithm, i.e. *core points*; points inside of the cluster and *border points*; points on the border of the cluster. An *Eps-neighborhood* of a border point contains significantly less points than an *Eps-neighborhood* of a core point. For every point *p* in a cluster *C* there is a point *q* in *C* so that *p* is inside of the Eps-neighborhood of *q* and $N_{Eps}(q)$ contains at least *MinPts* points. A point p is directly density-reachable from a point q wrt. Eps, MinPts if

1) $p \in N_{Eps}(q)$
2) $|N_{Eps}(q)| \geq Minpts$ $(Core\ point\ condition)$

Directly density-reachable is symmetric for a pair of core points and it is not symmetric if one core point and one border point are involved. A point p is density-reachable from a point q wrt. *Eps* and *MinPts* if there is a chain of points $p1, ..., pn,\ p_1 = q,\ p_n = p$ such that p_{i+1} is directly density-reachable from p_i. A point p is density- connected to a point q wrt. *Eps* and *MinPts* if there is a point *o* such that both, *p* and *q* are density-reachable from *o* wrt. *Eps* and *MinPts*. A cluster C wrt. *Eps* and *MinPts* is a non-empty subset of *D*, where *D* is a database of points, satisfy the following conditions:

1) $\forall p, q$: *if* $p \in C$ and *q* is density-reachable from p wrt. *Eps* and *MinPts*, then $q \in C$. *(Maximality)*
2) $\forall p, q \in C$: *p* is density-connected to q wrt. EPS and MinPts. *(Connectivity)*

The noise is defined as the set of points in the database *D* not belonging to any cluster *Ci* , i.e. noise = $\{p \in D \mid \forall i: p \notin Ci\}$.

Algorithm: DBSCAN starts with an arbitrary point *p* and retrieves all points density-reachable from *p* wrt. *Eps* and *MinPts*. If *p* is a core point, this procedure yields a cluster wrt. *Eps* and *MinPts*. If *p* is a border point, no points are density-reachable from *p* and DBSCAN visits the next point of the database. Merge two clusters, if two clusters of different density are "close" to each other. The algorithm may need to be called recursively with a higher value for *MinPts* if "close" clusters need to be merged because they are within the same *Eps* threshold. The expected time complexity of DBSCAN is $O(n \log n)$.

GDBSCAN

GDBSCAN - can cluster point objects as well as spatially extended objects according to both, their spatial and their non-spatial attributes (Jörg Sander , Martin Ester , Hans-Peter Kriegel , Xiaowei Xu, 1998). GDBSCAN generalizes DBSCAN in two important ways. Any notion of a neighborhood of an object can be used, if the definition of the neighborhood is based on a binary predicate which is symmetric and reflexive. Instead of simply counting the objects in the neighborhood of an object, other measures can be used for example, considering the non-spatial attributes such as the average income of a city, to define the "cardinality" of that neighborhood.

To find a density-connected set, GDBSCAN starts with an arbitrary object p and retrieves all objects density-reachable from p with respect to *NPred;* neighborhood of the object and *MinWeight*; minimum weighted cardinality. If p is a core object, this procedure yields a density-connected set with respect to *NPred* and *MinWeight* . If p is not a core object, no objects are density-reachable from p and p is assigned to Noise, where Noise is defined as the set of objects in the database *D* not belonging to any density-connected set *Ci*. This procedure is iteratively applied to each object p which has not yet been classified.

OPTICS

OPTICS creates an augmented ordering of the database representing its density-based clustering structure (Ankerst, Mihael, Markus M. Breunig, Hans-Peter Kriegel, and Jörg Sander, 1999). Let DB be a database containing n points. The OPTICS algorithm generates an ordering of the points $o:\{1..n\} \rightarrow$DATABASE and corresponding reach ability-values $r:\{1..n\} \rightarrow R \geq 0$. OPTICS does not assign cluster memberships. Instead, the algorithm store the order in which the objects are

processed and the information which would be used by an extended DBSCAN algorithm to assign cluster memberships. This information consists of only two values for each object: the core-distance and a reachability distance. The core-distance of an object *p* is simply the smallest distance *ε'* between *p* and an object in its *ε*-neighborhood such that *p* would be a core object with respect to *ε'* if this neighbor is contained in $N_\varepsilon(p)$. Otherwise, the core-distance is *UNDEFINED*. The reachability-distance of an object *p* with respect to another object *o* is the smallest distance such that *p* is directly density-reachable from *o* if *o* is a core object. Depending on the size of the database, the cluster-ordering can be represented graphically for small data sets or can be represented using appropriate visualization technique for large data sets.

DBCLASD

Distribution Based Clustering of Large Spatial Databases (DBCLASD) is another locality-based clustering algorithm, but unlike DBSCAN, the algorithm assumes that the points inside each cluster are uniformly distributed (Xiaowei Xu , Martin Ester , Hans-Peter Kriegel , Jörg Sander, 1998). Three parameters are defined in the algorithm; $NN_S(q)$, $NN_{Dist}(q)$, and $NN_{DistSet}(S)$. Let q be a query point and *S* be a set of points. Then the nearest neighbor of *q* in *S*, denoted by $NN_S(q)$, is a point *p* in $S - \{q\}$ which has the minimum distance to *q*. The distance from *q* to its nearest neighbor in *S* is called the nearest neighbor distance of *q*, $NN_{Dist}(q)$ for short. Let *S* be a set of points and e_i be the elements of S. The nearest neighbor distance set of S, denoted by $NN_{DistSet}(S)$, or distance set for short, is the multi-set of all values. The probability distribution of the nearest neighbor distances of a cluster is analysed based on the assumption that the points inside of a cluster are uniformly distributed, i.e. the points of a cluster are distributed as a homogeneous Poisson point process restricted to a certain part of the data space. A grid-based representation is used to approximate the clusters as part of the probability calculation. DBCLASD is an incremental algorithm. Points are processed based on the points previously seen, without regard for the points yet to come which makes the clusters produced by DBCLASD dependent on input order. The major advantage of DBCLASD is that it requires no outside input which makes it attractive for larger data sets and sets with larger numbers of attributes.

Others

SParClus

SpaRClus (Spatial RelationshipPattern-Based Hierarchical Clustering) to cluster image data is based on an algorithm, SpIBag (Spatial Item Bag Mining), which discovers frequent spatial patterns in images (S. Kim, X. Jin, and J. Han, 2008). SpIBag is invariant on semi-affine transformations. Semi-affine transformation is a way to express or detect shape preserving images. SpaRClus uses internally SpIBag algorithm to mine frequent patterns, and generates a hierarchical structure of image clusters based on their representative frequent patterns. When SpIBag algorithm generates a frequent n-pattern *p*, SpaRClus computes a scoring function of its support image set $\widetilde{Ip}$ and decides if *p* will be used or not to join with other n-patterns, which enables more pruning power than using SpIBag alone.

C2P

C^2P, Clustering based on Closest Pairs, exploits spatial access methods for the determination of closest pairs (Nanopoulos, A., Theodoridis, Y., and Manolopoulos, 2001). C2P consists of two main phases. The first phase efficiently determines a number of sub-clusters. The first phase of C2P has as input n points and produces m sub-clusters, and it is iterative. The first phase of C2P has the objective of efficiently producing a number of sub-clusters which capture the shape of final clusters. Therefore, it represents clusters with their center points. The second phase uses a different cluster representation scheme to produce the final clustering. The second phase performs the final clustering by using the sub-clusters of the first phase and a different cluster representation scheme. The second phase merges two clusters at each step in order to better control the clustering procedure. The second phase is a specialization of the first, i.e., the latter can be modified in: a) finding different points to represent the cluster instead the center point, b) finding at each iteration only the closest pair of clusters that will be merged, instead of finding for each cluster the one closest to it. The time complexity of $C^2P$for large datasets is O(nlog n), thus it scales well to large inputs.

DBRS+

Density-Based Spatial Clustering in the Presence of Obstacles and Facilitators (DBRS+) aims to cluster spatial data in the presence of both obstacles and facilitators (Wang, X., Rostoker, C., and Hamilton, H. J, 2004). The authors claim that without preprocessing, DBRS+ processes constraints during clustering. It can also find clusters with arbitrary shapes and varying densities. DBRS is a density-based clustering method with three parameters, *Eps*, *MinPts*, and *MinPur*. DBRS repeatedly picks an unclassified point at random and examines its neighborhood, i.e., all points within a radius *Eps* of the chosen point. The purity of the neighborhood is defined as the percentage of the neighbor points with the same non-spatial property as the central point. If the neighborhood is sparsely populated (≤*MinPts*) or the purity of the points in the neighborhood is too low (≤*MinPur*) and disjoint with all known clusters, the point is classified as noise. Otherwise, if any point in the neighborhood is part of a known cluster, this neighborhood is joined to that cluster, i.e., all points in the neighborhood are classified as being part of the known cluster. If neither of these two possibilities applies, a new cluster is begun with this neighborhood. The time complexity of DBRS is *O(n log n)* if an R-tree or SR-tree is used to store and retrieve all points in a neighborhood.

CONCLUSION

The main objective of spatial data mining is to find patterns in data with respect to its locational significance. The scope of spatial mining increases as the data generated from various sources which are geographically referenced increases. Every aspects of applications like health services, marketing, environmental agencies make use of spatial mining to find information contained within. The major challenges in spatial data mining are that the spatial data repositories are tend to be very large and the range and diversity in representing the spatial and non-spatial data attributes in the same canvas. Though the above discussed clustering algorithms try to resolve issues like scalability and complexity, it can be observed that a perfect clustering algorithm which comprehends all the issues with the dataset is an idealistic notion. Current research progresses on more dynamic, adaptive and innovative methods that decipher meaningful patterns that effectively satisfy the requirements of dealing huge volumes of data of higher

dimensionality, insensitive to large noises, unaffected by the order of input, and having no prior knowledge of the domain.

REFERENCES

1. A. Hinneburg and D. A. Keim. (1998). An Efficient Approach to Clustering in Large Multimedia Databases with Noise. *International Conference on Knowledge Discovery and Data Mining*, (pp. 58-65).
2. Ankerst, Mihael, Markus M. Breunig, Hans-Peter Kriegel, and Jörg Sander. (1999). OPTICS: ordering points to identify the clustering structure. *SIGMOD Rec. 28* , (pp. 49-60).
3. E. Schikuta. (1996). Grid-Clustering: An Efficient Hierarchical Clustering Method for Very Large Data Sets. *Proceedings of the 13th International Conference on Pattern Recognition*, (pp. 101-105).
4. Erich Schikuta , Martin Erhart. (1997). The BANG-Clustering System: Grid-Based Data Analysis. *Proceedings of the Second International Symposium on Advances in Intelligent Data Analysis,Reasoning about Data*, (pp. 513-524).
5. Ester, M., Frommelt, A., Kriegel, H.-P., and Sander, J. (1998). Algorithms for characterization and trend detection in spatial databases. *Int. Conf. on Knowledge Discovery and Data Mining*, (pp. 44-50). New York City, NY.
6. George Karypis , Eui-Hong (Sam) Han , Vipin Kumar. (1999). Chameleon: Hierarchical Clustering Using Dynamic Modeling. *Computer, 32*, 68-75.
7. Gholamhosein Sheikholeslami , Surojit Chatterjee , Aidong Zhang. (2000). WaveCluster: a wavelet-based clustering approach for spatial data in very large databases. *The VLDB Journal — The International Journal on Very Large Data Bases, 8*(3-4), 289-304.
8. Güting, R. H. (1994). An Introduction to Spatial Database Systems. *VLDB:Special Issue on Spatial Database System, 3*(4).
9. Jiawei Han , Yandong Cai , Nick Cercone. (1992). Knowledge Discovery in Databases: An Attribute-Oriented Approach. *Proceedings of the 18th International Conference on Very Large Data Bases*, (pp. 547-559).
10. Jiyeon Choo, Rachsuda Jiamthapthaksin , Chun-sheng Chen, Oner Ulvi Celepcikay, Christian Giusti and Christoph F. Eick. (2007). C.F.: MOSAIC: A proximity graph approach for agglomerative clustering. *International Conferenceon Data Warehousing and Knowledge Discovery.* Springer Berlin/Heidelberg.
11. Jörg Sander , Martin Ester , Hans-Peter Kriegel , Xiaowei Xu. (1998). Density-Based Clustering in Spatial Databases: The Algorithm GDBSCAN and Its Applications. *Data Mining and Knowledge Discovery, 2*(2), 169-194.
12. M. Ester, H.-P. Kriegel, S. Jörg, and X. Xu. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. *Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining* , (pp. 226-231).
13. M. Ester, H.-P. Kriegel, S. Jörg, and X. Xu. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. *Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining (KDD-96)*, (pp. 226-231).
14. Nanopoulos, A., Theodoridis, Y., and Manolopoulos. (2001). C 2 P: Clustering based on Closest Pairs. *In Proceedings of the 27th international Conference on Very Large Data Bases* , (pp. 331--340).
15. Ng, Raymond T. and Jiawei Han. (1994). Efficient and Effective Clustering Methods for Spatial Data Mining. *Proceedings of the 20th International Conference on Very Large Data Bases.* (pp. 144-155). San Francisco, CA: USA: Morgan Kaufmann Publishers Inc.
16. Rakesh Agrawal , Johannes Gehrke , Dimitrios Gunopulos , Prabhakar Raghavan. (1998). Automatic subspace clustering of high dimensional data for data mining applications. *Proceedings of the 1998 ACM SIGMOD international conference on Management of data*, (pp. 94-105). Seattle, Washington, United States.
17. S. Kim, X. Jin, and J. Han. (2008). SpaRClus: Spatial Relationship Pattern-Based Hierarchical Clustering. *SIAM International Conference on Data Mining - SDM*, (pp. 49-60).
18. Sudipto Guha , Rajeev Rastogi , Kyuseok Shim. (1998). CURE: an efficient clustering algorithm for large databases. *Proceedings of the 1998 ACM SIGMOD international conference on Management of data*, (pp. 73-84). Seattle.
19. Sudipto Guha, Rajeev Rastogi, Kyuseok Shim. (1999). ROCK: A Robust Clustering Algorithm for Categorical Attributes. *Proceedings of the 15th international Conference on Data Engineering* (pp. 512-521). IEEE Computer Society .
20. Tian Zhang , Raghu Ramakrishnan , Miron Livny. (1996). BIRCH: an efficient data clustering method for very large databases. *Proceedings of the 1996 ACM SIGMOD international conference on Management of data*, (pp. 103-114). Montreal, Canada.
21. Usama Fayyad, Gregory Piatetsky-shapiro,Padhraic Smyth. (1996). Knowledge Discovery and Data Mining: Towards a Unifying Framework.
22. Wang, X., Rostoker, C., and Hamilton, H. J. (2004). Density-based spatial clustering in the presence of obstacles and facilitators. *European Conference on Principles and Practice of Knowledge Discovery in Databases* (pp. 446-458). Pisa, Italy: J. Boulicaut, F. Esposito, F. Giannotti, and D. Pedreschi, Eds.
23. Wei Wang , Jiong Yang , Richard R. Muntz. (1997). STING: A Statistical Information Grid Approach to Spatial Data Mining. *Proceedings of the 23rd International Conference on Very Large Data Bases*, (pp. 186-195).
24. Wei Wang, Jiong Yang, Richard Muntz. (1999). STING+: An Approach to Active Spatial Data Mining. *Proceedings of the 15th International Conference on Data Engineering*, (pp. 116-125).
25. Xiaowei Xu , Martin Ester , Hans-Peter Kriegel , Jörg Sander. (1998). Distribution-Based Clustering Algorithm for Mining in Large Spatial Databases. *Proceedings of the Fourteenth International Conference on Data Engineering*, (pp. 324-331).