International Journal of
Emerging Research in Management &Technology
ISSN: 2278-9359 (Volume-3, Issue-12)

Research Article | **December 2014**

Universally used Clustering Algorithms in Spatial Data Mining – A Survey

[1]G. Dona Rashmi[*], [2]Dr. V. Narayani
[1]Assistant Professor, Department of Master of Computer Applications, Karpagam College of Engineering, India
[2]Director i/c/MCA, Karpagam College of Engineering, India

Abstract –

Spatial data mining is the discovery of interesting relationships and characteristics that may exist implicitly in spatial databases. Spatial clustering, which groups similar spatial objects into classes, is an important component of spatial data mining. Spatial clustering can be used in the identification of areas of similar land usage in an earth observation database or in merging regions with similar weather pattern, etc. As a data mining function, spatial clustering can be used as a stand-alone too to gain insight into the distribution of data, to observe the characteristics of each cluster and to focus on particular set of clusters in future analysis. It may also serve as a preprocessing step for other algorithms, such as classification and characterization, which will operate on the detected clusters. Due to its immense applications in various areas, spatial clustering has been a highly active topic in data mining research, with fruitful, scalable clustering methods developed recently. Spatial clustering methods can be classified into four categories: partitioning method, hierarchical method, density-based method and grid-based method. In this paper we present the functionalities of these clustering algorithms, finally a comparative study is proposed.

Keywords -Spatial Database Management System (SDMS), Medoids, Agglomerative, Divisive, statistical density

I. INTRODUCTION

Spatial clustering is the process of grouping a set of objects into classes or clusters so that objects within a cluster have high similarity in comparison to one another, but are dissimilar to objects in other clusters [1]. As a branch of statistics, cluster analysis has been studied extensively for many years, focusing mainly on distance-based cluster analysis.

It is a main task of exploratory data mining, and a common technique for statistical data analysis, used in many fields, including machine learning, pattern recognition, image analysis, information retrieval, and bioinformatics[4]. We focus on clustering algorithms which works reasonably well for large geographical database. These algorithms mostly work on numerical attributes and can be separated into four general categories: partitioning method, hierarchical method, density based method and grid based method.

II. PARTITIONING METHODS

Suppose we are given a database of n objects, the partitioning method construct k partition of data. Each partition will represents a cluster and k≤n[2]. It means that it will classify the data into k groups, which satisfy the following requirements:

- Each group contains at least one object.
- Each object must belong to exactly one group.

Construct a partition of a database ***D*** of ***n*** objects into a set of ***k*** clusters, s.t., min sum of squared distance

$$\sum_{m=1}^{k}\sum_{t_{mi}\in Km}(C_m - t_{mi})^2$$

Given a *k*, find a partition of *k clusters* that optimizes the chosen partitioning criterion

- Global optimal: exhaustively enumerate all partitions
- Heuristic methods: *k-means* and *k-medoids* algorithms
- *k-means* : Each cluster is represented by the center of the cluster
- *k-medoids* or PAM (Partition around medoids) : Each cluster is represented by one of the objects in the cluster

A. The K-Means Clustering Method

Given *k*, the *k-means* algorithm is implemented in four steps:
Step 1: Partition objects into *k* nonempty subsets
Step 2: Compute seed points as the centroids of the clusters of the current partition (the centroid is the center, i.e., *mean point*, of the cluster)
Step 3: Assign each object to the cluster with the nearest seed point
Step4: Go back to Step 2, stop when no more new assignment

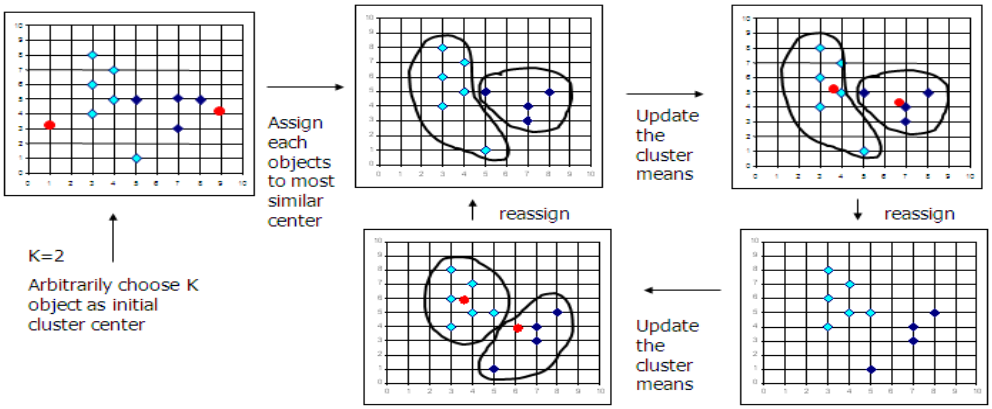

Fig 1: Example for k-means clustering

Comments on the *K-Means* Method

Strength:

Relatively efficient: *O*(*tkn*), where *n* is # objects, *k* is # clusters, and *t* is # iterations. Normally, *k*, *t* << *n*.

Comparing: PAM: $O(k(n-k)^2)$, CLARA: $O(ks^2 + k(n-k))$

Comment: Often terminates at a *local optimum*[6]. The *global optimum* may be found using techniques such as: *deterministic annealing* and *genetic algorithms*

Weakness

- Applicable only when *mean* is defined, not about categorical data
- Need to specify *k*, the *number* of clusters, in advance
- Unable to handle noisy data and *outliers*
- Not suitable to discover clusters with *non-convex shapes*

Problem of the K-Means Method

- The k-means algorithm is sensitive to outliers. Since an object with an extremely large value may substantially distort the distribution of the data.
- K-Medoids: Instead of taking the **mean** value of the object in a cluster as a reference point, **medoids** can be used, which is the **most centrally located** object in a cluster.

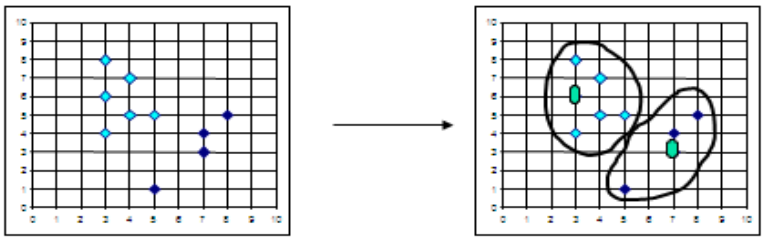

Fig 2: k-Means Method

B.The K-Medoids Clustering Method

Finding *representative* objects, called medoids, in clusters.*PAM* (Partitioning Around Medoids) starts from an initial set of medoids and iteratively replaces one of the medoids by one of the non-medoids if it improves the total distance of the resulting clustering[3].

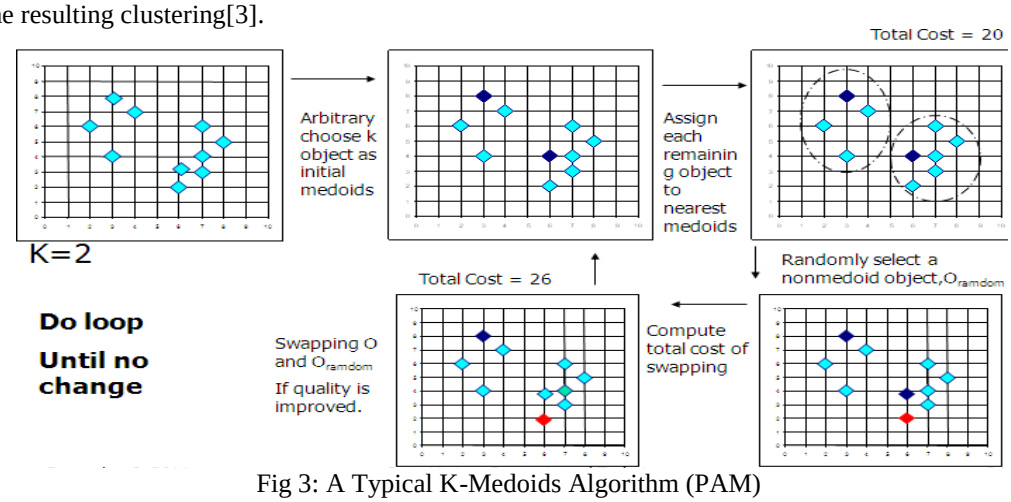

Fig 3: A Typical K-Medoids Algorithm (PAM)

PAM works effectively for small data sets, but does not scale well for large data sets. It also includes the following:

- *CLARA*
- *CLARANS*: Randomized sampling

PAM (Partitioning Around Medoids)

It uses real object to represent the cluster. The following steps are followed:

Step 1: Select *k* representative objects arbitrarily
Step 2: For each pair of non-selected object *h* and selected object *i*, calculate the total swapping cost TC_{ih}
Step 3: For each pair of *i* and *h*,

If $TC_{ih} < 0$, *i* is replaced by *h*

Then assign each non-selected object to the most similar representative object

Step 4: Repeat steps 2-3 until there is no change

Problem with PAM

- Pam is more robust than k-means in the presence of noise and outliers because a medoid is less influenced by outliers or other extreme values than a mean
- Pam works efficiently for small data sets but does not scale well for large data sets.
- $O(k(n-k)^2)$ for each iteration ,where n is # of data, k is # of clusters

CLARA (Clustering Large Applications)

It draws *multiple samples* of the data set, applies *PAM* on each sample, and gives the best clustering as the output [8]. It is built in statistical analysis packages, such as S+.

Strength: Deals with larger data sets than *PAM*

Weakness:

- Efficiency depends on the sample size
- A good clustering based on samples will not necessarily represent a good clustering of the whole data set if the sample is biased

CLARANS ("Randomized" CLARA)

CLARANS (A Clustering Algorithm based on Randomized Search), draws sample of neighbors dynamically. The clustering process can be presented as searching a graph where every node is a potential solution, that is, a set of *k* medoids[7]. If the local optimum is found, *CLARANS* starts with new randomly selected node in search for a new local optimum. It is more efficient and scalable than both *PAM* and *CLARA*. Focusing techniques and spatial access structures may further improve its performance.

III. HIERARCHICAL CLUSTERING METHODS

Hierarchical clustering involves creating clusters that have a predetermined ordering from top to bottom. For example, all files and folders on the hard disk are organized in a hierarchy. There are two types of hierarchical clustering, *Divisive* and *Agglomerative*.

i)Agglomerative Approach

This approach is also known as bottom-up approach. In this we start with each object forming a separate group. It keeps on merging the objects or groups that are close to one another[2]. It keeps on doing so until all of the groups are merged into one or until the termination condition holds.

ii)Divisive Approach

This approach is also known as top-down approach. In this we start with all of the objects in the same cluster. In the continuous iteration, a cluster is split up into smaller clusters. It is down until each object in one cluster or the termination condition holds.

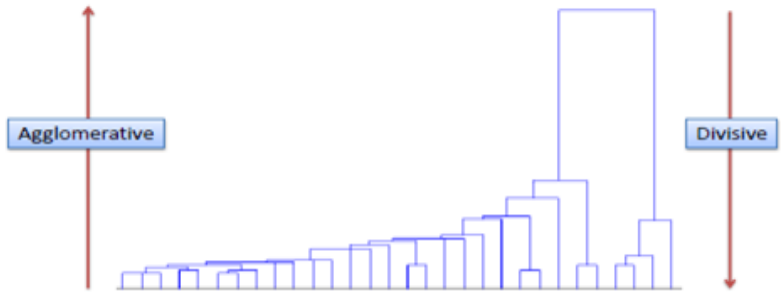

Fig 4: Hierarchical clustering

AGNES (Agglomerative Nesting)

It was introduced was Kaufmann and Rousseeuw (1990), It was implemented in statistical analysis packages, e.g., Splus . It uses the Single-Link method and the dissimilarity matrix [1]. It also merges nodes that have the least dissimilarity. It just goes on in a non-descending fashion, eventually all nodes belong to the same cluster

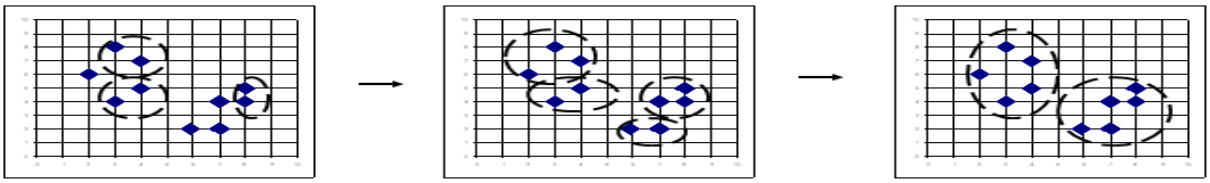

Fig 5: AGNES

DIANA (Divisive Analysis)
It was introduced by Kaufmann and Rousseeuw (1990), implemented in statistical analysis packages, e.g., Splus. It is inverse order of AGNES, eventually each node forms a cluster on its own.

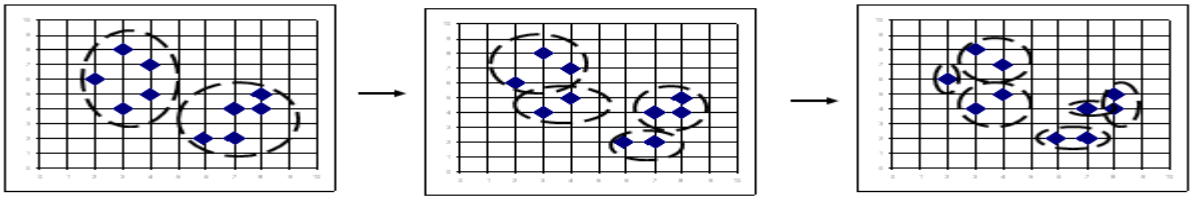

Fig 6: DIANA

Recent Hierarchical Clustering Methods

Major weakness of agglomerative clustering methods is it does not scale well. Time complexity of at least $O(n^2)$, where n is the number of total objects. It can never undo what was done previously. Integration of hierarchical with distance-based clustering leads to following clustering methods:

- BIRCH : uses CF-tree and incrementally adjusts the quality of sub-clusters
- ROCK : clustering categorical data by neighbor and link analysis
- CHAMELEON : hierarchical clustering using dynamic modeling

BIRCH(Balanced Iterative Reducing and Clustering using Hierarchies)

It incrementally constructs a CF (Clustering Feature) tree, a hierarchical data structure for multiphase clustering. The following phases to be followed:

- Phase 1: scan DB to build an initial in-memory CF tree (a multi-level compression of the data that tries to preserve the inherent clustering structure of the data)
- Phase 2: use an arbitrary clustering algorithm to cluster the leaf nodes of the CF-tree

Strength: Finds a good clustering with a single scan and improves the quality with a few additional scans
Weaknes: Handles only numeric data, and sensitive to the order of the data record.
The ROCK Algorithm(RObust Clustering using links)
It was introduced by S. Guha, R. Rastogi & K. Shim. The major ideas are:

- Use links to measure similarity/proximity
- Not distance-based
- Computational complexity

Algorithm: sampling-based clustering
Step 1: Draw random sample
Step 2: Cluster with links
Step 3: Label data in disk

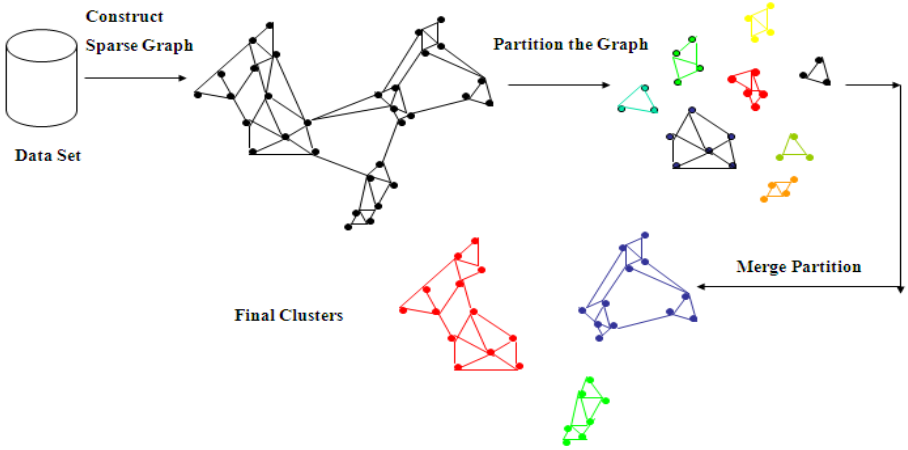

Fig 7: Overall Framework of CHAMELEON

CHAMELEON: Hierarchical Clustering Using Dynamic Modeling

CHAMELEON was given by G. Karypis, E.H. Han, and V. Kumar'99. It measures the similarity based on a dynamic mode [3]. Two clusters are merged only if the *interconnectivity* and *closeness (proximity)* between two clusters are high *relative to* the internal interconnectivity of the clusters and closeness of items within the clusters. **Cure** ignores information about **interconnectivity** of the objects; **Rock** ignores information about the **closeness** of two clusters

A two-phase algorithm

- Use a graph partitioning algorithm: cluster objects into a large number of relatively small sub-clusters
- Use an agglomerative hierarchical clustering algorithm: find the genuine clusters by repeatedly combining these sub-clusters

IV. DENSITY BASED CLUSTERING METHODS

This method is based on the notion of density. The basic idea is to continue growing the given cluster as long as the density in the neighborhood exceeds some threshold i.e. for each data point within a given cluster [5]; the radius of a given cluster has to contain at least a minimum number of points.

Major features:

- Discover clusters of arbitrary shape
- Handle noise
- One scan
- Need density parameters as termination condition

Several other density based clustering methods are: DBSCAN, OPTICS, DENCLUE, CLIQUE

DBSCAN: Density Based Spatial Clustering of Applications with Noise

- Relies on a *density-based* notion of cluster: A *cluster* is defined as a maximal set of density-connected points
- Discovers clusters of arbitrary shape in spatial databases with noise

DBSCAN: The Algorithm

Step 1: Arbitrary select a point *p*

Step 2: Retrieve all points density-reachable from *p* w.r.t. *Eps* and *MinPts*.

Step 3: If *p* is a core point, a cluster is formed.

Step 4: If *p* is a border point, no points are density-reachable from *p* and DBSCAN visits the next point of the database.

Step 5: Continue the process until all of the points have been processed.

OPTICS: A Cluster-Ordering Method

OPTICS(Ordering Points To Identify the Clustering Structure) was introduced by Ankerst, Breunig, Kriegel, and Sander. It produces a special order of the database and its density-based clustering structure [5]. This cluster-ordering contains information equivalent to the density-based clusterings corresponding to a broad range of parameter settings.

It is good for both automatic and interactive cluster analysis, including finding intrinsic clustering structure; it can be represented graphically or using visualization techniques. It is an extension from DBSCAN.

DENCLUE: Using Statistical Density Functions

DENsity-based CLUstEring was introduced by Hinneburg & Keim. It is using statistical density functions:

$$f_{Gaussian}(x,y) = e^{-\frac{d(x,y)^2}{2\sigma^2}}$$

$$f^{D}_{Gaussian}(x) = \sum_{i=1}^{N} e^{-\frac{d(x,x_i)^2}{2\sigma^2}}$$

$$\nabla f^{D}_{Gaussian}(x,x_i) = \sum_{i=1}^{N} (x_i - x) \cdot e^{-\frac{d(x,x_i)^2}{2\sigma^2}}$$

Major features

- Solid mathematical foundation
- Good for data sets with large amounts of noise
- Allows a compact mathematical description of arbitrarily shaped clusters in high-dimensional data sets
- Significant faster than existing algorithm (e.g., DBSCAN)
- But needs a large number of parameters

CLIQUE (Clustering In QUEst)

CLIQUE was given by Agrawal, Gehrke, Gunopulos, Raghavan .It automatically identifying subspaces of a high dimensional data space that allow better clustering than original space. CLIQUE can be considered as both density-based and grid-based. It partitions each dimension into the same number of equal length interval

It partitions an m-dimensional data space into non-overlapping rectangular units. A unit is dense if the fraction of total data points contained in the unit exceeds the input model parameter [8]. A cluster is a maximal set of connected dense units within a subspace

The Major Steps

- Partition the data space and find the number of points that lie inside each cell of the partition.
- Identify the subspaces that contain clusters using the Apriori principle
- Identify clusters
    - Determine dense units in all subspaces of interests
    - Determine connected dense units in all subspaces of interests.
- Generate minimal description for the clusters
    - Determine maximal regions that cover a cluster of connected dense units for each cluster
    - Determination of minimal cover for each cluster

Strength

- *automatically* finds subspaces of the highest dimensionality such that high density clusters exist in those subspaces
- *insensitive* to the order of records in input and does not presume some canonical data distribution
- scales *linearly* with the size of input and has good scalability as the number of dimensions in the data increases

Weakness

- The accuracy of the clustering result may be degraded at the expense of simplicity of the method

V. GRID-BASED CLUSTERING METHOD

The grid-based clustering approach differs from the conventional clustering algorithms in that it is concerned not with the data points but with the value space that surrounds the data points [2]. In general, a typical grid-based clustering algorithm consists of the following five basic steps:

1. Creating the grid structure, i.e., partitioning the data space into a finite number of cells.
2. Calculating the cell density for each cell.
3. Sorting of the cells according to their densities.
4. Identifying cluster centers.
5. Traversal of neighbor cells.

It uses multi-resolution grid data structure. Several interesting methods are:

- STING (a STatistical INformation Grid approach) by Wang, Yang and Muntz
- WaveCluster by Sheikholeslami, Chatterjee, and Zhang - A multi-resolution clustering approach using wavelet method
- CLIQUE by Agrawal - On high-dimensional data (thus put in the section of clustering high-dimensional data

STING: A Statistical Information Grid Approach

The spatial area is divided into rectangular cells. There are several levels of cells corresponding to different levels of resolution[1]. Each cell at a high level is partitioned into a number of smaller cells in the next lower level. Statistical information of each cell is calculated and stored before hand and is used to answer queries.

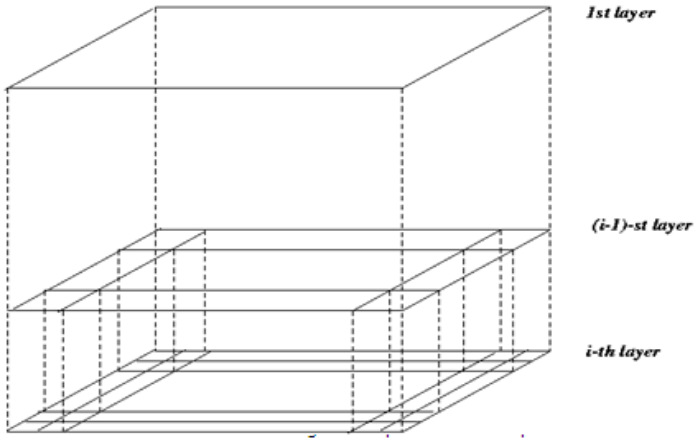

Fig 8: The STING Clustering Method

Parameters of higher level cells can be easily calculated from parameters of lower level cell (*count*, *mean*, *s*, *min*, *max*, type of distribution—normal, *uniform*, etc.). Use a top-down approach to answer spatial data queries. It starts from a pre-selected layer—typically with a small number of cells. For each cell in the current level compute the confidence interval. Remove the irrelevant cells from further consideration.When finish examining the current layer, proceed to the next lower level Repeat this process until the bottom layer is reached

Advantages:

- Query-independent, easy to parallelize, incremental update
- *O(K),* where *K* is the number of grid cells at the lowest level

Disadvantages:

- All the cluster boundaries are either horizontal or vertical, and no diagonal boundary is detected

WaveCluster: Clustering by Wavelet Analysis

It was given by Sheikholeslami, Chatterjee, and Zhang. It is a multi-resolution clustering approach which applies wavelet transform to the feature space. Steps to apply wavelet transform to find clusters

- Summarizes the data by imposing a multidimensional grid structure onto data space
- These multidimensional spatial data objects are represented in a n-dimensional feature space
- Apply wavelet transform on feature space to find the dense regions in the feature space
- Apply wavelet transform multiple times which result in clusters at different scales from fine to coarse

Wavelet transform is a signal processing technique that decomposes a signal into different frequency sub-band (can be applied to n-dimensional signals)[2]. Data are transformed to preserve relative distance between objects at different levels of resolution. It allows natural clusters to become more distinguishable

Major features:

- Complexity O(N)
- Detect arbitrary shaped clusters at different scales
- Not sensitive to noise, not sensitive to input order
- Only applicable to low dimensional data

VI. MODEL-BASED CLUSTERING

It is attempt to optimize the fit between the given data and some mathematical model. It is based on the assumption that data are generated by a mixture of underlying probability distribution. Typical methods used are:

- Statistical approach - EM (Expectation maximization)
- Machine learning approach - COBWEB, CLASSIT
- Neural network approach - SOM (Self-Organizing Feature Map)

EM — Expectation Maximizationx

EM is a popular iterative refinement algorithm. It is an extension to k-means [3]; assign each object to a cluster according to a weight (prob. distribution).New means are computed based on weighted measures.

General idea is: Starts with an initial estimate of the parameter vector, iteratively rescores the patterns against the mixture density produced by the parameter vector, the rescored patterns are used to update the parameter updates, patterns belonging to the same cluster, if they are placed by their scores in a particular component. Algorithm converges fast but may not be in global optima

Algorithm

- Initially, randomly assign k cluster centers
- Iteratively refine the clusters based on two steps
- Expectation step: assign each data point X_i to cluster C_i with the following probability

$$P(X_i \in C_k) = p(C_k|X_i) = \frac{p(C_k)p(X_i|C_k)}{p(X_i)},$$

- Maximization step: Estimation of model parameters

$$m_k = \frac{1}{N}\Sigma_{i=1}^{N}\frac{X_i P(X_i \in C_k)}{\Sigma_j P(X_i \in C_j)}.$$

VII. CONSTRAINT-BASED CLUSTER ANALYSIS

Constrained clustering clusters that satisfy user-specified constraints is highly desirable in many applications. Constrained clustering is a class of semi-supervised learning algorithms. Typically, constrained clustering incorporates either a set of must-link constraints, cannot-link constraints, or both, with a Data clustering algorithm [8]. Both a must-link and a cannot-link constraint define a relationship between two data instances. A must-link constraint is used to specify that the two instances in the must-link relation should be associated with the same cluster. A cannot-link constraint is used to specify that the two instances in the cannot-link relation should *not* be associated with the same cluster.

Different constraints in cluster analysis:

- Constraints on individual objects (do selection first) - Cluster on houses worth over $300K
- Constraints on distance or similarity functions - Weighted functions, obstacles (e.g., rivers, lakes)

- Constraints on the selection of clustering parameters - # of clusters, MinPts, etc.
- User-specified constraints - Contain at least 500 valued customers and 5000 ordinary ones
- Semi-supervised: giving small training sets as "constraints" or hints

Clustering With Obstacle Objects
K-medoids is more preferable since k-means may locate the ATM center in the middle of a lake[2]. It uses visibility graph and shortest path, triangulation and micro-clustering. Two kinds of join indices (shortest-paths) worth pre-computation

- VV index: indices for any pair of obstacle vertices
- MV index: indices for any pair of micro-cluster and obstacle indices

Clustering with User-Specified Constraints
Example: Locating k delivery centers, each serving at least m valued customers and n ordinary ones. Efficiency is improved by micro-clustering.
Proposed approach:

- Find an initial "solution" by partitioning the data set into k groups and satisfying user-constraints
- Iteratively refine the solution by micro-clustering relocation (e.g., moving δ μ-clusters from cluster C_i to C_j) and "deadlock" handling (break the micro clusters when necessary)

VII. COMPARATIVE STUDY

Algorithm	**Type of Cluster**	**Dimensionality of Data**	**Parameters Used**	**Shapes of clusters**	**Worst Case**
K-means method	Local Clustering	Scalar data	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 inter0connectivity RI(Ci,Cj), relative closeness RC(Ci,Cj)	Arbitrary shapes	$O(nm+n\log n+m^2\log m)$
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)

VIII. CONCLUSION

We have presented an overview of clustering algorithms that are useful to geographical community. The spatial database is characterized with some unique features and so has specific requirements in a clustering algorithm. Specifically the geographic databases in the SDBMS (Spatial Database Management System) have noisy data and outliers, which the algorithm should handle effectively. A recent clustering algorithm group's datasets into clusters with the spatial object as such as the core object and it utilizes Euclidean distance for that purpose. The algorithm that is efficient in processing geographic databases of the application would be implemented in the future work.

REFERENCES

[1] 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.
[2] Chen Guang-xue, Li Xiao-zhou, Chen Qi-feng and Li Xiaozhou, 2010. *Clustering Algorithms for Area Geographical Entities in Spatial Data Mining.* Seventh International Conference on Fuzzy Systems and Knowledge Discovery, pp. 1630-1633.
[3] 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.
[4] J. Han and M. Kamber, "*Data mining: Concepts and Techniques*," Academic Press, 2001.

[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] 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.

[7] R. T. Ng and J. Han, "*CLARANS: A method for cluster-ing objects for spatial data mining*," IEEE Transactions on Knowledge and Data Engineering, Vol. 14, No. 5, pp. 1003–1016, 2002.

[8] 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.

