

An Analysis on Density Based Clustering of Multi Dimensional Spatial Data

K. Mumtaz¹

Assistant Professor, Department of MCA

Vivekanandha Institute of Information and Management Studies, Tiruchengode, Tamil Nadu, India

mumtazmunaf@yahoo.co.in

Dr. K. Duraiswamy²

Dean (Academic)

KS Rangasamy College of Technology, Tiruchengode, Tamil Nadu, India

drkduraiswamy@yahoo.co.in

Abstract

Mining knowledge from large amounts of spatial data is known as spatial data mining. It becomes a highly demanding field because huge amounts of spatial data have been collected in various applications ranging from geo-spatial data to bio-medical knowledge. The amount of spatial data being collected is increasing exponentially. So, it far exceeded human's ability to analyze. Recently, clustering has been recognized as a primary data mining method for knowledge discovery in spatial database. The development of clustering algorithms has received a lot of attention in the last few years and new clustering algorithms are proposed. DBSCAN is a pioneer density based clustering algorithm. It can find out the clusters of different shapes and sizes from the large amount of data containing noise and outliers. This paper shows the results of analyzing the properties of density based clustering characteristics of three clustering algorithms namely DBSCAN, k-means and SOM using synthetic two dimensional spatial data sets.

Keywords: Clustering, DBSCAN, k-means, SOM.

Introduction

Clustering is considered as one of the important techniques in data mining and is an active research topic for the researchers. The objective of clustering is to partition a set of objects into clusters such that objects within a group are more similar to one another than patterns in different clusters. So far, numerous useful clustering algorithms have been developed for large databases, such as K-MEANS [4], CLARANS [6], BIRCH [10], CURE [3], DBSCAN [2], OPTICS [1], STING [9] and CLIQUE [5]. These algorithms can be divided into several categories. Three prominent categories are partitioning, hierarchical and density-based. All these algorithms try to challenge the clustering problems treating huge amount of data in large databases. However, none of them are the most effective.

In density-based clustering algorithms, which are designed to discover clusters of arbitrary shape in databases with noise, a cluster is defined as a high-density region partitioned by low-density regions in data space. Density Based Spatial Clustering of Applications with Noise (DBSCAN) [2] is a typical density-based clustering algorithm. In this paper, we analyze the properties of density based clustering characteristics of three clustering algorithms namely DBSCAN, k-means and SOM.

DBSCAN Algorithm

Density-Based Spatial Clustering and Application with Noise (DBSCAN) was a clustering algorithm based on density. It did clustering through growing high density area, and it can find any shape of clustering (Rong *et al.*, 2004). The idea of it was:

1. ϵ -neighbor: the neighbors in ϵ semi diameter of an object
2. Kernel object: certain number ($MinP$) of neighbors in ϵ semi diameter
3. To a object set D , if object p is the ϵ -neighbor of q , and q is kernel object, then p can get "direct density reachable" from q .
4. To a ϵ , p can get "direct density reachable" from q ; D contains $Minp$ objects; if a series object $p_1, p_2, \dots, p_n, p_1 = q, p_n = p$. then p_{i+1} can get "direct density reachable" from $p_i, p_i \in D, 1 \leq i \leq n$
5. To ϵ and $MinP$, if there exist a object $o(o \in D)$, p and q can get "direct density reachable" from o , p and q are density connected.

Explanation of DBSCAN Steps

- DBSCAN requires two parameters: epsilon (eps) and minimum points (minPts). It starts with an arbitrary starting point that has not been visited. It then finds all the neighbor points within distance eps of the starting point.
- If the number of neighbors is greater than or equal to minPts, a cluster is formed. The starting point and its neighbors are added to this cluster and the starting point is marked as visited. The algorithm then repeats the evaluation process for all the neighbors recursively.
- If the number of neighbors is less than minPts, the point is marked as noise.
- If a cluster is fully expanded (all points within reach are visited) then the algorithm proceeds to iterate through the remaining unvisited points in the dataset.

Advantages

1. DBSCAN does not require you to know the number of clusters in the data a priori, as opposed to k-means.
2. DBSCAN can find arbitrarily shaped clusters. It can even find clusters completely surrounded by (but not connected to) a different cluster. Due to the MinPts parameter, the so-called single-link effect (different clusters being connected by a thin line of points) is reduced.
3. DBSCAN has a notion of noise.
4. DBSCAN requires just two parameters and is mostly insensitive to the ordering of the points in the database.

Disadvantages

1. DBSCAN can only result in a good clustering as good as its distance measure is in the function `getNeighbors(P,epsilon)`. The most common distance metric used is the euclidean distance measure. Especially for high-dimensional data, this distance metric can be rendered almost useless.
2. DBSCAN does not respond well to data sets with varying densities (called hierarchical data sets).

k-means Algorithm

The naive k-means algorithm partitions the dataset into 'k' subsets such that all records, from now on referred to as points, in a given subset "belong" to the same center. Also the points in a given subset are closer to that center than to any other center.

The algorithm keeps track of the centroids of the subsets, and proceeds in simple iterations. The initial partitioning is randomly generated, that is, we randomly initialize the centroids to some points in the region of the space. In each iteration step, a new set of centroids is generated using the existing set of centroids following two very simple steps. Let us denote the set of centroids after the i^{th} iteration by $C^{(i)}$. The following operations are performed in the steps:

- (i) Partition the points based on the centroids $C^{(i)}$, that is, find the centroids to which each of the points in the dataset belongs. The points are partitioned based on the Euclidean distance from the centroids.
- (ii) Set a new centroid $c^{(i+1)} \in C^{(i+1)}$ to be the mean of all the points that are closest to $c^{(i)} \in C^{(i)}$
 - (i) The new location of the centroid in a particular partition is referred to as the new location of the old centroid.

The algorithm is said to have converged when recomputing the partitions does not result in a change in the partitioning. In the terminology that we are using, the algorithm has converged completely when $C^{(i)}$ and $C^{(i-1)}$ are identical. For configurations where no point is equidistant to more than one center, the above convergence condition can always be reached. This convergence property along with its simplicity adds to the attractiveness of the k-means algorithm.

The k-means needs to perform a large number of "nearest-neighbour" queries for the points in the dataset. If the data is 'd' dimensional and there are 'N' points in the dataset, the cost of a single iteration is $O(kdN)$. As one would have to run several iterations, it is generally not feasible to run the naïve k-means algorithm for large number of points.

Sometimes the convergence of the centroids (i.e. $C^{(i)}$ and $C^{(i+1)}$ being identical) takes several iterations. Also in the last several iterations, the centroids move very little. As running the expensive iterations so many more times might not be efficient, we need a measure of convergence of the centroids so that we stop the iterations when the convergence criteria are met. Distortion is the most widely accepted measure.

SOM Algorithm

A Self-Organizing Map (SOM) or self-organizing feature map (SOFM) is a neural network approach that uses competitive unsupervised learning. Learning is based on the concept that the behavior of a node should impact only those nodes and arcs near it. Weights are initially assigned randomly and adjusted during the learning process to produce better results. During this learning process, hidden features or patterns in the data are uncovered and the weights are adjusted accordingly. The model was first described by the Finnish professor Teuvo Kohonen and is thus sometimes referred to as a Kohonen map.

The self-organizing map is a single layer feed forward network where the output syntaxes are arranged in low dimensional (usually 2D or 3D) grid. Each input is connected to all output neurons. There is a weight vector attached to every neuron with the same dimensionality as the input vectors. The goal of the learning in the self-organizing map is to associate different parts of the SOM lattice to respond similarly to certain input patterns.

Training of SOM

Initially, the weights and learning rate are set. The input vectors to be clustered are presented to the network. Once the input vectors are given, based on the initial weights, the winner unit is calculated either by Euclidean distance method or sum of products method.

Based on the winner unit selection, the weights are updated for that particular winner unit. An epoch is said to be completed once all the input vectors are presented to the network. By updating the learning rate, several epochs of training may be performed.

A two dimensional Kohonen Self Organizing Feature Map network is shown in figure 1 which is given below.

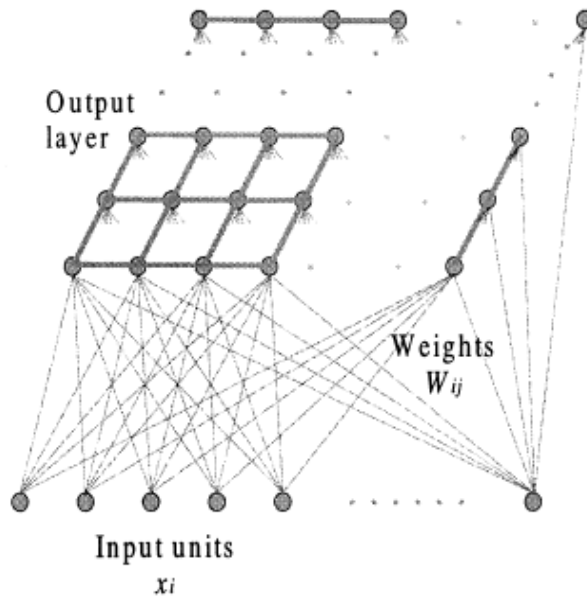


Fig. 1. The SOM Network

Evaluation and Results

The Test Database

To evaluate the performance of the clustering algorithms, two dimensional spatial data sets were used and the properties of density based clustering characteristics of the clustering algorithms were evaluated.

The first type of data sets was prepared from the guideline images of some of the main reference papers of DBSCAN algorithm. So that data was handled from image format. The figure 2 shows the type of spatial data used for testing the algorithms.

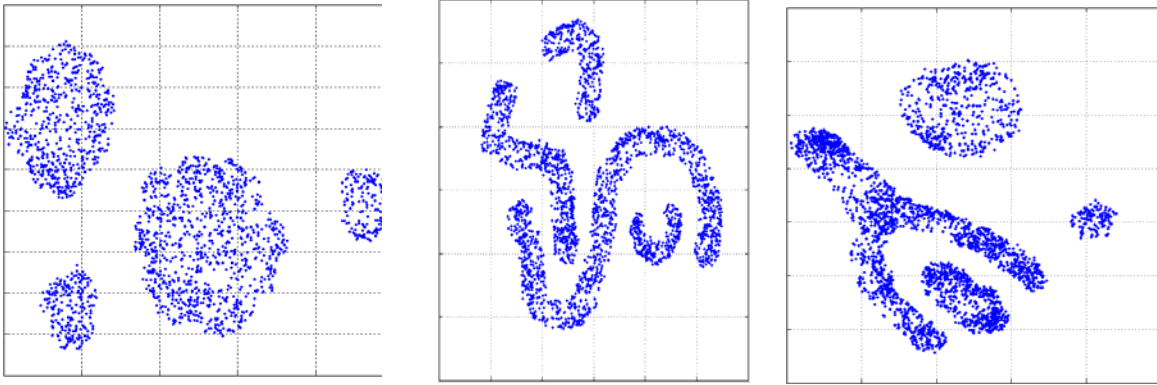
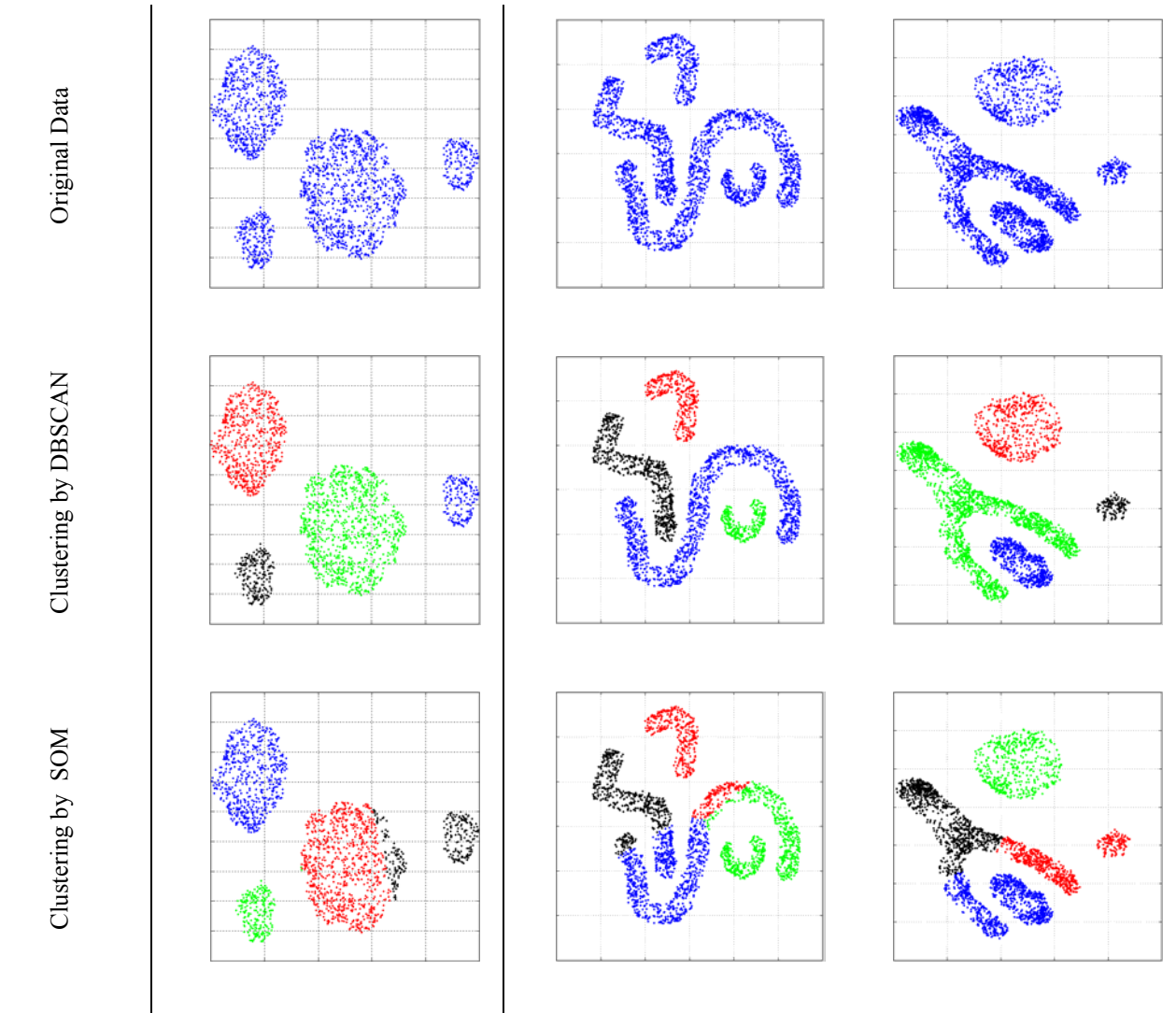


Fig 2: Spatial data used for testing

The figure 3 shows the properties of density based clustering characteristics of DBSCAN, SOM and k-means.



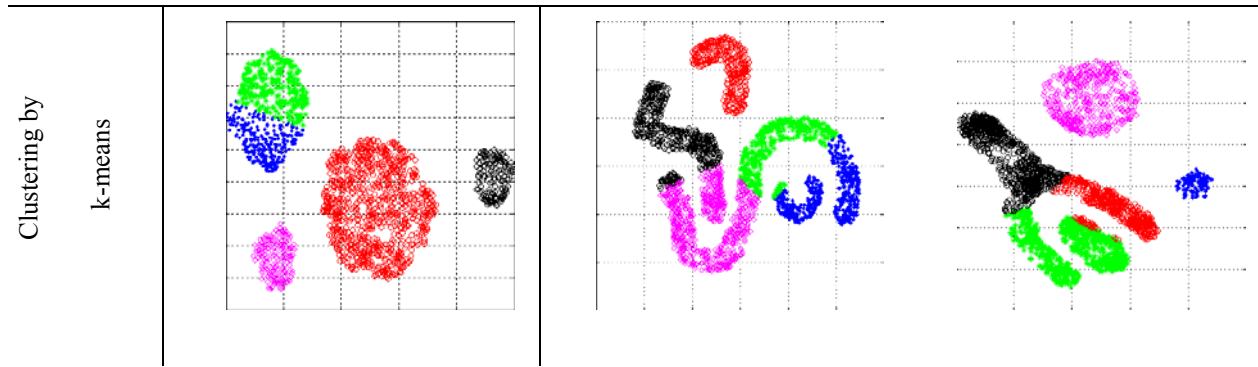


Fig 3 : Clustering characteristics of DBSCAN, SOM and k-means clustering algorithms

From the plotted results, it is noted that DBSCAN performs better for spatial data sets and produces the correct set of clusters compared to SOM and k-means algorithms. DBSCAN responds well to spatial data sets.

Conclusion

The Clustering algorithms are attractive for the task of class identification in spatial databases. This paper evaluated the efficiency of clustering algorithms namely DBSCAN, k-means and SOM for a synthetic, two dimensional spatial data sets. The implementation was carried out using MATLAB 6.5. Among the three algorithms DBSCAN responds well to the spatial data sets and produces the same set of clusters as the original data.

References

- [1] Ankerst M., Markus M. B., Kriegel H., Sander J(1999), "OPTICS: Ordering Points To Identify the Clustering Structure", Proc.ACM SIGMOD'99 Int. Conf. On Management of Data, Philadelphia, PA, pp.49-60.
- [2] Ester M., Kriegel H., Sander J., Xiaowei Xu (1996), "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise", KDD'96, Portland, OR, pp.226-231.
- [3] Guha S, Rastogi R, Shim K (1998), "CURE: An efficient clustering algorithm for large databases", In: SIGMOD Conference, pp.73~84.
- [4] Kaufman L. and Rousseeuw P. J (1990), "Finding Groups in Data: An Introduction to Cluster Analysis", John Wiley & Sons.
- [5] Rakesh A., Johanners G., Dimitrios G., Prabhakar R(1999), "Automatic subspace clustering of high dimensional data for data mining applications", In: Proc. of the ACM SIGMOD, pp.94~105.
- [6] Raymond T. Ng and Jiawei Han (2002), "CLARANS: A Method for Clustering Objects for Spatial Data Mining", IEEE Transactions on Knowledge and Data Engineering, Vol. 14, No. 5.
- [7] Sivanandam S N and Paulraj M (2004), "Introduction to Artificial Neural Networks", Vikas Publishing House Pvt Ltd, pp. 8 – 95.
- [8] Sivanandam S N, Sumathi S and Deepa S N (2008), "Introduction to Neural Networks using MATLAB 6.0", Tata McGraw-Hill Publishing company Limited, New Delhi, pp. 21- 223.
- [9] Wang W., Yang J., Muntz R(1997), "STING: A statistical information grid approach to spatial data mining", In: Proc. of the 23rd VLDB Conf. Athens, pp.186~195.
- [10] Zhang T, Ramakrishnan R., Livny M (1996), "BIRCH: An efficient data clustering method for very large databases", In: SIGMOD Conference, pp.103~114.