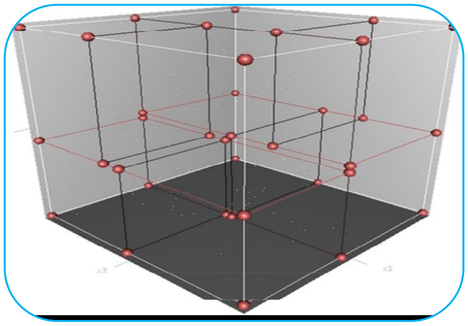
INDIAN STREAMS RESEARCH JOURNAL

ISSN: 2230-7850
IMPACT FACTOR : 5.1651 (UIF)
VOLUME - 10 | ISSUE - 11 | DECEMBER - 2020

MULTI DIMENSIONAL SPATIAL DATA

Prof. Santosh Kulkarni
Assistant Professor,
Prin. K.P. Mangalvedhekar Institute of Management,
Career Development & Research , Solapur.

ABSTRACT



Spatial data mining is the process of extracting knowledge from large amounts of spatial data. It turns into a profoundly requesting field on the grounds that tremendous measures of spatial information have been gathered in different applications going from geo-spatial information to bio-clinical information. The amount of spatial data that is being gathered is growing at an exponential rate. As a result, it far surpassed human analysis capabilities. Clustering has recently been recognized as one of the most important data mining techniques for knowledge discovery in spatial databases. In the past few years, new clustering algorithms have been proposed and the development of clustering algorithms has received a lot of attention. A pioneering density-based clustering algorithm is DBSCAN. From a large amount of data with noise and outliers, it can identify clusters of varying sizes and shapes. Using synthetic two-dimensional spatial data sets, this paper demonstrates the outcomes of analyzing the density based clustering characteristics of three clustering algorithms: DBSCAN, k-means, and SOM.

KEYWORDS : *Clustering, DBSCAN, k-means, SOM.*

INTRODUCTION

Clustering is one of the most important data mining techniques, and researchers are actively researching it. The target of bunching is to segment a bunch of items into groups to such an extent that articles inside a gathering are more like each other than designs in various bunches. K-MEANS [4], CLARANS [6], BIRCH [10], CURE [3], DBSCAN [2], OPTICS [1], STING [9], and CLIQUE [5] are just a few of the useful clustering algorithms that have been developed so far for large databases. There are many different types of these algorithms. Hierarchical, density-based, and partitioning are the three most common types. The clustering issues associated with handling massive amounts of data in large databases are the subject of all of these algorithms. Nevertheless, none of them are the most successful.

A cluster is defined as a high-density region divided by low-density regions in data space in density-based clustering algorithms, which are made to find clusters of any shape in databases with noise. A typical density-based clustering algorithm is Density Based Spatial Clustering of Applications with Noise (DBSCAN) [2]. Density-based clustering characteristics of three clustering algorithms—DBSCAN, k-means, and SOM—are examined in this paper.

Explanation of DBSCAN Steps

- There are two requirements for DBSCAN: minimum points (minPts) and epsilon (eps). It begins at a random location that has not been visited. After that, it locates all neighbor points that are within eps of the starting point.
- A cluster is formed if the number of neighbors is greater than or equal to minPts. This cluster includes the starting point and its neighbors, and the starting point is marked as visited. The evaluation procedure is then repeated recursively by the algorithm for each neighbor.
- The point is labeled as noise if the number of neighbors is less than minPts.
- The algorithm iterates through the dataset's remaining unvisited points if a cluster is fully expanded (every point within reach is visited).

Advantages

1. Unlike k-means, DBSCAN does not require you to know the number of clusters in the data beforehand.
2. Clusters can be found in any shape using DBSCAN. It can even find clusters that are completely surrounded by another cluster but not connected to it. The so-called single-link effect, in which multiple clusters are linked by a thin line of points, is lessened by the MinPts parameter.
3. DBSCAN has an idea of what noise is.
4. DBSCAN is largely insensitive to the database's point order and only requires two parameters.

Disadvantages

1. The distance measure used by DBSCAN in the function getNeighbors(P,epsilon) determines how well it can cluster the data. Euclidean distance is the most widely used distance metric. This distance metric can be rendered virtually useless, particularly for data with high dimensionality.
2. DBSCAN doesn't answer well to informational indexes with shifting densities (called various leveled informational collections) .

k-means Algorithm

The dataset is divided up into "k" subsets by the naive k-means algorithm, where each record in a given subset "belongs" to the same center. Likewise the focuses in a given subset are nearer to that middle than to some other focus.

Simple iterations are used by the algorithm to track the subsets' centroids. The initial partitioning is generated at random, or we initialize the centroids at random to specific points in the space. Following two very straightforward steps, a new set of centroids is generated from the existing set of centroids in each iteration step. Let's use C(i) to denote the set of centroids following the sixth iteration. The steps carry out the following operations:

(I) Segment the focuses in view of the centroids C(i), that is, track down the centroids to which each of the focuses in the dataset has a place. The Euclidean distance between the centroids is used to divide the points.

(II) The new location of the old centroid in a particular partition is referred to as the new location of the old centroid. (i) Set a new centroid to be the mean of all points that are closest to c(i) C (i+1).

When recompiling the partitions does not alter the partitioning, the algorithm is said to have converged. When C(i) and C(i – 1) are identical, the algorithm has converged completely, according to our terminology. For designs where no point is equidistant to more than one community, the above intermingling condition can continuously be reached. The kmeans algorithm is attractive because of its simplicity and convergence property.

Numerous "nearest-neighbor" queries for the points in the dataset are required by the k-means. O(kdN) is the cost of one iteration for data with d dimensions and N points in the dataset. The naive k-means algorithm is generally not practical for a large number of points because it would require multiple iterations.

Occasionally, it takes several iterations for the centroids to converge, with C(i) and C(i+1) being identical. The centroids also move very little over the course of the last few iterations. We need a measure of centroids' convergence in order to stop the iterations when the convergence criteria are met—running the expensive iterations so many times might not be efficient. The most widely accepted method is distortion.

SOM Algorithm

A neural network approach known as a self-organizing map (SOM) or self-organizing feature map (SOFM) makes use of competitive unsupervised learning. The idea that a node's behavior should only affect the nodes and arcs that are close to it is the foundation of learning. In the beginning, weights are assigned at random, which is changed during the learning process to get better results. The data's hidden features or patterns are discovered during this learning process, and the weights are adjusted accordingly. Teuvo Kohonen, a professor from Finland, was the first to describe the model, which is why it is sometimes called a Kohonen map.

The output syntaxes in the self-organizing map are arranged in a low dimensional (typically 2D or 3D) grid, and it is a feed forward network with only one layer. All output neurons are connected to each input. Every neuron carries a weight vector with the same dimensionality as the input vectors. The self-organizing map's learning process aims to associate various components of the SOM lattice with similar responses to particular input patterns.

Training of SOM

Weights and learning rates are initially set. The network is shown the input vectors that need to be clustered. The winner unit is determined using either the Euclidean distance method or the sum of products method after the input vectors are provided and the initial weights are taken into account.

The weights for that particular winner unit are updated based on the winner unit selection. Once the network has received all of the input vectors, an epoch is said to have ended. Multiple training epochs can be performed by updating the learning rate.

CONCLUSION

For class identification in spatial databases, the Clustering algorithms are appealing. DBSCAN, k-means, and SOM were the three clustering algorithms examined in this study for their efficacy on fake, two-dimensional spatial data sets. MATLAB version 6.5 was used to carry out the implementation. DBSCAN is the best of the three algorithms when it comes to spatial data sets and produces clusters that are identical to 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.