ISSN:0975-9646

Combination of Density Based and Partition Based Clustering Algorithm-DBK Means

B.L. Krishna, P.Jhansi Lakshmi , P.Satya Prakash
Department Of Computer Science , Vignan University,
Vadlamudi, Guntur, India

Abstract- **Mining or extracting the knowledge from the large amount of data is known as data mining. Here, the collection of data increases exponentially so that for extracting the efficient data we need good methods in data mining. Data mining analyzes several methods for extracting the data. Clustering is one of the methods for extracting the data from large amount of data. Multiple clustering algorithms were developed for clustering. Multiple clustering can be combined so that the final partitioning of data provides better clustering. Efficient density based k-means clustering algorithm has been proposed to overcome the drawbacks of dbscan and k-means clustering algorithms. The algorithm performs better than dbscan while handling clusters of circularly distributed data points and slightly overlapped clustering.**

Index Terms : clustering analysis, k-means, and dbscan.

I. INTRODUCTION

Cluster analysis divides data into meaningful or useful groups (clusters). If meaningful clusters are the goal, then the resulting clusters should capture the “natural” structure of the data. For example, cluster analysis has been used to group related documents for browsing, to find genes and proteins that have similar functionality, and to provide a grouping of spatial locations prone to earthquakes. However, in other cases, cluster analysis is only a useful starting point for other purposes, e.g., data compression or efficiently finding the nearest neighbors of points. Whether for understanding or utility, cluster analysis has long been used in a wide variety of fields: psychology and other social sciences, biology, statistics, pattern recognition, information retrieval, machine learning, and data mining.

What Cluster Analysis Is

Cluster analysis groups objects (observations, events) based on the information found in the data describing the objects or their relationships. The goal is that the objects in a group will be similar (or related) to one other and different from (or unrelated to) the objects in other groups. The greater the similarity (or homogeneity) within a group, and the greater the difference between groups, the “better” or more distinct the clustering.

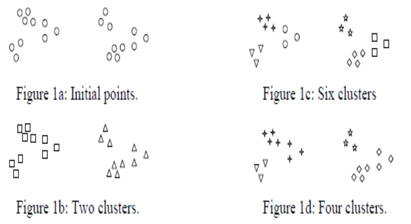


Figure 1 : examples of clustering analysis

DISTANCE MEASURES

The number of points in a dataset is denoted by N. Each point is denoted by Pi, Pj and so on. k denotes the number of clusters and d denotes the number of dimensions of a point. D denotes the set of dimensions and Dix, Djy represent the subsets of dimensions of the points Pi and Pj respectively. l,m and x are simply used as indices.

EUCLIDEAN DISTANCE

An N x N matrix Me is calculated. For points with d dimensions, the Euclidean distance Me(Pi, Pj) between two points Pi and Pj is defined as follows:

$$M_e(p_i, p_j) = \sqrt{\sum_{z=1}^{d}(\mathbf{piz} - \mathbf{pjz})}^2$$

Where Pixand Pjxrepresent the xth dimension values of Pi and Pj respectively. Also, Me is a symmetric matrix.

II. RELATED WORK

K-MEANS :

The term "k-means" was first used by James Mac Queen in 1967, though the idea goes back to Hugo Steinhaus in 1956. The standard algorithm was first proposed by Stuart Lloyd in 1957 as a technique for pulse-code modulation, though it wasn't published until 1982.

In statistics and machine learning, k-means clustering is a method of cluster analysis which aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean. It is similar to the expectation-maximization algorithm for mixtures of Gaussians in that they both attempt to find the centers of natural clusters in the data as well as in the iterative refinement approach employed by both algorithms.

The kmeans implementation is easy when compared to the other clustering algorithms. This simply involves the selection of the initial indices and the calculation of the distances to these indices to the every other point in the data base and there by forming the clusters. This is a continuous procedure and the procedure is repeated until the same means for all the clusters are repeated.

KMEANS ALGORITHM:

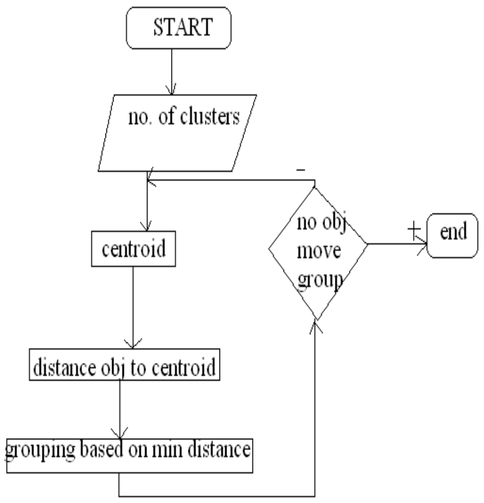
```
void cluster( Data D, clno k, Num n )
{
If( k <=n)
        {
```

```
        Randomly pick k-objects as initial clusters
          Repeat:
  {
            For all n objects calculate the distance and assign the clusters
            For obtained clusters calculate Means $M_i$ & update clusters $C_i$
            }
     Until:  no change in Means ( i=1 to k $M_i$==$M_i$-1)  ;
          }
      Else
         Clustering not possible
}
```

Where	K	-	No of clusters.
	D	-	Data set taken.
	N	-	Size of the data set (N by 2)
	M_i	-	Means to the clusters.
	C_i	-	Clusters formed.

From the algorithm it's clear that if that if the no of clusters required is greater than the size of the data it's not possible to cluster the data. The actual implementation is seen in the next section.

KMEANS DATA FLOW DIAGRAM



K MEANS ADVANTAGES

- With a large number of variables, K-Means may be computationally faster than hierarchical clustering (if K is small).
- K-Means may produce tighter clusters than hierarchical clustering, especially if the clusters are globular.

KMEANS DISADVANTAGES

- Difficulty in comparing quality of the clusters produced (e.g. for different initial partitions or values of K affect outcome).
- Fixed number of clusters can make it difficult to predict what K should be.
- Does not work well with non-globular clusters.
- Different initial partitions can result in different final clusters. It is helpful to rerun the program using the same as well as different K values, to compare the results achieved.

PROBLEMS WITH K-MEANS

- ➢ When the numbers of data are not so many, initialgrouping will determine the cluster significantly.
- ➢ The result is circular cluster shape because based on distance.
- ➢ The number of cluster, K, must be determined before hand. Selection of value of K is itself an issue and sometimes its hard to predict before hand the number of clusters that would be there in data.
- ➢ We never know the real cluster, using the same data, if it is inputted in a different order may produce different cluster if the number of data is few.
- ➢ Sensitive to initial condition. Different initial condition may produce different result of cluster.The algorithm may be trapped in the local optimum.
- ➢ We never know which attribute contributes more to the grouping process since we assume that each attribute has the same weight.

DBSCAN:

Dbscan (Density-Based Spatial Clustering of Applications with Noise) is a data clustering algorithm proposed
by MartinEster, HansPeterKriegel, JorgSander and Xiaoweiu in 1996. It is a density-based clustering algorithm because it finds a number of clusters starting from the estimated density distribution of corresponding nodes. DBSCAN is one of the most common clustering algorithms and also most cited in scientific literature. The main implementation of this algorithm depends up on the selection of the nearest neighbors. This is done based on the given density & the distance value. This is the finest algorithm used to find the outliers.

DENSITY REACHABILITY AND DENSITY CONNECTIVITY*:*

Density reachability is the first building block in dbscan. It defines whether two distance close points belong to the same cluster. Points p1 is density reachable from p2 if two conditions are satisfied: (i) the points are close enough to each other: distance (p1, p2) <e, (ii) there are enough of points in is neighborhood: |{ r : distance(r,p2)}|>m, where r is a database point. Density connectivity is the last building step of dbscan. Points p0 and pn are density connected, if there is a sequence of density reachable points p1,i2,...,i(n-1) from p0 to pn such that p(i+1) is density reachable from pi. A dbscan cluster is a set of all density connected points.

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 theremaining unvisited points in the dataset.

DBSCAN ALGORITHM:

```
        void dbscan(Data D, minpts min, eps ep)
{
        Repeat:
        If i==1 select as default index else Select randomly
$d_i$ as initial index
        Repeat:
             Get neighbours (di , min , ep) :
        For each neighbour: Mark as visited for that loop
                ::Get neighbours (di , min , ep) :
             (UNTIL) : no neighbour is obtained;
        if(  minpoints present )
        add to cluster else leave & mark as unvisited;
             Until all the nodes are visited;
}
```

Where D - Data set.
Min - Minimum points required to form a cluster.
Ep - Minimum distance required to select as a neighbor.

DBSCAN ADVANTAGES

1. DBSCAN does not require one to specify 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 a cluster 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. (However, points sitting on the edge of two different clusters might swap cluster membership if the ordering of the points is changed, and the cluster assignment is unique only up to isomorphism.)

DBSCAN DISADVANTAGES

- The quality of DBSCAN depends on the distance measure used in the function regionQuery(P,ε). The most common distance metric used isEuclidean distance. Especially for high-dimensional data, this metric can be rendered almost useless due to the so-called "Curse of dimensionality", making it difficult to find an appropriate value for ε. This effect, however, is also present in any other algorithm based on Euclidean distance.
- DBSCAN cannot cluster data sets well with large differences in densities, since the minPts-ε combination cannot then be chosen appropriately for all clusters.

III. PROBLEM IDENTIFICATION

Some type clusters are formed in below figure format by using dbscan.

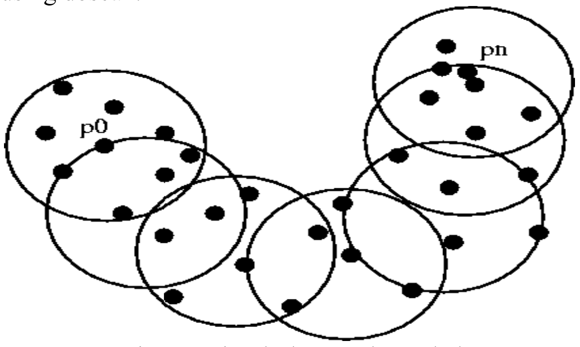


Figure2:circularly over lapped clusters

In order to avoid this type of circularly overlapped clusters in dbscan we have to follow the dbkmeans

IV. IMPLEMENTATION OF PROPOSED WORK DBKMEANS

The DBKMEANS combines the both kmeans and dbscan algorithms. In order to overcome the drawbacks of the both algorithms.

```
        Void DBKMEANS (clno k , min ep , dist ε, Dataset
D)
{
 c=0
for each unvisited point p in dataset D
{
N = getNeighbors (p, ε)
if (sizeof(N) < ep)
mark p as NOISE
else
++ c
mark p as visited
add p to cluster c
recurse (N)
}
Now will have m clusters
for each detected clusters
{
```

```
find the cluster centers Cmby taking the mean
find the total number of points in each cluster
}
If m>k
{
# Join two or more as follows
select two cluster based on density and
number of points satisfying the application criteria
and joint them and find the new cluster center and
repeat it until achieving k clusters.
Finally we will have Ck centers
}
else {
l =k-m
# split one or more as follows
if ( m >=l )
 {
select a cluster based on density and number of
points satisfying the application criteria and split it
using kmeans clustering algorithm and repeat it
until achieving k clusters.
Finally we will have Ck centers
}
```

Steps

- Initially we apply the dbscan then we have m clusters.
- Then again we apply the kmeans taking k as input.
- If k<m then the m is reduced to m by clubbing the clusters
- If k>m then the m is partitioned to k clusters.

CONCLUSION

In this project we have successfully implemented the DBKMEANS which is the improved version of the kmeans. In this project we have taken many considerations to improve the kmeans such as the taking the means as the initial means & taking the silhouette width etc. We got the good results by using the DBKMEANS than the kmeans & the dbscan. We also showed the difference using the visualization. Finally by using this algorithm we overcome the many disadvantages in the kmeans & some disadvantages in the dbscan.

ACKNOWLEDGEMENT

This research is supported by the vignan university, AndhraPradesh, india. Mr. Krishna kishore, HOD and assistance professor Mrs. P.jhanasi lakshmi .

REFERENCES

[1] Kaufman L. and Rousseeuw P. J., Finding Groups in Data: An Introduction to Cluster Analysis, John Wiley & Sons, 1990.
[2] Raymond T. Ng and Jiawei Han, CLARANS: A Method for Clustering Objects for Spatial Data Mining, IEEE TRANSACTIONS ON KNOWLEDGE and DATA ENGINEERING, Vol. 14, No. 5, SEPTEMBER 2002.
[3] Zhang T, Ramakrishnan R., Livny M., and BIRCH: An efficient data clustering method for very large databases, In: SIGMOD Conference, pp.103~114, 1996.
[4] Guha S, Rastogi R, Shim K., and CURE: An efficient clustering algorithm for large databases, In: SIGMOD Conference, pp.73~84, 1998.
[5] Ester M., Kriegel H., Sander J., Xiaowei Xu, A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise, KDD'96,Portland, OR, pp.226-231, 1996.
[6] Ankerst M., Markus M. B., Kriegel H., Sander J., OPTICS: Ordering Points To Identify the Clustering Structure, Proc.ACM SIGMOD'99 Int.Conf. On Management of Data, Philadelphia, PA, pp.49-60, 1999.
[7] Wang W., Yang J., Muntz R., STING: A statistical information grid approach to spatial data mining, In: Proc. of the 23rd VLDB Conf. Athens, pp.186~195, 1997.
[8] Rakesh A., Johanners G., Dimitrios G., Prabhakar R., Automatic subspace clustering of high imensional data for data mining applications, In: Proc. of the ACM SIGMOD, pp.94~105, 1999.
[9] http://en.wikipedia.org/wiki/DBSCAN
[10] http://en.wikipedia.org/wiki/K-means_clustering
[11] http://users.csc.calpoly.edu/~dekhtyar/560- Fall2009 /lectures /lec08.466.pdf
[12] http://www.mathworks.com/access/helpdesk/help/