**DOCUMENTATION**

**Implementation of clustering techniques on large data set**

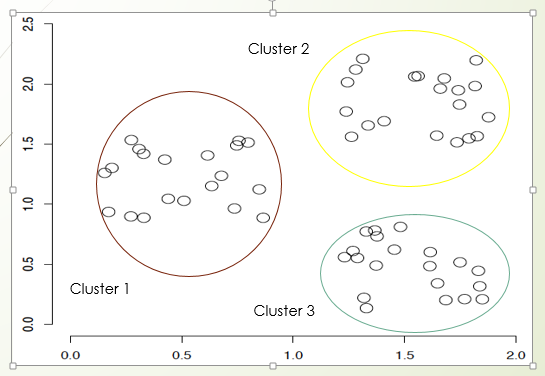
**Overview**

This project mainly focuses on brief description of clustering and its techniques DBSCAN and K-Means. It shows how both clustering techniques work, what are the factors or parameters needed to create clusters, how it classifies high density areas and low density areas and how it classifies various outliers (noise points). It also depicts comparison between DBSCAN and K-means on dataset of text document.

**Clustering**

Clustering is the process of grouping a set of data points into groups of similar characteristics. The main objective of clustering is to increase intra-group similarity and inter-group dissimilarity[9].

The clustering techniques are widely used in variety of applications like grouping customer of similar purchasing behavior so that they can be easily targeted for recommendation. Clustering is unsupervised machine learning technique because it automatically divides the data into clusters, or groups of similar items.



**Types of clustering techniques**

* Centroid Based Clustering
  + This is basically one of iterative clustering algorithm in which the clusters are formed by the closeness of data points to the *centroid* of clusters.
  + Here the cluster center i.e. *centroid* is formed such that the distance of data points is minimum with the center.
* Density Based Clustering
  + Density-based clustering algorithms makes the clusters based on density of data points in a region [11].
  + In this clustering model there will be a searching of data space for areas of varied density of data points in the data space.
  + It isolates various density regions based on different densities present in the data space [11].
* Hierarchical Based Clustering
  + These clustering is based on the notion that the data points closer in data space exhibit more similarity to each other than the data points lying farther away [10].
  + In the first approach, they start with classifying all data points into separate clusters & then aggregating them as the distance decreases. In the second approach, all data points are classified as a single cluster and then partitioned as the distance increases.

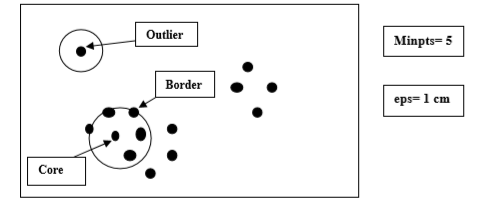
**DBSCAN**

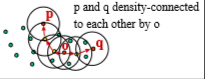
In DBSCAN, clustering is done on basis of density (population) of data points. Region with more number of data points is considered as existence of cluster and region with less number of data points is considered as noise cluster [1].

It requires two parameters to create cluster ie ε( radius) and minpts (minimum no. of points require to form cluster). Size of radius is to be defined. A point is to be taken as core point if sufficient number of points comes in to radius are to be involved in cluster and rest are the outliers[1].

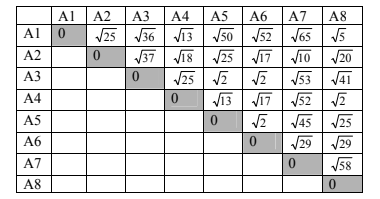
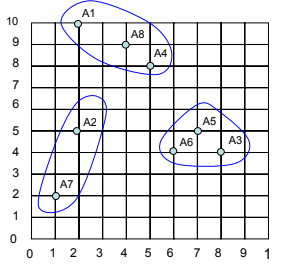
DBSCAN is used with dynamic database where the data may gets frequently updated. But when any data is inserted or deleted, DBSCAN rescans the database. Rescanning consumes lot of time which reduces efficiency. DBSCAN creates clusters of same densities. It can be used for the categorical data. The algorithm overcomes the impact of the inadequate of the memory when clustering the large scale data set and also accurately reflect the characteristics of the data set [6].

DBSCAN does not work well when dealing with clusters of varying densities and struggles with clusters of similar density. Sometimes Struggles with high dimensionality of data and discrete data. If the data points are in very scattered form then it gets difficult to form clusters by DBSCAN. (<https://www.youtube.com/watch?v=C3r7tGRe2eI>)





**Example of DBSCAN**

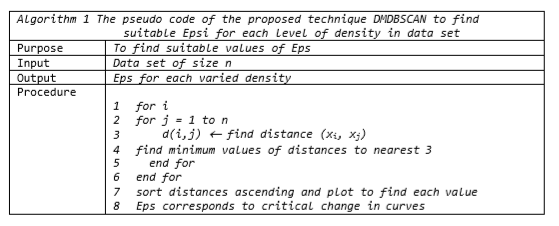
* There are 8 data points:
  + A1(2,10), A2(2,5), A3(8,4), A4(5,8), A5(7,5), A6(6,4), A7(1,2), A8(4,9)
* Epsilon(radius)=√10 and Minimum no. of points=2
* Calculate distance between every data points using Euclidean distance.
* Distance matrix based on Euclidean distance is given below.
* 
* A1, A8 and A4 will be in cluster **C1** , A5, A6, A3 will be in cluster **C2**, A2 and A7 will be in cluster **C3**.
* 

**Optimal value of epsilon**

Calculate the Euclidean distance considering one point at a time to its 2nd nearest point. Nearest point is that point itself.

Euclidean distance of each point to its second nearest point is sorted in ascending order and plot it into graph[7].

 In graph, the point where the slope changes or line gets curve is significant Epsilon optimal value[8].



        [8]

The value of minimum points depends on how many clusters need to be generated.

If you want to generate big clusters and less number of clusters then set minPts value high.

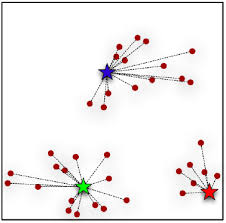
Too low value of minPts leads to generate more clusters from noise points. High or low value for minPts is relative and strongly depends on the size of the dataset. (<https://www.youtube.com/watch?v=TGad0nc-8gU>)

**K-Means**

the K-means algorithm identifies *k* number of centroids, and then allocates every data point to the nearest cluster, while keeping the centroids as small as possible. *‘means’* in the K-means refers to averaging of the data, which helps in finding the centroid.

K-means algorithm starts with a first group of randomly selected centroids, which are used as the beginning points for every cluster.Now new data point is to be entered in cluster on basis of distance between that point and centroids.

Whichever point has less distance with centroid, is to be entered in that particular cluster.After entering into the cluster, centroid of that cluster is changed by calculating mean of old point and new point recently added.

****

**REFERENCES**

[1] Sauravjyoti Sarmah , Dhruba K. Bhattacharyya,”An Effective Technique for Clustering Incremental Gene Expression data”, IJCSI International Journal of Computer Science Issues, Vol. 7, Issue 3, No 3, May 2010.

[2] Martin Ester,Hans-Peter Kriegel,Jorg Sander,Michael Wimmer,Xiaowei Xu,” Incremental clustering for mining in a data ware housing”, University of Munich Oettingenstr. 67, D-80538 München, Germany.

[3] K.M. Hammouda, M.S. Kamel, Efficient phrase-based document indexing for web document clustering, in: IEEE TransKnowledge and Data Eng., vol. 16, no. 10, Oct. 2004, pp. 1279–1296.

[4] M. Ester, H.P. Kriegel, J. Sander, M. Wimmer, X. Xu, Incremental clustering for mining in a data warehousing environment, in: Proceedings of the 24th VLDB Conference, Institute for Computer Science, University of Munich, Germany, New York, USA, Aug 1998.

[5] Glory H. Shah, C. K. Bhensdadia, Amit P. Ganatra , “An Empirical Evaluation of Density-Based Clustering Techniques”, International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-2, Issue-1, March 2012

[6] Xiaoke Su, Yang Lan, Renxia Wan, and Yuming “A Fast Incremental Clustering Algorithm”, international Symposium on Information Processing (ISIP’09), Huangshan, P.R.China, August-21-23,pp:175-178,2009.

[7] Gaonkar MN, Sawant K. 2013. AutoEps DBSCAN: DBSCAN with Eps automatic for large dataset. International Journal on Advanced Computer Theory and Engineering.

[8] Elbatta MNT. 2012. An improvement for  DBSCAN algorithm for best results in varied densities [disertasi]. Gaza (PS): Islamic University of Gaza.

[9] VikramPudi, P. Radha Krishna “Data Mining” Oxford University Press 2009.

[10] B. Rama et. Al., “A Survey on clustering Current Status and challenging issues”(IJCSE) International Journal on Computer Science and Engineering Vol. 02, No. 9, pp. 2976-2980, 2010.

[11] Stefan Brecheisen, Hans-Peter Kriegel, and Martin Pfeifleisen , Multi-Step Density-Based Clustering , Knowledge and information system (KAIS), Vol. 9 , No. 3 , 2006.

.