International Journal of Engineering, Management, Humanities and Social Sciences Paradigms (IJEMHS)
Volume 30, Issue 02, Quarter 02 (April to June 2018)
An Indexed and Referred Journal with Impact Factor: 2.75
ISSN (online): 2347-601x
www.ijemhs.com

# Enhancement of Clustering Mechanism

**Yashi Shrivastava[1] and Tarun Dalal[2]**

**[1]Department of Computer Science and Engineering, CBS Group of Institution, Jhajjar, Haryana (India)**
**[2]Assistant Professor, Department of Computer Science and Engineering, CBS Group of Institution, Jhajjar, Haryana (India)**

**Publishing Date: June 19, 2018**

**Abstract**

Cluster analysis or clustering is task of grouping a set of objects within in such a way that objects within same group (called a cluster) are more similar (in some sense or another) to each other than to those within other groups (clusters). It is a main task of exploratory data mining, & a common technique for statistical data analysis, used within many fields, including machine learning, pattern recognition, image analysis, information retrieval, bioinformatics, data compression, & computer graphics. Cluster is not one special algorithm, but commonly works to be solved. It could be achieved by various algorithms that differ significantly within their notion of what constitutes a cluster & how to efficiently find them. Popular notions of clusters include groups within short distances large cluster set, dense areas of data space, intervals or particular statistical distributions. Clustering could therefore be formulated as a multi-objective optimization problem. Appropriate clustering algorithm depends on individual data set & intended use of results. Cluster analysis as such is not an automatic task, but an iterative process of knowledge discovery or interactive multi-objective optimization that involves trial & failure. It is arrogant to update data pre procedure & model parameters until result achieves desired properties.

***Keywords:*** *Clustering Mechanism, Hierarchical Clustering, Density-reachability.*

## I. Introduction

Big data is very huge data sets that could be analysed computationally in order to check trends, patterns and associations. This pattern could be relating to human behaviour & interactions. Lot of IT investment is interested in managing & maintaining big data. This research deals with such huge data set or big data as research objective is to enhance the clustering mechanisms. It is very essential to use huge data set in order to make analysis of performance of enhanced clustering mechanism. Discussion is made to introduction of Data mining ,Classification of Data Mining System, Data Warehouse, Big Database in Data Mining, Big Data, Clustering, Application of Cluster Analysis, Requirement of Clustering in Data Mining, Clustering. This chapter has also explained the objective of research.

## II. Clustering

Clustering is a way of finding structures from a collection of unlabelled gene expression data. A number of algorithms are developed to tackle problem of clustering gene expression data. It is important for solving problems that originate due to unsupervised learning. This paper presents a performance analysis on various clustering algorithm namely K-means, expectation maximization, & density based clustering in order to identify best clustering algorithm for microarray data.

## III. Requirements of Clustering

- **Scalability**: we need highly scalable clustering algorithms to deal with large databases.
- **Ability to deal with different kinds of attributes**: Algorithms should be capable to be applied on any kind of data such as interval-based (numerical) data, categorical, and binary data.
- **Discovery of clusters with attribute shape**: The clustering algorithm should be capable of detecting clusters of arbitrary shape. They should not be bounded to only distance measures that tend to find spherical cluster of small sizes.
- **High dimensionality**: The clustering algorithm should not only be able to handle low-dimensional data but also the high dimensional space.
- **Ability to deal with noisy data**: Databases contain noisy, missing or erroneous data. Some algorithms are sensitive to such data and may lead to poor quality clusters.
- **Interpretability**: The clustering results should be interpretable, comprehensible, and usable.

## IV. Clustering Techniques

Clustering algorithms can be categorized into

<partition-based algorithms,

<hierarchical-based algorithms,

<density-based algorithms and

<grid-based algorithms.

These methods vary in:

-the procedures used for measuring the similarity (within and between clusters)

-the use of thresholds in constructing clusters

-the manner of clustering, that is, whether they allow objects to belong to strictly to one cluster or can belong to more clusters in different degrees and the structure of the algorithm .

The different clustering techniques are stated as follows:

### 1) Partition Clustering

**a) K-Means**: K-Means clustering is a clustering method in which given data set is divided into K number of clusters.
Experimental results of K-means clustering &its performance in case of execution time are discussed here. But there are certain limitations in K-means clustering algorithm such as it takes more time for execution. So in order to reduce execution, time they are using Ranking Method &also shown that how clustering is performed in less execution time as compared to traditional method. This work makes an attempt at studying feasibility of K-means clustering algorithm in data mining using Ranking Method.

**K-Means Algorithm**

MSE=largenumber;
Select initial cluster centroids {mj}j
k=1;
Do
OldMSE=MSE;
MSE1=0;
For j=1 to k
mj=0; nj=0;
endfor
For i=1 to n
For j=1 to k
Compute squared Euclidean
distance d2(xi, mj);
endfor
Find closest centroid mj to xi;
mj=mj+xi; nj=nj+1;
MSE1=MSE1+d2(xi, mj);
endfor
For j=1 to k
nj=max(nj, 1); mj=mj/nj;
endfor
MSE=MSE1;
while (MSE<OldMSE)

**b) K-Medoids:** This algorithm is computationally demanding and may take some time to complete. Several options are available to the user, via the **options** button in the Statistics page:

- Maximum k-medoids clusters: the maximum value of k for which clusters are computed
- Minimum k-medoids clusters: the minimum value of k for which clusters are computed
- Number of reseeds: the number of times the algorithm is reseeded with randomly chosen initial centroids and the algorithm re-run.
- Number of simulations: the number of times simulated data is clustered, in order to adjust Silhouette Index values of cluster validity.

Output data are displayed in several ways:

- Global Silhouette Indices for the different values of k are displayed in the Summary text box. Higher values of the Global Silhouette Index indicate greater separation between clusters and/or cohesion within clusters, and may indicate a natural clustering solution.

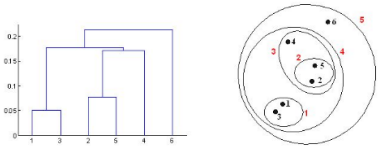
### 2) Hierarchial Clustering

Hierarchical *clustering* groups data into a multilevel cluster tree or dendrogram. If your data is hierarchical, this technique can help you choose the level of clustering that is most appropriate for your application.

International Journal of Engineering, Management, Humanities and Social Sciences Paradigms (IJEMHS)
Volume 30, Issue 02, Quarter 02 (April to June 2018)
An Indexed and Referred Journal with Impact Factor: 2.75
ISSN (online): 2347-601x
www.ijemhs.com

**Hierarchical Clustering**

- Produces a set of nested clusters organized as a hierarchical tree
- Can be visualized as a dendrogram
  - A tree like diagram that records the sequences of merges or splits



### a) Agglomerative

<♣ BIRCH ♣ CHAMELEON

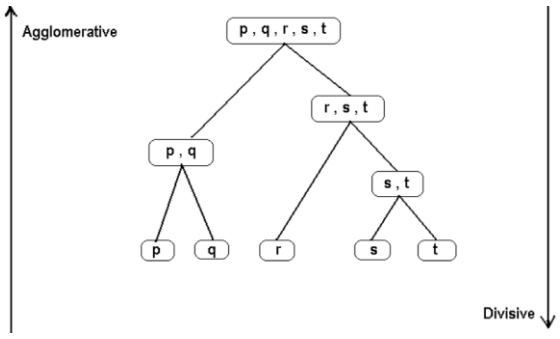
```
import numpy as np
import AgglomerativeClustering # Make sure to
use the new one!!!
d = np.array(
   [
      [1, 2, 3],
      [4, 5, 6],
      [7, 8, 9]
   ]
)

clustering =
AgglomerativeClustering(n_clusters=2,
compute_full_tree=True,
   affinity='euclidean', linkage='complete')
clustering.fit(d)
print clustering.distance
```

That example has the following output:

```
[ 5.19615242  10.39230485]
```

### b) Divisive




### 3) Density-Based Clustering

Density-based spatial clustering of applications with noise (DBSCAN)[1] is a density-based clustering algorithm. It gives a set of points in some space, it groups together points that are closely packed together (points with many nearby neighbors), marking as outliers points that lie alone in low-density regions. In 2014, the algorithm was awarded the test of time award at the leading data mining conference, KDD.

**Density Definition**

- ε (eps) Neighborhood – Objects within a radius of ε from an object
- "High density" - ε-Neighborhood of an object contains at least MinPts of objects

$$N_\varepsilon(p) : \{q \mid d(p,q) \le \varepsilon\}$$

**Core, Border & Outlier**

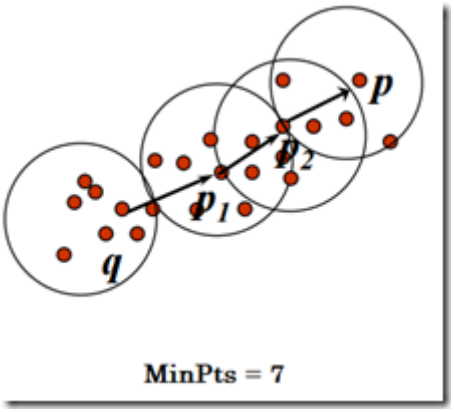
A point is a core point if it has more than a specified number of points (MinPts) within Eps—These are points that are at the interior of a cluster.

A border point has fewer than MinPts within Eps, but is in the neighbourhood of a core point.

**Density-Reachability**

Reachability is not a symmetric relation since, by definition, no point may be reachable from a non-core point, regardless of distance. Two points p and q are density-connected if there is a point o such that both p and q are density-reachable from o. Density-connectedness is symmetric.

Let check with example




– A point p is directly density-reachable from p2
– p2 is directly density-reachable from p1

– p1 is directly density-reachable from q
– p <- p2 <- p1 <- q form a chain

p is (indirectly) density-reachable from q
q is not density-reachable from p

**a) DBSCAN**

Density = number of points within a specified radius r (Eps)

A point is a core point if it has more than a specified number of points (MinPts) within Eps

These are points that are at the interior of a cluster

A border point has fewer than MinPts within Eps, but is in the neighborhood of a core point

A noise point is any point that is not a core point or a border point.

**b) DENCLUE**

1. Create a graph whose nodes are the points to be clustered
2. For each core-point c create an edge from c to every point p in the ε-neighborhood of c
3. Set N to the nodes of the graph;
4. If N does not contain any core points terminate
5. Pick a core point c in N
6. Let X be the set of nodes that can be reached from c by going forward;
   Create a cluster containing $X \cup \{c\}$
   $N=N/(X \cup \{c\})$
7. Continue with step 4

## 4) Grid Based Clustering :

Grid-Based Methods, the clustering methods discussed so far are data-driven – they partition the set of objects and adapt to the distribution of the objects in the embedding space. Alternatively, a grid-based clustering method takes a space-driven approach by partitioning the embedding space into cells independent of the distribution of the input objects. The grid-based clustering approach uses a multiresolution grid data structure. It quantizes the object space into a finite number of cells that form a grid structure on which all of the operations for clustering are performed. The main advantage of the approach is its fast processing time, which is typically independent of the number of data objects, yet dependent on only the number of cells in each dimension in the quantized space. In this section, we illustrate grid-based clustering using three typical examples. STING (Section 10.5.1) explores statistical information stored in the grid cells. CLIQUE (Section 10.5.2) represents a grid- and density-based approach for subspace clustering in a high-dimensional data space.

**a) STING**

The algorithm is given below:
1. Determine a layer to begin with.
2. For each cell of this layer, we calculate the confidence interval (or estimated range) of probability that this cell is relevant to the query. 3. From the interval calculated above, we label the cell as relevant or not relevant.
4. If this layer is the bottom layer, go to Step 6; otherwise, go to Step 5.
5. We go down the hierarchy structure by one level. Go to Step 2 for those cells that form the relevant cells of the higher level layer.
6. If the specification of the query is met, go to Step 8; otherwise, go to Step 7.
7. Retrieve those data fall into the relevant cells and do further processing. Return the result that meets the requirement of the query. Go to Step 9. 8. Find the regions of relevant cells. Return those regions that meet the requirement of the query. Go to Step 9.
9. Stop [3]

**b) Clique**

The clustering process in CLIQUE involves: 1. CLIQUE partitions the d- dimensional data space into non-overlapping rectangular units called grids according to the given grid size and then find out the dense region according to a given threshold value. A unit is dense if the data points in this are exceeding the threshold value. 2. Clusters are generated from the all dense subspaces by using the apriori approach . CLIQUE algorithm generates minimal description for the clusters obtained by first determining the maximal dense regions in the subspaces and then minimal cover for each cluster from that maximal region. It repeats the same procedure until covered all the dimensions.

## V. Conclusion

A comparison of these four algorithms is given on basis of percentage of incorrectly classified instances. Performance of clustering method is measured by percentage of incorrectly classified instances. Farthest first clustering gives better performance compared to k means clustering, Density based clustering & filtered clustering. Also this algorithm's result is independent of number of cluster while k means algorithm result is highly dependent on number of cluster. Farthest first clustering though gives a fast analysis when taken an account of time domain, but makes comparatively high error rate.

## References

[1] G.J., Kaiser, W. J.: Wireless integrated network sensors. Communications of the ACM 43(5), 51–58 (2000)
[2] Ye, W., Heidemann, J., Estrin, D.: An energy-efficient mac protocol for wireless sensor networks. IEEE Infocom, 1567–1576 (June 2002)
[3] Cerpa, A., Estrin, D: Acent: Adaptive self-configuring sensor networks topologies. In: IEEE Infocom, pp. 1278–1287 (June 2002)
[4] Chen, B., Jamieson, K., Balakrishnan, H., Morris, R: SPAN: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks. ACM/IEEE Mobicom 85–96 (July 2001)
[5] Kawadia, V., Kumar, P.R.: Power control and clustering in ad hoc networks. IEEE Infocom, 459–469 (April 2003)