Manish Verma, Mauly Srivastava, Neha Chack, Atul Kumar Diswar, Nidhi Gupta / International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com
Vol. 2, Issue 3, May-Jun 2012, pp.1379-1384

A Comparative Study of Various Clustering Algorithms in Data Mining

Manish Verma, Mauly Srivastava, Neha Chack, Atul Kumar Diswar, Nidhi Gupta
GLNA Institute of Technology, Mathura

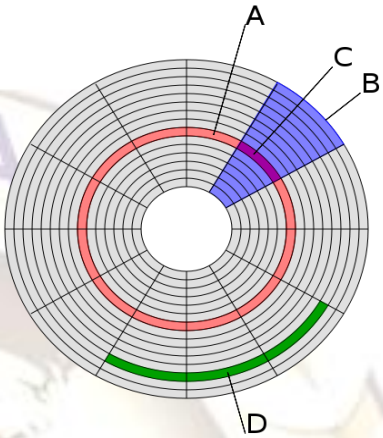
Abstract-Data clustering is a process of putting similar data into groups. A clustering algorithm partitions a data set into several groups such that the similarity within a group is larger than among groups.This paper reviews six types of clustering techniques- k-Means Clustering, Hierarchical Clustering, DBScan clustering, Density Based Clustering, Optics , EM Algorithm. These clustering techniques are implemented and analysed using a clustering tool WEKA.Performance of the 6 techniques are presented and compared.

Index Terms-Data clustering, K-Means Clustering, Hierarchical Clustering, DB Scan Clustering, Density Based Clustering, OPTICS, EM Algorithm

I. INTRODUCTION

CLUSTERING is a data mining technique to group the similar data into a cluster and dissimilar data into different clusters**.** Clustering can be considered the most important *unsupervised learning* technique so as every other problem of this kind, it deals with finding a *structure* in a collection of unlabeled data.Clustering is "the process of organizing objects into groups whose members are similar in some way". A *cluster* is therefore a collection of objects which are "similar" between them and are "dissimilar" to the objects belonging to other clusters. Clustering is the unsupervised classification of patterns (observations, data items, or feature vectors) into groups (clusters).
Data clustering is a process of putting similar data into groups. A clustering algorithm partitions a data set into several groups such that the similarity within a group is larger than among groups. Moreover, most of the data collected in many problems seem to have some inherent properties that lend themselves to natural groupings. Clustering algorithms are used extensively not only to organize and categorize data, but are also useful for data compression and model construction.

Finding these groupings or trying to categorize the data is not a simple task for or three dimensions at maximum.) Another reason for clustering is to discover relevance knowledge in data. Data cluster are created to meet specific requirements that cannot created using any of the categorical levels. One can combine data subjects as a temporary group to get a data cluster.



Disk structure:
(A)Track
(B) geometrical sector
(C) Track sector
(D) Cluster

The common approach of all the clustering techniques presented here is to find *cluster centers* that will represent each cluster. A cluster center is a way to tell where the heart of each cluster is located, so that later when presented with an input vector, the system can tell which cluster this vector belongs to by measuring a similarity metric between the input vector and all the cluster centers, and determining which cluster is the *nearest* or most similar one.

Some of the clustering techniques rely on knowing the number of clusters apriori. In that case the algorithm tries to partition the data into the given number of clusters. K-means and Fuzzy
C-means clustering are of that type.

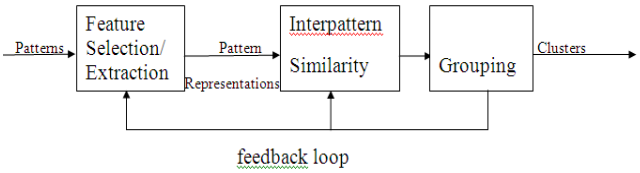


Figure:- Stages in clustering.

The *grouping* step can be performed in a number of ways. The output clustering (or clusterings) can be hard (a partition of the data into groups) or fuzzy (where each pattern has a variable degree of membership in each of the output clusters).

II. DATA CLUSTERING TECHNIQUES

In this section a detailed discussion of each technique is presented. Implementation and results are presented in the following sections.

A. K-means Clustering

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. The algorithm is called *k-means, where k is the number of clusters we want, since* a case is assigned to the cluster for which its distance to the cluster mean is the smallest. The action in the algorithm centers around finding the *k-means. We start out with an* initial set of means and classify cases based on their distances to the centers. Next, we compute the cluster means again, using the cases that are assigned to the cluster; then, we reclassify all cases based on the new set of means. We keep repeating this step until cluster means don't change much between successive steps. Finally, we calculate the means of the clusters once again and assign the cases to their permanent clusters.

Algorithmic steps for k-means clustering

Let $X = \{x_1,x_2,x_3,\ldots\ldots\ldots,x_n\}$ be the set of data points and $V = \{v_1,v_2,\ldots\ldots\ldots,v_c\}$ be the set of centers.

1) Randomly select *'c'* cluster centers.

2) Calculate the distance between each data point and cluster centers.

3) Assign the data point to the cluster center whose distance from the cluster center is minimum of all the cluster centers..

4) Recalculate the new cluster center using:

$$v_i = (1/c_i) \sum_{j=1}^{c_i} x_i$$

Where, *'c_i'* represents the number of data points in i^{th} cluster.

5) Recalculate the distance between each data point and new obtained cluster centers.

Distances: Quantitative Variables in K-Means

Identity (absolute) error

$$d_j(x_{ij}, x_{i'j}) = I(x_{ij} \neq x_{i'j})$$

Squared distance

$$d_j(x_{ij}, x_{i'j}) = (x_{ij} - x_{i'j})^2$$

L_q norms

$$L_{qii'} = [\sum_j |x_{ij} - x_{i'j}|^q]^{1/q}$$

Canberra distance

$$d_{ii'} = \sum_j \frac{|x_{ij} - x_{i'j}|}{|x_{ij} + x_{i'j}|}$$

Correlation

$$\rho(x_i, x_{i'}) = \frac{\sum_j (x_{ij} - \bar{x}_i)(x_{i'j} - \bar{x}_{i'})}{\sqrt{\sum_j (x_{ij} - \bar{x}_i)^2 \sum_j (x_{i'j} - \bar{x}_{i'})^2}}$$

6) If no data point was reassigned then stop, otherwise repeat from step 3

Variance clustering of the dataset into *k* clusters is that of finding *k* points {*mj*} (*j*=1,2,…,*k*) in **Rd** such that is minimized, where *d*(*xi*,*mj*)denotes the Euclidean distance between *xi* and *mj*. The points {*mj*} (*j*=1, 2… *k*) are known as cluster centroids.

- Ordinal variables can be forced to lie within (0, 1) and then a quantitative metric can be applied:
- For categorical variables, distances must be specified by user between each pair of categories.

B. Hierarchical clustering

HIERARCHICAL CLUSTERING builds a cluster hierarchy or, in other words, a tree of clusters, also known as a dendrogram. Every cluster node contains child clusters; sibling clusters partition the points covered by their common parent.

Agglomerative (bottom up)

1. Start with 1 point (singleton).
2. Recursively add two or more appropriate clusters.
3. Stop when k number of clusters is achieved.

Divisive (top down)

1. Start with a big cluster.
2. Recursively divides into smaller clusters.
3. Stop when k number of clusters is achieved.

General steps of Hierarchical Clustering:

Given a set of N items to be clustered, and an N*N distance (or similarity) matrix, the basic process of hierarchical clustering (defined by S.C. Johnson in 1967) is this:

- Start by assigning each item to a cluster, so that if we have N items, we now have N clusters, each containing just one item. Let the distances (similarities) between the clusters the same as the distances (similarities) between the items they contain.
- Find the closest (most similar) pair of clusters and merge them into a single cluster, so that now we have one cluster less.
- Compute distances (similarities) between the new cluster and each of the old clusters.
- Repeat steps 2 and 3 until all items are clustered into K number of clusters.

C. DB Scan Clustering

DBSCAN finds all clusters properly, independent of the size, shape, and location of clusters to each other, and is superior to a widely used Clarans method. DBscan is based on two main concepts: *density reachability* and *density connectability*. These both concepts depend on two input parameters of the dbscan clustering: *the size of epsilon neighborhood e* and *the minimum points in a cluster m*. The number of points parameter impacts detection of outliers. Points are declared to be outliers if there are few other points in the *e*-Euclidean neighborhood. *e* parameter controls the size of the neighborhood, as well as the size of the clusters. An open set in the Euclidean space can be divided into a set of its connected components. The implementation of this idea for partitioning of a finite set of points requires concepts of density, connectivity and boundary.

Density Functions:

Hinneburg & Keim [1998] shifted the emphasis from computing densities pinned to data points to computing density functions defined over the underlying attribute space. They proposed the algorithm DENCLUE (Density-based Clustering). Along with DBCLASD, it has a firm mathematical foundation. DENCLUE uses a density function

$$f^D(x) = \bullet \quad \underset{y \in D}{\bullet} f(x,y)$$

Crucial concepts of this section are density and connectivity both measured in terms of local distribution of nearest neighbors. The algorithm DBSCAN (Density Based Spatial Clustering of Applications with Noise) targeting low-dimensional spatial data is the major representative in this category. Two input parameters are used to define:

1) A, neighborhood

$$N_\varepsilon(x) = \{y \in X \mid d(x,y) \le \varepsilon\}$$

Of the point x,
2) A core object (a point with a neighborhood consisting of more than MinPts points)
3) A concept of a point y density-reachable from a core object x (a finite sequence of core objects between x and y exists such that each next belongs to an- neighborhood of its predecessor)
4) A density-connectivity of two points x, y (they should be density-reachable from a common core object).

D. OPTICS

OPTICS ("Ordering Points to Identify the Clustering Structure") is an algorithm for finding density-based clusters in spatial data. It was presented by Michael Ankerst, Markus M. Breunig, Hans-Peter Kriegel and Jörg Sander. Its basic idea is similar to DBSCAN, but it addresses one of DBSCAN's major weaknesses: the problem of detecting meaningful clusters in data of varying density. In order to do so, the points of the database are (linearly) ordered such that points which are spatially closest become neighbors in the ordering. Additionally, a special distance is stored for each point that represents the density that needs to be accepted for a cluster in order to have both points belong to the same cluster.

It is required to cut off the density of clusters that is no longer considered to be interesting and to speed up the algorithm this way.

The parameter ε is strictly speaking not necessary. It can be set to a maximum value. When a spatial index is available, it does however play a practical role when it comes to complexity. It is often claimed that OPTICS abstracts from DBSCAN by removing this each point is assigned a *core distance* that basically describes the distance to its $MinPts$th point:-

$$\text{core-distance}_{\varepsilon, MinPts}(p) = \begin{cases} \text{UNDEFINED} & \text{if } |N_\varepsilon(p)| < MinPts \\ \text{distance to the } MinPts\text{-th point} & \text{otherwise} \end{cases}$$

The *reachability-distance* of a point p from another point o is the distance between p and o, or the core distance of o:-

$$\text{reachability-distance}_{\varepsilon,MinPts}(p,o) = \begin{cases} \text{UNDEFINED} & \text{if } |N_\varepsilon(o)| < MinPts \\ \max(\text{core-distance}_{\varepsilon,MinPts}(o), \text{distance}(o,p)) & \text{otherwise} \end{cases}$$

Basically, if o and p are nearest neighbors, this is the we need to assume in order to have o and p belong to the same cluster.

Both the core-distance and the reachability-distance are undefined if no sufficiently dense cluster (w.r.t. ε) is available. Given a sufficiently large ε, this will never happen, but then every ε-neighborhood query will return the entire database, resulting in $O(n^2)$ runtime. Hence, the ε parameter, at least to the amount of only has to give a maximum value.

E. EM Algorithm

EM ALGORITHM is an iterative method for finding maximum likelihood or maximum a posteriori (MAP) estimates of parameters in statistical models, where the model depends on unobserved latent variables . The EM iteration alternates between performing an expectation (E) step, which computes the expectation of the log-likelihood evaluated using the current estimate for the parameters, and maximization (M) step, which computes parameters maximizing the expected log-likelihood found on the *E* step. These parameter-estimates are then used to determine the distribution of the latent variables in the next E step.

1. Expectation: Fix model and estimate missing labels.
2. Maximization: Fix missing labels (or a distribution over the
missing labels) and find the model that maximizes the expected log-likelihood of the data.

General EM Algorithm in English:

Alternate steps until model parameters don't change much:
E step:
Estimate distribution over labels given a certain fixed model.
M step:
Choose new parameters for model to maximize expected log-likelihood of observed data and hidden variables.

III. Comparison and Result

Above section involves the study of each of the six techniques introduced previously, and testing each one of them using **Weka Clustering** Tool on a set of banking data related to customer information. The whole data set consists of 11 attributes and 600 entries. Clustering of the data set is done with each of the clustering algorithm using Weka tool and the results are:

Comparison & Results using Weka Clustering Tool

Name	Number of clusters	Cluster Instances	Number of Iterations	Within clusters sum of squared errors	Time taken to build model	Log likelihood	Unclustered Instances
K-Means Algorithm	2	0:254(42%) 1:346(58%)	4	2016.675252093805 3	0.08 Seconds		0
EM Algorithm	6	0:31 (5%) 1:97 (16%) 2:65 (11%) 3:184(31%) 4:92(15%) 5:131(22%)			76.94 seconds	-21.09024	0
DBSCAN	3	0:10 (40%) 1:6 (24%) 2:9 (36%)			1.03 Seconds		575
Hierarchical Clustering	2	0:599(100%) 1:1 (0%)			1.16 Seconds		0
Density based Clusters	2	0:239(40%) 1:361(60%)	4	2016.675252093805 3	0.06 Seconds	-22.04211	0
OPTICS	0				1.37 seconds		600

IV. Conclusion

After analyzing the results of testing the algorithms and running them under different factors and situations, we can obtain the following conclusions:

- Performance of K-Means algorithm increases as the RMSE decreases and the RMSE decreases as the number of cluster increases.
- The performance of K-Means algorithm is better than Hierarchical Clustering algorithm.
- All the algorithms have some ambiguity in some (noisy) data when clustered.
- The quality of EM and K-Means algorithm become very good when using huge dataset.
- DBSCAN and OPTICS does not perform well on small datasets.
- K-Means and EM algorithm are very sensitive for noise in dataset. This noise makes it difficult for the algorithm to cluster data into suitable clusters, while affecting the result of the algorithm.
- K-Means algorithm is faster than other clustering algorithm and also produces quality clusters when using huge dataset.
- Hierarchical clustering algorithm is more sensitive for noisy data.
- Running the clustering algorithm using any software produces almost the same result even when changing any of the factors because most of the clustering software uses the same procedure in implementing any algorithm.

V. Future Work

In this paper we have intended to compare the pre-defined six algorithm and we have given some conclusions above.But still we were not able to cover all the factors for comparing these six algorithms.

As a future work,comparison between these algorithms(or may other algorithms) may be done using different parameters other than considered in this paper.

One important factor is normalization.Comparing between the results of algorithms using normalized data and non-normalized data will give different results. Of course normalization weill affect the performance of the algorithm and the quality of the results

VI. References

[1] Khalid Hammouda, Prof. Fakhreddine Karray,"A Comparative Study of Data Clustering Techniques" University of Waterloo, Ontario, Canada N2L 3G1

[2] Pavel Berkhin,"Survey of Clustering Data Mining Techniques", Accrue Software, Inc.

3] Tapan kanungo, senior member IEEE David M. Mount, member IEEE "An Efficient –means algorithm: analysis and implementation"

[4] A.K. JAIN, Michigan State University,M.N. MURTY, Indian Institute of Science AND P.J. FLYNN, the Ohio State UniversityData Clustering: A Review, ACM Computing Surveys, Vol. 31, No. 3, September 1999

[5] Glenn Fung,"A Comprehensive Overview of Basic Clustering Algorithms"., June 22, 2001

[6] OsamaAbuAbbas, Department of computer Science, Yarmouk University, Jordan,"Comparison Between Data Clustering Algorithm",The International Arab Journal Of InformationTechnology, vol.5, No.3, July 2008

[7] Chakraborty, S. and Nagwani, N.K., "Analysis and study of Incremental K-Means clustering Algorithm", Communication in Computer and Information Science, 1, Volume 169, High Performance Architecture and Grid Computing, Part 2, Pages 338-341, 2011.

[8] J.A.Hatigan and M.A.Wong,"a K-Means Clustering Algorithm, "Applied statistics, 28:100—108, 1979

[9] Azuaje, F., Dubitzky, W., Black, N., Adamson, K., "Discovering Relevance Knowledge in Data: A Growing Cell Structures Approach," IEEE Transactions on Systems, Man, and Cybernetics Part B: Cybernetics, Vol. 30, No. 3, June 2000 (pp.448)

[10] ALLEN, P. A. AND ALLEN, J. R. 1990. "Basin Analysis: Principles and Applications". Blackwell Scientific Publications, Inc., Cambridge, MA.

[11] BACKER, E. 1995. "Computer-Assisted Reasoning in Cluster Analysis". Prentice Hall International (UK) Ltd., Hertfordshire, UK.

[12] AGGARWAL, C.C., HINNEBURG, A., and KEIM, D.A. 2000. "On the surprising behavior of Distance metrics in high dimensional space". IBM Research report, RC 21739.

[13] ARSLAN, A.N. and EGECIOGLU, O. 2000. "Efficient algorithms for normalized edit Distance". Journal of Discrete Algorithms, 1, 1.

[14] Isabelle Guyon, Ulrike von Luxburg, and Robert C. Williamson. "Clustering: Science or art?" NIPS 2009 Workshop on Clustering Theory, Vancouver, Canada, 2009

[15] Berry, M.J.A. & Linoff, G.S.," Mastering Data Mining", New York, NY: Wiley Computer Publishing,200