International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Impact Factor (2012): 3.358

A Study on Clustering Techniques on Matlab

A.M Nirmala[1], Dr. S. Saravanan [2]

[1]Assistant Professor, Department of Computer Science, Muthayammal Arts & Science College, Rasipuram, Tamilnadu, India

[2]Professor & Head, Department of EEE, Muthayammal Engineering College, Rasipuram, Tamilnadu, India

Abstract: ***Clustering is the process of grouping similar object from the large dataset. It helps to arranging data into its logical group based on an attribute or a set of attributes. Clustering is the subject of active research in several fields such as statistics, pattern recognition, and machine learning. This survey focuses on different methods on clustering. The technique adds to clustering the complications of very large datasets with very many attributes of different types. This imposes unique computational requirements on relevant clustering algorithms. A variety of algorithms have recently emerged that meet these requirements and were successfully applied to real-life data mining problems. This paper is going to explore a variety of clustering methods and brief their working styles. The different techniques discussed here are just a snap shot of clustering algorithms. The Partitional clustering algorithms are have been used to develop clustering methods like K-Means, Clara, Clarans and implemented using Matlab environment.***

Keyword: Clustering, Partitional clustering, Hierarchical clustering, Matlab, K-Means

1. Introduction

A. Clustering

Clustering is a technique [12] to group together a set of items having similar characteristics. Clustering and classification are both fundamental tasks in Data Mining. Classification is used mostly as a supervised learning method, clustering for unsupervised learning (some clustering models are for both). The goal of clustering is descriptive, that of classification is predictive (Veyssieres and Plant, 1998). Since the goal of clustering is to discover a new set of categories, the new groups are of interest in themselves, and their assessment is intrinsic. In classification tasks, however, an important part of the assessment is extrinsic, since the groups must reflect some reference set of classes. *"Understanding our world requires conceptualizing the similarities and differences between the entities that compose it"* (Tyron and Bailey, 1970). Clustering can be considered the most important *unsupervised learning* problem, so, as every other problem of this kind, it deals with finding a *structure* in collection of unlabeled data.

A loose definition of clustering could be "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.

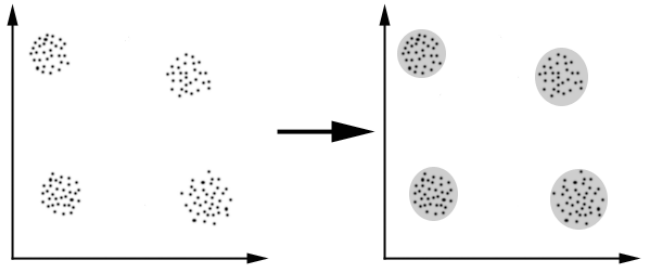

Figure 1: Clustering

B. Goals of Clustering

The goal of clustering is to determine the intrinsic grouping in a set of unlabeled data. It can be shown that there is no absolute "best" criterion which would be independent of the final aim of the clustering. For instance, we could be interested in finding representatives for homogeneous groups (data reduction), in finding "natural clusters" and describe their unknown properties ("natural" data types), in finding useful and suitable groupings ("useful" data classes) or in finding unusual data objects (outlier detection). The main requirements that a clustering algorithm should satisfy are:

- Scalability
- Dealing with different types of attributes
- Discovering clusters with arbitrary shape
- Minimal requirements for domain knowledge to determine input parameters
- Ability to deal with noise and outliers;
- Insensitivity to order of input records;
- High dimensionality
- Interpretability and usability

There are a number of problems with clustering:

- Current clustering techniques do not address all the requirements adequately (and concurrently);
- Dealing with large number of dimensions and large number of data items can be problematic because of time complexity;
- The effectiveness of the method depends on the definition of "distance" (for distance-based clustering);
- If an obvious distance measure doesn't exist we must "define" it, which is not always easy, especially in multi-dimensional spaces;

The result of the clustering algorithm (that in many cases can be arbitrary itself) can be interpreted in different ways.

2. Clustering Techniques

A large number of techniques have been proposed for forming clusters from distance matrices. The most important types are hierarchical techniques, optimization techniques and mixture models. We discuss the first two types here. We will discuss mixture models in a separate note that includes their use in classification and regression as well as clustering.

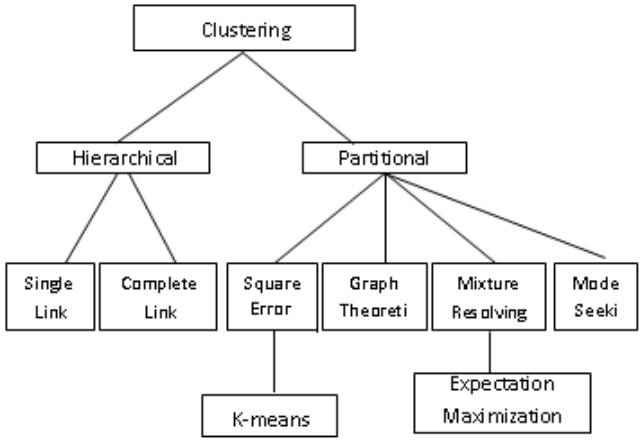

Figure 1: Types of clustering methods

Clustering algorithms can be divided into two broad classes:

- **Hierarchical approaches:** We begin assuming that each point is a cluster by itself. We repeatedly merge nearby clusters, using some measure of how close two clusters are (e.g., distance between their centroids), or how good a cluster the resulting group would be (e.g., the average distance of points in the cluster from the resulting centroid).
- **Centroid approaches:** We guess the centroids or central point in each cluster, and assign points to the cluster of their nearest centroid.

2.1 Hierarchical Clustering Algorithms

A Hierarchical clustering [10] algorithm yields a dendogram, representing the nested grouping of patterns and similarity levels at which groupings change. The dendogram can be broken at different levels to yield different clustering of the data. Most hierarchical clustering algorithms are variants of the single-link, complete-link, and minimum-variance algorithms. The single-link and complete link algorithms are most popular. These two algorithms differ in the way they characterize the similarity between a pair of clusters. In the single-link method, the distance between two clusters is the minimum of the distances between all pairs of patterns drawn from the two clusters (one pattern from the first cluster, the other from the second). In the complete-link algorithm, the distance between two clusters is the minimum of all pair wise distances between patterns in the two clusters. In either case, two clusters are merged to form a larger cluster based on minimum distance criteria. The complete-link algorithm produces tightly bound or compact clusters. The single-link algorithm, by contrast, suffers from a chaining effect. It has a tendency to produce clusters that are straggly or elongated. The clusters obtained by the complete link algorithm are more compact than those obtained by the single-link algorithm.

This algorithm is an agglomerative algorithm1 [11] that has several variations depending on the metric used to measure the distances among the clusters. The Euclidean distance is usually used for individual points. There are no known criteria of which clustering distance should be used, and it seems to depend strongly on the dataset. Among the most used variations of the hierarchical clustering based on different distance measures.

1. Average linkage clustering
The similarity between clusters is calculated using average values. The average distance is calculated from the distance between each point in a cluster and all other points in another cluster. The two clusters with the lowest average distance are joined together to form the new cluster.

2. Complete linkage clustering (Maximum or Furthest-Neighbour Method)
The similarity between 2 groups is equal to the greatest similarity between a member of cluster i and a member of cluster j. This method tends to produce very tight clusters of similar cases.

3. Single linkage clustering (Minimum or Nearest-Neighbour Method)
The similarity between 2 clusters is the minimum similarity between members of the two clusters. This method produces long chains which form loose, straggly clusters.

4. Ward's Method
Cluster membership is assigned by calculating the total sum of squared deviations from the mean of a cluster. The criterion for fusion is that it should produce the smallest possible increase in the error sum of squares.

2.2 Density-Based Methods

Density-based methods assume that the points that belong to each cluster are drawn from a specific probability distribution (Banfield and Raftery, 1993). The overall distribution of the data is assumed to be a mixture of several distributions. The aim of these methods is to identify the clusters and their distribution parameters. These methods are designed for discovering clusters of arbitrary shape which are not necessarily convex. The idea is to continue growing the given cluster as long as the density (number of objects or data points) in the neighbourhood exceeds some threshold. Namely, the neighbourhood of a given radius has to contain at least a minimum number of objects. When each cluster is characterized by local mode or maxima of the density function, these methods are called mode-seeking. Much work in this field has been based on the underlying assumption that the component densities are multivariate Gaussian (in case of numeric data) or multi nominal (in case of nominal data).

An acceptable solution in this case is to use the maximum likelihood principle. According to this principle, one should choose the clustering structure and parameters such that the probability of the data being generated by such clustering structure and parameters is maximized. The expectation maximization algorithm EM (Dempster *et al.*, 1977), which is a general-purpose maximum likelihood algorithm for

missing-data problems, has been applied to the problem of parameter estimation. This algorithm begins with an initial estimate of the parameter vector and then alternates between two steps (Farley and Raftery, 1998): an "E-step", in which the conditional expectation of the complete data likelihood given the observed data and the current parameter estimates is computed, and an "M-step", in which parameters that maximize the expected likelihood from the E-step are determined. This algorithm was shown to converge to a local maximum of the observed data likelihood. The *K*-means algorithm may be viewed as a degenerate EM algorithm, in which:

$$p(k/x) = \begin{cases} 1 & k = \underset{k}{\operatorname{argmax}}\{\hat{p}(k/x)\} \\ 0 & \text{otherwise} \end{cases}$$

Assigning instances to clusters in the *K*-means may be considered as the E-step; computing new cluster centers may be regarded as the M-step. The DBSCAN algorithm (density-based spatial clustering of applications with noise) discovers clusters of arbitrary shapes and is efficient for large spatial databases.

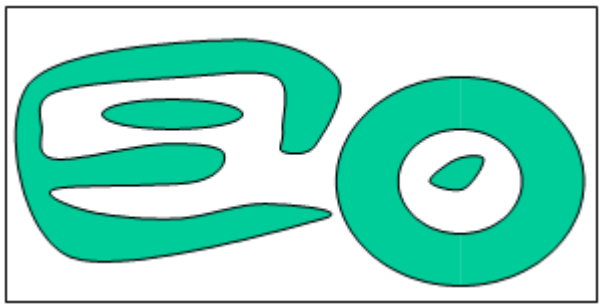

Figure 3: Clusters of arbitrary shape

The algorithm searches for clusters by searching the neighbourhood of each object in the database and checks if it contains more than the minimum number of objects (Ester *et al*., 1996). AUTOCLASS is a widely-used algorithm that covers a broad variety of distributions, including Gaussian, Bernoulli, Poisson, and log-normal distributions (Cheeseman and Stutz, 1996). Other well-known density-based methods include: SNOB (Wallace and Dowe, 1994) and MCLUST (Farley and Raftery, 1998).Density-based clustering may also employ nonparametric methods, such as searching for bins with large counts in a multidimensional histogram of the input instance space (Jain *et al*., 1999).

2.3 Model-based Clustering Methods [3]

These methods attempt to optimize the fit between the given data and some mathematical models. Unlike conventional clustering, which identifies groups of objects; model-based clustering methods also find characteristic descriptions for each group, where each group represents a concept or class. The most frequently used induction methods are decision trees and neural networks.

1. Decision Trees.
In decision trees [1], the data is represented by a hierarchical tree, where each leaf refers to a concept and contains a probabilistic description of that concept. Several algorithms produce classification trees for representing the unlabelled data. The most well-known algorithms are:

COBWEB: This algorithm assumes that all attributes are independent (an often too naive assumption). Its aim is to achieve high predictability of nominal variable values, given a cluster. This algorithm is not suitable for clustering large database data (Fisher, 1987). CLASSIT, an extension of COBWEB for continuous-valued data, unfortunately has similar problems as the COBWEB algorithm.

2. Neural Networks [4].
This type of algorithm represents each cluster by a neuron or "prototype". The input data is also represented by neurons, which are connected to the prototype neurons. Each such connection has a weight, which is learned adaptively during learning. A very popular neural algorithm for clustering is the self-organizing map (SOM). This algorithm constructs a single-layered network. The learning process takes place in a "winner-takes-all" fashion: The prototype neurons compete for the current instance. The winner is the neuron whose weight vector is closest to the instance currently presented. The winner and its neighbours learn by having their weights adjusted. The SOM algorithm is successfully used for vector quantization and speech recognition. It is useful for visualizing high-dimensional data in 2D or 3D space. However, it is sensitive to the initial selection of weight vector, as well as to its different parameters, such as the learning rate and neighbourhood radius.

2.4 Grid- Based Methods

These methods partition the 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. Grid-based clustering [13] algorithms first quantize the clustering space into a finite number of cells and then perform the required operations on the quantized space. Cells that contain more than certain number of points are treated as dense and the dense cells are connected to form the clusters. Some of the grid-based clustering algorithms are: Statistical INformation Grid-based method – STING, Wave Cluster, and Clustering In QUEst – CLIQUE. STING [8] first divides the spatial area into several levels of rectangular cells in order to form a hierarchical structure. The cells in a high level are composed from the cells in the lower level. It generates a hierarchical structure of the grid cells so as to represent the clustering information at different levels. Although STING generates good clustering results in a short running time, there are two major problems with this algorithm. Firstly, the performance of STING relies on the granularity of the lowest level of the grid structure.

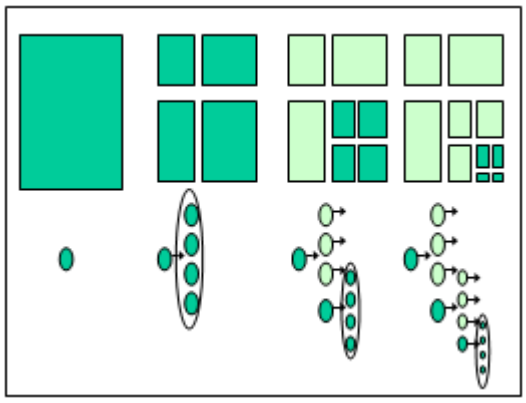

Figure 3: Cell generation and structuring in tree

Secondly, the resulting clusters are all bounded horizontally or vertically, but never diagonally. This shortcoming might greatly affect the cluster quality. CLIQUE is another grid-based clustering algorithm that starts by finding all the dense areas in the one-dimensional spaces corresponding to each attribute, and then generates the set of two-dimensional cells that might possibly be dense by looking at dense one-dimensional cells. Generally, CLIQUE generates the possible set of k-dimensional cells that might possibly be dense by looking at dense (k - 1) dimensional cells. CLIQUE produces identical results irrespective of the order in which the input records are presented. In addition, it generates cluster descriptions in the form of DNF expressions for ease of comprehension. Moreover, empirical evaluation shows that CLIQUE scales linearly with the number of instances, and has good scalability as the number of attributes is increased. Unlike other clustering methods, Wave Cluster does not require users to give the number of clusters applicable to low dimensional space. It uses a wavelet transformation to transform the original feature space resulting in a transformed space where the natural clusters in the data become distinguishable. Grid based methods help in expressing the data at varied level of detail based on all the attributes that have been selected as dimensional attributes. In this approach r representation of cluster data is done in a more meaningful manner.

2.4 Applications of Clustering

Clustering algorithms can be applied in many fields, for instance:

- Marketing: finding groups of customers with similar behavior given a large database of customer data containing their properties and past buying records
- Biology: classification of plants and animals given their features
- Libraries: book ordering
- Insurance: identifying groups of motor insurance policy holders with a high average claim cost; identifying frauds
- City-planning: identifying groups of houses according to their house type, value and geographical location
- Earthquake studies: clustering observed earthquake epicenters to identify dangerous zones
- WWW: document classification; clustering web log data to discover groups of similar access patterns.

3. Partitional Clustering Algorithms

A Partitional clustering [2] algorithm obtains a single partition of the data instead of a clustering structure, such as dendogram produced by a hierarchical technique. Partitional methods have advantages in applications involving large data sets for which the construction of a dendogram is computationally prohibitive. A problem accompanying the use of a Partitional algorithm is the choice of the number of desired output clusters. The Partitional technique [4] usually produce clusters by optimizing a criterion function defined either locally or globally. Combinatorial search of the set of possible labelling for an optimum value of a criterion is clearly computationally prohibitive. In practice, therefore, the algorithm is typically run multiple times with different starting states, and the best configuration obtained from all of the runs issued as the output clustering.

Combinatorial search of the set of possible labelling for an optimum value of a criterion is clearly computationally prohibitive. In practice, therefore, the algorithm is typically run multiple times with different starting states, and the best configuration obtained from all of the runs issued as the output clustering.

3.1 Types of Partitional Algorithms

- Squared Error Algorithms
- Graph-Theoretic Clustering
- Mixture-Resolving
- Mode-Seeking Algorithms

1. Graph-Theoretic Clustering.
Graph theoretic methods are methods that produce clusters via graphs. The edges of the graph connect the instances represented as nodes. A well-known graph-theoretic algorithm is based on the Minimal Spanning Tree—MST. Inconsistent edges are edges whose weight (in the case of clustering-length) is significantly larger than the average of nearby edge lengths. Another graph-theoretic approach constructs graphs based on limited neighbourhood sets (Urquhart, 1982). There is also a relation between hierarchical methods and graph theoretic clustering:

Single-link clusters are sub graphs of the MST of the data instances. Each sub graph is a *connected component*, namely a set of instances in which each instance is connected to at least one other member of the set, so that the set is maximal with respect to this property. These sub graphs are formed according to some similarity threshold.

Complete-link clusters are *maximal complete sub graphs*, formed using a similarity threshold. A maximal complete sub graph is a sub graph such that each node is connected to every other node in the sub graph and the set is maximal with respect to this property

2. K-Means Algorithm:
K-Means [5], [7], [14], [15] is one of the simplest unsupervised learning algorithms that solve the well-known clustering problem. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters (assume k clusters) fixed a priori. The main idea is

to define k centroids, one for each cluster. The next step is to take each point belonging to a given data set and associate it to the nearest centroid. When no point is pending, the first step is completed and an early group is done. At this point, it is need to re-calculate k new centroids [6], [9] as centres of the clusters resulting from the previous step. After these k new centroids, a new binding has to be done between the same data points and the nearest new centroid. A loop has been generated. As a result of this loop it may notice that the k centroids change their location step by step until no more changes are done. In other words centroids do not move any more. The algorithm is composed of the following steps:

Algorithm 1: K-Means Clustering Algorithm

1. **Initialize the number of clusters k.**
2. **Randomly selecting the centroids in the given dataset** $(c_1, c_2, \ldots, c_k)$.
3. **Compute the distance between the centroids and objects using the Euclidean Distance equation.**
   $d_{ij} = \|x_i - c_k\|^2$
4. **Update the centroids.**
5. **Stop the process when the new centroids are nearer to old one. Otherwise, go to step-3.**

Advantages of K-Means Algorithm

- It is easy to implement and works with any of the standard norms.
- It allows straightforward parallelization.
- It is incentive with respect to data ordering

Drawbacks of k-means algorithm

- The final clusters do not represent a global optimization result but only the local one, and complete different final clusters can arise from difference in the initial randomly chosen cluster centres.

4. MATLAB Environment

A. MATLAB

MATLAB is a high-level language and interactive environment for numerical computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. Using MATLAB, you can analyse data, develop algorithms, and create models and applications. The language tools and built-in math functions enable you to explore multiple approaches and reach a solution faster than with spreadsheets or traditional programming languages, such as C/C++ or Java. You can use MATLAB for a range of applications, including signal processing and communications, image and video processing, control systems, test and measurement, computational finance, and computational biology.

B. Key Features

- High-level language for technical computing
- Development environment for managing code, files, and data
- Interactive tools for iterative exploration, design, and problem solving
- Mathematical functions for linear algebra, statistics, Fourier analysis, filtering, optimization, and numerical integration
- 2-D and 3-D graphics functions for visualizing data
- Tools for building custom graphical user interfaces
- Functions for integrating MATLAB based algorithms with external applications and languages, such as C, C++, FORTRAN, Java, COM, and Microsoft Excel.

C. MATLAB Product Family Highlights

- MATLAB: Unified functions for 1-D, 2-D, and 3-D numerical integration and improved performance of basic math and interpolation functions
- MATLAB Compiler: MATLAB Compiler Runtime (MCR) available for download, simplifying distribution of compiled applications and components
- Image Processing Toolbox: Automatic image registration using intensity metric optimization
- Statistics Toolbox: Enhanced interface for fitting, prediction, and plotting with linear, generalized linear, and nonlinear regression
- System Identification Toolbox: Identification of continuous-time transfer functions

5. Conclusion

In this paper, clustering and its different clustering techniques are studied very well. The hierarchical method, Density based clustering method and Partitional clustering method are widely used in the recent research field. In our survey mainly focusing the Partitional clustering method and its different Partitional clustering algorithms like K-Means, K-Medoids and Graph Theoretic method. The K-Means and other Partitional clustering algorithms will be develop using Matlab tool and validate by the cluster validity index which helps to identify the good clustering method is our future work.

References

[1] Cohen S., Rokach L., Maimon O., Decision Tree Instance Space Decomposition with Grouped Gain-Ratio, Information Science, Volume 177, Issue 17, pp. 3592-3612, 2007.
[2] Fortier, J.J. and Solomon, H. 1996. Clustering procedures. In proceedings of the Multivariate Analysis, ’66, P.R. Krishnaiah (Ed.), pp. 493-506.
[3] Fraley C. and Raftery A.E., “How Many Clusters? Which Clustering Method? Answers Via Model-Based Cluster Analysis”, Technical Report No. 329. Department of Statistics University of Washington, 1998.
[4] Han, J. and Kamber, M. Data Mining: Concepts and Techniques. Morgan Kaufmann Publishers, 2001.
[5] HARTIGAN, J. and WONG, M. Algorithm AS136: “A k-means clustering algorithm”. Applied Statistics, 28, 100-108, 1979.
[6] Hinneburg A. and D.A. Keim, “A General Approach to Clustering in Large Databases with Noise,” Knowledge and Information Systems (KAIS), vol. 5, no. 4, pp. 387415, 2003.

[7] HEER, J. and CHI, E. 2001. "Identification of Web user traffic composition using multimodal clustering and information scent." 1st SIAM ICDM, Workshop on Web Mining, 51-58, Chicago, IL
[8] Huang, Z., Extensions to the k-means algorithm for clustering large data sets with categorical values. Data Mining and Knowledge Discovery, 2(3), 1998.
[9] Madhu Yedla, Srinivasa Rao Pathakota, T M Srinivasa , "Enhancing K-Means Clustering Algorithm with Improved Initial Center" , Madhu Yedla et al. / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 1 (2), pp121-125, 2010
[10] Murtagh, F. A survey of recent advances in hierarchical clustering algorithms which use cluster centers.. 26 354-359, 1984.
[11] Oded Maimon, Lior Rokach, "Data mining and Knowledge Discovery Handbook", Springer, second edition.2010.
[12] Pavel Berkhin , "Survey of Clustering Data Mining Techniques ".
[13] Rama. B "A Survey on clustering Current status and challenging issues" International Journal on Computer Science and Engineering (IJCSE) Vol. 02, No. 09, 2010, 2976-2980.
[14] Sauravjoyti Sarmah and 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.
[15] Velmurugan T and T. Santhanam, "Computational Complexity between K-Means and K-Medoids Clustering Algorithms," *Journal of Computer Science*, vol. 6, no. 3, 2010.