
Sejal Patel¹ Ankita Shah²
¹²Department of Computer Science and Engineering
1,2Namarayan Shastri Institute of Technology, Jetalpur, Ahmedabad, Gujarat, India

Abstract—Clustering is a useful technique that organizes a large quantity of unordered text documents into a small number of meaningful and coherent clusters. This paper presents the comparative study on clustering algorithms i.e. K-means and Hierarchical Clustering using MATLAB. MATLAB is a multi-paradigm numerical computing environment and fourth-generation programming language, developed by MathWorks.

Key words: Document Clustering, K-means, Hierarchical Clustering, MATLAB

I. INTRODUCTION
Cluster analysis or simply clustering is the progress of partitioning a set of data objects into subsets. Here, each subset is a cluster, that objects in a cluster are similar to one another, yet dissimilar to objects in other cluster. Clustering is sometimes called automatic Classification [1]. A critical difference here is that clustering can automatically find the grouping. Clustering is also called data segmentation in some application because clustering partitions large datasets into groups according to their similarity. Clustering can also be used for outlier detection, where outliers may be more interesting than common areas.

Clustering is known as unsupervised learning because the class label information is not present [2]. No super-vision means that there is no human expert who has assigned documents to classes.

Clustering has been widely used in many applications as below:
- Business intelligence
- Image pattern recognition
- Web search
- Biology
- Security

The popular areas of research in the cluster analysis include data mining, machine learning, statistics, spatial database, information retrieval (IR), web search, biology, marketing and many other application areas.

There are many requirements for clustering as a data mining tool, as well as aspects that can be used for comparing clustering methods. The following are typical requirements of clustering in data mining:
- Scalability
- Ability to deal with different types of attributes
- Discovery of clusters with arbitrary shape
- Requirements for domain knowledge to determine input parameters
- Ability to deal with noisy data
- Incremental clustering and insensitivity to input order
- Capability of clustering high-dimensionality data
- Constraint-based clustering

II. CLUSTERING ALGORITHMS
There are many clustering algorithms. Among which the two algorithms are going to be discussed. First, the partitional K-means algorithm and Second, the hierarchical clustering algorithm.

A. K-means Clustering Algorithm:
The K-means is the most important flat, hard clustering algorithm. It is iterative by nature. It is also referred to as Lloyd’s algorithm particularly in the computer science community [3]. The objective function of K-means is to minimize the average squared distance of objects from their cluster centers, where a cluster center is defined as the mean or centroid of the objects in a cluster. K-means clustering is the method of vector quantization, originally from signal processing.

A centroid-based clustering technique uses the centroid of a cluster, Cᵢ, to represent that cluster. The difference between an object p∈Cᵢ and cᵢ, the centroid, is measured by d(p,cᵢ), where d(x,y) is the Euclidean Distance between two points x and y. The quality of cluster Cᵢ can be measured by the within-cluster variation, which is the sum of squared error between all objects in Cᵢ and the centroid cᵢ, defined as

\[ E = \sum_{i=1}^{k} \sum_{p \in C_i} d(p, c_i)^2 \]

Where E is the sum of the squared error for objects in the dataset; p is the point in space representing a given object.

The K-means algorithms steps are shown below:
1) Step 1:
   - Choose the number of clusters ("K" in the K-means) required.
   - Choose the centroids for the clusters randomly from among the given set of elements.
2) Step 2:
   - Compute the distance of each of the element from the centroids.
Assign the element to the cluster with the closest centroid.

3) Step 3:
- Re-compute the cluster centroids based on the assignments done in step 2.

4) Step 4:
- Repeat Step 2 and Step 3 until the algorithm converges.

The generated clustering solutions are locally optimal for the given data set and the initial seeds [4]. The different choices of initial seed sets can result in very difficult final partitions.

1) 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.

2) 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.

B. Hierarchical Clustering Algorithm:

When partitioning methods meet the basic clustering requirement of organizing a set of objects into a number of exclusive groups, in some situations the examiner want to partition into groups at different levels such as in a hierarchy [4]. A hierarchical clustering method works by grouping data objects into hierarchy or “tree” of clusters.

Strategies for hierarchical clustering generally fall into two types:
1) Agglomerative:
This uses a “bottom up” strategy. It typically starts by letting each object from its own cluster and iteratively merges clusters into larger and larger clusters, until all the objects are in a single cluster or certain termination conditions are satisfied.

2) Divisive:
This uses a “top down” strategy. It starts by placing all objects in one cluster, which is the hierarchy’s root. It then divides the root cluster into several smaller subclusters, and recursively partitions those clusters into smaller ones.

III. COMPARATIVE STUDY OF ALGORITHMS

We have taken one wine.csv (comma separated value) file for executing the above two algorithms. The file has to be imported into MATLAB into matrix form. The file contains the numerical values of wine stock and sales. When it is imported in the MATLAB, it forms the matrix 178*14 named wine.

Now, K-means and Hierarchical clustering algorithms have been executed on this matrix.

A. K-means Clustering Algorithm in MATLAB:
The Statistics Toolbox in matlab provides a function K-Means to cluster the data [6]. The following is the description of the function K-Means with an example.

1) Syntax:
- IDX = kmeans(X,k)
- [IDX,C] = kmeans(X,k)
- [IDX,C,sumd] = kmeans(X,k)
- [IDX,C,sumd,D] = kmeans(X,k)
- [...] = kmeans(...,param1,val1,param2,val2,...)

2) Description:
IDX = kmeans(X,k) partitions the points in the n-by-p data matrix X into k clusters. This iterative partitioning minimizes the sum, over all clusters, of the within-cluster sums of point-to-cluster-centroid distances. Rows of X correspond to points, columns correspond to variables. kmeans returns an n-by-1 vector IDX containing the cluster indices of each point [7]. By default, kmeans uses squared Euclidean distances. When X is a vector, kmeans treats it as an n-by-1 data matrix, regardless of its orientation.
- [IDX,C] = kmeans(X,k) returns the k cluster centroid locations in k-by-p matrix C.
- [IDX,C,sumd] = kmeans(X,k) returns the within-cluster sums of point-to-centroid distances in the 1-by-k vector sumd.
- [IDX,C,sumd,D] = kmeans(X,k) returns distances from each point to every centroid in the n-by-k matrix D.
- [...] = kmeans(...,param1,val1,param2,val2,...) enables you to specify optional parameter/value pairs to control the iterative algorithm used by kmeans.

3) Implemented K-means:
The syntax for K-means used in our implementation is as below:
idx = kmeans(wine,6);
c1 = vendor(idx == 1,:);
c2 = vendor(idx == 2,:);
c3 = vendor(idx == 3,:);
c4 = vendor(idx == 4,:);
c5 = vendor(idx == 5,:);
c6 = vendor(idx == 6,:);
plot(c1(:,[1,2]),'*'); hold all
plot(c2(:,[1,2]),'*')
plot(c3(:,[1,2]),'*')
plot(c4(:,[1,2]),'*')
plot(c5(:,[1,2]),'*')
plot(c6(:,[1,2]),'*')
silhouette(wine,idx);

4) Hierarchical Clustering Algorithm in MATLAB:
The Statistics Toolbox in MATLAB provides a function cluster data which supports agglomerative clustering and performs all of the necessary steps. It incorporates the pdist, linkage, and cluster functions, which can be used separately for more detailed analysis. The dendrogram function plots the cluster tree [7]. Here a complete description is explained followed by an example.

1) Syntax:
- Z = linkage(X)
- Z = linkage(X,method)
- Z = linkage(X,method,metric)
- Z = linkage(X,method,pdist_inputs)
- Z = linkage(Y,method,metric,‘savememory’,value)
- Z = linkage(Y)
- Z = linkage(Y,method)

2) Description:
- Z = linkage(X) returns a matrix Z that encodes a tree of hierarchical clusters of the rows of the real matrix X.
- Z = linkage(X,method) uses a specified method, where method describes how to measure the distance between clusters.
- Z = linkage(X,method,metric) performs the distance measure metric to compute distances between the rows of X.
- Z = linkage(X,method,pdist_inputs) passes parameters to the pdist function, which is the function that computes the distance between rows of X.
- Z = Linkage (X,method,metric,’ savememory’,value) uses a memory saving algorithm when value is 'true', and uses the standard algorithm when value is 'false'.
- Z = linkage(Y) uses a vector representation Y of a distance matrix. Y can be a distance matrix as computed by pdist, or a more general dissimilarity matrix conforming to the output format of pdist.
- Z = linkage(Y,method) uses a specified method, where method describes how to measure the distance between clusters. The following cluster clearly explain the process.
3) Implemented Hierarchical Clustering:
The syntax for hierarchical clustering used in our implementation is as below:

```
wine;
Y=pdist(wine);
Z=linkage(Y);
dendrogram(Z);
```

![dendrogram for hierarchical clustering](image)

**Fig. 5:** dendrogram for hierarchical clustering

C. Comparison Table:

<table>
<thead>
<tr>
<th>Partitioning Algorithm</th>
<th>Hierarchical Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of clusters to be specified</td>
<td>No need to specify number of clusters</td>
</tr>
<tr>
<td>Un-nested</td>
<td>Nested</td>
</tr>
<tr>
<td>Non-deterministic: will in general produce different clusters with different initializations</td>
<td>Deterministic: produces the same clustering each time</td>
</tr>
<tr>
<td>Running time: O(n)</td>
<td>Running time: O(n^2)</td>
</tr>
<tr>
<td>Low memory usage: stores Mean or centroids only</td>
<td>Huge memory requirements: stores the n x n similarity matrix</td>
</tr>
<tr>
<td>Efficiency based usage</td>
<td>Quality based usage</td>
</tr>
<tr>
<td>Used for large data sets</td>
<td>Used for small data sets</td>
</tr>
<tr>
<td>Less Expensive</td>
<td>More Expensive</td>
</tr>
</tbody>
</table>

Table 1: Comparison between both algorithms

IV. CONCLUSION

Both K-means and hierarchical clustering algorithms have their own advantages and disadvantages considering the different parameters. For gaining useful and proper output, the examiners can use them according to the data they have. Any of the clustering algorithms tend to give the results whether the data are relevant or irrelevant.

V. FUTURE ENHANCEMENT

For the Better usage of both the algorithms and enhancement in the execution process, the examiners should use the hybrid algorithm of both of these K-means and hierarchical algorithms, such as Bisecting K-means. One can also prepare his own algorithm for clustering techniques or modify the existing algorithms.

REFERENCES

[1] Jiawei Han, Micheline Kamber, and jian Pei, Data Mining: Concepts and Techniques, 3rd ed. USA: Morgan Kaumann, 2012.