Data Warehousing & Data Mining (SE – 407a)

DELHI TECHNOLOGICAL UNIVERSITY

PROJECT REPORT

ON

TOPIC: Partitional and Hierarchical Clustering

Submitted To:

Sonika Dahiya

Submitted By:

Aman Bhatia (2K18/SE/019)

Ashish Kumar (2K18/SE/041)

Madhur Vaidya (2K18/SE/076)

INTRODUCTION

In Data Mining, Clustering is a general technique for statistical data analysis, which is used in dissimilar fields, including machine learning, pattern recognition, image analysis and bioinformatics. Clustering is an excellent data mining tool for a huge and multivariate database. It is the one of data mining techniques in which data is separated into the set of related objects. Clustering is an appropriate example of unsupervised classification. It means that clustering does not depend on predefined classes and training examples through classifying the data objects. A Partitioning and Hierarchical algorithm in data mining is the most active research algorithm among proposed algorithms. In this project, we **examined these clustering techniques i.e, partitioning and hierarchical based clustering techniques.**

Types of Clustering

There are four types of clustering:

1. Partitional Clustering
2. Hierarchical Clustering
3. Density Based Clustering
4. Grid-Based Clustering

But in this report, we only focus on first two clustering techniques.

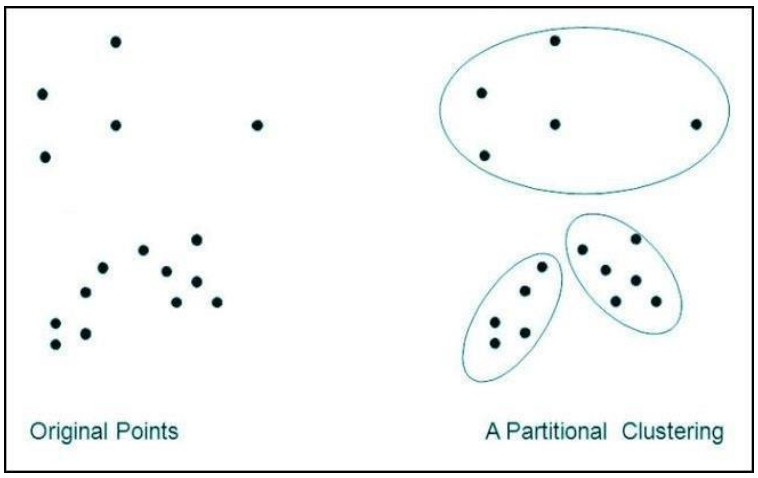
1. Partitioning Clustering

Partitioning clustering is considered to be the most popular class of clustering algorithm also known as **iterative relocation algorithm.** These algorithms minimize a given clustering criterion by iteratively relocating data points between clusters until an optimal partition is attained. Partition clustering algorithm splits the data points into k partition, where each partition represents a cluster. The partition is done based on a certain objective function. The cluster are formed to optimize an objective partitioning criterion, such as a dissimilarity function based on distance, so that the objects within a cluster are “similar”, whereas the objects of different cluster are “dissimilar”.

Partitioning clustering methods are useful for the applications where a fixed number of clusters are required. Partitioning-based clustering is highly efficient in terms of simplicity, proficiency, and easy to deploy, and computes all attainable clusters synchronously.

There are various types of partitioning clustering algorithms: K-Means, PAM (Partitioning around Medoids), FCM (Fuzzy C-Means), CLARANS (Clustering Large Applications Based on Randomized Search).

But in this section, we only examine K-means and PAM clustering algorithms.

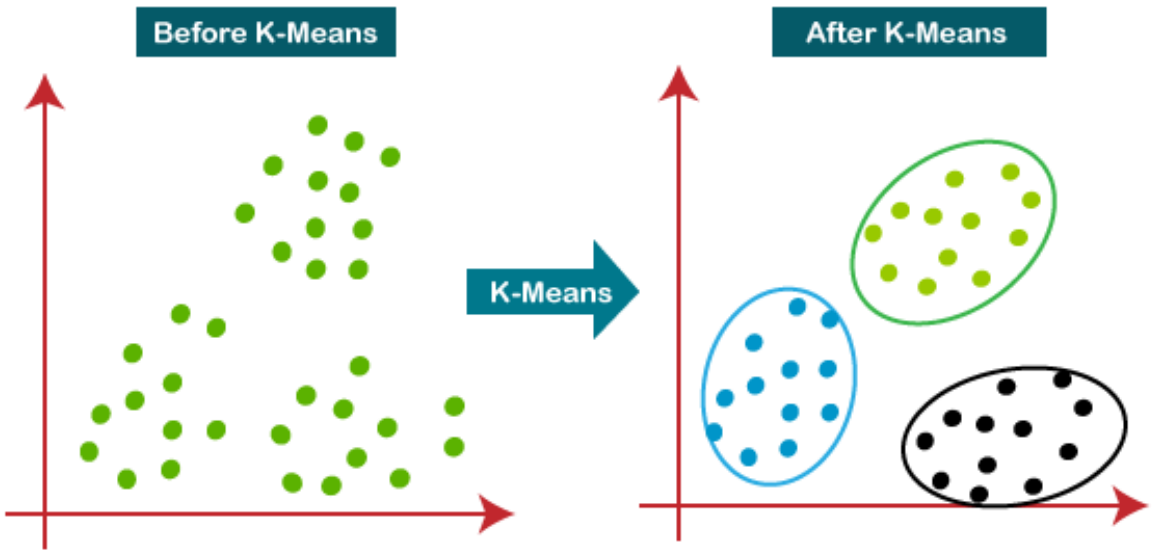


i) K-Means Algorithm

The k-means grouping algorithm was initially proposed by MacQueen and later enhanced by Hartigan and Wong. K-Means algorithm is the well-liked clustering algorithm that is used in scientific and industrial applications. It iteratively computes the clusters and their centroids which is a mean (average pt.) of points within a cluster. It is a top down approach to clustering. It is used for making and analyzing the clusters with 'n' amount of data points, point is separated into 'K' clusters supporting the similarity measure criterion where k is predetermined. The result generated by the algorithm generally depends on initial cluster centroids chosen.

The below diagram explains the working of the K-means Clustering Algorithm:

Here we choose the number of clusters to be 3. Then we assign each data point to their closest centroid, which will form the predefined K clusters. After applying K-means algorithm, the final model is ready.



Steps involved in K-Means Clustering:

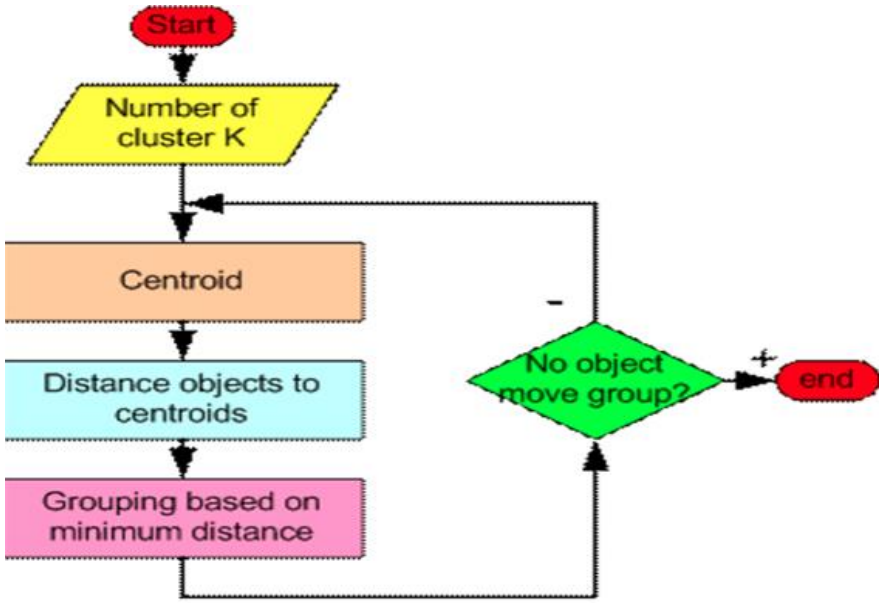
1. The first step when using k-means clustering is to indicate the number of clusters (k) that will be generated in the final solution.

2. The algorithm starts by randomly selecting k objects from the data set to serve as the initial centers for the clusters. The selected objects are also known as cluster means or centroids.

3. Next, each of the remaining objects is assigned to its closest centroid, where closest is defined using the Euclidean distance between the object and the cluster mean.

4. After that the algorithm computes the new mean value of each cluster. Now that the centers have been recalculated, every observation is checked again to see if it might be closer to a different cluster. All the objects are reassigned again using the updated cluster means.

5. The cluster assignment and centroid update steps are iteratively repeated until the cluster assignments stop changing.



The k-means algorithm has the following significant properties:

1. It is effective in dealing out huge data sets.

2. It frequently terminates at a local optimum.

3. It works just on numeric values.

Applications of K-Means Clustering Algorithm:

These are the applications where K-means clustering can be used −

- Market segmentation
- Document Clustering
- Image segmentation
- Image compression
- Customer segmentation
- Analyzing the trend on dynamic data

Real life implementation of K-means algorithm

K Means on Image Compression

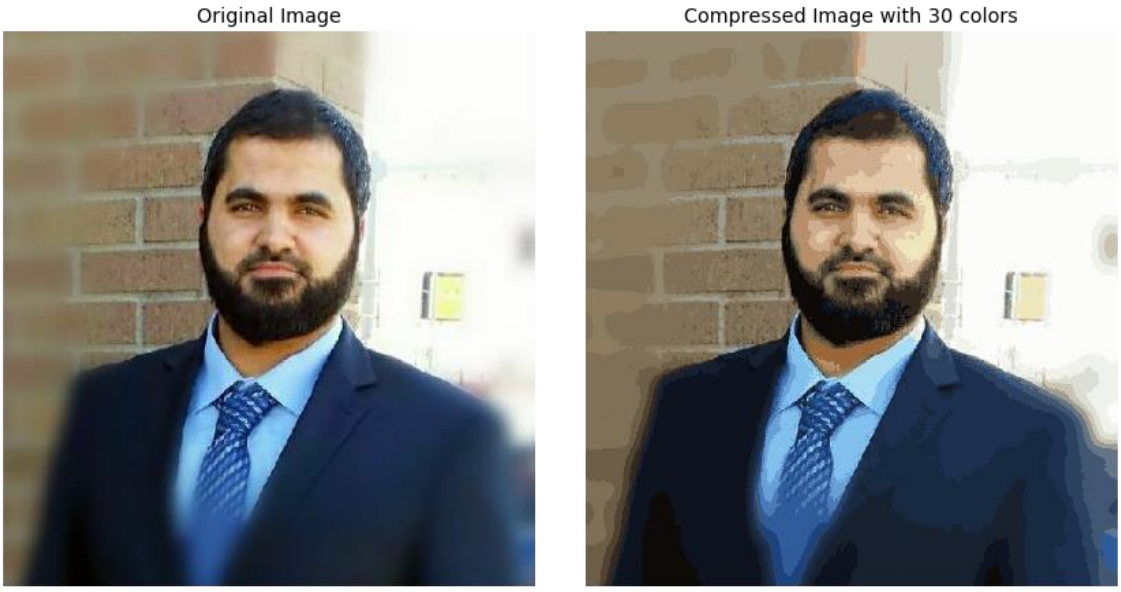
In this part, we implemented the k means algorithm to compress an input image. The image that we'll be working on is of dimensions 396 x 396 x 3. The 3 represents the 8 bit values for the RGB colours. Our goal is to reduce the number of colors to 30 and represent (compress) the photo using those 30 colors only and hence we use 30 clusters.

For this, every input pixel is used as a data point for the k-means algorithm. Hence, we reshape the image into a single vector of 396*396 which is equal to 156.816 data points. Doing so will allow us to represent the image using the 30 centroids for each pixel and would significantly reduce the size of the image by a factor of 6. This allows us to reduce the size of the input image drastically.

The Steps that we followed to make model:

1. First we choose Number of clusters to be 30 because the number of colors in the final image was 30.
2. Then we calculate Centroid for the image and use inbuilt Kmeans algorithm function function available in python.
3. Then we reshape the image to get the final compressed image having the same dimensions as of original image.

After processing input image through our model, following compressed image is produced:



Disadvantages of K-Means algorithm:

1. Generally terminates at the local optimum, and not the global optimum.

2. Can only be used when the mean is defined and therefore requires specifying k, the number of clusters, in advance.

3. The k-Means is not suitable for all types of data. For example, k-Means does not work on categorical data because mean (or centroids) cannot be defined.

ii) PAM (Partitioning Around Medoids)

The Partitioning Around Medoids (PAM) algorithm was introduced by Kaufman and Rousseeuw. The K-medoids algorithm is also termed as PAM (Partitioning Around Medoids) algorithm. It is based on the k representative objects, called medoids, among the objects of the dataset. The medoids are points with the smallest average dissimilarity to all other points. The algorithm follows the same steps that are followed by the k-means algorithm, but the use of medoids instead of means makes the algorithm more robust to outliers.

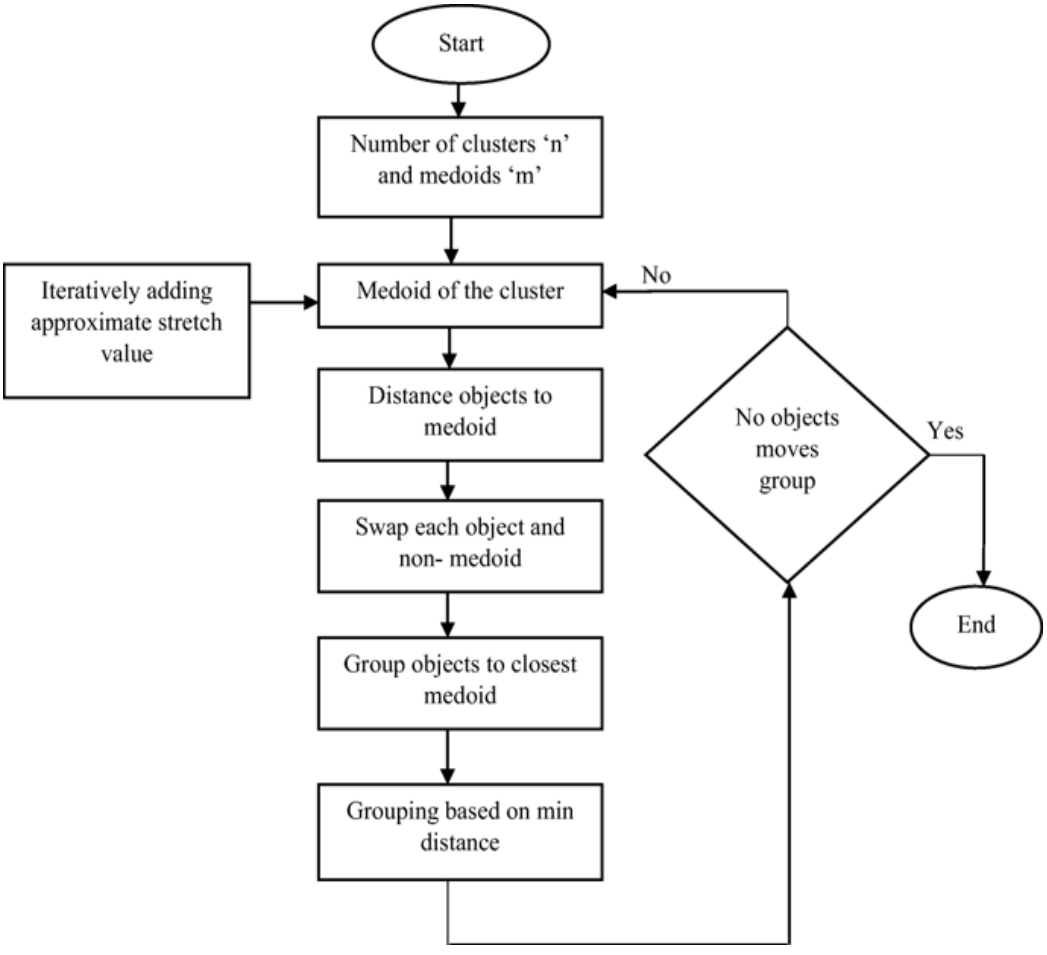
Steps involved in PAM Clustering Algorithm:

1. The PAM algorithm is based on the search for "k" representative objects or medoids among the observations of the data set.

2. After finding a set of "k" medoids, clusters are constructed by assigning each observation to the nearest medoid.

3. Next, each selected medoid "m" and each non-medoid data point are swapped and the objective function is computed. The objective function corresponds to the sum of the dissimilarities of all objects to their nearest medoid.

4. The SWAP step attempts to improve the quality of the clustering by exchanging selected objects (medoids) and non-selected objects. If the objective function can be reduced by interchanging a selected object with an unselected object, then the swap is carried out. This is continued until the objective function can no longer be decreased. The goal is to find k representative objects which minimize the sum of the dissimilarities of the observations to their closest representative object.



Advantages of K-Medoids (or PAM) Algorithm:

1. PAM works efficiently for small data sets, however does not scale well for huge data sets.
2. PAM can also be used in datasets that have categorical and/or other types of discrete data, such as binary data.
3. K-Medoid Algorithm is fast and converges in a fixed number of steps.
4. PAM is less sensitive to outliers than other partitioning algorithms.

Disadvantages of K-Medoids (or PAM) Algorithm:

1. The main disadvantage of K-Medoid algorithms is that it is not suitable for clustering non-spherical (arbitrary shaped) groups of objects. This is because it relies on minimizing the distances between the non-medoid objects and the medoid (the cluster centre) – briefly, it uses compactness as clustering criteria instead of connectivity.

2. It may obtain different results for different runs on the same dataset because the first k-medoids are chosen randomly.

3. One of the problems of the PAM algorithm is that the desired number of clusters must be predetermined.

Difference between K-Means clustering and K-medoids (or PAM) clustering:

K-Means	**K-Medoids**
Complexity is O(ikn)	Complexity is $O(i\,k(n-k)^2)$
More efficient.	Comparatively less efficient.
Sensitive to outliers.	Not sensitive to outliers.
Efficient for large data sets.	Efficient for small data sets.
Cannot be used in categorical type of data.	Can be used in categorical type of data.

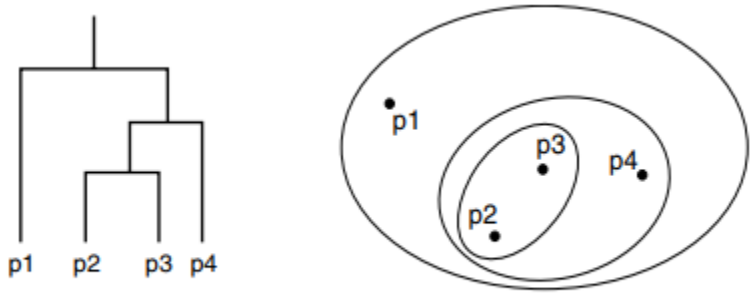
Note: In time complexity, "n" is the number of data points to cluster, "i" is the number of iterations needed and "k" is the number of clusters (or centroids).

Pros and Cons of Partitioning Algorithm: It is simple to understand and implement. It takes less time to execute as compared to other techniques. The drawback of this algorithm is the user has to provide a predetermined value of k and it produces spherical shaped clusters. It cannot handle with noisy data objects.

2. Hierarchical Clustering

Hierarchical clustering, also known as **hierarchical cluster analysi**s, is an algorithm that groups similar objects into groups called **clusters**. The endpoint is a set of clusters, where each cluster is distinct from each other cluster, and the objects within each cluster are broadly similar to each other. For example, all files and folders on the hard disk are organized in a hierarchy.

In Hierarchical Clustering, the aim is to produce a hierarchical series of nested clusters. A diagram called **Dendrogram** (A Dendrogram is a tree-like diagram that statistics the sequences of merges or splits) graphically represents this hierarchy and is an inverted tree that describes the order in which factors are merged (bottom-up view) or cluster are break up (top-down view).



(a) Dendrogram. (b) Nested cluster diagram.

Hierarchical clustering begins by treating every data points as a separate cluster. Then, it repeatedly executes the subsequent steps:

1. Identify the 2 clusters which can be closest together, and
2. Merge the 2 maximum comparable clusters. We need to continue these steps until all the clusters are merged together.

There are two types of hierarchical clustering algorithms:

1. Agglomerative
2. Divisive

i) Agglomerative

Agglomerative clustering starts with each observation as its own cluster. The two closest clusters are joined into one cluster. The next closest clusters are grouped together and this process continues until there is only one cluster containing the entire data set.
In this algorithm, it is started with n clusters and at the end only 1 cluster remains.

The algorithm steps of Agglomerative algorithms are

i) Preparing the data:

   a) rows representing observations (individuals);

   b) and columns representing variables.

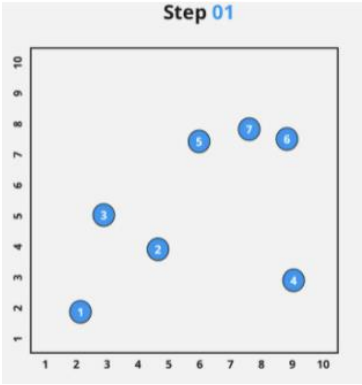
ii) Computing (dis)similarity information between every pair of objects in the data set: The results of this computation is known as a distance or dissimilarity matrix the distance is calculated using euclidean or manhattan distances.

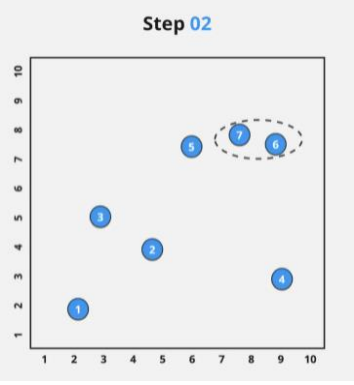
iii) Consider every data point as a individual cluster

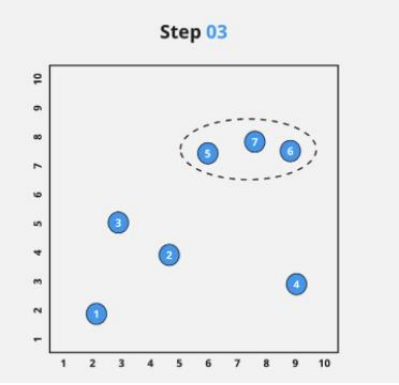
iv) Merge the clusters which are highly similar or close to each other.

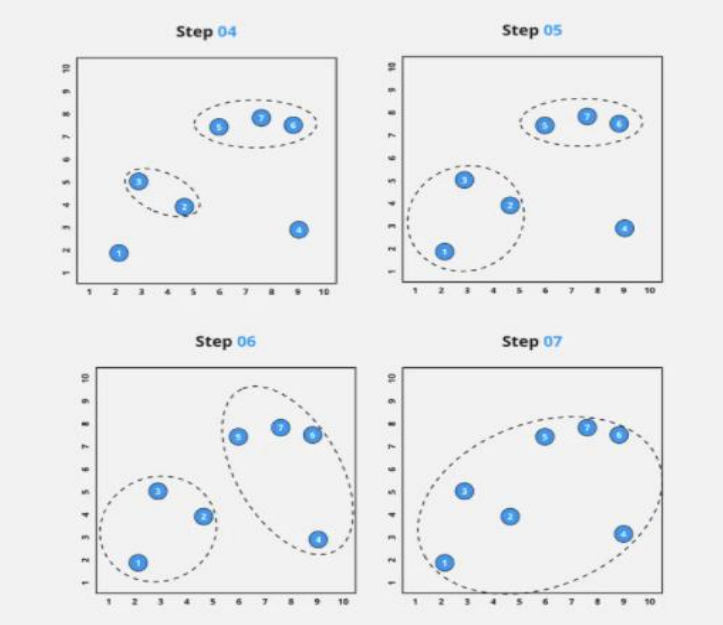
v) Recalculate the proximity matrix for each cluster

vi) Repeat Step 4 and 5 until only a single cluster remains.







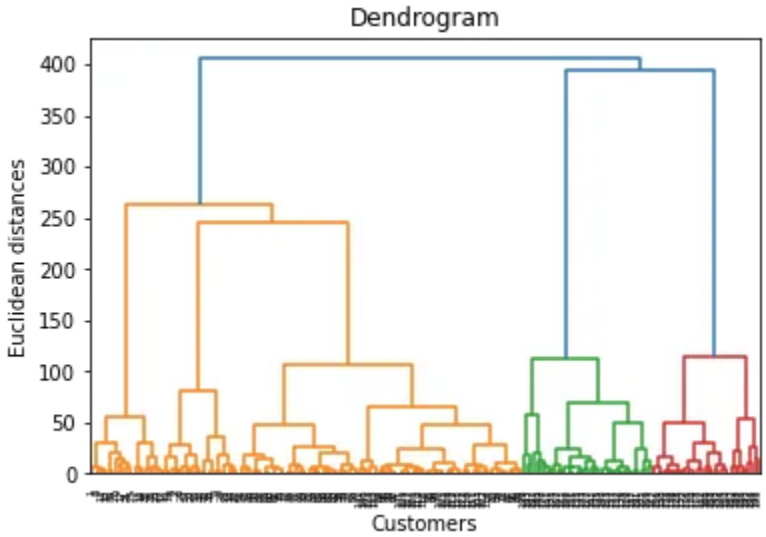


Real World implementation of Agglomerative clustering

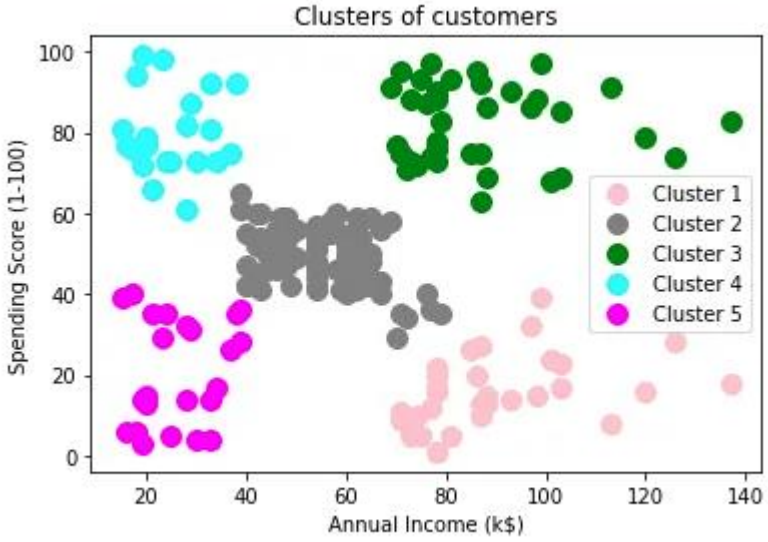
Taking a mall for example and extracting the relation between the amount spent and the annual income.

	CustomerID	Genre	Age	Annual Income (k$)	Spending Score (1-100)
0	1	Male	19	15	39
1	2	Male	21	15	81
2	3	Female	20	16	6
3	4	Female	23	16	77
4	5	Female	31	17	40

A dendrogram is built using the ward's method of calculating linkage:



Now while building the model for clustering we specify the number of clusters to be 5 and the linkage type to be ward.



The following clusters are made.

This image shows how the data points are now divided into 5 clusters. The cluster shows a link between the spending score and the annual income, for example it can be said that people belonging in cluster 5 tend to spend less due to a lower income however people in cluster 1, even though having a higher income still decide to spend similar to the ones in cluster 5.

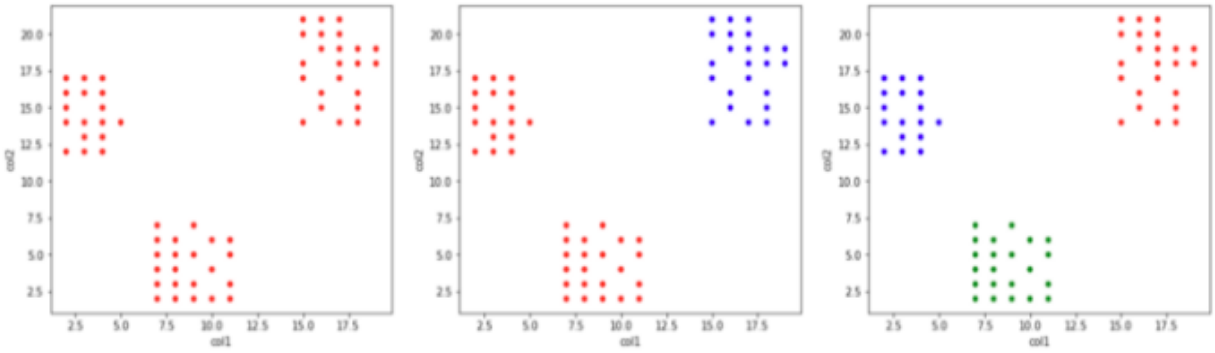
ii) Devisive

Divisive hierarchical clustering works by starting with 1 cluster containing the entire data set. The observation with the highest average dissimilarity (farthest from the cluster by some metric) is reassigned to its own cluster. Any observations in the old cluster closer to the new cluster are assigned to the new cluster. This process repeats with the largest cluster until each observation is its own cluster.

In this algorithm, it is started with 1 cluster of all the data points and at the end there are n clusters.

The algorithm steps of divisive algorithms are:

i) Initially, all the objects or points in the dataset belong to one single cluster.
ii) Partition the single cluster into two least similar clusters.
iii) And continue this process to form the new clusters until the desired number of clusters means one cluster for each observation.



Difference between Agglomerative and Divisive Clustering Algorithms:

Divisive	Agglomerative
Divisive uses top-bottom approach in which the parent is visited first then the child.	Agglomerative Hierarchical clustering method allows the clusters to be read from bottom to top and it follows this approach so that the program always reads from the sub-component first then moves to the parent.
Divisive the parent cluster is divided into smaller cluster and it keeps on dividing till each cluster has a single object to represent.	Agglomerative hierarchical method consists of objects in which each object creates its own clusters and these clusters are grouped together to create a large cluster. It defines a process of merging that carries on till all the single clusters are merged together into a complete big cluster that consists of all the objects of child clusters.

Some more advanced Hierarchical Clustering techniques are:

1. **BIRCH**: BIRCH (balanced iterative reducing and clustering using hierarchies) is an unsupervised data mining algorithm used to perform hierarchical clustering over particularly large data-sets.
2. **Chameleon:** Chameleon is a new agglomerative hierarchical clustering algorithm that overcomes the limitations of existing clustering algorithms. The Chameleon algorithm's key feature is that it accounts for both interconnectivity and closeness in identifying the most similar pair of clusters.
3. **Probabilistic:** uses probabilistic models to measure distances between clusters.

Advantages of hierarchical clustering:

1. The agglomerative technique is easy to implement.
2. It can produce an ordering of objects, which may be informative for the display.
3. In agglomerative Clustering, there is no need to pre-specify the number of clusters.
4. By the Agglomerative Clustering approach, smaller clusters will be created, which may discover similarities in data.

Disadvantages of hierarchical clustering:

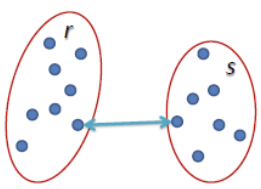
1. The agglomerative technique gives the best result in some cases only.
2. The algorithm can never undo what was done previously, which means if the objects may have been incorrectly grouped at an earlier stage, and the same result should be close to ensure it.
3. The usage of various distance metrics for measuring distances between the clusters may produce different results. So performing multiple experiments and then comparing the result is recommended to help the actual results' veracity.

Types of Linkage

In the above section, we neglected to define what "close" means. There are a variety of possible metrics, but I will list the 5 most popular: single-linkage, complete-linkage, average-linkage, centroid-linkage and ward linkage.

i) **Single Linkage:**

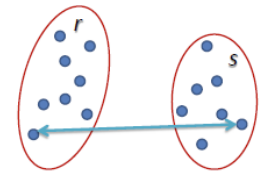
a. In the Single Linkage method, the distance of two clusters is defined as the minimum distance between an object (point) in one cluster and an object (point) in the other cluster. This method is also known as the nearest neighbour method.



$$L(r,s) = \min(D(x_{ri}, x_{sj}))$$

ii) **Complete Linkage**:

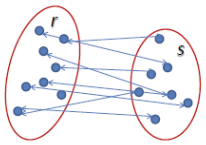
a. Complete-linkage (farthest neighbour) is where distance is measured between the farthest pair of observations in two clusters. This method usually produces tighter clusters than single-linkage, but these tight clusters can end up very close together. Along with average-linkage, it is one of the more popular distance metrics.



$$L(r,s) = \max(D(x_{ri}, x_{sj}))$$

iii) **Average Linkage:**

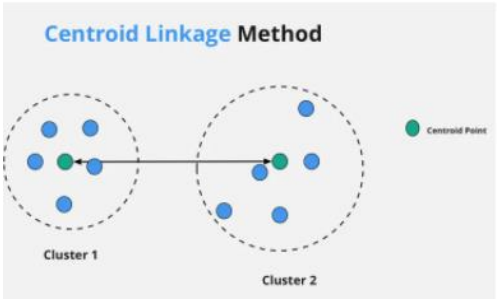
a. Average-linkage is where the distance between each pair of observations in each cluster are added up and divided by the number of pairs to get an average inter-cluster distance. Average-linkage and complete-linkage are the two most popular distance metrics in hierarchical clustering.



$$L(r,s) = \frac{1}{n_r n_s} \sum_{i=1}^{n_r} \sum_{j=1}^{n_s} D(x_{ri}, x_{sj})$$

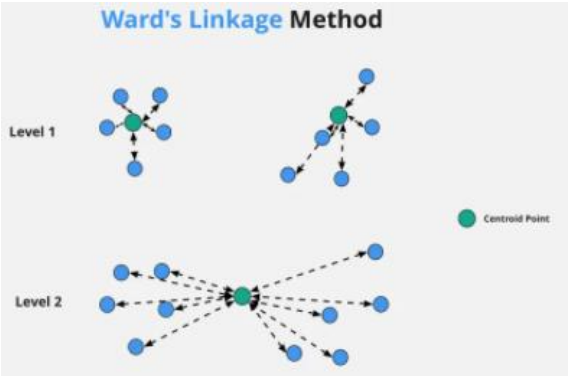
iv) **Centroid Method:**

a. Centroid-linkage is the distance between the centroids of two clusters. As the centroids move with new observations, it is possible that the smaller clusters are more similar to the new larger cluster than to their individual clusters causing an inversion in the dendrogram. This problem doesn't arise in the other linkage methods because the clusters being merged will always be more similar to themselves than to the new larger cluster.



v) **Ward's Method:**

a. Ward's Linkage method is the similarity of two clusters which is based on the increase in squared error when two clusters are merged, and it is similar to the group average if the distance between points is distance squared.



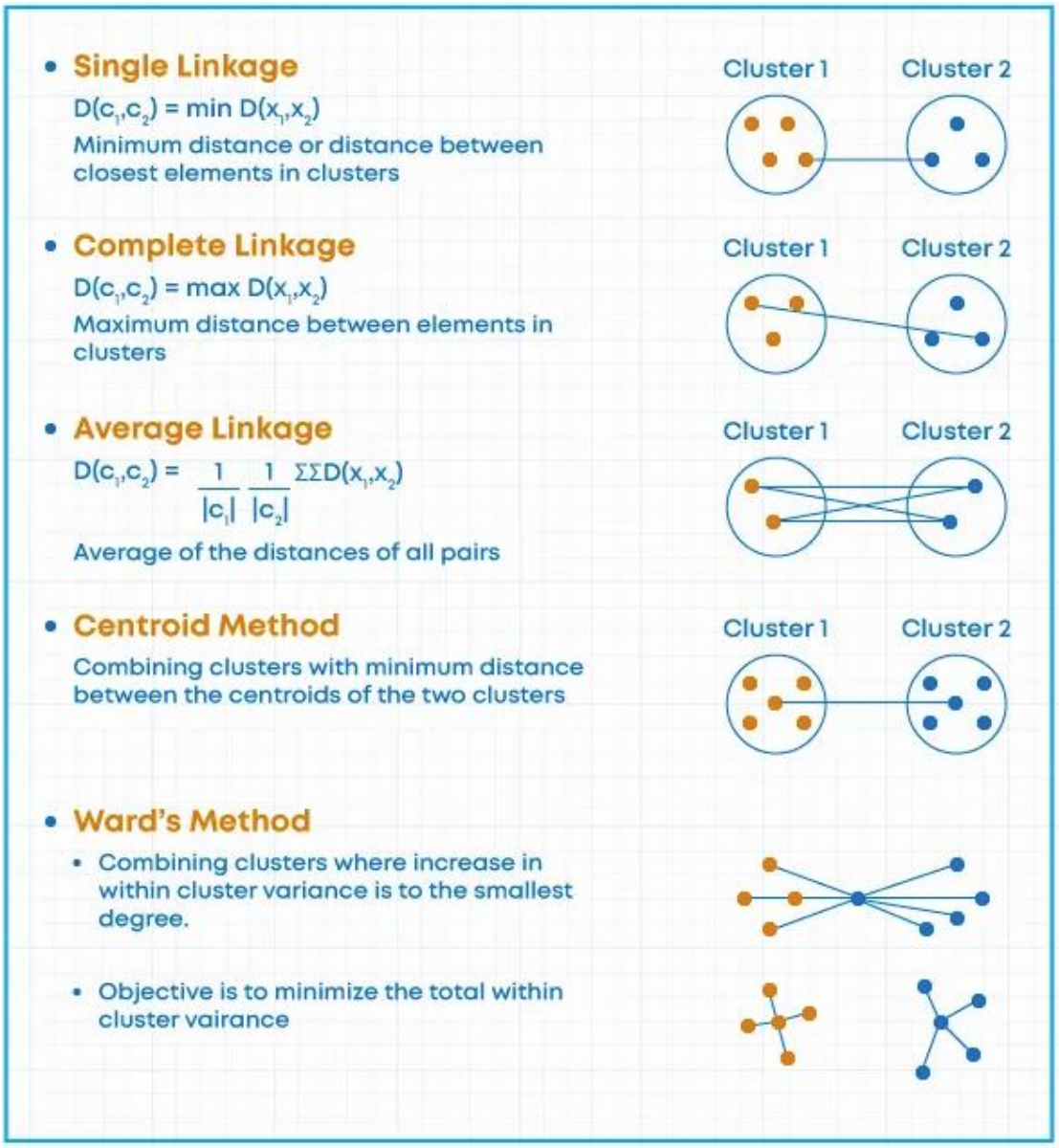


Fig. Types of Linkage

Difference between Hierarchical and Partitional Clustering

Hierarchical and Partitional Clustering have key differences in running time, assumptions, input parameters and resultant clusters.

1. Typically, partitional clustering is faster than hierarchical clustering.
2. Hierarchical clustering requires only a similarity measure, while partitional clustering requires stronger assumptions such as number of clusters and the initial centers.
3. Hierarchical clustering does not require any input parameters, while partitional clustering algorithms require the number of clusters to start running.
4. Hierarchical clustering returns a much more meaningful and subjective division of clusters but partitional clustering results in exactly k clusters.
5. Hierarchical clustering algorithms are more suitable for categorical data as long as a similarity measure can be defined accordingly.

CONCLUSION

After taking a deep understanding of the two topics it can be understood that each algorithm has its strengths and weaknesses. It depends on the situation and the type of data one is dealing with to decide which algorithm will work and provide the most appropriate result according to the present circumstances.

- Partitional clustering is considerably faster than hierarchical clustering and in instances with huge data chunks and less time it will be preferable to apply partitional clustering.

- Hierarchical Clustering is often used in the form of descriptive rather than predictive modeling.

- Mostly we use Hierarchical Clustering when the application requires a hierarchy. The advantage of Hierarchical Clustering is we don't have to pre-specify the clusters. However, it doesn't work very well on vast amounts of data or huge datasets. Moreover, in hierarchical Clustering, once a decision is made to combine two clusters, it cannot be undone.

- If the data is categorical it is more advantageous to use hierarchical clustering compared to partition clustering.

REFERENCES

1. https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.433.3348&rep=rep1&type=pdf
2. https://medium.com/analytics-vidhya/partitional-clustering-181d42049670
3. https://www.geeksforgeeks.org/ml-k-medoids-clustering-with-example/
4. https://www.ripublication.com/ijaer18/ijaerv13n24_12.pdf
5. https://www.ijcsmc.com/docs/papers/August2017/V6I8201725.pdf
6. https://web.itu.edu.tr/uzunper/documents/Document_Clustering.pdf
7. https://www.differencebetween.com/difference-between-hierarchical-and-vs-partitional-clustering/
8. https://towardsdatascience.com/k-means-clustering-algorithm-applications-evaluation-methods-and-drawbacks-aa03e644b48a
9. https://www.javatpoint.com/k-means-clustering-algorithm-in-machine-learning
10. https://towardsdatascience.com/hierarchical-clustering-agglomerative-and-divisive-explained-342e6b20d710
11. https://www.javatpoint.com/hierarchical-clustering-in-machine-learning
12. https://www.saedsayad.com/clustering_hierarchical.htm
13. https://towardsdatascience.com/agglomerative-clustering-and-dendrograms-explained-29fc12b85f23
14. https://www.datanovia.com/en/lessons/agglomerative-hierarchical-clustering/
15. https://towardsdatascience.com/hierarchical-clustering-agglomerative-and-divisive-explained-342e6b20d710
16. https://dataaspirant.com/hierarchical-clustering-algorithm/#t-1608531820435
17. https://towardsdatascience.com/hierarchical-clustering-and-its-applications-41c1ad4441a6
18. https://www.geeksforgeeks.org/ml-hierarchical-clustering-agglomerative-and-divisive-clustering/
19. https://www.sawaal.com/data-warehousing-interview-questions/what-is-the-difference-between-agglomerative-and-divisive-hierarchical-clustering_7149
20. https://www.displayr.com/what-is-hierarchical-clustering/