Clustering

Clustering involves searching for groups of similar observations in data. The goal of clustering is to partition the observations into clusters so that the difference between observations within a cluster is much smaller than the distance between observations between clusters.

1 K-means Clustering

1.1 General Overview

- The k-means algorithm is an iterative descent algorithm that tries to partition a dataset into k pre-defined non-overlapping clusters, where each observation belongs to one cluster only.
- Observations are assigned to a cluster such that the sum of squared distances between the observations and the cluster centroids are minimised. Where the cluster centroid is defined as the arithmetic mean of all observations within a cluster.
- The lower the variation present within a cluster the more homogeneous the observations are within that same cluster.
- Generally speaking, Euclidean distance is used as the distance/dissimilarity measure for the k-means algorithm.

1.2 General Outline of the Algorithm

The k-means algorithm is a partition based clustering algorithm and is performed as follows:

1. Define a specific number of clusters k.
2. Randomly initialise the cluster centroids, one method is to randomly select k observations (without replacement) and define these values as the initial cluster centroids.
3. For each observation compute its distance to each centroid and reassign the observation to its nearest centroid.
4. Recompute each clusters centroid based on the newly updated cluster memberships.
5. Repeat steps 3 and 4 until the cluster memberships remain unchanged or a maximum number of iterations is reached.

1.3 K-means Clustering Algorithm

The k-means clustering algorithm with a Euclidean distance/dissimilarity measure has the following objective function:

$$J = \sum_{i=1}^{m} \sum_{k=1}^{K} z_{ik} ||x_i - \mu_k||^2 \tag{1}$$

where

$$z_{ik} = \begin{cases} 1, & \text{if observation i belongs to cluster k.} \\ 0, & \text{otherwise.} \end{cases}$$

Below is the mathematical derived version of the k-means algorithm:

1. First specify the number of clusters k and randomly select k observations (without replacement) to represent the initial centroids of each cluster, $\mu_1, ..., \mu_k$.
2. The objective function is minimised in a two step procedure, first with respect to z_{ik} which identifies any changes in cluster memberships, and subsequently with respect to μ_k which calculates the new cluster centroids.

3. First minimise the objective function J with respect to z_{ik}, the cluster memberships, keeping μ_k fixed.

$$\begin{aligned} \frac{\partial J}{\partial z_{ik}} &= \frac{\partial}{\partial z_{ik}} \sum_{i=1}^{m} \sum_{k=1}^{K} z_{ik} ||x_i - \mu_k||^2 \\ \frac{\partial J}{\partial z_{ik}} &= \sum_{i=1}^{m} \sum_{k=1}^{K} ||x_i - \mu_k||^2 \end{aligned} \tag{2}$$

$$\implies z_{ik} = \begin{cases} 1, & \text{if } k = \underset{j}{\operatorname{argmin}} ||x_i - \mu_j||^2 \text{ (find the closest cluster centroid).} \\ 0, & \text{otherwise.} \end{cases}$$

4. Reassign each observation x_i to its nearest centroid.

5. Now, minimise the objective function J with respect to μ_k, the cluster centroids, keeping z_{ik} fixed.

$$\frac{\partial J}{\partial \mu_k} = \frac{\partial}{\partial \mu_k} \sum_{i=1}^{m} \sum_{k=1}^{K} z_{ik} ||x_i - \mu_k||^2 \tag{3}$$

$$\frac{\partial J}{\partial \mu_k} = -2 \sum_{i=1}^{m} z_{ik}(x_i - \mu_k) = 0 \tag{4}$$

$$\implies \sum_{i=1}^{m} z_{ik} x_i - \sum_{i=1}^{m} z_{ik} \mu_k = 0 \tag{5}$$

$$\sum_{i=1}^{m} z_{ik} \mu_k = \sum_{i=1}^{m} z_{ik} x_i \tag{6}$$

$$\mu_k \sum_{i=1}^{m} z_{ik} = \sum_{i=1}^{m} z_{ik} x_i \tag{7}$$

$$\implies \mu_k = \frac{\sum_{i=1}^{m} z_{ik} x_i}{\sum_{i=1}^{m} z_{ik}} \tag{8}$$

where μ_k can be interpreted as the mean of the observations present in cluster k.

6. Next, recompute the centroids based on the newly updated cluster memberships.

7. Repeat the two minimisation steps until the cluster memberships remain unchanged or a maximum number of iterations is reached. In other words, continue iteratively reassigning the cluster memberships and subsequently recalculating each clusters centroid until the cluster memberships remain unchanged.

1.4 Additional Comments and Drawbacks

- The k-means algorithm uses distance based measurements, like the Euclidean distance, to determine the similarity between observations. Features measured on different scales can have a great impact on the k-means results, i.e. the variable with the greatest variance will figure most prominently in the clustering solution. Therefore, it is recommended to standardise the dataset prior to applying the k-means algorithm.

- Different clusters may be identified depending on the initialisation of the k-means algorithm, this can be due to the algorithm getting stuck in a local minimum and may not converge to the global minimum. Therefore, it is important to use different starting cluster centroid values and to pick the solution which yields the lowest sum of squared distance.

- k-means is designed to capture clusters which have a spherical shape around the centroid. Therefore, the k-means algorithm performs poorly on data whose underlying clusters have more complicated geometric shapes, e.g. elliptical shaped clusters.

- k-means is also sensitive to outliers (consider using the k-medoids algorithm).

- k-means is unable to deal with categorical data (consider using the k-modes algorithm).

1.5 Centroid Initialisation

The cluster centroid initialisation method mentioned above simply selects k observations from the dataset (without replacement) and defines these values as the clusters centroids. However, this is not the only initialisation method, some others will be discussed below:

- randomly assign observations to k clusters and calculate the k cluster means/centroids.
- specify your own initial cluster centroids.
- use results from an exploratory hierarchical clustering algorithm.

2 Hierarchical Clustering

Hierarchical clustering is a form of unsupervised learning technique which aims to group similar observations together and keep dissimilar observations apart. Hierarchical clustering involves constructing a tree-like structure to show groups of observations, where the final clustering is built up over a number of steps where similar observations are joined together. Hierarchical clustering requires the user to specify a measure of dissimilarity between (disjoint) groups of observations, based on the pairwise dissimilarities among the observations in the two groups. It produces hierarchical representations in which the clusters at each level of the hierarchy are created by merging clusters at the next lower level. At the lowest level, each cluster contains a single observation. At the highest level there is only one cluster containing all of the data.

It is up to the user to decide which level (if any) actually represents a 'natural' clustering in the sense that observations within each of its groups are sufficiently more similar to each other than to observations assigned to different groups at that level.

2.1 Agglomerative Hierarchical Clustering Algorithm

The hierarchical clustering algorithm is performed as follows:

1. Agglomerative hierarchical clustering starts by assigning each observation to its own group.
2. A distance matrix is calculated between all observations using a specified distance measurement e.g. Euclidean, Manhattan or binary depending on the data type.
3. A specified linkage method is then used to measure the dissimilarity between two groups. Groups which are closest to each other are merged together to form one single group.
4. This process is repeated until only one group remains.

Hierarchical clustering results are presented as a dendrogram. A dendrogram provides a highly interpretable complete description of the hierarchical clustering in a graphical format. In a dendrogram the height of each node is proportional to the value of the intergroup dissimilarity between its two daughters, i.e. the y-axis represents the distance between the groups when they were joined together. The groups joined together at the bottom are close together while groups at the top are far apart. A dendrogram should be viewed mainly as a description of the clustering structure of the data and not a hard partitioning of the data. Cutting a dendrogram horizontally at a particular height splits the data into a specific number of clusters, where each cluster is represented by the vertical lines which intersect the horizontal divisor. Generally speaking we search for a relatively wide range of distances over which the number of clusters in the solution does not change.

There are many different types of linkage method which measure the dissimilarity between groups. Examples of linkage methods include single, complete and average linkage, each of which calculates the dissimilarity between groups in different ways. An alternative linkage method, called Ward's linkage, merges pairs of clusters based on those which minimise the error sum of squares value.

2.2 Linkage Methods

Given two groups A and B, the dissimilarity $d(A, B)$ between A and B is computed from the set of pairwise observation dissimilarities $d_{ii'}$ where one member of the pair i is in A and the other i' is in B. It is possible to measure the dissimilarity between these two groups $d(A, B)$ using a linkage method. There are many different linkage methods, some of the most common linkage methods are described below.

2.2.1 Single Linkage Method

The single linkage method measures the dissimilarity between two groups as the minimum distance between two pairs of elements within each group. In other words, the smallest distance between any two elements of two separate groups is defined as the dissimilarity between those two groups. The two groups with the smallest dissimilarity between each other are combined together, this process is repeated until no groups remain.

$$d(A, B) = \min_{x \in A, y \in B} d(x, y) \tag{9}$$

Single linkage is prone to chaining. Chaining is the process where single observations are continuously added to a group which just gets larger. This phenomenon results in long string like clusters where observations at opposite ends of such clusters are often quite dissimilar. Given that single linkage joins clusters by the shortest link, it can be poor at discerning between poorly separated clusters. For this reason single linkage is rarely used in practice.

2.2.2 Complete Linkage Method

Complete linkage is the opposite of single linkage where the dissimilarity between two groups is defined as the largest distance between a pair of elements from each group. Two clusters are subsequently combined together based on the smallest of these dissimilarity values. Clusters combined together until no groups remain.

$$d(A, B) = \max_{x \in A, y \in B} d(x, y) \tag{10}$$

Complete linkage will tend to produce compact clusters with small diameters, however, it can produce clusters that violate the 'closeness' property. That is, observations assigned to a cluster can be much closer to members of other clusters than they are to some members of their own cluster.

2.2.3 Average Linkage Method

Average linkage finds the average pairwise distance between two groups, with this value defined as the dissimilarity between two clusters. As before, groups with the smallest dissimilarity between them are combined together, this process is repeated until no clusters remain.

$$d(A, B) = \frac{1}{|A||B|} \sum_{x \in A} \sum_{y \in B} d(x, y) \tag{11}$$

Average linkage can represent a compromise between the two extremes of single and complete linkage. It attempts to produce relatively compact clusters that are relatively far apart. However, its results depend on the numerical scale on which the observation dissimilarities $d_{ii'}$ are measured

2.2.4 Ward's Linkage Method

Ward's linkage method is an alternative approach to the usual linkage methods like single, average or complete linkage. Ward's linkage method examines cluster analysis problems as analysis of variance problems. Therefore, clusters/groups are merged when the error sum of squares is minimised or the r^2 value is maximized between two separate clusters/groups. The r^2 is a measure of the proportion of variation explained by a certain clustering of observations within the dataset. The r^2 value is defined as:

$$r^2 = \frac{\text{TSS} - \text{ESS}}{\text{TSS}} \tag{12}$$

Where ESS is given as:

$$\sum_i \sum_j \sum_k (x_{ijk} - \bar{x}_{i.j})^2 \tag{13}$$

And TSS is defined as:

$$\sum_i \sum_j \sum_k (x_{ijk} - \bar{x}_{..k})^2 \tag{14}$$

3 Assessing Clustering Solutions

It is important to assess how well we are clustering the data. Here are some of the methods used to assess the clustering results.

3.1 Rand Index and Adjusted Rand Index

The Rand index is used as a measure for the correspondence between two cluster solutions. The Rand index is defined between the values of 0 and 1, with values close to 1 indicating high similarity between two cluster solutions and a low Rand index value indicating low similarity between two cluster solutions.

A simple definition of the Rand index between two clustering solutions $S1$ and $S2$ is given as follows:

$$\text{R(S1, S2)} = \frac{\sum_{i<j}^{n} \gamma_{ij}}{\binom{n}{2}} \tag{15}$$

where

$$\gamma_{ij} = \begin{cases} 1, & \text{if } i \text{ and } j \text{ are in the same cluster in } S1 \text{ and in the same cluster in } S2. \\ 1, & \text{if } i \text{ and } j \text{ are in different clusters in } S1 \text{ and in different clusters in } S2. \\ 0, & \text{otherwise.} \end{cases}$$

More formally, the Rand index is defined as:

$$\text{Rand Index} = \frac{\binom{n}{2} + 2\sum_{i=1}^{c_1}\sum_{j=1}^{c_2}\binom{n_{ij}}{2} - \left[\sum_{i=1}^{c_1}\binom{n_{i.}}{2} + \sum_{j=1}^{c_2}\binom{n_{.j}}{2}\right]}{\binom{n}{2}} \tag{16}$$

Where n_{ij} is the number of observations contained within cluster i for the first clustering method and cluster j for the second clustering method. c_1 and c_2 are the number of clusters in the first and second clustering methods, respectively. Also $n_{i.} = \sum_{j=1}^{c_2} n_{ij}$ and $n_{.j} = \sum_{i=1}^{c_1} n_{ij}$.

One of the issues with the Rand index is that it can produce large values, even when labels are assigned randomly to observations.

It is for this reason that the adjusted Rand index was created. The adjusted Rand index is very similar to the Rand index, however it has been adjusted for agreement due to chance that can occur between two cluster solutions. The adjusted Rand index has an upper bound of 1, where values close to 1 indicate high similarly between two cluster solutions, similarly low values indicate little agreement between two clustering results. However, unlike the Rand index a negative adjusted Rand index value can be obtained, this also indicates little or no agreement between two clustering solutions. The adjusted Rand index is defined as follows:

$$\text{Adjusted Rand Index} = \frac{\binom{n}{2}\sum_{i=1}^{c_1}\sum_{j=1}^{c_2}\binom{n_{ij}}{2} - \sum_{i=1}^{c_1}\binom{n_{i.}}{2}\sum_{j=1}^{c_2}\binom{n_{.j}}{2}}{\frac{1}{2}\binom{n}{2}\left[\sum_{i=1}^{c_1}\binom{n_{i.}}{2} + \sum_{j=1}^{c_2}\binom{n_{.j}}{2}\right] - \sum_{i=1}^{c_1}\binom{n_{i.}}{2}\sum_{j=1}^{c_2}\binom{n_{.j}}{2}} \tag{17}$$

Where n_{ij} is the number of observations contained within cluster i for the first clustering method and cluster j for the second clustering method. c_1 and c_2 are the number of clusters in the first and second clustering methods, respectively. Also $n_{i.} = \sum_{j=1}^{c_2} n_{ij}$ and $n_{.j} = \sum_{i=1}^{c_1} n_{ij}$.

3.2 Silhouette Plots

Silhouette plots are a visual method that are used to help interpretation and validation of a clustering solution. The silhouette value is a measure of how similar an object is to its own cluster compared to other clusters. Silhouette values range from -1 to 1, where a high value represents an observation that is well matched with its own cluster and poorly matched with the other clusters. A large proportion of high values suggest that the clustering results are an appropriate solution for the dataset, however, large amounts of low or negative values suggests that this is not the optimal clustering solution.

1. Suppose the data are clustered into k clusters via some clustering technique.
2. Let $a(i)$ be the average dissimilarity/distance from observation i to the other members of the same cluster C_i i.e.
$$a(i) = \frac{1}{|C_i| - 1} \sum_{j \in C_i, i \neq j} d(i, j) \tag{18}$$
where $d(i, j)$ is the distance between the observations i and j in cluster C_i (we divide by $|C_i| - 1$ as we do not include the distance $d(i, i)$ in the sum). $a(i)$ can be interpreted as a measure of how well i is assigned to its cluster (the smaller the value the better the assignment).
3. Next, we compute the average dissimilarity/distance from observation i to the members of another cluster C_k ($C_i \neq C_k$).
4. For each observation $i \in C_i$, let $b(i)$ be the minimum such dissimilarity, i.e.
$$b(i) = \min_{k \neq i} \frac{1}{|C_k|} \sum_{j \in C_k} d(i, j) \tag{19}$$
where b(i) can be interpreted as the smallest mean distance of i to all points in any other cluster, of which i is not a member.
5. The silhouette value for observation i is defined as:
$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))} \tag{20}$$
$$\implies s(i) = \begin{cases} 1 - \dfrac{a(i)}{b(i)}, & \text{if a(i) < b(i).} \\ 0, & \text{a(i) = b(i).} \\ \dfrac{b(i)}{a(i)} - 1, & \text{if a(i) > b(i).} \end{cases}$$
where $-1 \leq s(i) \leq 1$. For $s(i)$ to be close to 1 we require $a(i) << b(i)$, as $a(i)$ is a measure of how dissimilar i is to its own cluster, a small value means it is well matched. On the other hand, a large $b(i)$ means that i is more matched to a neighbouring cluster. Therefore, an $s(i)$ value close to 1 indicates that the data is appropriately clustered. On the other hand an $s(i)$ value close to -1 suggests that observation i should be clustered with one of its neighbouring clusters. An $s(i)$ value of approximately 0 suggests that the observation i is on the border between two clusters.
6. Finally, calculate the mean of $s(i)$ over all points of a cluster, this measures how tightly grouped all the points in a cluster are. Averaging $s(i)$ over the whole dataset, represents an overall measure of how well the data have been clustered.

3.3 Elbow Plot Method

- This method is specific to the k-means algorithm and it is generally not possible to apply to the hierarchical clustering algorithm.
- The elbow plot is a method that helps to identify the optimal number of clusters to group a dataset.
- Different values of k are plotted against their corresponding sum of squared distance (SSE) between the observations and their assigned clusters centroids.
- k is chosen at the point where the SSE starts to plateau, ie.e where there appears to be an 'elbow' in the plot.