CHAPTER 6

Text Mining 2: Clustering

Motivation. We have 1000 tweets. What are they about? Can we summarise them?

Clustering allows us to group similar documents or words, allowing us to identify topics in the document set.

1. Introduction to Clustering

What is Clustering?

Clustering. To gather objects into clusters.

Cluster. A subset where each element of the subset is similar using some measurement and each element is dissimilar to element of other clusters.

Therefore, for a given clustering, items in a cluster should be close, but items in different clusters should be distant.

Simple Clustering Example. We can cluster by age: We can cluster by gender: We can cluster by hair colour: We can cluster by age and hair colour (here we have assumed that brown is more similar to blonde than black):

	Age	Gender	Hair Colour
Person 1	7	Male	Brown
Person 2	5	Female	Black
Person 3	12	Male	Black
Person 4	32	Female	Brown
Person 5	45	Female	Blonde
Person 6	28	Male	Brown

Clustering by observation. Clusters may be identifiable for low dimensional data, or data projected to two dimensions using Multidimensional Scaling:

Computing Clusters. Once we have a set of objects (e.g tweets) and a distance metric over the objects (e.g. TF-IDF with cosine, or binary metric), we can use a clustering algorithm to cluster the objects.

In this Unit, we will examine the clustering algorithms:

- K-means clustering
- Hierarchical clustering

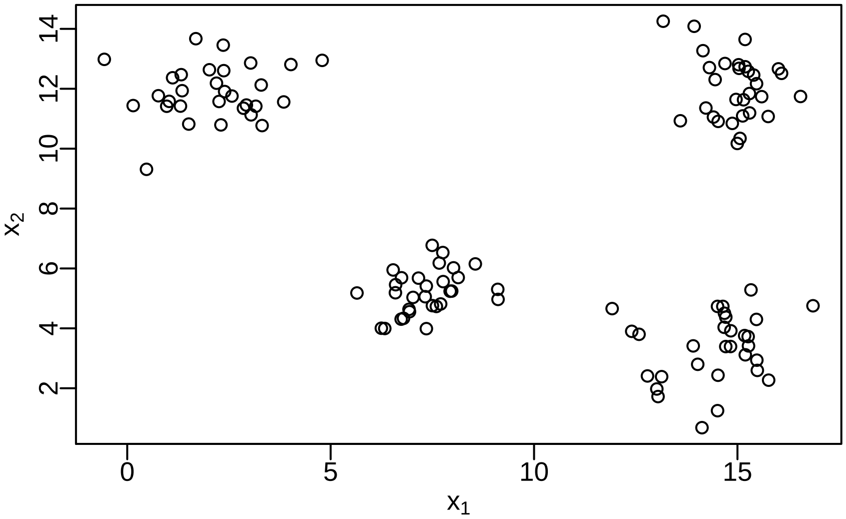

FIGURE 1.

2. K-means Clustering

Examining K-means.

K-means. K-means clustering was designed to be used in a Euclidean space, meaning it uses the Euclidean distance:

$$d(\vec{x}_i, \vec{x}_j)^2 = \sum_{n=1}^{N} (x_{in} - x_{jn})^2 = \|\vec{x}_i - \vec{x}_j\|_2^2$$

We want to minimise the within-point scatter over all clusters:

$$W(C) = \sum_{k=1}^{K} \sum_{C(i)=k} \sum_{C(j)=k} \|\vec{x}_i - \vec{x}_j\|_2^2$$

K is the number of clusters, $C(i) = k$ if point $\vec{x}_i$ belongs to cluster k, and $W(C)$ is the sum of the distances between all points in each cluster. By minimising $W(C)$, we obtain tight clusters.

K-means Algorithm. K-means iterates through two steps, until the centres stabilise (stop moving).

Compute which object belongs to which cluster. For each object $\vec{x}_i$:

$$C(i) = \operatorname*{arg\,min}_{1 \le k \le K} \|\vec{x}_i - \vec{m}_k\|$$

Compute the centre of each cluster. For each cluster k:

$$\begin{aligned} \vec{m}_k &= \operatorname*{arg\,min}_{\vec{m} \in \mathbb{R}^m} \sum_{C(i)=k} \|\vec{x}_i - \vec{m}\| \\ &= \frac{1}{N_k} \sum_{C(i)=k} \vec{x}_i \end{aligned}$$

Complete K-means Algorithm.

1. Choose K the number of clusters wanted.
2. Assign the cluster centres $\vec{m}_k$ randomly.
3. While the new centres are different to the last centres:
   i. Assign each object to its closest cluster centre.
   ii. Compute the cluster mean from the cluster objects.

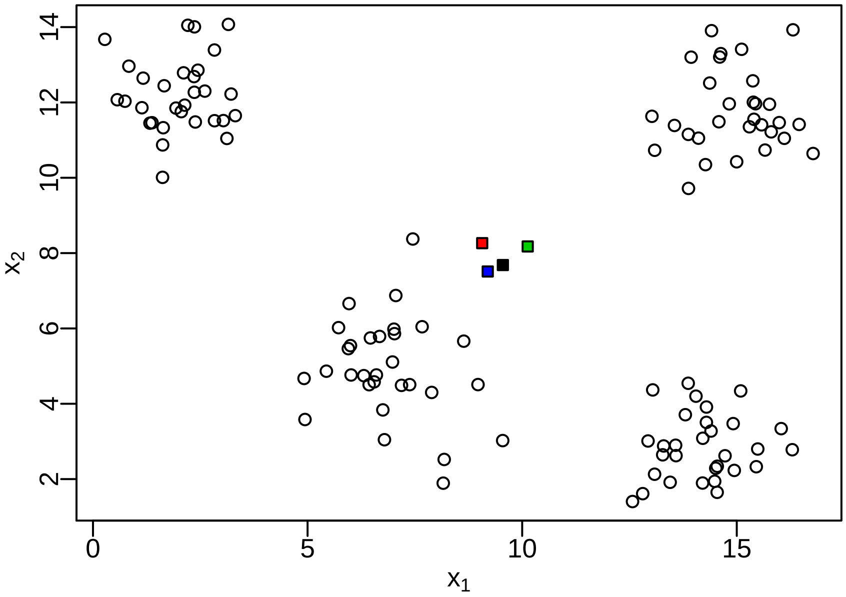

FIGURE 2.

K-means Example: Assign Centres.

K-means Example: Assign cluster membership.

K-means Example: Recompute cluster centres.

K-means Example: Reassign membership.

K-means Example: Recompute cluster centres.

K-means numerical Example.

Problem. Three two dimensional data points. Let's find two clusters.

```
##      [,1] [,2]
## [1,]    1    1
## [2,]    2    1
## [3,]    4    5
```

Begin with the randomly allocated centres:

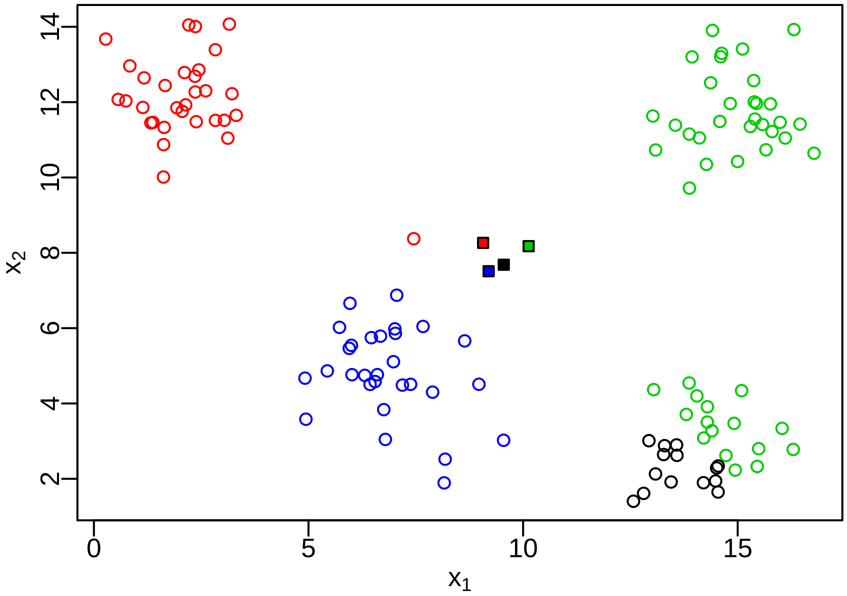

FIGURE 3.

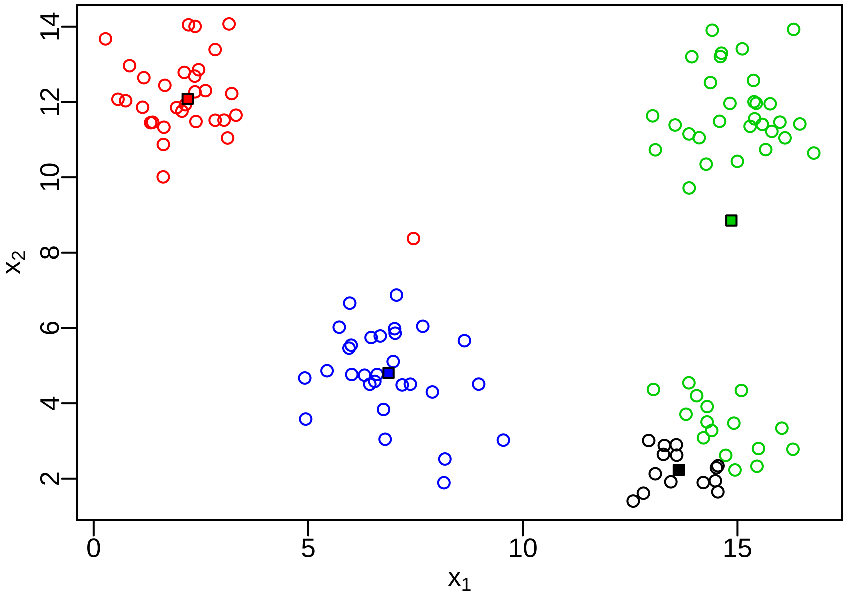

FIGURE 4.

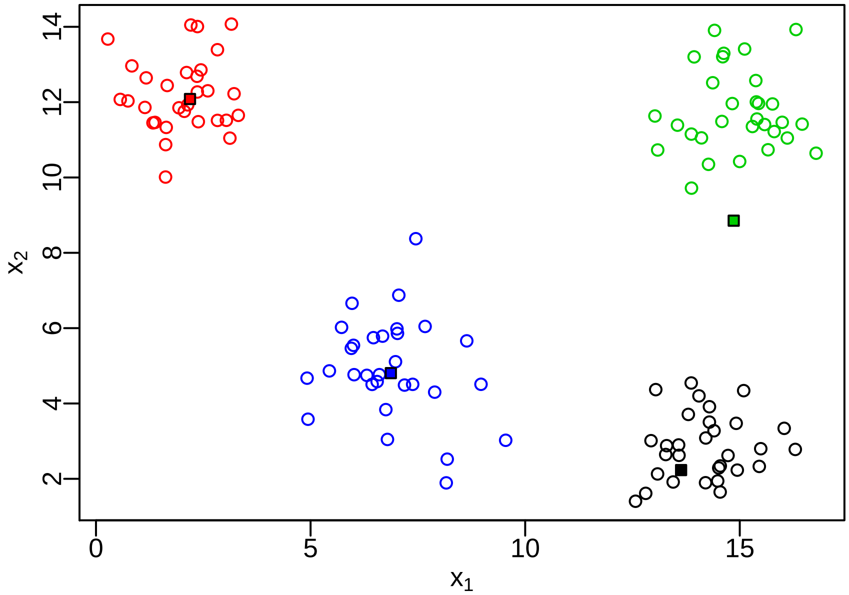

FIGURE 5.

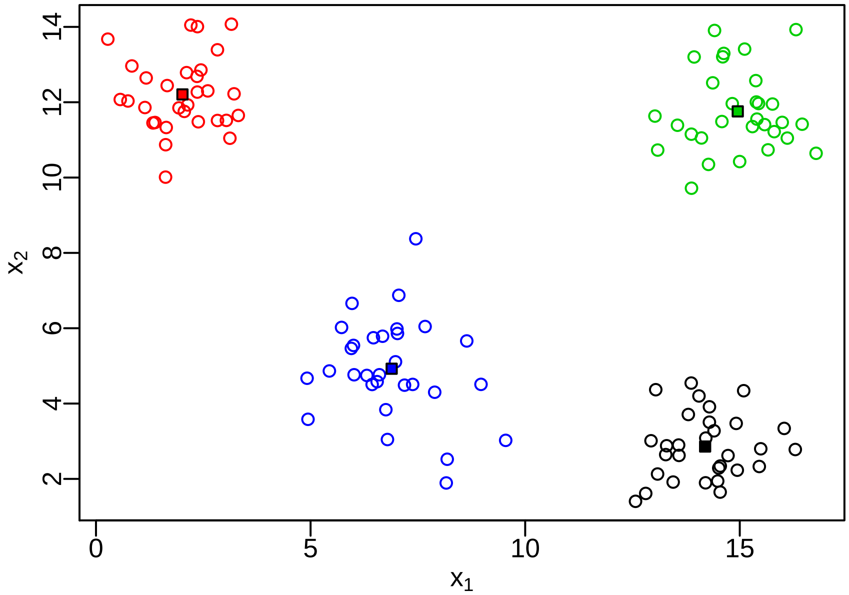

FIGURE 6.

```
##      [,1] [,2]
## [1,]    2    2
## [2,]    3    3
```

Problems with K-means. K-means clustering is one of the most widely used clustering algorithms due to its simplicity. But it does have problems.

1. K means requires that the cluster centres are positioned somewhere before the iteration begins. The initial cluster centres are usually chosen randomly, so there is a chance that we obtain *different clusters* each time we run K-means.
2. K-means requires us to set the number of clusters before the algorithm is run, but we usually *don't know how many clusters* there are. We will later examine how to set the number of clusters.
3. K-means uses Euclidean distance, which *may not be appropriate for our data*. How can we ensure that our data distances are measured using Euclidean distance?

We will now examine how to deal with these problems.

Initial position of cluster centres.

Initialisation of Cluster Centres. The location of the initial cluster centres can effect the final clustering. How can we place these points to ensure a good clustering?

We can either:

1. Use another clustering algorithm to identify rough cluster centres, or
2. Run k-means many times using random initialisation and use the *best* clustering.

We can perform choice 2 in R's `kmeans` function using the `nstart` parameter.

But what does *best* clustering mean?

k means best clustering. K means defines the best clustering of point to satisfy:

- all points within a cluster are close
- all points in different clusters are distant.

Where the distance is measured using Euclidean distance. There are many distances, we need to combine them into a single score.

To combine the distances use use sum of squares:

- SSW: Within Sum of Squares (sum of squared distances between points and cluster centre)
- SSB: Between Sum of Squares (sum of squared distances between cluster centre and centre of all points)

So if we have two different clusterings (from different random starts), we choose the clustering that has the smallest SSW and the largest SSB.

Within Sum of Squares (SSW) distances.

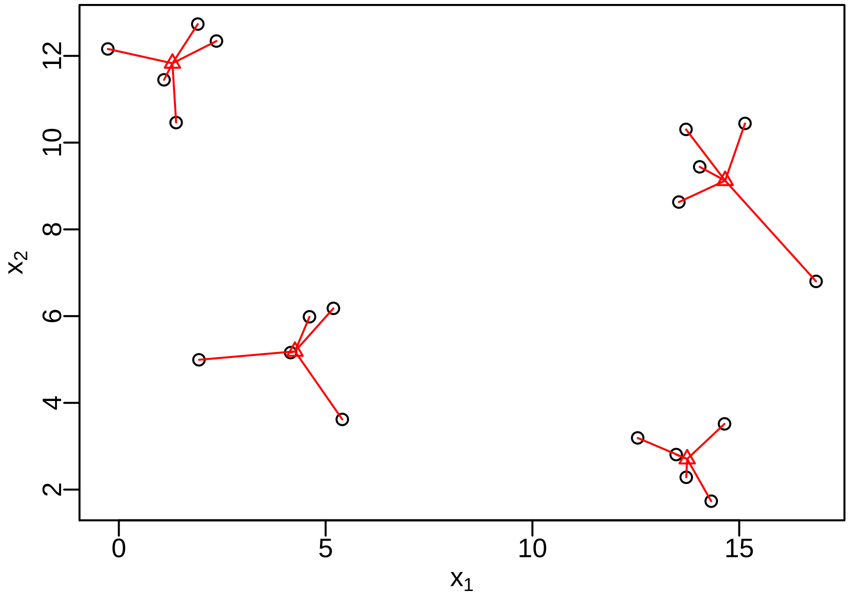

FIGURE 7.

Within Sum of Squares. The Euclidean distance from a point to the point's cluster centre is:

$$d(x, c) = \|x - c\|$$

The Within Sum of Squares (SSW) is the sum of all squared distances from each point to its cluster centre:

$$SSW = \sum_{x,c=C(x)} \|x - c\|^2$$

where $c = C(x)$ is the centre of the cluster containing x.

Between Cluster Sum of Squares (SSB) distances.

Between Sum of Squares. The Euclidean distance from a point's cluster centre to the mean of all points is:

$$d(c, \bar{x}) = \|c - \bar{x}\|$$

The Between Sum of Squares (SSB) is the sum of all squared distances from each point's cluster centre to the mean of all points:

$$SSB = \sum_{x,c=C(x)} \text{|C(x)|} \; \|c - \bar{x}\|^2$$

where $c = C(x)$ is the centre of the cluster containing x, and $\bar{x} = \sum x/n$, n is the number of points.

|C(x)| is the number of points in each cluster. Note that we *sum over the set of all points*, not the set of clusters.

Comparing SSW and SSB. We choose the clustering that has the smallest SSB and largest SSW, but what if one clustering has the smallest SSB and another has the largest SSW?

It can never happen! The value SSW + SSB is constant for any set of points! So as SSW increases, SSB must decrease at the same rate.

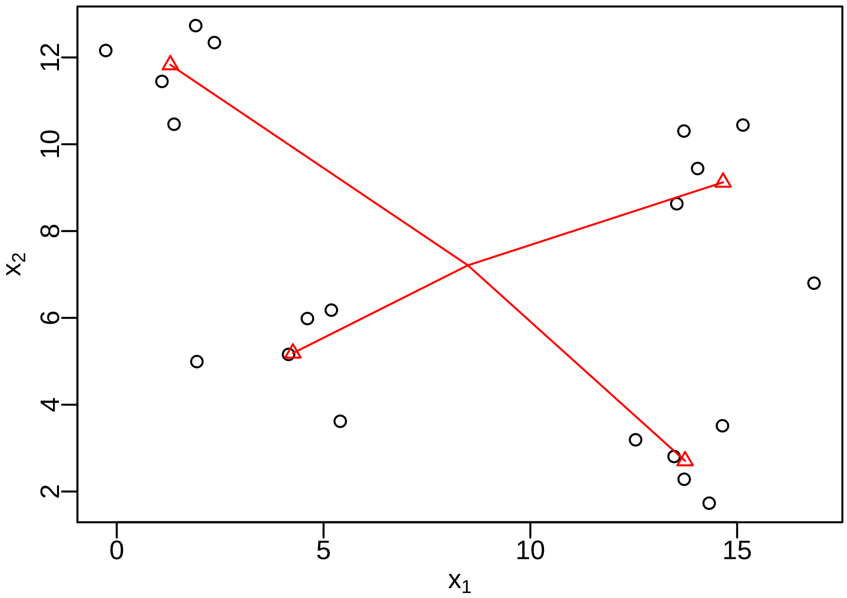

FIGURE 8.

Total sum of squares (SST). The total sum of squares is the sum of squared distances of each point to the mean of the set of points (not dependent on clustering).

$$SST = SSW + SSB$$

Total Sum of Squares (SST).

Total Sum of Squares Calculation. SST is the sum of squares of all points from the centre of the data.

$$SST = \sum_{i=1}^{N} \left(\vec{x}_i - \vec{\bar{x}}\right)^2$$

The mean point is:

$$\vec{\bar{x}} = \frac{1}{N}\sum_{i=1}^{N} \vec{x}_i$$

where N is the number of points.

Note that both SSW and SSB depend on the position of the cluster centres, but SST does not.

Computing SSW, SSB and SST.

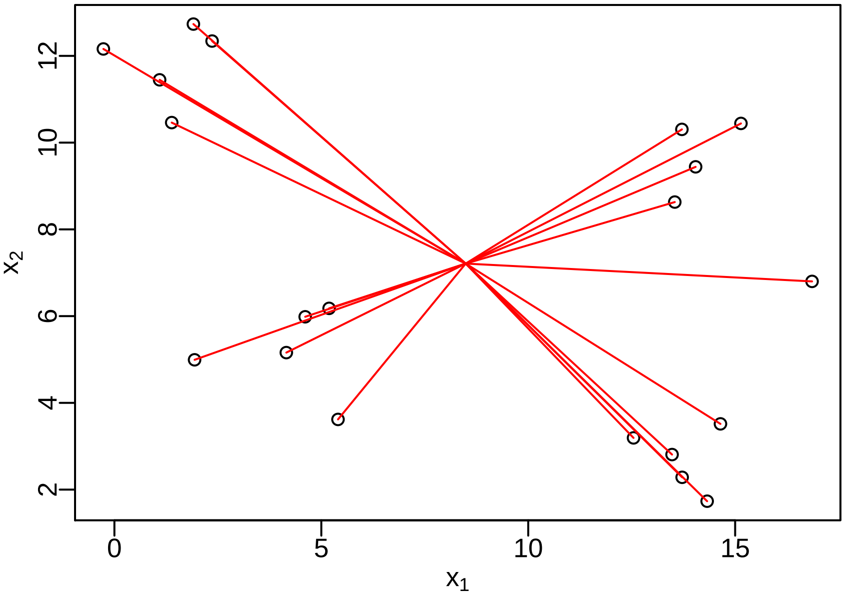

FIGURE 9.

Problem. Compute SSW, SSB and SST for the given set of points and the previously computed cluster centres.

```
##      [,1] [,2]
## [1,]    1    1
## [2,]    2    1
## [3,]    4    5
```

Compare to the SSW, SSB and SST using the initial cluster centres.

```
##        [,1] [,2]
## [1,]    1.5  1
## [2,]    4    5
```

Choosing the number of clusters.

Choosing the number of Clusters. K-means requires us to set the number of clusters before we run the algorithm.

To decide on the number of clusters, we must remember that a good clustering will have:

- Small SSW (all points within a cluster are close), and
- Large SSB (all points in different clusters are distant)

Therefore, we can examine the SSW for clusterings with 2, 3, 4, ..., k clusters, and choose the clustering with the smallest SSW.

Unfortunately, as we increase the number of clusters SSW will decrease, so SSW will always be the smallest for k clusters.

Let's examine the change in SSW as we increase the number of clusters.

Comparing SSW and SSB. For the previous set of points, we have SSB and SSW:

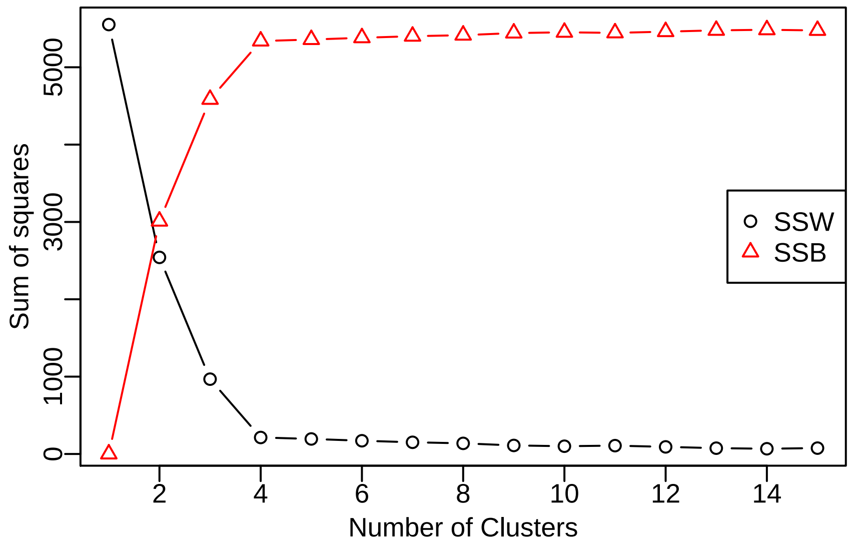

FIGURE 10.

Elbow method. When choosing the number of clusters:

- If we increase the number of clusters, we reduce SSW and increase SSB.
- If we choose to have too many clusters, the clustering becomes worthless.

How do we trade off between minimising the number of clusters and minimising SSB?

Elbow method. The number of clusters is provided by the clustering where the reduction of SSW slows (the elbow bend).

Elbow method example. The number of clusters is given by the position of the *elbow* bend.

Elbow method problems. How many clusters should we find based on these elbow plots?

Forcing Euclidean Distance.

K-means with non-Euclidean distance. We stated that K-means identifies clusters where Euclidean distance is used.

But what if our data metric is not Euclidean distance?

We project our data into a Euclidean space using Multi-dimensional Scaling (MDS)!

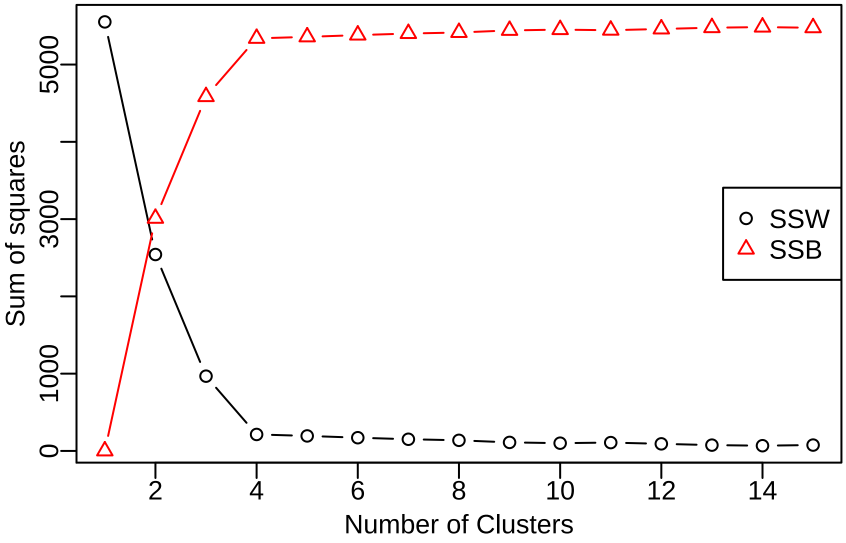

FIGURE 11.

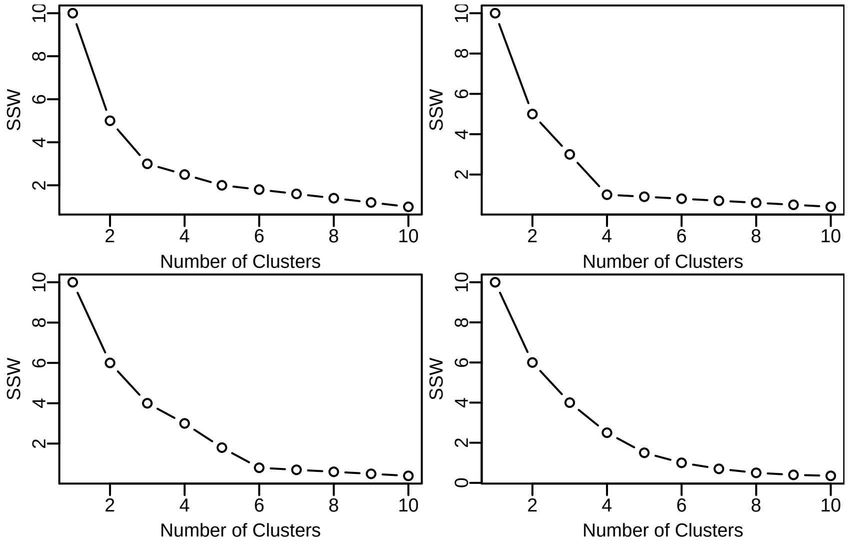

FIGURE 12.

K-means with non-Euclidean metric. We previously used MDS to visualise our data by projecting the data into a 2D Euclidean space (space using Euclidean distance), but remember that projecting into a small dimensional space may lead to data loss.

We can project our data into a *higher dimensional Euclidean space using MDS* to minimise the data loss, then use K-means on the projected data.

We will explore this in the lab.

3. Hierarchical Clustering

Hierarchy of Points. A hierarchy shows the order of its items, usually in the form of a tree. The higher an item is in the tree, the more important it is.

Hierarchical clustering provides us with a tree of clusters, called a *dendrogram* that shows the division of clusters.

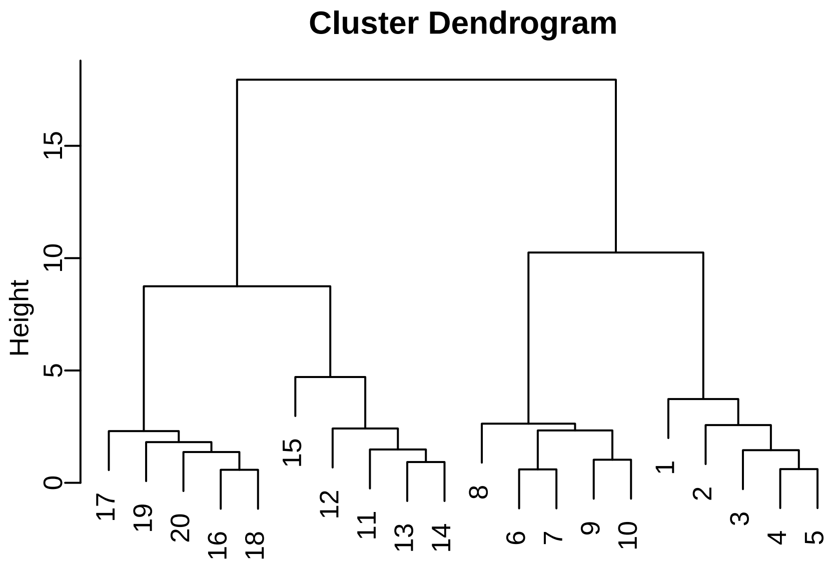

FIGURE 13.

Hierarchical Clustering Example.

Types of Hierarchical Clustering.

Agglomerative Clustering (bottom up).

1. Treat all points as clusters
2. Iteratively merge the closest clusters, until we are left with one cluster of all points.

Divisive Clustering (top down).

1. Treat all points as being in one cluster
2. Iteratively split the cluster with the largest gap, until all points are in their own cluster.

We will not further examine Divisive Clustering in this Unit.

Types of Agglomerative Clustering. Agglomerative clustering requires us to select the two most similar clusters and merge them. There are many ways which we can define cluster similarity, therefore there are many forms of agglomerative clustering.

We will examine:

- Single Linkage Clustering
- Complete Linkage Clustering
- Group Average Clustering

Each of these methods begin with all points as a cluster, then the clusters are merged one by one until we have one cluster of all points.

Single Linkage Clustering. Single linkage clustering defines the distance between clusters G and H as:

$$d_{\text{SL}}(G,H) = \min_{i \in G, j \in H} d_{ij}$$

where d_{ij} is the distance between points $\vec{x}_i$ and $\vec{x}_j$.

The distance between two clusters is the *minimum distance* over all pairs of points A and B, where point A belongs to one cluster and point B belongs to the other cluster.

Complete Linkage Clustering. Complete linkage clustering defines the distance between clusters G and H as:

$$d_{\text{CL}}(G,H) = \max_{i \in G, j \in H} d_{ij}$$

where d_{ij} is the distance between points $\vec{x}_i$ and $\vec{x}_j$.

The distance between two clusters is the *maximum distance* over all pairs of points A and B, where point A belongs to one cluster and point B belongs to the other cluster.

Group Average Clustering. Group average clustering defines the distance between clusters G and H as:

$$d_{\text{GA}}(G,H) = \frac{1}{N_G N_H} \sum_{i \in G} \sum_{j \in H} d_{ij}$$

where d_{ij} is the distance between points $\vec{x}_i$ and $\vec{x}_j$, and N_G and N_H are the number of points in clusters G and H respectively.

The distance between two clusters is the *average distance* over all pairs of points A and B, where point A belongs to one cluster and point B belongs to the other cluster.

Single Linkage Example. Let's find the hierarchy of clusters using the following set of points and Euclidean distance.

Complete Linkage Problem.

Problem. Find the hierarchy of clusters from the following plot using complete linkage.

Single Linkage using a Distance Matrix. We saw that Multidimensional Scaling is a powerful dimension reduction method because it works off the distance matrix, rather than on the points themselves.

Single Linkage Clustering also only requires the distance matrix.

To perform clustering:

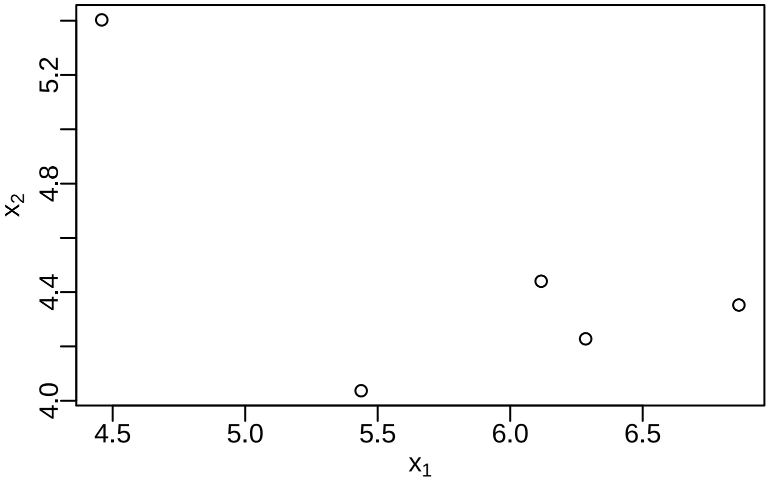

FIGURE 14.

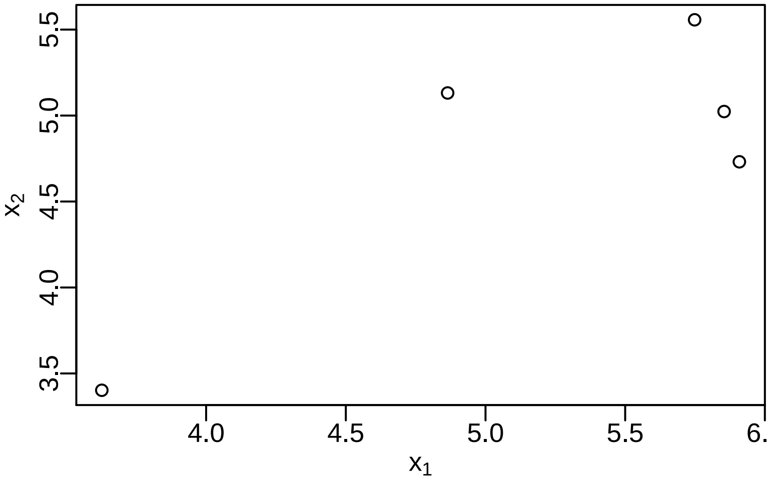

FIGURE 15.

1. Find the row i and column j associated to the smallest distance.
2. Merge the rows i and j and columns i and j by taking the minimum of each cell.
3. Repeat until we have 1 cluster.

Single Linkage Distance Matrix Example. The distance between points x_1, x_2, x_3, x_4 and x_5.

	x_1	x_2	x_3	x_4	x_5
x_1	0	1	7	5	6
x_2	1	0	4	8	6
x_3	7	4	0	2	8
x_4	5	8	2	0	3
x_5	6	6	8	3	0

The table is symmetric since the distance between x_i and x_j is the same as the distance between x_j and x_i.

1. Find the smallest distance (the closest points).
2. Merge the associated points.

Single Linkage Distance Matrix Problem.

Problem. Find the hierarchy of clusters from the following dissimilarity matrix using single linkage clustering.

	x_1	x_2	x_3	x_4	x_5
x_1	0	3	7	8	9
x_2	3	0	5	7	6
x_3	7	5	0	2	1
x_4	8	7	2	0	3
x_5	9	6	1	3	0

Summary. From this lecture, we have learnt:

- Clustering allows us to summarise data
- We can cluster data using a clustering algorithm; we examined K-means and two forms of Hierarchical clustering.
- The clusters obtained depend on the distance between the objects, therefore we must measure the distance appropriately (choose the right metric).
- We must choose the number of clusters for k-means clustering, but we are provided all clusters using hierarchical clustering.

Next Week. Graphs 1: Introduction to Graphs and Their Parameters