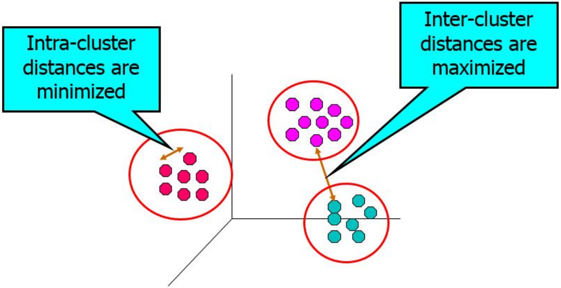
Machine learning

Clustering

The process of grouping a set of physical or abstract objects into classes of similar objects is called **clustering.**

A cluster is a collection of data objects that are similar to one another within the same cluster and are dissimilar to the objects in other clusters.



Intra-cluster distances are minimized
Inter-cluster distances are maximized

Clustering distance measures

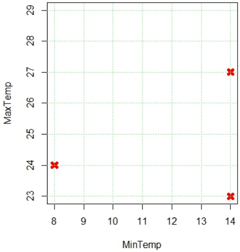
The classification of observations into groups requires some methods for computing the **distance or the (dis)similarity** between each pair of observations. The result of this computation is known as a **dissimilarity or distance matrix**.

- The choice of distance measures is a critical step in clustering.
- It defines how the similarity of two elements (x, y) is calculated and it will influence the shape of the clusters.

Clustering distance measures

Let us consider the following example

```
weather<-read.csv("http://rattle.togaware.com/weather.csv")
(x <- round(weather$MinTemp[1:3]))              #x=  8 14 14
(y <- round(weather$MaxTemp[1:3]))              #y= 24 27 23
plot(x, y, ylim=c(23, 29),pch=4,
     lwd=5, xlab="MinTemp",ylab="MaxTemp", col = "red")
grid(col = "lightgreen")
```



How can we measure the distances between the 3 observations?

Clustering distance measures

Measuring the distance of two observations requires taking a measure of the distance between two objects **x** and **y** represented by points in a multidimensional space.

The mathematic requirements of a distance function:

- $d(x, y) \geq 0$ is a nonnegative number.
- $d(x, x) = 0$ the distance of an observation to itself is 0.
- $d(x, y) = d(y, x)$ is a symmetric function.
- $d(x, y) \leq d(x, c) + d(c, y)$ the distance from observation x to observation z is at most as large as the sum of the distance from observation x to observation c and the distance from observation c to observation z.(triangular inequality).

Clustering distance measures

Suppose the observations are described by a set of continuous attributes, then the classical methods for distance measures are Euclidean and Manhattan distances:

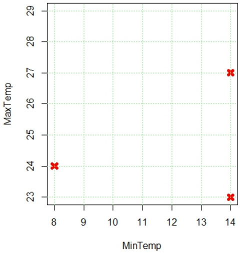
- **Euclidean distance**

$$D\left(x_i, x_j\right) = \sqrt{\sum_{k=1}^{n} \left(x_{ik} - x_{jk}\right)^2};$$

- **Manhattan distance**

$$D\left(x_i, x_j\right) = \sum_{k=1}^{n} \left|x_{ik} - x_{jk}\right|;$$

Clustering distance measures



	MinTemp	MaxTemp
1	8	24
2	14	27
3	14	23

$x_1 = (8, 24); x_2 = (14, 27); x_3 = (14, 23);$

- **Euclidean distance** $D(x_1, x_2) = 6.7; D(x_1, x_3) = 6.1; D(x_2, x_3) = 4;$
- **Manhattan distance** $D(x_1, x_2) = 9; D(x_1, x_3) = 7; D(x_2, x_3) = 4;$

```
x1<-c(8,24)

x2<-c(14,27)

x3<-c(14,23)

df3<-rbind(x1,x2,x3)

dist(df3)

dist(df3, method = "manhattan")
```

Clustering distance measures

What type of distance measures should we choose?

- The choice of distance measures is very important, as it has a strong influence on the clustering results.
- For most common clustering software, the default distance measure is the Euclidean distance.

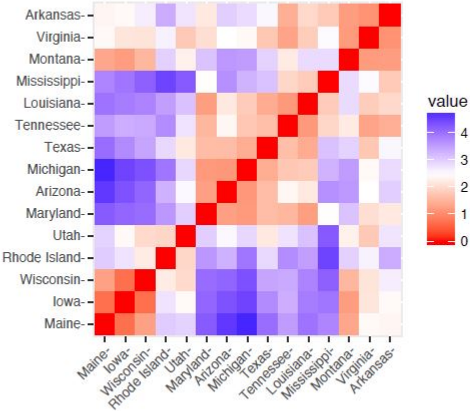
Distance matrix computations

There are many R functions for computing distances between pairs of observations:

- **dist()** R base function [stats package]: Accepts only numeric data as an input.
- **get_dist()** function [factoextra package]: Accepts only numeric data as an input.
- **daisy()** function [cluster package]: Able to handle other variable types (e.g. nomminal, ordinal, (a)symmetric binary). In that case, of the Gower coefficient will be automatically used as the metric. It is one of the most popular measures of proximity for mixed data types.

Visualizing distance matrices

A simple solution for visualizing the distance matrices is to use the function **fviz_dist()** [factoextra package].



The color level is proportional to the value of the dissimilarity between observations: pure red if $dist(x_i, x_j) = 0$ and pure blue if $dist(x_i, x_j) = 1$. Objects belonging to the same cluster are displayed in consecutive order.

Clustering distance measures

Suppose the observations have attributes: drug dose, temperature. For three patients we have

$$x_1 = (5000, 36.6); x_2 = (5000, 39); x_3 = (4000, 39);$$

Obviously, patients x_1 i x_2 are close. Changing the dose record from mg to g

$$x_1 = (5, 36.6); x_2 = (5, 39); x_3 = (4, 39);$$

brings about changes in the cluster, bringing about x_2 to x_3.

The value of distance measures is intimately related to the scale on which measurements are made.

Transforming the Data

- With **normalization**, we transform the original data from the measured units to a interval [0,1].

$$x^* = \frac{x - \min(x_i)}{\max(x_i) - \min(x_i)}.$$

- With **z-score, or zero mean normalization**, we transform the original data from the measured units to a normalized values that have a **mean of 0** and **a standard deviation of 1**.

$$x^* = \frac{x - \bar{x}}{\sigma},$$

Transforming the Data

```
v=weather$MinTemp[1:3]
scale(v)
mean(scale(v))
sd(scale(v))

normalize <- function(x) {
  return((x - min(x)) / (max(x) - min(x)))
}
normalize(v)
```

Dissimilarity between symmetric binary variables

A binary variable is **symmetric** if both of its states are equally valuable and carry the same weight. There is no preference on which outcome should be coded as 0 or 1.

Dissimilarity (or distance) measure between observation x_i and x_j we define as follows

$$D(x_i, x_j) = \frac{r+s}{p}$$

where

		obs.	x_j	
		1	**0**	*suma*
obs.	**1**	q	r	$q+r$
x_i	**0**	s	t	$s+t$
	suma	$q+s$	$r+t$	p

and $p = q + r + s + t$.

Dissimilarity between symmetric binary variables

patient	gender	fever	*testHIV*	*testHCV*
Kate	*f*	*T*	*T*	*F*
John	*m*	*T*	*T*	*F*
Mary	*f*	*T*	*F*	*T*

$\Downarrow$

patient	gender	fever	*testHIV*	*testHCV*
Kate	1	1	1	0
John	0	1	1	0
Mary	1	1	0	1

	1	0	
1	q	r	$q+r$
0	s	t	$s+t$
	$q+s$	$r+t$	p

$$D(x_i, x_j) = \frac{r+s}{p}$$

Dissimilarity between symmetric binary variables

patient	gender	fever	*testHIV*	*testHCV*
Kate	1	1	1	0
John	0	1	1	0
Mary	1	1	0	1

	1	0	
1	q	r	$q+r$
0	s	t	$s+t$
	$q+s$	$r+t$	p

$$D(x_i, x_j) = \frac{r+s}{p}$$

- for Kate and John

	1	0	
1	2	1	3
0	0	1	1
	2	2	4

$$D(Kate, John) = \frac{1+0}{4} = \frac{1}{4}$$

- for Kate and Mary; for John and Mary

$$D(Kate, Mary) = \frac{2}{4} = \frac{1}{2}; D(John, Mary) = \frac{3}{4}$$

```
kate<-c(1,1,1,1)
john<-c(0,1,1,0)
mary<-c(1,1,0,1)
patients<-
  rbind(kate,john,mary)
dist(patients,
 method = "binary")
```

Dissimilarity between asymmetric binary variables

A binary variable is **asymmetric** if the outcomes of the states are not equally important, such as the positive and negative outcomes of a disease test. By convention, we shall code the most important outcome, which is usually the rarest one, by 1 (e.g., HIV positive) and the other by 0 (e.g., HIV negative). Given two asymmetric binary variables, the agreement of two 1s (a positive match) is then considered more significant than that of two 0s (a negative match). Therefore, such binary variables are often considered “monary” (as if having one state).

The dissimilarity based on such variables is called asymmetric binary dissimilarity between observation x_i and x_j, where the number of negative matches, t, is considered unimportant and thus is ignored:

$$D(x_i, x_j) = \frac{r+s}{q+r+s}$$

		obs.	x_j	
		1	0	*suma*
obs.	1	q	$\mathbf{r}$	$q+r$
x_i	0	$\mathbf{s}$	t	$s+t$
	suma	$q+s$	$r+t$	p

Dissimilarity between asymmetric binary variables

patient	gender	fever	*testHIV*	*testHCV*
Kate	1	1	1	0
John	0	1	1	0
Mary	1	1	0	1

- for Kate and John

	1	0	
1	2	1	3
0	0	1	1
	2	2	4

$$D(Kate, John) = \frac{1+0}{3} = \frac{1}{3}$$

- for Kate and Mary; for John and Mary

$$D(Kate, Mary) = \frac{2}{4} = \frac{1}{2}; D(John, Mary) = \frac{3}{4}$$

	1	0	
1	q	$\mathbf{r}$	$q+r$
0	$\mathbf{s}$	t	$s+t$
	$q+s$	$r+t$	p

$$D(x_i, x_j) = \frac{r+s}{q+r+s}$$

Categorical Variables

A categorical variable is a generalization of the binary variable in that it can take on more than two states. For example, map color is a categorical variable that may have, say, five states: red, yellow, green, pink, and blue.

The dissimilarity between two objects (x_i and x_j can be computed based on the ratio of mismatches:

$$D(x_i, x_j) = \frac{p - m}{p} = \frac{n}{p}$$

where **p** is the total number of variables, **m** is the number of matches (i.e., the number of variables for which (x_i and x_j are in the same state).

Variables of Mixed Types

So, how can we compute the dissimilarity between objects of mixed variable types?

Gower metrics is technique that combines the different variables into a single dissimilarity matrix, bringing all of the meaningful variables onto a common scale of the interval [0.0,1.0]

- the metric takes values from 0 to 1;
- from all variables a linear combination is built using weights (usually just averages);
- **advantages** is intuitive and simple to calculate;
- **disadvantage** is sensitive to distributions that are not normal and outliers present in continuous variables.

Variables of Mixed Types

$$s_{ij} = \frac{\sum_{k=1}^{N} w_{ijk}^{k} s_{ij}^{k}}{\sum_{k=1}^{N} w_{ij}^{k}}$$

where

- w_{ijk}^{k} weighting between observations i and j taking into account column k;
- s_{ijk}^{k} distance between observations i and j taking into account column k

$$s_{ijk}^{k} = \begin{cases} 1 \text{ jeżeli } x_{ik} \neq x_{jk} \\ 0 \text{ jeżeli } x_{ik} = x_{jk} \end{cases} \quad \text{for } \textbf{Categorical Variables}$$

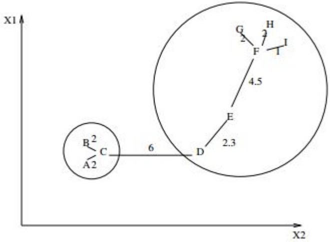
$$s_{ijk}^{k} = \frac{\|x_{ik} - x_{jk}\|}{r_k} \quad \textbf{for numeric variables}$$

and $r_k = max(x_k) - min(x_k)$.

Clustering methods

The major clustering methods can be classified into the following categories:

- **partitioning methods**
- **hierarchical methods**
- **model-based methods**



Partitioning methods

Given a database **D** of n objects or data tuples.

- Partitioning method constructs k partitions of the data, where each partition represents a **cluster** and $k \leq n$.
- That is, it classifies the data into k groups, which together satisfy the following requirements:
    - each group must contain at least one object,
    - each object must belong to exactly one group.
- Given k, the number of partitions to construct, a partitioning method creates an initial partitioning. It then uses an iterative relocation technique that attempts to improve the partitioning by moving objects from one group to another.

The general **criterion of a good partitioning** is that objects in the same cluster are **close or related to each other**, whereas objects of different clusters are **far apart or very different**

.

Partitioning methods

Main heuristic methods:

- the **k-means algorithm**, where **each cluster is represented by the mean value of the objects in the cluster**, and
- the **k-medoids algorithm**, where **each cluster is represented by one of the objects located near the center of the cluster**.

These heuristic clustering methods work well for finding spherical-shaped clusters in small to medium-sized databases.

The k-Means Method

The k-means algorithm takes the input parameter, **k**, and partitions a set of n objects into k clusters so that the resulting intracluster similarity is high but the intercluster similarity is low.

The k-Means Method

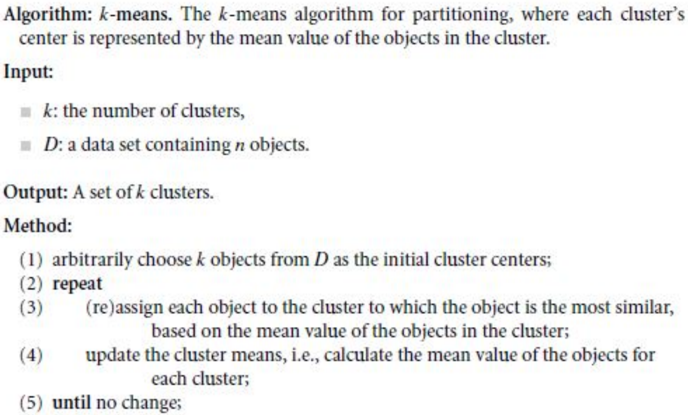
How does the k-means algorithm work?

- randomly selects k of the objects, each of which initially represents a cluster mean or center.
- for each of the remaining objects, an object is assigned to the cluster to which it is the most similar, based on the distance between the object and the cluster mean.
- it then computes the new mean for each cluster.
- this process iterates until the criterion function converges. Typically, the square-error criterion is used.

$$tot.withinss = WCSS = \sum_{i=1}^{k} \sum_{p \in C_i} d\left(p, m_i\right)^2 ,$$

where p is the point in space representing a given object; and m_i is the mean of cluster C_i (both p and m_i are multidimensional). In other words, **for each object in each cluster, the distance from the object to its cluster center is squared, and the distances are summed.**

The k-Means Method



Algorithm: k**-means.** The k-means algorithm for partitioning, where each cluster's center is represented by the mean value of the objects in the cluster.

Input:

- k: the number of clusters,
- D: a data set containing n objects.

Output: A set of k clusters.

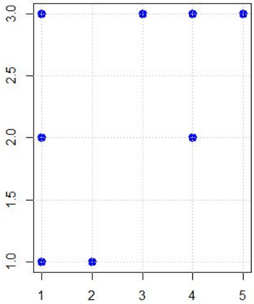
Method:

```
arbitrarily choose $k$ objects from $D$ as the initial cluster centers;
repeat
    (re)assign each object to the cluster to which the object is the most similar,
            based on the mean value of the objects in the cluster;
    update the cluster means, i.e., calculate the mean value of the objects for
            each cluster;
until no change;
```

The k-Means Method

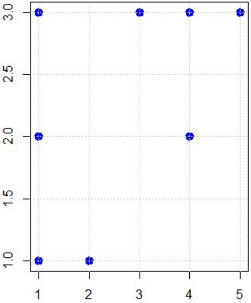
Suppose, that we have eight observations in the table below and we are interested in discovering $k = 2$ groups

a	(1,3)
b	(3,3)
c	(4,3)
d	(5,3)
e	(1,2)
f	(4,2)
g	(1,1)
h	(2,1)



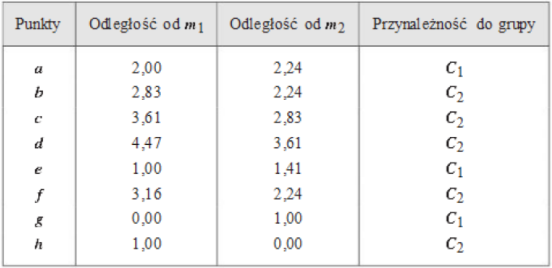
The k-Means Method

We randomly assign k observations as the initial centres of the groups. In this example, we assign the group centres as $m_1 = (1,1), m_2 = (2,1)$.



The k-Means Method

- for each observation, we search for the nearest centre of the group. The table below contains the (rounded) Euclidean distances between each point and each centre of the group $(C_1)m_1 = (1,1)$ i $(C_2)m_2 = (2,1)$, together with an indication of which measure is closer. Therefore **group 1** contains points **a,e,g**. **Group 2** contains points **b,c,d,f,h**.



Punkty	Odległość od m_1	Odległość od m_2	Przynależność do grupy
a	2,00	2,24	C_1
b	2,83	2,24	C_2
c	3,61	2,83	C_2
d	4,47	3,61	C_2
e	1,00	1,41	C_1
f	3,16	2,24	C_2
g	0,00	1,00	C_1
h	1,00	0,00	C_2

The k-Means Method

Once group membership has been assigned, the sum of squared errors can be calculated.

$$\begin{aligned} WCSS &= \sum_{i=1}^{k} \sum_{p \in C_{\mathbf{1}}} d\left(p, m_i\right)^2 = 2^2 + (2.24)^2 + (2.84)^2 + (3.61)^2 \\ & \quad +1^2 + (2.24)^2 + 0^2 + 0^2 = 36 \end{aligned}$$

As we noted earlier, we would like our clustering approach to **maximises the variability between groups relative to the variability within the group.**

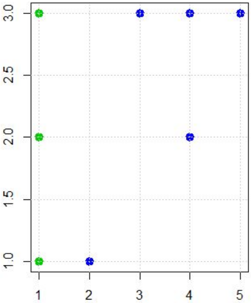
- as **BCV (variability between groups - ang. between-cluster variation)** let's denote $d(m_1, m_2)$
- as**WCV (internal group variability - ang, within-cluster variation)** let's denote WCSS

$$\frac{BCV}{WCV} = \frac{d(m_1, m_2)}{SSE} = \frac{1}{36} = 0.0278.$$

We expect this ratio to increase in subsequent iterations.

The k-Means Method

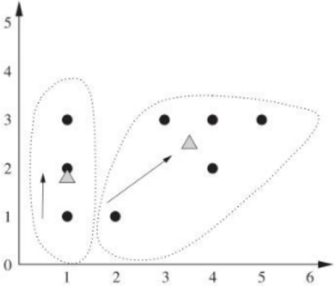
- for each cluster k, find its new centre. In our case, for a cluster 1 (C_1) is $m_1 = \left(\frac{1+1+1}{3}, \frac{3+2+1}{3}\right) = (1, 2)$ and for cluster 2 $m_2 = \left(\frac{3+4+5+4+2}{5}, \frac{3+3+3+2+1}{5}\right) = (3.6, 2.4)$



Groups and measures after the first iteration.

The k-Means Method

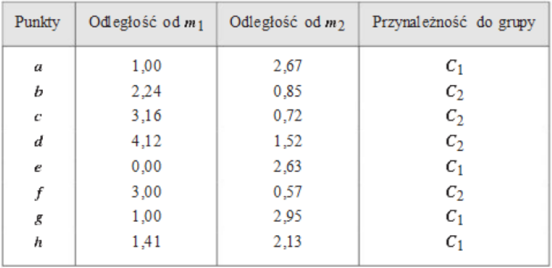
- the groups and centres (triangles) at the end of the first iteration are shown in the figure.
- note that m_1 has moved from the point of $(1, 1)$ to $(1, 2)$, dupwards to the centre for three points in group 1.
- in the meantime, the second centre m_2 has moved from $(2, 1)$ to $(3.6, 2.4)$, to the up and to the right.



Groups and measures after the first iteration.

The k-Means Method

- we repeat steps 3 and 4 until convergence or completion. In our case, the means have been shifted, so we return to step 3 in the second iteration of the algorithm.
- for each observation, we determine the nearest centre of the group. The table shows the distances between each point and each updated centre of the group $m_1 = (1, 2)$, $m_2 = (3.6, 2.4)$.

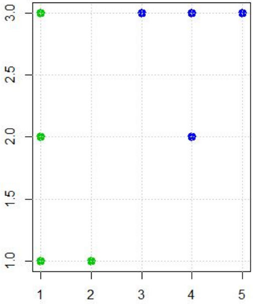


Punkty	Odległość od m_1	Odległość od m_2	Przynależność do grupy
a	1,00	2,67	C_1
b	2,24	0,85	C_2
c	3,16	0,72	C_2
d	4,12	1,52	C_2
e	0,00	2,63	C_1
f	3,00	0,57	C_2
g	1,00	2,95	C_1
h	1,41	2,13	C_1

Points in the second iteration

The k-Means Method

- Note that we have shifted single record (h) from group 2 to group 1. The relatively large change in the value of m_2 has meant that observation h is now closer to m_1 than m_2, therefore h now belongs to group 1. All the other observations remain in the same groups as before. Therefore, Therefore, **group 1 is a,e,g,h,** and **group 2 is b,c,d,f**.



Points in the second iteration

The k-Means Method

The new value of the sum of squared errors **(intra-group variability)** is

$$WCSS = 7.86,$$

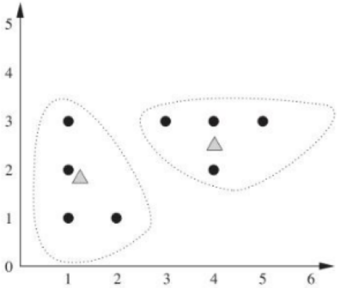
and therefore less than before. This means that we have found a better solution to the clustering task. Furthermore,

$$\frac{BCV}{WCV} = \frac{d\left(m_1, m_2\right)}{WCSS} = \frac{2.63}{7.86} = 0.3346,$$

has a higher value than before.

The k-Means Method

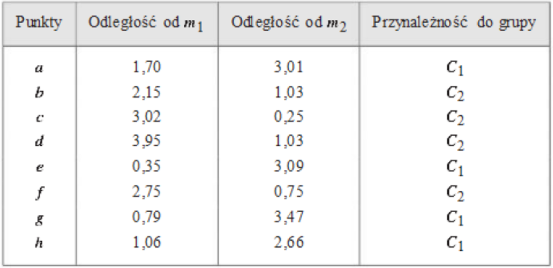
- For each of the k groups, we find a new centre of the group. The new measure for group 1 is $[(1+1+1+2)/4, (3+2+1+1)/4] = (1.25, 1.75)$. The new measure for group 2 is $[(3+4+5+4)/4, (3+3+3+2)/4] = (4; 2.75)$. The groups and centers at the end of the second iteration are shown in the figure. The centres m_1 and m_2 have been slightly shifted.



Groups and centres after the second iteration.

The k-Means Method

- the centres have been moved, so we return to step 3 in the third (as it will turn out the last) iteration of the algorithm.
- for each observation, we find the nearest centre of the group. The table shows the distances between the points and each newly updated group centre $m_1 = (1.25; 1.75)$ and $m_2 = (4; 2.75)$, together with the group membership. Note that the observations did not change the group membership from the previous iteration.



Punkty	Odległość od m_1	Odległość od m_2	Przynależność do grupy
a	1,70	3,01	C_1
b	2,15	1,03	C_2
c	3,02	0,25	C_2
d	3,95	1,03	C_2
e	0,35	3,09	C_1
f	2,75	0,75	C_2
g	0,79	3,47	C_1
h	1,06	2,66	C_1

Third iteration.

The k-Means Method

New value for the sum of squared errors

$$WCSS = 6.23$$

is slightly less than the previous WCSS value of 7.86, which indicates the closeness of the best clustering solution. In addition, we have

$$\frac{BCV}{WCV} = \frac{d\left(m_1, m_2\right)}{WCSS} = \frac{2.97}{6.23} = 0.4703,$$

and therefore more than the previous value of 0.3346. This indicates an increase in inter-group variability relative to the variability within a group.

The k-Means Method

- for each of the k groups, we look for the centre of the group.
- since no observation has changed group membership, the centres of the groups will also remain unchanged.

Determining the Optimal Number of Clusters

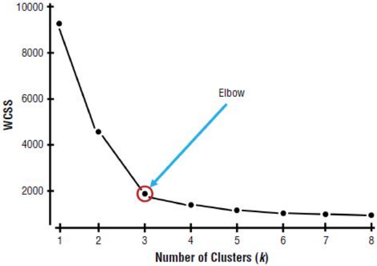
Determining the optimal number of clusters in a data set is a fundamental issue in partitioning clustering, which requires the user to specify the number of clusters k to be generated.

The optimal number of clusters is somehow subjective and depends on the method used for measuring similarities and the parameters used for partitioning.

Determining the Optimal Number of Clusters- Elbow method

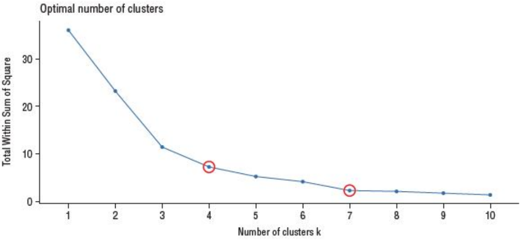
$$WCSS = \sum_{i=1}^{k} \sum_{p \in C_i} d\,(p, m_i)^2\,,$$

- the closer the elements in the cluster are to the centre, the smaller the WCSS value.
- the smaller the WCSS value, the more the elements in the cluster are similar to each other.
- if the value of k increases, then the elements in each cluster move closer towards the centre and the total WCSS decreases.



Elbow method

- if we increase the value of k, then the WCSS value decreases, but the the reduction in the WCSS value achieved for each unit increase in the value of k also decreases.
- at some point in the curve, a visible bend occurs that represents the point at which increasing the value for k no longer yields a significant reduction in WCSS. This point is known as the elbow, and the k value at this point is usually expected to be the appropriate number of clusters for the dataset.



$$WCSS = \sum_{i=1}^{\mathbf{k}} \sum_{p \in C_i} d\left(p, m_i\right)^2$$

Average silhouette method

The silhouette s_x for observation x is defined as follows

$$s_x = \frac{b(x) - a(x)}{max\{a(x), b(x)\}}$$

where

- $a(x)$ denotes the average measure of dissimilarity (distance) between x and all observations from the same group (the smaller the value the better the fit of the observations to the group),
- $b(x)$ denotes the minimum from the mean distances between an observation x and other clusters.

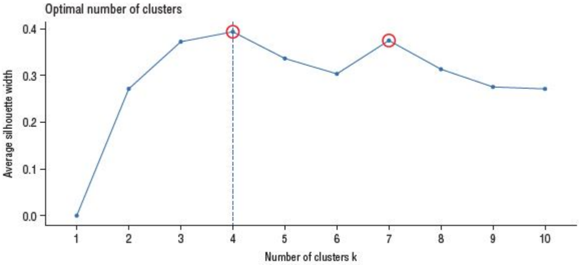
Average silhouette method computes the average silhouette of observations for different values of k. The optimal number of clusters k is the one that maximize the average silhouette over a range of possible values for k. A value close to 1 indicates that the observation is in the correct cluster, while a value close to -1 indicates that it is in the wrong cluster.

Average silhouette method

Average silhouette $\bar{s}_k$ for each cluster k is equal $\bar{s}_k = mean(s_x)$

Average silhouette of all clusters $S_{avg} = \frac{1}{m} \sum_{k=1}^{m} s_k$

Silhouette coefficient is of the form $SC = max_k S_{avg}$

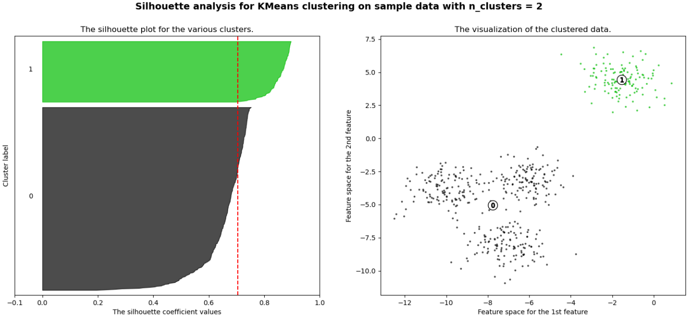


Average silhouette method

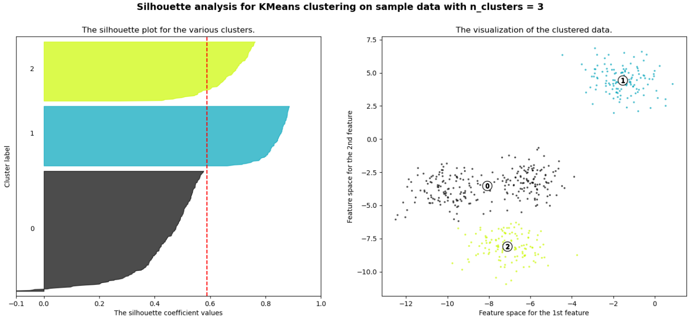
- In this example the silhouette analysis is used to choose an optimal value for **nclusters**.
- The silhouette plot shows that the **nclusters** value of **3, 5 and 6** are a bad choice for the data given, due to
  - the presence of clusters with silhouette scores below the mean;
  - large variations in the size of the silhouette graphs.
- Silhouette analysis suggests a number of clusters between 2 and 4.

Based on the thickness of the silhouette graph, the size of the cluster can be visualised. The silhouette graph when nclusters is equal to 2, has a larger size due to the grouping of 3 sub-clusters into one large cluster. On the other hand, when nclusters is equal to 4, all graphs are more or less similar in thickness and therefore size, which can also be seen in the scatter plot on the right.

Average silhouette method



Average silhouette method

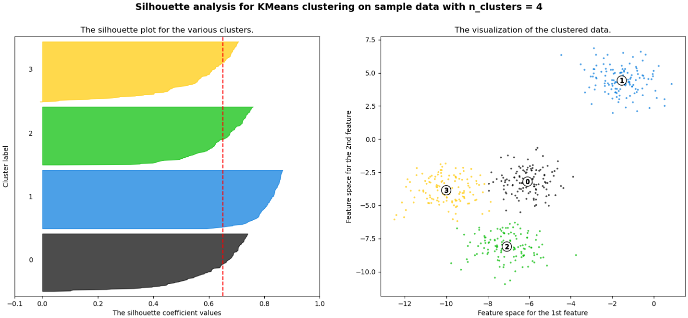


Silhouette analysis for KMeans clustering on sample data with n_clusters = 3

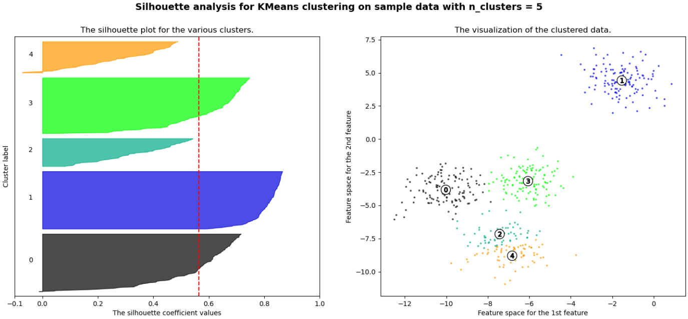
The silhouette plot for the various clusters.

The visualization of the clustered data.

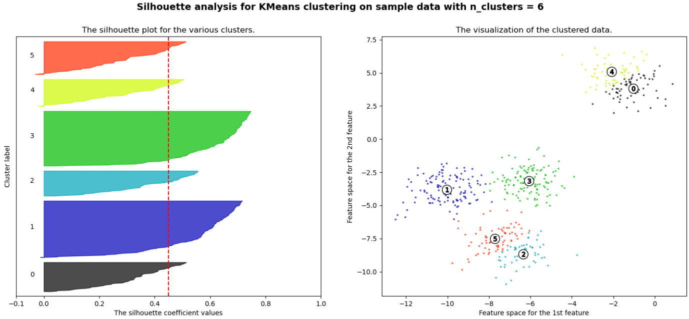
Average silhouette method



Average silhouette method



Average silhouette method



Silhouette analysis for KMeans clustering on sample data with n_clusters = 6

The silhouette plot for the various clusters.

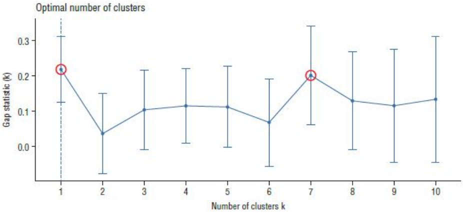
The visualization of the clustered data.

Gap Statistic

The Gap Statistic compares the difference between clusters created from the observed data and clusters created from a randomly generated dataset, known as the reference dataset.

For a given k, the gap statistic is the difference in the total WCSS for the observed data and that of the reference dataset.

The optimal number of clusters is denoted by the k value that yields the largest gap statistic.



The Random Initialization Trap

- the initial cluster centers don't have to represent actual points in the original dataset.
- the final set of clusters is sensitive to the location of the initial set of cluster centers.
- this means that we could run the k-means clustering process several times and end up with different looking clusters each time, depending on the choice of initial cluster centers.
- **This is known as the random initialization trap.**

The **k-means++** method assume to choose an initial set of cluster centers that are as far away as possible from each other. By doing this, we minimize the impact of randomness on the final clusters.

Strengths and Weaknesses of k-Means Clustering

the strengths:

- this approach is also flexible - simply changing the the value of k to change the number of subgroups into which the observations are grouped.
- the underlying mathematical principles behind k-means clustering (such as Euclidean distance) are not difficult to understand, so is commonly used.

the weaknesses:

- requires that the value for k be set by the user.
- because distance can be calculated only between numeric values, k-means clustering works only with numeric data.
- the k-means algorithm is not good at modeling clusters that have a complex geometric shape (nonspherical clusters).
- the use of random or pseudorandom initial centroids means that the approach, to some extent, relies on chance.

Thank you for your attention!!!