Department of Statistics
Lund University

Data Mining
Fall

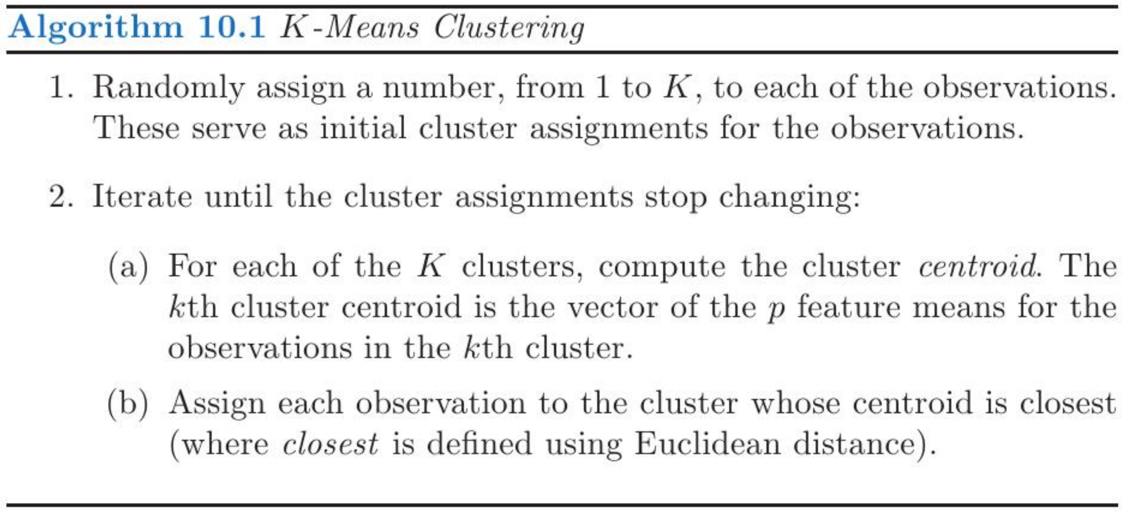
# Project 4: Unsupervised learning

*Clustering methods*

*Peform all requested task. Some useful R-code that can help in completing Project 4 can be found through the Data for Download link in our webpage.*

**Part One – $K$-clustering**

$K$-means algorithm is performed according to the following scheme



**Algorithm 10.1** *$K$-Means Clustering*

1. Randomly assign a number, from 1 to $K$, to each of the observations. These serve as initial cluster assignments for the observations.
2. Iterate until the cluster assignments stop changing:
   (a) For each of the $K$ clusters, compute the cluster *centroid*. The $k$th cluster centroid is the vector of the $p$ feature means for the observations in the $k$th cluster.
   (b) Assign each observation to the cluster whose centroid is closest (where *closest* is defined using Euclidean distance).

The function `kmeans()` performs K-means clustering in R and is part of the standard package. Use `help()` function to get some information about `kmeans()`.

(1) Simulated data in two dimensions that are made of two clusters obtained by shifting their means. Check on a scatterplot that the data indeed constitute two clusters.
(2) Perform $K$-means clustering with $K = 2$. Check what are results of applying function `kmeans()`.
(3) Plot the data, with each observation colored according to its cluster assignment. Does the $K$-means cluster data well?
(4) Check what happens if you repeat clustering with $K = 3$. Repeat this part several times without setting the seed to see how the clustering depends on the intial (random) choice of starting points.
(5) There is a special meaning to `nstart` argument (with default set to one). If a value of nstart greater than one is used, then K-means clustering will be performed using multiple (`nstart`) random assignments in Step 1 of Algorithm 10.1, and the `kmeans()` function will report only the best results. Compare using `nstart=1` to `nstart=20`.
(6) Use `km.out$tot.withinss`, which is the total within-cluster sum of squares, which we seek to minimize by performing $K$-means clustering, to quantify comparisons. The individual within-cluster sum-of-squares are contained in the vector `km.out$withinss`.

**Remark:** *It is advisable to run $K$-means clustering with a large value of* `nstart`*, such as 20 or 50, since otherwise an undesirable local optimum may be obtained. When performing $K$-means clustering, in addition to using multiple initial cluster assignments, it is also important to set a random seed using the* `set.seed()` *function. This way, the initial cluster assignments in Step 1 can be replicated, and the $K$-means output will be fully reproducible, which is important when reporting your finding to the public.*

## Part Two – Hierarchical clustering

The `hclust()` function implements hierarchical clustering in R . In the following example we use the simulated data from the previous part to plot the hierarchical clustering dendrogram using complete, single, and average linkage clustering, with Euclidean distance as the dissimilarity measure.

(1) Perform clustering observations using complete linkage. The `dist()` function is used to compute the $50 \times 50$ inter-observation Euclidean distance matrix.
(2) Use the `cutree()` function to determine the cluster labels for each observation associated with a given cut of the dendrogram.
(3) Correlation-based distance can be computed using the `as.dist()` function, which converts an arbitrary square symmetric matrix into a form that the `hclust()` function recognizes as a distance matrix. However, this only makes sense for data with at least three features since the correlation between any two points is always 1. Hence, you need to create a three-dimensional data set, before applying hierarchical clustering.

## Part Three – Clustering cities around the world

The following data (collected by the Economic Research Department of the Union Bank of Switzerland) represent economic indicators in 46 cities year 1991.

| City | Work | Price | Salary |
|---|---|---|---|
| Amsterdam | 1714 | 65.6 | 49.0 |
| Athens | 1792 | 53.8 | 30.4 |
| Bogota | 2152 | 37.9 | 11.5 |
| Bombay | 2052 | 30.3 | 5.3 |
| Brussels | 1708 | 73.8 | 50.5 |
| Buenos Aires | 1971 | 56.1 | 12.5 |
| Caracas | 2041 | 61.0 | 10.9 |
| Chicago | 1924 | 73.9 | 61.9 |
| Copenhagen | 1717 | 91.3 | 62.9 |
| Dublin | 1759 | 76.0 | 41.4 |
| Dusseldorf | 1693 | 78.5 | 60.2 |
| Frankfurt | 1650 | 74.5 | 60.4 |
| Geneva | 1880 | 95.9 | 90.3 |
| Helsinki | 1667 | 113.6 | 66.6 |
| Hong Kong | 2375 | 63.8 | 27.8 |
| Houston | 1978 | 71.9 | 46.3 |
| Johannesburg | 1945 | 51.1 | 24.0 |
| Kuala Lumpur | 2167 | 43.5 | 9.9 |
| Lagos | 1786 | 45.2 | 2.7 |
| Lisbon | 1742 | 56.2 | 18.8 |
| London | 1737 | 84.2 | 46.2 |
| Los Angeles | 2068 | 79.8 | 65.2 |
| Luxembourg | 1768 | 71.1 | 71.1 |
| Madrid | 1710 | 93.8 | 50.0 |
| Manila | 2268 | 40.0 | 4.0 |
| Mexico City | 1944 | 49.8 | 5.7 |
| Milan | 1773 | 82.0 | 53.3 |
| Montreal | 1827 | 72.7 | 56.3 |
| Nairobi | 1958 | 45.0 | 5.8 |
| New York | 1942 | 83.3 | 65.8 |
| Nicosia | 1825 | 47.9 | 28.3 |
| Oslo | 1583 | 115.5 | 63.7 |
| Panama | 2078 | 49.2 | 13.8 |
| Paris | 1744 | 81.6 | 45.9 |
| Rio de Janeiro | 1749 | 46.3 | 10.5 |
| Sao Paulo | 1856 | 48.9 | 11.1 |
| Seoul | 1842 | 58.3 | 32.7 |
| Singpore | 2042 | 64.4 | 16.1 |
| Stockholm | 1805 | 111.3 | 39.2 |
| Sydney | 1668 | 70.8 | 52.1 |
| Taipei | 2145 | 84.3 | 34.5 |
| Tel Aviv | 2015 | 67.3 | 27.0 |
| Tokyo | 1880 | 115.0 | 68.0 |
| Toronto | 1888 | 70.2 | 58.2 |
| Vienna | 1780 | 78.0 | 51.3 |
| Zurich | 1868 | 100.0 | 100.0 |

Data is available in the file `cities4.dat` We will use different *hierarchial clustering methods* to look a similarities between the cities.

We will use *agglomerative clustering methods* such as **(a) Single linkage**, **(b) Complete linkage** and **(c) Average linkage**.

(1) Load the data.
(2) Read help pages for two R-functions that can be used for the hierarchical data clustering: `dist()` and `hclust()`.
(3) Perform a 'complete linkage' cluster analysis, based on the variables Work, Price and Salary.
(4) Plot the dendrograms. Which of the methods seems to produce most readable clusters?