WORKSHOP CLUSTERING

K-MEANS

The data that is going to be used is the Iris dataset which is included in the R installation. This dataset contains data on 50 flowers for 3 species of iris. It gives the measurements in centimeters of the sepal length and width and petal length and width.

Load the data into R

> data(iris)

Since K-Means clustering only works with numerical data, we will check if the data contains any non-numerical columns

> str(iris)

This shows us that the data contains 150 observations, and 5 columns. The first 4 columns contain numerical data and the last column contains a non-numerical field.

Remove the last column and also scale the data. Scaling means each variable will now have a mean of zero and a standard deviation of one. Ideally, you want a unit in each coordinate to represent the same degree of difference. Scaling will make the standard deviation the unit of measurement in each column. This is done to avoid the clustering algorithm to depend on a single column with large values. For this use the **scale** function.

> iris.scaled <- scale(iris)

Check the difference after the scaling using the summary function

> summary(iris)

> summary(iris.scaled)

Since k-means use a random generator for starting points, set the random generator seed. This will ensure that results will be the same in this sample and the results will be the same when we rerun the algorithm.

> set.seed(12345)

Since we do not know how the data is partitioned and how many clusters would be good for this dataset, start clustering using any number of centers you think would be good.

Use the function **kmeans** for clustering and store the result in **iris.clusters**. See http://stat.ethz.ch/R-manual/R-devel/library/stats/html/kmeans.html for an overview of all the possible parameters, and the possible values of the result. Hint: nstart is how many times k-means will run with different starting cluster points. The outcome will be the run with the best result.

Results and properties of the clustering are stored in **iris.clustered**. Let us investigate what information this result contains.

EVALUATION

One of the most basic ways to evaluate is to check the number of samples falling in each clusters. This can be done by using the size component on the iris.clustered variable:

> iris.clustered$size

This will show you the number of samples in each cluster. If a cluster contains too little or to many samples it will be unlikely that this will be a good cluster. From the outcome can be seen that the clusters are ok, no cluster contains a high or low number of samples.

Let us check the clusters in more detail. For this use the **centers** component.

This will give insight on the location of the centers of the clusters. Since we scaled the original dataset, the data of the centers will be somewhere around zero.

Try some different parameters for the k-means, such as the number of clusters, nstart or maybe even try a different type of k-means clustering algorithm and compare the results.

Also try to use the dataset without scaling and check the results.

HIERARCHICAL CLUSTERING

First the pairwise distance matrices have to be calculated. You can use different methods. Choose one you like. See http://stat.ethz.ch/R-manual/R-devel/library/stats/html/dist.html

> iris.dist <- dist(iris.scaled, method=".....")

Next the hierarchical cluster can be created. For this use the **hclust** function. Use the **iris.dist** created pr eviously as dissimilarity structure for the hclust function. For more info on the parameters and result val ues, check https://stat.ethz.ch/R-manual/R-devel/library/stats/html/hclust.html

You can plot the dendogram using the **plot** function, where **iris.hc** is the result of the **hclust** function.

> plot(iris.hc, labels=FALSE, hang=-1)

Try out the hierarchical clustering with different parameters (for example the method) and see the differences.

The dendogram gives us an indication of where to split the data into clusters. To actually assign each sample to a cluster use **cutree**. This will cut a dendogram into several groups either by specifying the

desired number of groups (k) or the cut height(b). At least one of them must be specified. See http://stat.ethz.ch/R-manual/R-devel/library/stats/html/cutree.html

> hc.clust <- cutree(iris.hc, k=4)

This will give you a vector of where each sample is assigned to a cluster.

ELBOW METHOD

If the dataset is bigger (i.e. more variables) than it would become harder to manually investigate for the correct number of clusters. R is able to plot an elbow diagram for the clustering. So we will run k-means for different number of k and plot that in a diagram. First, the random generator is fixed and we will set the max number of k to 15. We will then use different k as input parameter for k-means and the tot.withinss is taken as result and stored in wss. sapply ensures that all the results are transformed into a vector. Last, the different results for k can be plotted in a diagram.

> set.seed(12345)

> k.max <- 15

> wss <- sapply(1:k.max, function(k){ kmeans(iris.scaled, k, nstart=15) $tot.withinss })

> plot(1:k.max, wss, type="b", pch=19, frame=FALSE, xlab="Number of clusters K", ylab="Total within-clusters sum of squares")

Determine the elbow point.

When the data was loaded into R, the last column was removed from our dataset since it was non-numerical. You may have noticed that this column contains a few different values. This column is actually the class of each record.

Run the K-Means algorithm and the Hierarchical clustering with your elbow point and compare the results of the algorithm with the classes of each record. See if the algorithm clusters each record correctly. Hint: use the function **table**, i.e. **table(datavector1, datavector2)**

EXTRA

Apply the kmeans clustering and hierarchical clustering on the customers.csv dataset to cluster the samples and try to find some meaning in the clustering. This dataset contains statistics about the purchase behavior. It can be found in the clustering handson folder.

> customers <- read.csv("filelocation", header=TRUE)

Hint: select appropriate columns, check if scaling is necessary, use elbow method to find a correct cluster number. Also plot a dendogram to look for the cluster number.

Attribute Information Customers Data Set:

1) FRESH: annual spending (m.u.) on fresh products (Continuous);
2) MILK: annual spending (m.u.) on milk products (Continuous);
3) GROCERY: annual spending (m.u.)on grocery products (Continuous);
4) FROZEN: annual spending (m.u.)on frozen products (Continuous)
5) DETERGENTS_PAPER: annual spending (m.u.) on detergents and paper products (Continuous)
6) DELICATESSEN: annual spending (m.u.)on and delicatessen products (Continuous);
7) CHANNEL: customersâ€™ Channel - Horeca (Hotel/Restaurant/Cafe) or Retail channel (Nominal)
8) REGION: customers Region â€“ Lisnon, Oporto or Other (Nominal)
Descriptive Statistics:

(Minimum, Maximum, Mean, Std. Deviation)
FRESH (3, 112151, 12000.30, 12647.329)
MILK (55, 73498, 5796.27, 7380.377)
GROCERY (3, 92780, 7951.28, 9503.163)
FROZEN (25, 60869, 3071.93, 4854.673)
DETERGENTS_PAPER (3, 40827, 2881.49, 4767.854)
DELICATESSEN (3, 47943, 1524.87, 2820.106)

REGION Frequency
Lisbon 77
Oporto 47
Other Region 316
Total 440

CHANNEL Frequency
Horeca 298
Retail 142
Total 440