Hochschule Konstanz
Fakultät Informatik

Data Science

Clustering

Aufgabe 1 K-Means und PCA

This is adopted from ILSR.

In this problem, you will generate simulated data, and then perform PCA and K-means clustering on the data.

a) Generate a simulated data set with 20 observations in each of three classes (i.e. 60 observations total), and 50 variables. Hint: There are a number of functions in R that you can use to generate data. One example is the `rnorm()` function; Be sure to add a mean shift to the observations in each class so that there are three distinct classes. Other functions you might need are `rbind()` and `matrix()`.

b) Perform PCA on the 60 observations and plot the first two principal component score vectors. Use a different color to indicate the observations in each of the three classes. If the three classes appear separated in this plot, then continue on to part (c). If not, then return to part (a) and modify the simulation so that there is greater separation between the three classes. Do not continue to part (c) until the three classes show at least some separation in the first two principal component score vectors.

c) Perform K-means clustering of the observations with K = 3. How well do the clusters that you obtained in K-means cluster- ing compare to the true class labels? Hint: You can use the table() function in R to compare the true class labels to the class labels obtained by clustering. Be careful how you interpret the results: K-means clustering will arbitrarily number the clusters, so you cannot simply check whether the true class labels and clustering labels are the same.

d) Perform K-means clustering with K = 2. Describe your results.

e) Now perform K-means clustering with K = 4, and describe your results.

f) Now perform K-means clustering with K = 3 on the first two principal component score vectors, rather than on the raw data. That is, perform K-means clustering on the 60×2 matrix of which the first column is the first principal component score vector, and the second column is the second principal component score vector. Comment on the results.

g) Using the scale() function, perform K-means clustering with K = 3 on the data after scaling each variable to have standard deviation one. How do these results compare to those obtained in (b)? Explain.

h) Look at the total within sum of squares of the clusters for varying number of k's. Which is the best number of k?

Aufgabe 2 Customer Segmentation

In customer segmentation, you try to get insight into your customer base by identifying homogenuous customer groups. The following data set hold information on wine customers and is taken from the book *Data Smart* (http://www.john-foreman.com/data-smart-book.html).

The original data set contains two tables, stored in a Excel-Sheet. The first data frame `offers` contains information about campaigns, each offering wine of a specific varietal (German: Rebsorte). The second data frame `transactions` contains the transaction data describing which customer responded to which campaigns by putting an order. We joined this information together in a data frame `dat_num`. The first column holds the name of the customer and all other columns are titled by id of the campaigns along with the corresponding varietal and hold the number of orders the customer put in these campaigns. All three data frames are stored `WineKMC.Rdata`.

a) Please load `WineKMC.Rdata` and familiarize yourself with the data. Then browse through a customer segmentation analysis of this data which was done using *python* and is described in the blog post (http://blog.yhat.com/posts/customer-segmentation-using-python.html). What do you think, which distance has been used in this analysis?

b) Visualize the data in a 2D MDS or score plot. How many clusters do you see in the plot? Perform a k-means clustering like described in the blog (with k=5 and k=3) and color/label the points in the 2D plot to indicate the cluster association. To get a feeling for the cluster quality also produce a silhouette plots. Comment on your results.

c) Determine Euclidean distances and do a hierarchical cluster analysis with the *ward* method. Looking at the cluster dendrogram, find a good count for the number of clusters to cut to tree. Then cut the tree and plot the silhouette distance. Comment on the results

d) Up to now, we used Euclidean distances. Now, we want to use a more appropriate distance to quantify dissimilarities between customers according to their wine ordering behavior. Which distance would you use (look at the number of 0's) in the data set.

Hint: daisy cannot work with this data_frame, first use `as.matrix()` of the appropriate columns to transfer it into a matrix. Also use the argument `type` in `daisy` to specify the appropriate distance.

e) Perform again a hierarchical clustering using the appropriate distance matrix using the ward method. Looking at the cluster tree, find a good count for the number of clusters to cut to tree. Then cut the tree and plot the silhouette distance. Do you get a better clustering with the new distance measure compared to the Euclidean distance?

f) Let's have a look into the cluster which is most homogeneous (look at the silhouette) plot. Find out in which campaigns how many orders have been put by this customer segment. What do this customers all have in common? You might want to use the `colSums` function. Compare your insights with the insights gained in the blog post.