Multivariate Statistics: Exercise 2

October 17, 2018

Cluster analysis:

Load the data `olives` from the package `classifly`. The data contain measurements on fatty acids in Italian olive oils. The oils originate from tree different regions in Italy (`$Region`) which are again split into subregions (`$Area`). The remaining variables represent the fatty acids.

The data should be clustered, where the resulting clusters should ideally represent the regions or even the subregions. Use for the further analyses only the first seven fatty acids (why not *eicosenoic*?). Why do you first need to scale the data (`scale()`)?

- Apply k-means clustering (`kmeans()`) with $k = 3$. How many objects are misclassified (i.e. not correctly assigned to the 3 regions)? How sensitive is the result with respect to the random initialization within the algorithm? How could you get rid of this sensitivity?

- How could you find the "optimal" value of k? Compute for this purpose some prominent validity measures for a range of values of k:

  - *Calinski-Harabasz index*

    $$\mathrm{CH}_k = \frac{B_k/(k-1)}{W_k/(n-k)}$$

    and the *Hartigan index*

    $$\mathrm{H}_k = \ln \frac{B_k}{W_k},$$

    where n is the number of observations, k is the number of clusters, B_k is the between-cluster sum-of-squares with k clusters, and W_k is the within-cluster sum-of-squares with k clusters.

  - *Silhouette plot:* implemented in the package `cluster` as function `silhouette()`. Look at the help pages and figure out the idea behind this validity measure. The plot is done by: `plot(silhouette(clustervector,distances))`, where `clustervector` is a vector with the assignments of the observations to the clusters, and `distances` is the distance matrix of the data, obtained by the function `dist()`.

  - *Gap statistic:* implemented in the package `cluster` as function `clusGap()`. Look at the help pages and figure out the idea behind this method. Details are at `https://statweb.stanford.edu/~gwalther/gap`. Results are computed e.g. by `clusGap(x, FUN = kmeans, K.max=10)`. The results can be shown graphically using `plot()`, and the function `maxSE()` returns the optimal k.

  Compare the different approaches in terms of the resulting optimal values of k.

- Apply hierarchical cluster analysis (`hclust()`), using the methods *complete linkage*, *single linkage*, and *average linkage*. Here you cannot directly use the data matrix as an input, but you need to provide a distance matrix (`dist()`). Visualize the results with a dendrogram (`plot()` the resulting object). Select the 3-cluster solution using `cutree()` with option `k=3`. How many observations are misclassified? How sensitive is the result with respect to the distance measure used?

- In the package *mclust* you can find procedures for model-based clustering. Use the function `Mclust()`, provide the data matrix, and possibly a vector with the desired numbers of clusters, see helpfile. Compare the results to the methods from above.

Save your (successful) R code together with short documentations and interpretations of results in a text file (= R script file), named as *Matrikelnummer_2.R* (no word document, no plots). Submit this file to Exercise 2 of our tuwel course (deadline October 16).