Multivariate Statistics: Exercise 12

January 9, 2019

Discriminant analysis:

We consider the data set *winwequality-red.csv*, which is available from the TUWEL course of our exercises – see Exercise 7. Convert the variable *quality* to a factor variable with levels "Low" (quality 3, 4, 5) and "High" (quality 6, 7, 8), and delete the variable *quality* from the data frame for the following tasks. The new factor variable is the class variable for discriminant analysis. Split the data set randomly into about 2/3 of training data and 1/3 of test data. In all following tasks, the discriminant function should be estimated only based on the training data, and the evaluation should be done for the test data.

1. Use in addition to the class variable only the variables *alcohol* and *fixed.acidity.* Plot the training data and visualize the class by color.

   (a) Use `lda()` from the package `MASS` for linear discriminant analysis (LDA) (only for training data). Predict the class membership (`predict.lda()`) for the test data and compute the misclassification rate (based on `table(truth, prediction)`).

   (b) Visualize the classification border in the plot.
   Hint: This is a bit tricky. One can generate new "test" data, which are defined on a grid of all x/y values in the domain of the plot, and predict the class. The classification boundary should then be linear. Do an internet search, and you will soon find elegant solutions . . .

   (c) Perform 1.(a) and 1.(b) for quadratic discriminant analysis (QDA) (`qda()`).

   (d) The package `rrcov` also contains an implementation of LDA (`LdaClassic()`) and QDA (`QdaClassic()`), but also of robust LDA (`Linda()`) and robust QDA (`QdaCov()`). Use all these methods and perform tasks 1.(a) and 1.(b). Be careful, you get S4 objects, and e.g. the predictions are obtained by `predict(object,newdata)@classification`.

   (e) Compare all the separation lines in the plot and the resulting misclassification rates. What do you conclude?

2. Now use all variables in the data frame.

   (a) Use `lda()` and `LdaClassic()` for the training data, and compute the misclassification rate for the test data.

   (b) It seems that `LdaClassic()` gives very poor results. Run the procedure again. Why do the results change (but are still bad)? What could be the problem for this instability? How could you solve the problem?
   Hint: dimension reduction

Save your (successful) R code together with short documentations and interpretations of results in a text file (= R script file), named as *Matrikelnummer_12.R* (no word document, no plots). Submit this file to Exercise 12 of our tuwel course (deadline January 8).