Series 11

1. In this series we are going to explore the dataset `vehicle.dat` which can be found at `"http://stat.ethz.ch/Teaching/Datasets/NDK/vehicle.dat"`. The dataset contains 846 observations of 19 variables. The aim is to classify the response (which is named `Class`) into four different car types (`bus,van,saab,opel`) by means of 18 predictors such as compactness, some information about the car axes and certain length ratios of the cars' silhouettes. For this, we are going to use CART's with cost-complexity-optimized size. The optimal tree size can be found automatically using the methods from the package `rpart`.

a) First of all, generate a classification tree using the methods from `rpart`. Set the options `cp = 0` and `minsplit = 30` such that the resulting tree becomes too large and overfits the data. Comment on the tree.
R-Hints:
```
library(rpart)
tree <- rpart(Class ~ ., data = ?,
              control = rpart.control(cp = 0.0, minsplit = 30))
```
To visualize the tree use:
```
plot(tree, uniform = TRUE)
text(tree, use.n=TRUE, all=TRUE, cex=0.8, fancy=FALSE, pretty=3)
```

b) Now it comes to pruning the tree from part **a)**. We let `rpart` perform a cost-complexity-analysis to find an optimal `cp`-value by cross-validating a sequence of subtrees of the tree in **a)**. Generate a cost-complexity table and explain it. Determine the optimal `cp` according to the *one standard-error rule*. Is this the same model as the one with the minimal cross-validation error? Visualize the pruned tree with the optimal `cp`, compare it to the full tree, and calculate its misclassification rate.
R-Hints:
- To access the cost-complexity table use `printcp(tree)`, to plot classification error (relative to root tree) vs. the subtree size (dotted line represents one standard error limit) use `plotcp(tree)`.
- To prune the tree use `tree.pruned <- prune.rpart(tree, cp = ?)`.
- For the misclassification rate look at `?residuals.rpart`.

c) To investigate the predictive power, compute the bootstrap generalization error and the leave-one-out cross-validated performance (based on 0-1 loss) for the `cp`-optimal tree from above. Use $B = 1000$ bootstrap-samples, and `set.seed(100)` for reproducibility. Comment on the different values you get.
R-Hint: to predict classes from an rpart object `tree` use
`predict(tree, newdata = ?, type = "class")`

d) **(optional)** Finally, calculate the out-of-bootstrap sample generalization error (cf. Chapter 5.2.5 of the lecture notes). Compare the value you get to the (standard) bootstrap generalization error and the cross-validation error from **c)**.

2. **a)** Let's consider the general linear regression model:

$$y_i = \beta_0 + \sum_{j=1}^{p} \beta_j \cdot x_{ij}.$$

Show that this model is equivalent to the following one:

$$y_i - \bar{y} = \sum_{j=1}^{p} \beta_j \cdot (x_{ij} - \bar{x}_{\cdot j}).$$

Therefore by centering the variables it is always possible to get rid of the intercept β_0.

b) Show that the ridge-regression solution defined as

$$\tilde{\beta}^*(s) = \underset{\|\beta\|^2 \leq s}{\arg\min} \|\mathbf{Y} - X\beta\|^{\mathbf{2}}$$

is given by

$$\hat{\beta}^*(\lambda) = (X^\intercal X + \lambda I)^{-1} X^\intercal \mathbf{Y}.$$

where λ is a suitably chosen Lagrange-multiplicator. Therefore the ridge estimator is still linearly depending on the response $\mathbf{Y}$. Note that (at least) for large λ the ridge solution exists even if $X^\intercal X$ has not full rank or if it is computationally close to singular. Therefore ridge regression is practicable also if $n \ll p$.

c) The *ridge traces* $\hat{\beta}^*(\lambda)$ can computationally easily be determined by using a *singular value decomposition* of the data matrix $X = UDV^\intercal$ where $U(n \times p)$ and $V(p \times p)$ are orthogonal and D is diagonal. Show that:

$$\hat{\beta}^*(\lambda) = V(D^2 + \lambda I)^{-1} DU^\intercal Y.$$

d) Show that the ridge regression fit is just a linear combination of shrinked response-components y_i with respect to the orthogonal basis defined by U. More explicitly show that:

$$\hat{y}_{ridge}(\lambda) = \sum_{j=1}^{p} \mathbf{u_j} \frac{d_j^2}{d_j^2 + \lambda} \mathbf{u_j}^\intercal \mathbf{y},$$

where d_j are the diagonal elements of D. In fact one can show that the directions defined by $\mathbf{u_j}$ are the so called *principal components* of the dataset X. The smaller the corresponding d_j-value, the smaller the data variance in direction u_j. For directions with small data variance, the gradient estimation for the minimization problem is difficult, therefore ridge regression shrinks the corresponding coefficients the most.

e) Ridge regression can also be motivated by Bayesian theory. We assume that

$$\mathbf{Y}|\beta \sim \mathcal{N}(X\beta, \sigma^{\mathbf{2}} I) \text{ and } \beta \sim \mathcal{N}(\mathbf{0}, \tau I).$$

Show that the ridge estimator $\hat{\beta}^*(\lambda)$ is the mean of the posterior distribution. What is the relationship between λ, τ and σ^2?

Preliminary discussion: Friday, May 27.

Deadline: Friday, June 03.