Exercise 11

1. **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:

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

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

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

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 practiable 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:

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

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?

2. Once again we look at the dataset `vehicle.dat` which still can be found at:

`"http://www.ethz.ch/Teaching/Datasets/NDK/vehicle.dat"`

This time we apply two shrinking regression methods as our classifiers, namely *ridge- regression* and *lasso*. The aims of this exercise are to find the optimal degree of shrinking by cross-validation, to evaluate the methods' classfication accuracy and predictive power and finally to compare the results to the `rpart`- and nnet-results from exercise 10.

Ridge-regression is performed by the function `lm.ridge` which can be found in the `MASS`-package, whereas lasso is provided as function `lars` in the homonymous package `lars`.

a) Because we use plain non-generalized regression methods as classifiers in a mutlticlass-classification problem (remember that the `Class`-variable consists of *four* factors `bus`,`van`, `saab`,`opel`) we can choose a *one against the rest*- approach (as described in the manuscript on p.56). Write functions `cl.lasso` and `cl.ridge` which calculate the misclassification rate. Write the functions in such a way that they can as well be used later in your CV-code to determine optimal tuning parameters for the shrinkage-process.

R-Hints: From the help-file `?lm.ridge` we learn that the parameter `lambda` which determines the degree of penalization on the regression coefficient vector's L_2-norm can be given as a whole vector. A good choice could be:

```
lambda <- c(0,2^c(-10,-5, seq(0,10, length=101)))
m.ridge <- lm.ridge(formula = ???,data=???,lambda=lambda,...)
```

There is *no* predefined `predict`-method for `ridgelm`- objects. You have to calculate the prediction - probabilities for each factor on your own. Because `lm.ridge` does centering and scaling of the input data, you need to backtransform by something like (help file for explanation!):

```
prob[,i,]  <-  with(m.ridge, ym * Inter + ((x.new-rep(xm,each = l.new))*
                         Inter/rep(scales,each = l.new)) %*% coef)
```

where `x.new` is the matrix of predictors. Because `m.ridge$coef` gives the regression coefficients for *all* values of the tuning parameter vector `lambda` at once, it is convenient to store the probabilites in a 3-dimensional array, where the first index is over the datapoints, the second over the factors and the third over the components of `lambda`.
For `lars`-objects there are methods `predict()` and `coef()` which can be used for prediction. See `predict.lars` for details. Use the option `mode = "fraction"` for `predict`. Then the tuning parameter `s` can nicely be interpreted as it corresponds to a regression coefficient whose L_1-norm is $s\%$ of the corresponding least-squares coefficient vector's L_1-norm. Therefore a convenient choice for s is:

```
s <- seq(0,1,length=100)
prob[,i,]  <- predict(m.lasso,newx=x.new,s=s.range,type="fit",
              mode="fraction")$fit
```

Again prediction is made for *all* s-components at once.

b) Write functions `CV.ridge` and `CV.lasso` to determine optimal values for the tuning parameters `lambda` and `s`. Choose misclassification-error as your CV-criterion. Note that there can be several grid points at which the minimum is attained because softmax-classification may stay the same in a small neighbourhood of a given shrinkage parameter because in such a neighbourhood the regression coefficients and thus also the probabilities for the different factors will change only a little. From all CV-optimal models choose the one with the lowest misclassification error on the *whole* `vehicle` data-set.

c) Plot the lasso- and ridge traces, fit the optimal models and compare their perfomance with `rpart` and `nnet` from exercise 10. Because of L_1-penalization many of the fitted lasso-method regression coefficients can become 0. As `rpart` the lasso can thus be used for *variable selection*. Compare the selections of relevant predictors made by the lasso to those made by `rpart`.

R-Hints: for traces-plotting you can use the ordinary `plot`-function for `lars` and `ridgelm`-objects. For `lars` you can also look at `?plot.lars`.

Deadline: Friday, July 1, 2005, at the beginning of the lecture.

Advice: Thursdays from 12.00-13.00, LEO C12.1, Leonhardstr. 27. Or contact either Maik Berchtold, `berchtld@stat.math.ethz.ch`, or Lukas Meier, `meier@stat.math.ethz.ch`.