TMA4268 V2021 Exam

TMA4268 Statistical Learning V2021

Stefanie Muff, Department of Mathematical Sciences, NTNU

May 26, 2021

Warming up

Problem 1 (Fill-in-the-blank text, 7P)

Read the whole text and fill in the blanks such that the whole text makes sense (you might only understand which answer is correct after you continued reading):

We have discussed a lot of methods and models, and in *supervised* (*unsupervised*, *parametric*, *non-parametric*) methods there were two main purposes: *prediction* (*inference*, *bias reduction*, *variance reduction*, *supervised learning*, *supervised learning*) and *inference* (*prediction*, *bias reduction*, *variance reduction*, *unsupervised learning*, *supervised learning*). In both cases we want to learn from data and build a model that relates a set of variables to an outcome, but in the first case we do not care about the actual model parameters, because we do not want to interpret them. Some of the methods we learned about were *parametric* (*non-parameteric*, *supervised*, *unsupervised*) and others were *non-parametric* (*parameteric*, *supervised*, *unsupervised*), whereas the former tend to be more more rigid and thus less flexible – and thus possibly more biased – than the latter ones.

In any case, the aim of a supervised model fitting procedure in statistical learning is to minimize the *reducible error* (alternatives: *irreducible error*, *bias*, *variance*, *overfitting*, *underfitting*) of the estimated relation between some predictor variables and a response. The *irreducible error* (alternatives: *reducible error*, *bias*, *variance*, *overfitting*, *underfitting*), on the other hand, corresponds to the expected test error when we find the best possible function that relates the predictors to the response.

Conceptual / theoretical questions

Problem 2 (Multiple choice, 6P)

a) (2P)

Which of the following methods is/are suitable to do model selection / variable selection?

- Support vector classifiers
- [correct] Lasso
- Ridge regression
- Partial least squares regression

b) (2P)

Which of the following are assumptions that are made when fitting a normal multiple linear regression model $y_i = \beta_0 + \beta_x x_i + \beta_z z_i + \epsilon_i$ with $1 \leq i \leq n$ and n number of data points?

- The response variables y_i are independent and identically distributed.
- The covariates x and z are not collinear.
- [correct] The error terms are normally distributed.
- [correct] The error terms have a constant variance and are independent of each other.

c) (2P)

Which of the following are tuning parameters in at least one of the methods we discussed in the course?

- [correct] The number of trees in boosted regression trees.
- The width of the margin M in support vector classifiers.
- [correct] The number of principal component included in principal component regression.
- [correct] K in k-nearest-neighbor (KNN) classification.

Problem 3 (Free text questions 7P)

a) (3P)

One of the central topics in the course was the bias-variance trade-off.

(i) (1P) Say in 2-3 sentences what the bias variance trade-off means.
(ii) (2P) Give two examples of methods in the course where there were (tuning) parameters that could be tweaked such that there was a bias variance trade-off.

Solution: (i) In brief, it means that for more flexible models, the variance will increase and the bias will decrease and vice versa (this answer is enough to get the point). While the training error decreases for more flexible models, the test error has an optimum at some (usually) intermediate point. It is also a trade-off between under- and overfitting. (ii) Many examples, but some are K in KNN or KNN regression; the degree of a polynomial or the number of covariates in a regression model; the depth of a regression/classification tree; etc.

b) (2P)

Assume you are a statistician that is involved in an epidemiological study. The researchers are interested in finding variables that are associated with obesity, such as physiological measures, personal behaviour, food habits or genetic components. The aim of the study is to help obese people lose weight.

(i) (1P) What is the purpose of finding a good model for this question: prediction or inference? Please justify your anwer.
(ii) (1P) When would analysing the same dataset serve the opposite purpose (i.e., if you chose prediction in (i), which question would make it an inference type of problem, and vice versa)?

Solution: (i) The aim is inference, because we want to understand how variables affect the outcome. (ii) The opposite purpose is prediction, which is for example the case when we only want to give predictions for obesity to people early on, in the sense of "how likely is it that you get obese, given your behavior and physiology?".

c) (2P)

We have learned about K-means clustering and hierarchical clustering. Give one advantage and one disadvantage of K-means clustering compared to hierarchical clustering.

Soultion: Disadvantages: - K-means results obtained will depend on the initial (random) cluster assignment. - Number of clusters have to be fixed in the beginning

Advantages: - Simple and intuitive - Works well on large data sets

Problem 4 (5P)

In the module about support vector machines we have mentioned the *hinge loss* $L(\boldsymbol{x}_i, y_i, \boldsymbol{\beta}) = \max(0, 1 - y_i f(\boldsymbol{x}_i))$ where in the simplest case, $f(\boldsymbol{x}_i)$ is a linear function of the covariates and some parameter vector $\boldsymbol{\beta}$. In the case where $f(\boldsymbol{x}_i) = \beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip}$ we saw that the hinge loss and the logistic regression loss were very similar, see Figure 9.12 of the course book:

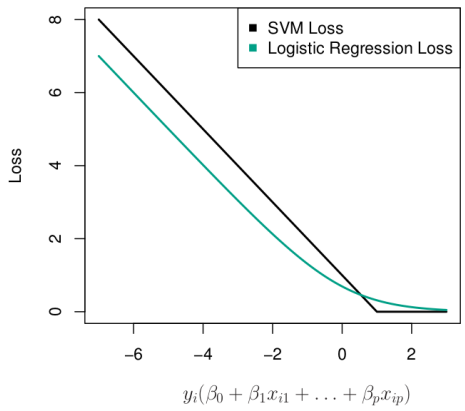


Figure 1: ISLR Figure 9.12: hinge loss - loss 0 for observations on the correct side of the margin

Show that the loss function

$$\log(1 + \exp(-y_i f(\boldsymbol{x}_i)))$$

is the negative log-likelihood for the $y_i = -1, 1$ encoding in a logistic regression model.

Solution: Using that $\mathrm{P}(Y = 1) = \frac{\exp(f)}{1+\exp(f)}$ and $\mathrm{P}(Y = -1) = \frac{1}{1+\exp(f)}$, the contribution of an observation i, $(\boldsymbol{x}_i, y_i)$, to the likelihood is

$$\mathrm{L}(f; (\boldsymbol{x}_i, y_i)) = \left(\frac{\exp(f(\boldsymbol{x}_i))}{1 + \exp(f(\boldsymbol{x}_i))}\right)^{\mathrm{I}(y_i=1)} \cdot \left(\frac{1}{1 + \exp(f(\boldsymbol{x}_i))}\right)^{\mathrm{I}(y_i=-1)} .$$

Therefore, the negative log-likelihood (which is proportional to the deviance) is given as

$$\begin{aligned}
-\log \mathrm{L}(f; (\boldsymbol{x}_i, y_i)) =& -\mathrm{I}(y_i = 1) \log(\frac{\exp(f(\boldsymbol{x}_i))}{1 + \exp(f(\boldsymbol{x}_i))}) \\
& -\mathrm{I}(y_i = -1) \cdot \log(\frac{1}{1 + \exp(f(\boldsymbol{x}_i))}) \\
=& \mathrm{I}(y_i = 1) \log\left(\frac{1 + \exp(f(\boldsymbol{x}_i))}{\exp(f(\boldsymbol{x}_i))}\right) \\
& + \mathrm{I}(y_i = -1) \cdot \log\left(1 + \exp(f(\boldsymbol{x}_i))\right) \\
=& \mathrm{I}(y_i = 1) \log\left(\exp(-f(\boldsymbol{x}_i)) + 1\right)) \\
& + \mathrm{I}(y_i = -1) \cdot \log\left(1 + \exp(f(\boldsymbol{x}_i))\right) \ .
\end{aligned}$$

Therefore

$$\begin{aligned}
-\log \mathrm{L}(f; (\boldsymbol{x}_i, y_i)) =& \mathrm{I}(y_i = 1) \log\left(1 + \exp(-y_i f(\boldsymbol{x}_i))\right) + \\
& \mathrm{I}(y_i = -1) \log\left(1 + \exp(-y_i f(\boldsymbol{x}_i))\right) \ .
\end{aligned}$$

This expression can now be summarized to the result

$$\log\left(1 + \exp(-y_i f(x_i))\right) \ .$$

Problem 5 – Data analysis 1 (20P)

In this task we are again using the bodyfat example from the course, but this time we use a version with more covariates. The data set can be loaded and split into a training and test set as described below. Please look at the data set yourself, for example by using the `pairs()` and `str()` functions, before you start working on the analysis.

```
id <- "1dNLfx9Dbs2gYIooUxA6HMxK_MPFwE3Hn"  # google file ID
d.bodyfat <- read.csv(sprintf("https://docs.google.com/uc?id=%s&export=download",
    id), header = T)
```

The variable `bodyfat` is the reponse variable, all other variables are covariates.

We are splitting into a training set (180 observations) and a test set (63 observations):

```
set.seed(1234)
samples <- sample(1:243, 180, replace = F)
d.body.train <- d.bodyfat[samples, ]
d.body.test <- d.bodyfat[-samples, ]
```

a) (4P)

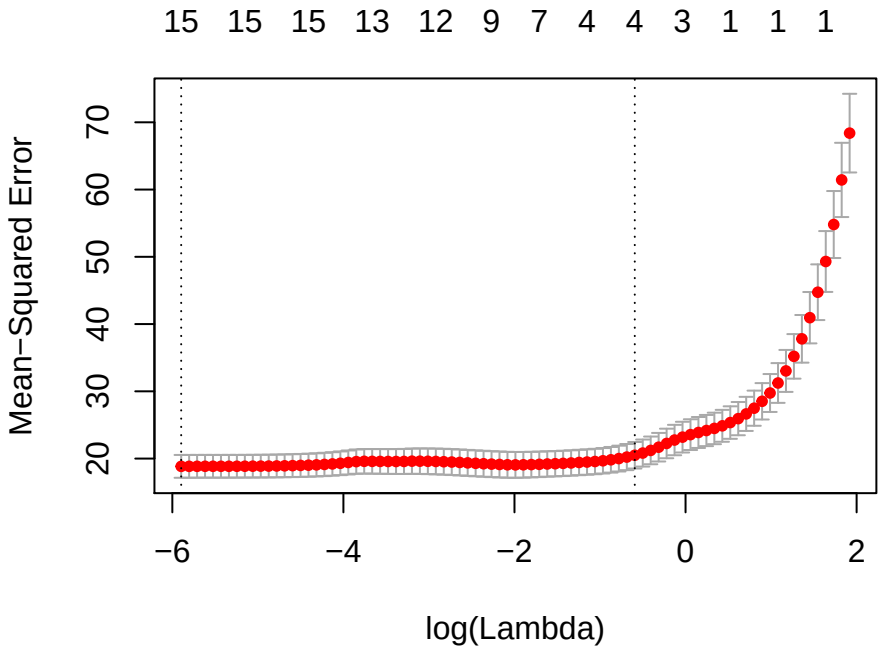
Carry out Lasso regression on the training set, and say how you choose λ. Report the MSE on the test data.

Solution

The students should choose lambda with cross-validation and then use the minimal value, or the smallest value within 1 SE for the prediction model.

```
x.train <- model.matrix(bodyfat ~ ., data = d.body.train)[, -1]
y.train <- d.body.train$bodyfat
x.test = model.matrix(bodyfat ~ ., data = d.body.test)[, -1]
y.test = d.body.test$bodyfat
```

```
library(glmnet)
cv.lasso <- cv.glmnet(x.train, y.train, alpha = 1)
plot(cv.lasso)
```




```
cv.lasso$lambda.1se
```

```
## [1] 0.5523759
```

```
bodyfat.lasso <- glmnet(x.train, y.train, alpha = 1, lambda = cv.lasso$lambda.1se)

coef(bodyfat.lasso)
```

```
## 16 x 1 sparse Matrix of class "dgCMatrix"
##                        s0
## (Intercept) -19.79483282
## age           0.03073728
## weight        .
## height       -0.05366148
## bmi           .
## neck          .
## abdomen       0.67712360
## hip           .
## chest         .
## thigh         .
## knee          .
## ankle         .
## biceps        .
## forearm       .
## wrist        -0.85988914
## head          .
```

```
mse.lasso = mean((y.test - predict(bodyfat.lasso, newx = x.test))^2)
mse.lasso
```

```
## [1] 18.88053
```

Solution with minimal lambda (only one coef is =0 in that case!):

```
cv.lasso$lambda.min
```

```
## [1] 0.002749188
```

```r
bodyfat.lasso.min <- glmnet(x.train, y.train, alpha = 1, lambda = cv.lasso$lambda.min)

coef(bodyfat.lasso.min)
```

```
## 16 x 1 sparse Matrix of class "dgCMatrix"
##                         s0
## (Intercept) -167.03250191
## age            0.07876188
## weight        -1.06535033
## height         0.88721158
## bmi            3.33811274
## neck          -0.42853459
## abdomen        0.87148140
## hip           -0.27689454
## chest         -0.12067733
## thigh          0.35260128
## knee           0.21913683
## ankle         -0.36212334
## biceps         0.34855918
## forearm       -0.08501644
## wrist         -1.57358210
## head           .
```

```r
mse.lasso.min = mean((y.test - predict(bodyfat.lasso.min, newx = x.test))^2)
mse.lasso.min
```

```
## [1] 21.58997
```

b) (4P)

(i) (2P) Fit a linear regression model with all covariates to the training data and calculate the MSE on the test data.
(ii) (1P) Compare the test MSE to the one you obtain from a) and interpret the difference.
(iii) (1P) Systematically compare the regression coefficients between linear regression and Lasso, and give a theoretical explanation for the pattern you see.

Solution:

```r
r.lm.bodyfat <- lm(bodyfat ~ ., d.body.train)

mse.lm = mean((y.test - predict(r.lm.bodyfat, newdata = d.body.test))^2)
mse.lm
```

```
## [1] 22.10107
```

```r
cbind(r.lm.bodyfat$coef[-1], coef(bodyfat.lasso)[-1])
```

```
##                 [,1]        [,2]
## age      0.078823204  0.03073728
## weight  -1.237664889  0.00000000
## height   1.045418867 -0.05366148
## bmi      3.886616206  0.00000000
## neck    -0.427587053  0.00000000
## abdomen  0.874043244  0.67712360
## hip     -0.298990981  0.00000000
## chest   -0.126480827  0.00000000
```

```
## thigh    0.363928629  0.00000000
## knee     0.247814245  0.00000000
## ankle   -0.374916332  0.00000000
## biceps   0.374657808  0.00000000
## forearm -0.108045422  0.00000000
## wrist   -1.561702755 -0.85988914
## head     0.004003341  0.00000000
```

(ii) The test MSE is smaller for Lasso, which is expected since Lasso is reducing variance at the cost of bias, and λ was selected such that the bias-variance trade-off was optimal in some sense.

(iii) The Lasso coefficients tend to be smaller than for linear regression due to shrinkage. (Note that here only one of them is zero when lambda.min is used – indicating that most variables are having some predictive ability, but this is not part of the answer I expect).

c) (4P)

Fit a GAM on the training data, including

- a polynomial of degree 2 for age,
- a natural cubic spline with 3 degrees of freedom for height,
- a natural cubic spline for abdomen with three knots at the 25%, 50% and 75% quantiles, respectively,
- a smoothing spline for hip,
- a linear term for weight and bmi.

Calculate the MSE for the test set.

Solution:

```
# library(gam) library(mgcv)
quantile(d.bodyfat$abdomen)[2:4]
```

```
##   25%   50%   75%
## 84.90 91.00 99.15
```

```
r.gam <- gam(bodyfat ~ poly(age, degree = 2) + ns(height, df = 3) + weight +
    bmi + ns(abdomen, knots = c(84.9, 91, 99.15)) + s(hip), data = d.body.train)

summary(r.gam)
```

```
##
## Call: gam(formula = bodyfat ~ poly(age, degree = 2) + ns(height, df = 3) +
##     weight + bmi + ns(abdomen, knots = c(84.9, 91, 99.15)) +
##     s(hip), data = d.body.train)
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -12.7750  -2.6791   0.1481   2.4430   8.7911
##
## (Dispersion Parameter for gaussian family taken to be 18.4341)
##
##     Null Deviance: 12288.44 on 179 degrees of freedom
## Residual Deviance: 3023.193 on 164.0001 degrees of freedom
## AIC: 1052.618
##
## Number of Local Scoring Iterations: 2
##
## Anova for Parametric Effects
##                                        Df Sum Sq Mean Sq  F value    Pr(>F)
```

```
## poly(age, degree = 2)                     2 1199.3   599.6  32.5291 1.264e-12
## ns(height, df = 3)                        3  110.2    36.7   1.9924    0.1172
## weight                                    1 5922.0  5922.0 321.2518 < 2.2e-16
## bmi                                       1  294.8   294.8  15.9919 9.602e-05
## ns(abdomen, knots = c(84.9, 91, 99.15))   4 1693.1   423.3  22.9621 4.426e-15
## s(hip)                                    1    2.5     2.5   0.1350    0.7138
## Residuals                               164 3023.2    18.4
##
## poly(age, degree = 2)                   ***
## ns(height, df = 3)
## weight                                  ***
## bmi                                     ***
## ns(abdomen, knots = c(84.9, 91, 99.15)) ***
## s(hip)
## Residuals
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Anova for Nonparametric Effects
##                                         Npar Df Npar F   Pr(F)
## (Intercept)
## poly(age, degree = 2)
## ns(height, df = 3)
## weight
## bmi
## ns(abdomen, knots = c(84.9, 91, 99.15))
## s(hip)                                        3 2.2309 0.08657 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
mse.gam = mean((y.test - predict(r.gam, newdata = d.body.test))^2)
c(mse.gam, mse.lm, mse.lasso)
```

```
## [1] 20.39062 22.10107 18.88053
```

d) (3P)

In the course you also heard about *partial least squares (PLS)* regression, which is a smart approach that uses the principal component regression idea, but finds the components that are most correlated with the response. For the bodyfat example do the following:

(i) (1P) Run a PLS regression on the training data (don't forget to scale the variables, `scale=TRUE`).
(ii) (1P) Choose the smallest number of components such that at least 95% of the covariate variance in the training data is explained.
(iii) (1P) Report the MSE of the test data when using the respective number of components.

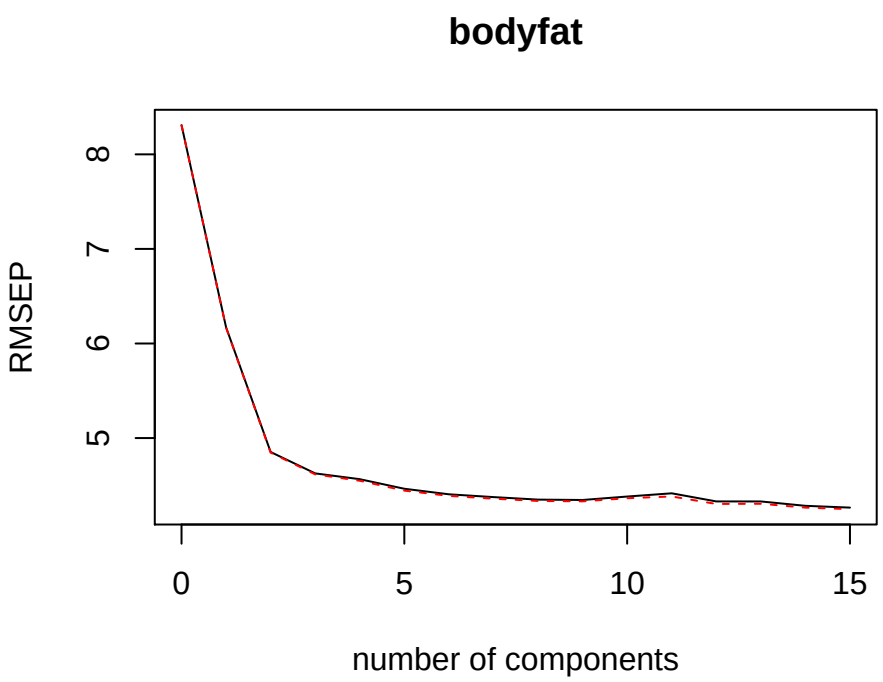
Solution:

```
library(pls)
set.seed(4268)
r.pls <- plsr(bodyfat ~ ., data = d.body.train, scale = T, validation = "CV")
summary(r.pls)
```

```
## Data:    X dimension: 180 15
##  Y dimension: 180 1
## Fit method: kernelpls
```

```
## Number of components considered: 15
##
## VALIDATION: RMSEP
## Cross-validated using 10 random segments.
##        (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
## CV           8.309    6.171    4.853    4.626    4.565    4.464    4.406
## adjCV        8.309    6.167    4.846    4.616    4.549    4.444    4.389
##        7 comps  8 comps  9 comps  10 comps  11 comps  12 comps  13 comps
## CV       4.376    4.349    4.345     4.382     4.416     4.330     4.330
## adjCV    4.359    4.335    4.331     4.364     4.383     4.303     4.305
##        14 comps  15 comps
## CV        4.284     4.265
## adjCV     4.265     4.248
##
## TRAINING: % variance explained
##          1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps  8 comps
## X          58.36    70.33    75.49    79.44    82.86    86.58    90.88    93.18
## bodyfat    45.86    67.46    71.35    73.09    74.73    75.46    75.71    75.81
##          9 comps  10 comps  11 comps  12 comps  13 comps  14 comps  15 comps
## X          94.95     96.34     96.83     97.73     98.53     98.84    100.00
## bodyfat    75.87     76.01     76.62     76.88     77.01     77.04     77.04
```

```
# The following plot is not expected, but can be informative:
validationplot(r.pls)
```



10 components is the first number that crosses the 95% limit

```
mse.pls = mean((d.body.test$bodyfat - predict(r.pls, x.test, ncomp = 10))^2)
mse.pls
```

```
## [1] 19.63273
```

e) (3P)

Finally, fit either a random forest or a boosted regression tree to the training data, and calculate the MSE on the test data. Explain your choic(es) that you make for the tuning parameter(s) in your selected method.

Solution

Solutions are expected to differ between students. For each mistake we deduct -1P.

Students that choose boosted regression trees should show that they understand that the number of trees is a tuning parameter, so they should mention this. Also interaction depth and the shrinkage factor are tuning parameters

If the choice is a random forest, the number of trees is no tuning parameter (but no point deducted, because the point is apparently debated in the literature). In the RF approach the students should explain how they use mtry (it should be around $p/3$, thus $15/3 = 5$, but they might choose another, similar value and justify why).

```
library(gbm)

set.seed(4268)

ntrees <- 5000
boost = gbm(bodyfat ~ ., d.body.train, distribution = "gaussian", n.trees = ntrees,
    interaction.depth = 2, shrinkage = 0.001)
yhat.boost = predict(boost, newdata = d.body.test, n.trees = ntrees)
mse.boost = mean((d.body.test$bodyfat - yhat.boost)^2)
mse.boost
```

```
## [1] 20.55672
```

Random forest example:

```
library(randomForest)
r.rf <- randomForest(bodyfat ~ ., data = d.body.train, mtry = 5, ntree = 2000,
    importance = TRUE)

mse.rf <- mean((d.body.test$bodyfat - predict(r.rf, d.body.test))^2)
```

f) (2P)

Compare all the MSEs you found in a) to e) for the test set. Which of these methods seems to do best and worst for the given test data?

Solution:

```
c(mse.rf, mse.boost, mse.pls, mse.gam, mse.lm, mse.lasso)
```

```
## [1] 20.88733 20.55672 19.63273 20.39062 22.10107 18.88053
```

Problem 6 – Data analysis 2 (15P)

In this data analysis problem we are using data collected by people at the Centre for Biodiversity Dynamics at NTNU. The main study question was whether inbreeding is influencing the probability that young sparrows survive until the second year (denoted as recruitement). The list of covariates is as follows:

- `sex`: Sex of the animal.

- `lnrhday`: Day in the year when the bird hatched (enumerated from 1 to 365)
- `clsize`: The clutch size of the clutch the bird was born in
- `hyear`: Hatch year
- `f`: Inbreeding coefficient
- `hisl`: Hatch island (this is a categorical variable!)
- `H1`: Proportion of heterozygous loci
- `GTloci`: number of genotyped microsatellite loci
- `Hloci`: number of heterozygous loci
- `geno`: A variable where we do not know what it means
- `recruit`: The binary response variable for survival to the second year

The data can be loaded as follows:

```
id <- "1cSVIJv-OoAwkhUAuun2qQyOfiuZzkmo3"  # google file ID
d.sparrows <- read.csv(sprintf("https://docs.google.com/uc?id=%s&export=download",
    id), header = T)
```

It is advisable to look at the dataset, for example by using `pairs(d.sparrows)` and `str(d.sparrows)` before you do the following analyses:

a) (4P)

(i) (2P) Fit a logistic regression model on the full data set with `recruit` as response variable, using all the covariates plus an interaction term between `sex` and `f`. Remember that hatch island (`hisl`) is a categorical variable.

(ii) (1P) Fit a second model, but this time without hatch island as covariate. Use the anova function to compare to the model in (i).

(iii) (1P) Is there evidence that survival probabilities differed between hatch islands? Explain your response.

R-hints

- To compare two models using an anova table, use `anova(model1,model2, test="Chisq")`.

Solution:

(i) and (ii)

```
d.sparrows$hisl <- as.factor(d.sparrows$hisl)
r.glm <- glm(recruit ~ . + sex:f, d.sparrows, family = "binomial")

summary(r.glm)
```

```
##
## Call:
## glm(formula = recruit ~ . + sex:f, family = "binomial", data = d.sparrows)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.5536  -0.7744  -0.5230   0.7866   2.1427
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) 423.153540 349.742787   1.210   0.2263
## sex          -0.821220   0.447836  -1.834   0.0667 .
## lnrhday       0.020683   0.010802   1.915   0.0555 .
## clsize        0.196824   0.190545   1.033   0.3016
## hyear        -0.222854   0.179995  -1.238   0.2157
```

```
## f            -0.132373   0.087483  -1.513   0.1302
## hisl20       -0.137605   0.788102  -0.175   0.8614
## hisl26       -0.532817   0.567052  -0.940   0.3474
## hisl28       -1.876165   1.150416  -1.631   0.1029
## hisl38        1.237168   0.811106   1.525   0.1272
## H1           16.215035  22.584974   0.718   0.4728
## GTloci        2.204134   2.472942   0.891   0.3728
## Hloci        -1.673924   2.753501  -0.608   0.5432
## geno          0.002901   0.005606   0.518   0.6048
## sex:f         0.048489   0.058147   0.834   0.4043
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 193.82  on 168  degrees of freedom
## Residual deviance: 166.20  on 154  degrees of freedom
## AIC: 196.2
##
## Number of Fisher Scoring iterations: 5
```

```
r.glm2 <- glm(recruit ~ . + sex:f - hisl, d.sparrows, family = "binomial")
anova(r.glm, r.glm2, test = "Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: recruit ~ sex + lnrhday + clsize + hyear + f + hisl + H1 + GTloci +
##     Hloci + geno + sex:f
## Model 2: recruit ~ sex + lnrhday + clsize + hyear + f + hisl + H1 + GTloci +
##     Hloci + geno + sex:f - hisl
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1       154     166.21
## 2       158     176.92 -4  -10.713  0.02999 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

(iii) The p-value for the anova test is 0.030, thus there is evidence that hatch island matters.

b) (3P)

Now split the dataset into a training and a test sample for prediction (assuming our aim is to predict survival). Split the dataset as in the code below, using the same seed. Then

(i) (1P) Fot logistic regression for the training data, but without hatch island and without interaction between f and sex.
(ii) (1P) Use the fitted model to predict survival in the test set using a probability cutoff of $p = 0.5$.
(iii) (1P) Generate the confusion table and calculate sensitivity and specificity for the prediction on the test set.

```
set.seed(123456)
samples <- sample(1:169, 120, replace = F)
d.sparrows.train <- d.sparrows[samples, ]
d.sparrows.test <- d.sparrows[-samples, ]
```

Solution:

```
r.glm.train <- glm(recruit ~ . - hisl, d.sparrows.train, family = "binomial")

predLogistic = predict(r.glm.train, newdata = d.sparrows.test, type = "response")
# The round() function uses the 0.5 cutoff automatically:
logClass = round(predLogistic)
ta = table(true = d.sparrows.test$recruit, predict = logClass)
ta
```

```
##     predict
## true  0  1
##    0 30  4
##    1 14  1
```

```
sensLog = ta[2, 2]/(sum(ta[2, ]))
spesLog = ta[1, 1]/sum(ta[1, ])
c(sensitivity = sensLog, specificity = spesLog)
```

```
## sensitivity specificity
##  0.06666667  0.88235294
```

c) (3P)

Repeat the task from b), but now use a quadratic discriminant analysis (QDA) instead of logistic regression. Calculate sensitivity and specificity for QDA. Calculate sensitivity and specificity for the predictions on the test set.

Solution:

```
library(MASS)
qdaMod = qda(recruit ~ . - hisl, data = d.sparrows.train)
# $class is doing the 0.5 cutoff, while $posterior gives the actual
# probabilities:
postQDA = predict(qdaMod, newdata = d.sparrows.test)$posterior
predQDA = predict(qdaMod, newdata = d.sparrows.test)$class
tQDA = table(true = d.sparrows.test$recruit, predict = predQDA)
tQDA
```

```
##     predict
## true  0  1
##    0 29  5
##    1 10  5
```

```
sensQDA = tQDA[2, 2]/(sum(tQDA[2, ]))
spesQDA = tQDA[1, 1]/(sum(tQDA[1, ]))
c(sensitivity = sensQDA, specificity = spesQDA)
```

```
## sensitivity specificity
##   0.3333333   0.8529412
```

d) (5P)

Finally we are using a neural network approach for our classification task. Prepare the data as indicated in the R-hints

(i) Fit a neural network with two hidden layers to the training set, where

- the first hidden layer has 32 units and the second hidden layer 64.
- the hidden layers have ReLU and the output layer a sigmoid activation function.
- you add 20% dropout in both hidden layers.
- you use RMSprop optimization.
- you use a batch size of 16.
- you use a validation split of 0.5 (**R-hint:** fit(..., validation_split=0.5)).
- you train the model for 25 epochs.
- you use `set.seed(1234)` before you train the network.

(ii) Calculate sensitivity and specificity for the predictions (probability cut-off 0.5) for the test set, and compare to the values you got from logistic regression and QDA above.

R-hints

```
library(keras)
library(caret)
x_train <- d.sparrows.train[, -c(6, 11)]
x_test = d.sparrows.test[, -c(6, 11)]

mean = apply(x_train, 2, mean)
std = apply(x_train, 2, sd)
x_train = scale(x_train, center = mean, scale = std)
x_test = scale(x_test, center = mean, scale = std)

y_train = as.numeric(d.sparrows.train$recruit)
y_test = as.numeric(d.sparrows.test$recruit)
```

To receive predictions from the neural network output, use

```
predictionsNN <- model %>% predict_classes(x_test)
```

Solution:

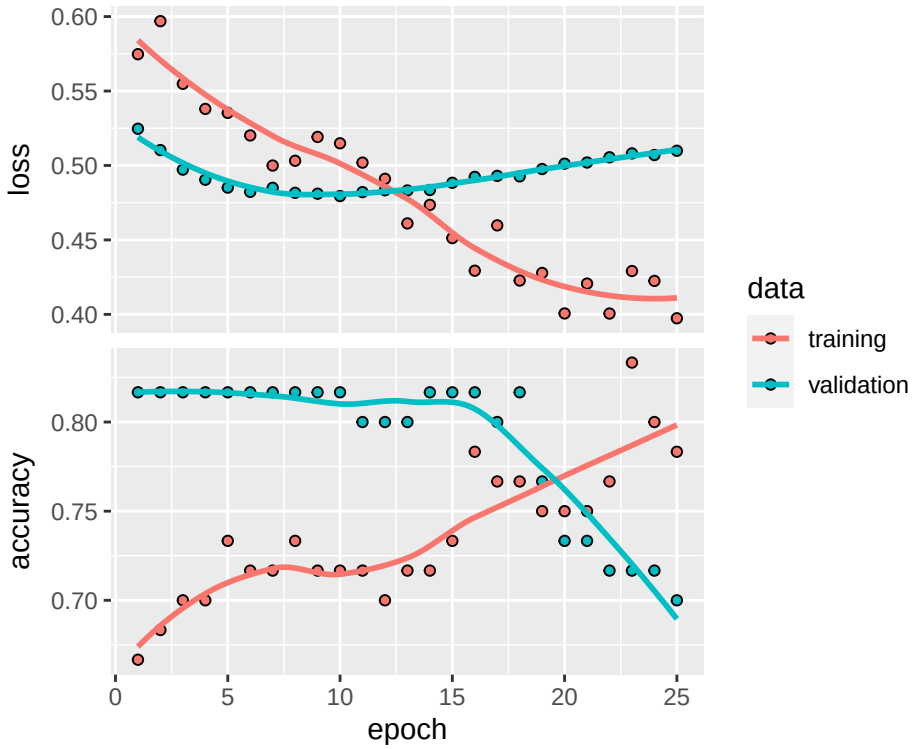
```
set.seed(1234)
model <- keras_model_sequential()

model %>% layer_dense(units = 32, activation = "relu", input_shape = ncol(x_train)) %>%
    layer_dropout(rate = 0.2) %>% layer_dense(units = 64, activation = "relu") %>%
    layer_dropout(rate = 0.2) %>% layer_dense(units = 1, activation = "sigmoid")

model %>% compile(loss = "binary_crossentropy", optimizer = "rmsprop",
    metrics = c("accuracy"))

history <- model %>% fit(x_train, y_train, epochs = 25, batch_size = 16,
    validation_split = 0.5)

plot(history)
```




```
predictionsNN <- model %>% predict_classes(x_test)

tNN <- table(true = d.sparrows.test$recruit, predict = predictionsNN)

sensNN = tNN[2, 2]/(sum(tNN[2, ]))
spesNN = tNN[1, 1]/(sum(tNN[1, ]))
c(sensitivity = sensNN, specificity = spesNN)
```

```
## sensitivity specificity
##   0.2666667   0.9117647
```

Multiple and single choice questions

Problem 7 (4P, single choice, 1P each)

a)

Look at the estimated coefficients from regression model $y_i = \beta_0 + \beta_x x_i + \beta_z z_i + \beta_w w_i + \epsilon_i$ with continuous covariate x and binary covariates z and w.

```
##
## Call:
## lm(formula = y ~ x + z + w)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.6167 -3.6347  0.5217  3.0458 12.0439
##
```

```
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.6311     2.4346   0.670   0.5046
## x             0.9826     0.1091   9.008 3.05e-14 ***
## z            -2.4697     1.0861  -2.274   0.0253 *
## w             2.4402     1.4355   1.700   0.0926 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.222 on 91 degrees of freedom
## Multiple R-squared:  0.491,  Adjusted R-squared:  0.4742
## F-statistic: 29.26 on 3 and 91 DF,  p-value: 2.463e-13
```

What is the predicted outcome ($\hat{y}_i$) for an individual i with $x_i = 20$, $z_i = 0$ and $w_i = 1$?

- 17.2
- 3.4
- 19.6
- 18.8
- [correct] 23.7

b)

Using again the same example as in a). Which of the following statements is true?

- The 95% confidence interval for $\hat{\beta}_x$ ranges from 0.87 to 1.20.
- The 95% confidence interval for $\hat{\beta}_z$ ranges from -3.56 to -1.38.
- The R-squared is the proportion of residual variability on the total variability of the response variable.
- The total number of data points in the above analysis was 94.
- [correct] None of the other statements are correct.

c)

We have 9 covariates, X_1 to X_9 , each of them uniformly distributed in the interval $[0, 1]$. To predict a new test observation $(X_1, \ldots, X_9)$ in a K-nearest neighbor (KNN) approach, we use all observations within 15% of the range closest to each of the covariates (that is, in each dimension). Which proportion of available (training) observations can you expect to use for prediction?

- [correct] $3.8 \cdot 10^{-8}$
- $1.5 \cdot 10^{-9}$
- $0.15 \cdot 10^{-9}$
- 0.15
- 10^{-9}

d)

Let us look at a neural network with one single output node. By which of the following users can this network potentially be used for the specified task?

- By the Norwegian post to read zip codes on letters.
- By a medical institution to discriminate gene expression patterns into three different disease statuses.
- By the criminal police to assign handwriting to different persons.
- [correct] By a hospital to classify X-ray pictures into healthy and unhealthy.
- Non of the other alternatives.

Problem 8 (6P, multiple choice, 2P each)

a)

In a study it was investigated how yearly income (in 1000 Eur) of adults varies with age (years) and education status (low, medium or high education). The result from the analysis is given in the following tables:

```
##                       Estimate Std. Error    t value     Pr(>|t|)
## (Intercept)         33.3820487 3.46209745  9.6421459 2.404054e-20
## age                  1.0217459 0.07639833 13.3739291 2.977750e-35
## educationhigh        7.9713187 4.77037713  1.6710039 9.532491e-02
## educationmedium      3.6151331 4.70229197  0.7688023 4.423610e-01
## age:educationhigh    0.3496907 0.10748204  3.2534806 1.214376e-03
## age:educationmedium  0.1468468 0.10516250  1.3963800 1.631981e-01
```

```
## Analysis of Variance Table
##
## Response: income
##                Df Sum Sq Mean Sq  F value    Pr(>F)
## age             1 108000  108000 712.5665 < 2.2e-16 ***
## education       2  43378   21689 143.0993 < 2.2e-16 ***
## age:education   2   1621     810   5.3474  0.005027 **
## Residuals     518  78510     152
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Please select all statements that are true, according to the summary and anova output:

(i) [correct] People with high education tend to earn more than people with medium or low education.
(ii) Someone with a medium education earns, on average, EUR 102927 at age 50.
(iii) [correct] Income seems to increase with age, and the rate of the increase depends on the education.
(iv) Education does not seem to play a role for people at higher ages.

b)

Which of the following statements are true, which false?

(i) The maximal margin hyperplane approach is equivalent to a linear discriminant analysis when the covariates are normally distributed with identical covariance matrix in the different groups.
(ii) The support vector classifier is equivalent to quadratic discriminant analysis when the covariates are normally distributed with group-specific covariance matrices.
(iii) [correct] Logistic regression, LDA and support vector machines tend to perform similar when decision boundaries are linear, unless classes are linearly separable.
(iv) [correct] An advantage of logistic regression over SVMs is that it is easier to do feature selection and to interpret the results.

Solution FALSE - FALSE - TRUE - TRUE

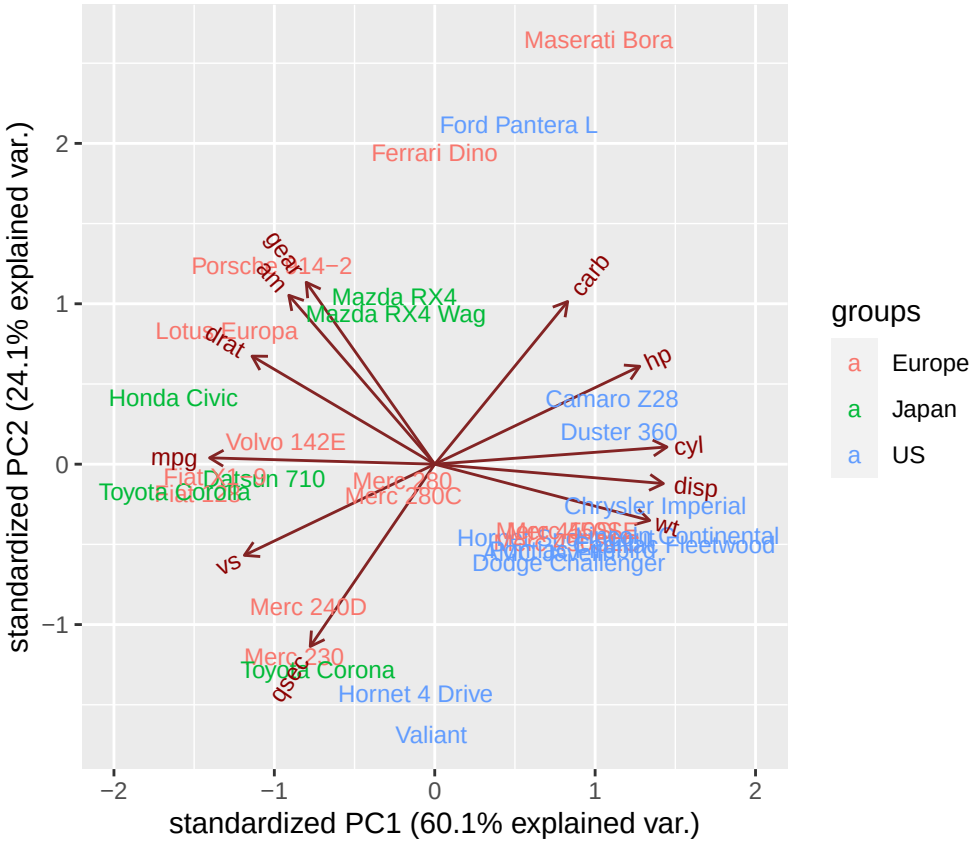
Comment on (i) and (ii): The support vector classifer is not equivalent to a LDA or QDA. One reason to understand why is because only a limited number of data points (the support vectors) influences the estimation of the classifier function of a support vector classifier, while LDA and QDA take all data points into account by estimating the class mean and covariance matrices from the data.

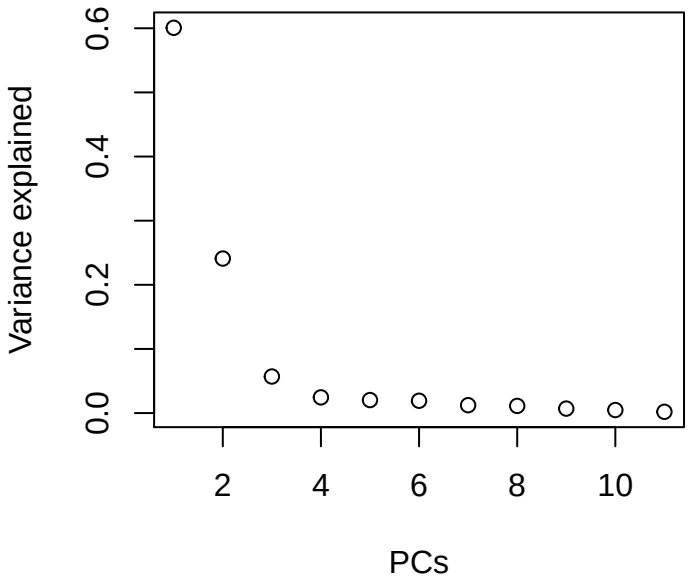
c)

We are looking at the `mtcars` dataset given in R. This dataset consists of data on 32 models of cars, taken from an American motoring magazine (1974 Motor Trend magazine). For each car, you have 11 features, expressed in varying units (US units). You can check in R via ?mtcars to see what the different variables mean.

Here we carried out a principal component analysis and give the biplot and the scree plot below. In the biplot we also color the cars according to their origin. Which of the following statements are correct?

(i) The first two components explain 84.2% of the variability of the response variable.
(ii) Japanese cars tend to have lower miles per gallon (mpg) and more cylinders (cyl variable) than American cars, which indicates that Japanese cars are more environment friendly.
(iii) [correct] mpg has a very small loading for the second principal component.
(iv) [correct] The first three PCs explain most of the variability in the data.




Variance explained
0.0
0.2
0.4
0.6
2
4
6
8
10
PCs