

# Compulsory Exercise 3

## TMA4268 Statistical Learning V2020

Martina Hall, Michail Spitieris, Stefanie Muff, Department of Mathematical Sciences, NTNU

14 mai, 2021

Last changes: 13.04.2020

---

### Problem 1 (9P)

In compulsory exercise 2 we used the `College` data from the `ISLR` library, where we wanted to predict `Outstate`.

```
library(ISLR)
library(keras)
set.seed(1)
College$Private = as.numeric(College$Private)
train.ind = sample(1:nrow(College), 0.5 * nrow(College))
college.train = College[train.ind, ]
college.test = College[-train.ind, ]
str(college.train)

## 'data.frame':   388 obs. of  18 variables:
## $ Private      : num  1 2 1 2 2 1 2 2 2 2 ...
## $ Apps         : num 1401 344 4216 427 2929 ...
## $ Accept       : num 1239 264 2290 385 1834 ...
## $ Enroll       : num  605 97 736 143 622 ...
## $ Top10perc    : num  10 11 20 18 20 10 27 50 62 13 ...
## $ Top25perc    : num  34 42 52 38 56 35 50 77 93 33 ...
## $ F.Undergrad  : num 3716 500 4296 581 2738 ...
## $ P.Undergrad  : num  675 331 1027 533 1662 ...
## $ Outstate     : num 7100 12600 5130 12700 12600 ...
## $ Room.Board   : num 4380 5520 4690 5800 5610 ...
## $ Books        : num  540 630 600 450 450 537 450 525 500 570 ...
## $ Personal     : num 2948 2250 1450 700 3160 ...
## $ PhD          : num  63 77 73 81 90 77 77 76 94 66 ...
## $ Terminal     : num  88 80 75 85 90 84 98 92 96 83 ...
## $ S.F.Ratio    : num 19.4 10.4 17.9 10.3 15.1 21 21.5 10.1 9.6 16 ...
## $ perc.alumni  : num  0 7 18 37 9 16 21 57 20 14 ...
## $ Expend       : num 5389 9773 5125 11758 9084 ...
## $ Grad.Rate    : num  36 43 56 84 84 54 64 77 93 66 ...
```

The task here is to fit densely connected neural networks using the package `keras` in order to predict `Outstate`.

### a) (2P)

Preprocessing is important before we fit a neural network. Apply feature-wise normalization to the predictors (but not to the response!).

```
# Feature-wise normalization added to the predictors (not the response).
train.target <- college.train$Outstate
college.train <- subset(college.train, select = -c(Outstate))
test.target <- college.test$Outstate
college.test <- subset(college.test, select = -c(Outstate))

mean <- apply(college.train, 2, mean)
std <- apply(college.train, 2, sd)
college.train <- scale(college.train, center = mean, scale = std)
college.test <- scale(college.test, center = mean, scale = std)
```

### b) (2P)

Write down the equation which describes a network that predicts `Outstate` with 2 hidden layers and `relu` activation function with 64 units each. What activation function will you choose for the output layer?

The equation which describes a network that predicts `Outstate` with 2 hidden layers and `relu` activation function with 64 units each, is

$$\hat{y}_1(\mathbf{x}) = \beta_{01} + \sum_{m=1}^{64} \beta_{m1} \max(\gamma_{0m} + \sum_{l=1}^{64} \gamma_{lm} \max(\alpha_{0l} + \sum_{j=1}^{17} \alpha_{jl} x_j, 0), 0)$$

Since `Outstate` is a continuous variable, I would use a linear activation function for the output layer (for regression).

### c) (3P)

- (i) Train the network from b) for the training data using the library `keras`; use 20% of the training data as your validation subset (1P).

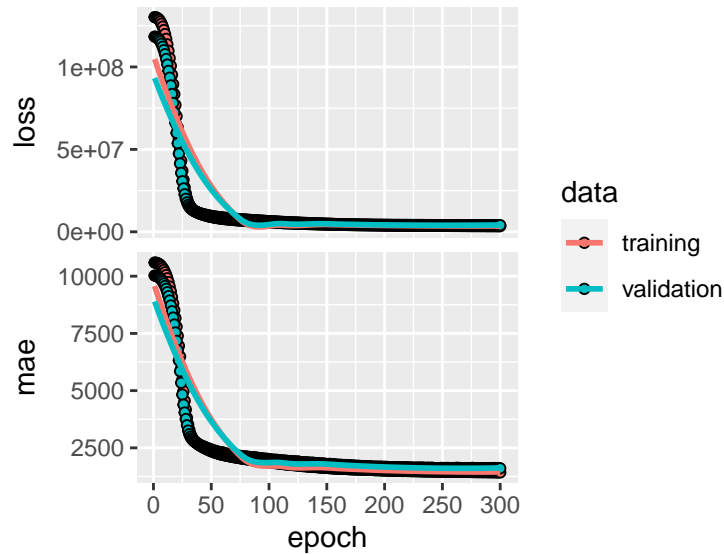
```
set.seed(123)
# Build the model
model <- keras_model_sequential() %>% layer_dense(units = 64, activation = "relu",
  input_shape = ncol(college.train)) %>% layer_dense(units = 64, activation = "relu") %>%
  layer_dense(units = 1)

model %>% compile(optimizer = "rmsprop", loss = "mse", metrics = c("mae"))

# Train
history <- model %>% fit(college.train, train.target, epochs = 300, batch_size = 8,
  validation_split = 0.2, verbose = 0)
```

- (ii) Plot the training and validation error as a function of the epochs (1P).

```
plot(history)
```



(iii) Report the MSE of the test set and compare it with methods that you used in Compulsory 2 (1P).

```
result <- model %>% evaluate(college.test, test.target, verbose = 0)
result[1]
```

```
##      loss
## 3614758
```

#### d) (2P)

Apply one of the regularization techniques you heard about in the course (easiest to use dropout or weight decay with L1/L2 norms). Does this improve the performance of the network? Please again use `set.seed(123)` to make results comparable.

```
set.seed(123)

# Regularization with weight decay with l2. Build the model
model.kernel.reg <- keras_model_sequential() %>% layer_dense(units = 64, activation = "relu",
  input_shape = ncol(college.train), kernel_regularizer = regularizer_l2(l = 0.001)) %>%
  layer_dense(units = 64, activation = "relu", kernel_regularizer = regularizer_l2(l = 0.001)) %>%
  layer_dense(units = 1)

model.kernel.reg %>% compile(optimizer = "rmsprop", loss = "mse", metrics = c("mae"))

# Train
history.kernel.reg <- model.kernel.reg %>% fit(college.train, train.target, epochs = 300,
  batch_size = 8, validation_split = 0.2, verbose = 0)

# Result for weight decay with l1.
result.kernel.reg <- model.kernel.reg %>% evaluate(college.test, test.target, verbose = 0)
result.kernel.reg[1]
```

```
##      loss
## 3826504
```

```
# Regularization with dropout. Build the model
model.drop.reg <- keras_model_sequential() %>% layer_dense(units = 64, activation = "relu",
```

```

input_shape = ncol(college.train)) %>% layer_dropout(rate = 0.4) %>% layer_dense(units = 64,
activation = "relu") %>% layer_dropout(rate = 0.4) %>% layer_dense(units = 1)

model.drop.reg %>% compile(optimizer = "rmsprop", loss = "mse", metrics = c("mae"))

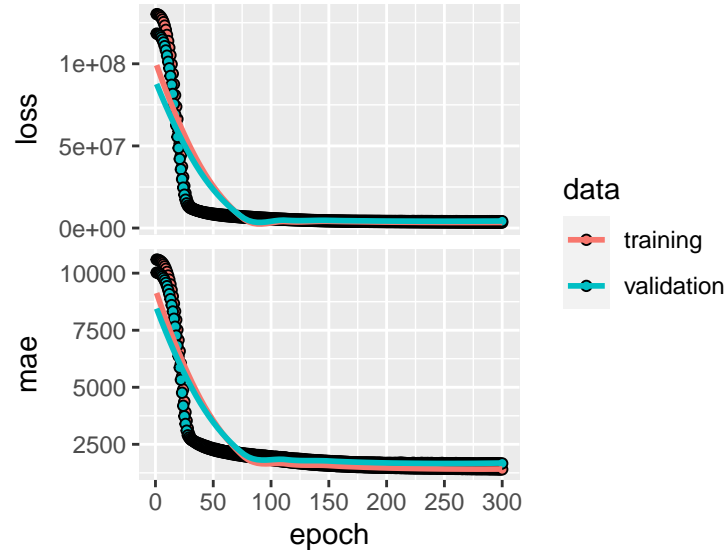
# Train
history.drop.reg <- model.drop.reg %>% fit(college.train, train.target, epochs = 300,
batch_size = 8, validation_split = 0.2, verbose = 0)

# Result for dropout..
result.drop.reg <- model.drop.reg %>% evaluate(college.test, test.target, verbose = 0)
result.drop.reg[1]

##      loss
## 3525276

par(mfrow = c(1, 2))
plot(history.kernel.reg)

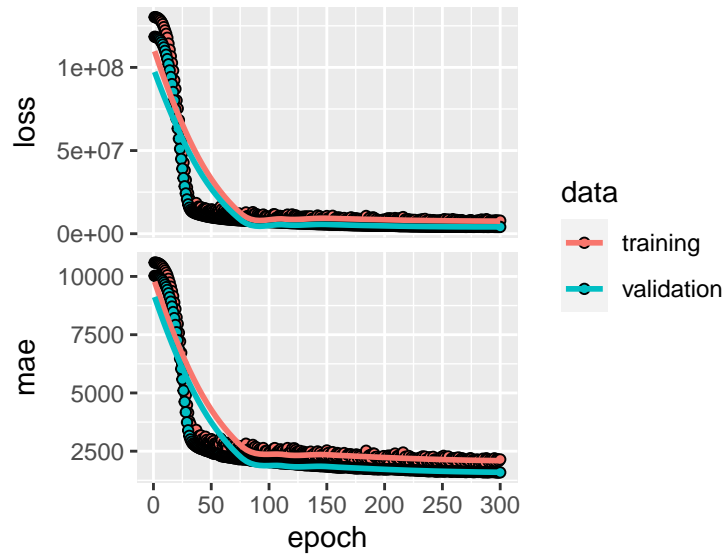
```



```

plot(history.drop.reg)

```



```
par(mfrow = c(1, 1)) # reset plotting grid.
```

The dropout seems to improve the model fit, while the regularization with weight decay with L2-norm does not seem to boost the performance compared to no use of regularization (from earlier).

## Problem 2 (10P)

In this problem, we will use a real dataset of individuals with the Covid-19 infection. The data were downloaded from <https://www.kaggle.com/shirmani/characteristics-corona-patients> on 30. March 2020, and have only been cleaned for the purpose of this exercise. The dataset consists of 2010 individuals and four columns,

- **deceased:** if the person died of corona (1:yes, 0:no)
- **sex:** male/female
- **age:** age of person (ranging from 2 years to 99 years old)
- **country:** which country the person is from (France, Japan, Korea or Indonesia)

Note that the conclusions we will draw here are probably not scientifically valid, because we do not have enough information about how data were collected.

Load your data into R using the following code:

```
id <- "1CA1RPRYqU9oTlaHfSroitnWrI6WpUeBw" # google file ID
d.corona <- read.csv(sprintf("https://docs.google.com/uc?id=%s&export=download",
  id), header = T)
```

### a) Inspecting your data (1P)

Inspect the data by reporting **tables** for

- the number of deceased for each country,
- the number of deceased for each sex, and
- for each country: the number of deceased, separate for each sex.

```
table(d.corona$country, d.corona$deceased)
```

```
##
##           0    1
##   France   100  14
##   indonesia  67   2
##   japan    291   3
##   Korea    1507  26

table(d.corona$sex, d.corona$deceased)

##
##           0    1
##   female 1075  14
##   male   890  31

France <- d.corona[which(d.corona$country == "France"), ]
Japan <- d.corona[which(d.corona$country == "japan"), ]
Korea <- d.corona[which(d.corona$country == "Korea"), ]
Indonesia <- d.corona[which(d.corona$country == "indonesia"), ]

table(France$sex, France$deceased)

##
##           0    1
##   female  55   5
##   male   45   9

table(Japan$sex, Japan$deceased)

##
##           0    1
##   female 120   0
##   male   171   3

table(Korea$sex, Korea$deceased)

##
##           0    1
##   female 871   8
##   male   636  18

table(Indonesia$sex, Indonesia$deceased)

##
##           0    1
##   female  29   1
##   male   38   1
```

## b) Multiple choice (2P)

Answer the following multiple choice questions by using the data above to model the probability of deceased as a function of **sex**, **age** and **country** (with France as reference level; no interactions).

Which of the following statements are true, which false?

```
glm.fit <- glm(deceased ~ ., family = "binomial", data = d.corona)
summary(glm.fit)
```

```
##
## Call:
```

```
## glm(formula = deceased ~ ., family = "binomial", data = d.corona)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.2797  -0.1855  -0.1009  -0.0553   3.2233
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -7.633051   0.897063  -8.509 < 2e-16 ***
## sexmale         1.137246   0.343706   3.309 0.000937 ***
## age             0.068012   0.009846   6.907 4.94e-12 ***
## countryindonesia -0.754259   0.815127  -0.925 0.354796
## countryjapan    -2.434101   0.667826  -3.645 0.000268 ***
## countryKorea    -1.366797   0.374837  -3.646 0.000266 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 430.92  on 2009  degrees of freedom
## Residual deviance: 321.07  on 2004  degrees of freedom
## AIC: 333.07
##
## Number of Fisher Scoring iterations: 8
```

(i) Country is not a relevant variable in the model.

Answer: FALSE

(ii) The slope for indonesia has a large  $p$ -value, which shows that we should remove the Indonesian population from the model, as they do not fit the model as well as the Japanese population.

Answer: FALSE

(iii) Increasing the age by 10 years,  $x_{age}^* = x_{age} + 10$ , and holding all other covariates constant, the odds ratio to die increases by a factor of 1.97.

Answer: TRUE

(iv) The probability to die is approximately 3.12 larger for males than for females.

Answer: Kom hit ;)

### c) (1P)

Create a plot of probabilities to die of coronavirus as a function of age, separately for the two sexes and each country.

Hints:

- Make one plot and add lines for each country/sex.
- A useful function to generate gridded data for prediction is `expand.grid()`. For example `newdata = expand.grid(sex="male", age= seq(20,100,1) ,country="France")` generates a grid for males in France over a range of ages between 20 and 100.

### d) (3P)

As a statistician working on these data, you are asked the following questions:

(i) Have males generally a higher probability to die of coronavirus than females?

- (ii) Is age a greater risk factor for males than for females?
- (iii) Is age a greater risk factor for the French population than for the Korean population?

Answer the questions by fitting appropriate models (1P each).

### e) Interpret your model (1P)

According to your model fitted in part b), it looks like the French population is at a much higher risk of dying from Covid-19 than the other countries. Do you trust this result? How could it be influenced by the way the data were collected?

### f) Multiple choice (2P)

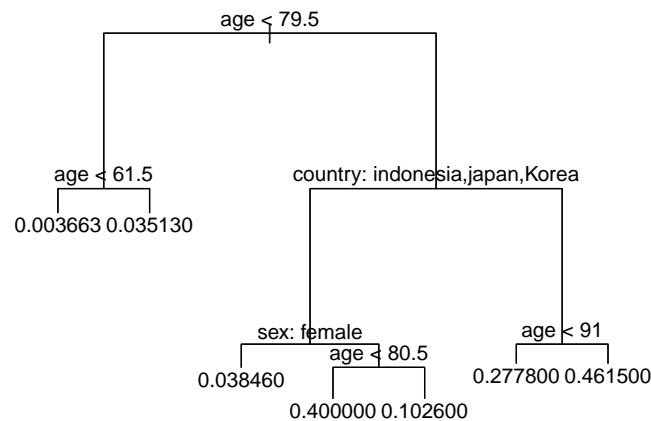
Which of the following statements are true, which false?

Consider the classification tree below to answer:

- (i) The probability of dying (`deceased = 1`) is about 0.46 for a French person with age above 91.
- (ii) Age seems to be a more important predictor for mortality than sex.

Consider the LDA code and output below:

- (iii) The “null rate” for misclassification is 2.24%, because this is the proportion of deaths among all cases in the dataset. No classifier should have a higher misclassification rate.
- (iv) LDA is not a very useful method for this dataset, among other reasons because it does not estimate probabilities, but also because the misclassification error is too high.



```
library(MASS)
table(predict = predict(lda(deceased ~ age + sex + country, data = d.corona))$class,
      true = d.corona$deceased)
```

```
##      true
## predict  0    1
##         0 1926  31
##         1   39  14
```



## Problem 3 (14P)

The `d.support` dataset (source *F. E. Harrell, Regression Modeling Strategies*) contains the total hospital costs of 9105 patients with certain diseases in American hospitals between 1989 and 1991. The different variables are

Variable	Meaning
<code>totcst</code>	Total costs
<code>age</code>	Age of the patients
<code>dzgroup</code>	' Disease group
<code>num.co</code>	Number of co-morbidities
<code>edu</code>	Years of education
<code>scoma</code>	Measure for Glasgow coma scale
<code>income</code>	Income
<code>race</code>	Rasse
<code>meanbp</code>	Mean blood pressure
<code>hrt</code>	Heart rate
<code>resp</code>	Respiratory frequency
<code>temp</code>	Body temperature
<code>pafi</code>	PaO2/FiO2 proportion (blood-gas mixture)

Data are loaded as follows (and we reduce the number of patients to the 4960 complete cases with total costs larger than 0):

```
id <- "1heRtzi8vBoBGMaM2-ivBQI5Ki3HgJTm0" # google file ID
d.support <- read.csv(sprintf("https://docs.google.com/uc?id=%s&export=download",
  id), header = T)
# We only look at complete cases
d.support <- d.support[complete.cases(d.support), ]
d.support <- d.support[d.support$totcst > 0, ]
```

We would like to build models that help us to understand which predictors are mostly driving the total cost, but also models for prediction.

### a) (1P)

Before we start analysing the data, visualize the distributions of all continuous or integer variables with histograms. Suggest a transformation for the response variable `totcst` (hint: it is a *standard transformation* that we have used earlier in the course). Important: **you should fit all models with the transformed version of the response variable `totcst` from now on. Leave all other variables untransformed.**

### b) (3P)

Fit a multiple linear regression model with the six covariates `age`, `temp`, `edu`, `resp`, `num.co` and `dzgroup` and the (transformed version of the) response `totcst`.

- How much/by which factor are the total costs expected to change when a patient's age increases by 10 years, given that all other characteristics of the patient are the same? Use the transformed response to fit the model, but report the result on the original (back-transformed) scale of the response. (1P)
- Do a residual analysis using the Tukey-Anscombe plot and the QQ-diagram. Are the assumptions fulfilled? (1P)
- Does the effect of age depend on the disease group? Do a formal test and report the  $p$ -value. (1P)

### c) (3P)

In order to build a more robust model for inference and prediction of the total costs, continue using ridge regression. Create a training set with 80% of the data and a test set with the remaining 20% (1P). Run cross-validation to find the largest value of  $\lambda$  such that the error is within 1 standard error of the smallest  $\lambda$  (1P). Report the test MSE of the ridge regression where you used the respective  $\lambda$  (1P).

Be careful: we still use the same transformation for the response as in b) – you should report the MSE using the transformed version of `totcst` (i.e., do **not back-transform** the MSE to the original scale).

**R-hints:**

```
set.seed(12345)
train.ind = sample(1:nrow(d.support), ... * nrow(d.support))
d.support.train = d.support[... , ]
d.support.test = d.support[... , ]
```

### d) (3P)

Now assume that our sole aim is prediction. In the course you heard about *partial least squares (PLS)*. It is a smart approach that uses the principal component regression idea, but finds the components that are best correlated with the response.

Proceed as follows:

- (i) Run a PLS regression (don't forget to scale the variables, `scale=TRUE`) (1P).
- (ii) Choose an optimal number of principal components (PCs) using cross-validation (1P).
- (iii) Report the MSE of the test set when using the respective set of PCs and compare to the result from ridge regression. Conclusion? (1P)

### e) (4P)

Now choose two other methods that you know from the course and try to build models with even lower test MSEs than those found so far (imagine that this is a competition where the lowest test MSE wins). Use the same training and test dataset as generated above. And remember that we are still *always* working with the transformed version of the response variable (`totcst`). In particular, use

- (i) One model that involves non-linear transformations of the covariates (e.g., splines, natural splines, polynomials etc) that are combined to a GAM (2P).
- (ii) One model/method based on regression trees (2P).

Very briefly discuss or explain your choices (1-2 sentences each).

## Problem 4 (Mixed questions; 6P)

### a) 2P

We look at the following cubic regression spline model:

$$Y = \begin{cases} \beta_0 + \beta_1x + \beta_2x^2 + \beta_3x^3 + \epsilon, & \text{if } x \leq 1, \\ \beta_0 + \beta_1x + \beta_2x^2 + \beta_3x^3 + \beta_4(x-1)^3 + \epsilon, & \text{if } 1 < x \leq 2, \\ \beta_0 + \beta_1x + \beta_2x^2 + \beta_3x^3 + \beta_4(x-1)^3 + \beta_5(x-2)^3 + \epsilon, & \text{if } x > 2. \end{cases}$$

Write down the basis functions (1P) and the design matrix (1P) of this model.

## b) Multiple choice - 2P

Inference vs prediction: Which of the following methods are suitable when the aim of your analysis is inference?

- (i) Lasso and ridge regression
- (ii) Multiple linear regression with interaction terms
- (iii) Logistic regression
- (iv) Support Vector Machines

## c) Multiple choice - 2P

We again look at the Covid-19 dataset from Problem 2 to study some properties of the bootstrap method. Below we estimated the standard errors of the regression coefficients in the logistic regression model with `sex`, `age` and `country` as predictors using 1000 bootstrap iterations (column `std.error`). These standard errors can be compared to those that we obtain by fitting a single logistic regression model using the `glm()` function. Look at the R output below and compare the standard errors that we obtain from these two approaches (note that the `t1*` to `t6*` variables are sorted in the same way as for the `glm()` output).

```
library(boot)
boot.fn <- function(data, index) {
  return(coefficients(glm(deceased ~ sex + age + country, family = "binomial",
    data = data, subset = index)))
}
boot(d.corona, boot.fn, 1000)
```

```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = d.corona, statistic = boot.fn, R = 1000)
##
##
## Bootstrap Statistics :
##      original      bias    std. error
## t1* -7.63305130 -0.1263323277 0.767744824
## t2*  1.13724644  0.0206113065 0.356972527
## t3*  0.06801169  0.0007610352 0.008393884
## t4* -0.75425940 -2.1964990968 5.423256337
## t5* -2.43410057 -0.9420486455 3.703570262
## t6* -1.36679680  0.0060403860 0.386818463
```

```
# Logistic regression
r.glm <- glm(deceased ~ sex + age + country, d.corona, family = "binomial")
summary(r.glm)$coef
```

```
##              Estimate Std. Error    z value    Pr(>|z|)
## (Intercept) -7.63305130 0.897063042 -8.5089352 1.755379e-17
## sexmale      1.13724644 0.343705727  3.3087794 9.370363e-04
## age          0.06801169 0.009846377  6.9072806 4.940322e-12
## countryindonesia -0.75425940 0.815127165 -0.9253273 3.547957e-01
## countryjapan   -2.43410057 0.667826265 -3.6448111 2.675883e-04
## countryKorea   -1.36679680 0.374836917 -3.6463772 2.659635e-04
```

Which of the following statements are true?

- (i) There are large differences between the estimated standard errors, which indicates a problem with the bootstrap.
- (ii) The differences between the estimated standard errors indicate a problem with the assumptions taken about the distribution of the estimated parameters in logistic regression.
- (iii) The `glm` function leads to too small  $p$ -values for the differences between countries, in particular for the differences between Indonesia and France and between Japan and France.
- (iv) The bootstrap relies on random sampling the same data without replacement.

## Problem 5 (Multiple and single choice questions; 11P)

### a) Multiple choice - 2P

Which of the following are techniques for regularization?

- (i) Lasso
- (ii) Ridge regression
- (iii) Forward and backward selection
- (iv) Stochastic gradient descent

### b) Multiple choice - 2P

Which of the following statements about principal component regression (PCR) and partial least squares (PLS) are correct?

- (i) PCR involves the first principal components that are most correlated with the response.
- (ii) PLS involves the first principal components that are most correlated with the response.
- (iii) The idea in PLS is that we choose the principal components that explain most variation among all covariates.
- (iv) The idea in PCR is that we choose the principal components that explain most variation among all covariates.

### c) Single choice - 1P

In ridge regression, we estimate the regression coefficients in a linear regression model by minimizing

$$\sum_{i=1}^n \left( y_i - \beta_0 - \sum_{j=1}^p \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^p \beta_j^2 .$$

What happens when we increase  $\lambda$  from 0? Choose the single correct statement:

- (i) The training RSS will steadily decrease.
- (ii) The test RSS will steadily decrease.
- (iii) The test RSS will steadily increase.
- (iv) The bias will steadily increase.
- (v) The variance of the estimator will steadily increase.

### d) Single choice - 1P

Which statement about the *curse of dimensionality* is correct?

- (i) It means that we have a bias-variance tradeoff in  $K$ -nearest neighbor regression, where large  $K$  leads to more bias but less variance for the predictor function.
- (ii) It means that the performance of the  $K$ -nearest neighbor classifier gets worse when the number of predictor variables  $p$  is large.

- (iii) It means that the  $K$ -means clustering algorithm performs bad if the datapoints lie in a high-dimensional space.
- (iv) It means that support vector machines with radial kernel function should be avoided, because radial kernels correspond to infinite-dimensional polynomial boundaries.
- (v) It means that we should never measure too many covariates when we want to do classification.

### e) Single choice - 1P

Now assume you have 10 covariates,  $X_1$  to  $X_{10}$ , each of them uniformly distributed in the interval  $[0, 1]$ . To predict a new test observation  $(X_1^{(0)}, \dots, X_{10}^{(0)})$  in a  $K$ -nearest neighbor (KNN) clustering approach, we use all observations within 20% 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?

- (i)  $1.02 \cdot 10^{-7}$
- (ii)  $2.0 \cdot 10^{-3}$
- (iii) 0.20
- (iv) 0.04
- (v)  $10^{-10}$

### f) Multiple choice - 2P

This example is taken from a real clinical study by *Ikeda, Matsunaga, Irabu, et al. Using vital signs to diagnose impaired consciousness: cross sectional observational study. BMJ 2002;325:800*. Researchers investigated the use of vital signs as a screening test to identify brain lesions in patients with impaired consciousness. The setting was an emergency department in Japan. The study included 529 consecutive patients that arrived with consciousness. Patients were followed until discharge. The vital signs of systolic and diastolic blood pressure and pulse rate were recorded on arrival. The aim of this study was to find a quick test for assessing whether the newly arrived patient suffered from a brain lesion. While vital signs can be measured immediately, the actual diagnosis of a brain lesion can only be determined on the basis of brain imaging and neurological examination at a later stage, thus the quick measurements of blood pressure and heart rate are important to make a quick assessment. In total, 312 patients (59%) were diagnosed with a brain lesion.

The performance of each vital sign (systolic blood pressure, diastolic blood pressure and heart rate) was separately evaluated as a screening test to quickly diagnose brain lesions. To assess the quality of each of these vital signs, different thresholds were taken successively to discriminate between “negative” and “positive” screening test result. For each vital sign and each threshold the sensitivity and specificity were derived and used to plot a receiver operating characteristic (ROC) curve for the vital sign (Figure 1):

Which of the following statements are true?

- (i) The value of 1-specificity represents the proportion of patients without a diagnosed brain lesion identified as positive on screening.
- (ii) When we use different cut-offs, sensitivity increases at the cost of lower specificity, and vice versa.
- (iii) A perfect diagnostic test has an AUC of 0.5.
- (iv) The vital sign that is most suitable to distinguish between patients with and without brain lesion is systolic blood pressure.

### g) Multiple choice - 2P

We study the `decathlon2` dataset from the `factoextra` package in R, where Athletes' performance during a sporting meeting was recorded. We look at 23 athletes and the results from the 10 disciplines in two competitions. Some rows of the dataset are displayed here:

```
decathlon2.active[c(1, 3, 4), ]

##      100m long_jump shot_put high_jump 400m 110.hurdle discus pole_vault
## SEBRLE 11.04      7.58    14.83     2.07 49.81      14.69 43.75      5.02
```

```
## BERNARD 11.02      7.23      14.25      1.92 48.93      14.99 40.87      5.32
## YURKOV  11.34      7.09      15.19      2.10 50.42      15.31 46.26      4.72
##          javeline 1500m
## SEBRLE   63.19 291.7
## BERNARD   62.77 280.1
## YURKOV    63.44 276.4
```

From a principal component analysis we obtain the biplot given in Figure 2.

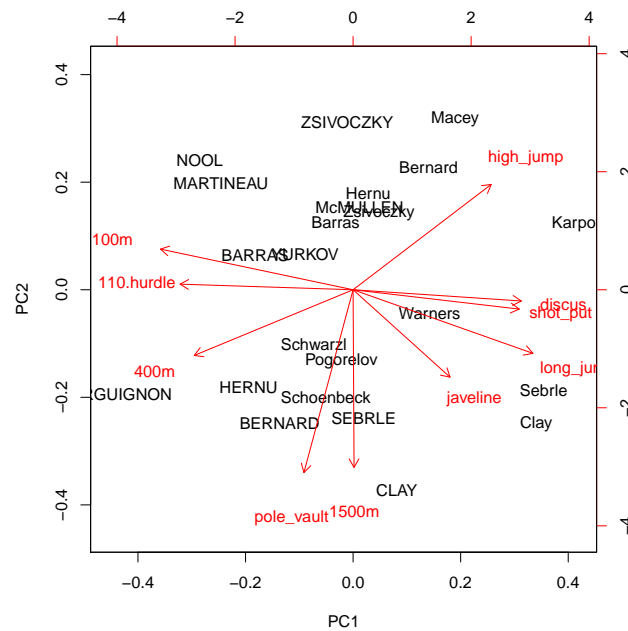


Figure 1: Figure for question 5g).

Which of the following statements are true, which false?

- (i) The athlete named CLAY seems to be one of the fastest 1500m runners.
- (ii) Athletes that are good in 100m tend to be also good in long jump.
- (iii) The first principal component has the highest loadings for 100m and long jump.
- (iv) 110m hurdle has a very small loading for PC2.