Compulsory Exercise 1

TMA4268 Statistical Learning V2021

Emma Skarstein, Michail Spitieris, Stefanie Muff, Department of Mathematical Sciences, NTNU

Hand out date: February 8, 2021

The submission deadline is Monday, 22. February 2021, 23:59h using Blackboard

Introduction

Maximal score is 50 points. Your score will make up 20% points of your final grade.

Supervision

Supervisions will be via Whereby by the teaching assistants and the lecturer during the usual lecture and exercise hours on February 15th and 16th. Ideally, only one member of a group comes with the questions that the group compiled. This ensures that all groups have the opportunity to ask questions. More information is as always available from the course website.

General supervision:

- Monday, February 15, 14.15-16.00
- Tuesday February 15, 10.15-12.00
- Tuesday February 16, 16.15-18.00

Remember that there is also the Piazza forum, and we strongly encourage you to use it for your questions - this ensures that all other students benefit from the answers.

Practical issues (Please read carefully)

- Group size is 2 or 3 - join a group (self enroll) before handing in on Bb. We prefer that you do not work alone.
- Please organize yourself via the Piazza forum to find a group. Once you formed a group, log into Blackboard and add yourself to the same group there.
- If you did not find a group even when using Piazza, you can email Stefanie (stefanie.muff@ntnu.no) and I will try to match you with others that are alone (please use this really only if you have already tried).
- Remember to write your names and group number on top of your submission.
- The exercise should be handed in as **one R Markdown file and a pdf-compiled version** of the R Markdown file (if you are not able to produce a pdf-file directly please make an html-file, open it in your browser and save as pdf - no, not landscape - but portrait please). We will read the pdf-file and use the Rmd file in case we need to check details in your submission.
- You may want to work through the R Mardown bonus part in the R course (https://digit.ntnu.no/courses/course-v1:NTNU+IMF001+2020/about)
- In the R-chunks please use both `echo=TRUE` and `eval=TRUE` to make it simpler for us to read and grade.

- Please do not include all the text from this file (that you are reading now) - we want your R code, plots and written solutions - use the template from the course page (https://wiki.math.ntnu.no/tma4268/2021v/subpage6).
- Please **not more than 12 pages** in your pdf-file! (This is a request, not a requirement.)
- Please save us time and **do not submit word or zip**, and do not submit only the Rmd. This only results in extra work for us!

R packages

You need to install the following packages in R to run the code in this file.

```
install.packages("knitr")  #probably already installed
install.packages("rmarkdown")  #probably already installed
install.packages("ggplot2")  #plotting with ggplot
install.packages("ggfortify")
install.packages("MASS")
install.packages("class")
install.packages("pROC")
install.packages("plotROC")
```

Multiple/single choice problems

There will be a few *multiple choice questions*. This is how these will be graded:

- **Multiple choice questions (2P)**: There are four choices, and each of them can be TRUE or FALSE. If you make one mistake (either wrongly mark an option as TRUE/FALSE) you get 1P, if you have two or more mistakes, you get 0P. Your answer should be given as a list of answers, like TRUE, TRUE, FALSE, FALSE, for example.

Problem 1 – 11P

We consider the following regression problem

$$Y = f(\mathbf{x}) + \varepsilon, \text{ where } \mathrm{E}(\varepsilon) = 0 \text{ and } \mathrm{Var}(\varepsilon) = \sigma^2.$$

Assume now that the true function $f(\mathbf{x})$ is a linear combination of the observed covariates, that is $f(\mathbf{x}) = \mathbf{x}^T\boldsymbol{\beta} = \beta_0 + \beta_1 x_1 + \ldots + \beta_p x_p$, where $\mathbf{x}$ and $\boldsymbol{\beta}$ are both vectors of length $p+1$.

We know that the OLS estimator in this case is $\hat{\boldsymbol{\beta}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}$, with design matrix $\mathbf{X}$ and response vector $\mathbf{Y}$. We look now at a competing estimator $\widetilde{\boldsymbol{\beta}} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{Y}$ (ridge regression estimator) where $\lambda \geq 0$ is a constant tuning parameter and controls the bias-variance trade-off. Observe that for $\lambda = 0$ the ridge regression estimator is equivalent to $\hat{\boldsymbol{\beta}}$.

We will first derive mathematical formulas for the bias and variance and then we will plot these curves in R.

a) (1P)

Find the expected value and the variance-covariance matrix of $\widetilde{\boldsymbol{\beta}}$.

b) (2P)

Let $\widetilde{f}(\mathbf{x}_0) = \mathbf{x}_0^T\widetilde{\boldsymbol{\beta}}$ be the prediction at a new covariate vector $\mathbf{x}_0$. Using a) find the expected value and variance for $\widetilde{f}(\mathbf{x}_0) = \mathbf{x}_0^T\widetilde{\boldsymbol{\beta}}$.

c) (2P)

Find the expected MSE at $\mathbf{x}_0$, $E[(y_0 - \widetilde{f}(\mathbf{x}_0))^2]$.

Hint: Use that $\mathrm{E}[(y_0 - \widetilde{f}(\mathbf{x}_0))^2] = [\mathrm{E}(\widetilde{f}(\mathbf{x}_0) - f(\mathbf{x}_0))]^2 + \mathrm{Var}(\widetilde{f}(\mathbf{x}_0)) + \mathrm{Var}(\varepsilon)$

Plotting the bias-variance trade-off

The estimator $\widetilde{\boldsymbol{\beta}}$ is a function of the tuning parameter λ, which controls the bias-variance trade-off. Using the decomposition derived in c) we will plot the three elements (bias, variance and irreducible error) using the values in the code chunk below. Values is a list with the design matrix `X`, the vector $\mathbf{x}_0$ as `x0`, the parameters vector `beta` and the irreducible error `sigma`.

```
id <- "1X_8OKcoYbng1XvYFDirxjEWr7LtpNr1m"  # google file ID
values <- dget(sprintf("https://docs.google.com/uc?id=%s&export=download", id))

X = values$X
dim(X)
```

```
## [1] 100  81
```

```
x0 = values$x0
dim(x0)
```

```
## [1] 81  1
```

```
beta = values$beta
dim(beta)
```

```
## [1] 81  1
```

```
sigma = values$sigma
sigma
```

```
## [1] 0.5
```

d) (2P)

First we will create the squared bias function (`bias`) which takes as inputs the parameter `lambda`, `X`, `x0`, `beta` and returns the squared bias. You have only to fill the `value =` and run the chunk of code to plot the squared bias, where value is the squared bias as derived in c). Comment on what you see.

```
library(ggplot2)
bias = function(lambda, X, x0, beta) {
    p = ncol(X)
    value = ...
    return(value)
}
lambdas = seq(0, 2, length.out = 500)
BIAS = rep(NA, length(lambdas))
```

```
for (i in 1:length(lambdas)) BIAS[i] = bias(lambdas[i], X, x0, beta)
dfBias = data.frame(lambdas = lambdas, bias = BIAS)
ggplot(dfBias, aes(x = lambdas, y = bias)) + geom_line(color = "red") + xlab(expression(lambda)) +
    ylab(expression(bias^2))
```

e) (2P)

Now we will create the variance function which takes the same inputs as the squared bias. As in d) you have to fill only the `value =` and run the code to plot the variance. Comment on what you see.

```
variance = function(lambda, X, x0, sigma) {
    p = ncol(X)
    inv = solve(t(X) %*% X + lambda * diag(p))
    value = ...
    return(value)
}
lambdas = seq(0, 2, length.out = 500)
VAR = rep(NA, length(lambdas))
for (i in 1:length(lambdas)) VAR[i] = variance(lambdas[i], X, x0, sigma)
dfVar = data.frame(lambdas = lambdas, var = VAR)
ggplot(dfVar, aes(x = lambdas, y = var)) + geom_line(color = "green4") + xlab(expression(lambda)) +
    ylab("variance")
```

f) (2P)

Fill in the exp_mse of the following code to calculate the expected MSE (left hand side of Hint in c)) for all the lambda values that you plugged in above (you have used λ between 0 and 2, that is `lambdas = seq(0, 2, length.out = 500)` in the hints above). Run the code to plot all the components of the expected MSE and find the value of λ which minimizes it.

```
exp_mse = ...
lambdas[which.min(exp_mse)]
```

```
dfAll = data.frame(lambda = lambdas, bias = BIAS, var = VAR, exp_mse = exp_mse)
ggplot(dfAll) + geom_line(aes(x = lambda, y = exp_mse), color = "blue") + geom_line(aes(x = lambda,
    y = bias), color = "red") + geom_line(aes(x = lambda, y = var), color = "green4") +
    xlab(expression(lambda)) + ylab(expression(E(MSE)))
```

Run the following code to plot the decomposition and comment on what you see.

When you create the pdf file don't forget to change `eval=FALSE` to `eval=TRUE`.

Problem 2 – 13P

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 March 30 2020, and have only been **modified and 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:

```
# read file
id <- "1yYlEl5gYY3BEtJ4d7KWaFGIOEweJIn__"  # google file ID
d.corona <- read.csv(sprintf("https://docs.google.com/uc?id=%s&export=download",
    id), header = T)
```

a) Inspecting your data (2P)

Inspect the data by reporting **tables** for

- the number of deceased and non-deceased
- for each country the number of males and females
- the number of deceased and non-deceased for each sex, and
- the number of deceased and non-deceased in France, separate for each sex.

Each correct table gives 0.5P.

b) (5P)

Fit an appropriate regression model to understand and/or predict how sex, age and the country of residence influence the probability to decease of covid-19. Use all variables in the model, but do not use any variable transformations or interactions.

Answer the following questions by doing the appropriate analyses and **giving a short explanation**. Points can only be given with the correct explanation.

(i) What is the probability to die of covid for a male age 75 living in Korea?
(ii) Is there evidence that males have higher probability to die than females?
(iii) Is there evidence that the country of residence has an influence on the probability to decease?
(iv) Quantify how the odds to die changes when someone with otherwise identical covariates is 10 years older than another person.

c) (4P)

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

(i) Is age a greater risk factor for males than for females?

(ii) Is age a greater risk factor for the French population than for the Indonesian population?

Answer the questions by fitting appropriate models (1P each for the correct model) and giving the correct reasoning of your answer (1P per question). Important: Carefully think what "appropriate model" means. Do not throw away information in the data...

d) Multiple choice (2P)

We are looking and linear and quadratic discriminant analysis (LDA, QDA) to classify deceased versus non-deceased. In both LDA and QDA we use the three predictor variables. Say for each statement whether it is true or false.

(i) The "null rate" for misclassification is 5.22%, because this is the proportion of deaths among all cases in the dataset.

(ii) LDA is not a very useful method for this dataset.

(iii) LDA has a specificity of 1.

(iv) QDA has a lower sensitivity to classify deceased compared to LDA.

Problem 3 – 15P

In this problem, we will use a data set regarding the presence of *diabetes* from a population of women of Pima heritage in the US. For each woman we have we have the following information:

diabetes: 1 = present, 0 = not present

npreg: number of pregnacies

glu: plasma glucose concentration in an oral glucose tolerance test

bp: diastolic blood pressure (mm Hg)

skin: triceps skin fold thickness (mm)

bmi: body mass index

ped: diabetes pedigree function

age: age in years

The aim is to use the methods you have learned so far in order to make a classication rule for diabetes (or not) based on the available data. We will use the training set, `train` to fit the models and at the end we will use the test set, `test` to compare the fitted models. The training set consists of 300 observations, 200 non-diabetes and 100 diabetes cases and the testing set includes 232 observations, 155 non-diabetes and 77 diabetes cases.

```
# read file
id <- "1i1cQPeoLLC_FyAH0nnqCnnrSBpn05_h0"  # google file ID
diab <- dget(sprintf("https://docs.google.com/uc?id=%s&export=download", id))

t = MASS::Pima.tr2
train = diab$ctrain
test = diab$ctest
```

a) (3P)

We first fit a logistic regression model where the probability of diabetes is given by

$$P(y_i = 1|\mathbf{X} = \mathbf{x}_i) = p_i = \frac{e^{\beta_0+\beta_1 x_{i1}+\beta_2 x_{i2}+\cdots+\beta_7 x_{i7}}}{1 + e^{\beta_0+\beta_1 x_{i1}+\beta_2 x_{i2}+\cdots+\beta_7 x_{i7}}} ,$$

where $y_i = 1$ is presence of diabetes ($y_i = 0$ is non-presence), x_{i1} is the value of *npreg* for the ith observation, x_{i2} is the value of *glu*, et cetera.

```r
logReg = glm(diabetes ~ ., data = train, family = "binomial")
summary(logReg)
```

```
##
## Call:
## glm(formula = diabetes ~ ., family = "binomial", data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.8155  -0.6367  -0.3211   0.6147   2.2408
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -10.583538   1.428276  -7.410 1.26e-13 ***
## npreg         0.105109   0.062721   1.676 0.093775 .
## glu           0.035586   0.005892   6.039 1.55e-09 ***
## bp           -0.014654   0.013982  -1.048 0.294615
## skin          0.020379   0.020575   0.990 0.321962
## bmi           0.094683   0.031265   3.028 0.002458 **
## ped           1.931666   0.529573   3.648 0.000265 ***
## age           0.038291   0.020247   1.891 0.058594 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 381.91  on 299  degrees of freedom
## Residual deviance: 253.84  on 292  degrees of freedom
## AIC: 269.84
##
## Number of Fisher Scoring iterations: 5
```

(i) (1P) Show that the logit function (or the log-odds) $\text{logit}(p_i) = \log(\frac{p_i}{1-p_i})$ is a linear function of the covariates.

(ii) (2P) Use the `logReg` model to classify the test set observations to presence or non-presence of diabetes. Use a cut-off probabilty of 0.5 and create the confusion table. Calculate sensitivity and specificity.

b) (5P)

Now we will use LDA and QDA to estimate the probability of diabetes given the covariates. In LDA with K classes, we assign a class to a new observation based on the posterior probability

$$P(y = k|\mathbf{X} = \mathbf{x}) = \frac{\pi_k f_k(\mathbf{x})}{\sum_{l=1}^{K} \pi_l f_l(\mathbf{x})},$$

where

$$f_k(\mathbf{x}) = \frac{1}{(2\pi)^{p/2}|\mathbf{\Sigma}|^{1/2}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu_k})^T \mathbf{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu_k})}.$$

(i) (2P) Explain what $\pi_k, \boldsymbol{\mu}_k, \mathbf{\Sigma}$ and $f_k(x)$ are in the diabetes classification problem.

(ii) (3P) Fit LDA and QDA models and explain what is the difference between these two methods. Predict on the test data using 0.5 cut-off and create confusion tables.

c) (4P)

We will use now the KNN algorithm to classify the presence of diabetes in the data set.

(i) (1P) Explain how a new observation is classified in the KNN approach.

(ii) (1P) Explain briefly how you would choose the tuning parameter k.

(iii) (2P) Use the function `knn()` from the `class` library, with $k = 25$ to classify the presence of diabetes in the testing data. Give the confusion tables and derive sensitivity and specificity.

R-hints: In the `knn()` function set `prob=T` to ensure you get the class probabilities that you then need in d):

```
knnMod = knn(train = ..., test = ..., cl = ..., k = 25, prob = T)
```

d) (3P)

Produce ROC curves for all 3 models (LDA, QDA and KNN) in a single plot. The ROC curve should be generated for different cutoffs for the classification probabilities. Which model performs better? If the task is to create an interpretable model, which model would you choose?

R-hints:

- To obtain $P(y = 1)$ from the `knn()` output you have to be aware that the respective probabilities

```
attributes(knnMod)$prob
```

are the success probability for the actual class where the categorization was made. So if you want to get a vector for $P(y = 1)$, you have to use $1 - P(y = 0)$ for the cases where the categorization was 0:

```
probKNN = ifelse(knnMod == 0, 1 - attributes(knnMod)$prob, attributes(knnMod)$prob)
```

- You might find the functions `roc()` and `ggroc()` from the package `pROC` useful, but there are many ways to plot ROC curves.

Problem 4 – 6P

a) (4P)

To calculate the LOOCV statistic for N training data we typically have to fit N models and evaluate the error of each fit on the left out observation. This task can be computationaly intensive when N is large. Fortunately, for linear models there is formula for doing this with fitting just the full data model. Show that for the linear regression model $Y = X\beta + \varepsilon$ the LOOCV statistic can be computed by the following formula

$$\text{CV} = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{y_i - \hat{y}_i}{1 - h_i}\right)^2,$$

where $h_i = \mathbf{x}_i^T(X^TX)^{-1}\mathbf{x}_i$ and $\mathbf{x}_i^T$ is the ith row of X.

Hint 1: You need to calculate $\hat{y}_{(-i)}$, which is the predicted value of y when the ith observation is kept out. This can be written as $\hat{y}_{(-i)} = \mathbf{x}_i^T\hat{\boldsymbol{\beta}}_{(-i)}$.

Hint 2: $X_{(-i)}^TX_{(-i)} = X^TX - \mathbf{x}_i\mathbf{x}_i^T$, where $X_{(-i)}$ is the X matrix without the ith row.
Hint 2.1: Analogously $X_{(-i)}^T\mathbf{y}_{(-i)} = X^T\mathbf{y} - \mathbf{x}_iy_i$.

Hint 3: The Sherman–Morrison formula will be useful.

b) Multiple choice (2P)

Say for each statement whether it is true or false.

(i) The LOOCV will lead to more bias, but less variance than 10-fold CV in the estimated prediction error.
(ii) The formula from a) is valid for polynomial regression.
(iii) The formula from a) is not valid when we use the log transform of the response in linear regression.
(iv) The validation set-approach in the same as 2-fold CV.

Problem 5 – 5P

Load the bodyfat dataset that we used in the lecture to module 3:

```
id <- "19auu8YlUJJJUsZY8JZfsCTWzDm6doE7C"  # google file ID
d.bodyfat <- read.csv(sprintf("https://docs.google.com/uc?id=%s&export=download",
    id), header = T)
```

We are looking at the model where we model bodyfat as the response in a linear regression, with age, weight and bmi as predictor variables.

a) (1P)

Fit the respective regression model and report the R^2.

b) (4P)

You might have noticed that the R^2 is just a point estimate, without any uncertaint associated to it. We now use the bootstrap to estimate the uncertainty of the R^2 derived in a). To this end:

(i) Generate 1000 bootstrap samples of the R^2 (2P).
(ii) Plot the respective distribution of the values (0.5P).
(iii) Derive the standard error and the 95% confidence interval (1P).
(iv) Interpret what you see (0.5P).

R-hint: Use `set.seed(4268)` at the beginning of the bootstrap iterations to ensure that your results are reproducible.