Compulsory exercise 1: Group 5

TMA4268 Statistical Learning V2021

Hans Røhjell Odland and Aksel Haugen Madslien

2/9/2021

Problem 1

a)

We consider $Y = f(\mathbf{x}) + \varepsilon,$ where $\mathrm{E}(\varepsilon) = 0$ and $\mathrm{Var}(\varepsilon) = \sigma^2$.
We find the expected value for $\tilde{\beta}$ as

$$
\begin{aligned}
E(\tilde{\beta}) &= E[(\mathbf{x}^T\mathbf{x} + \lambda\mathbf{I})^{-1}\mathbf{x}^T\mathbf{y}] && (1)\\
&= (\mathbf{x}^T\mathbf{x} + \lambda\mathbf{I})^{-1}\mathbf{x}^T E[\mathbf{y}] && (2)\\
&= (\mathbf{x}^T\mathbf{x} + \lambda\mathbf{I})^{-1}\mathbf{x}^T\mathbf{x}\beta + \lambda\mathbf{I}\beta - \lambda\mathbf{I}\beta && (3)\\
&= (\mathbf{x}^T\mathbf{x} + \lambda\mathbf{I})^{-1}(-\lambda\mathbf{I})\beta + \mathbf{I}\beta && (4)\\
&= \beta - \lambda(\mathbf{x}^T\mathbf{x} + \lambda\mathbf{I})\beta && (5)
\end{aligned}
$$

b)

We let $\widetilde{f}(\mathbf{x}_0) = \mathbf{x}_0^T\widetilde{\boldsymbol{\beta}}$ The variance for $\widetilde{f}(\mathbf{x}_0)$ then becomes

$$
\begin{aligned}
E[\widetilde{f}(\mathbf{x}_0)] &= E[\mathbf{x}_0^T\widetilde{\boldsymbol{\beta}}] && (6)\\
&= \mathbf{x}_0^T E[\widetilde{\boldsymbol{\beta}}] && (7)\\
&= \mathbf{x}_0^T(\beta - \lambda(\mathbf{x}^T\mathbf{x} + \lambda\mathbf{I})\beta) && (8)
\end{aligned}
$$

For the variation we get

$$
\begin{aligned}
Var[\widetilde{f}(\mathbf{x}_0)] &= \mathbf{x}_0^T Var[\widetilde{\boldsymbol{\beta}}]\mathbf{x} && (9)\\
&= \mathbf{x}_0^T(\beta - \lambda(\mathbf{x}^T\mathbf{x} + \lambda\mathbf{I})\beta)\mathbf{x} && (10)
\end{aligned}
$$

c)

$$
\begin{aligned}
&\mathrm{E}[(y_0 - \widetilde{f}(\mathbf{x}_0))^2] = [\mathrm{E}(\widetilde{f}(\mathbf{x}_0) - f(\mathbf{x}_0))]^2 + \mathrm{Var}(\widetilde{f}(\mathbf{x}_0) && (11)\\
& && (12)
\end{aligned}
$$

d)

```
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
```

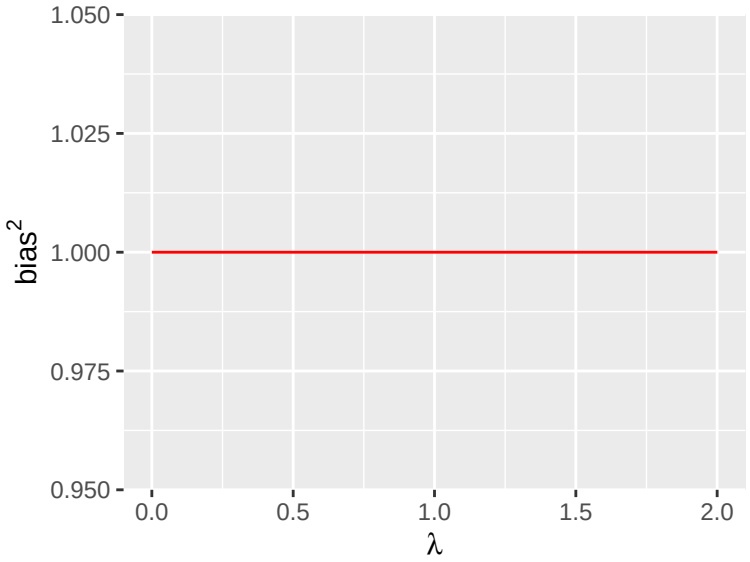
```
beta = values$beta
dim(beta)
```

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

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

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

```
bias = function(lambda, X, x0, beta) {
    p = ncol(X)
    value = -lambda * (t(X) %*% X %*% beta + lambda * beta)
    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))
```



Problem 2

a)

```
id <- "1yYlEl5gYY3BEtJ4d7KWaFGIOEweJIn__"  # google file ID

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

Number of deceased and non-deceased [Non-deceased = 0, deceased = 1]

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

```
##
##    0    1
## 1905  105
```

The number of males and females for each country

```
table(Country = d.corona$country, sex = d.corona$sex)
```

```
##           sex
## Country    female male
##   France       60   54
##   indonesia    30   39
##   japan       120  174
##   Korea       879  654
```

The number of deceased and non-deceased for each sex [Non-deceased = 0, deceased = 1]

```r
table(sex = d.corona$sex, deceased = d.corona$deceased)
```

```
##         deceased
## sex         0    1
##   female 1046   43
##   male    859   62
```

The number of deceased and non-deceased in France, separated for each sex [Non-deceased = 0, deceased = 1]

```r
francedf <- subset(d.corona, country == "France")
table(francedf$sex, francedf$deceased)
```

```
##
##           0  1
##   female 55  5
##   male   43 11
```

b)

i)

The covariates sex, country and age is included to inspect te probability of dying of covid at age 75 in Korea. The function is fitted and summarized to get the coefficients. To get the age of 75 we had to multiply the covariate for age with 75.

```r
fit <- lm(as.numeric(deceased) ~ sex + country + age, data = d.corona)
summary(fit)$coef  # show results
```

```
##                      Estimate   Std. Error   t value     Pr(>|t|)
## (Intercept)       0.043861643 0.0252284684  1.738577 8.226271e-02
## sexmale           0.030814710 0.0099018024  3.112030 1.884264e-03
## countryindonesia -0.053478088 0.0335835996 -1.592387 1.114555e-01
## countryjapan     -0.097524595 0.0242695290 -4.018397 6.076062e-05
## countryKorea     -0.071966403 0.0215415300 -3.340821 8.506420e-04
## age               0.001304608 0.0002180126  5.984098 2.571005e-09
```

```r
deceasedmale <- fit$coefficients[6] * 75 + fit$coefficients[2] + fit$coefficients[5]
```

The probability of dying of Covid-19 for a male at age 75 in Korea is found to be `5.669`%.

ii)

Do males have higher probability to die than females?

```r
fit <- glm(deceased ~ sex, data = d.corona, family = binomial)
summary(fit)$coef
```

```
##              Estimate Std. Error    z value     Pr(>|z|)
## (Intercept) -3.191529  0.1556015 -20.510908 1.720507e-93
## sexmale      0.562894  0.2037278   2.762971 5.727782e-03
```

The estimate readings for men dying of corona is positive. At the same time the p-value for age is significant. This means that we can conclude that men have a higher probability of dying of corona than women.

iii)

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

```
##                     Estimate Std. Error   z value     Pr(>|z|)
## (Intercept)      -1.8123788  0.2696369 -6.721552 1.797990e-11
## countryindonesia -0.7370664  0.5369628 -1.372658 1.698586e-01
## countryjapan     -1.4351729  0.4088358 -3.510389 4.474509e-04
## countryKorea     -1.1833535  0.2951062 -4.009925 6.073807e-05
```

From these readings we can conclude that there is not enough evidence to say that there is a higher risk of dying of corona in Indonesia than in France, since the p-value is not significant. For Japan and Korea the p-value is much more significant and less than the alpha value of 5%, and also has a negative estimate, which means that there is a higher risk of dying of corona in Japan and Korea than in France.

iv)

A person is 10 years older than another person. The probability of dying is linear in terms of age because of the logic regression, so we can see the probability of a person dying at an age of 65 and an age of 75 from task i and see that there is an increase of risk to die in case of higher age.

```
deceasedmale75 <- fit$coefficients[6] * 75 + fit$coefficients[2] + fit$coefficients[5]
deceasedmale65 <- fit$coefficients[6] * 65 + fit$coefficients[2] + fit$coefficients[5]

diff <- (deceasedmale75 - deceasedmale65) * 100
```

This gives an age difference in NA%

c)

i)

```
fit <- glm(deceased ~ age * sex, data = d.corona, family = "binomial")
summary(fit)$coef
```

```
##                  Estimate  Std. Error     z value     Pr(>|z|)
## (Intercept) -4.7759607615 0.484626626 -9.85492852 6.526311e-23
## age          0.0278572418 0.007278416  3.82737708 1.295160e-04
## sexmale      0.5588617879 0.628776551  0.88880825 3.741061e-01
## age:sexmale  0.0005070244 0.009476422  0.05350378 9.573305e-01
```

Here we see that the `age:sexmale` coefficients has a positive estimate, but doesn't have a significant p-value. Age has a slightly lower p-value, and we can see that age is not a greater risk factor for males than for females.

ii)

We fitted the function to find the relation with country and age.

```
fit <- glm(deceased ~ age * country, data = d.corona, family = "binomial")
summary(fit)$coef
```

```
##                          Estimate Std. Error   z value     Pr(>|z|)
## (Intercept)          -6.65232422 1.68621990 -3.945111 7.976311e-05
## age                   0.06637008 0.02081162  3.189088 1.427225e-03
## countryindonesia      4.27185119 2.11130583  2.023322 4.303998e-02
## countryjapan          2.05835682 2.00125741  1.028532 3.036998e-01
## countryKorea          2.33159626 1.72031415  1.355332 1.753119e-01
## age:countryindonesia -0.06981659 0.03206143 -2.177588 2.943672e-02
## age:countryjapan     -0.04403013 0.02627239 -1.675909 9.375603e-02
## age:countryKorea     -0.04173146 0.02148983 -1.941917 5.214712e-02
```

We found that the coefficient for the `age:countryindonesia` interaction is negative, which means that Indonesia is lower than it is for France. The p-value is slightly significant, which gives a low but greater risk factor for the Indonesian population than for the French.

d)

First we fitted the dataset with all the covariates

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

```
##                     Estimate  Std. Error    z value     Pr(>|z|)
## (Intercept)      -3.99348478 0.462190319 -8.6403471 5.604250e-18
## sexmale           0.62606777 0.209044668  2.9948995 2.745353e-03
## age               0.02713421 0.004736262  5.7290352 1.010034e-08
## countryindonesia -0.41185539 0.550050523 -0.7487592 4.540024e-01
## countryjapan     -1.34338289 0.417195836 -3.2200295 1.281774e-03
## countryKorea     -0.77389515 0.307979819 -2.5128112 1.197734e-02
```

Then we used the three predictor variables age, sex and country for LDA to print the confusion table for LDA

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

```
##        true
## predict    0    1
##       0 1905  105
##       1    0    0
```

and did the same for the counfusion matrix for QDA

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

```
##        true
## predict    0    1
##       0 1751   84
##       1  154   21
```

The answers will then be FALSE, TRUE, TRUE, FALSE

Problem 3

a)

```
# read file
id <- "1i1cQPeoLLC_FyAHOnnqCnnrSBpn05_hO"  # 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
```

```
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)

prove that logit is linear By denoting the term $\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_7 x_{i7}$ as β_{sum}, we can derive that

$$\text{logit}(p_i) = \log(\frac{p_i}{1-p_i}) = \log(\frac{\frac{e^{\beta_{sum}}}{1+e^{\beta_{sum}}}}{1-\frac{e^{\beta_{sum}}}{1+e^{\beta_{sum}}}}) \tag{13}$$

$$= \log(\frac{\frac{e^{\beta_{sum}}}{1+e^{\beta_{sum}}}}{\frac{1+e^{\beta_{sum}}}{1+e^{\beta_{sum}}} - \frac{e^{\beta_{sum}}}{1+e^{\beta_{sum}}}}) \tag{14}$$

$$= \log(\frac{\frac{e^{\beta_{sum}}}{1+e^{\beta_{sum}}}}{\frac{1+e^{\beta_{sum}}-e^{\beta_{sum}}}{1+e^{\beta_{sum}}}}) \tag{15}$$

$$= \log(\frac{e^{\beta_{sum}}}{1+e^{\beta_{sum}}-e^{\beta_{sum}}}) \qquad = \beta_{sum} \tag{16}$$

As we know,

$$\beta_{sum} = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_7 x_{i7}$$

is linear for the covariates x_n

ii)

```r
# make predictions on test set
pred <- predict(logReg, newdata = test, type = "response")

# convert predictions to numeric values (0/1)
numpred <- as.numeric(pred > 0.5)

# create reference on test set
reference <- test$diabetes

# ensure correct levels
u <- union(numpred, reference)
t <- table(factor(numpred, u), factor(reference, u))

# make confusion matrix (and sensitivity/specificity etc.)
confusionMatrix(t)$table
```

```
##
##        1   0
##   1   48  18
##   0   29 137
```

```r
confusionMatrix(t)$byClass
```

```
##          Sensitivity          Specificity       Pos Pred Value
##            0.6233766            0.8838710            0.7272727
##       Neg Pred Value            Precision               Recall
##            0.8253012            0.7272727            0.6233766
##                   F1           Prevalence       Detection Rate
##            0.6713287            0.3318966            0.2068966
## Detection Prevalence    Balanced Accuracy
##            0.2844828            0.7536238
```

b)

(i) π_k is the prior probability that a random observation comes from the $k-th$ class. As we have two classes, where class 0 have 200 observations and class 1 have 100 observations from the training set, we obtain that $\pi_0 \approx 0.67$ and $\pi_1 \approx 0.33$

μ_k is the mean of X. It is a vector of size p, where p is the number of predictors, and it can be estimated to

$$\hat{\mu_k} = \frac{1}{n_k} \sum_{i:y_i=k} x_i$$

$\mathbf{\Sigma}$ is the $p \times p$-covariance matrix of X which is common for all classes.

$f_k(x)$ is the multivariate Gaussian density, meaning the ...

ii)

```
# table(predict = predict(lda(diabetes ~ glu + bmi + ped, data = train))$class,
# true = train$diabetes)


lda.fit = lda(diabetes ~ ., data = train)
lda.pred = predict(lda.fit, test, type = "prob")
lda.table = table(predict = lda.pred$class, true = test$diabetes)
lda.table
```

```
##         true
## predict   0   1
##       0 138  30
##       1  17  47
```

```
qda.fit = qda(diabetes ~ ., data = train)
qda.pred = predict(qda.fit, test)
table(predict = qda.pred$class, true = test$diabetes)
```

```
##         true
## predict   0   1
##       0 131  32
##       1  24  45
```

c)

i)

A new observation x_0 is classified to the most occuring class of the K nearest nodes in the training data by Eucleidian distance. For K=1 x_0 is simply classified as the same class as the nearest node.

ii)

When choosing the tuning parameter k, one must consider the bias-variance trade off (small k leads to large bias, but low varaiance and vice verca). For this case we need a loss function and a validation set, which we will come back to. First, we split our data in to a test set, and a training set. Then, by applying cross validation to the training set, splitting up in training and validation sets, we are able to use the whole training set and still ensure a valid test set. The model is then fitted on the cross validated sets for different K, and a loss is computed for the validation set. The tuning parameter k is determined by evaluating for which K the valiation error is lowest.

an other approach is to apply bootstrapping on the training set.

(iii)

```
# traincl <- factor(diab[train, 'classifications'])
knnMod = knn(train = train, test = test, cl = train$diabetes, k = 25, prob = T)
knnConfMat = table(knnMod, test$diabetes)

knnConfMat
```

```
##
## knnMod   0   1
##      0 144  36
##      1  11  41
```

```
sensitivity = knnConfMat[2, 2]/(knnConfMat[2, 2] + knnConfMat[1, 2])
sensitivity
```

```
## [1] 0.5324675
```

```
specificity = knnConfMat[1, 1]/(knnConfMat[1, 1] + knnConfMat[2, 1])
specificity
```

```
## [1] 0.9290323
```

Problem 4

Problem 5

Loading the bodyfat dataset given in the problem:

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

a)

```
r.bodyfat <- lm(bodyfat ~ ., data = d.bodyfat)

rsqrd <- summary(r.bodyfat)$r.squared

rsqrd
```

```
## [1] 0.7281213
```

The R^2 is found to be 0.728.

b)

i)

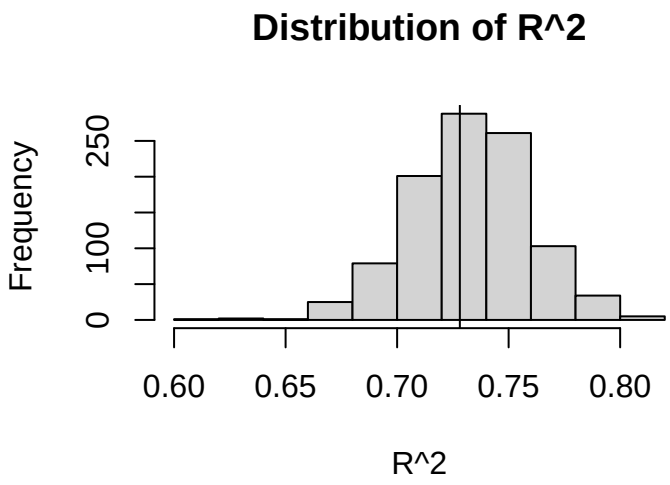
Generate 1000 bootstrap samples of the R^2

```
N = length(d.bodyfat[, 1])  #Finding length of bodyfat


set.seed(4268)
# boot(data = r.bodyfat, statistic = , R = 1000)
index <- sample(1:N, N, replace = TRUE)

newdata.d <- d.bodyfat[index, ]

B = 1000

r2stored <- rep(0, B)  #Generating a list with B zeros

for (i in 1:B) {
    index <- sample(1:N, N, replace = TRUE)  #Generating randon integers from 1, length of dataset
    newbodyfat.d <- d.bodyfat[index, ]  #creating a new dataset with the generated indexes
    bodyfat.boot <- lm(bodyfat ~ ., data = newbodyfat.d)  #fitting the regression model
    r2stored[i] = summary(bodyfat.boot)$r.squared  #appending r squared to the r2stored list.

}
```

ii)

Plotting the distribution of the values of R^2

```
hist(r2stored, main = "Distribution of R^2", xlab = "R^2", ylab = "Frequency")
abline(v = rsqrd)
```



iii)

```
index <- sample(1:N, N, replace = TRUE)

mean.r2 = mean(r2stored)
SE = sqrt(1/(B - 1) * sum((r2stored - mean.r2)^2))
confInverval = quantile(r2stored, probs = c(5, 95)/100)

SE
```

```
## [1] 0.02662221
```

```
confInverval
```

```
##        5%       95%
## 0.6887567 0.7769587
```

iv)

The R^2 value found in problem 5a)