Compulsory exercise 1: Group 5

TMA4268 Statistical Learning V2021

Hans Røhjell Odland and Aksel Haugen Madslien

2/9/2021

For some problems you will need to include some LaTex code. Please install latex on your computer and then consult Compulsory1.Rmd for hints how to write formulas in LaTex.

An example:

$$Y_i = f(x_i) + \varepsilon_i \ ,$$

Or the same formula

$$Y_i = f(x_i) + \varepsilon_i$$

in-line.

Problem 1

a)

We consider

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

We find the expected value for

$$\tilde{\beta}$$

as

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

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

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

For the variation we get

$$Var[\widetilde{f}(\mathbf{x}_0)] = \mathbf{x}_0^T Var[\widetilde{\boldsymbol{\beta}}]\mathbf{x} \tag{8}$$
$$= \mathbf{x}_0^T (\beta - \lambda(\mathbf{x}^T\mathbf{x} + \lambda\mathbf{I})\beta)\mathbf{x} \tag{9}$$

c)

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
```

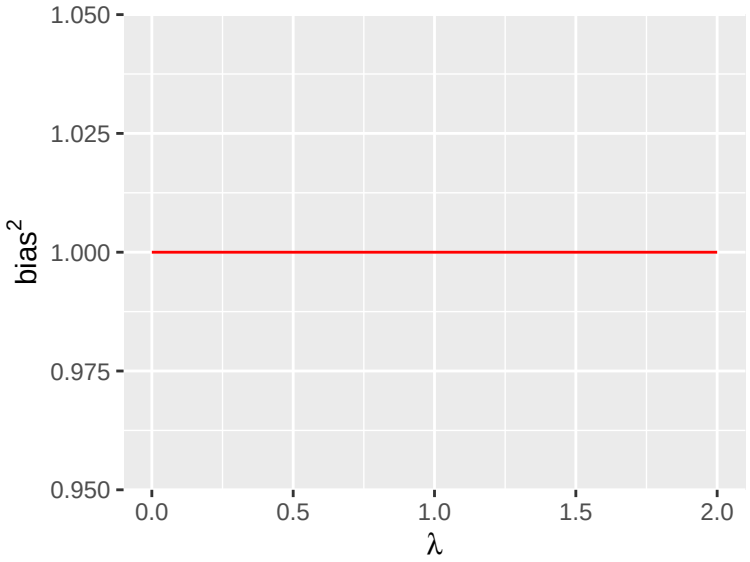
```
beta = values$beta
dim(beta)
```

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

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

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

```
bias = function(lambda, X, x0, beta) {
    p = ncol(X)
    value = 1
    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)


table(d.corona$deceased)
```

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

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

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

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

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

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

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

b)

i)

```r
# Multiple Linear Regression Example
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
# predict(fit)

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

deceasedmale
```

```
##        age
## 0.05669394
```

The probability of dying of Covid-19 for a male age in Korea is found to be 5.7%.

ii)

Does 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, which means that we can conclude that men have a higher probability of dying of corona than women.

###iii)

```r
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 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.

```r
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 1.3% and we see that the model is linear.

c)

i)

```r
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)

```r
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
```

No, 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)

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

```
##
## Call:
## glm(formula = deceased ~ ., family = "binomial", data = d.corona)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.9050  -0.3508  -0.2761  -0.2144   3.1165
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -3.993485   0.462190  -8.640  < 2e-16 ***
## sexmale           0.626068   0.209045   2.995  0.00275 **
## age               0.027134   0.004736   5.729 1.01e-08 ***
## countryindonesia -0.411855   0.550051  -0.749  0.45400
## countryjapan     -1.343383   0.417196  -3.220  0.00128 **
## countryKorea     -0.773895   0.307980  -2.513  0.01198 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 824.32  on 2009  degrees of freedom
## Residual deviance: 766.16  on 2004  degrees of freedom
## AIC: 778.16
##
## Number of Fisher Scoring iterations: 6
```

```
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
```

```
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
```

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

$$\text{logit}(p_i) = \log(\frac{p_i}{1-p_i}) = \log(\frac{\frac{e^{\beta_{sum}}}{1+e^{\beta_{sum}}}}{1-\frac{e^{\beta_{sum}}}{1+e^{\beta_{sum}}}}) \tag{10}$$

$$= \log(\frac{\frac{e^{\beta_{sum}}}{1+e^{\beta_{sum}}}}{\frac{1+e^{\beta_{sum}}}{1+e^{\beta_{sum}}} - \frac{e^{\beta_{sum}}}{1+e^{\beta_{sum}}}}) \tag{11}$$

$$= \log(\frac{\frac{e^{\beta_{sum}}}{1+e^{\beta_{sum}}}}{\frac{1+e^{\beta_{sum}}-e^{\beta_{sum}}}{1+e^{\beta_{sum}}}}) \tag{12}$$

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

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
# install package e1071 (required to ensure correct levels in given manner)


# 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)
```

```
## Confusion Matrix and Statistics
##
##
##       1   0
##   1  48  18
##   0  29 137
##
##                Accuracy : 0.7974
##                  95% CI : (0.7399, 0.8472)
##     No Information Rate : 0.6681
##     P-Value [Acc > NIR] : 9.374e-06
##
##                   Kappa : 0.5262
##
##  Mcnemar's Test P-Value : 0.1447
##
##             Sensitivity : 0.6234
##             Specificity : 0.8839
##          Pos Pred Value : 0.7273
##          Neg Pred Value : 0.8253
##              Prevalence : 0.3319
##          Detection Rate : 0.2069
##    Detection Prevalence : 0.2845
##       Balanced Accuracy : 0.7536
##
##        'Positive' Class : 1
##
```

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)
qda.table = table(predict = qda.pred$class, true = test$diabetes)
qda.table
```

```
##         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])


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

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

```
B = 1000

for (i in i:B) {

}
```