

# Project 10

## Contents

<b>Problem 27</b>	<b>1</b>
<b>Problem 28 (a)</b>	<b>2</b>
<b>Problem 28 (b)</b>	<b>3</b>
<b>Problem 29</b>	<b>3</b>
Splitting data into training and testing subsets . . . . .	3
Cross-validation of elastic-net model . . . . .	4
Fitting model on the whole training set . . . . .	6
Testing model . . . . .	6

## Problem 27

Showing that  $X\hat{\beta}^{(1)} = X\hat{\beta}^{(2)}$  ( $\hat{\beta}^{(1)} = \hat{\beta}^{(2)}$ ) give the same Lasso predictions.

### Proof

The statement will be proven by contradiction. Let's assume that  $\hat{\beta}^{(1)} \neq \hat{\beta}^{(2)}$ .

Let's define  $f(u) = \|y - u\|_2^2$ ,  $l_1 = \|u\|_1$ , and  $g(u) = \frac{1}{2}f(u) + \lambda l_1(u)$ .

Let's say that  $u = \alpha\hat{\beta}^{(1)} + (1 - \alpha)\hat{\beta}^{(2)}$ , which is in the Lasso solution set for  $\forall \alpha \in (0, 1)$ .

$$g(\alpha\hat{\beta}^{(1)} + (1 - \alpha)\hat{\beta}^{(2)}) = \frac{1}{2}f(\alpha\hat{\beta}^{(1)} + (1 - \alpha)\hat{\beta}^{(2)}) + \lambda l_1(\alpha\hat{\beta}^{(1)} + (1 - \alpha)\hat{\beta}^{(2)}) \stackrel{\text{convexity of } l_1}{\leq} \quad (1)$$

$$\leq \frac{1}{2}f(\alpha\hat{\beta}^{(1)} + (1 - \alpha)\hat{\beta}^{(2)}) + \lambda \alpha l_1(\hat{\beta}^{(1)}) + \lambda(1 - \alpha)l_1(\hat{\beta}^{(2)}) \stackrel{\text{strict convexity of } f}{<} \quad (2)$$

$$< \frac{1}{2}\alpha f(\hat{\beta}^{(1)}) + \frac{1}{2}(1 - \alpha)f(\hat{\beta}^{(2)}) + \lambda \alpha l_1(\hat{\beta}^{(1)}) + \lambda(1 - \alpha)l_1(\hat{\beta}^{(2)}) = \quad (3)$$

The following lines display rearrangement of members.

$$= \frac{1}{2}\alpha f(\hat{\beta}^{(1)}) + \lambda \alpha l_1(\hat{\beta}^{(1)}) + \frac{1}{2}(1 - \alpha)f(\hat{\beta}^{(2)}) + \lambda(1 - \alpha)l_1(\hat{\beta}^{(2)}) = \quad (4)$$

$$= \alpha \left[ \frac{1}{2}f(\hat{\beta}^{(1)}) + \lambda l_1(\hat{\beta}^{(1)}) \right] + (1 - \alpha) \left[ \frac{1}{2}f(\hat{\beta}^{(2)}) + \lambda l_1(\hat{\beta}^{(2)}) \right] = \quad (5)$$

$$= \alpha c^* + (1 - \alpha)c^* = \alpha c^* + c^* - \alpha c^* = c^* \quad (6)$$

$$\Rightarrow g(u) = g(\alpha\hat{\beta}^{(1)} + (1 - \alpha)\hat{\beta}^{(2)}) < c^* \quad (7)$$

It implies that  $u$  does not belong to the solution set of Lasso, which imposes contradiction. Therefore our initial assumption that  $\hat{\beta}^{(1)} \neq \hat{\beta}^{(2)}$  is incorrect.  $\square$

## Problem 28 (a)

Show that for some  $\lambda$ , the Ridge regression coefficients are equivalent to the maximum a posteriori (MAP) estimator, if we assume a normal prior for the coefficients.

### Proof

Solution of Ridge regression can be written as (with  $\lambda \geq 0$ ):

$$\beta_{Ridge} = \underset{\beta}{\operatorname{argmin}} \left( \|y - X\beta\|^2 + \lambda \|\beta\|^2 \right) \quad (8)$$

Posterior distribution of  $\beta$  is proportional to product of likelihood and the prior:

$$p(\beta|X, y) \propto p(y|X, \beta) \cdot p(\beta) \quad (9)$$

Since  $y = X\beta + \epsilon$ , where  $\epsilon \sim N(0, \sigma^2 I_n)$ ,  $y \sim N(X\beta, \sigma^2 I_n)$ . Therefore  $p(y|X, \beta)$  follows  $N(X\beta, \sigma^2 I_n)$  and  $p(\beta)$  is a given prior.

$$\begin{aligned} \beta_{MAP} &= \underset{\beta}{\operatorname{argmax}} \left( p(y|X, \beta) \cdot p(\beta) \right) = \underset{\beta}{\operatorname{argmin}} \left( -\ln(p(y|X, \beta) \cdot p(\beta)) \right) = \\ &= \underset{\beta}{\operatorname{argmin}} \left( -\ln(p(y|X, \beta)) - \ln(p(\beta)) \right) \end{aligned} \quad (10)$$

$$\begin{aligned} \ln(p(y|X, \beta)) &= \ln \left( \prod_{i=1}^n \frac{1}{\sqrt{2\pi}\sigma} \exp \left( -\frac{1}{2} \frac{(y_i - X\beta_i)^2}{\sigma^2} \right) \right) = \ln \left( \frac{n}{\sqrt{2\pi}\sigma} \right) + \sum_{i=1}^n \ln \left( \exp \left( -\frac{1}{2} \frac{(y_i - X\beta_i)^2}{\sigma^2} \right) \right) = \\ &= \ln \left( \frac{n}{\sqrt{2\pi}\sigma} \right) + \sum_{i=1}^n \left( -\frac{1}{2} \frac{(y_i - X\beta_i)^2}{\sigma^2} \right) \end{aligned} \quad (11)$$

$$\begin{aligned} \ln(p(\beta)) &= \ln \left( \prod_{k=1}^p \frac{1}{\sqrt{2\pi}\sigma_\beta} \exp \left( -\frac{1}{2} \frac{\beta_k^2}{\sigma_\beta^2} \right) \right) = \ln \left( \frac{p}{\sqrt{2\pi}\sigma_\beta} \right) + \sum_{k=1}^p \ln \left( \exp \left( -\frac{1}{2} \frac{\beta_k^2}{\sigma_\beta^2} \right) \right) = \\ &= \ln \left( \frac{p}{\sqrt{2\pi}\sigma_\beta} \right) + \sum_{k=1}^p \left( -\frac{1}{2} \frac{\beta_k^2}{\sigma_\beta^2} \right) \end{aligned} \quad (12)$$

By collecting members that depend on  $\beta$ , we get  $\beta_{MAP}$  expression:

$$\begin{aligned} \beta_{MAP} &= \underset{\beta}{\operatorname{argmin}} \left( \frac{1}{2\sigma^2} \|y - X\beta\|^2 + \frac{1}{2\sigma_\beta^2} \|\beta\|^2 \right) \stackrel{!}{=} \underset{\beta}{\operatorname{argmin}} \left( \|y - X\beta\|^2 + \frac{\sigma^2}{\sigma_\beta^2} \|\beta\|^2 \right) \\ &\Rightarrow \lambda = \frac{\sigma^2}{\sigma_\beta^2} \end{aligned} \quad (13)$$

For  $\lambda = \frac{\sigma^2}{\sigma_\beta^2}$  Ridge regression coefficients are equivalent to the MAP estimator, if normal prior for coefficients is assumed.

## Problem 28 (b)

Show that for some  $\lambda$ , the Lasso regression coefficients are equivalent to the maximum a posteriori (MAP) estimator, if we assume prior  $\pi(\beta) = \prod_{k=1}^p \frac{1}{2b} \exp(-\frac{|\beta_k|}{b})$ .

### Proof

Solution of Lasso regression can be written as (with  $\lambda \geq 0$ ):

$$\beta_{Lasso} = \underset{\beta}{\operatorname{argmin}} \left( \|y - X\beta\|^2 + \lambda \|\beta\|_1 \right) \quad (14)$$

Posterior distribution of  $\beta$  is proportional to product of likelihood and the prior and since  $y$ ,  $X$ ,  $\beta$ , and  $\epsilon$  stay the same as in part *a*, we can recycle the computations of log-likelihood and take a look only at the part of the prior (having  $\pi(\beta) = p(\beta)$ ).

$$\begin{aligned} \ln(p(\beta)) &= \ln\left(\prod_{k=1}^p \frac{1}{2b} \exp(-\frac{|\beta_k|}{b})\right) = \ln\left(\frac{p}{2b}\right) + \sum_{k=1}^p \ln\left(\exp(-\frac{|\beta_k|}{b})\right) = \\ &= \ln\left(\frac{p}{2b}\right) + \sum_{k=1}^p \left(-\frac{|\beta_k|}{b}\right) \end{aligned} \quad (15)$$

By collecting members that depend on  $\beta$ , we get  $\beta_{MAP}$  expression:

$$\begin{aligned} \beta_{MAP} &= \underset{\beta}{\operatorname{argmin}} \left( \frac{1}{2\sigma^2} \|y - X\beta\|^2 + \frac{1}{b} \|\beta\|_1 \right) \stackrel{|\cdot 2\sigma^2}{=} \underset{\beta}{\operatorname{argmin}} \left( \|y - X\beta\|^2 + \frac{2\sigma^2}{b} \|\beta\|_1 \right) \\ &\Rightarrow \lambda = \frac{2\sigma^2}{b} \end{aligned} \quad (16)$$

For  $\lambda = \frac{2\sigma^2}{b}$  Lasso regression coefficients are equivalent to the MAP estimator, if the given prior  $\pi(\beta)$  is assumed.

## Problem 29

```
library(caret)
library(glmnet)
library(pROC)

load(file='yeastStorey.rda')

print(paste("Number of samples (N):", nrow(data)))

## [1] "Number of samples (N): 112"
print(paste("Number of features (p):", ncol(data)))

## [1] "Number of features (p): 232"
```

### Splitting data into training and testing subsets

```
set.seed(42)
trainIndex <- createDataPartition(data$Marker, p=0.7, list=FALSE, times=1)
trainData <- data[trainIndex,]
testData <- data[-trainIndex,]
```

## Cross-validation of elastic-net model

```
# Preparing data for cv.glmnet
x <- trainData[, !(names(trainData) %in% c("Marker"))]
x <- as.matrix(x)
y <- trainData$Marker

# Executing 10-fold CV for each value of alpha
foldid <- sample(1:10, size=length(y), replace=TRUE)
alphas <- seq(0, 1, by=0.1)

elasticNetCVAAlpha <- function(alpha) {
  cv.glmnet(x, y, family="binomial", alpha=alpha, nfolds=10, foldid=foldid)
}

resultsCV <- lapply(alphas, elasticNetCVAAlpha)

# Finding the optimal alpha
minMeanCVMIdx <- 1
minMeanCVM <- mean(resultsCV[[1]]$cvm)
for(i in 1:length(alphas)) {
  if(minMeanCVM > mean(resultsCV[[i]]$cvm)) {
    minMeanCVM <- mean(resultsCV[[i]]$cvm)
    minMeanCVMIdx <- i
  }
  # Reporting mean of mean cross-validated error of each alpha
  print(paste0("alpha=", alphas[i], "; error=", mean(resultsCV[[i]]$cvm)))
}
```

```
## [1] "alpha=0; error=1.41826065852264"
## [1] "alpha=0.1; error=1.207662476829"
## [1] "alpha=0.2; error=1.07852495274017"
## [1] "alpha=0.3; error=0.978145211748884"
## [1] "alpha=0.4; error=0.892525261344348"
## [1] "alpha=0.5; error=0.81568741249565"
## [1] "alpha=0.6; error=0.745849929007219"
## [1] "alpha=0.7; error=0.678769142955298"
## [1] "alpha=0.8; error=0.60340112911262"
## [1] "alpha=0.9; error=0.519596711095501"
## [1] "alpha=1; error=0.430489062157447"
```

## Finding optimal alpha

$\alpha$  with which mean of mean cross-validated error is the smallest:  $\alpha = 1$ . This  $\alpha$  will be considered as optimal.

```
print(paste("Min. mean of mean cross-validated error:", minMeanCVM))
```

```
## [1] "Min. mean of mean cross-validated error: 0.430489062157447"
```

```

optimalAlphaIdx <- minMeanCVMIdx
optimalAlpha <- alphas[optimalAlphaIdx]

```

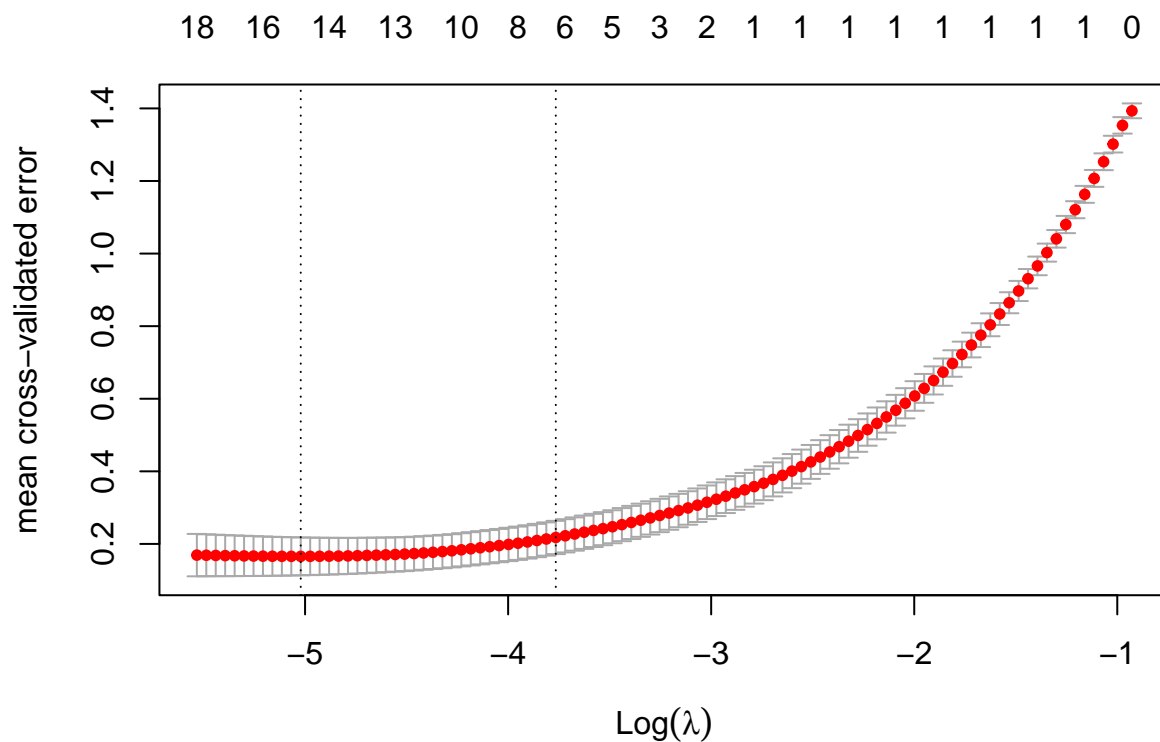
### Plotting mean cross-validated error

Cross-validated error function is binomial deviance. The plot of  $\log(\lambda)$  versus mean cross-validated error is done using results retrieved with  $\alpha = 1$ .

```

plot(resultsCV[optimalAlphaIdx], ylab="mean cross-validated error")

```



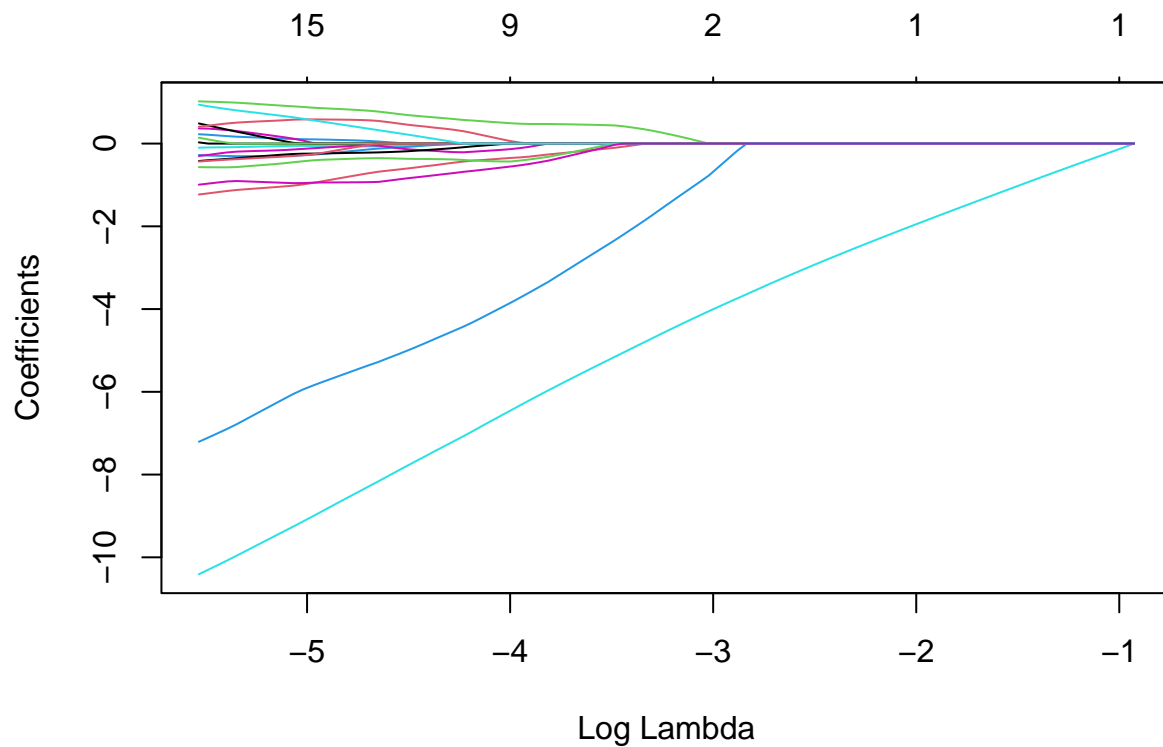
### Plotting trace curve of coefficients

The plot of  $\log(\lambda)$  versus coefficients is done using results retrieved with  $\alpha = 1$ .

```

plot(resultsCV[optimalAlphaIdx]$glmnet.fit, "lambda")

```



### Picking optimal lambda

```
optimalLambdaIdx <- which.min(resultsCV[[optimalAlphaIdx]]$cvm)
optimalLambda <- resultsCV[[optimalAlphaIdx]]$lambda[[optimalLambdaIdx]]
```

Optimal  $\lambda = 0.0065961$ .

### Fitting model on the whole training set

```
trainedModel <- glmnet(x, y, family="binomial", alpha=optimalAlpha, lambda=optimalLambda)
trainedModel
```

```
##
## Call:  glmnet(x = x, y = y, family = "binomial", alpha = optimalAlpha,      lambda = optimalLambda)
##
##   Df %Dev  Lambda
##  1 15 96.72 0.006596
```

### Testing model

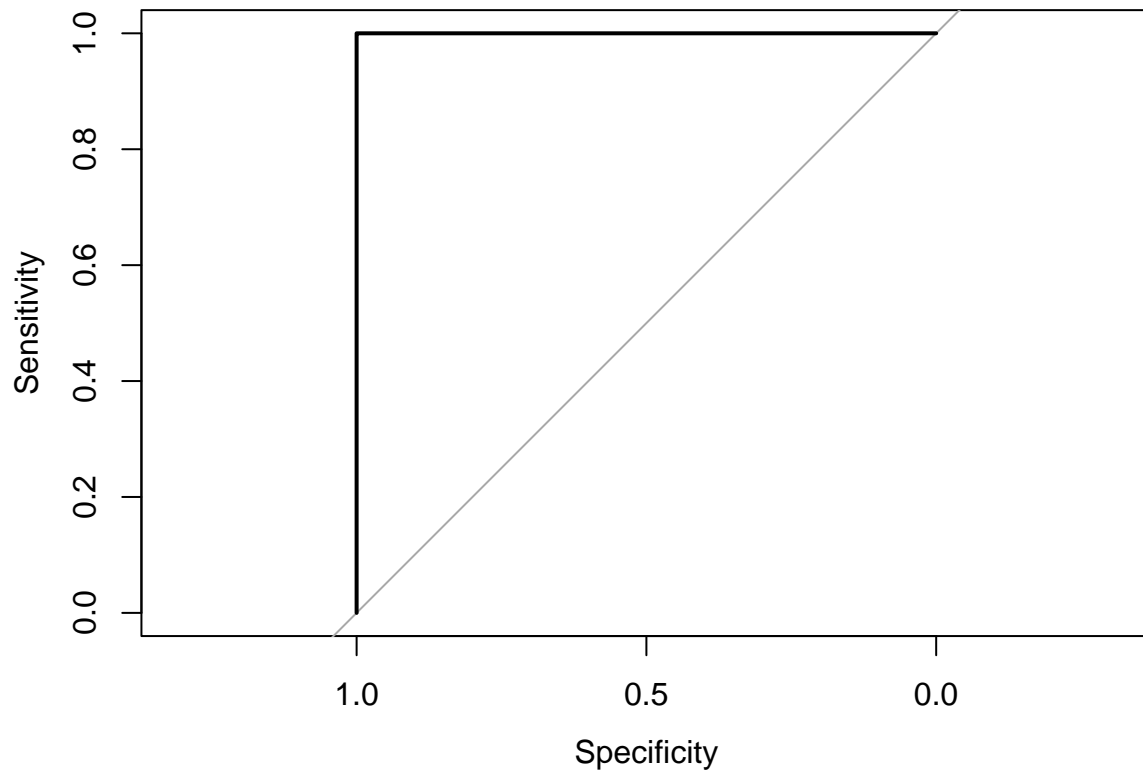
```
# Preparing data for inference using fitted glmnet
xTest <- testData[, !(names(testData) %in% c("Marker"))]
xTest <- as.matrix(xTest)
yTest <- testData$Marker
```

```

# Making predictions and evaluating performance
predictions <- predict(trainedModel, newx=xTest)
resultsTest <- assess.glmnet(predictions, newy=yTest, family="binomial")

roc(yTest, predictions, plot=TRUE)

```



```

##
## Call:
## roc.default(response = yTest, predictor = predictions, plot = TRUE)
##
## Data: predictions in 17 controls (yTest 0) < 16 cases (yTest 1).
## Area under the curve: 1

library(rmarkdown)
render("project10.Rmd", pdf_document(TRUE), "Indilewitsch_Toidze_Houhamdi_Pudziuvelyte_Project10.pdf")

```