Project 10

Contents

Problem 27

Showing that $X\hat{\beta}^{(1)} = X\hat{\beta}^{(2)}$, $\hat{\beta}^{(1)}$ and $\hat{\beta}^{(2)}$) 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 $f(u) = ||y - u||_2^2$, $l_1(u) = ||u||_1$, 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 $\forall\alpha \in (0, 1)$.

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

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

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

The following lines display rearrangement of members.

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

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

$$= \alpha c^* + (1-\alpha)c^* = \alpha c^* + c^* - \alpha c^* = c^* \tag{6}$$

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

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

Problem 28 (a)

Show that for some λ, 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 $\lambda \geq 0$):

$$\beta_{Ridge} = \underset{\beta}{argmin}\Big(||y - X\beta||^2 + \lambda||\beta||^2\Big) \tag{8}$$

Posterior distribution of β is proportional to product of likelihood and the prior:

$$p(\beta|X, y) \propto p(y|X, \beta) \cdot p(\beta) \tag{9}$$

Since $y = X\beta + \epsilon$, where $\epsilon \sim N(0, \sigma^2 I_n)$. Therefore $p(y|X, \beta)$ follows $N(X\beta, \sigma^2 I_n)$ and $p(\beta) = \prod_{k=1}^{p} \frac{1}{\sqrt{2\pi}\sigma_\beta} exp(-\frac{1}{2}\frac{\beta_k^2}{\sigma_\beta^2})$.

$$\beta_{MAP} = \underset{\beta}{argmax}\Big(p(y|X, \beta) \cdot p(\beta)\Big) = \underset{\beta}{argmin}\Big(-ln(p(y|X, \beta) \cdot p(\beta))\Big) =$$

$$= \underset{\beta}{argmin}\Big(-ln(p(y|X, \beta)) - ln(p(\beta))\Big) \tag{10}$$

$$ln(p(y|X, \beta)) = ln\Big(\prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma} exp(-\frac{1}{2}\frac{(y_i - X\beta_i)^2}{\sigma^2})\Big) = ln\Big(\frac{1}{\sqrt{2\pi}\sigma}\Big)^n + \sum_{i=1}^{n} ln\Big(exp(-\frac{1}{2}\frac{(y_i - X\beta_i)^2}{\sigma^2})\Big) =$$

$$= ln\Big(\frac{1}{\sqrt{2\pi}\sigma}\Big)^n + \sum_{i=1}^{n}\Big(-\frac{1}{2}\frac{(y_i - X\beta_i)^2}{\sigma^2}\Big) \tag{11}$$

$$ln(p(\beta)) = ln\Big(\prod_{k=1}^{p} \frac{1}{\sqrt{2\pi}\sigma_\beta} exp(-\frac{1}{2}\frac{\beta_k^2}{\sigma_\beta^2})\Big) = ln\Big(\frac{1}{\sqrt{2\pi}\sigma_\beta}\Big)^p + \sum_{k=1}^{p} ln\Big(exp(-\frac{1}{2}\frac{\beta_k^2}{\sigma_\beta^2})\Big) =$$

$$= ln\Big(\frac{1}{\sqrt{2\pi}\sigma_\beta}\Big)^p + \sum_{k=1}^{p}\Big(-\frac{1}{2}\frac{\beta_k^2}{\sigma_\beta^2}\Big) \tag{12}$$

By collecting members that depend on β, we get β_{MAP} expression:

$$\beta_{MAP} = \underset{\beta}{argmin}\Big(\frac{1}{2\sigma^2}||y - X\beta||^2 + \frac{1}{2\sigma_\beta^2}||\beta||^2\Big) \overset{|\cdot 2\sigma^2}{=} \underset{\beta}{argmin}\Big(||y - X\beta||^2 + \frac{\sigma^2}{\sigma_\beta^2}||\beta||^2\Big) \tag{13}$$

$$\Rightarrow \lambda = \frac{\sigma^2}{\sigma_\beta^2}$$

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 λ, 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 $\lambda \geq 0$):

$$\beta_{Lasso} = \underset{\beta}{argmin}\Big(||y - X\beta||^2 + \lambda||\beta||_1\Big) \tag{14}$$

Posterior distribution of β is proportional to product of likelihood and the prior and, since y, X, β, and ϵ 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 $\pi(\beta) = p(\beta)$).

$$ln(p(\beta)) = ln\Big(\prod_{k=1}^{p} \frac{1}{2b} exp(-\frac{|\beta_k|}{b})\Big) = ln\Big(\frac{1}{2b}\Big)^p + \sum_{k=1}^{p} ln\Big(exp(-\frac{|\beta_k|}{b})\Big) =$$

$$= ln\Big(\frac{1}{2b}\Big)^p + \sum_{k=1}^{p}\Big(-\frac{|\beta_k|}{b}\Big) \tag{15}$$

By collecting members that depend on β, we get β_{MAP} expression:

$$\beta_{MAP} = \underset{\beta}{argmin}\Big(\frac{1}{2\sigma^2}||y - X\beta||^2 + \frac{1}{b}||\beta||_1\Big) \overset{|\cdot 2\sigma^2}{=} \underset{\beta}{argmin}\Big(||y - X\beta||^2 + \frac{2\sigma^2}{b}||\beta||_1\Big) \tag{16}$$

$$\Rightarrow \lambda = \frac{2\sigma^2}{b}$$

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

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

```r
load(file='yeastStorey.rda')
```

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

```
## [1] "Number of samples (N): 112"
```

```r
print(paste("Number of features (p):", ncol(data)))
```

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

Splitting data into training and testing subsets

```r
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

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

```r
# 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)

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

resultsCV <- lapply(alphas, elasticNetCVAlpha)
```

```r
# 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

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

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

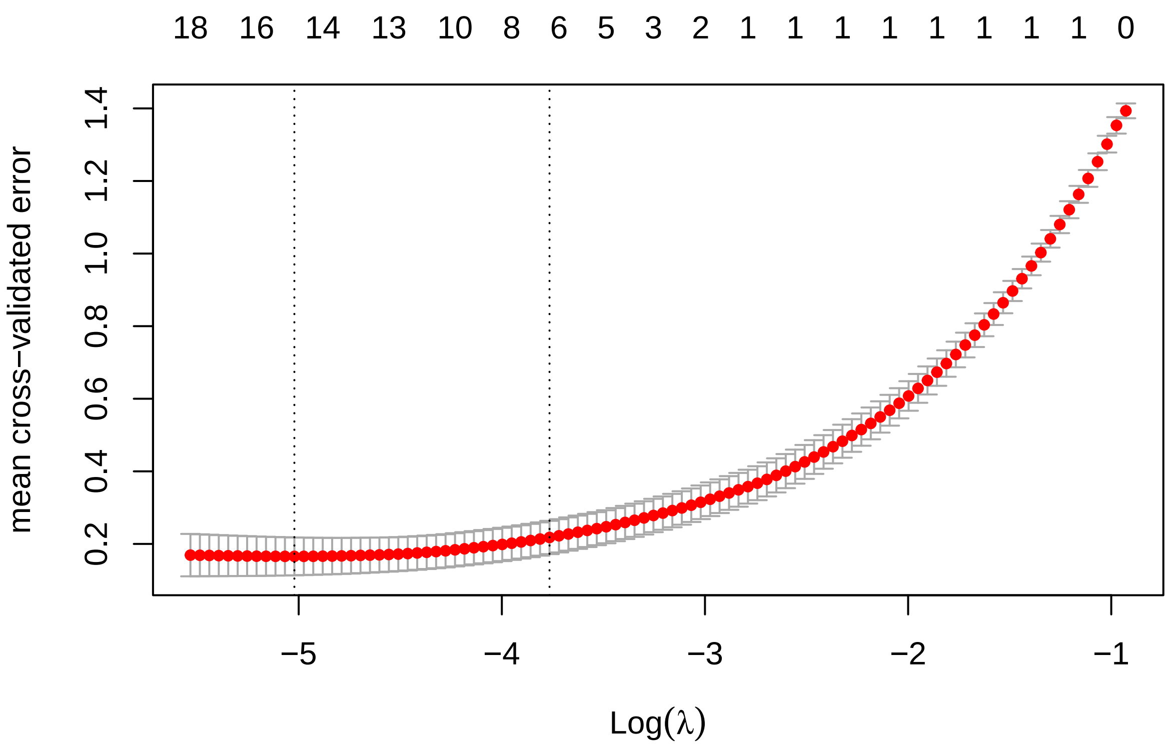
```
## [1] "Min. mean of mean cross-validated error: 0.430489062157447"
```

```r
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 $\alpha = 1$.

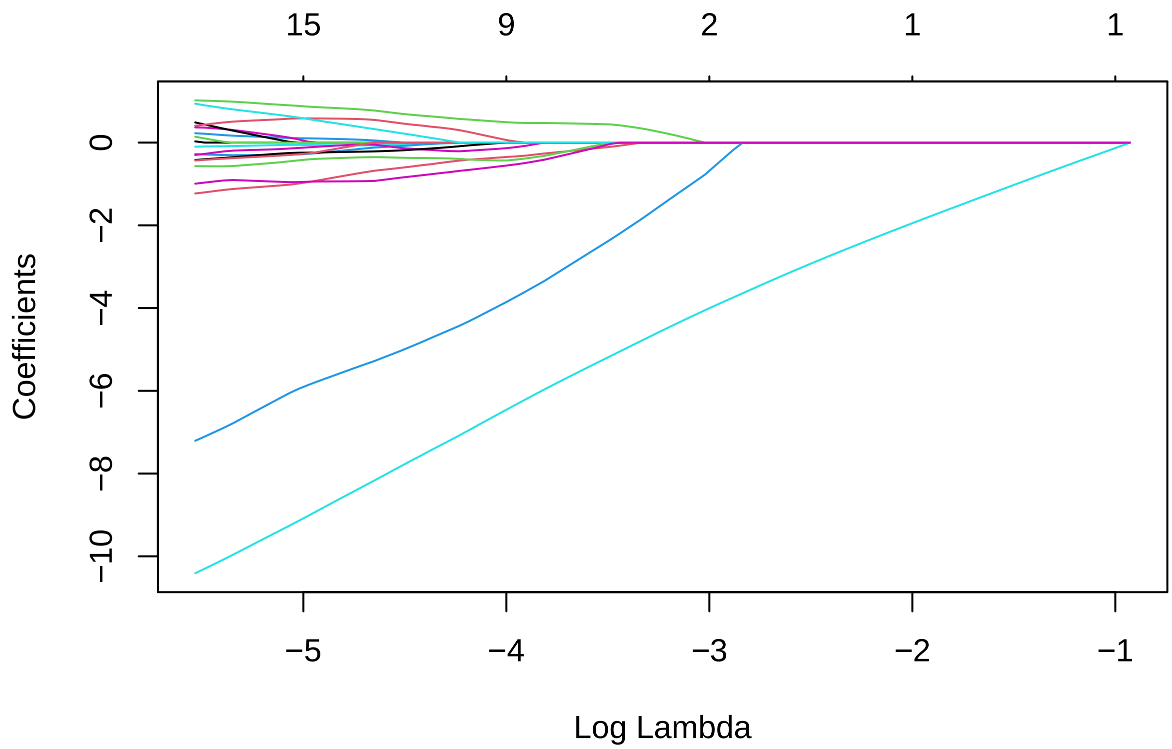
```r
plot(resultsCV[[optimalAlphaIdx]], ylab="mean cross-validated error")
```



Plotting trace curve of coefficients

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

```r
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
```

```
# Non-zero coefficients
coef(trainedModel)
```

```
## 232 x 1 sparse Matrix of class "dgCMatrix"
##                        s0
## (Intercept)   3.03682485
## YAL046C       .
## YAL061W       .
## YAR029W       .
```

```
## YBL009W      .
## YBL059W      .
## YBL081W      .
## YBR025C      .
## YBR108W      .
## YBR139W      .
## YBR141C      .
## YBR147W      .
## YBR187W      .
## YBR238C      .
## YBR246W      .
## YCL002C      .
## YCL044C      .
## YCL045C      .
## YCR016W      .
## YCR061W      .
## YCR076C      .
## YCR090C      .
## YCR095C      .
## YCR102C      .
## YDL119C      .
## YDL121C      .
## YDL133W      .
## YDL156W      .
## YDL157C      .
## YDL180W      .
## YDL193W     -0.99373409
## YDL203C      .
## YDL233W      .
## YDR049W      .
## YDR061W      .
## YDR067C      .
## YDR115W      .
## YDR119W      .
## YDR124W      .
## YDR161W      .
## YDR186C      .
## YDR196C      .
## YDR248C      .
## YDR339C      .
## YDR348C      .
## YDR391C      .
## YDR437W      .
## YDR531W      .
## YEL007W      .
## YEL043W      0.84129092
## YEL057C      .
## YER071C      .
## YER078C      .
## YER079W      .
## YER128W      .
## YER139C      .
## YER156C      .
## YER163C      .
```

```
## YER187W       .
## YFL049W       .
## YFL052W       .
## YFL054C       .
## YFR011C       .
## YFR042W       .
## YGL036W       0.06459498
## YGL057C       .
## YGL081W       .
## YGL114W       .
## YGL196W       .
## YGL235W       .
## YGL242C       .
## YGL250W       .
## YGR016W       .
## YGR021W       .
## YGR046W       .
## YGR050C      -0.04368219
## YGR058W       .
## YGR106C       0.08040409
## YGR122W       .
## YGR130C       .
## YGR203W       .
## YGR237C       .
## YGR251W       .
## YGR277C       .
## YHL018W       .
## YHL029C       0.53966984
## YHR020W       .
## YHR036W       .
## YHR045W       .
## YHR048W       .
## YHR087W       .
## YHR112C       .
## YHR140W       .
## YHR177W       .
## YIL006W       .
## YIL039W       .
## YIL054W       .
## YIL055C       .
## YIL083C       .
## YIR016W       .
## YIR024C      -0.28248755
## YIR042C       .
## YJL010C       .
## YJL045W       .
## YJL046W       .
## YJL051W       .
## YJL057C       .
## YJL068C       .
## YJL070C       .
## YJL097W       .
## YJL123C       .
## YJL131C       .
```

```
## YJL147C       .
## YJL149W       .
## YJL160C       .
## YJL205C       .
## YJR011C       .
## YJR039W       .
## YJR054W       .
## YJR085C       .
## YJR088C       .
## YJR120W       .
## YJR141W       .
## YKL033W       .
## YKL037W       .
## YKL044W       .
## YKL069W       .
## YKL098W       .
## YKL121W       .
## YKL137W       .
## YKL171W       .
## YKL206C       .
## YKL207W       .
## YKR018C       .
## YKR023W       .
## YKR043C       .
## YKR096W       .
## YKR104W     -9.14087563
## YLL023C       .
## YLL029W       .
## YLL032C       .
## YLL033W       .
## YLL049W       .
## YLR012C       .
## YLR021W     -0.19429268
## YLR065C       .
## YLR149C       .
## YLR152C       .
## YLR168C       .
## YLR193C       .
## YLR218C       .
## YLR225C       .
## YLR243W       .
## YLR278C       .
## YLR281C       .
## YLR287C     -0.24255609
## YLR290C       .
## YLR345W       .
## YLR346C       .
## YLR404W       .
## YLR415C       .
## YLR426W       .
## YML003W       .
## YML020W       .
## YML030W       .
## YML053C       .
```

```
## YML108W      .
## YML125C      .
## YMR018W      .
## YMR031C      .
## YMR034C      .
## YMR111C      .
## YMR155W      .
## YMR166C      .
## YMR233W      .
## YMR269W      .
## YMR291W      .
## YMR293C     -0.29438359
## YMR315W      .
## YNL010W      .
## YNL040W      .
## YNL056W      .
## YNL080C      .
## YNL086W      .
## YNL100W      .
## YNL115C      .
## YNL149C      .
## YNL181W     -0.46827635
## YNL194C      .
## YNL213C      .
## YNL260C     -6.00008468
## YNR020C      .
## YOL003C      .
## YOL008W      .
## YOL036W      .
## YOL053W      .
## YOL057W      .
## YOL111C      0.60468672
## YOL124C      .
## YOL138C      .
## YOR131C      .
## YOR164C      .
## YOR186W      .
## YOR228C      .
## YOR246C      .
## YOR262W      .
## YOR285W      .
## YOR292C      .
## YOR356W      .
## YPL030W      .
## YPL034W      .
## YPL039W      .
## YPL041C      .
## YPL066W      .
## YPL098C     -0.95966632
## YPL109C      .
## YPL141C      .
## YPL144W      .
## YPL150W      .
## YPL158C      .
```

```
## YPL183C      .
## YPL184C      .
## YPR022C      .
## YPR045C      .
## YPR078C      .
## YPR085C      .
## YPR089W      .
## YPR098C      .
## YPR117W      .
## YPR118W      .
## YPR152C      .
## YPR172W      .
```

Selected variables were those that are associated with non-zero coefficients.

```
# Selected variables
which(rowSums(coef(trainedModel)) != 0)
```

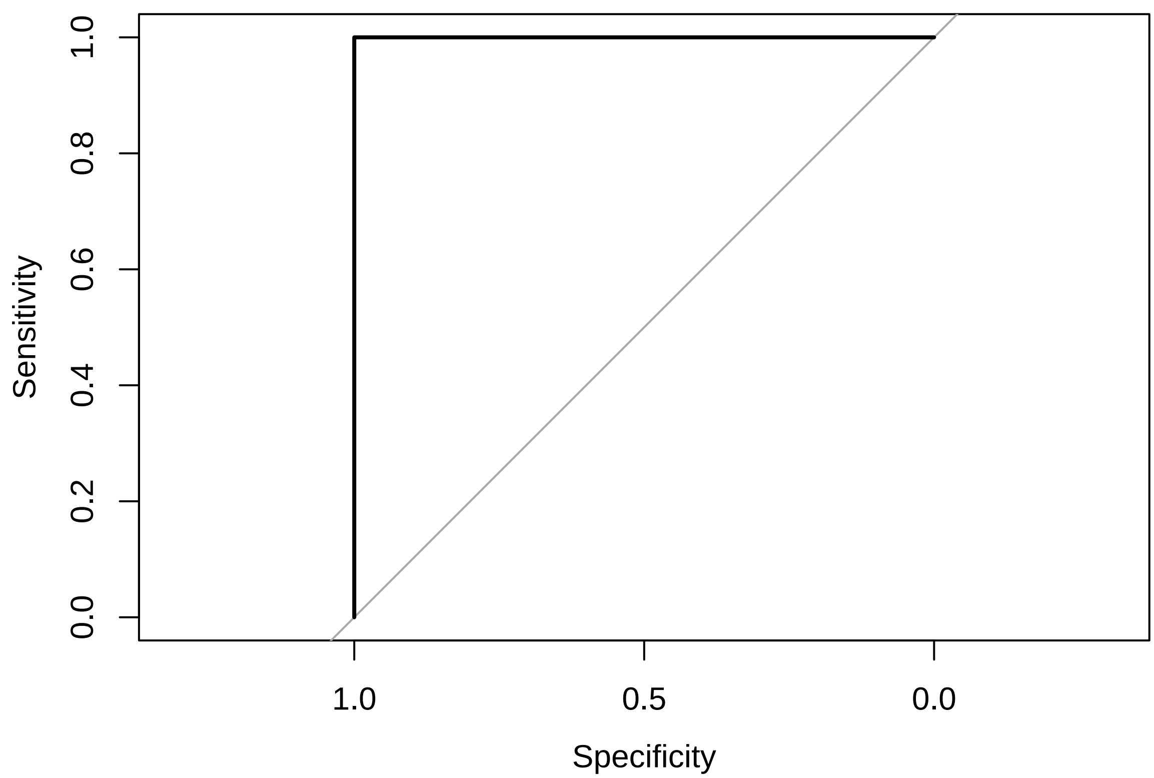
```
## (Intercept)     YDL193W     YEL043W     YGL036W     YGR050C     YGR106C     YHL029C     YIR024C
##           1          31          50          65          76          78          86         101
##     YKR104W     YLR021W     YLR287C     YMR293C     YNL181W     YNL260C     YOL111C     YPL098C
##         138         145         156         178         188         191         198         215
```

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
```

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