Project 10

Santiago Castro Dau, June Monge, Rachita Kumar, Sarah Lötscher

2022-05-05

Problem 27: Uniqueness of predictions from the lasso

1.

We will prove the statement of the exercise by contradiction. Let us define two solutions $\hat{\beta}_1$ and $\hat{\beta}_2 \in S$ such that $X\hat{\beta}_1 \neq X\hat{\beta}_2$. Since the solution set S of the convex minimization problem is convex, $\alpha\hat{\beta}_1 + (1-\alpha)\hat{\beta}_2 \in S$, and given the strict convexity of the loss function, $f(\alpha\hat{\beta}_1 + (1-\alpha)\hat{\beta}_2) < \alpha f(\hat{\beta}_1) + (1-\alpha)f(\hat{\beta}_2)$, we obtain the following:

$$\frac{1}{2}||y - X(\alpha\hat{\beta}_1 + (1-\alpha)\hat{\beta}_2)||_2^2 + \lambda||\alpha\hat{\beta}_1 + (1-\alpha)\hat{\beta}_2||_1 < \alpha\left(\frac{1}{2}||y - X\hat{\beta}_1)||_2^2 + \lambda||\hat{\beta}_1||_1\right) + (1-\alpha)\left(\frac{1}{2}||y - X\hat{\beta}_2)||_2^2 + \lambda||\hat{\beta}_2||_1\right)$$

$$\frac{1}{2}||y - X(\alpha\hat{\beta}_1 + (1-\alpha)\hat{\beta}_2)||_2^2 + \lambda||\alpha\hat{\beta}_1 + (1-\alpha)\hat{\beta}_2||_1 < \alpha c^* + (1-\alpha)c^*$$

$$\frac{1}{2}||y - X(\alpha\hat{\beta}_1 + (1-\alpha)\hat{\beta}_2)||_2^2 + \lambda||\alpha\hat{\beta}_1 + (1-\alpha)\hat{\beta}_2||_1 < c^*$$

Note that due to the convexity of the problem $\{\alpha\hat{\beta}_1 + (1-\alpha)\hat{\beta}_2,\ \hat{\beta}_1,\ \hat{\beta}_2\} \in S, \forall\alpha \in (0,1)$, thus, the c corresponding to the solution $\alpha\hat{\beta}_1 + (1-\alpha)\hat{\beta}_2$ should not be lower than c, as we have stablished that $\hat{\beta}_1$ and $\hat{\beta}_2 \in S$ give rise to the optimal criterion value c^*. Overall, this implies that $X\hat{\beta}_1 = X\hat{\beta}_2$.

2.

As proven above, given that $X\hat{\beta}_1 = X\hat{\beta}_2$, then it holds that:

$$\frac{1}{2}||y - X\hat{\beta}_1)||_2^2 = \frac{1}{2}||y - X\hat{\beta}_2)||_2^2$$

Given that the optimal criterion value c^* is identical for both solutions, we obtain that

$$\frac{1}{2}||y - X\hat{\beta}_1)||_2^2 + \lambda||\hat{\beta}_1||_1 = \frac{1}{2}||y - X\hat{\beta}_1)||_2^2 + \lambda||\hat{\beta}_2||_1\lambda||\hat{\beta}_1||_1 = \lambda||\hat{\beta}_2||_1$$

This shows that $||\hat{\beta}_1||_1 = ||\hat{\beta}_2||_1$.

Problem 28: Bayesian priors as regularizers

To recover the least squared estimator in the Bayesian setting one seeks to perform MAP estimation on the probability of the weights given the data $P(w|y_{1:n}, x_{1:n})$, since maximizing this probability is equivalent to finding the most probable weights $\hat{w}$, where w is the weight vector (note that we use w instead of β). Here $y_{1:n}$ are the labels and $x_{1:n}$ are the corresponding feature vectors. Decomposing $P(w|y_{1:n}, x_{1:n})$ using Bayes rule yields the following.

$$P(w|y_{1:n}, x_{1:n}) = \frac{P(w, y_{1:n}, x_{1:n})}{P(y_{1:n}, x_{1:n})}$$

$$= \frac{P(y_{1:n}|w, x_{1:n})P(w|x_{1:n})P(x_{1:n})}{P(y_{1:n}|x_{1:n})P(x_{1:n})}$$

Since the prior doesn't depend on the data we can simplify $P(w|x_{1:n})$ to $P(w)$.

$$= \frac{P(y_{1:n}|w, x_{1:n})P(w)}{P(y_{1:n}|x_{1:n})}$$

Since we are performing MAP estimation we seek to maximized this probability by finding the best set of weights $\hat{w}$.

$$\hat{w} = \underset{w}{\text{argmax}} \frac{P(y_{1:n}|w, x_{1:n})P(w)}{P(y_{1:n}|x_{1:n})}$$

First we will derive the expression for the likelihood $P(y_{1:n}|w, x_{1:n})$ by performing MLE, which we can then use in this primary optimization problem.

The likelihood

The linear regression model is given by $Y = h(X) + \epsilon$, where $\epsilon \sim \mathcal{N}(0, \sigma^2)$ and $h(X) = w^T X$. Then we seek to find a simplified expression of this quantity $P(y_{1:n}|w, x_{1:n})$ by performing MLE, which means finding w such that $P(y_{1:n}|w, x_{1:n})$ is maximized. Since σ^2 is assumed to be fixed, we can omit it in our expressions of conditional probability.

$$\hat{w} = \underset{w}{\text{argmax}} P(y_{1:n}|x_{1:n}, w)$$

Assuming that all n data points are $i.i.d$ and by applying the log, we can rewrite this expression as follows.

$$= \underset{w}{\text{argmin}} - \sum_{i=1}^{n} \log P(y_i|x_i, w)$$

Then we can plug in our assumptions about the model for each data point.

$$\begin{aligned} -\log P(y_i|x_i, w, \sigma^2) &= -\log \mathcal{N}(y_i; w^T x_i, \sigma^2) \\ &= -\log \frac{1}{\sqrt{2\pi\sigma^2}} \exp -\frac{(y - w^T x)^2}{2\sigma^2} \\ &= \frac{1}{2} \log 2\pi\sigma^2 + \frac{(y - w^T x)^2}{2\sigma^2} \end{aligned}$$

Then for the whole data set we can wirte the following.

$$\hat{w} = \underset{w}{\text{argmin}} \sum_{i=1}^{n} (\frac{1}{2} \log 2\pi\sigma^2 + \frac{(y - w^T x)^2}{2\sigma^2})$$

$$= \underset{w}{\text{argmin}} \frac{n}{2} \log 2\pi\sigma^2 + \frac{1}{2\sigma^2} \sum_{i=1}^{n} (y - w^T x)^2$$

Since the firs term is independent of w we can eliminate it, yielding the expression for the least square loss. From this result we can deduce that least squares linear regression is then equivalent to performing MLE for a conditional linear Gaussian likelihood. One can also clearly see from this demonstration that by performing OLS regression one implicitly places some statistical assumptions on the data, namely that the error is normally distributed and independent of the input (homoscedasticity). This assumption is characteristic of the least squared loss.

$$\hat{w} = \underset{w}{\operatorname{argmin}} \sum_{i=1}^{n} (y - w^T x)^2$$

Comming back to our original problem

Now that we have an expression for the likelihood $P(y_{1:n}|x_{1:n}, w, \sigma^2)$ we can plug it into our original problem along with a Laplassian prior and aplaying the log function to the whole expression. By doing this we obtain the Lasso regression estimator.

$$\hat{w} = \underset{w}{\operatorname{argmin}} - \log P(w) - \log P(y_{1:n}|w, x_{1:n}) + \underbrace{\log P(y_{1:n}|x_{1:n})}_{\text{indp. of w}}$$

$$= \underset{w}{\operatorname{argmin}} - \log \prod_{j=1}^{p} \frac{1}{2b} \exp\left(\frac{-|w_j|}{b}\right) + \frac{1}{2\sigma^2} \sum_{i=1}^{n} (y - w^T x)^2$$

$$- \log \prod_{j=1}^{p} \frac{1}{2b} \exp\left(\frac{-|w_j|}{b}\right) = -\left(\underbrace{\sum_{j=1}^{p} \log \frac{1}{2b}}_{\text{indp.of w}} + \sum_{j=1}^{p} \log \exp\left(\frac{-|w_j|}{b}\right)\right)$$

$$\hat{w} = \underset{w}{\operatorname{argmin}} \frac{1}{b} \sum_{j=1}^{p} |w_j| + \frac{1}{2\sigma^2} \sum_{i=1}^{n} (y - w^T x)^2$$

Finally we can group together the constants into a single one λ and write the expression in matrix notation.

$$\hat{w} = \underset{w}{\operatorname{argmin}} \lambda ||w||_1 + \frac{1}{2} ||\mathrm{y} - \mathrm{X}w||_2^2$$

Problem 29: Variable selection under various norms :

```
## Loading required package: lattice
## Loading required package: Matrix
## Warning: package 'Matrix' was built under R version 4.0.5
## Loaded glmnet 4.1-4
## Type 'citation("pROC")' for a citation.
##
## Attaching package: 'pROC'
## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
##
## Attaching package: 'dplyr'
## The following objects are masked from 'package:stats':
##
##     filter, lag
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

1. Load the data and construct the design matrix X and response variable y, respectively. Randomly split the data into training set (70%) and test set (30%). For reproducibility set the seed to 42 in the beginning.

```
#Read in dataset
set.seed(42)
load("./yeastStorey.rda")
```

```
#Sample data
Sampledata<- createDataPartition(data$Marker,p = 0.7,list = FALSE)

#Create training data
Traindata<-data[Sampledata, ]

#Create test data
Testdata<-data[-Sampledata, ]
```

```
#Create X and y train
X_train <- apply(as.matrix.noquote(Traindata[,-1]), 2,as.numeric)

Y_train<- Traindata[,1]

#Create X and y test
X_test <- apply(as.matrix.noquote(Testdata[,-1]), 2, as.numeric)

Y_test <- Testdata[,1]
```

2. Using 10-fold cross-validation, find the optimum and optimum using elastic-net model on the training set. Fit the final model with the optimal parameters on the training set. For reducing computation time restrict the search space of to {0, 0.1, 0.2, · · · , 1}.

```
#Get alpha vector    = {0, 0.1, 0.2, · · · , 1}
alphavec= seq(0, 1, 0.1)
```

```
#Get the foldid
foldid <- sample(1:10, size=length(Sampledata), replace = TRUE )


#Find best lambda using 10-fold cross-validation
crossval <- lapply(alphavec, function(n)
  {cv.glmnet(X_train, Y_train, alpha=n, family = "binomial",foldid = foldid, type.measure="mse")})


#Make dataset with the new values for plotting
dd<- do.call(rbind,lapply(1:length(alphavec), function(x){
  cbind.data.frame(alphavec[x],crossval[[x]]$lambda,crossval[[x]]$cvm)}))

colnames(dd)<- c('alpha','lambda','cvm')
```

3. Predict the response on the test dataset using the final model. Plot the cross-validation error as a function of log , trace curve of coefficients as a function of log , and the ROC curve for the optimal . Lastly, report the corresponding AUC (area under the curve) of the ROC curve and the variables selected.

```
#Predict the response on the test dataset

#Get alpha and lambda
param <- filter(dd ,cvm== min(cvm))

#Fit model using trainingdata
model <-glmnet(X_train, Y_train, alpha = param$alpha, family = "binomial" ,lambda = param$lambda)


#Predict Y of testdata
prob <- predict(model, newx = X_test, type = "response")
predY <- ifelse(prob>0.5,1,0)
```

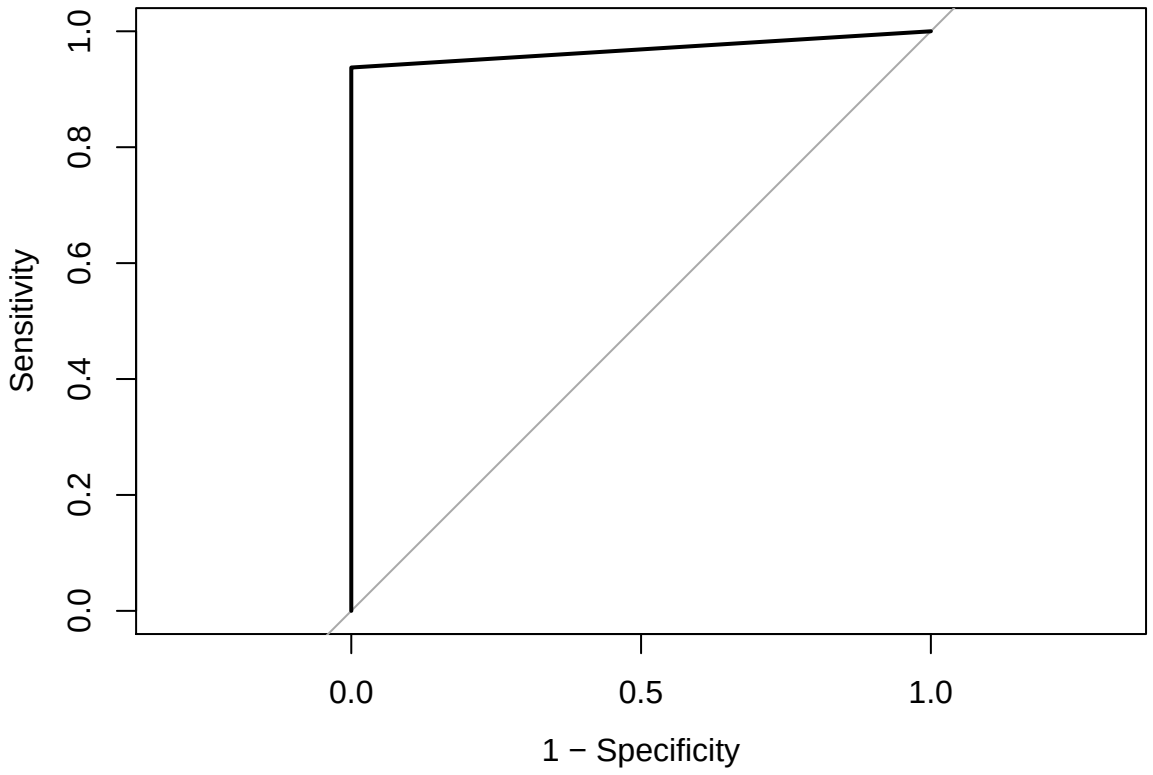
```
#Plot cross-validation error as a function of log

Acc <- roc(Y_test,as.vector(predY))
```

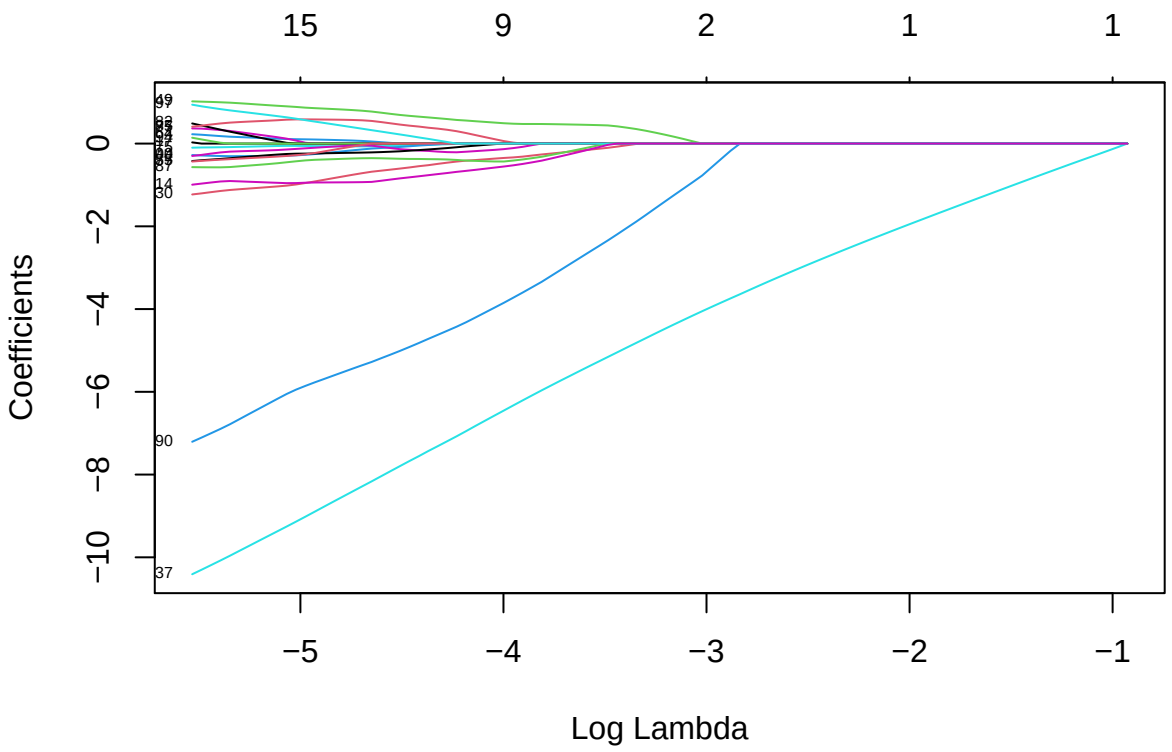
```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

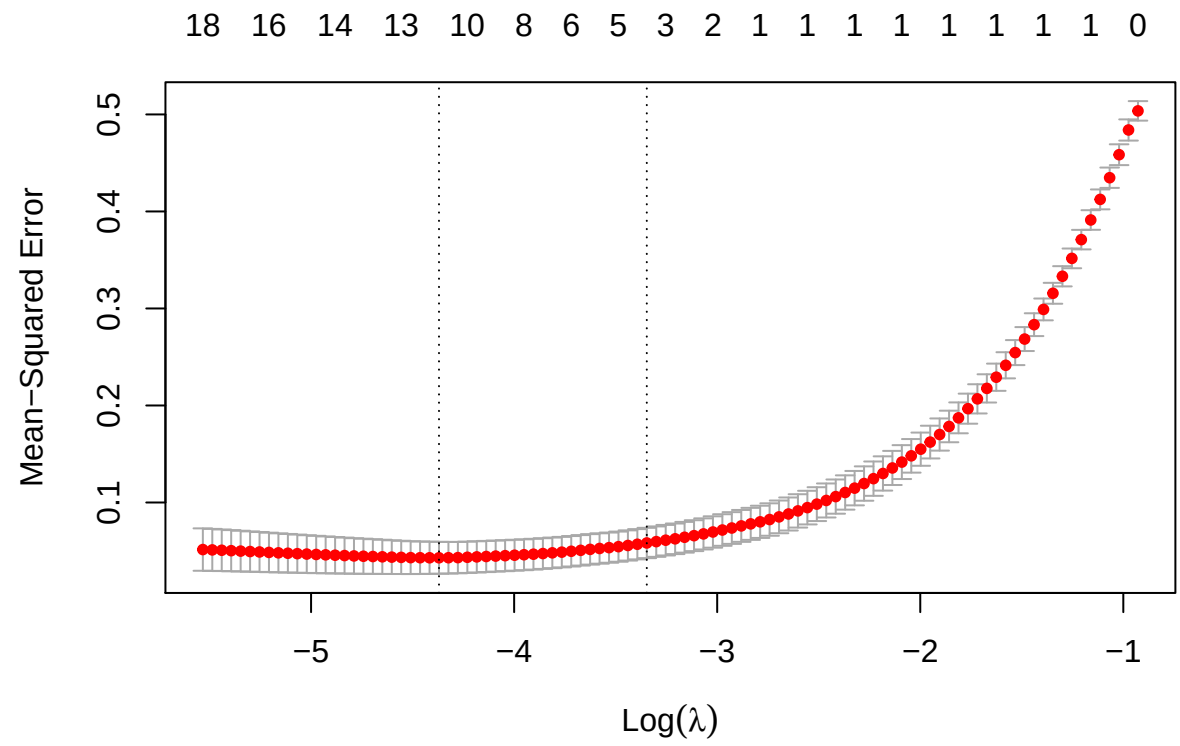
```
plot(Acc, legacy.axes=TRUE)
```




```
#Plot trace curve of coefficients as a function of log
plot(crossval[[11]]$glmnet.fit,"lambda",label=TRUE)
```




```
#ROC curve for the optimal
plot(crossval[[11]])
```




```
#Get the AUC
Acc$auc
```

```
## Area under the curve: 0.9688
```

```r
#Variables/predictors selected
predictors<-coef(crossval[[11]], param$lambda)
predictors@Dimnames[[1]][predictors@i]
```

```
##  [1] "YDL180W" "YEL007W" "YGR046W" "YHL018W" "YIR016W" "YKR096W" "YLR012C"
##  [8] "YLR281C" "YNL149C" "YNL213C" "YOL057W" "YPL066W"
```