

Project 10 Structured sparsity in genetics

Team K - Minghang Li

May 07, 2023

Problem 28: Uniqueness of predictions from the lasso

Proof. To prove that $\mathbf{X}\hat{\beta}_1 = \mathbf{X}\hat{\beta}_2$, it's equivalent to prove $\hat{\beta}_1 = \hat{\beta}_2$. We'll prove by contradiction.

Let the solution set be denoted as S , we have $\hat{\beta}_1, \hat{\beta}_2 \in S$.

Since S is convex, we also have $\alpha\hat{\beta}_1 + (1 - \alpha)\hat{\beta}_2 \in S$.

So that we have:

$$\frac{1}{2} \left\| \mathbf{y} - \mathbf{X} (\alpha\hat{\beta}_1 + (1 - \alpha)\hat{\beta}_2) \right\|_2^2 + \lambda \left\| \alpha\hat{\beta}_1 + (1 - \alpha)\hat{\beta}_2 \right\|_1 = c^* \quad (1)$$

Define the loss function as $f : S \mapsto \left\| \mathbf{y} - \mathbf{X}\beta \right\|_2^2$. We know that the linear loss function is strictly convex, so that we have:

$$f(\alpha\hat{\beta}_1 + (1 - \alpha)\hat{\beta}_2) < \alpha f(\hat{\beta}_1) + (1 - \alpha)f(\hat{\beta}_2) \quad (2)$$

Similarly, since l_1 -norm is convex, we have:

$$\left\| \alpha\hat{\beta}_1 + (1 - \alpha)\hat{\beta}_2 \right\|_1 \leq \alpha \left\| \hat{\beta}_1 \right\|_1 + (1 - \alpha) \left\| \hat{\beta}_2 \right\|_1 \quad (3)$$

Combining Equation 2 and Equation 3, we can start the following derivation:

$$\begin{aligned} & \frac{1}{2} f(\alpha\hat{\beta}_1 + (1 - \alpha)\hat{\beta}_2) + \lambda \left\| \alpha\hat{\beta}_1 + (1 - \alpha)\hat{\beta}_2 \right\|_1 \\ & < \alpha (f(\hat{\beta}_1) + \lambda \left\| \hat{\beta}_1 \right\|_1) + (1 - \alpha) (f(\hat{\beta}_2) + \lambda \left\| \hat{\beta}_2 \right\|_1) \\ & = \alpha c^* + (1 - \alpha)c^* \\ & = c^* \end{aligned} \quad (4)$$

Just a re-write:

$$\frac{1}{2} f(\alpha\hat{\beta}_1 + (1 - \alpha)\hat{\beta}_2) + \lambda \left\| \alpha\hat{\beta}_1 + (1 - \alpha)\hat{\beta}_2 \right\|_1 < c^* \quad (5)$$

Equation 5 contradicts with Equation 1, which suggests that our assumption $\hat{\beta}_1 \neq \hat{\beta}_2$ is wrong.

□

\end{proof}

Problem 29: Ridge regression solution

Given $n \times p$ matrix \mathbf{X} , response n -vector \mathbf{y} , and the parameter p -vector β , consider augmenting \mathbf{X} with rows corresponding to $\sqrt{\lambda}$ times a $p \times p$ identity matrix \mathbf{I} :

$$\mathbf{X}_* = \begin{pmatrix} \mathbf{X} \\ \sqrt{\lambda}\mathbf{I} \end{pmatrix}$$

and \mathbf{y} is similarly augmented with p zeros at its end into \mathbf{y}_* .

Now, the least square objective function on the modified dataset turns out to be:

$$\begin{aligned} & (\mathbf{y}_* - \mathbf{X}_*\beta)^T(\mathbf{y}_* - \mathbf{X}_*\beta) \\ &= \mathbf{y}_*^T\mathbf{y}_* - 2\mathbf{y}_*^T\mathbf{X}_*\beta + \beta^T\mathbf{X}_*^T\mathbf{X}_*\beta \end{aligned} \tag{6}$$

The solution to the linear regression on the modified dataset hence follows:

$$\hat{\beta} = (\mathbf{X}_*^T\mathbf{X}_*)^{-1}\mathbf{X}_*^T\mathbf{y}_* \tag{7}$$

Since we are just padding \mathbf{y} with zeros, the augmented part of \mathbf{X}_* will not have any effect when it is multiplied by \mathbf{y}_*^T :

$$\mathbf{y}_*^T\mathbf{X}_* = \mathbf{y}^T\mathbf{X} \tag{8}$$

And finally with augmented dataset \mathbf{X}_* we have:

$$\begin{aligned} & \mathbf{X}_*^T\mathbf{X}_* \\ &= (\mathbf{X}^T \quad \sqrt{\lambda}\mathbf{I}) \begin{pmatrix} \mathbf{X} \\ \sqrt{\lambda}\mathbf{I} \end{pmatrix} \\ &= \mathbf{X}^T\mathbf{X} + \lambda\mathbf{I} \end{aligned} \tag{9}$$

Plug Equation 8 and Equation 9 back into Equation 7, we delightfully see that performing linear regression on the modified dataset gives:

$$\hat{\beta} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{y} \tag{10}$$

which is just the ridge regression solution.

Problem 30: Variable selection under various norms

Load data and split into training and test set

```
# load data
set.seed(42)
load("yeastStorey.rda")

sample <- createDataPartition(data$Marker, p = 0.7, list = FALSE)

train <- data[sample, ]
test  <- data[-sample, ]

X_train <- as.matrix(train[, -1])
y_train <- train[, 1]

X_test  <- as.matrix(test[, -1])
y_test  <- test[, 1]
```

Do 10-fold cross validation

```
alphas <- seq(0, 1, 0.1)

foldid <- sample(1:10, size = length(sample), replace = TRUE)

num_cores <- detectCores()
registerDoParallel(num_cores)
start <- Sys.time()
res <- foreach (alpha = alphas, .combine = c, .options.RNG = 42) %dorng% {
  cv_res <- cv.glmnet(X_train, y_train, alpha = alpha,
                      family = "binomial", foldid = foldid,
                      type.measure = "mse")

  list(cv_res)
}
end <- Sys.time()
end - start

## Time difference of 0.3338723 secs

stopImplicitCluster()

df <- do.call(rbind,
             lapply(1:length(alphas),
                    function(x) {
                      cbind.data.frame(alphas[x],
                                       res[[x]]$lambda,
                                       res[[x]]$cvm)
                    })
            )
colnames(df) <- c("alpha", "lambda", "cvm")
```

```
param <- filter(df, cvm = min(cvm))
param
```

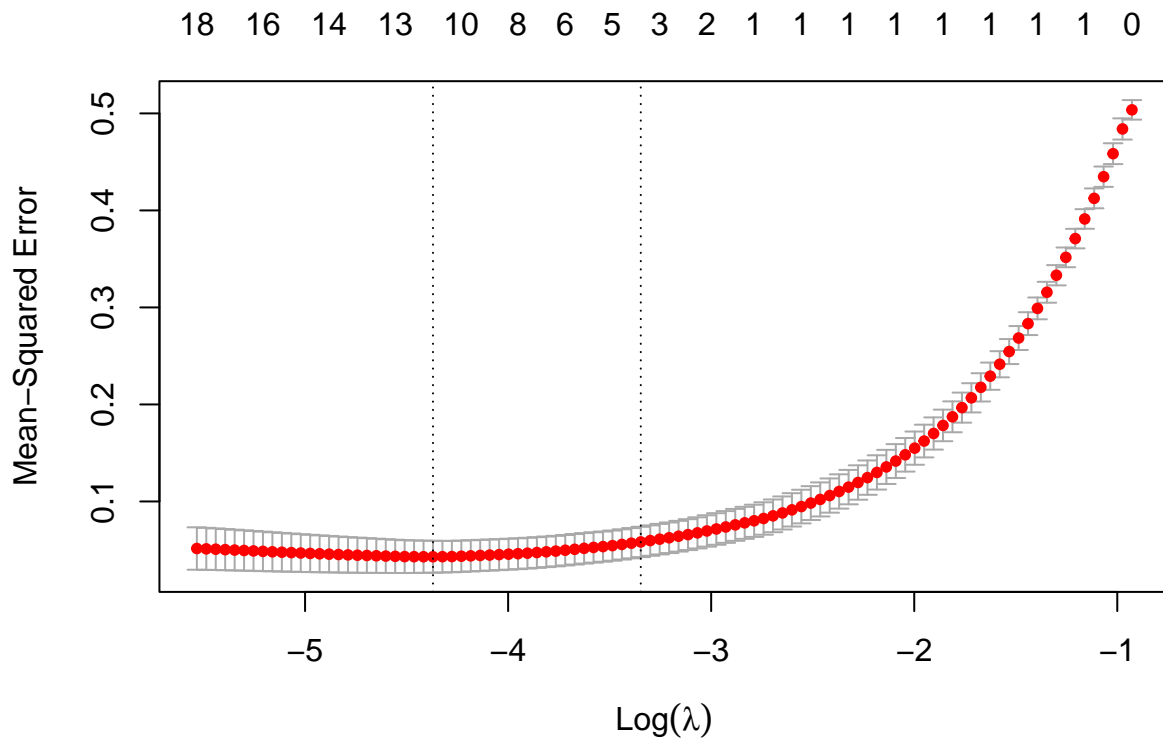
```
##   alpha   lambda      cvm
## 1     1 0.0126508 0.04302968
```

For the optimal α (here $\alpha = 1$)...

```
opt_idx <- which(alphas == param$alpha)
res.opt_alpha <- res[[opt_idx]]
```

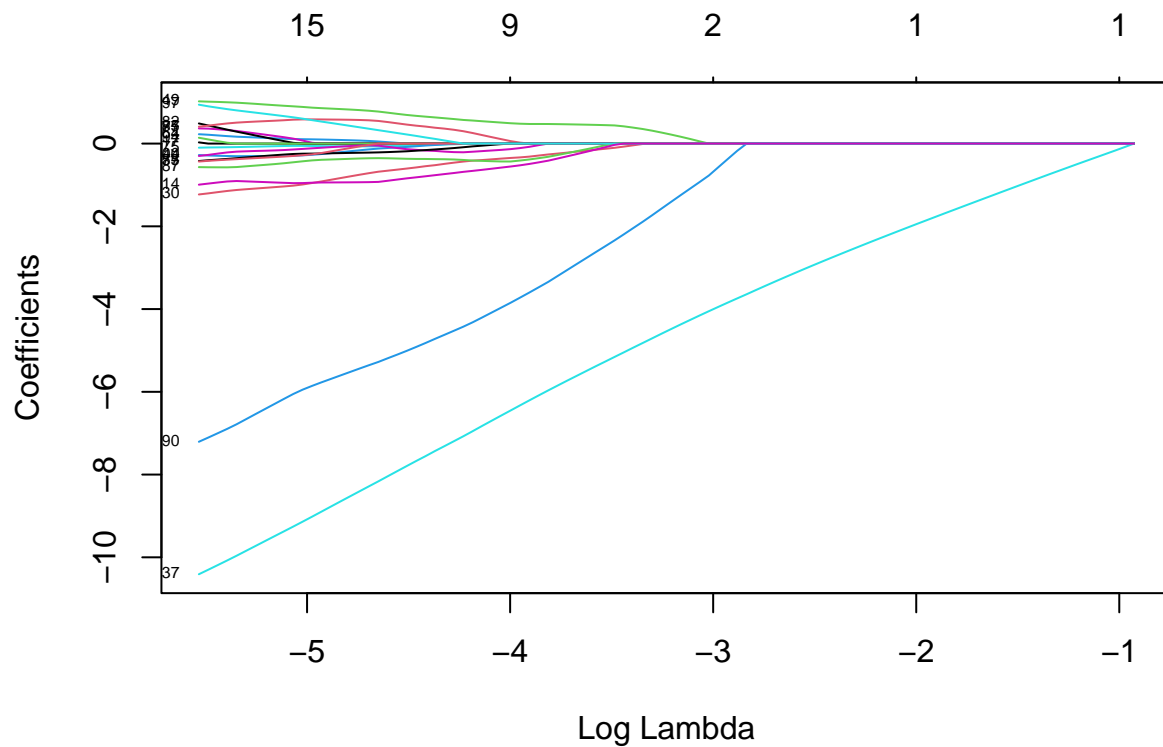
Plot the mean cross-validated error as a function of $\log \lambda$.

```
plot(res.opt_alpha)
```



Plot the trace curve of coefficients as a function of $\log \lambda$.

```
plot(res.opt_alpha$glmnet.fit, "lambda", label=TRUE)
```



Fit the final model and predict the response

```
model <- glmnet(X_train, y_train,
               alpha = param$alpha,
               family = "binomial",
               lambda = param$lambda)
```

```
prob <- predict(model, newx = X_test, type = "response")
y_pred <- ifelse(prob > 0.5, 1, 0)
```

Report the variables selected.

```
selected <- coef(res.opt_alpha, param$lambda)
selected@Dimnames[[1]][selected@i]
```

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

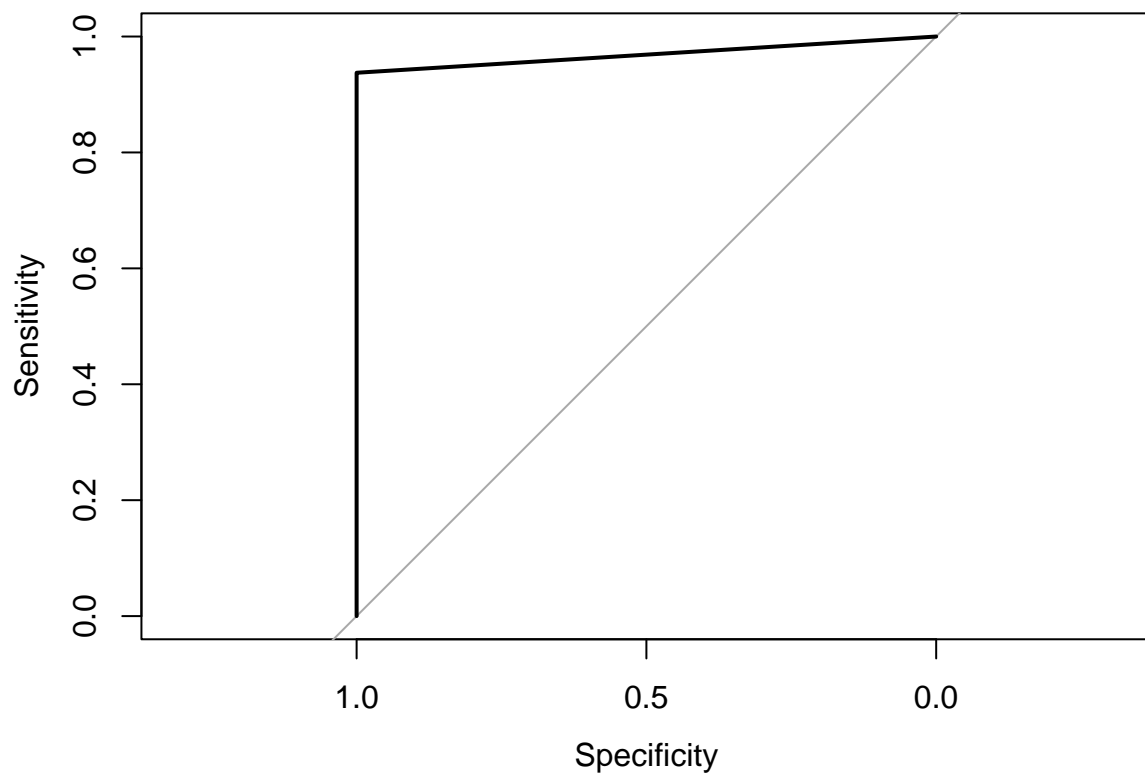
Plot ROC...

```
test_roc <- roc(y_test, as.vector(y_pred))
```

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

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

```
plot(test_roc)
```



... and report the corresponding AUC.

```
test_roc$auc
```

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