In-class session 2

Ishwara Hegde

11/1/2020

The goal of this in-class session was to explore:

1. Prediction – how to split data into test and train and why we need to do it.
2. Regularization using the glmnet package.
3. Get some practice using Random Forests in R for prediction.

Preamble

Loading packages and setting seed :

```
#Loading the packages needed:

library(glmnet)
library(splines)
library(pdp)
library(randomForest)

#Setting the seed and clearing workspace
set.seed(1234)
rm(list = ls())
```

Loading data:

```
data_africa <-
read.table("http://www-stat.stanford.edu/~tibs/ElemStatLearn/datasets/SAheart.data",
    sep=",",head=T,row.names=1)

#Extracting the chd, sbp, alcohol, tobaco and obesity columns as
#matrices of dim 462x1.

Y <- matrix(data_africa[,'chd'],nrow=462,ncol = 1)
X_1 <- matrix(data_africa[,'sbp'],nrow=462,ncol=1)
X_2 <- matrix(data_africa[,'alcohol'],nrow=462,ncol=1)
X_3 <- matrix(data_africa[,'tobacco'],nrow=462,ncol=1)
X_4 <- data_africa[,'obesity']

df_af <- 10

#Creating the splines with polynomials of order
X_1s <- ns(X_1, df = df_af)
X_2s <- ns(X_2, df = df_af)
X_3s <- ns(X_3, df = df_af)
X_4s <- ns(X_4, df = df_af)
```

```
data_X <- cbind(X_1s ,X_2s ,X_3s, X_4s)
data_Y <- Y
```

Problem 1

Your goal is to predict Y using $ X_{ks} $.

Split your data randomly into training set and test set.

```
# Creating the test-train split

n <- length(Y)
n_train <- floor(n*0.8)
n_test <- n - n_train
index_tr <- sample(1:n,n_train)
index_test <- setdiff(1:n,index_tr)

data_X_train <- data_X[index_tr,]
data_Y_train <- data_Y[index_tr]

data_X_test <- data_X[index_test,]
data_Y_test <- data_Y[index_test]

#Training the models
#Recall that alpha =0 is the ridge penalty and alpha =1 is the lasso one

glm_res <- glmnet(data_X_train,data_Y_train, family = 'binomial',
                  alpha = 1)
glm_cv <- cv.glmnet(data_X_train,data_Y_train, family = 'binomial',
                    alpha = 1)
opt_lambda <- glm_cv$lambda.min
glm_res_opt <- glmnet(data_X_train,data_Y_train, family = 'binomial',

                                          alpha = 1,lambda = opt_lambda)

#Fitting a lasso with 0 regularization penalty
glm_res_nonr <- glmnet(data_X_train,data_Y_train, family = 'binomial',
                       alpha = 1,lambda =0)

#Testing our models against the test data
link_pred_reg <- predict(glm_res_opt,data_X_test,type = 'link')
link_pred_nonr <- predict(glm_res_nonr,data_X_test,type = 'link')
#basic pr is just a prediction based on the log odds ratio of the mean
#of the training data

basic_pr <- log(mean(data_Y_train)/(1-mean(data_Y_train)))

#evaluating the empirical risk using logistic loss function
emp_risk_test_reg <-mean(log(1+exp(link_pred_reg)) - data_Y_test*link_pred_reg)
emp_risk_test_nonr <-mean(log(1+exp(link_pred_nonr)) -
                                  data_Y_test*link_pred_nonr)

emp_risk_test_simp <-mean(log(1+exp(basic_pr)) - data_Y_test*basic_pr)
```

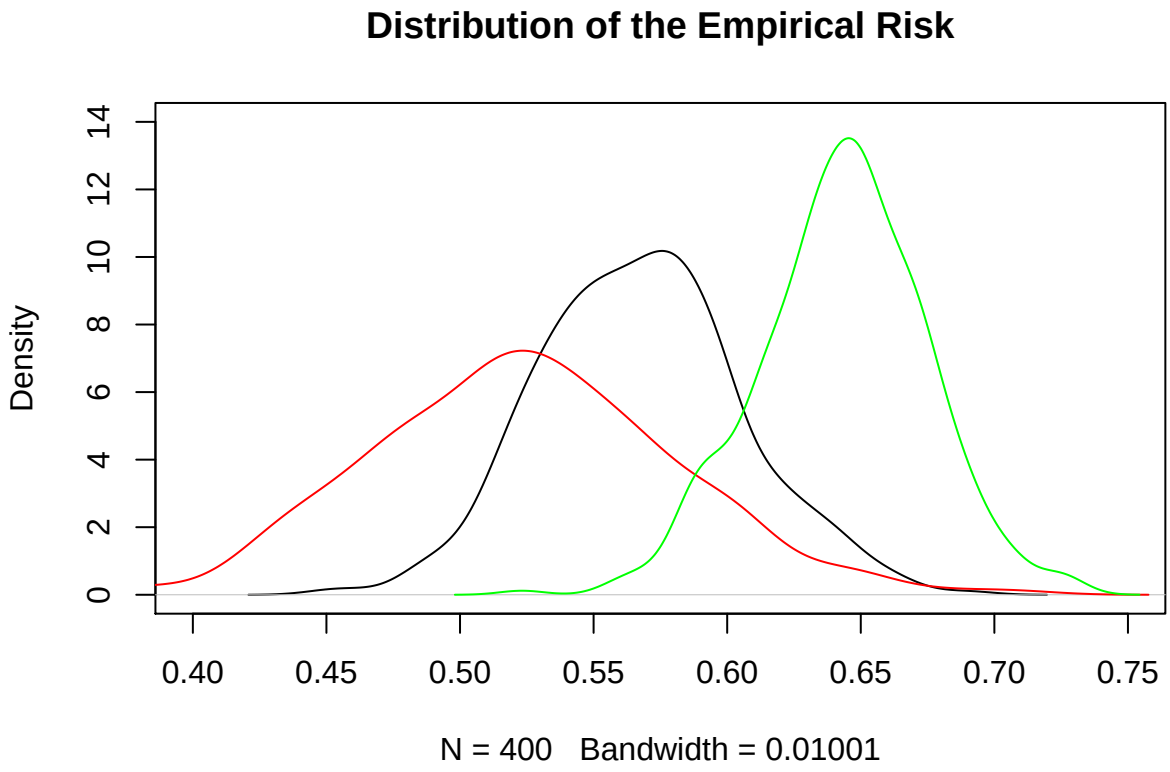
We see that in this case when we include a large number of splines (10) regu- larization becomes important. The risk of the model with no regularization is dominated both by the simple model and the regularized model. The regularized model performs best overall.

Problem 2

Do the same exercise as in Problem 1, but now apply it to bootstrap replications of the data: sample n datapoints with replacement, and do the same thing as above. Repeat it B = 400 times and plot the distribution of the empirical risk on the test data for three estimators.

```
B <- 400
results <- do.call(rbind,lapply(1:B, function(b){

    index_b <- sample(1:n,n,replace = TRUE)
    data_X_b <- data_X[index_b,]
    data_Y_b <- data_Y[index_b]


    index_tr_b <- sample(1:n,n_train)
    index_test_b <- setdiff(1:n,index_tr)

    data_X_train_b <- data_X_b[index_tr_b,]
    data_Y_train_b <- data_Y_b[index_tr_b]

    data_X_test_b <- data_X_b[index_test_b,]
    data_Y_test_b <- data_Y_b[index_test_b]

    glm_res_b <- glmnet(data_X_train_b,data_Y_train_b,
                        family = 'binomial',alpha = 1)
    glm_cv_b <- cv.glmnet(data_X_train_b,data_Y_train_b,
                          family = 'binomial',alpha = 1)
    opt_lambda_b <- glm_cv_b$lambda.min
    glm_res_opt_b <- glmnet(data_X_train_b,data_Y_train_b,
                    family = 'binomial',alpha = 1,lambda = opt_lambda)
    glm_res_nonr_b <- glmnet(data_X_train_b,data_Y_train_b,
                             family = 'binomial',alpha = 1,lambda =0)

    link_pred_reg_b <- predict(glm_res_opt_b,data_X_test_b,type = 'link')
    link_pred_nonr_b <- predict(glm_res_nonr_b,data_X_test_b,type = 'link')
    basic_pr_b <- log(mean(data_Y_train_b)/(1-mean(data_Y_train_b)))

    emp_risk_test_reg_b <-mean(log(1+exp(link_pred_reg_b)) -
                                       data_Y_test_b*link_pred_reg_b)
    emp_risk_test_nonr_b <-mean(log(1+exp(link_pred_nonr_b)) - data_Y_test_b*link_pred_nonr_b)
    emp_risk_test_simp_b <-mean(log(1+exp(basic_pr_b)) -
                                        data_Y_test_b*basic_pr_b)

    return(c(emp_risk_test_reg_b,emp_risk_test_nonr_b,emp_risk_test_simp_b))
}))



plot(density(results[,1]),main = 'Distribution of the Empirical Risk',
     col = 'black',xlim = c(0.4,0.75), ylim =c(0,14))
lines(density(results[,2]), col = 'red')
```

```
lines(density(results[,3]), col = 'green')
```

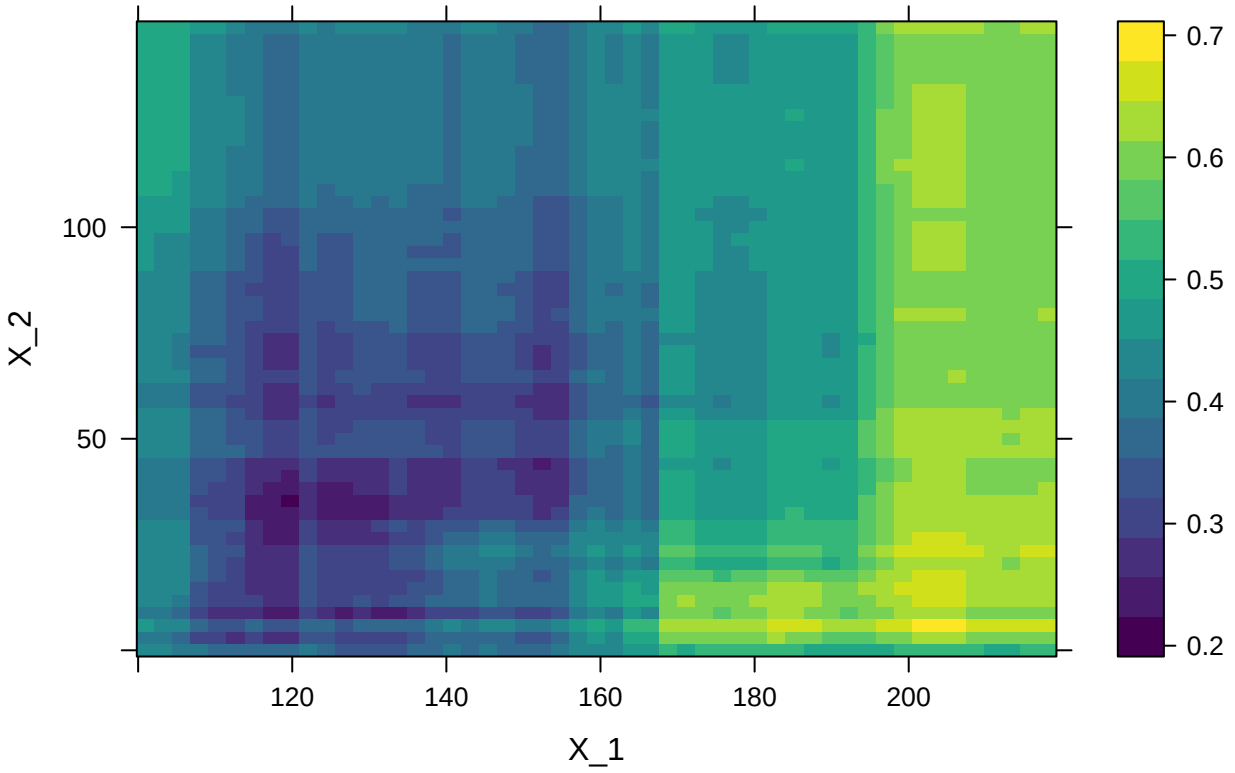


Black is the risk from the regularized model, red from the non-regularized and green from the simple model.

Problem 3

Run a random forest algorithm to predict Y with original characteristics. Then construct a partial dependence plot for X_1 and X_2

```
data_rf <- cbind(Y,X_1,X_2,X_3,X_4)
colnames(data_rf)<-c("Y","X_1","X_2","X_3","X_4")
rf_res <- randomForest(Y~X_1+X_2+X_3+X_4, data = data_rf,importance = TRUE)
pd <- partial(rf_res, pred.var = c("X_1","X_2"))

pdp1 <- plotPartial(pd)
pdp1
```



This Partial Dependence Plot shows the probability of Coronary Heart Disease (chd) and the interaction of X_2 (current alcohol consumption) and X_4 (obesity).

For instance, low obesity and high alcohol consumption show a high probability of heart disease. Perhaps, due to very high levels of alcohol consumption per unit of bodyweight. Similarly, those with higher obesity (>36) also show a higher probability of heart disease.