University of Montpellier Year 2024-2025
HAX912X - IDIL - Machine Learning

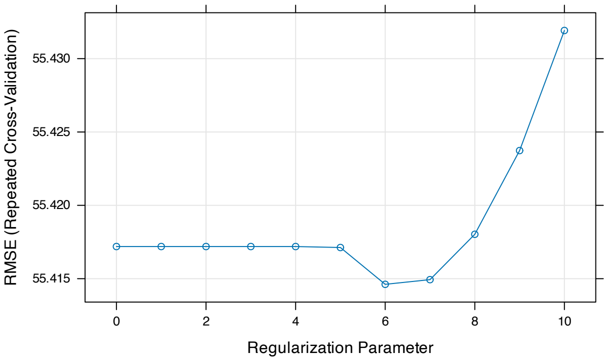
Continuous assessment exam

Duration 1 hour - Without document
October 3, 2024

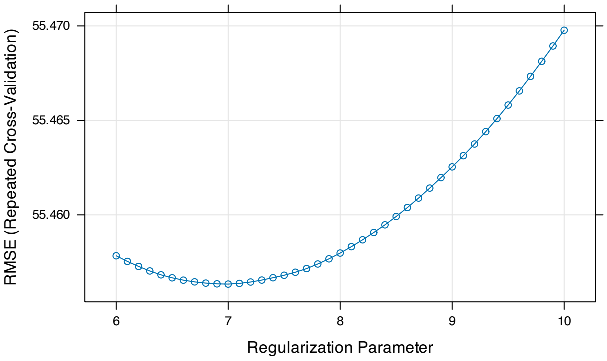
Exercise 1 (14 pts)

Describe and comment on the results obtained using the R code below.

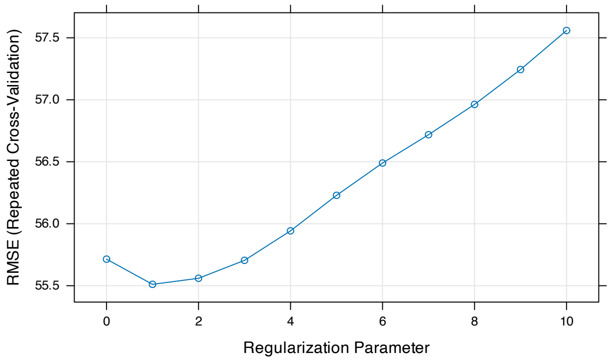
```
data(diabetes)
dim(diabetes$x)
# [1] 442  10
x <- diabetes$x
y <- diabetes$y

colMeans(diabetes$x)
#           age           sex           bmi           map            tc           ldl
# -3.587816e-16  9.321980e-17 -7.993543e-16  1.360463e-16 -9.117895e-17  1.263679e-16
#           hdl           tch           ltg           glu
# -4.505571e-16  3.834131e-16 -3.814488e-16 -3.428522e-16

diag(var(diabetes$x))
#         age         sex         bmi         map          tc         ldl         hdl
# 0.002267574 0.002267574 0.002267574 0.002267574 0.002267574 0.002267574 0.002267574
#         tch         ltg         glu
# 0.002267574 0.002267574 0.002267574

as.vector(createDataPartition(1:20, p = 0.8, list = FALSE))
# [1]  1  3  4  5  6  7  9 10 12 13 14 15 17 18 19 20

set.seed(123)
trainIndex <- createDataPartition(y, p = 0.8, list = FALSE)
x_train <- x[trainIndex, ]
x_test <- x[-trainIndex, ]
y_train <- y[trainIndex]
y_test <- y[-trainIndex]

train_control <- trainControl(method = "repeatedcv", number = 5, repeats = 10)

model1 <- train(
  x_train, y_train,
  method = "glmnet",
  tuneGrid = expand.grid(alpha = 0, lambda = seq(0, 10, by = 1)),
  trControl = train_control
)
plot(model1)
```



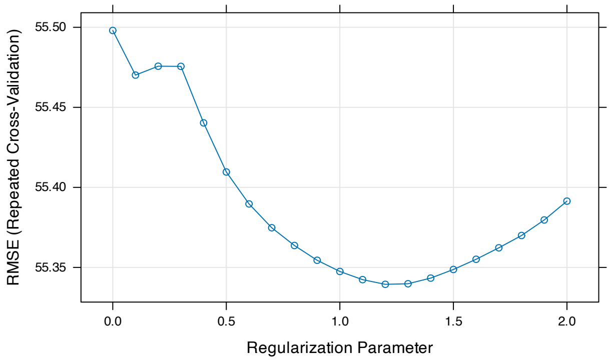
```
model1 <- train(
  x_train, y_train,
  method = "glmnet",
  tuneGrid = expand.grid(alpha = 0, lambda = seq(6, 10, by = 0.1)),
  trControl = train_control
)
plot(model1)
```



```
model1$bestTune$lambda
# [1] 7
model1$results[model1$results$lambda==model1$bestTune$lambda,]
#    alpha lambda     RMSE  Rsquared      MAE   RMSESD RsquaredSD    MAESD
# 11     0      7 55.45634 0.4748846 45.12797 3.486439 0.07605376 3.179491

model2 <- train(
  x_train, y_train,
  method = "glmnet",
  tuneGrid = expand.grid(alpha = 1, lambda = seq(0, 10, by = 1)),
  trControl = train_control
)
plot(model2)
```



```
model2 <- train(
  x_train, y_train,
  method = "glmnet",
  tuneGrid = expand.grid(alpha = 1, lambda = seq(0, 2, by = 0.1)),
  trControl = train_control
)
plot(model2)
```



```
model2$bestTune$lambda
# [1] 1.2
model2$results[model2$results$lambda==model2$bestTune$lambda,]
#    alpha lambda     RMSE  Rsquared      MAE   RMSESD RsquaredSD    MAESD
# 13     1    1.2 55.33944 0.4743875 45.07996 3.610067 0.07401547 3.271096

pred1 <- predict(model1, newdata = x_test)
pred2 <- predict(model2, newdata = x_test)

res1 <- sqrt(mean((pred1 - y_test)^2))
res1
# [1] 52.97107
res2 <- sqrt(mean((pred2 - y_test)^2))
res2
# [1] 53.00034
```

Solutions

This exercise is worth a total of 14 points, each of the following items is worth 2 points

- The data set under consideration consists of $n = 442$ individuals and 10 predictors that have already been normalised.
- The dataset is divided into two parts : a training sample containing 80% of the data and a test sample.
- A first model (model 1), which includes all variables and uses Ridge regularisation, is implemented. The associated regularisation parameter is selected by cross-validation in 5 sets, repeated 100 times, using the RMSE criterion.
- To find the best value for the regularisation parameter, we focused on values between 6 and 10. The best value is 7 and the estimated RMSE is around 55.46.
- A second model (model 2), which includes all variables and uses Lasso regularisation, is implemented. The associated regularisation parameter is selected by cross-validation in 5 sets, repeated 100 times, using the RMSE criterion.
- To find the best value for the regularisation parameter, we focused on values between 0 and 2. The best value is 1.2 and the estimated RMSE is around 55.33.
- As with cross-validation, the results of models 1 and 2 are highly similar when evaluated on a set of test data.

Exercise 2 (6 pts)

Describe and comment on the results obtained using the R code below.

```
n <- 100
x1 <- rnorm(n)
x2 <- rnorm(n)
x3 <- rnorm(n)
u <- rnorm(n)
y <- 1+1.5*x1+2*x2-0.7*x3+u
d <- 80
z <- matrix(runif(n*d),n)

app <- data.frame(y=y,x1=x1,x2=x2,x3=x3,z=z)
names(app)[-(1:4)] <- paste("z",1:80,sep="")

model1 <- lm(y~1,data=app)
model2 <- lm(y~.,data=app)
model3 <- lm(y~x1+x2+x3,data=app)

library(glmnet)
library(caret)
x <- as.matrix(app[,-1])
model4 <- glmnet(x,y,family="gaussian",nlambda=50,alpha=1)
model5 <- train(x,y,method="glmnet",metric="RMSE",
                trControl=trainControl(method="repeatedcv",
                                       number=5,repeats=100),
                tuneGrid=data.frame(alpha=1,lambda=model4$lambda))
```

```
x1test <- rnorm(n)
x2test <- rnorm(n)
x3test <- rnorm(n)
utest <- rnorm(n)
ytest <- 1+1.5*x1test+2*x2test-0.7*x3test+utest
ztest <- matrix(runif(n*d),n)

test <- data.frame(y=ytest,x1=x1test,x2=x2test,x3=x3test,z=ztest)
names(test)[-(1:4)] <- paste("z",1:80,sep="")

mean((predict(model3,test)-ytest)^2)
# 1.000278
mean((predict(model1,test)-ytest)^2)
# 10.37723
mean((predict(model2,test)-ytest)^2)
# 6.840056
mean((predict(model5,test)-ytest)^2)
# 1.217017
```

Solutions

This exercise is worth a total of 6 points, each of the following items is worth one point

- The model is based on the data of $n = 100$ individuals. The true model is $y = 1 + 1.5x_1 + 2x_2 + 0.7x_3 + u$. The model includes three predictors and 80 noise predictors.
- In Model 1, only the intercept is included in the model. In contrast, Model 2 incorporates all the variables, including those that are noise variables.
- Model 3 is the Oracle model that includes the variables pertinent to the model.
- Model 5 incorporates all the variables and employs Lasso regularisation.
- The Lasso regularisation parameter is selected by 5-fold cross-validation, repeated 100 times, using the RMSE criteria.
- The results of the Oracle model and the Lasso model are very similar on a test dataset.