ORF525, Assignment 2, Problem 6

Igor Silin

Data preparation

Let's load the data:

```
train.data <- read.csv("train.data.csv", header=TRUE)
test.data <- read.csv("test.data.csv", header=TRUE)
train.data$zipcode <- as.factor(train.data$zipcode)
test.data$zipcode <- as.factor(test.data$zipcode)

all.data = rbind(train.data, test.data)
all.data[ , c("X", "id", "date")] = list(NULL)
```

Now we extract the train and test response - vectors $Y.train$ and $Y.test$:

```
Y.train = as.numeric(train.data[, "price"])
Y.test = as.numeric(test.data[, "price"])
N.train = length(Y.train)
N.test = length(Y.test)

all.data["price"] = list(NULL)
```

Further, we create the features for settings (a) - (b). They will be stored in matrices $X.train$ and $X.test$. First 17 features are everything except zipcode:

```
X.all = matrix(0L, nrow=N.train+N.test, ncol=23)
X.all[ , 1:17] = as.matrix(all.data[ , colnames(all.data)[colnames(all.data) != "zipcode"]])
```

Next 6 features are interaction terms of "bedrooms", "bathrooms", "sqft_living", "sqft_lot":

```
X.all[ , 18] = all.data[, "bedrooms"] * all.data[, "bathrooms"]
X.all[ , 19] = all.data[, "bedrooms"] * all.data[, "sqft_living"]
X.all[ , 20] = all.data[, "bedrooms"] * all.data[, "sqft_lot"]
X.all[ , 21] = all.data[, "bathrooms"] * all.data[, "sqft_living"]
X.all[ , 22] = all.data[, "bathrooms"] * all.data[, "sqft_lot"]
X.all[ , 23] = 0.001*all.data[, "sqft_living"]*all.data[, "sqft_lot"] # to avoid overflow
```

Zipcode is treated as feature as well. But since it is a categorical variable (has no numerical meaning), we have to use dummy variables to encode all possible zipcodes. It gives 69 columns (there are 70 distinct zipcodes, but one is removed to avoid collinearity):

```
library("fastDummies")
zipcodes = fastDummies::dummy_cols(all.data["zipcode"],
                                   select_columns = "zipcode",
                                   remove_first_dummy = TRUE)
zipcodes["zipcode"] = list(NULL)          # remove zipcode itself
X.all = cbind(X.all, zipcodes)
```

Finally, we create the most sophisticated 11 features:

```
X.all = cbind(X.all, as.numeric(all.data["view"] == 0))
X.all = cbind(X.all, all.data[ , "sqft_living"]^2)

q = quantile(train.data[, "sqft_living"], probs=0.1*(1:9))
```

```
for (i in 1:9) {
  X.all = cbind(X.all, pmax( all.data[, "sqft_living"] - q[i], 0)^2)
}
```

Now, all features are generated. For (a) we will use the first 92 columns, for (b) - all columns. Let's split all back into train matrix and test matrix:

```
X.train = as.matrix(X.all[1:N.train, ])
X.test = as.matrix(X.all[(N.train+1) : (N.train+N.test), ])
```

Now we are ready to fit models based on different features.

```
library("glmnet")
```

```
## Loading required package: Matrix
```

```
## Loading required package: foreach
```

```
## Loaded glmnet 2.0-16
```

```
library("ncvreg")
```

(a)

First, we look at out-of-sample quality of Ridge regression:

```
RIDGEcv <- cv.glmnet(X.train[,1:92], Y.train, alpha = 0, nfolds = 10)
Y.pred = predict(RIDGEcv, newx=X.test[,1:92], s="lambda.min")

R2 = 1 - sum((Y.test - Y.pred)^2)/sum((Y.test - mean(Y.train))^2)
sprintf("(a) Ridge Regression: Out-of-sample R^2 = %f", R2)
```

```
## [1] "(a) Ridge Regression: Out-of-sample R^2 = 0.817970"
```

Then, we apply Lasso:

```
LASSOcv <- cv.glmnet(X.train[,1:92], Y.train, alpha = 1, nfolds = 10)
Y.pred = predict(LASSOcv, newx=X.test[,1:92], s="lambda.min")

R2 = 1 - sum((Y.test - Y.pred)^2)/sum((Y.test - mean(Y.train))^2)
sprintf("(a) LASSO: Out-of-sample R^2 = %f", R2)
```

```
## [1] "(a) LASSO: Out-of-sample R^2 = 0.824919"
```

Finally, we look at the performance of SCAD:

```
SCADcv = cv.ncvreg(X.train[,1:92], Y.train, penalty="SCAD", nfolds=10)
Y.pred = predict(SCADcv, X=X.test[,1:92], lambda=SCADcv$lambda.min)

R2 = 1 - sum((Y.test - Y.pred)^2)/sum((Y.test - mean(Y.train))^2)
sprintf("(a) SCAD: Out-of-sample R^2 = %f", R2)
```

```
## [1] "(a) SCAD: Out-of-sample R^2 = 0.824318"
```

(b)

First, we look at out-of-sample quality of Ridge regression:

```
RIDGEcv <- cv.glmnet(X.train, Y.train, alpha = 0, nfolds = 10)
Y.pred = predict(RIDGEcv, newx=X.test, s="lambda.min")

R2 = 1 - sum((Y.test - Y.pred)^2)/sum((Y.test - mean(Y.train))^2)
sprintf("(b) Ridge Regression: Out-of-sample R^2 = %f", R2)
```

```
## [1] "(b) Ridge Regression: Out-of-sample R^2 = 0.830357"
```

Then, we apply Lasso:

```
LASSOcv <- cv.glmnet(X.train, Y.train, alpha = 1, nfolds = 10)
Y.pred = predict(LASSOcv, newx=X.test, s="lambda.min")

R2 = 1 - sum((Y.test - Y.pred)^2)/sum((Y.test - mean(Y.train))^2)
sprintf("(b) LASSO: Out-of-sample R^2 = %f", R2)
```

```
## [1] "(b) LASSO: Out-of-sample R^2 = 0.829195"
```

Finally, we look at the performance of SCAD:

```
SCADcv = cv.ncvreg(X.train, Y.train, penalty="SCAD", nfolds=10)
Y.pred = predict(SCADcv, X=X.test, lambda=SCADcv$lambda.min)

R2 = 1 - sum((Y.test - Y.pred)^2)/sum((Y.test - mean(Y.train))^2)
sprintf("(b) SCAD: Out-of-sample R^2 = %f", R2)
```

```
## [1] "(b) SCAD: Out-of-sample R^2 = 0.828516"
```

The resuls of these three algorithms are pretty much the same for each setting.