Mathieu Sauterey
UNI: mjs2364

STAT GR5241
HW3

03/19/2018

Pledge: The solutions in this homework represent my own work and I did not copy solutions from the work of other students.

Problem 1 (Ridge Regression and Lasso for Correlated Variables, 10 points)

It is well-known that ridge regression tends to give similar coefficient values to correlated variables, whereas the lasso may give quite different coefficient values to correlated variables. We will now explore this property in a very simple setting.

Suppose that $n = 2, p = 2, x_{11} = x_{12}, x_{21} = x_{22}$. Furthermore, suppose that $y_1 + y_2 = 0$ and $x_{11} + x_{21} = 0$ and $x_{12} + x_{22} = 0$, so that the estimate for the intercept in a least squares, ridge regression, or lasso model is zero: $\widehat{\beta_0} = 0$

Homework Problems.

1. Write out the ridge regression optimization problem in this setting.
2. Argue that in this setting, the ridge coefficient estimates satisfy $\widehat{\beta_1} = \widehat{\beta_2}$.
3. Write out the lasso optimization problem in this setting.
4. Argue that in this setting, the lasso coefficients $\widehat{\beta_1}$ and $\widehat{\beta_2}$ are not unique - in other words, there are many possible solutions to the optimization problem in 3. Describe these solutions.

Response:

1. $n = 2, p = 2$ and $\widehat{\beta_0} = 0$, so the ridge regression coefficient estimates $\widehat{\beta_1}$ and $\widehat{\beta_2}$ are the values that minimize:

$$\min_{\widehat{\beta_1},\widehat{\beta_2}} \sum_{i=1}^{2} (y_i - \beta_1 x_{i1} - \beta_2 x_{i2})^2 + \lambda \sum_{j=1}^{2} \beta_j^2$$

$$\Leftrightarrow \boldsymbol{\min_{\widehat{\beta_1},\widehat{\beta_2}} \left[(y_1 - \beta_1 x_{11} - \beta_2 x_{12})^2 + (y_2 - \beta_1 x_{21} - \beta_2 x_{22})^2 + \lambda(\beta_1^2 + \beta_2^2)\right]}$$

2. To find β_1 and β_2 that minimize the ridge regression above, we must set the derivatives of the equation with respect to β_1 and β_2 both equal to zero:

   Let's start with $\frac{d}{d\beta_1} = 0$:

   $\Leftrightarrow -2x_{11}(y_1 - \beta_1 x_{11} - \beta_2 x_{12})^2 - 2x_{21}(y_2 - \beta_1 x_{21} - \beta_2 x_{22})^2 + 2\lambda\beta_1 = 0$

   Then $\frac{d}{d\beta_2} = 0$:

   $\Leftrightarrow -2x_{12}(y_1 - \beta_1 x_{11} - \beta_2 x_{12})^2 - 2x_{22}(y_2 - \beta_1 x_{21} - \beta_2 x_{22})^2 + 2\lambda\beta_2 = 0$

   Let $\frac{d}{d\beta_1} = \frac{d}{d\beta_2}$:

   $-2x_{11}(y_1 - \beta_1 x_{11} - \beta_2 x_{12})^2 - 2x_{21}(y_2 - \beta_1 x_{21} - \beta_2 x_{22})^2 + 2\lambda\beta_1$
   $= -2x_{12}(y_1 - \beta_1 x_{11} - \beta_2 x_{12})^2 - 2x_{22}(y_2 - \beta_1 x_{21} - \beta_2 x_{22})^2 + 2\lambda\beta_2$

   $\Leftrightarrow -x_{11}(y_1 - \beta_1 x_{11} - \beta_2 x_{12})^2 - x_{21}(y_2 - \beta_1 x_{21} - \beta_2 x_{22})^2 + \lambda\beta_1$
   $= -x_{12}(y_1 - \beta_1 x_{11} - \beta_2 x_{12})^2 - x_{22}(y_2 - \beta_1 x_{21} - \beta_2 x_{22})^2 + \lambda\beta_2$

   We also know that $x_{11} = x_{12}, x_{21} = x_{22}$, so we rewrite the above as follows:

   $-x_{11}(y_1 - \beta_1 x_{11} - \beta_2 x_{12})^2 - x_{22}(y_2 - \beta_1 x_{21} - \beta_2 x_{22})^2 + \lambda\beta_1$
   $= -x_{11}(y_1 - \beta_1 x_{11} - \beta_2 x_{12})^2 - x_{22}(y_2 - \beta_1 x_{21} - \beta_2 x_{22})^2 + \lambda\beta_2$

   $\Leftrightarrow \lambda\beta_1 = \lambda\beta_2$

   $\Leftrightarrow \boldsymbol{\widehat{\beta_1} = \widehat{\beta_2}}$

3. $n = 2, p = 2$ *and* $\widehat{\beta_0} = 0$, so the lasso regression coefficient estimates $\widehat{\beta_1}$ and $\widehat{\beta_2}$ are the values that minimize:

   $min_{\widehat{\beta_1},\widehat{\beta_2}} \sum_{i=1}^{2} (y_i - \beta_1 x_{i1} - \beta_2 x_{i2})^2 + \lambda \sum_{j=1}^{2} |\beta_j|$

   $\Leftrightarrow \boldsymbol{\min_{\widehat{\beta_1},\widehat{\beta_2}} \left[(y_1 - \beta_1 x_{11} - \beta_2 x_{12})^2 + (y_2 - \beta_1 x_{21} - \beta_2 x_{22})^2 + \lambda(|\beta_1| + |\beta_2|)\right]}$

4. To prove that lasso regression coefficients β_1 and β_2 are not unique, we must consider the other formulation of the lasso optimization problem:

   $\min_{\widehat{\beta_1},\widehat{\beta_2}}(y_1 - \beta_1 x_{11} - \beta_2 x_{12})^2 + (y_2 - \beta_1 x_{21} - \beta_2 x_{22})^2$ subject to $|\beta_1| + |\beta_2| \leq s$

   This optimization problem can be visualized as a constraint region shaped as a diamond centered in (0,0) that intersects x and y axis at a distance equal to s.

   Let's rewrite the optimization problem knowing that $x_{11} = x_{12}, x_{21} = x_{22}, y_1 = -y_2$:

   $\min_{\widehat{\beta_1},\widehat{\beta_2}}(y_1 - \beta_1 x_{11} - \beta_2 x_{12})^2 + (y_2 - \beta_1 x_{21} - \beta_2 x_{22})^2$

   $\Leftrightarrow \min_{\widehat{\beta_1},\widehat{\beta_2}}(y_1 - x_{11}(\beta_1 + \beta_2))^2 + (-y_1 - x_{22}(\beta_1 + \beta_2))^2$

   We also know that $x_{12} + x_{22} = 0$ and $x_{11} = x_{12}$, thus $x_{11} + x_{22} = 0$ and $x_{11} = -x_{22}$:

   $\Leftrightarrow \min_{\widehat{\beta_1},\widehat{\beta_2}}(y_1 - x_{11}(\beta_1 + \beta_2))^2 + (-y_1 + x_{11}(\beta_1 + \beta_2))^2$

   $\Leftrightarrow \min_{\widehat{\beta_1},\widehat{\beta_2}}(y_1 - x_{11}(\beta_1 + \beta_2))^2 + ((-1)(y_1 - x_{11}(\beta_1 + \beta_2))^2$

   $\Leftrightarrow \min_{\widehat{\beta_1},\widehat{\beta_2}} 2(y_1 - x_{11}(\beta_1 + \beta_2))^2$

   The above must be greater or equal to zero because the square root function is strictly positive.

   $\Leftrightarrow \min_{\widehat{\beta_1},\widehat{\beta_2}} 2(y_1 - x_{11}(\beta_1 + \beta_2))^2 \geq 0$

   The above is minimized when the left-hand side is equal to 0:

   $\Leftrightarrow 2\left(y_1 - x_{11}(\widehat{\beta_1} + \widehat{\beta_2})\right)^2 = 0$

   $\Leftrightarrow \mathbf{\frac{y_1}{x_{11}} = (\widehat{\beta_1} + \widehat{\beta_2})}$

   Therefore, the contour of the sum of squared error is a straight line. Let's recall that the constraint region is of the form $|\beta_1| + |\beta_2| \leq s$.Thus, when $\beta_1 \geq 0, \beta_2 \geq 0$ or $\beta_1 \leq 0, \beta_2 \leq 0$, the constraint region follows $\beta_1 + \beta_2 \leq s$ or $\beta_1 + \beta_2 \leq -s$ respectively. For these two cases, the contour line will be exactly lying on one edge of the diamond-shaped constraint region and it follows from this observation that there is an infinite number of possible coefficient intersects. **To conclude, these coincident lines have an infinity of intersects and thus the lasso regression coefficient estimates $\widehat{\beta_1}$ and $\widehat{\beta_2}$ are not unique. It happens when $\beta_1 \geq 0, \beta_2 \geq 0$ which yields the constraint $\beta_1 + \beta_2 \leq s$ or when $\beta_1 \leq 0, \beta_2 \leq 0$ which yields the constraint $\beta_1 + \beta_2 \leq -s$.**

Problem 2 (Smoothing Splines, ISL 7.5, 10 points)

Consider two curves $\hat{g}_1$ and $\hat{g}_2$, defined by

$$\hat{g}_1 = \arg\min_g \left(\sum_{i=1}^{n} \left(y_i - g(x_i)\right)^2 + \lambda \int \left[g^{(3)}(x)\right]^2 dx \right)$$

$$\hat{g}_2 = \arg\min_g \left(\sum_{i=1}^{n} \left(y_i - g(x_i)\right)^2 + \lambda \int \left[g^{(4)}(x)\right]^2 dx \right)$$

where $g^{(m)}$ represents the mth derivative of g.

Homework Problems.

1. As $\lambda \to \infty$, will $\hat{g}_1$ or $\hat{g}_2$ have the smaller training RSS, or is there no definite answer? Explain briefly.

2. As $\lambda \to \infty$, will $\hat{g}_1$ or $\hat{g}_2$ have the smaller test RSS, or is there no definite answer? Explain briefly.

3. As $\lambda = 0$, will $\hat{g}_1$ or $\hat{g}_2$ have the smaller training and test RSS, or is there no definite answer? Explain briefly.

Response:

1. We know that for any polynomial of degree d, the d-1th derivative describes the roughness of the function g. The penalty term of $\hat{g}_2$ is one degree higher than $\hat{g}_1$. Therefore, we can expect that $\hat{g}_2$ is a polynomial that has one more degree order than $\hat{g}_1$. **This means that $\hat{g}_2$ is more flexible than $\hat{g}_1$ and so as $\lambda \to \infty$, $\hat{g}_2$ has the smaller training RSS.**

2. We can argue that most of the times $\hat{g}_1$ will have smaller variance and thus smaller test RSS because $\hat{g}_1$ is less flexible than $\hat{g}_2$. Yet, this is not always true as the distribution of the data may be better suited for a higher order polynomial. **So, we conclude that there is no definite answer as we would need more information, however most of the times we would expect $\hat{g}_1$ to have smaller test RSS than $\hat{g}_2$.**

3. As $\lambda = 0$, $\hat{g}_1 = \hat{g}_2 = \arg\min_g \left(\sum_{i=1}^{n} \left(y_i - g(x_i)\right)^2 \right)$ therefore **$\hat{g}_1$ and $\hat{g}_2$ will have the same training and test RSS.**

Problem 3 (SVM, 20 points)

In this problem, we will apply a support vector machine to classify hand-written digits. You do not have to implement the SVM algorithm: The R library e1071 provides an implementation, see

http://cran.r-project.org/web/packages/e1071/index.html

Download the digit data set from the course website. The zip archive contains two files: Both files are text files. Each file contains a matrix with one data point (= vector of length 256) per row. The 256-vector in each row represents a 16x16 image of a handwritten number. The data contains two classes - the digits 5 and 6 - so they can be labeled as -1 and +1, respectively. The image on the right shows the first row, re-arranged as a 16x16 matrix and plotted as a gray scale image.

- Randomly select about 20% of the data and set it aside as a test set.
- Train a linear SVM with soft margin. Cross-validate the margin parameter.
- Train an SVM with soft margin and RBF kernel. You will have to cross-validate both the soft-margin parameter and the kernel bandwidth.
- After you have selected parameter values for both algorithms, train each one with the parameter value you have chosen. Then compute the misclassification rate (the proportion of misclassified data points) on the test set.

Homework Questions.

1. Plot the cross-validation estimates of the misclassification rate. Please plot the rate as
    a) a function of the margin parameter in the linear case.
    b) a function of the margin parameter and the kernel bandwidth in the non-linear case (you are encouraged to use heat map here).

2. Report the test set estimates of the misclassification rates for both cases, with the parameter values you have selected, and compare the two results. Is a linear SVM a good choice for this data, or should we use a non-linear one?

Mathieu Sauterey - UNI: MJS2364

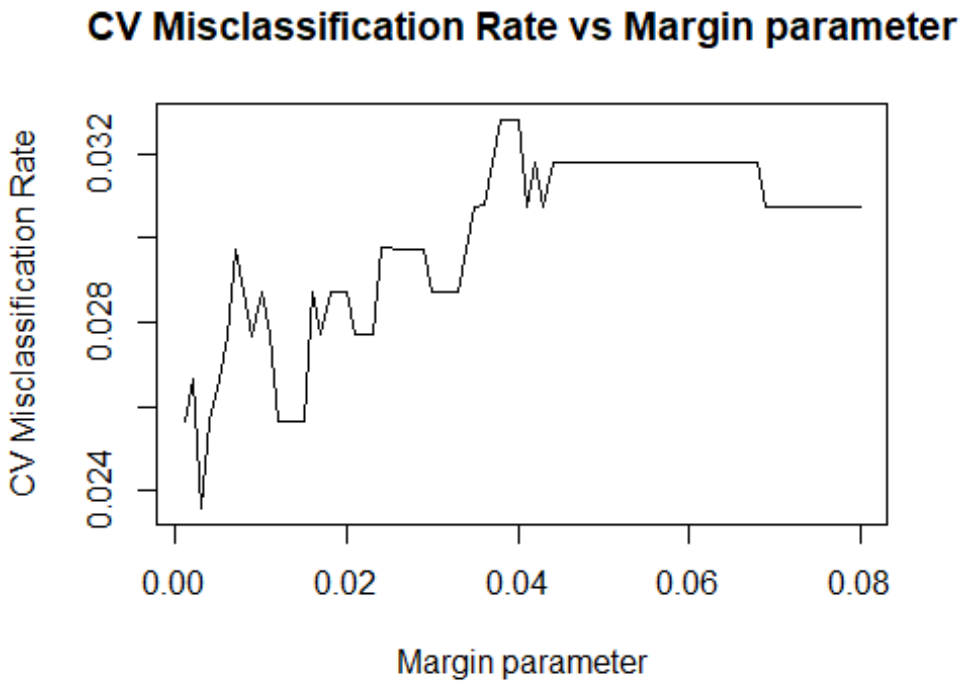
March 21th, 2018

Problem 3 – SVM (20 points)

```
set.seed(1)
library(e1071)

# Loads the data containing the digit "5" and adds the label in a new column
data_5 <- read.table("train.5.txt", as.is = TRUE, sep=",")
data_5 <- cbind(rep(-1,nrow(data_5)),data_5)
colnames(data_5) <- c("Label",paste("X",1:256,sep=""))


# Loads the data containing the digit "6" and adds the label in a new column
data_6 <- read.table("train.6.txt", as.is = TRUE, sep=",")
data_6 <- cbind(rep(1,nrow(data_6)),data_6)
colnames(data_6) <- c("Label",paste("X",1:256,sep=""))


# Combines the two digits tables into a training dataframe
data_set <- rbind.data.frame(data_5, data_6)


# Randomly select about 20% of the data and set it aside as a test set
index_test <- sample(x=1:nrow(data_set), size = 0.2*nrow(data_set))
data_test <- data_set[index_test,]
data_train <- data_set[-index_test,]


## LINEAR SOFT MARGIN SVM

# Exclude label column and columns with constant value
nocst <- c(1,2,17)

# Tune the cost parameter of the soft margin SVM (linear kernel)
# "tune.svm" uses 10-fold cross validation
tune.soft <- tune.svm(kernel="linear", x=data_train[,-nocst],
                      y=factor(data_train$Label),
                      cost = seq(0.001,0.08, by = 0.001))

# Tuned margin parameter
soft_cost <- tune.soft$best.model$cost
soft_cost
```

```
## [1] 0.003
```

```r
# Table of error rates for different parameter values
soft_perfo <- tune.soft$performances

# Plots the CV estimates of the misclassification rate vs margin parameter
plot(soft_perfo$cost, soft_perfo$error, xlab = "Margin parameter",
     ylab = "CV Misclassification Rate", type="l",
     main = "CV Misclassification Rate vs Margin parameter")
```

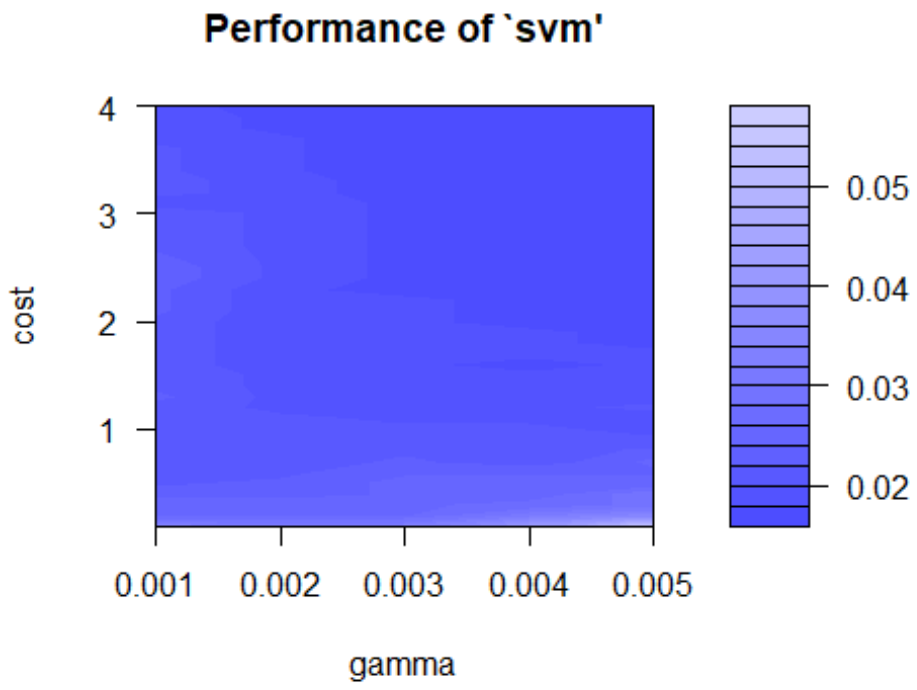



```r
# Train an SVM with soft margin (linear kernel)
soft_svm <- svm(x=data_train[,-nocst],y=factor(data_train$Label),
                kernel="linear", cost=soft_cost)

# Compute the misclassification rate on the test set for the soft margin.
test_pred_soft <- predict(soft_svm, data_test[,-nocst])
soft_rate <- mean(test_pred_soft != data_test$Label)
print(paste(soft_rate*100, "%"))
```

```
## [1] "1.63934426229508 %"
```

The cross-validation estimate of misclassification rate in the case of the soft margin SVM on the test set is **1.64%.**

```r
set.seed(1)
## RBF KERNEL SVM

# Tune the cost and gamma parameters of the RBF kernel SVM
tune.rbf <- tune.svm(kernel="radial", x=data_train[,-nocst],
                     y=factor(data_train$Label), cost = seq(0.1,4, by = 0.1),
                     gamma = seq(0.001,0.005, by = 0.001))

# Heatmap of CV misclassification rate vs kernel bandwidth, margin parameter
plot(tune.rbf)
```




```r
# Tuned margin parameter
rbf_cost <- tune.rbf$best.model$cost
rbf_cost

## [1] 4

# Tuned bandwidth parameter
rbf_gamma <- tune.rbf$best.model$gamma
rbf_gamma

## [1] 0.002

# Train an SVM with RBF kernel)
rbf_svm <- svm(x=data_train[,-nocst],y=factor(data_train$Label),
               kernel="radial", cost=rbf_cost, gamma=rbf_gamma)
```

```r
# Compute the misclassification rate on the test set for the rbf kernel SVM
test_pred_rbf <- predict(rbf_svm, data_test[,-nocst])
rbf_rate <- mean(test_pred_rbf != data_test$Label)
print(paste(rbf_rate*100, "%"))

## [1] "0.819672131147541 %"
```

The cross-validation estimate of misclassification rate in the case of the RBF kernel SVM on the test set is **0.82%.**

Therefore, we conclude that **the non-linear SVM with RBF kernel should be used** instead of the linear soft margin SVM, because the non-linear SVM has a lower test misclassification rate.