

Homework 5

Na Yun Cho

```
library(tidyverse)

## -- Attaching packages ----- tidyverse 1.3.0 --

## v ggplot2 3.3.3      v purrr  0.3.4
## v tibble  3.0.6      v dplyr  1.0.4
## v tidyr   1.1.2      v stringr 1.4.0
## v readr   1.4.0      v forcats 0.5.1

## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()

library(mlbench)
library(ISLR)
library(caret)

## Loading required package: lattice

##
## Attaching package: 'caret'

## The following object is masked from 'package:purrr':
##
##      lift

library(e1071)
library(kernlab)

##
## Attaching package: 'kernlab'

## The following object is masked from 'package:purrr':
##
##      cross

## The following object is masked from 'package:ggplot2':
##
##      alpha
```

```
library(DALEX)
```

```
## Welcome to DALEX (version: 2.2.0).  
## Find examples and detailed introduction at: http://ema.drwhy.ai/
```

```
##  
## Attaching package: 'DALEX'
```

```
## The following object is masked from 'package:dplyr':  
##  
## explain
```

```
data(OJ)  
# partition the data into training set and test set  
oj = OJ %>% mutate(Store7 = recode(Store7, '1' = 'Yes', '0' = 'No'), Store7 = as.numeric(Store7))  
set.seed(1)  
rowTrain <- createDataPartition(y = oj$Purchase, p = 799/1070, list = FALSE)  
  
train_df = oj[rowTrain,]  
test_df = oj[-rowTrain,]  
  
dim(train_df) #has 800 observations
```

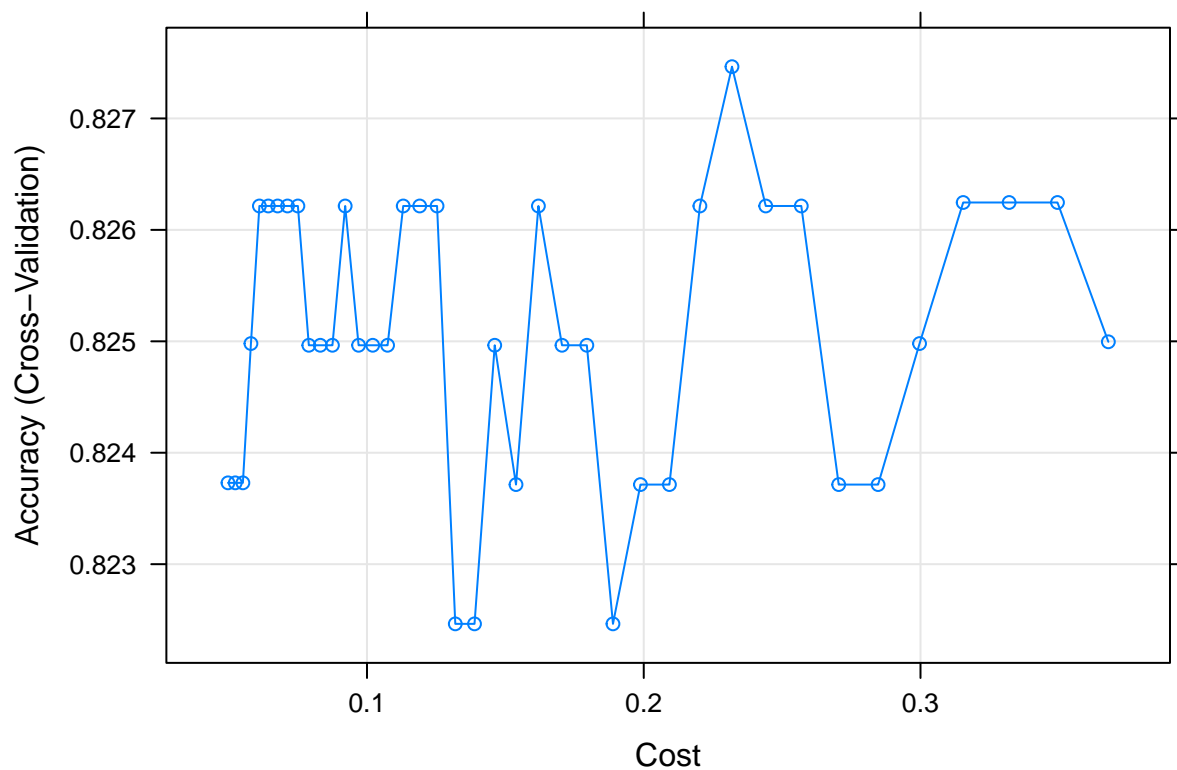
```
## [1] 800 18
```

```
dim(test_df) #has 270 observations
```

```
## [1] 270 18
```

(a)

```
ctrl <- trainControl(method = "cv")  
  
set.seed(1)  
svml.fit <- train(Purchase ~ .,  
                  data = train_df,  
                  method = "svmLinear2",  
                  preProcess = c("center", "scale"),  
                  tuneGrid = data.frame(cost = exp(seq(-3, -1, len = 40))),  
                  trControl = ctrl)  
  
plot(svml.fit, highlight = TRUE)
```



```
#training error rate
mean(predict(svm1.fit) != train_df$Purchase)
```

```
## [1] 0.16875
```

```
#test error rate
mean(predict(svm1.fit, newdata = test_df) != test_df$Purchase)
```

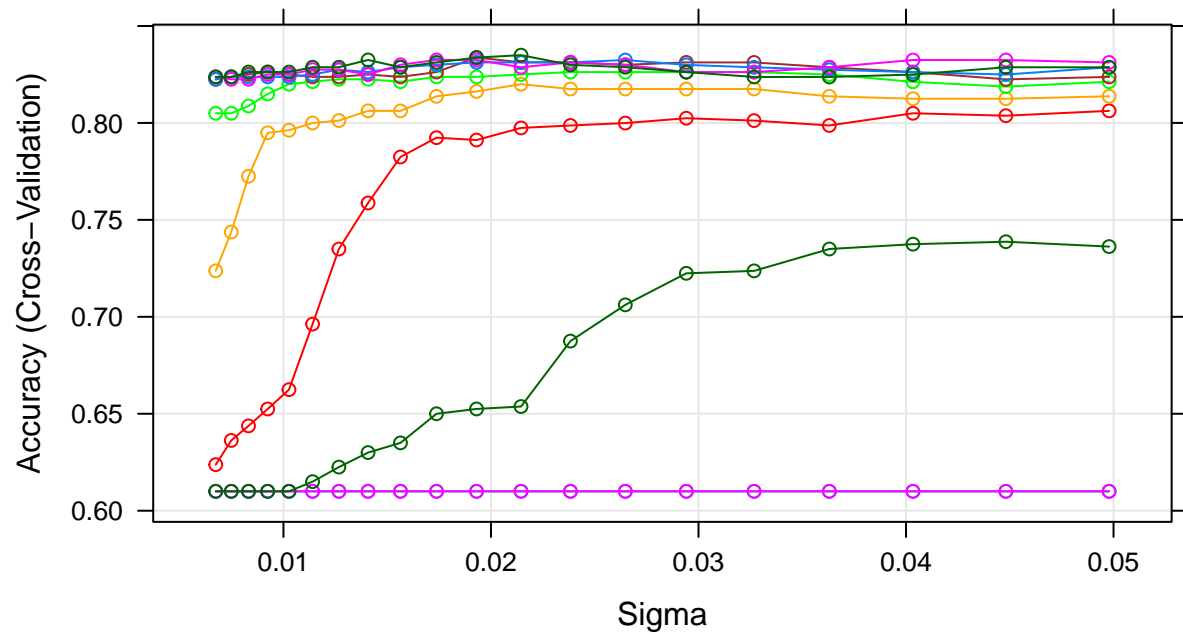
```
## [1] 0.1518519
```

(b)

```
svmr.grid <- expand.grid(C = exp(seq(-4,0,len=10)),
                        sigma = exp(seq(-5,-3,len=20)))
set.seed(1)
svmr.fit <- train(Purchase~.,
                  data = train_df,
                  method = "svmRadialSigma",
                  preProcess = c("center", "scale"),
                  tuneGrid = svmr.grid,
                  trControl = ctrl)

plot(svmr.fit, highlight = TRUE)
```

		Cost	
12	○ —	0.0694834512228015	○ — 0.263597138115727
14	○ —	0.108368023221896	○ — 0.411112290507187
17	○ —	0.169013315406066	○ — 0.641180388429955



```
#training error rate
mean(predict(svmr.fit) != train_df$Purchase)
```

```
## [1] 0.15625
```

```
#test error rate
mean(predict(svmr.fit, newdata = test_df) != test_df$Purchase)
```

```
## [1] 0.1555556
```