Homework 5

Ngoc Duong

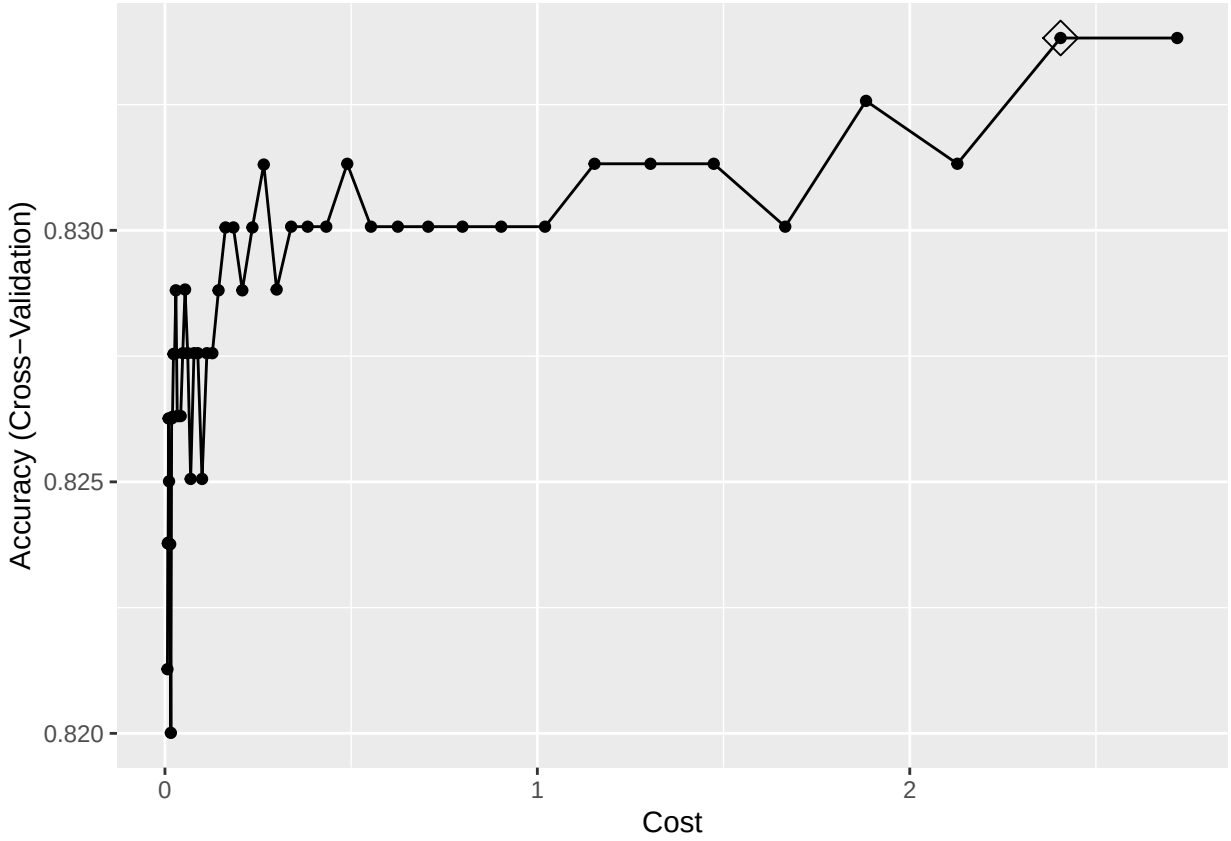
5/4/2020

Import data OJ from the ISLR package

```
data(OJ)

#create training set with a random sample of 800 observations
set.seed(13)
rowTrain <-createDataPartition(y = OJ$Purchase,
                               p = 799/1070,
                               list = FALSE)
oj_train = OJ[rowTrain,]
oj_test = OJ[-rowTrain,]
```

a) Fit a support vector classifier (linear kernel) to the training data with Purchase as the response

```
ctrl <-trainControl(method = "cv")

set.seed(13)
svml.fit <-train(Purchase~.,data = oj_train,
                 method = "svmLinear2",
                 preProcess =c("center", "scale"),
                 tuneGrid =data.frame(cost =exp(seq(-5,1,len=50))),
                 trControl = ctrl)

ggplot(svml.fit, highlight = TRUE)
```



Find the training and test error rate

```
pred.svml.test <-predict(svml.fit, newdata = oj_test)
confusionMatrix(data = pred.svml.test, reference = oj_test$Purchase)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  CH  MM
##         CH 150  32
##         MM  15  73
##
##                Accuracy : 0.8259
##                  95% CI : (0.7753, 0.8692)
##     No Information Rate : 0.6111
##     P-Value [Acc > NIR] : 1.626e-14
##
##                   Kappa : 0.6227
##
##  Mcnemar's Test P-Value : 0.0196
##
##             Sensitivity : 0.9091
##             Specificity : 0.6952
##          Pos Pred Value : 0.8242
##          Neg Pred Value : 0.8295
##              Prevalence : 0.6111
##          Detection Rate : 0.5556
##    Detection Prevalence : 0.6741
```

```
##       Balanced Accuracy : 0.8022
##
##        'Positive' Class : CH
##
```

Test error rate: $(15+22)/270 = 0.137$

```
pred.svml.train <-predict(svml.fit, newdata = oj_train)
confusionMatrix(data = pred.svml.train, reference = oj_train$Purchase)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  CH  MM
##         CH 426  68
##         MM  62 244
##
##                Accuracy : 0.8375
##                  95% CI : (0.8101, 0.8624)
##     No Information Rate : 0.61
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.6573
##
##  Mcnemar's Test P-Value : 0.661
##
##             Sensitivity : 0.8730
##             Specificity : 0.7821
##          Pos Pred Value : 0.8623
##          Neg Pred Value : 0.7974
##              Prevalence : 0.6100
##          Detection Rate : 0.5325
##    Detection Prevalence : 0.6175
##       Balanced Accuracy : 0.8275
##
##        'Positive' Class : CH
##
```

Training error rate: $(59+77)/800 = 0.17$

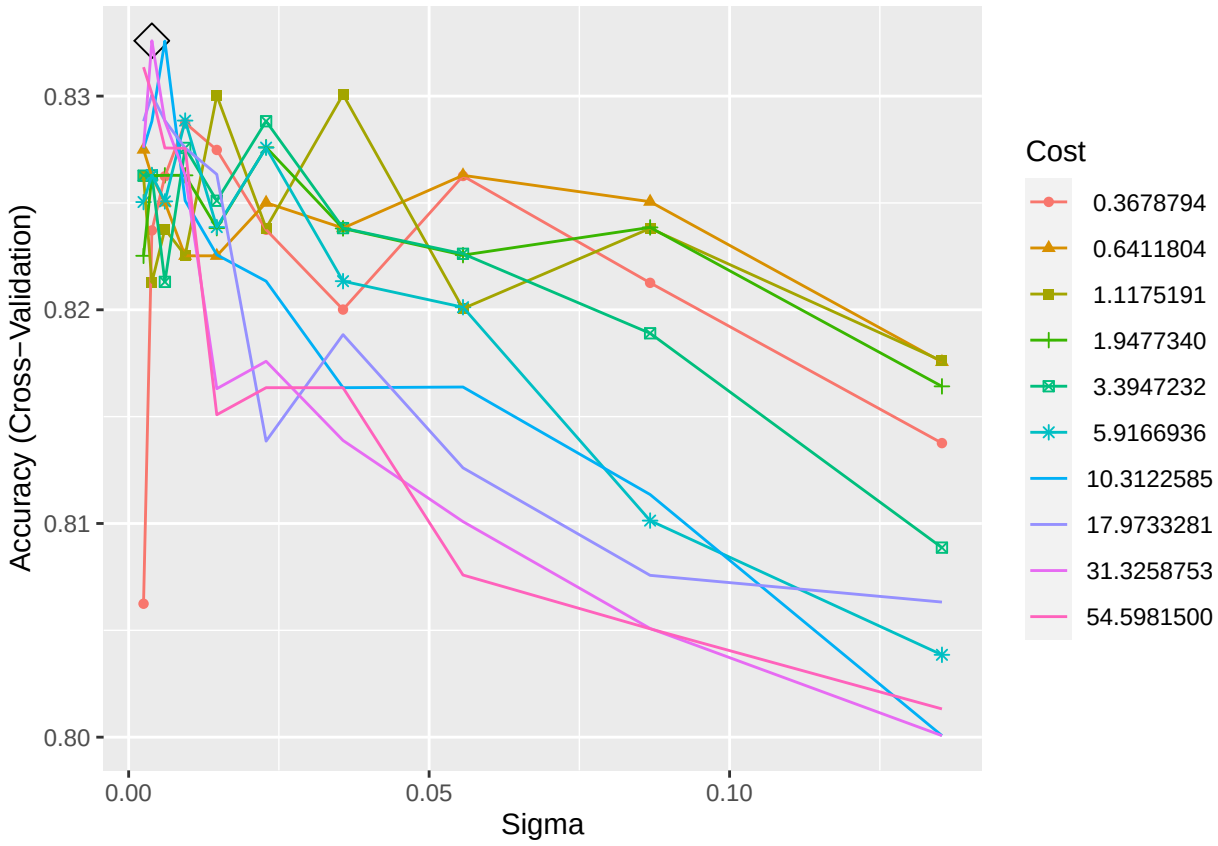
b) Fit a support vector machine (radial kernel) to the training data with Purchase as the response

```
svmr.grid <-expand.grid(C =exp(seq(-1,4,len=10)),
                        sigma =exp(seq(-6,-2,len=10)))

set.seed(13)
svmr.fit <-train(Purchase~.,data = oj_train,
                 method = "svmRadial",
                 preProcess =c("center", "scale"),
                 tuneGrid = svmr.grid,
                 trControl = ctrl)

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

```
## Warning: The shape palette can deal with a maximum of 6 discrete values because
```

```
## more than 6 becomes difficult to discriminate; you have 10. Consider
## specifying shapes manually if you must have them.
```

```
## Warning: Removed 40 rows containing missing values (geom_point).
```



Find the training and test error rate

```
pred.svmr.test <-predict(svmr.fit, newdata = oj_test)
confusionMatrix(data = pred.svmr.test, reference = oj_test$Purchase)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  CH  MM
##         CH 149  32
##         MM  16  73
##
##                Accuracy : 0.8222
##                  95% CI : (0.7713, 0.8659)
##     No Information Rate : 0.6111
##     P-Value [Acc > NIR] : 4.866e-14
##
##                   Kappa : 0.6153
##
##  Mcnemar's Test P-Value : 0.03038
##
##             Sensitivity : 0.9030
##             Specificity : 0.6952
##          Pos Pred Value : 0.8232
```

```
##          Neg Pred Value : 0.8202
##              Prevalence : 0.6111
##          Detection Rate : 0.5519
##    Detection Prevalence : 0.6704
##       Balanced Accuracy : 0.7991
##
##        'Positive' Class : CH
##
```

Test error rate: $(15+22)/270 = 0.137$

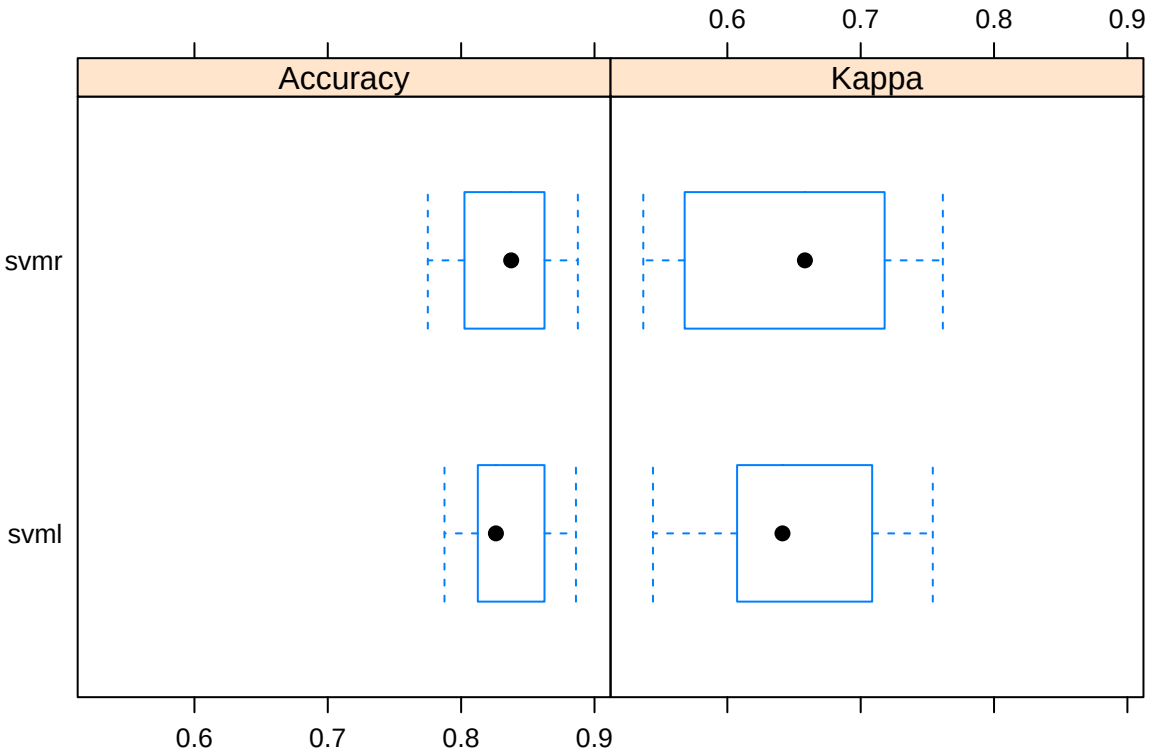
```
pred.svmr.train <-predict(svmr.fit, newdata = oj_train)
confusionMatrix(data = pred.svmr.train, reference = oj_train$Purchase)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  CH  MM
##         CH 432  67
##         MM  56 245
##
##                Accuracy : 0.8462
##                  95% CI : (0.8194, 0.8706)
##     No Information Rate : 0.61
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.6748
##
##  Mcnemar's Test P-Value : 0.3672
##
##             Sensitivity : 0.8852
##             Specificity : 0.7853
##          Pos Pred Value : 0.8657
##          Neg Pred Value : 0.8140
##              Prevalence : 0.6100
##          Detection Rate : 0.5400
##    Detection Prevalence : 0.6238
##       Balanced Accuracy : 0.8353
##
##        'Positive' Class : CH
##
```

Training error rate: $(53+78)/800 = 0.163$

Compare from the two

```
resamp <-resamples(list(svmr = svmr.fit, svml = svml.fit))
bwplot(resamp)
```



The two models' classification performance are very simiilar. However, support vector machines have only marginally better accuracy and kappa statistic than the support vector classifier.