DS2_HW5

Siyan Chen

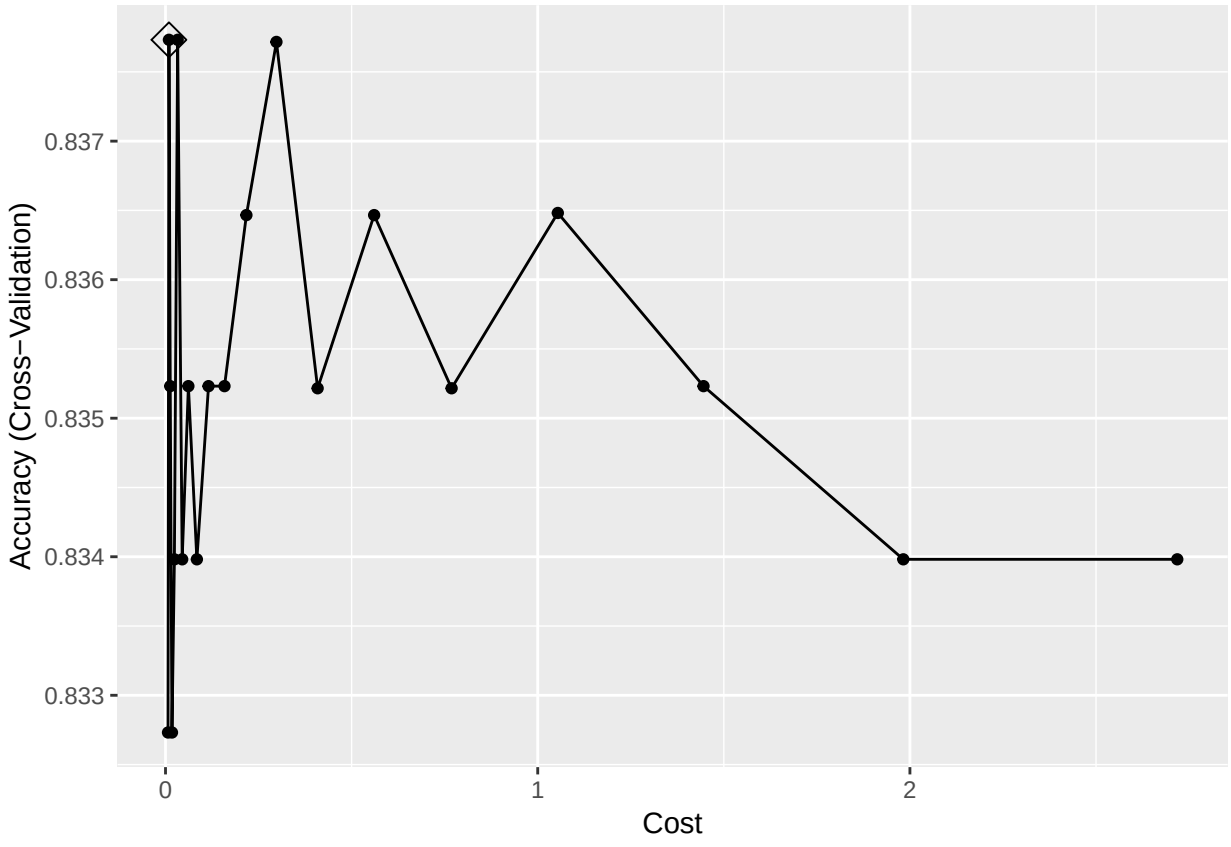
4/24/2019

(a) Fit a support vector classifier (linear kernel) to the training data with Purchase as the response and the other variables as predictors. What are the training and test error rates?

```
set.seed(1)
rowtrain = createDataPartition(y = OJ$Purchase,
                               p = 800/1070,
                               list = FALSE)

ctr1 = trainControl(method = "cv")

set.seed(1)
svml.fit = train(Purchase~.,
                 data = OJ[rowtrain,],
                 method = "svmLinear2",
                 preProcess = c("center", "scale"),
                 tuneGrid = data.frame(cost = exp(seq(-5,1,len = 20))),
                 trControl = ctr1)
ggplot(svml.fit, highlight = TRUE)
```



```
best.svml = svml.fit$finalModel
summary(best.svml)
```

```
##
## Call:
## svm.default(x = as.matrix(x), y = y, kernel = "linear", cost = param$cost,
##     probability = classProbs)
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  linear
##        cost:  0.009240026
##       gamma:  0.05882353
##
## Number of Support Vectors:  442
##
##  ( 220 222 )
##
##
## Number of Classes:  2
##
## Levels:
##  CH MM
```

```
### train error
pred.svml_train = predict(svml.fit, newdata = OJ[rowtrain,])
1-confusionMatrix(data = pred.svml_train, reference = OJ$Purchase[rowtrain])$overall[["Accuracy"]]
```

```
## [1] 0.1560549
```

```
### test error
pred.svml = predict(svml.fit, newdata = OJ[-rowtrain,])
mean(pred.svml != OJ[-rowtrain,]$Purchase)
```

```
## [1] 0.1858736
```

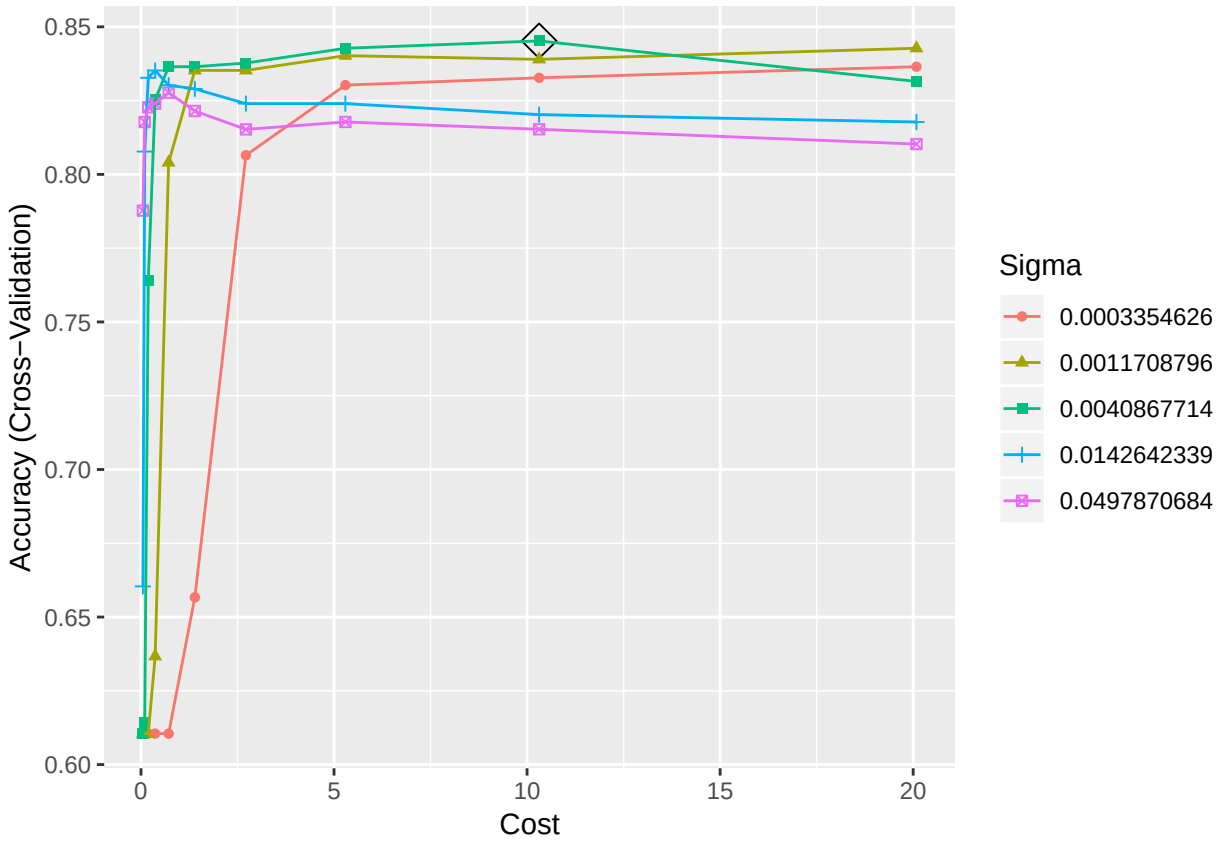
```
# OR 1-confusionMatrix(data = pred.svml, reference = OJ$Purchase[-rowtrain])$overall[["Accuracy"]]
```

The trainning error rate is 0.1560549 and test error rate is 0.1858736.

(b) Fit a support vector machine with a radial kernel to the training data. What are the training and test error rates?

```
svmr.grid = expand.grid(C = exp(seq(-3, 3, len = 10)),
                        sigma = exp(seq(-8, -3, len = 5)))
set.seed(1)
svmr.fit = train(Purchase~.,
                 data = OJ[rowtrain,],
                 method = "svmRadial",
                 preProcess = c("center", "scale"),
                 tuneGrid = svmr.grid,
                 trControl = ctr1)

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




```
### train error
pred.svmr_train = predict(svmr.fit, newdata = OJ[rowtrain,])
1-confusionMatrix(data = pred.svmr_train, reference = OJ$Purchase[rowtrain])$overall[["Accuracy"]]
```

```
## [1] 0.1510612
```

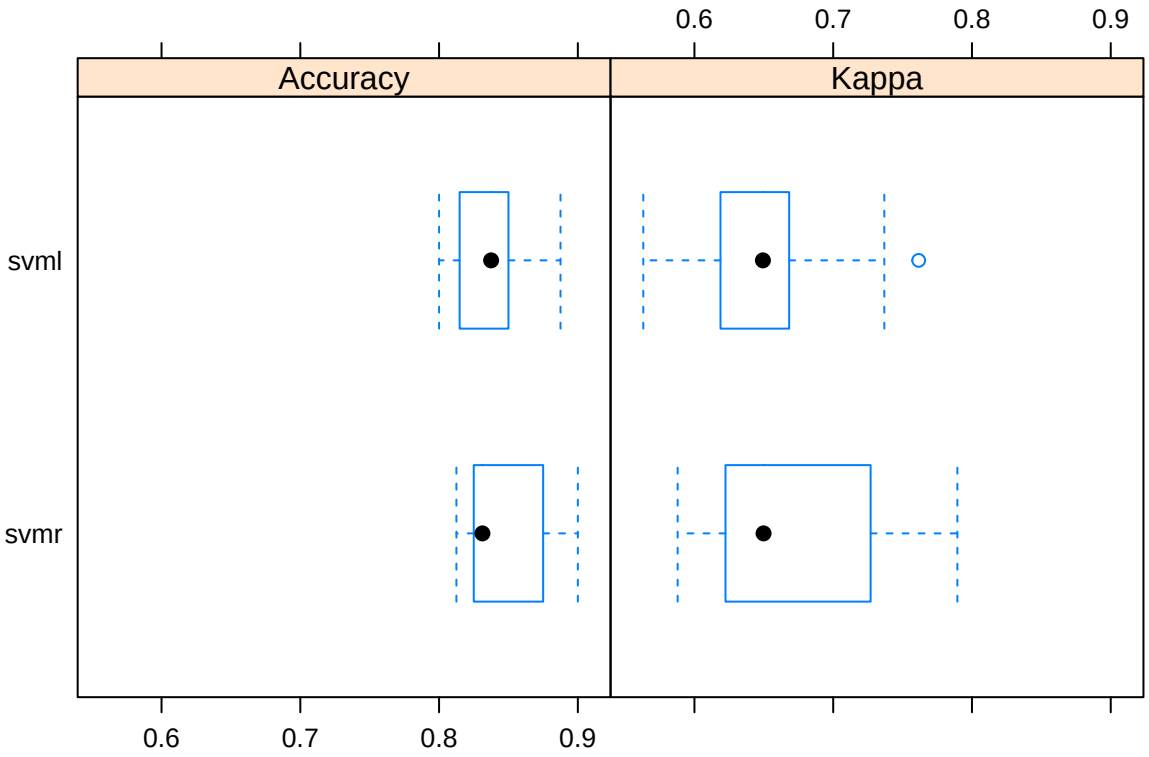
```
### test error rate
pred.svmr = predict(svmr.fit, newdata = OJ[-rowtrain,])
mean(pred.svmr != OJ[-rowtrain,]$Purchase)
```

```
## [1] 0.2007435
```

Train error rate is 0.1510612 and test error rate is 0.2007435.

(c) Which approach seems to give a better result on this data?

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




```
summary(resamp)
```

```
##
## Call:
## summary.resamples(object = resamp)
##
## Models: svml, svmr
## Number of resamples: 10
##
## Accuracy
##        Min.   1st Qu.  Median      Mean  3rd Qu.   Max. NA's
## svml 0.8000 0.8173611 0.83750 0.8377315 0.846875 0.8875    0
## svmr 0.8125 0.8250000 0.83125 0.8452315 0.868750 0.9000    0
##
## Kappa
##           Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## svml 0.5631399 0.6208016 0.6493465 0.6541754 0.6651167 0.7615894    0
## svmr 0.5879121 0.6246244 0.6498108 0.6707525 0.7133106 0.7894737    0
```

According the boxplot, both accuracy and kappa of the model with radial kernel is greater which suggests that it has smaller train error rate and greater inter-rater aggrement. Therefore, model with radial kernel is better.