

# Homework 5

*Ziyi Zhao*

*5/9/2020*

## Import data OJ

### linear boundary

```
##
## Parameter tuning of 'svm':
##
## - sampling method: 10-fold cross validation
##
## - best parameters:
##      cost
## 6.405409
##
## - best performance: 0.16625
##
## - Detailed performance results:
##      cost  error dispersion
## 1  0.1353353 0.17000 0.03395258
## 2  0.1561180 0.17250 0.03216710
## 3  0.1800923 0.17125 0.03283481
## 4  0.2077482 0.17125 0.03283481
## 5  0.2396510 0.17000 0.03446012
## 6  0.2764530 0.17125 0.03488573
## 7  0.3189066 0.17375 0.03143004
## 8  0.3678794 0.17000 0.03184162
## 9  0.4243728 0.17250 0.03216710
## 10 0.4895417 0.17000 0.03291403
## 11 0.5647181 0.17125 0.03586723
## 12 0.6514391 0.17000 0.03593976
## 13 0.7514773 0.17000 0.03689324
## 14 0.8668779 0.17125 0.03537988
## 15 1.0000000 0.17000 0.03736085
## 16 1.1535650 0.16875 0.03738408
## 17 1.3307122 0.17125 0.03910900
## 18 1.5350630 0.17125 0.03910900
## 19 1.7707950 0.17125 0.03910900
## 20 2.0427271 0.17125 0.03910900
## 21 2.3564184 0.17000 0.03736085
## 22 2.7182818 0.17000 0.03736085
## 23 3.1357148 0.17000 0.03827895
## 24 3.6172508 0.17000 0.03641962
## 25 4.1727339 0.17125 0.03634805
## 26 4.8135197 0.17125 0.03438447
## 27 5.5527079 0.16750 0.03343734
## 28 6.4054094 0.16625 0.03335936
```

```

## 29 7.3890561 0.16625 0.03283481
## 30 8.5237565 0.16875 0.03131937
## 31 9.8327071 0.16875 0.03131937
## 32 11.3426667 0.16875 0.03131937
## 33 13.0845032 0.16750 0.03129164
## 34 15.0938249 0.16875 0.03186887
## 35 17.4117081 0.17000 0.03016160
## 36 20.0855369 0.17000 0.03016160
## 37 23.1699723 0.17125 0.03064696
## 38 26.7280690 0.17125 0.03064696
## 39 30.8325648 0.17000 0.02898755
## 40 35.5673674 0.17375 0.03251602
## 41 41.0292700 0.17375 0.03251602
## 42 47.3299296 0.17500 0.03173239
## 43 54.5981500 0.17500 0.03173239
## 44 62.9825147 0.17375 0.03030516
## 45 72.6544242 0.17375 0.03030516
## 46 83.8116005 0.17250 0.03106892
## 47 96.6821285 0.17250 0.03106892
## 48 111.5291191 0.17000 0.03496029
## 49 128.6560877 0.17000 0.03496029
## 50 148.4131591 0.17000 0.03291403

##
## Call:
## best.svm(x = purchase ~ ., data = jojo[jo_train, ], cost = exp(seq(-2,
## 5, len = 50)), kernel = "linear")
##
##
## Parameters:
## SVM-Type: C-classification
## SVM-Kernel: linear
## cost: 6.405409
##
## Number of Support Vectors: 315
##
## ( 158 157 )
##
##
## Number of Classes: 2
##
## Levels:
## CH MM

## Confusion Matrix and Statistics
##
## Reference
## Prediction CH MM
## CH 146 27
## MM 19 78
##
## Accuracy : 0.8296
## 95% CI : (0.7794, 0.8725)
## No Information Rate : 0.6111

```

```

##      P-Value [Acc > NIR] : 5.295e-15
##
##              Kappa : 0.6365
##
## Mcnemar's Test P-Value : 0.302
##
##      Sensitivity : 0.8848
##      Specificity : 0.7429
##      Pos Pred Value : 0.8439
##      Neg Pred Value : 0.8041
##      Prevalence : 0.6111
##      Detection Rate : 0.5407
##      Detection Prevalence : 0.6407
##      Balanced Accuracy : 0.8139
##
##      'Positive' Class : CH
##

```

The training error rate using linear boundary is 0.18375.

The test error rate is 0.1407.

## radical kernel

```

##
## Parameter tuning of 'svm':
##
## - sampling method: 10-fold cross validation
##
## - best parameters:
##      gamma      cost
## 0.011109 0.7165313
##
## - best performance: 0.16125
##
## - Detailed performance results:
##      gamma      cost  error dispersion
## 1 0.000911882 0.006737947 0.39000 0.05706965
## 2 0.003182781 0.006737947 0.39000 0.05706965
## 3 0.011108997 0.006737947 0.39000 0.05706965
## 4 0.038774208 0.006737947 0.39000 0.05706965
## 5 0.135335283 0.006737947 0.39000 0.05706965
## 6 0.000911882 0.014666017 0.39000 0.05706965
## 7 0.003182781 0.014666017 0.39000 0.05706965
## 8 0.011108997 0.014666017 0.39000 0.05706965
## 9 0.038774208 0.014666017 0.39000 0.05706965
## 10 0.135335283 0.014666017 0.39000 0.05706965
## 11 0.000911882 0.031922492 0.39000 0.05706965
## 12 0.003182781 0.031922492 0.39000 0.05706965
## 13 0.011108997 0.031922492 0.38875 0.06022239
## 14 0.038774208 0.031922492 0.31625 0.06998264
## 15 0.135335283 0.031922492 0.39000 0.05706965
## 16 0.000911882 0.069483451 0.39000 0.05706965

```

```

## 17 0.003182781 0.069483451 0.39000 0.05706965
## 18 0.011108997 0.069483451 0.25250 0.07163759
## 19 0.038774208 0.069483451 0.18500 0.04632314
## 20 0.135335283 0.069483451 0.21500 0.05062114
## 21 0.000911882 0.151239760 0.39000 0.05706965
## 22 0.003182781 0.151239760 0.30750 0.05177408
## 23 0.011108997 0.151239760 0.18375 0.04489571
## 24 0.038774208 0.151239760 0.16750 0.03827895
## 25 0.135335283 0.151239760 0.19250 0.03545341
## 26 0.000911882 0.329192988 0.37625 0.05573063
## 27 0.003182781 0.329192988 0.19250 0.04495368
## 28 0.011108997 0.329192988 0.16750 0.03736085
## 29 0.038774208 0.329192988 0.16250 0.03280837
## 30 0.135335283 0.329192988 0.18125 0.03346329
## 31 0.000911882 0.716531311 0.24250 0.06800735
## 32 0.003182781 0.716531311 0.16750 0.03827895
## 33 0.011108997 0.716531311 0.16125 0.03251602
## 34 0.038774208 0.716531311 0.16250 0.02763854
## 35 0.135335283 0.716531311 0.17625 0.02531057
## 36 0.000911882 1.559623498 0.18125 0.03784563
## 37 0.003182781 1.559623498 0.17000 0.03073181
## 38 0.011108997 1.559623498 0.16750 0.03016160
## 39 0.038774208 1.559623498 0.16500 0.02687419
## 40 0.135335283 1.559623498 0.18125 0.02901748
## 41 0.000911882 3.394723187 0.17125 0.03682259
## 42 0.003182781 3.394723187 0.17125 0.03682259
## 43 0.011108997 3.394723187 0.16500 0.02415229
## 44 0.038774208 3.394723187 0.16875 0.02585349
## 45 0.135335283 3.394723187 0.19250 0.03073181
## 46 0.000911882 7.389056099 0.17125 0.03335936
## 47 0.003182781 7.389056099 0.17125 0.03335936
## 48 0.011108997 7.389056099 0.16250 0.03061862
## 49 0.038774208 7.389056099 0.17625 0.02461509
## 50 0.135335283 7.389056099 0.18875 0.02598744

##
## Call:
## best.svm(x = purchase ~ ., data = jojo[jo_train, ], gamma = exp(seq(-7,
##      -2, len = 5)), cost = exp(seq(-5, 2, len = 10)))
##
##
## Parameters:
##   SVM-Type:  C-classification
##   SVM-Kernel: radial
##      cost:  0.7165313
##
## Number of Support Vectors:  410
##
## ( 206 204 )
##
##
## Number of Classes:  2
##
## Levels:

```

```

## CH MM

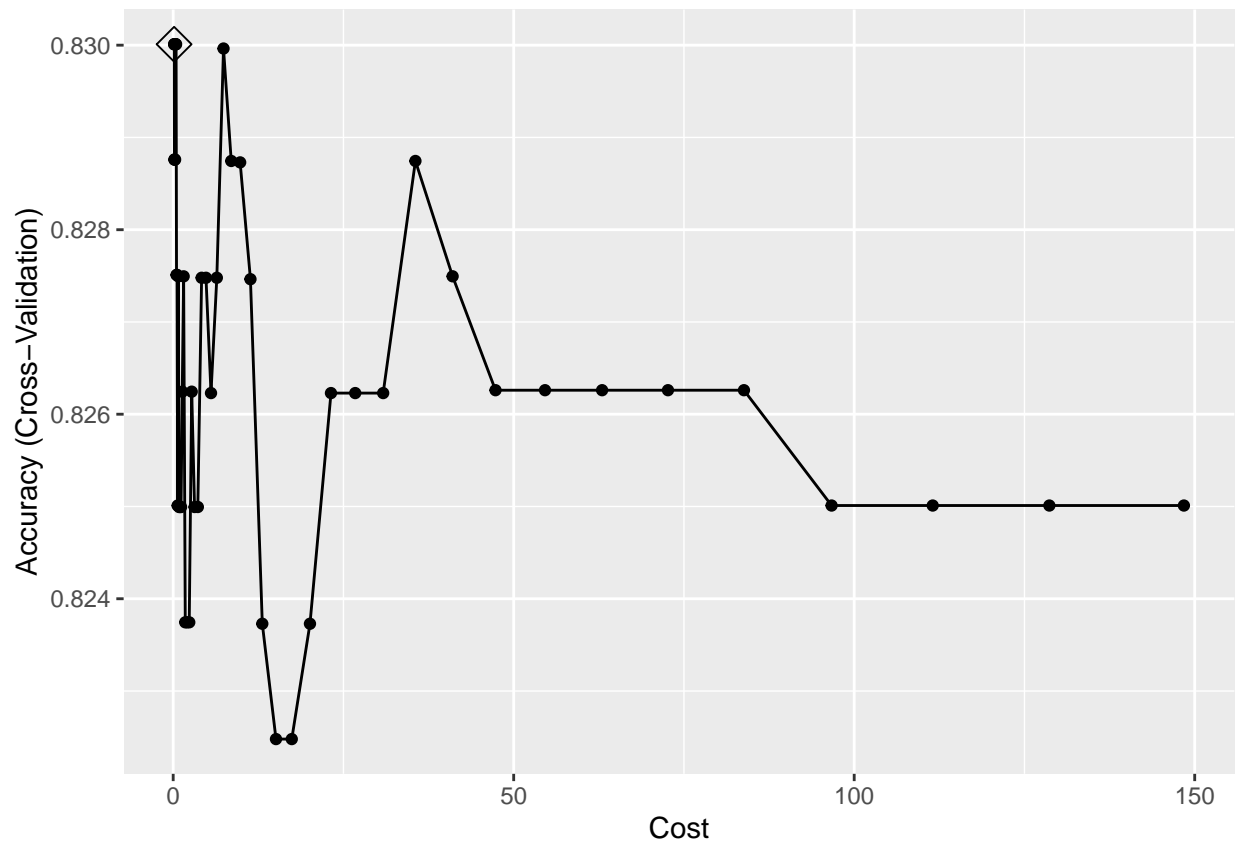
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  CH  MM
##           CH 143  29
##           MM  22  76
##
##           Accuracy : 0.8111
##           95% CI : (0.7592, 0.856)
##           No Information Rate : 0.6111
##           P-Value [Acc > NIR] : 1.126e-12
##
##           Kappa : 0.5977
##
## Mcnemar's Test P-Value : 0.4008
##
##           Sensitivity : 0.8667
##           Specificity : 0.7238
##           Pos Pred Value : 0.8314
##           Neg Pred Value : 0.7755
##           Prevalence : 0.6111
##           Detection Rate : 0.5296
##           Detection Prevalence : 0.6370
##           Balanced Accuracy : 0.7952
##
##           'Positive' Class : CH
##

```

The training error rate using radical kernel is 0.1775.

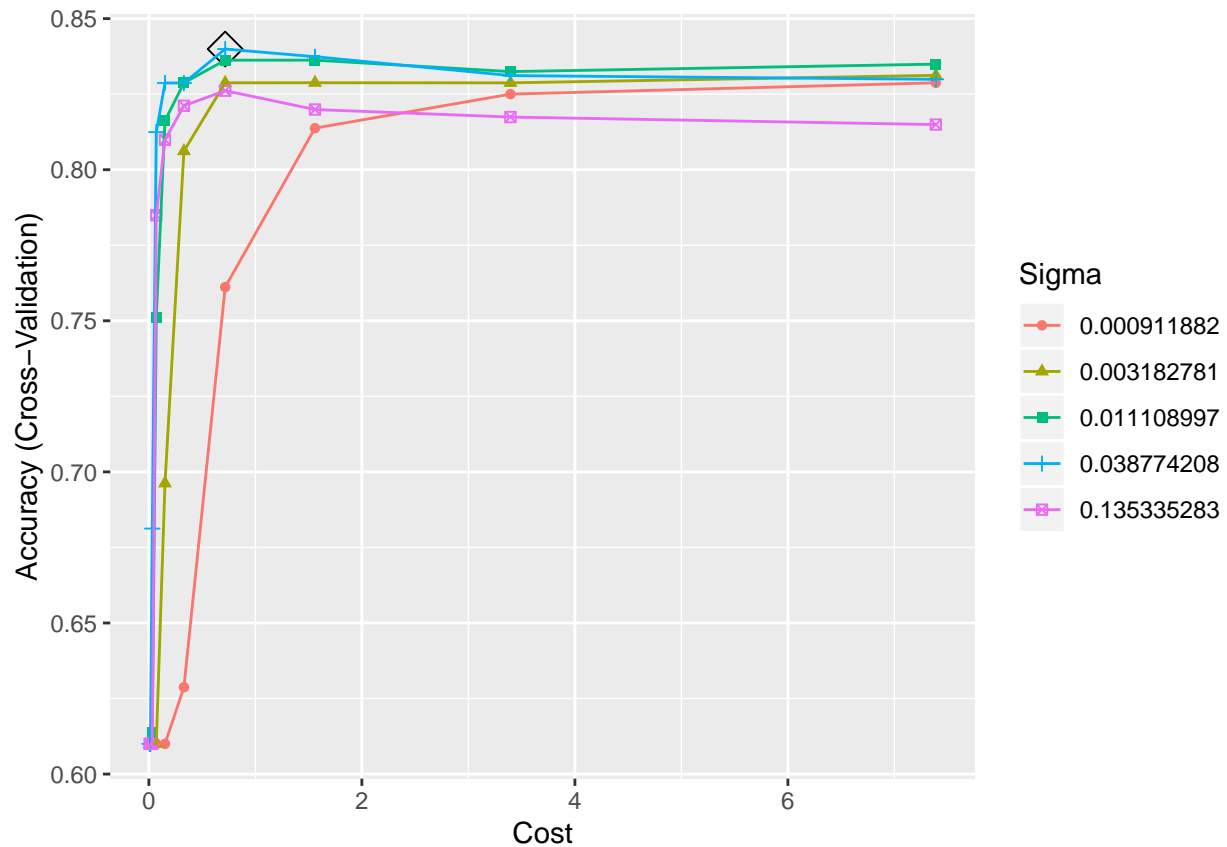
The test error rate is 0.1259.

## summary using caret



```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  CH  MM
##      CH 145  27
##      MM  20  78
##
##           Accuracy : 0.8259
##           95% CI : (0.7753, 0.8692)
##      No Information Rate : 0.6111
##      P-Value [Acc > NIR] : 1.626e-14
##
##           Kappa : 0.6293
##
##  Mcnemar's Test P-Value : 0.3815
##
##           Sensitivity : 0.8788
##           Specificity : 0.7429
##      Pos Pred Value : 0.8430
##      Neg Pred Value : 0.7959
##           Prevalence : 0.6111
##      Detection Rate : 0.5370
##      Detection Prevalence : 0.6370
```

```
##      Balanced Accuracy : 0.8108
##
##      'Positive' Class : CH
##
```

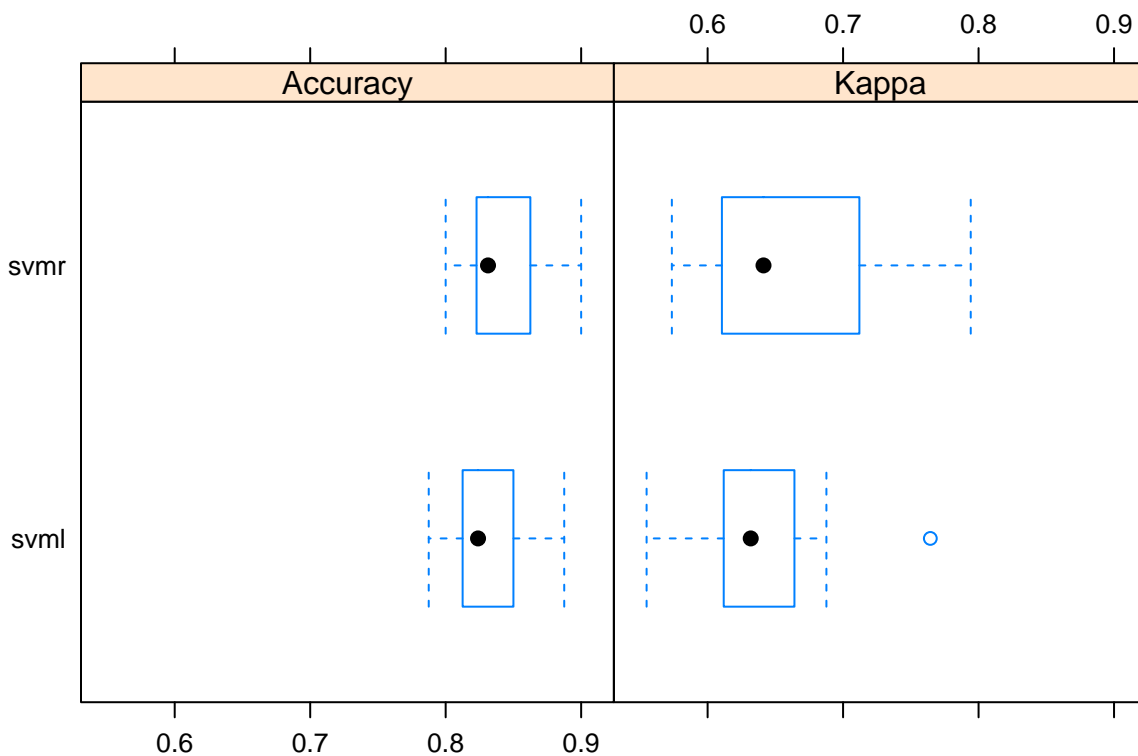


```
## Confusion Matrix and Statistics
##
##      Reference
## Prediction CH MM
##      CH 142  31
##      MM  23  74
##
##      Accuracy : 0.8
##      95% CI : (0.7472, 0.846)
##      No Information Rate : 0.6111
##      P-Value [Acc > NIR] : 2.106e-11
##
##      Kappa : 0.5733
##
##      McNemar's Test P-Value : 0.3408
##
##      Sensitivity : 0.8606
##      Specificity : 0.7048
##      Pos Pred Value : 0.8208
##      Neg Pred Value : 0.7629
##      Prevalence : 0.6111
```

```

##          Detection Rate : 0.5259
##    Detection Prevalence : 0.6407
##      Balanced Accuracy : 0.7827
##
##      'Positive' Class : CH
##
##
## Call:
## summary.resamples(object = res)
##
## Models: svm1, svmr
## Number of resamples: 10
##
## Accuracy
##      Min.   1st Qu.   Median     Mean 3rd Qu.   Max. NA's
## svm1 0.7875 0.8130787 0.8238924 0.8300100 0.846875 0.8875    0
## svmr 0.8000 0.8233386 0.8312500 0.8399328 0.859375 0.9000    0
##
## Kappa
##      Min.   1st Qu.   Median     Mean 3rd Qu.   Max. NA's
## svm1 0.5549738 0.6127724 0.6318311 0.6402035 0.6598802 0.7643979    0
## svmr 0.5736176 0.6123594 0.6412723 0.6587101 0.7059572 0.7942122    0

```



The linear boudary has higher accuracy and Kappa statistics value.