Support Vector Machines(SVMs) Tutorial

Sonali Narang

11/12/2019

Support Vector Machines(SVMs)

A Support Vector Machine (SVM) is a discriminative classifier formally defined by a separating hyperplane. Given labeled training data, the algorithm outputs an optimal hyperplane which categorizes new examples.

The Breast Cancer Dataset

699 Observations, 11 variables Predictor Variable: Class–benign or malignant

```
data(BreastCancer)

#bc = BreastCancer %>%
#  mutate_if(is.character, as.numeric)
#bc[is.na(bc)] = 0

BreastCancer_num = transform(BreastCancer, Id = as.numeric(Id),
                         Cl.thickness = as.numeric(Cl.thickness),
                         Cell.size = as.numeric(Cell.size),
                         Cell.shape = as.numeric(Cell.shape),
                         Marg.adhesion = as.numeric(Marg.adhesion),
                         Epith.c.size = as.numeric(Epith.c.size),
                         Bare.nuclei = as.numeric(Bare.nuclei),
                         Bl.cromatin = as.numeric(Bl.cromatin),
                         Normal.nucleoli = as.numeric(Normal.nucleoli),
                         Mitoses = as.numeric(Mitoses))

BreastCancer_num[is.na(BreastCancer_num)] = 0

train_size = floor(0.75 * nrow(BreastCancer_num))
train_pos <- sample(seq_len(nrow(BreastCancer_num)), size = train_size)

train_classification <- BreastCancer_num[train_pos, ]
test_classification <- BreastCancer_num[-train_pos, ]
```

##SVM

```
set.seed(1112)
control = trainControl(method = "repeatedcv", repeats = 5, classProbs = T, savePredictions = T)

svm = train(Class ~ Id + Cl.thickness + Cell.size + Cell.shape + Marg.adhesion + Epith.c.size + Bare.nu

svm
```

```
## Support Vector Machines with Linear Kernel
##
```

```
## 524 samples
##  10 predictor
##   2 classes: 'benign', 'malignant'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold, repeated 5 times)
## Summary of sample sizes: 472, 472, 472, 472, 471, 471, ...
## Resampling results:
##
##   Accuracy   Kappa
##   0.9667774  0.9259438
##
## Tuning parameter 'C' was held constant at a value of 1
```

##Receiver operating characteristic(ROC) curve

```
roc(predictor = svm$pred$malignant, response = svm$pred$obs)$auc
```

```
## Setting levels: control = benign, case = malignant
```

```
## Setting direction: controls < cases
```

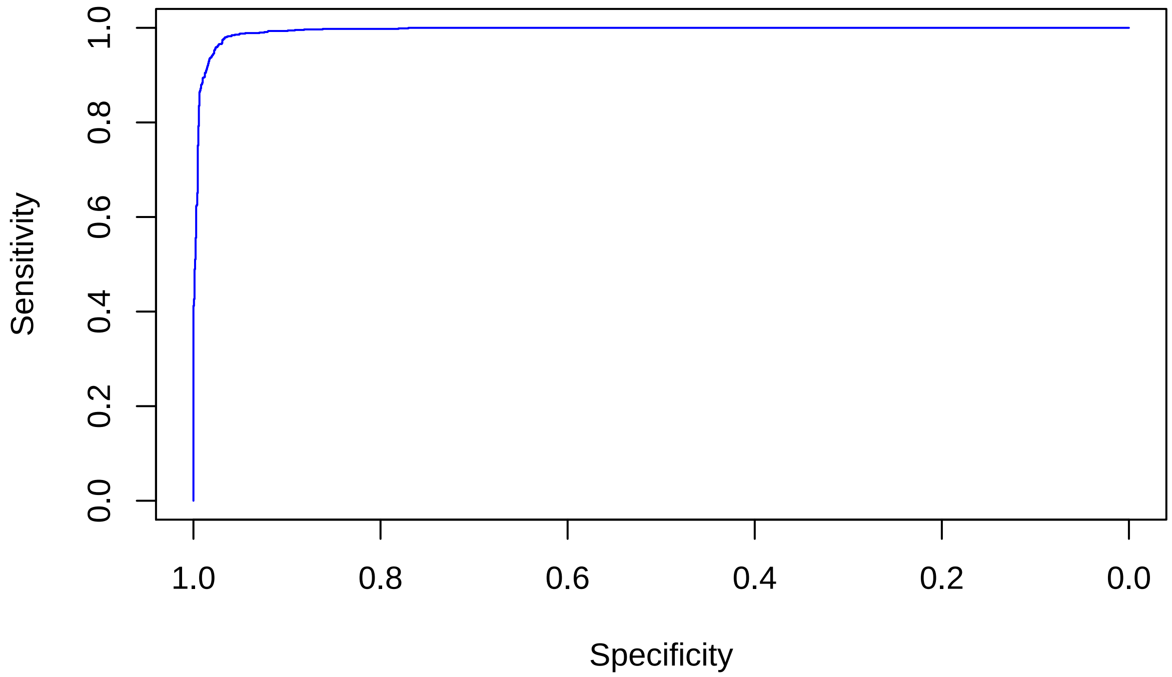
```
## Area under the curve: 0.9946
```

```
plot(x = roc(predictor = svm$pred$malignant, response = svm$pred$obs)$specificities, y = roc(predictor =
```

```
## Setting levels: control = benign, case = malignant
## Setting direction: controls < cases
```

```
## Setting levels: control = benign, case = malignant
```

```
## Setting direction: controls < cases
```



## Test Set

```
svm_test = predict(svm, newdata = test_classification)
confusionMatrix(svm_test, reference = test_classification$Class)
```

```
## Confusion Matrix and Statistics
##
##            Reference
## Prediction  benign malignant
##   benign       111         5
##   malignant      4        55
##
##                Accuracy : 0.9486
##                  95% CI : (0.9046, 0.9762)
##     No Information Rate : 0.6571
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.8854
##
##  Mcnemar's Test P-Value : 1
##
##             Sensitivity : 0.9652
##             Specificity : 0.9167
##          Pos Pred Value : 0.9569
##          Neg Pred Value : 0.9322
##              Prevalence : 0.6571
##          Detection Rate : 0.6343
##    Detection Prevalence : 0.6629
##       Balanced Accuracy : 0.9409
##
##        'Positive' Class : benign
##
```

SVM with a radial kernel

```
set.seed(1112)
control = trainControl(method = "repeatedcv", repeats = 5, classProbs = T, savePredictions = T)

svm = train(Class ~ Id + Cl.thickness + Cell.size + Cell.shape + Marg.adhesion + Epith.c.size + Bare.nu

svm
```

```
## Support Vector Machines with Radial Basis Function Kernel
##
## 524 samples
##  10 predictor
##   2 classes: 'benign', 'malignant'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold, repeated 5 times)
## Summary of sample sizes: 472, 472, 472, 472, 471, 471, ...
## Resampling results across tuning parameters:
##
##   C       Accuracy   Kappa
##     0.25  0.9462256  0.8845771
##     0.50  0.9481487  0.8886031
##     1.00  0.9519513  0.8964471
##     2.00  0.9511821  0.8947132
##     4.00  0.9500355  0.8923302
##     8.00  0.9508047  0.8939457
##    16.00  0.9492662  0.8907169
##    32.00  0.9500282  0.8923031
##    64.00  0.9504128  0.8930790
##   128.00  0.9508047  0.8939448
##
## Tuning parameter 'sigma' was held constant at a value of 0.7192712
## Accuracy was used to select the optimal model using the largest value.
## The final values used for the model were sigma = 0.7192712 and C = 1.
```

##Receiver operating characteristic(ROC) curve

```
roc(predictor = svm$pred$malignant, response = svm$pred$obs)$auc
```

```
## Setting levels: control = benign, case = malignant
```

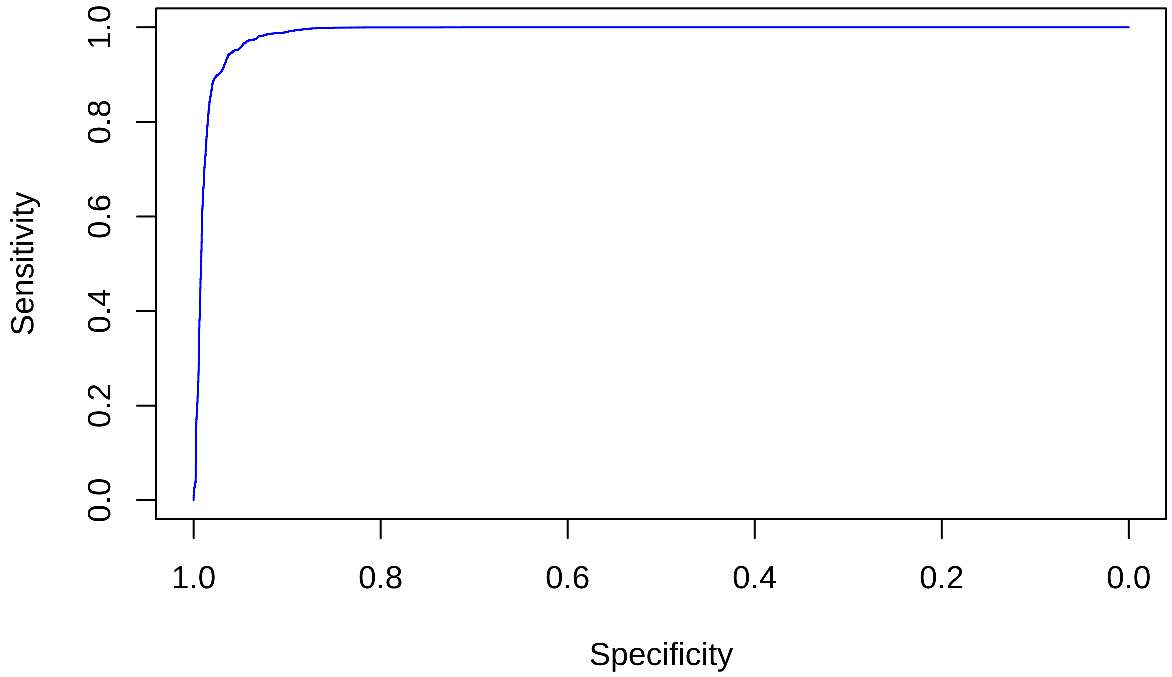
```
## Setting direction: controls < cases
## Area under the curve: 0.9873
```

```
plot(x = roc(predictor = svm$pred$malignant, response = svm$pred$obs)$specificities, y = roc(predictor
```

```
## Setting levels: control = benign, case = malignant
## Setting direction: controls < cases
```

```
## Setting levels: control = benign, case = malignant
```

```
## Setting direction: controls < cases
```



Test Set

```
svm_test = predict(svm, newdata = test_classification)
confusionMatrix(svm_test, reference = test_classification$Class)
```

```
## Confusion Matrix and Statistics
##
##            Reference
## Prediction  benign malignant
##   benign       109         2
##   malignant      6        58
```

```
## 
##                Accuracy : 0.9543          
##                  95% CI : (0.9119, 0.9801)
##     No Information Rate : 0.6571          
##     P-Value [Acc > NIR] : <2e-16          
##                                           
##                   Kappa : 0.9001          
##                                           
##  Mcnemar's Test P-Value : 0.2888          
##                                           
##             Sensitivity : 0.9478          
##             Specificity : 0.9667          
##          Pos Pred Value : 0.9820          
##          Neg Pred Value : 0.9062          
##              Prevalence : 0.6571          
##          Detection Rate : 0.6229          
##    Detection Prevalence : 0.6343          
##       Balanced Accuracy : 0.9572          
##                                           
##        'Positive' Class : benign          
## 
```

##Homework

1. Choose an appropriate machine learning dataset and use SVM with two different kernels. Campare the results.
2. Attempt using SVM after using a feature selection method. Do the results improve? Explain.