**Assignment #7: SVM**

**Submit through link: eCampus -> Assignments->Assignment 7 Submission**

**Deadline: December 3 (Tuesday) @5:00 pm**

**The filename should have this format: LastName-FirstName-hw07.doc**

**Problem 1**

In this problem, you will use support vector approaches to predict whether a given car gets high or low gas mileage based on the Auto data set in the ISLR package.

(a) Create a binary variable that takes on a 1 for cars with gas mileage above the median, and a 0 for cars with gas mileage below the median. Use this variable as response in the following analysis.

(b) Fit a support vector classifier to the data with various values of cost, to predict whether a car gets high or low gas mileage. Report the cross-validation errors associated with different values of this parameter. Comment on your results.

(c) Now repeat (b), this time using SVMs with radial and polynomial kernels, with different values of gamma, degree and cost. Comment on your results.

**Problem 2**

This problem uses the OJ data set in the ISLR package.

(a) Create a training set containing a random sample of 800 observations, and a test set containing the remaining observations.

> set.seed(1)

> data('OJ')

> inTrain <- sample(nrow(OJ), 800, replace = FALSE)

> training <- OJ[inTrain,]

> testing <- OJ[-inTrain,]

(b) Fit a support vector classifier to the training data using cost=0.01, with Purchase as the response and the other variables as predictors. Use the summary() function to produce summary statistics, and describe the results obtained.

> svm\_linear <- svm(Purchase ~ ., data = training,

+ kernel = 'linear',

+ cost = 0.01)

> summary(svm\_linear)

Call:

svm(formula = Purchase ~ ., data = training, kernel = "linear", cost = 0.01)

Parameters:

SVM-Type: C-classification

SVM-Kernel: linear

cost: 0.01

Number of Support Vectors: 435

( 219 216 )

Number of Classes: 2

Levels:

CH MM

**The number of support vectors are 435 with 219 of class CH and 216 of class MM**

(c) What are the training and test error rates?

Training:

> postResample(predict(svm\_linear, training), training$Purchase)

Accuracy Kappa

0.8250000 0.6313971

Testing:

> postResample(predict(svm\_linear, testing), testing$Purchase)

Accuracy Kappa

0.8222222 0.6082699

(d) Use the tune() function to select an optimal cost. Consider value in the range 0.01 to 10.

> svm\_linear\_tune <- train(Purchase ~ ., data = training,

+ method = 'svmLinear2',

+ trControl = trainControl(method = 'cv', number = 10),

+ preProcess = c('center', 'scale'),

+ tuneGrid = expand.grid(cost = seq(0.01, 10, length.out = 20)))

> svm\_linear\_tune

Support Vector Machines with Linear Kernel

800 samples

17 predictor

2 classes: 'CH', 'MM'

Pre-processing: centered (17), scaled (17)

Resampling: Cross-Validated (10 fold)

Summary of sample sizes: 721, 720, 720, 720, 721, 719, ...

Resampling results across tuning parameters:

cost Accuracy Kappa

0.0100000 0.8199215 0.6202565

0.5357895 0.8273760 0.6360834

1.0615789 0.8236101 0.6284665

1.5873684 0.8261105 0.6333280

2.1131579 0.8261105 0.6333280

2.6389474 0.8273605 0.6362121

3.1647368 0.8261105 0.6338114

3.6905263 0.8248605 0.6309732

4.2163158 0.8248605 0.6309732

4.7421053 0.8261105 0.6338114

5.2678947 0.8273605 0.6361662

5.7936842 0.8273605 0.6361662

6.3194737 0.8260947 0.6331693

6.8452632 0.8260947 0.6331693

7.3710526 0.8260947 0.6331693

7.8968421 0.8273605 0.6361662

8.4226316 0.8273605 0.6361662

8.9484211 0.8273605 0.6361662

9.4742105 0.8248447 0.6308145

10.0000000 0.8248447 0.6308145

Accuracy was used to select the optimal model using the largest value.

The final value used for the model was cost = 0.5357895.

(e) Compute the training and test error rates using this new value for cost.

> postResample(predict(svm\_linear\_tune, training), training$Purchase)

Accuracy Kappa

0.8350000 0.6524601

> postResample(predict(svm\_linear\_tune, testing), testing$Purchase)

Accuracy Kappa

0.8444444 0.6585983

(f) Repeat parts (b) through (e) using a support vector machine with a radial kernel. Use the tune() function to select an optimal cost and gamma.

> svm\_radial <- svm(Purchase ~ ., data = training,

+ method = 'radial',

+ cost = 0.01)

> summary(svm\_radial)

Call:

svm(formula = Purchase ~ ., data = training, method = "radial", cost = 0.01)

Parameters:

SVM-Type: C-classification

SVM-Kernel: radial

cost: 0.01

Number of Support Vectors: 634

( 319 315 )

Number of Classes: 2

Levels:

CH MM

> postResample(predict(svm\_radial, training), training$Purchase)

Accuracy Kappa

0.60625 0.00000

> postResample(predict(svm\_radial, testing), testing$Purchase)

Accuracy Kappa

0.6222222 0.0000000

> svm\_radial\_tune

Support Vector Machines with Radial Basis Function Kernel

800 samples

17 predictor

2 classes: 'CH', 'MM'

Pre-processing: centered (17), scaled (17)

Resampling: Cross-Validated (10 fold)

Summary of sample sizes: 721, 720, 719, 719, 720, 721, ...

Resampling results across tuning parameters:

C Accuracy Kappa

0.0100000 0.6062600 0.0000000

0.5357895 0.8224203 0.6213094

1.0615789 0.8286553 0.6357260

1.5873684 0.8224512 0.6218004

2.1131579 0.8250299 0.6266581

2.6389474 0.8212795 0.6188102

3.1647368 0.8200137 0.6161205

3.6905263 0.8200449 0.6160717

4.2163158 0.8187949 0.6131171

4.7421053 0.8200295 0.6155611

5.2678947 0.8187637 0.6131963

5.7936842 0.8150445 0.6047736

6.3194737 0.8138100 0.6022973

6.8452632 0.8150445 0.6047736

7.3710526 0.8150445 0.6047736

7.8968421 0.8150445 0.6047736

8.4226316 0.8137787 0.6023727

8.9484211 0.8125287 0.5995732

9.4742105 0.8100441 0.5942645

10.0000000 0.8100441 0.5942645

Tuning parameter 'sigma' was held constant at a value of 0.05691

Accuracy was used to select the optimal model using the largest value.

The final values used for the model were sigma = 0.05691 and C = 1.061579.

> postResample(predict(svm\_radial\_tune, training), training$Purchase)

Accuracy Kappa

0.8525000 0.6854487

> postResample(predict(svm\_radial\_tune, testing), testing$Purchase)

Accuracy Kappa

0.8185185 0.6024878

(g) Repeat parts (b) through (e) using a support vector machine with a polynomial kernel. Set degree=2. Use the tune() function to select an optimal cost.

> svm\_poly <- svm(Purchase ~ ., data = training,

+ method = 'polynomial', degree = 2,

+ cost = 0.01)

> summary(svm\_poly)

Call:

svm(formula = Purchase ~ ., data = training, method = "polynomial", degree = 2, cost = 0.01)

Parameters:

SVM-Type: C-classification

SVM-Kernel: radial

cost: 0.01

Number of Support Vectors: 634

( 319 315 )

Number of Classes: 2

Levels:

CH MM

> postResample(predict(svm\_poly, training), training$Purchase)

Accuracy Kappa

0.60625 0.00000

> postResample(predict(svm\_poly, testing), testing$Purchase)

Accuracy Kappa

0.6222222 0.0000000

> svm\_poly\_tune <- train(Purchase ~ ., data = training,

+ method = 'svmPoly',

+ trControl = trainControl(method = 'cv', number = 10),

+ preProcess = c('center', 'scale'),

+ tuneGrid = expand.grid(degree = 2,

+ C = seq(0.01, 10, length.out = 20),

+ scale = TRUE))

> svm\_poly\_tune

Support Vector Machines with Polynomial Kernel

800 samples

17 predictor

2 classes: 'CH', 'MM'

Pre-processing: centered (17), scaled (17)

Resampling: Cross-Validated (10 fold)

Summary of sample sizes: 720, 720, 721, 719, 720, 721, ...

Resampling results across tuning parameters:

C Accuracy Kappa

0.0100000 0.8125299 0.5951821

0.5357895 0.7975125 0.5687539

1.0615789 0.7987934 0.5721940

1.5873684 0.7987621 0.5721616

2.1131579 0.7975121 0.5686575

2.6389474 0.8000125 0.5734773

3.1647368 0.8000125 0.5734773

3.6905263 0.8049975 0.5845053

4.2163158 0.8037475 0.5820222

4.7421053 0.8049975 0.5843887

5.2678947 0.8050287 0.5846129

5.7936842 0.8050287 0.5846129

6.3194737 0.8050441 0.5846338

6.8452632 0.8037941 0.5819771

7.3710526 0.8037941 0.5819771

7.8968421 0.8025596 0.5790185

8.4226316 0.8038254 0.5820067

8.9484211 0.8038254 0.5820067

9.4742105 0.8025908 0.5795958

10.0000000 0.8013408 0.5766545

Tuning parameter 'degree' was held constant at a value of 2

Tuning parameter 'scale' was held constant at a value of TRUE

Accuracy was used to select the optimal model using the largest value.

The final values used for the model were degree = 2, scale = TRUE and C = 0.01.

> postResample(predict(svm\_poly\_tune, training), training$Purchase)

Accuracy Kappa

0.850000 0.678295

> postResample(predict(svm\_poly\_tune, training), training$Purchase)

Accuracy Kappa

0.850000 0.678295

(h) Overall, which approach seems to give the best results on this data?

**Overall the models are very similar, but the radial kernel does best by a small margin.**