Week 1 HW - ISYE 6501

May 23, 2018

Question 2.1

We would like to predict which secondary students will need remedial math services at the start of the school year. This is a binary classification problem, so we may try using classifiers such as logistic regression, svm, knn etc.

Some predictors we can use are as follows (all are for the previous school year): Math class grade, Standardized test scores, Number of days student was absent from school. The analysis may be performed at a district level in order to identify all students who need additional services.

Question 2.2

1: Load relevant libraries and read in the data:

Setting up the provided model:

```
model <- ksvm(as.matrix(data[,1:10]),as.factor(data[,11]),type="C-svc",kernel="vanilladot",C=100,scaled
```

```
##  Setting default kernel parameters
```

Calculate $a_1, ..., a_m$:

```
a<-colSums(model@xmatrix[[1]]*model@coef[[1]])
a
```

```
##            V1            V2            V3            V4            V5
## -0.0010065348 -0.0011729048 -0.0016261967  0.0030064203  1.0049405641
##            V6            V7            V8            V9           V10
## -0.0028259432  0.0002600295 -0.0005349551 -0.0012283758  0.1063633995
```

```
a0<- -model@b
a0
```

```
## [1] 0.08158492
```

The equation of our SVM classifier is

$a_0 + a_1V_1 + a_2V_2 + a_3V_3 + a_4V_4 + a_5V_5 + a_6V_6 + a_7V_7 + a_8V_8 + a_9V_9 + a_{10}V_{10} = 0$

with the coefficients as determined above. Now we see what the model predicts:

```
pred <- predict(model, data[,1:10])
```

Percent of model's predictions that match the actual data:

```
sum(pred==data[,11])/nrow(data) * 100
```

```
## [1] 86.39144
```

Question: Can we choose a value of C in ksvm that yields higher accuracy? We set up some test values of C to choose the correct order of magnitude, and then loop the model over each choice of C to see what yields the highest accuracy. Our test vector was

$$C_{vals} = c(0.0000001, 0.000001, 0.00001, 0.0001, 0.001, 0.01, 0.1, 1, 10, 100, 1000, 10000)$$

```
## Best SVM is model number  6
## Best C value is  0.01

## Best validation set correctness is  0.8639144

##  Setting default kernel parameters

## Performance on test data =  0.8639144
```

We see that $C = 0.01$ yields the highest accuracy of the possible C values, with a validation set accuracy of 0.8639144. Now the question becomes whether we can find a higher validation accuracy by searching more carefully close to 0.01. We run the same loop as above, just with a more precise vector of C's: $C_{vals} = seq(0.001, 0.05, by = 0.00001)$. Please reference my R code to see the implementation.

This method yields C=0.00139 with validation accuracy of 0.8685015 (improvement of 0.0045871).

2: (optional)

3: Implement knn, find a good value of k, and show how well it classifies the data in the full dataset.

We follow the code provided in TA video 3 to loop over the values of k (from 1 to 100) and choose the model with the highest accuracy:

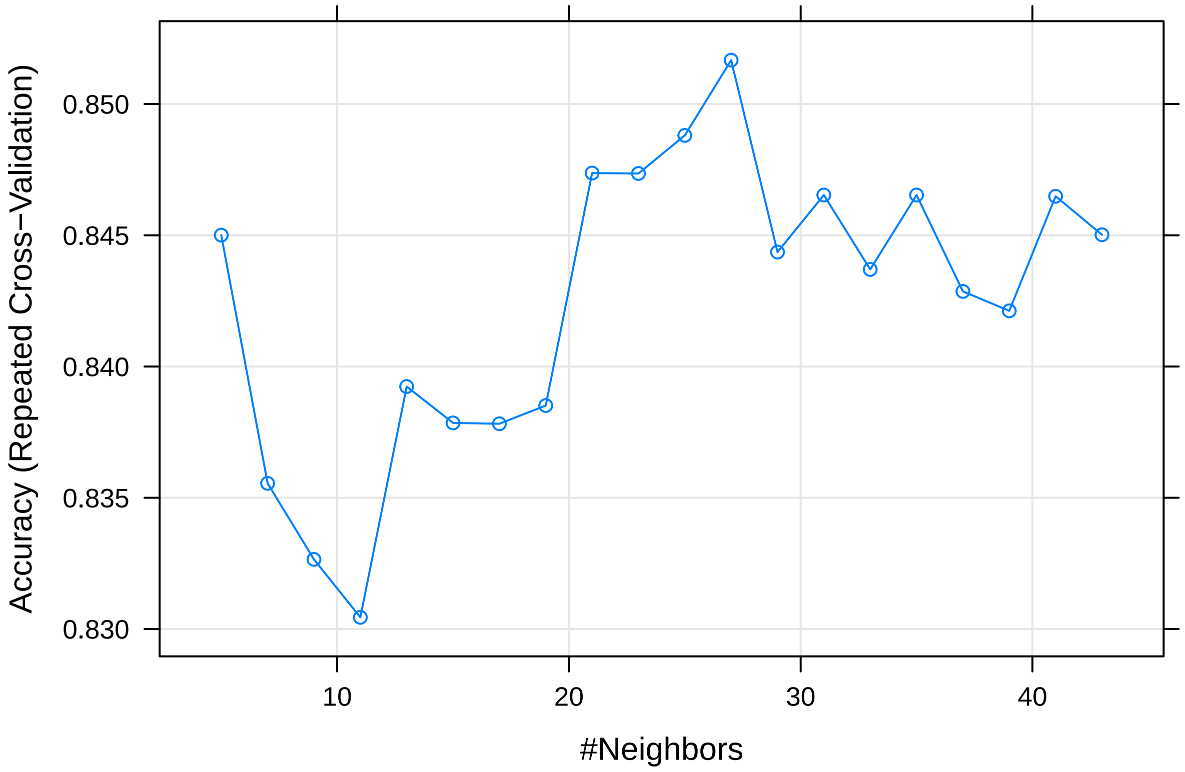
```
## Desired k value:  12

## Associated accuracy:  0.853211
```

Question 3.1

(a) We want to implement the k-nearest-neighbors classifier using cross-validation. We use the caret package's built-in cross validation feature to decide on the number of clusters, and decided on a 70-30 train-test split. The caret package produces some nice graphics:

```
## k-Nearest Neighbors
##
## 458 samples
##  10 predictor
##   2 classes: '0', '1'
##
## Pre-processing: centered (10), scaled (10)
## Resampling: Cross-Validated (10 fold, repeated 3 times)
## Summary of sample sizes: 411, 413, 412, 412, 413, 412, ...
## Resampling results across tuning parameters:
##
##   k   Accuracy   Kappa
##    5  0.8450057  0.6868307
##    7  0.8355497  0.6687414
##    9  0.8326519  0.6620713
##   11  0.8304444  0.6566408
##   13  0.8392373  0.6727919
##   15  0.8378511  0.6697094
##   17  0.8378203  0.6697948
##   19  0.8385141  0.6704676
##   21  0.8473694  0.6879160
##   23  0.8473546  0.6875115
##   25  0.8488039  0.6904732
##   27  0.8516709  0.6961117
##   29  0.8443602  0.6809580
##   31  0.8465334  0.6853636
##   33  0.8436986  0.6794304
##   35  0.8465341  0.6846056
##   37  0.8428619  0.6772105
##   39  0.8421218  0.6752421
##   41  0.8464864  0.6838844
##   43  0.8450204  0.6808513
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was k = 27.
```



We see that the optimal number of clusters is 27. The accuracy associated with this value of k is 0.8516709.

(b) We now break up the data into training/test/validation sets manually. We chose a 70/15/15 train/test/validate split.

We choose a svm classifier and iterate over the values of C (they are the same as in 2.1) to choose the best classifier on the validation set.

Our results show that the value of C yielding the highest accuracy is again 0.01. As before, we investigate values close to this more carefully. We set $C_{vals} = seq(0.001, 0.05, by = 0.00001)$ and iterate through to find the value of C with highest validation accuracy. The final chosen value of C is 0.00196, with a re-trained test set performance of 0.8979592.