ISSR Short Course

Gregory Matthews [1]

[1]Department of Mathematics and Statistics
Loyola University Chicago

June 2015

Outline

- The last thing we are going to do is document classification.
- We can split our corpus into two pieces:
  - Training Data
  - Test Data
- We use the training data to build a model.
- Then the test data gets classified based on the model.

```
#http://journal.r-project.org/archive/2013-1/collingwood-j
library(RTextTools)

## Loading required package:  SparseM
##
## Attaching package:  'SparseM'
##
## The following object is masked from
'package:base':
##
##    backsolve

data(USCongress)
#help(USCongress)
summary(USCongress)

##         ID              cong         billnum      h_or_sen
##  Min.   :   1   Min.   :107   Min.   :   1   HR:1269   M
##  1st Qu.:1113   1st Qu.:107   1st Qu.:1113   S :3180   1
##  Median :2225   Median :107   Median :2225
```

```
#Create a document matrix
doc_matrix <- create_matrix(USCongress$text, language="english", removeNumbers=TRUE,
                            stemWords=TRUE, removeSparseTerms=.998)
#create a container
container <- create_container(doc_matrix, USCongress$major, trainSize=1:4000,
                              testSize=4001:4449, virgin=FALSE)
```

```r
#Build the training data set.
SVM <- train_model(container,"SVM")
```

```
summary(SVM)
```

```
##
## Call:
## svm.default(x = container@training_matrix, y = container@training_codes,
##     kernel = kernel, cost = cost, cross = cross, probability = TRUE,
##     method = method)
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  radial
##        cost:  100
##       gamma:  0.001106195
##
## Number of Support Vectors:  2890
##
##  ( 118 194 268 234 160 239 75 269 106 260 26 128 131 75 154 132 78 101 83 59 )
##
##
## Number of Classes:  20
##
## Levels:
##  1 2 3 4 5 6 7 8 10 12 13 14 15 16 17 18 19 20 21 99
```

```
#Classify using the model we just built
SVM_CLASSIFY <- classify_model(container, SVM)
#Let's look at analytics
analytics <- create_analytics(container,
                              cbind(SVM_CLASSIFY))
```

```
summary(analytics)

## ENSEMBLE SUMMARY
##
##          n-ENSEMBLE COVERAGE n-ENSEMBLE RECALL
## n >= 1                     1                0.75
##
##
## ALGORITHM PERFORMANCE
##
## SVM_PRECISION    SVM_RECALL    SVM_FSCORE
##        0.6525        0.6430        0.6390
```

```
GLMNET <- train_model(container,"GLMNET")
#Load a previously saved version
load("/Users/gregorymatthews/Dropbox/ISSRshortCourse/GLMNET.RData")
#save(GLMNET,"/Users/gregorymatthews/Dropbox/ISSRshortCourse/GLMNET.RData")
GLMNET_CLASSIFY <- classify_model(container, GLMNET)
#here we are looknig at the performance of both SVM and GLMNET
analytics <- create_analytics(container,cbind(SVM_CLASSIFY,GLMNET_CLASSIFY))
summary(analytics)
create_ensembleSummary(analytics@document_summary)
```

- Cross validation splits the data into n different sets of approximately equal size.
- Each set of data is then classified using training data that is all of the data outside of the set.
- Here we will use $n = 4$ as an example.

```
#Cross validation
cross_validate(container, 4, "SVM")
#Fold 1 Out of Sample Accuracy = 0.731641
#Fold 2 Out of Sample Accuracy = 0.7130045
#Fold 3 Out of Sample Accuracy = 0.7146814
#Fold 4 Out of Sample Accuracy = 0.7195122
```

```
#Cross validation
cross_validate(container, 4, "GLMNET")
#Fold 1 Out of Sample Accuracy = 0.2060345
#Fold 2 Out of Sample Accuracy = 0.2251356
#Fold 3 Out of Sample Accuracy = 0.2241071
#Fold 4 Out of Sample Accuracy = 0.2248354
```