Submission 9

Shannon Rush

July 19, 2014

A single generalized boosted regression model.

Loading

```r
source("../helpers/predictions.R")
```

```r
library(caret)
```

```
## Loading required package: lattice
## Loading required package: ggplot2
```

```r
library(doMC)
```

```
## Loading required package: foreach
## Loading required package: iterators
## Loading required package: parallel
```

```r
train <- read.csv("../../data/processed/processed_train.csv")
test <- read.csv("../../data/original/test.csv")
```

Data

```r
set.seed(123)
training.indices <- createDataPartition(train$Label, p=0.6, list=F)
training <- train[training.indices,]
validation <- train[-training.indices,]
```

Models

Fit a GBM with training data

```r
predictors <- training[,setdiff(names(training),c("Label","EventId"))]
registerDoMC(cores=8)
gbm.fit <- train(x=predictors, y=training$Label, method="gbm")
```

```
## Loading required package: gbm
## Loading required package: survival
## Loading required package: splines
##
## Attaching package: 'survival'
```

```
##
## The following object is masked from 'package:caret':
##
##     cluster
##
## Loaded gbm 2.1
```

```
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.2303             nan     0.1000    0.0277
##      2        1.1848             nan     0.1000    0.0228
##      3        1.1489             nan     0.1000    0.0178
##      4        1.1170             nan     0.1000    0.0160
##      5        1.0889             nan     0.1000    0.0139
##      6        1.0654             nan     0.1000    0.0116
##      7        1.0424             nan     0.1000    0.0114
##      8        1.0245             nan     0.1000    0.0090
##      9        1.0072             nan     0.1000    0.0087
##     10        0.9936             nan     0.1000    0.0066
##     20        0.9022             nan     0.1000    0.0032
##     40        0.8392             nan     0.1000    0.0009
##     60        0.8109             nan     0.1000    0.0004
##     80        0.7928             nan     0.1000    0.0005
##    100        0.7805             nan     0.1000    0.0003
##    120        0.7714             nan     0.1000    0.0002
##    140        0.7634             nan     0.1000    0.0002
##    150        0.7602             nan     0.1000    0.0000
```

```
    gbm.fit
```

```
## Stochastic Gradient Boosting
##
## 150001 samples
##     30 predictors
##      2 classes: 'b', 's'
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
##
## Summary of sample sizes: 150001, 150001, 150001, 150001, 150001, 150001, ...
##
## Resampling results across tuning parameters:
##
##   interaction.depth  n.trees  Accuracy  Kappa  Accuracy SD  Kappa SD
##   1                   50      0.8       0.6    0.001        0.003
##   1                  100      0.8       0.6    0.001        0.003
##   1                  200      0.8       0.6    0.001        0.003
##   2                   50      0.8       0.6    0.001        0.002
##   2                  100      0.8       0.6    0.001        0.003
##   2                  200      0.8       0.6    0.001        0.003
##   3                   50      0.8       0.6    0.002        0.004
##   3                  100      0.8       0.6    0.001        0.003
##   3                  200      0.8       0.6    0.001        0.003
##
```

```
## Tuning parameter 'shrinkage' was held constant at a value of 0.1
## Accuracy was used to select the optimal model using  the largest value.
## The final values used for the model were n.trees = 150,
##  interaction.depth = 3 and shrinkage = 0.1.
```

```
    val <- validation[,setdiff(names(validation),c("EventId","Label"))]
    pred.val <- predict(gbm.fit, val)
    confusionMatrix(pred.val, validation$Label)
```

```
## Warning: NAs produced by integer overflow
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction     b     s
##          b 58897 10421
##          s  6836 23845
##
##                Accuracy : 0.827
##                  95% CI : (0.825, 0.83)
##     No Information Rate : 0.657
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : NA
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.896
##             Specificity : 0.696
##          Pos Pred Value : 0.850
##          Neg Pred Value : 0.777
##              Prevalence : 0.657
##          Detection Rate : 0.589
##    Detection Prevalence : 0.693
##       Balanced Accuracy : 0.796
##
##        'Positive' Class : b
##
```

```
    pred.test <- predict(gbm.fit, test, type="prob")
    pred.df <- PrepPrediction(pred.test, test)
    WriteSubmission(pred.df, 9)
```