

Trying various classification algorithms

First let's try SVM with radial kernel for classification. We train on the first 5000 data and test on the next 5000 data from the connect four board configuration dataset provided by kaggle. The description of the dataset can be found here <https://www.kaggle.com/tbrewer/connect-4>

```
library(tidyverse)

## -- Attaching packages ----- tidyverse 1.3.0 --
## v ggplot2 3.2.1      v purrr   0.3.3
## v tibble  2.1.3      v dplyr  0.8.3
## v tidyr   1.0.0      v stringr 1.4.0
## v readr   1.3.1      v forcats 0.4.0

## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()     masks stats::lag()

library(e1071)
data <- read.csv("./c4_game_database.csv")
data$winner <- as.factor(data$winner)

data.train <- data[1:5000,]
svm.fit <- svm(winner~.,data=data.train,kernel="radial",cost=10,gamma=1)
#summary(svm.fit)
#table(svm.fit$fitted,data.train$winner)

data.test <- data[5001:10000,]
svm.predict <- predict(svm.fit,newdata=data.test)
table(svm.predict,data.test$winner)

##
## svm.predict    -1     0     1
##              -1    48     0     0
##              0     0     0     0
##              1  2303   203  2446
```

Appears to have 2506 testing errors or 50.12% test error rate.

Let's now try gradient boosting for classification. We have not tuned the parameters of gradient boosting, but just roughly try with 1000 trees and a default learning rate.

```
library(gbm)

## Loaded gbm 2.1.5

set.seed(1)
boost.model <- gbm(winner~.,data=data.train,distribution="multinomial",n.trees=1000,interaction.depth=4)
predBST <- predict(boost.model,newdata=data.test,n.trees=1000,type='response')
p.predBST <- apply(predBST,1,which.max)
boost.prediction <- colnames(predBST)[p.predBST]
#head(boost.prediction)
table(boost.prediction,data.test$winner)

##
## boost.prediction  -1     0     1
```

```
##           -1 2170    0 148
##           0   7 203    4
##           1  174    0 2294
```

Appears to have 333 testing errors or a 6.66% test error rate

We then try multinomial logistic regression on the same training and test data set.

```
library(nnet)
glm.fit <- multinom(winner~.,data=data.train)
```

```
## # weights: 132 (86 variable)
## initial value 5493.061443
## iter 10 value 2692.186810
## iter 20 value 2469.709080
## iter 30 value 2001.386535
## iter 40 value 1850.467144
## iter 50 value 1811.470718
## iter 60 value 1765.411343
## iter 70 value 1744.627933
## iter 80 value 1732.523694
## iter 90 value 1724.663940
## iter 100 value 1711.992902
## final value 1711.992902
## stopped after 100 iterations
```

```
prediction <- predict(glm.fit,data.test)
table(prediction,data.test$winner)
```

```
##
## prediction  -1    0    1
##           -1 2063   91 291
##           0   0    4   0
##           1  288  108 2155
```

There are 778 test errors, or a 15.56% test error rate.