Lab 6 – MATH 243

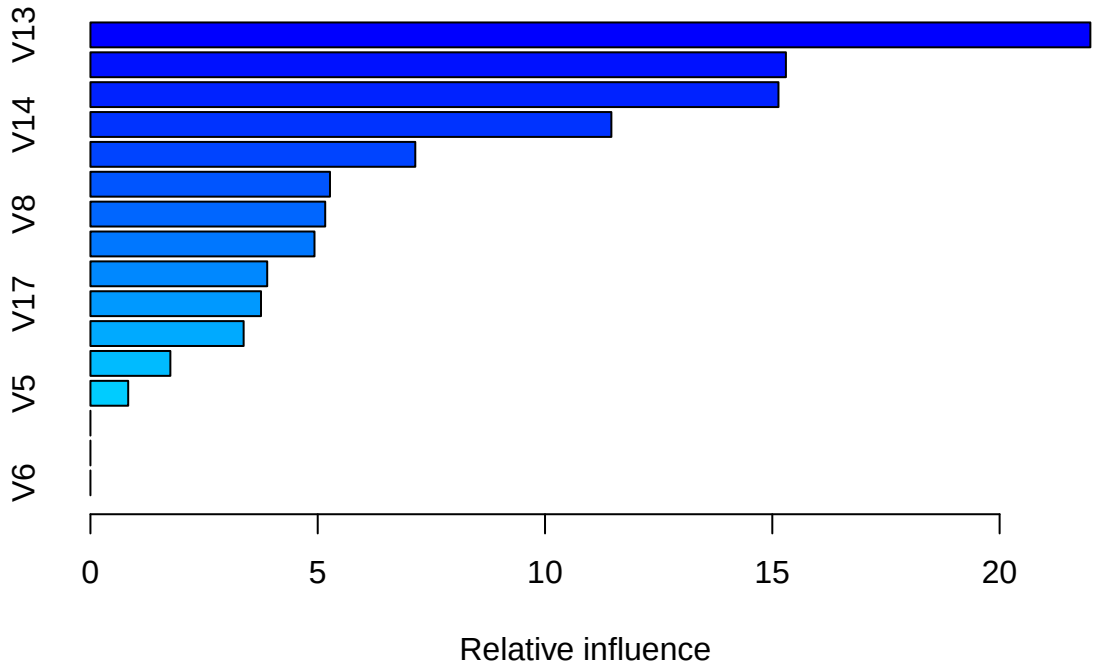
Theodore Dounias

November 6, 2017

Build a Boosted Tree

```
boost.letters <- gbm(V1~., data = lettersdf[train,], distribution = "multinomial", n.trees = 50,
                     interaction.depth = 1, shrinkage = 0.1)

summary(boost.letters)
```




```
##     var    rel.inf
## V13 V13 21.9965247
## V11 V11 15.2993528
## V12 V12 15.1356499
## V14 V14 11.4595433
## V9   V9  7.1464605
## V15 V15  5.2694983
## V8   V8  5.1645842
## V10 V10  4.9290784
## V4   V4  3.8886848
## V17 V17  3.7521023
## V16 V16  3.3698812
## V7   V7  1.7579011
```

```
## V5   V5  0.8307385
## V2   V2  0.0000000
## V3   V3  0.0000000
## V6   V6  0.0000000
```

Variable V13 seems to be the most important variable.

Assessing Predictions

```
yhat.boost <- predict(boost.letters, newdata = lettersdf[-train, ], n.trees = 50)

predicted <- LETTERS[apply(yhat.boost, 1, which.max)]

#1
conf_tb <- table(predicted, lettersdf$V1[-train])
conf_tb
```

```
##
## predicted   A   B   C   D   E   F   G   H   I   J   K   L   M   N   O   P
##         A 176   0   0   0   0   0   1   0   1   0   2  10   5   0   0   0
##         B   0 129   0  26   5  15   3   7  12  17   2   3   1   5   1   6
##         C   3   0 130   0  26   0  15   0   1   3   7   7   1   3   1   0
##         D   0  20   0 131   0  13   6  10   6   6   4   0   1   4  10  13
##         E   0   0  11   1  72   1   3   0   0   0   5   1   0   0   0   1
##         F   0   0   3   0   0 119   0   1   2   3   0   0   0   0   0  16
##         G   1   2   6   0  22   6 112   4   1   0   4   8   0   0   5   3
##         H   0   0   0   1   0   0   1  82   0   0   4   0   1   1   0   0
##         I   0   0   0   0   0   4   0   0 148   2   0   0   0   0   0   1
##         J   3   0   0   7   0   2   0   1   9 131   0   0   0   1   0   2
##         K   0   1  17   2  10   0   4  13   0   0 108   3   3   0   1   0
##         L   2   0   0   0   0   0   2   0   0   0   1 146   0   0   3   0
##         M   3   7   0   1   0   1   0   3   0   3   6   0 178   8   0   0
##         N   0   2   0   4   1   0   0   5   0   0   5   0   5 157   1   0
##         O   5   1   9   7   0   0   1  28   0   4   0   0   5   8 147  10
##         P   0   0   0   8   0  14   0   0   3   3   0   0   0   9   0 134
##         Q   1   1   1   0   8   0  19   6   1   5   0   2   1   0   5   1
##         R   0  12   0   7   7   3  13   9   2   7  17   2   1   2   2   0
##         S   2   5   4   6   9   5   6   2   2   7   0   3   1   0   0   0
##         T   0   0   0   2   0   6   0   2   0   0   0   0   0   0   0   0
##         U   0   0   2   0   2   1   0  12   0   0   0   0   0   1   2   0
##         V   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
##         W   0   1   2   0   0   0   6   6   0   0   2   0   4   4  10   9
##         X   5   3   0   0  31   2   0   3   5   0  10   1   0   0   0   0
##         Y   2   0   0   0   2   5   0   0   0   0   2   8   0   5   0   4
##         Z   1   0   1   0  11   0   2   0   0   0   0   0   0   0   0   0
##
## predicted   Q   R   S   T   U   V   W   X   Y   Z
##         A   0   0   8   0   0   0   0   0   0   0
##         B   8  16  13   2   1   0   0   8   1   4
##         C   6   0   0   0   1   0   0   0   0   0
##         D   0   8   5   0   1   0   0   4   4   1
##         E   1   5   2   4   3   0   0   2   0   7
##         F   0   0   2  15   0   4   4   0   5   0
##         G  14   1   1   0   1   0   0   0   0   1
##         H   0   0   2   0   0   0   0  12   0   0
##         I   0   0   4   5   0   0   0   0   2   0
```

```
##         J   2   0   1   0   0   0   0   0   0   2
##         K   0   2   0   2   2   0   0   8   0   1
##         L  11   0   1   0   0   0   0   0   0   0
##         M   1   9   0   0  13   2  13   1   1   0
##         N   0   2   0   5  15   7   4   0   0   0
##         O  16   3   3   3  11   1   4   4   1   0
##         P   0   0   0   1   0   2   0   0   1   0
##         Q  98   1   3   0   3   4   0   0   8   1
##         R   1 152  11   0   0   0   1   1   0   4
##         S   6   0 122   0   0   1   0   4   4  17
##         T   0   0   2 137   2   5   0   0  11   3
##         U   0   0   1   5 145   3   0   0   3   0
##         V   0   0   0  10   5 127   2   0  15   0
##         W   1   3   0   0   1  14 142   0   2   0
##         X   0   3  15   8   0   0   0 128   0   3
##         Y   1   0   1  11   2   4   0   9 120   0
##         Z   0   0   3   2   0   0   0   5   0 133
```

```
#2
mcr <- 1 - sum(diag(conf_tb))/5000
mcr
```

```
## [1] 0.3192
```

```
#3
df_conf_pred <- data.frame(pred = predicted)
df_conf_real <- data.frame(real = lettersdf$V1[-train])

df_conf_pred <- df_conf_pred %>%
  group_by(pred) %>%
  summarize(N = n())

df_conf_real <- df_conf_real %>%
  group_by(real) %>%
  summarize(N = n())

df_full <- inner_join(df_conf_pred, df_conf_real, by = c("pred" = "real"))

df_full <- df_full %>%
  mutate(rate = abs(N.x - N.y)/N.y)
```

```
## Warning: package 'bindrcpp' was built under R version 3.3.3
```

```
df_full[df_full$rate == max(df_full$rate), ]
```

```
## # A tibble: 1 x 4
##     pred   N.x   N.y     rate
##   <fctr> <int> <int>    <dbl>
## 1      B   285   184 0.548913
```

A note on problem 3. It is unclear here what being most difficult to predict means. I assumed that, out of the total number of each letter the method was presented with, the worse predicted would be that for which the frequency at which mistakes are made is higher. That is, the letter around which most error happens, which is not necessarily the same as being difficult to predict. B is also the letter for which the absolute number of errors made is highest, if we go with that interpretation.

4. In terms of letter pairs, BD, XE, EC seem to be particularly hard to discern. Also, several other letters

in combination with B are hard as well, validating our previous claim about the letter B.

Slow Learning