Ensemble Methods and Mixture Models

Maximilian Pfundstein (maxpf364)

27 November 2018

Contents

1 Ensemble Methods

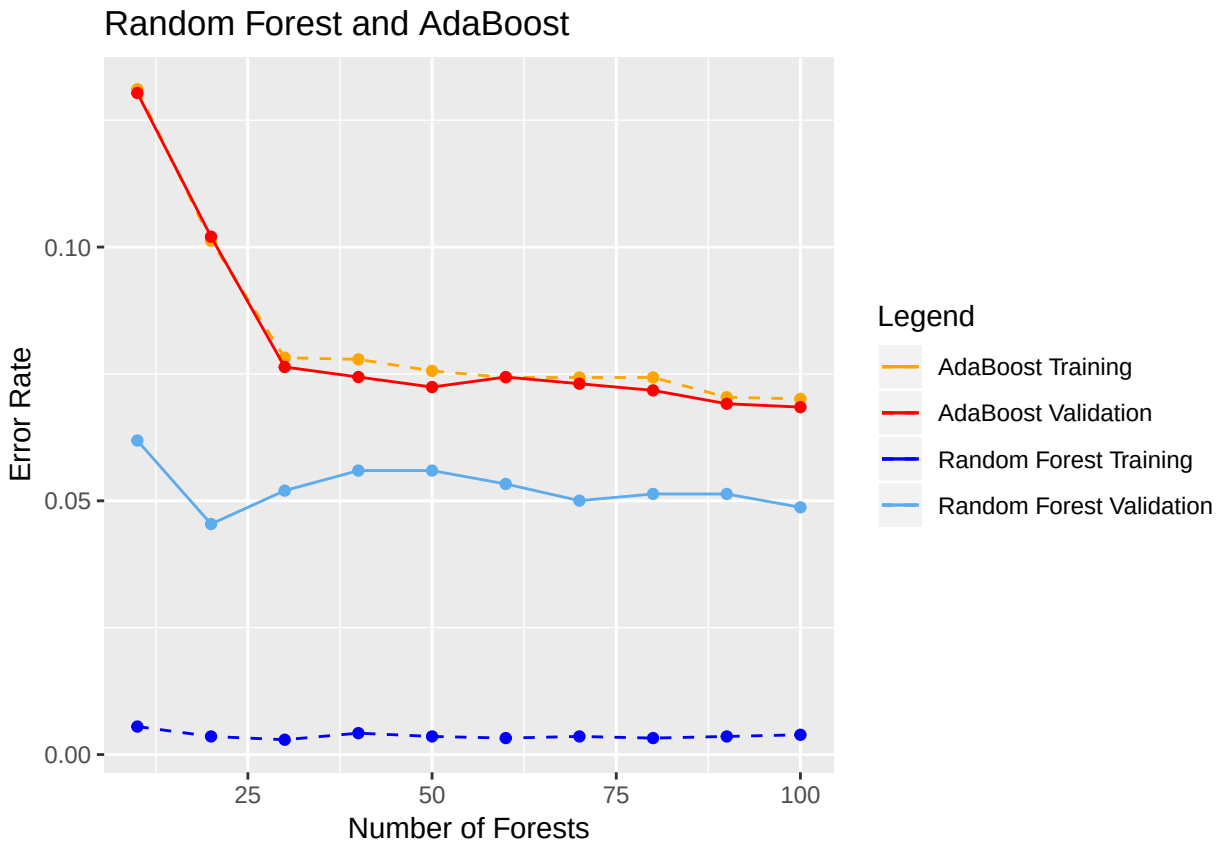
Let's load the dataset and have a look at it.

```
ggplot(adb_errors) +
  geom_line(aes(x = n, y = error_rate_training,
                colour = "AdaBoost Training"), linetype = "dashed") +
  geom_point(aes(x = n, y = error_rate_training), colour = "orange") +

  geom_line(aes(x = n, y = error_rate_validation,
                colour = "AdaBoost Validation")) +
  geom_point(aes(x = n, y = error_rate_validation), colour = "red") +

  geom_line(aes(x = n, y = error_rate_training,
                colour = "Random Forest Training"),
            data = rf_errors, linetype = "dashed") +
  geom_point(aes(x = n, y = error_rate_training),
             colour = "blue", data = rf_errors) +

  geom_line(aes(x = n, y = error_rate_validation,
                colour = "Random Forest Validation"), data = rf_errors) +
  geom_point(aes(x = n, y = error_rate_validation),
             colour = "steelblue2", data = rf_errors) +
  labs(title = "Random Forest and AdaBoost", y = "Error Rate",
       x = "Number of Forests", color = "Legend") +
  scale_color_manual(values = c("orange", "red", "blue", "steelblue2"))
```




```
print(rf_errors)
```

```
##      n error_rate_training error_rate_validation
## 1   10          0.005515899             0.06188282
## 2   20          0.003569111             0.04542462
## 3   30          0.002920182             0.05200790
## 4   40          0.004218040             0.05595787
## 5   50          0.003569111             0.05595787
## 6   60          0.003244646             0.05332456
## 7   70          0.003569111             0.05003292
## 8   80          0.003244646             0.05134957
## 9   90          0.003569111             0.05134957
## 10 100          0.003893576             0.04871626
```

```
print(adb_errors)
```

```
##      n error_rate_training error_rate_validation
## 1   10           0.13108371             0.13034891
## 2   20           0.10123297             0.10204082
## 3   30           0.07819598             0.07636603
## 4   40           0.07787151             0.07439105
## 5   50           0.07560026             0.07241606
## 6   60           0.07430240             0.07439105
## 7   70           0.07430240             0.07307439
## 8   80           0.07430240             0.07175774
## 9   90           0.07040883             0.06912442
## 10 100           0.07008436             0.06846610
```

2 Mixture Models

Appendix

```
knitr::opts_chunk$set(echo = TRUE)
library(mboost)
library(randomForest)
library(ggplot2)
library(knitr)
set.seed(1234567890)
spambase = read.csv("spambase.csv", sep=";", dec = ",")
spambase$Spam = as.factor(spambase$Spam)

n = dim(spambase)[1]
id = sample(1:n, floor(n*0.67))
train_spambase = spambase[id,]
val_spambase = spambase[-id,]

kable(head(spambase[,48:58]), caption = "spambase.csv")

# General Information
c_formula = Spam ~ .
tree_sizes = seq(from = 10, to = 100, by = 10)

# Random Forest
rf_errors = data.frame(n = numeric(), error_rate_training = numeric(),
                       error_rate_validation = numeric())

for (i in tree_sizes) {

  # Create the forest
  c_randomForest =
    randomForest(formula = c_formula, data = train_spambase, ntree = i)

  # Do the prediction on the validation dataset
  c_prediction_training =
    predict(object = c_randomForest, newdata = train_spambase)
  c_prediction_validation =
    predict(object = c_randomForest, newdata = val_spambase)

  # Get the error rate
  c_error_rate_training = 1 - sum(c_prediction_training ==
                                    train_spambase$Spam)/nrow(train_spambase)
  c_error_rate_validation = 1 - sum(c_prediction_validation ==
                                      val_spambase$Spam)/nrow(val_spambase)

  rf_errors = rbind(rf_errors,
                    list(n = i,
                         error_rate_training = c_error_rate_training,
                         error_rate_validation = c_error_rate_validation))
}
```

```
# AdaBoost
adb_errors = data.frame(n = numeric(), error_rate_training = numeric(),
                        error_rate_validation = numeric())

for (i in tree_sizes) {

  # Create the model
  c_adaBoost = blackboost(formula = c_formula,
                          data = train_spambase,
                          family = AdaExp(),
                          control=boost_control(mstop=i))

  # Do the prediction on the validation dataset
  c_prediction_training =
    predict(object = c_adaBoost, newdata = train_spambase, type = "class")
  c_prediction_validation =
    predict(object = c_adaBoost, newdata = val_spambase, type = "class")

  # Get the error rate
  c_error_rate_training = 1 - sum(c_prediction_training ==
                                    train_spambase$Spam)/nrow(train_spambase)
  c_error_rate_validation = 1 - sum(c_prediction_validation ==
                                      val_spambase$Spam)/nrow(val_spambase)

  adb_errors = rbind(adb_errors,
                     list(n = i,
                          error_rate_training = c_error_rate_training,
                          error_rate_validation = c_error_rate_validation))
}


ggplot(adb_errors) +
  geom_line(aes(x = n, y = error_rate_training,
                colour = "AdaBoost Training"), linetype = "dashed") +
  geom_point(aes(x = n, y = error_rate_training), colour = "orange") +

  geom_line(aes(x = n, y = error_rate_validation,
                colour = "AdaBoost Validation")) +
  geom_point(aes(x = n, y = error_rate_validation), colour = "red") +

  geom_line(aes(x = n, y = error_rate_training,
                colour = "Random Forest Training"),
            data = rf_errors, linetype = "dashed") +
  geom_point(aes(x = n, y = error_rate_training),
             colour = "blue", data = rf_errors) +

  geom_line(aes(x = n, y = error_rate_validation,
                colour = "Random Forest Validation"), data = rf_errors) +
  geom_point(aes(x = n, y = error_rate_validation),
             colour = "steelblue2", data = rf_errors) +
  labs(title = "Random Forest and AdaBoost", y = "Error Rate",
       x = "Number of Forests", color = "Legend") +
```

```
  scale_color_manual(values = c("orange", "red", "blue", "steelblue2"))
print(rf_errors)
print(adb_errors)
```

Bibliography