A Look Into The Random Forest Algorithm And It's Applications

Daerian Dilkumar, Guan Yu Chen, Derek Frempong, Zong Li

April 4, 2018

Contents

1 Introduction

This project was aimed at building and using an algorithm that will create predictions for classification and regression problems by using the idea of "random forests." The algorithm heavily relies on the aspect of "randomness" to reduce bias, since it attempts to randomly select sample spaces AND feature spaces from the original data, and does it hundreds of times over. The goal is to then to train a series of models and gain inputs from each, contributing to a final result. The algorithm does best at predicting the regressive variant for quantitative predictions.

A "random forest" is a collections of decision trees which each perform predictions, after which the forest will aggregate these or select a prediction which it will return. Each tree is trained over a bootstrapped sample of the training set, with a randomly selected set of variables to create it's feature space. This means that each tree has high bias, however since the forest is intended for use with hundreds of trees, the bias in the overall forest drops significantly lower than other models for this result.

Bootstrapping is the process by which we create a sample of the original data, which has the same size as the original data. This is made possible because the sampling is done with replacement. Therefore, the same observations can be used repeatedly, with a uniform distribution chance to be chosen.

2 Methodologies

The procedure is as follows:

- Select a number of trees for the forest, and how big the feature space is.
- For each tree, choose variables from the feature space, and get a bootstrap.
- Train the model for each tree with these variables for the given labels.
- Collect the trees into a forest to get them ready for predictions
- Finally, given testing data, use the forest to create as many predictions are there are trees.
- Aggregate the predictions for a final prediction for each observation in the testing data.

The libraries we used are as follows:

- tidyverse
- dplyr
- rpart
- rpart.plot

3 Development Process

4 Examples And Results

Note: We will be working with our functions and algorithms, which are stored in the file `RandomForest.R` so we will be first setting this source before we proceed.

4.1 Wine Data

We will create an example using Wine Data. This data performs well using a regression model. The data is formatted so that the quality is the last column of the data set. We will properly read and process the data so that we can run it through our random forest algorithm. To start we start at $M = 1$ (1 variable is chosen per tree), and $B = 500$ (500 trees).

```
# Read red wine and white wine data from their respective filess
redWineData = read_delim("winequality-red.csv", delim = ";")
whiteWineData = read_delim("winequality-white.csv", delim = ";")
```

```
# Remove NA values
redWineData = (redWineData[complete.cases(redWineData),])
whiteWineData = (whiteWineData[complete.cases(whiteWineData),])

# Prep Sets for merging library
redWineData = redWineData %>% mutate(Type = "Red")
whiteWineData = whiteWineData %>% mutate (Type = "White")

# Merge Data and get Final Dataset
wineData = rbind(redWineData,whiteWineData)

# Remove spaces from column names
names(wineData) = gsub(" ","_", names(wineData))
wineData$quality  = as.factor(wineData$quality)
wineData$Type = as.factor(wineData$Type)
wineData = wineData[,-ncol(wineData)]
```

```
training_set = sample_n(wineData, nrow(wineData)/2, replace=FALSE)
testing_set = anti_join(wineData,training_set)
```

```
## Joining, by = c("fixed_acidity", "volatile_acidity", "citric_acid", "residual_sugar", "chlorides", "
```

```
# labels for the training set
training_labels = as.numeric(unlist(training_set[ncol(training_set)]))

# labels for the testing set
testing_labels = as.numeric(unlist(testing_set[ncol(testing_set)]))

training_set = training_set[,-ncol(training_set)]
testing_set = testing_set[,-ncol(testing_set)]

# this is the number of trees we want to create
B = 500

# this is the number of variables we want to use
M = 1
f = PerformRegression(training_set,training_labels,testing_set,testing_labels,B,M)
```

```
## [1] "Results:"
## [1] "MSE = 45.7595132803216"
## [1] "R2 = -58.160212845936"
## [1] "Timings: "
##    user  system elapsed
##    9.79    0.06    9.89
```

The data above says that the accuracy is

Now, we try again with a higher number of variables. We find that with M = 3, the result changes.

4.2 Breast Cancer

Heres another example, but instead we will do classification on this dataset rather than regression. This data is is preprocessed already so we have to read in the data only.

```
BreastCancer = read.table("BreastCancer.csv",header=T, sep=",")
```

Now that we have the data we can then perform Classification on it and see how it fares.

```
training_set2 = sample_n(BreastCancer, nrow(BreastCancer)/2, replace=FALSE)
testing_set2 = anti_join(BreastCancer,training_set2)
```

```
## Joining, by = c("Cl.thickness", "Cell.size", "Cell.shape", "Marg.adhesion", "Epith.c.size", "Bare.nu
```

```
labelsBC = as.numeric(unlist(training_set2[ncol(training_set2)]))#training_set labels
labelsBC2 = as.numeric(unlist(testing_set2[ncol(testing_set2)]))# testing_set Labels
training_set2 = training_set2[,-ncol(training_set2)] #getting rid of the labels to  prepare
# toperform classification
testing_set2 = testing_set2[,-ncol(testing_set2)]
B2 = 500
M2 = 1
f = PerformClassification(training_set2,labelsBC,testing_set2,labelsBC2,B2,M2,BreastCancer)
```

```
## [1] "Results:"
## [1] "Loss = 0.469767441860465"
## [1] "Timings: "
##    user  system elapsed
##    3.14    0.00    3.17
```

5 Wrap-Up

6 Appendix

Note: The following libraries are used throughout the code: `tidyverse`, `dplyr`, `rpart`, and `rpart.plot`.

7 References

Gareth, J. *An Introduction to Statistical Learning: with Applications in R*. Springer.

Hastie, T. *The elements of statistical learning [electronic resource] : data mining, inference, and prediction* (2nd ed.). Springer.

Random forest. (2018). *En.wikipedia.org*. Retrieved 22 January 2018, from https://en.wikipedia.org/wiki/Random_forest