Random Forest Demonstration in R

Vinh Nguyen

Abstract—The purpose of this documentation is to demo how Random Forest algorithm works in R

Index Terms—Randome Forest, algorithm, R, bootstraping

1 DATASET

In this experiment, we use a very popular dataset for machine learning, that is, the IRIS dataset introduced by Fisher. This dataset contains 150 observations for three species of flowers (setosa, versicolor, and virginica) with 4 variables (Sepal.Length, Sepal.Width, Petal.Length, Petal.Width)

```
> summary(iris)
  Sepal.Length    Sepal.Width     Petal.Length    Petal.Width   
 Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100  
 1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300  
 Median :5.800   Median :3.000   Median :4.350   Median :1.300  
 Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199  
 3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800  
 Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500  
       Species  
 setosa    :50  
 versicolor:50  
 virginica :50  
```

2 DATA PROCESSING

We split our dataset in two parts: training part and testing part with the following commands
data_set_size <- floor(nrow(iris)/2)
indexes <- sample(1:nrow(iris), data_set_size)
training <- iris[indexes,]
test <- iris[-indexes,]

3 RUNNING ALGORITHM

rf_classifier <- randomForest(Species ~., data = training, ntree=100, mtry=2, importance=TRUE)

In the command above, we have 5 parameters. The first parameter (Species ~.) indicates that our target class is Species in which we need our algorithm have to classify, the dot (.) shows that we will put all other variables/features into training, data=training points out the training data which we created in the previous step. ntree = 100 shows that we will create 100 random trees, mtry = sqrt (features/variables) = sqrt (4) = 2, important = TRUE means that we want to output the important value of variables.

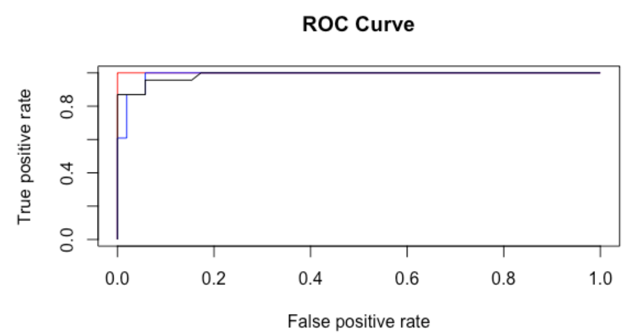
4 RESULT

```
        OOB estimate of  error rate: 5.33%
Confusion matrix:
           setosa versicolor virginica class.error
setosa         21          0         0  0.00000000
versicolor      0         25         2  0.07407407
virginica       0          2        25  0.07407407
```

The result shows that our out-of-bag error rate is 5.33% = 4/75 as depicted in the confusion matrix when training the model.

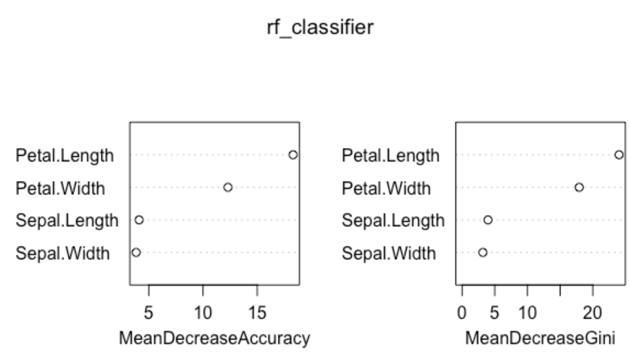
```
            predicted
observed     setosa versicolor virginica
  setosa         29          0         0
  versicolor      0         20         3
  virginica       0          1        22
```

The figure above shows the confusion matrix with testing data, the error rate is the same compared to training data.



The ROC curve:
 Red: setosa
 Blue: versicolor
 Black: virginica
It can quickly be seen that setosa is the ideal case with 100 accuracy, followed by virginica and versicolor.

The figure below shows the importance of each variable, it can be



seen that Petal.Length plays the most important role for classification task.

- *Vinh Nguyen. Email: vinh.nguyen@ttu.edu*