Classification: Bagging, Random Forest and Boosting

Ruoyuan Li (001223313)

Data Description & Question addressed

The data set I used is called "user knowledge Modelling Data set" which can be found on UCI-Machine Learning repository. It contain six variables and there are tonal 258 observations. Five out of six variables are predictor variables, they all ranges from 0 to 1 since original data may be scaled, namely:

- STG which is the degree of study time for goal object materials
- SCG which is the degree of repetition number of user for goal object materials
- STR The degree of study time of user for related objects with goal object
- LPR The exam performance of user for related objects with goal object
- PEG The exam performance of user for goal objects

The response variables is called UNS which is knowledge level of user(categorical variable). It has four different levels, "High","Middle","Low", and "Very low".

In this analysis report, bagging, random forest and boosting are using for building classification tree. Goal is building a suitable classification tree with good predicting performance. First building classification tree based on training data, and validating classification performance through test data. Since more than one method are using in this report, so it included a table which contain misclassified rate and corrected rand index for comparison purpose.

Bagging & Random Forest & Boosting and Comparison

Overall, first dividing data into train data which contain half of the observations and test data which contain another half of the observations. Also need to set seed to ensure to get same results. After setting train and test data, then starting with bagging, next move to random forest, and boosting at the end.

For bagging and random forest, using function **randomforest**, "mtry" is number of variables randomly sampled as candidates at each split. setting as 5 since there are 5 predictor variables in the data set, and the detail of the parameters setting as following:

```
bag.user=randomForest(UNS~.,data=user,subset=train,
                      mtry=5,importance=TRUE,type="class")
```

Out of bagging (OBB) rate is 9.3% which is acceptable for now. And predicting test data via this random forest object leads to result with misclassified rate is 6.98% which is acceptable and corrected rand index for agreement by chance is 0.844. From the variable importance plot of bagging, "PEG" (MeanDecreaseAccuracy equals to 140 and Gini equals to 60) and "LPR" (MeanDecreaseAccuracy equals to 50 and Gini equals to 20) are contributes more than other, they are ranked higher in usefulness than other variables for correctly classifying data.

Then move to random forest, with only change parameter "mtry" equals to 3. Again, OBB rate is 8.53% is better which decreasing a little than bagging. Predicting test data again based on this random forest object, ended with lower misclassified rate 5.6% and corrected rand index equals to 0.867. Then for variable importance, "PEG" and "LPR" ranking as more helpful than other, same results from bagging. We may interested in decrease parameter "mtry" again for better analysis. However, decreasing "mtry" to 2 leading a worse result. OBB rate is 9.3% and for prediction with misclassified rate 13.2%. From bagging and random forest, best results can get is with parameter "mtry" equals to 3 so far. Next moving to boosting.

Using **gbm** function for boosting purpose, formula and parameters setting as following:

```
boost.user=gbm(UNS~.,data=user[train,],distribution="multinomial",
               n.trees=3000,interaction.depth=4)
```

Unlike to bagging and random forest, "PEG" has largest relative influence which more than 80, other variables have lower than 10 relative influence. For missclassified rate of predicting test data based on this boosting method is 15.5%, this is the highest misclassified rate so far of the analysis. So we may need to adjust parameter to create a better result. Changing λ to 0.01 and keep interaction same as before. Also this time using **gbm.perf** for selecting trees for boosting, it suggests 230 trees. This new boosting ends with misclassified rate 16.3% and corrected rand index is 0.628. Even worse than before. The number of trees may be under estimate, so may increase the number of trees to 400 and do the analysis again. Also maybe need to change interaction depth as well. I put some

of the boosting results in the table. Even keep changing trees and other parameters, the results from boosting still not as good as expected. There are many options. However, the details do not address due limit pages. I used a table to showing all the results together for comparison purpose.

	Bagging	Random Forest	Random Forest
parameters			
	• ntree=500 • mtry=5	• ntree=500 • mtry=3	• ntree=500 • mtry=2
rand.index	0.844	0.867	0.705
corrected			
misclassified rate	6.98%	5.43%	13.2%

|	boosting	boosting	boosting
parameters			
	• tree=3000 • d=4 • $\lambda = 0.001$	• tree=230 • d=4 • $\lambda = 0.01$	• tree=158 • d=1 • $\lambda = 0.01$
rand.index	0.637	0.628	0.610
corrected			
misclassified rate	15.5%	16.3%	17.0%

In conclusion, random forest with 3 variables at each split has best result, it has lowest misclassifed rate equals to 5.43% and highest correct rand index equals to 0.867. It is a good result from classifcation. Also, from all the variable importance plot ans relative influence plot, we can conclude "PEG" is most helpful variable for correctly classifying data.