The 6th SISC Machine Learning Challenge – Accuracy Group 1

* Increase the number of decision trees used in random forest.

Because number of decision trees is equal to number of features selected. More features selected absolutely makes more comparison and more accurate

More trees mean more accuracy, but drawbacks are loss of time and memory efficiencies

* Increase the leaf (the end node) of the decision tree can increase the accuracy.

Large size of leaf nodes make the machine capture less of the noise of the training and testing data

* Increase the max feature

Max feature is an amount of features can be used in one decision tree. More features mean more diversities and more accuracy.

However, from L.Breiman (2001) said that max feature should be set for sqrt(n) because there’s no significant difference in more than that.

* Increase data sample amount

To increase the accuracy of the random forest algorithm, the sample amount of the data should be increased. Increasing the sample amount would allow for more variables to be selected, making a larger forest.

For example,

Badminton Expert A has a 0.8 accuracy, meaning that he has a 80% chance of a correct prediction.

Badminton Experts A, B, C, D, E each have an accuracy of 0.8 independent of other experts.

To calculate the probability that three or more experts are correct,

(5!/3!2!)(0.8)^3(0.2)^2 + (5!/4!1!)(0.8)^4(0.2)^1 + (5!/5!)(0.8)^5(0.2)^0 =0.94208

This example proves that by adding a larger sample size, the accuracy has increased by 0.14208 and is closer to 1.

* Filling the missing data would also increase the accuracy of the algorithm.

By adding some of the missing parameters for example, if there are some missing data, the accuracy will decrease.

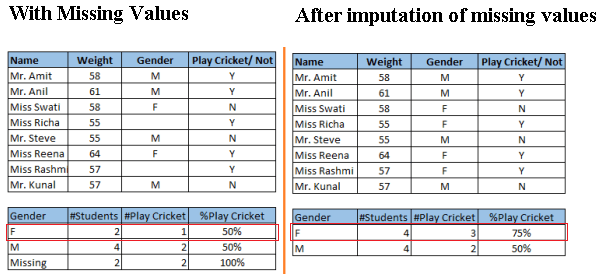


Figure 1: the chances of males and females playing cricket. (Sunil Ray, 2015)

The diagram above shows the chances of females playing cricket. With the missing values, the table suggest there the chances of playing cricket by males and females are equal. However, after imputation of the missing values, it can be seen that females have a higher chance at playing cricket in comparison to males.

http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

[https://link.springer.com/article/10.1007/s10994-006-6226-1](https://l.facebook.com/l.php?u=https%3A%2F%2Flink.springer.com%2Farticle%2F10.1007%2Fs10994-006-6226-1&h=ATOf3mBE3kgrVTbRoi93Byj4H5R0opeTp36XHpYt8UDfJWyAnektelLjVYdBDcvch-g9tFySw11zxUq9Eu-gMCcQt6i9W-DZSqyYnfLf6V8VgyK60_gwIMl99lQIwrSoGQh4UebEkhk4OQ)

The 6th SISC Machine Learning Challenge – Efficiency Group 1

* Basically the opposite of how to increase the accuracy because increasing means the program has to analyze more data, making the efficiency drops.
* Decrease data sample size/Select less features

To increase the efficiency of the random forest algorithm, there should be a lower sample size for the code to process making it quicker. This however lowers the accuracy of the random forest algorithm and a balance should be calculated where the sample size will maximise efficiency and accuracy.

* Treat the dataset better (treating values and cutting outliers.)
* Use the “%timeit” function which will find the fastest loop time.

EX:model = RandomForestRegressor(n\_estimator = 100,oob\_score = TRUE,n\_jobs = -1,random\_state =1) best loop time=1.7s/loop

EX:model = RandomForestRegressor(n\_estimator = 100,oob\_score = TRUE,n\_jobs = 1,random\_state =1) best loop time= 1.1s/loop

* Use random\_state parameter to help make a solution easy to replicate.
* Use the oop\_score function. This method tags every observation used in different tress and finds out the maximum vote score for every observation based on trees which did not use this particular observation to train itself.

EX:model = RandomForestRegressor(n\_estimator = 100, oob\_score = TRUE, n\_jobs = -1,random\_state =50,                                         max\_features = "auto", min\_samples\_leaf = 50)

* Decrease the number of trees

The 6th SISC Machine Learning Challenge – Scrutability Group 1

* Balance the decision trees to simplify the code for programmers
* Try not to bagging. Bagging can increase the accuracy but will decrease the scrutability.
* To make decision trees in random forest can be visualized, it can be exported to .dot file by export\_graphviz

1. If the decision trees are too large for graphviz, we can use this below method for finding conditions in condition nodeBinary search for the condition on the root node
2. Breadth first search by applying the value which will return false and which will return true. Use pointer to keep that which subtree connected to each answer
3. Binary search for the condition on each node.
4. Repeat 2.-3. Until reach the last leaf node.

* Do 1-4 for each tree.

http://scikit-learn.org/stable/modules/generated/sklearn.tree.export\_graphviz.html