CS412 Data Mining
Fall 2016

Assignment 4

Bangqi Wang
Dec. 4, 2016

Introduction
This project uses decision tree and random forest to predict the labels. The decision tree uses Gini index and the random forest uses random linear combinations.

Decision Tree
The project built decision tree according to the data sets. Each node in decision tree contains one attributes, and each edge stands for the value of this attribute. The decision tree decides the attribute with Gini index. The Gini index calculates the conditional probability of the attribute and label and selects the attribute that provides the largest reduction in impurity.

$$gini(D) = 1 - \sum_{i=1}^{n} p_j^2$$

$$gini_A(D) = \frac{|D_1|}{|D|} gini(D_1) + \frac{|D_2|}{|D|} gini(D_2)$$

(Equations from lecture note)

Implementation
The decision tree includes three files. One main file as DecisionTree.py, one tree class file as tree.py, and one node class file as node.py.

Node Class
The node class takes data table as input and calculate the necessary information according to the data table. To improve the runtime of calculating Gini index, the node.construct() function parses the table and stores all necessary information for future calculation. If the data table only contains one class or the data table only contains class information without any attribute, the node is a leaf node and node.label will save the prediction of this node. Otherwise, the function will calculate the Gini index and split the data table into according to the selected attribute.

Tree Class
The tree class takes original data table from the main file and constructs node with parsed data table. After constructing each node, the tree class will check the children list and uses the returned data table to construct children node. The tree class also has a testing function which can pass data from the root and get the label from the leaf. The decision tree will return the majority label so far if it meets unseen values or features.

Random Forest
Random forest uses a random linear combination of multiple decision trees. The program trains each decision tree with selected data sample and each decision tree calculates Gini index with randomly selected attributes. The prediction is the majority vote from all decision trees.

Module Evaluation

Balance

	Decision Tree			*Random Forest*		
Accuracy	58.22%			70.67%		

Class	**1**	**2**	**3**	**1**	**2**	**3**
Sensitivity	0.0	0.6471	0.6436	0.0	0.7549	0.8119
Specificity	0.8374	0.8130	0.6935	0.9803	0.7805	0.7177
Precision	0.0	0.7416	0.6311	0.0	0.7404	0.7009
Recall	0.0	0.6471	0.6436	0.0	0.7549	0.8119
F-1 Score	0.0	0.6911	0.6373	0.0	0.7476	0.7523
F-0.5 Score	0.0	0.7205	0.6335	0.0	0.7432	0.7206
F-2 Score	0.0	0.6640	0.6410	0.0	0.7520	0.7869

Nursery

	Decision Tree					*Random Forest*				
Accuracy	97.60%					97.66%				

Class	**1**	**2**	**3**	**4**	**5**	**1**	**2**	**3**	**4**	**5**
Sensitivity	0.968	0.746	0.979	1.0	0.0	0.979	0.623	0.979	1.0	0.0
Specificity	0.981	0.992	0.997	1.0	0.999	0.975	0.998	0.994	1.0	1.0
Precision	0.961	0.708	0.993	1.0	0.0	0.951	0.880	0.986	1.0	0.0
Recall	0.968	0.746	0.979	1.0	0.0	0.979	0.623	0.979	1.0	0.0
F-1 Score	0.964	0.727	0.986	1.0	0.0	0.956	0.730	0.982	1.0	0.0
F-0.5 Score	0.962	0.715	0.990	1.0	0.0	0.956	0.813	0.984	1.0	0.0
F-2 Score	0.967	0.738	0.981	1.0	0.0	0.973	0.662	0.980	1.0	0.0

Led

	Decision Tree		*Random Forest*	
Accuracy	85.80%		85.89%	
Class	**1**	**2**	**1**	**2**
Sensitivity	0.7806	0.8927	0.7806	0.8940
Specificity	0.8927	0.7806	0.8940	0.7806
Precision	0.7654	0.9008	0.7675	0.9009
Recall	0.7806	0.8927	0.7806	0.8940
F-1 Score	0.7729	0.8967	0.7740	0.8974
F-0.5 Score	0.7684	0.8992	0.7701	0.8995
F-2 Score	0.7775	0.8943	0.7780	0.8954

Poker

	Decision Tree		*Random Forest*	
Accuracy	63.13%		67.85%	
Class	**1**	**2**	**1**	**2**
Sensitivity	0.7952	0.2877	0.9956	0.0137
Specificity	0.2877	0.7952	0.0137	0.9956
Precision	0.7006	0.4013	0.6790	0.6000
Recall	0.7952	0.2877	0.9956	0.0137
F-1 Score	0.7449	0.3351	0.8074	0.0268
F-0.5 Score	0.7177	0.3719	0.7252	0.0628
F-2 Score	0.7743	0.3049	0.9107	0.0170

Parameter Selection

The program has several parameters that may affect the performance of the classification. The decision has no attributes because the runtime for the largest data set is less than one second and it is no need for limiting the tree depth. For the random forest, I created several parameters.

Sample Rate = 0.5

This parameter decides the size of sample data set for each decision tree. I set this parameter as 0.5. When I parse the data, there is 50% probability that the sample training data set will select this data. This parameter decides the probability for selecting each record, so the total length of sample data sets may be different for each decision tree. When the parameter is close to 1, the random forest may have similar predictions as a single decision tree. If the parameter is too small, the predictions of decision trees have too many difference and the total accuracy will decrease. Therefore, I chose 0.5 as the sample rate.

# of Decision Trees = 200

The parameter represents the number of decision trees in a random forest. As the number of decision trees increase, the accuracy will increase until the accuracy converges to a stable value. In this project, I used 200 decision trees in each random forest. The runtime for a single decision tree with the largest dataset is around 0.7 seconds. The total runtime for 200 decision tree with sampled data is no longer than one minute. I tried the random forest with 500 decision tree, and the total runtime for the largest data set is about 2 minutes, but the output accuracy has no significant difference with the output of random forest with 200 trees.

# of Attributes for Gini Index = $\sqrt{\#\ of\ remained\ attributes}$

The number of attributes uses to calculate the Gini index. The program will randomly select the attributes up to this limitation and calculate the Gini index with the attributes selected. For the projects, the datasets contain about 5~13 attributes, and the number of selected attributes is about 1~4. The decision tree will only use about 1/3 attributes to calculate Gini index. Therefore, each decision tree in the random forest may have a significant difference.

Results Analysis

The ensemble method, random forest, improves the performance of the basic classification, decision tree for all data sets.

Balance

The accuracy for this data set increases from 58% to 71% with the help of random forest. The accuracies for class 2 and 3 increase significantly, especially class 3. The precision for class 2 remains the same but the precision for class 3 increases from 0.64% to 70%. This means the random forest predicts class 3 more accurately. The specificity for class 1 increases from 84% to 98% and the random forest makes less confusion between other classes and class 1.

Nursery
The decision tree has high accuracy in this data set as 97.60%. Random forest only increases 0.06%. The accuracy increases for class 1, decreases for class 2, remain the same for class 3. The specificity for class 5 increases from 0.999 to 1.0. Random forest makes no confusion about class 5 and other classes. Although the sensitivity for class 2 decreases, the precision for class 2 increases from 71% to 89%. Random forest predicts class 2 much more accurately.

Led
The prediction from decision tree and random forest are almost the same. There is only one more correct prediction for class 2. Therefore, I believe the performances for decision tree and random forest are the same.

Poker
The accuracy increases from 63% to 68%. However, I believe that the performance of random forest is worse than the performance of decision tree. The accuracy for class 1increas from 80% to 99%. However, the specificity for class 1 decreases from 29% to 1%. In this case, random forest predicts most of the data as class 1. Indeed, the precision of class one only decreases from 70% to 68%. Therefore, I can conclude that the most of the data set is in class 1 and the random forest almost predicts every data as class 1. The performance may become worse if the data is balanced.