CS412 Assignment-4

Yuetong Liu
Department of Statistics

In this assignment, a general purpose classification framework was built from scratch. This framework contains two classification methods: a basic method and an ensemble method. More specifically, decision tree and random forest. Given certain training dataset following a specific data format, the classification framework generated a classifier, and use this classifier to assign labels to unseen test data instances. Last, 8 metrics were calculated to evaluated the performance of classification on test data.

1 Classification Methods Implementation

In this section, I will give a brief introduction of classification methods in the classification framework. It contains two specific parts: Decision Tree and Random Forest.

1.1 Decision Tree

A decision tree is a flowchart-like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (or terminal node) holds a class label. The topmost node in a tree is the root node.

Since assignment 4 only deals with categorical features, in each splitting node (internal node), we only consider whether the data belongs to certain category in certain attribute. According to the result, the data set can be split into two parts below this internal node. Thus, this decision tree is a binary tree with each node has two children nodes at most.

The implementation of decision tree follows the procedure of recursively partitioning. First, initialized the root node containing the whole training data set. Then find the most appropriate splitting rule and split the data set. Next, recursively apply the procedure on each child node. Last, predict each terminal node using within-node data.

As for the splitting rule, we used **Gini-index** as the attribute selection measure (the gini-index score is computed by considering every possible categorical value of the feature as a branch). The splitting rule with the most information gain was selected among all possible splitting attribute and category candidates. The Gini-index is defined as $gini(D) = 1 - \sum_{j=1}^{n} p_j^2$, where p_j is the relative frequency of class j in data set D.

There are three situations where a terminal node will be created. 1. If all samples for a given node belong to the same class; 2. There are no remaining attributes for further partitioning, that is, there are no splitting rules to make the information gain greater than 0; 3. There are no samples left.

1.2 Random Forest

Random forest is an ensemble method where a collection of decision tree classifiers are built. The individual decision trees are generated using a random selection of attributes at each node to determine the split. More formally, each tree depends on the the randomly sampling data from the original training data set using bootstrap method. During classification, each tree votes and the most popular class is returned.

Different from the decision tree, the individual classifier in random forest won't take all attributes for consideration in each splitting node. But just randomly select certain amount of attributes as splitting candidates to choose from. The number of attributes are regarded as a tunning parameter in this model.

There are two tunning parameters in this random forest method. One is the number of attributes selected from at each node. Here we choose the closet integer of square root of total number of attributes. Sometimes, 1 may do a good job but not in these cases. Another tunning parameter is the number of decision trees in random forest. After several attempts ranging from 100 to 1000, **400** will give the best result and won't take a long time to achieve the classification.

2 Model Evaluation

After applying both basic classification method and your ensemble classification method on each data set, the following model evaluation metrics were calculated using the output of classification methods for both training and test data sets.

For overall performance, the accuracy is used to evaluated. For each class, Sensitivity, Specificity, Precision, Recall, F-1 Score, F-score ($\beta = 0.5$ and $\beta = 2$) are calculated to measure the performance of model.

The results for 4 data sets are shown below.

1. **Balance data set**

	Decision Tree	Random Forest
Accuracy	0.7111	0.8311

Metrics	**Decision Tree**			**Random Forest**		
	Class 1	Class 2	Class 3	Class 1	Class 2	Class 3
Sensitivity	0	0.8037	0.7629	0.3333	0.8067	0.8585
Specificity	0.8922	0.8644	0.7891	0.9022	0.9434	0.9160
Precision	0	0.8037	0.7629	0.3333	0.8067	0.8585
Recall	0	0.8431	0.7327	0	0.9412	0.9010
F_1 score	0	0.8230	0.7475	0	0.8688	0.8792
$F_{\beta=0.5}$ score	0	0.8113	0.7566	0	0.8304	0.8667
$F_{\beta=2}$ score	0	0.8350	0.7385	0	0.9108	0.8922

2. **Nursery data set**

	Decision Tree	Random Forest
Accuracy	0.9922	0.9536

Metrics	**Decision Tree**				**Random Forest**			
	Class 1	Class 2	Class 3	Class 4	Class 1	Class 2	Class 3	Class 4
Sensitivity	0.9893	1	1	0	0.9101	1	0.9503	0.2
Specificity	0.9942	0.9987	1	1	0.9759	0.9822	0.9807	1
Precision	0.9893	1	1	0	0.9101	1	0.9503	0.2
Recall	0.9881	0.9538	1	0.2	0.9518	0.3385	0.9583	0.2
F_1 score	0.9887	0.9764	1	0	0.9305	0.5057	0.9543	0.2000
$F_{\beta=0.5}$ score	0.9891	0.9904	1	0	0.9182	0.7190	0.9519	0.2
$F_{\beta=2}$ score	0.9883	0.9627	1	0	0.9431	0.3901	0.9567	0.2

3. **Led data set**

	Decision Tree	Random Forest
Accuracy	0.8580	0.8642

Metrics	**Decision Tree**		**Random Forest**	
	Class 1	Class 2	Class 1	Class 2
Sensitivity	0.7654	0.9008	0.8012	0.8897
Specificity	0.9008	0.7654	0.8897	0.8012
Precision	0.7654	0.9008	0.8012	0.8897
Recall	0.7806	0.8927	0.7464	0.9170

F_1 score	0.7729	0.8967	0.7729	0.9031
$F_{\beta=0.5}$ score	0.7684	0.8992	0.7896	0.8950
$F_{\beta=2}$ score	0.7775	0.8943	0.7568	0.9114

4. **Poker data set**

	Decision Tree	Random Forest
Accuracy	0.6209	0.6814

Metrics	**Decision Tree**		**Random Forest**	
	Class 1	Class 2	Class 1	Class 2
Sensitivity	0.7104	0.4040	0.6838	0.5882
Specificity	0.4040	0.7104	0.5882	0.6838
Precision	0.7104	0.4040	0.6838	0.5882
Recall	0.7429	0.3653	0.9847	0.0457
F_1 score	0.7263	0.3837	0.8071	0.0847
$F_{\beta=0.5}$ score	0.7167	0.3956	0.7283	0.1742
$F_{\beta=2}$ score	0.7362	0.3724	0.9051	0.0560

3 Conclusion

From the results in four data set, we can see that, in most cases the ensemble method improves the performance of the basic classification method. Like only except the second data set Nursery, the overall accuracy of ensemble method is greater than the basic classification method. The decision tree model already predict well in nursery data set with the accuracy 0.9922. Perhaps due to the randomness, random forest may not achieve that accuracy.

Also, the ensemble method tend to do a better job in each class than the basic classification method. Take the first data set balance as example, in class 2 and class 3, all 7 metrics computed in random forest are better than decision tree. As for class 1, because the training data set is unbalanced data set, with only 7.14% of data labeled as class 1. So both decision tree and random forest didn't predict well on class 1. Thus, we can come to the conclusion that ensemble method tends to perform better than basic classification method, unless the basic classification method already did a great classification.