HW4

Liam Adams

December 11 2018

Test Accuracy

Table 1: Test Accuracy

	Decision Tree	Random Forest
balance.scale.test	.724	.773
led.test	.859	.862
nursery.test	.968	.985
synthetic.social.test	.464	.527

Classification Methods

For Decision Tree I implemented both full split and binary split, with Gini Index as the attribute selection method. I went with binary split in order to meet the accuracy threshold for synthetic social. I also limited the depth of the tree to 5 for synthetic social in order to meet the 3 minute runtime requirement. I did not limit the depth of the trees for the other 3 training sets.
For Random Forest I created 31 binary trees for each of the balance, led and nursery training sets. For balance I created the trees by randomly selecting 2 attributes as candidates to split for each node, for led 3 attributes, and for nursery 4 attributes. I used Gini Index to choose the attribute to split on at each node. I did not limit the depth of the trees for these 3 training sets. For synthetic social I created 7 binary trees, randomly selecting 20 attributes as split candidates at each node. The best attribute to split was selected using Gini Index. I limited the depth of the trees to 6 for synthetic social.

Decision Tree Test F1 Scores

Below are the F1 scores for each class of each test set for Decision Tree using

$$F1 = \frac{2TP}{2TP + FP + FN}$$

Table 2: Decision Tree Balance Test F1 Scores

Class Label	Score
1	0
2	.832
3	.763

Table 3: Decision Tree Led Test F1 Scores

Class Label	Score
1	.773
2	.898

Table 4: Decision Tree Nursery Test F1 Scores

Class Label	Score
1	.950
2	.861
3	.960
4	1

Table 5: Decision Tree Synthetic Test F1 Scores

Class Label	Score
1	.504
2	.409
3	.449
4	.465

Random Forest Test F1 Scores

Below are the F1 scores for each class of each test set for Random Forest using

$$F1 = \frac{2TP}{2TP + FP + FN}$$

Table 6: Random Forest Balance Test F1 Scores

Class Label	Score
1	0
2	.821
3	.816

Table 7: Random Forest Led Test F1 Scores

Class Label	Score
1	.773
2	.9

Table 8: Random Forest Nursery Test F1 Scores

Class Label	Score
1	.972
2	.96
3	.975
4	1

Table 9: Random Forest Synthetic Test F1 Scores

Class Label	Score
1	.586
2	.423
3	.518
4	.528

Training Accuracy

Table 10: Training Accuracy

	Decision Tree	Random Forest
balance.scale.train	1	1
led.train	.86	.858
nursery.train	1	1
synthetic.social.train	.503	.606

Decision Tree Training F1 Scores

Below are the F1 scores for each class of each training set for Decision Tree using

$$F1 = \frac{2TP}{2TP + FP + FN}$$

Table 11: Decision Tree Balance Training F1 Scores

Class Label	Score
1	1
2	1
3	1

Table 12: Decision Tree Led Training F1 Scores

Class Label	Score
1	.770
2	.90

Table 13: Decision Tree Nursery Training F1 Scores

Class Label	Score
1	1
2	1
3	1
4	1
5	1

Table 14: Decision Tree Synthetic Training F1 Scores

Class Label	Score
1	.492
2	.461
3	.509
4	.554

Random Forest Training F1 Scores

Below are the F1 scores for each class of each training set for Random Forest using

$$F1 = \frac{2TP}{2TP + FP + FN}$$

Table 15: Random Forest Balance Training F1 Scores

Class Label	Score
1	1
2	1
3	1

Table 16: Random Forest Led Training F1 Scores

Class Label	Score
1	.767
2	.9

Table 17: Random Forest Nursery Training F1 Scores

Class Label	Score
1	1
2	1
3	1
4	1
5	1

Table 18: Random Forest Synthetic Training F1 Scores

Class Label	Score
1	.616
2	.585
3	.654
4	.658

Choice of Parameters

In Decision Tree I did not parameterize balance scale, led, or nursery and let the trees grow to completion. I chose to do this because the algorithm built the trees very quickly (less than a second). For synthetic social I limited the depth of the tree to 5. I chose to limit the depth to meet the 3 minute time limit, and also because there was only a slight decrease in accuracy between the depth limited tree and a full tree (less than 10 percent).
For Random Forest I chose 31 trees each for balance scale, led, and nursery. I chose 31 because it seemed like a big enough number for one class to reach a convincing majority when the trees vote. For balance scale the trees randomly select 2 attributes as split candidates for each node, for led the number is 3, and for nursery 4. I chose these numbers as they are all about 50 percent of the total attributes of their respective training sets, and felt choosing less would leave too little data to split on.
For synthetic social Random Forest I chose to build 7 trees, limiting the depth to 6, and randomly select 20 attributes as split candidates at each node. I went through many permutations of these 3 parameters to find the combination that would run in under 3 minutes and produce an accuracy of .5. It appeared that the number of trees and depth were most important for accuracy while reducing the number of attributes to split on did not have as large of an effect. In the end I tried to maximize the number of trees and depth and sacrificed the number of attributes to meet the runtime standard.

Conclusion

I think the ensemble method definitely improves the performance of my basic decision tree classification. It improved the accuracy for each of the 4 data sets, although the improvements were very small for led and nursery. In ensemble, since each tree is slightly different due to sampling and random attribute selection, it may remove some noise and bias present in the original training set and allow the pool of trees to converge on the most important factors for classification. The biggest improvement was in synthetic social, I think the biggest improvements for ensemble may be in data that have a lot of attributes, because many of the attributes may be irrelevant to the classification task. Randomly

selecting a subset may aid in reducing the influence of noisy attributes.