CS 435 Artificial Intelligence: Homework 4

Results and Analytical Questions

Due: Friday December 4, 2015

Li-Yi Lin / llin34@jhu.edu
Chenhao Han / charleshan@jhu.edu

1. Decision Tree:

1) Report:

Data Set: Congressional Voting Records

Settings:

- Training Ratio = 0.6
- Pruning threshold = 0.05

Without Pruning

Metric	Information Gain		Information Gain Ratio	
Datasets	Training	Testing	Training	Testing
Accuracy	1.0	0.9425	1.0	0.9482
Precision	Label 0: 1.0 Label 1: 1.0	Label 0: 0.92 Label 1: 0.9595	Label 0: 1.0 Label 1: 1.0	Label 0: 0.9444 Label 1: 0.9509
Recall	Label 0: 1.0 Label 1: 1.0	Label 0: 0.9452 Label 1: 0.9405	Label 0: 1.0 Label 1: 1.0	Label 0: 0.9315 Label 1: 0.9603

With Pruning: Threshold = 0.05

Metric	Information Gain		Information Gain Ratio	
Datasets	Training	Testing	Training	Testing
Accuracy	0.9923	0.9482	1.0	0.9483
Precision	Label 0: 1.0 Label 1: 0.9881	Label 0: 0.9324 Label 1: 0.96	Label 0: 1.0 Label 1: 1.0	Label 0: 0.9444 Label 1: 0.9509
Recall	Label 0: 0.978 Label 1: 1.0	Label 0: 0.9452 Label 1: 0.9505	Label 0: 1.0 Label 1: 1.0	Label 0: 0.9315 Label 1: 0.9604

Data set: MONKS Problems data 1

Without pruning

Metric	Information Gain		Information Gain Ratio	
Datasets	Training	Testing	Training	Testing
Accuracy	1.0	0.8287	1.0	0.8194
Precision	Label 0: 1.0 Label 1: 1.0	Label 0: 0.8446 Label 1: 0.8141	Label 0: 1.0 Label 1: 1.0	Label 0: 0.8349 Label 1: 0.8053
Recall	Label 0: 1.0 Label 1: 1.0	Label 0: 0.8055 Label 1: 0.8518	Label 0: 1.0 Label 1: 1.0	Label 0: 0.7962 Label 1: 0.8426

With pruning: Threshold = 0.05

Metric	Information Gain		Information Gain Ratio	
Datasets	Training	Testing	Training	Testing
Accuracy	0.8225	0.8055	0.8226	0.8055
Precision	Label 0: 0.7381 Label 1: 1.0	Label 0: 0.7391 Label 1: 0.9231	Label 0: 0.7381 Label 1: 1.0	Label 0: 0.7391 Label 1: 0.9444
Recall	Label 0: 1.0 Label 1: 0.645	Label 0: 0.9444 Label 1: 0.6666	Label 0: 1.0 Label 1: 0.645	Label 0: 0.9231 Label 1: 0.6666

Data set: MONKS Problems data 2

Without pruning

Metric	Information Gain		Information Gain Ratio	
Datasets	Training	Testing	Training	Testing
Accuracy	1.0	0.6921	1.0	0.6967
Precision	Label 0: 1.0 Label 1: 1.0	Label 0: 0.7896 Label 1: 0.5279	Label 0: 1.0 Label 1: 1.0	Label 0: 0.8023 Label 1: 0.5325
Recall	Label 0: 1.0 Label 1: 1.0	Label 0: 0.7379 Label 1: 0.5986	Label 0: 1.0 Label 1: 1.0	Label 0: 0.7276 Label 1: 0.6338

With pruning: Threshold = 0.05

Metric	Information Gain		Information Gain Ratio	
Datasets	Training	Testing	Training	Testing
Accuracy	0.6213	0.6712	0.6213	0.6712
Precision	Label 0: 0.6213 Label 1: 0.0	Label 0: 0.6712 Label 1: 0.0	Label 0: 0.6213 Label 1: 0.0	Label 0: 0.6712 Label 1: 0.0
Recall	Label 0: 1.0 Label 1: 0.0	Label 0: 1.0 Label 1: 0.0	Label 0: 1.0 Label 1: 0.0	Label 0: 1.0 Label 1: 0.0

Data set: MONKS Problems data 3

Without pruning

Metric	Information Gain		Information Gain Ratio	
Datasets	Training	Testing	Training	Testing
Accuracy	1.0	0.9444	1.0	0.9027
Precision	Label 0: 1.0 Label 1: 1.0	Label 0: 0.9091 Label 1: 0.9811	Label 0: 1.0 Label 1: 1.0	Label 0: 0.8648 Label 1: 0.9428
Recall	Label 0: 1.0 Label 1: 1.0	Label 0: 0.9804 Label 1: 0.9123	Label 0: 1.0 Label 1: 1.0	Label 0: 0.9411 Label 1: 0.8684

With pruning: Threshold = 0.05

Metric	Information Gain		Information Gain Ratio	
Datasets	Training	Testing	Training	Testing
Accuracy	1.0	1.0	1.0	1.0
Precision	Label 0: 1.0 Label 1: 1.0	Label 0: 1.0 Label 1: 1.0	Label 0: 1.0 Label 1: 1.0	Label 0: 1.0 Label 1: 1.0
Recall	Label 0: 1.0 Label 1: 1.0	Label 0: 1.0 Label 1: 1.0	Label 0: 1.0 Label 1: 1.0	Label 0: 1.0 Label 1: 1.0

Data set: Iris
Settings:

- Training Ratio = 0.6
- Pruning threshold = 0.05

Without pruning

Metric	Information Gain		Information Gain Ratio	
Datasets	Training	Testing	Training	Testing
Accuracy	0.9666	0.9333	0.9666	0.95
Precision	Label 0: 1.0 Label 1: 0.9642 Label 2: 0.9259	Label 0: 1.0 Label 1: 0.84 Label 2: 1.0	Label 0: 1.0 Label 1: 0.9642 Label 2: 0.9259	Label 0: 1.0 Label 1: 0.875 Label 2: 1.0
Recall	Label 0: 1.0 Label 1: 0.9310 Label 2: 0.9615	Label 0: 1.0 Label 1: 1.0 Label 2: 1.0	Label 0: 1.0 Label 1: 0.9310 Label 2: 0.9615	Label 0: 1.0 Label 1: 1.0 Label 2: 0.875

With pruning: Threshold = 0.05

Metric	Information Gain		Information Gain Ratio	
Datasets	Training	Testing	Training	Testing
Accuracy	0.9666	0.9333	0.9666	0.95
Precision	Label 0: 1.0 Label 1: 0.9642 Label 2: 0.9259	Label 0: 1.0 Label 1: 0.84 Label 2: 1.0	Label 0: 1.0 Label 1: 0.9642 Label 2: 0.9259	Label 0: 1.0 Label 1: 0.875 Label 2: 1.0
Recall	Label 0: 1.0 Label 1: 0.9310 Label 2: 0.9615	Label 0: 1.0 Label 1: 1.0 Label 2: 0.8333	Label 0: 1.0 Label 1: 0.9310 Label 2: 0.9615	Label 0: 1.0 Label 1: 1.0 Label 2: 0.875

2) Discussion:

a) Difference between using information gain and information gain ratio:
   When using information gain ratio, the decision tree is more likely to overfit on data, so the testing accuracy will be relatively lower than the training accuracy. The reason is that we consider more aspects of data when introducing information gain ratio, so it will more likely to be overfitting. Also, comparing to not using information gain ratio, the accuracy of testing data is higher when attributes are continuous value.
b) Difference between using pruning and not using pruning:

When the maximum information gain or information gain ratio is smaller than the pruning threshold (0.05), the decision tree will stop expanding subtrees. It, instead, returns the most common label in the given examples in the current node. The effectiveness of pruning depends on the setting of pruning threshold. If the pruning threshold is too high, the decision will stop too early and the accuracy will become lower. On the other hand, the pruning will not have much influence on the decision tree if the pruning threshold is too low, making the decision tree more likely overfit the training data.

2. Naive Bayes:

1) Report:

Data Set: Congressional Voting Records

Settings:

- Training Ratio = 0.6

Datasets	Training	Testing
Accuracy	0.8965	0.8965
Precision	Label 0: 0.8148 Label 1: 0.9542	Label 0: 0.8873 Label 1: 0.9029
Recall	Label 0: 0.9263 Label 1: 0.8795	Label 0: 0.8630 Label 1: 0.9207

MONKS Problems data 1

Datasets	Training	Testing
Accuracy	0.7984	0.7129
Precision	Label 0: 0.7466 Label 1: 0.8775	Label 0: 0.6932 Label 1: 0.7371
Recall	Label 0: 0.9032 Label 1: 0.6935	Label 0: 0.7638 Label 1: 0.6620

MONKS Problems data 2

Datasets	Training	Testing
Accuracy	0.6390	0.6157
Precision	Label 0: 0.6571 Label 1: 0.8761	Label 0: 0.6703 Label 1: 0.3235
Recall	Label 0: 0.5517 Label 1: 0.25	Label 0: 0.8413 Label 1: 0.1549

MONKS Problems data 3

Datasets	Training	Testing
Accuracy	0.9344	0.9722
Precision	Label 0: 0.95 Label 1: 0.9193	Label 0: 0.9444 Label 1: 1.0
Recall	Label 0: 0.9193 Label 1: 0.95	Label 0: 1.0 Label 1: 0.9473

Data set: Iris

Settings:

- Training Ratio = 0.6

Datasets	Training	Testing
Accuracy	0.9333	0.95
Precision	Label 0: 1.0 Label 1: 0.8709 Label 2: 0.9166	Label 0: 1.0 Label 1: 0.875 Label 2: 1.0
Recall	Label 0: 1.0 Label 1: 0.9310 Label 2: 0.8461	Label 0: 1.0 Label 1: 1.0 Label 2: 0.857

3. Neural Network

1) Report:

Settings:

- Training ratio = 0.6
- Number of hidden layer nodes = 2 * number of input layer nodes
- Trainging iteration = 5000 (feed forward and backpropagation)
- Learning rate: 0.01
- Deep neural network layers: 3 hidden layers
- Weight initialization methods:
    - default: [-0.12, 0.12]
    - shallow: [-1/n^0.5, 1/n^0.5], where n is the number of nodes in previous layer
    - deep: [-6^0.5/(n+m)^0.5, 6^0.5/(n+m)^0.5], where n and m are the numbers of nodes in the previous and next layers respectively

Data Set: Congressional Voting Records

Without momentum

Weight	Default: [-0.12, 0.12]		Shallow		Deep	
Datasets	Training	Testing	Training	Testing	Training	Testing
Accuracy	0.9923	0.9482	1.0	0.9597	1.0	0.9425
Precision	Label 0: 0.9894 Label 1: 0.9939	Label 0: 0.9324 Label 1: 0.9452	Label 0: 1.0 Label 1: 1.0	Label 0: 0.9230 Label 1: 0.9895	Label 0: 1.0 Label 1: 1.0	Label 0: 0.92 Label 1: 0.9595
Recall	Label 0: 0.9894 Label 1: 0.9939	Label 0: 0.96 Label 1: 0.9504	Label 0: 1.0 Label 1: 1.0	Label 0: 0.9863 Label 1: 0.9405	Label 0: 1.0 Label 1: 1.0	Label 0: 0.9452 Label 1: 0.9405

With momentum (0.8)

Weight	Default: [-0.12, 0.12]		Shallow		Deep	
Datasets	Training	Testing	Training	Testing	Training	Testing
Accuracy	0.9961	0.9367	1.0	0.9483	0.9961	0.9482
Precision	Label 0: 1.0 Label 1: 0.994	Label 0: 0.9189 Label 1: 0.95	Label 0: 1.0 Label 1: 1.0	Label 0: 0.9211 Label 1: 0.9694	Label 0: 1.0 Label 1: .994	Label 0: 0.9324 Label 1: 0.96
Recall	Label 0: 0.9894 Label 1: 1.0	Label 0: 0.9315 Label 1: 0.9406	Label 0: 1.0 Label 1: 1.0	Label 0: 0.9589 Label 1: 0.9406	Label 0: 0.9894 Label 1: 1.0	Label 0: 0.9452 Label 1: 0.9505

Data set: MONKS Problems data 1

Without momentum

Weight	Default: [-0.12, 0.12]		Shallow		Deep	
Datasets	Training	Testing	Training	Testing	Training	Testing
Accuracy	1.0	1.0	1.0	1.0	1.0	1.0
Precision	Label 0: 1.0 Label 1: 1.0	Label 0: 1.0 Label 1: 1.0	Label 0: 1.0 Label 1: 1.0	Label 0: 1.0 Label 1: 1.0	Label 0: 1.0 Label 1: 1.0	Label 0: 1.0 Label 1: 1.0
Recall	Label 0: 1.0 Label 1: 1.0	Label 0: 1.0 Label 1: 1.0	Label 0: 1.0 Label 1: 1.0	Label 0: 1.0 Label 1: 1.0	Label 0: 1.0 Label 1: 1.0	Label 0: 1.0 Label 1: 1.0

With momentum (0.8)

Weight	Default: [-0.12, 0.12]		Shallow		Deep	
Datasets	Training	Testing	Training	Testing	Training	Testing
Accuracy	1.0	1.0	1.0	1.0	1.0	0.9907
Precision	Label 0: 1.0 Label 1: 1.0	Label 0: 1.0 Label 1: 1.0	Label 0: 1.0 Label 1: 1.0	Label 0: 1.0 Label 1: 1.0	Label 0: 1.0 Label 1: 1.0	Label 0: 1.0 Label 1: 0.9818
Recall	Label 0: 1.0 Label 1: 1.0	Label 0: 1.0 Label 1: 1.0	Label 0: 1.0 Label 1: 1.0	Label 0: 1.0 Label 1: 1.0	Label 0: 1.0 Label 1: 1.0	Label 0: 0.9814 Label 1: 1.0

Data set: MONKS Problems data 2

Without momentum

Weight	Default: [-0.12, 0.12]		Shallow		Deep	
Datasets	Training	Testing	Training	Testing	Training	Testing
Accuracy	0.8639	0.7175	0.8934	0.7037	0.8698	0.7476
Precision	Label 0: 0.8534 Label 1: 0.8867	Label 0: 0.7857 Label 1: 0.5724	Label 0: 0.9393 Label 1: 0.8285	Label 0: 0.8319 Label 1: 0.5372	Label 0: 0.832 Label 1: 0.9772	Label 0: 0.7608 Label 1: 0.6941
Recall	Label 0: 0.9428 Label 1: 0.7343	Label 0: 0.7965 Label 1: 0.5563	Label 0: 0.8857 Label 1: 0.9062	Label 0: 0.7 Label 1: 0.7112	Label 0: 0.9904 Label 1: 0.6718	Label 0: 0.9103 Label 1: 0.4154

With momentum (0.8)

Weight	Default: [-0.12, 0.12]		Shallow		Deep	
Datasets	Training	Testing	Training	Testing	Training	Testing
Accuracy	0.9822	0.7847	0.9704	0.7222	0.9349	0.7431
Precision	Label 0: 0.9903 Label 1: 0.9692	Label 0: 0.8661 Label 1: 0.6503	Label 0: 0.9807 Label 1: 0.9538	Label 0: 0.8512 Label 1: 0.5578	Label 0: 0.9051 Label 1: 1.0	Label 0: 0.8013 Label 1: 0.6148
Recall	Label 0: 0.9809 Label 1: 0.9843	Label 0: 0.8034 Label 1: 0.7464	Label 0: 0.9714 Label 1: 0.9687	Label 0: 0.7103 Label 1: 0.7464	Label 0: 1.0 Label 1: 0.8281	Label 0: 0.8206 Label 1: 0.5845

Data set: MONKS Problems data 3

Without momentum

Weight	Default: [-0.12, 0.12]		Shallow		Deep	
Datasets	Training	Testing	Training	Testing	Training	Testing
Accuracy	0.9918	0.8773	1.0	0.8935	1.0	0.8796
Precision	Label 0: 0.9841 Label 1: 1.0	Label 0: 0.8479 Label 1: 0.9069	Label 0: 1.0 Label 1: 1.0	Label 0: 0.8526 Label 1: 0.9375	Label 0: 1.0 Label 1: 1.0	Label 0: 0.8619 Label 1: 0.8963
Recall	Label 0: 1.0 Label 1: 0.9833	Label 0: 0.9019 Label 1: 0.8552	Label 0: 1.0 Label 1: 1.0	Label 0: 0.9362 Label 1: 0.8552	Label 0: 1.0 Label 1: 1.0	Label 0: 0.8872 Label 1: 0.8728

With momentum (0.8)

Weight	Default: [-0.12, 0.12]		Shallow		Deep	
Datasets	Training	Testing	Training	Testing	Training	Testing
Accuracy	1.0	0.8726	1.0	0.8865	1.0	0.9051
Precision	Label 0: 1.0 Label 1: 1.0	Label 0: 0.8341 Label 1: 0.9138	Label 0: 1.0 Label 1: 1.0	Label 0: 0.8475 Label 1: 0.9282	Label 0: 1.0 Label 1: 1.0	Label 0: 0.8721 Label 1: 0.9389
Recall	Label 0: 1.0 Label 1: 1.0	Label 0: 0.9117 Label 1: 0.8377	Label 0: 1.0 Label 1: 1.0	Label 0: 0.9264 Label 1: 0.8508	Label 0: 1.0 Label 1: 1.0	Label 0: 0.9362 Label 1: 0.8771

Data set: Iris

Without momentum

Weight	Default: [-0.12, 0.12]		Shallow		Deep	
Datasets	Training	Testing	Training	Testing	Training	Testing
Accuracy	0.9888	0.9666	0.9888	0.9666	0.9888	0.9333
Precision	Label 0: 1.0 Label 1: 1.0 Label 2: 0.9629	Label 0: 1.0 Label 1: 0.9130 Label 2: 1.0	Label 0: 1.0 Label 1: 1.0 Label 2: 0.9629	Label 0: 1.0 Label 1: 0.9130 Label 2: 1.0	Label 0: 1.0 Label 1: 1.0 Label 2: 0.9629	Label 0: 1.0 Label 1: 0.84 Label 2:
Recall	Label 0: 1.0 Label 1: 0.9655 Label 2: 1.0	Label 0: 1.0 Label 1: 1.0 Label 2: 0.9166	Label 0: 1.0 Label 1: 0.9655 Label 2: 1.0	Label 0: 1.0 Label 1: 1.0 Label 2: 0.9166	Label 0: 1.0 Label 1: 0.9655 Label 2: 1.0	Label 0: 1.0 Label 1: 1.0 Label 2: 0.8333

With momentum (0.8)

Weight	Default: [-0.12, 0.12]		Shallow		Deep	
Datasets	Training	Testing	Training	Testing	Training	Testing
Accuracy	0.9888	0.8833	0.9777	0.9833	1.0	0.9166
Precision	Label 0: 1.0 Label 1: 0.9666 Label 2: 1.0	Label 0: 1.0 Label 1: 0.75 Label 2: 1.0	Label 0: 1.0 Label 1: 1.0 Label 2: 0.9285	Label 0: 1.0 Label 1: 0.9545 Label 2: 1.0	Label 0: 1.0 Label 1: 1.0 Label 2: 1.0	Label 0: 1.0 Label 1: 0.8077 Label 2: 1.0
Recall	Label 0: 1.0 Label 1: 1.0 Label 2: 0.9615	Label 0: 1.0 Label 1: 1.0 Label 2: 0.7083	Label 0: 1.0 Label 1: 0.931 Label 2: 1.0	Label 0: 1.0 Label 1: 1.0 Label 2: 0.9583	Label 0: 1.0 Label 1: 1.0 Label 2: 1.0	Label 0: 1.0 Label 1: 1.0 Label 2: 0.7916

2) Discussion:

a) Difference between weight initialization methods:
Because the deep neural network has one more factor, number of layers, that is different from default and shallow neural network, we will discuss the difference between the default and the shallow methods only. The outcome shows that it has better results when using the shallow method. The reason might be that it takes more consideration of the node size of each neural network layer into account. By doing so, therefore, the neural network will have a better starting weights for each layer and find a better result within the given fixed number of iterations.

b) Influence of introducing momentum term:
When introducing the momentum term, we keep track of weight in previous iteration and contribute to updating current weight, therefore, the network will converge faster than before. As a result, given same number of iterations, the network using momentum term results in higher accuracy.

4. Summary:

The metric we used here is accuracy of testing data.

For each different method, we compute average testing accuracy,called performance in the following discussion, which averaging over all different parameter settings.

1) For Congressional Voting Records Dataset:
The best performance for congress dataset is getting from neural network and the worst one goes to Naive Bayes.
2) For MONKS problems dataset:
The best performance for this dataset is getting from neural network. The worst one is getting from the Naive Bayes.
3) For Iris dataset:
Both the Naive Bayes and decision tree get the best performance on the Iris data set. We think the reason for this result is because we floor the continuous data for both the Naive Bayes and decision tree.

We think the nature of the dataset consists of the number of features, range of each feature value, data type (continuous or discrete value). Based on these properties of each dataset, we choose accuracy as the metric to evaluate which algorithm is best, because accuracy tend to be a general way for evaluation when properties of these datasets are relatively similar.

We have observed the following phenomenon from the results:

- From the results above, we observed that Iris dataset prefer Naive Bayes and decision tree to Neural Network, which usually expected to be the best. The reason we concluded is that we floor the continuous value before training decision tree and naive bayes while provide raw data for neural network. Among all methods of preprocessing the continuous data, flooring, ceiling, and rounding, we found that flooring is the best for iris dataset. Also, this preprocessing does not affect the way decision tree and naive bayes makes its judgment. As a result, after preprocessed the iris data with flooring, the performance of decision tree and naive bayes prevail the neural network.

- Based on the results of congressional and monks datasets using decision tree method, we also observe that, for simple data set, it is more likely for decision tree overfits on dataset. When this happen, the accuracy of training set is higher than that of testing set. As the complexity increased, for example the Monk2 dataset, the decision tree (with pruning) will less likely to overfit on dataset.
- For preventing overfitting, we know pruning is a good method to try on decision tree. We indeed observed that pruning had prevented the decision tree from overfitting since the accuracy for testing data set is better than that of training data set, especially for the monks problems data 2. However, the nature of the dataset is too simple, namely the attribute number and attribute value for each attribute are not too complex, the decision tree can learn the classification rule for the dataset perfectly. It is not necessarily better to apply pruning on simple data.