HW 4 Artificial Intelligence

Justin Chung jchung51

Ryan Demo rdemo1

Notes

Justin worked on the descision tree and the naive bayes implementations. Ryan worked on the neural network, trying a range of iterative and matrix-based implementations to find the best one. Our only lingering error is that our pruning does not work.

Algorithm types:

- decision_tree
- naive_bayes
- neural_network

Decision Tree

Training includes recursive call to add to decision tree. We go through every attribute and caculate the gain which includes finding the entropy. This returns the most likely value for an instance. Ran into a bug with pruning, and could only get non-pruning to function correctly.

To call information gain ratio for training:

`python classify.py --data [train_file] --mode train --model-file name.model --algorithm [algorithm_type] --gain ratio`

Accuracy Summary

Data	gain	gainRatio
House Votes	64/87	63/87
Iris	23/30	12/30
Monks1	374/432	264/432
Monks2	272/432	297/432
Monks3	380/432	266/432

Cases

```
house-votes-84
Accuracy:  0.735632183908      64.0 / 87
Label republican :
    Precision: 0.642857142857  27 / 42
    Recall:    0.794117647059  27 / 34
Label democrat :
    Precision: 0.860465116279  37 / 43
    Recall:    0.698113207547  37 / 53
```

```
house-votes-84-gainratio
Accuracy:  0.724137931034      63.0 / 87
Label republican :
    Precision: 0.651162790698  28 / 43
    Recall:    0.823529411765  28 / 34
Label democrat :
    Precision: 0.875   35 / 40
    Recall:    0.660377358491  35 / 53
```

```
iris
Accuracy:  0.766666666667      23.0 / 30
Label Iris-virginica :
    Precision: 1.0     6 / 6
    Recall:    0.6     6 / 10
Label Iris-setosa :
    Precision: 1.0     10 / 10
    Recall:    1.0     10 / 10
Label Iris-versicolor :
    Precision: 1.0     7 / 7
    Recall:    0.7     7 / 10
```

```
iris-gainratio
Accuracy:  0.4         12.0 / 30
Label Iris-virginica :
    Precision: 1.0     1 / 1
    Recall:    0.1     1 / 10
Label Iris-setosa :
    Precision: 1.0     7 / 7
    Recall:    0.7     7 / 10
Label Iris-versicolor :
    Precision: 0.666666666667  4 / 6
    Recall:    0.4     4 / 10
```

```
monks1
Accuracy:  0.865740740741      374.0 / 432
Label 1 :
    Precision: 0.979591836735  192 / 196
    Recall:    0.888888888889  192 / 216
Label 0 :
```

```
    Precision: 0.883495145631  182 / 206
    Recall:    0.842592592593  182 / 216
```

```
monks1-gainratio
Accuracy:  0.611111111111      264.0 / 432
Label 1 :
    Precision: 0.771084337349  128 / 166
    Recall:    0.592592592593  128 / 216
Label 0 :
    Precision: 0.715789473684  136 / 190
    Recall:    0.62962962963   136 / 216
```

```
monks2
Accuracy:  0.62962962963       272.0 / 432
Label 1 :
    Precision: 0.459259259259  62 / 135
    Recall:    0.43661971831   62 / 142
Label 0 :
    Precision: 0.736842105263  210 / 285
    Recall:    0.724137931034  210 / 290
```

```
monks2-gainratio
Accuracy:  0.6875      297.0 / 432
Label 1 :
    Precision: 0.632653061224  62 / 98
    Recall:    0.43661971831   62 / 142
Label 0 :
    Precision: 0.767973856209  235 / 306
    Recall:    0.810344827586  235 / 290
```

```
monks3
Accuracy:  0.87962962963       380.0 / 432
Label 1 :
    Precision: 0.94    188 / 200
    Recall:    0.824561403509  188 / 228
Label 0 :
    Precision: 0.880733944954  192 / 218
    Recall:    0.941176470588  192 / 204
```

```
monks3-gainratio
Accuracy:  0.615740740741      266.0 / 432
Label 1 :
    Precision: 0.716666666667  129 / 180
    Recall:    0.565789473684  129 / 228
Label 0 :
    Precision: 0.671568627451  137 / 204
    Recall:    0.671568627451  137 / 204
```

Information gain ratio is the information gain divided by the range of values of the attributes. Thus if the range of values of attributes is greater, the information gain is less. This makes attributes with a smaller range of variance more valuable.

Naive Bayes

We assume that each predictor is independent of one another. For training, we store a dictionary where the unique labels match to a list of mean and standard deviation pairs. Then when we predict an instance, we calculate the probabilities for each attribute and determine the best classification.

Accuracy Summary

Data	Accuracy
House Votes	78/87
Iris	29/30
Monks1	292/432
Monks2	228/432
Monks3	390/432

Cases

```
house-votes-84
Accuracy:  0.896551724138      78.0 / 87
Label republican :
    Precision: 0.837837837838  31 / 37
    Recall:    0.911764705882  31 / 34
Label democrat :
    Precision: 0.94    47 / 50
    Recall:    0.88679245283   47 / 53
```

```
iris
Accuracy:  0.966666666667      29.0 / 30
Label Iris-virginica :
    Precision: 0.909090909091  10 / 11
    Recall:    1.0     10 / 10
Label Iris-setosa :
    Precision: 1.0     10 / 10
    Recall:    1.0     10 / 10
Label Iris-versicolor :
    Precision: 1.0     9 / 9
    Recall:    0.9     9 / 10
```

```
monks1
Accuracy:  0.675925925926      292.0 / 432
Label 1 :
    Precision: 0.66814159292   151 / 226
    Recall:    0.699074074074  151 / 216
Label 0 :
    Precision: 0.684466019417  141 / 206
    Recall:    0.652777777778  141 / 216
```

```
monks2
Accuracy:  0.527777777778      228.0 / 432
Label 1 :
    Precision: 0.349514563107  72 / 206
    Recall:    0.507042253521  72 / 142
Label 0 :
    Precision: 0.690265486726  156 / 226
    Recall:    0.537931034483  156 / 290
```

```
monks3
Accuracy:  0.902777777778      390.0 / 432
Label 1 :
    Precision: 1.0     186 / 186
    Recall:    0.815789473684  186 / 228
Label 0 :
    Precision: 0.829268292683  204 / 246
    Recall:    1.0     204 / 204
```

Neural Network

We iterate over the number of epochs, but parallelize all of the operations for a set of training cases instead of using iteration. Using gradient descent, we update the learning rate as the epochs increase.

Accuracy Summary

Data	Accuracy
House Votes	79/87
Iris	29/30
Monks1	292/432
Monks2	290/432
Monks3	338/432

```
house-votes-84
Accuracy:  0.908045977011      79.0 / 87
Label republican :
    Precision: 0.809523809524  34 / 42
    Recall:    1.0     34 / 34
Label democrat :
    Precision: 1.0     45 / 45
    Recall:    0.849056603774  45 / 53
```

```
iris
Accuracy:  0.966666666667      29.0 / 30
Label Iris-virginica :
    Precision: 1.0     9 / 9
    Recall:    0.9     9 / 10
Label Iris-setosa :
    Precision: 1.0     10 / 10
    Recall:    1.0     10 / 10
Label Iris-versicolor :
    Precision: 0.909090909091  10 / 11
    Recall:    1.0     10 / 10
```

```
monks1
Accuracy:  0.675925925926      292.0 / 432
Label 1 :
    Precision: 0.688118811881  139 / 202
    Recall:    0.643518518519  139 / 216
Label 0 :
    Precision: 0.665217391304  153 / 230
    Recall:    0.708333333333  153 / 216
```

```
monks2
Accuracy:  0.671296296296      290.0 / 432
Label 0 :
    Precision: 0.671296296296  290 / 432
    Recall:    1.0     290 / 290
```

```
monks3
Accuracy:  0.782407407407      338.0 / 432
Label 1 :
    Precision: 0.908536585366  149 / 164
    Recall:    0.65350877193   149 / 228
Label 0 :
    Precision: 0.705223880597  189 / 268
    Recall:    0.926470588235  189 / 204
```

Evidently, a neural net works best on classification-type problems, specifically house votes (which it performed the best on) and iris (performed the best).

Different Weight Initializations

```
house-votes-84
Accuracy:  0.862068965517      75.0 / 87
Label republican :
    Precision: 0.84375     27 / 32
    Recall:    0.794117647059  27 / 34
Label democrat :
    Precision: 0.872727272727  48 / 55
    Recall:    0.905660377358  48 / 53
```

```
iris
Accuracy:  0.933333333333      28.0 / 30
Label Iris-virginica :
    Precision: 0.833333333333  10 / 12
    Recall:    1.0     10 / 10
Label Iris-setosa :
    Precision: 1.0     10 / 10
    Recall:    1.0     10 / 10
Label Iris-versicolor :
    Precision: 1.0     8 / 8
    Recall:    0.8     8 / 10
```

```
monks1
Accuracy:  0.685185185185      296.0 / 432
Label 1 :
    Precision: 0.717391304348  132 / 184
    Recall:    0.611111111111  132 / 216
Label 0 :
    Precision: 0.661290322581  164 / 248
    Recall:    0.759259259259  164 / 216
```

```
monks2
Accuracy:  0.671296296296      290.0 / 432
Label 0 :
    Precision: 0.671296296296  290 / 432
    Recall:    1.0     290 / 290
```

```
monks3
Accuracy:  0.782407407407      338.0 / 432
Label 1 :
    Precision: 0.89880952381   151 / 168
    Recall:    0.662280701754  151 / 228
Label 0 :
    Precision: 0.708333333333  187 / 264
    Recall:    0.916666666667  187 / 204
```

We did not observe much of a deviation from the standard weight init by using the alternate weight initialization for shallow neural networks. In iris, we were slightly less accurate, and in monks1, we were slightly more accurate. In all other cases, this remained the same. This is probably because we were only using one hidden layer in the first place, and this alternate weight init would show its benefits more in a multi-hidden layer neural net.

All Algorithms

Accuracy Summary

Algorithm	House Votes	Iris	Monks1	Monks2	Monks3
Decision Tree	64/87, 63/87	23/30, 12/30	**374/432**, 264/432	272/432, **297/432**	380/432, 266/432
Naive Bayes	78/87	**29/30**	292/432	228/432	**390/432**
Neural Net	**79/87**, 79/87	**29/30**, 28/30	292/432, 296/432	290/432, 290/432	338/432, 338/432

Based on accuracy, the neural net was the best algorithm to use on house votes and on iris. Decision tree came out on top for monks1 and monks2. Naive Bayes tied with neural net on iris and won on monks3.

Accuracy was high for house votes across the board. Iris did okay in decision tree, but naive bayes and neural net both only misclassified one test case. All of the monks models had accuracies that were in the 0.6-0.8 range, and decision tree did better than the others in two of the three monks models.

In decision tree, we observed IG ratio performing significantly worse than IG except for in monks2. This is probably because this dataset benefitted from a reduction in bias toward multi-valued attributes by taking the number and size of branches into account when choosing an attribute.