CIS 419 HW1

Varun S Rajagopal

February 9, 2018

1. (a) Current entropy (35 positive examples, 15 negative examples) =
$H = -\frac{35}{50} \times log_2(35/50) - \frac{15}{50} \times log_2(15/50) \approx 0.881$

Information Gain: Sunny

Yes: playing = 20, not playing = 1
$H_{suny} = -\frac{20}{21} \times log_2(20/21) - \frac{1}{21} \times log_2(1/21) \approx 0.276$

No: playing = 15, not playing = 14
$H_{sunn} = -\frac{15}{29} \times log_2(15/29) - \frac{14}{29} \times log_2(14/29) \approx 0.999$

Expected Entropy:$\frac{21}{50} \times 0.276 + \frac{29}{50} \times 0.999 \approx 0.695$

Information Gain $= 0.881 - 0.695 \approx 0.186$

Information Gain: Snow

Yes: playing = 25, not playing = 10
$H_{snoy} = -\frac{25}{35} \times log_2(25/35) - \frac{10}{35} \times log_2(10/35) \approx 0.863$

No: playing = 10, not playing = 5
$H_{snon} = -\frac{10}{15} \times log_2(10/15) - \frac{5}{15} \times log_2(5/15) \approx 0.918$

Expected Entropy: $\frac{35}{50} \times 0.863 + \frac{15}{50} \times 0.918 \approx 0.880$

Information Gain $= 0.881 - 0.880 \approx 0.001$

$\implies$ By ID3, we choose Sunny to split on as it has a higher information gain (0.186 > 0.001).

(b) New information gain formula:

$$MajorityError(S) - \sum_{v \in values(S)} \frac{|S_v|}{|S|} MajorityError(S_v)$$

The final tree looks as follows:

```
if Size = Large :
   if Color = Blue :
       if Act = Dip :
         class = T
       if Act != Dip :
            if Age = Adult:
                class = F
            if Age != Adult:
                class = T
   if Color != Blue :
      class = T
if Size != Large :
    if Color = Blue :
        class = F
    if Color != Blue :
        if Act = Dip :
           class = T
        if Act != Dip :
            if Age = Adult:
                class = F
            if Age != Adult:
                class = T
```

Color:
Blue: T: 3 F: 5
Red: T: 6 F: 2
Size:
Small: T: 3 F: 5
Large: T: 6 F: 2
Act:
Stretch: T: 3 F: 5
Dip: T: 6 F: 2
Age:
Adult: T: 3 F: 5
Child: T: 6 F: 2

Since all of the above have the same majority error (due to their similar distribution of values), it is arbitrary which we set to be our root feature. The majority error value for each $= \frac{1}{2} \times \frac{3}{8} + \frac{1}{2} \times \frac{1}{4} = \frac{5}{16} = 0.3125$. Suppose then that we choose Size as our root attribute. Now, we aim to minimize the majority error value summed over our subsets as shown below.

When Size = Large, the majority error for the three other attributes is as follows:
MajorityError(Color) $= \frac{1}{2} \times \frac{1}{4} + \frac{1}{2} \times \frac{1}{4} = 0.25$
MajorityError(Act) $= \frac{1}{2} \times \frac{1}{4} + \frac{1}{2} \times \frac{1}{4} = 0.25$
MajorityError(Age) $= \frac{1}{2} \times \frac{1}{2} + \frac{1}{2} \times \frac{0}{4} = 0.25$

In the large case, we therefore arbitrarily pick our next attribute. Lets say we choose color. Then, Act and Age have the same majority error again $= \frac{1}{2} \times \frac{1}{2} + \frac{1}{2} \times 0 = 0.25$. Again we arbitrarily pick between them, and we choose Act, then Age.

When Size = Small, the majority error for the three other attributes is as follows:
MajorityError(Color) $= \frac{1}{2} \times 0 + \frac{1}{2} \times \frac{1}{4} = 0.125$
MajorityError(Act) $= \frac{1}{2} \times \frac{1}{4} + \frac{1}{2} \times \frac{1}{2} = 0.375$
MajorityError(Age) $= \frac{1}{2} \times \frac{1}{4} + \frac{1}{2} \times \frac{1}{2} = 0.375$

Hence, in the small case we choose color next. Again, the two remaining have similar majority error values as can be observed from looking at the table ($\frac{1}{2} \times \frac{1}{4} + \frac{1}{2} \times \frac{1}{2} = 0.375$) and so we choose Act followed by Age.

(c) ID3 does not guarantee a globally optimal tree due to the fact the algorithm does not backtrack. Once the algorithm selects a particular attribute at a given level in the tree, the algorithm does not go back to reconsider the choice. As a result, the algorithm can only guarantee a locally optimal solution (turns out it does), but not a globally optimal one. The reason that it in essence produces a locally optimal solution is that the solution we get is the best possible decision tree given the ID3 algorithm's choice of path. However as indicated earlier, there is the added possibility that by going down a different path, the algorithm produces a globally optimal decision tree in terms of depth and fitting the data.

Evaluation Report

Table 1: HW1 Compiled Results

Algorithm	Average Accuracy	Training Accuracy	Confidence Interval	Features/Parameters
SGD Decision Stumps (8)	82.1%	86.1%	[78.3%,86.0%]	Used Decision Stumps for generating features Learning Rate = 0.1
Decision Tree (8)	75.0%	84.7%	[70.1%,80.0%]	Maximum Depth of 8
SGD Decision Stumps (4)	72.2%	76%	[59.3%,84.0%]	Used Decision Stumps for generating features Learning Rate = 0.1
Full Decision Tree	71.9%	100%	[63.4%,80.4%]	No limitations on tree depth
Simple SGD	71.0%	84.9%	[68.1%,73.9%]	Learning Rate = 0.1 Error Threshold = 0.005
Decision Tree (4)	68.2%	71.4%	[63.1%,73.5%]	Maximum Depth of 4

The above table provides a summary of all the relevant statistics encountered in the course of running 6 different algorithms on the training data. They are sorted by average accuracy (p_a) and the following is a more in depth discussion of the results in order of performance of the algorithms. Aside from the two decision stump algorithms, the features for everything was standard and constructed based on the first 5 letters in the first and last names. The decision stump algorithms were based on features that were the predicted values of 100 different decision trees.

1. **SGD Decision Stumps (Depth 8):**
   By far, the best perfoming algorithm was the last one implemented, which used decision stumps as features. It fared well against the training data, and the disparity between p_a and tr_a isn't very high (82.1 versus 86.1 percent). A learning rate of 0.1 was used for the algorithm after starting with a lower value of 0.01 and incrementally working upwards. Additionally, an error threshold of 0.001 (the deafult value) was used. The decision stumps were probably expressive enough to the point where SGD could reliably use the features generated by them, hence leading to their relative better performance. The t-statistic was 3.09 with a p-value of 0.0365. Hence at the 99% level we fail to reject the null hypothesis of mean differences between decision stumps and trees of depth 8.

2. **Decision Tree (Depth 8):**
   An expressive decision tree was created with depth 8 as it had a high average accuracy and a significantly higher training accuracy (75.0 percent vs 84.7 percent). This disparity is likely due to some overfitting of the data (perhaps a smaller depth would have the two figures closer to one another, expecially since the depth 4 decision tree has values within 3 percent of one another). The best decision tree got 116 out of 140 correct and 24 incorrect.The t-statistic was 0.948 with a p-value of 0.397. Hence at the 99% level we fail to reject the null hypothesis of mean differences between trees of depth 8 and stumps of depth 4. The following is a screenshot of the tree that resulted:

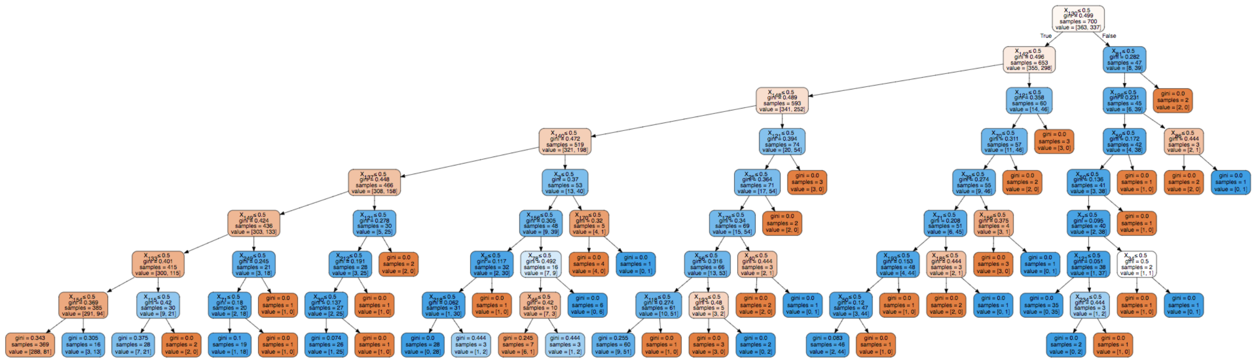

Figure 1: Decision Tree (Depth 8)

3. **SGD Decision Stumps (Depth 4):**
   In determining why the SGD with decision stumps of depth 8 was so succesful, I decided to compare it to an SGD Decision Stump of depth 4. As it turns out, the decision stump is nowhere near as expressive and only had an average accuracy of 72.2 percent indicating that it doesn't pick up on the relevant features prior to applying stochastic gradient descent. The training accuracy was 76 percent so these were close to one another.The t-statistic was 0.09. Hence at the 99% level we fail to reject the null hypothesis of mean differences between stumps of depth 4 and full decision trees.

4. **Full Decision Tree:**
   As can be seen in the image attached, the full decision tree was massive as it had no limitation on depth and therefore perfectly fit the data. The training accuracy of 100 percent is therefore not entirely meaningful as it constructed a perfect tree corresponding to the training data provided. The tree of highest accuracy is similar in depth but the average accuracy is only 71.9 percent. A clear indication of overfitting is that this model performs worse than that with maximum depth 8.The best decision tree got 114 out of 140 correct and 26 incorrect. The t-statistic was 0.255. Hence at the 99% level we fail to reject the null hypothesis of mean differences between full decision trees and simple SGD.

5. **Simple SGD:**
   The first algorithm to be implemented was definitely not the right choice for the classification problem at hand given our choice of features. The average accuracy was 71.0 percent but the disparity compared to training accuracy is immense at around 14 percent. The learning rate was 0.1 and the error threshold was 0.005 (both chosen as they converged well).The t-statistic was 1.63 with a p-value of 0.18. Hence at the 99% level we fail to reject the null hypothesis of mean differences between simple SGD and DT(4).

6. **Decision Tree (4):**
   The worst performing model was the decision tree of depth 4 likely as it didn't allow for good classification because it did not split on enough features. This is an indication

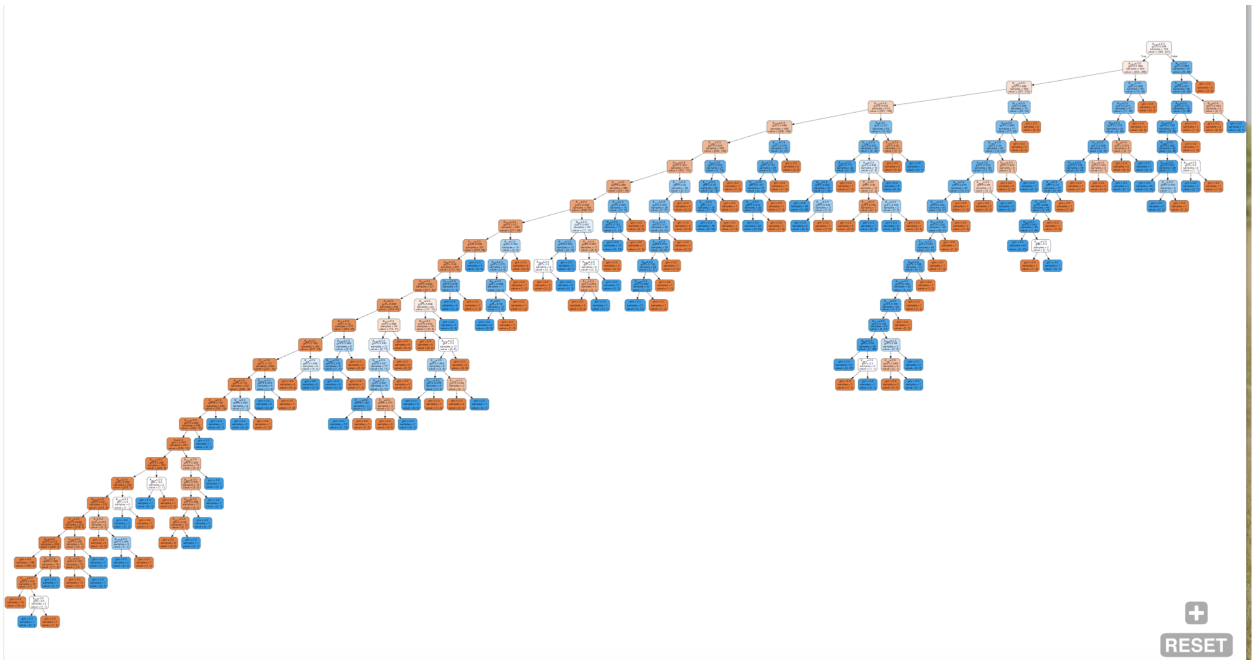

Figure 2: Decision Tree Unlimited Depth

of underfitting and shows why pruning too much can be problematic. The average accuracy was 68.4 percent and the training accuracy was close at 71.4 percent.The best decision tree got 104 out of 140 correct and 36 incorrect.

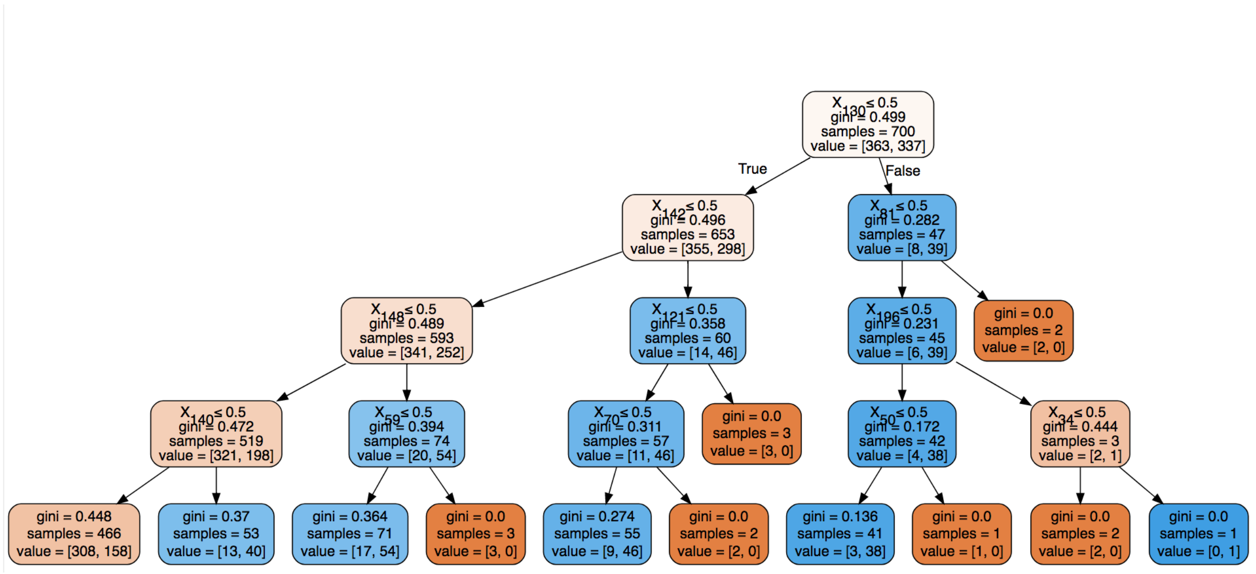
$X_{130} \leq 0.5$ gini = 0.499 samples = 700 value = [363, 337]
True
False
$X_{142} \leq 0.5$ gini = 0.496 samples = 653 value = [355, 298]
$X_{81} \leq 0.5$ gini = 0.282 samples = 47 value = [8, 39]
$X_{148} \leq 0.5$ gini = 0.489 samples = 593 value = [341, 252]
$X_{121} \leq 0.5$ gini = 0.358 samples = 60 value = [14, 46]
$X_{196} \leq 0.5$ gini = 0.231 samples = 45 value = [6, 39]
gini = 0.0 samples = 2 value = [2, 0]
$X_{140} \leq 0.5$ gini = 0.472 samples = 519 value = [321, 198]
$X_{59} \leq 0.5$ gini = 0.394 samples = 74 value = [20, 54]
$X_{70} \leq 0.5$ gini = 0.311 samples = 57 value = [11, 46]
gini = 0.0 samples = 3 value = [3, 0]
$X_{50} \leq 0.5$ gini = 0.172 samples = 42 value = [4, 38]
$X_{34} \leq 0.5$ gini = 0.444 samples = 3 value = [2, 1]
gini = 0.448 samples = 466 value = [308, 158]
gini = 0.37 samples = 53 value = [13, 40]
gini = 0.364 samples = 71 value = [17, 54]
gini = 0.0 samples = 3 value = [3, 0]
gini = 0.274 samples = 55 value = [9, 46]
gini = 0.0 samples = 2 value = [2, 0]
gini = 0.136 samples = 41 value = [3, 38]
gini = 0.0 samples = 1 value = [1, 0]
gini = 0.0 samples = 2 value = [2, 0]
gini = 0.0 samples = 1 value = [0, 1]

Figure 3: Decision Tree (Depth 4)

In conclusion, most of the results were as expected. Based on the training accuracies, average accuracies and most significant difference in performance compared to the other algorithms, I would conclude that the SGD Decision Stumps algorithm was the best performing one. However, given other machine learning techniques that we've yet to cover, it is definitely likely that there are other algorithms that would work better for the classification problem at hand.