CS446: Machine Learning **Fall 2016**

Problem Set 2

Kexin Hui *Handed In: February 16, 2017*

1. Answer to problem 1

   a. In order to determine the root attribute, we need to find out the attribute with the highest information gain. Here is my calculation, where '+' stands for yes and '−' stands for no.

   The data set we are given has 35 positive examples and 15 negatives, so that $p_+ = \frac{35}{50} = 0.7$ and $p_- = \frac{15}{50} = 0.3$. The entropy of the data set can be concluded to $Entropy(D) = Entropy(\{p_+, p_-\}) = -p_+ log(p_+) - p_- log(p_-) = -0.7 \times log(0.7) - 0.3 \times log(0.3) = 0.8813$.

   For the subset $\{Holiday = Yes\}$, the output label with $\{Study\ Today = Yes\}$ has 20 events while the output label with $\{Study\ Today = No\}$ has 1 event. Therefore, $p_+ = \frac{20}{21} = 0.9524$ and $p_- = \frac{1}{21} = 0.0476$. The entropy of this subset is given by $Entropy(\{Holiday = Yes\}) = Entropy(\{p_+, p_-\}) = -p_+ log(p_+) - p_- log(p_-) = -0.9524 \times log(0.9524) - 0.0476 \times log(0.0476) = 0.2761$.

   For the subset $\{Holiday = No\}$, the output label with $\{Study\ Today = Yes\}$ has 15 events while the output label with $\{Study\ Today = No\}$ has 14 events. Therefore, $p_+ = \frac{15}{29} = 0.5172$ and $p_- = \frac{14}{29} = 0.4828$. The entropy of this subset is given by $Entropy(\{Holiday = Yes\}) = Entropy(\{p_+, p_-\}) = -p_+ log(p_+) - p_- log(p_-) = -0.5172 \times log(0.5172) - 0.4828 \times log(0.4828) = 0.9991$.

   Thus, the information gain of Attribute $Holiday$ is given by $Gain(D, `Holiday') = 0.8813 - \frac{21}{50} \times 0.2761 - \frac{29}{50} \times 0.9991 = 0.186$.

   By the same token, the entropy for subset $\{ExamTomorrow = Yes\}$ where 10 positives and 5 negatives is given by $Entropy(\{ExamTomorrow = Yes\}) = Entropy(\{p_+, p_-\}) = -p_+ log(p_+) - p_- log(p_-) = -0.6667 \times log(0.6667) - 0.3333 \times log(0.3333) = 0.9183$.

   The entropy for subset $\{ExamTomorrow = No\}$ where 25 positives and 10 negatives is given by $Entropy(\{ExamTomorrow = No\}) = Entropy(\{p_+, p_-\}) = -p_+ log(p_+) - p_- log(p_-) = -0.7143 \times log(0.7143) - 0.2857 \times log(0.2857) = 0.8631$.

   Thus, the information gain of Attribute $ExamTomorrow$ is given by $Gain(D, `ExamTomorrow') = 0.8813 - \frac{15}{50} \times 0.9183 - \frac{35}{50} \times 0.8631 = 0.0016$.

Based on my calculation, Attribute $Holiday$ has the highest information gain, and is determined to be the root attribute.

b. We will always choose the split node as the attribute that has the smallest MajorityError.

---

1. if $Color = Blue$:
2. if $Size = Small$:
3. $Inflated = F$;
4. if $Size = Large$:
5. if $Act = Dip$:
6. $Inflated = T$;
7. if $Act = Stretch$:
8. if $Age = Adult$:
9. $Inflated = F$;
10. if $Age = Child$:
11. $Inflated = T$;
12. if $Color = Red$:
13. if $Act = Dip$:
14. $Inflated = T$;
15. if $Act = Stretch$:
16. if $Age = Adult$:
17. $Inflated = F$;
18. if $Age = Child$:
19. $Inflated = T$;

---

c. No. ID3 will not guarantee a globally optimal decision tree. ID3 uses greedy search at each node without backtracking, thus it cannot guarantee optimality.

2. Answer to problem 2

(a) This part is simply to generate more features to 10 as described by modifying '$FeatureGenerator.java$'.

(b) We are now asked to implement two basic learning algorithms : decision tree, and stochastic gradient descent in five different combinations on our training data. Algorithms are sorted by average accuracy (p_A) from high to low, as shown in the table below.

Algorithm	Average Accuracy(%)	Standard Deviation
SGD+ Decision Stumps	72.8747	0.05279
Full Decision Tree	72.2292	0.5263
Simple SGD	70.1450	0.06186
Decision Stump of Depth 8	69.6730	0.5035
Decision Stump of Depth 4	65.6355	0.4972

1. Decision Stumps as Features
   Average Accuracy: 72.8747%
   Standard Deviation: 0.05279
   99% *Confidence Level*: $72.8747\% \pm 1.8128\%$
   To tune the parameters, I tried both the learning rate and the error threshold in the set $\{0.1, 0.01, 0.001, 0.0001\}$. As a result, I chose the learning rate to be 0.1 and the error threshold to be 0.0001 for the best average accuracy.

2. Grow Decision Tree
   Average Accuracy: 72.2292%
   Standard Deviation: 0.5263
   99% *Confidence Level*: $72.2292\% \pm 18.0734\%$
   Tree Display according to the 3^{rd} fold test set:

```
ID3

lastName0=m = 1
|  firstName2=e = 1: +
|  firstName2=e = 0
|  |  lastName1=o = 1: +
|  |  lastName1=o = 0
|  |  |  firstName0=p = 1: -
|  |  |  firstName0=p = 0
|  |  |  |  firstName0=r = 1: -
|  |  |  |  firstName0=r = 0
|  |  |  |  |  firstName0=y = 1: -
|  |  |  |  |  firstName0=y = 0
|  |  |  |  |  |  firstName1=u = 1: -
|  |  |  |  |  |  firstName1=u = 0
|  |  |  |  |  |  |  lastName3=a = 1
|  |  |  |  |  |  |  |  firstName3=r = 1: +
|  |  |  |  |  |  |  |  firstName3=r = 0: -
|  |  |  |  |  |  |  lastName3=a = 0: +
lastName0=m = 0
|  lastName1=l = 1
|  |  firstName0=d = 1: -
|  |  firstName0=d = 0: +
|  lastName1=l = 0
|  |  lastName2=l = 1
|  |  |  firstName2=r = 1: -
|  |  |  firstName2=r = 0
|  |  |  |  lastName4=n = 1: -
|  |  |  |  lastName4=n = 0
|  |  |  |  |  firstName2=h = 1: -
|  |  |  |  |  firstName2=h = 0: +
|  |  lastName2=l = 0
|  |  |  lastName2=o = 1
|  |  |  |  firstName0=b = 1: -
|  |  |  |  firstName0=b = 0: +
|  |  |  lastName2=o = 0
|  |  |  |  firstName3=f = 1: +
|  |  |  |  firstName3=f = 0
|  |  |  |  |  lastName4=l = 1
|  |  |  |  |  |  firstName2=h = 1: -
|  |  |  |  |  |  firstName2=h = 0
|  |  |  |  |  |  |  lastName0=l = 1: -
|  |  |  |  |  |  |  lastName0=l = 0
|  |  |  |  |  |  |  |  lastName0=q = 1: -
|  |  |  |  |  |  |  |  lastName0=q = 0: +
|  |  |  |  |  lastName4=l = 0
|  |  |  |  |  |  firstName1=o = 1
|  |  |  |  |  |  |  lastName0=f = 1: -
|  |  |  |  |  |  |  lastName0=f = 0
|  |  |  |  |  |  |  |  firstName2=e = 1: -
|  |  |  |  |  |  |  |  firstName2=e = 0
|  |  |  |  |  |  |  |  |  firstName3=a = 1
|  |  |  |  |  |  |  |  |  |  lastName0=h = 1: +
|  |  |  |  |  |  |  |  |  |  lastName0=h = 0: -
|  |  |  |  |  |  |  |  |  firstName3=a = 0
```

```
|  |  |  |  |  |  |  |  |  |  |  firstName2=n = 1: +
|  |  |  |  |  |  |  |  |  |  |  firstName2=n = 0
|  |  |  |  |  |  |  |  |  |  |  |  lastName2=n = 1: +
|  |  |  |  |  |  |  |  |  |  |  |  lastName2=n = 0
|  |  |  |  |  |  |  |  |  |  |  |  |  lastName1=a = 1: -
|  |  |  |  |  |  |  |  |  |  |  |  |  lastName1=a = 0
|  |  |  |  |  |  |  |  |  |  |  |  |  |  firstName3=g = 1: -
|  |  |  |  |  |  |  |  |  |  |  |  |  |  firstName3=g = 0: +
|  |  |  |  |  |  |  firstName1=o = 0
|  |  |  |  |  |  |  |  lastName0=l = 1
|  |  |  |  |  |  |  |  |  firstName1=a = 1
|  |  |  |  |  |  |  |  |  |  firstName0=d = 1: +
|  |  |  |  |  |  |  |  |  |  firstName0=d = 0: -
|  |  |  |  |  |  |  |  |  firstName1=a = 0: +
|  |  |  |  |  |  |  |  lastName0=l = 0
|  |  |  |  |  |  |  |  |  lastName3=m = 1
|  |  |  |  |  |  |  |  |  |  firstName2=r = 1: -
|  |  |  |  |  |  |  |  |  |  firstName2=r = 0: +
|  |  |  |  |  |  |  |  |  lastName3=m = 0
|  |  |  |  |  |  |  |  |  |  firstName1=e = 1
|  |  |  |  |  |  |  |  |  |  |  firstName2=n = 1: +
|  |  |  |  |  |  |  |  |  |  |  firstName2=n = 0
|  |  |  |  |  |  |  |  |  |  |  |  firstName2=o = 1
|  |  |  |  |  |  |  |  |  |  |  |  |  lastName0=b = 1: -
|  |  |  |  |  |  |  |  |  |  |  |  |  lastName0=b = 0: +
|  |  |  |  |  |  |  |  |  |  |  |  firstName2=o = 0
|  |  |  |  |  |  |  |  |  |  |  |  |  lastName2=r = 1
|  |  |  |  |  |  |  |  |  |  |  |  |  |  firstName0=m = 1: -
|  |  |  |  |  |  |  |  |  |  |  |  |  |  firstName0=m = 0: +
|  |  |  |  |  |  |  |  |  |  |  |  |  lastName2=r = 0: -
|  |  |  |  |  |  |  |  |  |  firstName1=e = 0
|  |  |  |  |  |  |  |  |  |  |  firstName0=t = 1
|  |  |  |  |  |  |  |  |  |  |  |  lastName4=e = 1: -
|  |  |  |  |  |  |  |  |  |  |  |  lastName4=e = 0
|  |  |  |  |  |  |  |  |  |  |  |  |  lastName0=s = 1: -
|  |  |  |  |  |  |  |  |  |  |  |  |  lastName0=s = 0: +
|  |  |  |  |  |  |  |  |  |  |  firstName0=t = 0
|  |  |  |  |  |  |  |  |  |  |  |  firstName3=o = 1
|  |  |  |  |  |  |  |  |  |  |  |  |  firstName0=a = 1: +
|  |  |  |  |  |  |  |  |  |  |  |  |  firstName0=a = 0
|  |  |  |  |  |  |  |  |  |  |  |  |  |  firstName0=m = 1: +
|  |  |  |  |  |  |  |  |  |  |  |  |  |  firstName0=m = 0: -
|  |  |  |  |  |  |  |  |  |  |  |  firstName3=o = 0
|  |  |  |  |  |  |  |  |  |  |  |  |  firstName4=o = 1
|  |  |  |  |  |  |  |  |  |  |  |  |  |  firstName0=s = 1: -
|  |  |  |  |  |  |  |  |  |  |  |  |  |  firstName0=s = 0: +
|  |  |  |  |  |  |  |  |  |  |  |  |  firstName4=o = 0
|  |  |  |  |  |  |  |  |  |  |  |  |  |  lastName0=s = 1
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  firstName0=d = 1: +
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  firstName0=d = 0
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  lastName4=h = 1
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  firstName0=s = 1: -
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  firstName0=s = 0: +
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  lastName4=h = 0: -
|  |  |  |  |  |  |  |  |  |  |  |  |  |  lastName0=s = 0
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  lastName3=l = 1
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  firstName0=d = 1: -
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  firstName0=d = 0: +
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  lastName3=l = 0: -
```

```
Correctly Classified Instances          35               76.087  %
Incorrectly Classified Instances        11               23.913  %
```

3. Stochastic Gradient Descent (SGD)
   Average Accuracy: 70.1450%
   Standard Deviation: 0.06186
   99% *Confidence Level*: $70.1450\% \pm 2.1243\%$
   To tune the parameters, I tried both the learning rate and the error threshold in the set $\{0.1, 0.01, 0.001, 0.0001\}$. As a result, I chose the learning rate to be 0.1

and the error threshold to be 0.0001 for the best average accuracy.

4. Grow Decision Tree of Depth 8
   Average Accuracy: 69.6730%
   Standard Deviation: 0.5035
   99% *Confidence Level*: $69.6730\% \pm 17.2904\%$
   Tree Display according to the 1^{st} fold test set:

```
ID3

firstName3=f = 1: +
firstName3=f = 0
|  lastName0=c = 1: -
|  lastName0=c = 0
|  |  lastName4=l = 1
|  |  |  lastName0=q = 1: -
|  |  |  lastName0=q = 0: +
|  |  lastName4=l = 0
|  |  |  firstName0=r = 1
|  |  |  |  firstName1=o = 1: +
|  |  |  |  firstName1=o = 0
|  |  |  |  |  firstName1=a = 1: +
|  |  |  |  |  firstName1=a = 0
|  |  |  |  |  |  firstName1=e = 1: +
|  |  |  |  |  |  firstName1=e = 0: -
|  |  |  firstName0=r = 0
|  |  |  |  lastName0=m = 1
|  |  |  |  |  firstName2=n = 1: -
|  |  |  |  |  firstName2=n = 0
|  |  |  |  |  |  firstName0=p = 1: -
|  |  |  |  |  |  firstName0=p = 0
|  |  |  |  |  |  |  lastName2=t = 1
|  |  |  |  |  |  |  |  firstName0=t = 1: +
|  |  |  |  |  |  |  |  firstName0=t = 0: -
|  |  |  |  |  |  |  lastName2=t = 0: +
|  |  |  |  lastName0=m = 0
|  |  |  |  |  lastName0=l = 1
|  |  |  |  |  |  firstName1=a = 1
|  |  |  |  |  |  |  firstName0=d = 1: +
|  |  |  |  |  |  |  firstName0=d = 0: -
|  |  |  |  |  |  firstName1=a = 0: +
|  |  |  |  |  lastName0=l = 0
|  |  |  |  |  |  lastName3=m = 1
|  |  |  |  |  |  |  firstName2=r = 1: -
|  |  |  |  |  |  |  firstName2=r = 0: +
|  |  |  |  |  |  lastName3=m = 0
|  |  |  |  |  |  |  lastName2=l = 1
|  |  |  |  |  |  |  |  firstName2=r = 1: -
|  |  |  |  |  |  |  |  firstName2=r = 0: +
|  |  |  |  |  |  |  lastName2=l = 0
|  |  |  |  |  |  |  |  lastName3=l = 1: +
|  |  |  |  |  |  |  |  lastName3=l = 0: -


Correctly Classified Instances          48               73.8462 %
Incorrectly Classified Instances        17               26.1538 %
```

5. Grow Decision Tree of Depth 4
   Average Accuracy: 65.6355%
   Standard Deviation: 0.4972
   99% *Confidence Level*: $65.6355\% \pm 17.0741\%$
   Tree Display according to the 4^{th} fold test set:

```
ID3

lastName2=l = 1
|  firstName2=r = 1: -
|  firstName2=r = 0
|  |  firstName2=m = 1: -
|  |  firstName2=m = 0: +
lastName2=l = 0
|  lastName2=o = 1
|  |  firstName0=d = 1: -
|  |  firstName0=d = 0
|  |  |  firstName2=l = 1: -
|  |  |  firstName2=l = 0: +
|  lastName2=o = 0
|  |  firstName3=f = 1: +
|  |  firstName3=f = 0
|  |  |  lastName0=m = 1
|  |  |  |  firstName0=n = 1: -
|  |  |  |  firstName0=n = 0: +
|  |  |  lastName0=m = 0
|  |  |  |  lastName1=l = 1: +
|  |  |  |  lastName1=l = 0: -


Correctly Classified Instances          48               72.7273 %
Incorrectly Classified Instances        18               27.2727 %
```

- Evaluation
  Decision Stumps as Features turns out to give out the best performance, with the highest averaged accuracy among all five methods we tried. Since the values predicted by Decision Tree method is already very reliable, as the second highest in the list, we just add an extra linear classifier. In other words, the SGD classifier in the end will be more likely to predict correctly given some preliminary knowledge provided by Decision Tree. It is surprising that full decision tree doesn't overfit the data. It kind of makes senses that this depth which is only 14 at full may not be able to overfit the data. Therefore, full decision tree better fits the data and gives good performance. Simple SGD comes at the third. With tuning the learning rate and error threshold, it can give a relatively high predict accuracy. I also try training for 1000 times, the accuracy is even much higher. The lowest two are decision tree with max depth at 8 and 4. When we limit the maximum depth to 8 or 4, not the full depth, the outcome cannot include too much information. As a result, these two methods run pretty fast at the cost of accuracy due to less knowledge for the original data.
  Additionally, based on my calculation:
  the difference between SGD+Decision Stumps and Full decision tree is not statistically significant;
  the difference between Full decision tree and Simple SGD is not statistically significant;
  the difference between Simple SGD and Decision tree with depth 8 is not statistically significant;
  the difference between Decision tree with depth 8 and Decision tree with depth 4 is not statistically significant;

- Conclusion
  To sum up, Decision Stumps as Features performed the best. SGD+Decision Stumps gives us much more flexibility and freedom to fit the data. In other words, we are no longer limited to the linear separability in simple SGD but we are open to any Boolean function to observe the data given the Decision Stumps. The basic assumption for this conclusion might be that the dataset should be the same.