CS446: Machine Learning **Spring 2017**

Problem Set 2

Zhiyu Zhu (zzhu24) *Handed In: February 17, 2017*

1. **Learning Decision Trees**

   (a) In order to know which attribute should be the root of the tree, we want to find the gain value of each attribute:

   Firstly, assume the root is Holiday:
   We want to find the entropy of Holiday_yes:
   $Entropy(S_{Holiday\_yes})$
   $= -P_{study=yes}log_2(P_{study=yes}) - P_{study=no}log_2(P_{study=no})$
   $= -\frac{20}{20+1}log_2(\frac{20}{20+1}) - \frac{1}{20+1}log_2(\frac{1}{20+1})$
   $= 0.070557$
   $Entropy(S_{Holiday\_no})$
   $= -P_{study=yes}log_2(P_{study=yes}) - P_{study=no}log_2(P_{study=no})$
   $= -\frac{15}{15+14}log_2(\frac{15}{15+14}) - \frac{14}{15+14}log_2(\frac{14}{15+14})$
   $= 0.999214$
   $Entropy(S_{Holiday})$
   $= -P_{study=yes}log_2(P_{study=yes}) - P_{study=no}log_2(P_{study=no})$
   $= -\frac{20+15}{20+15+1+14}log_2(\frac{20+15}{20+15+1+14}) - \frac{1+14}{20+15+1+14}log_2(\frac{1+14}{20+15+1+14})$
   $= 0.881291$
   Then we can find the gain from the entropy we found above:
   $Gain(S_{Holiday})$
   $= Entropy(S) - (\frac{\|S_{yes}\|}{\|S\|}Entropy(S_{yes}) + \frac{\|S_{no}\|}{\|S\|}Entropy(S_{no}))$
   $= 0.881291 - \frac{21}{50}0.070557 - \frac{29}{50}0.999214$
   $= 0.272155$

   Secondly, assume the root is Exam Tomorrow:
   We want to find the entropy of Exam_yes:
   $Entropy(S_{Exam\_yes})$
   $= -P_{study=yes}log_2(P_{study=yes}) - P_{study=no}log_2(P_{study=no})$
   $= -\frac{10}{10+5}log_2(\frac{10}{10+5}) - \frac{5}{10+5}log_2(\frac{5}{10+5})$
   $= 0.918296$
   $Entropy(S_{Exam\_no})$
   $= -P_{study=yes}log_2(P_{study=yes}) - P_{study=no}log_2(P_{study=no})$
   $= -\frac{25}{25+10}log_2(\frac{25}{25+10}) - \frac{10}{25+10}log_2(\frac{10}{25+10})$
   $= 0.863121$
   $Entropy(S_{Exam})$
   $= -P_{study=yes}log_2(P_{study=yes}) - P_{study=no}log_2(P_{study=no})$
   $= -\frac{10+25}{10+25+5+10}log_2(\frac{10+25}{10+25+5+10}) - \frac{10+5}{10+25+5+10}log_2(\frac{10+5}{10+25+5+10})$
   $= 0.881291$

Then we can find the gain from the entropy we found above:
$Gain(S_{Exam})$
$= Entropy(S) - (\frac{\|S_{yes}\|}{\|S\|} Entropy(S_{yes}) + \frac{\|S_{no}\|}{\|S\|} Entropy(S_{no}))$
$= 0.881291 - \frac{15}{50} 0.918296 - \frac{35}{50} 0.863121$
$= 0.001618$

According to the calculation, we can conclude that the root attribute should be Holiday because:
$Gain(S_{Holiday}) > Entropy(S_{Exam\_yes})$.

(b) According to the information given in the table, the decision tree should be:

```
if Color = Blue :
  if Size = Small :
    Inflated = F
  if Size = Large :
    if Act = Stretch :
      if Age = Adult :
        Inflated = F
      if Age = Child:
        Inflated = T
    if Act = Dip :
      Inflated = T
If Color Red:
  if Size = Small :
    if Act = Stretch :
      if Age = Adult :
        Inflated = F
      if Age = Child:
        Inflated = T
    if Act = Dip :
      Inflated = T
  if Size = Large :
    if Act = Stretch :
      if Age = Adult :
        Inflated = F
      if Age = Child:
        Inflated = T
    if Act = Dip :
      Inflated = T
```

(If-Then statement of decision tree)

(c) No, the ID3 does not guarantee a globally optimal decision tree. Since the ID3 focuses on the local minimal and tend to find the "greedy", this is the heuristic algorithm that might not find the best solution of the best attribute to be the root of the decision tree. Secondly, the ID3 focuses on the most closed splitting of the training data and after more than one splitting according to different attributes, it will be much less accurate. Thirdly, ID3's tend to focus on the most closed splitting, it also tends to choose the one holding more training data.

2. **Decision Trees as Features**

(a) **Feature Extraction and Instance Generation:** All the generators written in FeatureGenerator.Java in the zip file.
Also, there is another PDF that contains all the decision trees and the data of evaluation generated from the programming of all kinds of algorithm.

(b) **Algorithm:**

i. **SGD:** Wrote the file SGD.java and SGDrun.Java as the algorithm and the running-printing-evaluation files in the zip file

ii. **Decision Trees:** Wrote Depth4.Java, Depth8.Java and WholeTree.Java for the ID3 decision tree algorithm that are in the maximum depth of 4,8,14 and printing the result of evaluation.

iii. **Decision stumps as features:** Wrote the file Stump.Java that takes in the SGD.Java and ID3.Java as imports and the decision tree depth limitation of 4.

Part 1: Evaluation

Algorithm	Tree Depth	Learning Rate	$P_{correct}$	Confidence Interval
SGD	4	0.0001	52.31%	$[0.44, 0.61]$
$Tree_{depth=4}$	4	NONE	52.30%	$[0.43, 0.61]$
$Tree_{depth=8}$	8	NONE	67.68%	$[0.63, 0.73]$
$Tree_{depth=14}$	14	NONE	71.08%	$[0.67, 0.75]$
$Feature_{DecisionStump}$	4	0.0001	63.08%	$[0.56, 0.72]$

Table 1: Result Evaluation

According to the result generated from Java programing: (all results in the PDF files). The SGD has the worst performance. The Decision tree largely improved the performance and precision of the algorithm. And as the tree grows, the performance of the algorithm is improved accordingly. The Decision Stump also have a good performance but the result is not statistically significantly different from the decision tree with the max depth of 8.
We can find that the SGD is the worst algorithm for training and testing all the data because it takes the least features and cannot reach a result precise enough.
The decision trees improves the results a lots because it takes in much more features than the the SGD. As the tree grows, theoretically it will go into overfit that cause the performance go down. However, my experiments may not reach the limitation that it keeps go up as the tree analysis more features.
The Decision Stump also has a food performance but not as good as the the largest decision tree probably because it analysis 100 features that get into the situation of overfit of data. Also, as we train the Decision Stump more, the performance go up and sometimes reach over 70% of correctness in the algorithm.

Part 2: Conclusion Based on both the P_A value and the average performance of each algorithm, the large decision tree and the Decision Stump all have good performance. The correctness of prediction can be over 70% in both algorithem.