CPT_S 570 HW3

Yang Zhang

11529139

1. **(a)**

   We define g(y)=y^2 and $f(y) = \sum_i y$. Since g(y)'' = 2 which is bigger than 0, therefore g(y) is a convex function. By Jensen's inequality, g[f(y)] ≤ f[g(y)] holds true for random y and convex function g(y). Thus $(\sum_i y)^2 \le \sum_i y^2$, then we substitute y with (x-z):

   $$[\sum_i (x-z)]^2 \le \sum_i (x-z)^2$$

   And since d is large, so 1/d < 1. Therefore,

   $$\frac{1}{d}[\sum_i (x-z)]^2 \le \sum_i (x-z)^2$$

   Which is equivalent to

   $$(\frac{1}{\sqrt{d}}\sum_i x - \frac{1}{\sqrt{d}}\sum_i z)^2 \le \sum_i (x-z)^2$$

   (b)

   We could use the lower bound of the Euclidean distance (the right side of Jensen's inequality) to estimates the actual Euclidean distance of two random points in R-space. Then the estimation distance between x and z is

   $$\frac{1}{\sqrt{d}}\sum_i x - \frac{1}{\sqrt{d}}\sum_i z$$

   Which can be done in linear time instead of matter of polynomial

2. The key idea of LSH is to precompute a hash table that puts points with same probability in the same bucket. For input point p, the algorithm retrieves all the points in the same bucket with p (i.e. points that have the same probability as p). Then looping over the points we retrieved and calculating the distance from p. Finally, records the point if it is a correct answer.

3. It is not always possible to construct a decision tree from set of rules. If the rules in the rule set don't have intersection (common if-condition), then there is no such common node to connect the tree branch construct from the rule. For example, suppose the rule-sets has two rules:

   (1) If today is Sunday and sunny Jayce would play basketball
   (2) If it is windless and Jayce is happy then he would play basketball

   There is no way to adapt a decision based on the two rules.

4. For each of the features (Outlook, Temp, Humidity, Wind) has 9 yes and 5 no
So, the entropy H(Outlook) = H(Temp) = H(Humidity) = H(Wind) = $-\frac{9}{14}\ln\frac{9}{14} - \frac{5}{14}\ln\frac{5}{14} = 0.65$
Note: (x / y) denotes (# of yes / # of no)
The Outlook feature has 3 attributes Sunny (2 / 3), Overcast (4 / 0), Rain (3 / 2)
H(Sunny) = $-\frac{2}{5}\ln\frac{2}{5} - \frac{3}{5}\ln\frac{3}{5} = 0.67$
H(Overcast) = $-\frac{4}{4}\ln\frac{4}{4} - 0\ln 0 = 0$
H(Rain) = $-\frac{2}{5}\log_2\frac{2}{5} - \frac{3}{5}\log_2\frac{3}{5} = 0.67$

G(Outlook) = H(Outlook) – 5/14H(Sunny) – 4/14H(Overcast) – 5/14H(Rain)

= 0.65– 0.24 – 0 -0.24 =0.17

The Temp feature has 3 attributes Hot (2 / 2), Mild (4 / 2), Cool (3 / 1)
H(Hot) = $-\frac{2}{4}\ln\frac{2}{4} - \frac{2}{4}\ln\frac{2}{4} = 0.69$
H(Mild) = $-\frac{4}{6}\ln\frac{4}{6} - \frac{2}{6}\ln\frac{2}{6} = 0.64$
H(Cool) = $-\frac{3}{4}\ln\frac{3}{4} - \frac{1}{4}\ln\frac{1}{4} = 0.56$

G(Temp) = H(Temp) – 4/14H(Hot) – 6/14H(Mild) – 4/14H(Cool)

= 0.019

The Humidity feature has 2 attributes High (3 / 3), Normal (6 / 2)
H(High) = $-\frac{3}{6}\ln\frac{3}{6} - \frac{3}{6}\ln\frac{3}{6} = 0.69$
H(Normal) = $-\frac{2}{8}\ln\frac{2}{8} - \frac{6}{8}\ln\frac{6}{8} = 0.56$

G(Humidity) = H(Humidity) – 6/14H(High) – 8/14H(Normal)

= 0.034

The Wind feature has 2 attributes Strong (3 / 3), Weak (6 / 2)
H(Strong) = $-\frac{3}{6}\ln\frac{3}{6} - \frac{3}{6}\ln\frac{3}{6} = 0.69$
H(Weak) = $-\frac{2}{8}\ln\frac{2}{8} - \frac{6}{8}\ln\frac{6}{8} = 0.56$

G(Wind) = H(Wind) – 6/14H(Strong) – 8/14H(weak)

= 0.034

Therefore, we pick Outlook feature to split first since it has the max information gain.

5.

(b) The validation accuracy for Tree 1: 0.6007031642390759
The testing accuracy for Tree 1: 0.6033182503770739

Confusion matrix for test 1:

```
  ~0    ~1
0 665 335
1 460 531
```

```
The validation accuracy for Tree 2: 0.57
The testing accuracy for Tree 2: 0.5742904841402338
```

```
Confusion matrix for test 2:
  ~0 ~1
0 196 104
1 154 146
```

(d)

```
The validation accuracy for pruned Tree 1: 0.6601307189542484
The testing accuracy for pruned Tree 1: 0.6072325464590658
```

```
Confusion matrix for test 1:
  ~0 ~1
0 557 443
1 339 652
```

```
The validation accuracy for pruned Tree 2: 0.6277128547579299
The testing accuracy for pruned Tree 2: 0.5516666666666666
```

```
Confusion matrix for test 2:
  ~0 ~1
0 110 190
1 79  221
```

From the result above we can see that the accuracy of validation is improved, this is because we used the validation test to prune the decision tree, while pruning the tree doesn’t guarantee to have better result on testing data. There is a little bit testing accuracy improvement on tree 1, but on tree 2 the testing accuracy is worse than unpruned result.