Homework 1

Qinyun Song

1 Decision Tree

1.1 Write Decision Tree

Algorithm 1 Decision Tree 1

1.

```
$(x_1 \wedge x_2) \vee (x_1 \wedge x_3)$
if $x_1 = 0$ then
    return 0
else if $x_2 = 1$ then
    return 1
else if $x_3 = 1$ then
    return 1
else
    return 0
end if
```

Algorithm 2 Decision Tree 2

2.

```
$(x_1 \wedge x_2)\, xor\, x_3$
if $x_3 = 1$ then
    if $x_1 = 0$ then
        return 1
    else if $x_2 = 0$ then
        return 1
    else
        return 0
    end if
else
    if $x_1 = 0$ then
        return 0
    else if $x_2 = 0$ then
        return 0
    else
        return 1
    end if
end if
```

Algorithm 3 Decision Tree 2

3.

```
$\neg A \vee \neg B \vee \neg C \vee \neg D$
if $A = 0$ then
    return 1
else if $B = 0$ then
    return 1
else if $C = 0$ then
    return 1
else if $D = 0$ then
    return 1
else
    return 0
end if
```

1.2 Aliens invading

1. The number of possible functions is:

$$2^{2\times 2\times 4\times 4} = 2^{64}$$

The number of consistent possible functions is:

$$2^{64-9} = 2^{55}$$

2. The probability that the alien is going to invade the earth is $\frac{5}{9}$. The probability that it is not to do that is $\frac{5}{9}$. So the entropy of the data can be calculated as:

$$E = -\frac{5}{9}\log -\frac{5}{9} - \frac{4}{9}\log -\frac{4}{9} \approx 0.99$$

3. (a) For the feature Technology:

i. If Technology equals True, there is one example showing that the alien will invade and there are two show that it won't. So the entropy is:

$$E_{T_T} = -\frac{1}{3}\log\frac{1}{3} - \frac{2}{3}\log\frac{2}{3} \approx 0.92$$

ii. Similarly, when Technology is False, we can have:

$$E_{T_F} = -\frac{2}{6}\log\frac{2}{6} - \frac{4}{6}\log\frac{4}{6} \approx 0.92$$

So the overall information gain is:

$$Gain(All, T) = E - \frac{3}{9}0.92 - \frac{6}{9}0.92 = 0.07$$

(b) For the feature Environment: Similarly, we can have the following two equations:

$$\begin{aligned} E_{E_T} &= -\frac{4}{5}\log\frac{4}{5} - \frac{1}{5}\log\frac{1}{5} \approx 0.72 \\ E_{E_F} &= -\frac{1}{4}\log\frac{1}{4} - \frac{3}{4}\log\frac{3}{4} = \approx 0.81 \\ Gain(All, E) &= 0.99 - \frac{5}{9}\times 0.72 - \frac{4}{9}\times 0.81 = 0.23 \end{aligned}$$

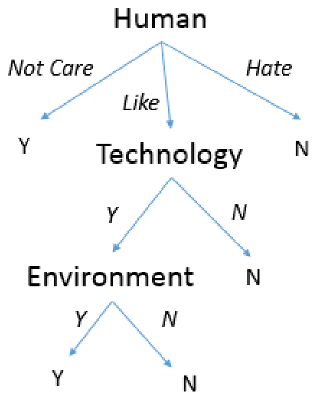
(c) for the feature Human, we have

$$\begin{aligned}
E_{H_{DC}} &= -\frac{4}{4}\log\frac{4}{4} = 0 \\
E_{H_L} &= -\frac{1}{4}\log\frac{1}{4} - \frac{3}{4}\log\frac{3}{4} =\approx 0.81 \\
E_{H_H} &= -\frac{1}{1}\log\frac{1}{1} = 0 \\
Gain(All, H) &= 0.99 - \frac{4}{9}\times 0 - \frac{4}{9}\times 0.81 - \frac{1}{9}\times 0 \\
&= 0.63
\end{aligned}$$

(d) For the feature distance, we have

$$\begin{aligned}
E_{D_1} &= -\frac{1}{2}\log\frac{1}{2} - \frac{1}{2}\log\frac{1}{2} = 1 \\
E_{D_2} &= -\frac{1}{1}\log\frac{1}{1} = 0 \\
E_{D_3} &= -\frac{2}{3}\log\frac{2}{3} - \frac{1}{3}\log\frac{1}{3} \approx 0.92 \\
E_{D_4} &= -\frac{1}{3}\log\frac{1}{3} - \frac{2}{3}\log\frac{2}{3} \approx 0.92 \\
Gain(All, D) &= 0.99 - 1\times\frac{2}{9} - 0\times\frac{1}{9} - 0.92\times\frac{3}{9} - 0.92\times\frac{3}{9} \approx 0.15
\end{aligned}$$

4. I would use the Human feature to construct the root of the decision tree. Because its information gain is the maximum.



5.

6. The three results from the decision tree constructed above are Yes, No, Yes. The accuracy of the classifier is $\frac{2}{3}$

1.3 Majority Error

1. Calculate the information gain of the four features using the majority error.

   The majority label among all the data is Yes. So the overall majority error would be:

$$M = 1 - \frac{5}{9} \approx 0.444$$

   i. For the feature Technology, when it is Yes, the majority error is:

$$M_{T_T} = 1 - \frac{2}{3} \approx 0.333$$

   When it is No, the majority error is:

$$M_{T_N} = 1 - \frac{4}{6} \approx 0.333$$

   So the information gain of Technology would be:

$$Gain(All, Technology) = 0.444 - \frac{3}{9} \times 0.333 - \frac{6}{9} \times 0.333 = 0.111$$

   ii. for the feature Environment, similarly, we can have:

$$\begin{aligned} M_{E_Y} &= 1 - \frac{4}{5} = 0.2 \\ M_{E_N} &= 1 - \frac{3}{4} = 0.25 \\ Gain(All, Environment) &= 0.444 - \frac{5}{9} \times 0.2 - \frac{4}{9} \times 0.25 \approx 0.222 \end{aligned}$$

   iii. for the feature Human, we have:

$$\begin{aligned} M_{H_N C} &= 1 - \frac{1}{1} = 0 \\ M_{H_L} &= 1 - \frac{3}{4} = 0.25 \\ M_{H_H} &= 1 - \frac{1}{1} = 0 \\ Gain(All, Human) &= 0.444 - 0 \times \frac{4}{9} - 0.25 \times \frac{4}{9} - 0 \times \frac{1}{9} \approx 0.333 \end{aligned}$$

   iv. for the feature Distance,

$$\begin{aligned} M_{D_1} &= 1 - \frac{1}{2} = 0.5 \\ M_{D_2} &= 1 - \frac{1}{1} = 0 \\ M_{D_3} &= 1 - \frac{2}{3} = 0.333 \\ M_{D_4} &= 1 - \frac{2}{3} = 0.333 \\ Gain(All, Distance) &= 0.444 - 0.5 \times \frac{2}{9} - 0 \times \frac{1}{9} - 0.333 \times \frac{3}{9} - 0.333 \times \frac{3}{9} \approx 0.111 \end{aligned}$$

2. According to the calculations above, the best attribute to choose could be the Technology or the Distance. Because their information gain are the same and are smaller than the other two. This is completely with the conclusion derived from entropy. So the tree derived here is different from the tree dirived previous.

2 Linear Classifier

1. The function could be:
$$o = 2x_1 + x_2 - x_3$$
2. Only the prediction of the fourth data is corerct. So the overall accuracy is $\frac{1}{7}$.
3. The linear classifier could be:
$$o = sgn(3x_1 - x_3 + 3x_4 - 1)$$
where the function sgn means that if the parameter bigger than 1 returns 1 otherwise returns -1.

3 Experiments

3.1 Implementation

1. To design the code, there are many things needed to be considered.
   - What features to use?
     I used four features.
     (a) The length of the name is smaller than four or between four to eight or bigger than eight.
     (b) The sum of the characters in the name mod three.
     (c) The fourth character in the name.
     (d) The sum of the first name of the person mod two.
   - How to implement it in object-oriente way?
     I don't want to put all the calculations in one code because I think that is a very bad idea. So I divide the calculation into several parts. One is to handle the people and get information from them. One is using a specific feature to calculate information gain and divide people into sub-group. And the last one is the structure helping us to construct the tree. So I defined three different kinds of classes to handle the three situations.
   - How to implement the features?
     I noticed that the features all need same functions. Like returns what entropy is or how to divide the people into sub-groups. So I first designed a virtual class as the father. And designed four feature classes that inherent from the virtual class. So that in the following calculating, one pointer in type father class can handle all different features.
   - How to handle the people?
     I use a class to handle a group of people. The class can help to calculate the entropy of the people and return what is the most frequent label. By doing this, during the calculation, we don't need to care about how to calculate these any more.
2. There are many other features that can be used, such as:

   (a) The occurance of one specific character like a.
   (b) The length of the first name or last name.
   (c) Number of spaces in the name.
   (d) If there is a special character like a dot.

3. The accuracy of my tree on the training data is 0.667416

4. The accuracy of my tree on the test data is 0.612613.

5. The maximum depth of the decision tree is 5.

3.2 Limiting Depth

1. Since the maximum depth of my decision tree is 5, I will only use the depth in the set $\{1, 2, 3, 4, 5\}$. The accuracy and the standard deviation are listed below.

Depth	1	2	3	4	5
Accuracy	0.623874	0.623874	**0.628378**	0.621622	0.621622
Standard Deviation	0.05406	0.05406	0.06258	0.05976	0.06021

   From the table we can see that, the best depth of my decision tree should be three. There are two reasons. First of all, it produces the best accuracy. Secondly, its standard deviation doesn't different from others much. So I would choose three as the depth.

2. By setting the maximum depth as three, the accuracy testing on the training data is 0.640449. The number on the testing data is 0.621622.

3. I do think limiting the depth of the tree is a good idea. The main reason is that, we can find out that, after limiting the depth from five to three, the accuracy on the training dasta set decreased. However, the accuracy on the test data increased. From this behavior, we can see that, we alleviate the overfitting problem. We can find a model that predicts better on the testing data although has poorer performance on the training data.

4 Decision Lists

Proof. For one decision list, suppose it contains n variables $x_1, \cdots, x_n$. Proof it by induction.

- For the first node, we only have requirement of one variable. So we only need to see whether this variable is true or false. If the condition is x_1, the equivalent linear function will be

$$x_1 > 0 \tag{1}$$

  If it is bigger than zero, output one. Otherwise, output zero. If the condition is $\not x_1$, the equivalent linear function will be

$$\begin{aligned} 1 - x_1 &> 0 \\ -x_1 &> -1 \end{aligned} \tag{2}$$

- So for the one decision list with only one node, it is linear seperable. Suppose we have prooved that, for one decision list with no more than $n-1$ nodes are all linear seperable. That is, we can have a linear function $w^T x \geq b$ that describe exact the one decision list. Now we add a new condition node with variable x_n.
  If it requires a positive variable, then we can add x_n to the left side of the linear function and 1 to the right side of it. Because the adding constriction shows that we need one more variable that is true showing that the sum of all variables now need to be increased by one. By doing this, the updated linear function can represent this decision tree uniquely.
  Otherwise, it would require a negative value. Similar as the above one, we only need to add $-x_n$ to the left side of the function and add -1 to the right side of it. Because we want to add one more constrain that requires one variable to be negative. So the sum should decreases by one. So the updated linear function can also represent the decision list.

So after all, every one decision tree corresponds to one linear function. So it is linear seperable. □