CS 5350 Assignment 1

Alec Adair u0725062

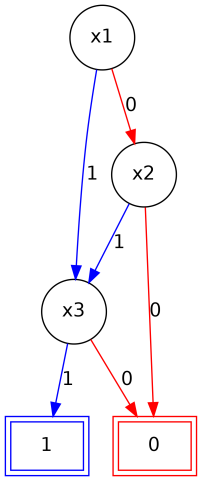
September 14, 2016

1 Decision Trees

1.a

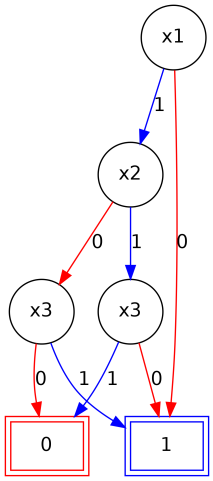
The inputs and output of each function can be written out using truth tables. From the truth table the boolean equation can be derived and simplified to find the simplest decision tree.

x_1	x_2	x_3	((x_1 ∨ x_2) ∧ x_3)
F	F	F	**F**
F	F	T	**F**
F	T	F	**F**
F	T	T	**T**
T	F	F	**F**
T	F	T	**T**
T	T	F	**F**
T	T	T	**T**



1.b

a	b	c	(a ∧ b) ⊕ (¬a ∨ c)
T	T	T	F
T	T	F	T
T	F	T	T
T	F	F	F
F	T	T	T
F	T	F	T
F	F	T	T
F	F	F	T

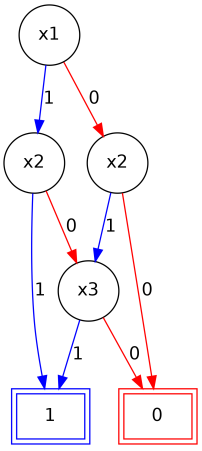


1.c

x1	x2	x3	f(x1,x2,x3)
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

This truth table can be translated into a boolean equation.

$f(x1, x2, x3) = \bar{x}1x2x3 + x1\bar{x}2x3 + x1x2\bar{x}3 + x1x2x3$



2

2.a

For the four variables there are $2X3X3X4 = 72$ combinations of variables. Since each combination can output 1 or 0, this means that there are 2^{72} total possible ways to map the four variables to boolean decisions.

2.b

There are equal numbers of positive and negative labels in this example, therefore the probability is 1/2 for each. The entropy for the data set is given by the following equation:

$Entropy = -(1/2)log_2(1/2) - (1/2)log_2(1/2)$
$= .5 - (-.5) = 1$
1 is the highest amount of entropy a data set can have.

2.c

For the Berry attribute there are two subsets, since the berry is a boolean value.
Using the entropy 1 calculated above for the entire set.

$Gain_{Berry} = Gain(S, A)$
S = Entire Data Set S_v = Subset of data set, where value $v = attributea$.
$Gain(S, A) = H(S) - \sum \frac{|S_v|}{|S|} H(S_v) = EntireDataSet$
$H(S_{yes}) = -\frac{6}{7} log_2(\frac{6}{7}) - \frac{1}{7} log_2(\frac{1}{7}) = .592$
$H(S_{no}) = -\frac{2}{9} log_2(\frac{2}{9}) - \frac{7}{9} log_2(\frac{7}{9}) = .764$
$Gain_{Berry} = 1 - \sum \frac{|S_v|}{|S|} H(S_v) = .311$

$Gain_{Ball}$
$|S_{poke}| = 6$
$|S_{great} = 7|$
$|S_{ultra} = 3|$
$H(poke) = -1/6(log_2 1/6) - 5/6(log_2 5/6) = .650$
$H(great) = -3/7(log_2 3/7) - (-4/7(log_2 4/7) = .524 + .461 = .985$
$H(ultra) = 3/3$ caught $\rightarrow Entropy = 0$
$Gain = 1 - ((6/16).65 + (7/16).985 + (3/16)0) = .675$

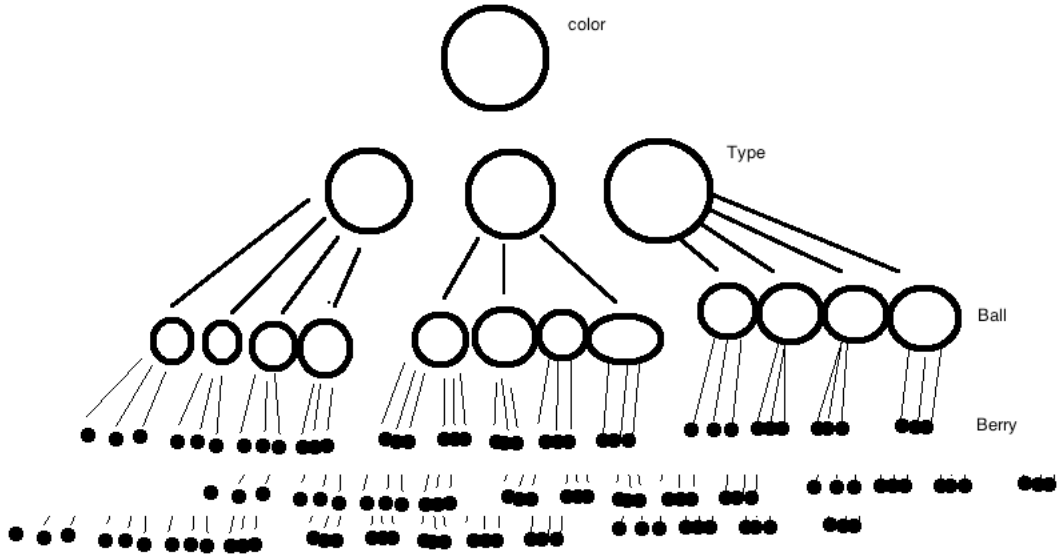
$Gain_{Color}$
$H(green) = -2/3(log_2 2/3) - 1/3(log_2 1/3) = .918$
$H(yellow) = -4/7(log_2 4/7) - 3/7 log_2 3/7 = .985$
$H(red) = -3/6 = 1$
$Gain(color) = 1 - (3/16(.918) + 7/16(.985) + 6/16) = .978$

$Gain_{Type}$
$H(Normal) = 1$
$H(water) = 1$
$H(flying) = -1/4 log_2 1/4 - 3/4 log_2 3/4 = .811$
$H(psychic) = 0$
$Gain(Type) = 1 - (6/16 + 4/16 + .811(4/16) = .828$

2.d

Based on my calculation Color should be chosen for the root node attribute, because it has the highest information gain.

2.e



Type:
i = 0 is normal
i = 1 is water
i = 2 is flying
i = 3 is psychic

Color:
i = 0 is green
i = 1 is yellow
i = 2 is red

Ball:
i = 0 is poke
i = 1 is great
i = 2 is ultra

Berry:
i = 0 is off
i = 1 is on

2.f

2.g

I think it is a good idea, because as more data comes in, it will be possible to build stronger more accurate trees to predict the game.

3

3.a

$Gini(type) = 1 - ((6/16)^2 + (4/16)^2 + (4/16)^2 + (2/16)^2)) = .719$
$Gini(color) = 1 - ((3/16)^2 + (7/16)^2 + (6/16)^2) = .633$
$Gini(berry) = 1 - ((1/2)^2 + (1/2)^2) = .5$
$Gini(ball) = 1 - ((3/16)^2 + (7/16)^2 + (6/16)^2) = .633$

3.b

According to the Gini Calculation, the type should be used as the root node. This is different than the entropy calculation, and therefor leads to a different tree.

4 Linear Classifiers

4.a

$sign(y = x_1 + x_2 + x_3 + x_4 - 1)$
$b = -1$
$w_1 = w_2 = w_3 = w_4 = 1$

4.b

The linear classifier correctly classifies all except for one. The case that it does not classify for is the zero case. I say it classifies incorrectly because sign(0) is not a negative number.

4.c

5 Experiments

5.a Setting A

1. Some choices to be made for the algorithm were what kind of data structure to house the data table in, what kind of functions to have. I implemented the algorithm in C++ and chose to use a vector of vectors of type char to act as the data table.

2. The error for training.data is zero for my tree.

3. The error for test.data is zero for my tree.

4. The maximum depth is 4.

5.b Limiting Depth Setting A

1.
```
Setting A
Depth Avg accuracy Std Dev
1 0.513658 0.00432735
2 0.513972 0.00432735
3 0.513972 0.00432735
4 0.514024 0.00431995
5 0.514024 0.00431995
10 0.514129 0.00430526
15 0.514704 0.00431259
20 0.514757 0.00430526
```

2.

5.c Setting B

1.

5.d Limiting Depth Setting B

```
Setting B
Depth Avg accuracy Std Dev
1 0 0
2 0.0392465 0.00511365
3 0.00193616 1.70508e-05
4 0.00549451 0.000145807
5 0.00868655 0.000218949
10 0.00130822 8.05056e-07
15 0.00324437 2.0846e-05
20 0.0021978 5.9147e-07
```

From these results the depth of 2 should be chosen because it has the highest accuracy.

6 Conclusion

I believe that I implemented the ID3 algorithm correctly. I do not believe that my test was implemented 100% accurately. For Setting A everything seemed to work fine, but then for testing Setting B my test would get stuck in a never ending loop. I thoroughly examined and tried to fix the test, but was unable to. Due to this I do not believe that my test information is 100% accurate.