CSCI 416 - Intro to Machine Learning **Homework 4** **Kyle Chen**

(Due December 5)

Question 2

a)

Given that

$$Gain(S, F) = Entropy(S) - \sum_{v \in Values(F)} (\frac{|S_v|}{|S|} Entropy(S_v)),$$

where $Entropy(S) = -p_1 log_2(p_1) - p_0 log_2(p_0)$ and F is a single feature in the set S, we will first find the entropy of the whole set S. So,

$$Entropy(S) = -1 \times \frac{3}{5} log_2(\frac{3}{5}) - \frac{2}{5} log_2(\frac{2}{5}) = 0.971$$

We must now calculate the individual entropy of each value of each feature. We will use the notation $S_{(f,r)}$, where f is the feature of the set and r is the output of the training data. Since each feature only has two values, we have:

$$\begin{aligned}
Entropy(S_{(x_1,+)}) &= -1 log_2(1) - 0 log_2(0) \\
&= 0 \\
Entropy(S_{(x_1,-)}) &= \frac{-1}{3} log_2(\frac{1}{3}) - \frac{2}{3} log_2(\frac{2}{3})) \\
&= 0.918 \\
Entropy(S_{(x_2,+)}) &= \frac{-3}{4} log_2(\frac{3}{4}) - \frac{1}{4} log_2(\frac{1}{4}) \\
&= 0.811 \\
Entropy(S_{(x_2,-)}) &= -0 log_2(0) - 1 log_2(1) \\
&= 0 \\
Entropy(S_{(x_3,+)}) &= \frac{-1}{3} log_2(\frac{1}{3}) - \frac{2}{3} log_2(\frac{2}{3})) \\
&= 0.918 \\
Entropy(S_{(x_3,-)}) &= -1 log_2(1) - 0 log_2(0) \\
&= 0
\end{aligned}$$

Thus, we will then calculate the gain for each feature x_1, x_2, x_3. So,

$$\begin{aligned}
Gain(S, x_1) &= 0.971 - \sum_{v \in x_1} \frac{|S_v|}{|S|} Entropy(S_v) \\
&= 0.971 - (\frac{2}{5} \times 0 + \frac{3}{5} \times 0.918) \\
&= 0.971 - 0.551 \\
&= 0.420 \\
Gain(S, x_2) &= 0.971 - \sum_{v \in x_2} \frac{|S_v|}{|S|} Entropy(S_v) \\
&= 0.971 - (\frac{4}{5} \times 0.811) \\
&= 0.322 \\
Gain(S, x_3) &= 0.971 - \sum_{v \in x_3} \frac{|S_v|}{|S|} Entropy(S_v) \\
&= 0.971 - (\frac{3}{5} \times 0.918) \\
&= 0.971 - 0.551 \\
&= 0.420
\end{aligned}$$

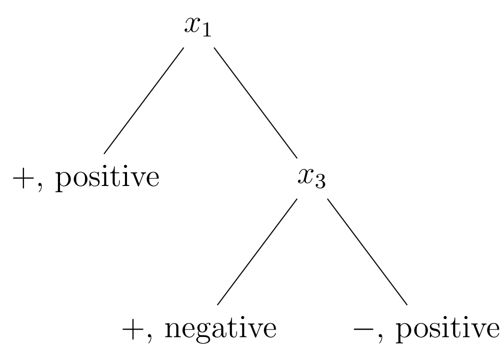
Since the gain for features x_1, x_3 are the same, we will choose x_1 to be the root of our decision tree. We will then recalculate the Entropy of x_2, x_3 on S^1, where $S^1 = S_{x_1,2}$ and being the set of remaining uncertain values in the set S using feature x_1 as a decision boundary. So, using the same methodology as above,

$$\begin{aligned}
Entropy(S^1_{(x_2,+)}) &= \frac{-1}{2} log_2(\frac{1}{2}) - \frac{1}{2} log_2(\frac{1}{2}) \\
&= -1 \times -1 \\
&= 1 \\
Entropy(S^1_{(x_2,-)}) &= -1 log_2(1) - 0 log_2(0) \\
&= 0 \\
Entropy(S^1_{(x_3,+)}) &= -0 log_2(0) - 1 log_2(1) \\
&= 0 \\
Entropy(S^1_{(x_3,-)}) &= 1 log_2(1) - 0 log_2(0) \\
&= 0
\end{aligned}$$

So, our gains on S_1 should therefore be:

$$
\begin{aligned}
Gain(S_1, x_2) &= 0.918 - \frac{2}{3} \times 1 \\
&= 0.251 \\
Gain(S_1, x_3) &= 0.918 - 0 \\
&= 0.918
\end{aligned}
$$

Thus, we choose x_3 as our second decision boundary. Since all the training data has been classified, we can now construct the decision tree, with the value on the left being the specific datapoint for the feature and the value on the right being the return value:



Thus, for our new datapoints,

$$
\begin{aligned}
d_6 :& x_1 = +, x_2 = -, x_3 = - \\
d_7 :& x_1 = -, x_2 = -, x_3 = - \\
d_8 :& x_1 = +, x_2 = -, x_3 = +
\end{aligned}
$$

We have

$$d_6 \ \rightarrow \ +, \ d_7 \ \rightarrow \ +, \ d_8 \ \rightarrow \ +$$

and we are done.

b)

Given the weight update function of $w_i = \frac{w_i exp(-\alpha_t y_i h_t(x_i))}{Z_t}$, where w_i is the weight given to each datapoint per iteration of the AdaBoost algorithm, $\alpha_t = \frac{1}{2} ln(\frac{1-\epsilon_t}{\epsilon_t})$, $\epsilon_t = \sum w_i | y_i \times h_t(x_i) = -1$, $Z_t = \sum w_i$ after the application of $w_i = w_i exp(-\alpha_t y_i h_t(x_i))$ for all weights in the set, and t being the current iteration of the AdaBoost algorithm. Given the table:

	x_1	x_2	x_3	y
d_1	-1	1	1	-1
d_2	1	1	1	1
d_3	-1	1	-1	1
d_4	-1	-1	1	-1
d_5	1	1	-1	1

and that our max iterations $T = 3$, we can begin by arbitrarily choosing a feature, x_1, to calculate a weak classifier over. Since we only have 5 pieces of training data, we set each $w_i = \frac{1}{5}$. So, starting with $t = 1$, we have weak classifier

$$h_1(x_i) = \left\{ \begin{array}{l} 1, \text{ if feature } x_1 > 0 \\ -1, \text{ otherwise} \end{array} \right\}$$

So, we have $\epsilon_1 = w_3 = \frac{1}{5}$, $a_1 = \frac{1}{2}ln(4) = 0.693$. Thus,

$$\begin{aligned}
w_1 &= \frac{1}{5}exp(\frac{-1}{2}ln(4) \times 1) = \frac{1}{5} \times \frac{1}{2} = \frac{1}{10} \\
w_2 &= \frac{1}{5}exp(\frac{-1}{2}ln(4) \times 1) = \frac{1}{10} \\
w_3 &= \frac{1}{5}exp(\frac{-1}{2}ln(4) \times -1) = \frac{1}{5} \times 2 = \frac{2}{5} = \frac{4}{10} \\
w_4 &= \frac{1}{10} \\
w_5 &= \frac{1}{10}
\end{aligned}$$

So, $Z_1 = \frac{1}{10} \times 4 + \frac{4}{10} = \frac{8}{10} = \frac{4}{5}$. We can now divide the weights by Z_1:

$$\begin{aligned}
w_1 &= \frac{\frac{1}{10}}{\frac{4}{5}} = \frac{1}{8} \\
w_2 &= \frac{1}{8} \\
w_3 &= \frac{\frac{2}{5}}{\frac{4}{5}} = \frac{1}{2} \\
w_4 &= \frac{1}{8} \\
w_5 &= \frac{1}{8}
\end{aligned}$$

For $t = 2$, we have weak classifier

$$h_2(x_i) = \left\{ \begin{array}{l} 1, \text{ if feature } x_3 < 0 \\ -1, \text{ otherwise} \end{array} \right\}$$

Thus, $\epsilon_2 = w_2 = \frac{1}{8},\ \alpha_2 = \frac{1}{2}ln(7) = 0.973$. So,

$$
\begin{aligned}
w_1 &= \frac{1}{8}exp(\frac{-1}{2}ln(7) \times 1) = \frac{1}{8\sqrt{7}} \\
w_2 &= \frac{1}{8}exp(\frac{-1}{2}ln(7) \times -1) = \frac{\sqrt{7}}{8} = \frac{7}{8\sqrt{7}} \\
w_3 &= \frac{1}{2}exp(\frac{-1}{2}ln(7) \times 1) = \frac{1}{2\sqrt{7}} = \frac{4}{8\sqrt{7}} \\
w_4 &= \frac{1}{8\sqrt{7}} \\
w_5 &= \frac{1}{8\sqrt{7}}
\end{aligned}
$$

So, $Z_2 = \frac{14}{8\sqrt{7}} = \frac{\sqrt{7}}{4}$. As such:

$$
\begin{aligned}
w_1 &= \frac{\frac{1}{8\sqrt{7}}}{\frac{\sqrt{7}}{4}} = \frac{1}{14} \\
w_2 &= \frac{\frac{\sqrt{7}}{8}}{\frac{\sqrt{7}}{4}} = \frac{1}{2} \\
w_3 &= \frac{\frac{1}{2\sqrt{7}}}{\frac{\sqrt{7}}{4}} = \frac{2}{7} \\
w_4 &= \frac{1}{14} \\
w_5 &= \frac{1}{14}
\end{aligned}
$$

For $t = 3$, we have weak classifier

$$
h_3(x_i) = \left\{ \begin{array}{l} 1, \text{if feature } x_2 > 0 \\ -1, \text{otherwise} \end{array} \right\}
$$

So, $\epsilon_3 = w_1 = \frac{1}{14},\ \alpha_3 = \frac{1}{2}ln(13) = 1.28$. So,

$$
\begin{aligned}
w_1 &= \frac{1}{14}exp(\frac{-1}{2}ln(13) \times -1) = \frac{\sqrt{13}}{14} = \frac{13}{14\sqrt{13}} \\
w_2 &= \frac{1}{2}exp(\frac{-1}{2}ln(13) \times 1) = \frac{1}{2\sqrt{13}} = \frac{7}{14\sqrt{13}} \\
w_3 &= \frac{2}{7}exp(\frac{-1}{2}ln(13) \times 1) = \frac{2}{7\sqrt{13}} = \frac{4}{14\sqrt{13}} \\
w_4 &= \frac{1}{14}exp(\frac{-1}{2}ln(13) \times 1) = \frac{1}{14\sqrt{13}} \\
w_5 &= \frac{1}{14\sqrt{13}}
\end{aligned}
$$

So, $Z_3 = \frac{26}{14\sqrt{13}} = \frac{\sqrt{13}}{7}$. Thus,

$$
\begin{aligned}
w_1 &= \frac{\frac{\sqrt{13}}{14}}{\frac{\sqrt{13}}{7}} = \frac{1}{2} \\
w_2 &= \frac{\frac{1}{2\sqrt{13}}}{\frac{\sqrt{13}}{7}} = \frac{7}{26} \\
w_3 &= \frac{\frac{2}{7\sqrt{13}}}{\frac{\sqrt{13}}{7}} = \frac{2}{13} \\
w_4 &= \frac{\frac{1}{14\sqrt{13}}}{\frac{\sqrt{13}}{7}} = \frac{1}{26} \\
w_5 &= \frac{1}{26}
\end{aligned}
$$

Therefore, given the testing data

$$
\begin{aligned}
d_6 :& x_1 = 1, x_2 = -1, x_3 = -1 \\
d_7 :& x_1 = -1, x_2 = -1, x_3 = -1 \\
d_8 :& x_1 = 1, x_2 = -1, x_3 = 1
\end{aligned}
$$

we have the following predictions:

$$
\begin{aligned}
H(d_6) &= sign(0.693(1) + 1.28(-1) + 0.973(1)) \\
&= sign(0.386) \\
&= 1 \\
H(d_7) &= sign(0.693(-1) + 1.28(-1) + 0.973(1)) \\
&= sign(-1) \\
&= -1 \\
H(d_8) &= sign(0.693(1) + 1.28(-1) + 0.973(-1)) \\
&= sign(-1.56) \\
&= -1
\end{aligned}
$$

Question 3

Given the table:

ID	PAIN?	MALE?	SMOKES?	WORK OUT?	DISEASE
1.	yes	yes	no	yes	yes
2.	yes	yes	yes	no	yes
3.	no	no	yes	no	yes
4.	no	yes	no	yes	no
5.	yes	no	yes	yes	yes
6.	no	yes	yes	yes	no
7.	no	yes	yes	no	?

To choose whether or not the person of ID 7 has the disease, we will have to find $p(\text{Diseased} \mid \text{ID7})$ and $p(\text{Not Diseased} \mid \text{ID7})$ and compare the two. In order to best use the Naive Bayes Algorithm, we will first apply Laplace Smoothing to the sets of diseased and not diseased values. So, we have:

	PAIN?	MALE?	SMOKES?	WORK OUT?	DISEASE
Given	yes	yes	no	yes	yes
	yes	yes	yes	no	yes
	no	no	yes	no	yes
	yes	no	yes	yes	yes
Laplace Smoothing	no	yes	yes	no	yes
	yes	no	no	yes	yes

and

	PAIN?	MALE?	SMOKES?	WORK OUT?	DISEASE
Given	no	yes	no	yes	no
	no	yes	yes	yes	no
Laplace Smoothing	no	yes	yes	no	no
	yes	no	no	yes	no

So,

$$\begin{aligned}
p(\text{No Pain} \mid \text{Diseased}) &= \frac{2}{6} = \frac{1}{3} \\
p(\text{Male} \mid \text{Diseased}) &= \frac{3}{6} = \frac{1}{2} \\
p(\text{Smokes} \mid \text{Diseased}) &= \frac{4}{6} = \frac{2}{3} \\
p(\text{No Work Out} \mid \text{Diseased}) &= \frac{3}{6} = \frac{1}{2}
\end{aligned}$$

$$\begin{aligned}
p(\text{No Pain} \mid \text{Not Diseased}) &= \frac{3}{4} \\
p(\text{Male} \mid \text{Not Diseased}) &= \frac{3}{4} \\
p(\text{Smokes} \mid \text{Not Diseased}) &= \frac{2}{4} \\
p(\text{No Work Out} \mid \text{Not Diseased}) &= \frac{1}{4}
\end{aligned}$$

So,

$$p(\text{Diseased} \mid \text{ID7}) = \frac{1}{18}$$

and

$$p(\text{Not Diseased} \mid \text{ID7}) = \frac{9}{128}$$

giving us

$$p(\text{Diseased} \mid \text{ID7}) = \frac{2}{3} \times \frac{1}{18} = \frac{1}{27}$$

and

$$p(\text{Not Diseased} \mid \text{ID7}) = \frac{1}{3} \times \frac{9}{128} = \frac{3}{128}$$

Since $p(\text{Diseased} \mid \text{ID7}) > p(\text{Not Diseased} \mid \text{ID7})$, then we predict that $ID7$ has the disease, and we are done.

Question 4

Given that $k = 2$ to find the k-means of the one-dimensional data set

$$D = [4, 1, 9, 12, 6, 10, 2, 3, 9],$$

as well as the initial clusters and cluster centers:

$$\begin{aligned}
&Cluster_1 = [1, 2, 3],\ Center_1 = 1 \\
&Cluster_2 = [4, 9, 12, 6, 10, 9],\ Center_2 = 6
\end{aligned}$$

a) We will now iterate two more times through the k-means algorithm. First, we will find the new cluster centers:

$$Center_1 = mean(Cluster_1) = \frac{1+2+3}{3} = 2$$
$$Center_2 = mean(Cluster_2) = \frac{4+9+12+6+10+9}{6} = 8.33$$

With these new cluster centers, we can reform the clusters around these centers, giving us:

$$Cluster_1 = [1, 2, 3, 4],\ Center_1 = 2$$
$$Cluster_2 = [9, 12, 6, 10, 9],\ Center_2 = 8.33$$

We will iterate one more time, giving us:

$$Cluster_1 = [1, 2, 3, 4],\ Center_1 = 2.5$$
$$Cluster_2 = [9, 12, 6, 10, 9],\ Center_2 = 9.2$$

And we are done.

b) Yes, the k-means algorithm did converge for $k = 2$, as both clusters remained the same between the two iterations of the algorithm.