CS 5350/6350: Machine Learining Spring 2019

Homework 1

Handed out: 25 January, 2019
Due date: 11:59pm, 10 Feb, 2019

Author: Cade Parkison
Github: https://github.com/c-park/Machine-Learning-Library

1 Decision Tree [40 points + 10 bonus]

x_1	x_2	x_3	x_4	y
0	0	1	0	0
0	1	0	0	0
0	0	1	1	1
1	0	0	1	1
0	1	1	0.	0
1	1	0	0	0
0	1	0	1	0

Table 1: Training data for a Boolean classifier

1. [7 points] Decision tree construction.

   (a)

   The following are the steps used in generating the decision tree using the ID3 algorithm:

   i. Create a root node and calculate entropy of whole dataset

   ii. For each of the four features X_i, calculate the information gain. Initially, both X_2 and X_4 give the same information gain, so I picked x_2 since it was first and assigned it to the root node. The info gain calculations are shown below.

$$Gain(S, A) = Entropy(S) - \sum_v \frac{|S_v|}{|S|} Entropy(S_v)$$

Where:

$$Entropy(S) = \sum_{i=1}^{k} p_i \log_2(p_i)$$

iii. For each feature value of x_2 (0 and 1), I added an unassigned branch to the tree.

iv. Then, I split the dataset on this x_2 feature and removed it from the dataset.

v. Now there are two children of the root node, and each has it's own dataset. These subsets are shown below.

x_1	x_3	x_4	y
0	1	0	0
0	1	1	1
1	0	1	1

Table 2: $x_2 = 0$ data subset

x_1	x_3	x_4	y
0	0	0	0
0	1	0.	0
1	0	0	0
0	0	1	0

Table 3: $x_2 = 1$ data subset

vi. Repeating the steps above for the $x_2 = 0$ branch, I calculate entropy of this subset.

vii. I then calculate info gains for each remaining feature and split the subset on the one with the highest info gain, which is x_4.

viii. x_4 gets assigned to the $x_2 = 0$ branch, and the subset is split again into two branches, one for each feature value of x_4, these are shown below.

x_1	x_3	y
0	1	0

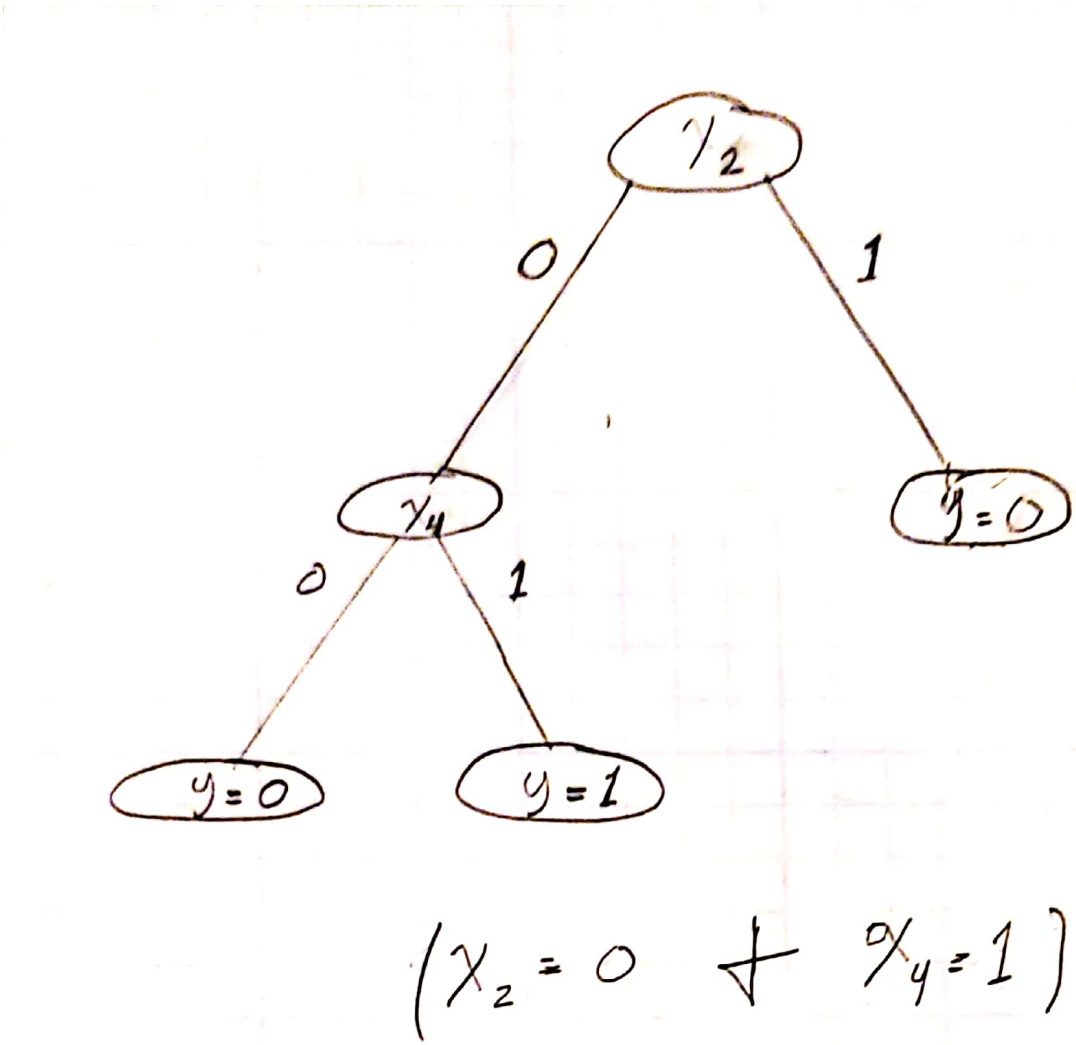
Table 4: $x_2 = 0$ and $x_4 = 0$ data subset

ix. At the next level of recursion, I found that all target feature values are the same, so the x_4 branch of the tree ends. This can be seen in the above tables.

x. Returning to the $x_2 = 1$ branch, we see that the subset for this branch all have the same target feature value, meaning that if $x_2 = 1$ we can predict the outcome $y = 0$

x_1	x_3	y
0	1	1
1	0	1

Table 5: $x_2 = 0$ and $x_4 = 1$ data subset



(b) Boolean Representation:

$$(x_2 = 0 \wedge x_4 = 1)$$

The above implicit equation is enough to fully describe the decision tree created in the previous problem.

2.

(a)

Information gain using Majority Error:

The following steps show the construction of the tree.

i. Create a root node and calculate majority error of whole dataset, S. Majority error is the error in labeling if the majority label were chosen. Since the majority of the examples are positive for playing tennis, this error is the ratio of negative examples to total examples.

$$ME(S) = \frac{5}{14}$$

ii. For each feature, calculate the information gain using majority error. The formula used and results are shown below.

$$Gain(S, A) = ME(S) - \sum_v \frac{|S_v|}{|S|} ME(S_v)$$

Info Gain for Outlook attribute (O):

$$\begin{aligned} G(S, O) &= ME(S) - \sum_v \frac{|S_v|}{|S|} ME(S_v) \\ &= ME(S) - \Big(\frac{|O = s|}{|S|} ME(O = s) + \frac{|O = o|}{|S|} ME(O = o) + \frac{|O = r|}{|S|} ME(O = r) \Big) \\ &= \frac{5}{14} - \Big(\frac{5}{14} \times \frac{2}{5} + \frac{4}{14} \times \frac{0}{4} + \frac{5}{14} \times \frac{2}{5} \Big) \\ &= \frac{1}{14} \end{aligned}$$

Info Gain for Temperature attribute (T):

$$\begin{aligned} G(S, T) &= \frac{5}{14} - \Big(\frac{4}{14} \times \frac{2}{4} + \frac{6}{14} \times \frac{2}{6} + \frac{4}{14} \times \frac{1}{4} \Big) \\ &= 0 \end{aligned}$$

Info Gain for Humidity attribute (H):

$$\begin{aligned} G(S, H) &= \frac{5}{14} - \Big(\frac{7}{14} \times \frac{3}{7} + \frac{7}{14} \times \frac{1}{7} \Big) \\ &= \frac{1}{14} \end{aligned}$$

Info Gain for Wind attribute (W):

$$G(S,W) = \frac{5}{14} - \left(\frac{8}{14} \times \frac{2}{8} + \frac{6}{14} \times \frac{3}{6}\right)$$
$$= 0$$

Both the Outlook and Humidity attributes have the highest information gain, so either one of these can be chosen for the first node of our tree. I then chose the Outlook attribute to split the data.

iii. For each attribute value of O, sunny overcast and rainy, I make a new branch of the tree. I then split the dataset based on this value, and remove the Overcast attribute from the data completely.

iv. I then repeat the above for each of the three data subsets by finding the best attribute to split the data subset based on the highest information gain.

v. Starting with the Outlook=sunny data subset, shown below, we calculate the majority error.

T	H	W	Play?
H	H	W	-
H	H	S	-
M	H	W	-
C	N	W	+
M	N	S	+

Table 6: Outlook=sunny data subset

$$ME(S_{O=sunny}) = \frac{2}{5}$$

vi. Next, I calculate the info gain for each of the remaining attributes using the same formulas as above.

$$Gain(T) = 0.32$$
$$Gain(H) = 0.4$$
$$Gain(W) = 0$$

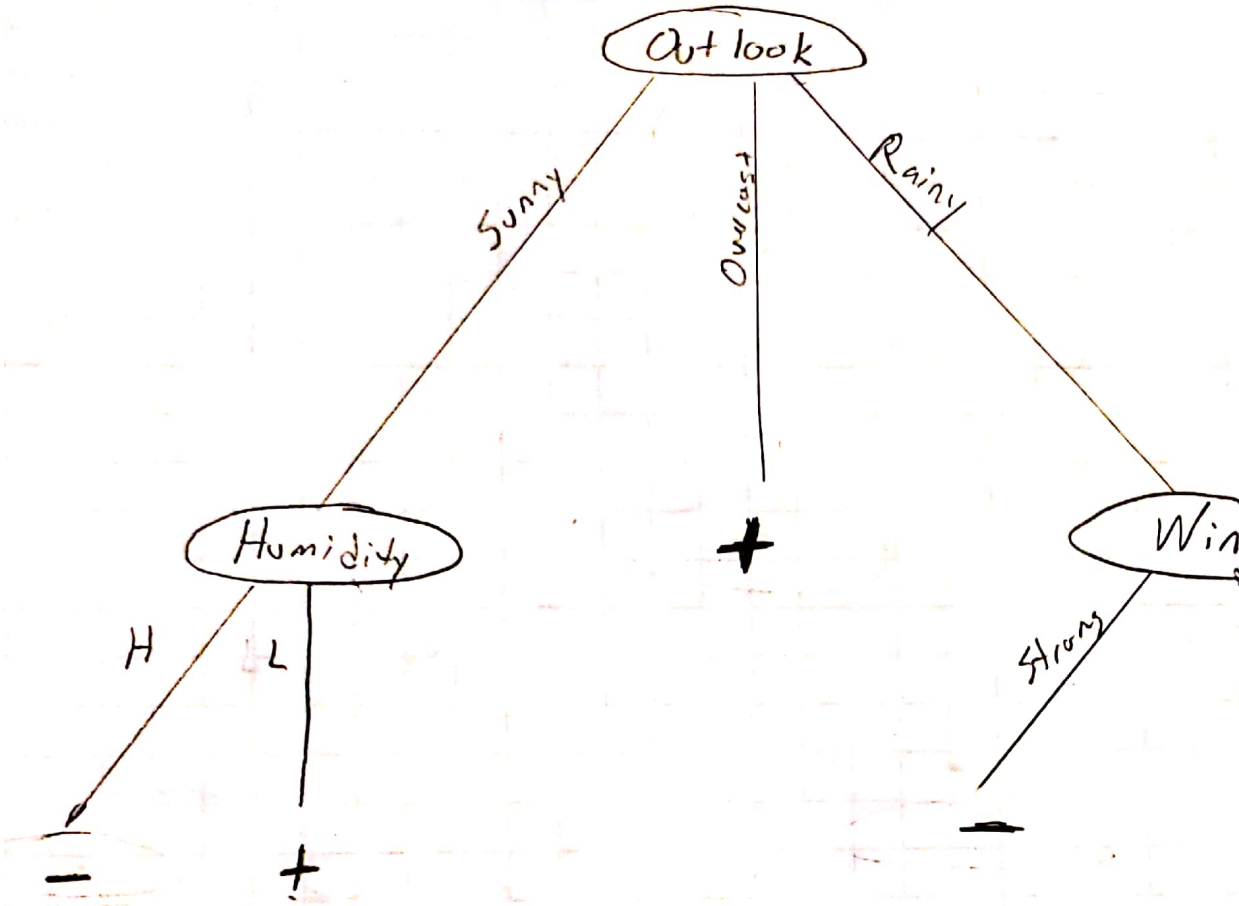
vii. As can be seen, the Humidity attribute has the highest information gain, so we split the subset on these attribute values and create new tree branches with them.

viii. For the O=sunny and Humidity data subset, all remaining subsets have equal target values. O=sunny and H=high gives Play=-, while O=sunny and H=low gives Play=+. We now traverse back up the tree and continue the recursion.

ix. All O=overcast data have the target value of Play=+, so this branch is also complete.

x. Finally, we take the O=rainy data subset and find the info gains for each remaining attribute. In this case, the Wind attribute gives the highest gain, so we split the remaining data on these attribute values.

xi. After splitting again, we see all remaining data have the same target values, so the tree is complete. See the picture below.


Outlook
Sunny
Overcast
Rainy
Humidity
H
L
+
Win
High
−
−
+

(b)

Gini Index:

The following steps are used to construct a tree using the Gini index.

i. Create root node and calculate Gini index of complete data set.

$$\begin{aligned} GI(S) &= 1 - (p_-^2 + p_+^2) \\ &= 1 - (\frac{9}{14})^2 - (\frac{5}{14})^2 \\ &= \frac{45}{98} \\ &= 0.45918 \end{aligned}$$

ii. For each feature, calculate information gain using this Gini index.

$$\begin{aligned} Gain(S, A) &= GI(S) - \sum_v \frac{|S_v|}{|S|} GI(S_v) \\ &= \\ &= \\ &= \end{aligned}$$

Info Gain for Outlook attribute (O):

$$\begin{aligned} G(S, O) &= GI(S) - \sum_v \frac{|S_v|}{|S|} GI(S_v) \\ &= \frac{45}{98} - \Big(\frac{5}{14} \times (1 - (\frac{2}{5})^2 - (\frac{3}{5})^2) + \frac{4}{14} \times (1 - (\frac{4}{4})^2 - (\frac{0}{4})^2) + \frac{5}{14} \times (1 - (\frac{3}{5})^2 - (\frac{2}{5})^ \\ &= \frac{45}{98} - \Big(\frac{5}{14} \times (\frac{12}{25}) + \frac{4}{14} \times (0) + \frac{5}{14} \times (\frac{12}{25})\Big) \\ &= 0.1163 \end{aligned}$$

Info Gain for Temperature attribute (T):

$$\begin{aligned} G(S, T) &= GI(S) - \sum_v \frac{|S_v|}{|S|} GI(S_v) \\ &= \frac{45}{98} - \Big(\frac{4}{14} \times (1 - (\frac{2}{4})^2 - (\frac{2}{4})^2) + \frac{6}{14} \times (1 - (\frac{4}{6})^2 - (\frac{2}{6})^2) + \frac{4}{14} \times (1 - (\frac{3}{4})^2 - (\frac{1}{4})^ \\ &= \frac{45}{98} - \Big(\frac{4}{14} \times (\frac{1}{2}) + \frac{6}{14} \times (\frac{4}{9}) + \frac{4}{14} \times (\frac{3}{8})\Big) \\ &= 0.0197 \end{aligned}$$

Info Gain for Humidity attribute (H):

$$\begin{aligned} G(S,H) &= GI(S) - \sum_v \frac{|S_v|}{|S|} GI(S_v) \\ &= \frac{45}{98} - \Big(\frac{7}{14} \times (1 - (\frac{3}{7})^2 - (\frac{4}{7})^2) + \frac{7}{14} \times (1 - (\frac{6}{7})^2 - (\frac{1}{7})^2)\Big) \\ &= \frac{45}{98} - \Big(\frac{7}{14} \times (\frac{24}{49}) + \frac{7}{14} \times (\frac{12}{49})\Big) \\ &= 0.0918 \end{aligned}$$

Info Gain for Wind attribute (W):

$$\begin{aligned} G(S,W) &= GI(S) - \sum_v \frac{|S_v|}{|S|} GI(S_v) \\ &= \frac{45}{98} - \Big(\frac{8}{14} \times (1 - (\frac{6}{8})^2 - (\frac{2}{8})^2) + \frac{6}{14} \times (1 - (\frac{3}{6})^2 - (\frac{3}{6})^2)\Big) \\ &= \frac{45}{98} - \Big(\frac{8}{14} \times (\frac{3}{8}) + \frac{6}{14} \times (\frac{1}{2})\Big) \\ &= 0.0306 \end{aligned}$$

iii. Since Outlook has the highest information gain, we choose this as our root node and split the dataset according to Outlook values.

iv. For each attribute value of O, sunny overcast and rainy, I make a new branch of the tree. I then split the dataset based on this value, and remove the Overcast attribute from the data completely.

v. I then repeat the above for each of the three data subsets by finding the best attribute to split the data subset based on the highest information gain.

vi. I now will repeat the above steps for the Outlook=sunny data subset. The gini index for this subset is the following.

$$GI = 1 - (\frac{3}{5})^2 - (\frac{2}{5})^2 = \frac{12}{25}$$

vii. Next, I calculate the info gain for the remaining three attributes. These values are shown below.

$$Gain(T) = 0.28$$

$$Gain(H) = 0.48$$

$$Gain(W) = 0.0133$$

Since the Humidity attribute has the highest info gain, I choose this as the attribute to split the remaining data on. After the split, the data subsets all have the same target values, so we move back up the tree.

(c) Comparing the two trees I just created to the one in the lecture notes, all the trees look the same. There are no differences due to the way the data is arranged.

The individual information gains are different for each of the three cases, but the ratios between each attribute remain fairly similar. This has the effect of choosing the same attributes for splitting regardless of the information gain method.

3.

(a)

For the attribute Outlook, the most common value is both S and R, so I picked S as the missing value. Using the original information gain with entropy, I calculate the entropy of the entire new data set first.

$$\begin{aligned} H &= -\frac{10}{15}\log_2(\frac{10}{15}) - \frac{5}{15}\log_2(\frac{5}{15}) \\ &= 0.9183 \end{aligned}$$

Information gain for Outlook attribute:

$$\begin{aligned} IG &= H - [\frac{6}{15}H_{O=s} + \frac{4}{15}H_{O=o} + \frac{5}{15}H_{O=r}] \\ &= 0.9183 - [\frac{6}{15}(1) + \frac{4}{15}(0) + \frac{5}{15}(0.97)] \\ &= 0.1943 \end{aligned}$$

Information gain for Temperature attribute:

$$\begin{aligned} IG &= H - [\frac{4}{15}H_{t=hot} + \frac{7}{15}H_{t=m} + \frac{4}{15}H_{t=c}] \\ &= 0.9183 - [\frac{4}{15}(1) + \frac{7}{15}(0.985) + \frac{4}{15}(0.97)] \\ &= -0.06 \end{aligned}$$

Information gain for Humidity attribute:

$$\begin{aligned} IG &= H - [\frac{7}{15}H_{h=h} + \frac{8}{15}H_{h=n}] \\ &= 0.9183 - [\frac{7}{15}(0.985) + \frac{8}{15}(0.543)] \\ &= 0.169 \end{aligned}$$

Information gain for Wind attribute:

$$\begin{aligned} IG &= H - [\frac{9}{15}H_{w=w} + \frac{6}{15}H_{w=s}] \\ &= 0.9183 - [\frac{9}{15}(0.764) + \frac{6}{15}(1)] \\ &= 0.0599 \end{aligned}$$

As can be seen above, the best feature (highest gain) is the Outlook feature.

4. [**Bonus question 1**]

5. [**Bonus question 2**]

2 Decision Tree Practice [60 points]

1. [5 Points]

2.

   (a)

   (b) After implementing my algorithm, I learned decision trees from the training data and tested them against the test data. I varied the tree depth from 1 to 6 as well. The percent errors for both data sets are shown below for entropy, majority error, and gini index.

Max Depth	Training	Testing
1	30.20	29.67
2	22.19	22.25
3	18.10	19.64
4	8.19	15.11
5	2.70	9.89
6	0.0	12.22

Table 7: Percent errors with entropy

Max Depth	Training	Testing
1	30.20	29.67
2	29.20	31.32
3	19.29	21.15
4	11.09	17.86
5	3.6	12.36
6	0.0	14.84

Table 8: Percent errors with majority error

Max Depth	Training	Testing
1	30.20	29.67
2	22.19	22.25
3	17.6	18.41
4	8.90	13.74
5	2.70	9.89
6	0.0	12.22

Table 9: Percent errors with gini index

   (c)

As can be seen in the above tables, the overall accuracy of my algorithm is fairly high, especially at high max depth numbers. In all three cases, the worst error seen at a depth of 6 is less than 15%. Comparing the three types of information gain, we can see that the gini index gave the best results, but not by much. Both gini index and info gain results are very similar, with only slight differences at certain max depths. Majority error showed promising results as well, only slightly worse than the other two. Comparing the training and testing errors, we can see both have similar errors at low max depth. As the depth increases, the training errors converge to zero, while the testing errors converge to an error around 12%. This makes sense, as the tree will overfit the training data if it can. Another interesting point is that the lowest testing errors were at a max depth of 5, which also confirms the overfitting issue.

3. [25 points]

   (a) Unknown as a particular attribute value

Max Depth	Training	Testing
1	11.92	12.48
2	10.60	11.14
3	10.06	10.70
4	7.92	11.50
5	6.12	12.2
6	4.72	13.24
7	3.48	14.22
8	2.86	14.64
9	2.30	15.08
10	1.70	15.54
11	1.44	15.62
12	1.36	15.56
13	1.36	15.56
14	1.36	15.54
15	1.36	15.54
16	1.36	15.54

Table 10: Percent errors with entropy

   (b) Unknown as attribute value missing

   (c) As can be seen in the above tables, there is an optimal tree depth where the testing data performs the best. This depth is lower than the max depth, and appeared to be around a depth of two or three for the results above. This is due to the model overfitting the training data.

Max Depth	Training	Testing
1	10.88	11.66
2	10.42	10.88
3	9.60	11.21
4	8.18	11.82
5	7.18	11.68
6	6.74	11.94
7	6.44	12.04
8	5.92	12.32
9	5.08	12.84
10	4.28	13.8
11	3.84	14.2
12	3.24	14.74
13	2.76	15.1
14	2.02	15.58
15	1.54	15.76
16	1.36	15.76

Table 11: Percent errors with majority error