CS 5350/6350: Machine Learning Spring 2019

Homework 1

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

- You are welcome to talk to other members of the class about the homework. I am more concerned that you understand the underlying concepts. However, you should write down your own solution. Please keep the class collaboration policy in mind.
- Feel free discuss the homework with the instructor or the TAs.
- Your written solutions should be brief and clear. You need to show your work, not just the final answer, but you do *not* need to write it in gory detail. Your assignment should be **no more than 15 pages**. Every extra page will cost a point.
- Handwritten solutions will not be accepted.
- *Your code should run on the CADE machines.* You should include a shell script, `run.sh`, that will execute your code in the CADE environment. Your code should produce similar output to what you include in your report.

  You are responsible for ensuring that the grader can execute the code using only the included script. If you are using an esoteric programming language, you should make sure that its runtime is available on CADE.
- Please do not hand in binary files! We will *not* grade binary submissions.
- The homework is due by **midnight of the due date**. Please submit the homework on Canvas.

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) [5 points] Use the ID3 algorithm with information gain to learn a decision tree from the training dataset in Table 1. Please list every step in your tree construction, including the data subsets, the attributes, and how you calculate the information gain of each attribute and how you split the dataset according to the selected attribute. Please also give a full structure of the tree. You can manually draw the tree structure, convert the picture into a PDF/EPS/PNG/JPG format and include it in your homework submission; or instead, you can represent the tree with a conjunction of prediction rules as we discussed in the lecture.

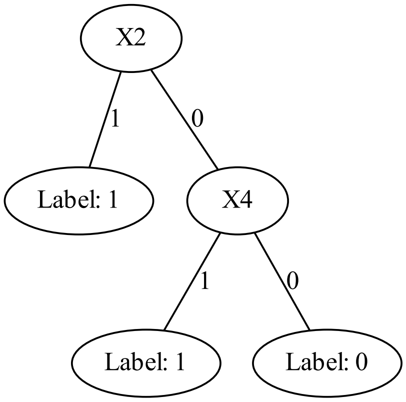
Solution:
Initial Entropy $= -(2/7)\log_2(2/7) - (5/7)\log_2(5/7) \approx 0.86$
Information gain of $x_1 = 0.86 - (\frac{5}{7}(.72) + \frac{2}{7}(1.0)) \approx 0.06$
Information gain of $x_2 = 0.86 - (\frac{3}{7}(.92) + \frac{4}{7}(.39)) \approx 0.47$
Information gain of $x_3 = 0.86 - (\frac{4}{7}(.81) + \frac{3}{7}(.92)) \approx 0.006$
Information gain of $x_4 = 0.86 - (\frac{4}{7}(0.0) + \frac{3}{7}(.92)) \approx 0.47$ (Same as x_2)
Split on x_2, giving S1 containing items [1, 3, 4] and S2 containing items [2, 5, 6, 7]

S1 Entropy: $-(1/3)\log_2(1/3) - (2/3)\log_2(2/3) \approx 0.92$
Information gain of $x_1 = 0.92 - (\frac{2}{3}(1.0) + \frac{1}{3}(0.0)) \approx 0.25$
Information gain of $x_3 = 0.92 - (\frac{1}{3}(0.0) + \frac{2}{3}(1.0)) \approx 0.25$
Information gain of $x_4 = 0.92 - (\frac{1}{3}(0.0) + \frac{2}{3}(0.0)) \approx 0.92$
Split on x_4, giving S3 = [1] and S4 = [3, 4]

S3 Entropy: $-(1/1)\log_2(1/1) - (0/1)\log_2(0/1) = 0$
S4 Entropy: $-(2/2)\log_2(2/2) - (0/2)\log_2(0/2) = 0$

S2 Entropy $= -(4/4)\log_2(4/4) - (0/4)\log_2(0/4) = 0$

Final Tree:



(b) [2 points] Write the boolean function which your decision tree represents. Please use a table to describe the function — the columns are the input variables and label, i.e., x_1, x_2, x_3, x_4 and y; the rows are different input and function values.

Solution:
$y = \neg x_2 \wedge x_4$

2. [17 points] Let us use a training dataset to learn a decision tree about whether to play tennis (**Page 39, Lecture: Decision Tree Learning**, accessible by clicking the link http://www.cs.utah.edu/˜zhe/teach/pdf/decision-trees-learning.pdf). In the class, we have shown how to use information gain to construct the tree in ID3 framework.

   (a) [7 points] Now, please use majority error (ME) to calculate the gain, and select the best feature to split the data in ID3 framework. As in problem 1, please list every step in your tree construction, the attributes, how you calculate the gain of each attribute and how you split the dataset according to the selected attribute. Please also give a full structure of the tree.

Solution:
Initial Majority Error $= \frac{5}{14} \approx 0.36$
ME for Outlook $= 0.36 - (\frac{5}{14}(0.4) + \frac{5}{14}(0.4) + \frac{4}{14}(0.0)) \approx 0.07$
ME for Temperature $= 0.36 - (\frac{4}{14}(0.5) + \frac{4}{14}(0.25) + \frac{6}{14}(0.33)) \approx$ 5.5e-17
ME for Humidity $= 0.36 - (\frac{7}{14}(0.43) + \frac{7}{14}(0.14) \approx 0.07$
ME for Wind $= 0.36 - (\frac{6}{14}(0.5) + \frac{8}{14}(0.25) \approx$ 5.5e-17
Split on Humidity (It was <1e-15 larger than Outlook), giving S1 (High) = [1, 2, 3, 4, 8, 12, 14] and S2 (Normal) = [5, 6, 7, 9, 10, 11, 13]

S1 Majority Error $= \frac{3}{7} \approx 0.42$
ME for Outlook $= 0.42 - (\frac{3}{7}(0.0) + \frac{2}{7}(0.5) + \frac{2}{7}(0.0)) \approx 0.29$
ME for Temperature $= 0.42 - (\frac{3}{7}(0.33) + \frac{4}{7}(0.5) = 0.0$
ME for Wind $= 0.42 - (\frac{3}{7}(0.33) + \frac{4}{7}(0.5) = 0.0$
Split on Outlook, giving S3 (Sunny) = [1, 2, 8], S4 (Overcast) = [3, 12], and S5 (Rainy) = [4, 14]

S3 Majority Error $= \frac{0}{3} = 0.0$, Done.
S4 Majority Error $= \frac{0}{2} = 0.0$, Done.

S5 Majority Error $= \frac{1}{2} = 0.5$
ME for Temperature $= 0.5 - (\frac{2}{2}(0.5) + \frac{0}{2}(0.0) = 0.0$
ME for Wind $= 0.5 - (\frac{1}{2}(0.0) + \frac{1}{2}(0.0) = 0.5$
Split on Wind, giving S6 (Strong) = [14] and S7 (Weak) = [4]

S6 Majority Error $= \frac{0}{1} = 0.0$, Done.
S7 Majority Error $= \frac{0}{1} = 0.0$, Done.

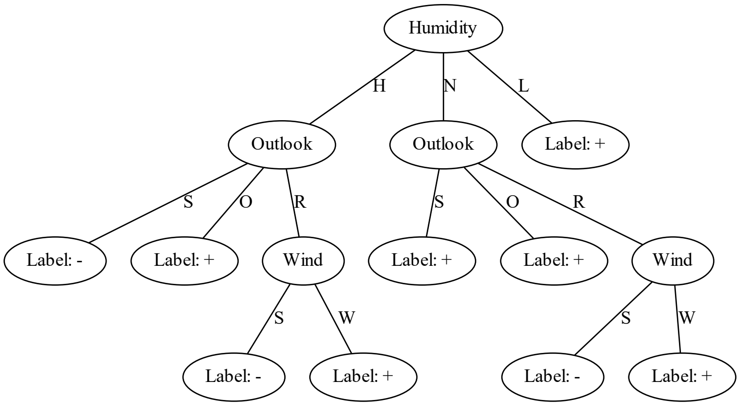
S2 Majority Error $= \frac{1}{7} \approx 0.14$
ME for Outlook $= 0.14 - (\frac{2}{7}(0.0) + \frac{3}{7}(0.33) + \frac{2}{7}(0.0)) = 0.0$
ME for Temperature $= 0.14 - (\frac{1}{7}(0.0) + \frac{4}{7}(0.25) + \frac{2}{7}(0.0)) = 0.0$
ME for Wind $= 0.14 - (\frac{3}{7}(0.33) + \frac{4}{7}(0.0)) = 0.0$
Break the tie and split on Outlook, giving S8 (Sunny) = [9, 11], S9 (Rainy) = [5, 6, 10], and S10 (Overcast) = [7, 13]

S8 Majority Error $= \frac{0}{2} = 0.0$, Done.

S8 Majority Error $= \frac{1}{3} \approx 0.33$
ME for Temperature $= 0.33 - (\frac{2}{3}(0.5) + \frac{1}{3}(0.5)) = 0.0$
ME for Wind $= 0.33 - (\frac{1}{3}(0.0) + \frac{2}{3}(0.0)) \approx 0.33$
Split on Wind, giving S11 (Strong) = [11] and S12 (Weak) = [9]

S11 Majority Error $= \frac{0}{1} = 0.0$, Done.
S12 Majority Error $= \frac{0}{1} = 0.0$, Done.
S10 Majority Error $= \frac{0}{2} = 0.0$, Done.

Final Tree:



(b) [7 points] Please use gini index (GI) to calculate the gain, and conduct tree learning with ID3 framework. List every step and the tree structure.

Solution:
Initial Gini Index $= 1 - (\frac{9}{14}^2 + \frac{5}{14}^2) \approx 0.46$

GI for Outlook $= 0.46 - (\frac{5}{14}(0.48) + \frac{5}{14}(0.48) + \frac{4}{14}(0.00)) \approx 0.12$

GI for Temperature $= 0.46 - (\frac{4}{14}(0.50) + \frac{4}{14}(0.38) + \frac{6}{14}(0.44)) \approx 0.02$

GI for Humidity $= 0.46 - (\frac{7}{14}(0.49) + \frac{7}{14}(0.24)) \approx 0.09$

GI for Wind $= 0.46 - (\frac{6}{14}(0.50) + \frac{8}{14}(0.38)) \approx 0.03$

Split on Outlook, giving S1 (Sunny) = [1, 2, 8, 9, 11], S2 (Rainy) = [4, 5, 6, 10, 14], and S3 (Overcast) = [3, 7, 12, 13]

S1 Gini Index $= 1 - (\frac{2}{5}^2 + \frac{3}{5}^2) = 0.48$

GI for Temperature $= 0.48 - (\frac{2}{5}(0.00) + \frac{1}{5}(0.00) + \frac{2}{5}(0.50)) \approx 0.28$

GI for Humidity $= 0.48 - (\frac{3}{5}(0.00) + \frac{2}{5}(0.00)) \approx 0.48$

GI for Wind $= 0.48 - (\frac{2}{5}(0.50) + \frac{3}{5}(0.44)) \approx 0.01$

Split on Humidity, giving S4 (High) = [1, 2, 8], S5 (Normal) = [9, 11]

S4 Gini Index $= 1 - (\frac{3}{3}^2 + \frac{0}{3}^2) = 0.00$, Done.

S5 Gini Index $= 1 - (\frac{2}{2}^2 + \frac{0}{2}^2) = 0.00$, Done.

S2 Gini Index $= 1 - (\frac{3}{5}^2 + \frac{2}{5}^2) = 0.48$

GI for Temperature $= 0.48 - (\frac{2}{5}(0.50) + \frac{3}{5}(0.44)+) \approx 0.01$

GI for Humidity $= 0.48 - (\frac{2}{5}(0.50) + \frac{3}{5}(0.44)+) \approx 0.01$

GI for Wind $= 0.48 - (\frac{2}{5}(0.00) + \frac{3}{5}(0.00)+) \approx 0.48$

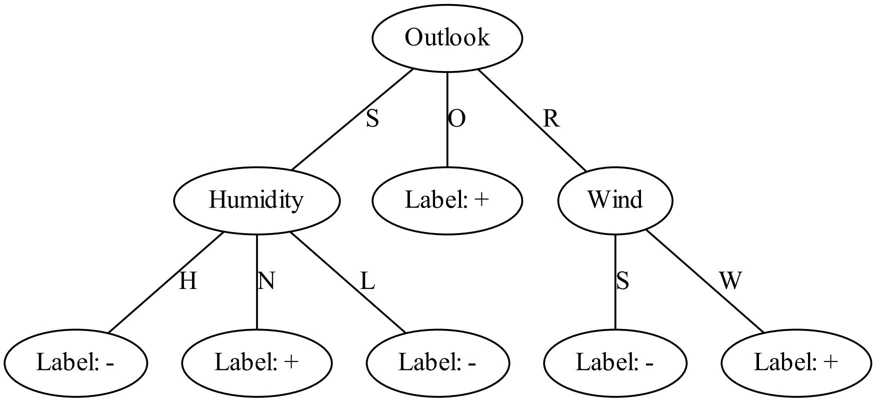
Split on Wind, giving S6 (Strong) = [6, 14] and S7 (Weak) = [4, 5, 10]

S6 Gini Index $= 1 - (\frac{2}{2}^2 + \frac{0}{2}^2) = 0.00$, Done.

S7 Gini Index $= 1 - (\frac{3}{3}^2 + \frac{0}{3}^2) = 0.00$, Done.

S3 Gini Index $= 1 - (\frac{4}{4}^2 + \frac{0}{4}^2) = 0.00$, Done.

Final Tree:



(c) [3 points] Compare the two trees you just created with the one we built in the class (see Page 58 of the lecture slides). Are there any differences? Why?

Solution:
My tree created using the gini index is exactly the same as the one created using entropy/information gain. However my tree created using majority error splits with Humidity first, then Outlook. This is because it had a marginally higher gain than Outlook, perhaps due to floating point rounding errors.

3. [16 points] Continue with the same training data in Problem 2. Suppose before the tree construction, we receive one more training instance where Outlook's value is missing: {Outlook: Missing, Temperature: Mild, Humidity: Normal, Wind: Weak, Play: Yes}.

(a) [3 points] Use the most common value in the training data as the missing value, and calculate the information gains of the four features. Indicate the best feature.

Solution:
New Outlook value: Sunny (tied with Rainy)

Initial Entropy $= -(10/15)\log_2(10/15) - (5/15)\log_2(5/15) \approx 0.92$

Information gain of Outlook $= 0.92 - (\frac{6}{15}(1.0) + \frac{4}{15}(0.0) + \frac{5}{15}(.97)) \approx 0.19$

Information gain of Temperature $= 0.92 - (\frac{4}{15}(1.0) + \frac{7}{15}(.86) + \frac{4}{15}(.81)) \approx 0.03$

Information gain of Wind $= 0.92 - (\frac{6}{15}(1.0) + \frac{9}{15}(.76)) \approx 0.06$

Information gain of Humidity $= 0.92 - (\frac{7}{15}(.99) + \frac{8}{15}(.54)) \approx 0.17$

Best Feature: Outlook

(b) [3 points] Use the most common value among the training instances with the same label, namely, their attribute "Play" is "Yes", and calculate the information gains of the four features. Indicate the best feature

Solution:
New Outlook value: Overcast

Information gain of Outlook $= 0.92 - (\frac{5}{15}(.97) + \frac{5}{15}(0.0) + \frac{5}{15}(.97)) \approx 0.27$

Information gain of Temperature $= 0.92 - (\frac{4}{15}(1.0) + \frac{7}{15}(.86) + \frac{4}{15}(.81)) \approx 0.03$

Information gain of Wind $= 0.92 - (\frac{6}{15}(1.0) + \frac{9}{15}(.76)) \approx 0.06$

Information gain of Humidity $= 0.92 - (\frac{7}{15}(.99) + \frac{8}{15}(.54)) \approx 0.17$

Best Feature: Outlook

(c) [3 points] Use the fractional counts to infer the feature values, and then calculate the information gains of the four features. Indicate the best feature.

Solution:
New Outlook value: 5/14 Sunny, 4/14 Overcast, 5/14 Rainy

Information gain of Outlook $= 0.92 - (\frac{5.36}{15}(.99) + \frac{4.29}{15}(0.0) + \frac{5.36}{15}(.95)) \approx 0.22$

Information gain of Temperature $= 0.92 - (\frac{4}{15}(1.0) + \frac{7}{15}(.86) + \frac{4}{15}(.81)) \approx 0.03$

Information gain of Wind $= 0.92 - (\frac{6}{15}(1.0) + \frac{9}{15}(.76)) \approx 0.06$

Information gain of Humidity $= 0.92 - (\frac{7}{15}(.99) + \frac{8}{15}(.54)) \approx 0.17$

Best Feature: Outlook

(d) [7 points] Continue with the fractional examples, and build the whole free with information gain. List every step and the final tree structure.

Solution:
New Outlook value: 5/14 Sunny, 4/14 Overcast, 5/14 Rainy

Information gain of Outlook $= 0.92 - (\frac{5.36}{15}(.99) + \frac{4.29}{15}(0.0) + \frac{5.36}{15}(.95)) \approx 0.22$

Information gain of Temperature $= 0.92 - (\frac{4}{15}(1.0) + \frac{7}{15}(.86) + \frac{4}{15}(.81)) \approx 0.03$

Information gain of Wind $= 0.92 - (\frac{6}{15}(1.0) + \frac{9}{15}(.76)) \approx 0.06$

Information gain of Humidity $= 0.92 - (\frac{7}{15}(.99) + \frac{8}{15}(.54)) \approx 0.17$

Split on Outlook, giving S1 (Sunny) = [1, 2, 8, 9, 11, 15], S2 (Overcast) = [3, 7, 12, 13, 16], and S3 (Rainy) = [4, 5, 6, 10, 14, 17]

S1 Initial Entropy $= -(2.35/5.35)\log_2(2.35/5.35) - (3/5.35)\log_2(3/5.35) \approx 0.99$

Info gain for Temperature $=0.99 - (\frac{2}{5.36}(0.00) + \frac{2.36}{5.36}(0.98) + \frac{1}{5.36}(0.00)) \approx 0.56$

Info gain for Wind $= 0.99 - (\frac{2}{5.36}(1.00) + \frac{3.36}{5.36}(0.97)) \approx 0.01$

Info gain for Humidity $= 0.99 - (\frac{3}{5.36}(0.00) + \frac{2.36}{5.36}(0.00) + \frac{0}{5.36}(0.00)) \approx 0.99$

Split on Humidity, giving S4 (High) = [1, 2, 8], S5 (Normal) = [9, 11, 15], and S6 (Low) = []

S4 Initial Entropy $= -(0/3)\log_2(0/3) - (3/3)\log_2(3/3) = 0$, Done.

S4 Initial Entropy $= -(0/2.36)\log_2(0/2.36) - (2.36/2.36)\log_2(2.36/2.36) = 0$, Done.

S6 Initial Entropy = 0, Done.

S2 Initial Entropy $= -(4.29/4.29)\log_2(4.29/4.29)) - (0/4.29)\log_2(0/4.29)) = 0$, Done.

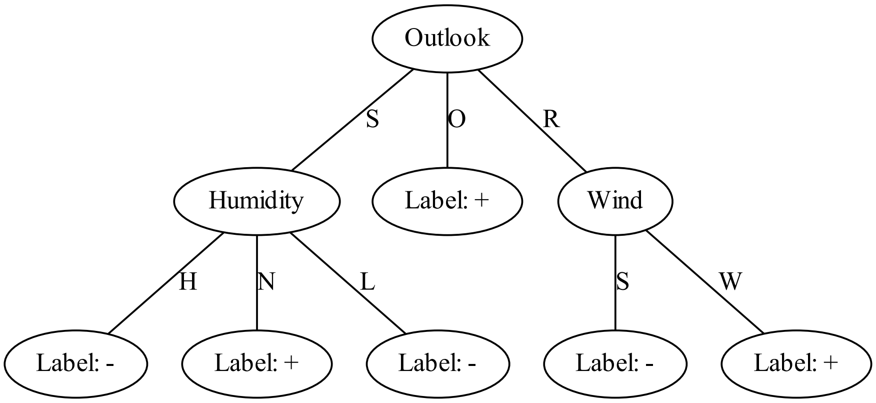
S3 Initial Entropy $= -(3.35/5.35)\log_2(3.35/5.35) - (2/5.35)\log_2(2/5.35) \approx 0.95$

Info gain for Temperature $= 0.95 - (\frac{0.00}{5.36}(0.00) + \frac{3.36}{5.36}(0.88) + \frac{2.00}{5.36}(1.00)) \approx 0.03$

Info gain for Wind $= 0.95 - (\frac{2.00}{5.36}(0.00) + \frac{3.36}{5.36}(0.00)+) \approx 0.95$
Info gain for Humidity $= 0.95 - (\frac{2.00}{5.36}(1.00) + \frac{3.36}{5.36}(0.88) + \frac{0.00}{5.36}(0.00)) \approx 0.03$
Split on Wind, giving S7 (Strong) = [6, 14] and S8 (Weak) = [4, 5, 10, 17]

S7 Initial Entropy $= -(0/2)\log_2(0/2) - (2/2)\log_2(2/2) = 0$, Done.
S8 Initial Entropy $= -(4.36/4.36)\log_2(4.36/4.36)) - (0/4.36)\log_2(0/4.36)) = 0$, Done.

Final Tree:



4. [**Bonus question 1**] [5 points]. Prove that the information gain is always non-negative. That means, as long as we split the data, the purity will never get worse! (Hint: use convexity)

5. [**Bonus question 2**] [5 points]. We have discussed how to use decision tree for regression (i.e., predict numerical values) — on the leaf node, we simply use the average of the (numerical) labels as the prediction. Now, to construct a regression tree, can you invent a gain to select the best attribute to split data in ID3 framework?

Solution:
Instead of information gain I would use some kind of least variance to split data. You want to converge on one number for any set of attributes, therefore you want to make splits that remove variance from the resulting subsets.

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

2 Decision Tree Practice [60 points]

1. [5 Points] Starting from this assignment, we will build a light-weighted machine learning library. To this end, you will first need to create a code repository in Github.com. Please refer to the short introduction in the appendix and the official tutorial to create an account and repository. Please commit a README.md file in your repository, and write one sentence: "This is a machine learning library developed by **Your Name** for CS5350/6350 in University of Utah". You can now create a first folder, "DecisionTree". Please leave the link to your repository in the homework submission. We will check if you have successfully created it.

   `https://github.com/willfrank98/5350-Machine_Learning`

2. [30 points] We will implement a decision tree learning algorithm for car evaluation task. The dataset is from UCI repository(`https://archive.ics.uci.edu/ml/datasets/car+evaluation`). Please download the processed dataset (car.zip) from Canvas. In this task, we have 6 car attributes, and the label is the evaluation of the car. The attribute and label values are listed in the file "data-desc.txt". All the attributes are categorical. The training data are stored in the file "train.csv", consisting of 1,000 examples. The test data are stored in "test.csv", and comprise 728 examples. In both training and test datasets, attribute values are separated by commas; the file "data-desc.txt" lists the attribute names in each column.

   Note: we highly recommend you to use Python for implementation, because it is very convenient to load the data and handle strings. For example, the following snippet reads the CSV file line by line and split the values of the attributes and the label into a list, "terms". You can also use "dictionary" to store the categorical attribute values. In the web are numerous tutorials and examples for Python. if you have issues, just google it!

```
with open(CSVfile, 'r') as f:
     for line in f:
          terms = line.strip().split(',')
          process one training example
```

   (a) [15 points] Implement the ID3 algorithm that supports, information gain, majority error and gini index to select attributes for data splits. Besides, your ID3 should allow users to set the maximum tree depth. Note: you do not need to convert categorical attributes into binary ones and your tree can be wide here.

   (b) [10 points] Use your implemented algorithm to learn decision trees from the training data. Vary the maximum tree depth from 1 to 6 — for each setting, run your algorithm to learn a decision tree, and use the tree to predict both the training and test examples. Note that if your tree cannot grow up to 6 levels, then you can stop at the maximum level. Report in a table the average prediction errors on each dataset when you use information gain, majority error and gini index heuristics, respectively.

Solution:
Training Error:

Tree Depth =	1	2	3	4	5	6
Information Gain	0.302	0.222	0.181	0.082	0.027	0.0
Majority Error	0.302	0.222	0.174	0.089	0.027	0.0
Gini Index	0.302	0.222	0.176	0.089	0.027	0.0

Testing Error:

Tree Depth =	1	2	3	4	5	6
Information Gain	0.297	0.222	0.196	0.151	0.084	0.084
Majority Error	0.297	0.222	0.187	0.137	0.084	0.084
Gini Index	0.297	0.222	0.184	0.137	0.084	0.084

(c) [5 points] What can you conclude by comparing the training errors and the test errors?

Solution:
For all algorithms a depth of 6 was enough to eliminate all training error. However there are obviously some examples in the test data which do not conform to the tree created from the training data, as even at depth 6 there is testing error across the board. Trees of depth 1 are also more accurate on the test data than the training data.

3. [25 points] Next, modify your implementation a little bit to support numerical attributes. We will use a simple approach to convert a numerical feature to a binary one. We choose the median (NOT the average) of the attribute values (in the training set) as the threshold, and examine if the feature is bigger (or less) than the threshold. We will use another real dataset from UCI repository(`https://archive.ics.uci.edu/ml/datasets/Bank+Marketing`). This dataset contains 16 attributes, including both numerical and categorical ones. Please download the processed dataset from Canvas (bank.zip). The attribute and label values are listed in the file "data-desc.txt". The training set is the file "train.csv", consisting of 5,000 examples, and the test "test.csv" with 5,000 examples as well. In both training and test datasets, attribute values are separated by commas; the file "data-desc.txt" lists the attribute names in each column.

   (a) [10 points] Let us consider "unknown" as a particular attribute value, and hence we do not have any missing attributes for both training and test. Vary the maximum tree depth from 1 to 16 — for each setting, run your algorithm to learn a decision tree, and use the tree to predict both the training and test examples. Again, if your tree cannot grow up to 16 levels, stop at the maximum level. Report in a table the average prediction errors on each dataset when you use information gain, majority error and gini index heuristics, respectively.

Solution:
Training Error:

Tree Depth =	1	2	3	4	5	6	7	8
Information Gain	0.1088	0.107	0.0934	0.0766	0.0668	0.0606	0.0584	0.0572
Majority Error	0.1088	0.1066	0.1006	0.0948	0.0828	0.0744	0.0716	0.0708
Gini Index	0.1088	0.107	0.0932	0.0762	0.0654	0.0602	0.0584	0.0572

Tree Depth =	9	10	11	12	13	14	15	16
Information Gain	0.0572	0.0572	0.0572	0.0572	0.0572	0.0572	0.0572	0.0572
Majority Error	0.07	0.0682	0.0678	0.065	0.063	0.0614	0.0602	0.0572
Gini Index	0.0572	0.0572	0.0572	0.0572	0.0572	0.0572	0.0572	0.0572

Testing Error:

Tree Depth =	1	2	3	4	5	6	7	8
Information Gain	0.1166	0.114	0.123	0.1348	0.1434	0.1466	0.1558	0.1562
Majority Error	0.1166	0.1134	0.117	0.1206	0.124	0.125	0.1256	0.1256
Gini Index	0.1166	0.114	0.123	0.1368	0.145	0.1516	0.1594	0.1592

Tree Depth =	9	10	11	12	13	14	15	16
Information Gain	0.1574	0.1574	0.1574	0.1574	0.1574	0.1574	0.1574	0.1574
Majority Error	0.1256	0.1256	0.1256	0.1256	0.1256	0.1256	0.1256	0.1256
Gini Index	0.1598	0.1598	0.1598	0.1598	0.1598	0.1598	0.1598	0.1598

(b) [10 points] Let us consider ”unknown” as attribute value missing. Here we simply complete it with the majority of other values of the same attribute in the training set. Vary the maximum tree depth from 1 to 16 — for each setting, run your algorithm to learn a decision tree, and use the tree to predict both the training and test examples. Report in a table the average prediction errors on each dataset when you use information gain, majority error and gini index heuristics, respectively.

Solution:

Training Error:

Tree Depth =	1	2	3	4	5	6	7	8
Information Gain	0.1088	0.1078	0.0968	0.085	0.0782	0.0728	0.0706	0.0702
Majority Error	0.1088	0.1074	0.1008	0.0944	0.0864	0.0836	0.082	0.0816
Gini Index	0.1088	0.1078	0.0968	0.0848	0.0782	0.0724	0.0706	0.0702

Tree Depth =	9	10	11	12	13	14	15	16
Information Gain	0.0702	0.0702	0.0702	0.0702	0.0702	0.0702	0.0702	0.0702
Majority Error	0.081	0.0808	0.0806	0.0804	0.0798	0.0764	0.073	0.070
Gini Index	0.0702	0.0702	0.0702	0.0702	0.0702	0.0702	0.0702	0.0702

Testing Error:

Tree Depth =	1	2	3	4	5	6	7	8
Information Gain	0.1166	0.1154	0.1208	0.1296	0.1336	0.143	0.1448	0.147
Majority Error	0.1166	0.1148	0.1178	0.1186	0.1198	0.1204	0.12	0.12
Gini Index	0.1166	0.1154	0.1208	0.1308	0.1364	0.1436	0.1454	0.1476

Tree Depth =	9	10	11	12	13	14	15	16
Information Gain	0.1476	0.1476	0.1476	0.1476	0.1476	0.1476	0.1476	0.1476
Majority Error	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12
Gini Index	0.1482	0.1482	0.1482	0.1482	0.1482	0.1482	0.1482	0.1482

(c) [5 points] What can you conclude by comparing the training errors and the test errors, with different tree depths, as well as different ways to deal with ”unknown” attribute values?

Solution:

On this larger data set training error never goes to 0, and testing error increases after a tree depth of 2, likely due to some kind of over fitting. Training error usually reaches a minimum after a depth of 8, except for Majority Error, and testing error usually reached a maximum at a tree depth of 9. Replacing ”unknown” with the most common training value results in higher training error, but lower testing error.