CS 5350/6350: Machine Learning Fall 2019

Ritesh Sharma

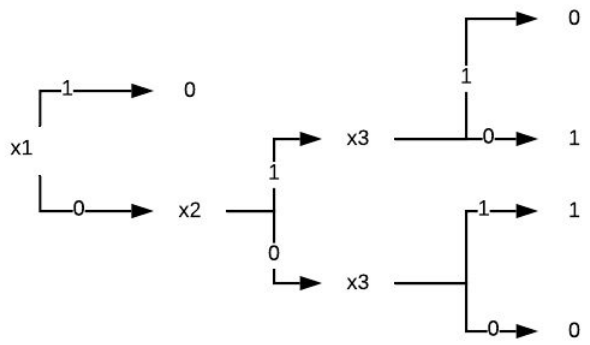
Homework 1

Handed out: August 28, 2019
Due date: September 10, 2019

1 Decision Trees

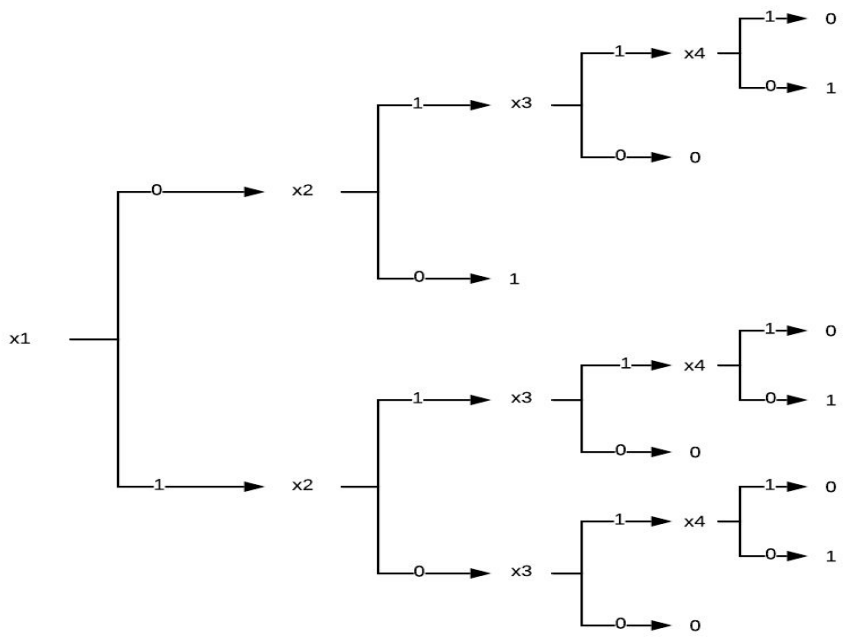
1. [6 points] Please indicate which of the following Boolean functions can be represented by decision trees or by linear classifiers or both. Show the corresponding decision tree or linear threshold unit if your answer is yes. (You can write your decision trees as a series of if-then-else statements, or use your favorite drawing program to draw a tree. You can use 1 to represent True and 0 to represent False. Also, recall that a linear classifier is described by a weight vector w and a bias b. The classifier predicts 1 if $w^T x + b \geq 0$ and -1 otherwise).

(a) $\neg x1 \wedge (x2 \text{ xor } x3)$



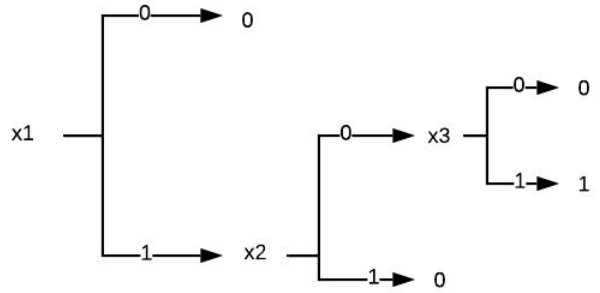
Boolean function can be represented as:
(1-x1) + ((1-x2)x3 + x2(1-x3)) >= 3
The function above is not linear so it cannot be represented as linear classifier.

(b) $(x1 \text{ nor } x2) \vee (x3 \wedge (\neg x4))$



Boolean function above can be represented as:
(1-x1)(1-x2) * (x3 + (1-x4)) >= 4
Since the function cannot be represented as linear classifier.

(c) x1 ∧ ¬x 2 ∧ x3



Boolean funtion above can be represented as:
x1 + (1-x2) + x3 >= 3, it is a linear function, so it can be represented as linear classifier.

2. [24 points] A group of CS college students want to rent an apartment. They want an automatic suggestion system that takes information from new posted apartments online and decides whether or not the apartment is worth renting. For this, they use

the following features:
(a) number of rooms (one, two, three or four rooms),
(b) apartment condition (poor, fair, good, excellent),
(c) distance from college (less than 1 mile, between 1 and 5 miles, more than 5 miles), and
(d) price (less than $500, between $500 and $1000, more than $1000).

(a) [3 points] How many possible functions are there to map these four features to a boolean decision (0 or 1)? How many functions are consistent with the given training dataset?

Answer : $2^{4*4*3*3} = \mathbf{2^{144}}$. All the functions are consistent with the given training dataset.

(b) [3 points] What is the entropy of the labels in this data? When calculating entropy, the base of the logarithm should be base 2.

Answer: $H(S) = -p_+\log(p_+) - p_-\log(p_-) = \mathbf{0.99}$, here $p_+ = 0.56$ and $p_- = 0.44$.

(c) [8 points] Compute the information gain of each feature and enter it into Table 2. Specify up to 3 decimal places.

Answer: Used type as an example given below:

First, get the entropy according to:

$H_{one}(S) = -p_+\log(p_+) - p_-\log(p_-) = 0$
$H_{Two}(S) = -p_+\log(p_+) - p_-\log(p_-) = 1$
$H_{Three}(S) = -p_+\log(p_+) - p_-\log(p_-) = 0.9182$
$H_{Four}(S) = -p_+\log(p_+) - p_-\log(p_-) = 0.9182$

Then,

$$\text{Gain}(S, \text{Type}) = \text{Entropy}(S) - \sum_{v \in \text{Value(Type)}} |S| / S \ \text{Entropy}(S_v)$$
$$= 0.99 - ((3/16 * 0) + (4/16 * 1) + (6/16 * 0.9182) + (3/16 * 0.9182))$$
$$= 0.2234$$

Table 2

Feature	Information Gain
Number of Rooms	**0.223**
Appartment Condition	**0.014**
Distance	**0.197**
Price	**0.185**

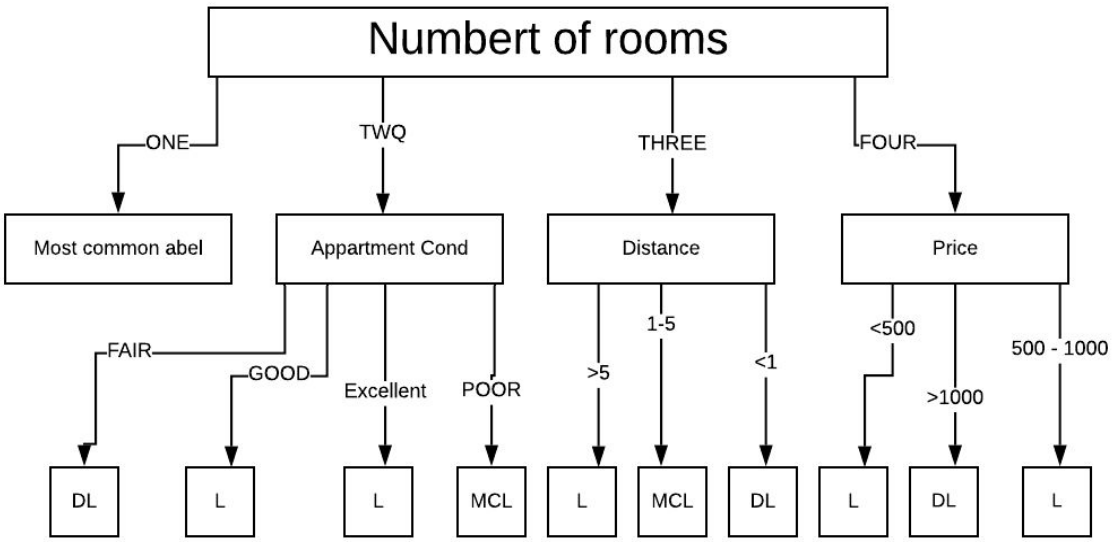
(d) [2 points] Which attribute will you use to construct the root of the tree using the

information gain heuristic of the ID3 algorithm?

Answer: Number of Rooms, because it has the most Information Gain.

(e) [4 points] Using the root that you selected in the previous question, construct a depth limited decision tree that represents the data. The max depth should be two (count maximum two branches down from the root node to the leaf). If at depth 2 you have not assigned a label, use the most common label in the dataset as the label on the leaf. You do not have to use the ID3 algorithm here, you can show any tree with the chosen root.

Answer:
Dislike : DL
Like : L
Most common label : MCL



(f) [4 points] Suppose you are given ten more examples, listed in Table 3. Use your decision tree to predict the label for each example. Also, report the accuracy of the classifier that you have learned.

Answer : After using the decision tree to predict the label for each example, the accuracy was 9/10.

Accuracy = 90% and Error = 10%

3. [10 points] Recall that in the ID3 algorithm, we want to identify the best attribute that splits the examples that are relatively pure in one label. Aside from entropy, which we saw in class and you used in the previous question, Gini impurity is a measure of how often a randomly chosen example from the training set would be incorrectly labeled if

it was randomly labeled according to the distribution of labels in the subset. We will use one of them called the Gini Index, which is calculated by subtracting the sum of the squared probabilities of each class from one.

(a) [4 points] Write down an expression that defines a new version of information gain that uses Gini Index in place of entropy.

Answer:

$$\text{Gain}(S, \text{Type}) = \text{Gini}(S) - \sum_{v \in \text{Value(Type)}} |S_v| / S \; \text{Gini}(S_v)$$

(b) [4 points] Calculate the value of your newly defined information gain from the previous question for the four features in the apartment dataset from 1. Use 3 significant digits. Enter the information gain into Table 4.

Answer: Answer: Used type as an example given below:

First, we get the entropy,

$$\text{Gini}(S) = 1 - (p_+^2 + p_-^2)$$
$$= 1 - (9/16)^2 - (7/16)^2$$
$$= \mathbf{0.4921}$$

$$\text{Gini}_{one} = 1 - (p_+^2 + p_-^2) = 0$$
$$\text{Gini}_{two} = 1 - (p_+^2 + p_-^2) = 0.5$$
$$\text{Gini}_{three} = 1 - (p_+^2 + p_-^2) = 0.444$$
$$\text{Gini}_{four} = 1 - (p_+^2 + p_-^2) = 0.444$$

Now, Using formula from part A,

$$\text{Gain}(S, \text{Type}) = 0.4921 - ((3/16) * 0 + (4/16) * 1 + (6/16) * 0.444 + (3/16) * 0.444)$$
$$= 0.4921 - 0.4999$$
$$= - 0.0078$$

Table 4

Feature	Information Gain(using Gini)
Number of Rooms	**-0.008**
Appartment Condition	**0.126**
Distance	**0.372**
Price	**0.120**

(c) [2 points] According to your results in the last question, which attribute should be

the root for the decision tree? Do these two measures (entropy and Gini Index) lead to the same tree?

Answer: Distance will be the root of the decision tree.
Since the root is different tree must be different.

2 Experiments

1. Implementation: Full trees
For this problem, your should use the data in data folder. This folder contains two files: train.csv and test.csv. You should train your algorithm on the training file. Remember that you should not look at or use your testing file until your training is complete.
In the first set of experiments, run the ID3 algorithm we saw in class without any depth restrictions. (That is, there are no hyperparameters for this setting.)

(a) [6 points] Implement the decision tree data structure and the ID3 algorithm for your decision tree (Remember that the decision tree need not be a binary tree!). For debugging your implementation, you can use the previous toy examples like the apartment data from Table 1. Discuss what approaches and design choices you had to make for your implementation and what data structures you used.

Answer: Used Data structure: Dictionary, List.
Build decision tree using Entropy, Information gain and recursion.

(b) Report the error of your decision tree on all the examples data/train.csv.
Answer:
Error = 0, because decision tree fits the whole training data, which gives zero error.
Accuracy = 1.0

(c) Report the error of your decision tree on the examples in data/test.csv.
Answer:
Error = 0.1380
Accuracy = 0.8618
This means that the decision tree is 86% accurate for testing data and has about 13% errors.

(d) Report the maximum depth of your decision tree.
Answer:
Max depth = 9, it may vary by one because depth = 0 can mean a tree with just a single node and no leaves and can also mean a tree with single node and several leaves from that node.

Your report should include the following information

(a) [2 points] The root feature that is selected by your algorithm
Answer: Spore-print-color(Index 19)

(b) [2 point] Information gain for the root feature
Answer: 0.4237

(c) [2 points] Maximum depth of the tree that your implementation gives
Answer: Max depth = 9

(d) [3 points] Error on the training set
Answer: Error on the traning set = 0, because tree will fit the whole training data.

(e) [5 points] Error on the test set
Answer: Error on test set: 0.1381

3. Limiting Depth
Next, you will perform 5-fold cross-validation to limit the depth of your decision tree, effectively pruning the tree to avoid overfitting. We have already randomly split the training data into five splits. You should use the 5 cross-validation files for this section, titled data/CVfolds/foldX.csv where X is a number between 1 and 5 (inclusive).

(a) [20 points] Run 5-fold cross-validation using the specified files. Experiment with depths in the set 1, 2, 3, 4, 5, 10, 15, reporting the average cross-validation accuracy and standard deviation for each depth. Explicitly specify which depth should be chosen as the best, and explain why.

Maximum Depth	Average Accuracy	Standard Deviation
1	0.07067	0.0104
2	0.08646	0.0124
3	0.09849	0.0107
4	0.22101	0.0434
5	0.54135	0.0236
9	0.85112	0.0219
10	0.85112	0.0219
15	0.85112	0.0219

- Depth 9 should be chosen as the best because depth 9 has the highest average accuracy.

(b) [15 points] Using the depth with the greatest cross-validation accuracy from your experiments: train your decision tree on the data/train.csv file. Report the accuracy of your decision tree on the data/test.csv file.

Answer:

Depth with greatest cross-validation accuracy is 9. This is the original depth of the decision tree that was produced without the hyper-parameter depth.

Accuracy of decision tree on the data test.csv : **0.86186**

(c) [5 points] Discuss the performance of the depth limited tree as compared to the full decision tree. Do you think limiting depth is a good idea? Why?

Answer:
For the given data set train.csv and test.csv, the full decision tree produced a depth of 9 with accuracy 0.86186. After doing cross validation on folds, the depth with the greatest cross validation accuracy was also 9. That mean for this given training data the ID3 algorithm produced the best possible depth.

The performance with limiting depth for this set of data was relatively lower than not limiting depth.
Limiting depth can be done to prevent over-fitting, it can be good idea if training data and test data largely vary.

Experiment Submission Guidelines

First imported all the files, then ran an ID3 algorithm using different functions such as calculating entropy which calculates entropy, split tree with takes the root of the tree and splits the tree. info_gain which calculated the information gain stores it in a list and returns the maximum information gain, column extractor which extracts a single column from the data, feature extractor which extracts only features from the data leaving labels, compare which compares or predicts the accuracy using two data sets testing and training, it also uses the set of rule and finally a crossvalidation funtion which takes in folds and computes the cross validation accuracy and standard deviation.

Most common label in the data with number(train): **('e ', 787)**
Entropy of the data(train): **0.9755834948606119**
Best Feature and Information Gain(Number is labels[0-21]: **19**
Accuracy on the training set: **0.9969924812030075**
Accuracy on the test set: **0.8618618618618619**
Cross_validation accuracies for each fold: **[0.8609022556390977, 0.8195488721804511, 0.868421052631579, 0.8308270676691729, 0.8759398496240601]**
Best Depth(9) Avg accuracy: **0.8511278195488721**
Accuracy on test set using best depth: **0.8618618618618619**
Error on test set using best depth: **0.13813813813813813**
Depth: **9**