Decision Trees

Vinay Bhaip*

September 2019

1 Introduction

Decision trees are powerful and interpretable classifiers that mirror human decisions unlike many other classifiers in supervised machine learning and are the building blocks of random forests.

2 Definition

In essence, decision trees ask a series of true/false questions to narrow down what class a particular sample belongs to. Here is an example of a decision tree one might use in real life to decide upon an activity on a given day:

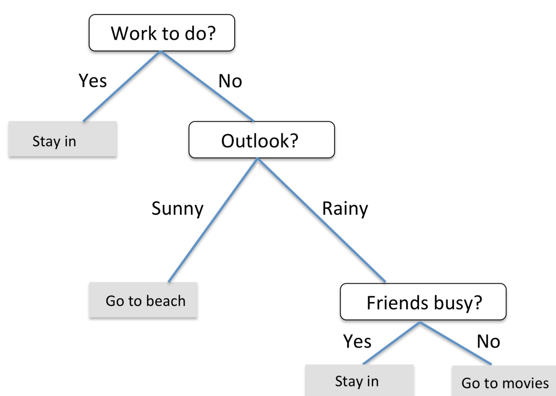


Figure 1: Real Life Decision Tree

Although this figure asks categorical variable-based questions, we can ask numerical-based questions like "$x_1 < 5$?" when the features are continuous. To build our tree, we start at the root node and ask a question that splits the data

*Adapted from Sylesh Suresh's lecture

based on a feature such that the information gain is maximized. We continuously do this for each node until the decision tree can classify all the training data. (Note that in practice this leads to overfitting, so the tree is usually pruned, i.e. a limit on the depth of the tree is set.)

2.1 Information Gain

We split each node on the feature and threshold that yields the most information gain. The formula for information gain in a binary decision tree is as follows:

$$IG(D_p, f) = I(D_p) - \frac{N_{left}}{N_p} I(D_{left}) - \frac{N_{right}}{N_p} I(D_{right})$$

D_p is the dataset of the parent node (the node which we are splitting), f is the feature of the dataset which we are splitting on, N_p is the total number of samples in the parent node, N_{left} and N_{right} are the number of samples in the datasets of the left child node and right child node respectively, and I is the impurity measure. A node is pure if all samples in its dataset belong to the same class and is most impure when an equal number of samples belong to each class. Essentially, information gain calculates the difference between the impurity of the parent node and the impurity of the child nodes.

One commonly used measure of impurity is Gini impurity:

$$I_G(i) = 1 - \sum_{k=1}^{c} p(k|i)^2$$

$p(k|i)$ is the proportion of samples of class k to the total number of samples in the dataset of the i^{th} node. The impurity is maximized when the classes of the node are perfectly mixed (for this example, consider a situation in which there are 2 classes, meaning c = 2):

$$1 - \sum_{k=1}^{c} 0.5^2 = 0.5$$

An alternative impurity measure is entropy, which is defined as:

$$-\sum_{k=1}^{c} p(k|i) log_2 p(k|i)$$

Note that this function has a maximum of 1.0, not 0.5. In practice, Gini impurity and entropy yield similar results, so it is more useful to test different pruning cut-offs rather than to evaluate trees with different impurity criteria.

To decide on a split for a specific node, we will search for the feature and the threshold (e.g. "petal length < 2.45 cm" for a flower classifier) that maximizes the information gain. We can choose the best threshold for a feature from the feature values in the training data or from the averages of every pair of feature values in the training set. Another method is to select the best threshold from the quartiles (20%, 40%, 60%, and 80% values) of the feature set.

Here is the pseudocode for determining the best split:

Algorithm 1 Best Split

```
IG ← 0
for each feature do
    for each threshold do
        pot_left, pot_right ← split(parent, feature, threshold)
        pot_ig ← information_gain(parent, pot_left, pot_right)
        if pot_ig > IG then
            left ← pot_left
            right ← pot_right
            IG ← pot_ig
        end if
    end for
end for
return left, right
```

Algorithm 2 Split

```
function SPLIT(dataset, feature, threshold)
    Initialize left and right lists
    for each data point in dataset do
        if feature of data point < threshold then
            append data point to left
        else
            append data point to right
        end if
    end for
    return left, right
end function
```

Algorithm 3 Gini Impurity

```
function SPLIT(dataset)
    sum ← 0
    for each class label c do
        ratio ← (number of class labels c)/size of dataset
        sum ← sum + ratio * ratio
    end for
    return 1 - sum
end function
```

3 Practice Problems

Consider the following dataset:

x1	x2	x3	y
0	1	0	-1
1	0	0	+1
0	1	1	+1
0	0	1	-1

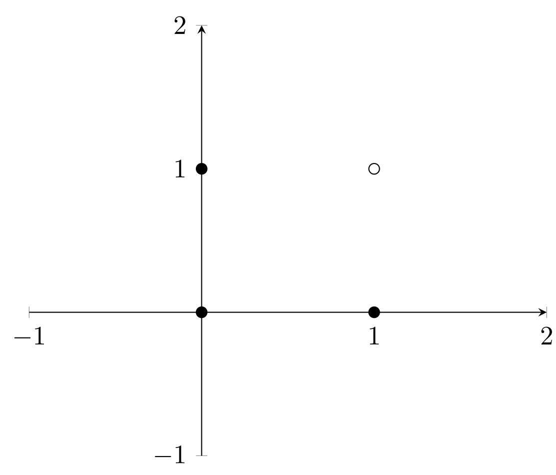
1. What feature will we split on at the root of our decision tree, and what will our information gain be from splitting on that feature using the Gini impurity measure?

2. Build a decision tree using the dataset. What is the depth of the tree?

3. What will the decision tree classify a data point with the features x1 = 0, x2 = 0, and x3 = 0 as (y = -1 or y = +1)?

x1	x2	y
0	0	+1
0	1	+1
1	0	+1
1	1	-1

It's often helpful to visualize your data, even if it seems simple. Let the $\bullet = +1$ and $\circ = -1$.



4. What will the information gain be after the first split in the above data set with the Gini impurity measure? With entropy as the impurity measure?

5. What is the depth of the final decision tree?

4 Competition

- The Decision Trees Problem Set is due next week. This will be taken into the ranking system.
- If you feel ready, try the Kaggle competition at `https://www.kaggle.com/c/tjml1920-decision-trees`. It ends in 2 weeks at 11:59 PM on Tuesday, October 8th. We will be going more in depth next week on how to do the competition, so feel free to wait till then if you'd like.