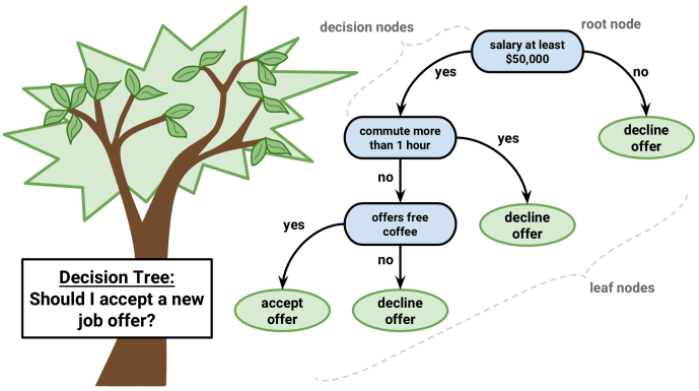
1 What's Decision Tree? 1
2 How to Build a Decision Tree? 1
2.1 Feature Selection 2
2.2.1 Information Entropy 2
2.2.2 Information Gain (IG) 2
2.2 Build Decision Tree 4
2.3 Pruning 4
3 Algorithms for Decision Tree 4
3.1 ID3 4
3.2 C4.5 5
3.3 CART 5
4 Pros and Cons 6

1 What's Decision Tree?

As the name goes, it uses a tree-like model of decisions. It has many analogies with a tree in real life, covering both **classification and regression**. In decision analysis, a decision tree can be used to visually and explicitly represent decisions and decision making.

Let's consider a basic example that predicts whether an offer will be accepted or declined. A decision tree is drawn upside down with its root at the top. In the image below, it includes condition/internal node represented by blue boxes, branches/edges represented by black arrows, and the decision/leaf node represented by green boxes.



- Root Node: including the entire samples
- Internal Node: including different features or attributes
- Leaf Node: including the outputs of decision

2 How to Build a Decision Tree?

Growing a tree involves decision on **which features (conditions) to choose** and **their order to use** for splitting, along with knowing when to stop. As a tree generally grows arbitrarily, you will need to **trim it down** for it to look beautiful.

2.1 Feature Selection

Feature selection determines which features to use for judgment. In the training data set, there may be many attributes of each sample, and different attributes have different effects. Therefore, to screen out features with high correlation with classification results is very important. **The criteria commonly used in feature selection are information gain.**

2.2.1 Information Entropy

"Information Entropy (IE)" is a commonly used index to measure the purity of samples, or you can say it represents the complexity of samples. The entropy H (Greek capital letter) of a discrete random variable X with possible values {x1, x2, x3, …, xn} can be written as:

$$H(X) = -\sum_{i-1}^{n} P(x_i) \log_b P(x_i) \tag{1}$$

B is the base of logarithm, which is often considered as 2.

[Example] There are two sets called A={1,1,1,1,1,1,1,1,2,2}, B={1,2,3,4,5,6,7,8,9,1}, what are the IE for these two sets?

IE for set A:

$$\begin{aligned} H(X) &= -P(x_1 = 1) \log_2 P(x_1 = 1) - P(x_2 = 2) \log_2 P(x_2 = 2) \\ &= -\frac{8}{10} \log_2 \frac{8}{10} - \frac{2}{10} \log_2 \frac{2}{10} \\ &= 0.722 \end{aligned}$$

IE for set B:

$$\begin{aligned} H(X) &= -P(x_1 = 1) \log_2 P(x_1 = 1) - P(x_2 = 2) \log_2 P(x_2 = 2) \\ &= -\frac{8}{10} \log_2 \frac{8}{10} - \frac{2}{10} \log_2 \frac{2}{10} \\ &= 3.122 \end{aligned}$$

2.2.2 Information Gain (IG)

Information Gain = New Information Entropy – Old Information Entropy

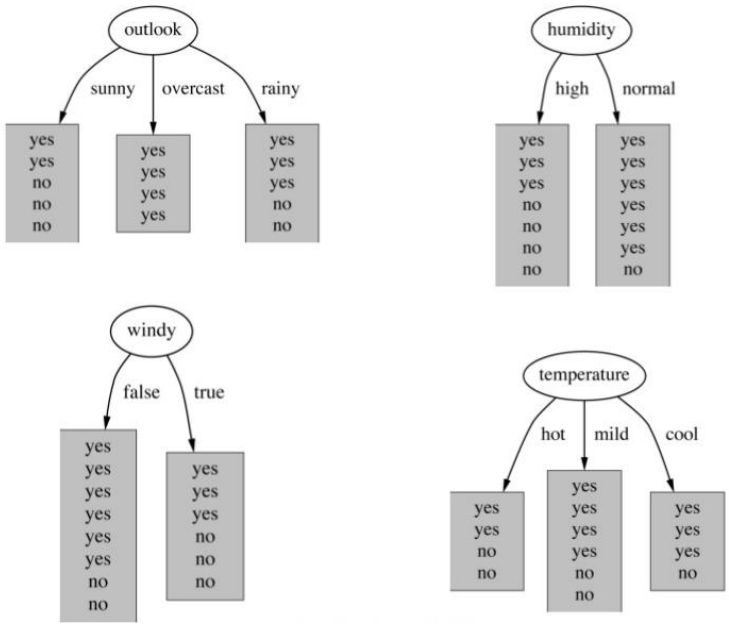
[Example] Based on weather, we predict if going outside to play tennis?

Outlook	Temperature	Humidity	Windy	PlayTennis
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Overcast	Hot	High	False	Yes
Rainy	Mild	High	False	Yes
Rainy	Cool	Normal	False	Yes
Rainy	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Sunny	Mild	High	False	No
Sunny	Cool	Normal	False	Yes
Rainy	Mild	Normal	False	Yes
Sunny	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Rainy	Mild	High	True	No

Old IE: In the historic dataset, there are 9 days playing, 5 days not playing, so the Old IE:

$$H(X) = -\frac{9}{14}\log_2\frac{9}{14} - \frac{5}{14}\log_2\frac{5}{14} = 0.940$$

Now, we have four features to make decisions: Outlook, Temperature, Windy, and Humidity. Let's see what happens to entropy when we make our first decision according to Outlook.



1. **Outlook: Information Gain**
   If we make a decision tree based on outlook, we have three branches possible; either it will be Sunny or Overcast or Raining.

Outlook	Yes	No	Number of instances
Sunny	2	3	5
Overcast	4	0	4
Rain	3	2	5

Sunny: In the given data, 5 days were sunny. Among those 5 days, tennis was played on 2 days and tennis was not played on 3 days. What is the entropy here?

- Probability of playing tennis = 2/5 = 0.4
- Probability of not playing tennis = 3/5 = 0.6
- Entropy when sunny = -0.4 * $\log_2(0.4)$ – 0.6 * $\log_2(0.6)$ = 0.97

Overcast: In the given data, 4 days were overcast and tennis was played on all the four days.

- Probability of playing tennis = 4/4 = 1
- Probability of not playing tennis = 0/4 = 0
- Entropy when overcast = 0.0

Raining: In the given data, 5 days were rainy. Among those 5 days, tennis was played on 3 days and tennis was not played on 2 days. What is the entropy here?

- Probability of not playing tennis = 2/5 = 0.4
- Probability of playing tennis = 3/5 = 0.6
- Entropy when rainy = -0.4 * log2(0.4) – 0.6 * log2(0.6) = 0.97

New IE: Entropy among the three branches:
= (number of sunny days)/(total days) * (entropy when sunny)
+ (number of overcast days)/(total days) * (entropy when overcast)
+ (number of rainy days)/(total days) * (entropy when rainy)
= (5/14) * 0.97) + (4/14) * 0) + (5/14) * 0.97) = 0.693

What is the reduction in randomness due to choosing outlook as a decision maker?
Reduction in randomness = entropy source – entropy of branches = 0.940 – 0.693 = **0.247**
This reduction in randomness is called **Information Gain**. Similar calculation can be done for other features.

2. **Temperature: Information Gain = 0.029**

Temperature	Yes	No	Number of instances
Hot	2	2	4
Cool	3	1	4
Mid	4	2	6

3. **Windy: Information Gain = 0.048**

Windy	Yes	No	Number of instances
Weak	6	2	8
Strong	3	3	6

4. **Humidity: Information Gain = 0.152**

Humidity	Yes	No	Number of instances
High	3	4	7
Normal	6	1	7

2.2 Build Decision Tree

Therefore, we can get IG for different features: **IG for Overlook > IG for Humidity > IG for Windy > IG for Temperature**, which means the order for features is Overlook, Humidity, Windy and Temperature.

2.3 Pruning

The main goal to prune is for **avoiding overfitting by removing some branches**. We can use pruning to limit the depth of tree, the number of leaf nodes, the number of samples of leaf nodes and information gain. There are two methods we commonly use:

- **Pre-pruning:** prune when decision tree is building. Stop growing when data split not statistically significant – e.g. in C4.5: Split only, if there are at least two descendant that have at least n examples.
- **Post-prune:** prune when decision tree is finished. Grow full tree, then post-prune.

3 Algorithms for Decision Tree

3.1 ID3

Iterative Dichotomies 3 (ID3) algorithm is to construct the decision tree by employing a top-down, greedy search through the given sets to test each attribute at every tree node. In order to select the attribute that is most useful for classifying a given sets, and **criteria of ID3 is Information Gain (IG)**.

3.2 C4.5

Based on ID3 algorithm, **the criteria of C4.5 is Information Gain Ratio (IGR)**. It's equal to information gain divided by its own entropy.

$$\text{Information Gain Ratio} = \frac{Information\ Gain}{Its\ own\ entropy} \tag{2}$$

3.3 CART

Classification and Regression Algorithm (CART) is another algorithm for decision tree. **The Gini Index is used in the CART decision tree to select the classification attribute.** The Gini index reflects the probability that two samples are randomly selected from the sample set and their category labels are inconsistent. Therefore, the smaller the Gini, the better. It stores the sum of squared probabilities of each class. We can formulate it as illustrated below.

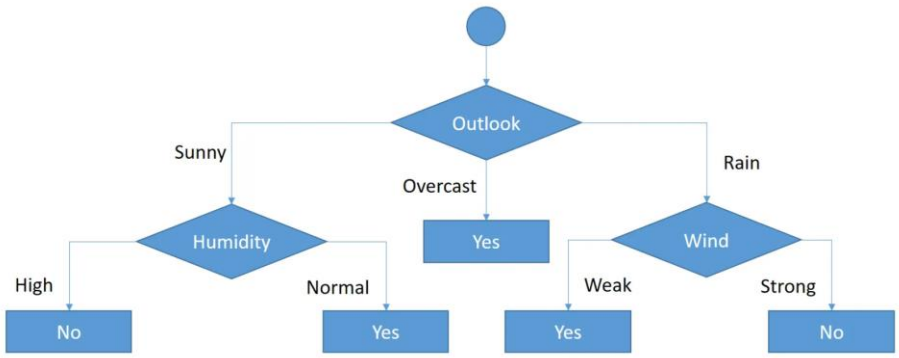
$$\text{Gini Index} = 1 - \sum_{i=1}^{n} {P_i}^2 \tag{3}$$

Now we compute Gini Index for the above example.

1. **Outlook: Gini Index**
   If we make a decision tree based on outlook, we have three branches possible; either it will be Sunny or Overcast or Raining.
   Sunny: In the given data, 5 days were sunny. Among those 5 days, tennis was played on 2 days and tennis was not played on 3 days.
   Overcast: In the given data, 4 days were overcast and tennis was played on all the four days.
   Raining: In the given data, 5 days were rainy. Among those 5 days, tennis was played on 3 days and tennis was not played on 2 days.
   - Gini (Outlook = Sunny) = $1 – (2/5)^2 – (3/5)^2 = 1 – 0.16 – 0.36 = 0.48$
   - Gini (Outlook = Overcast) = $1 – (4/4)^2 – (0/4)^2 = 0$
   - Gini (Outlook = Rain) = $1 – (3/5)^2 – (2/5)^2 = 1 – 0.36 – 0.16 = 0.48$
   - Gini (Outlook) = (5/14) x 0.48 + (4/14) x 0 + (5/14) x 0.48 = 0.171 + 0 + 0.171 = **0.342**

2. **Temperature: Gini Index = 0.439**
3. **Windy: Gini Index = 0.367**
4. **Humidity: Gini Index = 0.428**

Overall, the winner will be outlook feature because its cost is the lowest, next is windy, then are humidity and temperature. And we get the decision tree as below:



4 Pros and Cons

Pros:

- Easy to understand and explain, can be analyzed visually;
- It can process both nominal and numerical data;
- More suitable for handling samples with missing attributes;
- Able to handle irrelevant features;
- When testing data set, the running speed is fast;
- Able to produce feasible and effective results for large data sources in relatively short time.

Cons:

- Overfitting is easy to occur (random forest can greatly reduce overfitting);
- Easy to ignore the correlation of attributes in data set;
- For those data with different types of samples, different criteria will bring different preference for attribute selection when classifying attributes in the decision tree;
    - Information gain criteria (ID3 algorithm) is suitable to lager numbers of attributes
    - Information gain rate criterion (CART algorithm) is suitable to less numbers of attributes