Decision Tree

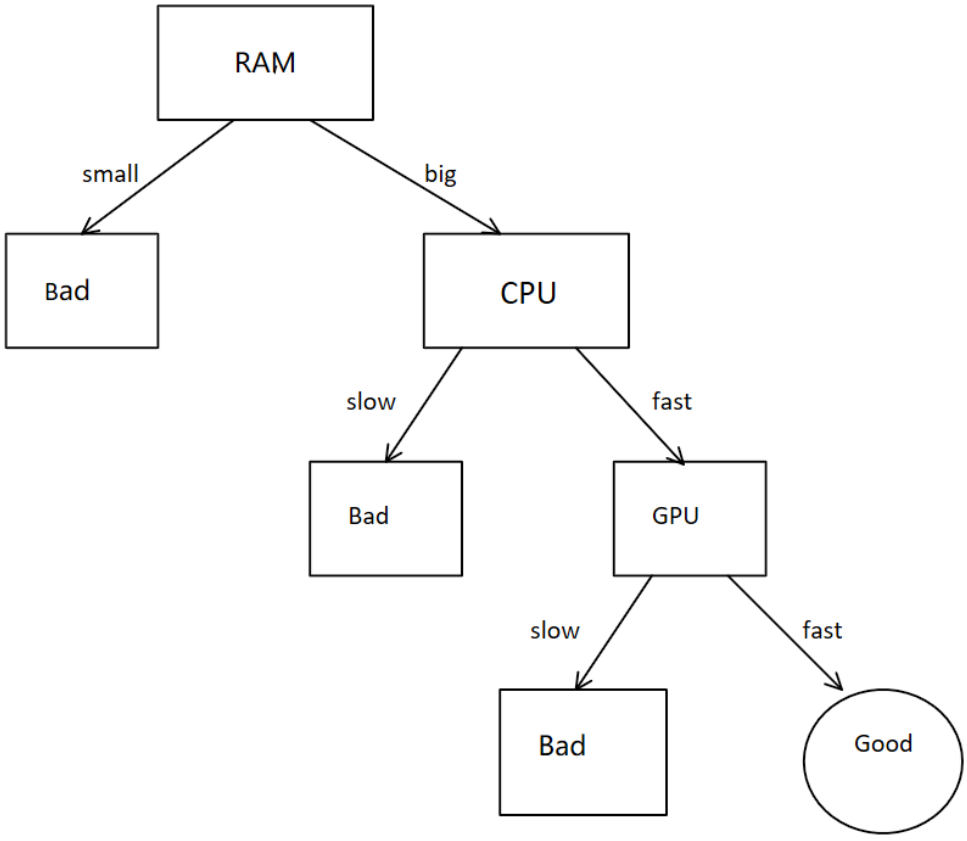
Author: Zhang Xiaozheng Date: 2019/10/20

What is Decision Tree

First, let's see the definition of the decision tree on Wiki.

> A **decision tree** is a decision support tool that uses a tree-like model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility. It is one way to display an algorithm that only contains conditional control statements.
>
> Decision trees are commonly used in operations research, specifically in decision analysis, to help identify a strategy most likely to reach a goal, but are also a popular tool in machine learning.

The decision tree is a way to do the classification. It works just like a human being. For example, when you want to know whether a computer is a good computer, you may first look at the RAM of this computer and then the CPU and then the GPU and so on. If we represent this process with a tree, we may do this:



First, we check the RAM of this computer, if it doesn't have large RAM, we think this computer isn't a good computer. If it has large RAM, we continue to check the CPU and so on.

So, if we can generate a tree like this from a data set, we can let the machine do the classification job based on this tree just like us. This is the basic idea of the decision tree.

How to create a decision tree

When we want to create a decision tree, the first thing we need to think is how to choose the feature on each node. To solve this question, we need to know information entropy and information gain.

- Information entropy

  For a given set ***D*** which has ***|y|*** classes, the proportion of the ***k-th*** class in set ***D*** is

  $$p_k \quad (k = 1, 2, 3.....|y|)$$

  The entropy of the set ***D*** is:

  $$Ent(D) = -\sum_{k=1}^{|y|} p_k log_2(p_k)$$

- Information gain

  Suppose a feature ***a*** has ***V*** different types which is:

  $$\{a^1, a^2, ..., a^v\}$$

  If we split the set ***D*** by feature ***a***, we will have ***v*** subsets. Each subsets has

  $$|D^v|$$

  elements and the set ***D*** has ***|D|*** elements, so the information gain is:

  $$Gain(D, a) = Ent(D) - \sum_{v=1}^{V} \frac{|D^v|}{|D|} Ent(D^v)$$

So we can choose the feature which makes the information gain largest to split the set ***D*** . This is the ***ID3*** algorithm.

But there is some problem with the ***ID3*** algorithm. For example, if we use the ID number of people in a training data as a feature, because everyone has a different ID number, the information gain of this feature will be very large, but the ID number may has nothing to do with the answer. To solve this problem, we can use the gain ratio to split the set ***D*** .

- Gain ratio

  $$GainRatio(D, a) = \frac{Gain(D, a)}{IV(a)}$$

  where

  $$IV(a) = -\sum_{v=1}^{V} \frac{|D^v|}{|D|} log_2 \frac{|D^v|}{|D|}$$

  It's the intrinsic value of a. The more classes feature ***a*** has, the bigger its intrinsic value is. So this can let the feature which has many classes have a little information gain.

But the gain ratio is more likely to choose those features who have little classes. So, in practice, we usually first choose those features whose information gain is larger than the average, then we choose the largest gain ratio in these features. This is the ***C4.5*** algorithm.

Question

The Decision Tree algorithm can be easily affect by the training data, why?