Data Mining Project

C4.5 Algorithm

Saber Salah
Naji Sami Abduljalil Abdulhak

Decembre 9, 2010
1.0 Introduction

Before start talking about C4.5 algorithm let’s see first what is machine learning? Human being learns from experience, so the target here is how to make the machine learn like a human? Or how the machine gets knowledge and gives result. Such a human learn from previous experience the machine learn from experience too ‘After giving a set of examples’, so the task of the programmer or the ‘teacher’ is to put a set of examples or instances into the input of the machine and analyzes the output ‘To check the learning accuracy’. In the literature there are a lot of learning methods which ‘Decision Tree’ is the most popular method, in this report we going to talk about the implementation of C4.5 algorithm which used for building a decision tree and we will discuss about its splitting criteria for the attributes, and we will try to discuss too about a new splitting criteria based on the probability.

2.0 Tools

Os Linux ‘Ubunto’

C language

Microsof word 2007

3.0 Description of C4.5 Algorithm

C4.5 algorithm is a process of a special instructions used to built or generate a decision tree, which helps for resolving a several problems such as classification or prediction problems. C4.5 algorithm is an extension of Id3 algorithm (Mr. Ross Quilan 1986), the main idea or the important thing is the splitting criteria used by C4.5, in order to compare the attributes which one is more informative or predictive C4.5 based on the entropy.

3.1 Entropy

The entropy is a mathematical method used to calculate the accuracy of the information for example imagine that we have two sources A and B and we would like to send an information I from A to B, the entropy seems like the fail rate of the information I to reach the destination B, so by changing the position of A and B for example we can get a best result, so in this case the entropy is less it means the fail rate is less. The same C4.5 algorithm uses the entropy to determine which attribute is more predictive for the classification.

\[
\text{Entrophy}(S) = \sum_{i=1}^{c} - p_i \log_2 p_i
\]

Where \( p_i \) is the proportion of the probability of \( S \) belonging to class \( i \).

3.2 Information Gain

It measures the expected reduction in entropy by partitioning the examples according to this attribute. The information gain, Gain(S, A) of an attribute A, relative to the collection of examples S, is defined as

\[
\text{Gain}(S, A) = \text{Entrophy}(S) - \sum_{v \in \text{Values}(A)} \left| \frac{S_v}{S} \right| \text{Entrophy}(S_v)
\]

where \( \text{Values}(A) \) is the set of all possible values for attribute A, and \( S_v \) is the subset of S for which the attribute A has value v.
3.2 C4.5 Algorithm

In general C4.5 uses 4 steps in order to generate a decision tree:

- Choose attribute for root node
- Create branch for each value of that attribute
- Split cases according to branches
- Repeat process for each for each branch until all cases in the branch have the same class

4.0 Implementation

Let’s take the example of ‘Vote Data’ we have 16 attributes and 300 items or examples.

C4.5 algorithm generate a pruning tree so here a comparison between the result of the pruning and not pruning

Case of the data ‘Vote’:

- Before pruning
  The size is 25, errors rate =2.7%
- After pruning
  The size is 7, errors rate=4.3%

Case of the data ‘Golf’:

- Before pruning
  The size is 21, errors rate =0.2%
- After pruning
  The size is 21, errors rate=0.2%

Here we got the same result before and after pruning!!

5.0 Discussion

As we have seen C4.5 uses the entropy as splitting criteria in order to choose the best predictive attribute, it means the attribute that has the minimum entropy value in other term the attribute which has the maximum Gain. Our idea here is: instead of using the entropy we can use the probability, in other sense which attribute has the highest probability to classify better. Thus we calculate the probability in each branch of the tree then we compare the probability of all the attributes in each branch. Let’s take an example to clarify the idea.

We suppose that our Data is ‘Experiment’, let’s X, Y and Z three Attributes and C{True, False} tow classes, suppose we have two values ‘Yes’ and ‘No’, and E1,E2,E3,E4,E5,E6,E7,E8,E9,E10 are ten examples or instances.
<table>
<thead>
<tr>
<th>Instances</th>
<th>X</th>
<th>Y</th>
<th>Z</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>E1</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>True</td>
</tr>
<tr>
<td>E2</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>True</td>
</tr>
<tr>
<td>E3</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>False</td>
</tr>
<tr>
<td>E4</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>True</td>
</tr>
<tr>
<td>E5</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>True</td>
</tr>
<tr>
<td>E6</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>False</td>
</tr>
<tr>
<td>E7</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>True</td>
</tr>
<tr>
<td>E8</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>False</td>
</tr>
<tr>
<td>E9</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>False</td>
</tr>
<tr>
<td>E10</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>True</td>
</tr>
</tbody>
</table>

Figure1 Data ‘Experiment’

We have 3 attributes, let’s start analyzing the first attribute X:
Case X= Yes the Probability for an example to belong to the class True is 4/7
The Probability for an example to belong to the class No is 3/7
Case X= No the Probability for an example to belong to the class True is 2/3
The Probability for an example to belong to the class No is 1/3.

Here the same method we calculate the probability in each branch of the attribute Z we find the following result:

Case Y= Yes the Probability for an example to belong to the class True is 3/3
The Probability for an example to belong to the class No is 0
Case Y= No the Probability for an example to belong to the class True is 3/7
The Probability for an example to belong to the class No is 4/7.

Case Z= Yes the Probability for an example to belong to the class True is 4/7
The Probability for an example to belong to the class No is 3/7
Case Z= No the Probability for an example to belong to the class True is 2/3
The Probability for an example to belong to the class No is 1/3.
So the conclusion is: by making a comparison between the proportion of the probability in each branch of the attributes we can get an idea which attribute is more predictive, it means the attribute that has the highest probability is the best attribute for the classification, so instead of calculate the entropy and determine the attribute which has the less entropy, we calculate directly the probability. The idea seems like the entropy method but we haven’t tried it within an experiment, however we think the complexity of the algorithm will be less than using the entropy because the number of the instruction will be less thus we will get a win in time.

6.0 Conclusion

C4.5 algorithm is an extension of Id3 algorithm, the main difference is the C4.5 can treat the categorical attribute and avoid the over fitting problem. Concerning the splitting criteria used by C4.5 it has some drawbacks such the long time reserved for the computation of the entropy. So the implementation of C4.5 was an opportunity for us to think about this problem and we have tried to think about the using of probability method.