Decision Tree Classifier

(C4.5 Algorithm)

Dr. JASMEET SINGH
ASSISTANT PROFESSOR, CSED
TIET, PATIALA

C4.5 Algorithm

- C4.5 algorithm is also proposed by Quinlan's and is an extension of earlier ID3 algorithm.
- It removes the restrictions and limitations of the ID3 variant of decision tree algorithm.
- It can work with both Discrete and Continuous Data
- C4.5 can handle the issue of incomplete data very well
- The algorithm is not biased towards the features with high number of distinct values.

Evaluating Split –C4.5

- ID3 algorithm's Information Gain metric is biased towards a feature with large number of distinct values.
- This limitation of ID3 algorithm is handled by normalizing the *Information Gain* metric using a parameter called *SplitInfo.* The normalized Information Gain is called *Gain Ratio.*
- *Gain Ratio of an attribute A for a given dataset is computed as:*

$$Gain\ Ratio(S, A) = \frac{Information\ Gain(S, A)}{SplitInfo(S, A)}$$

$$Information\ Gain(S, A) = Entropy(S) - \sum_{v=1}^{|v|} \frac{|S_v|}{|S|} Entropy(S_v)$$

$$SplitInfo(S, A) = -\sum_{v=1}^{|v|} \frac{|S_v|}{|S|} log_2\left(\frac{|S_v|}{|S|}\right)$$

where $\mathbf{S_v}$ is the set of rows in **S** for which the feature column **A** has value **v**, $|\mathbf{S_v}|$ is the number of rows in $\mathbf{S_v}$ and likewise $|\mathbf{S}|$ is the number of rows in **S.**

Handling Continuous Values-C4.5 Algorithm

- C4.5 algorithm partitions the continuous attribute value into a discrete set of intervals.
- C4.5 proposes to perform binary split based on a threshold value for features with continuous values.
- Threshold should be a value which offers maximum gain ratio (Information Gain) for that attribute.

C4.5 Algorithm-Pseudocode

1. Check for the base cases (as discussed in ID3 algorithm).
2. For each attribute a, find the normalised information gain ratio from splitting on a.
3. Let a_best be the attribute with the highest normalized information gain.
4. Create a decision node that splits on a_best.
5. Recur on the sublists obtained by splitting on a_best, and add those nodes as children of node.

Numerical Example 1-C4.5 Algorithm

Consider the dataset that informs about decision making factors to play tennis at outside for previous 14 days.

The dataset is similar to the ID3 dataset.

The difference is that temperature and humidity columns have continuous values instead of nominal ones.

Train a C4.5 Decision Tree Classifier.

Outlook	Temperature	Humidity	Windy	Play
sunny	85	85	false	no
sunny	80	90	true	no
overcast	83	78	false	yes
rain	70	96	false	yes
rain	68	80	false	yes
rain	65	70	true	no
overcast	64	65	true	yes
sunny	72	95	false	no
sunny	69	70	false	yes
rain	75	80	false	yes
sunny	75	70	true	yes
overcast	72	90	true	yes
overcast	81	75	false	yes
rain	71	80	true	no

Solution- Example 1

Compute Entropy of the entire dataset:

$$Entropy\ (S) = -\sum_{i=1}^{n} p_i log_2(p_i)$$

Positive Examples = 9

Negative Examples =5

Total= 14

$$Entropy(S) = -\frac{9}{9+5} log_2\left(\frac{9}{9+5}\right) - \frac{5}{9+5} log_2\left(\frac{5}{9+5}\right) = 0.940$$

Solution- Example 1 (Contd…)

Outlook Attribute

Outlook is a nominal attribute. Its possible values are Rainy, Overcast and Sunny.

$$Entropy(Outlook = Sunny) = -\frac{2}{2+3}log_2\left(\frac{2}{2+3}\right) - \frac{3}{2+3}log_2\left(\frac{3}{2+3}\right) = 0.971$$

$$Entropy(Outlook = Rainy) = -\frac{3}{2+3}log_2\left(\frac{3}{2+3}\right) - \frac{2}{2+3}log_2\left(\frac{2}{2+3}\right) = 0.971$$

$$Entropy(Outlook = Overcast) = -\frac{4}{4+0}log_2\left(\frac{4}{4+0}\right) - \frac{0}{4+0}log_2\left(\frac{0}{4+0}\right) = 0$$

$$Average\ Information\ Entropy = I(S, Outlook) = \sum_{v=1}^{|v|}\frac{|S_v|}{|S|}Entropy(S_v)$$

$$I(S, Outlook) = \frac{2+3}{9+5} \times 0.971 + \frac{3+2}{9+5} \times 0.971 + \frac{4+0}{9+5} \times 0 = 0.693$$

$$Information\ Gain(S, Outlook) = Entropy(S) - \mathrm{I(S, Outlook)} = 0.940 - 0.693 = 0.247$$

$$SplitInfo(S, Outlook) = -\frac{2+3}{9+5}log_2\left(\frac{2+3}{9+5}\right) - \frac{3+2}{9+5}log_2\left(\frac{3+2}{9+5}\right) + \frac{4+0}{9+5}log_2\left(\frac{4+0}{9+5}\right) = 1.577$$

$$\boldsymbol{Gain\ Ratio\ (S, Outlook) = \frac{Information\ Gain(S, Outlook)}{SplitInfo(S,, Outlook)} = \frac{0.247}{1.577} = 0.155}$$

Outlook	PlayTennis
Sunny	No
Sunny	No
Sunny	No
Sunny	Yes
Sunny	Yes

Outlook	PlayTennis
Rainy	Yes
Rainy	Yes
Rainy	No
Rainy	Yes
Rainy	No

Outlook	PlayTennis
Overcast	Yes
Overcast	Yes
Overcast	Yes
Overcast	Yes

Outlook	p	n	Entropy
Sunny	2	3	0.971
Rainy	3	2	0.971
Overcast	4	0	0

Solution- Example 1 (Contd…)

In ID3 algorithm, we've calculated gains for each attribute. Here, we need to calculate gain ratios instead of gains.

Wind Attribute

Wind is a nominal attribute. Its possible values are weak and strong.

$$Entropy(Windy = Strong) = -\frac{3}{3+3} log_2\left(\frac{3}{3+3}\right) - \frac{3}{3+3} log_2\left(\frac{3}{3+3}\right) = 1$$

$$Entropy(Windy = Weak) = -\frac{6}{6+2} log_2\left(\frac{6}{6+2}\right) - \frac{2}{6+1} log_2\left(\frac{2}{6+2}\right) = 0.811$$

$$Average\ Information\ Entropy = I(S, Windy) = \sum_{v=1}^{|v|} \frac{|S_v|}{|S|} Entropy(S_v)$$

$$I(S, Windy) = \frac{3+3}{9+5} \times 1 + \frac{6+2}{9+5} \times 0.811 = 0.892$$

$$Information\ Gain(S, Windy) = Entropy(S) - I(S, Windy) = 0.940 - 0.892 = 0.048$$

$$SplitInfo(S, Windy) = -\frac{3+3}{9+5} log_2\left(\frac{3+3}{9+5}\right) - \frac{6+2}{9+5} log_2\left(\frac{6+2}{9+5}\right) = 0.985$$

$$\boldsymbol{Gain\ Ratio\ (S, Windy) = \frac{Information\ Gain(S, Windy)}{SplitInfo(S, Windy)} = \frac{0.048}{0.985} = 0.049}$$

Windy	PlayTennis
Weak	No
Weak	Yes
Weak	Yes
Weak	Yes
Weak	No
Weak	Yes
Weak	Yes
Weak	Yes

Windy	PlayTennis
Strong	No
Strong	No
Strong	Yes
Strong	Yes
Strong	Yes
Strong	No

Windy	p	n	Entropy
Strong	3	3	1
Weak	6	2	0.811

Solution Example – 1 (Contd....)

- As an exception, humidity is a continuous attribute.
- We need to convert continuous values to nominal ones.
- Firstly, we need to sort humidity values smallest to largest (as shown in table).
- Now, we need to iterate on all humidity values and separate dataset into two parts as instances less than or equal to current value, and instances greater than the current value.
- We would calculate the gain or gain ratio for every threshold. The value which maximizes the gain would be the threshold.

Humidity	Play
65	Yes
70	No
70	Yes
70	Yes
75	Yes
78	Yes
80	Yes
80	Yes
80	No
85	No
90	No
90	Yes
95	No
96	Yes

Solution Example – 1 (Contd....)

Check 65 as a threshold for humidity

Humidity	Yes	No
<=65	1	0
>65	8	5

$$Entropy(Humidity \leq 65) = -\frac{1}{1+0} log_2\left(\frac{1}{1+0}\right) - \frac{0}{1+0} log_2\left(\frac{0}{1+0}\right) = 0$$

$$Entropy(Hunidity > 65) = -\frac{8}{8+5} log_2\left(\frac{8}{8+5}\right) - \frac{5}{8+5} log_2\left(\frac{5}{8+5}\right) = 0.961$$

$$Average\ Information\ Entropy = I(S, Humidity<>65) = \frac{1+0}{9+5} \times 0 + \frac{8+5}{9+5} \times 0.961 = 0.892$$

$$Information\ Gain(S, Humidity<>65) = Entropy(S) - \mathrm{I}(S, Humidity<>65) = 0.940 - 0.892 = 0.048$$

$$SplitInfo(S, Humidity<>65) = -\frac{1+0}{9+5} log_2\left(\frac{1+0}{9+5}\right) - \frac{8+5}{9+5} log_2\left(\frac{8+5}{9+5}\right) = 0.371$$

$$\boldsymbol{Gain\ Ratio\ (S, Humidity<>65) = \frac{Information\ Gain(S, Humidity<>65)}{SplitInfo(S, Humidity<>65)} = \frac{0.048}{0.371} = 0.126}$$

The statement $Humidity<>65$ *refers to that what would branch of decision tree be for less than or equal to 65, and greater than 65. It* ***does not*** *refer to that humidity is not equal to 65!*

Solution Example – 1 (Contd….)

Similarly, the process will be repeated for each unique value in Humidity and we will get following Gain Ratios:

For 70, Gain_Ratio(S, Humidity<> 70) = 0.016

For 75, Gain_Ratio(S, Humidity<> 75) = 0.047

For 78, Gain_Ratio(S, Humidity <> 78) =0.090

For 80, Gain_Ratio(S, Humidity <> 80) = 0.107

For 85, Gain_Ratio(S, Humidity <> 85) = 0.027

For 90, Gain_Ratio(S, Humidity <> 90) = 0.016

For 95, GainRatio(S, Humidity <> 95) = 0.118

Value 96 is ignored, because humidity cannot be greater than this value.

As seen, gain maximizes when threshold is equal to 65 for humidity. This means that we need to compare other nominal attributes and comparison of humidity to 65 to create a branch in our tree.

Solution Example – 1 (Contd....)

- Temperature is also a continuous attribute.
- We need to convert continuous values to nominal ones.
- Firstly, we need to sort temperature values smallest to largest (as shown in table).
- Now, we need to iterate on all temperature values and separate dataset into two parts as instances less than or equal to current value, and instances greater than the current value.
- We would calculate the gain or gain ratio for every step. The value which maximizes the gain would be the threshold.

Temperature	Play
64	yes
65	no
68	yes
69	yes
70	yes
71	no
72	no
72	yes
75	yes
75	yes
80	no
81	yes
83	yes
85	no

Solution Example – 1 (Contd....)

Check 68 as a threshold for Temperature

Temperature	Yes	No
<=68	2	1
>68	7	4

$$Entropy(Temp \leq 68) = -\frac{2}{2+1} log_2\left(\frac{2}{2+1}\right) - \frac{1}{2+1} log_2\left(\frac{1}{2+1}\right) = 0.915$$

$$Entropy(Temp > 68) = -\frac{7}{7+4} log_2\left(\frac{7}{7+4}\right) - \frac{4}{7+4} log_2\left(\frac{4}{7+4}\right) = 0.914$$

$$Average\ Information\ Entropy = I(S, \text{Temp<>68}) = \frac{2+1}{9+5} \times 0.915 + \frac{7+4}{9+5} \times 0.914 = 0.914$$

$$Information\ Gain(S, \text{Temp<>68}) = Entropy(\text{S}) - \text{I(S,Temp<>68)} = 0.940 - 0.914 = 0.026$$

$$SplitInfo(S, \text{Temp<>68}) = -\frac{2+1}{9+5} log_2\left(\frac{2+1}{9+5}\right) - \frac{7+4}{9+5} log_2\left(\frac{7+4}{9+5}\right) = 0.750$$

$$\boldsymbol{Gain\ Ratio\ (S, \text{Temp<>68}) = \frac{Information\ Gain(S, \text{Temp<>68})}{SplitInfo(S, \text{Temp<>68})} = \frac{0.026}{0.750} = 0.035}$$

Solution Example – 1 (Contd....)

Similarly, the process will be repeated for each unique value in Temperature and we will get maximum Gain Ratio for 83.

Gain_Ratio(S, Temperature<> 83) = 0.305

The dataset of continuous values is transformeed into nominal values as Humidity is now converted to Humiditiy>65 And Temperature is now converted to Temperature>83 with values Yes or no (as shown in Table)

Outlook	Temperature>83	Humidity>65	windy	play
overcast	No	Yes	Weak	yes
overcast	No	No	Strong	yes
overcast	No	yes	Strong	yes
overcast	No	yes	Weak	yes
rainy	No	yes	Weak	yes
rainy	No	yes	Weak	yes
rainy	No	yes	Strong	no
rainy	No	yes	Weak	yes
rainy	No	yes	Strong	no
sunny	Yes	Yes	Weak	no
sunny	No	Yes	Strong	no
sunny	No	Yes	Weak	no
sunny	No	yes	Weak	yes
sunny	No	yes	Strong	yes

Solution Example – 1 (Contd….)

Summary of Gain Ratio for Each Attribute

Attribute	Gain Ratio
Outlook	0.155
Temperature<>83	0.305
Humidity<>65	0.107
Windy	0.049

Temperature attribute comes with both maximized gain and gain ratio. This means that we need to put Temperature decision in root of decision tree.

Solution Example – 1 (Contd….)

So for each value of Temperature>83 i.e. Yes and No, a branch will be added and the C4.5 algorithm will be applied for instances corresponding to (Temperature>83)=Yes, and (Temperature>83=No)

Temperature>83	Outlook	Humidity>65	windy	play
No	overcast	Yes	Weak	yes
No	overcast	No	Strong	yes
No	overcast	yes	Strong	yes
No	overcast	yes	Weak	yes
No	rainy	yes	Weak	yes
No	rainy	yes	Weak	yes
No	rainy	yes	Strong	no
No	rainy	yes	Weak	yes
No	rainy	yes	Strong	no
No	sunny	Yes	Strong	no
No	sunny	Yes	Weak	no
No	sunny	Yes	Weak	yes
No	sunny	yes	Strong	yes

Temperature>83	Outlook	Humidity>65	windy	play
Yes	sunny	Yes	Weak	no

Solution Example – 1 (Contd….)

The Final Decision Tree is as follows:

```
if (Temperature>83)=No:
   if Outlook == 'Rain':
      if Wind == 'Weak':
         return 'Yes'
      elif Wind == 'Strong':
         return 'No'
   elif Outlook == 'Overcast':
      return 'Yes'
   elif Outlook == 'Sunny':
      if (Humidity>65)=Yes:
         if Wind == 'Strong':
            return ‘Yes'
         elif Wind == 'Weak':
            return ‘Yes’
elif (Temperature>83)=Yes:
   return 'No'
```