C4.5 Algorithm

Decision Tree Learning

Introduction

- ID3 is the most common conventional decision tree algorithm but it has bottlenecks.
- Attributes must be nominal values.
- Dataset must not include missing data.
- The algorithm tend to fall into over-fitting.
- Ross Quinlan, inventor of ID3, made some improvements for these bottlenecks and created a new algorithm named C4.5.
- The algorithm can create a more generalized models including continuous data and could handle missing data.
- some resources such as Weka named this algorithm as J48.
- Actually, it refers to re-implementation of C4.5 release 8.

Objective

- Decision rules will be found based on entropy and information gain ratio pair of each feature.
- In each level of decision tree, the feature having the maximum gain ratio will be the decision rule.

Data set

- We are going to create a decision table for the given dataset. It informs about decision making factors to play tennis at outside for previous 14 days. The dataset might be familiar from the ID3 and CART example. The difference is that temperature and humidity columns have continuous values instead of nominal ones.

Day	Outlook	Temp.	Humidity	Wind	Decision
1	Sunny	85	85	Weak	No
2	Sunny	80	90	Strong	No
3	Overcast	83	78	Weak	Yes
4	Rain	70	96	Weak	Yes
5	Rain	68	80	Weak	Yes
6	Rain	65	70	Strong	No
7	Overcast	64	65	Strong	Yes
8	Sunny	72	95	Weak	No
9	Sunny	69	70	Weak	Yes
10	Rain	75	80	Weak	Yes
11	Sunny	75	70	Strong	Yes
12	Overcast	72	90	Strong	Yes

Solution

- Firstly, we need to calculate global entropy.
- There are 14 examples; 9 instances refer to yes decision, and 5 instances refer to no decision.
- Entropy(Decision) $= \sum - p(I) \cdot \log_2 p(I)$
  $= - p(Yes) \cdot \log_2 p(Yes) - p(No) \cdot \log_2 p(No)$
  $= - (9/14) \cdot \log_2 (9/14) - (5/14) \cdot \log_2 (5/14)$
  $= 0.940$

Gain Ratio

- In ID3 algorithm, we've calculated gains for each attribute. Here, we need to calculate gain ratios instead of gains.
- GainRatio(A) = Gain(A) / SplitInfo(A)
- SplitInfo(A) = $-\sum |Dj|/|D| \times \log_2 |Dj|/|D|$

Wind Attribute

- Wind is a nominal attribute. Its possible values are weak and strong.
- Gain(Decision, Wind) = Entropy(Decision) – ∑ (p(Decision|Wind) . Entropy(Decision|Wind))
- Gain(Decision, Wind) = Entropy(Decision)
  - p(Decision|Wind=Weak) . Entropy(Decision|Wind=Weak)
  - p(Decision|Wind=Strong) . Entropy(Decision|Wind=Strong)
- There are 8 weak wind instances. 2 of them are concluded as no, 6 of them are concluded as yes.
- Entropy(Decision|Wind=Weak) = – p(No) . $\log_2$p(No) – p(Yes) . $\log_2$p(Yes)
  = – (2/8) . $\log_2$(2/8) – (6/8) . $\log_2$(6/8) = 0.811
- Entropy(Decision|Wind=Strong) = – (3/6) . $\log_2$(3/6) – (3/6) . $\log_2$(3/6) = 1
- Gain(Decision, Wind) = 0.940 – (8/14).(0.811) – (6/14).(1)
  = 0.940 – 0.463 – 0.428 = 0.049
- There are 8 decisions for weak wind, and 6 decisions for strong wind.
- SplitInfo(Decision, Wind) = -(8/14).$\log_2$(8/14) – (6/14).$\log_2$(6/14)
  = 0.461 + 0.524 = 0.985
- GainRatio(Decision, Wind) = Gain(Decision, Wind) / SplitInfo(Decision, Wind)
  = 0.049 / 0.985 = 0.049

Outlook Attribute

- Outlook is a nominal attribute, too. Its possible values are sunny, overcast and rain.
- Gain(Decision, Outlook) = Entropy(Decision) –
  ∑ (p(Decision|Outlook) . Entropy(Decision|Outlook))
- Gain(Decision, Outlook) = Entropy(Decision) –
  p(Decision|Outlook=Sunny) . Entropy(Decision|Outlook=Sunny) –
  p(Decision|Outlook=Overcast) . Entropy(Decision|Outlook=Overcast) –
  p(Decision|Outlook=Rain) . Entropy(Decision|Outlook=Rain)
- There are 5 sunny instances. 3 of them are concluded as no, 2 of them are concluded as yes.
- Entropy(Decision|Outlook=Sunny) = – p(No) . $\log_2$p(No) – p(Yes) . $\log_2$p(Yes) =
  -(3/5).$\log_2$(3/5) – (2/5).$\log_2$(2/5) = 0.441 + 0.528 = 0.970
- Entropy(Decision|Outlook=Overcast) = – p(No) . $\log_2$p(No) – p(Yes) . $\log_2$p(Yes) =
  -(0/4).$\log_2$(0/4) – (4/4).$\log_2$(4/4) = 0
- Entropy(Decision|Outlook=Rain) = – p(No) . $\log_2$p(No) – p(Yes) . $\log_2$p(Yes) =
  -(2/5).$\log_2$(2/5) – (3/5).$\log_2$(3/5) = 0.528 + 0.441 = 0.970
- Gain(Decision, Outlook) = 0.940 – (5/14).(0.970) – (4/14).(0) – (5/14).(0.970)
  = 0.246
- There are 5 instances for sunny, 4 instances for overcast and 5 instances for rain.
- SplitInfo(Decision, Outlook) = -(5/14).$\log_2$(5/14) -(4/14).$\log_2$(4/14) -(5/14).$\log_2$(5/14)
  = 1.577
- GainRatio(Decision, Outlook) = Gain(Decision, Outlook)/SplitInfo(Decision, Outlook)
  = 0.246/1.577 = 0.155

Humidity Attribute

- As an exception, humidity is a continuous attribute.
- We need to convert continuous values to nominal ones.
- C4.5 proposes to perform binary split based on a threshold value.
- Threshold should be a value which offers maximum gain for that attribute.
- Let’s focus on humidity attribute. Firstly, we need to sort humidity values smallest to largest.

Day	Humidity	Decision
7	65	Yes
6	70	No
9	70	Yes
11	70	Yes
13	75	Yes
3	78	Yes
5	80	Yes

Humidity Attribute

- 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 step.
- The value which maximizes the gain would be the threshold.

Humidity Attribute

- Check 65 as a threshold for humidity
- Entropy(Decision|Humidity<=65) = – p(No) . $\log_2$p(No) – p(Yes) . $\log_2$p(Yes)
  = -(0/1).$\log_2$(0/1) – (1/1).$\log_2$(1/1) = 0
- Entropy(Decision|Humidity>65) = -(5/13).$\log_2$(5/13) – (8/13).$\log_2$(8/13)
  =0.530 + 0.431 = 0.961
- Gain(Decision, Humidity<> 65) = 0.940 – (1/14).0 – (13/14).(0.961)
  = 0.048
- *The statement above 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!*
- SplitInfo(Decision, Humidity<> 65) = -(1/14).$\log_2$(1/14) -(13/14).$\log_2$(13/14)
  = 0.371
- GainRatio(Decision, Humidity<> 65) = 0.126

Humidity Attribute

- Check 70 as a threshold for humidity
- Entropy(Decision|Humidity<=70) = $-(1/4).\log_2(1/4) - (3/4).\log_2(3/4)$
  $= 0.811$
- Entropy(Decision|Humidity>70) = $-(4/10).\log_2(4/10) - (6/10).\log_2(6/10)$
  $= 0.970$
- Gain(Decision, Humidity<> 70) = $0.940 - (4/14).(0.811) - (10/14).(0.970)$
  $= 0.940 - 0.231 - 0.692 = 0.014$
- SplitInfo(Decision, Humidity<> 70) = $-(4/14).\log_2(4/14) -10/14).\log_2(10/14)$
  $= 0.863$
- GainRatio(Decision, Humidity<> 70) = 0.016

Humidity Attribute

- Check 75 as a threshold for humidity
- Entropy(Decision|Humidity<=75) = $-(1/5).\log_2(1/5) - (4/5).\log_2(4/5)$
  $= 0.721$
- Entropy(Decision|Humidity>75) = $-(4/9).\log_2(4/9) - (5/9).\log_2(5/9)$
  $= 0.991$
- Gain(Decision, Humidity<> 75) = $0.940 - (5/14).(0.721) - (9/14).(0.991)$
  $= 0.940 - 0.2575 - 0.637 = 0.045$
- SplitInfo(Decision, Humidity<> 75) = $-(5/14).\log_2(4/14) -(9/14).\log_2(10/14)$
  $= 0.940$
- GainRatio(Decision, Humidity<> 75) = 0.047

Humidity Attribute

- I *think calculation demonstrations are enough. Now, I skip the calculations and write only results.*
- Gain(Decision, Humidity <> 78) =0.090,
  GainRatio(Decision, Humidity <> 78) =0.090
- **Gain(Decision, Humidity <> 80) = 0.101,**
  GainRatio(Decision, Humidity <> 80) = 0.107
- Gain(Decision, Humidity <> 85) = 0.024,
  GainRatio(Decision, Humidity <> 85) = 0.027
- Gain(Decision, Humidity <> 90) = 0.010,
  GainRatio(Decision, Humidity <> 90) = 0.016
- Gain(Decision, Humidity <> 95) = 0.048,
  GainRatio(Decision, Humidity <> 95) = 0.128
- Here, I ignore the value 96 as threshold because humidity cannot be greater than this value.
- As seen, gain maximizes when threshold is equal to 80 for humidity. This means that we need to compare other nominal attributes and comparison of humidity to 80 to create a branch in our tree.

Temperature Attribute

- Temperature feature is continuous as well. When I apply binary split to temperature for all possible split points, the following decision rule maximizes for both gain and gain ratio.
- **Gain(Decision, Temperature <> 83) = 0.113,**

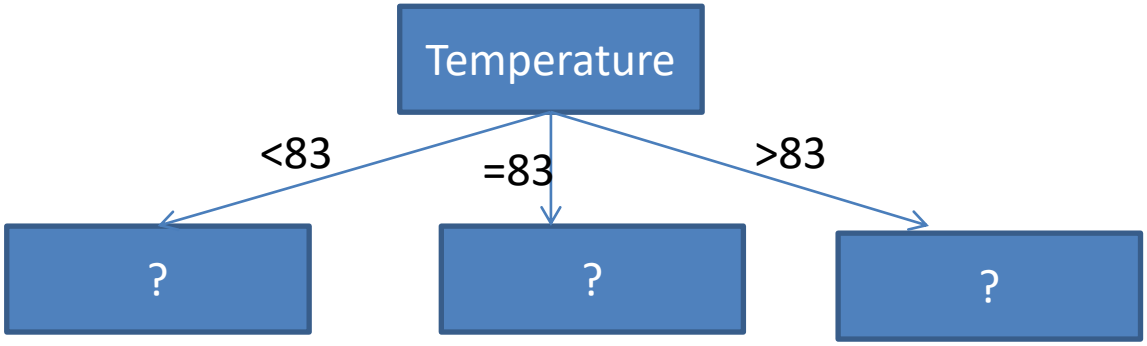
  GainRatio(Decision, Temperature<> 83) = 0.305

Time to decide

Attribute	Gain	GainRatio
Wind	0.049	0.049
Outlook	0.246	0.155
Humidity <> 80	0.101	0.107
Temperature <> 83	0.113	0.305

- If we will use gain metric as in ID3, then outlook will be the root node because it has the highest gain value.
- On the other hand, if we use gain ratio metric, then temperature will be the root node because it has the highest gain ratio value.

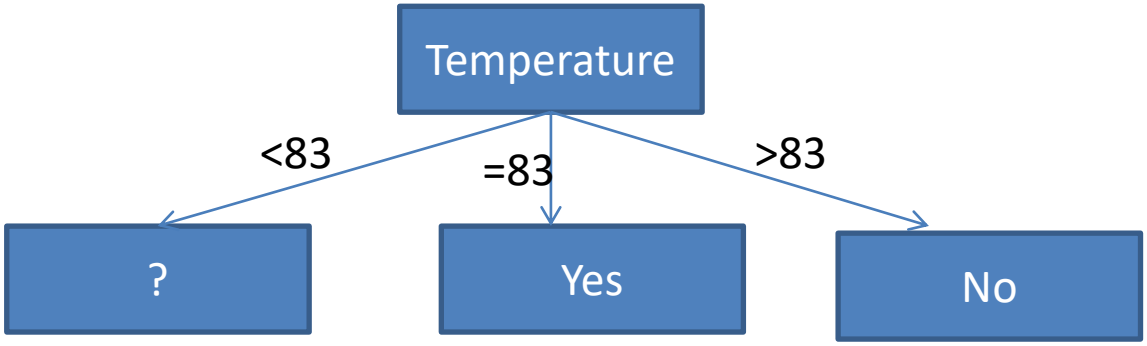
Root Node



- For temperature = 83, there is single example.
- Decision related to temperature = 83 is Yes.
- It will be a leaf node.

- For Temperature >83, there is one example.
- Decision related to example is No.
- It will be a leaf node.

Root Node



Second Level (Temperature < 83)

- Compute the gain ration for weather, wind and humidity.
- Number of examples correspond to temperature <83 are 12.
- Out of 12 examples, 8 instances refer to yes decision, and 4 instances refer to no decision.
- Entropy(decision | temp<83) =$\sum$ – p(I) . $\log_2$p(I)
  = – p(Yes) . $\log_2$p(Yes) – p(No) . $\log_2$p(No)
  = – (8/12) . $\log_2$(8/12) – (4/12) . $\log_2$(4/12)
  = 0.39+0.53= 0.92

Second Level (Temperature < 83) Wind Attribute

- Wind is a nominal attribute. Its possible values are weak and strong.
- There are 6 weak wind instances. 1 of them are concluded as no, 5 of them are concluded as yes.
- Entropy(temp<83 |Wind=Weak) = – p(No) . $\log_2$p(No) – p(Yes) . $\log_2$p(Yes)
  = – (1/6) . $\log_2$(1/6) – (5/6) . $\log_2$(5/6) = 0.43+0.22= 0.65
- Entropy(temp<83 |Wind=Strong) = – (3/6) . $\log_2$(3/6) – (3/6) . $\log_2$(3/6) = 1
- Gain(temp<83, Wind) = 0.92 – (6/12).(0.65) – (6/12).(1)
  = 0.095
- There are 6 decisions for weak wind, and 6 decisions for strong wind.
- SplitInfo(temp<83, Wind) = -(6/12).$\log_2$(6/12) – (6/12).$\log_2$(6/12)
  = 0.5 + 0.5 = 1
- GainRatio(temp<83, Wind) = Gain(temp<83, Wind) / SplitInfo(temp<83, Wind)
  = 0.095 / 1 = 0.095

Outlook Attribute

- Outlook is a nominal attribute, too. Its possible values are sunny, overcast and rain.
- Gain(Decision, Outlook) = Entropy(Decision) –
  ∑ (p(Decision|Outlook) . Entropy(Decision|Outlook))
- Gain(Decision, Outlook) = Entropy(Decision) –
  p(Decision|Outlook=Sunny) . Entropy(Decision|Outlook=Sunny) –
  p(Decision|Outlook=Overcast) . Entropy(Decision|Outlook=Overcast) –
  p(Decision|Outlook=Rain) . Entropy(Decision|Outlook=Rain)
- There are 4 sunny instances. 2 of them are concluded as no, 2 of them are concluded as yes.
- There are 3 overcast instances. All 3 are concluded as yes.
- There are 5 rainy instances. 2 of them are concluded as no, 2 of them are concluded as yes.

- Entropy(temp<83|Outlook=Sunny) = – p(No) . $\log_2$p(No) – p(Yes) . $\log_2$p(Yes) =
  -(2/4).$\log_2$(2/4) – (2/4).$\log_2$(2/4) = 0.5 + 0.5 = 1
- Entropy(temp<83|Outlook=Overcast) = – p(No) . $\log_2$p(No) – p(Yes) . $\log_2$p(Yes) =
  -(0/4).$\log_2$(0/4) – (4/4).$\log_2$(4/4) = 0
- Entropy(temp<83|Outlook=Rain) = – p(No) . $\log_2$p(No) – p(Yes) . $\log_2$p(Yes) =
  -(2/5).$\log_2$(2/5) – (3/5).$\log_2$(3/5) = 0.528 + 0.441 = 0.970
- Gain(temp<83, Outlook) = 0.92 – (4/12).(1) – (3/12).(0) – (5/12).(0.970)
  = 0.18
- There are 4 instances for sunny, 3 instances for overcast and 5 instances for rain.
- SplitInfo(temp<83, Outlook) = -(4/12).$\log_2$(4/12) -(3/12).$\log_2$(3/12) -(5/12).$\log_2$(5/12)
  = 0.53+0.5+0.53= 1.56
- GainRatio(temp<83, Outlook) = Gain(temp<83, Outlook)/SplitInfo(temp<83, Outlook)
  = 0.18/1.56 = 0.11

Exercise: Second Level (Temperature < 83) Humidity

- Compute gain ratio for humidity considering threshold = 80.

Conclusion

- C4.5 algorithm uses gain ratios instead of gains.
- In this way, it creates more generalized trees and not to fall into overfitting.
- Moreover, the algorithm transforms continuous attributes to nominal ones based on gain maximization.
- Additionally, it can ignore instances including missing data and handle missing dataset.
- On the other hand, both ID3 and C4.5 requires high CPU and memory demand.

Thank You !!!