C4.5 Algorithm

The **C4.5 algorithm**, developed by **Ross Quinlan**, is an extension of the **ID3 (Iterative Dichotomiser 3)** algorithm. It is used for **classification tasks** by constructing a decision tree based on the concept of **Information Gain Ratio** rather than just **Information Gain**, which was used in ID3. C4.5 handles both categorical and continuous attributes and is capable of dealing with missing values.

Entropy Calculation

Entropy is a measure of impurity or uncertainty in a dataset. It quantifies the amount of randomness or disorder in a system. The entropy for a dataset S with binary classification (Yes and No) is given by:

$$E(S) = -p_+ \log_2(p_+) - p_- \log_2(p_-)$$

where:

- p_+ = proportion of **positive** instances (Yes)
- p_- = proportion of **negative** instances (No)

A **higher entropy** value indicates greater disorder, meaning the data is more mixed. A **lower entropy** value indicates a more pure split.

Information Gain

Information Gain (IG) measures the effectiveness of an attribute in reducing uncertainty in classification. It is defined as:

$$IG(S, A) = E(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} E(S_v)$$

where:

- S = original dataset
- A = attribute for which IG is calculated
- $Values(A)$ = possible values of attribute A
- S_v = subset of S where attribute A has value v
- $|S_v|/|S|$ = proportion of S_v in S
- $E(S_v)$ = entropy of subset S_v

A higher **Information Gain** means that splitting on attribute A results in a more significant reduction in entropy, making it a better candidate for tree splitting.

Gain Ratio (GR) – Overcoming Bias in Information Gain

C4.5 improves upon ID3 by introducing **Gain Ratio**, which adjusts for the bias that Information Gain has towards attributes with a large number of unique values. The **Gain Ratio** is defined as:

$$GR(A) = \frac{IG(S, A)}{SplitInfo(A)}$$

where **Split Information** is calculated as:

$$SplitInfo(A) = -\sum_{v \in Values(A)} \frac{|S_v|}{|S|} \log_2 \frac{|S_v|}{|S|}$$

Why use Gain Ratio?

- **Information Gain alone** tends to favor attributes with many unique values.
- **Gain Ratio normalizes IG** by dividing it by Split Information, preventing bias towards attributes with more values.

Handling Continuous Attributes

C4.5 extends ID3 by handling continuous attributes. It does so by:

1. Sorting the values of the continuous attribute.
2. Finding possible split points based on class changes.
3. Selecting the best threshold using **Gain Ratio**.

For a continuous attribute A with values $v_1, v_2, ..., v_n$, the **best split** is determined by evaluating midpoints:

$$Threshold = \frac{v_i + v_{i+1}}{2}$$

where v_i and v_{i+1} belong to different classes.

Handling Missing Values

C4.5 can handle missing values by:

1. Ignoring missing values when computing entropy.
2. Assigning probabilities to missing attribute values based on frequency in the dataset.
3. Distributing instances with missing values proportionally across branches in the decision tree.

Pruning – Avoiding Overfitting

C4.5 uses **post-pruning** to prevent overfitting:

- **Reduced-Error Pruning:** Removes branches that do not significantly improve classification accuracy.
- **Subtree Replacement:** Replaces subtrees with leaf nodes if it improves performance.
- **Subtree Raising:** Moves subtrees up if it generalizes better.

Conclusion

C4.5 is an improvement over ID3 due to:

- **Use of Gain Ratio** instead of just Information Gain.
- **Handling of continuous attributes**.
- **Dealing with missing values**.
- **Pruning to reduce overfitting**.

It is widely used for classification problems and forms the foundation for many advanced decision tree algorithms, including **C5.0**.

C4.5 Decision Tree

Step 1: Sample Dataset

Let's consider a dataset for **Weather and Play Decision**:

ID	Outlook	Temperature	Humidity	Wind	Play (Target)
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

Total: 14 instances

- **Yes** = 9
- **No** = 5

Step 2: Calculate Entropy of the Dataset

Entropy Formula:

$$E(S) = -p_+ \log_2(p_+) - p_- \log_2(p_-)$$

where:

- p_+ = proportion of **Yes**

- p_- = proportion of **No**

$$E(S) = -\left(\frac{9}{14}\log_2\frac{9}{14}\right) - \left(\frac{5}{14}\log_2\frac{5}{14}\right)$$

$$= -(0.643 \times -0.636) - (0.357 \times -1.485)$$

$$= 0.636 + 0.530 = 0.940$$

So, **Entropy(S) = 0.940**.

Step 3: Compute Information Gain for Each Attribute

3.1 Information Gain for 'Outlook'

Outlook	Yes	No	Total	Entropy
Sunny	2	3	5	$E = -\frac{2}{5}\log_2\frac{2}{5} - \frac{3}{5}\log_2\frac{3}{5}$ = **0.971**
Overcast	4	0	4	$E = -\frac{4}{4}\log_2\frac{4}{4} - 0$ = **0.000**
Rain	3	2	5	$E = -\frac{3}{5}\log_2\frac{3}{5} - \frac{2}{5}\log_2\frac{2}{5}$ = **0.971**

$$E_{Outlook} = \frac{5}{14} \times 0.971 + \frac{4}{14} \times 0.000 + \frac{5}{14} \times 0.971$$

$$= 0.346 + 0 + 0.346 = 0.693$$

$$Gain(Outlook) = 0.940 - 0.693 = 0.247$$

3.2 Information Gain for 'Temperature'

Temperature	Yes	No	Total	Entropy
Hot	2	2	4	$E = 1.000$
Mild	4	2	6	$E = 0.918$
Cool	3	1	4	$E = 0.811$

$$E_{Temperature} = \frac{4}{14} \times 1.000 + \frac{6}{14} \times 0.918 + \frac{4}{14} \times 0.811$$

$$= 0.286 + 0.393 + 0.232 = 0.911$$

$$Gain(Temperature) = 0.940 - 0.911 = 0.029$$

3.3 Information Gain for 'Humidity'

Humidity	Yes	No	Total	Entropy
High	3	4	7	$E = 0.985$
Normal	6	1	7	$E = 0.591$

$$E_{Humidity} = \frac{7}{14} \times 0.985 + \frac{7}{14} \times 0.591$$

$$= 0.493 + 0.296 = 0.789$$

$$Gain(Humidity) = 0.940 - 0.789 = 0.151$$

3.4 Information Gain for 'Wind'

Wind	Yes	No	Total	Entropy
Weak	6	2	8	$E = 0.811$
Strong	3	3	6	$E = 1.000$

$$E_{Wind} = \frac{8}{14} \times 0.811 + \frac{6}{14} \times 1.000$$

$$= 0.463 + 0.429 = 0.892$$

$$Gain(Wind) = 0.940 - 0.892 = 0.048$$

Step 4: Choose Attribute with Highest Gain

- **Outlook** = **0.247** (Highest)
- **Humidity** = **0.151**
- **Wind** = **0.048**
- **Temperature** = **0.029**

So, **Outlook** is the root node.

Step 5: Build the Decision Tree

1. **If Outlook = Overcast → Play = Yes** (100% Yes)
2. **If Outlook = Sunny:**
   - If Humidity = High → Play = No
   - If Humidity = Normal → Play = Yes
3. **If Outlook = Rain:**
   - If Wind = Strong → Play = No
   - If Wind = Weak → Play = Yes

This is the final **C4.5 Decision Tree**, showing the best split based on **Entropy and Information Gain**.

@SSRoy