Decision Trees

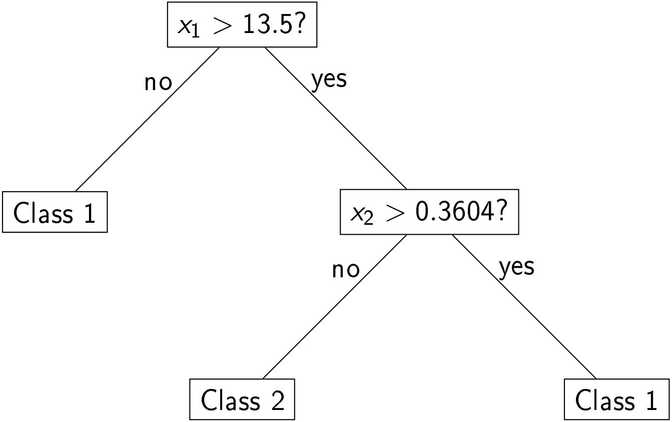
Ravi Kothari, Ph.D.
ravi.kothari@ashoka.edu.in

"It is in moments of your decision that your destiny is shaped" - Tony Robbins

Decision Trees

Decision Trees

- Intuitive and easy to interpret classification



Decision Tree Induction

Decision Tree Induction

- All the training patterns are at the root node to begin with

Decision Tree Induction

- All the training patterns are at the root node to begin with
- A *node splitting criteria* is used to separate the patterns amongst the children nodes. Entropy is a popular node splitting criteria and the attribute/criteria used for splitting is the one that results in the maximum decrease in entropy

Decision Tree Induction

- All the training patterns are at the root node to begin with
- A *node splitting criteria* is used to separate the patterns amongst the children nodes. Entropy is a popular node splitting criteria and the attribute/criteria used for splitting is the one that results in the maximum decrease in entropy
- The process continues until the leaf node is *pure* i.e. contains patterns of only one class

Decision Tree Induction

- All the training patterns are at the root node to begin with
- A *node splitting criteria* is used to separate the patterns amongst the children nodes. Entropy is a popular node splitting criteria and the attribute/criteria used for splitting is the one that results in the maximum decrease in entropy
- The process continues until the leaf node is *pure* i.e. contains patterns of only one class
- Each node thus partitions the input space and the decisions represent a greedy decision (why?)

Information Gain Based Node Splitting

Information Gain Based Node Splitting

- We assume there are a total of C classes denoted by $\Omega = \{\omega_1, \omega_2, \ldots, \omega_C\}$

Information Gain Based Node Splitting

- We assume there are a total of C classes denoted by $\Omega = \{\omega_1, \omega_2, \ldots, \omega_C\}$
- At a particular node in the tree, let there be N training examples represented by, $(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots, (x^{(N)}, y^{(N)})$

Information Gain Based Node Splitting

- We assume there are a total of C classes denoted by $\Omega = \{\omega_1, \omega_2, \ldots, \omega_C\}$
- At a particular node in the tree, let there be N training examples represented by, $(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots, (x^{(N)}, y^{(N)})$
- Of these N examples, N_{ω_k} belong to class ω_k. $\sum_k N_{\omega_k} = N$
- The decision rule at the node splits these examples into V partitions, or V *child* nodes, each of which has $N^{(v)}$ examples. In a particular partition, the number of examples of class ω_k is denoted by $N_{\omega_k}^{(v)}$. $\sum_k N_{\omega_k}^{(v)} = N^{(v)}$

C4.5

C4.5

- A widely used algorithm for decision tree induction

C4.5

- A widely used algorithm for decision tree induction
- Chooses a split that maximizes the gain in information

C4.5

- A widely used algorithm for decision tree induction
- Chooses a split that maximizes the gain in information Information gain can be computed as,

$$\begin{aligned} G(x_j) &= \left[\sum_{k=1}^{C} -\left(\frac{N_{\omega_k}}{N}\right)\log\left(\frac{N_{\omega_k}}{N}\right)\right] \\ &\quad - \left[\sum_{v=1}^{V}\left(\frac{N^{(v)}}{N}\right)\sum_{k=1}^{C} -\left(\frac{N_{\omega_k}^{(v)}}{N^{(v)}}\right)\log\left(\frac{N_{\omega_k}^{(v)}}{N^{(v)}}\right)\right] \end{aligned}$$

C4.5

- A widely used algorithm for decision tree induction
- Chooses a split that maximizes the gain in information Information gain can be computed as,

$$
\begin{aligned}
G(x_j) \;=\; & \left[\sum_{k=1}^{C} -\left(\frac{N_{\omega_k}}{N}\right)\log\left(\frac{N_{\omega_k}}{N}\right)\right] \\
& - \left[\sum_{v=1}^{V}\left(\frac{N^{(v)}}{N}\right)\sum_{k=1}^{C} -\left(\frac{N_{\omega_k}^{(v)}}{N^{(v)}}\right)\log\left(\frac{N_{\omega_k}^{(v)}}{N^{(v)}}\right)\right]
\end{aligned}
$$

- The first term is the entropy at the parent node and the second term is the weighted entropy of the child nodes. The attribute chosen is the one that results in the largest information gain

C4.5 (contd.)

C4.5 (contd.)

- It is possible that a large number of splits (children) result in an attempt to maximize the information gain. For example, a variable like date can be used to get pure leaf nodes (many many of them). However, such a tree would be very bad at prediction

C4.5 (contd.)

- It is possible that a large number of splits (children) result in an attempt to maximize the information gain. For example, a variable like date can be used to get pure leaf nodes (many many of them). However, such a tree would be very bad at prediction
- We thus introduce a term that penalizes too many splits. This term called *Split-Info* is defined as,

$$g = -\sum_{v=1}^{V} \left(\frac{N^{(v)}}{N} \right) \log \left(\frac{N^{(v)}}{N} \right)$$

C4.5 (contd.)

- It is possible that a large number of splits (children) result in an attempt to maximize the information gain. For example, a variable like date can be used to get pure leaf nodes (many many of them). However, such a tree would be very bad at prediction
- We thus introduce a term that penalizes too many splits. This term called *Split-Info* is defined as,

$$g = -\sum_{v=1}^{V} \left(\frac{N^{(v)}}{N} \right) \log \left(\frac{N^{(v)}}{N} \right)$$

- We thus use the attribute that maximizes the *Gain-Ratio* i.e. $G(x_j)/g$

Example

Consider the dataset (from `https://sefiks.com/2018/05/13/a-step-by-step-c4-5-decision-tree-example/`),

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
13	Overcast	81	75	Weak	Yes
14	Rain	71	80	Strong	No

Example

Example

- The entropy at the root node is, $H(0) = -\sum p_i \log_2 p_i$
$= -(5/14)\log(5/14) - (9/14 \log 9/14) = -0.940$. log is base 2

Example

- The entropy at the root node is, $H(0) = -\sum p_i \log_2 p_i$ $= -(5/14)\log(5/14) - (9/14 \log 9/14) = -0.940$. log is base 2
- Let us take each attribute at a time. Wind is either "Weak" or "Strong". When Wind is "Strong", we get 3 examples of class 'No" and 3 examples of class "Yes". When Wind is "Weak", we get 2 examples of class "No" and 6 examples of class "Yes". So, the weighted entropy at the child node is:

$$[(6/14)(-(3/6)\log(3/6) - (3/6)\log(3/6))] + [(8/14)(-(2/8)\log(2/$$

$$[(6/14)(1.0) + (8/14)(0.811)]$$

Example

- The entropy at the root node is, $H(0) = -\sum p_i \log_2 p_i$ $= -(5/14)\log(5/14) - (9/14 \log 9/14) = -0.940$. log is base 2
- Let us take each attribute at a time. Wind is either "Weak" or "Strong". When Wind is "Strong", we get 3 examples of class 'No" and 3 examples of class "Yes". When Wind is "Weak", we get 2 examples of class "No" and 6 examples of class "Yes". So, the weighted entropy at the child node is:

$$[(6/14)(-(3/6)\log(3/6) - (3/6)\log(3/6))] + [(8/14)(-(2/8)\log(2/$$
$$[(6/14)(1.0) + (8/14)(0.811)]$$

- So, the gain due to the use of Wind is $G(Wind) = 0.940 - (6/14)(1) - (8/14)(0.811) = 0.049$. For Split-Info, note that the Wind is "Strong" has 6 patterns and Wind is "Weak" has 8 patterns. So, $g = -(8/14)\log(8/14) - (6/14)\log(6/14) = 0.985$

Example

- The entropy at the root node is, $H(0) = -\sum p_i \log_2 p_i$
  $= -(5/14)\log(5/14) - (9/14 \log 9/14) = -0.940$. log is base 2
- Let us take each attribute at a time. Wind is either "Weak" or "Strong". When Wind is "Strong", we get 3 examples of class 'No" and 3 examples of class "Yes". When Wind is "Weak", we get 2 examples of class "No" and 6 examples of class "Yes". So, the weighted entropy at the child node is:

$$[(6/14)(-(3/6)\log(3/6) - (3/6)\log(3/6))] + [(8/14)(-(2/8)\log(2/$$

$$[(6/14)(1.0) + (8/14)(0.811)]$$

- So, the gain due to the use of Wind is $G(Wind) = 0.940 - (6/14)(1) - (8/14)(0.811) = 0.049$. For Split-Info, note that the Wind is "Strong" has 6 patterns and Wind is "Weak" has 8 patterns. So,
  $g = -(8/14)\log(8/14) - (6/14)\log(6/14) = 0.985$
  So, $GainRatio = 0.049/0.985 = 0.049$

Example

Example

- Now let us take humidity which is a continuous attribute. To convert that to a discrete attribute, we first sort the values and then explore each possible point as a potential threshold for discretizing. The threshold that provides the maximum gain is used

Example

- Now let us take humidity which is a continuous attribute. To convert that to a discrete attribute, we first sort the values and then explore each possible point as a potential threshold for discretizing. The threshold that provides the maximum gain is used
- As an illustration, consider the threshold of 75. So, Humidity ≤ 75 is "Low" and humidity greater than 75 is "High". So, the weighted entropy is,

$$[(5/14)(-(1/5)\log(1/5) - (4/5)\log(4/5))] + [(9/14)(-(4/9)\log(4/$$

$$[(5/14)(0.721) + (9/14)(0.991)]$$

Example

- Now let us take humidity which is a continuous attribute. To convert that to a discrete attribute, we first sort the values and then explore each possible point as a potential threshold for discretizing. The threshold that provides the maximum gain is used
- As an illustration, consider the threshold of 75. So, Humidity ≤ 75 is "Low" and humidity greater than 75 is "High". So, the weighted entropy is,

$$[(5/14)(-(1/5)\log(1/5) - (4/5)\log(4/5))] + [(9/14)(-(4/9)\log(4/$$

$$[(5/14)(0.721) + (9/14)(0.991)]$$

- So, the gain due to the use of Humidity is $G(Humidity \leq 75) = 0.940 - (5/14)(0.721) - (9/14)(0.991) = 0.045$. For Split-Info, note that the Humidity is "Low" has 5 patterns and Humidity is "High" has 9 patterns. So, $g = -(5/14)\log(5/14) - (9/14)\log(9/14) = 0.940$

Example

- Now let us take humidity which is a continuous attribute. To convert that to a discrete attribute, we first sort the values and then explore each possible point as a potential threshold for discretizing. The threshold that provides the maximum gain is used
- As an illustration, consider the threshold of 75. So, Humidity ≤ 75 is "Low" and humidity greater than 75 is "High". So, the weighted entropy is,

$$[(5/14)(-(1/5)\log(1/5) - (4/5)\log(4/5))] + [(9/14)(-(4/9)\log(4/$$
$$[(5/14)(0.721) + (9/14)(0.991)]$$

- So, the gain due to the use of Humidity is $G(Humidity \leq 75) = 0.940 - (5/14)(0.721) - (9/14)(0.991) = 0.045$. For Split-Info, note that the Humidity is "Low" has 5 patterns and Humidity is "High" has 9 patterns. So, $g = -(5/14)\log(5/14) - (9/14)\log(9/14) = 0.940$
  So, $Gain\text{-}Ratio = 0.045/0.940 = 0.047$

Example

- Now let us take humidity which is a continuous attribute. To convert that to a discrete attribute, we first sort the values and then explore each possible point as a potential threshold for discretizing. The threshold that provides the maximum gain is used
- As an illustration, consider the threshold of 75. So, Humidity ≤ 75 is "Low" and humidity greater than 75 is "High". So, the weighted entropy is,

$$[(5/14)(-(1/5)\log(1/5) - (4/5)\log(4/5))] + [(9/14)(-(4/9)\log(4/$$

$$[(5/14)(0.721) + (9/14)(0.991)]$$

- So, the gain due to the use of Humidity is $G(Humidity \leq 75) = 0.940 - (5/14)(0.721) - (9/14)(0.991) = 0.045$. For Split-Info, note that the Humidity is "Low" has 5 patterns and Humidity is "High" has 9 patterns. So, $g = -(5/14)\log(5/14) - (9/14)\log(9/14) = 0.940$