|  |  |
| --- | --- |
| Faculty of Computer & Information Sciences  Ain Shams University  Subject: (SCC 234)  Statistical Analysis and Applications  Year: (2nd year) undergraduate  Academic year: 2nd term 2019-2020 |  |

**Research Topic (C)**

**Title: Using Probability to Build Decision Trees for Classification**

**What ‘s a Decision Tree ?** (Ref. 1)

A decision tree is a diagram or chart that people use to determine a course of action or show a statistical probability. It forms the outline of the namesake woody plant, usually upright but sometimes lying on its side. Each branch of the decision tree represents a possible decision, outcome, or reaction. The farthest branches on the tree represent the end results.

Individuals use decision trees to clarify and find an answer to a complex problem. Decision trees are frequently employed in determining a course of action in finance, investing, or business.

The Basics of a Decision Tree :

A decision tree is a graphical depiction of a decision and every potential outcome or result of making that decision. Individuals deploy decision trees in a variety of situations, from something simple and personal ("Should I go out for dinner?") to more complex industrial, scientific or microeconomic undertakings.

By displaying a sequence of steps, decision trees give people an effective and easy way to visualize and understand the potential options of a decision and its range of possible outcomes. The decision tree also helps people identify every potential option and weigh each course of action against the risks and rewards each option can yield.

An organization may deploy decision trees as a kind of decision support system. The structured model allows the reader of the chart to see how and why one choice may lead to the next, with the use of the branches indicating mutually exclusive options. The structure allows users to take a problem with multiple possible solutions and to display those solutions in a simple, easy-to-understand format that also shows the relationship between different events or decisions.

In the decision tree, each end result has an assigned risk and reward weight or number. If a person uses a decision tree to make a decision, they look at each final outcome and assess the benefits and drawbacks. The tree itself can span as long or as short as needed in order to come to a proper conclusion.

**Mathematical formulation of Decision Trees.** (Ref. 2)

**Information gain** IG(A) is the measure of the difference in entropy from before to after the set S is split on an attribute A. In other words, how much uncertainty in S was reduced after splitting set S on attribute A.

IG(A,S) = H(S) -

Where,

* H(S) – Entropy of set S
* T – The subsets created from splitting set S by attribute A.
* P(t) – The proportion of the number of elements in t to the number of elements in set S
* H(t) – Entropy of subset t.

In ID3, information gain can be calculated (instead of entropy) for each remaining attribute. The attribute with the largest information gain is used to split the set S o()n this iteration.

**Entropy** H(S) is a measure of the amount of uncertainty in the (data) set S (i.e. entropy characterizes the (data) set S).

H(S) =

Where,

* S – is the current (data) set for which entropy is being calculated (changes every iteration of the ID3 algorithm)
* c – Set of classes in S c = { yes, no}
* p(c) – The proportion of the number of elements in class c to the number of elements in set S.

When H(S) = 0, the set S is classified (i.e. all elements in S are of the same class ).

In ID3, entropy is calculated for each remaining attribute. The attribute with the smallest entropy is used to split the set S on this iteration.

The higher the entropy, the higher the potential to improve the classification here.

**What are the terms in the Decision Trees mean and the intuitions behind them?**

Important Terminology related to Decision Trees

* Root Node: It represents the entire population or sample and this further gets divided into two or more homogeneous sets.
* Splitting: It is a process of dividing a node into two or more sub-nodes.
* Decision Node: When a sub-node splits into further sub-nodes, then it is called the decision node.
* Leaf / Terminal Node: Nodes do not split is called Leaf or Terminal node.
* Pruning: When we remove sub-nodes of a decision node, this process is called pruning. You can say the opposite process of splitting.
* Branch / Sub-Tree: A subsection of the entire tree is called branch or sub-tree.
* Parent and Child Node: A node, which is divided into sub-nodes is called a parent node of sub-nodes whereas sub-nodes are the child of a parent node.

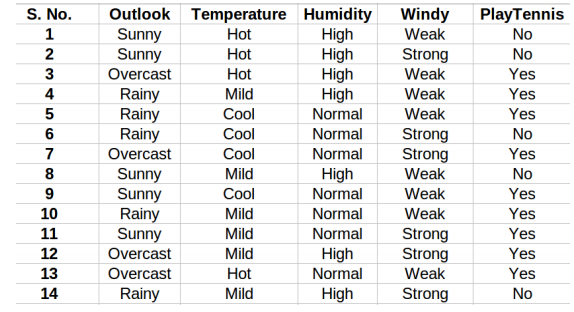
A decision tree is mainly made up of 6 different nodes (components), which are:

* Start Page: an optional introduction page
* Questions & Answers: ask your leads questions
* Decision: process the answers chosen by your leads
* Forms: to display legal disclaimers, terms & conditions, etc. and collect consent
* ESPs: email service providers such as Aweber, MailChimp and etc.
* Result Pages: display the results based on your leads answers.

**Explain the ID3 method to construct the decision tress how it is related to probability?**

**Working numerical examples of how Decision Trees are built.**

**Ex ;** Make a decision tree that predicts whether tennis will be played on the day?



**Solution in detail:**

**Step 1 : We’ll create a Root Node**

* How to choose the root node?

The attribute that best classifies the training data, use this

attribute at the root of the tree.

* How to choose the best attribute?

So from here, ID3 algorithm begins

- Calculate **Entropy** (The amount of uncertainty in dataset):

Entropy =

- Calculate **Average Information :**

I (Attribute) = Entropy (A)

- Calculate **Information Gain:** (Difference in Entropy before and after splitting dataset on attribute A)

**Gain = Entropy (S) – I (Attribute)**

1.compute the entropy for data-set Entropy(s)

2.for every attribute/feature:

1.calculate entropy for all other values Entropy(a)

2.take average information entropy for the current attribute

3.calculate gain for the current attribute

3. pick the highest gain attribute.

4. Repeat until we get the tree we desired.

P = 9 , N = 5 , Total = 14

* Calculate **Entropy (S)**

Entropy =

Entropy(S) =

Entropy(S) = = **0.940**

For Each Attribute: (let say **Outlook**)

* Calculate Entropy for each values, i.e. for ‘Sunny’, ‘Rainy’, ‘Overcast’

|  |  |
| --- | --- |
| **Outlook** | **Play Tennis** |
| Sunny | No |
| Sunny | No |
| Sunny | No |
| Sunny | Yes |
| Sunny | Yes |

|  |  |
| --- | --- |
| **Outlook** | **Play Tennis** |
| Rainy | Yes |
| Rainy | Yes |
| Rainy | No |
| Rainy | Yes |
| Rainy | Yes |

|  |  |
| --- | --- |
| **Outlook** | **Play Tennis** |
| 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 |

* Calculate **Entropy (Outlook = ‘Value’):**

E (Outlook = sunny) = = **0.971**

E (Outlook = overcast)=

Entropy(S) = = **0.971**

* Calculate **Average Information Entropy**:

I (Outlook) = Entropy (Outlook = Sunny) +

Entropy (Outlook = Rainy) +

Entropy (Outlook = Overcast)

I (Outlook) = \* 0.971 + \* 0.971 + \* 0 = 0.693

* Calculate **Gain**: attribute is Outlook

Gain = Entropy (S) – I (Attribute)

Entropy (S) = 0.940

Gain (Outlook) = 0.940 – 0.693 = 0.247

----------------------------------------------------

For Each Attribute: (let say **Temperature**)

* Calculate Entropy for each Temp, i.e. for ‘Hot’, ‘Mild, ‘Cool’

|  |  |
| --- | --- |
| **Temperature** | **Play Tennis** |
| Mild | Yes |
| Mild | No |
| Mild | Yes |
| Mild | Yes |
| Mild | Yes |
| Mild | No |

|  |  |
| --- | --- |
| **Temperature** | **Play Tennis** |
| Hot | No |
| Hot | No |
| Hot | Yes |
| Hot | Yes |

|  |  |
| --- | --- |
| **Temperature** | **Play Tennis** |
| Cool | Yes |
| Cool | No |
| Cool | Yes |
| Cool | Yes |

|  |  |  |  |
| --- | --- | --- | --- |
| **Temperature** | **p** | **n** | **Entropy** |
| Hot | 2 | 2 | 1 |
| Mild | 4 | 2 | 0.918 |
| Cool | 3 | 1 | 0.811 |

* Calculate **Average Information Entropy**:

I (Temperature) = Entropy (Temperature = Hot) +

Entropy (Temperature = Mild) +

Entropy (Temperature = Cool)

I (Temperature) = \* 0.811 => 0.911

* Calculate **Gain**: attribute is Temperature

Gain = Entropy (S) – I (Attribute)

Entropy (S) = 0.940

Gain (Temperature) = 0.940 – 0.911 = 0.029

----------------------------------------------------

For Each Attribute: (let say **Humidity**)

|  |  |
| --- | --- |
| **Humidity** | **Play Tennis** |
| Normal | Yes |
| Normal | No |
| Normal | Yes |
| Normal | Yes |
| Normal | Yes |
| Normal | Yes |
| Normal | Yes |

* Calculate Entropy for each Temp, i.e. for ‘High’, ‘Normal’

|  |  |
| --- | --- |
| **Humidity** | **Play Tennis** |
| High | No |
| High | No |
| High | Yes |
| High | Yes |
| High | No |
| High | Yes |
| High | No |

|  |  |  |  |
| --- | --- | --- | --- |
| **Humidity** | **p** | **n** | **Entropy** |
| High | 3 | 4 | 0.985 |
| Normal | 6 | 1 | 0.591 |

* Calculate **Average Information Entropy**:

I (**Humidity**) = Entropy (Humidity = High) +

Entropy (Humidity = Normal)

I (Humidity) = \* 0.811 => 0.788

* Calculate **Gain**: attribute is Humidity

Gain = Entropy (S) – I (Attribute)

Entropy (S) = 0.940

Gain (Humidity) = 0.940 – 0.788 = 0.152

------------------------------------------------------

For Each Attribute: (let say **Windy**)

* Calculate Entropy for each Windy, i.e. for ‘Strong’, ‘Weak’

|  |  |
| --- | --- |
| **Windy** | **Play Tennis** |
| Weak | No |
| Weak | Yes |
| Weak | Yes |
| Weak | Yes |
| Weak | No |
| Weak | Yes |
| Weak | Yes |
| Weak | Yes |

|  |  |
| --- | --- |
| **Windy** | **Play Tennis** |
| 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 |

* Calculate **Average Information Entropy**:

I (**Windy**) = Entropy (Windy = Strong) +

Entropy (Windy = )

I (Windy) = => 0.892

* Calculate **Gain**: attribute is Windy

Gain = Entropy (S) – I (Attribute)

Entropy (S) = 0.940

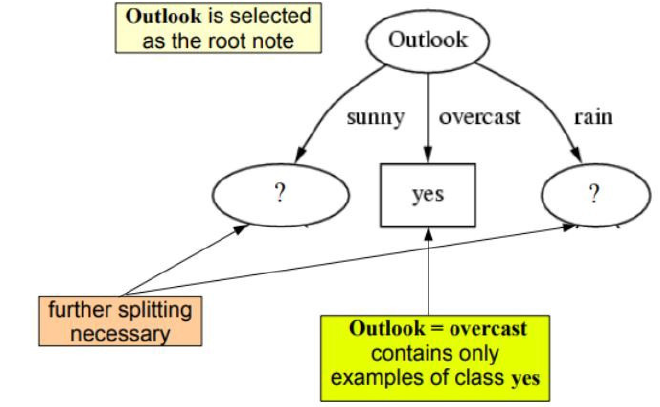
Gain (Windy) = 0.940 – 0.892 = 0.048

**Pick The Highest Gain Attribute.**

|  |  |
| --- | --- |
| **Attributes** | **Gain** |
| Outlook | 0.247 |
| Temperature | 0.029 |
| Humidity | 0.152 |
| Windy | 0.048 |

So 🡺  **Root Node: OUTLOOK 🡸**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Outlook** | **Temperature** | **Humidity** | **Windy** | **Play Tennis** |
| Overcast | Hot | High | Weak | Yes |
| Overcast | Cool | Normal | Strong | Yes |
| Overcast | Mild | High | Strong | Yes |
| Overcast | Hot | Normal | Weak | Yes |

****

* **We’ll repeat the same thing for Sub-Trees till we get the tree.**

Outlook: Sunny

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Outlook** | **Temperature** | **Humidity** | **Windy** | **Play Tennis** |
| Sunny | Hot | High | Weak | No |
| Sunny | Hot | High | Strong | No |
| Sunny | Mild | High | Weak | No |
| Sunny | Cool | Normal | Weak | Yes |
| Sunny | Mild | Normal | Strong | Yes |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Outlook** | **Temperature** | **Humidity** | **Windy** | **Play Tennis** |
| Rainy | Mild | High | Weak | Yes |
| Rainy | Cool | Normal | Weak | Yes |
| Rainy | Cool | Normal | Strong | No |
| Rainy | Mild | Normal | Weak | Yes |
| Rainy | Mild | High | Strong | No |

Outlook: Rainy

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Outlook** | **Temperature** | **Humidity** | **Windy** | **Play Tennis** |
| Sunny | Hot | High | Weak | No |
| Sunny | Hot | High | Strong | No |
| Sunny | Mild | High | Weak | No |
| Sunny | Cool | Normal | Weak | Yes |
| Sunny | Mild | Normal | Strong | Yes |

**P = 2**

**N = 3**

**Total = 5**

* **Entropy:**

Entropy =

Entropy (Sunny) = => **0.971**

For Each Attribute: (let say **Humidity**)

* Calculate Entropy for each Humidity, i.e. for ‘High’, ‘Normal’

|  |  |  |
| --- | --- | --- |
| **Outlook** | **Humidity** | **Play Tennis** |
| Sunny | High | No |
| Sunny | High | No |
| Sunny | High | No |
| Sunny | Normal | Yes |
| Sunny | Normal | Yes |

|  |  |  |  |
| --- | --- | --- | --- |
| **Humidity** | **p** | **n** | **Entropy** |
| high | 0 | 3 | 0 |
| normal | 2 | 0 | 0 |

* Calculate **Average Information Entropy**: I(Humidity) = 0
* Calculate **Gain:** Gain = 0.971

For Each Attribute: (let say **Windy**)

* Calculate Entropy for each Windy, i.e. for ‘Strong’, ‘Weak’

|  |  |  |
| --- | --- | --- |
| **Outlook** | **Windy** | **Play Tennis** |
| Sunny | Strong | No |
| Sunny | Strong | Yes |
| Sunny | Weak | No |
| Sunny | Weak | No |
| Sunny | Weak | Yes |

|  |  |  |  |
| --- | --- | --- | --- |
| **Windy** | **p** | **n** | **Entropy** |
| Strong | 1 | 1 | 1 |
| Weak | 1 | 2 | 0.918 |

* Calculate **Average Information Entropy**: I(Windy) = 0.951
* Calculate **Gain:** Gain = 0.020

For Each Attribute: (let say **Temperature**)

* Calculate Entropy for each Temp, i.e. for ‘Cool’ ,’Hot’, ‘Mild’

|  |  |  |
| --- | --- | --- |
| **Outlook** | **Temperature** | **Play Tennis** |
| Sunny | Cool | Yes |
| Sunny | Hot | No |
| Sunny | Hot | No |
| Sunny | Mild | No |
| Sunny | Mild | Yes |

|  |  |  |  |
| --- | --- | --- | --- |
| **Temperature** | **p** | **n** | **Entropy** |
| Cool | 1 | 0 | 0 |
| Hot | 0 | 2 | 0 |
| Mild | 1 | 1 |  |

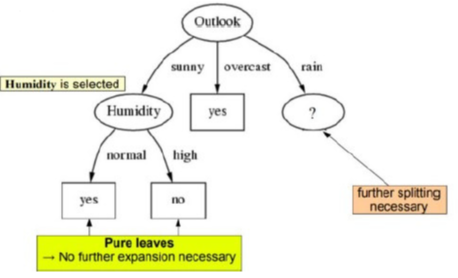
* Calculate **Average Information Entropy**: I(Temp) = 0.4
* Calculate **Gain:** Gain = 0.571

**Pick the highest gain attribute.**

|  |  |
| --- | --- |
| **Attributes** | **Gain** |
| Temperature | 0.571 |
| Humidity | 0.971 |
| Windy | 0.02 |

So 🡺  **Next Node in Sunny: Humidity 🡸**

|  |  |  |
| --- | --- | --- |
| **Outlook** | **Humidity** | **Play Tennis** |
| Sunny | High | No |
| Sunny | High | No |
| Sunny | High | No |
| Sunny | Normal | Yes |
| Sunny | Normal | Yes |

****

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Outlook** | **Temperature** | **Humidity** | **Windy** | **Play Tennis** |
| Rainy | Mild | High | Weak | Yes |
| Rainy | Cool | Normal | Weak | Yes |
| Rainy | Cool | Normal | Strong | No |
| Rainy | Mild | Normal | Weak | Yes |
| Rainy | Mild | High | Strong | No |

**P = 3**

**N = 2**

**Total = 5**

* **Entropy:**

Entropy =

Entropy (S Rainy) = => **0.971**

For Each Attribute: (let say **Humidity**)

* Calculate Entropy for each Humidity, i.e. for ‘High’, ‘Normal’

|  |  |  |
| --- | --- | --- |
| **Outlook** | **Humidity** | **Play Tennis** |
| Rainy | High | Yes |
| Rainy | High | No |
| Rainy | Normal | Yes |
| Rainy | Normal | No |
| Rainy | Normal | Yes |

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **p** | **n** | **Entropy** |
| high | 1 | 1 | 1 |
| normal | 2 | 1 | 0.918 |

* Calculate **Average Information Entropy**: I(Humidity) = 0.951
* Calculate **Gain:** Gain = 0.020

For Each Attribute: (let say **Windy**)

* Calculate Entropy for each Windy, i.e. for ‘Strong’, ‘Weak’

|  |  |  |
| --- | --- | --- |
| **Outlook** | **Windy** | **Play Tennis** |
| Rainy | Strong | No |
| Rainy | Strong | No |
| Rainy | Weak | Yes |
| Rainy | Weak | Yes |
| Rainy | Weak | Yes |

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **p** | **n** | **Entropy** |
| Strong | 0 | 2 | 0 |
| Weak | 3 | 0 | 0 |

* Calculate **Average Information Entropy**: I(Windy) = 0
* Calculate **Gain:** Gain = 0.971

For Each Attribute: (let say **Temperature**)

* Calculate Entropy for each Temp, i.e. for ‘Cool’ ,’Hot’, ‘Mild’

|  |  |  |
| --- | --- | --- |
| **Outlook** | **Temperature** | **Play Tennis** |
| Rainy | Mild | Yes |
| Rainy | Cool | Yes |
| Rainy | Cool | No |
| Rainy | Mild | Yes |
| Rainy | Mild | No |

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **p** | **n** | **Entropy** |
| Cool | 1 | 1 | 1 |
| Mild | 2 | 1 | 0.918 |

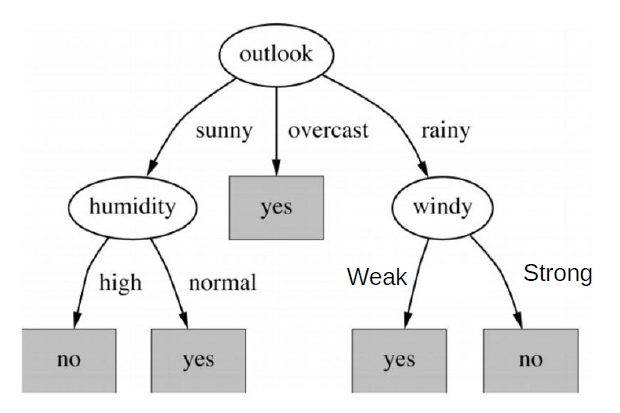
* Calculate **Average Information Entropy**: I(Temp) = 0.951
* Calculate **Gain:** Gain = 0.020

**Pick the highest gain attribute.**

|  |  |
| --- | --- |
| **Attributes** | **Gain** |
| Temperature | 0.02 |
| Humidity | 0.02 |
| Windy | 0.971 |

So 🡺  **Next Node in Rainy: Windy 🡸**

**Final Decision Tree**

****