DECISION TREE AND ID3ALGORITHM

OVERVIEW

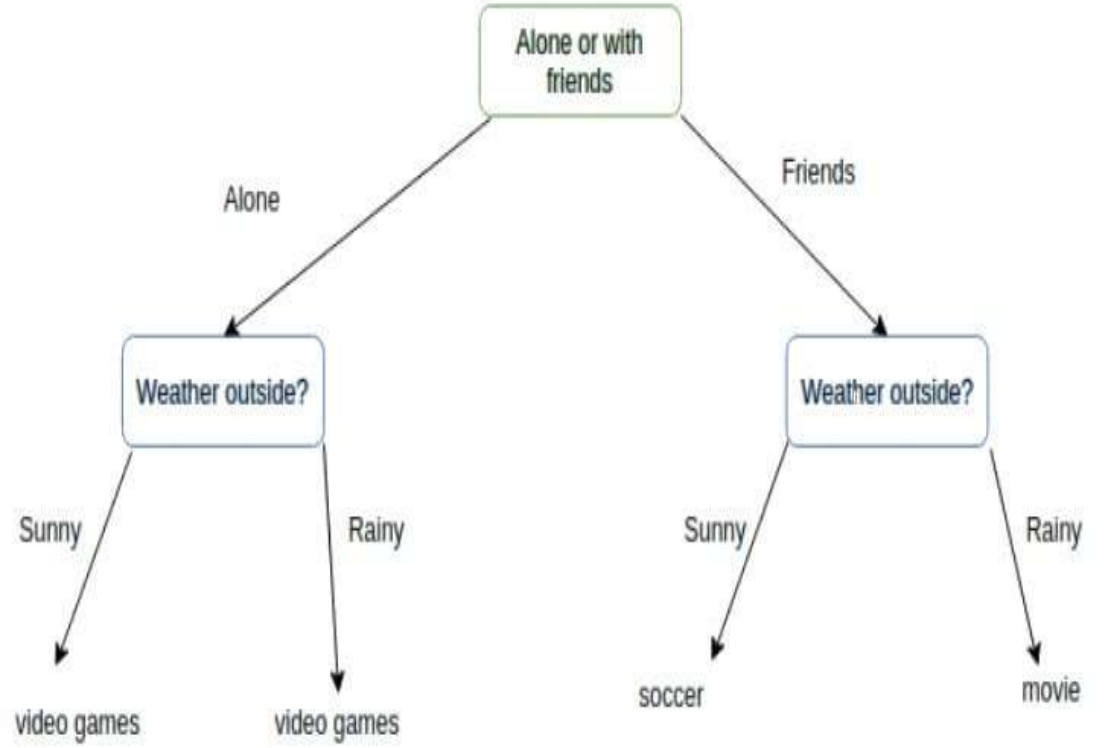
- Introduction
- Decision Tree Representation
- Appropriate Problems for Decision Tree Learning
- Basic Decision Tree Learning Algorithm (ID3)
- Hypothesis Space Search in decision Tree Learning
- Inductive Bias in Decision Tree Learning
- Issues in Decision Tree Learning

OVERVIEW

- **Introduction**
- Decision Tree Representation
- Appropriate Problems for Decision Tree Learning
- Basic Decision Tree Learning Algorithm (ID3)
- Hypothesis Space Search in decision Tree Learning
- Inductive Bias in Decision Tree Learning
- Issues in Decision Tree Learning

INTRODUCTION

- **Decision trees**, one of the simplest and yet most useful Machine Learning method.
- Decision trees, as the name implies, are trees of decisions.
- A decision tree can be used to visually and explicitly represent decisions and decision making.



•Take for example the decision about what activity you should do this weekend.

- ❑It might depend on whether or not you feel like ***going out with your friends*** or ***spending the weekend alone***; in both cases, your decision also ***depends on the weather***.
- ❑If it's ***sunny*** and ***your friends are available***, you may want to ***play soccer***.
- ❑If it ends up ***raining*** you'll go to a ***movie.***
- ❑And if ***your friends don't show*** up at all, well then you like ***playing video games*** no matter what the weather is like!.
- ❑These decisions can be represented by a tree as shown.

INTRODUCTION

- The Decision tree learning is a method for approximating discrete-valued target functions, in which the learned function is represented by a decision tree.
- Learned trees can also be re-represented as sets of if-then rules to improve human readability.
- These learning methods are among the most popular of inductive inference algorithms and have been successfully applied to a broad range of tasks from learning to diagnose medical cases to learning to assess credit risk of loan applicants.
- These decision tree learning methods search a completely expressive hypothesis space and thus avoid the difficulties of restricted hypothesis spaces.
- Their inductive bias is a preference for small trees over large trees.

OVERVIEW

- Introduction
- **Decision Tree Representation**
- Appropriate Problems for Decision Tree Learning
- Basic Decision Tree Learning Algorithm (ID3)
- Hypothesis Space Search in decision Tree Learning
- Inductive Bias in Decision Tree Learning
- Issues in Decision Tree Learning

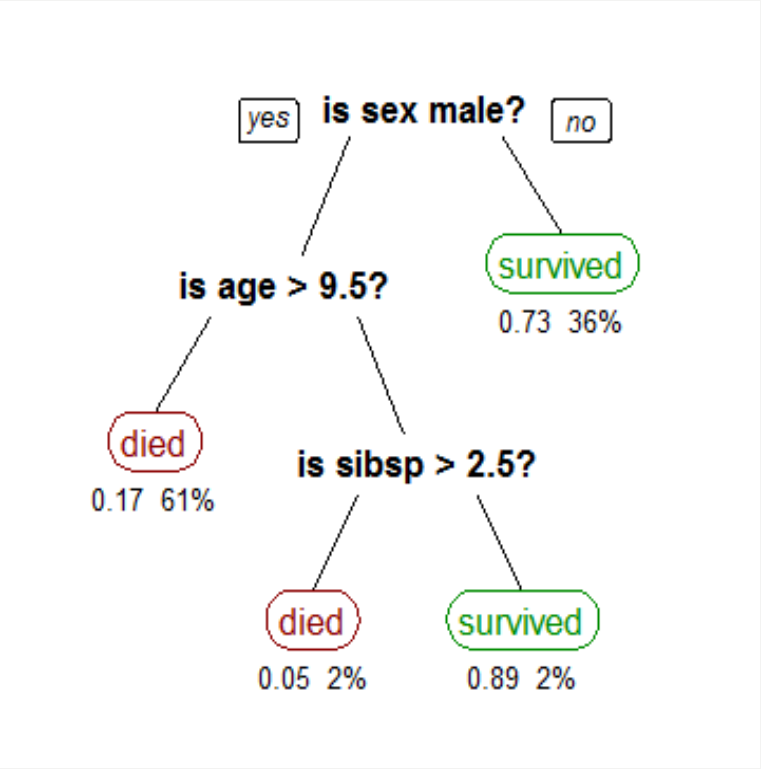
DECISION TREE REPRESENTATION

- A decision tree is drawn upside down with its root at the top.
- In the below figure, the bold text in black represents a condition/**internal node**, based on which the tree splits into branches/ **edges**.
- The end of the branch that doesn't split anymore is the decision/**leaf**, in this case, whether the passenger died or survived, represented as red and green text respectively.
- So the Decision trees classify instances by sorting them down the tree from the root to some leaf node, which provides the classification of the instance.

• **Each node** in the tree specifies a test of some **attribute of the instance**, and each branch descending from that node corresponds to one of the possible **values for this attribute.**

• An instance is classified by starting at the root node of the tree, testing the attribute specified by this node, then moving down the tree branch corresponding to the value of the attribute in the given example.

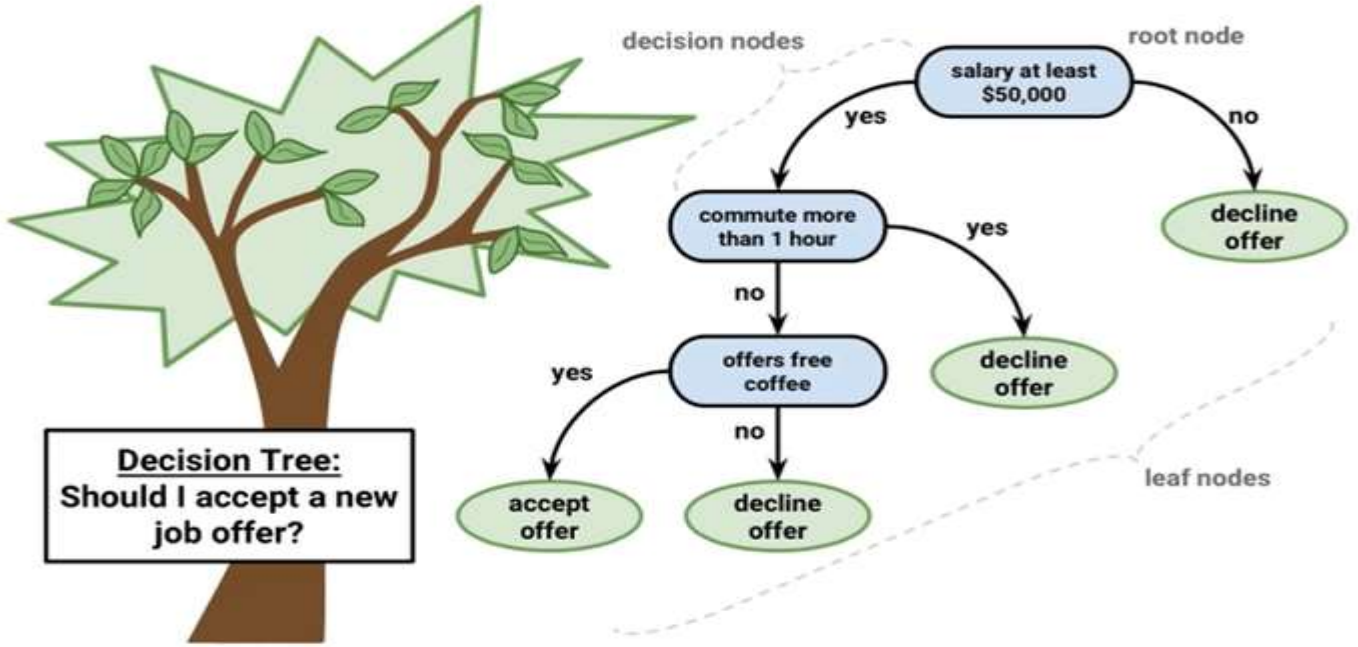
•This process is then repeated for the sub tree rooted at the new node.



DECISION TREE REPRESENTATION

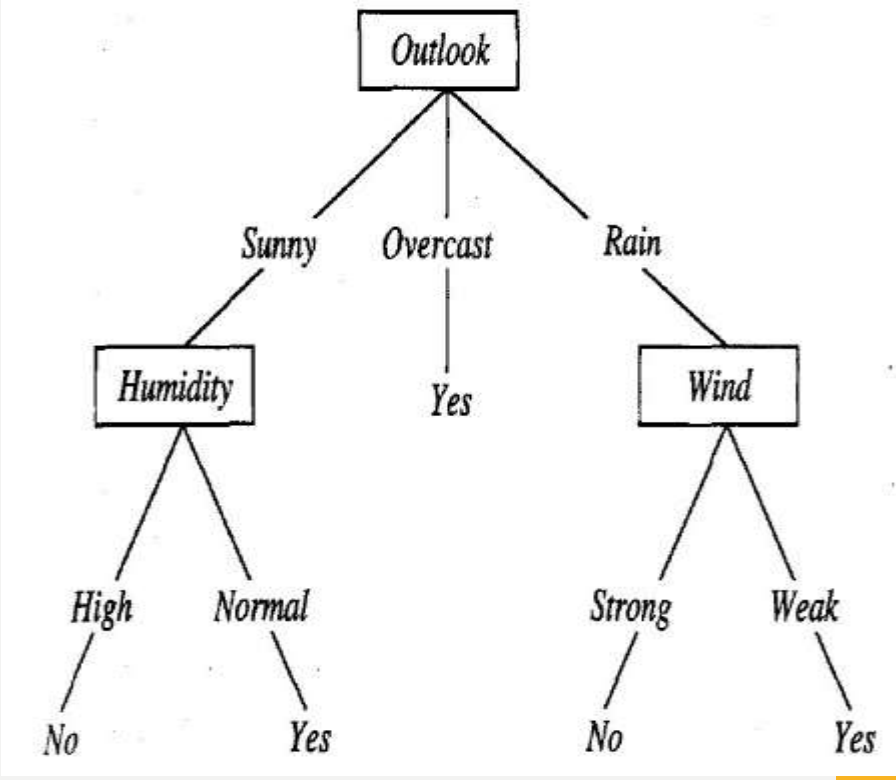
A ***DECISION TREE*** IS A TREE WHERE *EACH NODE* REPRESENTS A **FEATURE (ATTRIBUTE)**, EACH LINK (BRANCH) REPRESENTS A **DECISION (RULE)** AND EACH LEAF REPRESENTS AN **OUTCOME.**

"A **decision tree** is a graphical representation of all the possible solutions to a decision based on certain conditions"



DECISION TREE REPRESENTATION

- Take one more example. The below figure is a learned decision tree for the concept ***PlayTennis.***
- An example is classified by sorting it through the tree to the appropriate leaf node, then returning the classification associated with this leaf.



➢**What is the classification for the below instance ? and whether the day represented by the instance suits for playing tennis or not ?**

(Outlook = Sunny, Temperature = Hot, Humidity = High, Wind = Strong)

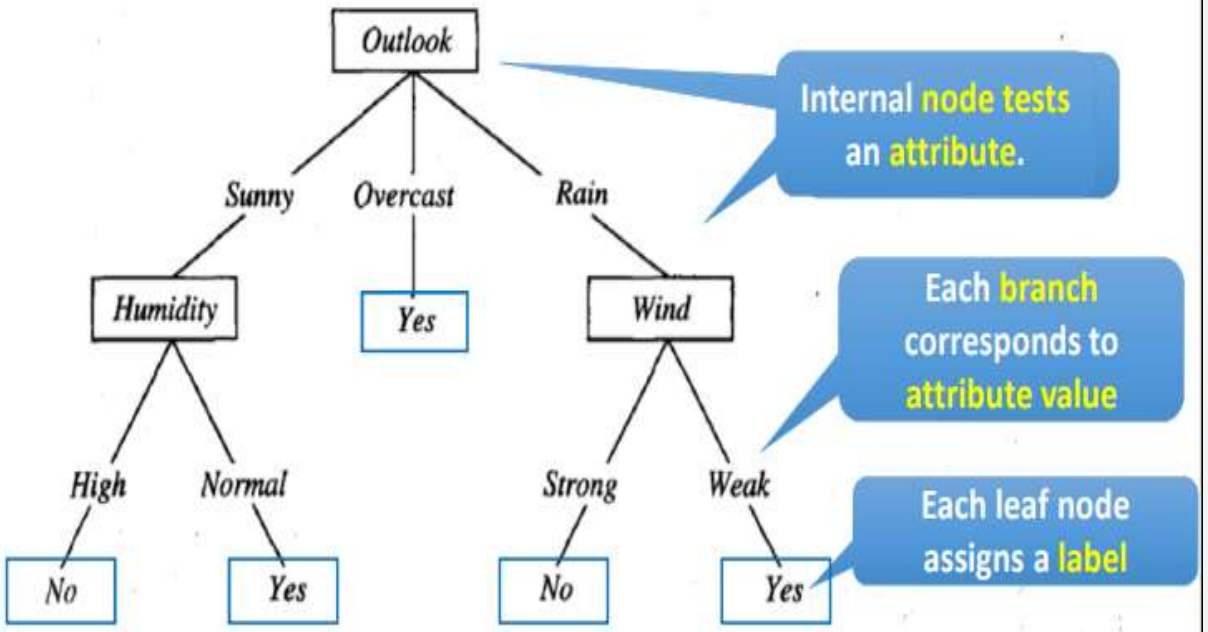
➢Based on the learnt decision tree, the instance is classified to be *negative instance* (i.e., the tree predicts that ***PlayTennis = no***).

➢In general, decision trees represent a disjunction of conjunctions of constraints on the attribute values of instances. **Each path** from the tree root to a leaf corresponds to a **conjunction of attribute** tests, and the **tree itself to a disjunction of these conjunctions.**

➢For example, the decision tree shown here corresponds to the below expression.

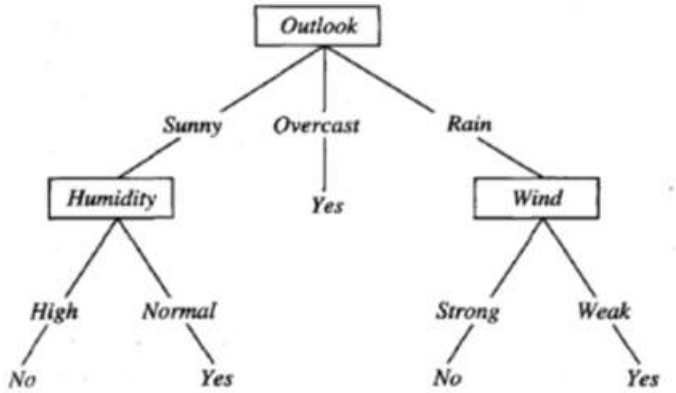
(Outlook = Sunny∧ Humidity = Normal) V (Outlook = Overcast) V (Outlook = Rain ∧ Wind = Weak)

DECISION TREE REPRESENTATION



$\langle Outlook = Sunny,\ Temperature = Hot,\ Humidity = High,\ Wind = Strong \rangle$

- It represent a disjunction of conjunctions of constraints on the attribute values of instances.



$$(Outlook = Sunny \wedge Humidity = Normal)$$
$$\vee \quad (Outlook = Overcast)$$
$$\vee \quad (Outlook = Rain \wedge Wind = Weak)$$

OVERVIEW

- Introduction
- Decision Tree Representation
- **Appropriate Problems for Decision Tree Learning**
- Basic Decision Tree Learning Algorithm (ID3)
- Hypothesis Space Search in decision Tree Learning
- Inductive Bias in Decision Tree Learning
- Issues in Decision Tree Learning

APPROPRIATE PROBLEMS FOR DECISION TREE LEARNING

Decision tree learning is generally best suited to problems with the following characteristics:

- **Instances are represented by attribute-value pairs.**
  - Instances are described by a fixed set of attributes (e.g., Temperature) and their values (e.g., Hot).
  - The easiest situation for decision tree learning is when each attribute takes on a small number of disjoint possible values (e.g., Hot, Mild, Cold).
- **The target function has discrete output values.**
  - The decision tree assigns a Boolean classification (e.g., yes or no) to each example.
  - Decision tree methods easily extend to learning functions with more than two possible output values.
- **Disjunctive descriptions may be required.**
  - As noted above, decision trees naturally represent disjunctive expressions.

APPROPRIATE PROBLEMS FOR DECISION TREE LEARNING

- **The training data may contain errors.**
  - Decision tree learning methods are robust to errors (error in attributes / error in targets)
- **The training data may contain missing attribute values.**
  - Decision tree methods can be used even when some training examples have unknown values (e.g., if the Humidity of the day is known for only some of the training examples).

APPROPRIATE PROBLEMS FOR DECISION TREE LEARNING

- Many practical problems have been found to fit these characteristics.
  - ***Medical diagnosis***
  - ***Equipment classification***
  - ***Credit risk analysis***
  - ***Several tasks in natural language processing***
- These problems, in which the task is to classify examples into one of a discrete set of possible categories, are often referred to as **classification problems**.

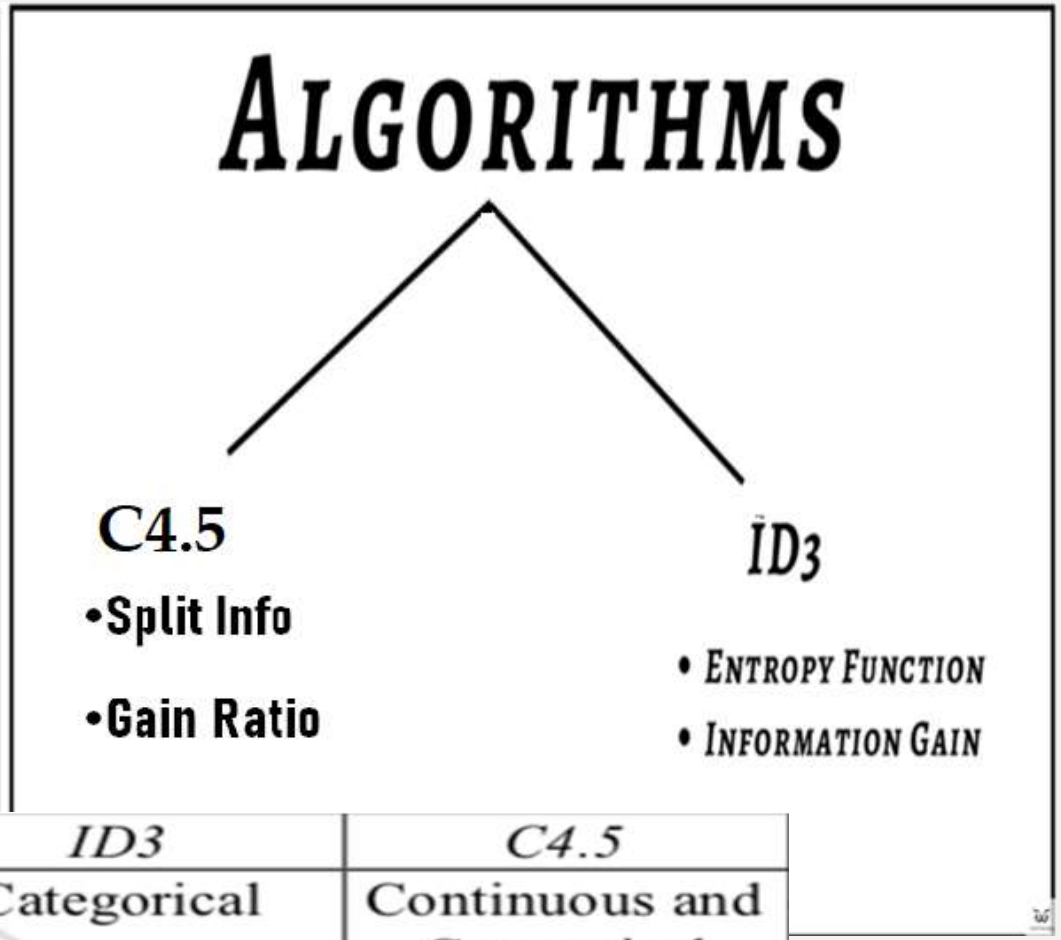
OVERVIEW

- Introduction
- Decision Tree Representation
- Appropriate Problems for Decision Tree Learning
- **Basic Decision Tree Learning Algorithm (ID3)**
- Hypothesis Space Search in decision Tree Learning
- Inductive Bias in Decision Tree Learning
- Issues in Decision Tree Learning

DECISION TREE LEARNING ALGORITHM

Inventor

- **John Ross Quinlan**
- He is a computer science researcher in data mining and decision theory.
- He has contributed extensively to the development of decision tree algorithms, including inventing the ID3 & canonical C4.5 algorithms.



Features	*ID3*	*C4.5*
Type of data	Categorical	Continuous and Categorical
Speed	Low	Faster than ID3
Formula	Use information entropy and information Gain	Use split info and gain ratio

Different Decision Tree Algorithms

1. ID3 (Iterative Dichotomiser 3)
2. C4.5 (successor of ID3)
3. CART (Classification And Regression Tree)
4. Chi-square automatic interaction detection (CHAID).
5. MARS: extends decision trees to handle numerical data better.

BASIC DECISION TREE LEARNING ALGORITHM (ID3)

- The core algorithm that employs a **top-down, greedy search** through the space of possible decision trees.
- **This approach is demonstrated by the ID3 algorithm (ID3-Iterative Dichotomiser 3)**
- ID3 basic algorithm, learns decision trees by constructing them top-down, beginning with the question "**which attribute should be tested at the root of the tree?**" .
- To answer this question,
  - Each instance attribute is evaluated using a statistical test to determine how well it alone classifies the training examples.
  - The best attribute is selected and used as the test at the root node of the tree.
  - A descendant of the root node is then created for each possible value of this attribute, and the training examples are sorted to the appropriate descendant node (i.e., down the branch corresponding to the example's value for this attribute).
  - The entire process is then repeated using the training examples associated with each descendant node to select the best attribute to test at that point in the tree.
  - This forms a greedy search for an acceptable decision tree, in which the algorithm never backtracks to reconsider earlier choices.

BASIC DECISION TREE LEARNING ALGORITHM

- A simplified version of the algorithm, specialized to learning boolean-valued functions (i.e., concept learning), is described in below.

ID3(*Examples*, *Target_attribute*, *Attributes*)

Examples are the training examples. Target_attribute is the attribute whose value is to be predicted by the tree. Attributes is a list of other attributes that may be tested by the learned decision tree. Returns a decision tree that correctly classifies the given Examples.

- Create a *Root* node for the tree
- If all *Examples* are positive, Return the single-node tree *Root*, with label = +
- If all *Examples* are negative, Return the single-node tree *Root*, with label = −
- If *Attributes* is empty, Return the single-node tree *Root*, with label = most common value of *Target_attribute* in *Examples*
- Otherwise Begin
    - $A \leftarrow$ the attribute from *Attributes* that best* classifies *Examples*
    - The decision attribute for *Root* $\leftarrow A$
    - For each possible value, v_i, of A,
        - Add a new tree branch below *Root*, corresponding to the test $A = v_i$
        - Let $Examples_{v_i}$ be the subset of *Examples* that have value v_i for A
        - If $Examples_{v_i}$ is empty
            - Then below this new branch add a leaf node with label = most common value of *Target_attribute* in *Examples*
            - Else below this new branch add the subtree ID3($Examples_{v_i}$, *Target_attribute*, *Attributes* − {*A*}))
- End
- Return *Root*

WHICH ATTRIBUTE IS THE BEST CLASSIFIER

- The central choice in the ID3 algorithm is selecting attribute that is most useful for classifying examples.

- We will define a statistical property, called ***information gain***, that measures how well a given attribute separates the training examples according to their target classification.

- ID3 uses this ***information gain*** measure to select among the candidate attributes at each step while growing the tree.

- ***ID3 algorithm, stands for Iterative Dichotomiser 3, is a classification algorithm that follows a greedy approach of building a decision tree by selecting a best attribute that yields maximum Information Gain (IG) or minimum Entropy (H).***

- ID3 stands for **Iterative Dichotomiser 3** and is **named** such because the algorithm ***iteratively (repeatedly) dichotomises(divides)*** features into two or more groups at each step.

DEFINITION: ENTROPY

- It is a Measurement of Homogeneity of Examples
- Given a collection S, containing +ve and -ve examples of some target concept, the entropy of S is given by

$$Entropy(S) \equiv -p_{\oplus} \log_2 p_{\oplus} - p_{\ominus} \log_2 p_{\ominus}$$

- Where $\mathbf{p_+}$ is the proportion of positive examples in S and $\mathbf{p_-}$ is the proportion of negative examples in S
- In all calculations involving entropy we define **0.log 0 = 0.**
- In general for c class classification

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

$$Entropy = \frac{-p}{p+n} log_2(\frac{p}{p+n}) - \frac{n}{p+n} log_2(\frac{n}{p+n})$$

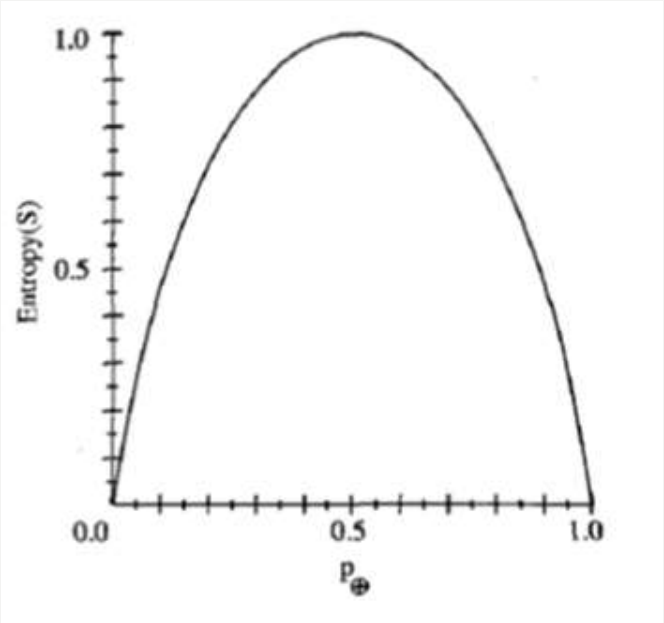
ENTROPY - ILLUSTRATION

- Let S is a collection of 14 examples of some boolean concept
- Let 9 positive and 5 negative examples [9+, 5-]
- Then the entropy of S relative to this Boolean classification is

$$Entropy(9+,5-) = -(9/14)\log_2(9/14)-(5/14)\log_2(5/14)$$

$$= 0.940$$

- Entropy function relative to a Boolean classification, as p_+, varies between 0 and 1



DEFINITION: INFORMATION GAIN

- It is the expected reduction in entropy caused by partitioning the examples according to some attribute A
- Split the node with attribute having highest Gain

- Calculate **Entropy** (Amount of uncertainity in dataset):

$$Entropy = \frac{-p}{p+n} log_2(\frac{p}{p+n}) - \frac{n}{p+n} log_2(\frac{n}{p+n})$$

- Calculate **Average Information**:

$$I(Attribute) = \sum \frac{p_i + n_i}{p+n} Entropy(A)$$

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

$$Gain = Entropy(S) - I(Attribute)$$

INFORMATION GAIN-ILLUSTRATION

- Calculate **Entropy** (Amount of uncertainity in dataset):

$$Entropy = \frac{-p}{p+n} log_2(\frac{p}{p+n}) - \frac{n}{p+n} log_2(\frac{n}{p+n})$$

- Calculate **Average Information**:

$$I(Attribute) = \sum \frac{p_i + n_i}{p+n} Entropy(A)$$

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

$$Gain = Entropy(S) - I(Attribute)$$

- Values (Wind) = Weak, Strong
- S = [9+,5-]
- S_{weak} = [6+,2-]
- S_{strong} = [3+, 3 −]

$Entropy(S) = -(9/14)log_2(9/14)-(5/14)log_2(5/14) = 0.940$

$Entropy(S_{weak}) = -(6/8)log_2(6/8)-(2/8)log_2(2/8) = 0.811$

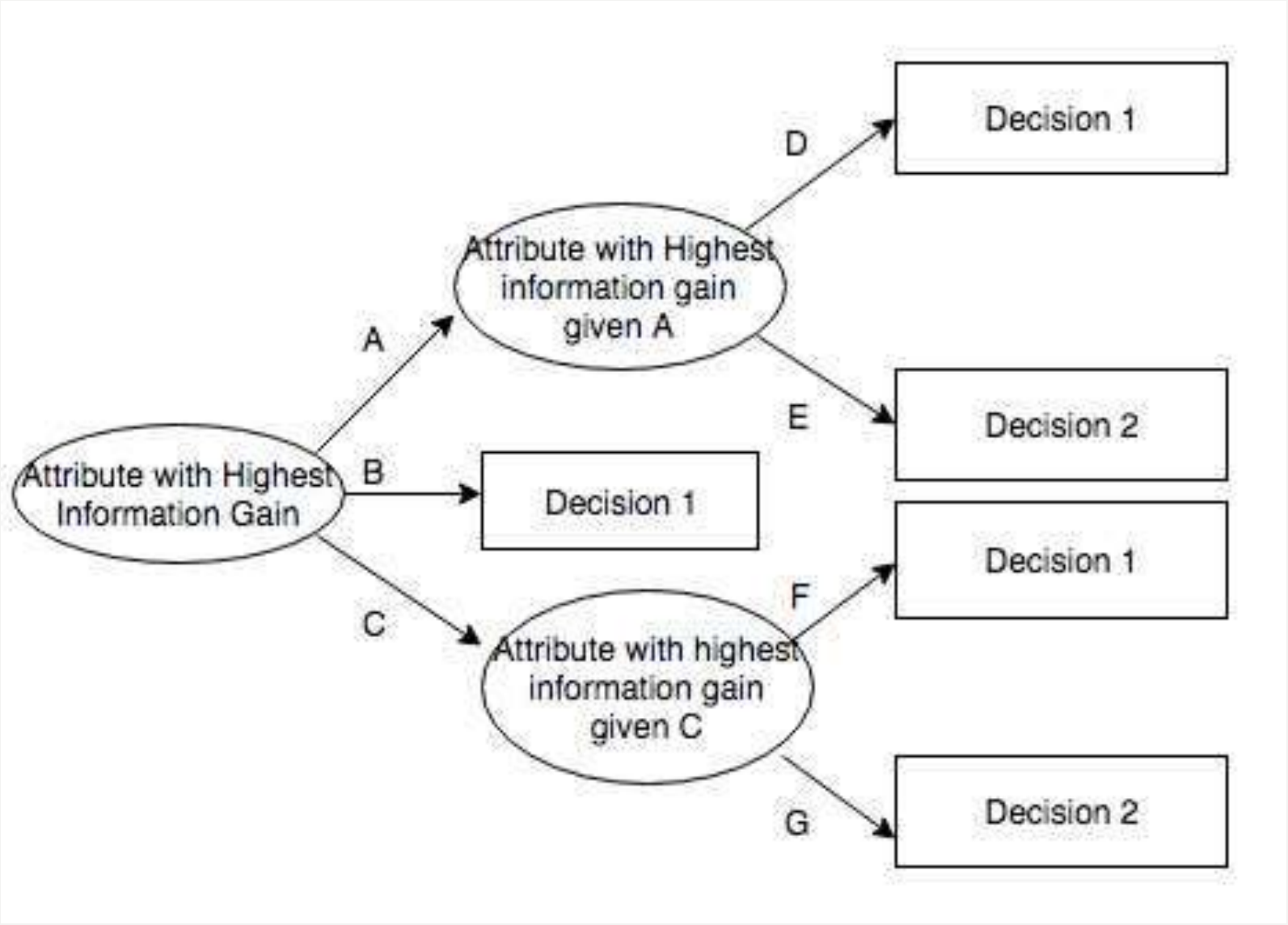
$Entropy(S_{strong}) = -(3/6)log_2(3/6) -(3/6)log_2(3/6)= 1$

$$I(Wind) = \frac{p_{weak}+n_{weak}}{p+n} Entropy(S_{weak}) + \frac{p_{strong}+n_{strong}}{p+n} Entropy(S_{strong})$$

$= 8/14(0.811)+6/14(1)= 0.892$

$Gain = Entropy(S) - I(Wind) = 0.94 – 0.892=0.048$

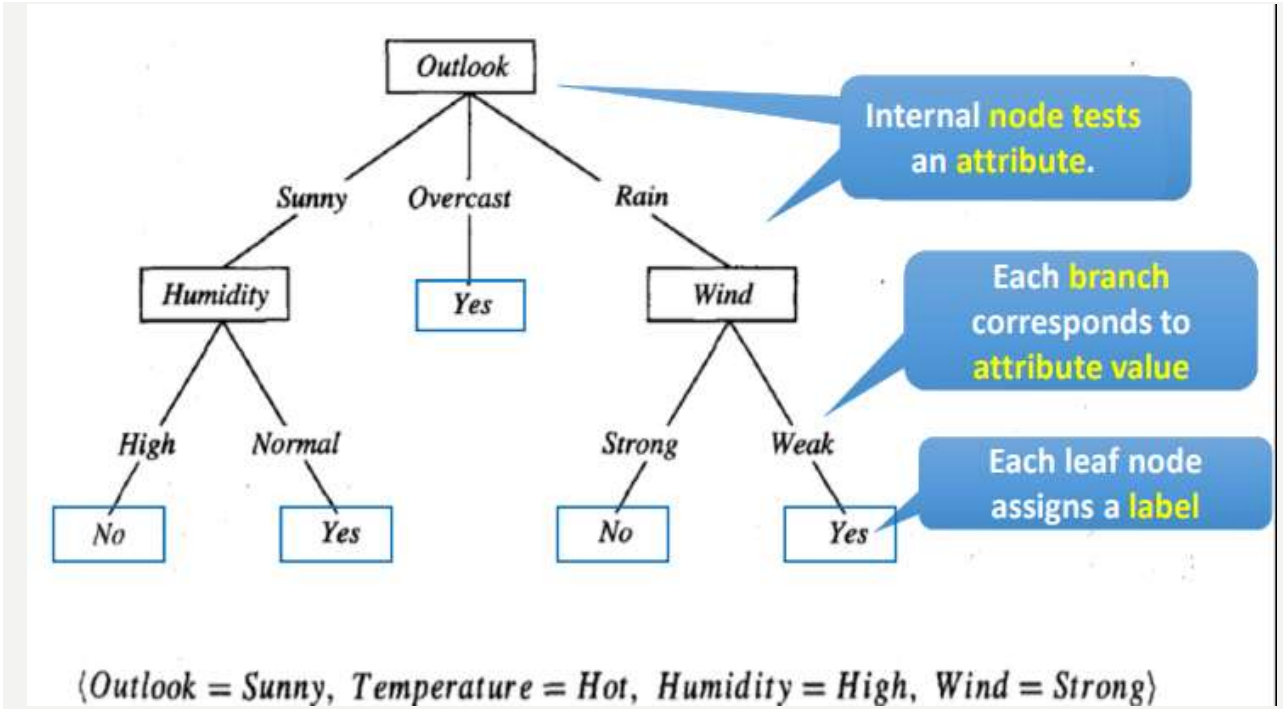
ID3 USING INFORMATION GAIN



SO FAR..

"A decision tree is a graphical representation of all the possible solutions to a decision based on certain conditions"

A **DECISION TREE** IS A TREE WHERE *EACH NODE* REPRESENTS A **FEATURE (ATTRIBUTE)**, EACH LINK (BRANCH) REPRESENTS A **DECISION (RULE)** AND EACH LEAF REPRESENTS AN **OUTCOME**.



- It represent a disjunction of conjunctions of constraints on the attribute values of instances.

(Outlook = Sunny ∧ Humidity = Normal) ∨ (Outlook = Overcast) ∨ (Outlook = Rain ∧ Wind = Weak)

SO FAR..

Appropriate Problems For Decision Tree Learning

- Instances are represented by attribute-value pairs.
- The target function has discrete output values.
- Disjunctive descriptions may be required.
- The training data may contain errors.
- The training data may contain missing attribute values.

SO FAR..

ID3 Algorithm

- ***John Ross Quinlan***

- *ID3 algorithm, stands for Iterative Dichotomiser 3, is a classification algorithm that follows a greedy approach of building a decision tree by selecting a best attribute that yields* ***maximum Information Gain (IG).***

- ID3 stands for **Iterative Dichotomiser 3** and is **named** such because the algorithm ***iteratively (repeatedly) dichotomises (divides)*** features into two or more groups at each step.

SO FAR..

ID3 Algorithm

- Calculate **Entropy** (Amount of uncertainity in dataset):

$$Entropy = \frac{-p}{p+n} log_2(\frac{p}{p+n}) - \frac{n}{p+n} log_2(\frac{n}{p+n})$$

- Calculate **Average Information**:

$$I(Attribute) = \sum \frac{p_i + n_i}{p+n} Entropy(A)$$

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

$$Gain = Entropy(S) - I(Attribute)$$

EXAMPLE

S. No.	Outlook	Temperature	Humidity	Windy	PlayTennis
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rainy	Mild	High	Weak	Yes
5	Rainy	Cool	Normal	Weak	Yes
6	Rainy	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rainy	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rainy	Mild	High	Strong	No

MAKE A DECISION TREE THAT PREDICTS WHETHER TENNIS WILL BE PLAYED ON THE DAY?

Step 1: 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.

Step 1: 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** (Amount of uncertainity in dataset):

$$Entropy = \frac{-p}{p+n} log_2(\frac{p}{p+n}) - \frac{n}{p+n} log_2(\frac{n}{p+n})$$

- Calculate **Average Information**:

$$I(Attribute) = \sum \frac{p_i + n_i}{p+n} Entropy(A)$$

- 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.

1.COMPUTE THE **ENTROPY** FOR DATA-SET **ENTROPY(S)**

Day	Outlook	Temperature	Humidity	Wind	Answer
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

$$Entropy(S) = -\frac{9}{14}\log_2\frac{9}{14} - \frac{5}{14}log_2\frac{5}{14} = 0.94$$

$$Entropy = \frac{-p}{p+n}log_2(\frac{p}{p+n}) - \frac{n}{p+n}log_2(\frac{n}{p+n})$$

S=(9+,5-)

$$Entropy(S) = -\frac{9}{14}\log_2\frac{9}{14} - \frac{5}{14}log_2\frac{5}{14} = 0.94$$

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

Attribute: Outlook

Values(Outlook)=*'Sunny', 'Overcast', 'Rain'*

Day	Outlook	Temperature	Humidity	Wind	Answer
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Day	Outlook	Answer
D1	Sunny	No
D2	Sunny	No
D8	Sunny	No
D9	Sunny	Yes
D11	Sunny	Yes

Day	Outlook	Answer
D3	Overcast	Yes
D7	Overcast	Yes
D12	Overcast	Yes
D13	Overcast	Yes

Day	Outlook	Answer
D4	Rain	Yes
D5	Rain	Yes
D6	Rain	No
D10	Rain	Yes
D14	Rain	No

$$Entropy(S) = -\frac{9}{14}\log_2\frac{9}{14} - \frac{5}{14}log_2\frac{5}{14} = 0.94$$

$S_{Sunny} \leftarrow (2+, 3-)$ $\quad Entropy\ (S_{Sunny}) = -\frac{2}{5}log_2\frac{2}{5} - \frac{3}{5}log_2\frac{3}{5} = 0.971$

$S_{Overcast} \leftarrow (4+, 0-)$ $\quad Entropy\ (S_{Overcast}) = -\frac{4}{4}log_2\frac{4}{4} - \frac{0}{4}log_2\frac{0}{4} = 0$

$S_{Rain} \leftarrow (3+, 2-)$ $\quad Entropy\ (S_{Rain}) = -\frac{3}{5}log_2\frac{3}{5} - \frac{2}{5}log_2\frac{2}{5} = 0.971$

$$I\ (Outlook) = \frac{p_{Sunny}+n_{Sunny}}{p+n}Entropy(S_{Sunny}) + \frac{p_{Overcast}+n_{Overcast}}{p+n}Entropy(S_{Overcast}) + \frac{p_{Rain}+n_{Rain}}{p+n}Entropy(S_{Rain})$$

$$I\ (Outlook) = \frac{2+3}{14} \times 0.971 + \frac{4+0}{14} \times 0 + \frac{3+2}{14} \times 0.971 = 0.693$$

$$Gain(S, Outlook) = Entropy(S) - I(Outlook) = 0.94 - 0.693 = 0.247$$

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

Attribute: Temperature

Values(Outlook)=*'Hot', 'Cool', 'Mild'*

Day	Outlook	Temperature	Humidity	Wind	Answer
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Day	Temperature	Answer
D1	Hot	No
D2	Hot	No
D3	Hot	Yes
D13	Hot	Yes

Day	Temperature	Answer
D5	Cool	Yes
D6	Cool	No
D7	Cool	Yes
D9	Cool	Yes

Day	Temperature	Answer
D4	Mild	Yes
D8	Mild	No
D10	Mild	Yes
D11	Mild	Yes
D12	Mild	Yes
D14	Mild	No

$$Entropy(S) = -\frac{9}{14}\log_2\frac{9}{14} - \frac{5}{14}log_2\frac{5}{14} = 0.94$$

$S_{Hot} \leftarrow (2+, 2-)$ $\quad Entropy\ (S_{Hot}) = -\frac{2}{4}log_2\frac{2}{4} - \frac{2}{4}log_2\frac{2}{4} = 1.0$

$S_{Cool} \leftarrow (3+, 1-)$ $\quad Entropy\ (S_{Cool}) = -\frac{3}{4}log_2\frac{3}{4} - \frac{1}{4}log_2\frac{1}{4} = 0.8113$

$S_{Mild} \leftarrow (4+, 2-)$ $\quad Entropy\ (S_{Mild}) = -\frac{4}{6}log_2\frac{4}{6} - \frac{2}{6}log_2\frac{2}{6} = 0.9183$

$$I\,(Sunny, Temperature) = \frac{p_{Hot}+n_{Hot}}{p+n}Entropy(S_{Hot}) + \frac{p_{Cool}+n_{Cool}}{p+n}Entropy(S_{Cool}) + \frac{p_{Mild}+n_{Mild}}{p+n}Entropy(S_{Mild})$$

$$I\,(Temperature) = \frac{2+2}{14} \times 1.0 + \frac{3+1}{14} \times 0.8113 + \frac{4+2}{14} \times 0.9183 = 0.9095$$

$$Gain(S, Temperature) = Entropy(S) - I(Temperature) = 0.94 - 0.9095 = 0.03$$

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

Attribute: Humidity

Values(Outlook)=*'High', 'Normal'*

Day	Outlook	Temperature	Humidity	Wind	Answer
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Day	Humidity	Answer
D1	High	No
D2	High	No
D3	High	Yes
D4	High	Yes
D8	High	No
D12	High	Yes
D14	High	No

Day	Humidity	Answer
D5	Normal	Yes
D6	Normal	No
D7	Normal	Yes
D9	Normal	Yes
D10	Normal	Yes
D11	Normal	Yes
D13	Normal	Yes

$$Entropy(S) = -\frac{9}{14}\log_2\frac{9}{14} - \frac{5}{14}log_2\frac{5}{14} = 0.94$$

$$S_{High} \leftarrow (3+, 4-) \qquad Entropy\ (S_{High}) = -\frac{3}{7}log_2\frac{3}{7} - \frac{4}{7}log_2\frac{4}{7} = 0.9852$$

$$S_{Normal} \leftarrow (6+, 1-) \qquad Entropy\ (S_{Normal}) = -\frac{6}{7}log_2\frac{6}{7} - \frac{1}{7}log_2\frac{1}{7} = 0.5916$$

$$I\ (Humidity) = \frac{p_{High}+n_{High}}{p+n}Entropy(S_{High}) + \frac{p_{Normal}+n_{Normal}}{p+n}Entropy(S_{Normal})$$

$$I\ (Humidity) = \frac{3+4}{14} \times 0.9852 + \frac{6+1}{14} \times 0.5916 = 0.7884$$

$$Gain(S, Humidity) = Entropy(S) - I(Humidity) = 0.94 - 0.7884 = 0.1516$$

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

Attribute: Wind

Values(Outlook)=*'Weak', 'Strong'*

$$Entropy(S) = -\frac{9}{14}\log_2\frac{9}{14} - \frac{5}{14}log_2\frac{5}{14} = 0.94$$

Day	Outlook	Temperature	Humidity	Wind	Answer
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Day	Wind	Answer
D1	Weak	No
D3	Weak	Yes
D4	Weak	Yes
D5	Weak	Yes
D8	Weak	No
D9	Weak	Yes
D10	Weak	Yes
D13	Weak	Yes

Day	Wind	Answer
D2	Strong	No
D6	Strong	No
D7	Strong	Yes
D11	Strong	Yes
D12	Strong	Yes
D14	Strong	No

$$S_{Weak} \leftarrow (6+, 2-) \qquad Entropy\ (S_{Weak}) = -\frac{6}{8}log_2\frac{6}{8} - \frac{2}{8}log_2\frac{2}{8} = 0.8113$$

$$S_{Strong} \leftarrow (3+, 3-) \qquad Entropy\ (S_{Strong}) = -\frac{3}{6}log_2\frac{3}{6} - \frac{3}{6}log_2\frac{3}{6} = 1$$

$$I\ (Wind) = \frac{p_{Weak}+n_{Weak}}{p+n} Entropy(S_{Weak}) + \frac{p_{Strong}+n_{Strong}}{p+n} Entropy(S_{Strong})$$

$$I\ (Wind) = \frac{6+2}{14} \times 0.8113 + \frac{3+3}{14} \times 1 = 0.892$$

$$Gain(S, Wind) = Entropy(S) - I(Wind) = 0.94 - 0.892 = 0.0478$$

3. PICK THE **HIGHEST GAIN ATTRIBUTE.**

Day	Outlook	Temperature	Humidity	Wind	Answer
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

$Gain(S, Outlook) = 0.247$

$Gain(S, Temperature) = 0.03$

$Gain(S, Humidity) = 0.1516$

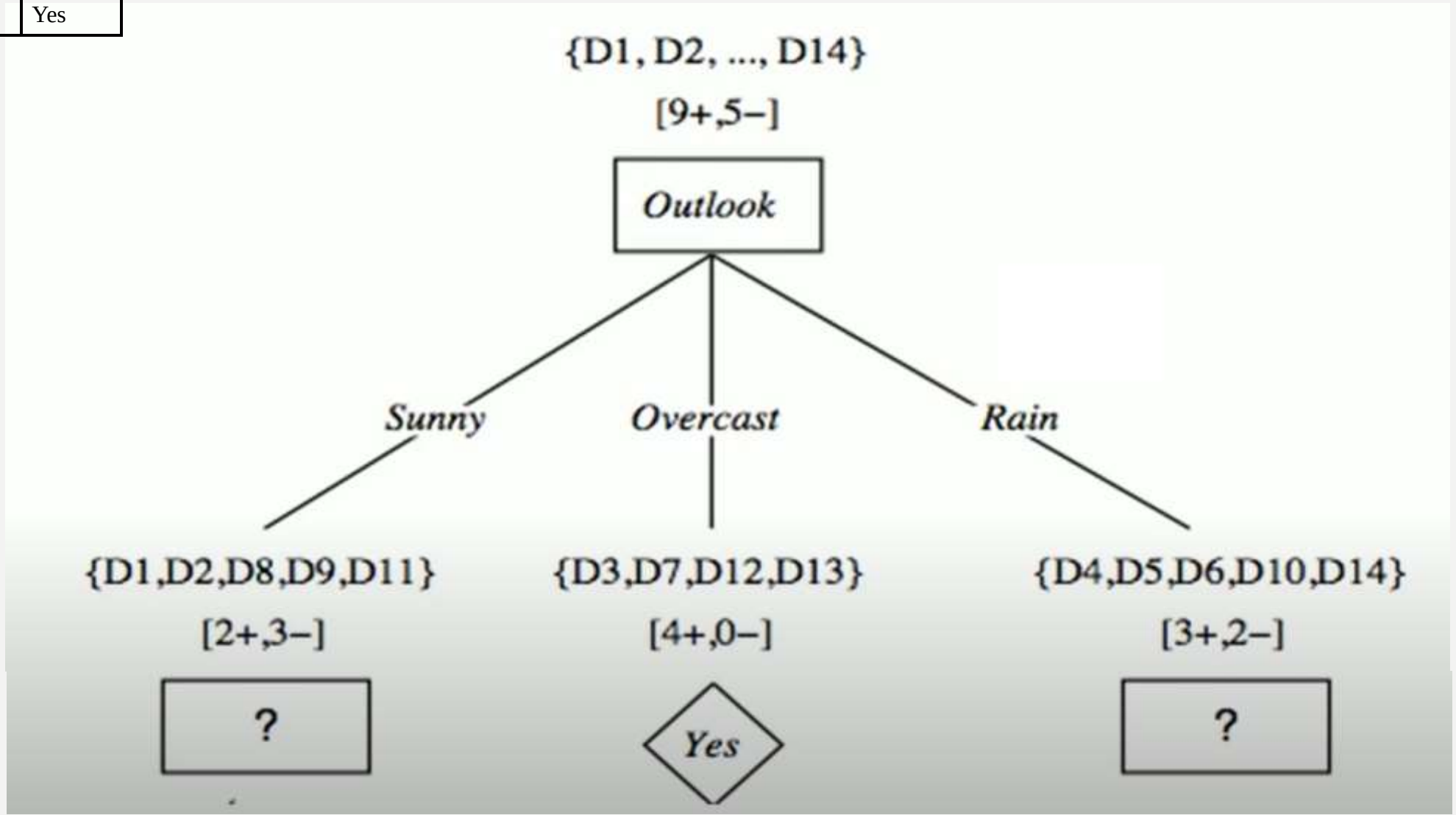
$Gain(S, Wind) = 0.0478$

Outlook is selected as the root note

Day	Outlook	Answer
D1	Sunny	No
D2	Sunny	No
D8	Sunny	No
D9	Sunny	Yes
D11	Sunny	Yes

Day	Outlook	Answer
D3	Overcast	Yes
D7	Overcast	Yes
D12	Overcast	Yes
D13	Overcast	Yes

Day	Outlook	Answer
D4	Rain	Yes
D5	Rain	Yes
D6	Rain	No
D10	Rain	Yes
D14	Rain	No



- Repeat the same thing for sub-trees till we get the tree.

Day	Outlook	Temperature	Humidity	Wind	Answer
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes

Outlook = "Sunny"

Day	Outlook	Temperature	Humidity	Wind	Answer
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D10	Rain	Mild	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Outlook = "Rain"

For Outlook=Sunny

Attribute: Temperature

Values(Temperature)='*Hot*', '*Cool*', '*Mild*'

Day	Outlook	Temperature	Humidity	Wind	Answer
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes

$$S_{Sunny} \leftarrow (2+, 3-) \quad Entropy\ (S_{Sunny}) = -\frac{2}{5}log_2\frac{2}{5} - \frac{3}{5}log_2\frac{3}{5} = 0.971$$

$$S_{Hot} \leftarrow (0+, 2-) \quad Entropy\ (S_{Hot}) = -\frac{0}{2}log_2\frac{0}{2} - \frac{2}{2}log_2\frac{2}{2} = 0$$

$$S_{Cool} \leftarrow (1+, 0-) \quad Entropy\ (S_{Cool}) = -\frac{1}{1}log_2\frac{1}{1} - \frac{0}{1}log_2\frac{0}{1} = 0$$

$$S_{Mild} \leftarrow (1+, 1-) \quad Entropy\ (S_{Mild}) = -\frac{1}{2}log_2\frac{1}{2} - \frac{1}{2}log_2\frac{1}{2} = 1$$

$$I\ (S_{Sunny}, Temperature) = \frac{p_{Hot}+n_{Hot}}{p+n} Entropy(S_{Hot}) + \frac{p_{Cool}+n_{Cool}}{p+n} Entropy(S_{Cool}) + \frac{p_{Mild}+n_{Mild}}{p+n} Entropy(S_{Mild})$$

Day	Outlook	Temperature	Answer
D1	Sunny	Hot	No
D2	Sunny	Hot	No

Day	Outlook	Temperature	Answer
D8	Sunny	Mild	No
D11	Sunny	Mild	Yes

$$I\ (S_{Sunny}, Temperature) = \frac{0+2}{5} \times 0 + \frac{1+0}{5} \times 0 + \frac{1+1}{5} \times 1 = 0.4$$

Day	Outlook	Temperature	Answer
D9	Sunny	Cool	Yes

$$Gain(S_{Sunny}, Temperature) = Entropy(S_{Sunny}) - I(S_{Sunny}, Temperature) = 0.971 - 0.4 = 0.57$$

For Outlook=Sunny
Attribute: Humidity
Values(Humidity)=*'High', 'Normal'*

Day	Outlook	Temperature	Humidity	Wind	Answer
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes

$$S_{Sunny} \leftarrow (2+, 3-) \quad Entropy\ (S_{Sunny}) = -\frac{2}{5}log_2\frac{2}{5} - \frac{3}{5}log_2\frac{3}{5} = 0.971$$

$$S_{High} \leftarrow (0+, 3-) \quad Entropy\ (S_{High}) = -\frac{0}{3}log_2\frac{0}{3} - \frac{3}{3}log_2\frac{3}{3} = 0$$

$$S_{Normal} \leftarrow (2+, 0-) \quad Entropy\ (S_{Normal}) = -\frac{2}{2}log_2\frac{2}{2} - \frac{0}{2}log_2\frac{0}{2} = 0$$

$$I\ (S_{Sunny}, Humidity) = \frac{p_{High}+n_{High}}{p+n} Entropy(S_{High}) + \frac{p_{Normal}+n_{Normal}}{p+n} Entropy(S_{Normal})$$

$$I\ (S_{Sunny}, Humidity) = \frac{3}{5} \times 0 + \frac{2}{5} \times 0 = 0$$

Day	Outlook	Humidity	Answer
D1	Sunny	High	No
D2	Sunny	High	No
D8	Sunny	High	No

$$Gain(S_{Sunny}, Humidity) = Entropy(S_{Sunny}) - I(S_{Sunny}, Humidity) = 0.971 - 0 = 0.971$$

Day	Outlook	Humidity	Answer
D9	Sunny	Normal	Yes
D11	Sunny	Normal	Yes

For Outlook=Sunny
Attribute: Wind
Values(Wind)=*'Weak', 'Strong'*

Day	Outlook	Temperature	Humidity	Wind	Answer
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes

$$S_{Sunny} \leftarrow (2+, 3-) \quad Entropy\ (S_{Sunny}) = -\frac{2}{5}log_2\frac{2}{5} - \frac{3}{5}log_2\frac{3}{5} = 0.971$$

$$S_{Weak} \leftarrow (1+, 2-) \quad Entropy\ (S_{Weak}) = -\frac{1}{3}log_2\frac{1}{3} - \frac{2}{3}log_2\frac{2}{3} = 0.9183$$

$$S_{Strong} \leftarrow (1+, 1-) \quad Entropy\ (S_{Strong}) = -\frac{1}{2}log_2\frac{1}{2} - \frac{1}{2}log_2\frac{1}{2} = 1$$

$$I\ (S_{Sunny}, Wind) = \frac{p_{Weak}+n_{Weak}}{p+n} Entropy(S_{Weak}) + \frac{p_{Strong}+n_{Strong}}{p+n} Entropy(S_{Strong})$$

Day	Outlook	Wind	Answer
D1	Sunny	Weak	No
D8	Sunny	Weak	No
D9	Sunny	Weak	Yes

$$I\ (S_{Sunny}, Wind) = \frac{3}{5} \times 0.9183 + \frac{2}{5} \times 1 = 0.95$$

Day	Outlook	Wind	Answer
D2	Sunny	Strong	No
D11	Sunny	Strong	Yes

$$Gain(S_{Sunny}, Wind) = Entropy(S_{Sunny}) - I(S_{Sunny}, Wind) = 0.971 - 0.95 = 0.02$$

3. PICK THE **HIGHEST GAIN ATTRIBUTE.**

Day	Outlook	Temperature	Humidity	Wind	Answer
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes

For Outlook=Sunny

$Gain(S_{Sunny}, Temperature) = 0.57$

$Gain(S_{Sunny}, Humidity) = 0.971$

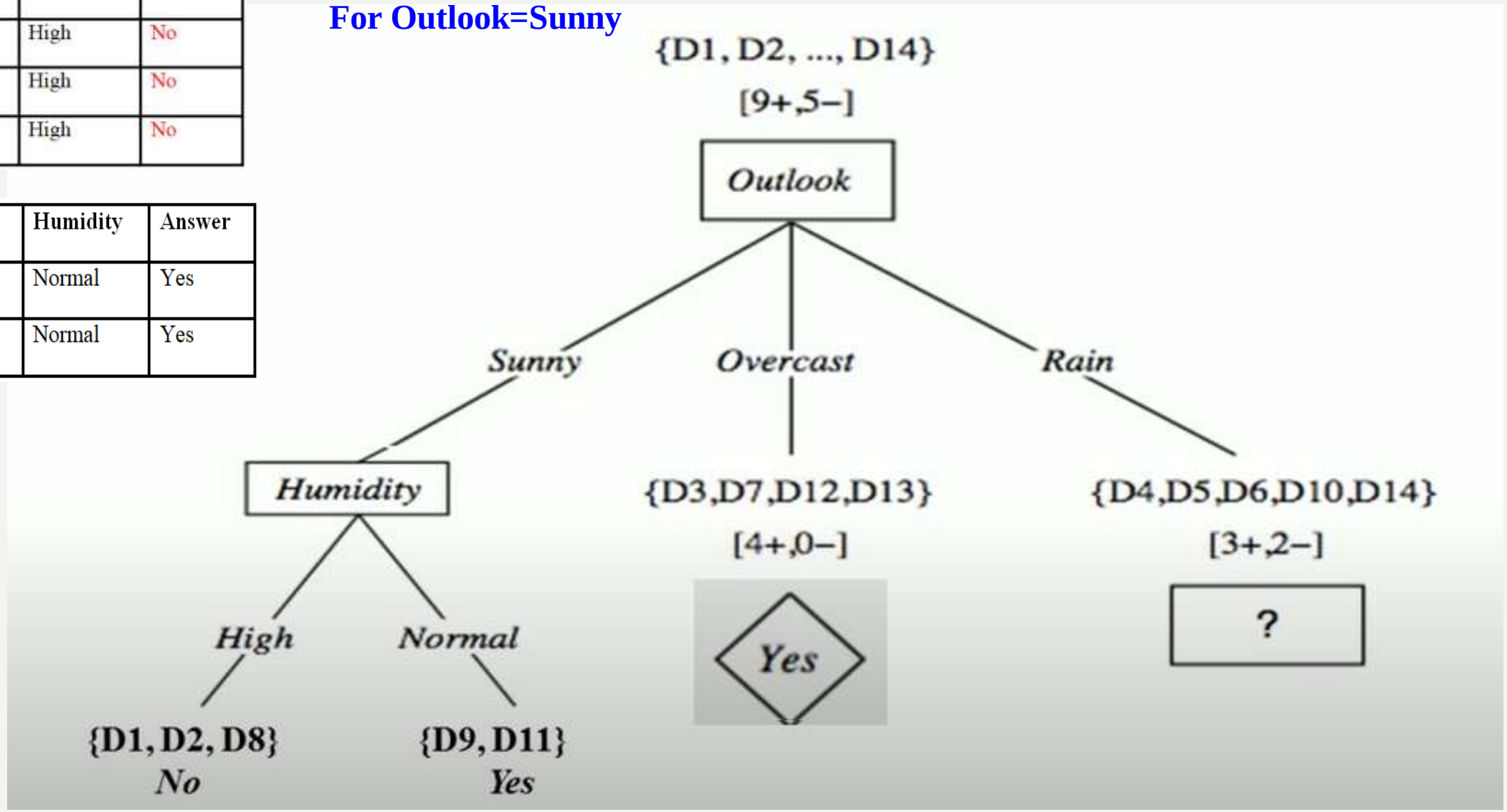
$Gain(S_{Sunny}, Wind) = 0.02$

Humidity is Selected

Day	Outlook	Humidity	Answer
D1	Sunny	High	No
D2	Sunny	High	No
D8	Sunny	High	No

Day	Outlook	Humidity	Answer
D9	Sunny	Normal	Yes
D11	Sunny	Normal	Yes

For Outlook=Sunny



For Outlook=Rain

Attribute: Temperature

Values(Temperature)=*'Hot', 'Cool', 'Mild'*

Day	Outlook	Temperature	Humidity	Wind	Answer
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D10	Rain	Mild	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

$S_{Rain} \leftarrow (3+, 2-)$ $\quad Entropy\ (S_{Rain}) = -\frac{3}{5}log_2\frac{3}{5} - \frac{2}{5}log_2\frac{2}{5} = 0.971$

$S_{Cool} \leftarrow (1+, 1-)$ $\quad Entropy\ (S_{Cool}) = -\frac{1}{2}log_2\frac{1}{2} - \frac{1}{2}log_2\frac{1}{2} = 1$

$S_{Mild} \leftarrow (2+, 1-)$ $\quad Entropy\ (S_{Mild}) = -\frac{2}{3}log_2\frac{2}{3} - \frac{1}{3}log_2\frac{1}{3} = 0.9183$

$$I\ (S_{Rain}, Temperature) = \frac{p_{Cool}+n_{Cool}}{p+n} Entropy(S_{Cool}) + \frac{p_{Mild}+n_{Mild}}{p+n} Entropy(S_{Mild})$$

$$I\ (S_{Rain}, Temperature) = \frac{2}{5} \times 1 + \frac{3}{5} \times 0.9183 = 0.95$$

Day	Outlook	Temperature	Answer
D4	Rain	Mild	Yes
D10	Rain	Mild	Yes
D14	Rain	Mild	No

Day	Outlook	Temperature	Answer
D5	Rain	Cool	Yes
D6	Rain	Cool	No

$$Gain(S_{Rain}, Temperature) = Entropy(S_{Rain}) - I(S_{Rain}, Temperature) = 0.971 - 0.95 = 0.02$$

For Outlook=Rain

Attribute: Humidity

Values(Humidity)=*'High', 'Normal'*

Day	Outlook	Temperature	Humidity	Wind	Answer
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D10	Rain	Mild	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

$S_{Rain} \leftarrow (3+, 2-)$ $\quad Entropy\ (S_{Rain}) = -\frac{3}{5}log_2\frac{3}{5} - \frac{2}{5}log_2\frac{2}{5} = 0.971$

$S_{High} \leftarrow (1+, 1-)$ $\quad Entropy\ (S_{High}) = -\frac{1}{2}log_2\frac{1}{2} - \frac{1}{2}log_2\frac{1}{2} = 1$

$S_{Normal} \leftarrow (2+, 1-)$ $\quad Entropy\ (S_{Normal}) = -\frac{2}{3}log_2\frac{2}{3} - \frac{1}{3}log_2\frac{1}{3} = 0.9183$

$$I\ (S_{Rain}, Humidity) = \frac{p_{High}+n_{High}}{p+n} Entropy(S_{High}) + \frac{p_{Normal}+n_{Normal}}{p+n} Entropy(S_{Normal})$$

$$I\ (S_{Rain}\ , Humidity) = \frac{2}{5} \times 1 + \frac{3}{5} \times 0.9183 = 0.95$$

Day	Outlook	Humidity	Answer
D4	Rain	High	Yes
D14	Rain	High	No

Day	Outlook	Humidity	Answer
D5	Rain	Normal	Yes
D6	Rain	Normal	No
D10	Rain	Normal	Yes

$$Gain(S_{Rain}, Humidity) = Entropy(S_{Rain}) - I(S_{Rain}, Humidity) \quad = 0.971 - 0.95 = 0.02$$

For Outlook=Rain
Attribute: Wind
Values(Wind)=*'Weak', 'Strong'*

Day	Outlook	Temperature	Humidity	Wind	Answer
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D10	Rain	Mild	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Day	Outlook	Wind	Answer
D4	Rain	Weak	Yes
D5	Rain	Weak	Yes
D10	Rain	Weak	Yes

Day	Outlook	Wind	Answer
D6	Rain	Strong	No
D14	Rain	Strong	No

$S_{Rain} \leftarrow (3+, 2-)$ $\quad Entropy\ (S_{Rain}) = -\frac{3}{5} log_2 \frac{3}{5} - \frac{2}{5} log_2 \frac{2}{5} = 0.971$

$S_{Weak} \leftarrow (3+, 0-)$ $\quad Entropy\ (S_{Weak}) = -\frac{3}{3} log_2 \frac{3}{3} - \frac{0}{3} log_2 \frac{0}{3} = 0$

$S_{Strong} \leftarrow (0+, 2-)$ $\quad Entropy\ (S_{Strong}) = -\frac{0}{2} log_2 \frac{0}{2} - \frac{2}{2} log_2 \frac{2}{2} = 0$

$$I\ (S_{Rain}, Wind) = \frac{p_{Weak}+n_{Weak}}{p+n} Entropy(S_{Weak}) + \frac{p_{Strong}+n_{Strong}}{p+n} Entropy(S_{Strong})$$

$$I\ (S_{Rain}, Wind) = \frac{3}{5} \times 0 + \frac{2}{5} \times 0 = 0$$

$$Gain(S_{Rain}, Wind) = Entropy(S_{Rain}) - I(S_{Rain}, Wind) = 0.971 - 0 = 0.971$$

3. PICK THE **HIGHEST GAIN ATTRIBUTE.**

For Outlook=Rain

Day	Outlook	Temperature	Humidity	Wind	Answer
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D10	Rain	Mild	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

$Gain(S_{Rain}, Temperature) = 0.02$

$Gain(S_{Rain}, Humidity) = 0.02$

$Gain(S_{Rain}, Wind) = 0.971$

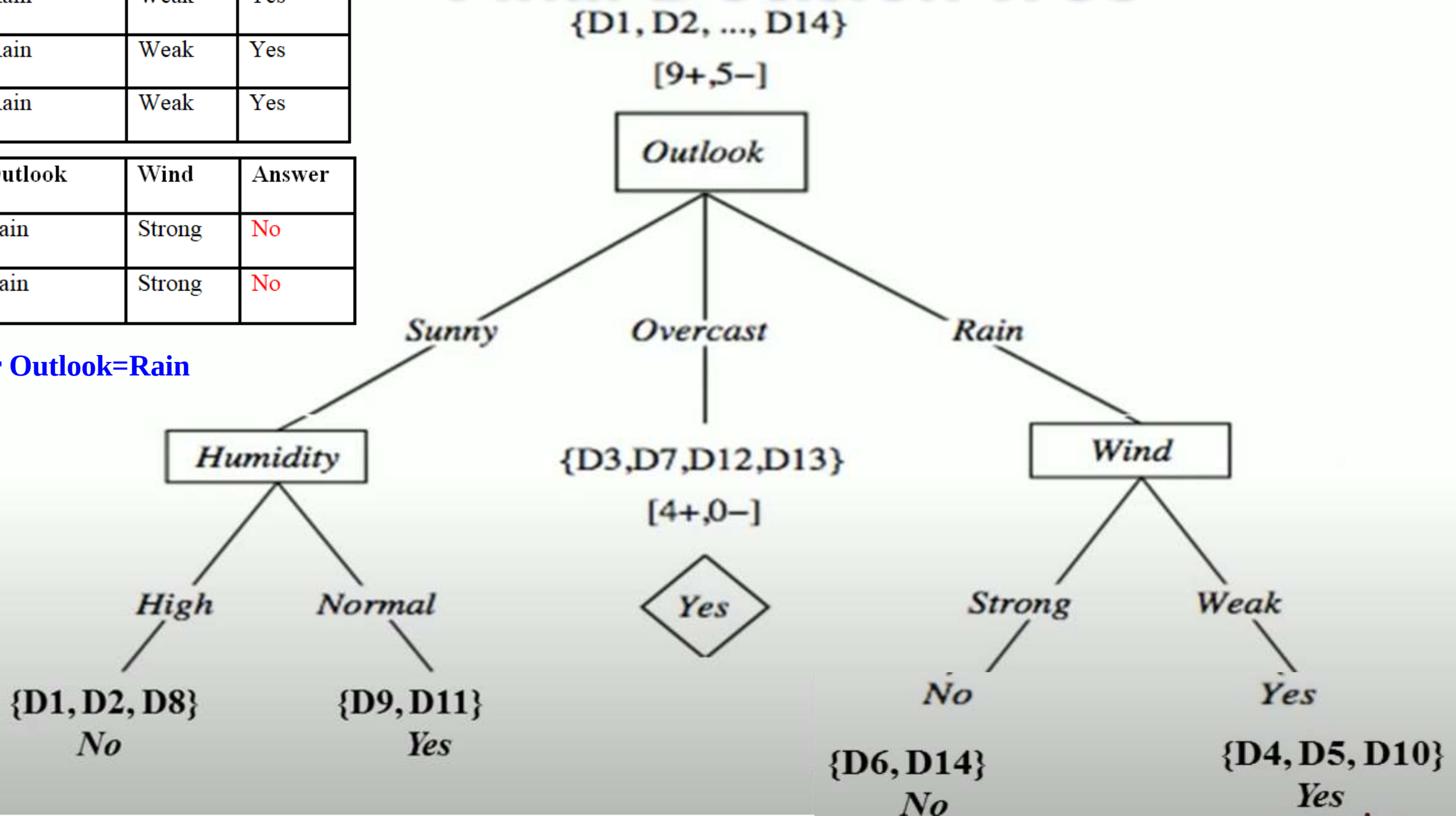
Wind is selected

Final Decision Tree

Day	Outlook	Wind	Answer
D4	Rain	Weak	Yes
D5	Rain	Weak	Yes
D10	Rain	Weak	Yes

Day	Outlook	Wind	Answer
D6	Rain	Strong	No
D14	Rain	Strong	No

For Outlook=Rain



Decision Tree Algorithm – ID3 Solved Example

1. What is the entropy of this collection of training examples with respect to the target function classification?
2. What is the information gain of *a1* and *a2* relative to these training examples?
3. Draw decision tree for the given dataset.

Instance	Classification	a1	a2
1	+	T	T
2	+	T	T
3	-	T	F
4	+	F	F
5	-	F	T
6	-	F	T

Attribute: a1

Values (a1): T, F

Instance	Classification	a1
1	+	T
2	+	T
3	-	T
4	+	F
5	-	F
6	-	F

$S \leftarrow (3+, 3-)$ $\quad Entropy\,(S) = -\frac{3}{6}log_2\frac{3}{6} - \frac{3}{6}log_2\frac{3}{6} = 1$

$S_T \leftarrow (2+, 1-)$ $\quad Entropy\,(S_T) = -\frac{2}{3}log_2\frac{2}{3} - \frac{1}{3}log_2\frac{1}{3} = 0.9183$

$S_F \leftarrow (1+, 2-)$ $\quad Entropy\,(S_F) = -\frac{1}{3}log_2\frac{1}{3} - \frac{2}{3}log_2\frac{2}{3} = 0.9183$

$$I\,(a1) = \frac{p_T+n_T}{p+n}Entropy(S_T) + \frac{p_F+n_F}{p+n}Entropy(S_F)$$

$$I\,(a1) = \frac{3}{6} \times 0.9183 + \frac{3}{6} \times 0.9183 = 0.9183$$

$$Gain(S, a1) = Entropy(S) - I(a1) = 1 - 0.9183 = 0.0817$$

Attribute:a2

Values(a2):T,F

Instance	Classification	a2
1	+	T
2	+	T
3	-	F
4	+	F
5	-	T
6	-	T

$S \leftarrow (3+, 3-)$ $\quad Entropy\ (S) = -\frac{3}{6}log_2\frac{3}{6} - \frac{3}{6}log_2\frac{3}{6} = 1$

$S_T \leftarrow (2+, 2-)$ $\quad Entropy\ (S_T) = -\frac{2}{4}log_2\frac{2}{4} - \frac{2}{4}log_2\frac{2}{4} = 1$

$S_F \leftarrow (1+, 1-)$ $\quad Entropy\ (S_F) = -\frac{1}{2}log_2\frac{1}{2} - \frac{1}{2}log_2\frac{1}{2} = 1$

$$I\ (a2) = \frac{p_T+n_T}{p+n} Entropy(S_T) + \frac{p_F+n_F}{p+n} Entropy(S_F)$$

$$I\ (a2) = \frac{4}{6} \times 1 + \frac{2}{6} \times 1 = 1$$

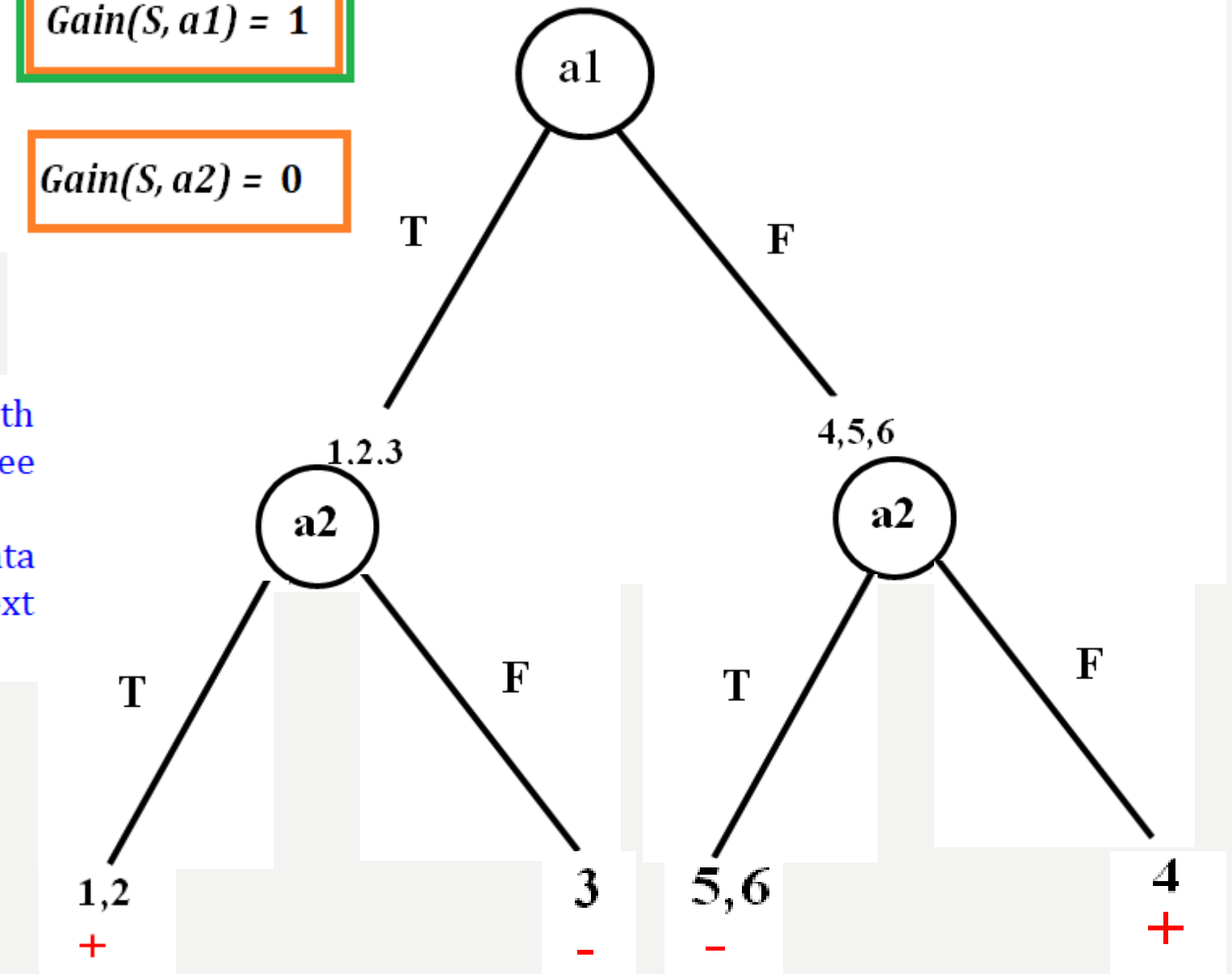
$$Gain(S, a2) = Entropy(S) - I(a2) = 1 - 1 = 0$$

Final Decision Tree

Instance	Classification	a1	a2
1	+	T	T
2	+	T	T
3	-	T	F
4	+	F	F
5	-	F	T
6	-	F	T

$Gain(S, a1) = 1$

$Gain(S, a2) = 0$



- Since a1 is the root, and this node with values T and F, the classification is impure, tree grows with next node.
- Since there exist only 2 attributes in the data set, a2 can be directly considered as the next node in the tree.

BASIC DECISION TREE LEARNING ALGORITHM

- A simplified version of the algorithm, specialized to learning boolean-valued functions (i.e., concept learning), is described in below.

ID3(*Examples*, *Target_attribute*, *Attributes*)

Examples are the training examples. Target_attribute is the attribute whose value is to be predicted by the tree. Attributes is a list of other attributes that may be tested by the learned decision tree. Returns a decision tree that correctly classifies the given Examples.

- Create a *Root* node for the tree
- If all *Examples* are positive, Return the single-node tree *Root*, with label = +
- If all *Examples* are negative, Return the single-node tree *Root*, with label = −
- If *Attributes* is empty, Return the single-node tree *Root*, with label = most common value of *Target_attribute* in *Examples*
- Otherwise Begin
    - $A \leftarrow$ the attribute from *Attributes* that best* classifies *Examples*
    - The decision attribute for *Root* $\leftarrow A$
    - For each possible value, v_i, of A,
        - Add a new tree branch below *Root*, corresponding to the test $A = v_i$
        - Let $Examples_{v_i}$ be the subset of *Examples* that have value v_i for A
        - If $Examples_{v_i}$ is empty
            - Then below this new branch add a leaf node with label = most common value of *Target_attribute* in *Examples*
            - Else below this new branch add the subtree ID3($Examples_{v_i}$, *Target_attribute*, *Attributes* − {*A*}))
- End
- Return *Root*

DECISION TREE FOR BOOLEAN FUNCTIONS.

(a) $A \wedge \neg B$

(b) $A \vee [B \wedge C]$

(c) A XOR B

(d) $[A \wedge B] \vee [C \wedge D]$

DECISION TREE FOR LOGICAL FUNCTIONS.

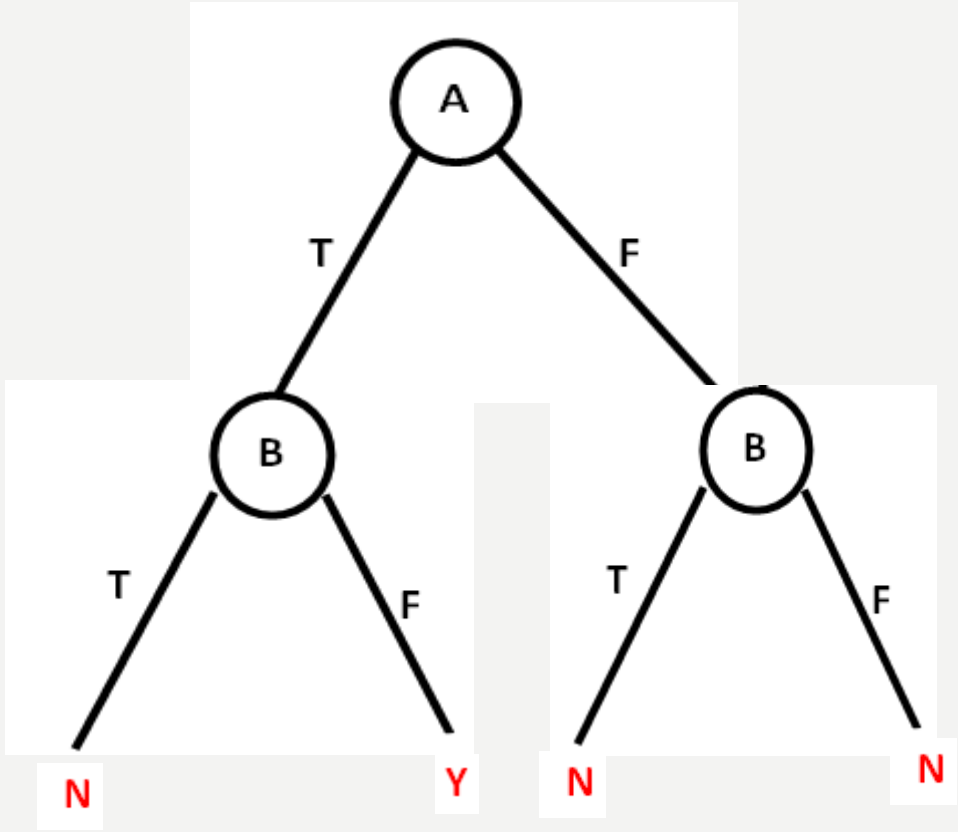
Solution:

- Every Variable in Boolean function such as A, B, C etc. has two possibilities that is True and False
- Every Boolean function is either True or False
- If the Boolean function is **True** we write YES **(Y)**
- If the Boolean function is **False** we write NO **(N)**

Decision Tree for Boolean Function

(a) $A \wedge \neg B$

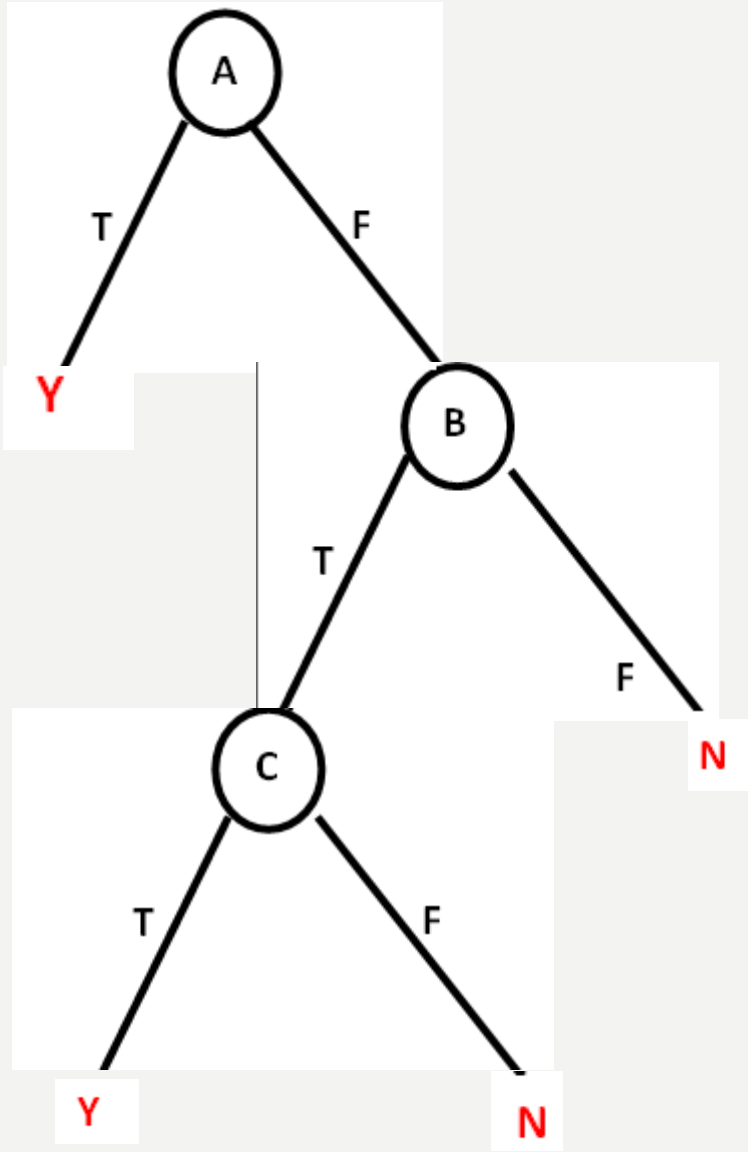
A	B	¬B	A ∧¬B
T	T	F	F→N
F	T	F	F→N
T	F	T	T→Y
F	F	T	F→N



Decision Tree for Boolean Function

(b) $A \vee [B \wedge C]$

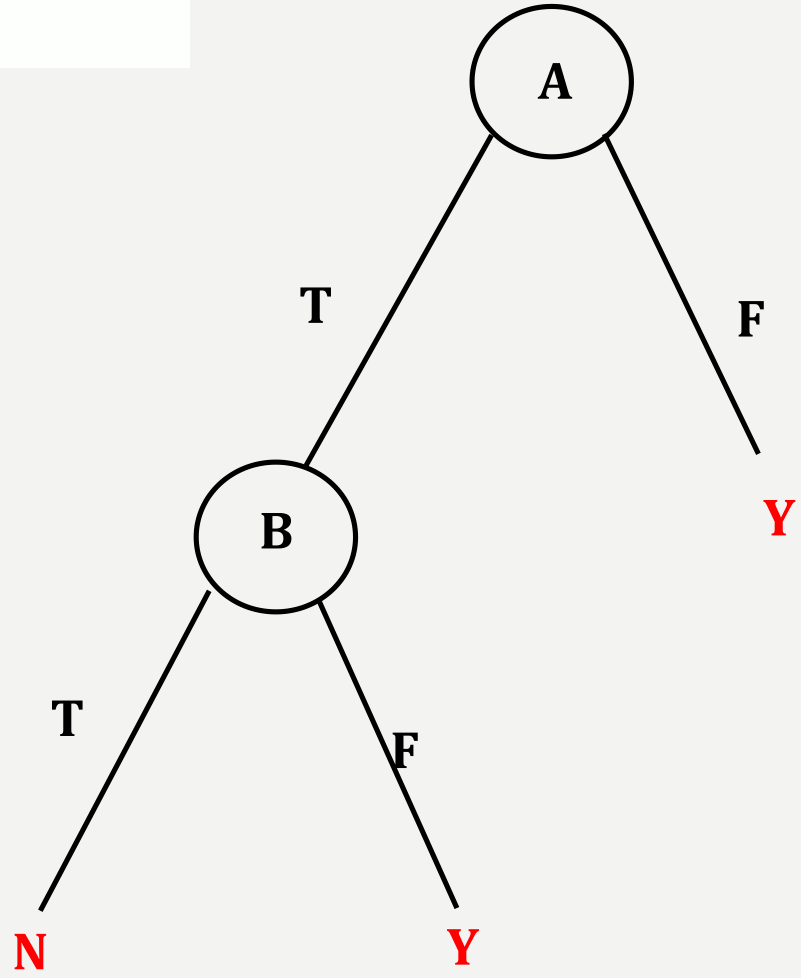
A	B	C	B∧C	A∨[B∧C]
T	T	T	T	T→Y
T	T	F	F	T→Y
T	F	T	F	T→Y
T	F	F	F	T→Y
F	T	T	T	T→Y
F	T	F	F	F→N
F	F	T	F	F→N
F	F	F	F	F→N



Decision Tree for Boolean Functions

(c) A XOR B

A	B	A XOR B
T	T	F→N
T	F	T→Y
F	T	T→Y
F	F	T→Y



Decision Tree for Boolean Function

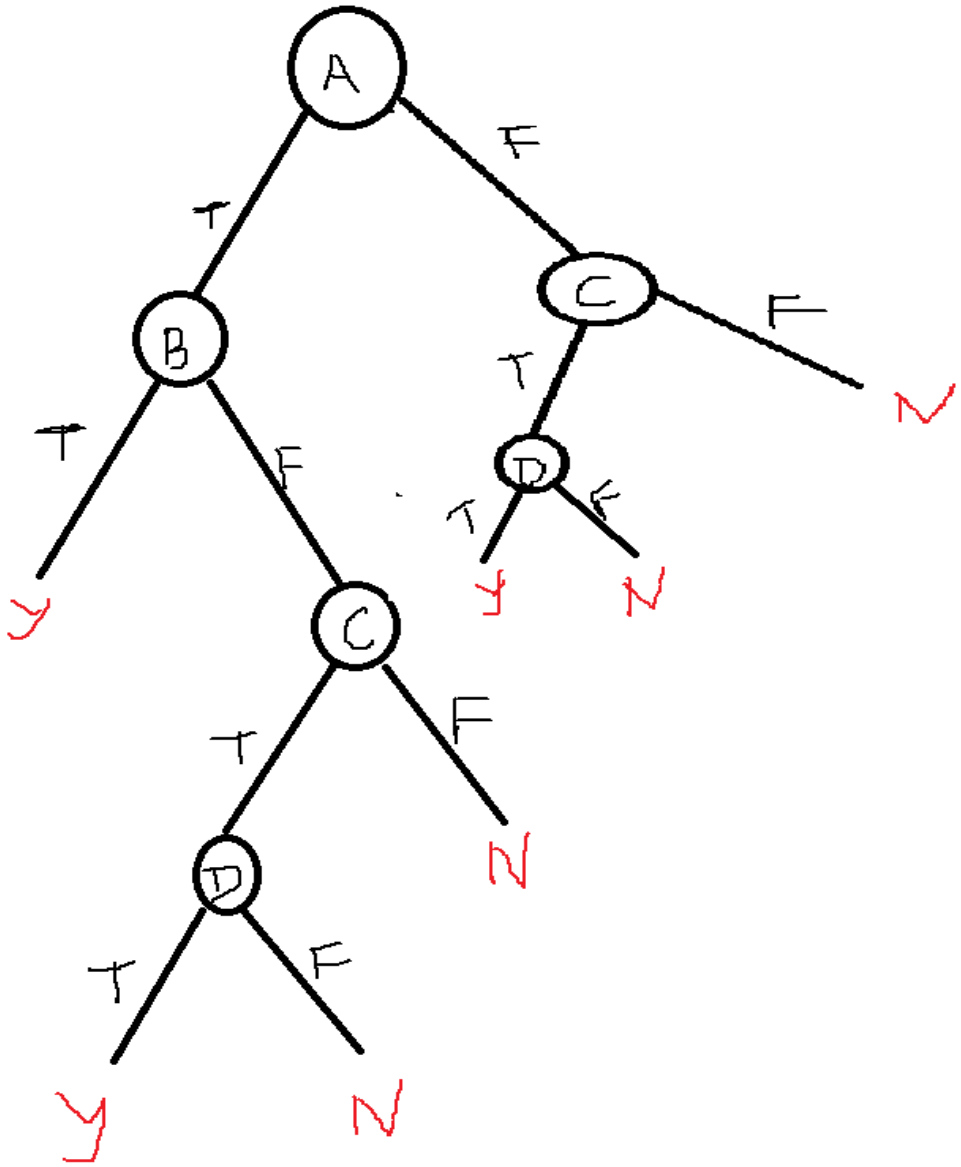
(d) $[A \wedge B] \vee [C \wedge D]$

A	B	A∧B
T	T	T
T	F	F
F	T	F
F	F	F

C	D	C∧D
T	T	T
T	F	F
F	T	F
F	F	F

A∧B	C∧D	[A∧B]∨[C∧D]
T	T	T→Y
F	F	F→N
F	F	F→N
F	F	F→N

A	B	C	D	[A∧B]∨ [C∧D]
T	T	T	T	T→Y
T	F	T	F	F→N
F	T	F	T	F→N
F	F	F	F	F→N



OVERVIEW

- Introduction
- Decision Tree Representation
- Appropriate Problems for Decision Tree Learning
- Basic Decision Tree Learning Algorithm (ID3)
- **Hypothesis Space Search in decision Tree Learning**
- Inductive Bias in Decision Tree Learning
- Issues in Decision Tree Learning

HYPOTHESIS SPACE SEARCH IN ID3

- ID3 can be characterized as searching a space of hypotheses for one that fits the training examples.
- The hypothesis space searched by ID3 is the set of possible decision trees.
- ID3 performs a simple-to-complex, hill-climbing search through this hypothesis space.
- Beginning with the empty tree, then considering progressively more elaborate hypotheses in search of a decision tree that correctly classifies the training data.
- The evaluation function that guides this hill-climbing search is the **information gain measure**.

HYPOTHESIS SPACE SEARCH IN ID3

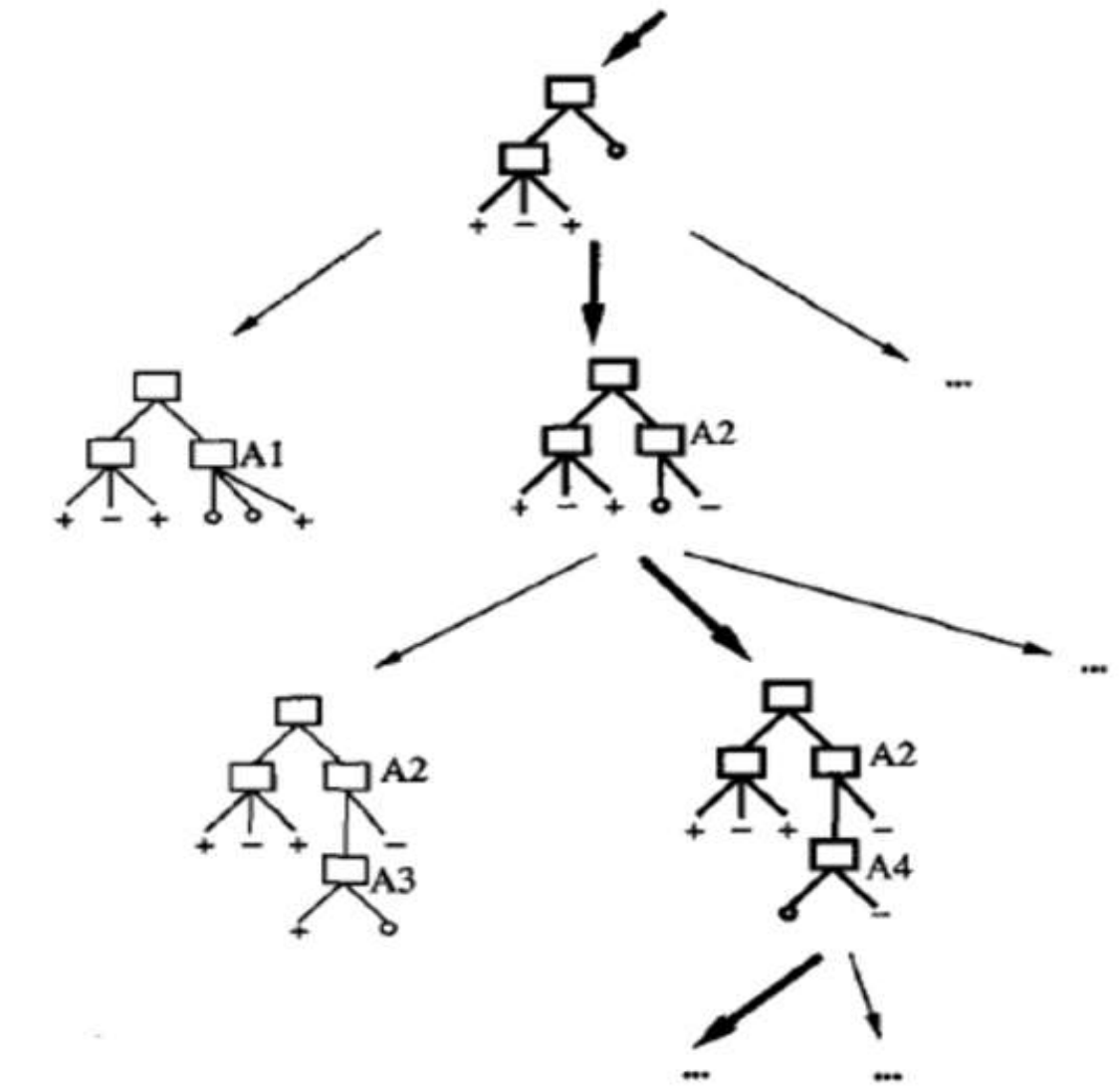


FIGURE 3.5
Hypothesis space search by ID3. ID3 searches through the space of possible decision trees from simplest to increasingly complex, guided by the *information gain heuristic.*

HYPOTHESIS SPACE SEARCH IN ID3

- **Hypothesis space:**
  - The hypothesis space searched by ID3 is the set of possible decision trees.
  - It is a complete space of finite discrete-valued functions, relative to the available attributes.
- **Search Method:**
  - ID3 performs a simple-to-complex, hill-climbing search.
- **Evaluate Function**: Information Gain

CAPABILITIES AND LIMITATIONS OF ID3

- **ID3's hypothesis space of all decision spaces is a *complete* space of finite discrete-valued functions, relative to the available attributes.**
  - Because every finite discrete-valued function can be represented by some decision tree, ID3 avoids one of the major risks of methods that search incomplete hypothesis spaces (such as version space methods that consider only conjunctive hypotheses): that the hypothesis space might not contain the target function.
- **ID3 maintains only a *single current hypothesis* as it searches through the space of decision trees.**
  - This contrasts, for example, with the earlier version space candidate-elimination method, which maintains the set of *all* hypotheses consistent with the available training examples.
  - However, by determining only a single hypothesis, ID3 loses the capabilities that follow from explicitly representing all consistent hypotheses.

CAPABILITIES AND LIMITATIONS OF ID3

- **ID3, in its pure form, performs *no backtracking* in its search (greedy algorithm).**
  - Once it selects an attribute to test at a particular level in the tree, it never backtracks to reconsider this choice; it is susceptible to the usual risks of hill-climbing search without backtracking: converging to locally optimal solutions that are not globally optimal.
- **ID3 uses all training examples at each step in the search to make statistically based decisions regarding how to refine its current hypothesis.**
  - This contrasts with methods that make decisions incrementally, based on individual training examples (eg. version space candidate-elimination).
  - One advantage of using statistical properties of all the examples is that the resulting search is much less sensitive to errors in individual training examples. ID3 can be easily extended to handle noisy training data by modifying its termination criterion to accept hypotheses that imperfectly fit the training data.

OVERVIEW

- Introduction
- Decision Tree Representation
- Appropriate Problems for Decision Tree Learning
- Basic Decision Tree Learning Algorithm (ID3)
- Hypothesis Space Search in decision Tree Learning
- **Inductive Bias in Decision Tree Learning**
- Issues in Decision Tree Learning

INDUCTIVE BIAS

- Inductive bias is the set of assumptions that, together with the training data, deductively justify the classifications assigned by the learner to future instances.
- Given a collection of training examples, there are typically many decision trees consistent with these examples.
- ***Describing the inductive bias of ID3*** therefore consists of describing the basis by which it chooses one of these consistent hypotheses over the others.
  - It chooses the first acceptable tree it encounters in its simple-to-complex, hill-climbing search through the space of possible trees.
- The ID3 search strategy

  (a) selects in favor of **shorter trees over longer ones**, and

  (b) selects trees that place the **attributes with highest information gain closest to the root**.

INDUCTIVE BIAS

- **Approximate inductive bias of ID3:** Shorter trees are preferred over larger trees.
  - ID3 can be viewed as an efficient approximation to BFS-ID3, using a greedy heuristic search to attempt to find the shortest tree without conducting the entire breadth-first search through the hypothesis space.
  - Because ID3 uses the information gain heuristic and a hill climbing strategy, it exhibits a more complex bias than BFS-ID3.
  - It **is biased to favor trees** that place **attributes with high information gain closest to the root**.
- **A closer approximation to the inductive bias of ID3**
  - Shorter trees are preferred over longer trees.
  - Trees that place high information gain attributes close to the root are preferred over those that do not.

RESTRICTION BIASES AND PREFERENCE BIASES

- Comparing inductive Bias of ID3 and Candidate Elimination algorithm

ID3	Candidate Elimination Algorithm
ID3 Searches : a complete hypothesis space	CEA :Searches, incomplete hypothesis space
It Searches incompletely through this hypothesis space from simple to complex hypothesis until the termination condition is met.	It Searches this space completely, finding every hypothesis consistent with the training data,
Inductive bias based on search strategy	Inductive bias based on hypothesis representation

RESTRICTION BIASES AND PREFERENCE BIASES

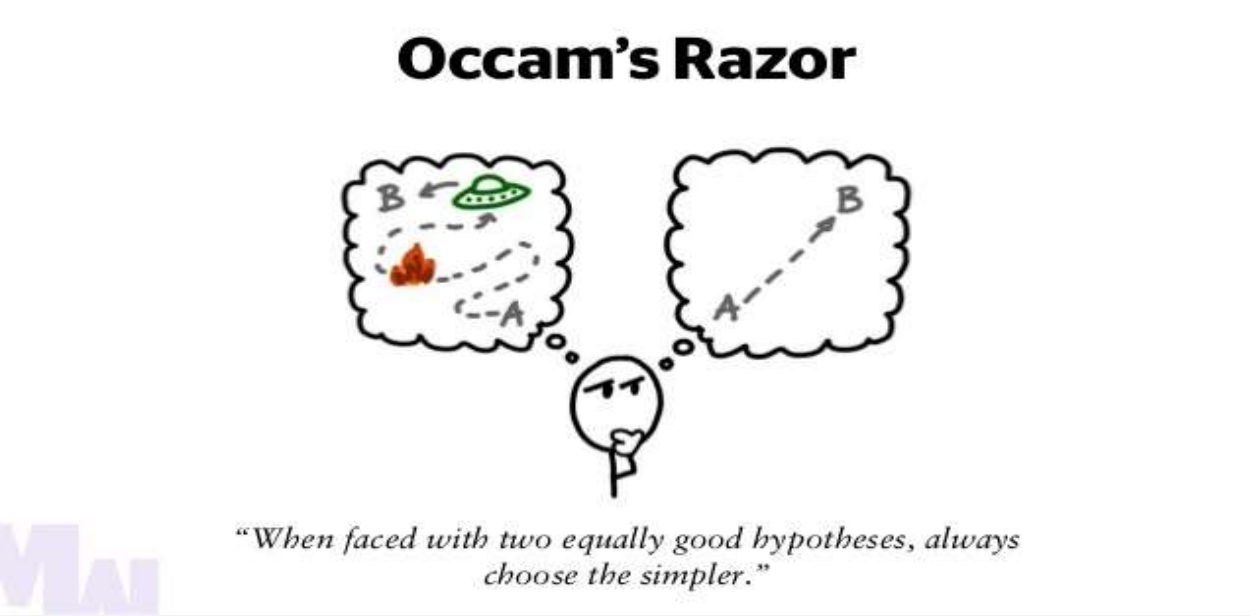
- The inductive bias of ID3 is thus a ***preference*** for certain hypotheses over others (e.g., for shorter hypotheses)
  - This form of bias is typically called a preference bias (or, alternatively, a search bias).
- In contrast, the bias of the CEA is in the form of a categorical restriction on the set of hypotheses considered.
  - This form of bias is typically called a restriction bias (or, alternatively, a language bias).
- A preference bias is more desirable than a restriction bias
- ID3 exhibits a purely preference bias and CEA is a purely restriction bias whereas some learning systems combine both.

WHY PREFER SHORT HYPOTHESES?

- **William of Occam** was one of the first to discuss the question, around the year 1320, so this bias often goes by the name of **Occam's razor.**

Occam's razor: *Prefer the simplest hypothesis that fits the data.*

- It is the problem solving principle that simplest solution tends to be right one.
- When presented with ***competing hypothesis*** to solve a problem, one should select a solution with the ***fewest assumptions.***
- ***Shorter hypothesis*** fits the training data which are less likely to be consist training data.



WHY PREFER SHORT HYPOTHESES?

Argument in favour:

- Fewer short hypotheses than long ones:
  - Short hypotheses fits the training data which are *less likely* to be *coincident*
  - Longer hypotheses fits the training data might be *coincident.*
- Many complex hypotheses that fit the current training data but fail to generalize correctly to subsequent data.

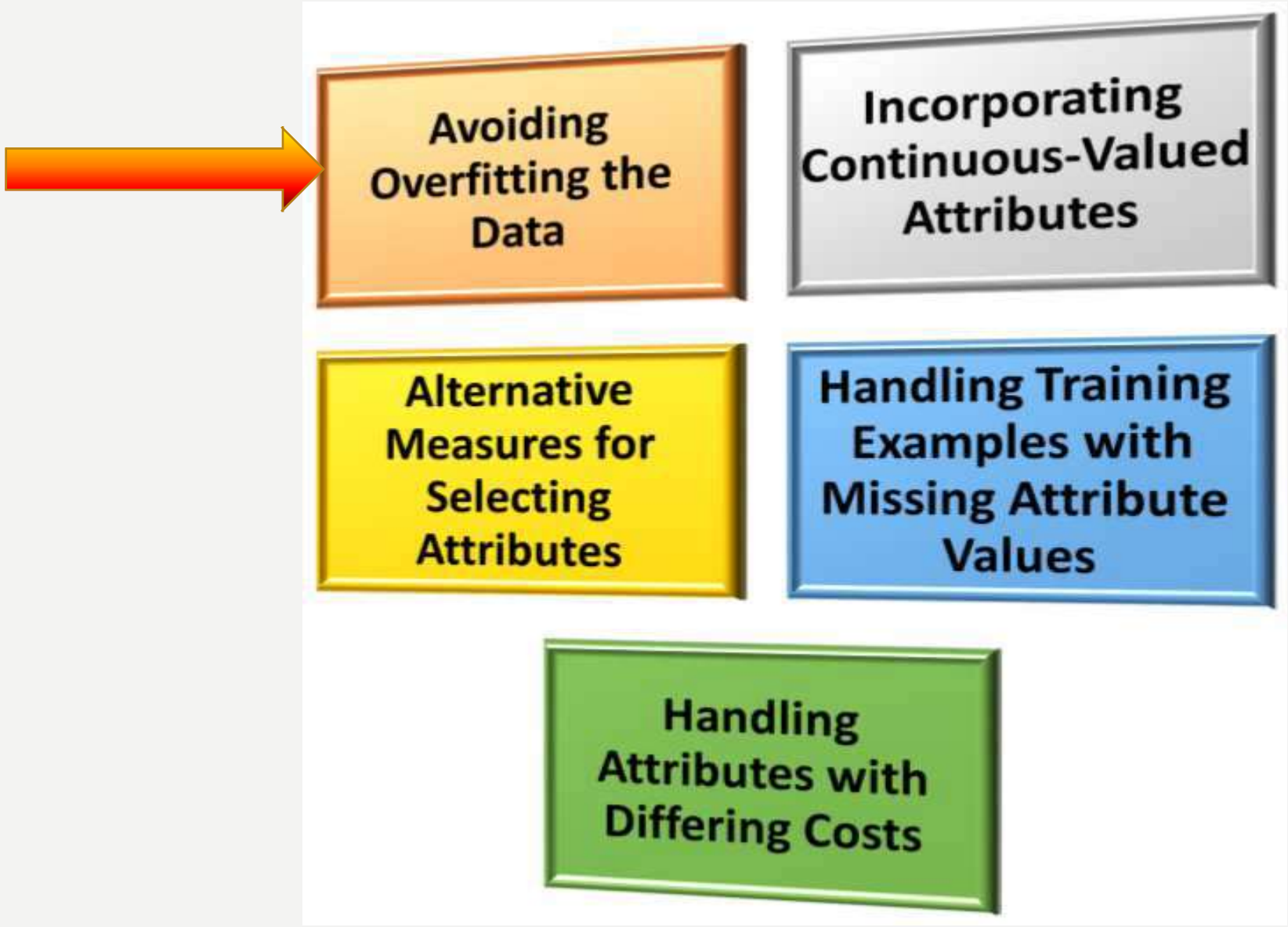
Argument opposed:

- There are few small trees, and our priori chance of finding one consistent with an arbitrary set of data is therefore small. The difficulty here is that there are very many small sets of hypotheses that one can define *but understood by fewer learner*.
- The size of a hypothesis is determined by the representation used ***internally*** by the learner. Occam's razor will produce *two different hypotheses from the same training examples when it is applied by two learners*, both justifying their contradictory conclusions by Occam's razor. On this basis we might be tempted to reject Occam's razor altogether.

OVERVIEW

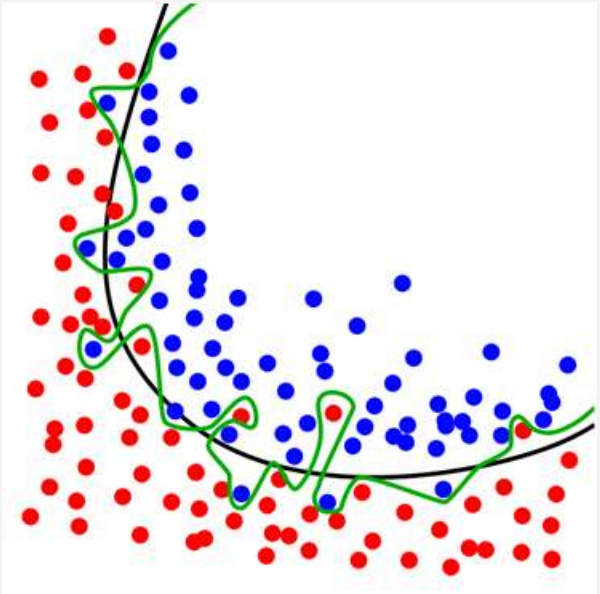
- Introduction
- Decision Tree Representation
- Appropriate Problems for Decision Tree Learning
- Basic Decision Tree Learning Algorithm (ID3)
- Hypothesis Space Search in decision Tree Learning
- Inductive Bias in Decision Tree Learning
- **Issues in Decision Tree Learning**

ISSUES IN DECISION TREE LEARNING



OVERFITTING

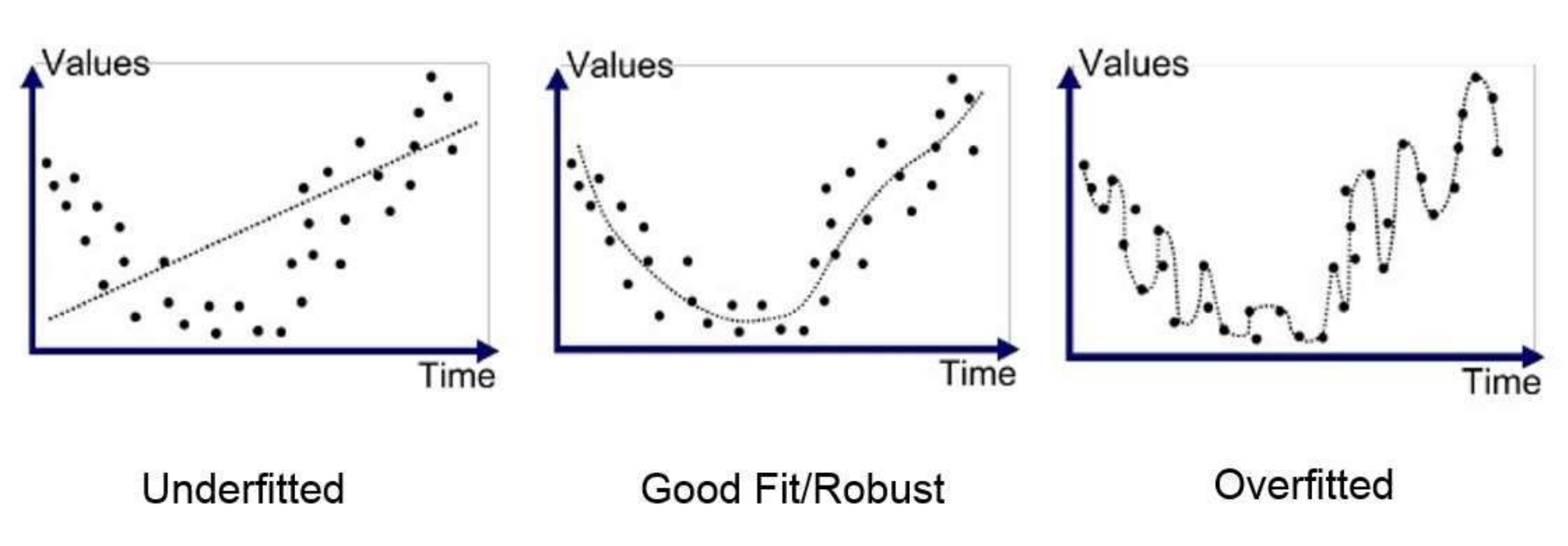
- Consider 2D data. +ve examples are plotted in Blue, -ve are in Red
- The green line represents an overfitted model and the black line represents a regularized model.
- While the green line best follows the training data, it is too dependent on that data and it is likely to have a higher error rate on new unseen data, compared to the black line.



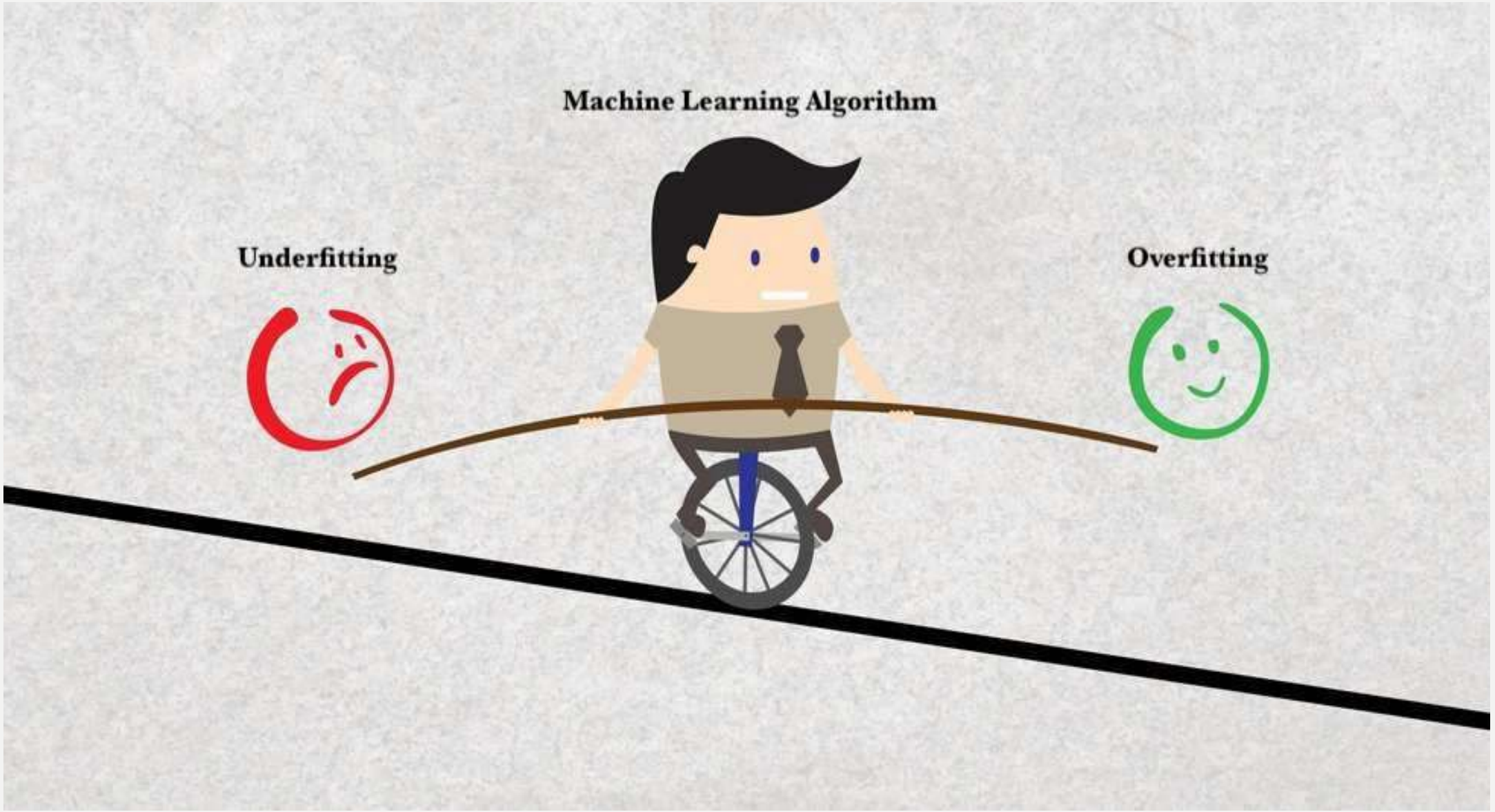
(Source: wikipedia)

UNDERFITTING

- **Underfitting** occurs when a statistical model cannot adequately capture the underlying structure of the data.



Source: http://blog.algotrading101.com/design-theories/what-is-curve-fitting-overfitting-in-trading/


Machine Learning Algorithm
Underfitting
Overfitting

OVERFITTING

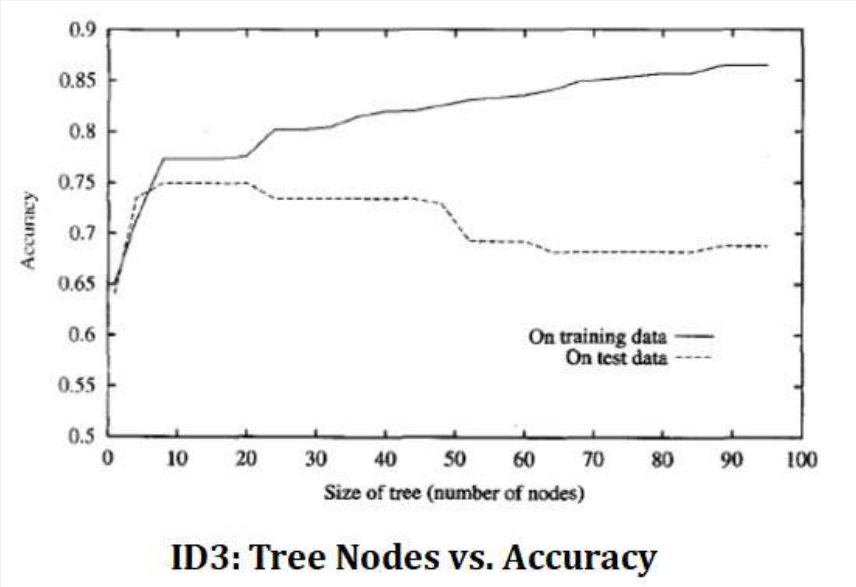
- The ID3 algorithm grows each branch of the tree just deeply enough to perfectly classify the training examples but it can lead to difficulties when there is noise in the data, or when the number of training examples is too small to produce a representative sample of the true target function. This algorithm can produce trees that ***overfit*** the training examples.

Definition: Overfitting

- Given a hypothesis space **H,** a hypothesis **h** ∈ **H** is said to **overfit the training data,** if there exists some alternative hypothesis **h'** ∈ **H**, such that **h** has smaller error than **h'** over the training examples, but **h'** has a smaller error than **h** over the entire distribution of instances.

OVERFITTING

- The below figure illustrates the impact of overfitting in a typical application of decision tree learning.



ID3: Tree Nodes vs. Accuracy

- The **horizontal axis** of this plot indicates the total number of nodes in the decision tree, as the tree is being constructed. The **vertical axis** indicates the accuracy of predictions made by the tree.
- The **solid line** shows the accuracy of the decision tree over the training examples. The **broken line** shows accuracy measured over an independent set of test example
- The **accuracy** of the tree over the **training examples increases** monotonically as the tree is grown.
- The **accuracy** measured over the independent test examples **first increases, then decreases**.

OVERFITTING

How can it be possible for tree h to fit the training examples better than h', but for it to perform more poorly over subsequent examples?

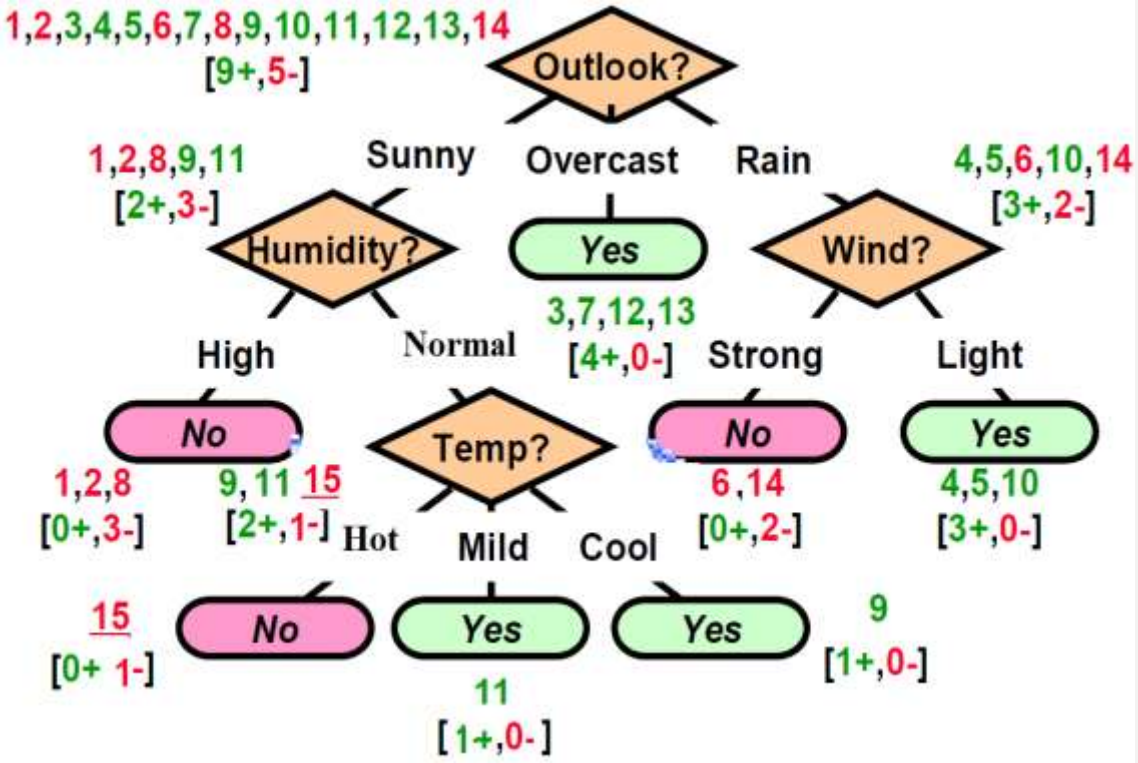
Reasons for Overfitting:

1. Overfitting can occur when the training examples contain random errors or noise
2. When small numbers of examples are associated with leaf nodes.

Noisy Training Example

Example 15: *<Sunny, Hot, Normal, Strong, ->*

- Example is noisy because the correct label is +
- Previously constructed tree misclassifies it



- **reasons for overfitting:**
  - noise in the data
  - number of training examples is to small too produce a representative sample of the target function

APPROACHES TO AVOIDING OVERFITTING IN DECISION TREE LEARNING

- **Pre-pruning (avoidance):** Stop growing the tree earlier, before it reaches the point where it perfectly classifies the training data
- **Post-pruning (recovery):** Allow the tree to overfit the data, and then post-prune the tree

Criterion used to determine the correct final tree size

- Use a separate set of examples, distinct from the training examples, to evaluate the utility of post-pruning nodes from the tree.
- Use all the available data for training, but apply ***a statistical test*** to estimate whether expanding (or pruning) a particular node is likely to produce an improvement beyond the training set.
- Use measure of the complexity for encoding the training examples and the decision tree, halting growth of the tree when this encoding size is minimized. This approach is called the Minimum Description Length:

MDL –> Minimize : size(tree) + size (misclassifications(tree))

APPROACHES TO AVOIDING OVERFITTING IN DECISION TREE LEARNING

- **how to avoid overfitting:**
  - **stop the tree grow earlier**, before it reaches the point where it perfectly classifies the training data
  - allow overfitting and then **post-prune** the tree (more successful in practice!)

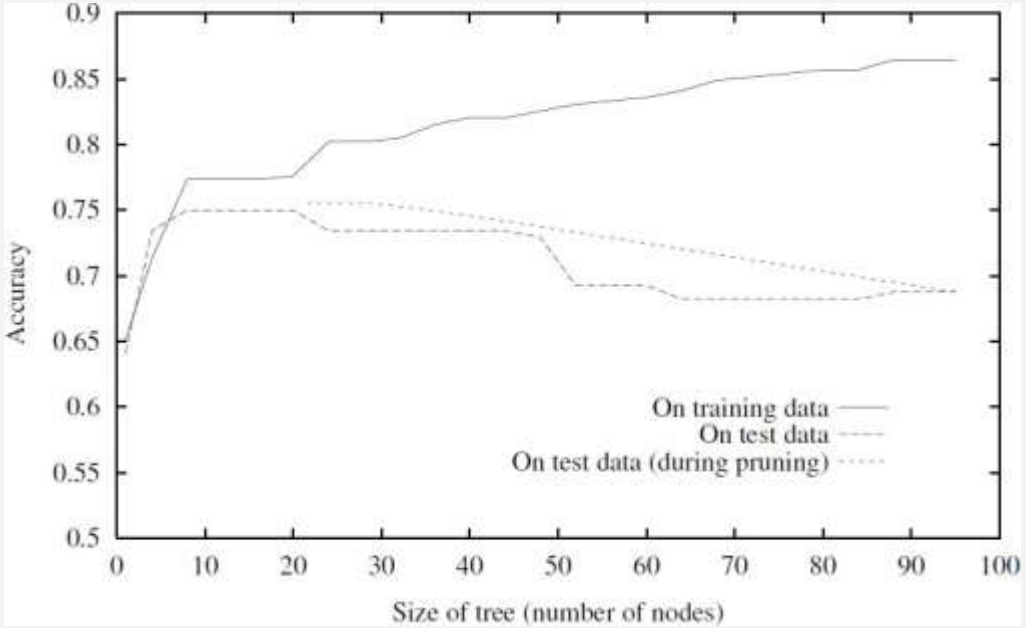
- **how to determine the perfect tree size:**
  - separate validation set to evaluate utility of post-pruning
  - apply statistical test to estimate whether expanding (or pruning) produces an improvement

REDUCED-ERROR PRUNING

- ***Reduced-error pruning***, is to consider each of the decision nodes in the tree to be candidates for pruning
- ***Pruning*** a decision node consists of removing the subtree rooted at that node, making it a leaf node, and assigning it the most common classification of the training examples affiliated with that node
- Nodes are removed only if the resulting pruned tree performs no worse than-the original over the validation set.
- Reduced error pruning has the effect that any leaf node added due to coincidental regularities in the training set is likely to be pruned because these same coincidences are unlikely to occur in the validation set

REDUCED-ERROR PRUNING

The impact of reduced-error pruning on the accuracy of the decision tree is illustrated in below figure



- The additional line in figure shows ***accuracy over the test examples*** as the tree is pruned. When pruning begins, the tree is at its maximum size and lowest accuracy over the test set. As pruning proceeds, the number of nodes is reduced and accuracy over the test set increases.
- The available data has been split into three subsets: **the training examples**, **the validation examples** used for pruning the tree, and a set of **test examples** used to provide an unbiased estimate of accuracy over future unseen examples.
- The plot shows accuracy over the training and test sets.

REDUCED-ERROR PRUNING
PROS AND CONS

Pros: Produces smallest version of most accurate T (subtree of T)

Cons:

- Uses less data to construct T
- Can afford to hold out $D_{validation}$?. If not (data is too limited), may make error worse (insufficient D_{train})

RULE POST-PRUNING

- Rule post-pruning is successful method for finding high accuracy hypotheses

Rule post-pruning involves the following steps:

1. Infer the decision tree from the training set, growing the tree until the training data is fit as well as possible and allowing overfitting to occur.
2. Convert the learned tree into an equivalent set of rules by creating one rule for each path from the root node to a leaf node.
3. Prune (generalize) each rule by removing any preconditions that result in improving its estimated accuracy.
4. Sort the pruned rules by their estimated accuracy, and consider them in this sequence when classifying subsequent instances.

CONVERTING DECISION TREES INTO RULES

For example,

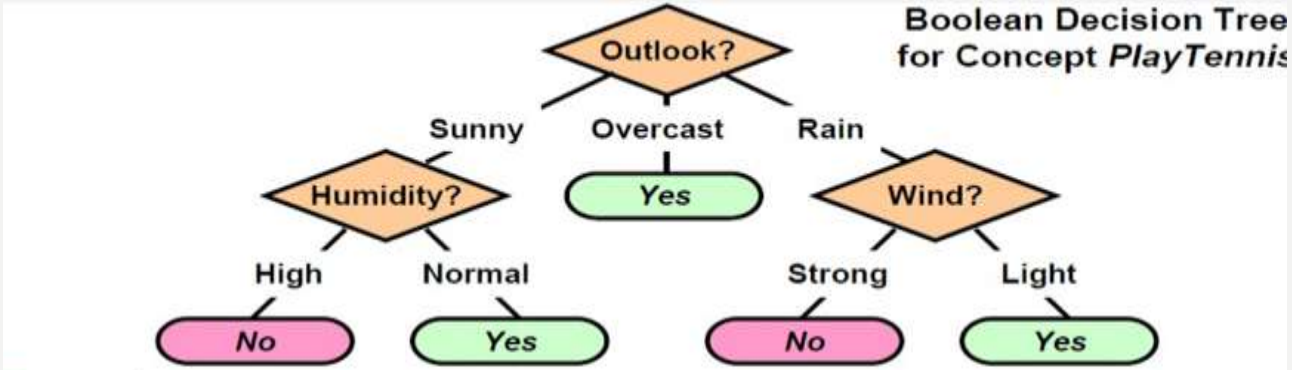
- Consider the decision tree. The leftmost path of the tree in below figure is translated into the rule.

IF (Outlook = Sunny) ^ (Humidity = High) THEN *PlayTennis* = No

- Given the above rule, rule post-pruning would consider removing the preconditions

(Outlook = Sunny) and (Humidity = High)

- It would select whichever of these pruning steps produced the greatest improvement in estimated rule accuracy, then consider pruning the second precondition as a further pruning step.
- No pruning step is performed if it reduces the estimated rule accuracy.



Example

R1: If (Outlook=Sunny) ∧ (Humidity=High) Then PlayTennis=No
R2: If (Outlook=Sunny) ∧ (Humidity=Normal) Then PlayTennis=Yes
R3: If (Outlook=Overcast) Then PlayTennis=Yes
R4: If (Outlook=Rain) ∧ (Wind=Strong) Then PlayTennis=No
R5: If (Outlook=Rain) ∧ (Wind=Weak) Then PlayTennis=Yes

CONVERTING DECISION TREES INTO RULES

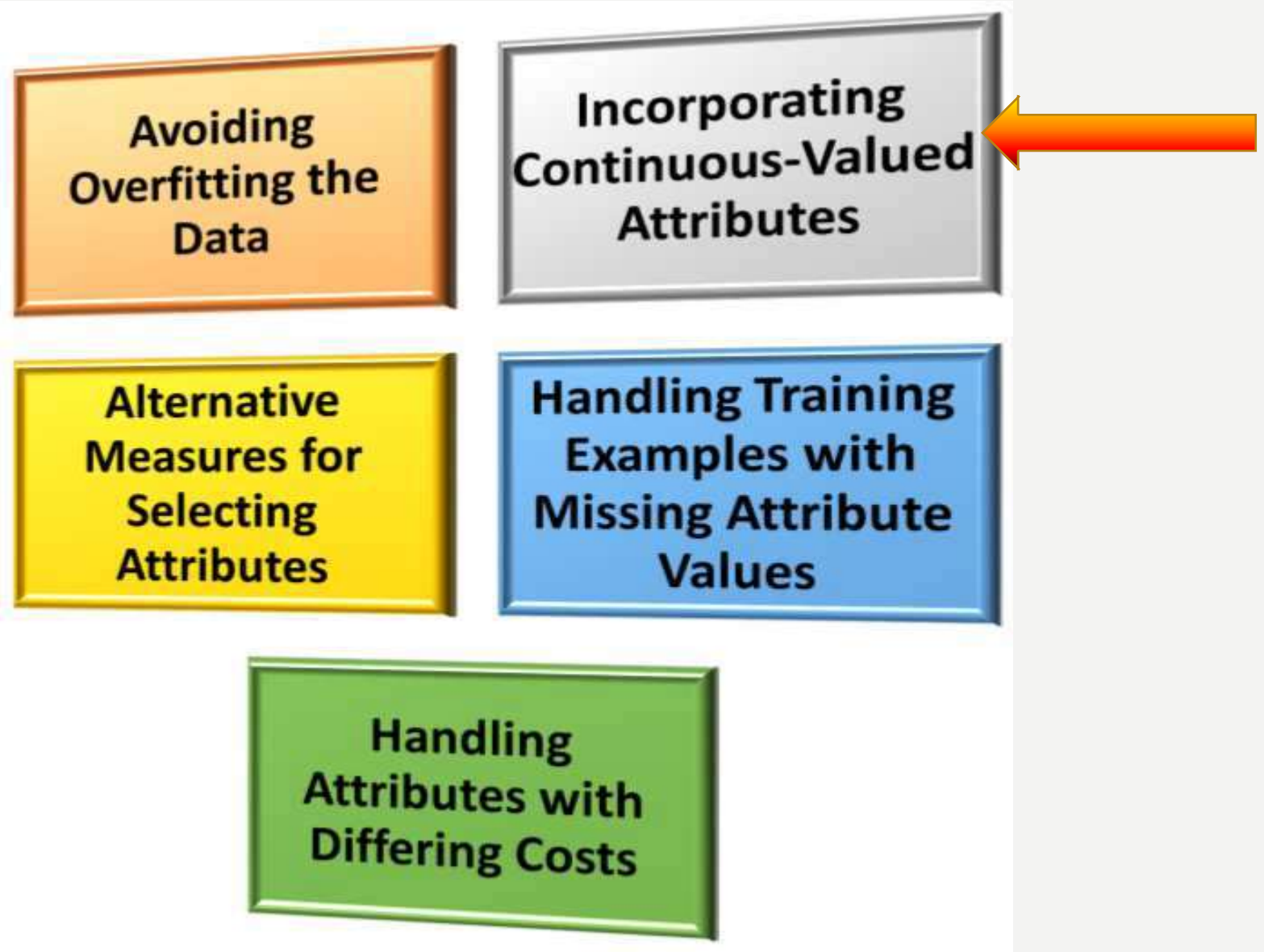
There are three main advantages by converting the decision tree to rules before pruning

- Converting to rules allows distinguishing among the different contexts in which a decision node is used. Because each distinct path through the decision tree node produces a distinct rule, the pruning decision regarding that attribute test can be made differently for each path.

- Converting to rules removes the distinction between attribute tests that occur near the root of the tree and those that occur near the leaves. Thus, it avoid messy bookkeeping issues such as how to reorganize the tree if the root node is pruned while retaining part of the subtree below this test.

- Converting to rules improves readability. Rules are often easier for to understand.

ISSUES IN DECISION TREE LEARNING



INCORPORATING CONTINUOUS VALUED ATTRIBUTES

- ID3 is restricted to attributes that take on a discrete set of values.
- Define new discrete valued attributes that partition the continuous attribute value into a discrete set of intervals
- For a continuous-valued attribute A that is, create a new boolean attribute Ac, that is true if A < c and false otherwise.
  - Select c using information gain
  - Sort examples according to the continuous attributeA,
  - Then identify adjacent examples that differ in their target classification
  - Generate candidate thresholds midway between corresponding values of A.
  - The value of c that maximizes information gain must always lie at a boundary.
  - These candidate thresholds can then be evaluated by computing the information gain associated with each.

INCORPORATING CONTINUOUS VALUED ATTRIBUTES

- Continuous-valued decision attributes can be incorporated into the learned tree.
- **There are two methods for Handling Continuous Attributes**

1. Define new discrete valued attributes that partition the continuous attribute value into a discrete set of intervals.

   E.g., {high ≡ Temp > 35º C, med ≡ 10º C < Temp ≤ 35º C, low ≡ Temp ≤ 10º C}

2. Using thresholds for splitting nodes

   E.g., A ≤ a produces subsets A ≤ a and A > a

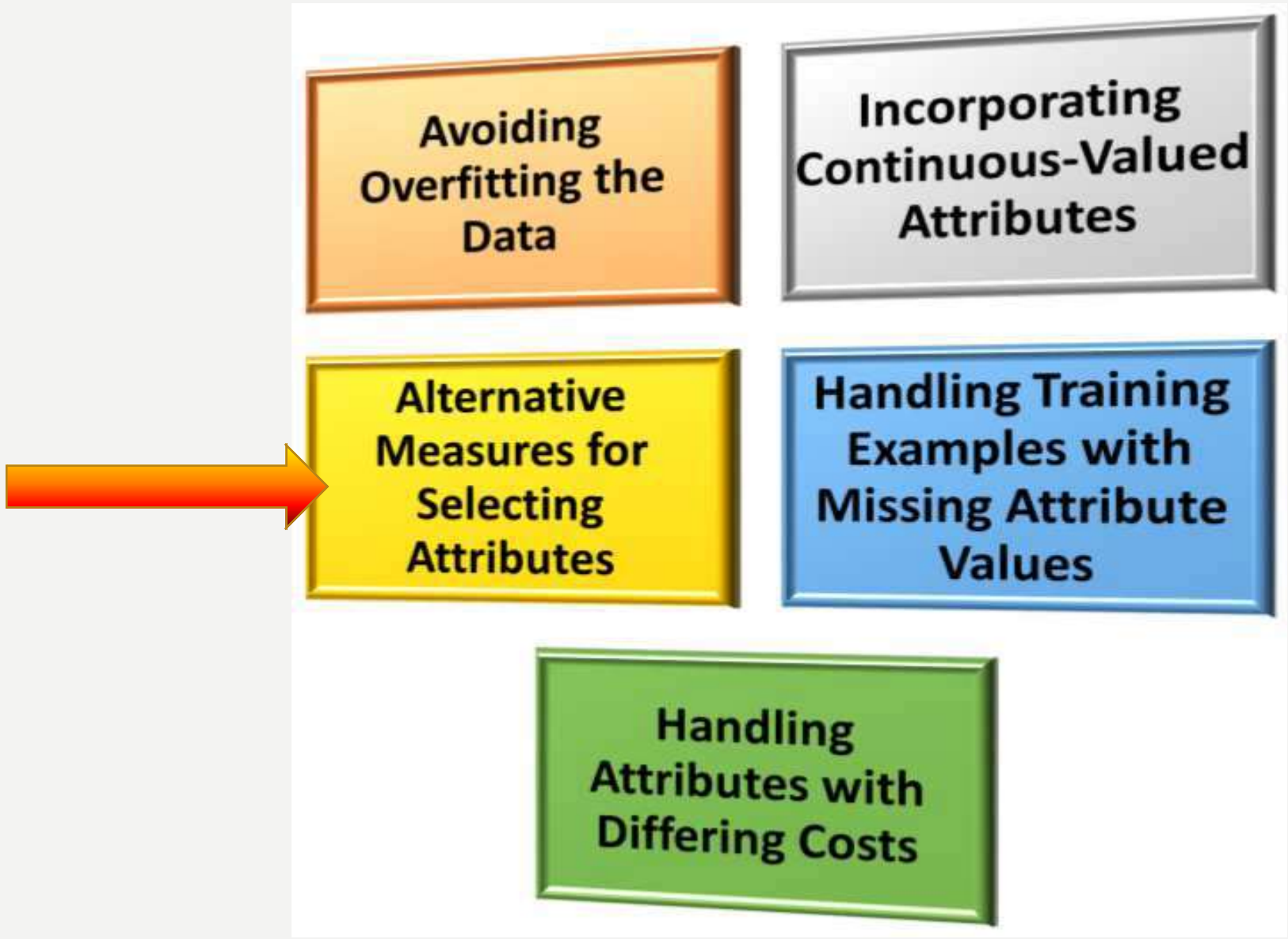
- ***What threshold-based Boolean attribute should be defined based on Temperature?***

Temperature:	40	48	60	72	80	90
PlayTennis:	No	No	Yes	Yes	Yes	No

- Pick a threshold, ***c,*** that produces the greatest information gain

  ➢In the current example, there are **two candidate thresholds**, corresponding to the values of Temperature at which the value of *PlayTennis* changes: (48 + 60)/2, and (80 + 90)/2. The information gain can then be computed for each of the candidate attributes, Temperature $_{>54}$, and Temperature $_{>85}$ and the best can be selected (Temperature $_{>54}$)

ISSUES IN DECISION TREE LEARNING



ALTERNATIVE MEASURES FOR SELECTING ATTRIBUTES

- The problem is if attributes with many values, ***Gain*** will select it ?

Example: consider the attribute ***Date***, which has a very large number of possible values. (e.g., March 4, 1979).

- If this attribute is added to the ***PlayTennis*** data, it would have the highest information gain of any of the attributes. This is because Date alone perfectly predicts the target attribute over the training data. Thus, it would be selected as the decision attribute for the root node of the tree and lead to a tree of depth one, which perfectly classifies the training data.
- This decision tree with root node ***Date*** is not a useful predictor because it perfectly separates the training data, but poorly predict on subsequent examples.

One Approach: Use *GainRatio* instead of *Gain*

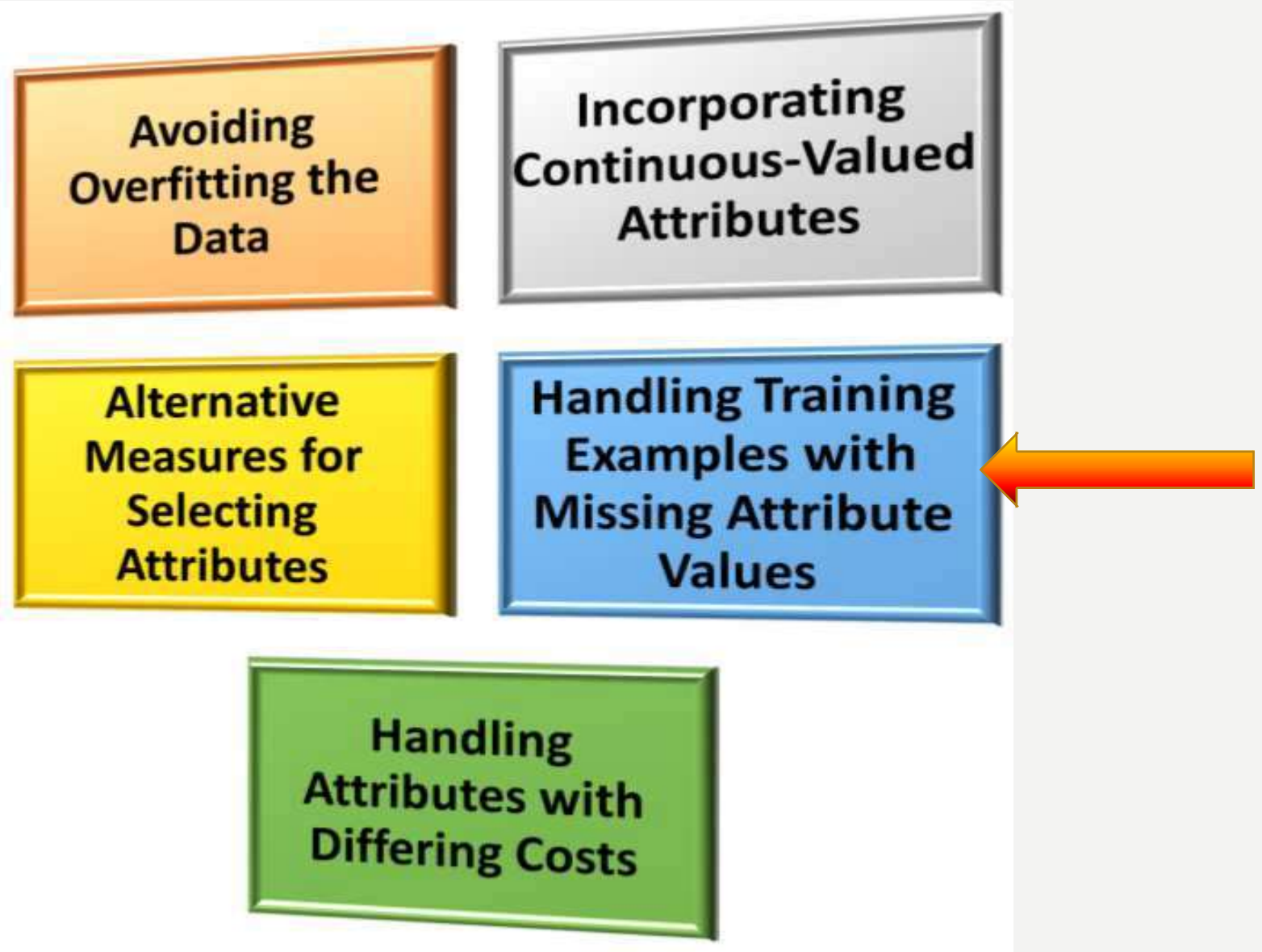
- The gain ratio measure penalizes attributes by incorporating a split information, that is sensitive to how broadly and uniformly the attribute splits the data.

$$GainRatio(S,A) \equiv \frac{Gain(S,A)}{SplitInformation(S,A)}$$

$$SplitInformation(S,A) \equiv -\sum_{i=1}^{c} \frac{|S_i|}{|S|} \log_2 \frac{|S_i|}{|S|}$$

- where S_i is subset of S, for which attribute A has value v_i

ISSUES IN DECISION TREE LEARNING



HANDLING TRAINING DATA MISSING ATTRIBUTE VALUES

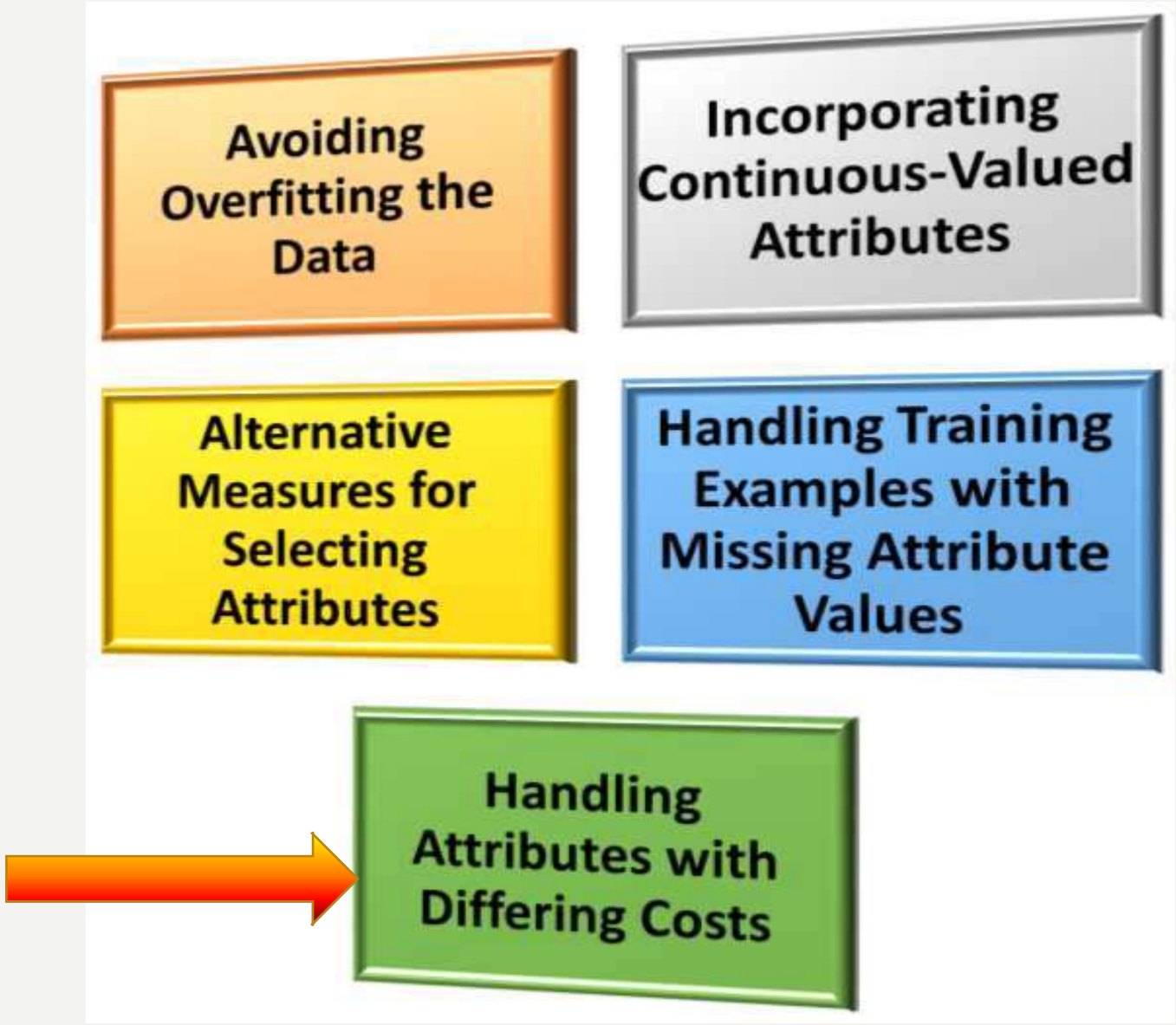
- The available data may be missing values for some attributes.
- It is common to estimate the missing attribute value based on other examples for which this attribute has a known value.
- Assume that an example (with classification c) in S has a missing value for attribute A.
  - Assign the most common value of A in S.
  - Assign the most common value of in the examples having c classification in S.
  - Or, use probabilty value for each possible attribute value.

MISSING ATTRIBUTE VALUES

Example : PlayTennis

Day	Outlook	Temperature	Humidity	Wind	*PlayTennis*?
1	Sunny	Hot	High	Light	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Light	Yes
4	Rain	Mild	High	Light	Yes
5	Rain	Cool	Normal	Light	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	???	Light	No
9	Sunny	Cool	Normal	Light	Yes
10	Rain	Mild	Normal	Light	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Light	Yes
14	Rain	Mild	High	Strong	No

ISSUES IN DECISION TREE LEARNING



HANDLING ATTRIBUTES WITH DIFFERENT COST

- In some learning tasks the instance attributes may have associated costs.
- **For example:**
  - In learning to classify medical diseases, the patients described in terms of attributes such as Temperature, BiopsyResult, Pulse, BloodTestResults, etc.
  - These attributes vary significantly in their costs, both in terms of monetary cost and cost to patient comfort.
  - Decision trees use low-cost attributes where possible, depends only on high-cost attributes only when needed to produce reliable classifications
- **How to Learn A Consistent Tree with Low Expected Cost?**

  One approach is replace Gain by ***Cost-Normalized-Gain***
- **Examples of normalization functions**

  • Tan and Schlimmer

$$\frac{Gain^2(S,A)}{Cost(A)}.$$

  • Nunez

$$\frac{2^{Gain(S,A)}-1}{(Cost(A)+1)^w}$$

where $w \in [0,1]$ determines importance of cost

SUMMARY

The main points in this module include:

- Decision tree learning provides a practical method for concept learning and for learning other discrete-valued functions.
- ID3 searches a complete hypothesis space
- The inductive bias implicit in ID3 includes a *preference for smaller trees*
- Overfitting the training data is an important issue in decision tree learning.
- A large variety of extensions to the basic ID3 algorithm has been developed by different researchers. These include methods for
  - post-pruning trees,
  - handling real-valued attributes
  - accommodating training examples with missing attribute values
  - incrementally refining decision trees as new training examples available
  - using attribute selection measures other than information gain
  - considering costs associated with instance attributes.

QUESTION ON DECISION TREES

1. What is decision tree and decision tree learning?
2. Explain representation of decision tree with example.
3. What are appropriate problems for Decision tree learning?
4. Explain the concepts of Entropy and Information gain.
5. Describe the ID3 algorithm for decision tree learning with example
6. Give Decision trees to represent the Boolean Functions:
    a. A && ~ B
    b. A V [B ∧C]
    c. A XOR B
    d. [A∧B] V [C∧D]

QUESTION ON DECISION TREES

7. Give Decision trees for the following set of training examples

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

QUESTION ON DECISION TREES

8. Consider the following set of training examples.

- **What is the entropy of this collection of training examples with respect to the target function classification?**
- **What is the information gain of a_2 relative to these training examples?**

Instance	Classification	a_1	a_2
1	+	T	T
2	+	T	T
3	–	T	F
4	+	F	F
5	–	F	T
6	–	F	T

QUESTION ON DECISION TREES

9. Identify the entropy, information gain and draw the decision tree for the following set of training examples

Gender	Car ownership	Travel cost	Income Level	Transportation (Class)
Male	Zero	Cheap	Low	Bus
Male	One	Cheap	Medium	Bus
Female	One	Cheap	Medium	Train
Female	Zero	Cheap	Low	Bus
Male	One	Cheap	Medium	Bus
Male	Zero	Standard	Medium	Train
Female	One	Standard	Medium	Train
Female	One	Expensive	High	Car
Male	Two	Expensive	Medium	Car
Female	Two	Expensive	High	Car

QUESTION ON DECISION TREES

10. Construct the decision tree for the following tree using ID3 Algorithm,

Instance	a1	a2	a3	Classification
1	True	Hot	High	No
2	True	Hot	High	No
3	False	Hot	High	Yes
4	False	Cool	Normal	Yes
5	False	Cool	Normal	Yes
6	True	Cool	High	No
7	True	Hot	High	No
8	True	Hot	Normal	Yes
9	False	Cool	Normal	Yes
10	False	Cool	High	Yes

QUESTION ON DECISION TREES

11. **Discuss Hypothesis Space Search in Decision tree Learning.**
12. **Discuss Inductive Bias in Decision Tree Learning.**
13. **What are Restriction Biases and Preference Biases and differentiate between them.**
14. **Write a note on Occam's razor and minimum description principal.**
15. **What are issues in learning decision trees?**