LEARNING MODELS

Decision Tree Learning

Introduction

1. Decision tree learning the most widely used.
2. It is practical methods for inductive inference.
3. It is a method that is robust to noisy data and capable of learning disjunctive expressions.
4. ID3, ASSISTANT, and C4.5 decision tree learning methods.

Introduction

- Decision tree learning is a method for approximating discrete-valued target functions, in that the learned function is represented by a decision tree.

- Learned trees can represented as sets of if-then rules to improve human readability.

DT Features

- Features
  - Method for approximating *discrete-valued* functions (including boolean)
  - Learned functions are represented as *decision trees* (or *if-then-else* rules)
  - Expressive hypotheses space, including disjunction
  - Robust to noisy data

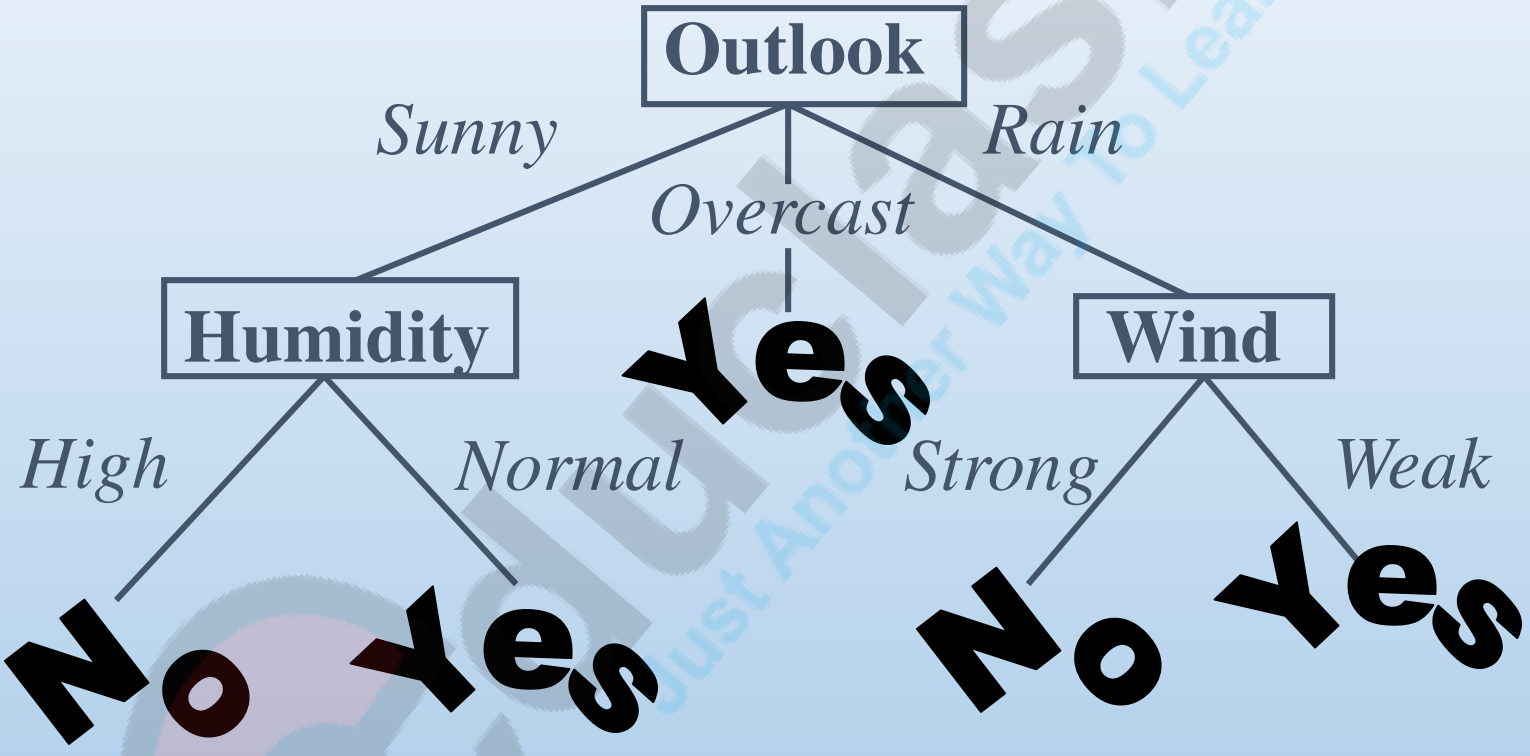
When to use Decision Trees

- Problem characteristics:
  - Instances can be described by attribute value pairs
  - Target function is discrete valued
  - Disjunctive hypothesis may be required
  - Possibly noisy training data samples
    - Robust to errors in training data
    - Missing attribute values
- Different classification problems:
  - Equipment or medical diagnosis
  - Credit risk analysis
  - Several tasks in natural language processing

Decision Tree Representation

- Decision trees classify instances by sorting them down the tree from the root to some leaf node.
- Each node in the tree specifies a test of some attribute of the instance.
- An instance is classified by starting at the root node of the tree, then moving down the tree branch corresponding to the value of the attribute.

Decision Tree Representation



A Decision Tree for the concept *PlayTennis*

Decision Tree For Playing Tennis

- ✓ 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

For example, the instance

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

Appropriate Problems For Decision Tree Learning

- The variety of decision tree learning methods have been developed with differing capabilities and requirements.

- The decision tree learning is generally best suited to problems with the Specific characteristics

Characteristics

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

DT Examples

- Many practical problems have been found to fit these characteristics.

1. Classify medical patients by their disease
2. Equipment malfunctions by their cause,
3. Loan applicants by their likelihood of defaulting on payments.

Classification

- The problems, in which the task is to classify into one of a discrete set of possible categories, are often referred to as classification problems.

  - Classification is a prediction technique.
  - For given the values of the independent attributes the class can be predicted.
  - Classification is used to predict categorical values while regression is used to predict continuous or ordered values.

The Basic Decision Tree Learning Algorithm

- Most algorithms that have been developed for learning decision trees:
  - Core Algorithm That Employs A Top-down
  - Greedy Search Through The Space.
- This approach is exemplified by
  - The ID3 algorithm quinlan 1986)
  - Its successor C4.5 (quinlan 1993),
-

Our basic algorithm, ID3

- Learns decision trees by constructing them top-down.
- "which attribute should be tested at the root of the tree?
- Answer :
- Each instance attribute is evaluated using a statistical test to determine how well it alone classifies the training examples.

Our basic algorithm, ID3

- The best attribute is selected and used as the test at the root node.
- A descendant of the root node is then created for each possible value of this attribute.
- The training examples are sorted to the appropriate descendant node.
- The entire process is then repeated using the training examples associated with each descendant node.

Which Attribute Is the Best Classifier

- The central choice in the ID3 algorithm is selecting which attribute to test at each node in the tree.
- Select the attribute that is most useful for classifying examples.
- Quantitative measure:
- Define a statistical property, called ***information gain.***

Which attribute is the best classifier?

- A statistical property called *information gain*, measures how well a given attribute separates the training examples
- Information gain uses the notion of *entropy*, commonly used in information theory
- *Information gain = expected reduction of entropy*

Entropy Measures Homogeneity Of Examples

- In order to define information gain precisely, we begin by defining a measure commonly used in information theory, called ***entropy, that characterizes the (im)purity*** of an arbitrary collection of examples.

- Given a collection S, containing positive and negative examples of some target concept, the entropy of S relative to this Boolean classification is:

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

DT Algorithm based on ID3

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 ← the attribute from Attributes that best* classifies Examples.

The decision attribute for Root ← A

For each possible value, vi, of A

- Add a new tree branch below Root, corresponding to the test A = vi

ID3 Steps

- 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*

Training examples for the target concept **Play Tennis.**

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

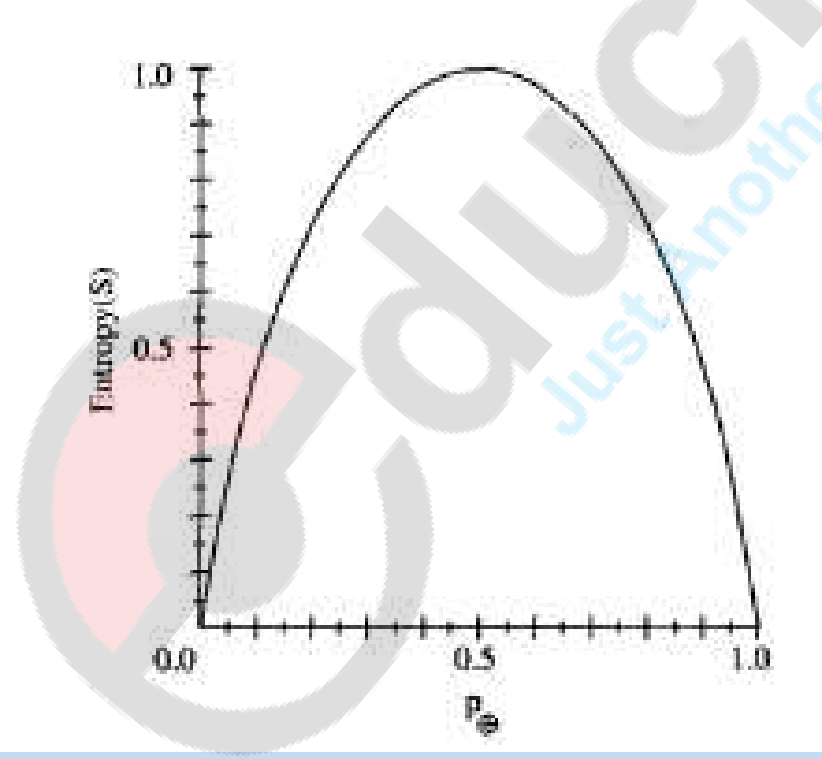
Example

- To illustrate, suppose S is a collection of 14 examples of some Boolean concept, including 9 positive and 5 negative examples (we adopt the notation [9+, 5-1 to summarize such a sample of data). Then the entropy of S relative to this boolean classification is

$$Entropy([9+, 5-]) = -(9/14)\log_2(9/14) - (5/14)\log_2(5/14)$$
$$= 0.940$$

The Entropy

The entropy function relative to a boolean classification, as the proportion, ***pe, of positive examples varies pe between 0 and 1.***



Information Gain Measures

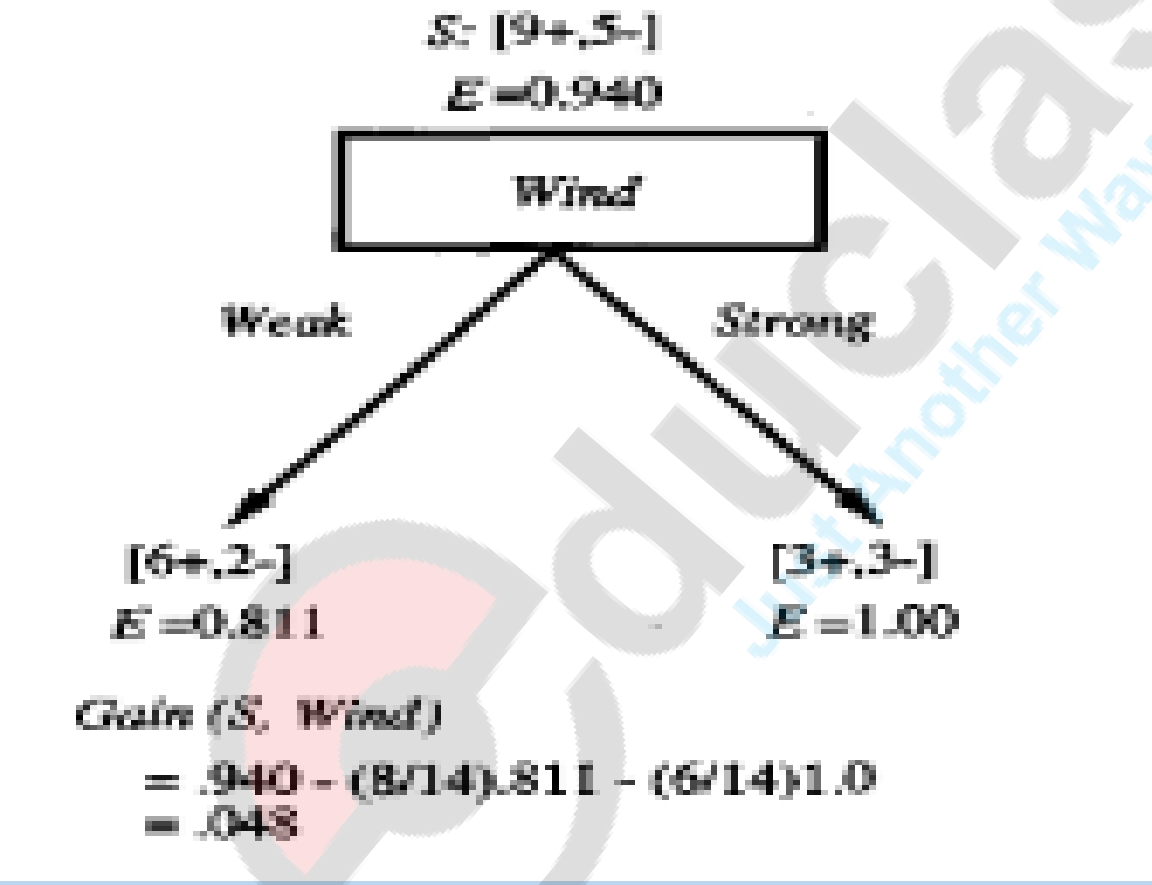
- The Expected Reduction In Entropy
- The measure information gain, is the expected reduction in entropy caused by partitioning the examples according to this attribute.
- The information gain, GAIN(S, A) of an attribute A, relative to a collec
- tion of examples S, is defined as

$$Gain(S, A) \equiv Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

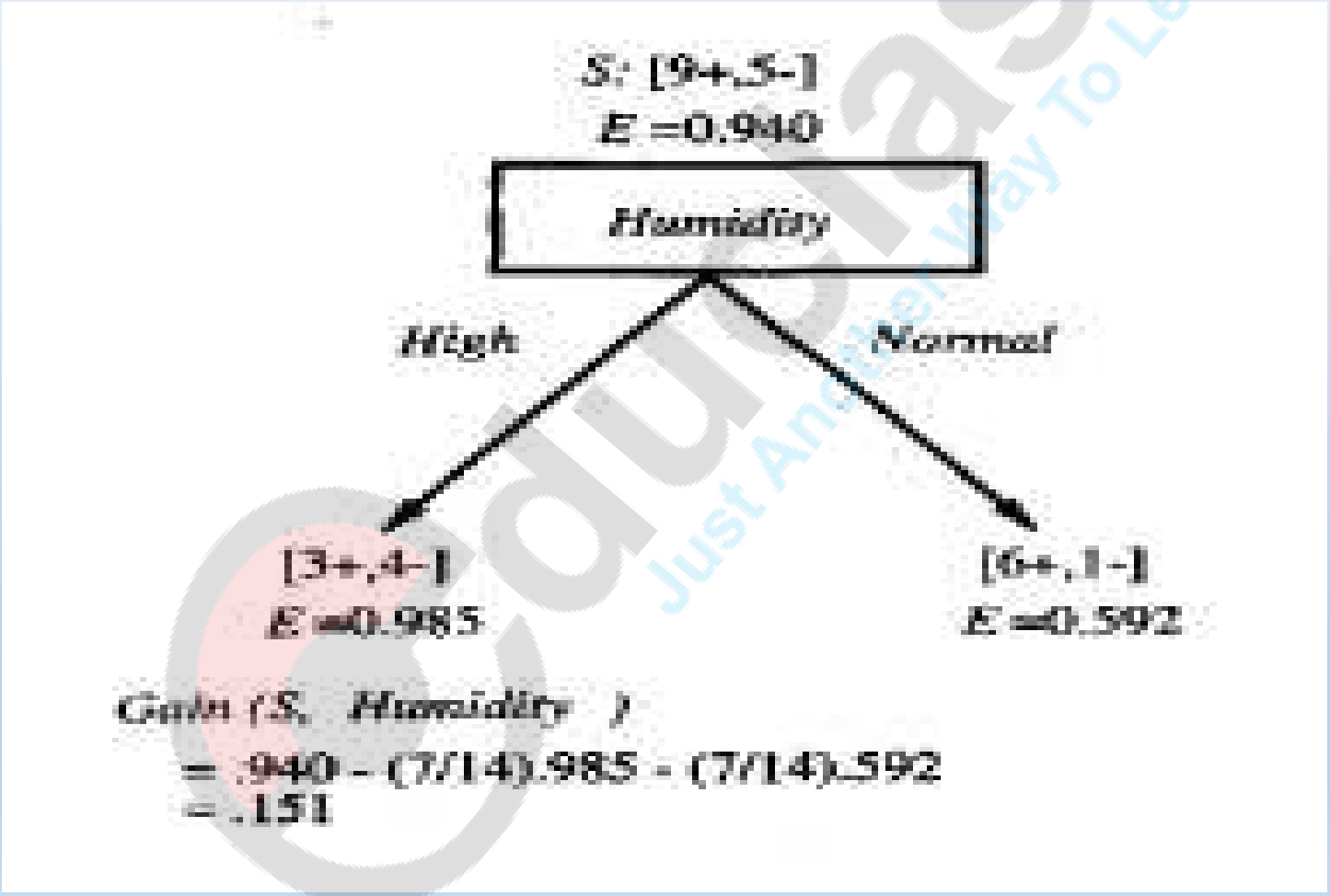
Wind-the values Weak or Strong

$$
\begin{aligned}
Values(Wind) &= Weak, Strong \\
S &= [9+, 5-] \\
S_{Weak} &\leftarrow [6+, 2-] \\
S_{Strong} &\leftarrow [3+, 3-] \\
Gain(S, Wind) &= Entropy(S) - \sum_{v \in \{Weak, Strong\}} \frac{|S_v|}{|S|} Entropy(S_v) \\
&= Entropy(S) - (8/14) Entropy(S_{Weak}) \\
&\quad - (6/14) Entropy(S_{Strong}) \\
&= 0.940 - (8/14)0.811 - (6/14)1.00 \\
&= 0.048
\end{aligned}
$$

Wind

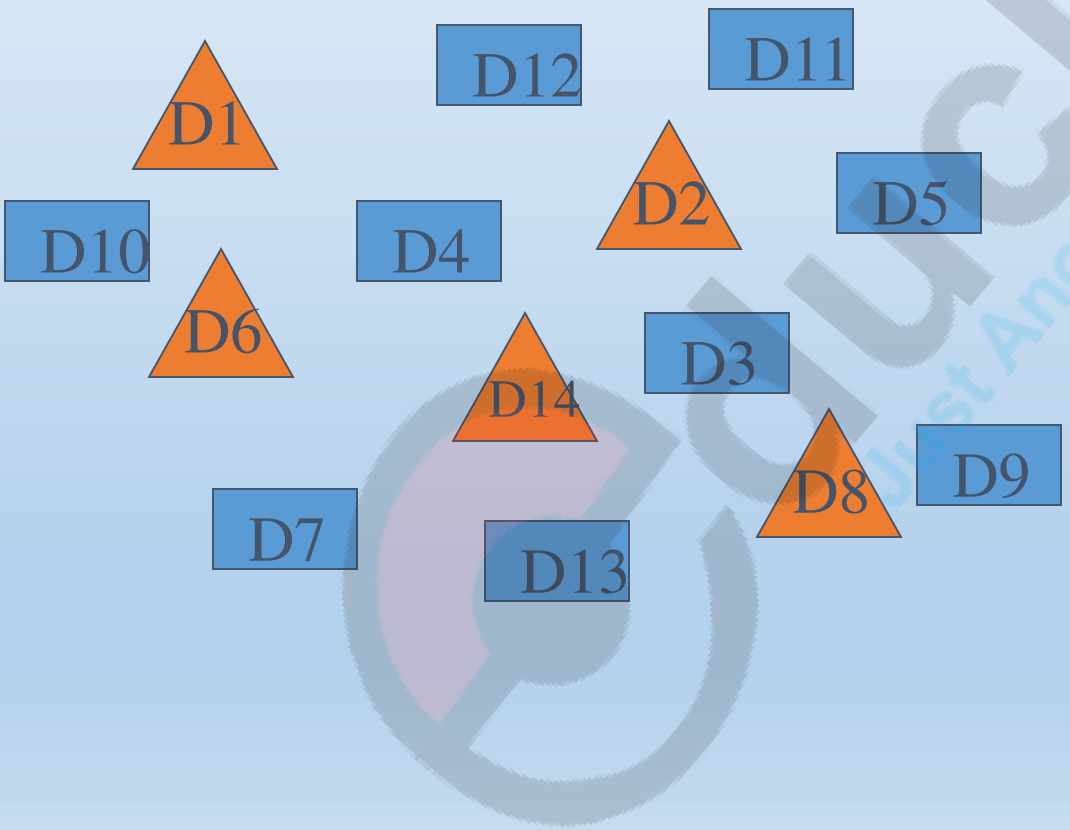


Humidity


S: [9+,5-]
E =0.940
Humidity
High
Normal
[3+,4-]
E =0.985
[6+,1-]
E =0.592
Gain (S, Humidity)
= .940 - (7/14).985 - (7/14).592
= .151

ID3: The Basic Decision Tree Learning Algorithm

- Database, See [Mitchell, p. 59]



What is the "best" attribute?

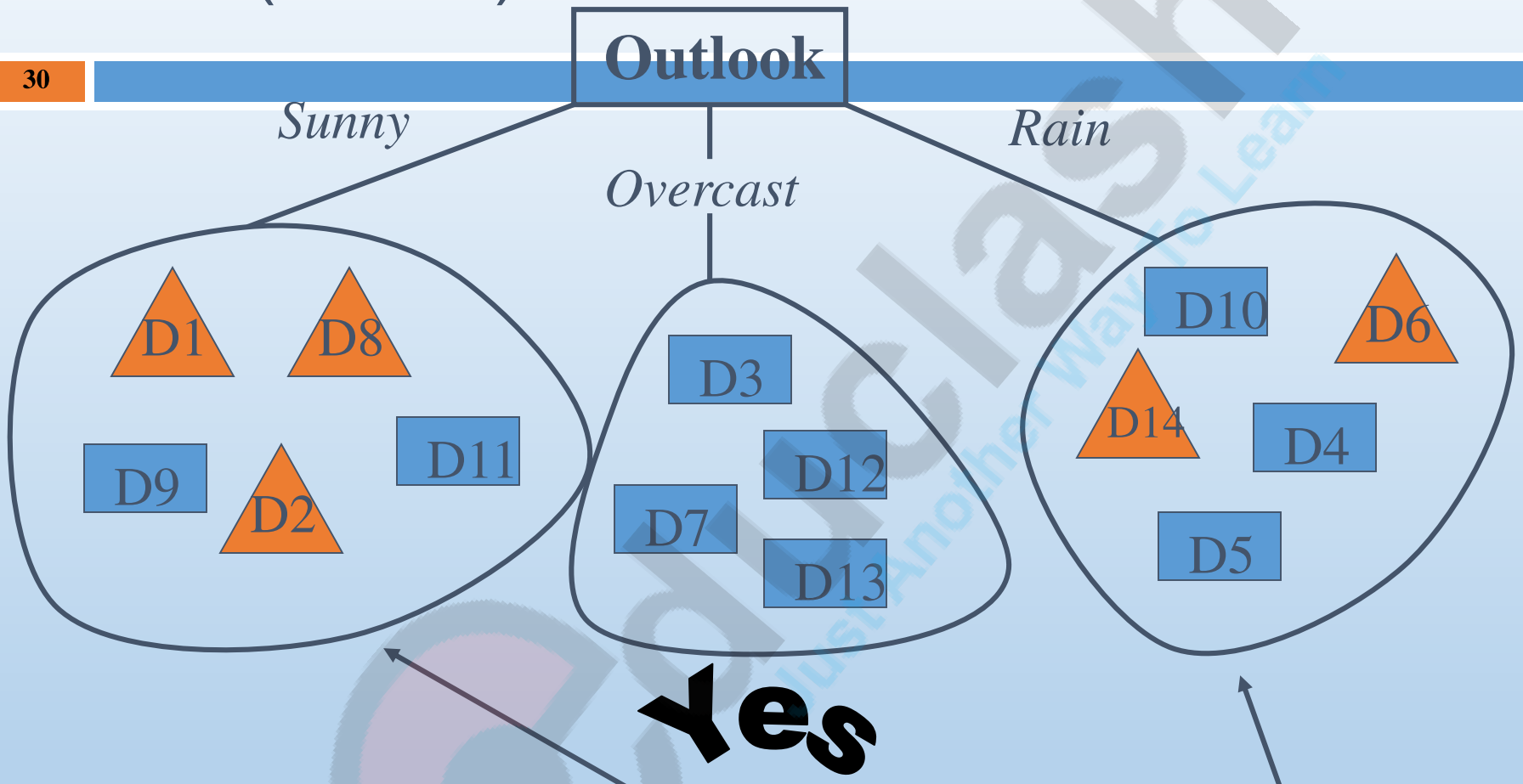
Answer: Outlook

["best" = with highest information gain]

Gains Of All Attributes

$$Gain(S, Outlook) = 0.246$$
$$Gain(S, Humidity) = 0.151$$
$$Gain(S, Wind) = 0.048$$
$$Gain(S, Temperature) = 0.029$$

ID3 (Cont'd)



What are the "best" attributes? **Humidity** and **Wind**

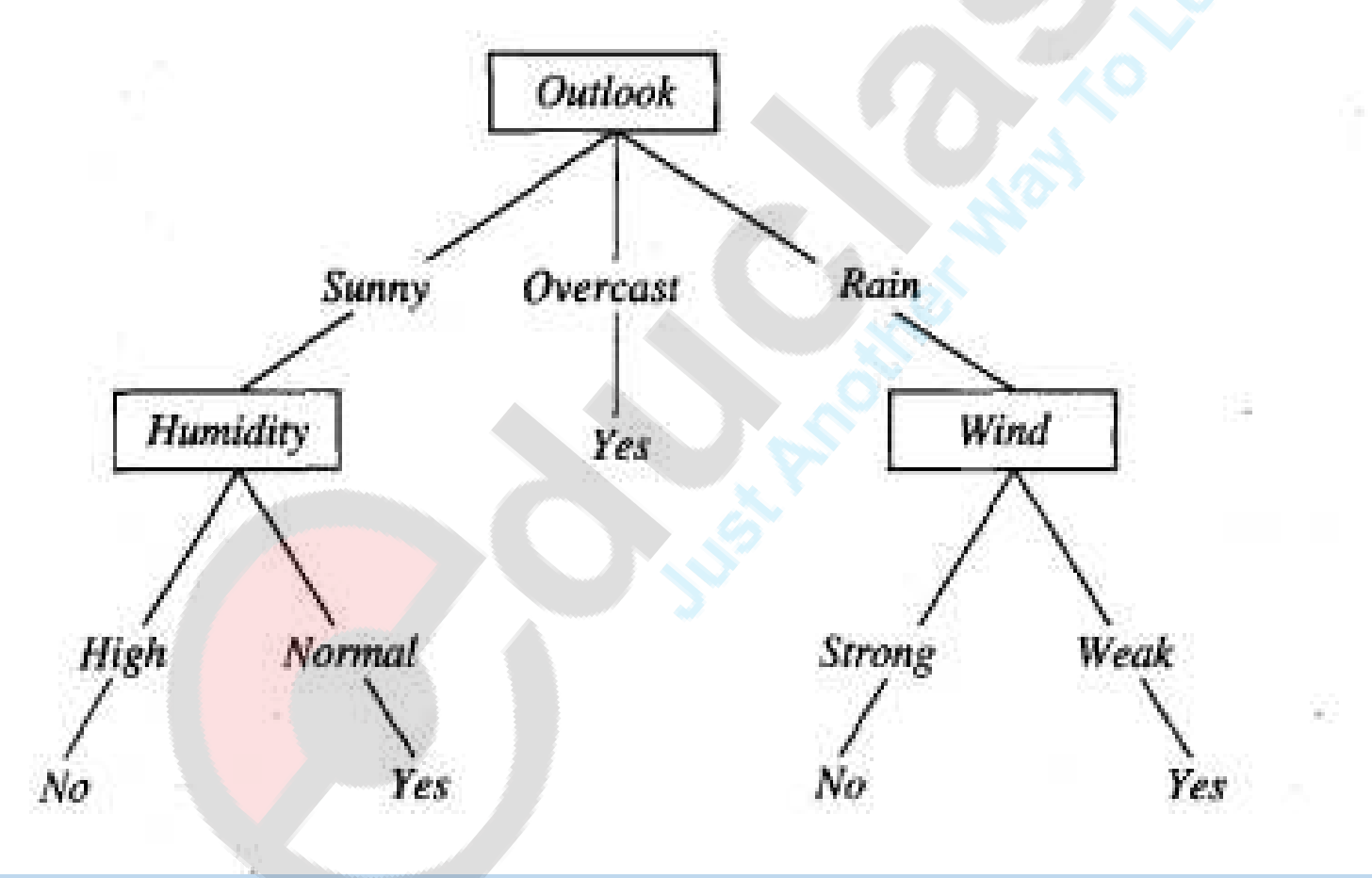
What Attribute to choose to "best" split a node?

- Choose the attribute that minimize the **Disorder (or Entropy**.

- **Disorder** and **Information** are **related** as follows:
- The more disorderly a set, the more information is required to correctly guess an element of that set.

- **Information:** how many questions do you need to ask in order to know the answer .

Tree using ID3



DT Example

- The process of selecting a new attribute and partitioning the training examples is now repeated for each nonterminal descendant node.

- Attributes that have been incorporated higher in the tree are excluded.
- This process continues for each new leaf until:
  1. Every attribute already included along this path through the tree. Or
  2. The training examples associated with this leaf node all have the same target attribute value.

Hypothesis Space Search In Decision Tree Learning

- D3 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.
- The evaluation function that guides this hill-climbing search is the information gain measure.

Hypothesis Space Search in Decision Tree Learning

- **Hypothesis Space**: Set of possible decision trees
- **Search Method**: Simple-to-Complex *Hill-Climbing* Search. **No Backtracking!!!**
- **Evaluation Function**: Information Gain Measure
- **Batch Learning:** ID3 uses all training examples at each step to make statistically-based decisions

ID3 Capabilities & Limitations

- Id3's hypothesis space of all decision trees 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 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.

ID3 Capabilities & Limitations

- ID3 in its pure form performs no backtracking in its search.
- ID3 uses all training examples at each step in the search to make statistically based decisions regarding.
- One advantage of using statistical properties of all the examples (e.g., information gain) is that the resulting search is much less sensitive to errors in individual training examples.

ID3 Capabilities & Limitations

- ID3 in its pure form performs no backtracking in its search.
- ID3 uses all training examples at each step in the search to make statistically based decisions regarding.
- This contrasts with methods that make decisions incrementally, based on individual training examples.
- One advantage of using information gain is that the resulting is much less sensitive to errors in individual training examples.
- ID3 can be easily handle noisy training data.

Inductive Bias In Decision Tree Learning

- That inductive bias is the set of assumptions that, together with the training data, deductively justify the classifications assigned by the learner to future instances.

- Describing the inductive bias of ID3 there fore consists of describing the basis by which it chooses one of these consistent hypotheses over the others.
- The id3 search strategy
- (A) selects in favor of shorter trees over longer ones.
- (B)selects trees that place the attributes with highest information gain closest to the root.

Inductive Bias in Decision Tree Learning

- **ID3's Inductive Bias:** *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.

- **Note:** this type of bias is different from the type of bias used by Candidate-Elimination: the inductive bias of ID3 follows from its search strategy (***preference*** or ***search*** bias) whereas the inductive bias of the Candidate-Elimination algorithm follows from the definition of its hypothesis space (***restriction*** or ***language*** bias).

Inductive Bias in Decision Tree Learning

- Approximate inductive bias of ID3: Shorter trees are preferred over larger trees.
- Because ID3 uses the information gain heuristic and a hill climbing strategy, it exhibits a more complex bias than BFS-ID3.
- **A closer approximation to the inductive bias of ID3: Shorter trees are preferred** over longer trees.

Why Prefer Short Hypotheses?

- **Occam'srazor:**
  Prefer the simplest hypothesis that fits the data

- Scientists seem to do that: E.g., Physicist seem to prefer simple explanations for the motion of planets, over more complex ones

Why Prefer Short Hypotheses?

- **Argument:** Since there are fewer short hypotheses than long ones, it is less likely that one will find a short hypothesis that coincidentally fits the training data.

- **Problem with this argument:** it can be made about many other constraints. Why is the "short description" constraint more relevant than others?

- **Nevertheless:** Occam's razor was shown experimentally to be a successful strategy!

Issues in Decision Tree Learning

- DT grows each branch of the tree just deeply enough to perfectly classify the training examples.
- In fact it can lead to difficulties when there is noise in the data.
- Definition: Given a hypothesis space H, a hypothesis h E H is said to overfit the training data if there exists some alternative hypothesis h' E 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.

Issues in Decision Tree Learning

1. Avoiding Over fitting the Data
   - Reduced Error Pruning
   - Rule Post-pruning
2. Incorporating Continuous-Valued Attributes
3. Alternative Measures for Selecting Attributes
4. Handling Training Examples with Missing Attribute Values .
5. Handling Attributes with Differing Costs.

Issues in Decision Tree Learning: I. Avoiding Overfitting the Data

- **Definition:** Given a hypothesis space H, a hypothesis $h \in H$ is said to ***overfit*** the training data if there exists some alternative hypothesis $h' \in H$, 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. (See curves in [Mitchell, p.67])

Avoiding Over fitting the Data

- Over fitting the training data is an important issue in decision tree learning.
- Because the training examples are only a sample of all possible instances.
- It is possible to add branches to the tree that improve performance on the training examples.
- Methods for post-pruning the decision tree are important to avoid over fitting.

Overfitting

- ✓ Over fitting is a significant practical difficulty for decision tree learning and many other learning methods.
- ✓ over fitting was found to decrease the accuracy of learned decision trees by 10-25% on most problems.
- ✓ There are two approaches for overfitting avoidance in Decision Trees:
  - ✓ Stop growing the tree before it perfectly fits the data
  - ✓ Allow the tree to overfit the data, and then ***post-prune*** it.

$$\langle Outlook = Sunny, Temperature = Hot, Humidity = Normal, Wind = Strong, PlayTennis = No \rangle$$

REDUCED ERROR PRUNING

- Reduced-error Pruning (Quinlan 1987), is to consider each of the decision nodes in the Tree to be candidates for pruning.
- Pruning a decision node consists of removing the sub tree rooted at that node, making it a leaf node, and assigning it the most common classification of the training examples.
- Nodes are removed only if the resulting pruned tree performs is increased.
- Nodes are pruned iteratively, removal should increases the accuracy over the validation set

Rule Post-pruning

- one quite successful method for finding high accuracy hypotheses is a technique we shall call rule post-pruning.

- A variant of this pruning method is used by C4.5 (Quinlan 1993), which is an outgrowth of the original ID3 algorithm

Rule Post-pruning

1. Infer the decision tree from the training set, growing the tree until the training data is fit as well as .
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.

Rule Post-pruning

1. Converting to rules allows distinguishing among the different context.
2. Converting to rules removes the distinction between attribute tests.
3. Converting to rules improves readability. Rules are often easier for to understand.

Incorporating Continuous-Valued Attributes

- First, the target attribute whose value is predicted by the learned tree must be discrete valued.
- Second, the attributes tested in the decision nodes of the tree must also be discrete valued.
- This second restriction can easily be removed so that continuous-valued decision attributes can be incorporated into the learned tree.

Incorporating Continuous-Valued Attributes

- The decision tree have the following values for ***temperature and the target*** attribute ***playtennis.***

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

Alternative Measures for Selecting Attributes

- There is a natural bias in the **information gain** measure that favors attributes with many values over those with few values.
- As an extreme example, consider the attribute Date, which has a very large number of possible values (e.g., March 4, 2015).
- One way to avoid this difficulty is to select decision attributes based on some measure other than information gain.

Alternative Measures for Selecting Attributes

- One alternative measure that has been used successfully is the gain ratio (Quinlan 1986).

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

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

Handling Training Examples with Missing Attribute Values

- The available data may be missing values for some attributes.

Handling Attributes with Differing Costs

- In some learning tasks the instance attributes may have associated costs.
- For example, in learning to classify medical diseases we might describe patients in terms of attributes such **as Temperature, BiopsyResult, Pulse, BloodTestResults,**
- Etc

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

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

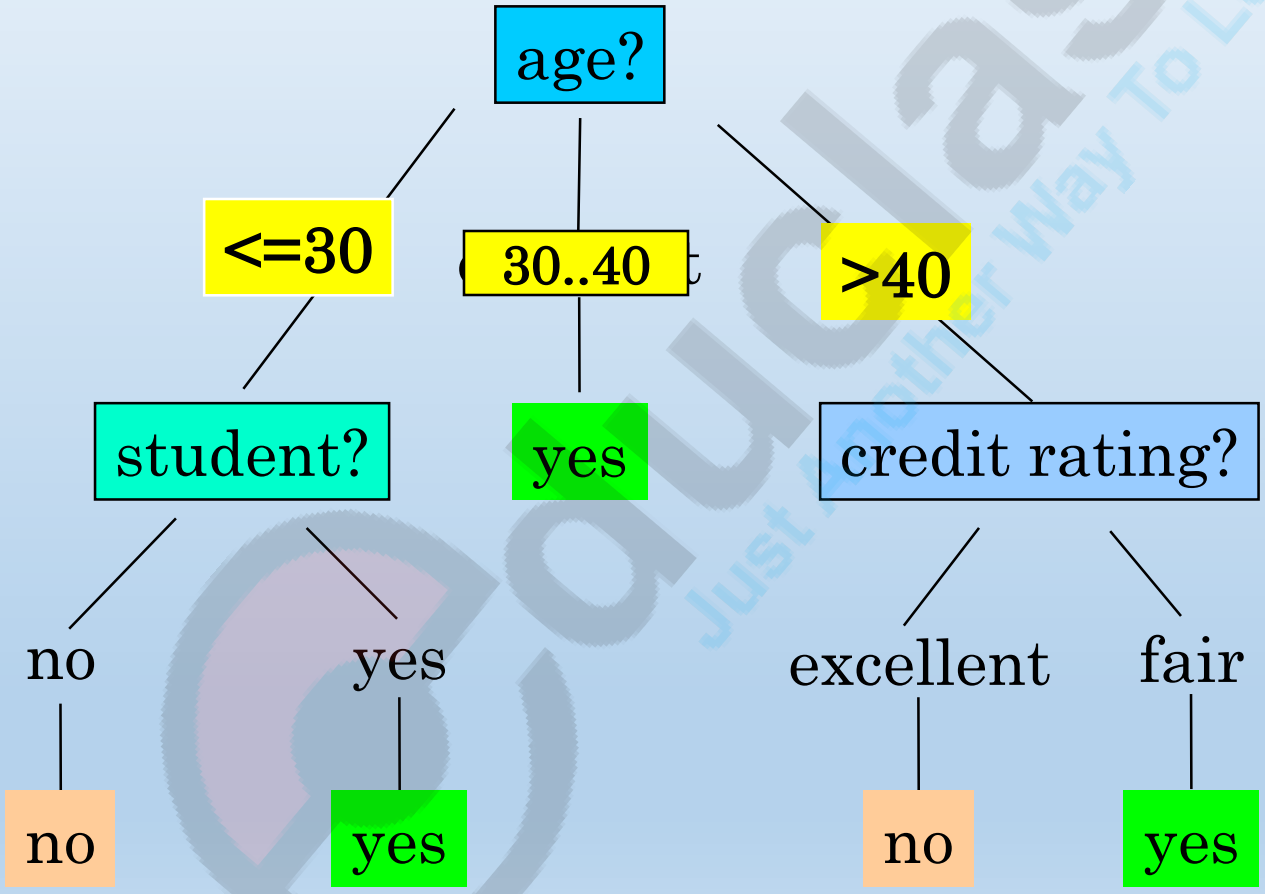
Issues in Decision Tree Learning: II. Other Issues

- Incorporating Continuous-Valued Attributes
- Alternative Measures for Selecting Attributes
- Handling Training Examples with Missing Attribute Values
- Handling Attributes with Differing Costs

Example (From Han and Kamber)

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
30…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

A decision Tree (From Han and Kamber)



An Example (from the text)

Sore Throat	Fever	Swollen Glands	Congestion	Headache	Diagnosis
Yes	Yes	Yes	Yes	Yes	Strep throat
No	No	No	Yes	Yes	Allergy
Yes	Yes	No	Yes	No	Cold
Yes	No	Yes	No	No	Strep throat
No	Yes	No	Yes	No	Cold
No	No	No	Yes	No	Allergy
No	No	Yes	No	No	Strep throat
Yes	No	No	Yes	Yes	Allergy
No	Yes	No	Yes	Yes	Cold
Yes	Yes	No	Yes	Yes	Cold

- First five attributes are symptoms and the last attribute is diagnosis. All attributes are categorical.
- Wish to predict the diagnosis class.