Lecture 5
Decision Tree Learning

1. **Introduction**
2. **Decision Tree Representation**
3. **The Basic Decision Tree Learning Algorithm**

1. Introduction

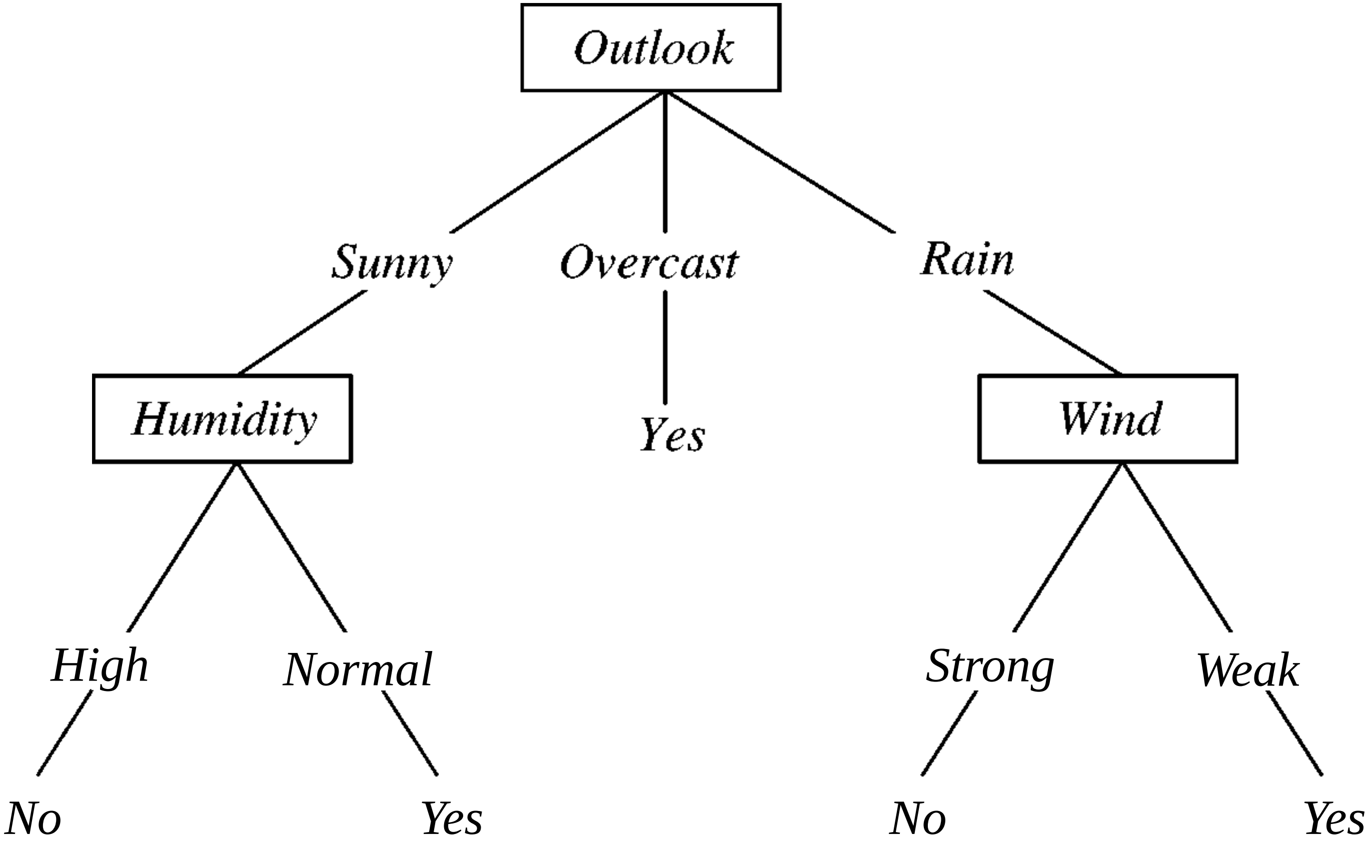
- Decision tree learning is one of the most widely used and practical methods for inductive inference.
- It is a method for approximating discrete-valued functions that is robust to noisy data and capable of learning disjunctive expressions.

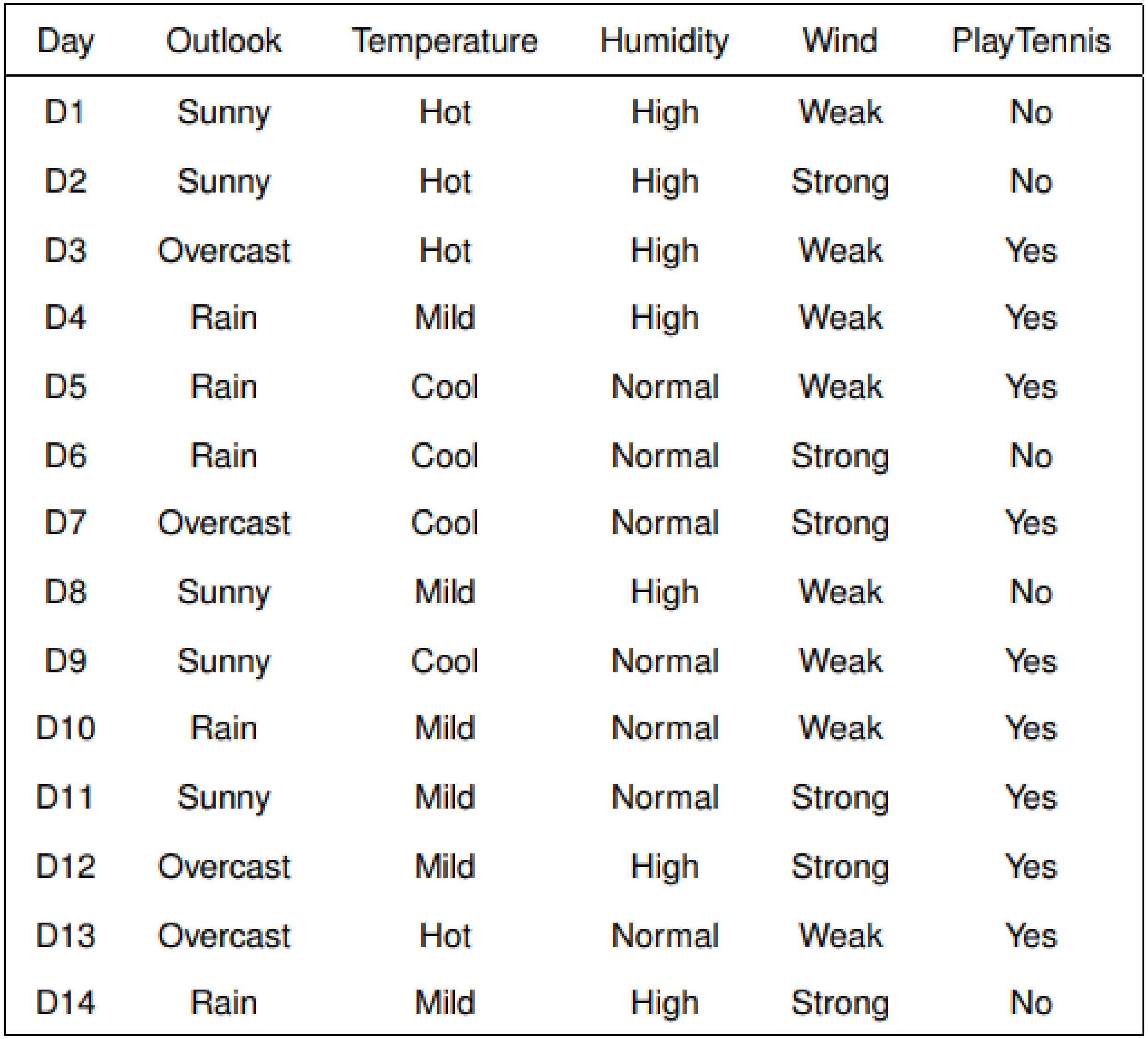
- The learned function is represented by a decision tree.
- Learned trees can also be represented as sets of if-then rules.
- It has been applied to a broad range of tasks from learning to diagnose medical cases to learning to assess credit risk of loan applicants.

2. Decision Tree Representation

- 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 of 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 repeated for the subtree rooted at the new node.

Example - 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

- A negative instance (i.e. the tree predicts that PlayTennis = no): (Outlook = Sunny, Temperature = Hot, Humidity = High, Wind = Strong)
- Positive Instance?
- In General, Decision trees represent a disjoint of conjunctions of constraints on the attribute value 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.

Appropriate Problems for Decision Tree

1. Instances describable by attribute-value pairs : e.g. attributes (Temperature) and their values (Hot).
2. Target function has discrete output values: e.g. boolean classification (yes or no).
3. Disjunctive hypothesis may be required: decision trees naturally represent disjunctive expressions.
4. Possibly noisy training data
5. Missing attribute values in trading data

Practical Problems

1. Equipment or medical diagnosis
2. equipment malfunctions by their cause
3. Credit risk analysis
4. Modeling calendar scheduling preferences

2. The Basic Decision Tree Learning Algorithm

- Top-down greedy search through the space of possible decision trees (ID3 algorithm and C4.5 Algorithm)
- “Which attribute should be tested at the root of the tree?".
- The best attribute is selected and used as the test at the root node of the tree.

Which Attribute Is the Best Classifier?

- Select the attribute that is most useful for classifying examples.
- “**Information gain**” 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.

Entropy Measures Homogeneity of Examples

- 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:

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

- $p_{\oplus}$ is the proportion of positive examples in *S*
- $p_{\ominus}$ is the proportion of negative examples in *S*

Entropy Example

- Suppose *S* is a collection of 14 examples of some Boolean concept, including 9 positive and 5 negative examples. [9+, 5-]

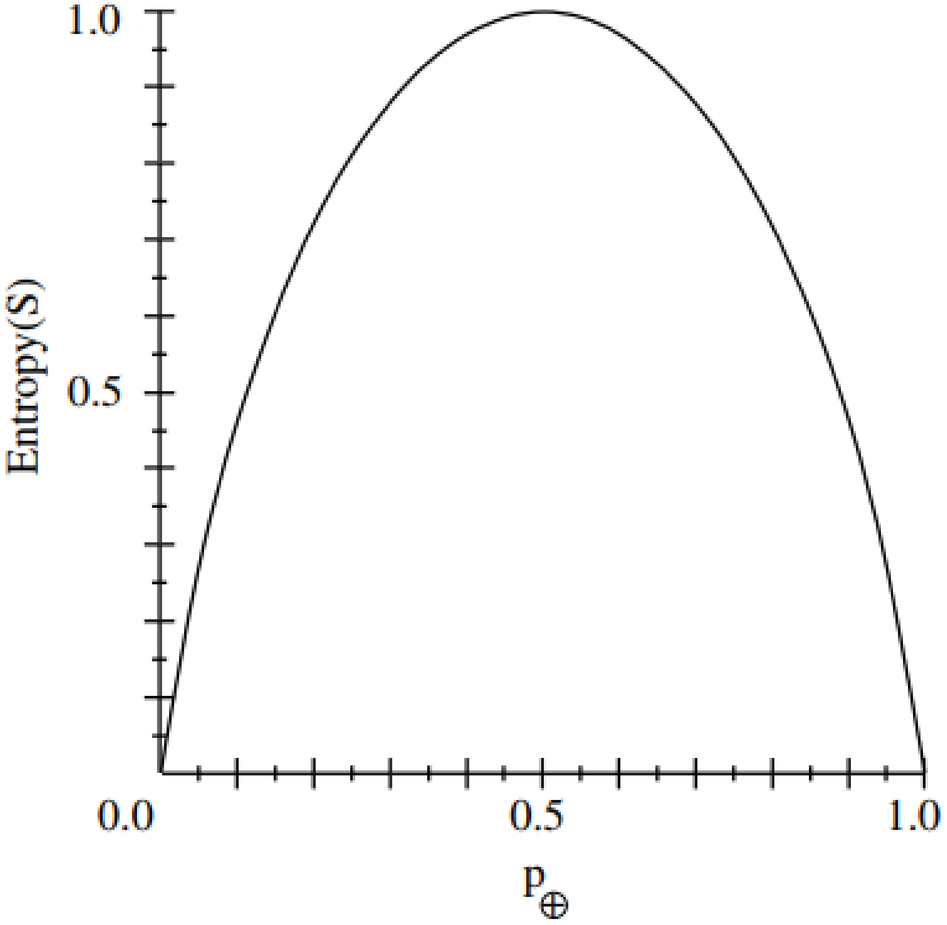
- The entropy of *S* relative to this Boolean classification is:

$$Entropy(\mathrm{S}) = -(9/14)\log_2(9/14) - (5/14)\log_2(5/14)$$

$$=0.940$$

- The entropy is 0 if all members of *S* belong to the same class.
- e.g. if all members of positive ($p_{\oplus}$ =1), then $p_{\ominus}$ is 0 and,
    - *Entropy(S*) = - (1) $\log_2(1)$ - (0) $\log_2 0$ = 0
- The entropy is 1 when the collection contains an equal number of positive and negative examples.
- If the collection contains unequal numbers of positive and negative examples, the entropy is between 0 and 1.

The Entropy Function Relative to a Boolean Classification



General Case

- If the target attribute can take on *c* different values, then the entropy of S relative to this c-wise classification is defined as:

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

- where p_i is the proportion of S belonging to class *i*

Information Gain Measures the Expected Reduction in Entropy

- **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 collection of examples S:

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

- where *V alues(A)* is the set of all possible values for attribute A, and S_v is the subset of *S* for which A has value *v*.

Example (Attribute Wind)

$$Values(Wind) = Weak, Strong$$

$$S = [9+, 5-]$$

$$S_{Weak} \leftarrow [6+, 2-]$$

$$S_{Strong} \leftarrow [3+, 3-]$$

$$Gain(S, Wind) \equiv Entropy(S) - \sum_{v \in \{Weak, Strong\}} \frac{|S_v|}{|S|} Entropy(S_v)$$

Example (Attribute Wind) - cont

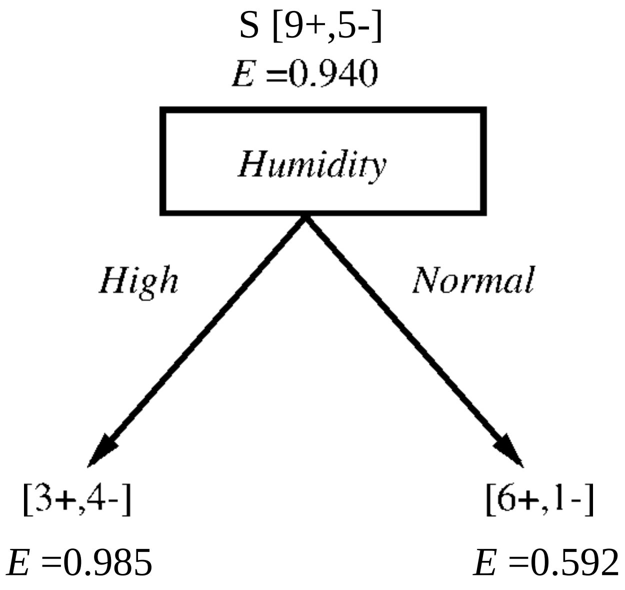
$$Entropy(S_{Weak}) = -(6/8)\log_2(6/8) - (2/8)\log_2(2/8) = 0.811$$

$$Entropy(S_{Strong}) = -(3/6)\log_2(3/6) - (3/6)\log_2(3/6) = 1.00$$

Example (Attribute Wind) - cont

$$
\begin{aligned}
Gain(S, Wind) &\equiv Entropy(S) - \sum_{v \in \{Weak, Strong\}} \frac{|S_v|}{|S|} Entropy(S_v) \\
&= Entropy(S) - (8/14) Entropy(S_{Weak}) \\
&\quad - (6/14) Entropy(S_{Strong}) \\
&= 0.940 - (8/14)0.811 - (6/14)1.00 \\
&= 0.048
\end{aligned}
$$

Which attribute is the best classifier?



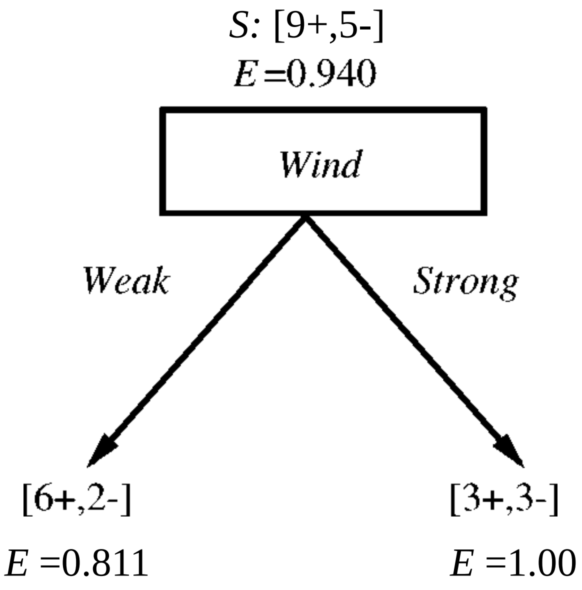
S [9+,5-]
E =0.940

Humidity

High *Normal*

[3+,4-] [6+,1-]

E =0.985 *E* =0.592



S: [9+,5-]
E =0.940

Wind

Weak *Strong*

[6+,2-] [3+,3-]

E =0.811 *E* =1.00

Gain (S, Humidity)

=.940 - (7/14).985 - (7/14).592
=.151

Gain (S, Wind)

=.940 - (8/14).811 - (6/14)1.0
=.048

Example (all attribute) - cont

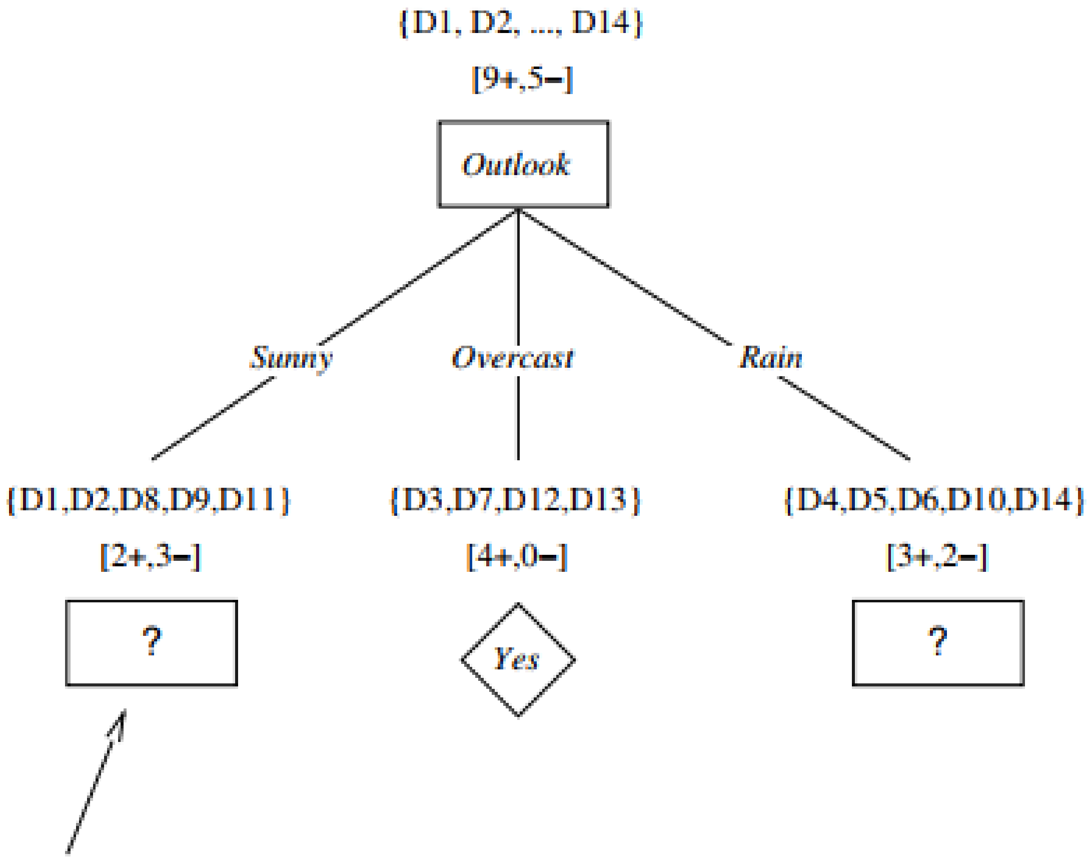
$$Gain(S, Outlook) = 0.246$$

$$Gain(S, Humidity) = 0.151$$

$$Gain(S, Wind) = 0.048$$

$$Gain(S, Temperature) = 0.029$$

- Attribute Outlook is selected as the decision attribute for the root node and branches are created below the root for each of its possible values.



{D1, D2, ..., D14}

[9+,5−]

Outlook

Sunny *Overcast* *Rain*

{D1,D2,D8,D9,D11}

[2+,3−]

?

{D3,D7,D12,D13}

[4+,0−]

Yes

{D4,D5,D6,D10,D14}

[3+,2−]

?

Which attribute should be tested here?

S_{sunny} = {D1,D2,D8,D9,D11}

$Gain\ (S_{sunny}\ ,\ Humidity)\ =\ .970\ -\ (3/5)\ 0.0\ -\ (2/5)\ 0.0\ =\ .970$

$Gain\ (S_{sunny}\ ,\ Temperature)\ =\ .970\ -\ (2/5)\ 0.0\ -\ (2/5)\ 1.0\ -\ (1/5)\ 0.0\ =\ .570$

$Gain\ (S_{sunny},\ Wind)\ =\ .970\ -\ (2/5)\ 1.0\ -\ (3/5)\ .918\ =\ .019$

- Note that every example for which *Outlook* = *Overcast* is also a positive example. Therefore, this node of the tree becomes a leaf node with the classification *PlayTennis* = *Yes*.
- The descendants corresponding to *Outlook* = *Sunny* and *Outlook* = *Rain* still have nonzero entropy, and the decision tree will be further elaborated below these nodes.
- The process of selecting a new attribute and partitioning the training examples is repeated for each nonterminal descendant node, using only the training examples associated with that node.
- Attributes that have been incorporated higher in the tree are excluded.

- This process continues for each new leaf node until either of two conditions is met:

1. Every attribute has already been included along this path through the tree, or
2. The training examples associated with this leaf node all have the same target attribute value (i.e. their entropy is zero).