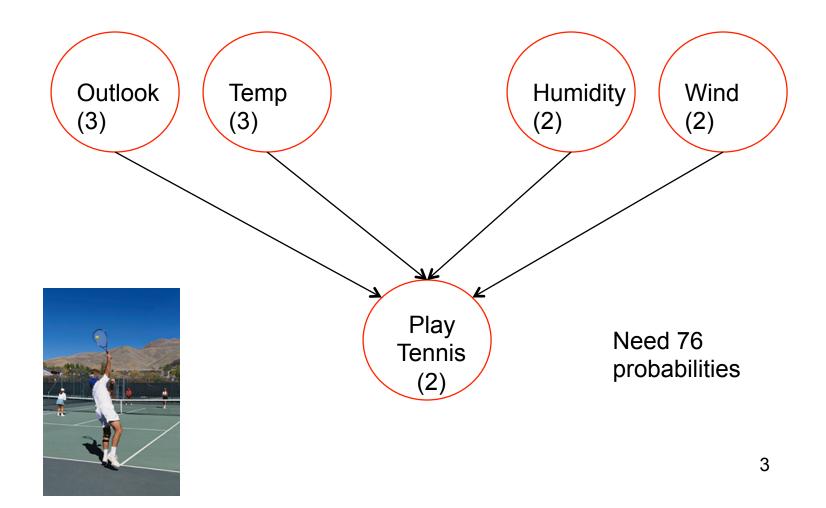
### **Decision Trees**

Sargur Srihari srihari@cedar.buffalo.edu

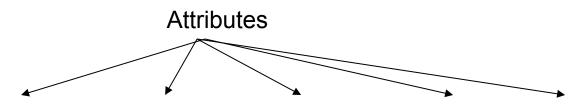
## Decision Tree Learning and Inductive Inference

- Widely used method for inductive inference
- Inductive Inference Hypothesis:
  - Any hypothesis found to approximate the target function well over a sufficiently large set of training examples will also approximate the target function well over the unobserved examples
- Inductive bias is a preference for small trees over large trees

### **Graphical Model**



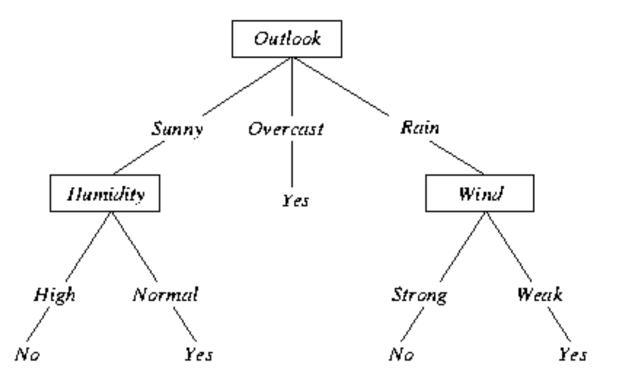
### **Learning Data**



Day	Outlook	Temp	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

### Decision Tree Example

- PlayTennis
- Learned function is a tree



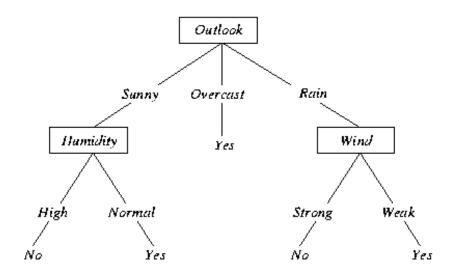


### **Decision Tree Learning**

- Approximates discrete-valued target functions as trees
- Robust to noisy data and capable of learning disjunctive expressions
- A family of decision tree learning algorithms includes ID3, ASSISTANT and C4.5
- Use a completely expressive hypothesis space

### **Decision Tree Classification**

- Classify instances
  - by sorting them down from the root to the leaf node,
- Each node specifies a test of an attribute
- Each branch descending from a node corresponds a possible value of this attribute

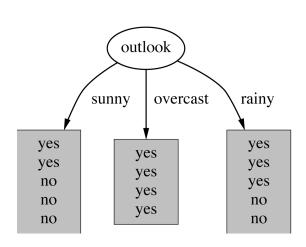


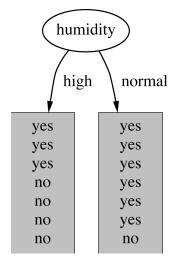
Srihari

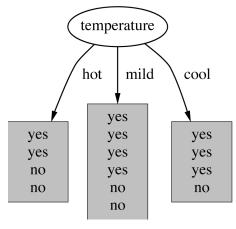
# Learning the Play Tennis Decision Tree

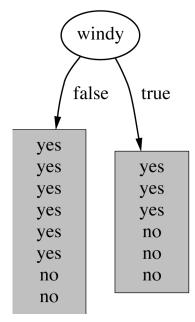
#### Four Attributes Humidity Wind **PlayTennis** Outlook Temp Day No D1 Sunny Hot High Weak D2 Sunny Hot High Strong No D3 High Weak **Overcast** Hot Yes **D**4 Rain Mild High Weak Yes D5 Normal Weak Rain Cool Yes D6 Cool Normal No Rain Strong D7 Cool Normal Strong **Overcast** Yes D8 Mild High Weak No Sunny D9 Cool Normal Weak Yes Sunny D10 Rain Mild Normal Weak Yes D11 Sunny Mild Normal Strong Yes D12 **Overcast** Mild High Strong Yes D13 Hot Normal Weak **Overcast** Yes Rain Mild Strong No D14 High

## Four Tree Stumps for PlayTennis









### A good attribute

- An attribute is good when:
  - for one value we get all instances as positive
  - for other value we get all instances as negative

#### Poor attribute

- An attribute is poor when:
  - it provides no discrimination
  - attribute is immaterial to the decision
  - for each value we have same number of positive and negative instances

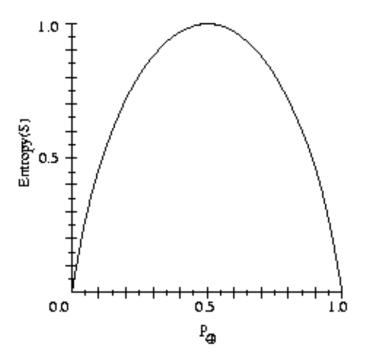
#### Entropy: Measure of Homogeneity of Examples

- •Entropy: Characterizes the (im)purity of an arbitrary collection of examples
- •Given a collection *S* of positive and negative examples, entropy of *S* relative to boolean classification is

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

Where p+ is proportion of positive examples and p- is proportion of negative examples

# Entropy Function Relative to a Boolean Classification



### **Entropy**

- Illustration:
- S is a collection of 14 examples with 9 positive and 5 negative examples
- Entropy of S relative to the Boolean classification:
  - Entropy  $(9+, 5-) = -(9/14)\log_2(9/14) (5/14)\log_2(5/14)$ = 0.940
- Entropy is zero if all members of S belong to the same class

# Entropy for multi-valued target function

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

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

#### Information Gain

- Entropy measures the impurity of a collection
- Information Gain is defined in terms of Entropy
  - expected reduction in entropy caused by partitioning the examples according to this attribute

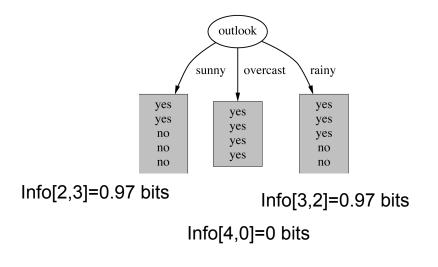
## Information Gain Measures the Expected Reduction in Entropy

 Information gain of attribute A is the reduction in entropy caused by partitioning the set of examples S

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

• where Values (A) is the set of all possible values for attribute A and  $S_v$  is the subset of S for which attribute A has value v

### Measure of purity: information (bits)



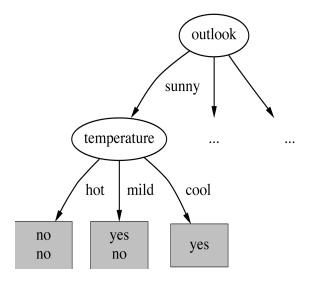
```
Info[2,3]=entropy(2/5, 3/5)
= -2/5 log 2/5 - 3/5 log 3/5
= 0.97 bits
```

```
Average info of subtree(weighted)= 0.97x5/14 + 0 \times 4/14 + 0
```

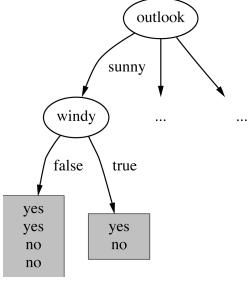
Info of all training samples, info[9,5] = 0.94

gain(outlook) = 0.94 - 0.693 = 0.247 bits

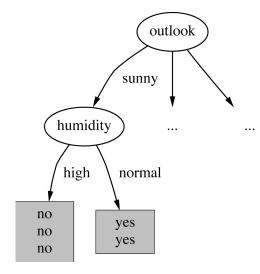
### Expanded Tree Stumps for PlayTennis for outlook=sunny



Info(2,3)=0.97 Info(0,2)=info(1,0)=0 Info(1,1)=0.5 Ave Info=0+0+(1/5) Gain(temp)=0.97- 0.2 =0.77 bits



Info(3,3)=1 Info(2,2)=Info(1,1)=1 Gain(windy)=1-1 =0 bits



Info (3,2)= 0.97 Info (0,3)=Info(2,0)=0 Gain(humidity)=0.97 bits

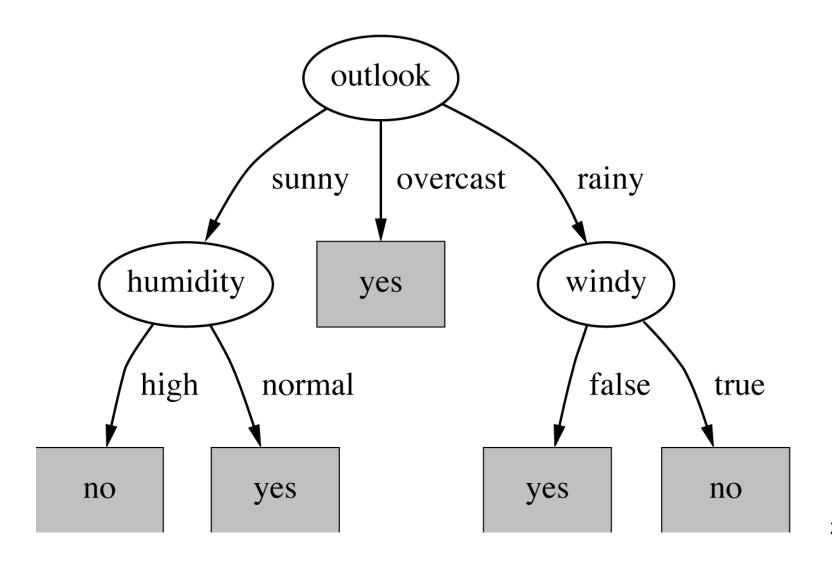
19

Since Gain(humidity) is highest, select humidity as splitting attribute. No need to split further

# Information gain for each attribute

- Gain(outlook) = 0.94 0.693 = 0.247
- Gain(temperature)= 0.94 0.911 = 0.029
- Gain(humidity)= 0.94 0.788 = 0.152
- Gain(windy) = 0.94 0.892 = 0.048
- arg Max {0.247, 0.029, 0.152, 0.048} = outlook
- Select outlook as the splitting attribute of tree

#### Decision Tree for the Weather Data



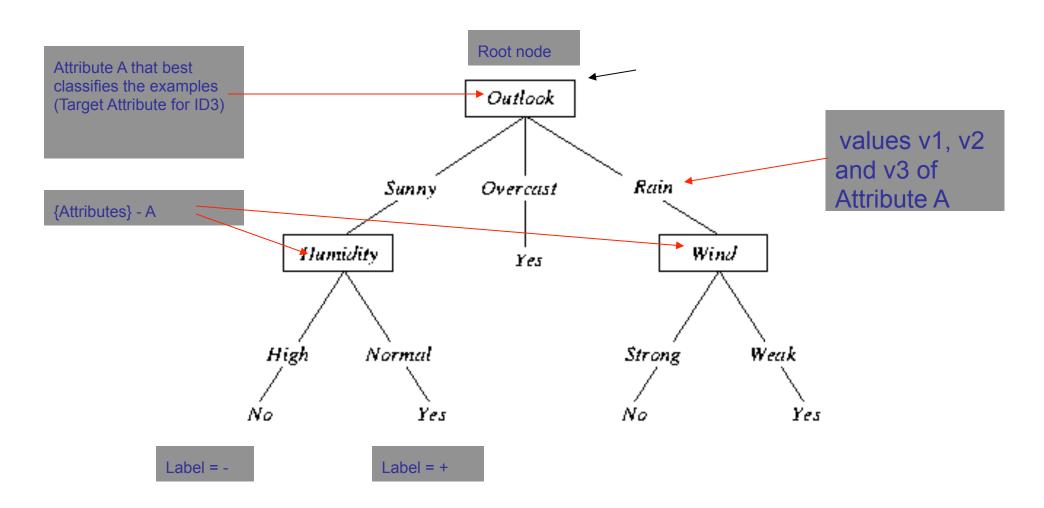
# The Basic Decision Tree Learning Algorithm (ID3)

- Top-down, greedy search (no backtracking) through space of possible decision trees
- Begins with the question
  - "which attribute should be tested at the root of the tree?"
- Answer
  - evaluate each attribute to see how it alone classifies training examples
- Best attribute is used as root node
  - descendant of root node is created for each possible value of this attribute

# Which Attribute Is Best for the Classifier?

- Select attribute that is most useful for classification
- ID3 uses Information gain as a quantitative measure of an attribute
- Information Gain: A statistical property that measures how well a given attribute separates the training examples according to their target classification.

### **ID3** Algorithm Notation



#### ID3 Algorithm to learn boolean-valued functions

#### ID3 (Examples, Target\_attribute, Attributes)

Examples are the training examples. Target\_attribute is the attribute (or feature) 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 (actually the root node of the 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 = the most common value of *Target\_attribute* in *Examples*
- \* %Note that we will return the name of a feature at this point

### ID3 Algorithm, continued

#### ·Otherwise Begin

- A the attribute from Attributes that best\* classifies Examples
- The decision attribute (feature) for *Root* 🖼 A
- · For each possible value  $v_i$ , of A,
  - · Add a new tree branch below *Root*, corresponding to test  $A = v_i$
  - Let Examples<sub>vi</sub> the subset of *Examples* that have value  $v_i$  for A
  - · If Examples<sub>vi</sub> 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

#### ·End

·Return Root

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

•The best attribute is the one with the highest information gain, as defined in Equation:

#### Perfect feature

- If feature outlook has two values: sunny and rainy
- If for sunny all 5 values of playtennis are yes
- If for rainy all 9 values of playtennis are no
- Gain(S,outlook) = 0.94 (5/9).0 (9/9).0
- $\bullet$  = 0.94

# Training Examples for Target Concept *PlayTennis*

Day	Outlook	Temp	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

Table 3.2

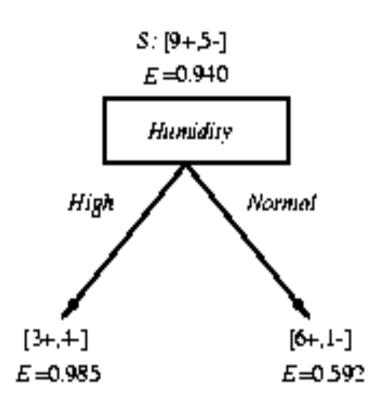
# Stepping through ID3 for the example

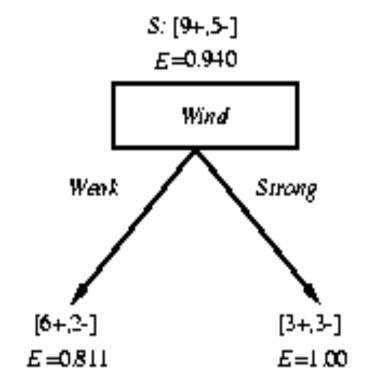
- Top Level (with S={D1,..,D14})
  - Gain(S, Outlook) = 0.246
  - Gain(S, Humidity)= 0.151
  - Gain(S,Wind) = 0.048
  - Gain(S,Temperature) = 0.029
- Example computation of Gain for Wind
  - Values(Wind) = Weak, Strong
  - -S = [9+,5-]
  - $-S_{\text{weak}} < --[6+,2-], S_{\text{strong}} < --[3+,3-]$
  - Gain(S,Wind) = 0.940 (8/14)0.811 (6/14)1.00 = 0.048

Best prediction of

target attribute

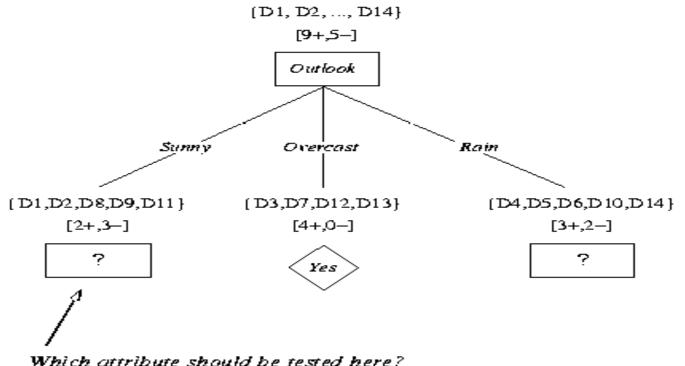
#### Sample calculations for Humidity and Wind





Srihari Machine Learning

#### The Partially Learned Decision Tree



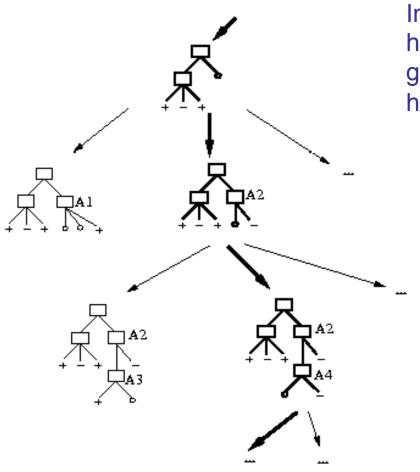
```
S_{SUMNV} = \{D1, D2, D8, D9, D11\}
  Gain(S_{sunny}, Humidity) = .970 - (3/5)0.0 - (2/5)0.0 = .970
  Gain(S_{sunny}, Temperature) = .970 - (2/5) 0.0 - (2/5) 1.0 - (1/5) 0.0 = .570
  Gain(S_{SURRY}, Wind) = .970 - (2.5) 1.0 - (3.5) .918 = .019
```

Humidity is a perfect feature since there is no uncertainty left when we know its value31

### Hypothesis Space Search in Decision Tree Learning

- ID3 searches a hypothesis space for one that fits training examples
- Hypothesis space searched is set of possible decision trees
- ID3 performs hill-climbing, starting with empty tree, considering progressively more elaborate hypotheses (to find tree to correctly classify training data)
- Hill climbing is guided by evaluation function which is the gain measure

# Hypothesis Space Search by ID3

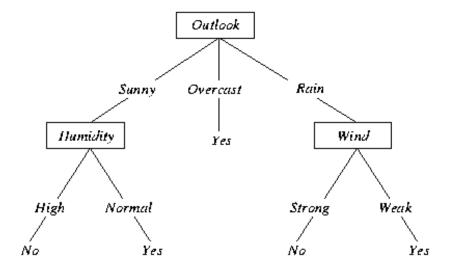


Information gain heuristic guides search of hypothesis space by ID3

Figure 3.5

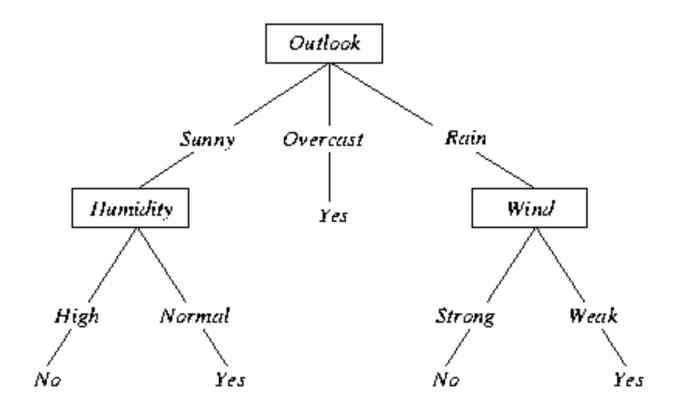
# Decision Trees and Rule-Based Systems

 Learned trees can also be re-represented as sets of *if-then* rules to improve human readability



## Decision Trees represent disjunction of conjunctions

(Outlook=Sunny ^ Humidity=Normal) v (Outlook=Overcast)v (Outlook=Rain ^ Wind=Weak)



### Appropriate Problems for Decision Tree Learning

- Instances are represented by attribute-value pairs
  - each attribute takes on a small no of disjoint possible values, eg, hot, mild, cold
  - extensions allow real-valued variables as well, eg temperature
- The target function has discrete output values
  - eg, Boolean classification (yes or no)
  - easily extended to multiple-valued functions
  - can be extended to real-valued outputs as well

# Appropriate Problems for Decision Tree Learning (2)

- Disjunctive descriptions may be required
  - naturally represent disjunctive expressions
- The training data may contain errors
  - robust to errors in classifications and in attribute values
- The training data may contain missing attribute values
  - eg, humidity value is known only for some training examples

# Appropriate Problems for Decision Tree Learning (3)

- Practical problems that fit these characteristics are:
  - learning to classify
    - medical patients by their disease
    - equipment malfunctions by their cause
    - loan applications by by likelihood of defaults on payments

# Capabilities and Limitations of ID3

- Hypothesis space is a complete space of all discrete valued functions
- Cannot determine how many alternative trees are consistent with training data (follows from maintaining a single current hypothesis)
- ID3 in its pure form performs no backtracking (usual risks of hill-climbing- converges to local optimum)
- ID3 uses all training examples at each step to make statistically based decisions regarding how to refine its current hypothesis
  - more robust than Find-S and Candidate
     Elimination which are incrementally-based