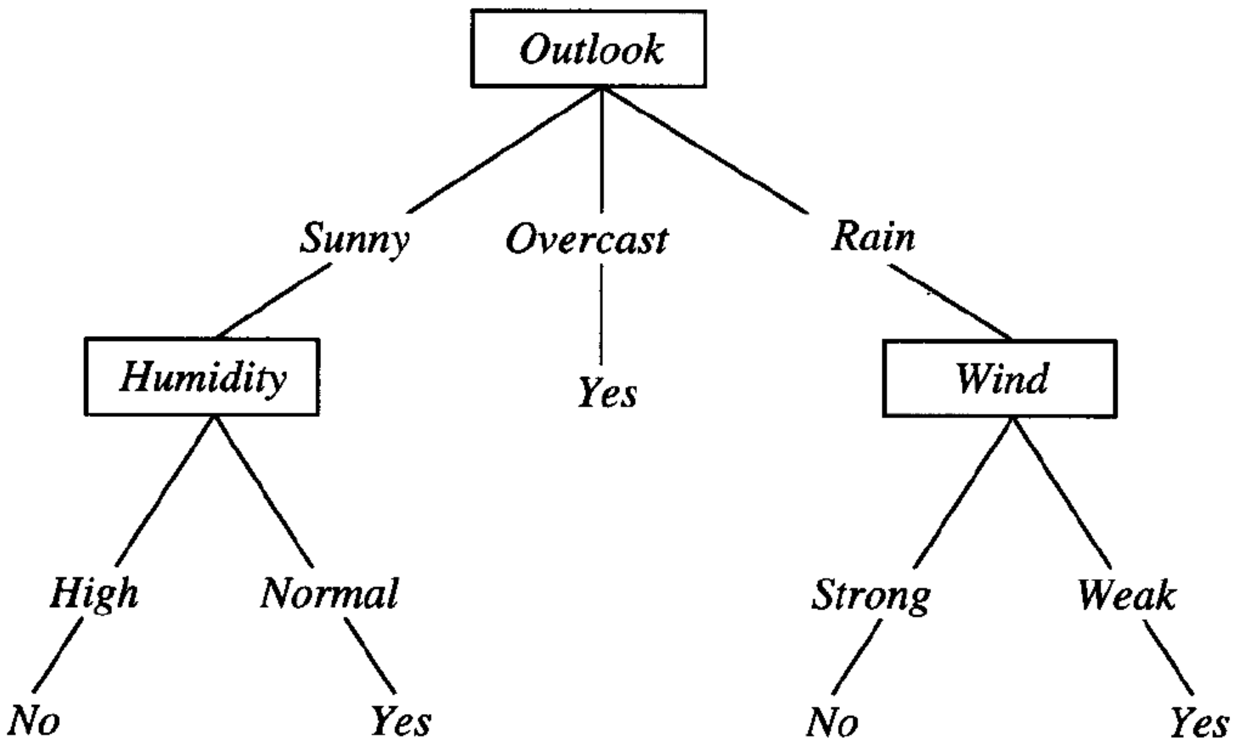
3. Decision Tree Learning

- Method for approximation of discrete-valued target functions (classification)
- One of the most widely known method for inductive inference
- Base for advanced methods

Example: PlayTennis

Day	Outlook	Temperature	Humidity	Wind	PlayTenn
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 for PlayTennis



Decision Tree Representation

- Each node tests some attribute of the instance
- Decision trees represent a disjunction of conjunctions of constraints on the attributes

Example:

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

Appropriate Problems for DTL

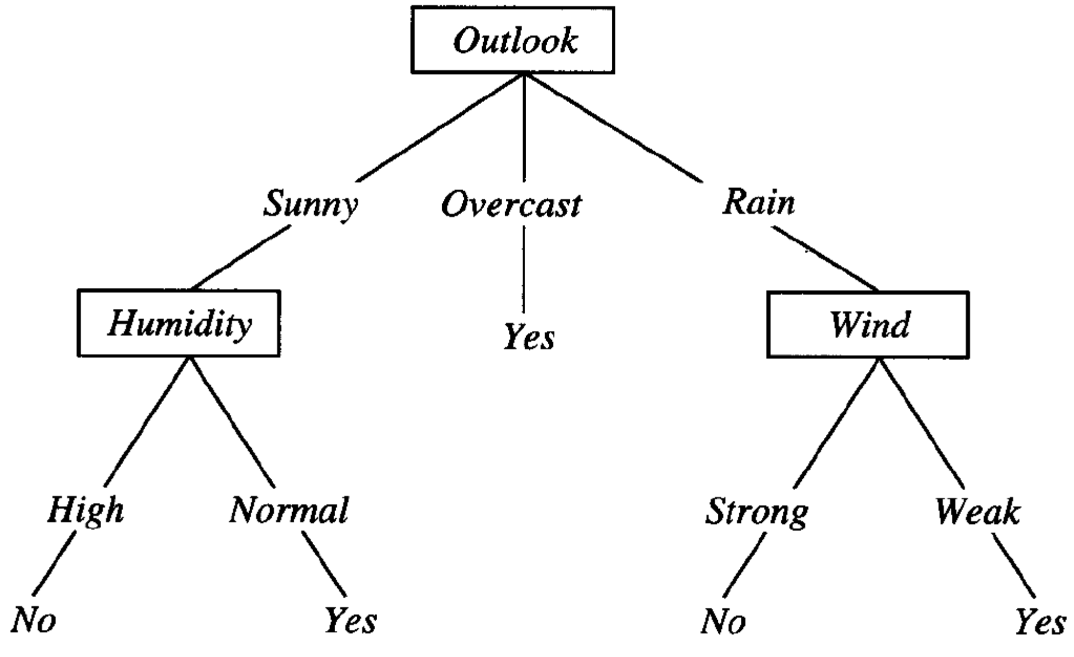
- Instances are represented by attribute-value pairs
- The target function has discrete output values
- Disjunctive descriptions may be required
- The training data may contain errors
- The training data may contain missing attributes values

The Basic DTL Algorithm

– Start, progress, stop

The Basic DTL Algorithm

- Start, progress, stop



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*

All examples?

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 $Root$

The Basic DTL Algorithm

- Start, progress, stop
- Root: best attribute for classification

Which attribute is the best classifier?

$\Rightarrow$ answer based on information gain

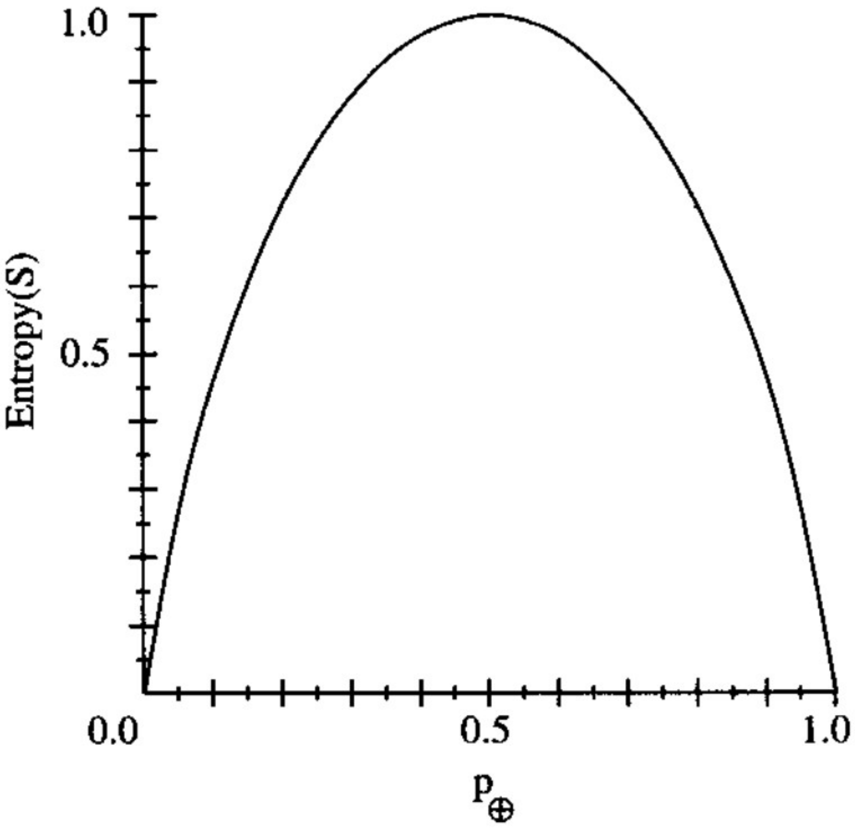
Entropy

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

$p_{+(-)}$ = proportion of positive (negative) examples

- Entropy specifies the minimum number of bits of information needed to encode the classification of an arbitrary member of *S*
- In general: $Entropy(S) = - \sum_{i=1,c} p_i \log_2 p_i$

Entropy



Entropy

$$\begin{aligned} Entropy([9+, 5-]) &= -(9/14)\log_2(9/14) - (5/14)\log_2(5/14) \\ &= 0.940 \end{aligned}$$

Information Gain

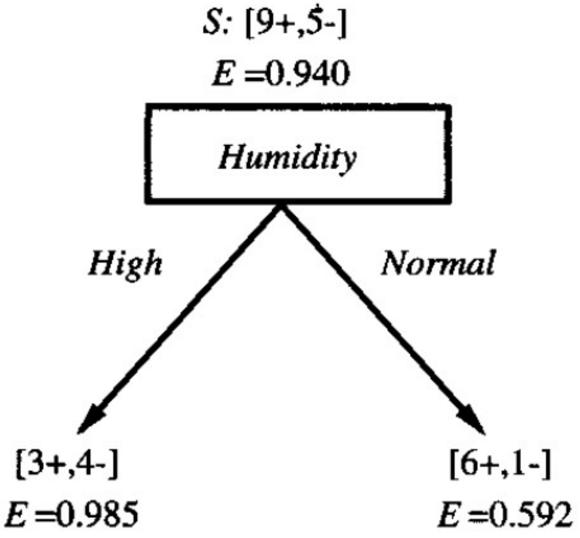
- Measures the expected reduction in entropy given the value of some attribute A

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

Values(A): Set of all possible values for attribute A

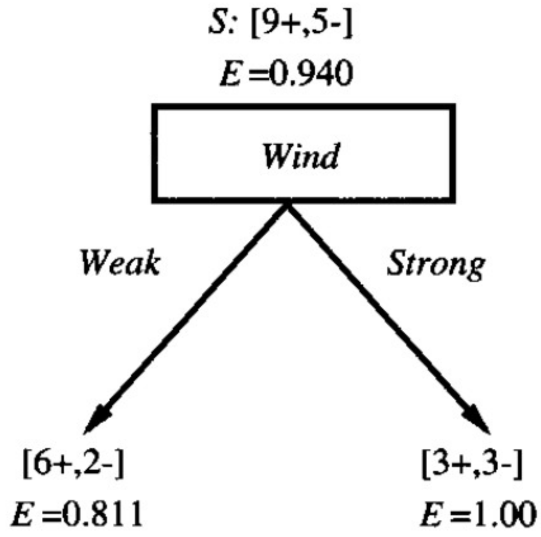
S_v: Subset of S for which attribute A has value v

Selecting the Root Attribute



Gain (S, Humidity)

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



Gain (S, Wind)

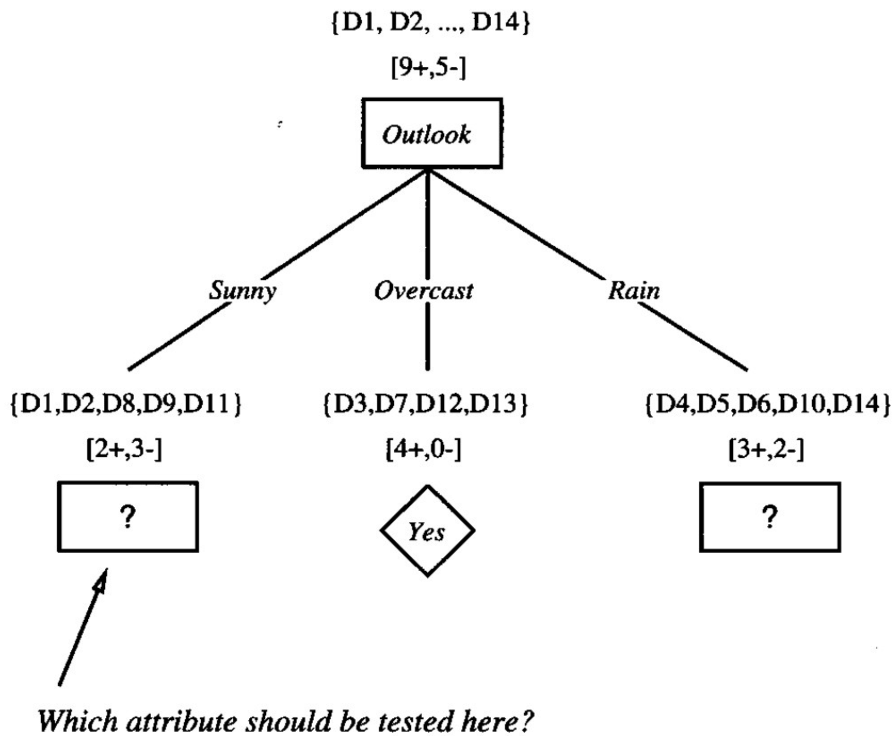
= .940 - (8/14).811 - (6/14)1.0
= .048

PlayTennis Problem

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

⇒ Outlook is the attribute of the root node

PlayTennis Problem



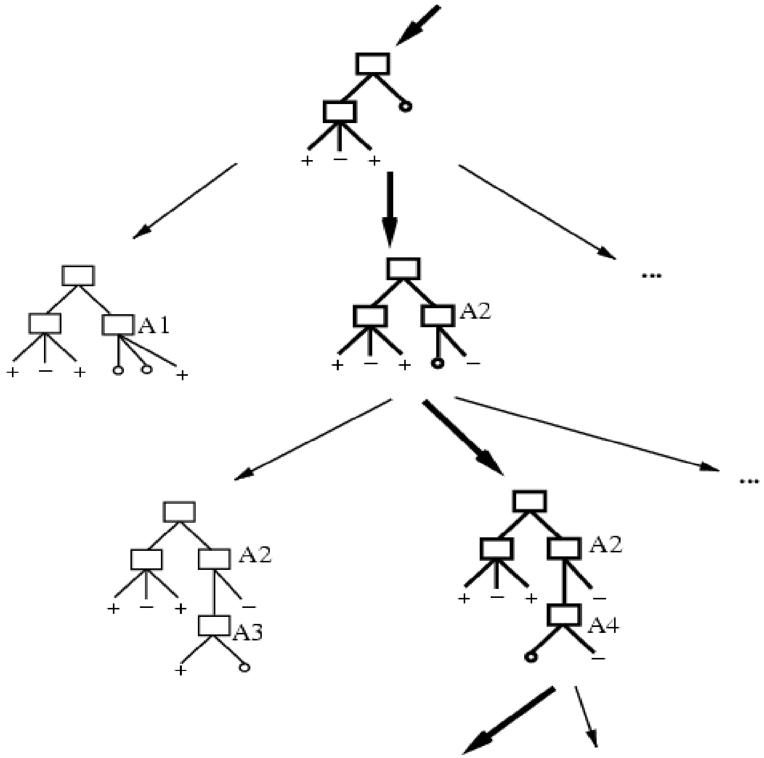
$S_{sunny} = \{D1,D2,D8,D9,D11\}$

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

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

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

Hypothesis Space Search



Simplest to complex, guided by an heuristic

Hypothesis Space Search

Hypothesis Space Search in Decision Tree Learning

- ID3's hypothesis space for all decision trees is a complete space of finite discrete-valued functions
- ID3 maintains only a single current hypothesis as it searches through the space of trees
- ID3 in its pure form performs no backtracking in its search
- ID3 uses all training examples at each step in the search (statistically based decisions)

Inductive Bias in DTL

Inductive Bias in DTL

Approximate Inductive bias of ID3: Shorter trees are preferred over larger trees. Trees that place high information gain attributes close to the root are preferred.

- ID3 searches incompletely a complete hypothesis space (preference bias)
- Candidate-Elimination searches completely an incomplete hypothesis space (language bias)

Inductive Bias in DTL

Approximate Inductive bias of ID3: Shorter trees are preferred over larger trees. Trees that place high information gain attributes close to the root are preferred.

– ID3 searc pletely a complete hypothesis

–

C4.5-Only gains greater than a value are considered

Inductive Bias in DTL

Approximate Inductive bias of ID3: Shorter trees are preferred over larger trees. Trees that place high information gain attributes close to the root are preferred.

- ID3 searches incompletely a complete hypothesis space (preference bias)
- Candidate-Elimination searches completely an incomplete hypothesis space (language bias)

Why Prefer Short Hypotheses?

Occam's Razor:

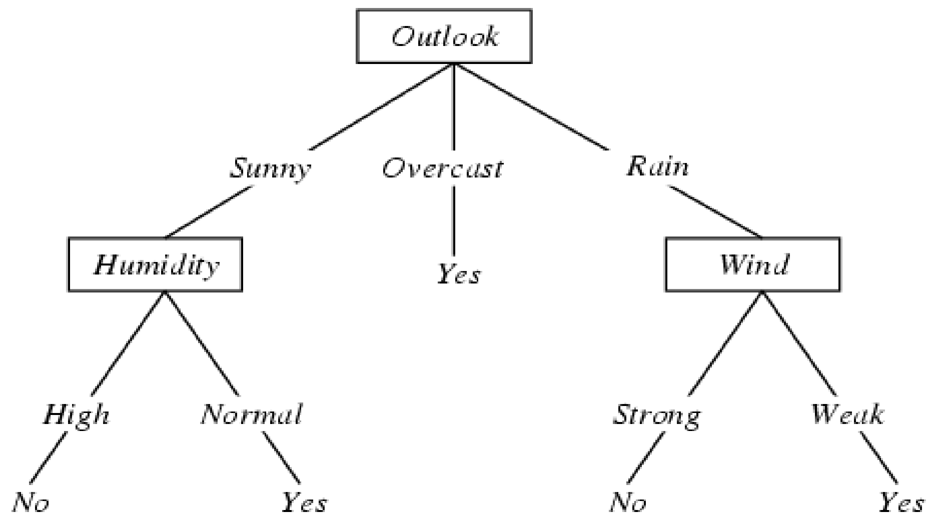
"Prefer the simplest hypothesis that fits the data"

Overfitting

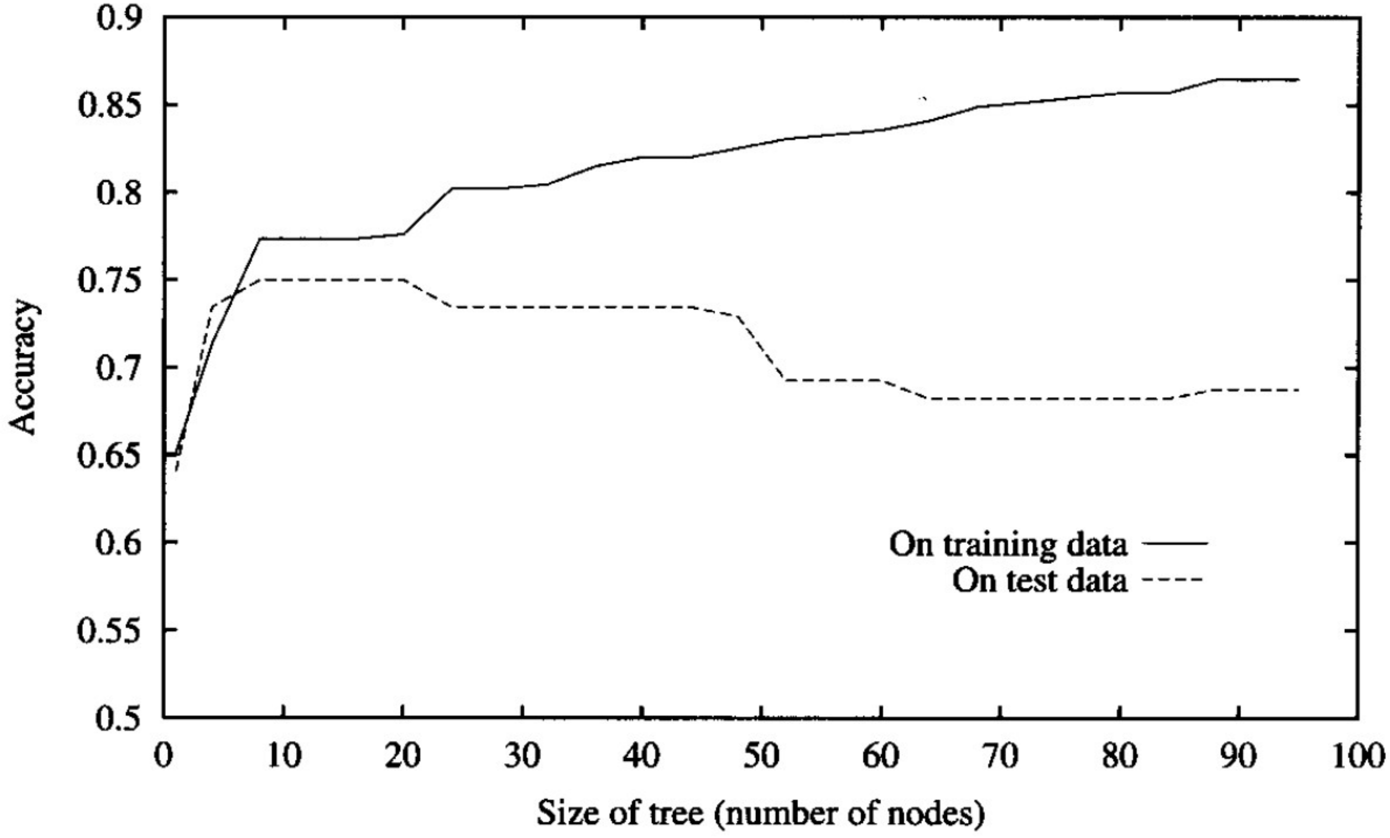
Consider adding noisy training example #15:

$Sunny,\ Hot,\ Normal,\ Strong,\ PlayTennis = No$

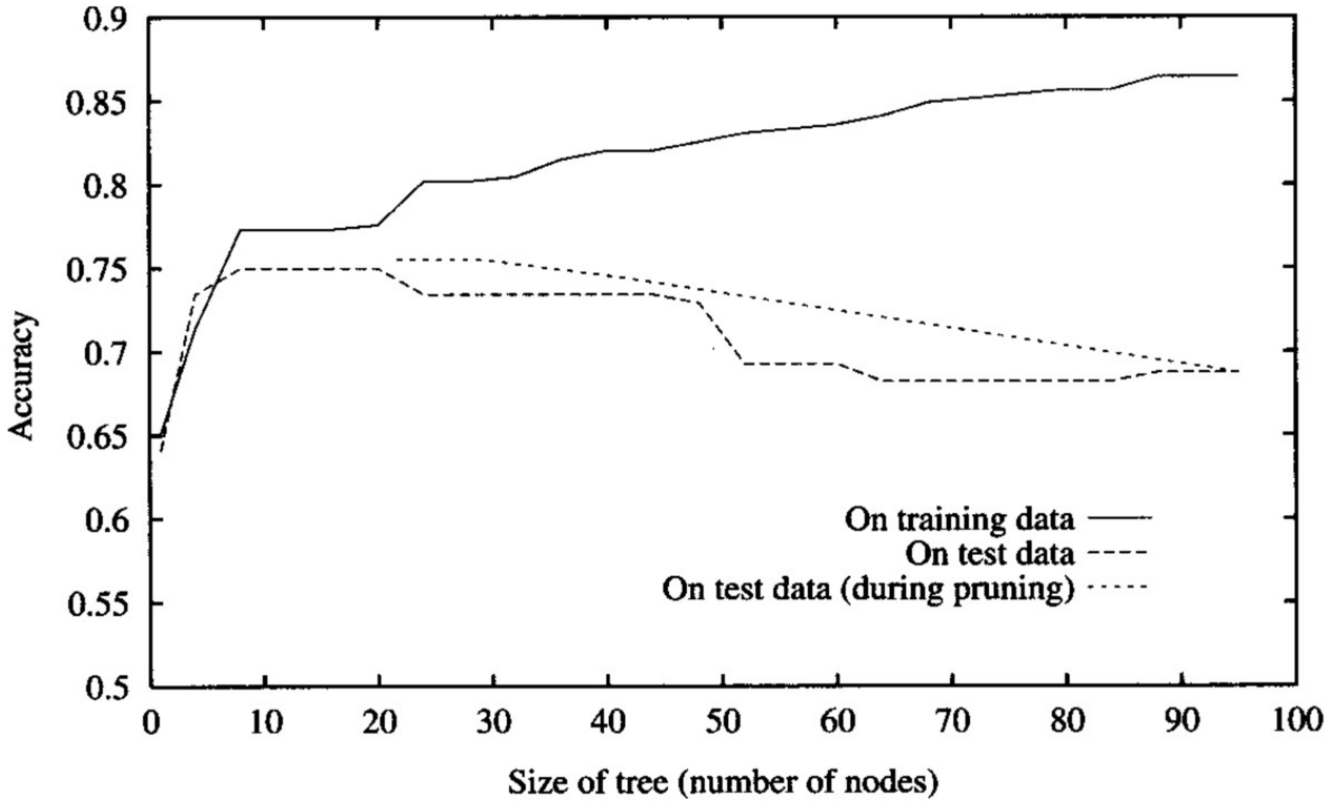
What effect on earlier tree?



Overfitting



Overfitting



Pruning

- Reduced-Error Pruning
  - Nodes are pruned iteratively, always choosing the node whose removal most increases the decision tree accuracy over the validation set
- Rule Pos-Pruning

  Example:

  IF (Outlook=Sunny) ^(Humidity=High)

  THEN PlayTennis = No

Advanced Material

- Incorporating continuous-valued attributes
- Alternative Measures for Selecting Attributes
- Handling Missing Attribute Values

Advanced Material

- Incorporating continuous-valued attributes

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

Advanced Material

- Incorporating continuous-valued attributes

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

T<54

Advanced Material

– Incorporating continuous-valued attributes
– Alternative Measures for Selecting Attributes
– Handling Missing Attribute Values

Advanced Material

- Incorporating continuous-valued attributes
- Alternative Measures for Selecting Attributes

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

Advanced Material

- Incorporating continuous-valued attributes
- Alternative Measures for Selecting Attributes

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

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

Advanced Material

- Incorporating continuous-valued attributes
- Alternative Measures for Selecting Attributes
- Handling Missing Attribute Values

Issues in Decision Tree Learning

Avoiding Overfitting the Data

- stop growing the tree earlier
- post-prune the tree

How?

- Use a separate set of examples
- Use statistical tests
- Minimize a measure of complexity of training examples plus decision tree

Decision Tree Learning: regression

Can we imagine a DTL method for regression?

-start, growing, stop

-value?

-attribute selection?