

Recap: Concept Learning

- Inductive inference (from examples to generalization)
- Hypothesis space
- Learning as search
- Inductive bias as base for generalization

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 (ensembles)

Example: Water Sport

“Days in which Aldo enjoys his favorite water sport”

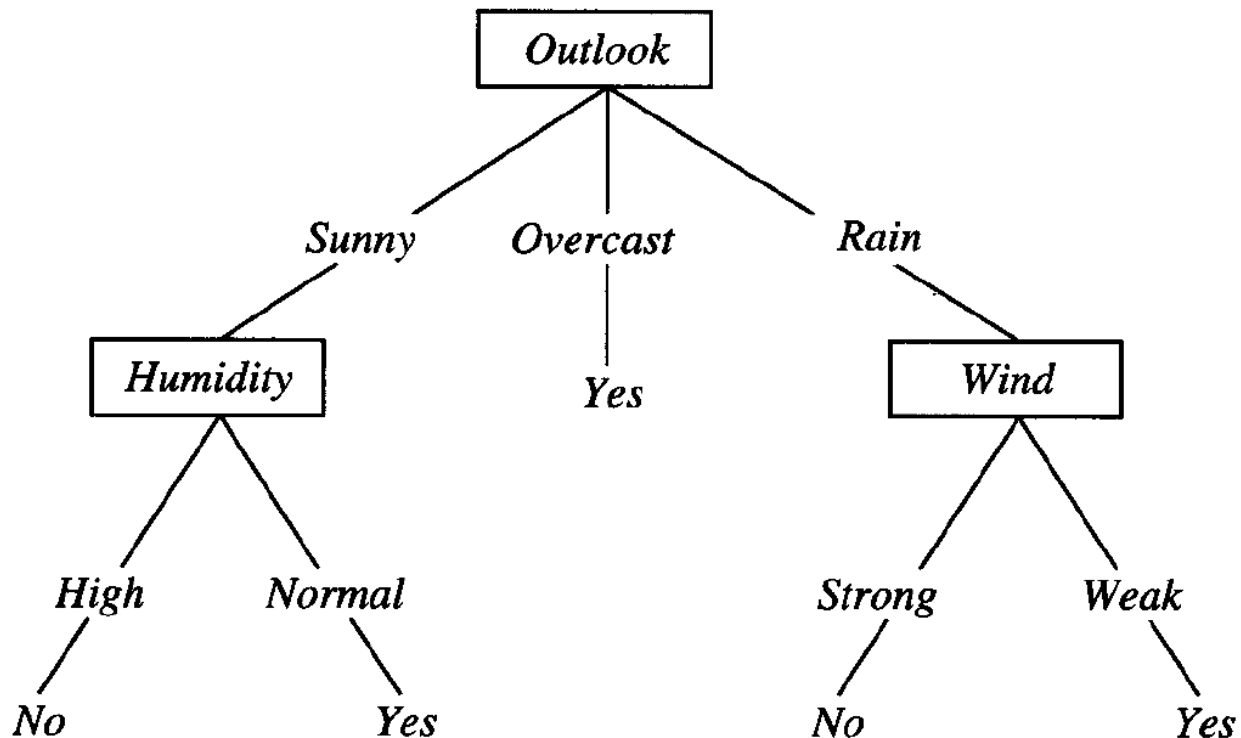
Example	Sky	AirTemp	Humidity	Wind	Water	Forecast	EnjoySport
1	Sunny	Warm	Normal	Strong	Warm	Same	Yes
2	Sunny	Warm	High	Strong	Warm	Same	Yes
3	Rainy	Cold	High	Strong	Warm	Change	No
4	Sunny	Warm	High	Strong	Cool	Change	Yes

A very simple answer.

Example: PlayTennis

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

Decision Tree for PlayTennis



Decision Tree Representation

- Each node tests some attribute of the instance
- Each branch selects one value for attribute
- Each leaf gives a response/class
- Decision trees represent a disjunction of conjunctions of constraints on the attributes

Example:

(Outlook=Sunny ^ Humidity=Normal)

∨

(Outlook = Overcast)

∨

(Outlook=Rain ^ 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

Hypothesis space for DTL

Set of function hypotheses $H = \{ h \mid h : X \rightarrow Y \}$

- each hypothesis h is a decision tree
- trees sort X values to a leaf, which assigns y

Decision trees represent a disjunction of conjunctions of constraints on the attributes

Which DT?

Multiple valid trees in most cases



Which DT?

Multiple valid trees in most cases

We want the smallest decision tree that correctly classifies all of the training examples

Which DT?

Multiple valid trees in most cases



We want the smallest decision tree that correctly classifies all of the training examples

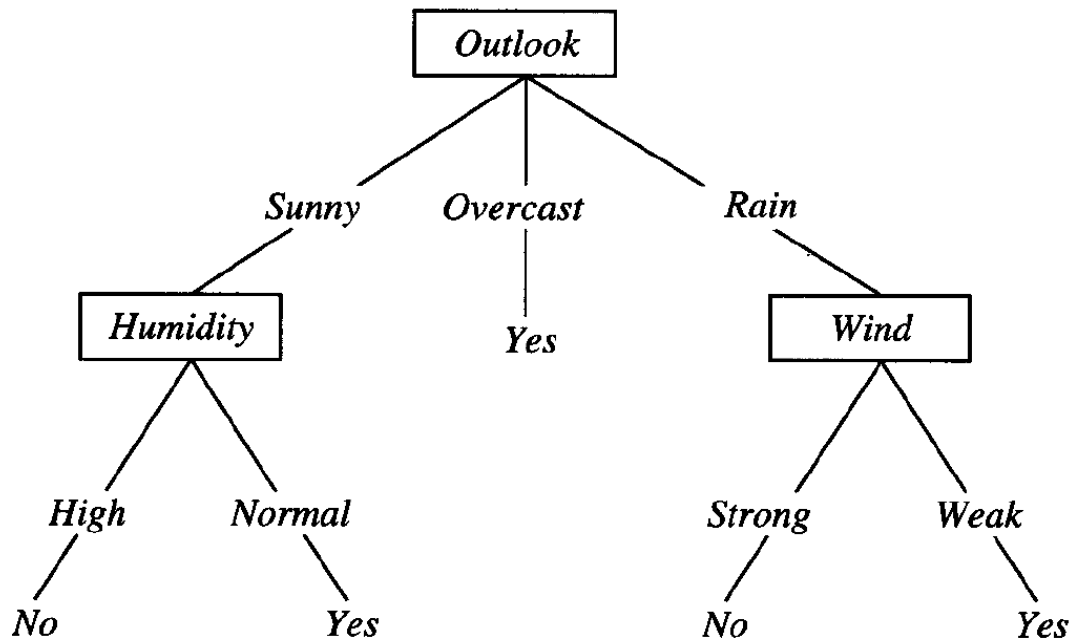
Finding the provably smallest decision tree is NP-hard (Quinlan 86).

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?

The Basic DTL Algorithm

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

Which attribute is the best classifier?

Random? More values? Less values?

The Basic DTL Algorithm

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

Which attribute is the best classifier?

⇒ answer based on information gain

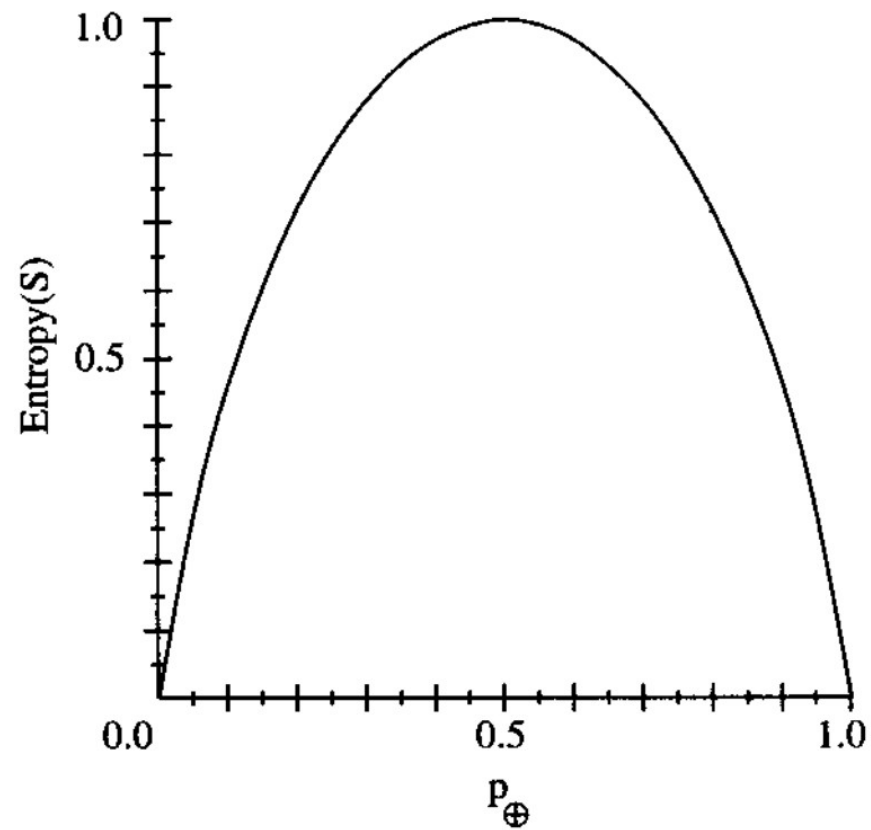
Entropy

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

$p_{+(-)}$ = 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
- Measure of impurity
- In general: $\text{Entropy}(S) = - \sum_{i=1,c} p_i \log_2 p_i$

Entropy



Entropy

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

Information Gain

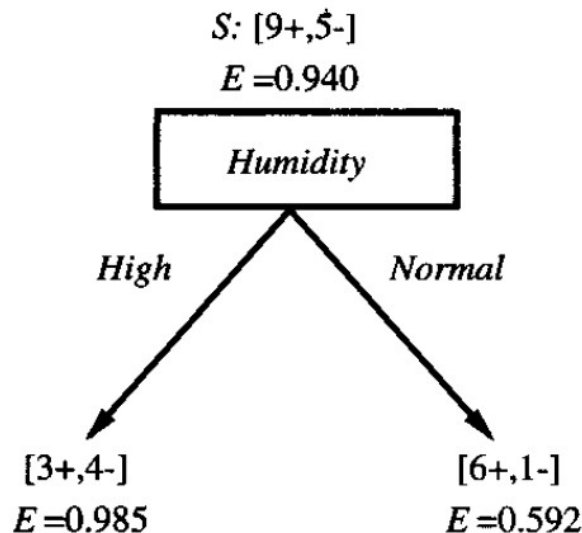
- Measures the expected reduction in entropy given the value of some attribute A

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

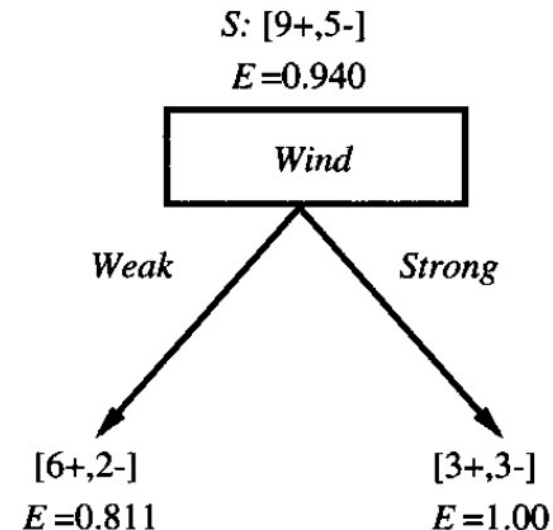
Values(A): Set of all possible values for attribute A

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

Selecting the Root Attribute



$$\begin{aligned} \text{Gain}(S, \text{Humidity}) &= .940 - (7/14) \cdot .985 - (7/14) \cdot .592 \\ &= .151 \end{aligned}$$



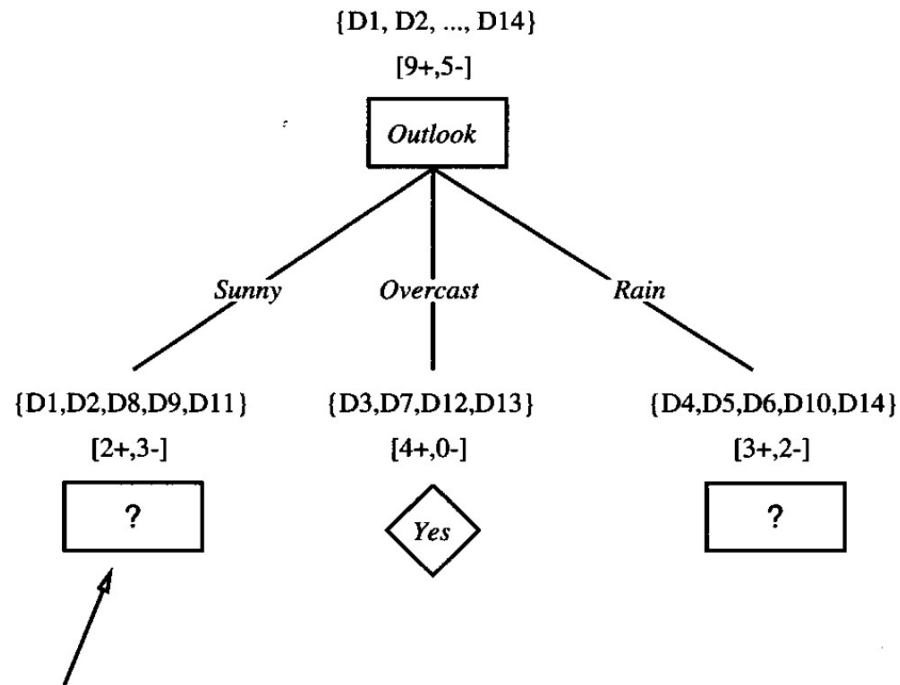
$$\begin{aligned} \text{Gain}(S, \text{Wind}) &= .940 - (8/14) \cdot .811 - (6/14) \cdot 1.0 \\ &= .048 \end{aligned}$$

PlayTennis Problem

- $\text{Gain}(S, \text{Outlook}) = 0.246$
- $\text{Gain}(S, \text{Humidity}) = 0.151$
- $\text{Gain}(S, \text{Wind}) = 0.048$
- $\text{Gain}(S, \text{Temperature}) = 0.029$

⇒ Outlook is the attribute of the root node

PlayTennis Problem



Which attribute should be tested here?

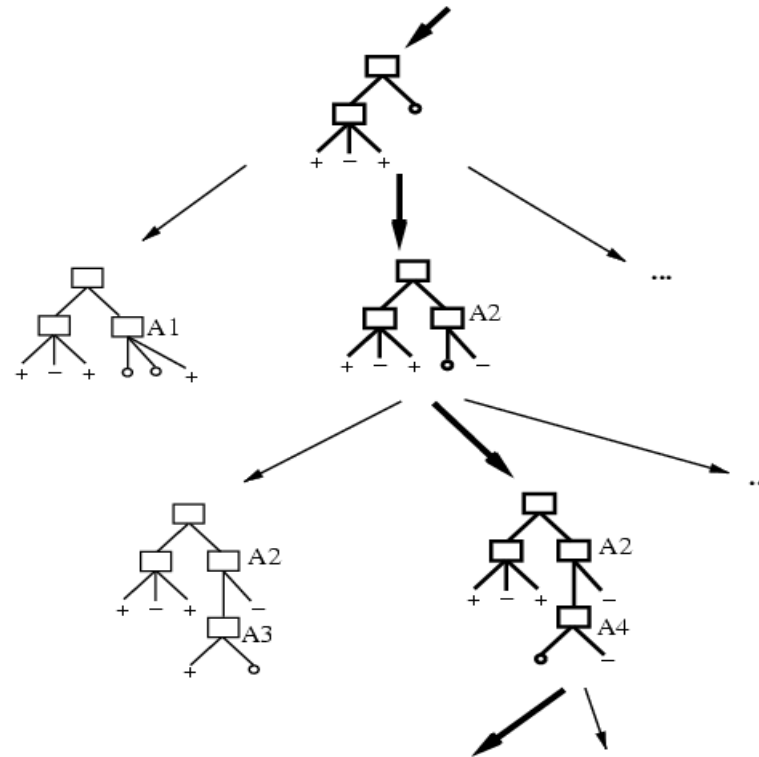
$$S_{\text{sunny}} = \{D1,D2,D8,D9,D11\}$$

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

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

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

Hypothesis Space Search



Simplest to complex, hill climbing, no backtracking

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 search is completely 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?



Why Prefer Short Hypotheses?

Occam's Razor:


“Prefer the simplest hypothesis
that fits the data”

Recap

DTL:

- Method for approximation of discrete-valued target functions (classification) ← So far
- Decision trees represent a disjunction of conjunctions of constraints on the attributes (highly expressive)
- Search: Simplest to complex, hill climbing, no backtracking.
- Statistical decisions on nodes and leaves

How good is our tree?

- No guaranties of perfect classification
 - We can measure the accuracy on the dataset
 - Proportion of correct labels
 - Is it a good measure of performance?
- 

How good is our tree?

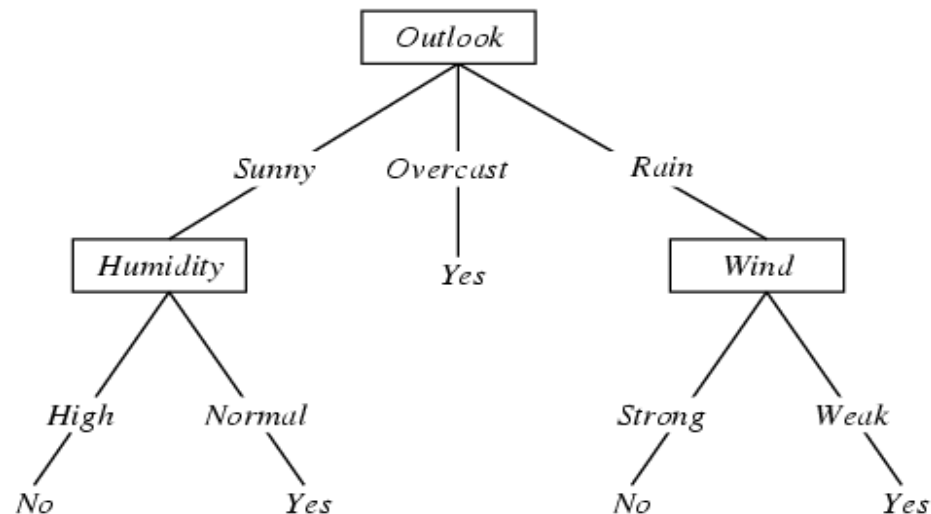
- No guaranties of perfect classification
- We can measure the accuracy on the dataset
 - Proportion of correct labels
- Is it a good measure of performance?
- Statistical bias: we need an independent estimation of the error on the distribution
 - We use a test set

Overfitting

Consider adding noisy training example #15:

Sunny, Hot, Normal, Strong, PlayTennis = No

What effect on earlier tree?

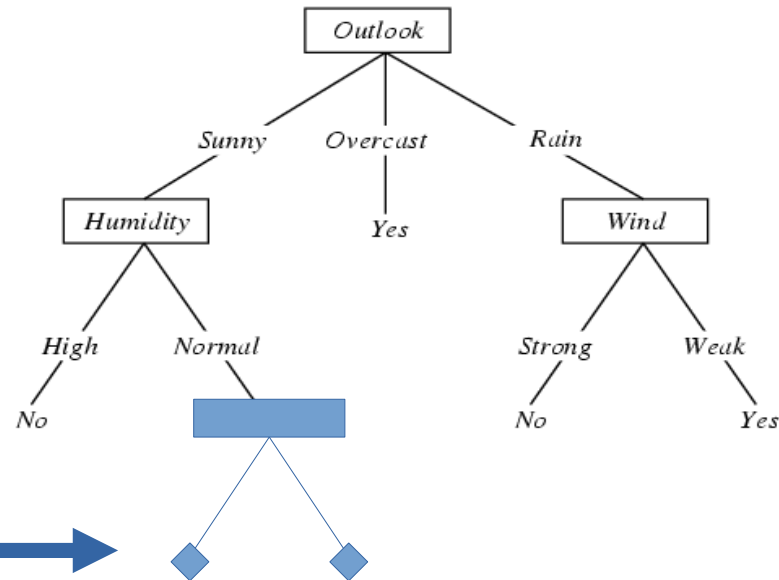


Overfitting

Consider adding noisy training example #15:

Sunny, Hot, Normal, Strong, PlayTennis = No

What effect on earlier tree?

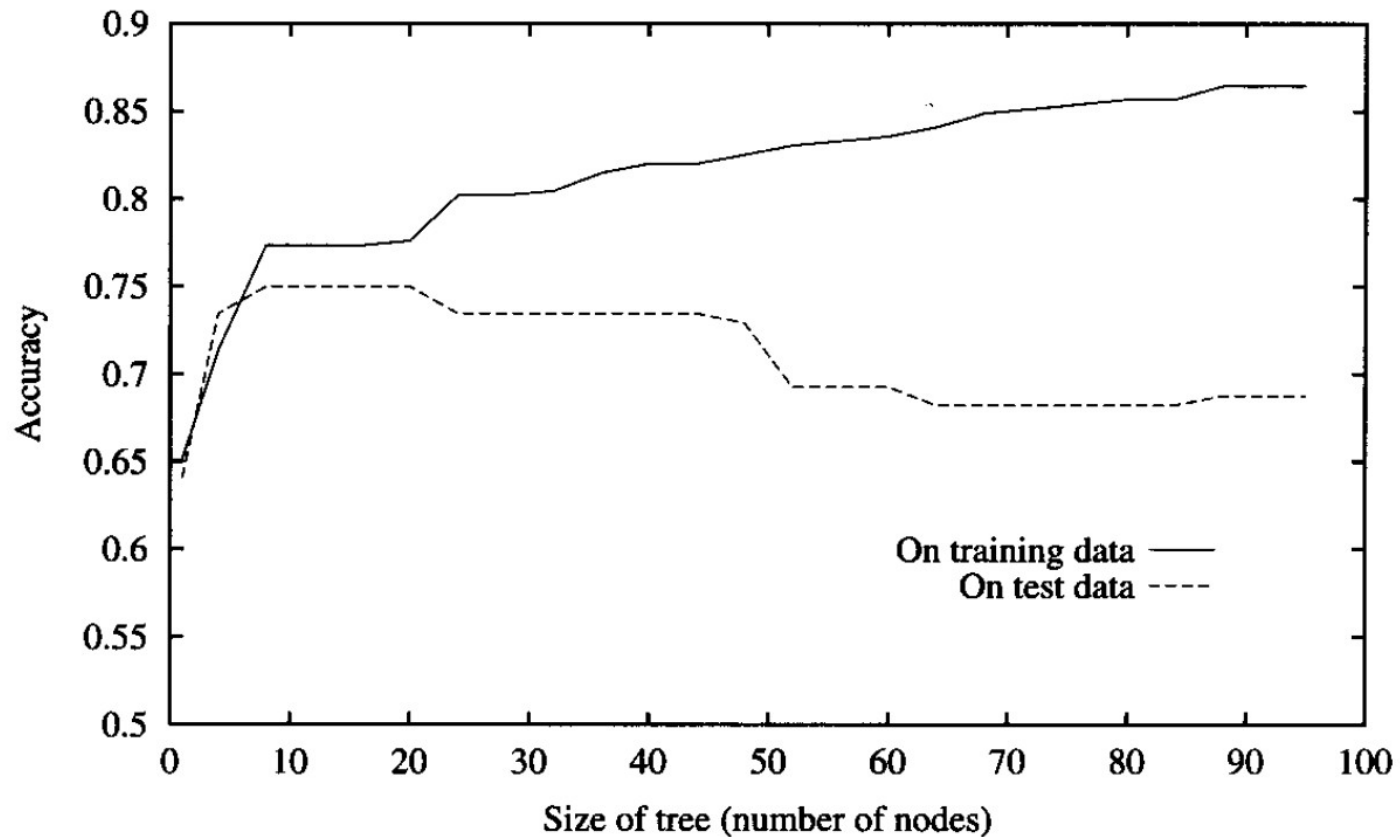


Adds new
branch! →

Learning noise


- Noisy samples can lead to a bigger tree
- Small samples can show regularities not present in the distribution
- In both cases, when the tree grows, it learns things not present in the complete distribution

Overfitting



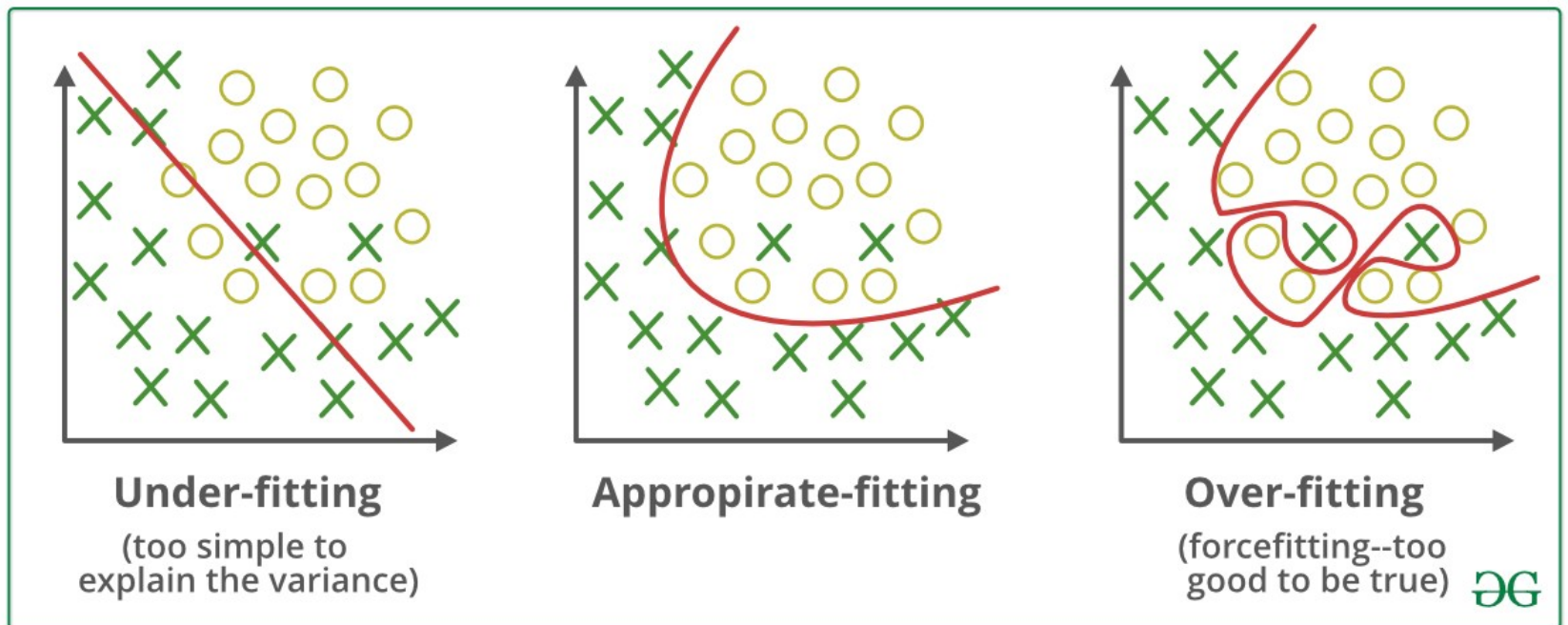
Overfitting

An hypothesis h is said to **OVERFIT** if there is another hypothesis h' such that h' has a lower accuracy on the training data but a higher accuracy on the full distribution.



Overfitting

A typical situation:



Pruning

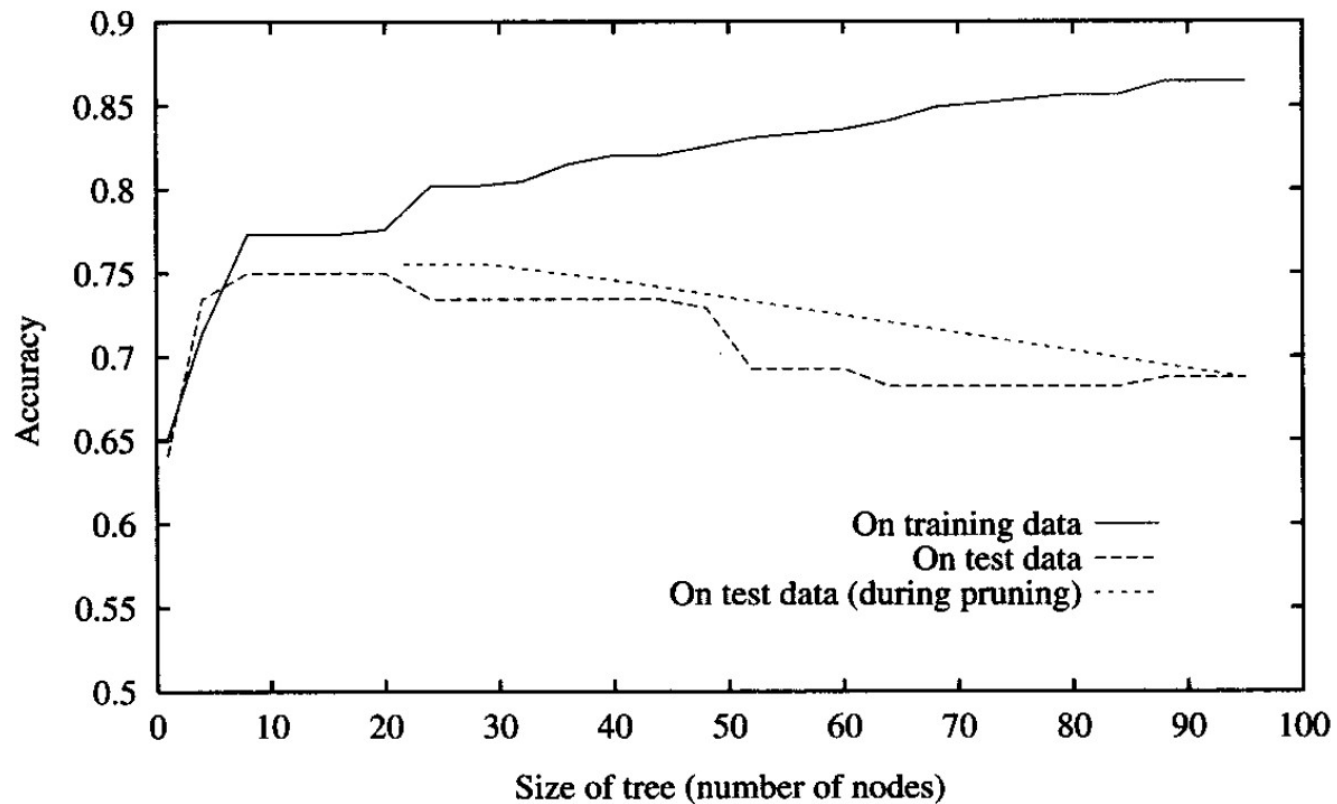
- Reduced-Error Pruning
 - Nodes are pruned iteratively, always choosing the node whose removal most increases the estimated decision tree accuracy over the full distribution

- Rule Pos-Pruning

Example:

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

Overfitting



Advanced Material

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

Advanced Material

- Incorporating continuous-valued attributes

<i>Temperature:</i>	40	48	60	72	80	90
<i>PlayTennis:</i>	No	No	Yes	Yes	Yes	No

Advanced Material

- Incorporating continuous-valued attributes

<i>Temperature:</i>	40	48	60	72	80	90
<i>PlayTennis:</i>	No	No	Yes	Yes	Yes	No

$T < 54$

Advanced Material

- Incorporating continuous-valued attributes
 - Continuous to binary
 - Multiple use of the same attribute
 - Sklearn only accept numerical attributes

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


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

Advanced Material

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

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

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

Advanced Material

- Incorporating continuous-valued attributes
- Alternative Measures for Selecting Attributes
- Handling Missing Attribute Values
 - Statistical decision at each node

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?

Recap

DTL:

- Method for approximation of discrete-valued target functions (classification) ← So far
- Decision trees represent a disjunction of conjunctions of constraints on the attributes (highly expressive)
- Search: Simplest to complex, hill climbing, no backtracking.
- Statistical decisions on nodes and leaves
- Possible overfitting in some situations
- Numerical data: discretization.