

Decision Tree Learning

Based on “Machine Learning”, T. Mitchell, McGRAW Hill, 1997, ch. 3

Acknowledgement:

The present slides are an adaptation of slides drawn by T. Mitchell

PLAN

1.

- DT Learning: **Basic Issues**

1. **Concept learning:** an example
2. Decision tree **representation**
3. ID3 learning **algorithm**

Hypothesis space search by ID3

Statistical measures in decision tree learning:
Entropy, Information gain

- DT Learning: **Advanced Issues**

1. **Inductive bias** in ID3
2. Avoiding **overfitting** of data:
reduced-error pruning, and rule post-pruning
3. Dealing with...
continuous-valued attributes
attributes with many values
attributes with different costs
training examples with missing attributes values

When to Consider Decision Trees

- Instances are described by attribute–value pairs
- Target function is discrete valued
- Disjunctive hypothesis may be required
- Possibly noisy training data

Examples:

- Equipment or medical diagnosis
- Credit risk analysis
- Modeling calendar scheduling preferences

1. Basic Issues in DT Learning

1.1 Concept learning: An example

Given the data:

Day	Outlook	Temperature	Humidity	Wind	EnjoyTennis
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

predict the value of **EnjoyTennis** for

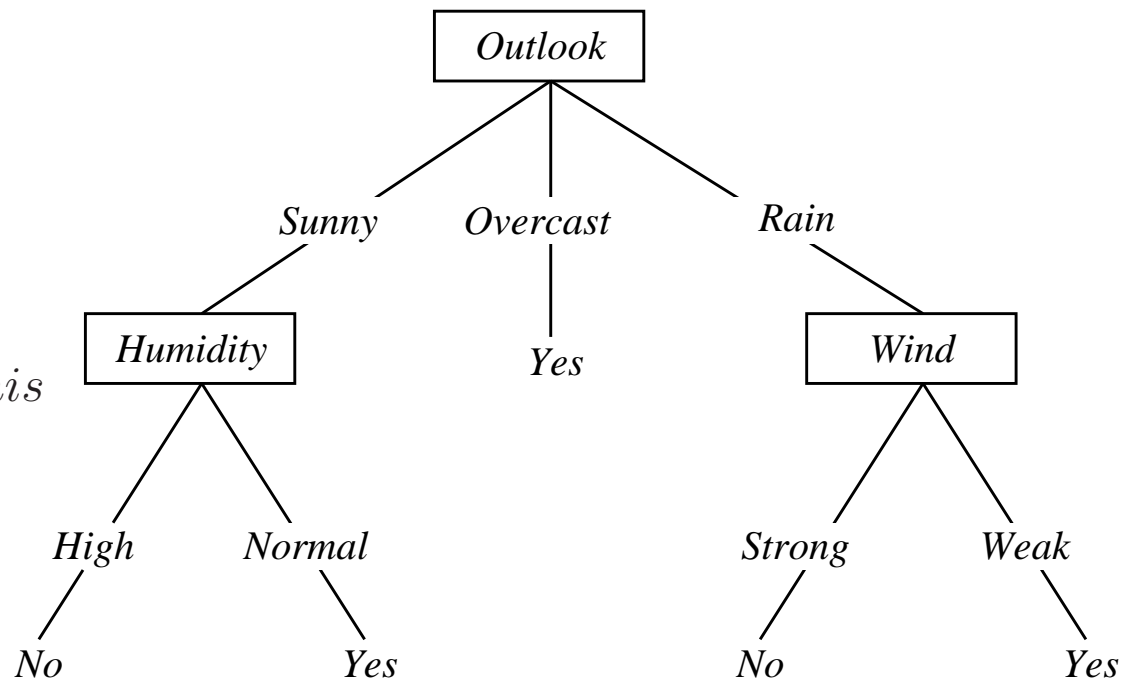
⟨Outlook = sunny, Temp = cool, Humidity = high, Wind = strong⟩

1.2. Decision tree representation

- Each internal node tests an attribute
- Each branch corresponds to attribute value
- Each leaf node assigns a classification

Example:

Decision Tree for *EnjoyTennis*



Another example: A Tree to Predict C-Section Risk

Learned from medical records of 1000 women

Negative examples are C-sections

```
[833+,167-] .83+ .17-
Fetal_Presentation = 1: [822+,116-] .88+ .12-
| Previous_Csection = 0: [767+,81-] .90+ .10-
| | Primiparous = 0: [399+,13-] .97+ .03-
| | Primiparous = 1: [368+,68-] .84+ .16-
| | | Fetal_Distress = 0: [334+,47-] .88+ .12-
| | | | Birth_Weight < 3349: [201+,10.6-] .95+ .05-
| | | | Birth_Weight >= 3349: [133+,36.4-] .78+ .22-
| | | Fetal_Distress = 1: [34+,21-] .62+ .38-
| Previous_Csection = 1: [55+,35-] .61+ .39-
Fetal_Presentation = 2: [3+,29-] .11+ .89-
Fetal_Presentation = 3: [8+,22-] .27+ .73-
```

1.3. ID3 Algorithm: Top-Down Induction of Decision Trees

START

create the root *node*;
assign all examples to root;

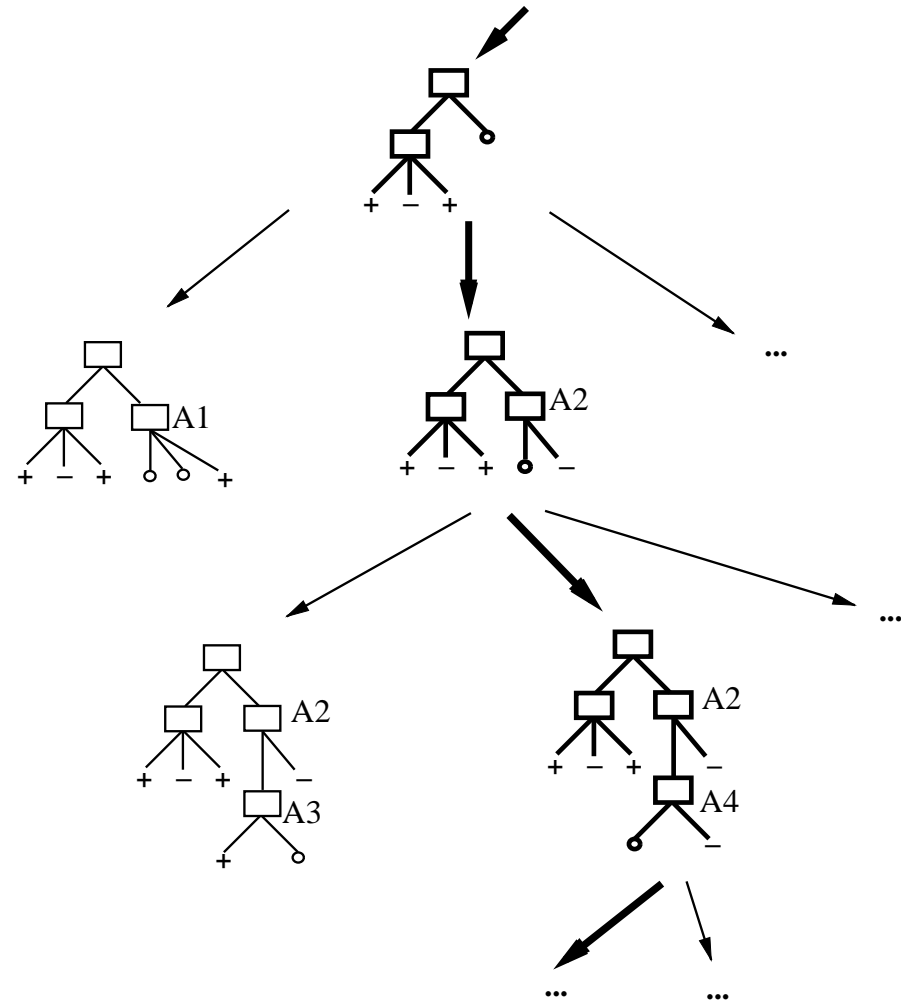
Main loop:

1. $A \leftarrow$ the “best” decision attribute for next *node*;
2. for each value of A , create a new descendant of *node*;
3. sort training examples to leaf nodes;
4. if training examples perfectly classified, then STOP;
else iterate over new leaf nodes

Hypothesis Space Search by ID3

7.

- Hypothesis space is complete!
The target function surely is in there...
- Outputs a single hypothesis
Which one?
- Inductive bias:
approximate “prefer the shortest tree”
- Statically-based search choices
Robust to noisy data...
- No back-tracking
Local minima...



Statistical measures in DT learning: Entropy, and Information Gain

Information gain:

the expected reduction of the entropy of the instance set S due to sorting on the attribute A

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

Entropy

- Let S be a sample of training examples
 p_{\oplus} is the proportion of positive examples in S
 p_{\ominus} is the proportion of negative examples in S

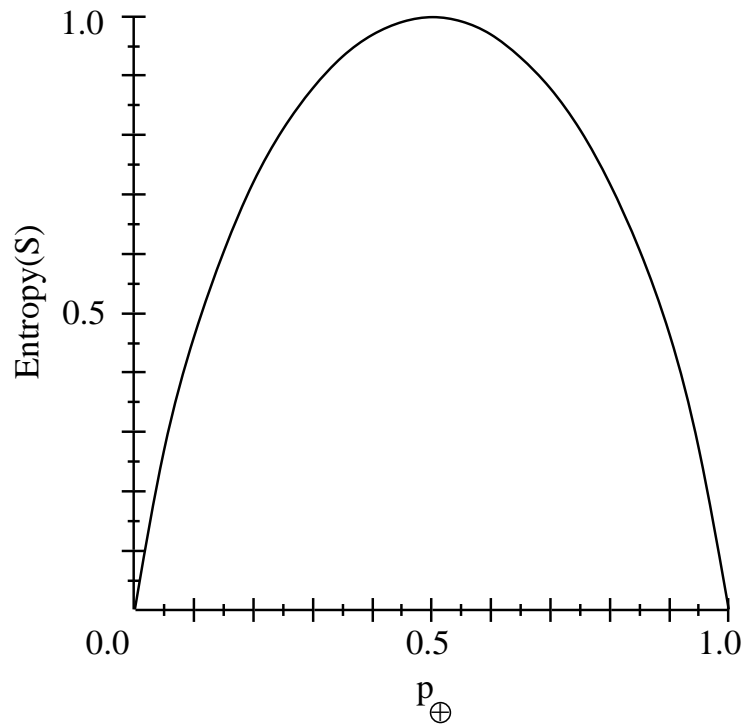
- Entropy measures the impurity of S

- **Information theory:**

Entropy(S) = expected number of bits needed to encode \oplus or \ominus for a randomly drawn member of S (under the optimal, shortest-length code)

The optimal length code for a message having the probability p is $-\log_2 p$ bits. So:

$$\text{Entropy}(S) = p_{\oplus}(-\log_2 p_{\oplus}) + p_{\ominus}(-\log_2 p_{\ominus}) = -p_{\oplus} \log_2 p_{\oplus} - p_{\ominus} \log_2 p_{\ominus}$$

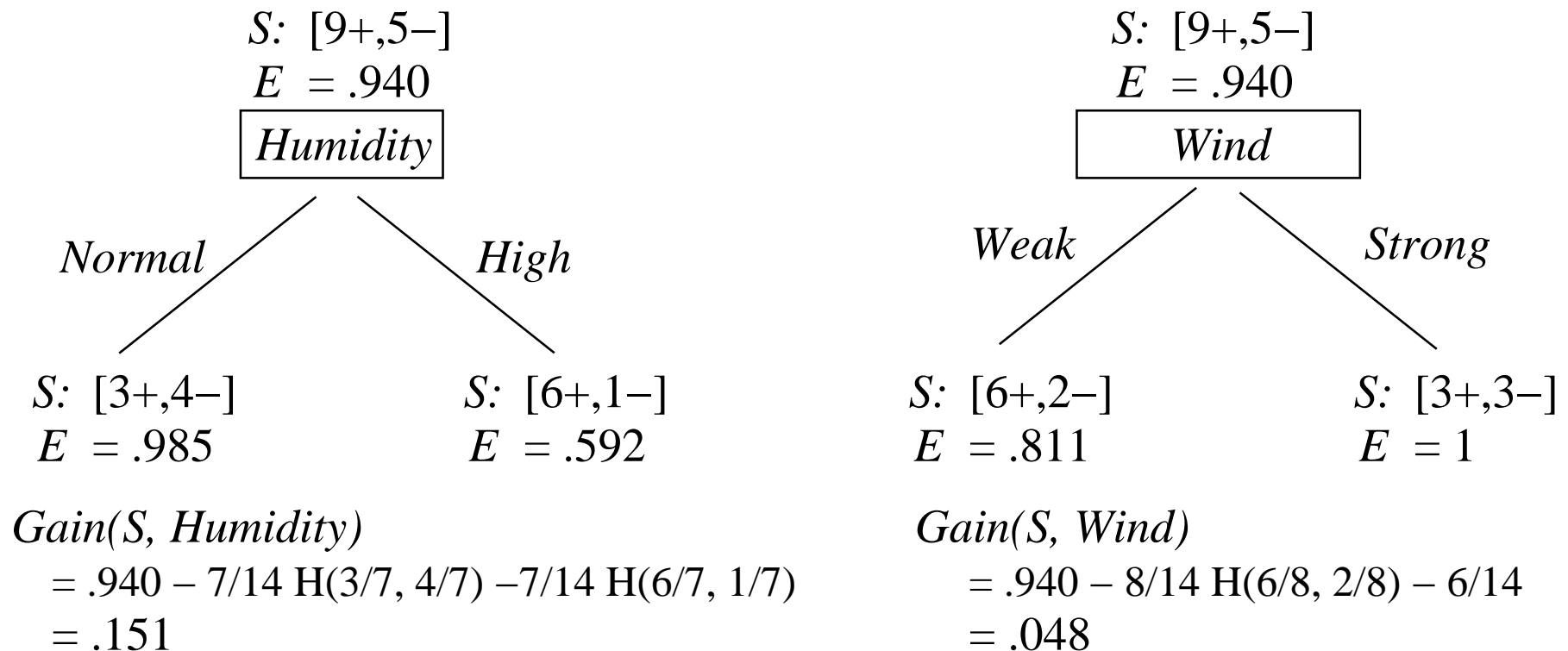


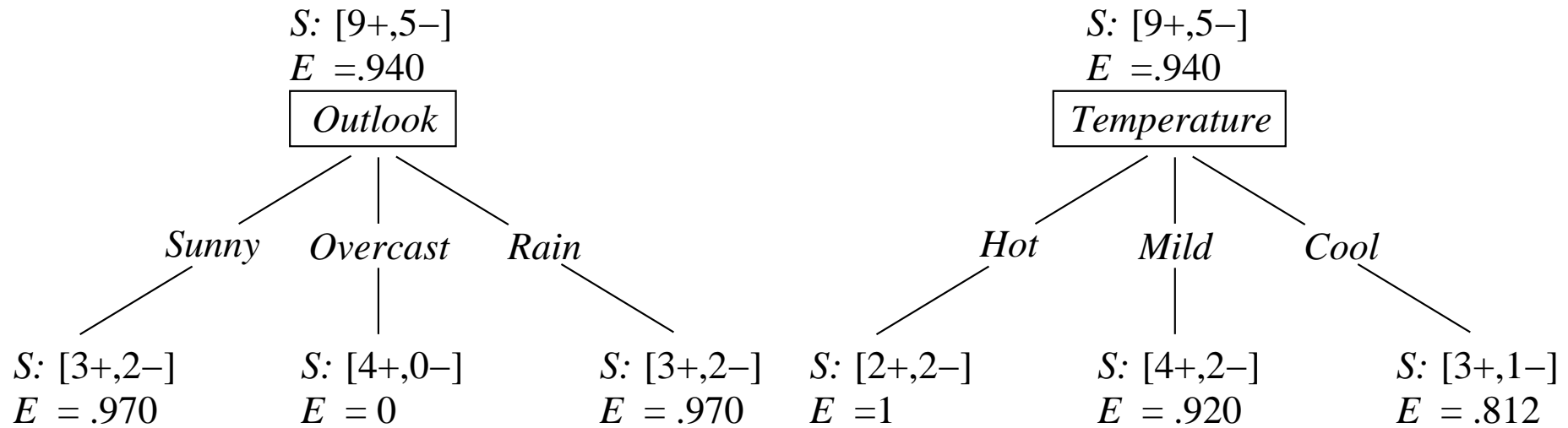
$$Entropy(S) = p_{\oplus} \log_2 \frac{1}{p_{\oplus}} + p_{\ominus} \log_2 \frac{1}{p_{\ominus}}$$

Note: By convention, $0 \cdot \log_2 0 = 0$.

Back to the *EnjoyTennis* example:

Selecting the root attribute





$Gain(S, Outlook)$

$$= .940 - 5/14 H(2/5, 2/5) - 5/14 H(2/5, 3/5)$$

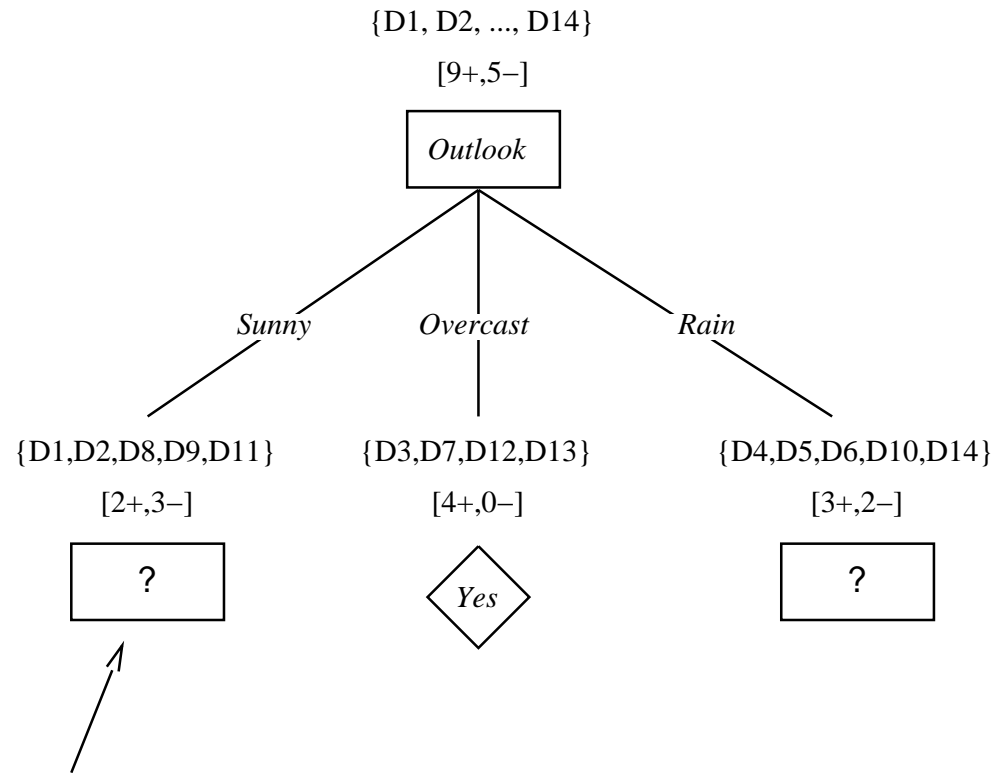
$$= 0.247$$

$Gain(S, Temperature)$

$$= .940 - 4/14 - 6/14 H(4/6, 2/6) - 4/14 H(1/4, 3/4)$$

$$= 0.028$$

A partially learned tree



Which attribute should be tested here?

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

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

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

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

2. Advanced Issues in DT Learning

2.1 Inductive Bias in ID3

Is ID3 unbiased?

Not really...

- Preference for short trees, and for those with high information gain attributes near the root
- The ID3 bias is a *preference* for some hypotheses (i.e., a *search bias*); there are learning algorithms (e.g. CANDIDATE-ELIMINATION, ch. 2) whose bias is a *restriction* of hypothesis space H (i.e., a *language bias*).
- Occam's razor: prefer the shortest hypothesis that fits the data

Occam's Razor

Why prefer short hypotheses?

Argument in favor:

- Fewer short hypotheses than long hypotheses
 - a short hypothesis that fits data unlikely to be coincidence
 - a long hypothesis that fits data might be coincidence

Argument opposed:

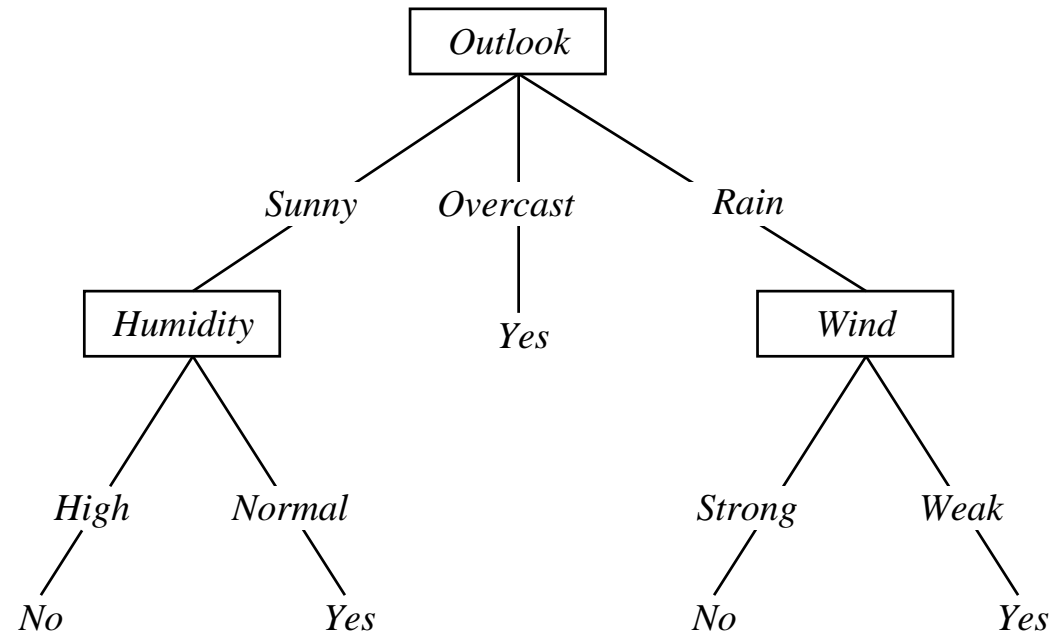
- There are many ways to define small sets of hypotheses (e.g., all trees with a prime number of nodes that use attributes beginning with "Z")
- What's so special about small sets based on the *size* of hypotheses?

2.2 Overfitting in Decision Trees

Consider adding **noisy training example #15**:

(Sunny, Hot, Normal, Strong, EnjoyTennis = No)

What **effect** does it produce
on the earlier tree?



Overfitting: Definition

Consider error of hypothesis h over

- training data: $error_{train}(h)$
- entire distribution \mathcal{D} of data: $error_{\mathcal{D}}(h)$

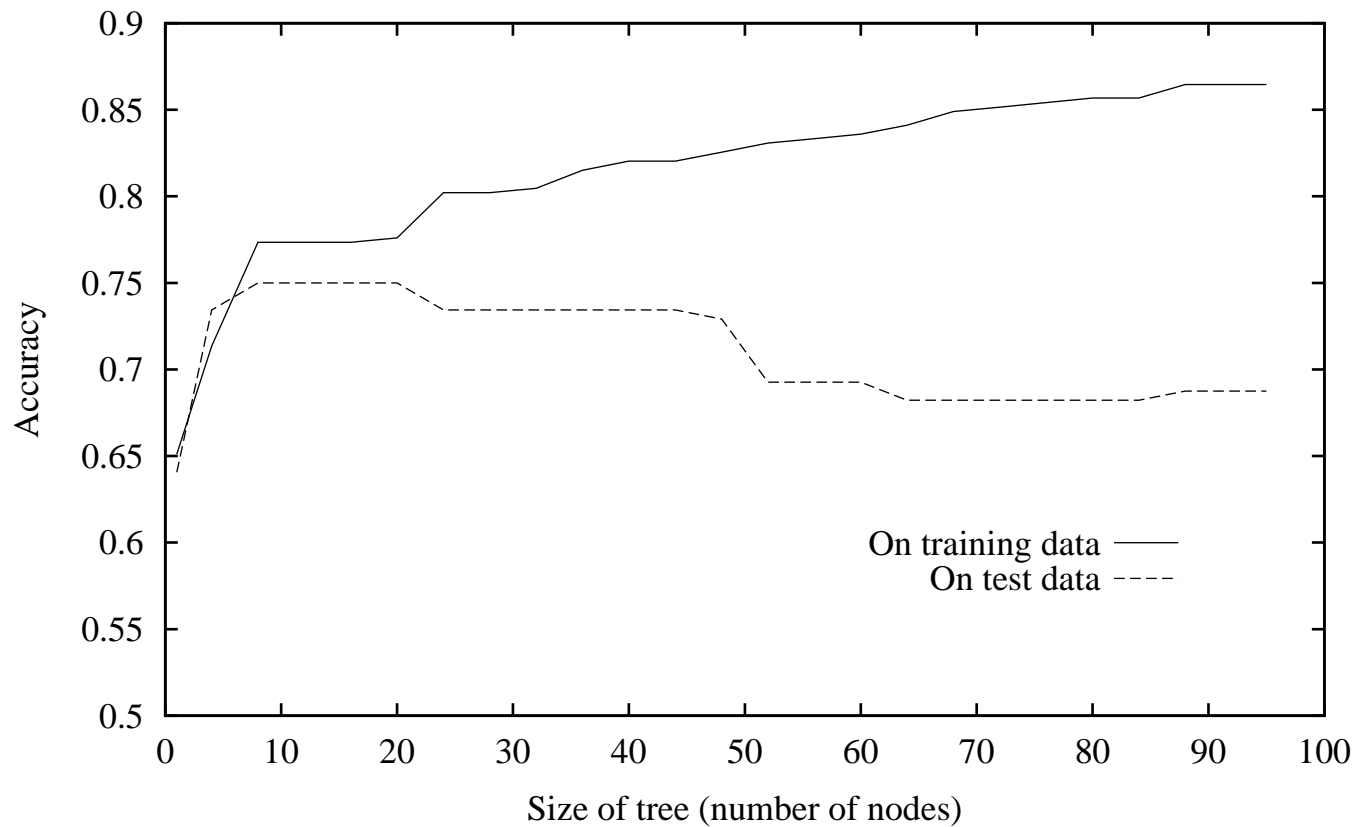
Hypothesis $h \in H$ overfits training data if
there is an alternative hypothesis $h' \in H$ such that

$$error_{train}(h) < error_{train}(h')$$

and

$$error_{\mathcal{D}}(h) > error_{\mathcal{D}}(h')$$

Overfitting in Decision Tree Learning



Avoiding Overfitting

How can we avoid overfitting?

- stop growing when the data split is not anymore statistically significant
- grow full tree, then post-prune

How to select the “best” tree:

- Measure performance over training data
- Measure performance over a separate validation data set
- Minimum Description Length (MDL) principle:
minimize $size(tree) + size(misclassifications(tree))$

2.2.1 Reduced-Error Pruning

Split data into **training set** and **validation set**

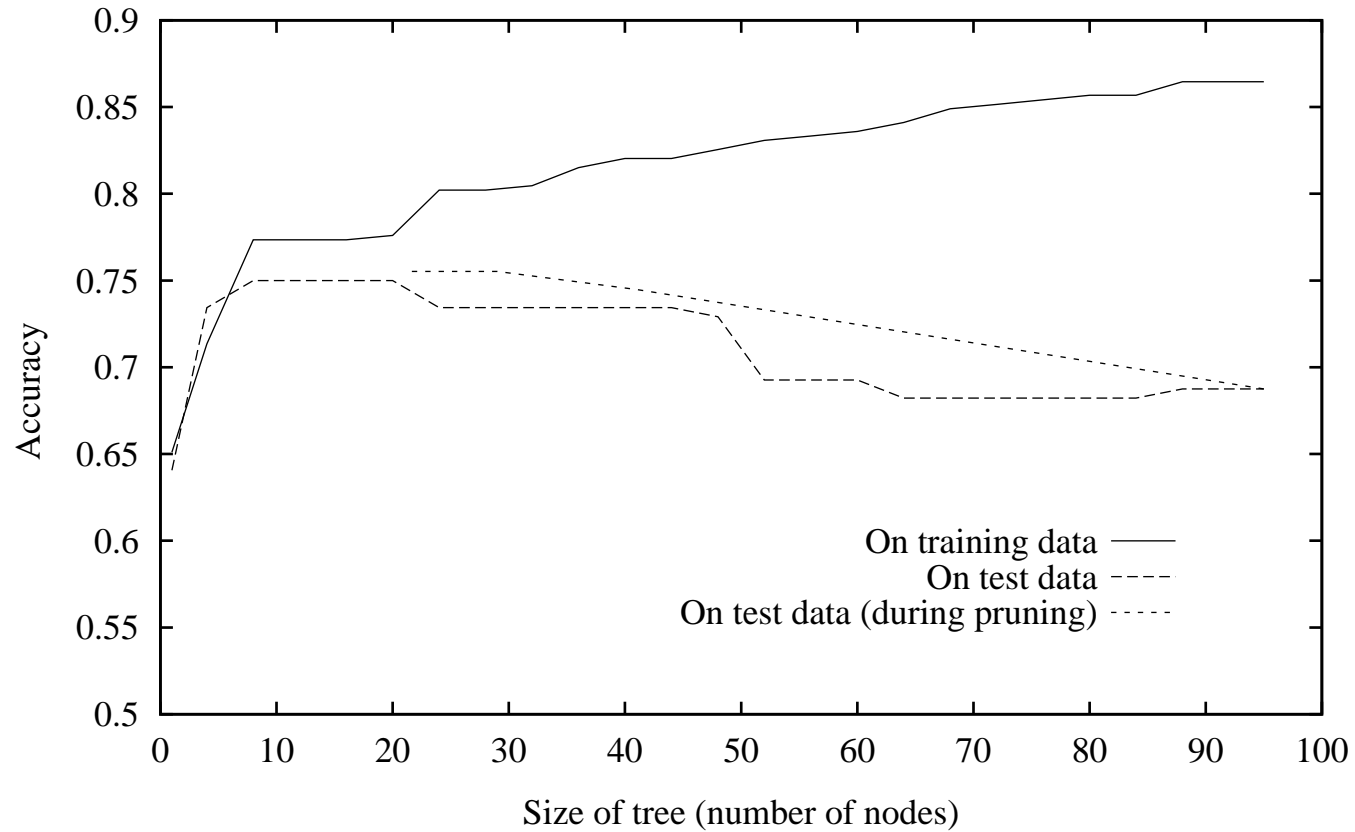
Do until further pruning is harmful:

1. Evaluate impact on validation set of pruning each possible node (plus those below it)
2. Greedily remove the one that most improves validation set accuracy

Effect: Produces the smallest version of most accurate subtree

Question: What if data is limited?

Effect of Reduced-Error Pruning



Note:

A validation set (distinct from both the training and test sets) was used for pruning. Accuracy over this validation set is not shown here.

2.2.2 Rule Post-Pruning

1. Convert tree to equivalent set of **rules**
2. **Prune** each rule **independently** of others
3. **Sort** the final rules into the desired sequence (e.g. according to the estimated accuracy) for use

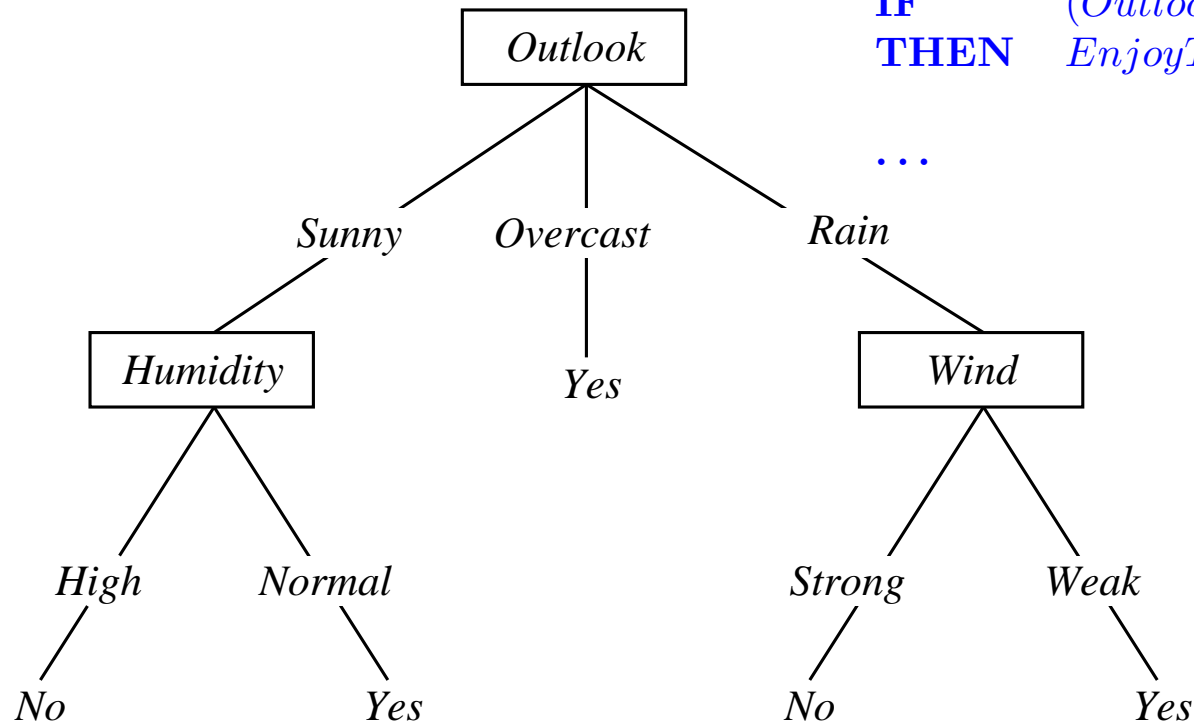
It is perhaps most frequently used method (e.g., C4.5)

Converting A Tree to Rules

IF $(Outlook = Sunny) \wedge (Humidity = High)$
THEN $EnjoyTennis = No$

IF $(Outlook = Sunny) \wedge (Humidity = Normal)$
THEN $EnjoyTennis = Yes$

...



2.3 Dealing with...

...Continuous valued attributes

Create one or more discrete attribute to test the continuous.

For instance:

Temperature = 82.5

$(\textit{Temperature} > 72.3) = t, f$

How to choose such (threshold) values:

Sort the examples according to the values of the continuous attribute, then identify examples that differ in their target classification.

For EnjoyTennis:

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

*Temperature*_{>54}

*Temperature*_{>85}

...Attributes with many values

Problem:

- If an attribute has many values, *Gain* will select it
- Imagine using *Date = Jun_3_1996* as attribute

One **approach**: use *GainRatio* instead

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

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

where S_i is the subset of S for which A has the value v_i

...Attributes with different costs

Consider

- medical diagnosis, *BloodTest* has cost \$150
- robotics, *Width_from_1ft* has cost 23 sec.

Question:

How to learn a consistent tree with low expected cost?

One **approach:** replace gain by

- $\frac{Gain^2(S,A)}{Cost(A)}$ (Tan and Schlimmer, 1990)
- $\frac{2^{Gain(S,A)} - 1}{(Cost(A) + 1)^w}$ (Nunez, 1988)

where $w \in [0, 1]$ determines importance of cost

...Training examples with unknown attribute values

Question:

What if an example is missing the value of an attribute A ?

Use the training example anyway, sort through tree

- If node n tests A , assign the most common value of A among the other examples sorted to node n , or
- assign the most common value of A among the other examples with same target value, or
- assign probability p_i to each possible value v_i of A ;
assign the fraction p_i of the example to each descendant in the tree.

Classify new examples in the same fashion.