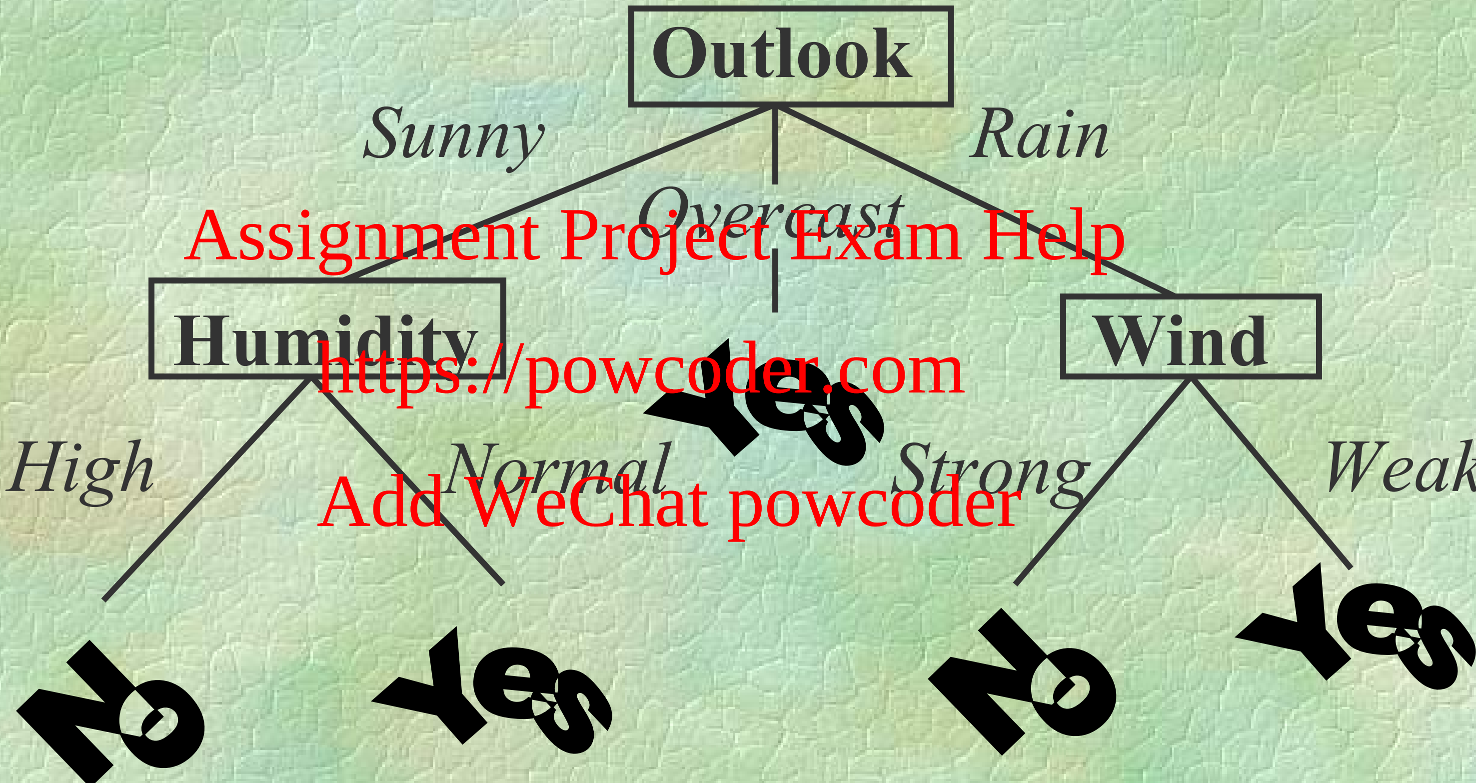
Machine Learning: Lecture 3

Decision Tree Learning

(Based on Chapter 3 of Mitchell T..,
Machine Learning, 1997)

thanks to Brian Pardo (http://bryanpardo.com) for the
illustrations on slides 9, 18

Decision Tree Representation



A Decision Tree for the concept *PlayTennis*

Appropriate Problems for Decision Tree Learning

- Instances are represented by discrete attribute-value pairs (though the basic algorithm was extended to real-valued attributes as well)

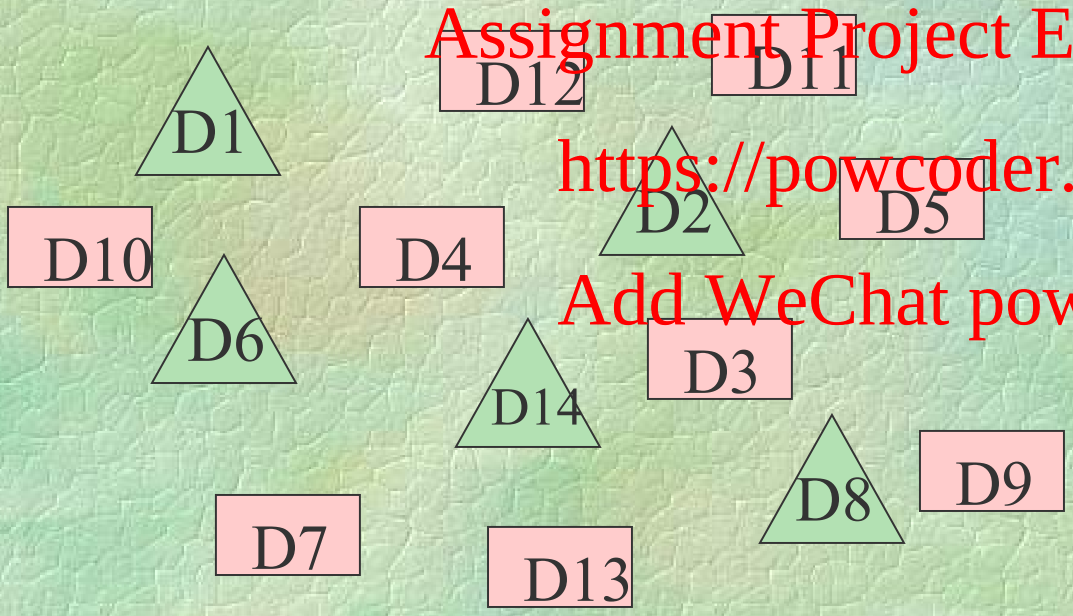
- The target function has discrete output values (can have more than two possible output values --> classes)

- Disjunctive hypothesis descriptions may be required

- The training data may contain errors
- The training data may contain missing attribute values

ID3: The Basic Decision Tree Learning Algorithm

See database on the next slide



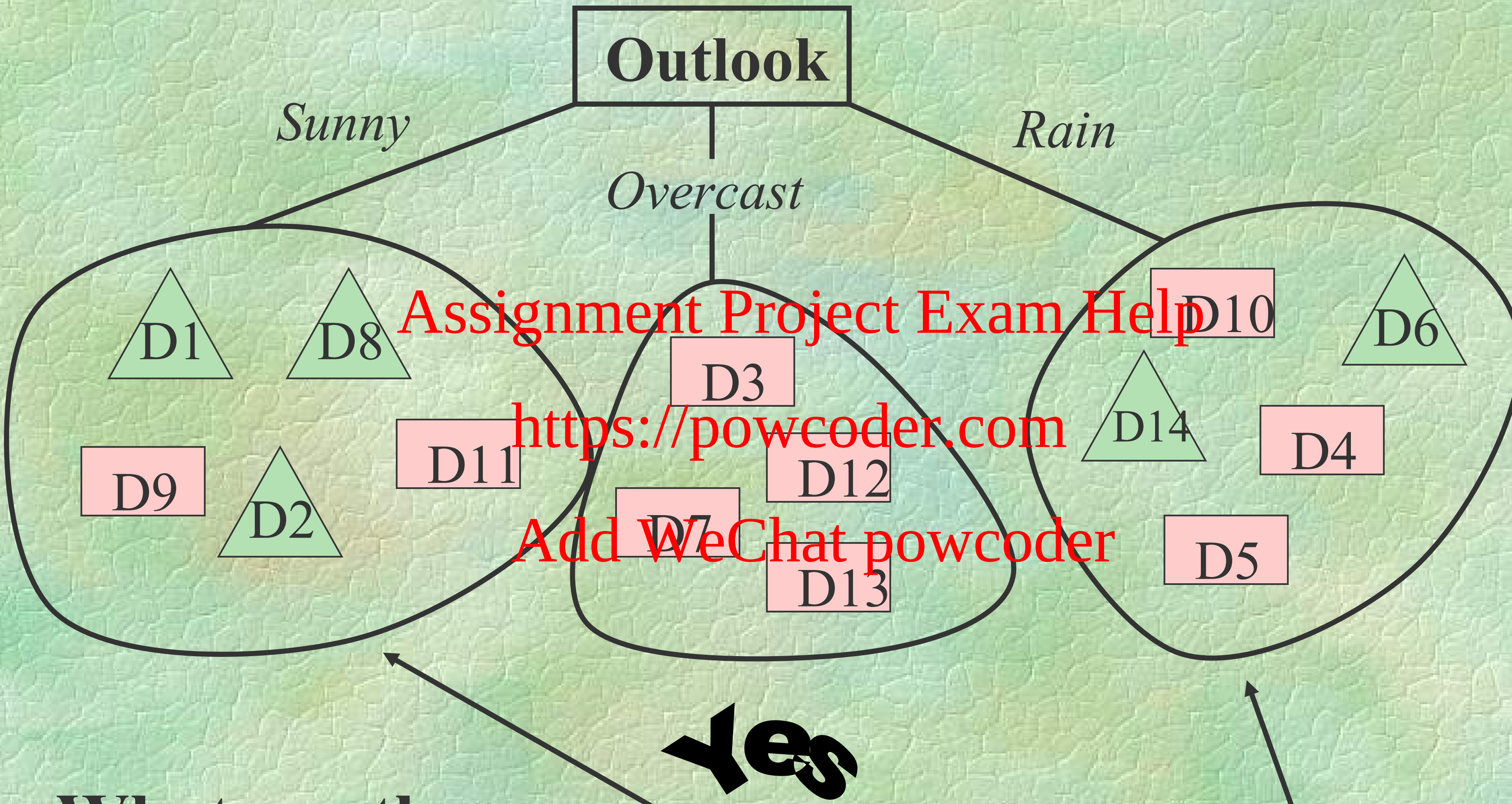
What is the "best" attribute?

Answer: Outlook

["best" = with highest information gain]

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

ID3 (Cont'd)



What are the "best" attributes? **Humidity** and **Wind**

What Attribute to choose to "best" split a node?

- Choose the attribute that minimizes the **Disorder (or Entropy)** in the subtree rooted at a given node.
- **Disorder** and **Information** are related as follows: the more disorderly a set, the more information is required to correctly guess an element of that set.
- **Information:** What is the best strategy for guessing a number from a finite set of possible numbers? i.e., how many questions do you need to ask in order to know the answer (we are looking for the minimal number of questions). Answer: $log_2|S|$, where S is the set of numbers and |S|, its cardinality.

Q1: is it smaller than 5?
Q2: is it smaller than 2?

E.g.: 0 1 2 3 4 5 6 7 8 9 10

Q2 Q1

Entropy (1)

- The entropy of a set is a measure for characterizing the degree of disorder or impurity in a collection of examples.
- The idea was borrowed from the field of thermodynamics and relates to the states (Gas/Liquid/Solid) of a system.

- For classification, we use the following formula:

$$Entropy(S) = -((p+) * log_2\,(p+)) - ((p-) * log_2\,(p-))$$

Where $p+$ / $p-$ represent the proportions of positive / negative examples, in S

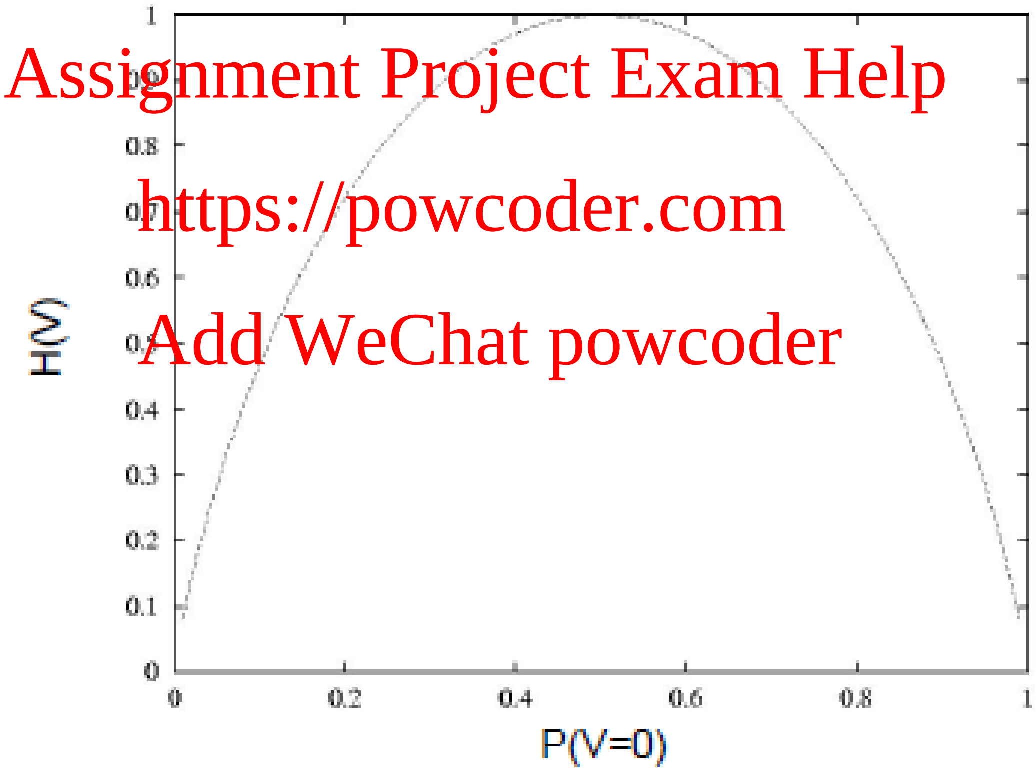
Entropy (2)

- The entropy can also be thought of as the minimum number of bits of information necessary to encode the classification of an arbitrary member of S.

- Examples:
  - If p+ is 1 ➔ All the examples are positive, i.e., no information is needed, *Entropy(S)=0*
  - If p+ is ½ ➔ 1 bit is required to indicate whether the example is positive or negative, Entropy(S) = 1
  - If p+ is .8 ➔ the class can be encoded by (on average) less than 1 bit by giving short codes to positive examples and large codes to negative ones

Entropy (3)

The entropy H(V) of a Boolean random variable V as the probability of V = 0 varies from 0 to 1



Bryan Pardo, EECS 349 Fall 2009

Entropy (4)

- When more than 2 classes are present, we have:

  - Entropy(S) $= \sum_{k=0}^{c} -p_i \log_2 p_i$

where c is the total number of classes

Information Gain (1)

- The information gain is a measure of the effectiveness of an attribute in classifying the training data.

- The information gain is also the expected reduction in entropy caused by partitioning the examples according to this attribute.
- Gain(S, A), is the expected reduction in entropy caused by knowing the value of Attribute A.

Information Gain (2)

- Gain(S, A) = Entropy(S) –

$$\sum_{v \in Values(A)} (|Sv|/|S|) \text{ Entropy}(Sv)$$

A = some attribute/feature

$Values(A)$ = set of possible values for attribute A.

Sv = subset of S for which Attribute A has value v.

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

Example: PlayTennis

- Calculate the Information Gain of attribute Wind
  - Step 1: Calculate Entropy(S)
  - Step 2: Calculate Gain(S, Wind)

- Calculate the Information Gain of all the other attributes (outlook, temperature and humidity)

- Choose the attribute with the highest information gain.

<See the calculations in class>

Hypothesis Space Search in Decision Tree Learning

- **Hypothesis Space**: Set of possible decision trees (i.e., complete space of finite discrete-valued functions).
- **Search Method**: Simple-to-Complex *Hill-Climbing* Search (only a single current hypothesis is maintained (≠ from candidate-elimination method)). **No Backtracking!!!**
- **Evaluation Function**: Information Gain Measure
- **Batch Learning:** ID3 uses all training examples at each step to make statistically-based decisions (≠ from candidate-elimination method which makes decisions incrementally). ==> the search is less sensitive to errors in individual training examples.

Inductive Bias in Decision Tree Learning

- **ID3's Inductive Bias:** *Shorter* trees are preferred over longer trees. Trees that place *high information gain attributes close to the root* are preferred over those that do not.

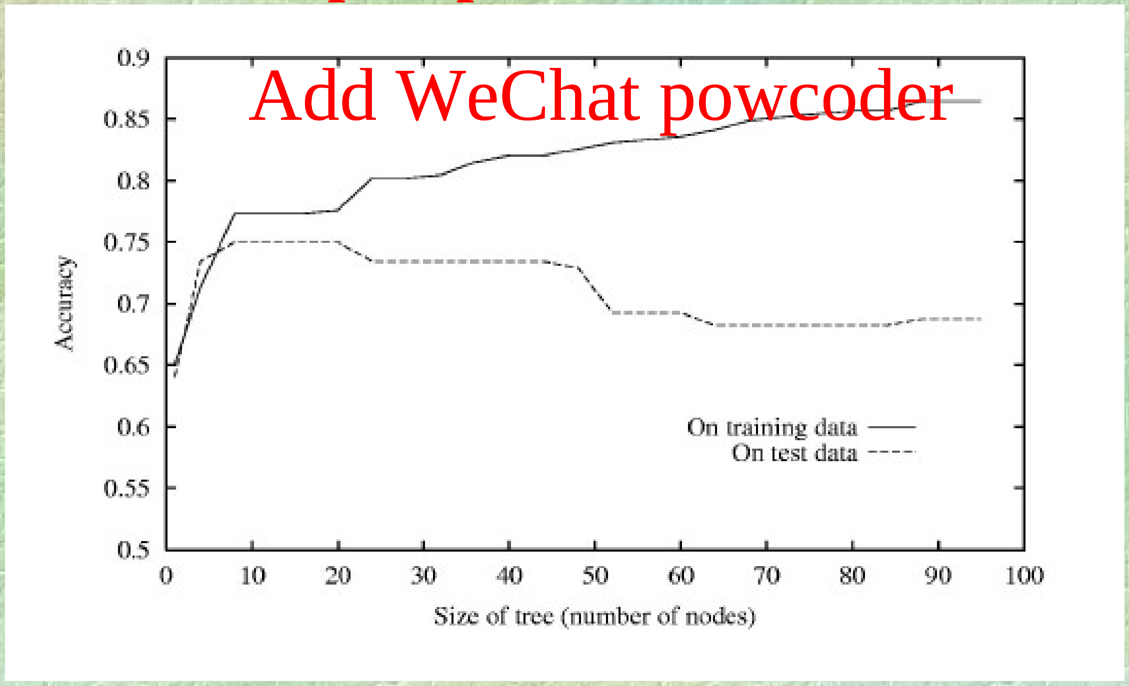
- **Note:** this type of bias is different from the type of bias used by Candidate-Elimination: the inductive bias of ID3 follows from its search strategy (***preference*** or ***search*** bias) whereas the inductive bias of the Candidate-Elimination algorithm follows from the definition of its hypothesis space (***restriction*** or ***language*** bias).

Why Prefer Short Hypotheses?

- **Occam's razor:** **Prefer the simplest hypothesis that fits the data** [William of Occam (Philosopher), circa 1320]
- Scientists seem to do that: E.g., Physicist seem to prefer simple explanations for the motion of planets, over more complex ones
- **Argument:** Since there are fewer short hypotheses than long ones, it is less likely that one will find a short hypothesis that coincidentally fits the training data.
- **Problem with this argument:** it can be made about many other constraints. Why is the "short description" constraint more relevant than others?
- **Nevertheless:** Occam's razor was shown experimentally to be a successful strategy!

Issues in Decision Tree Learning: I. Overfitting

- **Definition:** Given a hypothesis space *H*, a hypothesis *h*∈*H* is said to ***overfit*** the training data if there exists some alternative hypothesis *h'*∈*H*, such that *h* has smaller error than *h'* over the training examples, but *h'* has a smaller error than *h* over the entire distribution of instances.



Avoiding Overfitting the Data (1)

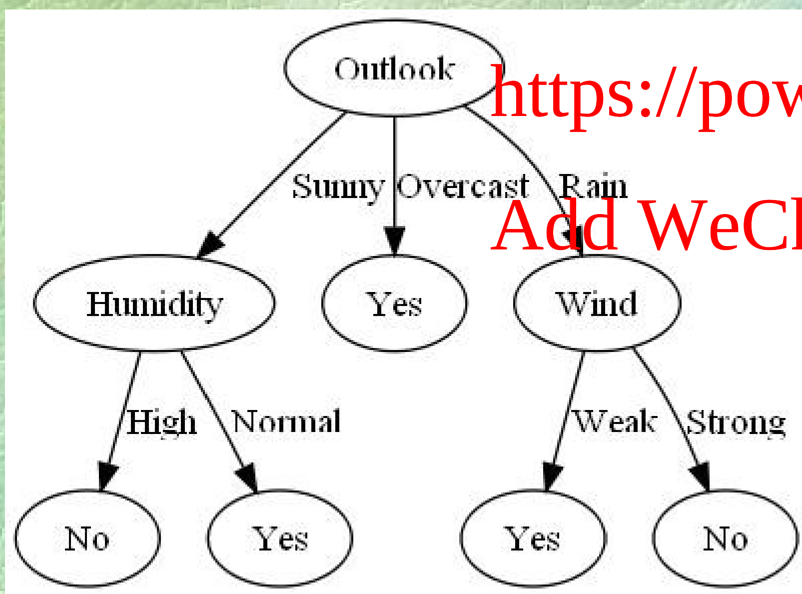
- There are two approaches for overfitting avoidance in Decision Trees:
  - Stop growing the tree before it perfectly fits the data
  - Allow the tree to overfit the data, and then ***post-prune*** it.

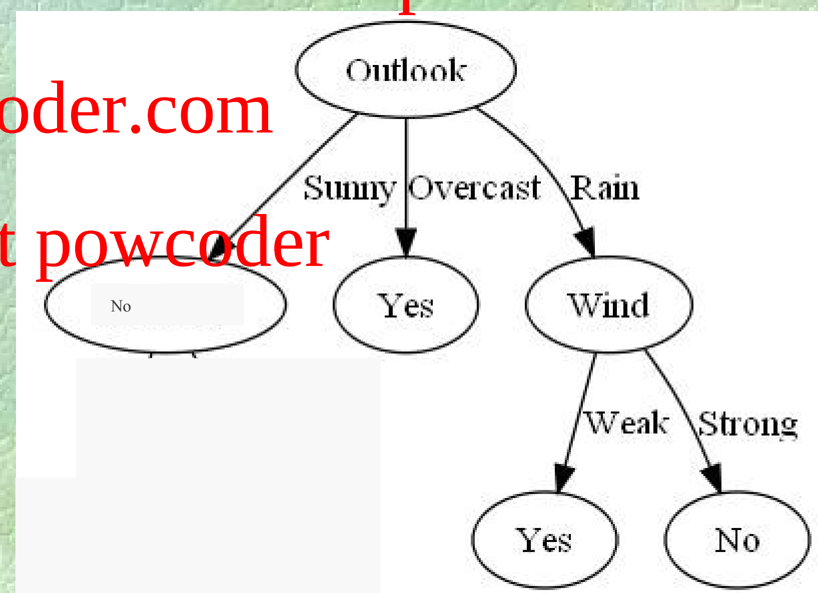
Avoiding Overfitting the Data (2)

- There are three criterion that can be used to determine the optimal final tree size:
  - Train and Validation Set Approach: use a separate set of examples (distinct from the training examples) to evaluate the utility → Reduced Error Pruning
  - Use all the available training data but apply a statistical test to estimate the effect of expanding or pruning a particular node.
  - Use an explicit measure of complexity for encoding the training examples and the decision tree.

Avoiding Overfitting the data (3) Reduced Error Prunning

1. Consider each of the decision nodes in the tree
2. For each node, compare the performance of the tree with that node expanded or not. Example:





3. If Right Tree is as accurate as Left Tree over the validation set then prune that node.

Avoiding Overfitting the data (4) Rule Post-Pruning

This is the strategy used in C4. 5:

- Grow a Tree
- Convert the tree into sets of rules
- Prune (generalize) each rule by removing any preconditions that result in improving its estimated accuracy.
- Sort the pruned rules by their estimated accuracy and consider them in this sequence when classifying subsequent instances.

Issues in Decision Tree Learning: II. Other Issues

- Incorporating Continuous-Valued Attributes

- Alternative Measures for Selecting Attributes

- Handling Training Examples with Missing Attribute Values
- Handling Attributes with Differing Costs