Decision Tree Learning

Adrian Barbu

Play Tennis Dataset

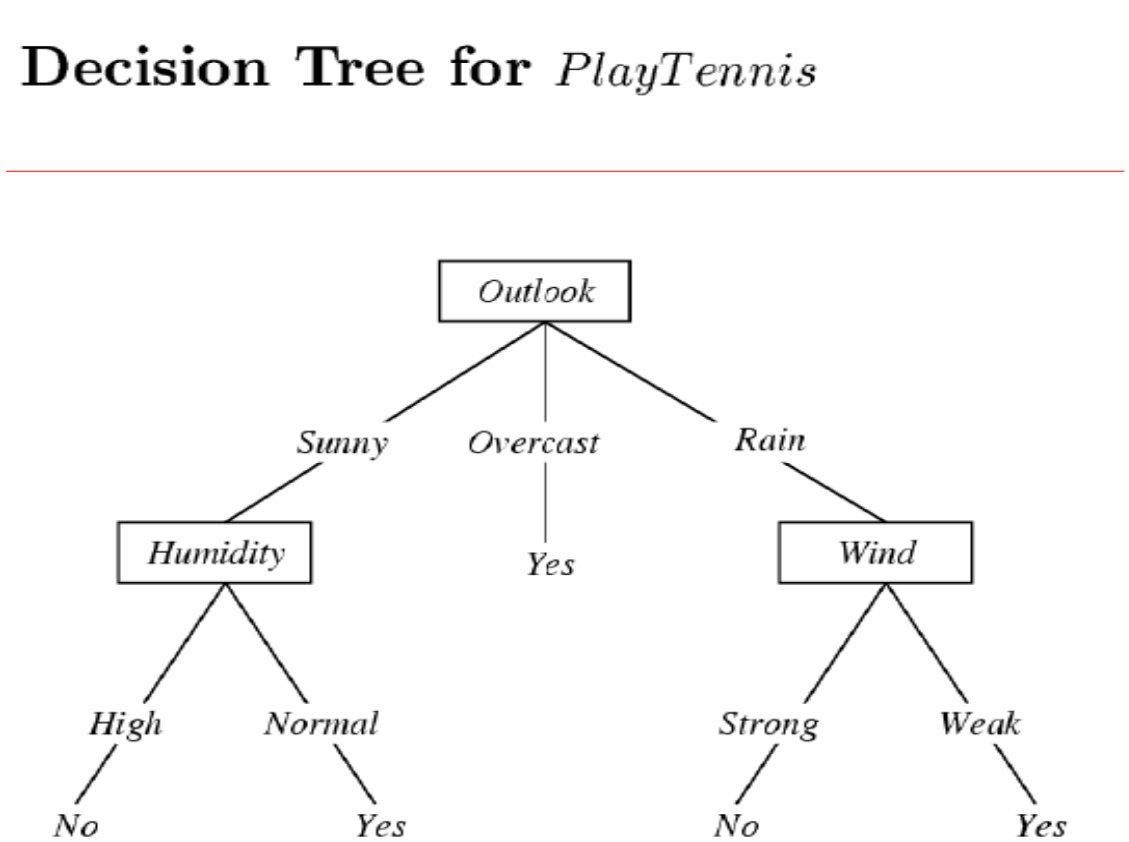
- Goal: decide whether to play tennis or not
- Attributes (features):
  - Outlook
  - Temperature
  - Humidity
  - Wind

Training set

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

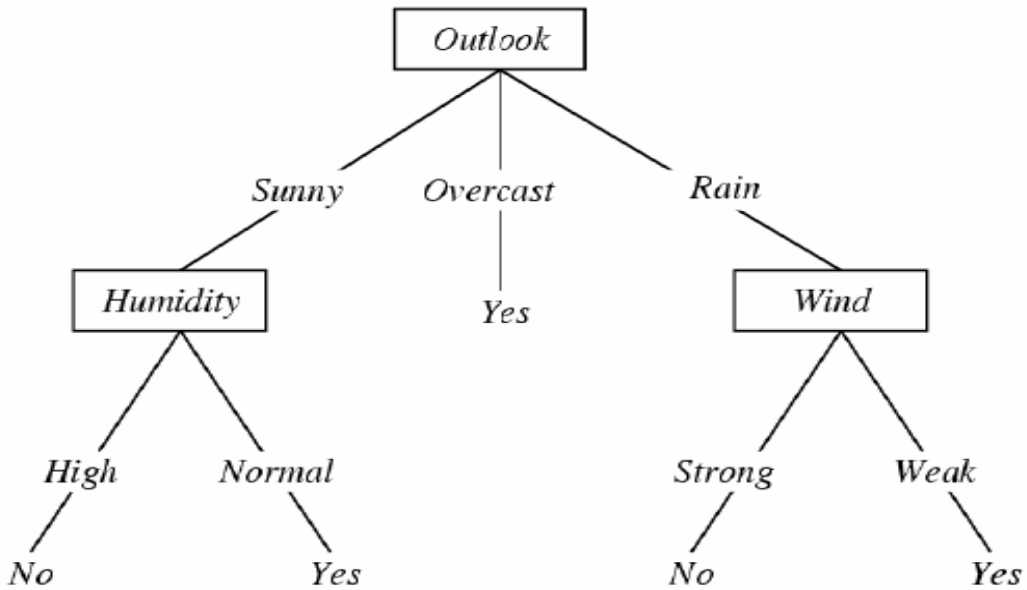
- Decision cannot be done based on a single attribute
- 20-questions kind of game:



Decision Tree Learning

- Used mostly for classification
- A set of discrete attributes (features) arranged in a tree
- Leaf nodes are the classification results or a probability.
- Based on the attribute values, only one path from root to leaves is selected

Decision Tree for *PlayTennis*



Learning to Predict Emergency C-sections

Sims et al., 2000

- Data set
  - 9714 records
  - 215 features each

Data:

Patient103 time=1	*Patient103* time=2	*Patient103* time=n
Age: 23	Age: 23	Age: 23
FirstPregnancy: no	FirstPregnancy: no	FirstPregnancy: no
Anemia: no	Anemia: no	Anemia: no
Diabetes: no	Diabetes: YES	Diabetes: no
PreviousPrematureBirth: no	PreviousPrematureBirth: no	PreviousPrematureBirth: no
Ultrasound: ?	Ultrasound: abnormal	Ultrasound: ?
Elective C–Section: ?	Elective C–Section: no	Elective C–Section: no
Emergency C–Section: ?	Emergency C–Section: ?	**Emergency C–Section: Yes**

One of 18 learned rules:

```
If   No previous vaginal delivery, and
     Abnormal 2nd Trimester Ultrasound, and
     Malpresentation at admission
Then Probability of Emergency C-Section is 0.6

 Over training data: 26/41 = .63,
 Over test data: 12/20 = .60
```

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+ .
| | | | Birth_Weight >= 3349: [133+,36.4-] .78+
| | | 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-
```

When to use Decision Trees?

We can use decision trees when:

- Can represent instances using a set of attributes and their values
- The target function is discrete: binary or multi-class *classification*

Advantages:

- Robust to errors in training data
- Training data can contain missing attribute values

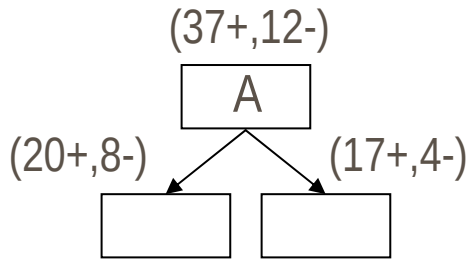
Disadvantages

- Easy to overfit

Basic Decision Tree Learning Algorithm for Binary Classification

ID3, C4.5

node=root



Main loop:

1. A=attribute with highest information gain
2. Assign A as the decision attribute for *node*
3. For each value of A, create a descendant from *node*
4. Take training examples to leaf nodes
5. If all examples are perfectly classified STOP, else iterate loop over the new leaf nodes

Entropy

- Entropy of a random variable X

$$H(X) = -\sum_{i=1}^{n} P(X = i) \log_2 P(X = i)$$

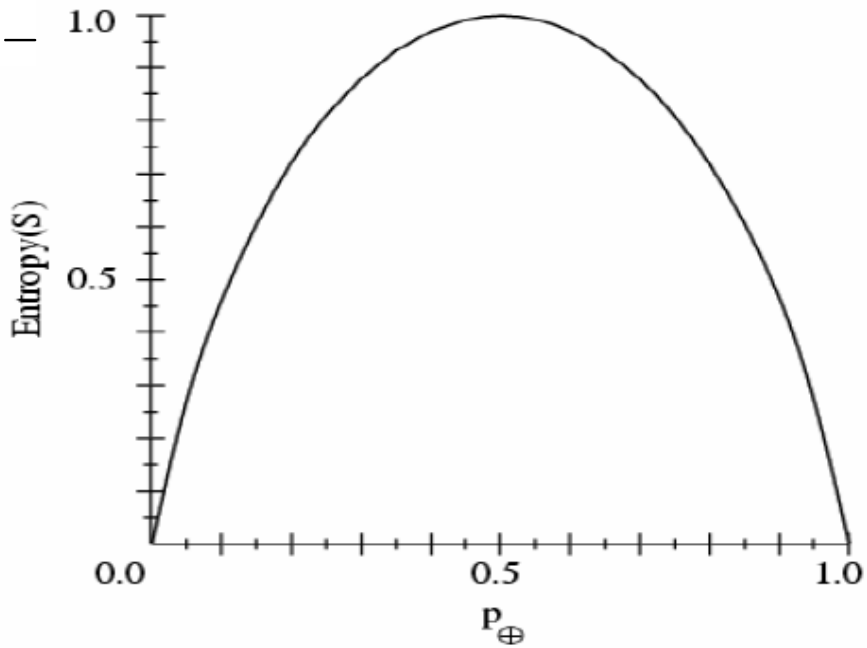
- It is the average number of bits required to optimally encode a random sample from X
- The optimal encoding is the Huffman coding
- Entropy is a measure of the degree of randomness
  - Smaller entropy=less randomness

Entropy for Training Examples

- At a node, we have (*a*+,*b*-), i.e. *a* positive samples and *b* negative examples reached this node.
- S=(*a*+,*b*-) means $p_+ = a/(a+b)$, $p_- = b/(a+b)$
- H(S)=entropy of S can be viewed as a function of p_+

$$H(S) = -p_+ \log_2 p_+ - p_- \log_2 p_-$$

- Entropy is max for uniform S.



Mutual Information

- Conditional entropy of X given Y=y

$$H(X|Y=y) = -\sum_{x \in Val(X)}^{n} P(X=x|Y=y) \log_2 P(X=x|Y=y)$$

- Conditional entropy of X given Y

$$H(X|Y) = \sum_{y \in Val(Y)} P(Y=y) H(X|Y=y)$$

- Mutual Information (Information Gain) of X and Y

$$I(X,Y) = H(X) - H(X|Y) = H(Y) - H(Y|X)$$

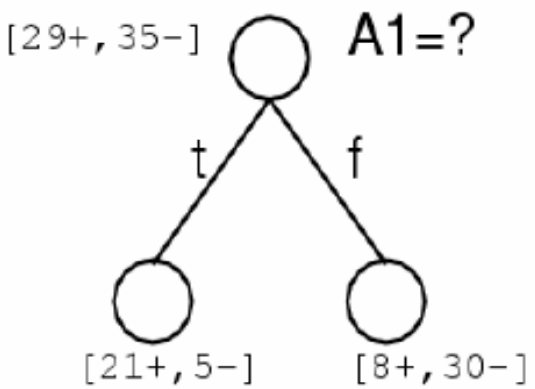
$$I(X,Y) = \sum_{x \in X} \sum_{y \in Y} P(x,y) \log\left(\frac{P(x,y)}{P(x)P(y)}\right)$$

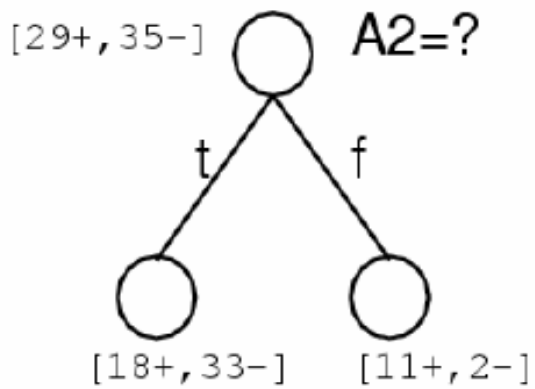
Information Gain

- Let S=(a+,b-), A an attribute (feature)
- Information gain of S and A

$$Gain(S, A) = H(S) - \sum_{v \in Val(A)} \frac{|S_v|}{|S|} H(S_v)$$

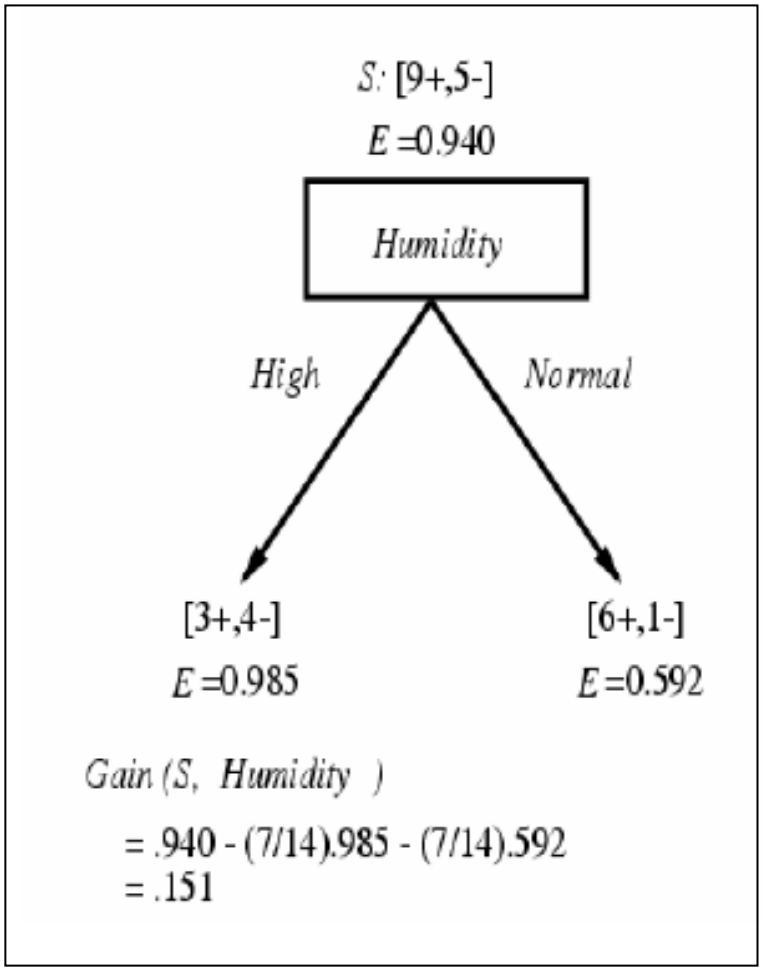
- *Gain*(S,A) = mutual information between A and target function over the samples of S

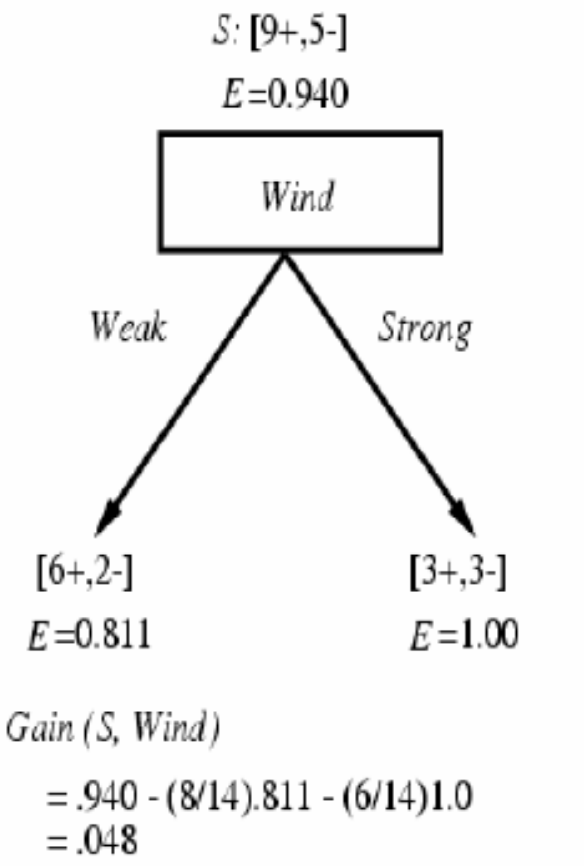




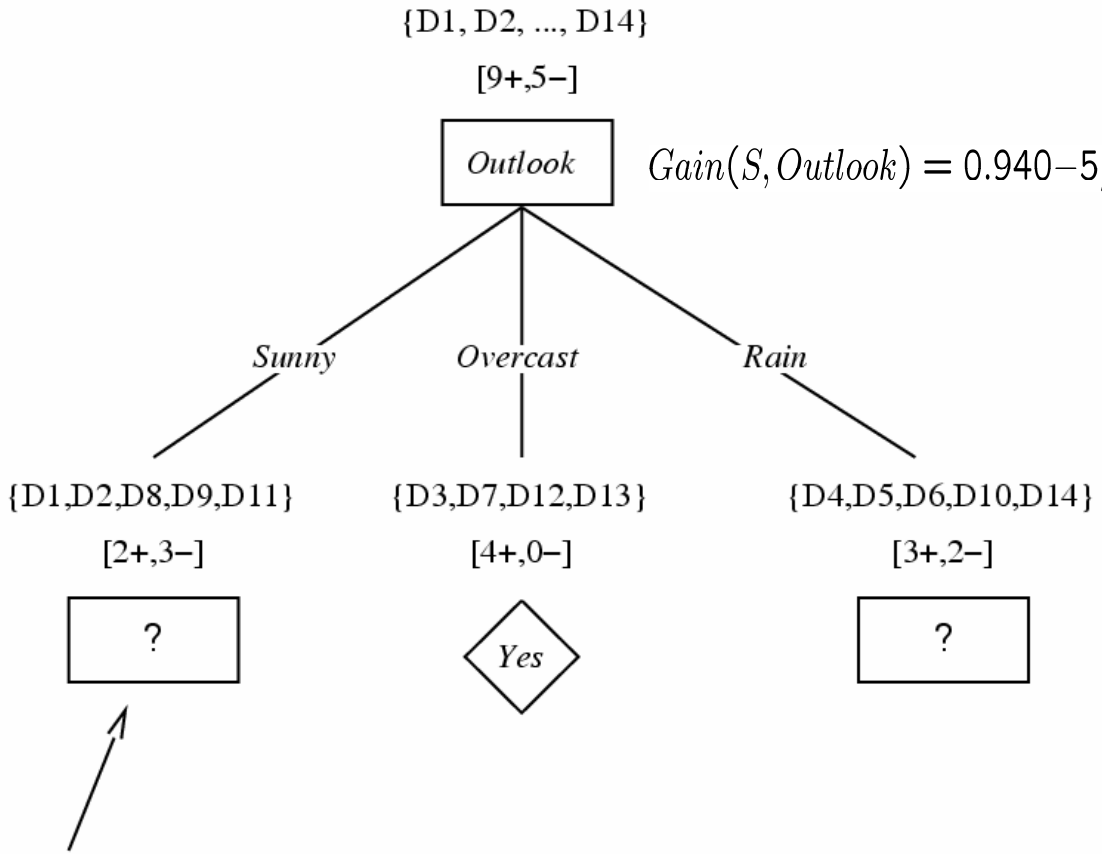
Information Gain

- Which attribute is best?





Play Tennis Example



$Gain(S, Outlook) = 0.940 - 5/14 * 0.971 - 0 - 5/14 * 0.971 = 0.246$

Which attribute should be tested here?

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

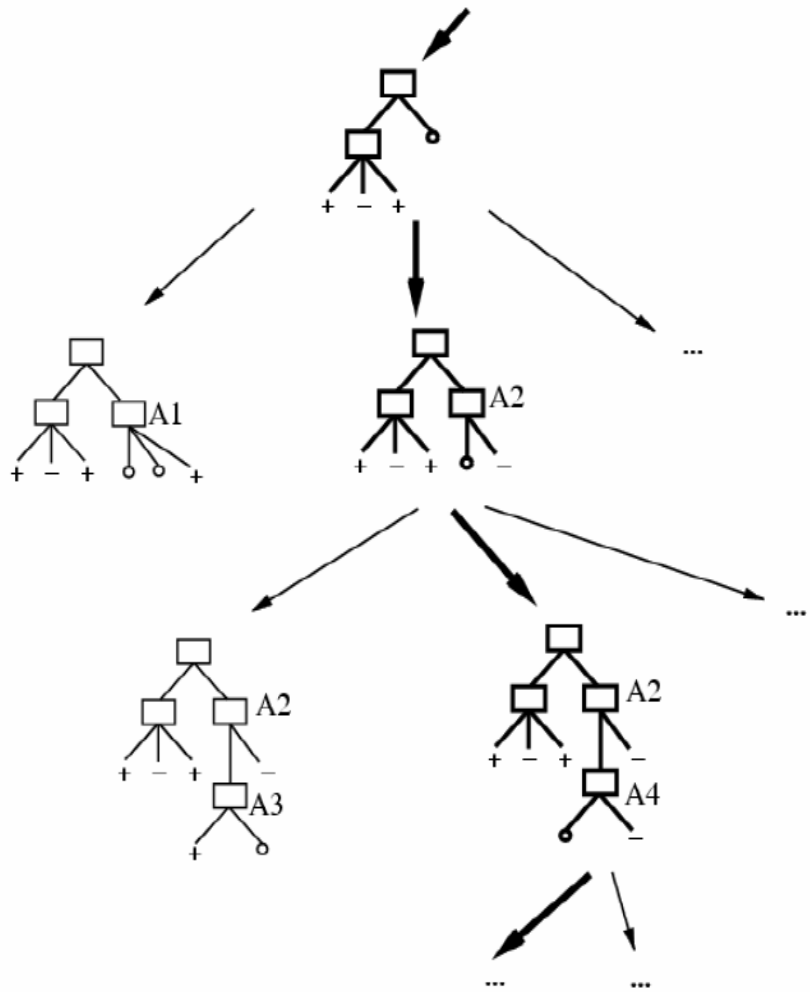
Decision Tree Learning

- The space of allowable functions

$$\mathcal{H} \subset \{h : X \rightarrow Y\}$$

  is the space of decision trees
- ID3 = greedy search in $\mathcal{H}$ using the information gain
- Greedy = suboptimal
- Prefers smaller trees
- This is good!

  Occam's razor



Occam's Razor

- William of Occam, philosopher, (*c*. 1285–1349)

  Entia non sunt multiplicanda praeter necessitatem or "Entities should not be multiplied unnecessarily."
- In other words:

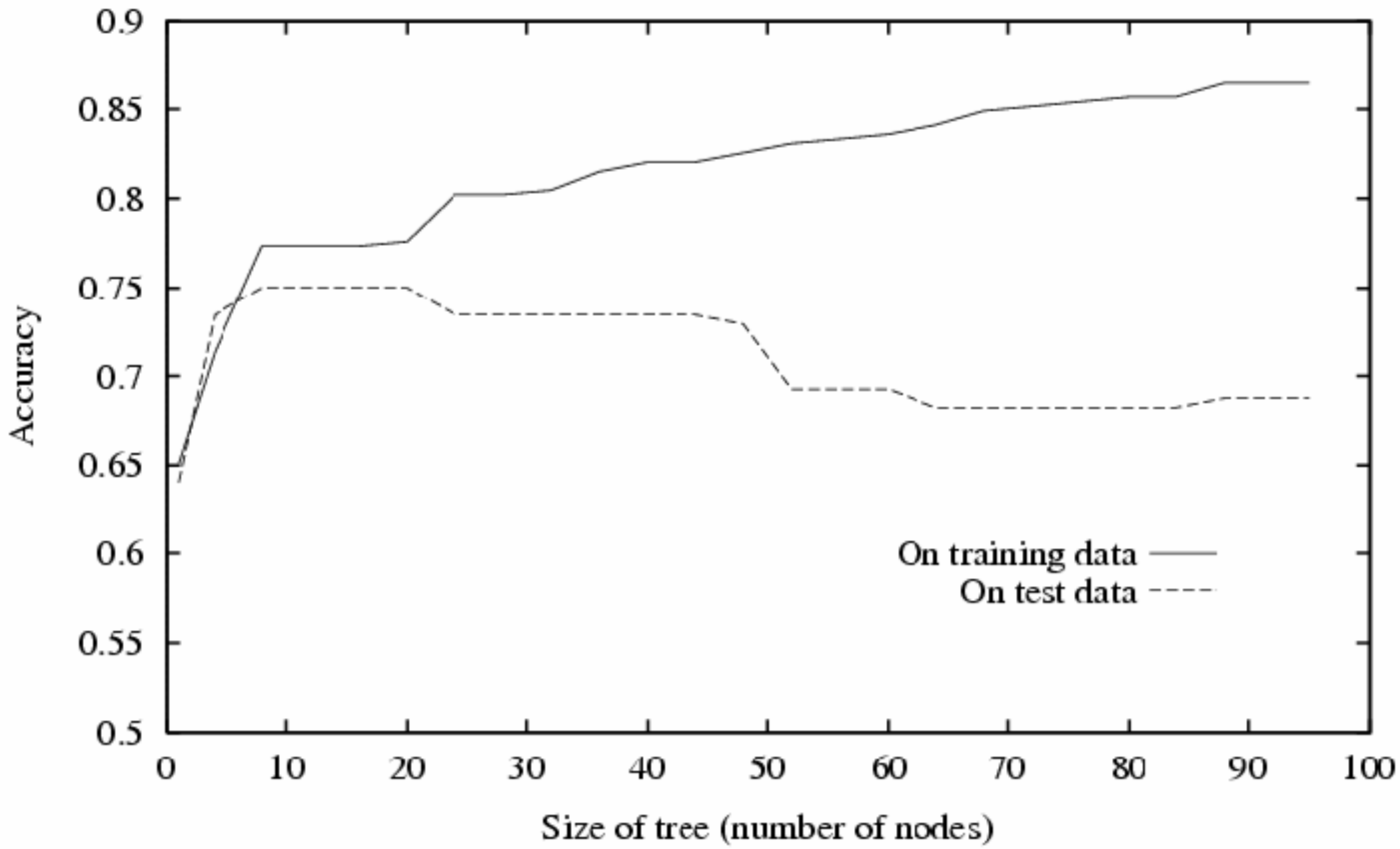
  "All other things being equal, the simplest solution is the best "
- Simpler explanations generalize better

- For Decision Trees

  Smaller trees that explain the training data are preferred.

Overfitting in Decision Trees

- Must balance tree size and accuracy



- Size>25 overfits the data in this example

Overfitting

- Hypothesis $h \in \mathcal{H}$ overfits the data if there exists $h' \in \mathcal{H}$ such that
  - $Err_{train}(h) < Err_{train}(h')$
  - Over the entire data we have

$$Err_D(h) > Err_D(h')$$

- E.g. the decision tree of size 60 from the graph

Avoiding Overfitting

Approaches:

- Stop growing the tree before it perfectly classifies the data
- Grow full-tree, then post-prune

Criteria:

- Use a separate set of examples (validation set) that was not used for training
- Use a statistical test for stopping the tree growth
- Use Minimum Description Length to measure tree complexity

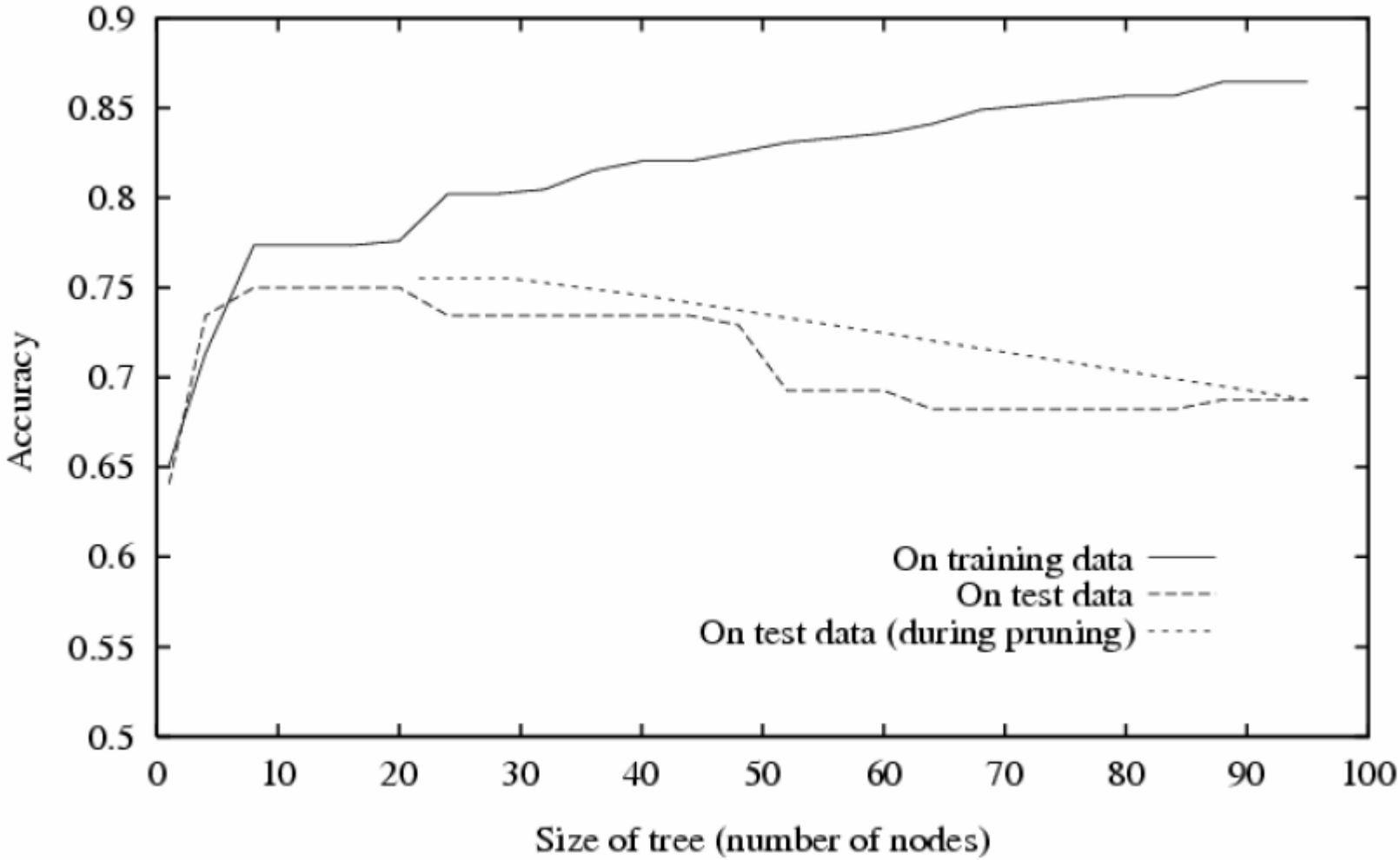
Reduced Error Pruning

- Use disjoint *training* and *validation* sets.
- Grow full tree on the training set
- Do while possible
  1. For each node, evaluate system accuracy after the node (and its subtree) is removed
  2. Greedily remove the node that gives largest increase in accuracy

Useful when lots of data is available

Reduced Error Pruning

- Improvement on unseen data (not used for validation)

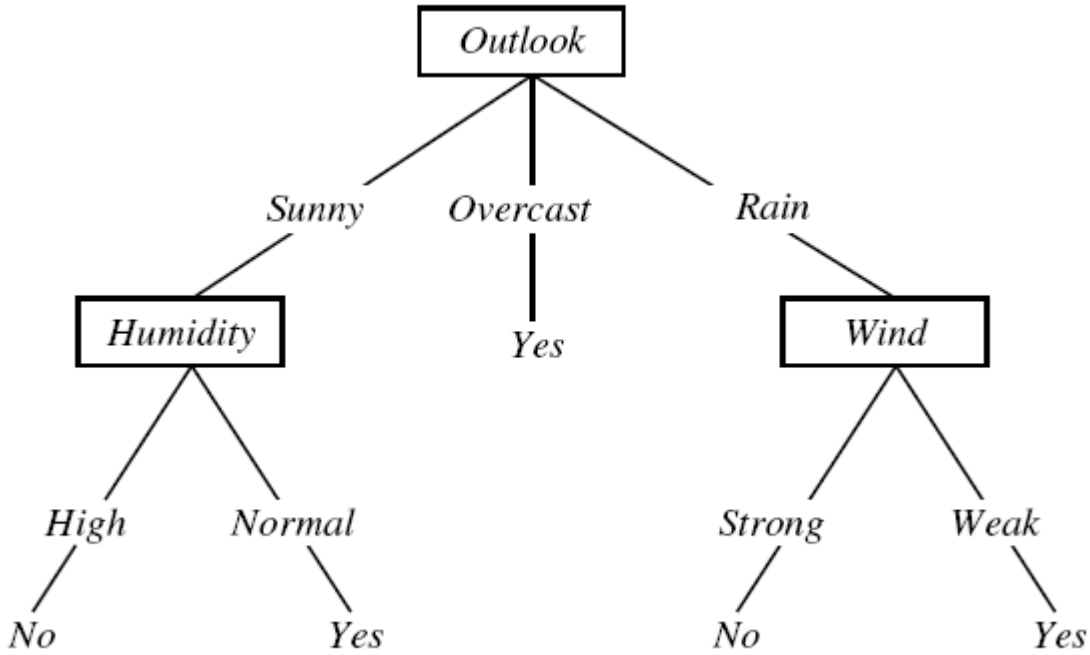


Rule Post-Pruning

1. Grow Full Tree
2. Convert the tree to a set of rules
   - one rule for each path from the root to a leaf.
3. Simplify each rule by removing any of the components if the accuracy increases
4. Sort the pruned rules by their accuracy and estimate them in this order

- How to prune:
  - Can use validation set
  - Can use a pessimistic measure of accuracy (C4.5, Quinlan 1993)

Converting a Tree to Rules



- One Rule for each leaf

$$\text{IF} \quad (Outlook = Sunny) \wedge (Humidity = High)$$
$$\text{THEN} \quad PlayTennis = No$$

$$\text{IF} \quad (Outlook = Sunny) \wedge (Humidity = Normal)$$
$$\text{THEN} \quad PlayTennis = Yes$$

$$\ldots$$

Handling Continuous Value Attributes

- Convert to discrete attributes using one or more thresholds
- E.g

$$(Temperature > 72.3) = t, f$$

$$(Temperature > 54) \wedge (Temperature < 85) = t, f$$

- Use thresholds between values where the target function changes

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

Alternative Measures for Attribute Selection

- Information Gain prefers attributes with many values
  - E.g. use *Date* as an attribute for playing tennis
  - Perfect fit on training data, but no generalization power
- Another measure = *Gain Ratio*

$$SplitInformation(S,A) = -\sum_{v \in Val(A)}^{c} \frac{|S_v|}{|S|} \log_2 \frac{|S_v|}{|S|}$$

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

Handling Missing Attribute Values

- What if there are training examples with missing attribute A
- Sort through the tree until node A is reached
- Three strategies
  - Assign to A the most common value
  - Assign to A the most common value for the same target
  - Assign to A all values with weights given by the frequencies take fractions of example through all the descendants
- New samples are classified in the same fashion

Conclusions

- Decision trees can learn a target function from training examples
- An important issue is overfitting
  - Use validation set
  - Smaller trees are preferred
  - Reduced Error Pruning
  - Rule Post-Pruning
- Decision Trees are used in recent learning advances
  - Random Forest
  - Boosting