

Decision Tree Learning

Vandi Verma
(CMU), Fall 2000



Lecture Outline

- Tree representations
- Learning Trees
- Overfitting and Occam's Razor
- Extensions



Supervised Learning: *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	Hot	Normal	Weak	Yes
D10	Rain	Mild	Normal	Strong	Yes
D11	Sunny	Cool	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

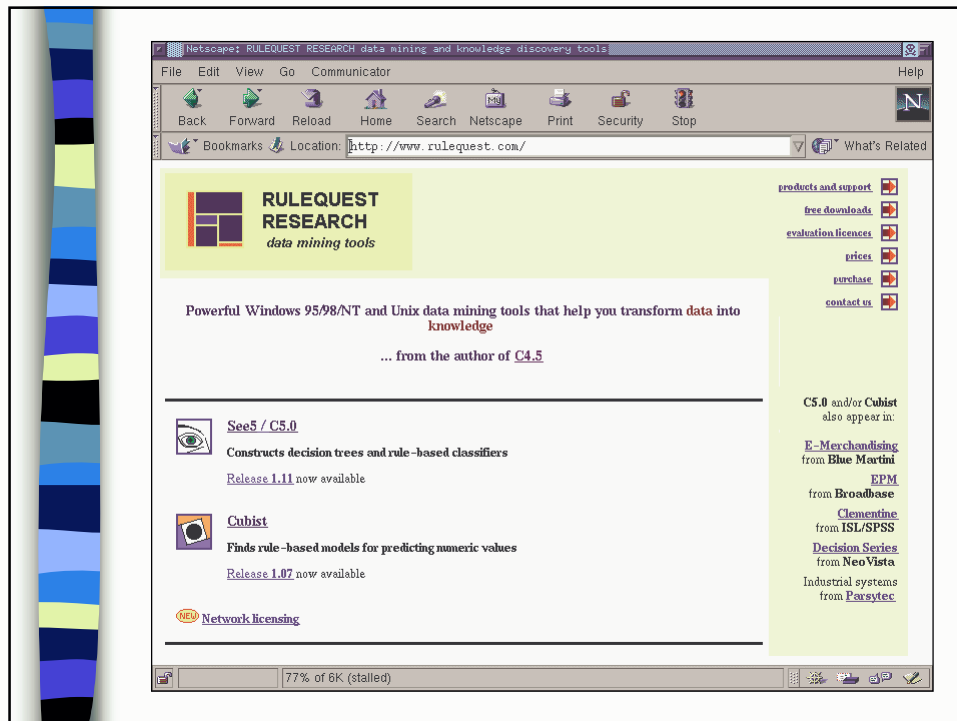


Decision Tree Learning

Extremely popular

- Credit risk assessment
- Medical diagnosis
- Market analysis
- Production control
- Poisonous mushroom detection ☺

...and many other domains with symbolic features



Outline

- Decision Trees
- ID3 Learning Algorithm
- Overfitting and Pruning
- Extension

Trees



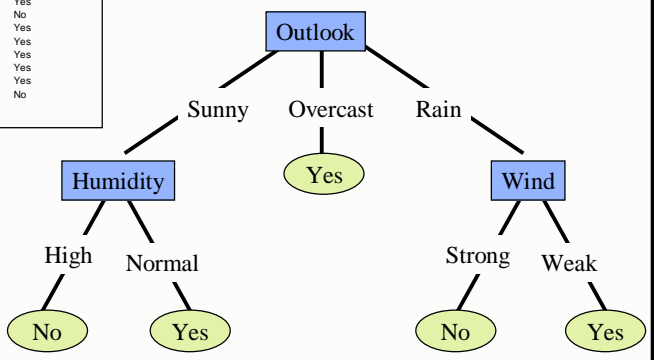
Training Examples *PlayTennis*

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
S2	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	Hot	Normal	Weak	Yes
D10	Rain	Mild	Normal	Strong	Yes
D11	Sunny	Cool	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*

Training Examples *PlayTennis*

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
S2	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	Hot	Normal	Weak	Yes
D10	Rain	Mild	Normal	Strong	Yes
D11	Sunny	Cool	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No



Representational Power

- Q: Can trees represent arbitrary Boolean expressions?
- A: Yes.
- Q: How many Boolean functions are there over N attributes?
- A: 2^{2^N}
- Q: How would we represent
 $A \wedge B$ $A \vee B$ $A \text{ XOR } B$



When To Consider Decision Trees

- Instances describable by attribute-value pair
- Target function discrete valued
- Possibly noisy training data

Examples:

- Medical diagnosis
- Credit risk analysis
- Mushrooms ☺



Lecture Outline

- Tree representations
- Learning Trees
- Overfitting and Occam's Razor
- Extensions

Learning: How to Generate trees?

- Learn trees from labeled data:
features -> class

Training Examples *PlayTennis*

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
S2	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	Hot	Normal	Weak	Yes
D10	Rain	Mild	Normal	Strong	Yes
D11	Sunny	Cool	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

A horribly Intractable Algorithm

- Generate all trees
- Check how well each tree describes the training set
- Pick the one that works best



A Better Algorithm

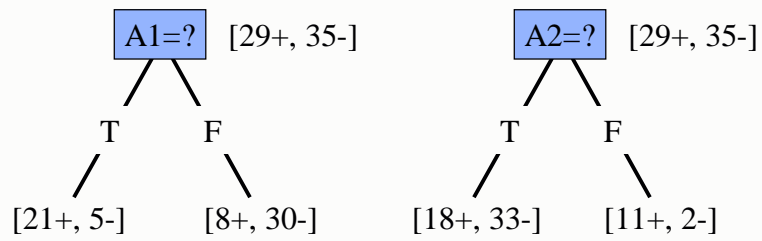
- Choose best attribute
- Split data set
- Recurse until each data item classified correctly



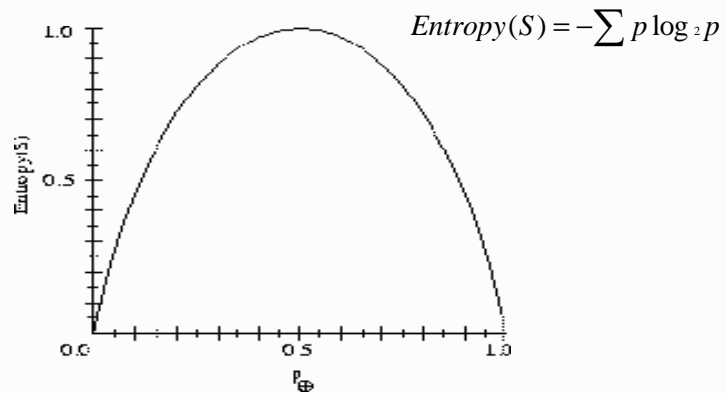
Top-Down Induction of DTs (ID3)

```
proc growtree(data)
  if (data not perfectly classified)
    find `best' splitting attribute A
    for each (a in A)
      create child a
      data_a = data restricted to A=a
      growtree(data_a)
    endfor
  endif
endproc
```

Which Attribute Is Best?



Entropy!



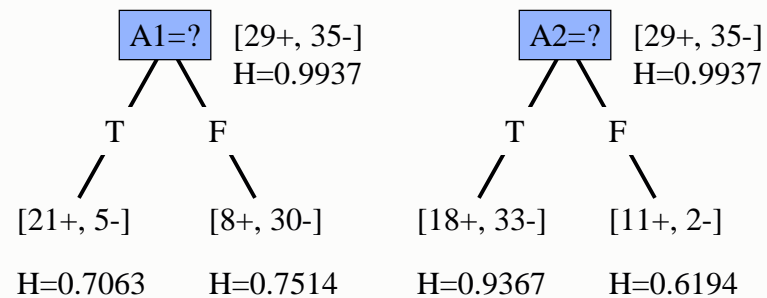
- Entropy measures the “impurity” of a sample set

Information Gain

- $Gain(S, A) =$ expected reduction in entropy if we know value of A

$$Gain(S, A) = Entropy(S) - \sum_{a \in A} \frac{|S_a|}{|S|} Entropy(S_a)$$

Which Attribute Is Best?



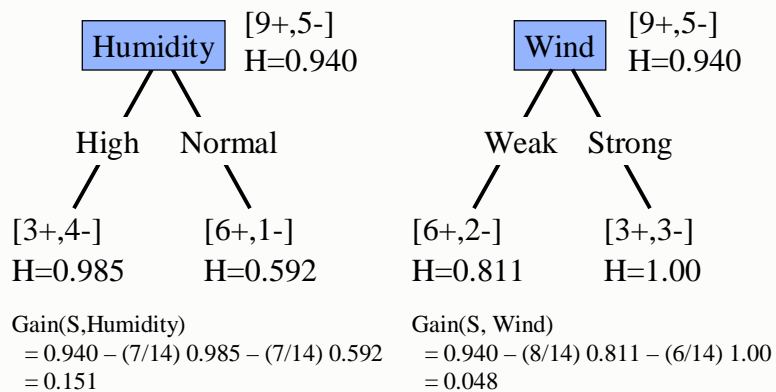
$$\begin{aligned}
 \text{Gain}(S, A1) &= 0.9937 - (26/64) 0.71 - (38/64) 0.75 \\
 &= 0.2606
 \end{aligned}$$

$$\begin{aligned}
 \text{Gain}(S, A2) &= 0.9937 - (51/64) 0.94 - (13/64) 0.62 \\
 &= 0.1215
 \end{aligned}$$

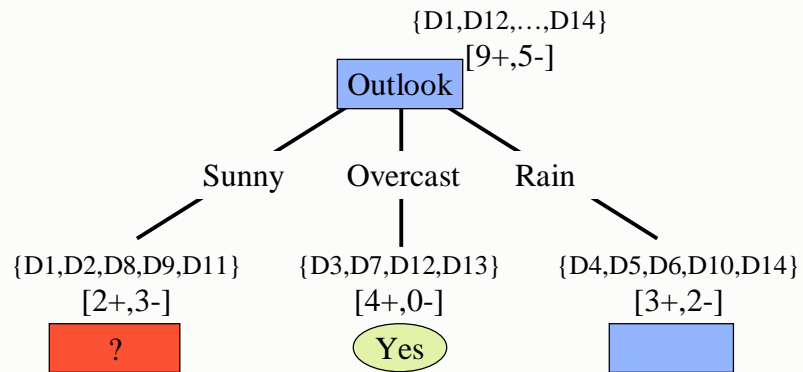
Training Examples *PlayTennis*

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	Hot	Normal	Weak	Yes
D10	Rain	Mild	Normal	Strong	Yes
D11	Sunny	Cool	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Selecting The First Attribute



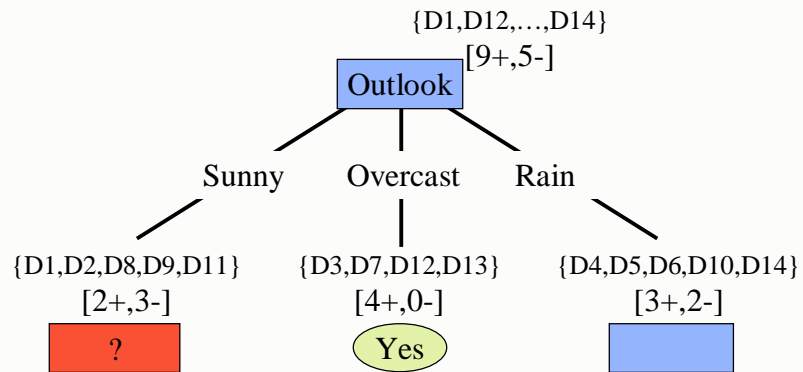
A Second Example



Training Examples *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	Hot	Normal	Weak	Yes
D10	Rain	Mild	Normal	Strong	Yes
D11	Sunny	Cool	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

A Second Example



$$\begin{aligned}
 \text{Gain}(S_{\text{sunny}}, \text{Temperature}) &= .970 - (3/5) .918 - (1/5) 1.0 - (1/5) 1.0 = .019 \\
 \text{Gain}(S_{\text{sunny}}, \text{Humidity}) &= .970 - (3/5) 0.0 - (2/5) 0.0 = .970 \\
 \text{Gain}(S_{\text{sunny}}, \text{Wind}) &= .970 - (2/5) 1.0 - (3/5) .918 = .019
 \end{aligned}$$

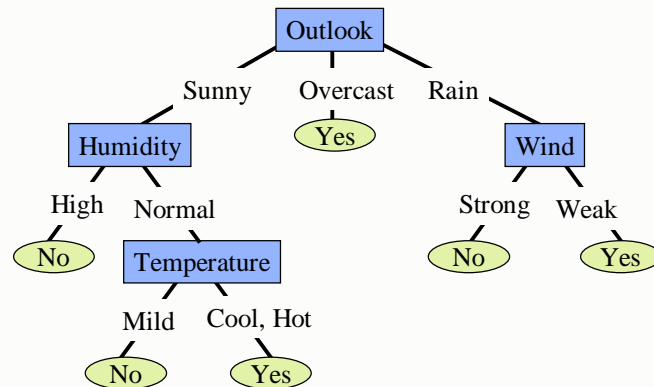
Lecture Outline

- Tree representations
- Learning Trees
- Overfitting and Occam's Razor
- Extensions

Overfitting in Decision Trees

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	Hot	Normal	Weak	Yes
D10	Rain	Mild	Normal	Strong	Yes
D11	Sunny	Cool	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No
D15	Sunny	Mild	Normal	Strong	No

Effect on Our Tree



Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D15	Sunny	Mild	Normal	Strong	No

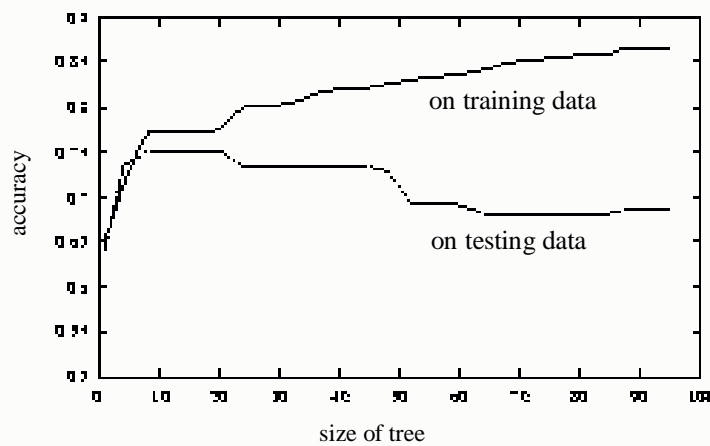
Overfitting

- Hypothesis h overfits iff $\exists h'$ with

$$\text{error}_{\text{train}}(h) < \text{error}_{\text{train}}(h')$$

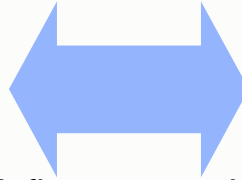
$$\text{error}_{\text{true}}(h) > \text{error}_{\text{true}}(h')$$

Overfitting in ID3



Bias Variance Trade-Off

High Bias

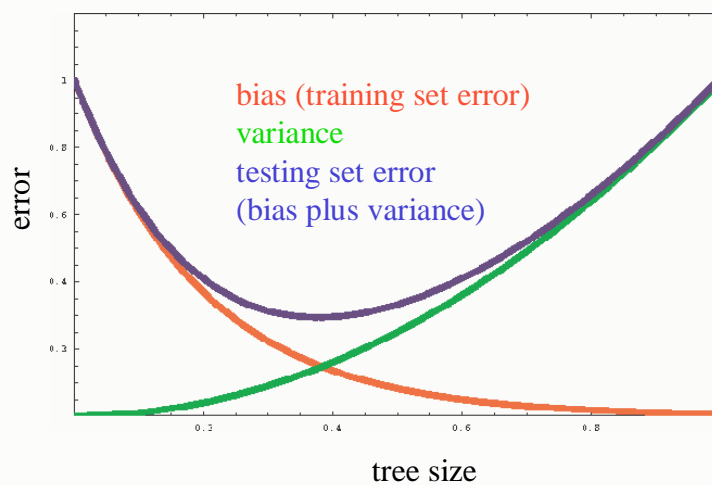


High Variance

Small trees can't fit
many data sets

Large trees sensitive
to randomness in
data selection

Bias Variance Trade-Off





Ockham's Razor

“Non sunt multiplicanda entia praeter necessitatem.”

William of Ockham (1285-1347)

In English: Entities are not to be multiplied beyond necessity (law of parsimony)

In machine learning (and science): Prefer simpler hypotheses over more complex ones (Occam's razor)



Ockham's Razor (cont'd)

“Everything should be made as simple as possible, *but not simpler*”

Albert Einstein (1879-1955)

“The only thing that interferes with my learning is my education”

“Common sense is the collection of prejudices acquired by age eighteen.”



How can we avoid overfitting?

- Stop growing in time
- Grow full tree, then prune
 - Reduced error pruning
 - Rule post-pruning
- (Mixing factors (shrinkage))

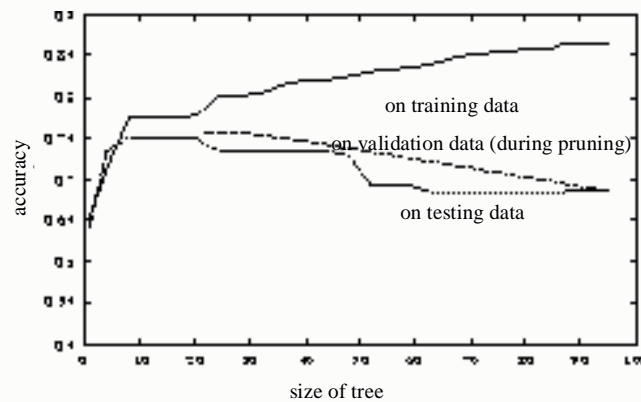


How to select “best” tree

- Measure performance over separate validation set
- MDL: minimize
$$\alpha \text{ size}(\text{tree}) + \#\text{misclassifications}(\text{tree})$$

Example: Reduced Error Pruning

- Split data into *training* and *validation* set
- Keep removing node that maximally increases validation set accuracy.



Rule Post-Pruning

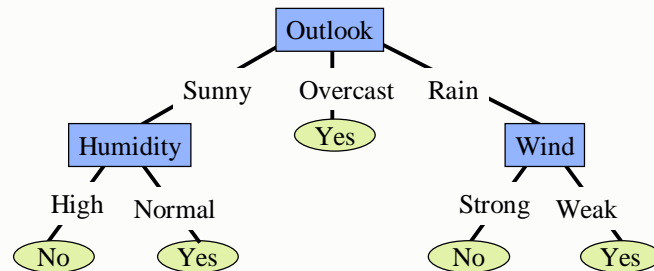
1. Convert tree into rules
2. Prune rules independently of each other
3. Sort final rules into desired sequence

Perhaps most frequently used method (eg, C4.5)

Lecture Outline

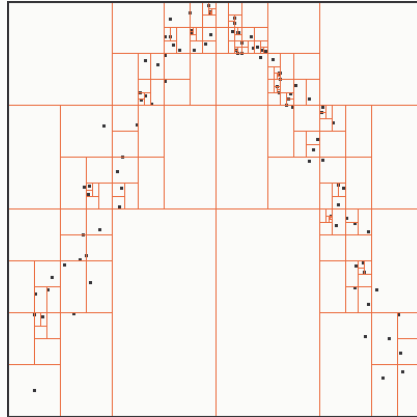
- Tree representations
- Learning Trees
- Overfitting and Occam's Razor
- Extensions

Converting A Tree to Rules



If (outlook = sunny) \wedge (humidity=high) then PlayTennis=No
If (outlook = sunny) \wedge (humidity=normal) then PlayTennis=Yes
If (outlook = overcast) then PlayTennis=Yes
If (outlook = rain) \wedge (wind=strong) then PlayTennis=No
If (outlook = rain) \wedge (wind=weak) then PlayTennis=Yes

Continuous Valued Attributes



- Finitely many tests!

Attributes with Many Values

- Imagine: Splitting on “Birthdate”??
- Idea: Use *GainRatio* instead of *InfoGain*

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

with

$$\text{SplitInformation}(S, A) = -\sum_{i=0}^c \frac{|S_i|}{|S|} \log_2 \frac{|S_i|}{|S|}$$



Splitting with Costs

- Example: Medical diagnosis: “*BloodTest*” costs \$150
- Ideas: Replace information gain by

$$\frac{Gain^2(S, A)}{Cost(A)} \quad (\text{Tan\&Schlimmer 90})$$

$$\frac{2^{Gain(S, A)} - 1}{Cost(A)} \quad (\text{Nunez 88})$$



Unknown Attribute Values

- Assign most common value in tree
- Assign most common value among training set
- Assign probability to each possible value of A

