

CSC421 Intro to Artificial Intelligence

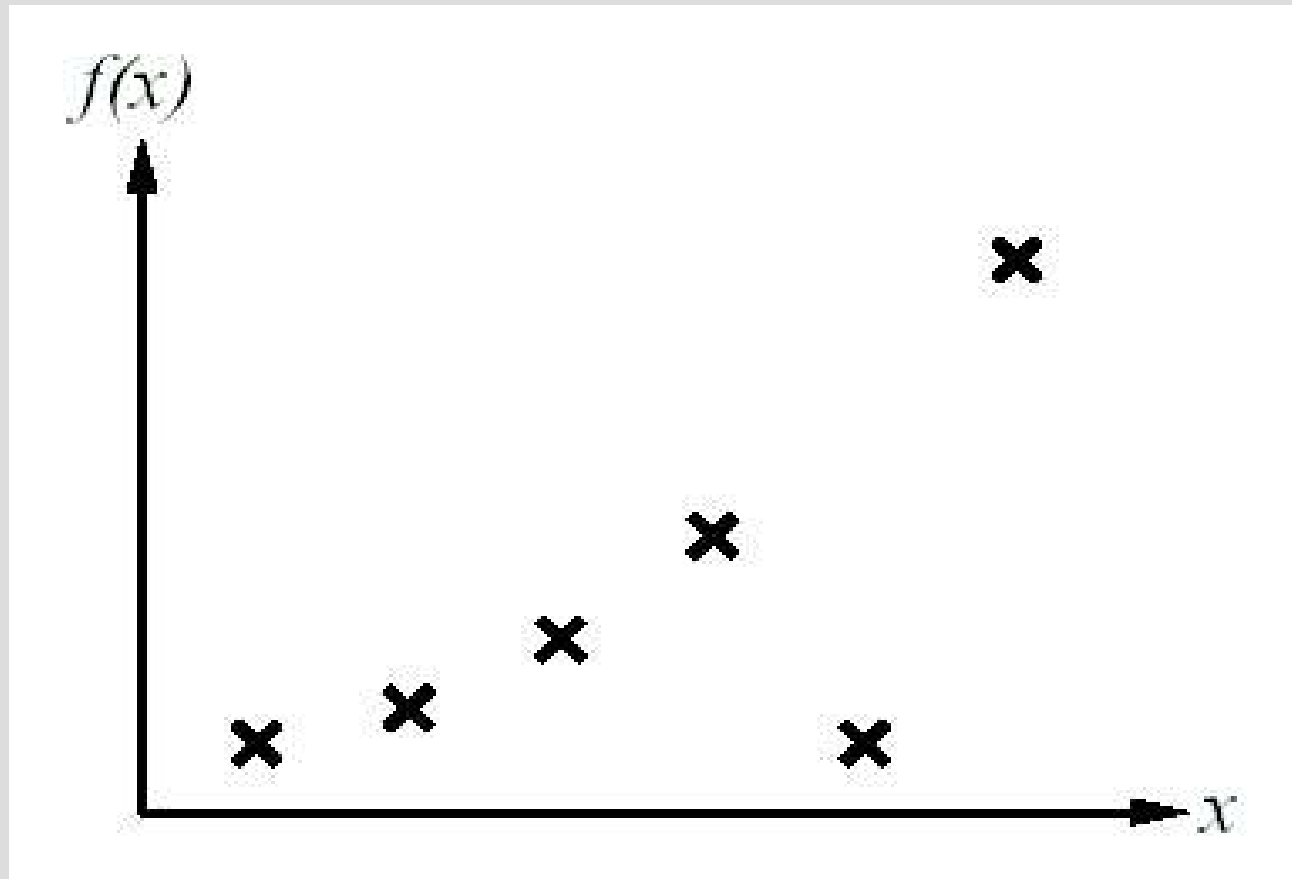


UNIT 29: Learning from observations

Inductive Learning Method



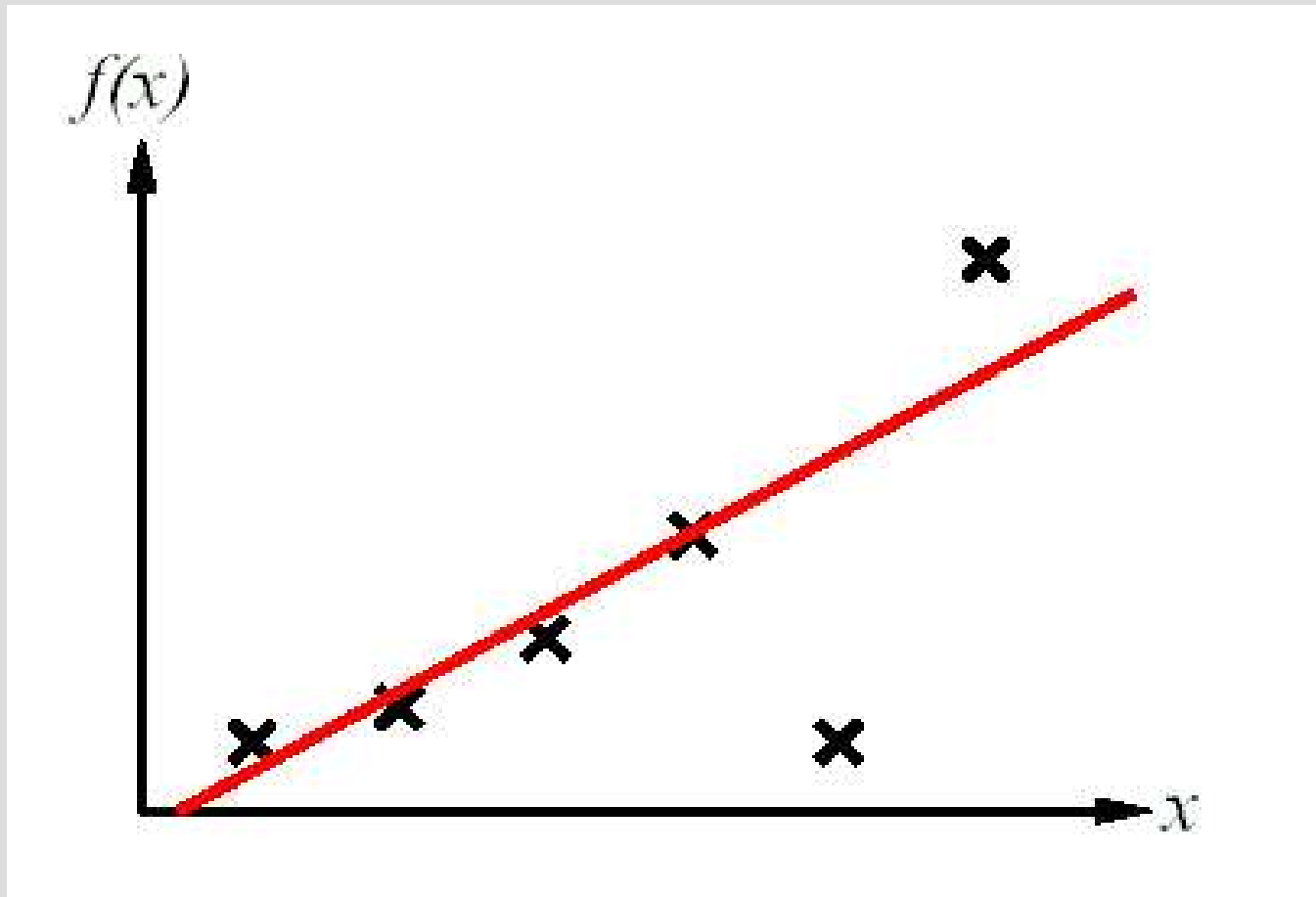
- Construct/adjust h to agree with f on training set e.g curve fitting



Inductive Learning Method



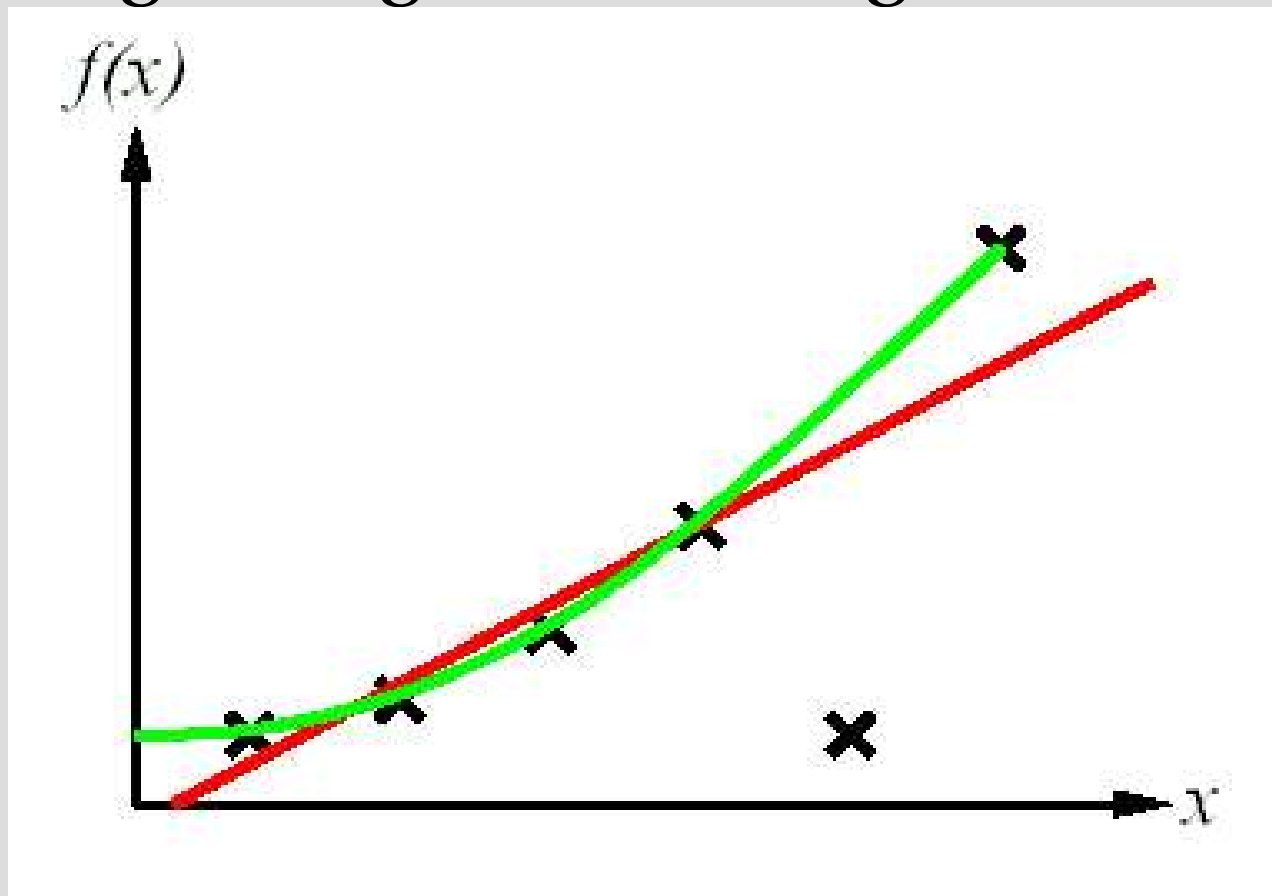
- Construct/adjust h to agree with f on training set e.g curve fitting



Inductive Learning Method



- Construct/adjust h to agree with f on training set e.g curve fitting

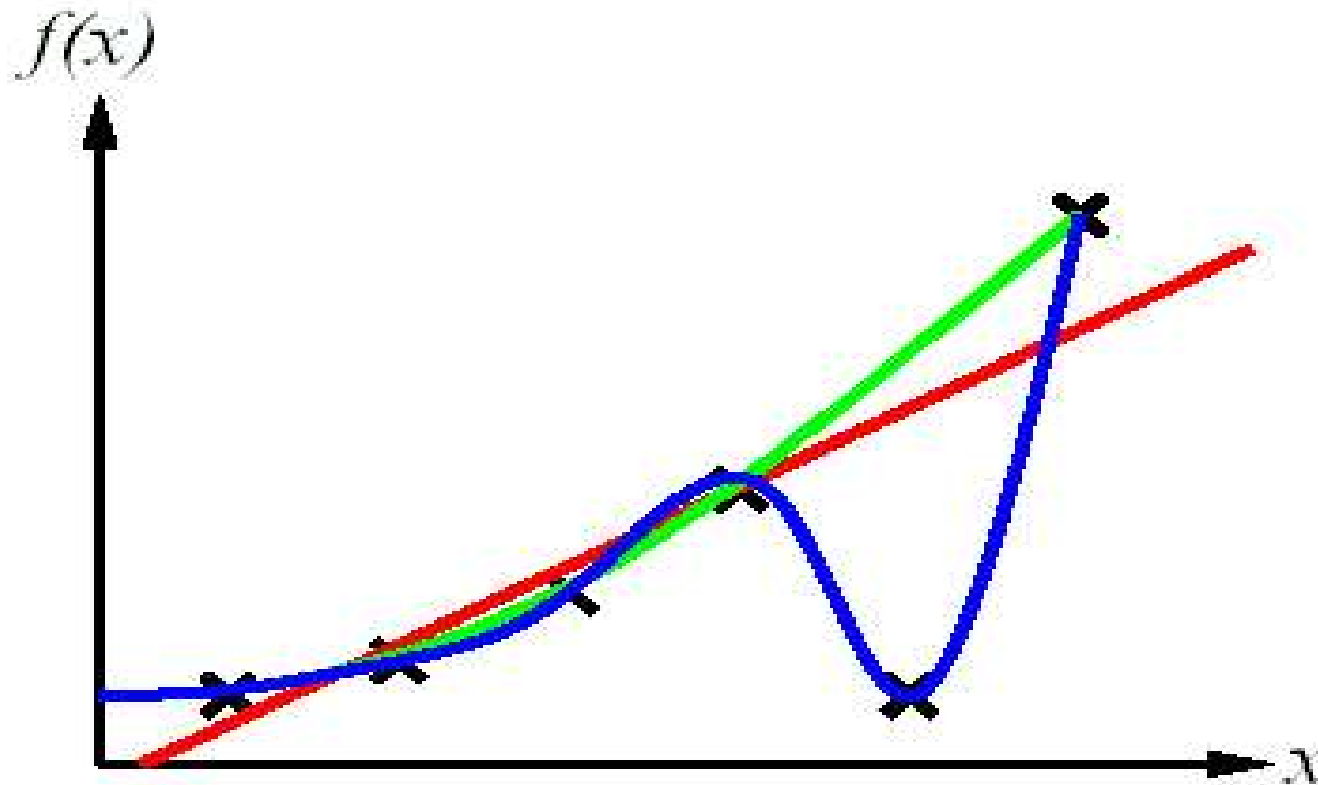


CONSISTENT
HYPOTHESIS

Inductive Learning Method



- Construct/adjust h to agree with f on training set e.g curve fitting

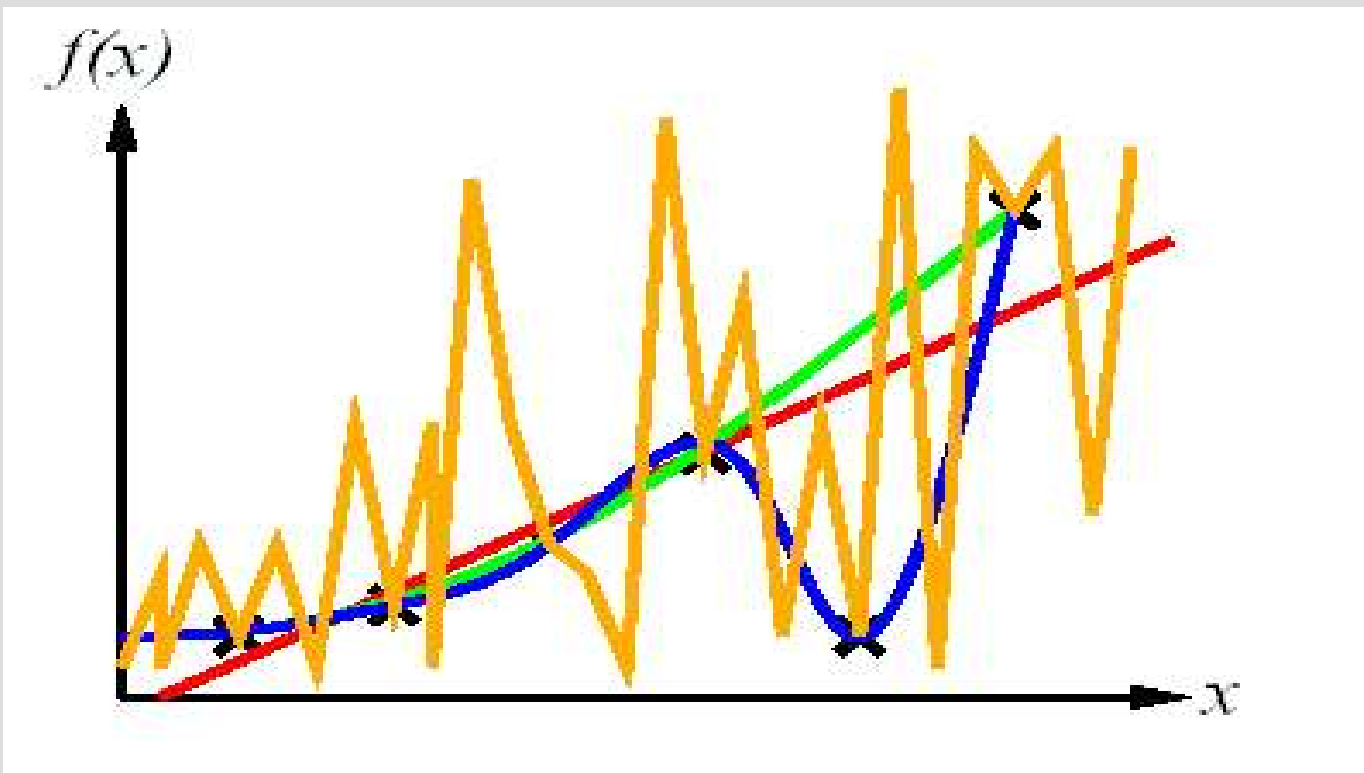


CONSISTENT
HYPOTHESIS

Inductive Learning Method



- Construct/adjust h to agree with f on training set e.g curve fitting



CONSISTENT
HYPOTHESIS

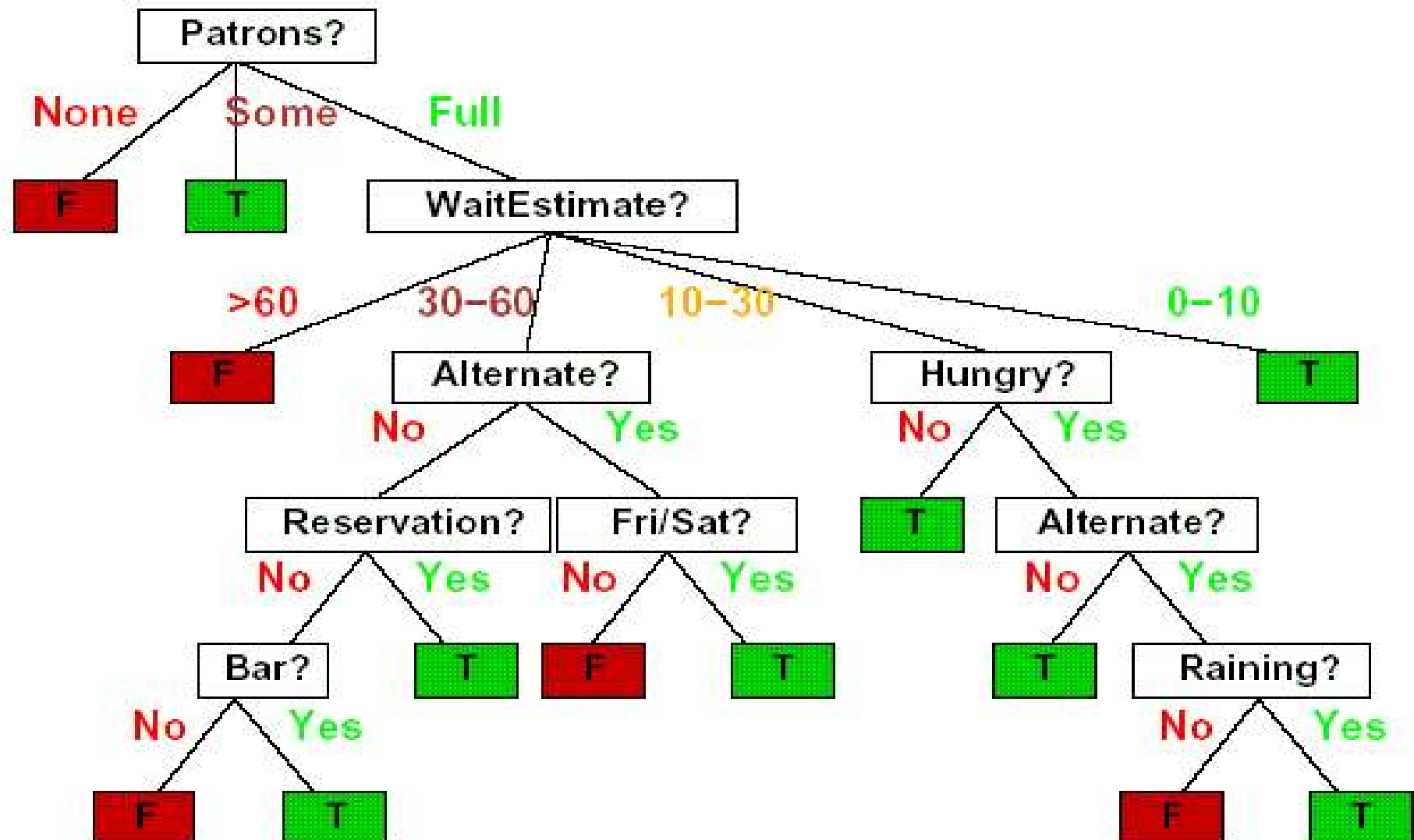
OCCAM'S
RAZOR
Maximize a
combination
of consistency
and simplicity

Attribute-value representations



Example	Attributes										Target
	<i>Alt</i>	<i>Bar</i>	<i>Fri</i>	<i>Hun</i>	<i>Pat</i>	<i>Price</i>	<i>Rain</i>	<i>Res</i>	<i>Type</i>	<i>Est</i>	<i>WillWait</i>
X_1	<i>T</i>	<i>F</i>	<i>F</i>	<i>T</i>	<i>Some</i>	<i>\$\$\$</i>	<i>F</i>	<i>T</i>	<i>French</i>	<i>0-10</i>	<i>T</i>
X_2	<i>T</i>	<i>F</i>	<i>F</i>	<i>T</i>	<i>Full</i>	<i>\$</i>	<i>F</i>	<i>F</i>	<i>Thai</i>	<i>30-60</i>	<i>F</i>
X_3	<i>F</i>	<i>T</i>	<i>F</i>	<i>F</i>	<i>Some</i>	<i>\$</i>	<i>F</i>	<i>F</i>	<i>Burger</i>	<i>0-10</i>	<i>T</i>
X_4	<i>T</i>	<i>F</i>	<i>T</i>	<i>T</i>	<i>Full</i>	<i>\$</i>	<i>F</i>	<i>F</i>	<i>Thai</i>	<i>10-30</i>	<i>T</i>
X_5	<i>T</i>	<i>F</i>	<i>T</i>	<i>F</i>	<i>Full</i>	<i>\$\$\$</i>	<i>F</i>	<i>T</i>	<i>French</i>	<i>>60</i>	<i>F</i>
X_6	<i>F</i>	<i>T</i>	<i>F</i>	<i>T</i>	<i>Some</i>	<i>\$\$</i>	<i>T</i>	<i>T</i>	<i>Italian</i>	<i>0-10</i>	<i>T</i>
X_7	<i>F</i>	<i>T</i>	<i>F</i>	<i>F</i>	<i>None</i>	<i>\$</i>	<i>T</i>	<i>F</i>	<i>Burger</i>	<i>0-10</i>	<i>F</i>
X_8	<i>F</i>	<i>F</i>	<i>F</i>	<i>T</i>	<i>Some</i>	<i>\$\$</i>	<i>T</i>	<i>T</i>	<i>Thai</i>	<i>0-10</i>	<i>T</i>
X_9	<i>F</i>	<i>T</i>	<i>T</i>	<i>F</i>	<i>Full</i>	<i>\$</i>	<i>T</i>	<i>F</i>	<i>Burger</i>	<i>>60</i>	<i>F</i>
X_{10}	<i>T</i>	<i>T</i>	<i>T</i>	<i>T</i>	<i>Full</i>	<i>\$\$\$</i>	<i>F</i>	<i>T</i>	<i>Italian</i>	<i>10-30</i>	<i>F</i>
X_{11}	<i>F</i>	<i>F</i>	<i>F</i>	<i>F</i>	<i>None</i>	<i>\$</i>	<i>F</i>	<i>F</i>	<i>Thai</i>	<i>0-10</i>	<i>F</i>
X_{12}	<i>T</i>	<i>T</i>	<i>T</i>	<i>T</i>	<i>Full</i>	<i>\$</i>	<i>F</i>	<i>F</i>	<i>Burger</i>	<i>30-60</i>	<i>T</i>

Decision Tree



Expressiveness



- Decision trees can express any function of the input attributes
 - Truth table row $\rightarrow$ path from root to leaf
- Trivially there is **consistent** decision tree for any training set w/ one path to leaf for each example but it probably won't generalize to new examples
- Prefer to find more **compact** decision trees

Hypothesis spaces



- How many distinct decision trees with n boolean attributes ?
 - = number of boolean functions
 - = number of distinct truth tables with 2^n rows = ?
- For 6 boolean attributes there are only 18,446,744, 073, 709,551,616 trees

Decision Tree Learning

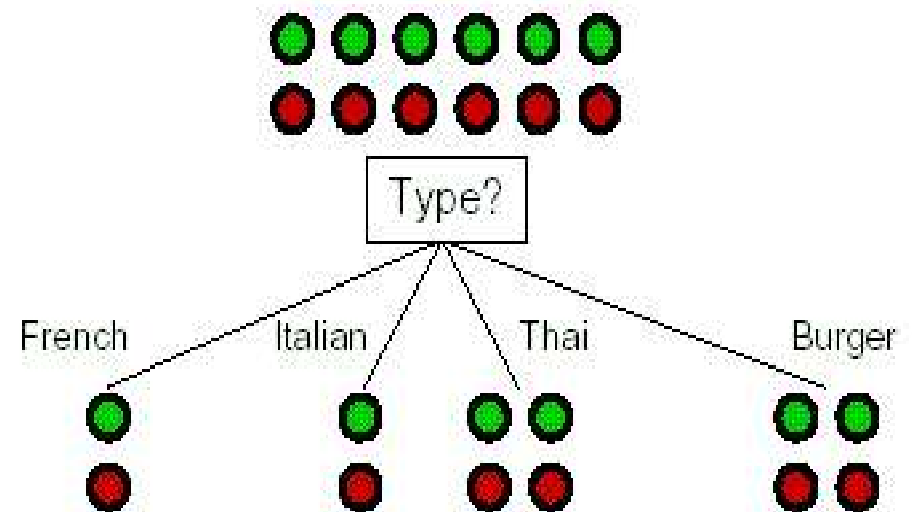
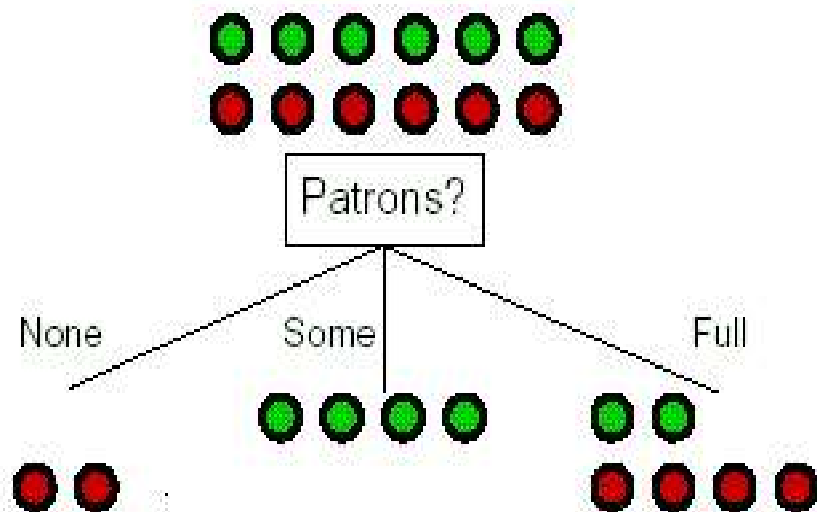


- Aim: Find a small tree consistent with training examples
- Idea: (recursively) choose the “most significant” attribute as root of the subtree

Choosing an attribute



- Idea: a good attribute splits examples into subsets that are “ideally” all + or -
- Patrons? is better choice – gives more “information” about classification



Information



- Information answers questions
- Scale = 1 bit = answer to Boolean question with prior $\langle 0.5, 0.5 \rangle$
- Information in an answer where prior is $\langle P_1, \dots, P_n \rangle$ is
- $H(\langle P_1, \dots, P_n \rangle) = \sum_{i=1}^n -P_i \log_2 P_i$
- Also called entropy of the prior

Historical Sidenote



- Claude E. Shannon – Information theory
- 1948 – Most efficient way to encode a message
- Other: juggling, unicycling
- Inventions:
 - Rocket powered pogo stick
 - Theseus mouse



Informaiton cont'd



- Suppose we have p positive and n negative examples at the root then:
 - $H(< p / p+n, n / p+n >)$ bits to classify new