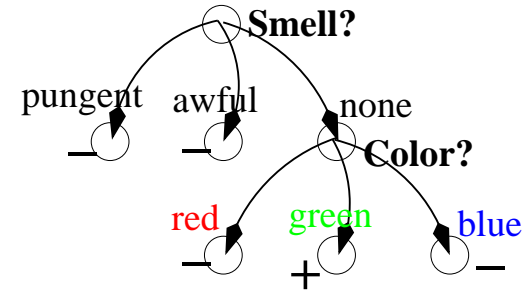


Lecture 5

Decision Trees

Decision Trees

some classifiers (k -nearest nbor) rely on distances between input vectors



what if want to classify food you like to eat?

classifiers for instances represented as lists of attributes (*nominal data*):
make decisions based on sets of attribute values

- internal nodes test *attributes*
- branches from node based on attribute *values*
- leaf nodes are outputs (*class*)

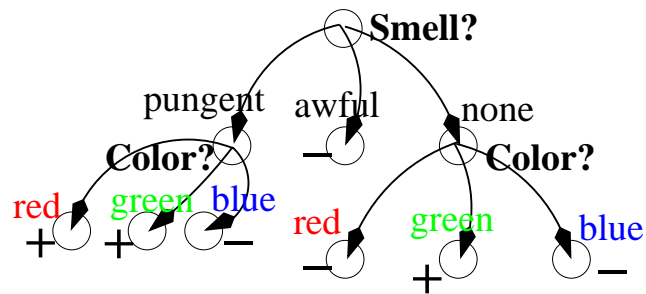
instances classified by following path from root to leaf of tree

Decision tree representations

represents a disjunction of conjunctions of constraints on attribute values

if features are continuous, may test value against threshold: $X_1 > 5$?

decision boundaries divide feature space into axis-parallel rectangles



interpretable system:

- understand decision for test example as series of decisions
(sweet-and-sour pork: $Smell = pungent \wedge Color = red \rightarrow +$)
- also understand category based on logical descriptions

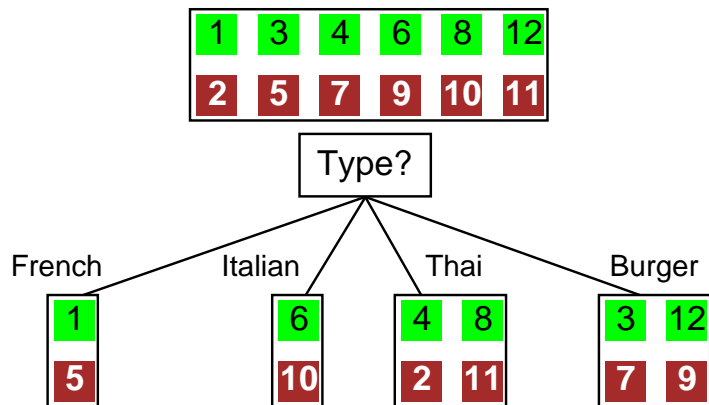
decision trees can represent any boolean function, but in the worst case requires exponentially many nodes

Decision Tree Algorithm

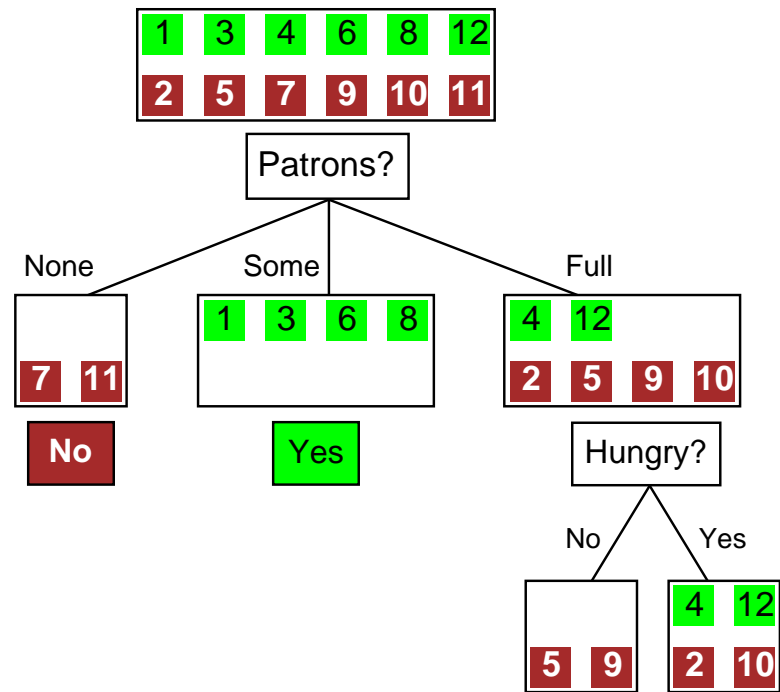
begin with simple, greedy, recursive approach, builds up tree node-by-node

1. pick an attribute to split at a non-terminal node
2. split examples into groups based on attribute value
3. for each group:
 - (a) if no examples – return majority from parent
 - (b) else if all examples same class – return class
 - (c) else if no attribute (non-deterministic domain or errors)
return majority
 - (d) else loop to step 1

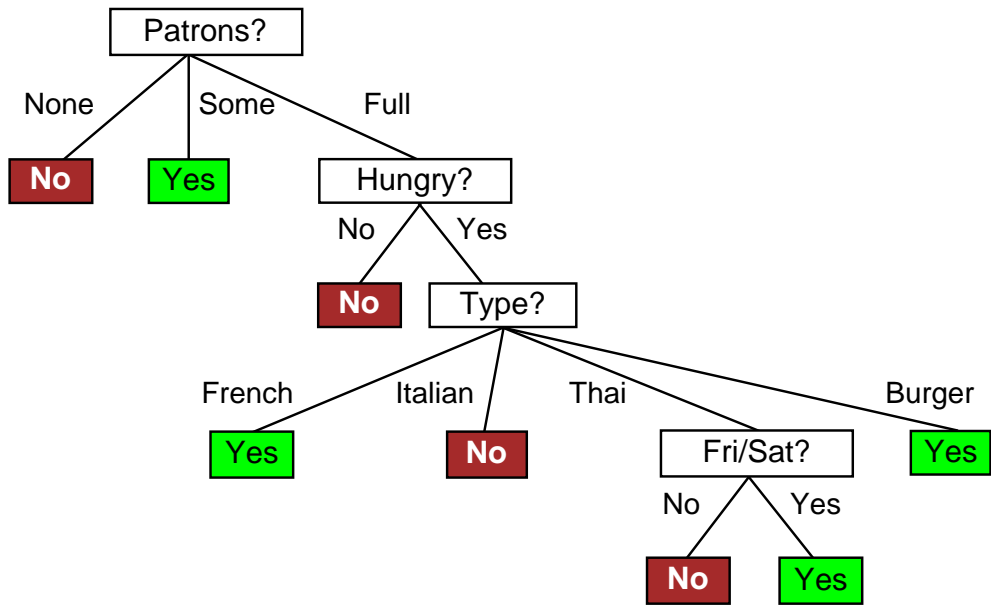
Ex.	Attributes										Target
	<i>Alt</i>	<i>Ba^r</i>	<i>Fri</i>	<i>Huⁿ</i>	<i>Pat</i>	<i>Price</i>	<i>Rain</i>	<i>Res</i>	<i>Type</i>	<i>Est</i>	<i>WillWait</i>
X_1	<i>T</i>	<i>F</i>	<i>F</i>	<i>T</i>	<i>Some</i>	<i>\$\$\$</i>	<i>F</i>	<i>T</i>	<i>French</i>	<i>0–10</i>	<i>T</i>
X_2	<i>T</i>	<i>F</i>	<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_{1^1}	<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_{1^2}	<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>

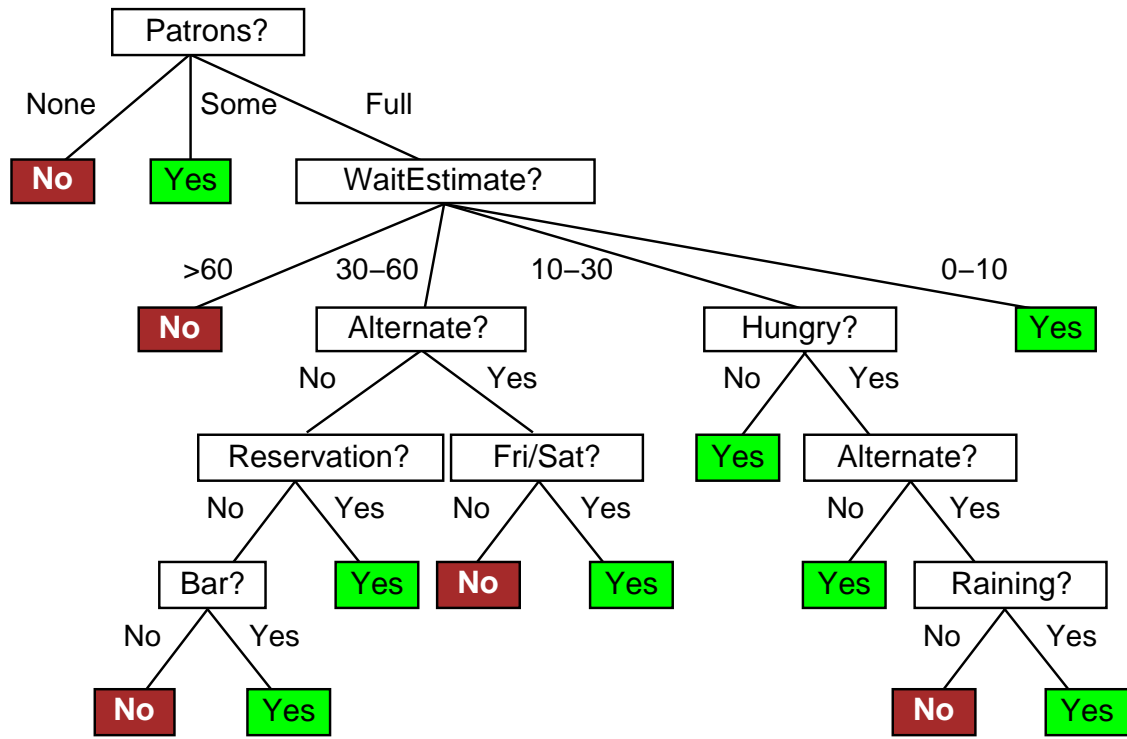


(a)



(b)





What makes a good tree?

Not too small – need to include enough attributes to handle possibly subtle distinctions in data

Not too big

- computational efficiency (avoid redundant, spurious attributes)
- avoid over-fitting training examples (noisy, scarce data)

Occam's Razor: find simplest hypothesis (tree) that is consistent with all observations

inductive bias – small trees, with informative nodes near root