Decision Tree Learning

Machine Learning (503025)

Outline

1. Problem Specification
2. Introduction Decision Tree
3. Decision Tree Learning
4. Performance Measures
5. Summary

Learning Decision Trees

- ***Problem:*** decide whether to wait for a table at a restaurant, based on the following attributes:
  1. *Alternate*: is there an alternative restaurant nearby?
  2. *Bar*: is there a comfortable bar area to wait in?
  3. *Fri/Sat*: is today Friday or Saturday?
  4. *Hungry*: are we hungry?
  5. *Patrons*: number of people in the restaurant (None, Some, Full)
  6. *Price*: price range ($, $$, $$$)
  7. *Raining*: is it raining outside?
  8. *Reservation*: have we made a reservation?
  9. *Type*: kind of restaurant (French, Italian, Thai, Burger)
  10. *WaitEstimate*: estimated waiting time (0-10, 10-30, 30-60, >60)

Attribute-based Representations

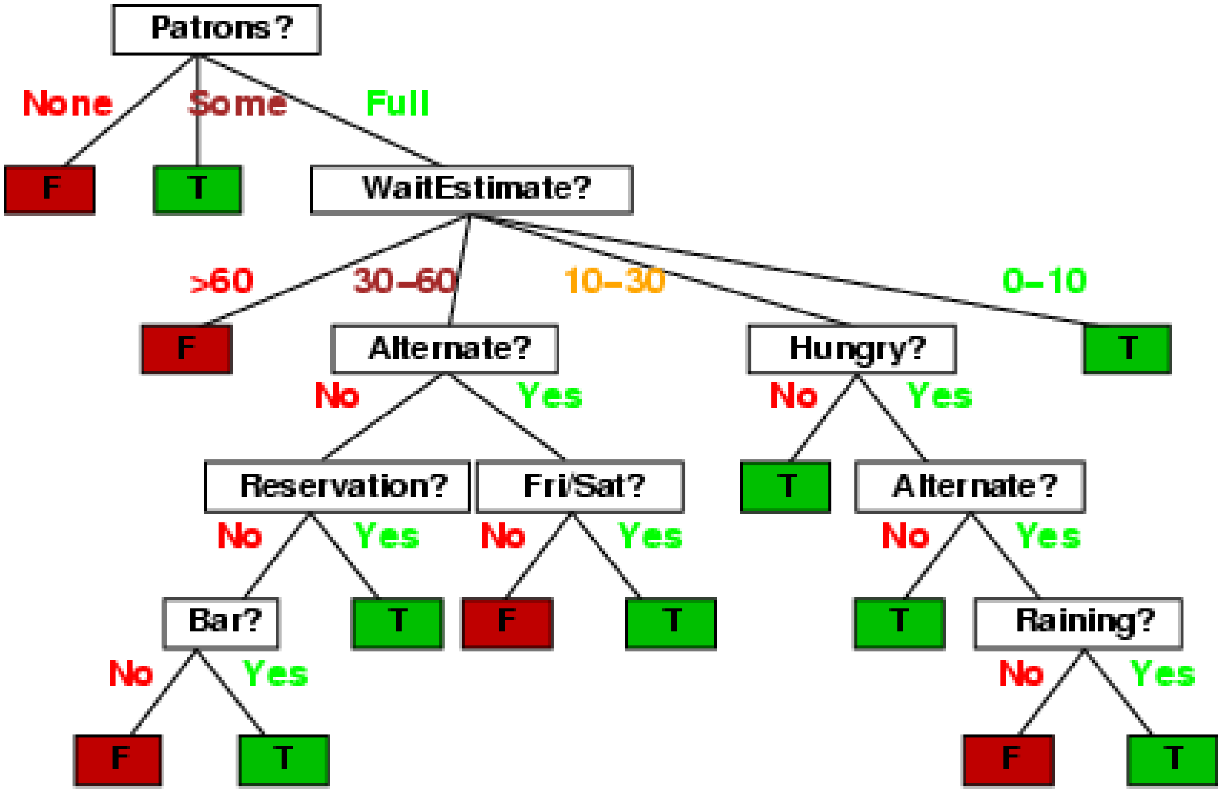
- Examples described by attribute values (Boolean, discrete, continuous)
- E.g., situations where I will/won't wait for a table:

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

- Classification of examples is **positive (T)** or **negative (F)**.

Decision Trees

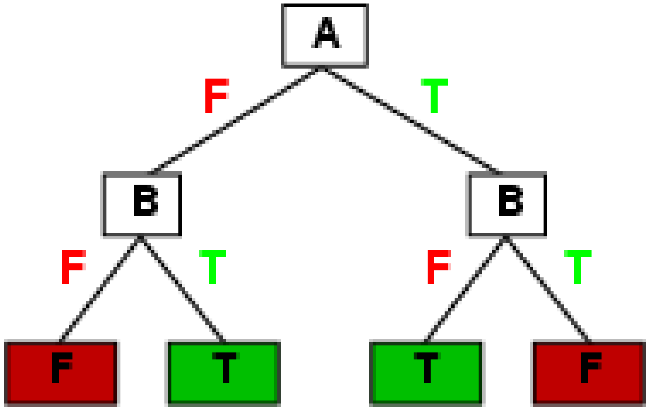
- One possible representation for hypotheses
- E.g., here is the “true” tree for deciding whether to wait:



Expressiveness

- Decision trees can express any function *f* of the input attributes.
- E.g., for Boolean functions, *truth table row → path to leaf*:

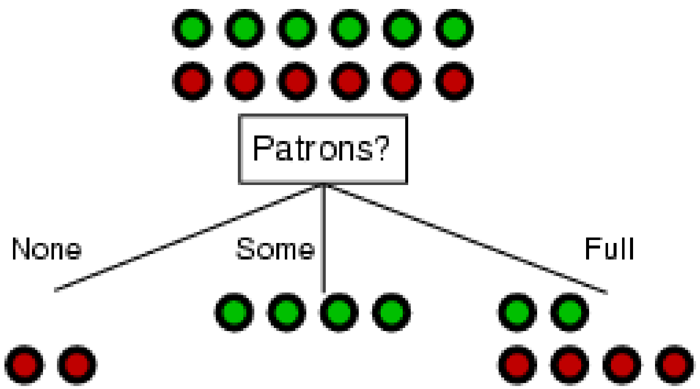
A	B	A xor B
F	F	F
F	T	T
T	F	T
T	T	F

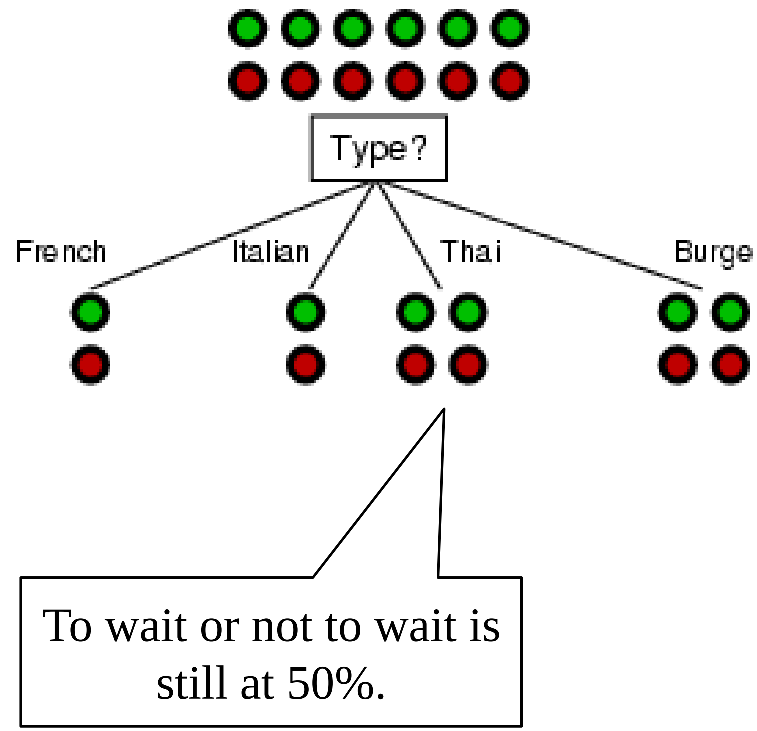


- Trivially, there is a consistent decision tree for any training set with one path to leaf for each example but it probably won't generalize to new examples. It's ***over-fitting*** case.
- Prefer to find more **compact** decision trees.

Choosing An Attribute

- Idea: a good attribute splits the examples into subsets that are (ideally) “all positive” or “all negative”.





- *Patrons?* is a better choice.

Information Gain

- **Entropy:**

$$Entropy(S) \equiv \sum_{i=1}^{c} -p_i \log_2 p_i$$

- **Information gain:**

$$Gain(S, A) \equiv Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

Play Tennis Example

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

Play Tennis Example (cont.)

$$
\begin{aligned}
Entropy([9+,5-]) &= -(9/14)\log_2(9/14) - (5/14)\log_2(5/14) \\
&= 0.940
\end{aligned}
$$

$$
\begin{aligned}
Values(Wind) &= Weak, Strong \\
S &= [9+,5-] \\
S_{Weak} &\leftarrow [6+,2-] \\
S_{Strong} &\leftarrow [3+,3-] \\
Gain(S, Wind) &= Entropy(S) - \sum_{v\in\{Weak,Strong\}} \frac{|S_v|}{|S|} Entropy(S_v) \\
&= Entropy(S) - (8/14)Entropy(S_{Weak}) \\
&\quad - (6/14)Entropy(S_{Strong}) \\
&= 0.940 - (8/14)0.811 - (6/14)1.00 \\
&= 0.048
\end{aligned}
$$

Play Tennis Example (cont.)

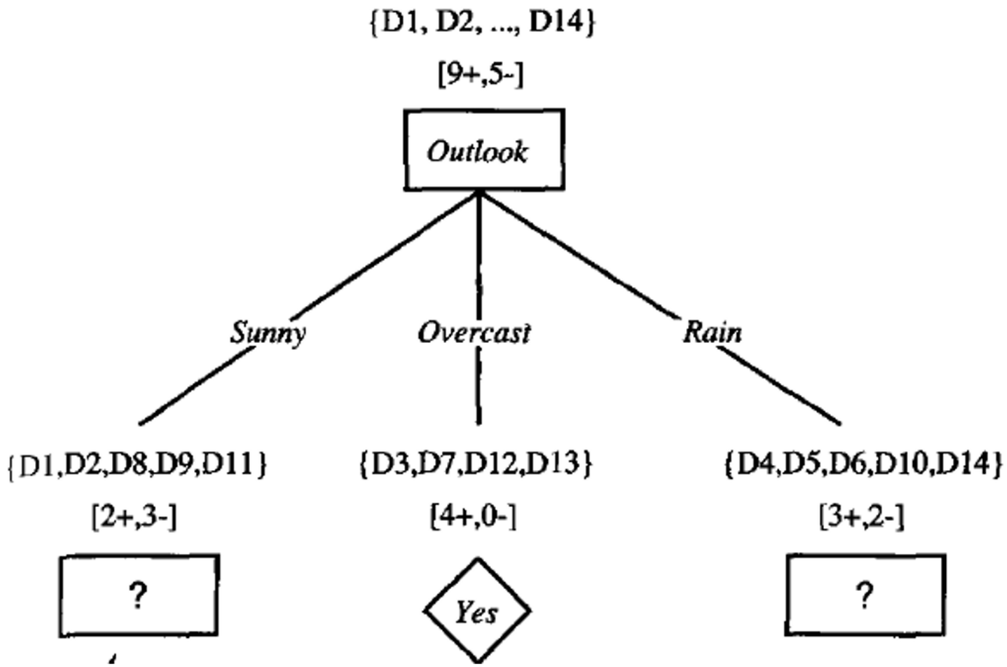
$$Gain(S, Outlook) = 0.246$$

$$Gain(S, Humidity) = 0.151$$

$$Gain(S, Wind) = 0.048$$

$$Gain(S, Temperature) = 0.029$$

Play Tennis Example (cont.)



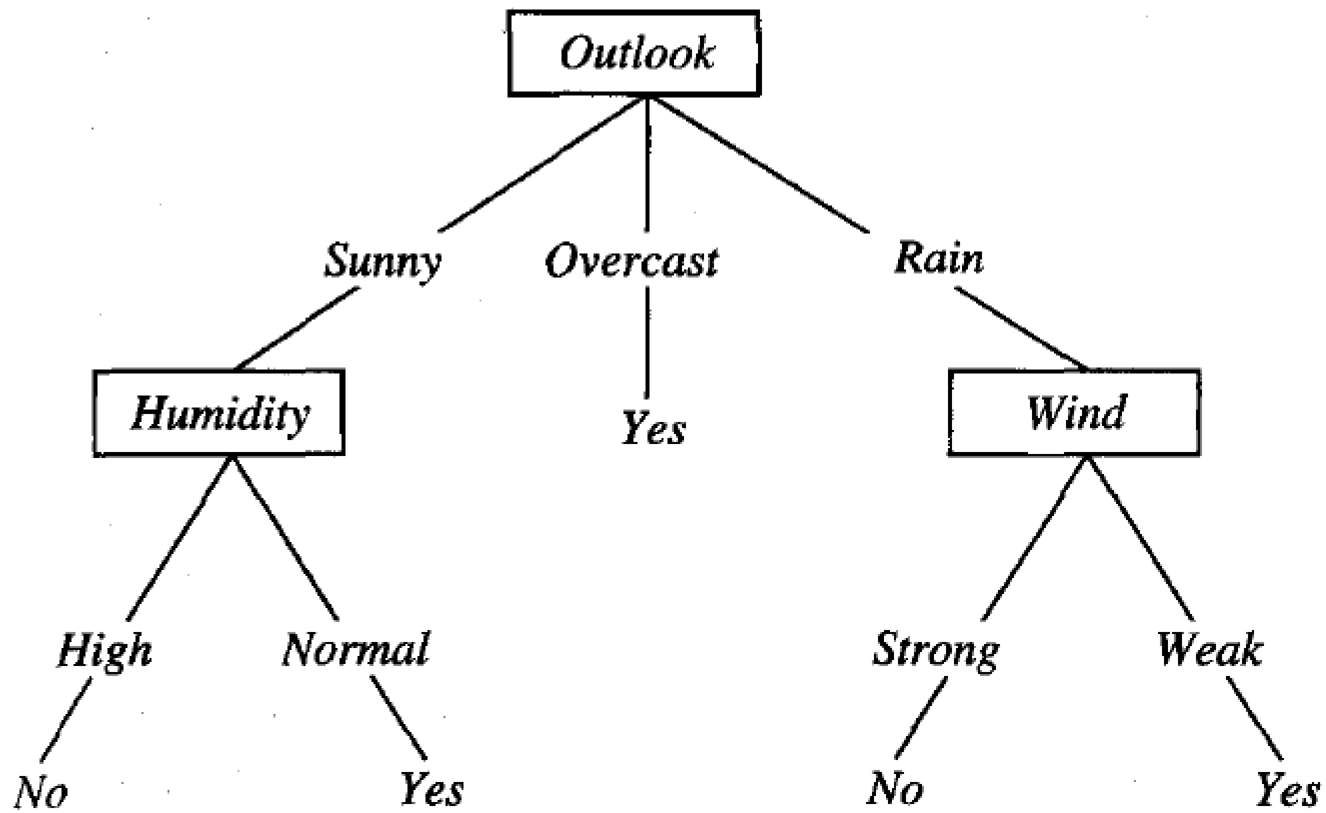
$S_{sunny} = \{D1,D2,D8,D9,D11\}$

$Gain\,(S_{sunny}, Humidity) = .970 - (3/5)\,0.0 - (2/5)\,0.0 = .970$

$Gain\,(S_{sunny}, Temperature) = .970 - (2/5)\,0.0 - (2/5)\,1.0 - (1/5)\,0.0 = .570$

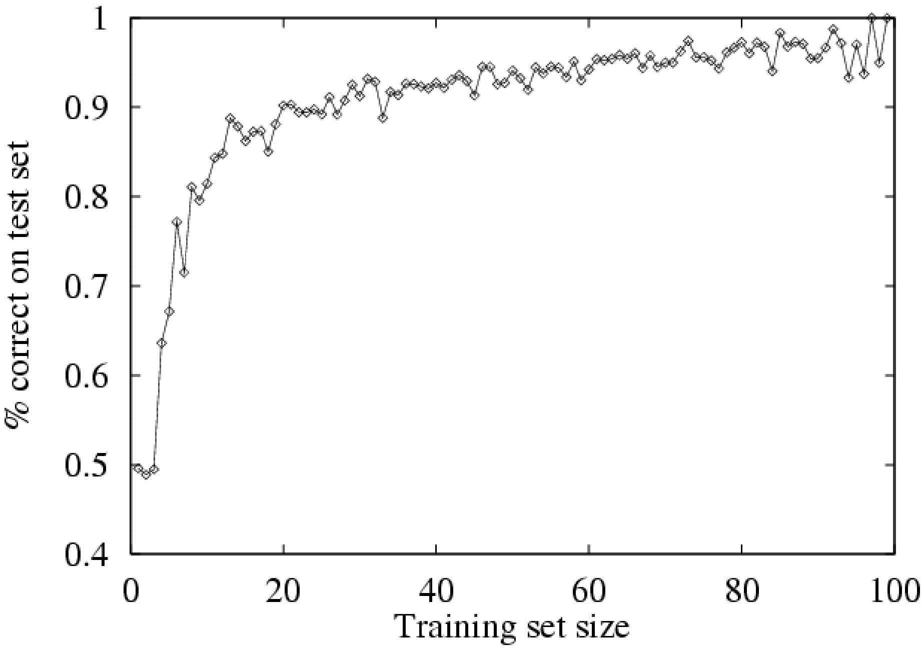
$Gain\,(S_{sunny}, Wind) = .970 - (2/5)\,1.0 - (3/5)\,.918 = .019$

Play Tennis Example (cont.)



Performance Measurement

- How do we know that $h \approx f$?
  - Use theorems of computational/statistical learning theory.
  - Try h on a new test set of examples (use same distribution over example space as training set).
- Learning curve = % correct on test set as a function of training set size.



Summary

- Learning needed for unknown environments, lazy designers.
- For supervised learning, the aim is to find a simple hypothesis approximately consistent with training examples.
- Decision tree learning using information gain.
- Learning performance = prediction accuracy measured on test set.