Foundations of Artificial Intelligence

CS472/3 — Fall 1999
Lecture #23

Bart Selman

Learning Decision Trees

Decision tree takes as input a set of properties and outputs yes/no "decisions".

Example:

goal predicate: *WillWait*

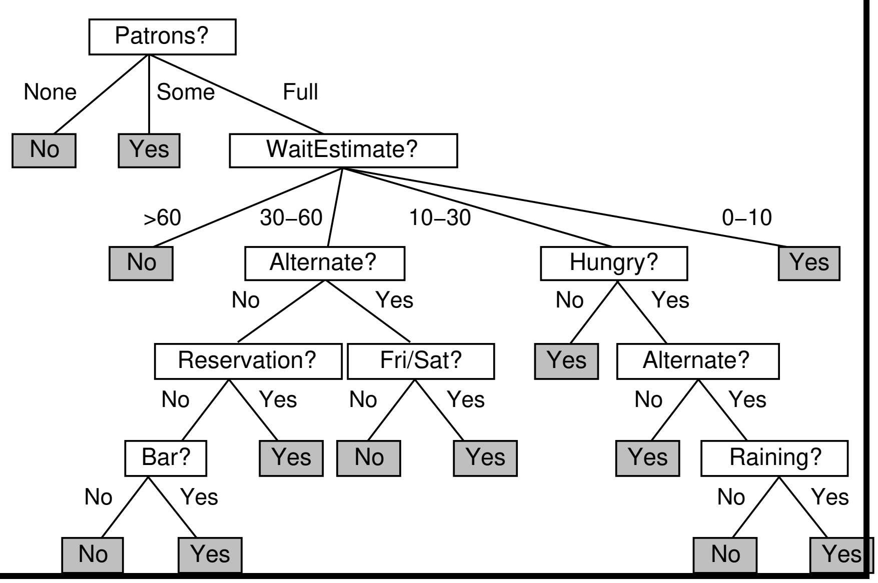
Example attributes:

1. Alternate
2. Bar
3. Fri/Sat
4. Hungry
5. Patrons
6. Price

etc.

$$\forall Patrons(r, Full) \wedge WaitEstimate(r, 10-30) \wedge Hungry(r, N)$$
$$\Rightarrow WillWait(r)$$

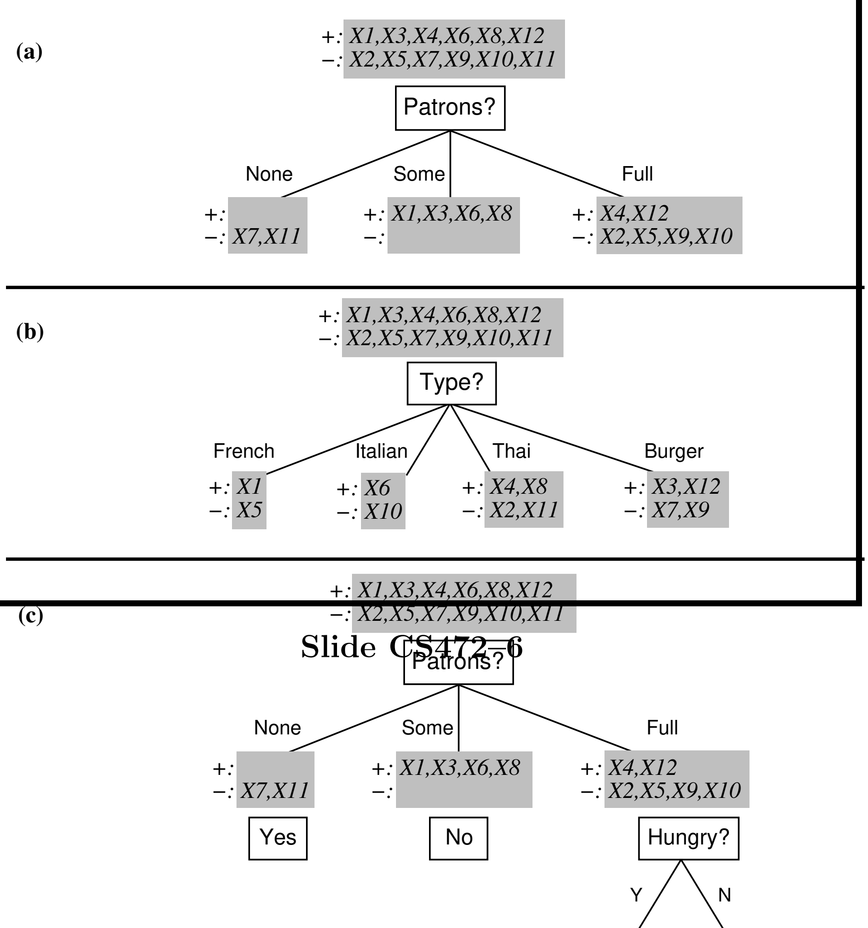
Slide CS472–3



Slide CS472–4

Example	Attributes										Goal
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	*Yes*	*No*	*No*	*Yes*	*Some*	*$$$*	*No*	*Yes*	*French*	*0–10*	*Yes*
X_2	*Yes*	*No*	*No*	*Yes*	*Full*	*$*	*No*	*No*	*Thai*	*30–60*	*No*
X_3	*No*	*Yes*	*No*	*No*	*Some*	*$*	*No*	*No*	*Burger*	*0–10*	*Yes*
X_4	*Yes*	*No*	*Yes*	*Yes*	*Full*	*$*	*No*	*No*	*Thai*	*10–30*	*Yes*
X_5	*Yes*	*No*	*Yes*	*No*	*Full*	*$$$*	*No*	*Yes*	*French*	*>60*	*No*
X_6	*No*	*Yes*	*No*	*Yes*	*Some*	*$$*	*Yes*	*Yes*	*Italian*	*0–10*	*Yes*
X_7	*No*	*Yes*	*No*	*No*	*None*	*$*	*Yes*	*No*	*Burger*	*0–10*	*No*
X_8	*No*	*No*	*No*	*Yes*	*Some*	*$$*	*Yes*	*Yes*	*Thai*	*0–10*	*Yes*
X_9	*No*	*Yes*	*Yes*	*No*	*Full*	*$*	*Yes*	*No*	*Burger*	*>60*	*No*
X_{10}	*Yes*	*Yes*	*Yes*	*Yes*	*Full*	*$$$*	*No*	*Yes*	*Italian*	*10–30*	*No*
X_{11}	*No*	*No*	*No*	*No*	*None*	*$*	*No*	*No*	*Thai*	*0–10*	*No*
X_{12}	*Yes*	*Yes*	*Yes*	*Yes*	*Full*	*$*	*No*	*No*	*Burger*	*30–60*	*Yes*

Slide CS472–5



Slide CS472–6

Best Property

- need to select property / feature / attribute
- goal find **short tree** (Occam's razor)

    select **most informative** feature
    one that best splits (classifies) the examples

use measure from **information theory**
    Claude Shannon (1949)

Slide CS472–7

Entropy / Information Content

- measures the "unpredictability" of an information source
    (loosely connected to physical chaos / randomness)

- measures number of **bits** needed to obtain full info

$$I(P(v_1), \ldots, P(v_n)) = \sum_{i=1}^{n} -P(v_i) \ log_2(P(v_i))$$

— $v_1, \ldots, v_n$ possible answers
— $P(v_i)$ probabibilty of answer v_i

Slide CS472–8

Some Examples

Source: fair coin

$$I(0.5, 0.5) = -0.5 \ \ log_2(0.5) - 0.5 \ \ log_2(0.5) = 1 \ \ bit$$

i.e., need 1 bit to convey the outcome
of the coin flip.

Source: biased coin

$$I(1/100, 99/100) = 0.08 \ \ bits$$

as the probability of heads goes to 1, the information of
the actual outcome goes to 0.

$I(0, 1) = I(1, 0) = 0 \ \ bits$

i.e., no uncertainty left in source. $(0.log_2(0) = 0)$

figure entropy — Boolean classification

Applied to a Collection of Examples

We don't have **exact** probabilities but our **training data** provides an estimate of the probabilities:

$$I(\frac{p}{p+n}, \frac{n}{p+n}) = -\frac{p}{p+n}\ \ log_2(\frac{p}{p+n}) - \frac{n}{p+n}\ \ log_2(\frac{n}{p+n})$$

Training set with p positive and n negative examples.

Example

our collection of 12 restaurant examples

$p = n = 6$, give $I(0.5, 0.5) = 1$ *bit*

so we need 1 *bit* of info to classify a randomly picked example.

Slide CS472–13

Intuition / Extremes

Entropy in collection is 0 if all examples in same class.

Entropy is 1 if equal number of positive and negative examples.

Intuition:

If you pick **random** example, how many **bits** do
you need to specify what class the example belongs too?

Slide CS472–14

Picking Attribute

Intuition: We want to pick the attribute that reduces the **entropy** (uncertainty) the **most**.

We therefore measure the information **gain** after testing on attribute A:

$$Gain(A) = I(\frac{p}{p+n}, \frac{n}{p+n}) - Remainder(A)$$

$Remainder(A)$ gives us the remaining uncertainty after getting info on attribute A.

Slide CS472–15

$Remainder(A)$

Gives the amount of info we still need after testing on A.

Assume A divides the training set E into $E_1, \ldots E_v$, where A has v distinct values.

Each subset E_i has p_i positive, and n_i negative examples. We again can compute entropy / information content of the remaining collection E_i

For total information content, need to weigh the contributions:

$$Remainder(A) = \sum_{i=1}^{v} \frac{p_i+n_i}{p+n} \; I(\frac{p_i}{p_i+n_i}, \frac{n_i}{p_i+n_i})$$

Slide CS472–16

Example Gains

Attributes *Patrons* and *Type* at top of tree:

$$Gain(Patrons) = 1 - [\tfrac{2}{12}I(0,1) + \tfrac{4}{12}I(1,0) + \tfrac{6}{12}I(\tfrac{2}{6},\tfrac{4}{6})]$$
$$\approx 0.541 \ \ bits$$

$$Gain(Type) = 1 - [\tfrac{2}{12}I(\tfrac{1}{2},\tfrac{1}{2}) + \tfrac{2}{12}I(\tfrac{1}{2},\tfrac{1}{2}) + \tfrac{4}{12}I(\tfrac{2}{4},\tfrac{2}{4}) + \tfrac{4}{12}I(\tfrac{2}{4},\tfrac{2}{4})$$
$$= 0.0 \ \ bits$$

Slide CS472–17

Patrons has the highest info gain of all attributes at the root. Will be picked first.

Slide CS472–18