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.

\( \forall \text{Patrons}(r, \text{Full}) \land \text{WaitEstimate}(r, 10 - 30) \land \text{Hungry}(r, N) \implies \text{WillWait}(r) \)
### Slide CS472–5

#### Example Attributes

<table>
<thead>
<tr>
<th>Example</th>
<th>Alt</th>
<th>Bar</th>
<th>Fri</th>
<th>Hun</th>
<th>Put</th>
<th>Price</th>
<th>Rain</th>
<th>Res</th>
<th>Type</th>
<th>Est</th>
<th>WillWait</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X_1$</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Some</td>
<td>$$$</td>
<td>No</td>
<td>Yes</td>
<td>French</td>
<td>0–10</td>
<td>Yes</td>
</tr>
<tr>
<td>$X_2$</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Full</td>
<td>$$</td>
<td>No</td>
<td>No</td>
<td>Thai</td>
<td>30–60</td>
<td>No</td>
</tr>
<tr>
<td>$X_3$</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Some</td>
<td>$$</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>0–10</td>
<td>Yes</td>
</tr>
<tr>
<td>$X_4$</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Full</td>
<td>$$</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>10–30</td>
<td>Yes</td>
</tr>
<tr>
<td>$X_5$</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Full</td>
<td>$$$</td>
<td>No</td>
<td>Yes</td>
<td>French</td>
<td>&gt;60</td>
<td>No</td>
</tr>
<tr>
<td>$X_6$</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Some</td>
<td>$$$</td>
<td>Yes</td>
<td>Yes</td>
<td>Italian</td>
<td>0–10</td>
<td>Yes</td>
</tr>
<tr>
<td>$X_7$</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>None</td>
<td>$$</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>0–10</td>
<td>No</td>
</tr>
<tr>
<td>$X_8$</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Some</td>
<td>$$$</td>
<td>Yes</td>
<td>Yes</td>
<td>Thai</td>
<td>0–10</td>
<td>Yes</td>
</tr>
<tr>
<td>$X_9$</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Full</td>
<td>$$$</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Burger&gt;60</td>
<td>No</td>
</tr>
<tr>
<td>$X_{10}$</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Full</td>
<td>$$$</td>
<td>No</td>
<td>Yes</td>
<td>Italian</td>
<td>10–30</td>
<td>No</td>
</tr>
<tr>
<td>$X_{11}$</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>None</td>
<td>$$</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Thai</td>
<td>0–10</td>
</tr>
<tr>
<td>$X_{12}$</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Full</td>
<td>$$</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Burger</td>
<td>30–60</td>
</tr>
</tbody>
</table>

### Slide CS472–6

#### Diagram (a)

```
(a) +: X1,X3,X4,X6,X8,X12
-: X2,X5,X7,X9,X10,X11
```

**Patrons?**

- None
- Some
- Full

```
+: X7,X11
-: X1,X3,X6,X8
```

#### Diagram (b)

```
(b) +: X1,X3,X4,X6,X8,X12
-: X2,X5,X7,X9,X10,X11
```

**Type?**

- French
- Italian
- Thai
- Burger

```
+: X1
+: X5
+: X3,X12
-: X6
-: X10
-: X4,X8
-: X2,X11
-: X7,X9
```

#### Diagram (c)

```
(c) +: X1,X3,X4,X6,X8,X12
-: X2,X5,X7,X9,X10,X11
```

**Patrons?**

- None
- Some
- Full

```
+: X7,X11
-: X1,X3,X6,X8
```

**Hungry?**

- Yes
- No

```
Y
N
```
**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)

---

**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
  - \(P(v_i)\) probability of answer \(v_i\)
Some Examples

Source: fair coin

\[ I(0.5, 0.5) = -0.5 \log_2(0.5) - 0.5 \log_2(0.5) = 1 \text{ bit} \]

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

Slide CS472–9

Source: biased coin

\[ I(1/100, 99/100) = 0.08 \text{ bits} \]

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

\[ I(0, 1) = I(1, 0) = 0 \text{ bits} \]

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

Slide CS472–10
Applied to a Collection of Examples

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

\[
I\left(\frac{p}{p+n}, \frac{n}{p+n}\right) = -\frac{p}{p+n} \log_2\left(\frac{p}{p+n}\right) - \frac{n}{p+n} \log_2\left(\frac{n}{p+n}\right)
\]

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.
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?

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 $A$:

\[
Gain(A) = I\left(\frac{p}{p+n}, \frac{n}{p+n}\right) - \text{Remainder}(A)
\]

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

---

Slide CS472–13

---

Slide CS472–14
**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:

$$\text{Remainder}(A) = \sum_{i=1}^{v} \frac{p_i + n_i}{p_i + n_i - 1} I\left( \frac{p_i}{p_i + n_i}, \frac{n_i}{p_i + n_i} \right)$$

---

**Example Gains**

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

$\text{Gain}(\text{Patrons}) = 1 - \left[ \frac{2}{12} I(0, 1) + \frac{1}{12} I(1, 0) + \frac{6}{12} I\left( \frac{2}{6}, \frac{4}{6} \right) \right]$

$\approx 0.541$ bits

$\text{Gain}(\text{Type}) = 1 - \left[ \frac{2}{12} I\left( \frac{1}{2}, \frac{1}{2} \right) + \frac{2}{12} I\left( \frac{1}{2}, \frac{1}{2} \right) + \frac{1}{12} I\left( \frac{2}{3}, \frac{1}{3} \right) + \frac{4}{12} I\left( \frac{2}{4}, \frac{2}{4} \right) \right]$

$= 0.0$ bits
*Patrons* has the highest info gain of all attributes at the root. Will be picked first.