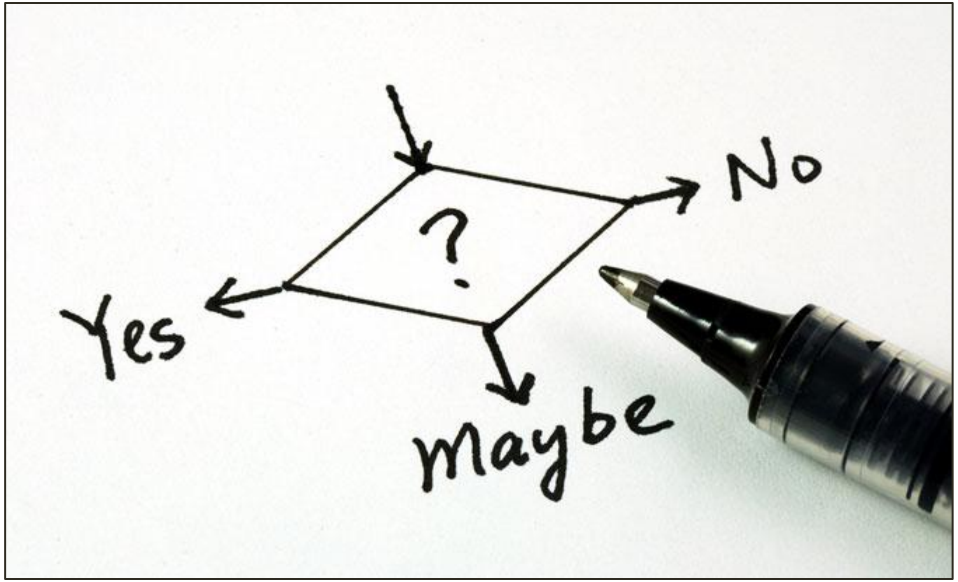
CSCI-620

Decision Trees

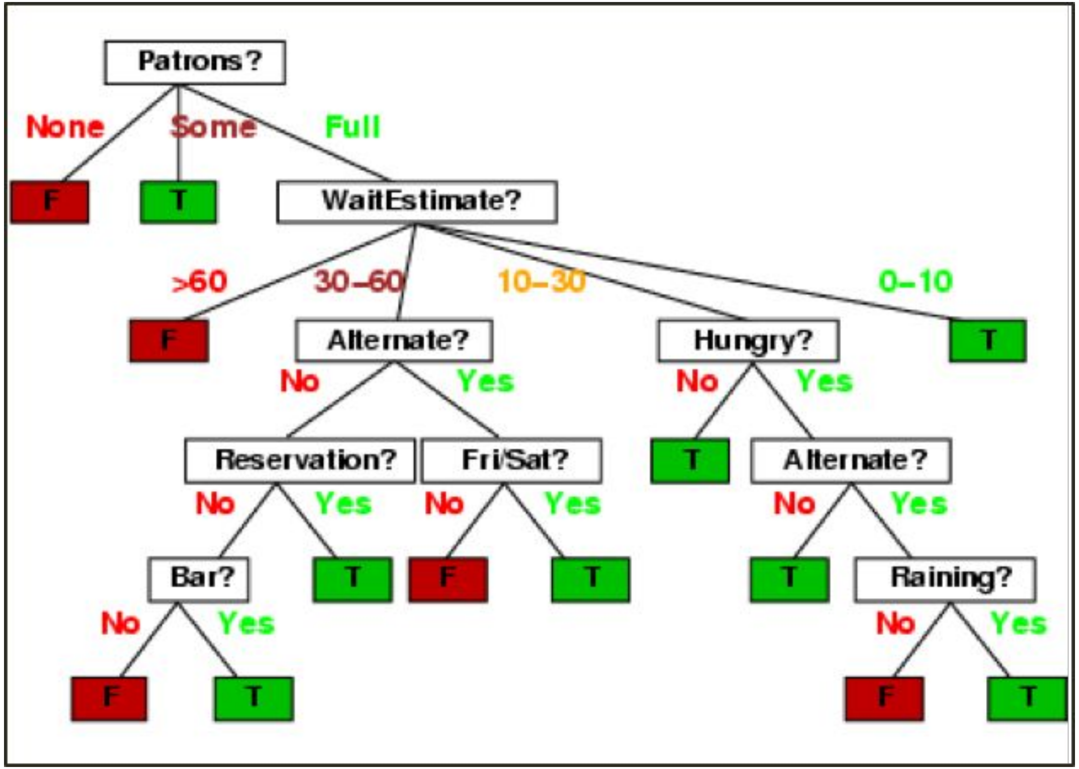


A tree-like structure that aids users make (complex) decisions with respect to a (large) number of factors and the consequences that may be derived.

Definition

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

Decision Trees



Decision Trees

Tree Elements

- For a set of records R with a set of attributes A, predict t wrt $A \setminus \{t\}$

- Branch nodes: fix an attribute $x \in A \setminus \{t\}$ and analyze its values
- Edges: values of x
- Leaf nodes: value of t after a path from the root

Construct a root node that *includes all the examples*, then for each node:

1. if there are both positive and negative examples, choose the best attribute to split them.
2. if all the examples are pos (neg) answer yes (no).
3. if there are no examples for a case (no observed examples) then choose a default based on the majority classification at the parent.
4. if there are no attributes left but we have both pos and neg examples, this means that the selected features are not sufficient for classification or that there is error in the examples. (can use majority vote.)

Decision Tree Building

Decision Tree Algorithms

- The main factor defining an algorithm is how we choose which attribute (and value) we should split on next
- We want to minimize the complexity (depth) of the tree, but still be accurate
- In practice, we will usually assume the data is noisy and ignore some

Having X = {x_1, x_2, ..., x_n}, the entropy of X is defined as follows:

$$\mathrm{H}(X) = -\sum_{i=1}^{n} \mathrm{P}(x_i) \log_b \mathrm{P}(x_i)$$

Where b = 10, and P(x_i) is the probability of x_i in X.

Note that if P(x_i)=0, then we consider P(x_i)log P(x_i) = 0.

Also, the closer H(X) is to zero the better (less entropy implies more information gain).

Choosing attributes

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

Example

Try splitting on **Alt**

Alt=T; W=T $\longrightarrow$ $\{x_1, x_4, x_{12}\}$; W=F $\longrightarrow$ $\{x_2, x_5, x_{10}\}$

Alt=F; W=T $\longrightarrow$ $\{x_3, x_6, x_8\}$; W=F $\longrightarrow$ $\{x_7, x_9, x_{11}\}$

$$
\begin{aligned}
H(\text{Alt}) &= -(3/6)\log(3/6) - (3/6)\log(3/6) \\
&\quad -(3/6)\log(3/6) - (3/6)\log(3/6) \\
&= -(0.5)(-0.3) - (0.5)(-0.3) - (0.5)(-0.3) - (0.5)(-0-.3) \\
&= 0.15 + 0.15 + 0.15 + 0.15 \\
&= 0.6
\end{aligned}
$$

Example

Try splitting on **Pat**

Pat=Full; W=T $\rightarrow$ $\{x_4, x_{12}\}$; W=F $\rightarrow$ $\{x_2, x_5, x_9, x_{10}\}$

Pat=Some; W=T $\rightarrow$ $\{x_1, x_3, x_6, x_8\}$; W=F $\rightarrow$ $\{\}$

Pat=None; W=T $\rightarrow$ $\{\}$; W=F $\rightarrow$ $\{x_7, x_{11}\}$

$$\begin{aligned} H(\text{Pat}) &= -(2/6)\log(2/6) - (4/6)\log(4/6) - (4/4)\log(4/4) \\ &\quad -(0/4)\log(0/4) - (0/2)\log(0/2) - (2/2)\log(2/2) \\ &= 0.16 + 0.12 + 0 + 0 + 0 + 0 \\ &= 0.28 \end{aligned}$$

Example

Having X = $\{x_1, x_2, \ldots, x_n\}$, the Gini index of X is defined as follows:

$$G(X) = \Sigma_{i=1}^{n} P(x_i)^2$$

Where $P(x_i)$ is the probability of x_i in X.

The minimum value is 0 when items are all of one class.

The maximum value is 1 – 1/*n* when the distribution is even.

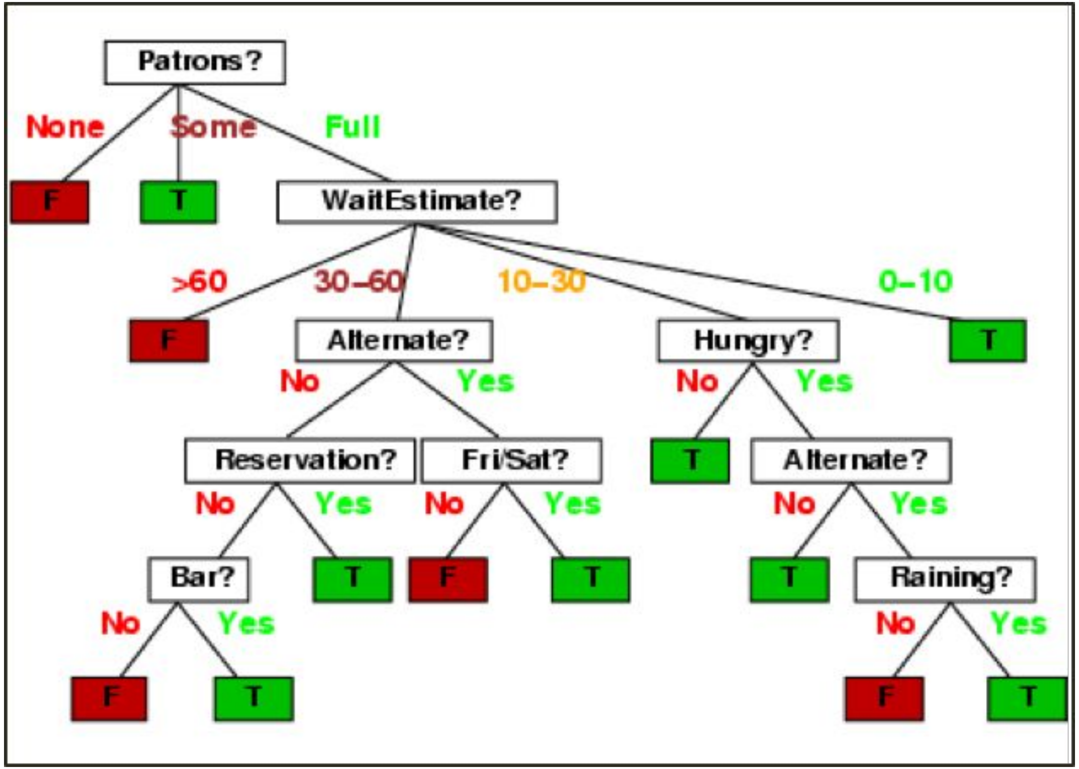
Gini index

Decision Rules

- We can *linearize* a decision tree by constructing decisions from branches

- There is one rule per leaf node in the tree based on the path from the root

- Rules take the form
  if *condition1* and *condition2* …
  then *outcome*



Decision Trees

Stopping conditions

- Maximum depth
- Minimum samples required to split
- Maximum number of features

Decision tree pruning

- To avoid overfitting, we may want to prune the decision tree
- For example, replace nodes with the most popular class starting bottom up
- Check the error against the data to decide when to stop

Decision trees

- Decision trees are easy to understand, explain, and evaluate
- Easy to construct with limited data
- Unstable and prone to change with small changes in input data
- Biased toward attributes with more categories

Bagging

- If one decision tree is prone to error, try many of them!
- Bootstrap aggregation: many decision trees on different subsets of the data
- Take the majority decision across all decision trees

Random forests

- Certain attributes will tend to dominate the decision nodes constructed
- In addition, we restrict each decision tree to $\sqrt{(p)}$ attributes of a possible p
- This gives much better performance and is very commonly used

Regression trees

- Decision trees can also be used to predict non-categorical values
- Calculate possible split points based on boundaries in the input data
- Select which split to use based on the sum of squared errors (SSE)
- Stop when the reduction in SSE is small enough and predict the mean