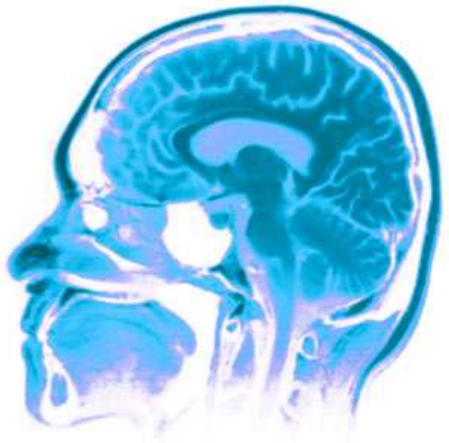
CPSC340



Decision trees

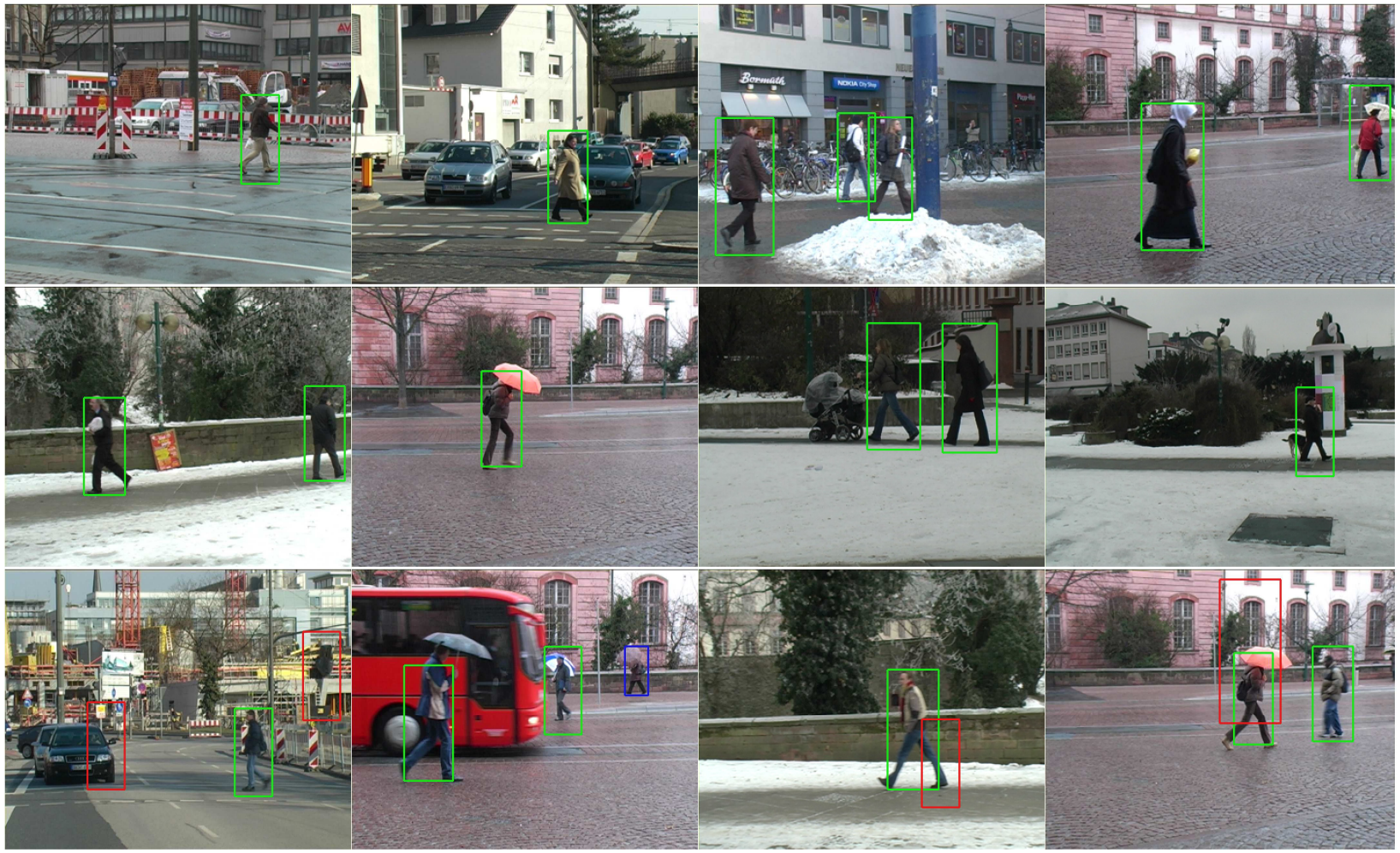
Nando de Freitas
November, 2012
University of British Columbia

Outline of the lecture

This lecture provides an introduction to decision trees. It discusses:

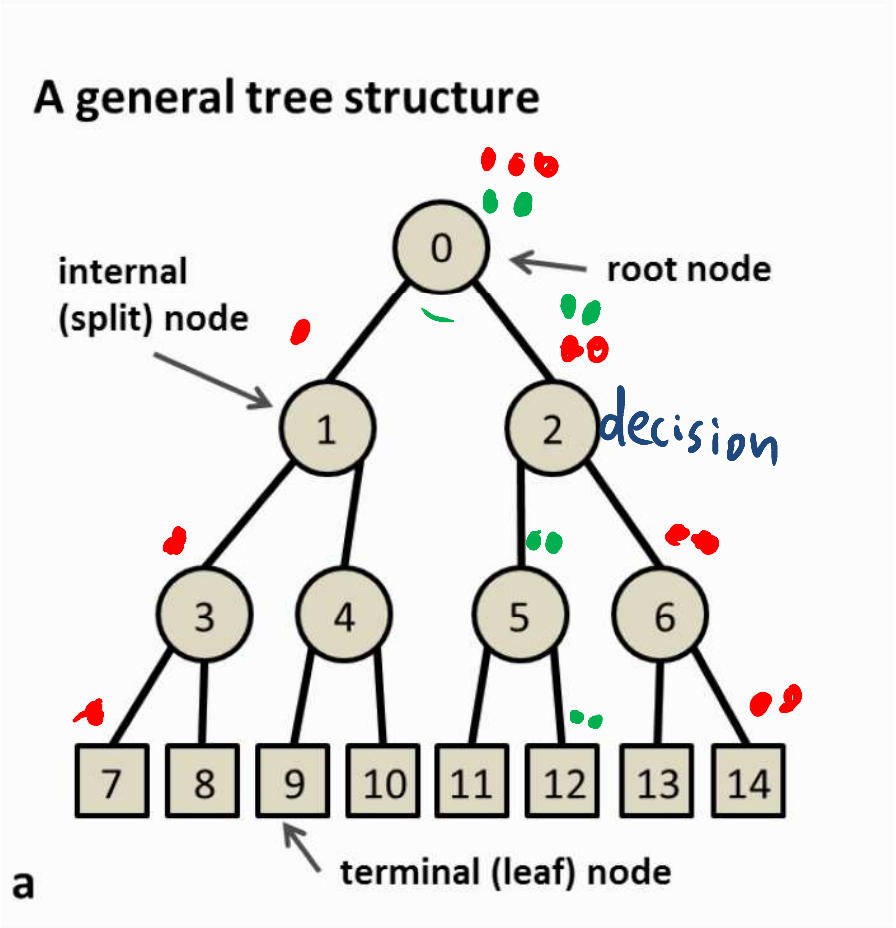
- ❑ Decision trees
- ❑ Using reduction in entropy as a criterion for constructing decision trees.
- ❑ Application of decision trees to classification

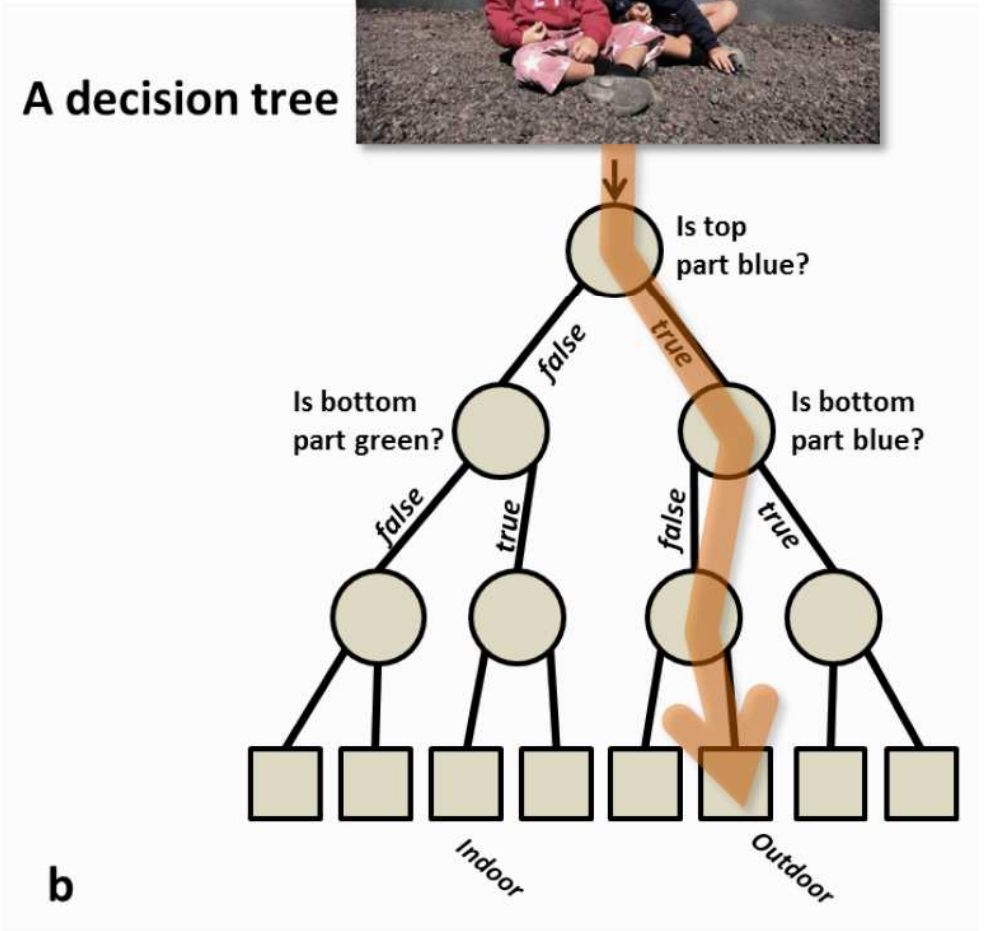
Motivation example 1: object detection



Motivation example 2: Kinect

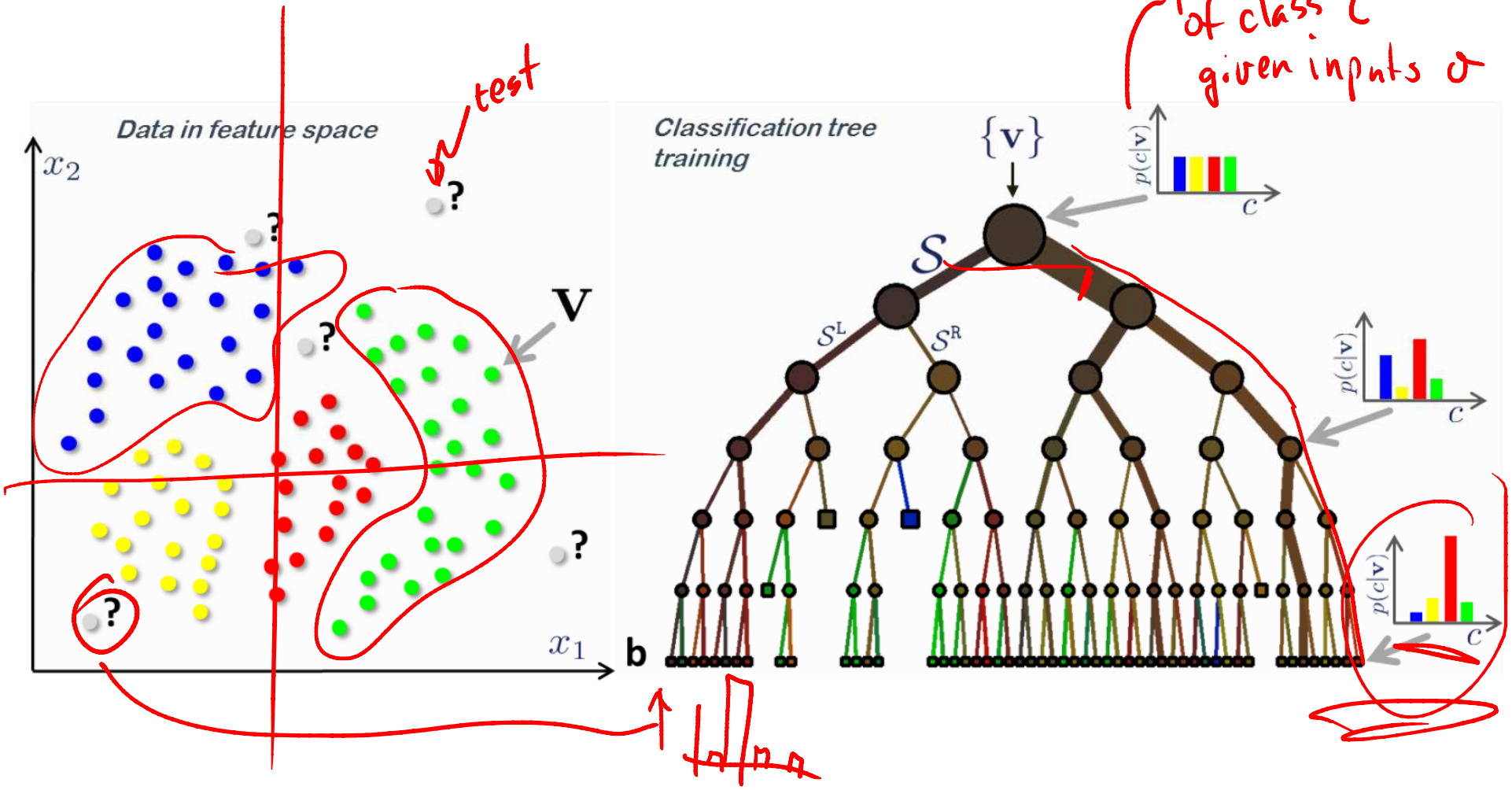
Image classification example





[Criminisi et al, 2011]

Classification tree

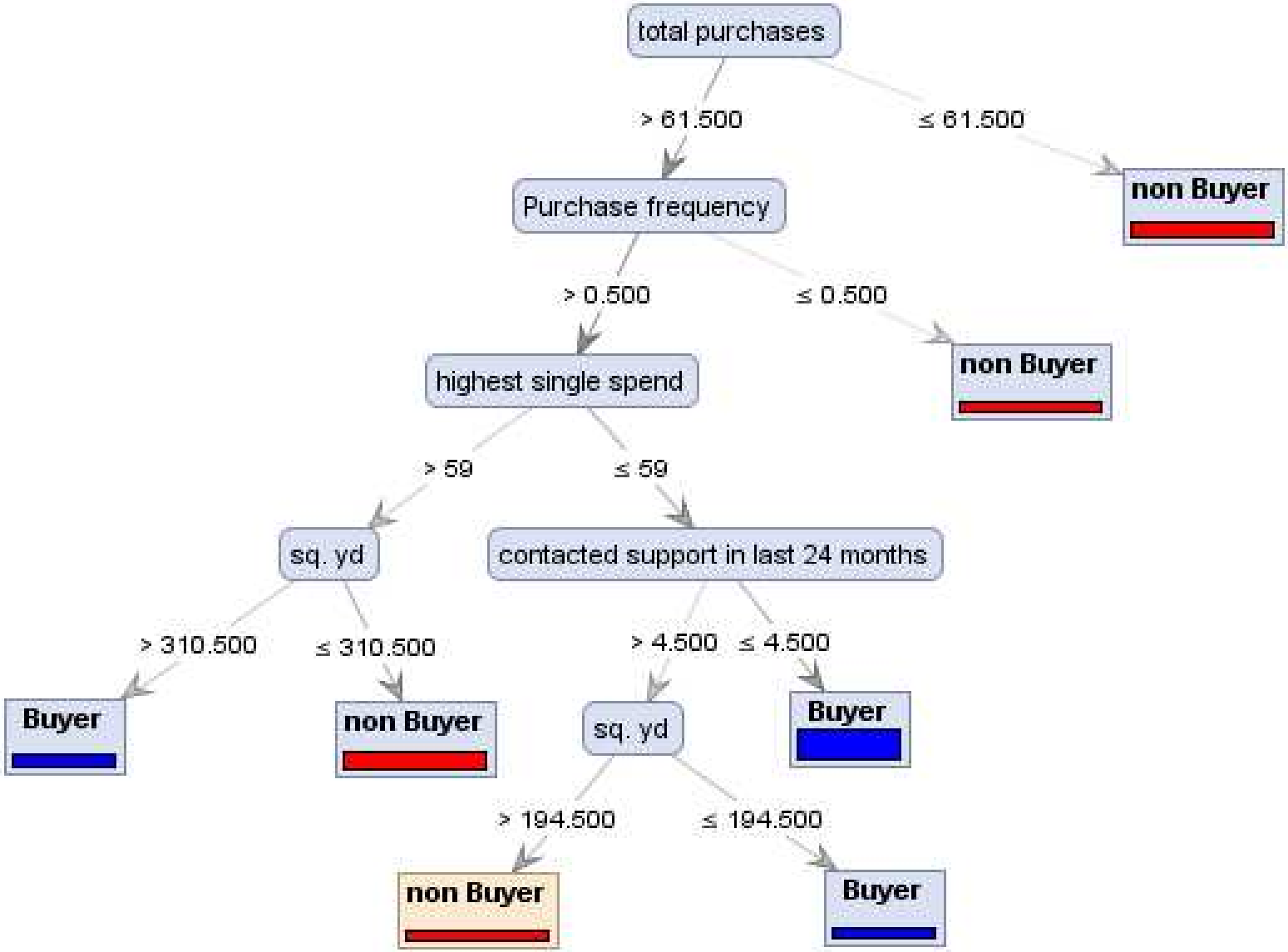


A generic data point is denoted by a vector $\mathbf{v} = (x_1, x_2, \cdots, x_d)$

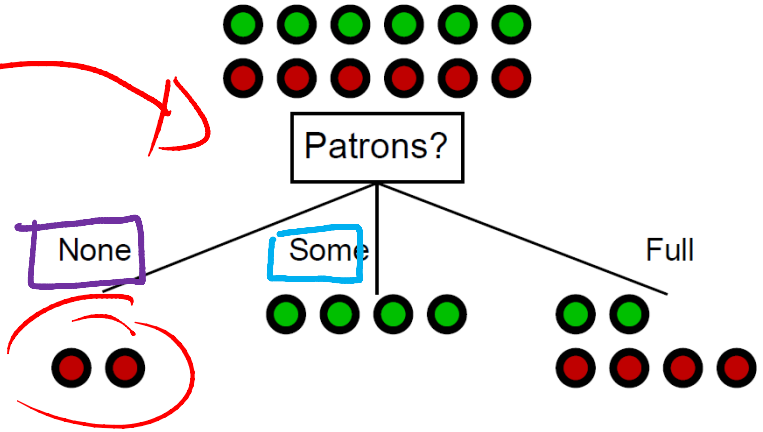
$\mathcal{S}_j = \mathcal{S}_j^{\mathrm{L}} \cup \mathcal{S}_j^{\mathrm{R}}$

[Criminisi et al, 2011]

Another commerce example



From a spreadsheet to a decision node



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

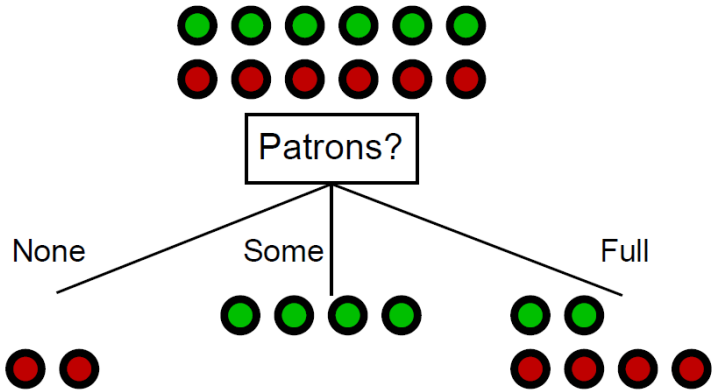
Patrons ?

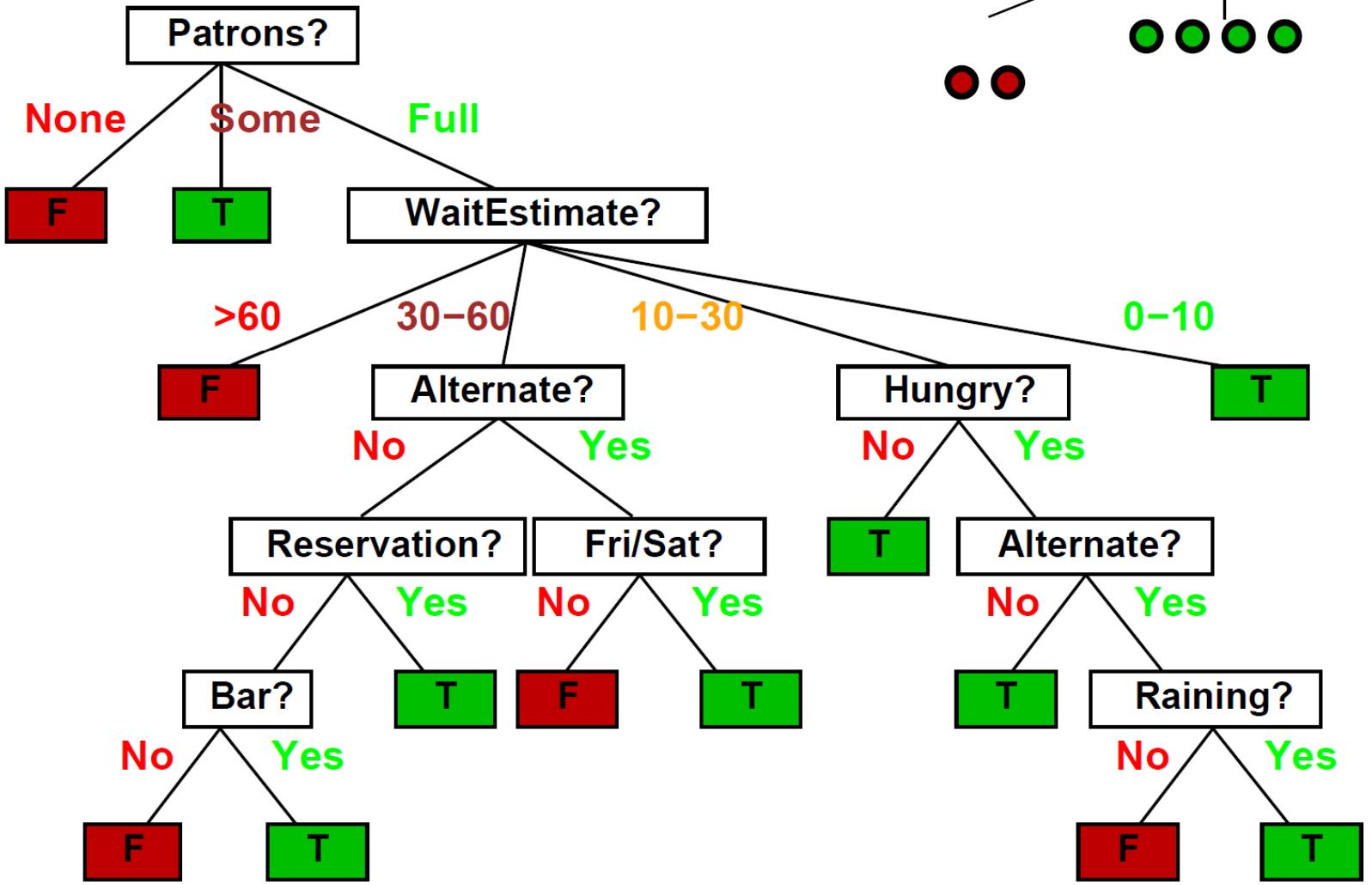
Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F	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)

[AI book of Stuart Russell and Peter Norvig]

A learned decision tree

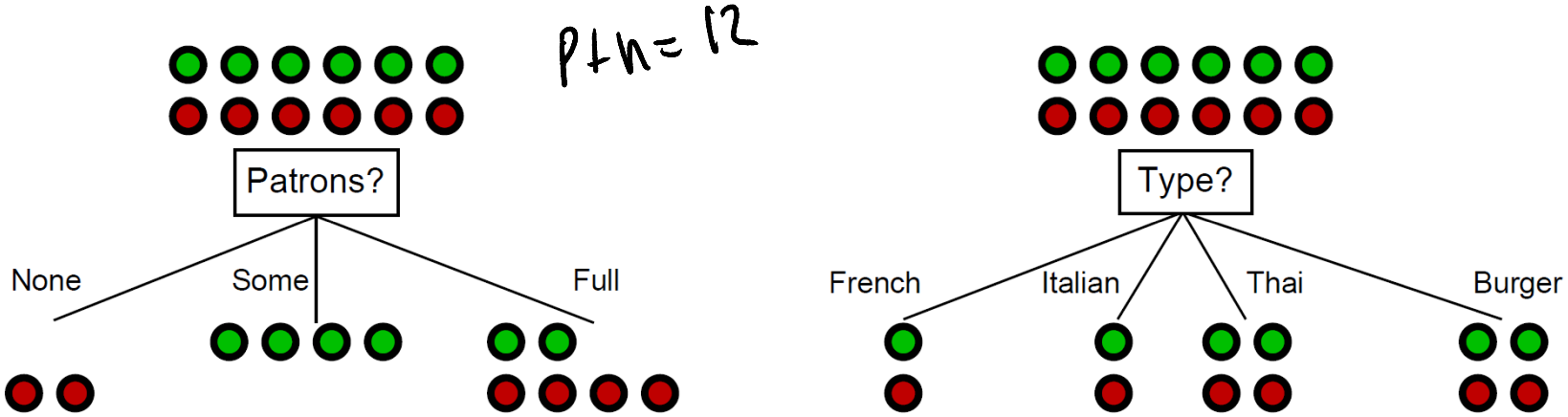




[AI book of Stuart Russell and Peter Norvig]

How do we construct the tree ? i.e., how to pick attribute (nodes)?

Idea: a good attribute splits the examples into subsets that are (ideally) "all positive" or "all negative"



Patrons? is a better choice—gives **information** about the classification

For a training set containing p positive examples and n negative examples, we have:

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

How to pick nodes?

- A chosen attribute ***A***, with ***K*** distinct values, divides the training set ***E*** into subsets $E_1, \ldots, E_K$.
- The **Expected Entropy (EH) remaining** after trying attribute ***A*** (with branches *i=1,2,…,**K***) is

$$EH(A) = \sum_{i=1}^{K} \frac{p_i + n_i}{p + n} H\left(\frac{p_i}{p_i + n_i}, \frac{n_i}{p_i + n_i}\right)$$

- **Information gain (I)** or **reduction in entropy** for this attribute is:

$$I(A) = H\left(\frac{p}{p+n}, \frac{n}{p+n}\right) - EH(A)$$

- Choose the attribute with the largest I

[Hwee Tou Ng & Stuart Russell]

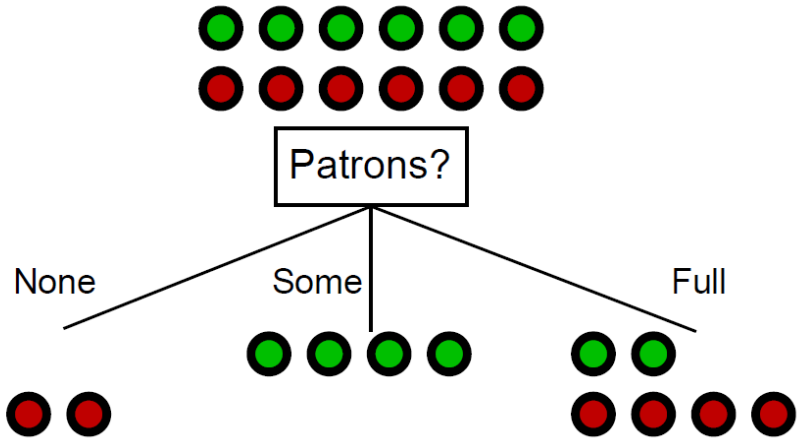
Example

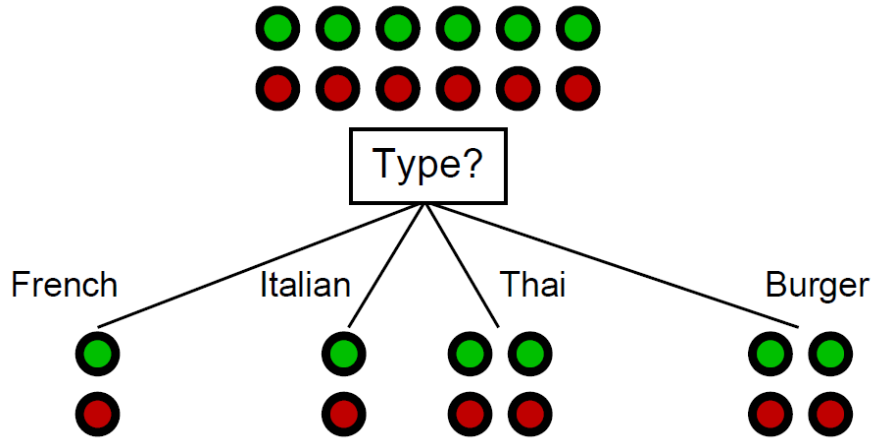
- **Convention**: For the training set, $p = n = 6$, $H(6/12, 6/12) = 1$ bit

- Consider the attributes *Patrons* and *Type* (and others too):

$$I(Patrons) = 1 - [\frac{2}{12} H(0,1) + \frac{4}{12} H(1,0) + \frac{6}{12} H(\frac{2}{6}, \frac{4}{6})] = .0541 \text{ bits}$$

$$I(Type) = 1 - [\frac{2}{12} H(\frac{1}{2}, \frac{1}{2}) + \frac{2}{12} H(\frac{1}{2}, \frac{1}{2}) + \frac{4}{12} H(\frac{2}{4}, \frac{2}{4}) + \frac{4}{12} H(\frac{2}{4}, \frac{2}{4})] = 0 \text{ bits}$$





[Hwee Tou Ng & Stuart Russell]

Weighted entropy is needed

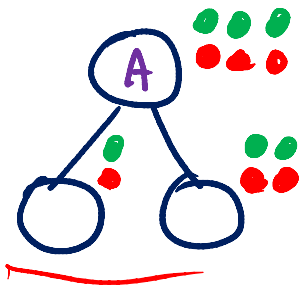
We want to optimize the **expected information gain** of an unseen **test** point (not in training data).

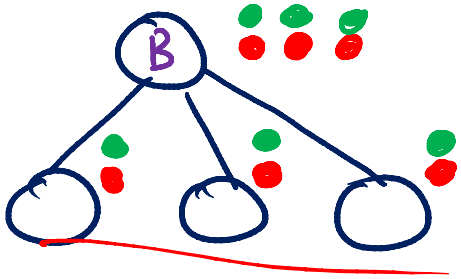
Since we cannot know what that test point will be (e.g., in the case of the restaurant data, we cannot know whether the next data point will have as a *Patrons* attribute, *none*, *some*, *or full*), we have to take an expectation over the distribution of data.

Since we do not know how the data is intrinsically distributed, we use the empirical distribution of the training data. Therefore when doing the split at a node, we need to weight the entropy of the leaf nodes according to the train population that goes in each leaf (what fraction of the training data went into which leaf).

Weighted entropy is needed

We need to used the weighted entropy of the children (as we did when we studied **maximum expected utility**). We need to weight by the probability of the data reaching each of the children nodes. If we exclude these weights, bad things happen, e.g. 2,





$$I(A) = 1 - \frac{2}{6}(1) - \frac{4}{6}(1) = 0$$ ✓

$$I(B) = 1 - \frac{1}{3}(1) - \frac{1}{3}(1) - \frac{1}{3}(1) = 0$$

No weighting

$1 - 1 - 1 = -1$ ✓

$1 - 1 - 1 - 1 = -2$

Next lecture

The next lecture introduces random forests.