Learning Decision Trees

Chapter 1~~8~~9, Section 3

Attribute-based representations

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

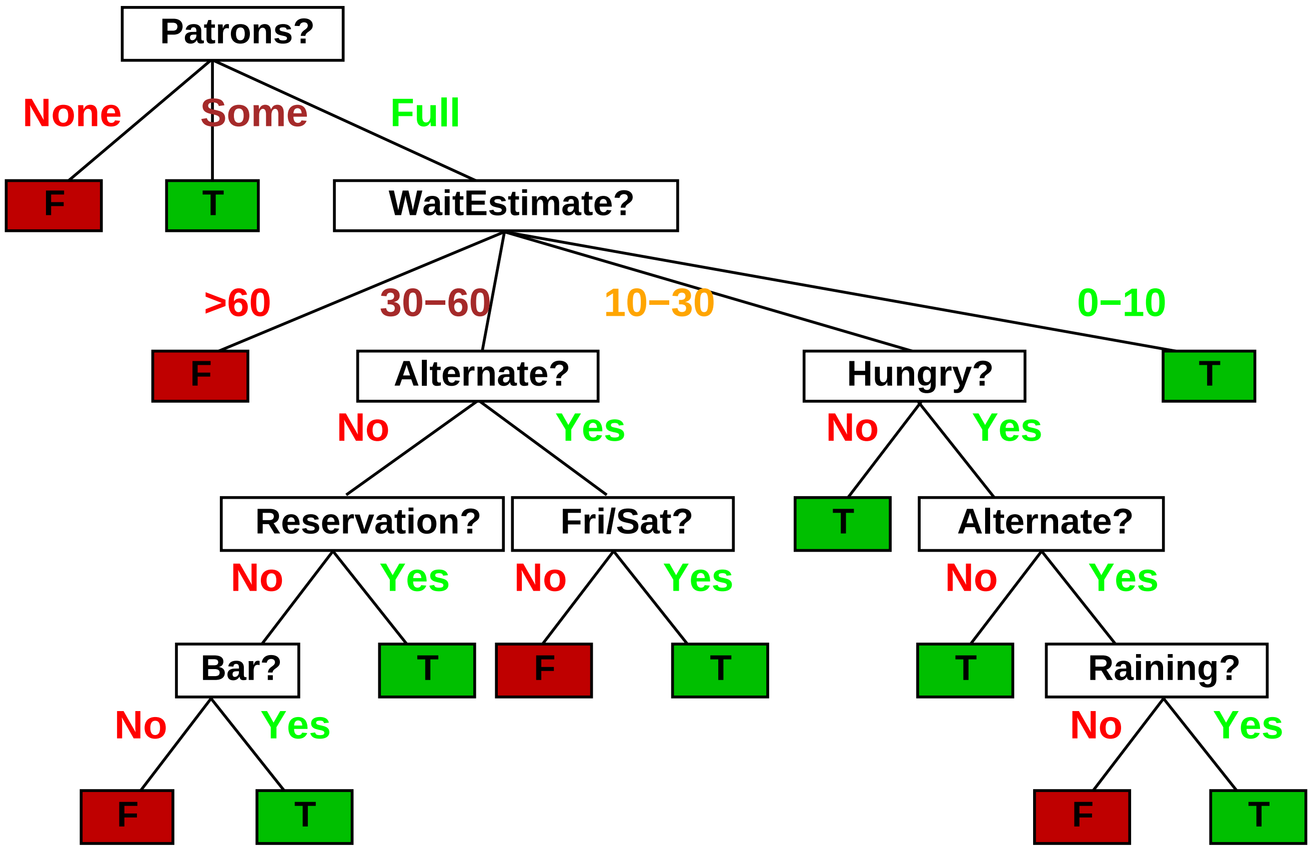
Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	*T*	*F*	*F*	*T*	*Some*	*$$$*	*F*	*T*	*French*	*0–10*	*T*
X_2	*T*	*F*	*F*	*T*	*Full*	*$*	*F*	*F*	*Thai*	*30–60*	*F*
X_3	*F*	*T*	*F*	*F*	*Some*	*$*	*F*	*F*	*Burger*	*0–10*	*T*
X_4	*T*	*F*	*T*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

Decision trees

One possible representation for hypotheses

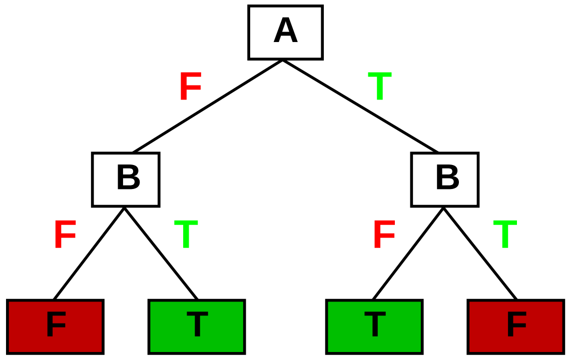
E.g., here is the "true" tree for deciding whether to wait for a table:



Expressiveness

Decision trees can express any function of the input attributes.
E.g., for Boolean functions, truth table row $\rightarrow$ 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 w/ one path to leaf for each example (unless f nondeterministic in x) but it probably won't generalize to new examples

Prefer to find more **compact** decision trees

Hypothesis spaces

How many distinct decision trees with n Boolean attributes??

Hypothesis spaces

How many distinct decision trees with n Boolean attributes??

$\approx$ number of Boolean functions

Hypothesis spaces

How many distinct decision trees with n Boolean attributes??

$\approx$ number of Boolean functions
$=$ number of distinct truth tables with 2^n rows

Hypothesis spaces

How many distinct decision trees with n Boolean attributes??

$\approx$ number of Boolean functions
$=$ number of distinct truth tables with 2^n rows $= 2^{2^n}$

Hypothesis spaces

How many distinct decision trees with n Boolean attributes??

$\approx$ number of Boolean functions
$=$ number of distinct truth tables with 2^n rows $= 2^{2^n}$

E.g., with 6 Boolean attributes, there are 18,446,744,073,709,551,616 trees

Hypothesis spaces

How many distinct decision trees with n Boolean attributes??

$\approx$ number of Boolean functions
$=$ number of distinct truth tables with 2^n rows $= 2^{2^n}$

E.g., with 6 Boolean attributes, there are 18,446,744,073,709,551,616 trees

How many purely conjunctive hypotheses (e.g., $Hungry \land \neg Rain$)??

Hypothesis spaces

How many distinct decision trees with n Boolean attributes??

$\approx$ number of Boolean functions
$=$ number of distinct truth tables with 2^n rows $= 2^{2^n}$

E.g., with 6 Boolean attributes, there are 18,446,744,073,709,551,616 trees

How many purely conjunctive hypotheses (e.g., $Hungry \wedge \neg Rain$)??

Each attribute can be in (positive), in (negative), or out
$\Rightarrow$ 3^n distinct conjunctive hypotheses

More expressive hypothesis space
– increases chance that target function can be expressed
– increases number of hypotheses consistent w/ training set
$\Rightarrow$ may get worse predictions

Decision tree learning

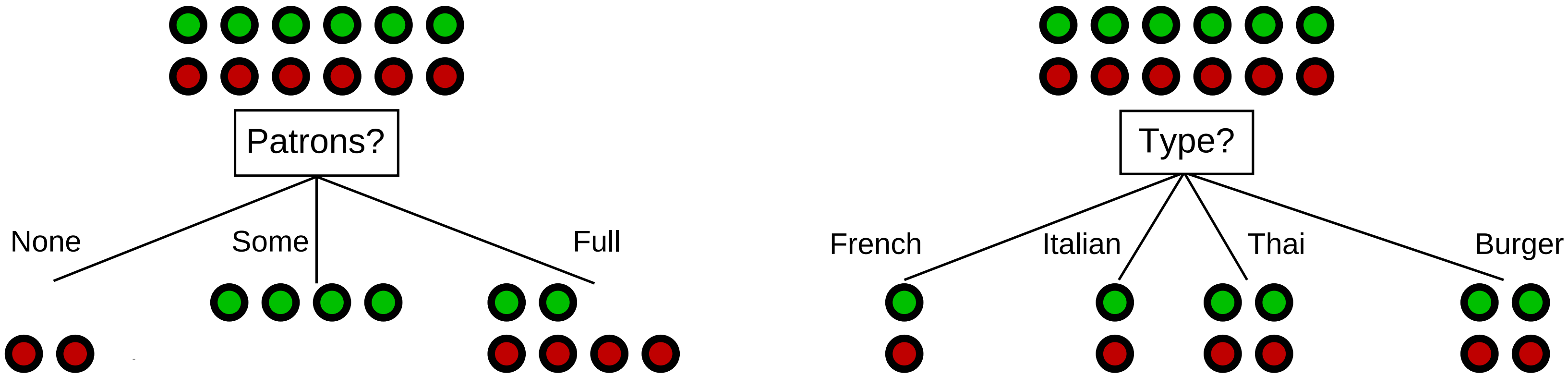
Aim: find a small tree consistent with the training examples

Idea: (recursively) choose "most significant" attribute as root of (sub)tree

```
function DTL(examples, attributes, default) returns a decision tree
   if examples is empty then return default
   else if all examples have the same classification then return the classification
   else if attributes is empty then return MODE(examples)
   else
       best ← CHOOSE-ATTRIBUTE(attributes, examples)
       tree ← a new decision tree with root test best
       for each value v_i of best do
           examples_i ← {elements of examples with best = v_i}
           subtree ← DTL(examples_i, attributes − best, MODE(examples))
           add a branch to tree with label v_i and subtree subtree
       return tree
```

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—gives **information** about the classification

Information

Information answers questions

The more clueless I am about the answer initially, the more information is contained in the answer

Scale: 1 bit = answer to Boolean question with prior $\langle 0.5, 0.5 \rangle$

Information in for an outcome X with probability $P(X)$ is

$$I(X) = \log_2 \frac{1}{P(X)} = -\log_2 P(X)$$

Information is high if $P(X)$ is low.

Information is zero if X is sure ($P(X) = 1$).

Information contd.

Information in a partioning C of objects into classes C_1, ..., C_n is the expected information of a test object.

$$H(\langle P(C_1), \ldots, P(C_k)\rangle) = \sum_{k=1}^{n} P(C_k) * I(C_k) = \sum_{k=1}^{n} -P(C_k) * \log_2 P(C_k)$$

This is also called entropy. It is a measure of unpredictability of the information.

We only have estimates for the probabilities:

$$P(C_k) \approx \frac{\text{number of examples in class } C_k}{\text{total number of examples}}$$

Suppose we have p positive and n negative examples at the root

$$\Rightarrow \quad P(\text{positive}) = \bar{p} = p/(p+n)$$
$$\Rightarrow \quad P(\text{negative}) = \bar{n} = n/(p+n)$$

$H(\langle \bar{p}, \bar{n}\rangle)$ bits needed to classify a new example
E.g., for 12 restaurant examples, $p = n = 6$ so we need 1 bit of information

Information contd.

An attribute splits the examples E into subsets E_i, each of which (we hope) needs less information to complete the classification

Let E_i have p_i positive and n_i negative examples

$\Rightarrow$ $H(\langle \bar{p}_i, \bar{n}_i \rangle)$ bits needed to classify a new example

$\Rightarrow$ **expected** number of bits per example over all branches is

$$\Sigma_i \ \frac{p_i + n_i}{p + n} H(\langle \bar{p}_i, \bar{n}_i \rangle)$$

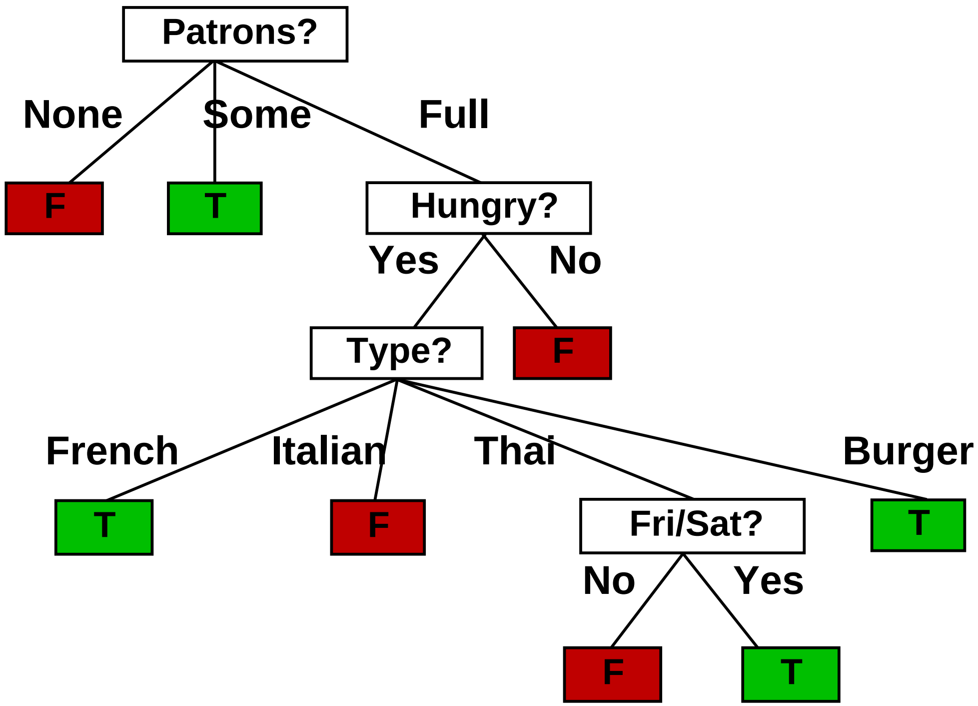
For $Patrons?$, this is 0.459 bits, for $Type$ this is (still) 1 bit

$\Rightarrow$ choose the attribute that minimizes the remaining information needed

(This state-of-the-art decision tree learning algorithm is called ID3 or, with slight improvements, C4.5.)

Example contd.

Decision tree learned from the 12 examples:



Substantially simpler than "true" tree—a more complex hypothesis isn't justified by small amount of data

Performance measurement

How do we know that we found a good hypothesis, i.e., $h \approx f$?

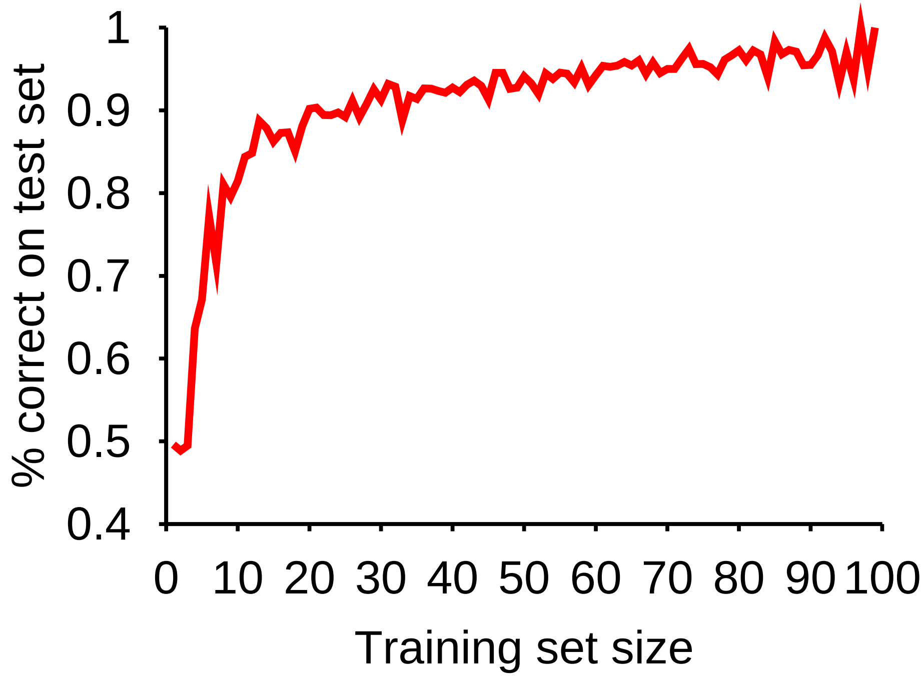
Hume's **Problem of Induction**: Just because we have not seen any different examples, does that mean there are none?

Performance measurement

How do we know that we found a good hypothesis, i.e., $h \approx f$?

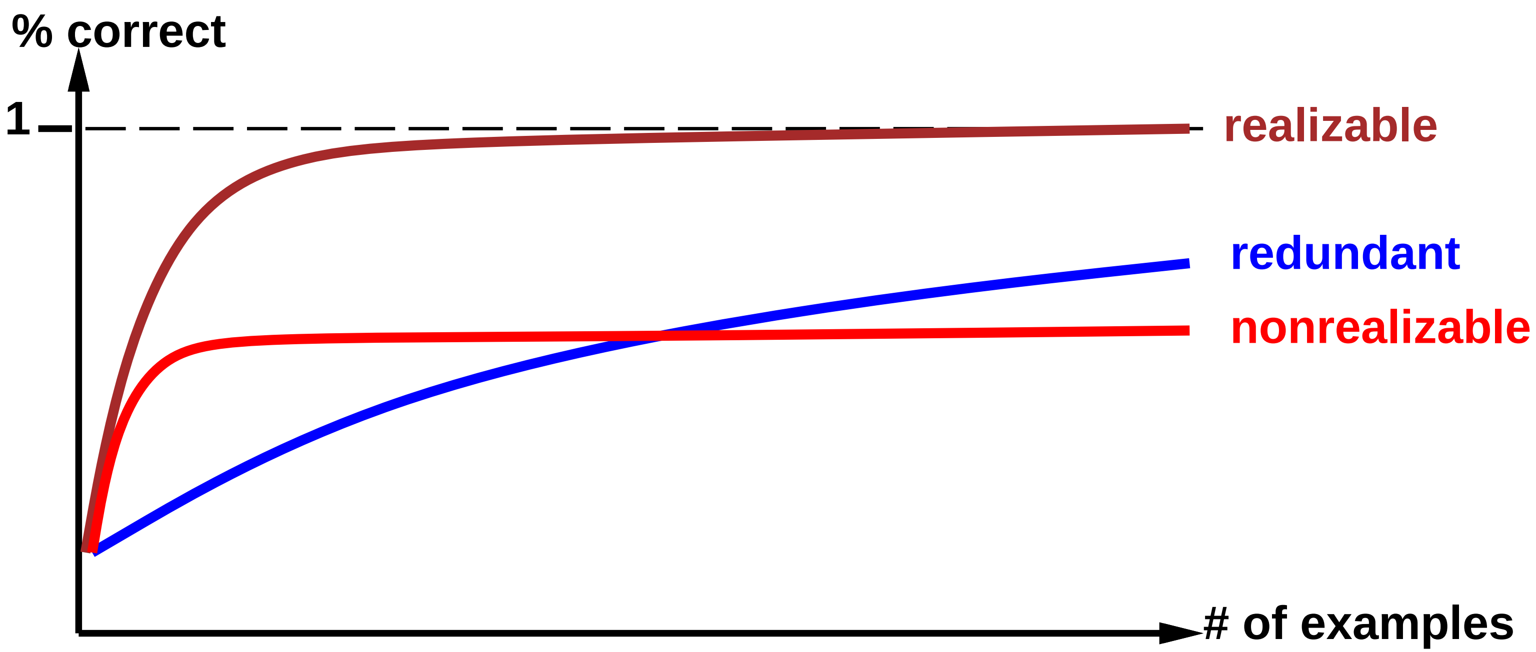
Try h on a new test set of examples
(use **same distribution over example space** as training set)

Learning curve = % correct on test set as a function of training set size



Performance measurement contd.

Learning curve depends on
- realizable (can h express target function) vs. non-realizable
  non-realizability can be due to missing attributes
  or restricted hypothesis class (e.g., thresholded linear function)
- redundant expressiveness (e.g., loads of irrelevant attributes)
- noise in the training data



Summary

For supervised learning, the aim is to find a **simple hypothesis** that is approximately consistent with training examples

Decision tree learning using information gain

Learning performance = prediction accuracy measured on test set