
Decision Trees

Hadi Moradi

AIMA 2nd edition

Russell, Norvig

Chapter 18

Introduction

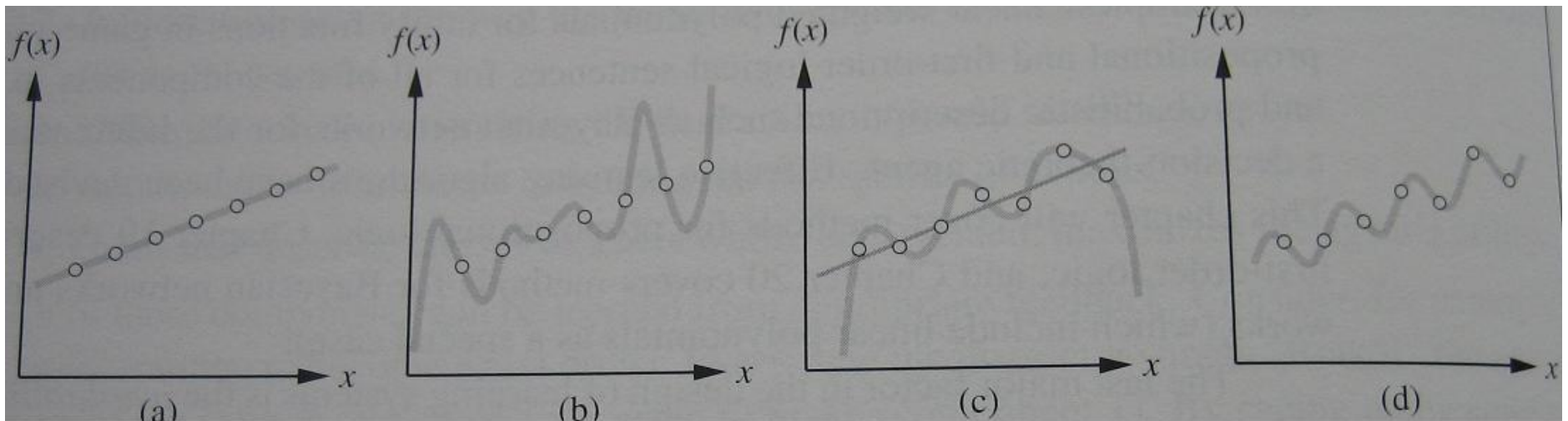
- Where are we?
 - We have studied
 - PL, FOL
 - Neural Networks (learning)
 - Uncertainty
 - Bayesian Network
 - Making decision

Learning

- Supervised learning
 - Learn from examples of the inputs and outputs
- Unsupervised learning
- Reinforcement Learning

Inductive Learning

- Given a collection of examples of f , return a function h that approximate f .
 - A supervised learning approach



- A learning problem is realizable if the hypothesis space contains the true function.
 - Otherwise unrealizable

Here is a dataset

age	employe	education	edun	marital	...	job	relation	race	gender	hour	country	wealth
39	State_gov	Bachelors	13	Never_mar	...	Adm_cleric	Not_in_fan	White	Male	40	United_Stat	poor
51	Self_emp_	Bachelors	13	Married	...	Exec_man	Husband	White	Male	13	United_Stat	poor
39	Private	HS_grad	9	Divorced	...	Handlers_c	Not_in_fan	White	Male	40	United_Stat	poor
54	Private	11th	7	Married	...	Handlers_c	Husband	Black	Male	40	United_Stat	poor
28	Private	Bachelors	13	Married	...	Prof_speci	Wife	Black	Female	40	Cuba	poor
38	Private	Masters	14	Married	...	Exec_man	Wife	White	Female	40	United_Stat	poor
50	Private	9th	5	Married_sp	...	Other_serv	Not_in_fan	Black	Female	16	Jamaica	poor
52	Self_emp_	HS_grad	9	Married	...	Exec_man	Husband	White	Male	45	United_Stat	rich
31	Private	Masters	14	Never_mar	...	Prof_speci	Not_in_fan	White	Female	50	United_Stat	rich
42	Private	Bachelors	13	Married	...	Exec_man	Husband	White	Male	40	United_Stat	rich
37	Private	Some_coll	10	Married	...	Exec_man	Husband	Black	Male	80	United_Stat	rich
30	State_gov	Bachelors	13	Married	...	Prof_speci	Husband	Asian	Male	40	India	rich
24	Private	Bachelors	13	Never_mar	...	Adm_cleric	Own_child	White	Female	30	United_Stat	poor
33	Private	Assoc_acc	12	Never_mar	...	Sales	Not_in_fan	Black	Male	50	United_Stat	poor
41	Private	Assoc_voc	11	Married	...	Craft_repa	Husband	Asian	Male	40	*MissingV	rich
34	Private	7th_8th	4	Married	...	Transport_	Husband	Amer_Indi	Male	45	Mexico	poor
26	Self_emp_	HS_grad	9	Never_mar	...	Farming_fi	Own_child	White	Male	35	United_Stat	poor
33	Private	HS_grad	9	Never_mar	...	Machine_c	Unmarried	White	Male	40	United_Stat	poor
38	Private	11th	7	Married	...	Sales	Husband	White	Male	50	United_Stat	poor
44	Self_emp_	Masters	14	Divorced	...	Exec_man	Unmarried	White	Female	45	United_Stat	rich
41	Private	Doctorate	16	Married	...	Prof_speci	Husband	White	Male	60	United_Stat	rich
:	:	:	:	:	:	:	:	:	:	:	:	:

48,000 records, 16 attributes [Kohavi 1995]

Classification

- A Major Data Mining Operation
- Give one attribute (e.g wealth), try to predict the value of new people's wealths by means of some of the other available attributes.

About this dataset

- It is a tiny subset of the 1990 US Census.
- It is publicly available online from the UCI Machine Learning Datasets repository

Used Attributes

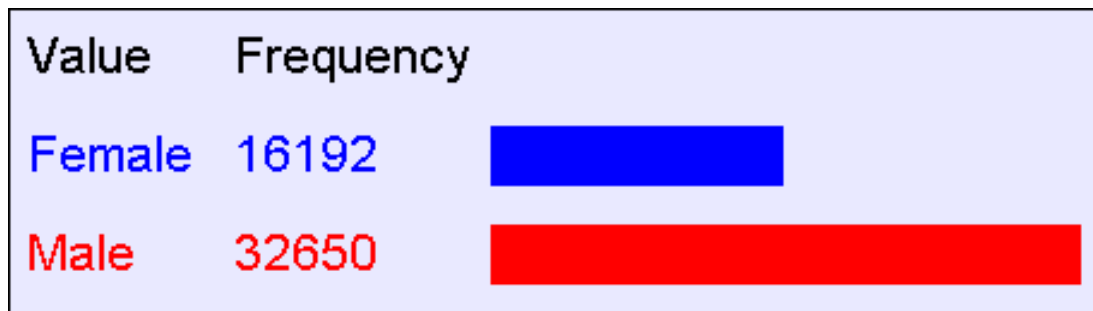
age	edunum	race	hours_worked
employment	marital	gender	country
taxweighting	job	capitalgain	wealth
education	relation	capitalloss	agegroup

This color = Real-valued This color = Symbol-valued

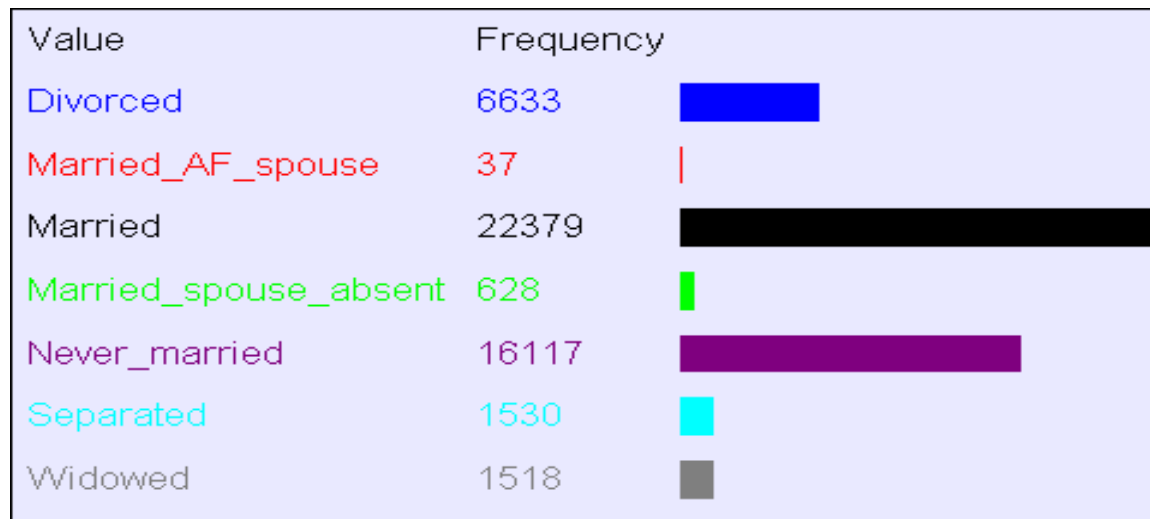
Successfully loaded a new dataset from the file \tadult.fds. It has 16 attributes and 48842 records.

What can you do with a dataset?

- Well, you can look at histograms...

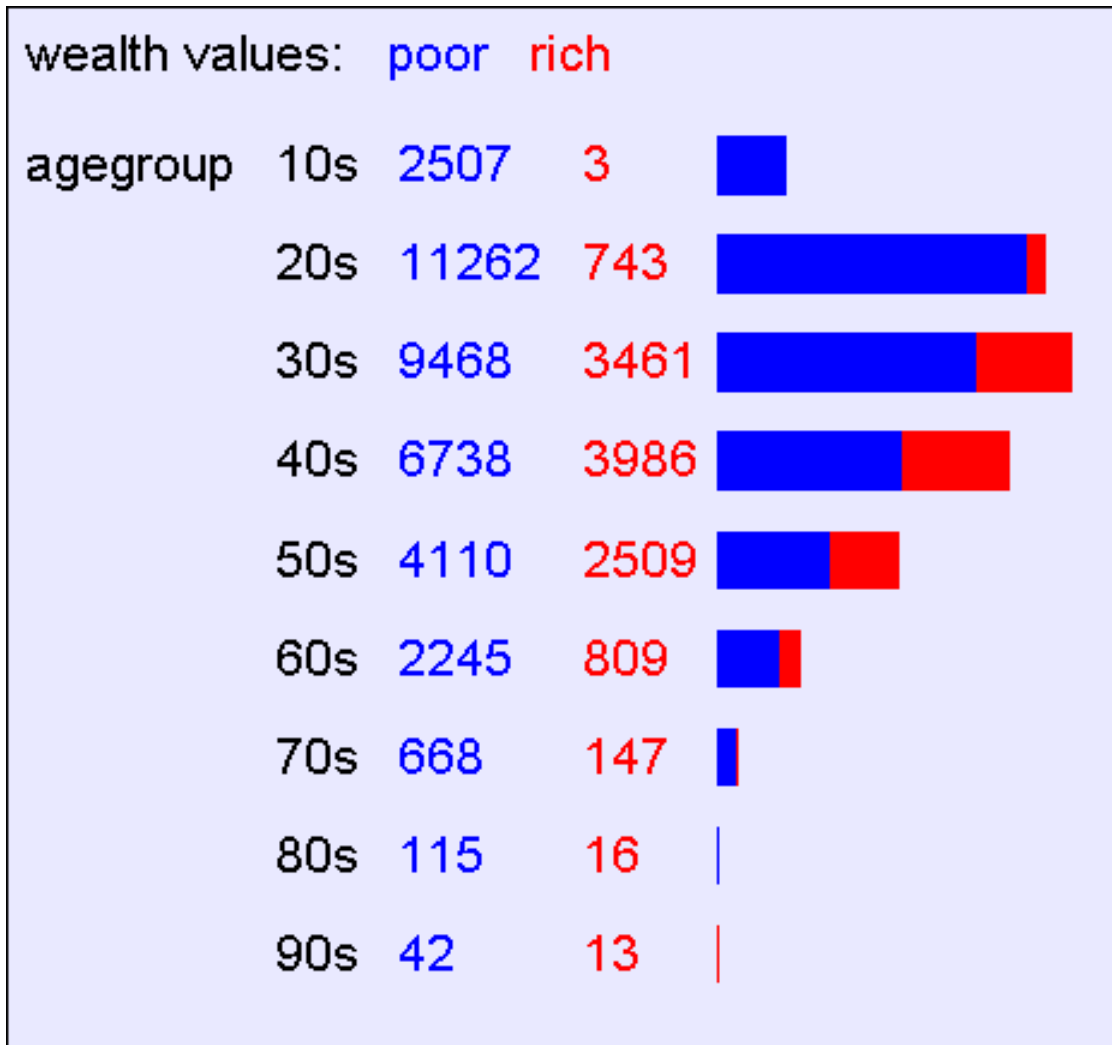


Gender



Marital
Status

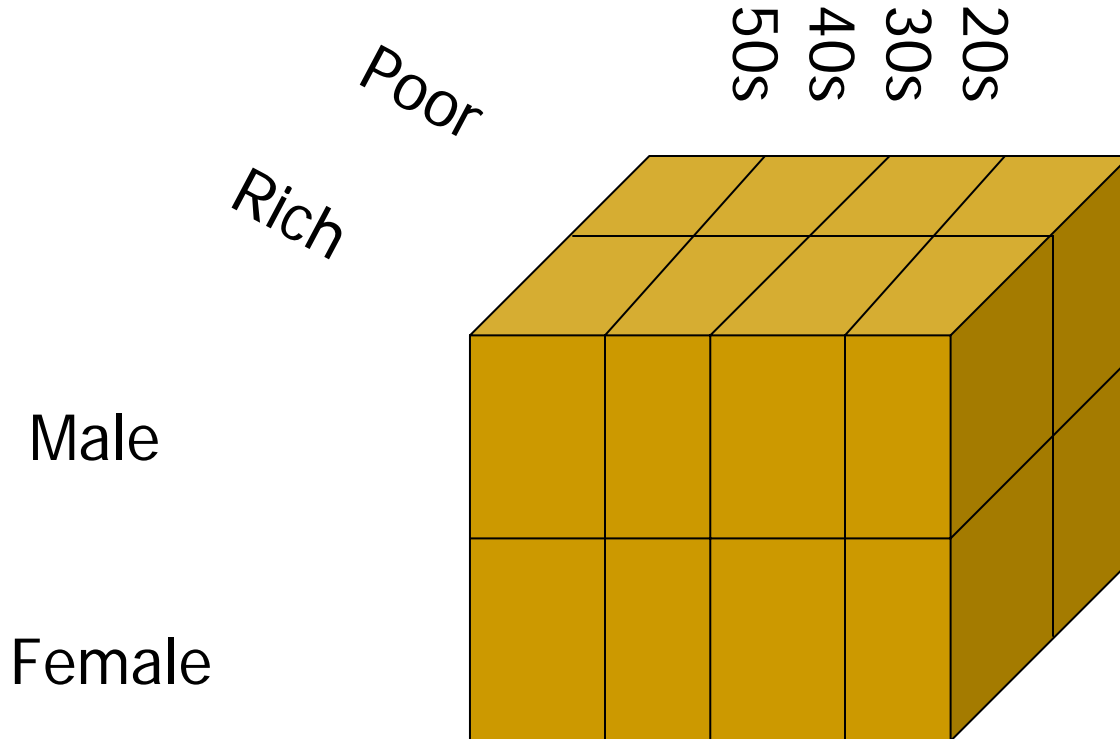
A 2-d Contingency Table



- Easier to appreciate graphically

3-d contingency tables (Histogram)

- These are harder to look at!



Let's think

- With 16 attributes, how many 1-d contingency tables are there?
- How many 2-d contingency tables?
- How many 3-d tables?
- With 100 attributes how many 3-d tables are there?

Data Mining

- Data Mining is all about automating the process of searching for patterns in the data.

Which patterns are interesting?

Which might be mere illusions?

And how can they be exploited?

Deciding whether a pattern is interesting

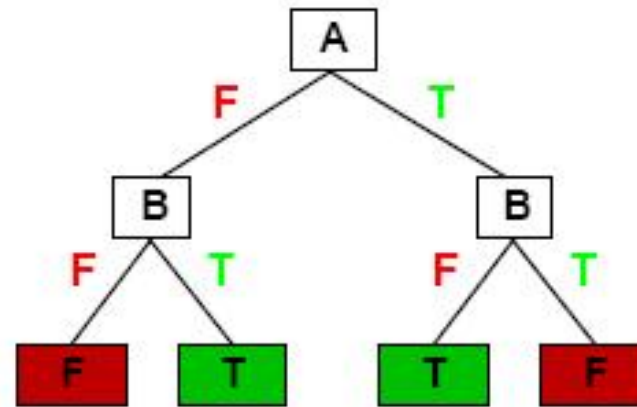
- We will use **information theory**
- A very large topic, originally used for compressing signals
- But more recently used for data mining...

Some Other Examples

- Considering multiple attributes:
 - Which play will work in sports?
 - Who should a bank give loans to?
 - What graduate student characteristics will lead to successful PhD process?
- Not remotely obvious or easy.

A Simple Decision Tree

A	B	A xor B
F	F	F
F	T	T
T	F	T
T	T	F



Restaurant Example

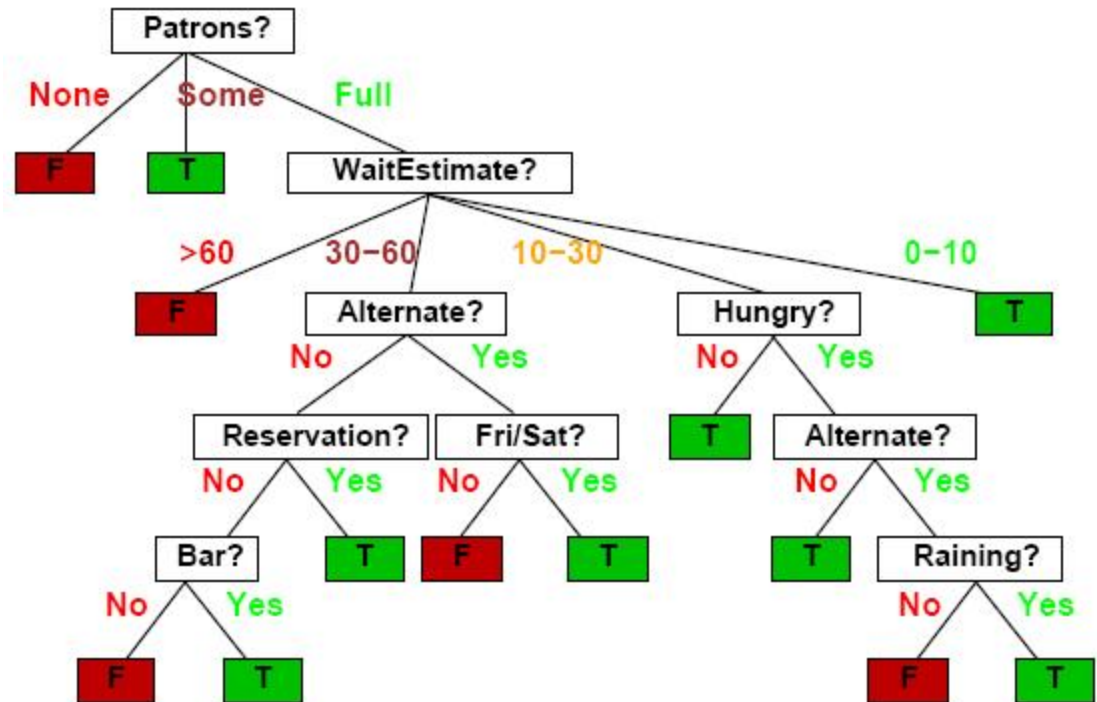
Example	Attributes										Target
	<i>Alt</i>	<i>Bar</i>	<i>Fri</i>	<i>Hun</i>	<i>Pat</i>	<i>Price</i>	<i>Rain</i>	<i>Res</i>	<i>Type</i>	<i>Est</i>	<i>WillWait</i>
X_1	<i>T</i>	<i>F</i>	<i>F</i>	<i>T</i>	<i>Some</i>	<i>\$\$\$</i>	<i>F</i>	<i>T</i>	<i>French</i>	<i>0-10</i>	<i>T</i>
X_2	<i>T</i>	<i>F</i>	<i>F</i>	<i>T</i>	<i>Full</i>	<i>\$</i>	<i>F</i>	<i>F</i>	<i>Thai</i>	<i>30-60</i>	<i>F</i>
X_3	<i>F</i>	<i>T</i>	<i>F</i>	<i>F</i>	<i>Some</i>	<i>\$</i>	<i>F</i>	<i>F</i>	<i>Burger</i>	<i>0-10</i>	<i>T</i>
X_4	<i>T</i>	<i>F</i>	<i>T</i>	<i>T</i>	<i>Full</i>	<i>\$</i>	<i>F</i>	<i>F</i>	<i>Thai</i>	<i>10-30</i>	<i>T</i>
X_5	<i>T</i>	<i>F</i>	<i>T</i>	<i>F</i>	<i>Full</i>	<i>\$\$\$</i>	<i>F</i>	<i>T</i>	<i>French</i>	<i>>60</i>	<i>F</i>
X_6	<i>F</i>	<i>T</i>	<i>F</i>	<i>T</i>	<i>Some</i>	<i>\$\$</i>	<i>T</i>	<i>T</i>	<i>Italian</i>	<i>0-10</i>	<i>T</i>
X_7	<i>F</i>	<i>T</i>	<i>F</i>	<i>F</i>	<i>None</i>	<i>\$</i>	<i>T</i>	<i>F</i>	<i>Burger</i>	<i>0-10</i>	<i>F</i>
X_8	<i>F</i>	<i>F</i>	<i>F</i>	<i>T</i>	<i>Some</i>	<i>\$\$</i>	<i>T</i>	<i>T</i>	<i>Thai</i>	<i>0-10</i>	<i>T</i>
X_9	<i>F</i>	<i>T</i>	<i>T</i>	<i>F</i>	<i>Full</i>	<i>\$</i>	<i>T</i>	<i>F</i>	<i>Burger</i>	<i>>60</i>	<i>F</i>
X_{10}	<i>T</i>	<i>T</i>	<i>T</i>	<i>T</i>	<i>Full</i>	<i>\$\$\$</i>	<i>F</i>	<i>T</i>	<i>Italian</i>	<i>10-30</i>	<i>F</i>
X_{11}	<i>F</i>	<i>F</i>	<i>F</i>	<i>F</i>	<i>None</i>	<i>\$</i>	<i>F</i>	<i>F</i>	<i>Thai</i>	<i>0-10</i>	<i>F</i>
X_{12}	<i>T</i>	<i>T</i>	<i>T</i>	<i>T</i>	<i>Full</i>	<i>\$</i>	<i>F</i>	<i>F</i>	<i>Burger</i>	<i>30-60</i>	<i>T</i>

Classification of examples is positive (T) or negative (F)

(This is the training set)

A Decision Tree

- Should you wait for a table in a restaurant.
 - It is manually generated.
 - Notice it doesn't use price for test



Should we wait at a restaurant?

Inductive Learning

- The table only shows 12 different cases
- The table has 10 attributes
 - 6 are 2-valued (alternative, bar, fri/sat, hungry, rain, reservation)
 - 2 are 3-valued (patrons, price)
 - 2 are 4-valued (type, estimate waiting time)
- To do a full table would have over 9,200 rows. ($2^6 + 3^2 + 4^2$)
- Each row has to be analyzed to give the Yes or No answer.

Hypothesis Spaces

- How many distinct decision trees with n Boolean attributes?
 - = Number of Boolean functions
 - = Number of distinct truth tables with 2^n rows = $2^{(2^n)}$
 - For 6 Boolean attributes, 18.4×10^{18} different trees.
 - Example:
 - $\text{Hungry} \wedge \sim \text{Rain}$

Inductive Learning –

How to build a decision tree with limited data

- We start with the learning vector e.g the restaurant table
 - $\{(x_i, y_i) \dots (x_n, y_n)\}$
 - where x_i is the vector of values
 - y_i is the output.
- We need a heuristic algorithm to find small tree.

Decision Tree Learning

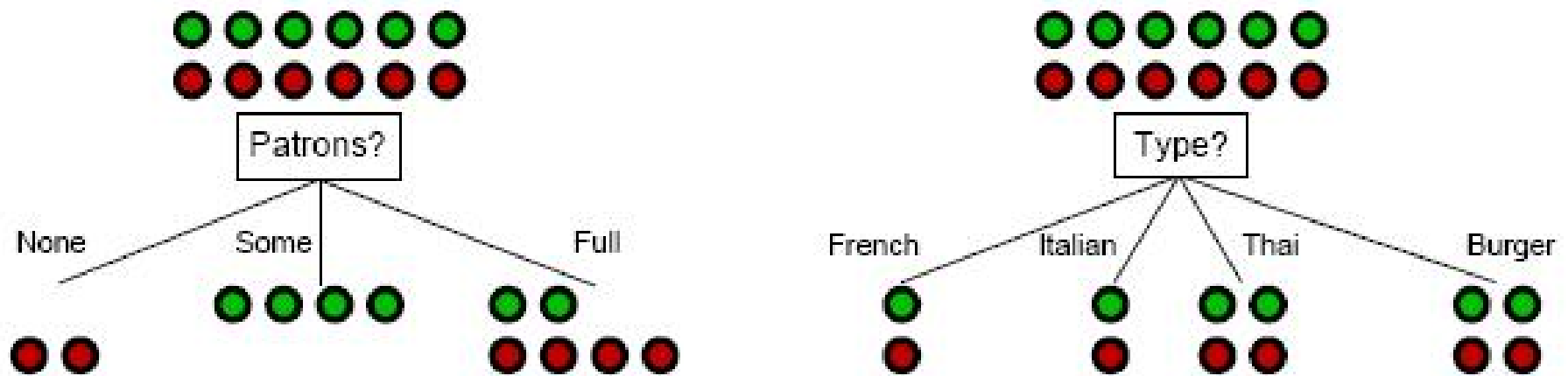
Aim: find a small tree consistent with the training examples

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 the Best Attribute

```
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
```



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

Better Attribute: Better Information

- Good early choices make for smaller trees.
- What makes an attribute good is whether it provides “information” or not.
 - = 1 if it splits perfectly
 - = 0 if it splits nothing
 - = [0,1] if it splits some.
- The more clueless I am about the answer initially, the more information contained in the answer.
- Scale: 1 bit = answer to Boolean question with prior $\langle 0.5, 0.5 \rangle$

Information

- Information in an answer with prior $\langle P_1, \dots, P_n \rangle$ is:

$$H(\langle P_1, \dots, P_n \rangle) = \sum_{i=1}^n -P_i \log_2 P_i$$

- In the 2nd edition I is used instead of H
- Also called entropy of the prior

Information (cont.)

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

$\Rightarrow H(\langle p/(p+n), n/(p+n) \rangle)$ bits needed to classify a new example

E.g., for 12 restaurant examples, $p = n = 6$ so we need 1 bit

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

$\Rightarrow H(\langle p_i/(p_i+n_i), n_i/(p_i+n_i) \rangle)$ bits needed to classify a new example

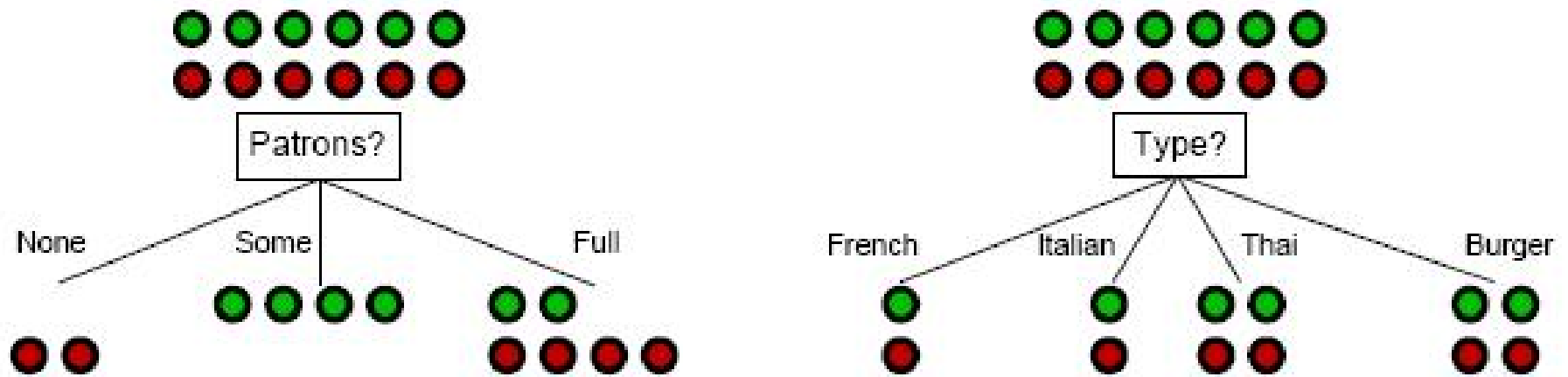
\Rightarrow **expected** number of bits per example over all branches is

$$\sum_i \frac{p_i + n_i}{p + n} H(\langle p_i/(p_i + n_i), n_i/(p_i + n_i) \rangle)$$

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

\Rightarrow choose the attribute that minimizes the remaining information needed

Information Gain Example

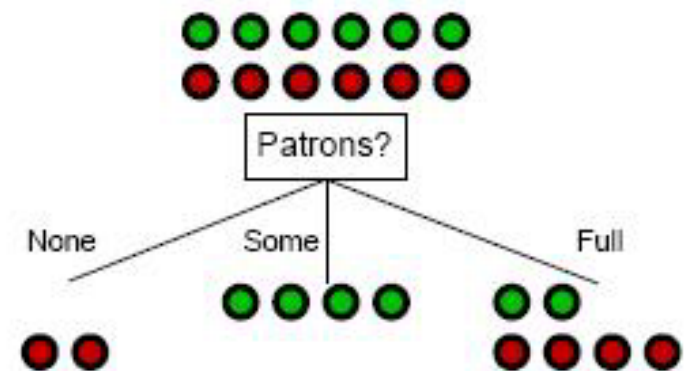


Patrons? is a better choice—gives **information** about the classification

- Information Gain = $H(p/p+n, n/p+n)$ - Remainder (expected)

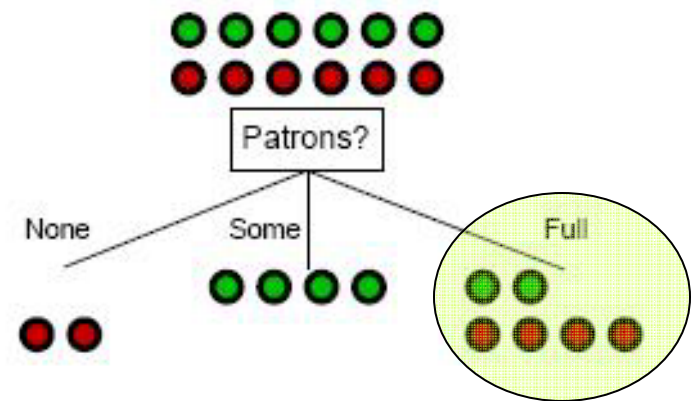
3 Special Cases – No Examples

- If there are no examples, return a default value. (You must decide it.)
- This means you've gone down a branch that hasn't been observed yet.
 - Suppose FULL was not yet observed in
 - That means that NONE and SOME appeared in all of the test cases, but not FULL.
 - We might see Full later in some real data.
 - We must return something, hence the default.



3 Special Cases – No Attributes

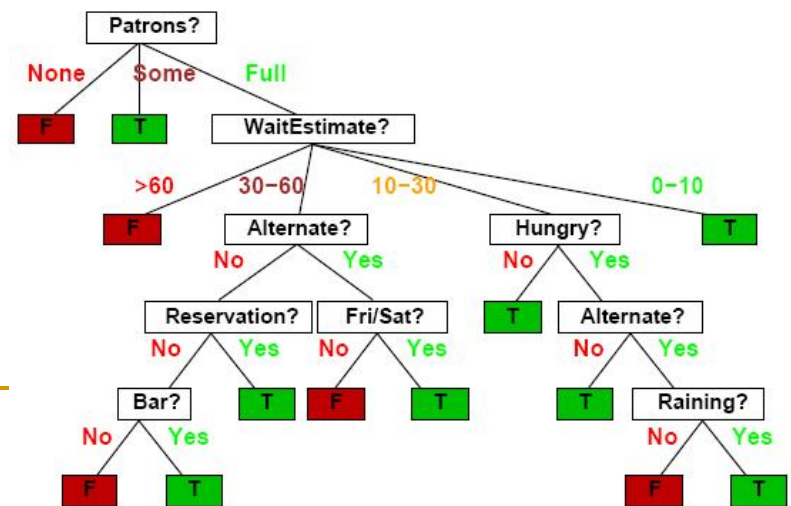
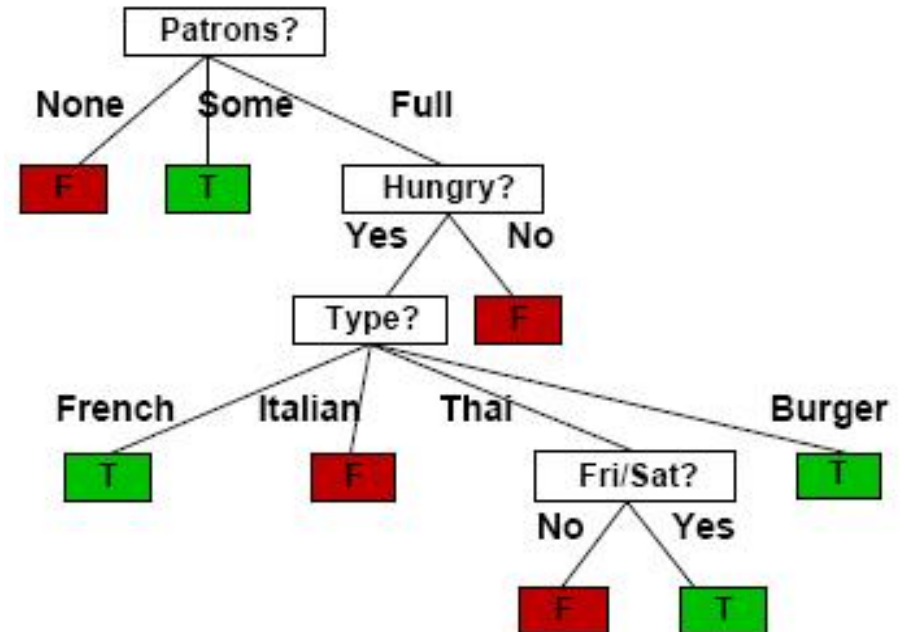
- If no attributes are left, return majority vote.
- This happens because, either:
 - Noise (bad data)
 - Need more attributes (problem under-modeled)
 - Non-deterministic choices (sometimes you said “yes” and sometimes “no” to same situation)



Patrons? is a better choice—gives **int**

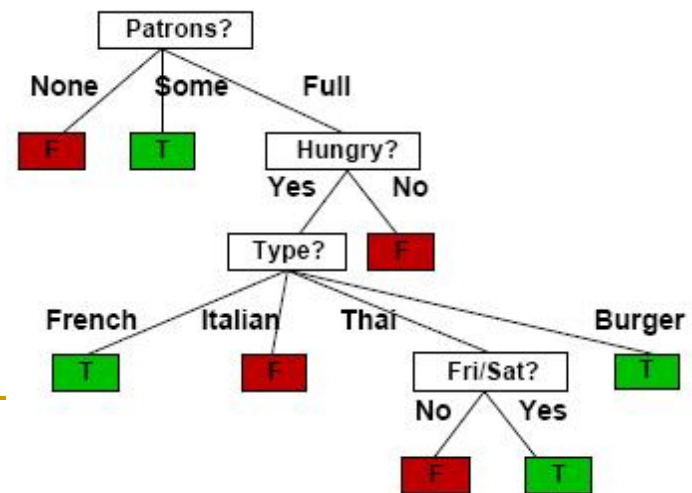
Decision Tree Learned

- Simpler than the manually made
- but is “bound” to be wrong
 - because there are so many relevant cases it hasn't seen yet (e.g. Full ^ 0-10)



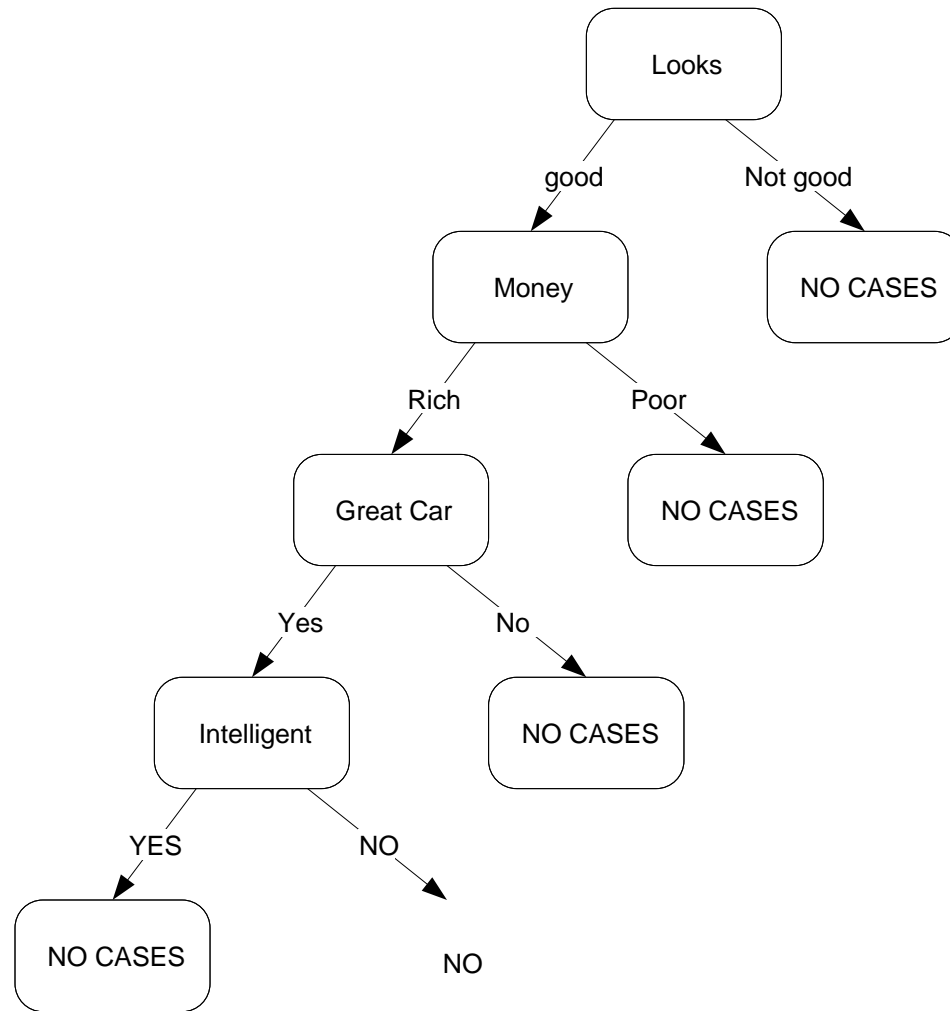
General Notes

- This is one of many trees that “fit” the data somehow.
- The generated tree may not match your intuition.
- Non-determinism may kill you, no matter what. Sometimes you wait at the restaurant, sometimes you don't. Based on what...luck, unknown attribute?
- Note that Type did not split data at top level, but did at 3rd level.



Another Problem - Gathering Data

- Consider problem of choosing a mate.
- Characteristics are: looks, sense of humor, money, car, intelligence, ...
- Idea is you meet a person and want to know if you should go out on a date?
- Young (overly discriminating) people with no experience have the following tree:



How Good is your Tree?

- Try it on new examples; see if it predicts correctly.
 - Originally you may gather a lot of data and break them into training and test sets.
- Remember the restaurant had 9200 cases.
- Problem is you want to train on $n \ll$ number of possible cases.
- Need a methodology.

Learning Methodology

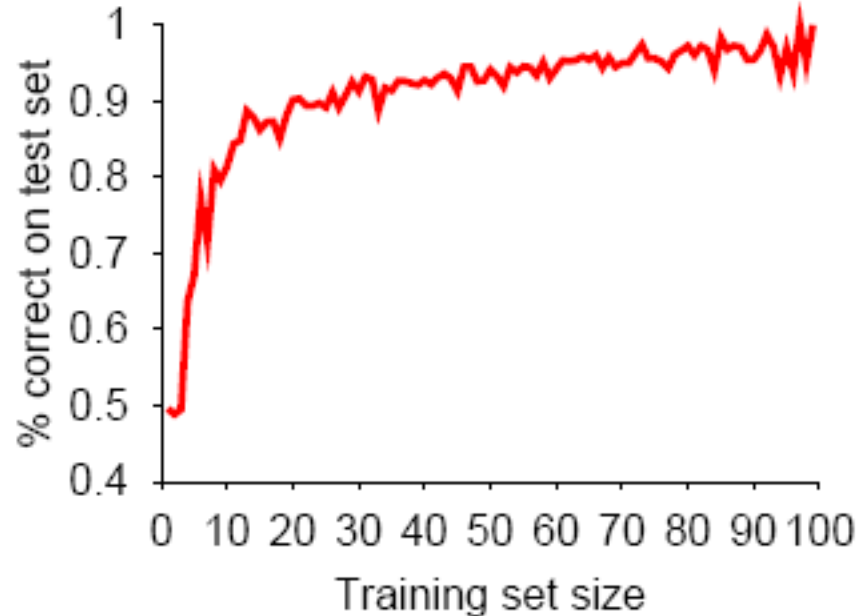
1. Collect n examples (the hard part)
2. Divide them into disjoint training and testing sets.
3. Use training set to generate the hypothesis.
4. Try the tree on the testing data.
5. Change test and training set until you get great results.

Performance Measurement

How do we know that $h \approx f$? (Hume's **Problem of Induction**)

- 1) Use theorems of computational/statistical learning theory
- 2) Try h on a new **test set** of examples
(use **same distribution over example space** as training set)

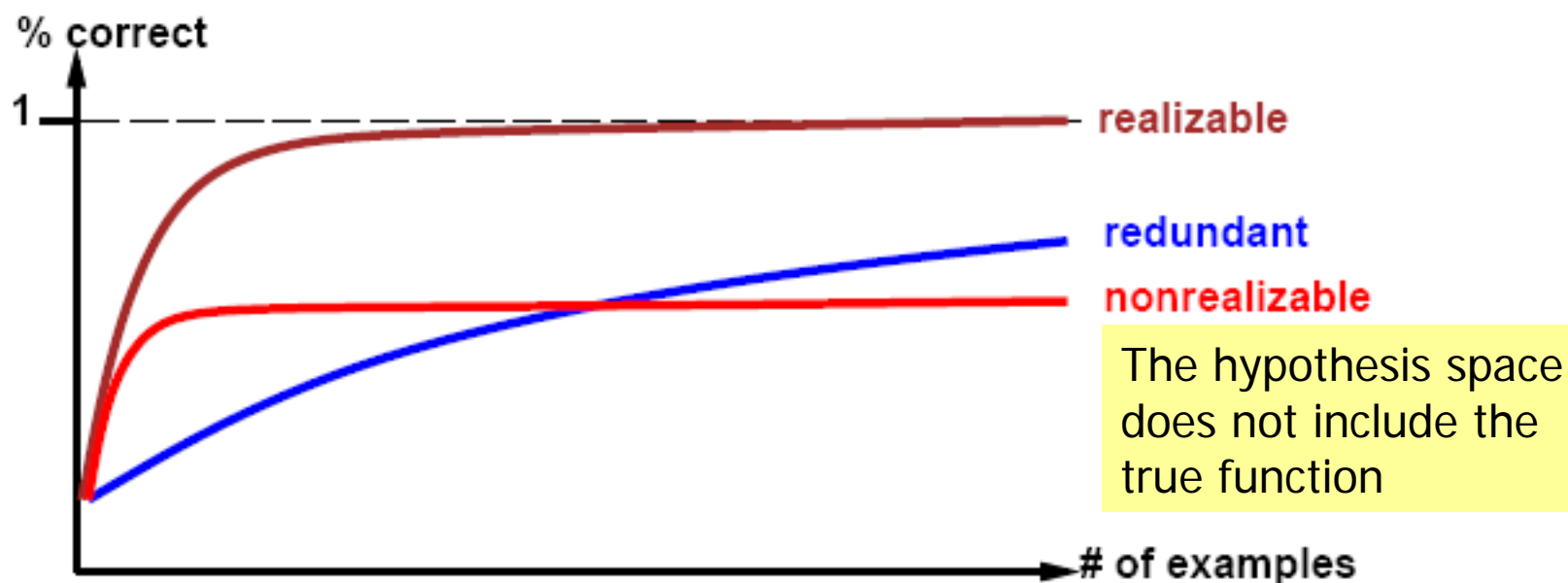
Learning curve = % correct on test set as a function of training set size



Performance Measurement (cont.)

Learning curve depends on

- **realizable** (can 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)



Overfitting - A common problem

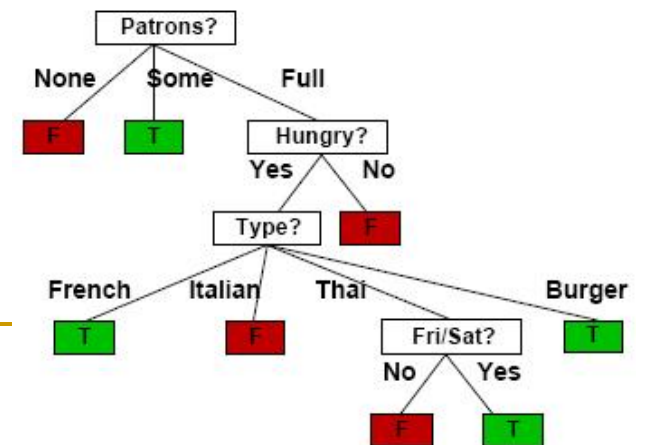
- Should you make a bet on the Dodgers?
 - You study the quality of the opponent;
 - Who is the starting pitcher;
 - Etc.
- You also notice:
 - You have won all your Saturday bets.
 - You have lost all your Sunday bets.
- Should you add DayOfTheWeek to your set of attributes, since it seems to split the data?

Pruning Irrelevant Attributes

- Superstitions are irrelevant attributes that accidentally split the data.
- It is important to prune away bad attributes.
- Do a statistical analysis, like Chi-Square pruning, to see how relevant an attribute is, i.e. do they correlate?

Cross Validation for Overfitting

- Estimate how each hypothesis predicts unseen data.
 - Run K experiments, each time setting aside 1/K of the test data.
 - Average the results (K is usually 5 or 10)
- Idea is to randomize the test data, eliminating the superstitious correlations.
- Has been done similarly in Neural Nets.
- Look at fig 18.6. Didn't use Price, Rain, etc.
 - With cross validation, we may find these to be irrelevant.



We've just scratched the surface

- Missing data – How do you handle it?
- Continuous or integer valued attributes.
- Input/Output may have an amount, not just Yes/No. (Need to do regression trees.)
- How do you combine hypotheses? Suppose you have 3. Can you use them all? Ensemble learning.
- Details about how many examples are needed.
- Etc, etc, ...

Examples

- Career Decision Tree for Psychology Students
 - <http://www.wku.edu/~sally.kuhlenschmidt/psycareer/>
- Export control Decision Tree at Stanford
 - http://www.stanford.edu/dept/DoR/exp_controls/tree/