COMP 472: Artificial Intelligence
Machine Learning
Decision Trees

- Russell & Norvig: Sections 19.3

Today

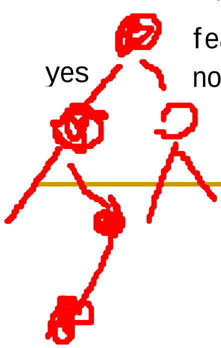
1. Introduction to ML
2. Naïve Bayes Classification
   a. Application to Spam Filtering
3. Decision Trees

4. (Evaluation
5. Unsupervised Learning)
6. Neural Networks
   a. Perceptrons
   b. Multi Layered Neural Networks

Guess Who?

一个游戏，每方想好一个人，每次轮流问一个YES OR NO的问题，排除以后关上一个窗，第一个guess出来的人获胜（用最少问题找到人的获胜）

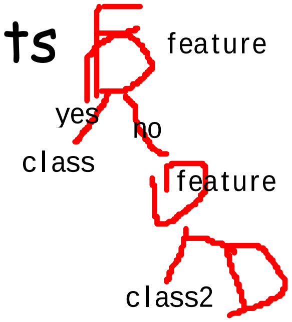
特征
feature
yes
no



这些特征组成最终答案class

Decision Trees

- Simplest, but most successful form of learning algorithm
- Very well-know algorithm is ID3 (Quinlan, 1987) and its successor C4.5



把所有的feature根据descrimating 分级（分辨力

1. Rank features based on how good they are to indicate the result
2. Put the most discriminating feature as a node (as a question) of a tree
3. Split the training examples so that those with different values for the chosen feature are in a different set
4. Repeat the same process with the nest most discriminating feature

http://www.rulequest.com/Personal/

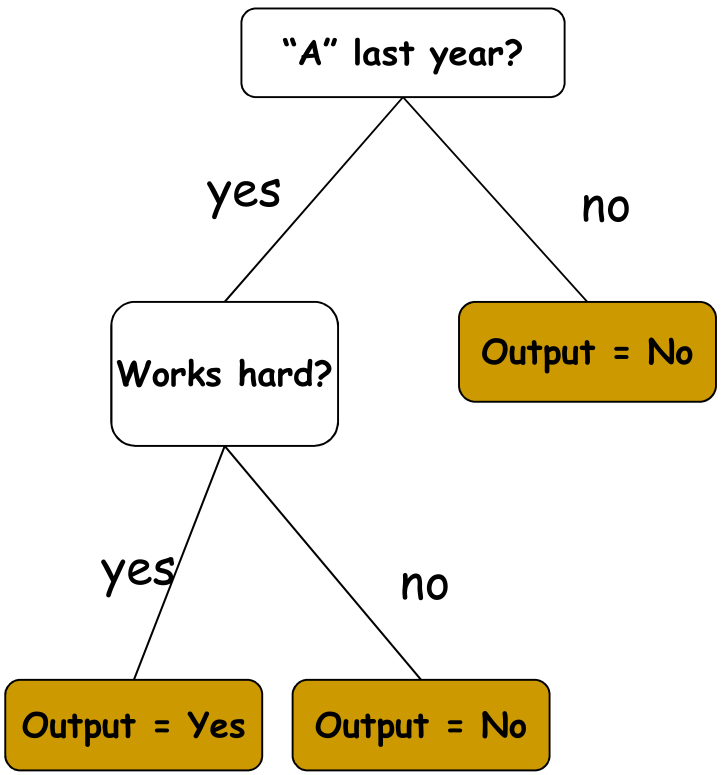
Example 1

Info on last year's students to determine if a student will get an 'A' this year

	Features (X)				Output f(X)
Student	**'A' last year?**	**Black hair?**	**Works hard?**	**Drinks?**	**'A' this year?**
X1: Richard	Yes	Yes	No	Yes	**No**
X2: Alan	Yes	Yes	Yes	No	**Yes**
X3: Alison	No	No	Yes	No	**No**
X4: Jeff	No	Yes	No	Yes	**No**
X5: Gail	Yes	No	Yes	Yes	**Yes**
X6: Simon	No	Yes	Yes	Yes	**No**

Example 1

A random decision tree that fits the dataset training set with 100% accuracy



	Features				Output f(X)
Student	‘A’ last year?	Black hair?	Works hard?	Drinks ?	**‘A’ this year?**
Richard	Yes	Yes	No	Yes	**No**
Alan	Yes	Yes	Yes	No	**Yes**
Alison	No	No	Yes	No	**No**
Jeff	No	Yes	No	Yes	**No**
Gail	Yes	No	Yes	Yes	**Yes**
Simon	No	Yes	Yes	Yes	**No**

但这里的feature是chosen at random,我们不知道是不是效率最高

是有可能有更小的DT decision tree的

Example 2: The Restaurant

- Goal: learn whether one should wait for a table
- Attributes
  1. Alternate: another suitable restaurant nearby
  2. Bar: comfortable bar for waiting
  3. Fri/Sat: true on Fridays and Saturdays
  4. Hungry: whether one is hungry
  5. Patrons: how many people are present (none, some, full)
  6. Price: price range ($, $$, $$$)
  7. Raining: raining outside
  8. Reservation: reservation made
  9. Type: kind of restaurant (French, Italian, Thai, Burger)
  10. WaitEstimate: estimated wait by host (0-10 mins, 10-30, 30-60, >60)

Example 2: The Restaurant

features

- Training data:

f(x

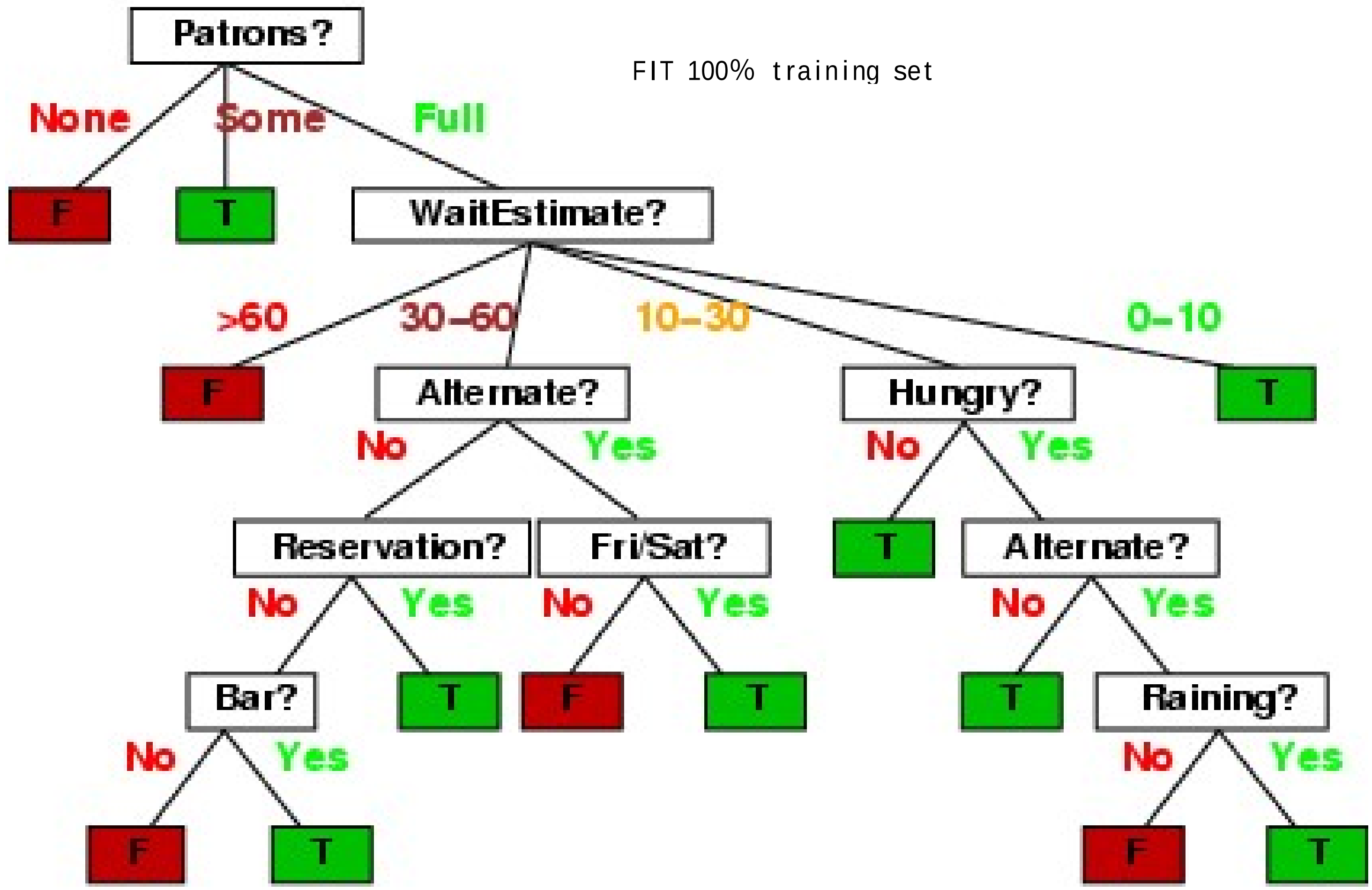
Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*Wait*
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F	T	F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

entropy=1

第一步：先不管attribute，统计f(x)的entropy，发现是1，，完全chaos
分别计算每个attribute/feature和target联合以后的entropy，how much knowing its value will reduce the entropy of class ,能减少entropy最多的，就是最具有决定性的attribute
或者说how much information have

source: Norvig (2003)

A First Decision Tree



FIT 100% training set

- But is it the best decision tree we can build?

source: Norvig (2003)

这是一个principle，由ockham发明

Ockham's Razor

It is vain to do more than can be done with less… Entities should not be multiplied beyond necessity. [Ockham, 1324]

- In other words… always favor the simplest answer that correctly fits the training data
- i.e. the smallest tree on average

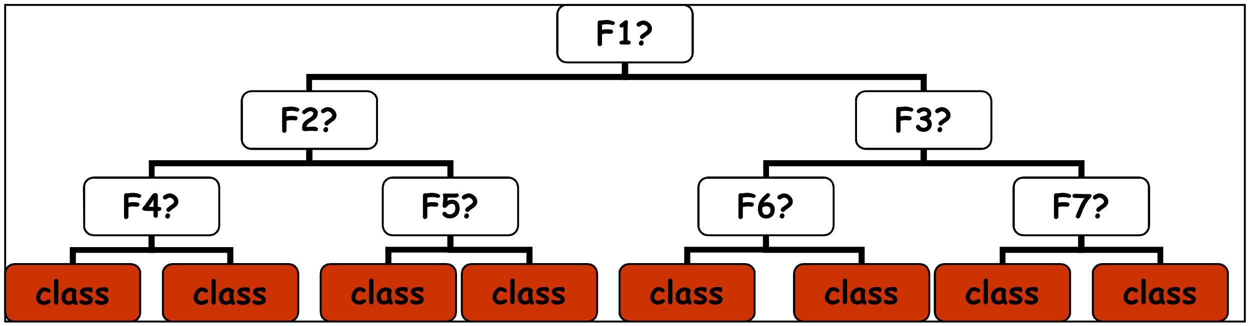
当我们有两个tree完美符合我们的training set的时候，倾向更小的那个

- This type of assumption is called inductive bias
  - inductive bias = making a choice beyond what the training instances contain

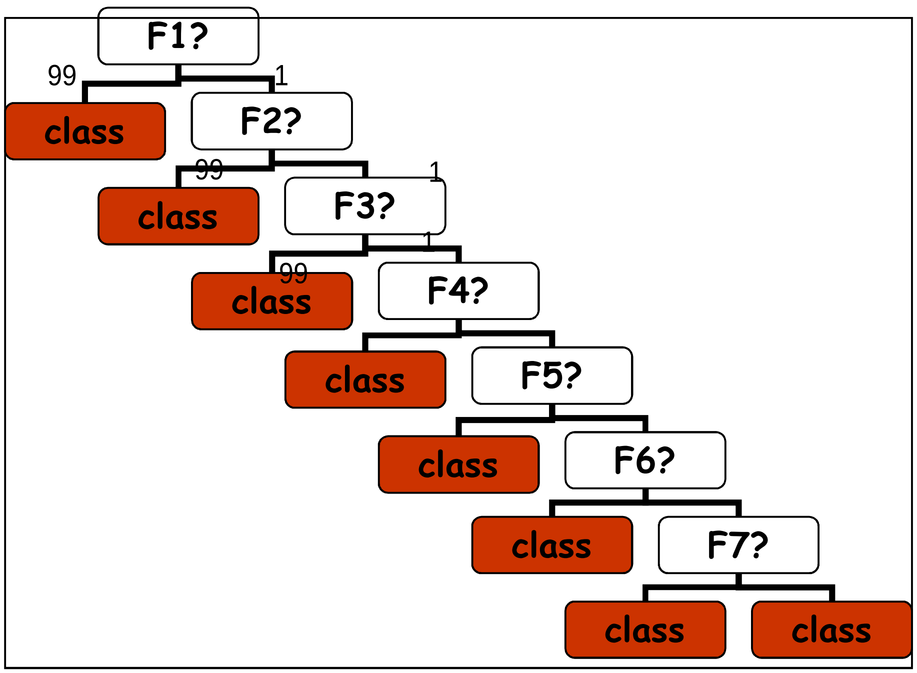
但其实这叫做inductive bias，因为两个tree都是正确的，我们做了超出training instances的内容的选择

https://www.britannica.com/topic/Occams-razor

Which Tree is Best?



这个通常来说是smallest，只要问三个问题，就有答案

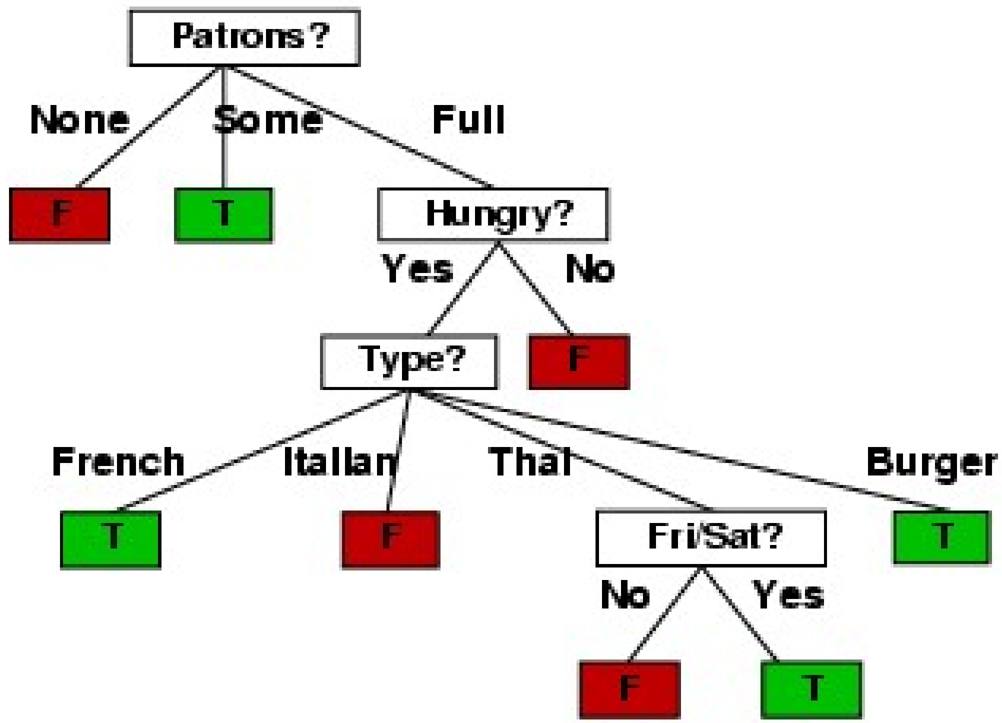


但假设所有比率都是99：1
那么这个可能更短，因为我们大概率一两步到位

depend the probability of each branch

A Better Decision Tree

- 4 tests instead of 9
- 11 branches instead of 21



source: Norvig (2003)

Choosing the Next Feature

- The key problem is choosing which feature to split a given set of examples
- Different measures have been proposed
- ID3 uses Maximum Information-Gain
  - i.e. we choose the feature that has the largest information gain
  - we expect this feature to result in the smallest tree on average we expect but cannot guarantee确保
  - based on information theory

Essential Information Theory

- Developed by Shannon in the 40s
- Shannon developed the notion of entropy (aka information content) of a random variable (RV)
  概念 熵 测量一个random variable有多informative

  or how much information the random variable can take
- Entropy measures how "predictable" a RV is
  - If you already have a good idea about the answer (e.g. 90/10 split)
    - → low entropy //sure thing， 例如扔一个硬币，90正面，10反面，low entropy
  - If you have no idea about the answer (e.g. 50/50 split)
    - → high entropy //total chaos

Dartmouth Conference: The Founding Fathers of AI

Entropy

离散的 random variable

- Let X be a discrete RV with *i* possible outcomes x_i
- Entropy (or information content) of X

例如字母表，有26个outcome
例如字典，有10000个outcome

$$H(X) = -\sum_{i=1}^{n} p(x_i)\log_2 p(x_i)$$

p就是posibility

这里必须是base 2

- measures:
  - the *amount of information* in a RV
  - *average uncertainty* of a RV
  - *average length of a message* needed to transmit an outcome x_i of that RV when encoded optimally over a binary channel
- measured in bits

entropy计量单位是bit

?

Entropy of a Coin Toss

$$H(X) = -\sum_{x_i \in X} p(x_i)\log_2 p(x_i)$$

Entropy (or information content)

$$H(\text{fair coin toss}) = -\sum_{x_i \in X} p(x_i)\log_2 p(x_i) = H\left(\frac{1}{2}, \frac{1}{2}\right)$$

$$= -\left(\frac{1}{2}\log_2\frac{1}{2} + \frac{1}{2}\log_2\frac{1}{2}\right) = 1\text{ bit}$$

entropy of a fair coin toss (the RV) with 2 possible outcomes, each with a probability of 1/2

这个就很简单，正反面几率是0.5，相加=1，单位是bit

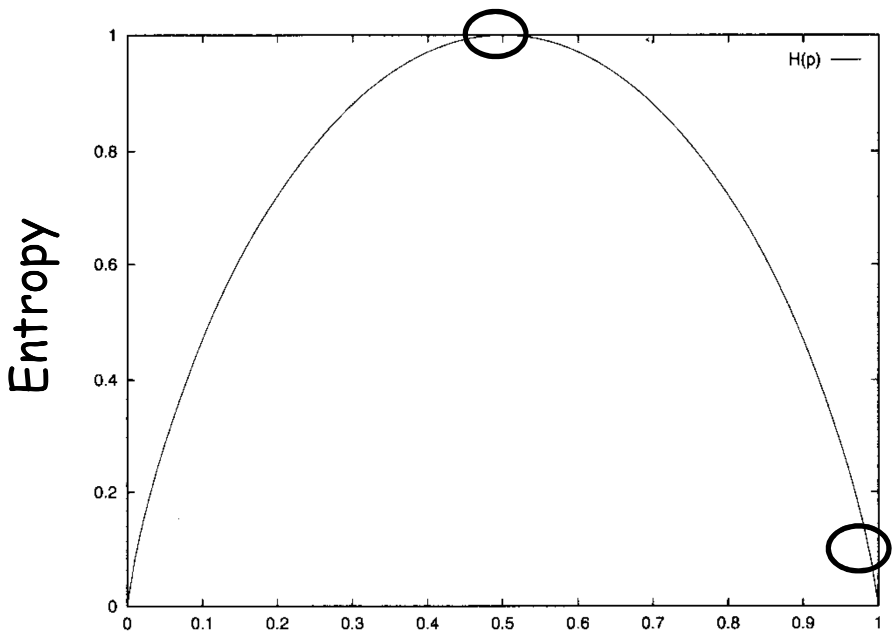
a RV with only 2 outcomes x_1 and x_2 will have $1 \geq H(X) \geq 0$

total chaos sure thing

Example: The Coin Toss

- Fair coin: $H(X) = -\sum_{i=1}^{n} p(x_i)\log_2 p(x_i) = -\left(\frac{1}{2}\log_2\frac{1}{2} + \frac{1}{2}\log_2\frac{1}{2}\right) = 1\ \text{bit}$

不正当的
- Rigged coin: $H(X) = -\sum_{i=1}^{n} p(x_i)\log_2 p(x_i) = -\left(\frac{99}{100}\log_2\frac{99}{100} + \frac{1}{100}\log_2\frac{1}{100}\right) = 0.08\ \text{bits}$



fair coin -> high entropy

rigged coin -> low entropy

P(head)

Information Gain

- information gain 知道一段信息后测量减少entropy的方法
  - measure the entropy reduction of a RV, once a piece of information is known
  - used to measure the "discriminating power" of an attribute A given a data set S 用来测量一个attribute的descriminating power

  - Let Values(A) = the set of values that attribute A can take
  - Let S_v = the set of examples in the data set which have value v for attribute A (for each value v from Values(A))

第八页

没统计过attribute的各种class的entropy

information gain (or entropy reduction)

$$gain(S, A) = H(S) - H(S|A)$$ 在attributeA下的entropy

$$= H(S) - \sum_{v \in values(A)} \frac{|S_v|}{|S|} \times H(S_v)$$

Some Intuition

Size	Color	Shape	Output
Big	Red	Circle	+
Small	Red	Circle	+
Small	Red	Square	-
Big	Blue	Circle	-

- *Size* is the least discriminating attribute (i.e. smallest information gain)

- *Shape* and *color* are the most discriminating attributes (i.e. highest information gain)

A Small Example (1)

Size	Color	Shape	Output
Big	Red	Circle	+
Small	Red	Circle	+
Small	Red	Square	-
Big	Blue	Circle	-

Values(Color) = {red,blue}

Color

red: 2+ 1-　　blue: 0+ 1-

$$gain(S, Color) = H(S) - \sum_{v \in values(Color)} \frac{|S_v|}{|S|} x\, H(S_v)$$

$$H(S) = -\left(\frac{2}{4}\log_2\frac{2}{4} + \frac{2}{4}\log_2\frac{2}{4}\right) = 1$$

这个就不解释了

for each v of Values(Color)

$$H(S\,|\,Color = red) = H\left(\frac{2}{3}, \frac{1}{3}\right) = -\left(\frac{2}{3}\log_2\frac{2}{3} + \frac{1}{3}\log_2\frac{1}{3}\right) = 0.918$$

$$H(S\,|\,Color = blue) = H(1,0) = -\left(\frac{1}{1}\log_2\frac{1}{1}\right) = 0$$

$$H(S\,|\,Color) = \frac{3}{4}(0.918) + \frac{1}{4}(0) = 0.6885$$

$$gain(Color) = H(S) - H(S\,|\,Color) = 1 - 0.6885 = 0.3115$$

例如我们要看color,分两步，
第一步按color divide，分别统计output
结果是red下两个正的一个负的，统计H熵
blue下一个负的，熵为0

第二步根据color各自比重，乘以他们的对应熵，
得到了加入color之后的entropy熵

相减得到gain

A Small Example (2)

Size	Color	Shape	Output
Big	Red	Circle	+
Small	Red	Circle	+
Small	Red	Square	-
Big	Blue	Circle	-

Shape

circle: 2+ 1-

square: 0+ 1-

Note: by definition,

- Log 0 = -∞
- 0log0 is 0

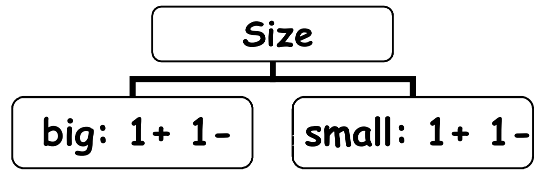
$$H(S) = -\left(\frac{2}{4}\log_2\frac{2}{4} + \frac{2}{4}\log_2\frac{2}{4}\right) = 1$$

$$H(S\,|\,Shape) = \frac{3}{4}(0.918) + \frac{1}{4}(0) = 0.6885$$

$$gain(Shape) = H(S) - H(S\,|\,Shape) = 1 - 0.6885 = 0.3115$$

A Small Example (3)

Size	Color	Shape	Output
Big	Red	Circle	+
Small	Red	Circle	+
Small	Red	Square	-
Big	Blue	Circle	-



$$H(S) = -\left(\frac{2}{4}\log_2\frac{2}{4} + \frac{2}{4}\log_2\frac{2}{4}\right) = 1$$

$$H(S\,|\,Size) = \frac{1}{2}(1) + \frac{1}{2}(1) = 1$$

$$gain(Size) = H(S) - H(S\,|\,Size) = 1 - 1 = 0$$

A Small Example (4)

Size	Color	Shape	Output
Big	Red	Circle	+
Small	Red	Circle	+
Small	Red	Square	-
Big	Blue	Circle	-

$$\text{gain}(\text{Shape}) = 0.3115$$
$$\text{gain}(\text{Color}) = 0.3115$$
$$\text{gain}(\text{Size}) = 0$$

所以ssize是最没用的attribute,我们建立树的时候优先shape color，最后size

- So first separate according to either *color* or *shape* (root of the tree)

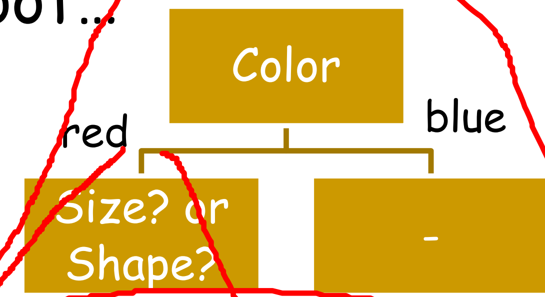
A Small Example (4)

- Let's assume we pick *Color* for the root…

构建完整的树，我们用4instance

Size	Color	Shape	Output
Big	Red	Circle	+
Small	Red	Circle	+
Small	Red	Square	-
Big	Blue	Circle	-

S_2 (first three rows)



$$H(S_2) = -\left(\frac{2}{3}\log_2\frac{2}{3} + \frac{1}{3}\log_2\frac{1}{3}\right)$$

子树，blue排除了，所以我们只用3instancce,所以我们要重新计算一遍

for each v of Values(Size)

$$H(S_2 \mid Size = big) = H\left(\frac{1}{1}, \frac{0}{1}\right) = 0$$

$$H(S_2 \mid Size = small) = H\left(\frac{1}{2}, \frac{1}{2}\right) = 1$$

$$H(S_2 \mid Size) = \frac{1}{3}(0) + \frac{2}{3}(1)$$

$$gain(Size) = H(S_2) - H(S_2 \mid Size)$$

for each v of Values(Shape)

$$H(S_2 \mid Shape = circle) = H\left(\frac{2}{2}, \frac{0}{2}\right) = 0$$

$$H(S_2 \mid Shape = square) = H\left(\frac{0}{1}, \frac{1}{1}\right) = 0$$

$$H(S_2 \mid Shape)$$

$$gain(Shape) = H(S_2) - H(S_2 \mid Shape)$$

Back to the Restaurant

- Training data:

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

source: Norvig (2003)

The Restaurant Example

$gain(alt) = \ldots$ $gain(bar) = \ldots$ $gain(fri) = \ldots$ $gain(hun) = \ldots$

$$gain(pat) = 1 - \left(\frac{2}{12} \times H\left(\frac{0}{2}, \frac{2}{2}\right) + \frac{4}{12} \times H\left(\frac{0}{4}, \frac{4}{4}\right) + \frac{6}{12} \times H\left(\frac{2}{6}, \frac{4}{6}\right) \right)$$

$$= 1 - \left(\frac{2}{12} \times -\left(\frac{0}{2}\log_2\frac{0}{2} + \frac{2}{2}\log_2\frac{2}{2}\right) + \frac{4}{12} \times -\left(\frac{0}{4}\log_2\frac{0}{4} + \frac{4}{4}\log_2\frac{4}{4}\right) + \ldots \right) \approx 0.541 \text{bits}$$

$gain(price) = \ldots$ $gain(rain) = \ldots$ $gain(res) = \ldots$

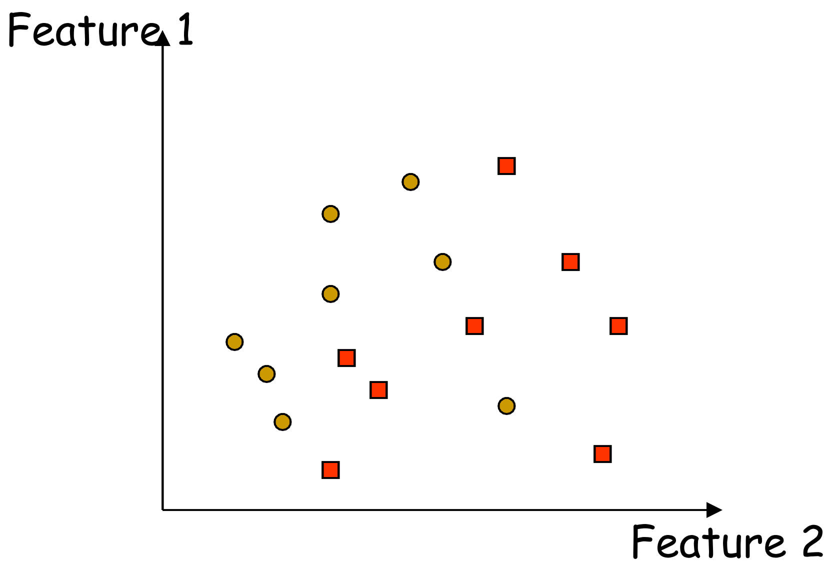
$$gain(type) = 1 - \left(\frac{2}{12} \times H\left(\frac{1}{2}, \frac{1}{2}\right) + \frac{2}{12} \times H\left(\frac{1}{2}, \frac{1}{2}\right) + \frac{4}{12} \times H\left(\frac{2}{4}, \frac{2}{4}\right) + \frac{4}{12} \times H\left(\frac{2}{4}, \frac{2}{4}\right) \right) = 0 \text{ bits}$$

$gain(est) = \ldots$

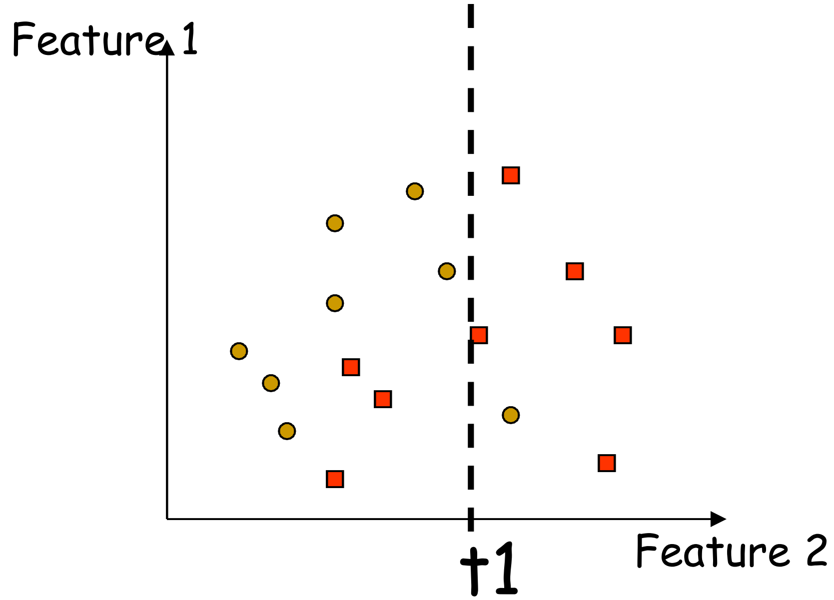
- Attribute pat (Patron) has the highest gain, so root of the tree should be attribute *Patrons*
- do recursively for subtrees

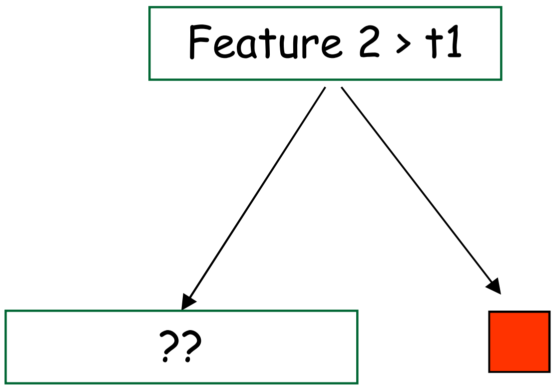
别忘了 do recursively for subtree

Decision Boundaries of Decision Trees

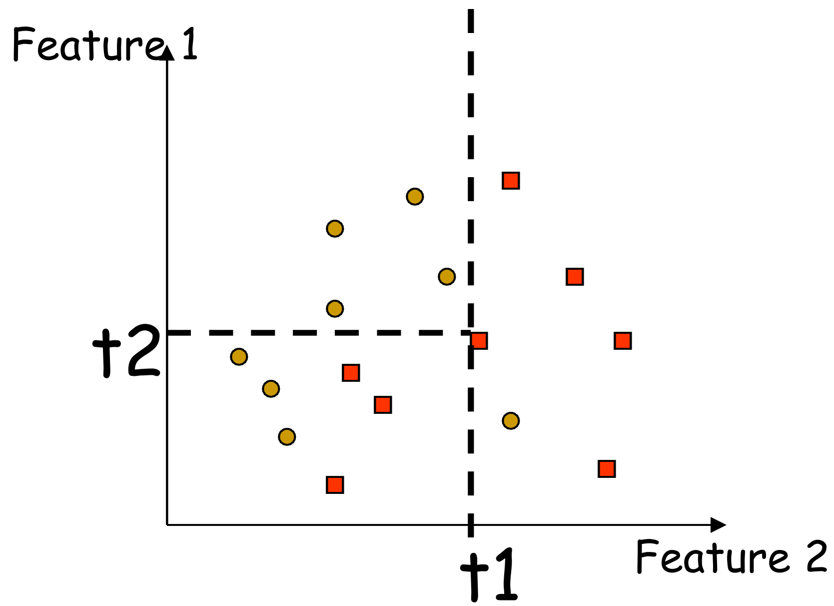


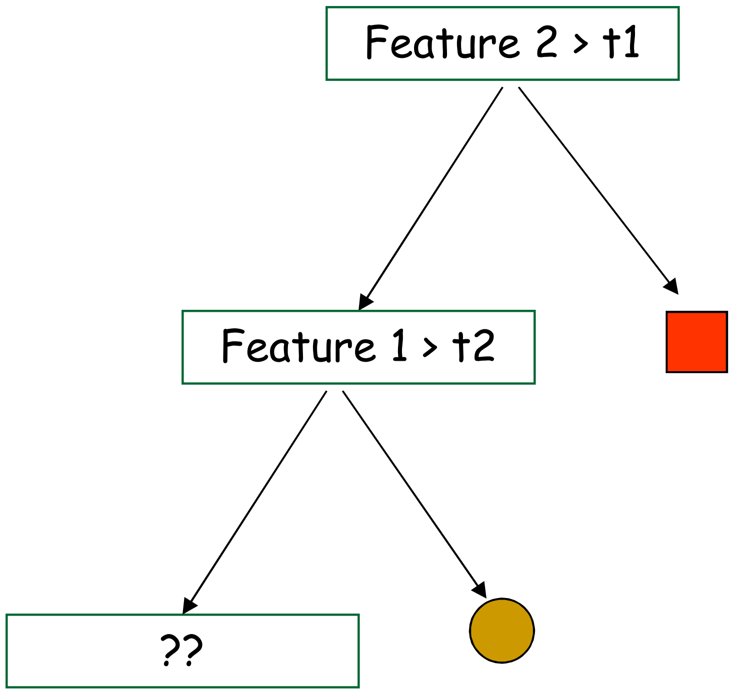
Decision Boundaries of Decision Trees



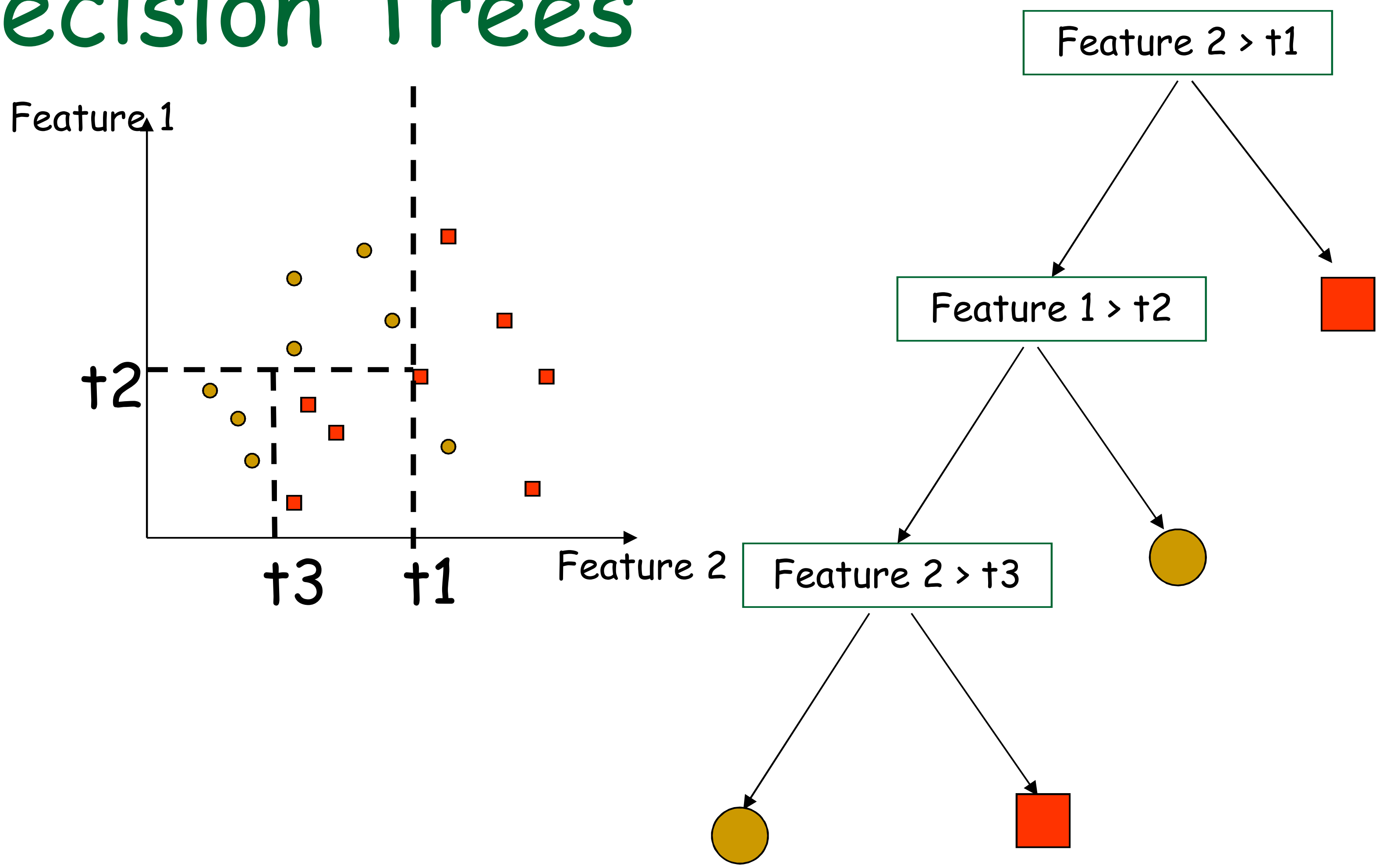


Decision Boundaries of Decision Trees





Decision Boundaries of Decision Trees



Applications of Decision Trees

- One of the most widely used learning methods in practice
  - Fast
  - Simple
  - Traceable (<-- very important!)

Today

1. Introduction to ML
2. Naïve Bayes Classification
   a. Application to Spam Filtering
3. Decision Trees
4. (Evaluation
5. Unsupervised Learning)
6. Neural Networks
   a. Perceptrons
   b. Multi Layered Neural Networks

Up Next

1. Introduction to ML
2. Naïve Bayes Classification
   a. Application to Spam Filtering
3. Decision Trees
4. (Evaluation
5. Unsupervised Learning)
6. Neural Networks
   a. Perceptrons
   b. Multi Layered Neural Networks