Decision Trees

15-381 lecture notes, 14 March 2000

David Cohn <www.cs.cmu.edu/~cohn>

Chapter 18 of Russell & Norvig

After this lecture, you should understand:

- what a decision tree is
- limits on decision tree representational power
- what Ockham's Razor has to do with it all
- how to build a decision tree
- several splitting criteria

Discriminative classification

- If domain has enough dimensions, can frequently find hyperplane which separates data
  - warning, in general, high dimensionality is a VERY BAD THING - requires huge amount of data
- In low dimensions, data may not be linearly separable

- [picture here]

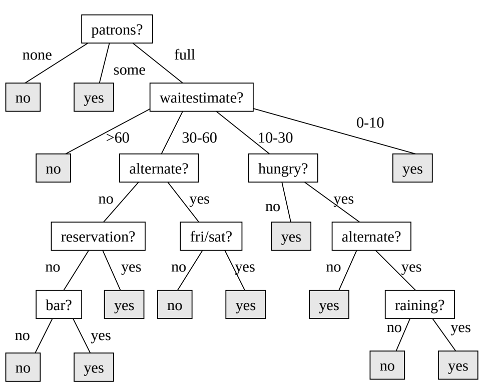
- Data may be categorical - not easily mapped to geometric space

- [picture here]

- What to do? Build a decision tree!

Decision trees - "Being Stuart Russell"

- You have found an interdimensional portal which drops you into the mind of Stuart Russell as he sets out to have dinner.
- He spots a restaurant, goes in and makes a decision whether or not to eat there, based on the following decision tree:



Decision trees

- Observables:
  - alternate (yes/no)
  - bar (yes/no)
  - fri/sat (yes/no)
  - hungry (yes/no)
  - patrons (none/some/full)
  - price ($,$$,$$$)
  - raining (yes/no)
  - reservation (yes/no)
  - type (french/italian/thai/burger)
  - waitestimate (0-10/10-30/30-60/>60)

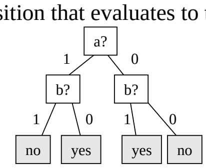
- Predict or determine: eat_here (yes/no)

- Some things to notice
  - not all observables appear in the decision tree!
  - some variables are continuous, some are categorical
  - leaf can take arbitrary values (not necessarily yes/no)

How expressive are decision trees?

- Given a proposition (a → b), can we build a decision tree that returns true if and only iff the proposition is true?

- Given a decision tree where all nodes are binary variables, can we build a proposition that evaluates to the same values of the leaves?



Decision trees in practice

- Bioinformatics (medical decision making)
  - e.g. predict risk of emergency c-section
- Credit card industry (fraud detection)
- Astronomy (classifying cosmic ray hits)
- DNA sequence analysis
- Data mining

- Given enough decision nodes, we can build a decision tree to represent *almost* anything we want
  - (aside: what sort of things *can't* we represent with the decision trees described so far?)

- So, how do we build a decision tree?
  - ask an expert how they make their decisions
  - *learn* one from data!

	attribute									
example	**alt**	**bar**	**fri**	**hun**	**pat**	**price**	**rain**	**res**	**type**	**est**
x1	y	n	n	y	some	$$$	n	y	french	0-10
x2	y	n	n	y	full	$	n	n	thai	30-60
x3	n	y	n	n	some	$	n	n	burger	0-10
x4	y	n	y	y	full	$	n	n	thai	10-30
x5	y	n	y	n	full	$$$	n	y	french	>60
x6	n	y	n	y	some	$$	y	y	italian	0-10
x7	n	y	n	n	none	$	y	n	burger	0-10
x8	n	n	n	y	some	$$	y	y	thai	0-10
x9	n	y	y	n	full	$	y	n	burger	>60
x10	y	y	y	y	full	$$$	n	y	italian	10-30
x11	n	n	n	n	none	$	n	n	thai	0-10
x12	y	y	y	y	full	$	n	n	burger	30-60

Building a decision tree

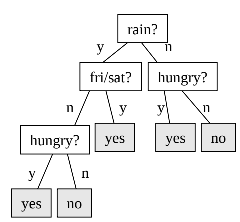
- You have found *another* interdimensional portal which drops you into the mind of Peter Norvig as he sets out to have dinner.
- He spots a restaurant, goes in and makes a decision, apparently at random, whether or not to eat there
- After spending several dinners in Peter's mind, we want to try to find some sort pattern - to predict whether or not he'll stick with this restaurant or go to another

	attribute										
example	**alt**	**bar**	**fri**	**hun**	**pat**	**price**	**rain**	**res**	**type**	**est**	**eat**
x1	y	n	n	y	some	$$$	n	y	french	0-10	**y**
x2	y	n	n	y	full	$	n	n	thai	30-60	**n**
x3	n	y	n	n	some	$	n	n	burger	0-10	**y**
x4	y	n	y	y	full	$	n	n	thai	10-30	**y**
x5	y	n	y	n	full	$$$	n	y	french	>60	**n**
x6	n	y	n	y	some	$$	y	y	italian	0-10	**y**
x7	n	y	n	n	none	$	y	n	burger	0-10	**n**
x8	n	n	n	y	some	$$	y	y	thai	0-10	**y**
x9	n	y	y	n	full	$	y	n	burger	>60	**n**
x10	y	y	y	y	full	$$$	n	y	italian	10-30	**n**
x11	n	n	n	n	none	$	n	n	thai	0-10	**n**
x12	y	y	y	y	full	$	n	n	burger	30-60	**y**

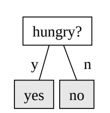
- Can you build a decision tree consistent with this data?
- Can you *always* build a decision tree consistent with any observed data?

Building a decision tree - Ockham's Razor

- Can you think of an procedure for automatically generating decision trees?
  - can you think of several?
- How many different decision trees can you generate for the restaurant data?
- Are some of these trees better than others?



rain	**hungry**	**fri/sat**	**eat**
y	y	y	**y**
n	y	y	**y**
y	n	n	**n**
n	n	y	**n**
y	y	n	**y**



"Non sunt multiplicanda entia praeter necessitatem."

("Do not multiply entities beyond necessity")

-William of Ockham (1285-1347)

- Translated to scientific practice: the best hypothesis is the simplest one that explains the data

Building decision trees

- How to find smallest (simplest) one that matches the data?
- The bad news: it's intractable
- The other bad news: not always a good idea, anyway
  - especially if you have noisy or stochastic data

Generally accepted good-thing-to-do:

- Heuristically build small tree in a greedy top-down way:
  - find "best" variable to split on
  - partition data to children
  - recurse on each child

- But - what do you mean by "best"?
  - Max minority
  - Sum minority
  - Information gain ⇐ most popular
  - Information ratio
  - Gini index...

Splitting criteria

- Making the first split
  - split on **rain**
    - *yes*: yyy, *no*: ynn
  - split on **alternate**
    - *yes*: yn, *no*: yyyn
  - split on **fri/sat**
    - *yes*: yyn, *no*:yyn

rain	alt	fri/sat	eat
y	n	y	**y**
n	n	y	**y**
n	n	n	**n**
n	y	y	**n**
y	n	n	**y**
y	y	n	**y**

A couple of simple criteria:

- Max minority: "minimize worst case child ambiguity"

- Sum minority: "minimize average child ambiguity"

Splitting criteria - information gain

- Introducing *entropy* - a measure of how diverse a set is

$$Entropy(S) = -\sum_{v \in S} p(v) \log_2 p(v)$$

- Entropy is always greater than or equal to zero (zero if and only if set is completely uniform)

- Information gain of attribute *A* is expected reduction in entropy of *S* if we know attribute *A*

$$Gain(S, A) = Entropy(S) - \sum_{a \in A} \frac{|S_a|}{|S|} Entropy(S_a)$$

- Making the first split
  - split on **rain**
    - *yes*: yyy, *no*: ynn
  - split on **alternate**
    - *yes*: yn, *no*: yyyn
  - split on **fri/sat**
    - *yes*: yyn, *no*:yyn

rain	alt	fri/sat	eat
y	n	y	**y**
n	n	y	**y**
n	n	n	**n**
n	y	y	**n**
y	n	n	**y**
y	y	n	**y**

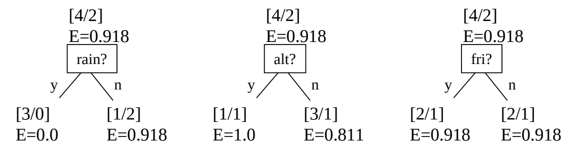
Splitting criteria - information gain

- Building the decision tree with information gain

$$Gain(S, A) = Entropy(S) - \sum_{a \in A} \frac{|S_a|}{|S|} Entropy(S_a)$$

- Making the first split
  - split on **rain**
    - *yes*: yyy, *no*: ynn
  - split on **alternate**
    - *yes*: yn, *no*: yyyn
  - split on **fri/sat**
    - *yes*: yyn, *no*:yyn

rain	alt	fri/sat	eat
y	n	y	**y**
n	n	y	**y**
n	n	n	**n**
n	y	y	**n**
y	n	n	**y**
y	y	n	**y**



0.918 -3/6*0 -3/6*0.918
= 0.459

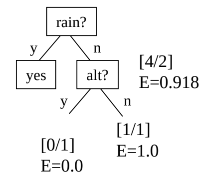
0.918 -1/2*1 -4/6*0.811
= 0.377

0.918 -3/6*0.918 -3/6*0.918
= 0.0

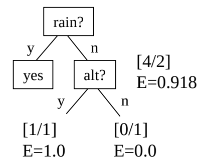
Splitting criteria - information gain

- Making the next split
- Given that it's not raining
  - split on **alternate**
    - *yes*: n, *no*: yn
  - split on **fri/sat**
    - *yes*: yn, *no*:n

rain	alt	fri/sat	eat
y	n	y	**y**
n	n	y	**y**
n	n	n	**n**
n	y	y	**n**
y	n	n	**y**
y	y	n	**y**

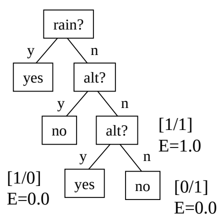


0.918 -1/3*0 -2/3*1.0
= 0.251



0.918 -2/3*1 -1/3*0.0
= 0.251

- No clear winner - pick one at random (or according to a secondary criterion) and continue



rain	alt	fri/sat	eat
y	n	y	**y**
n	n	y	**y**
n	n	n	**n**
n	y	y	**n**
y	n	n	**y**
y	y	n	**y**

What if an attribute has many values?

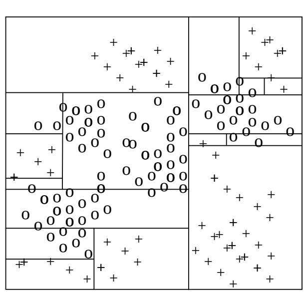
- What is information gain if we split on 'date': jan 1, jan 2,... dec 31
  - what does our information gain look like/
- Is this what we want?
  - how useful would this tree be for predicting the future?
- An alternative: Gain Ratio

$$GainRatio(S,A) = \frac{Gain(S,A)}{SplitInformation(S,A)}$$

$$SplitInformation(S,A) = -\sum_{a\in A} \frac{|S_a|}{|S|} \log_2 \frac{|S_a|}{|S|}$$

What about continuous values on inputs?

- Can find single splits continuous spaces:
  rain-per-minute < 0.43
  rain-per-minute ≥ 0.43
- How to find best split in a continuous space?
  - an optimization problem (we'll cover those soon!)
  - an easy method (but slow)
    - move split up over one point at a time, choose best



- Note - in continuous spaces, one frequently wants to split on the same dimension multiple times
- Even more esoteric - oblique splits!

When do we stop?

- Details coming up in next lecture - for now, take for granted that we frequently want to stop growing our tree before every leaf has zero entropy*
  (*almost correct - might grow exhaustively, then prune back)
- We find ourselves at a leaf node with values [y y y n n]
  - what should we return?
  - 'yes'?
  - how about 'yes' 3/5 of the time, and 'no' 2/5 of the time?

What about continuous values on outputs?

- Who says our output values have to be binary? Or even have a fixed number of values?
- What happens if output value is continuous?
  - in this case, we have a *regression* tree
- How do we have to change our splitting criterion?
- When we get to a leaf with multiple values, what value should we return?

Unknown attribute values

- Sometimes, we have missing values on some of our data
- We could
  - throw away those observations
  - try to fill in some default value
    - assign most common value in training set
    - assign most common value in node
    - assign probability to each possible value
- The lurking, unanswered question(s):
  - *How can I have any confidence about how my decision tree will perform on new data?*
  - *Do I always want to grow my decision trees out to the end?*
  - *What's all this about "overfitting"?*
- Stay tuned for the next gripping episode of "Introduction to Artificial Intelligence":

  Overfitting and Model Selection

Some practical decision tree names to be aware of

- ID3, by Ross Quinlan
  - the original "machine learning" decision tree algorithm
- CART - "Classification and Regression Trees," by Brieman, Friedman, Olshen and Stone
  - canonical look at decision trees, "done right" by statisticians
- C4.5, also by Ross Quinlan
  - souped-up decision tree system package
- OC1, by Sreerama Murthy, Simon Kasif and Steven Salzberg
  - free decision tree package, available at

    <http://www.tigr.org/~salzberg/announce-oc1.html>