Goals

- What is a Decision Tree
- How are Decisions tree used
- How are DTs constructed
- How to generate Expert Systems from DTs
- Entropy
- How do you know they work?

Background

Part of learning is simply remembering.
Computers are very good at that.

Another aspect of learning is generalizing from the observed to the unobserved.

This form of learning is inherently and unavoidably an errorful process.

After hearing 2,4,6,8... any number might come next.

Theoretically there is no best learning algorithm.

All data may be generated by noise.

Modeling the data too closely is called overfitting.
You may be modeling the noise.

Modeling the data too loosely is called underfitting.
You may be modeling too little of the data.

A general, apparently unsolvable problem, is how much of the data to believe.

Basic Components of Learning Program

Performance Element: program that uses the learned knowledge

Learning Element: program that produces a change in performance

Critic: evaluation of performance

Problem Generator In real situations, the world. More usually, a set of examples.

Types of Learning

Passive Learner Learn from observations; do not interact with the world.

Active Learner Conducts experiments or determine own experiences.

SpeedUp Learning Getting faster at any task, such as typing, coding, etc.

Classfication Learning Getter better at making decisions between a small number of classes.

Supervised Learning Here you are given a set of labelled examples, e.g. descriptions of patients with and without some disease. The goal is to predict accurately the disease status of a new patient.

Batch learning occurs if you have all the data available at once. This happens when you analysis a data base.

Incremental learning occurs if you are presented one instance at a time. This typically occurs in control problems or reactive situations, such as robotic control.

An important research issue is how to include prior knowledge in learning.

Inductive Learning

Input: Classified Examples
{glucose level=low, temp=97.5, ...} class= sick
Output: Decision Procedure
assigns a class label to any example

Example decision procedures:

- Expert Systems
- Decision Trees
- Nearest Neighbor
- Single Neuron
- Neural Net

Example: C4.5

C4.5 is the most widely used and generally effective learning algorithm.
It generates Decision Trees or Rules.

It is effective on training set size of thousands of examples.

The input files: *.names tells you what the classes and attributes are.

```
Play, Don't Play.   // classes

outlook: sunny, overcast, rain. // attributes
temperature: continuous.
humidity: continuous.
windy: true, false.
```

The Data

Notice that some attributes are real-valued.

```
sunny, 85, 85, false, Don't Play
sunny, 80, 90, true, Don't Play
overcast, 83, 78, false, Play
rain, 70, 96, false, Play
rain, 68, 80, false, Play
rain, 65, 70, true, Don't Play
overcast, 64, 65, true, Play
sunny, 72, 95, false, Don't Play
sunny, 69, 70, false, Play
rain, 75, 80, false, Play
sunny, 75, 70, true, Play
overcast, 72, 90, true, Play
overcast, 81, 75, false, Play
rain, 71, 80, true, Don't Play
```

The created decision tree:

```
---------------------------------------

Read 14 cases (4 attributes) from golf.data

Decision Tree:

outlook = overcast: Play (4.0)
outlook = sunny:
|   humidity <= 75 : Play (2.0)
|   humidity > 75 : Don't Play (3.0)
outlook = rain:
|   windy = true: Don't Play (2.0)
|   windy = false: Play (3.0)

Evaluation on training data (14 items):

Before Pruning          After Pruning
---------------   ------------------------
Size    Errors   Size   Errors   Estimate
8    0( 0.0%)    8    0( 0.0%)    (38.5%)
```

Trace of DT creation

```
Read 14 cases (4 attributes) from golf.data

14 items, total weight 14.0
Att outlook     inf 1.577, gain 0.247
Att tempe cut=70.500, inf 0.940, gain 0.045
Att humidity    cut=82.500, inf 0.940, gain 0.102
Att windy       inf 0.985, gain 0.048

 best attribute outlook inf 1.577 gain 0.247

5 items, total weight 5.0
Att temp cut=77.500, inf 0.971, gain 0.420
Att humidity    cut=77.500, inf 0.971, gain 0.971
Att windy       inf 0.971, gain 0.020
     average gain 0.470
best att
humidity cut 77.500 inf 0.971 gain 0.971 val 1.0

5 items, total weight 5.0
 Att temp  cut=70.500, inf 0.971, gain 0.020
   Att humidity    no gain
  Att windy       inf 0.971, gain 0.971
      average gain 0.495
 best attribute
 windy inf 0.971 gain 0.971 val 1.000
```

If the tree is converted into rules, it is usually easier to understand

```
Rule 2:  outlook = overcast
        ->  class Play  [70.7%]

Rule 4: outlook = rain
        windy = false
        ->  class Play  [63.0%]

Rule 1: outlook = sunny
        humidity > 75
        ->  class Don't Play  [63.0%]

Rule 3: outlook = rain
        windy = true
        ->  class Don't Play  [50.0%]

Default class: Play

Evaluation on training data (14 items):

Rule  Size  Error  Used  Wrong      Advantage
----  ----  -----  ----  -----      ---------
2  1  29.3%    4   0 (0.0%)  0 (0|0)    Play
4  2  37.0%    3   0 (0.0%)  0 (0|0)    Play
1  2  37.0%    3   0 (0.0%)  3 (3|0)    Don't
3  2  50.0%    2   0 (0.0%)  2 (2|0)    Don't

Tested 14, errors 0 (0.0%)   <<


       (a)  (b)      <-classified as
      ---- ----
         9           (a): class Play
              5      (b): class Don't Play
```

Basics

- Basic Assumption: concept is disjunct of conjuncts of attribute-values
  - sharp boundaries, i.e no degree of membership
- Input Representation
  - attributes with symbolic values
  - e.g. sex: male, female; income: low, average, high
- output: a decision tree

Properties

- learns with positive and negative examples
- noise tolerant
- general-to-specific search (reverse for pruning)
- batch (requires all the data) as opposed to incremental, where you learn after each example, which is what neural net algorithms do.
- Follows Divide-and-Conquer strategy, which has weaknesses of fracturing and diminishing training data.
- Learns discriminating rules

Basic Algorithm

Terms in <> need to be refined

Initial tree ← true (no tests)
While examples under leaf not <Pure>
<Choose> a <Test> and split the leaf by test's answers.
<Label> leaf with appropriate class.

Refinements

- Purity
  - all of one class
  - mostly of one class
- Tests
  - Nominal attribute: A=v1, A=v2, etc. : all attributes (C4.5)
  - A=v or not A=v (binary trees)
  - For real, $A < v$ or not (v midpoint of values)
- Choosing an attribute
  - at random
  - information gain (measure of mixedness)
- Labeling a leaf
  label with majority class of item

Logarithm Review

Definition: $log_2(x) = b$ means $2^b = x$
Properties:

- log(1)=0
- log(A*B)=log(A)+log(B)
- $log(A^n) = n * log(A)$
- log(A/B)=log(A)-log(B)
- log(1/A)= -log(A)
- 0*log(0) = 0
- $log(1/2^k) = k$

Information Gain

- Let p_i the probability of class i in set S
- $Info(S) = -\sum_i p_i * log_2(p_i)$
- Info(S)=0 if all of one class
- $Info(S) = log_2(n)$ if n equally likely classes.
- What about other splits?
- $gain(A) = Info(S) - \sum_{all\ attributes\ values\ v} Prob(A = v) * Info(S_v)$

From Trees to Expert Systems

- Each path to a leaf defines a rule
- allows combining decision process from multiple rules
- simplify rules by deleted unneeded conditions
- throw away redundant rules
- rules easier to understand (sometimes)
- Multiple trees yields multiple rules
- General Problem: combining Evidence.

Performance

- Efficiency
- Comprehensible
- Neural Nets sometimesn beat by small margin, but the knowledge gained is less comprehensible.

Successes

- Skicat (Fayyad): Binary trees with constructed attributes
  Classifies star data better than astronomers
  training classification: high quality pictures labelled by experts
  data: low quality pictures of entire sky
- Loan Applications: (Michie)
  better at evaluating candidates than people
- British Petroleum used DTs to generate 2500 rules for helping to control gas-oil separation. Claimed saving of millions.
- Instead of writing code to control an airplane, a DT approach was used. Positive examples (90,000) were taken from expert pilots. Negative examples were what they didn't do. The result was a program that was better than the experts.
- Sprouter optimized nuclear fuel plant. Given 30 parameters how to set? Used high, low performance class and history. 1/2 day of operation equalled payback.
- Expert Systems in general