Heuristic Discriminants

Road Map

- Simple Decision Trees
- The AdaBoost Algorithm
  - Boosted Decision Trees
Decision Tree Classification
Twenty Questions on Steroids

Decision Trees are a **heuristic**, and not really a statistical test
- Divides data into two alternate classes of data

Sequentially choose “cuts” to classify data
- This **really** is “twenty questions”
  - Consider 20 yes/no questions: Can isolate 1 out of a million
- Questions are chosen to optimize the information entropy

Advantages
- Explanation is very simple (see above)
- It's a “white box”
  - You can look at each individual decision and understand how it was arrived at.
- Easy to implement

Disadvantages
- Not all that common in physics
- Works best for “square” cut regions
To Play, or Not To Play

14 people show up at a tennis resort
How many of them to you expect to play on any given day?

Analysis

Overcast: Everybody Plays
Rain, but no wind: 60% Play
Sunny, but not humid: 40% Play

Notice: You can analyze how decisions were made
What you need to build a Decision Tree

- A set of training examples with attribute values
- Attributes can be
  - Discreet: The color of your toothbrush
  - Ordered and Discreet: The number of cars in the parking lot
  - Continuous: The room temperature
- The training set needs to be large
  - If not big enough you either
    - Will “over train” the DT
    - Won't be able to make enough decisions to get good separation between the data classes
Decision Trees Dissected

- Start with a set of training examples
  - Stop if the set is too small
  - Stop if all of the examples are the same type
- Find the cut (decision) that maximizes the figure of merit
  - This will generate two subsets of data
  - Inputs to the figure of merit
    - Weight (number) of examples correctly assigned to class 1
    - Weight of examples incorrectly assigned to class 1
    - Weight of examples correctly assigned to class 2
    - Weight of examples incorrectly assigned to class 2
- Recurse on the two new subsets of data
Training Set Example

Two classes of data
Blue Boxes
Red Rings
Training Set Example

First Decision
Training Set Example

Second Decision
Training Set Example

Over Trained!

Third Decision
Figures of Merit

- Tree split decisions as based on figures of merit
  - An arbitrary function, that tells you well a particular split works
- Several F.O.M.s have been suggested
  - Gini Impurity: Based on a measure of uniformity proposed by C. Gini in 1912
    - The continuous case is often used to measure income inequality
    - We use the binary case
  - Idiosyncratic Dichotomizer 3 (ID3): Measures the “Information Entropy”
    - Based on the ideas of information theory
  - Signal Significance: Separating out a signal in the presence of a large background
- Others...
The Figure of Merit: ID3

The original decision tree figure of merit (Quinlan, 1983)
- Based on the idea of information entropy, and minimizing the number of bits to uniquely identify an example

\[
\text{Entropy}(S) = P(\text{right}) \log_2 P(\text{right}) - P(\text{wrong}) \log_2 P(\text{wrong})
\]

- Minimizes the number of questions required to identify an example
  - Not very good a "partial classification"
    - This is when you ask a limited number of questions

- Entropy is minimized with classifications are all right, or all wrong.
The Figure of Merit: Gini Impurity

- A measure of the impurity in each node
  - It's zero when all nodes are pure

\[ \text{Gini}(S) = (\text{Class 1 right}) \times (\text{Class 2 wrong}) + (\text{Class 1 wrong}) \times (\text{Class 2 right}) \]

- Closely related to the Gini Coefficient invented in 1912 by “C. Gini”
  - Measures the deviation of a distribution from uniformity
Figure of Merit: Signal Significance

- Optimize the statistical significance of a signal in the presence of a large background
- This is an asymmetric figure of merit

\[
\text{Significance}(S) = \frac{\text{signal}}{\sqrt{\text{signal} + \text{background}}}
\]

- The significance depends on the total number of expected signal and background events
- Can't really express as a fraction...
Problems with Decision Trees

- Decision Trees are a very good heuristic to understand how to discriminate data

- But,

  - Over Training is a problem
    - The algorithm will sub-divide data until there is one example in each box
  
  - Works best for “rectangular cuts”
Adaptive Boosting

- An meta-algorithm for use with other discriminant/learning algorithms
  - Works with **Supervised Learning**
  - Works with any **Weak Discriminant**
    - i.e. Any other discriminant
- The basic idea
  - Assign a weight to each event in a training set i=1..n
    - The measurements are $x_i$
    - Start with uniform weights $W_1(i) = 1/n$
  - Construct a set of discriminants $t = 1 .. T$
    - Train a weak discriminant to classify the weighted events, $d_t(x)$
      - Could be a decision tree, or a MLP network
      - Calculate an error rate for the discriminant, $\alpha_t$
      - Calculate a new weight for each event
        $$w_{t+1}(i) = \begin{cases} W_t(i)e^{+\alpha_t} & \text{if identified correctly} \\ W_t(i)e^{-\alpha_t} & \text{if identified incorrectly} \end{cases}$$
  - Final discriminant is
    $$D(x) = \sum_{t=1}^{T} \alpha_t d_t(x)$$
    W increases if the event is misidentified
Adaptive Boosting

Pick the first discriminant with all the events having the same weight
Adaptive Boosting

Boost the weight of events that are misidentified.
Adaptive Boosting

Pick a new discriminant for the reweighted events.

Repeat for a bunch of discriminants. The final discriminant is a “vote”.
What's going on with AdaBoost

It's like taking an opinion poll
- Each weak discriminant has a particular bias
  - When an event is correctly identified by one discriminant, it's less likely to be correctly identified by the next
  - visa versa
- Combination of the discriminants makes a “smooth” approximation of the likelihood ratio.

Advantages
- It's a good way to combine many simple discriminants (e.g. DTrees)
- Each simple discriminant can be looked at independently

Disadvantages
- It's complicated enough that the final discriminant becomes a bit of a black box.
Example: Select a Correlated Gaussian Signal

Classification with one decision tree

Example Distribution

Combined Classifier Distribution
Example: Select a Correlated Gaussian Signal

Classification with 3 decision trees

Example Distribution

Combined Classifier Distribution
Example: Select a Correlated Gaussian Signal

Classification with 7 decision trees

Example Distribution

Combined Classifier Distribution
Example: Select a Correlated Gaussian Signal

Classification with 17 decision trees

Example Distribution

Combined Classifier Distribution
Example: Select a Correlated Gaussian Signal

Classification with 70 decision trees

Example Distribution

Combined Classifier Distribution
Finally

- Heuristic classification algorithms such as decision trees can be a good alternative to pure statistical methods
  - Attempt to approximate how a person learns to separate data
  - These algorithms don't try to approximate the Likelihood ratio
- Over-training can be a significant shortcoming
  - Simple heuristics can “memorize” your training set
  - An over-trained discriminant won't provide good separation when applied to a different set of data
  - Over-training is (partially) avoided by using one set of data to train, and another set to validate.
- Decision Trees and AdaBoost were used as examples
  - Lot's of other algorithms, each has strengths and weaknesses

The End