Machine Learning Intro

CPS 270
Ron Parr

Why Study Learning?

- Considered a hallmark of intelligence
- Viewed as way to reduce programming burden
- Many algorithms assume parameters that are difficult to determine exactly a priori

Formalizing Learning

- Learning Element
  - The thing that learns
- Performance Element
  - Objective measure of progress
- Learning is simply an increase in the ability of the learning element over time to achieve the task specified by the performance element

Types of Learning

- Inductive Learning
  - Acquiring new information that previously was not available
  - Learning concepts
- Speedup learning
  - Learning to do something you already "know" faster or better

Feedback in Learning

- Supervised Learning
  - Given examples of correct behavior
- Unsupervised Learning
  - No external notion of what is correct
  - Is this well-defined?
- Reinforcement Learning
  - Indirect indication of effectiveness

Learning Methodology

- Distinction between training and testing is crucial
- Correct performance on training set is just memorization!
- Researcher should ***never*** look at the test data
- Raises some troubling issues for "benchmark" learning problems

Computational Learning Theory

- Formal study of what can be learned from data
- Need to make some assumptions:
  - Training examples must be representative
  - Algorithm needn't always work, but should scale well
  - Number of samples or run time needed should have polynomial relationship to inverse probability of error and percentage of misclassifications
- Probably Approximately Correct (PAC) learning
- Goal: Algorithms that have a low error rate with high probability

COLT

- Learning theory is elegant and mathematically rich. However,
  - It sometimes isn't constructive
  - It sometimes tells us how many data are needed, but not how to manipulate the data efficiently
- Until recently, learning theory has been disconnected from practical learning algorithms
- New advances are leading to a rapprochement
  - Support vector machines
  - Boosting

Supervised Learning

- Classical framework
- Target concept, e.g., green
- Learner is presented with labeled instances
  - True: Green cones, green cubes, green spheres
  - False: Red cones, red cubes, red spheres, blue cones, blue cubes, blue spheres
- Learner must correctly identify the target concept from the training data

Performance Measure

- Training set won't have all possible objects
- Test set will contain novel objects
  - Blue cylinders, yellow tetrahedra
- To learn successfully, learner must have good performance even when confronted with novel objects
  - This is what we would expect from people
  - A blue Broccolisaurus is still blue

Examples

- Distinguish healthy/diseased tissue
- Predict good/bad credit risks
- Drilling sites likely to have oil
- Document categorization
- Predict failures in physical systems
- Predict if mushrooms are safe to eat
- Good/bad moves in a game
- Public database of learning problems:
  - http://www.ics.uci.edu/~mlearn/MLSummary.html

Why Learning Is Tricky

- Suppose we have seen:
  - Red tetrahedron(f), Blue sphere(t), Blue cone(t), green cube(f)
- Possible concepts:
  - Blue
  - (Blue Sphere) or (Blue Cone)
  - Objects a prime number from start
  - Objects with a circular cross-section
- What if some data are mislabeled?

Learning and Representation

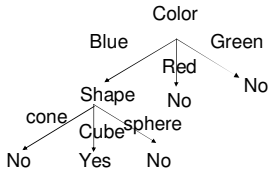
- Learning is very sensitive to representation
- Every learning algorithm can be viewed as a search procedure through a space of concepts
- Space of concepts determines
  - Difficulty of task
  - Appropriate algorithm
  - Restricting too aggressively can trivialize problem
- Space: Conjunctions of colors and shapes
  - Eliminates primes and (possibly) cross sections

Which Hypothesis is Best?

- Many approaches use Ockham's Razor:
  - All things being equal, favor simplest consistent hypothesis
- This cannot be proved correct, but the general understandability of the world seems to favor it
- Used in physics, philosophy
- Note: Defining "simplicity" not as simple as it sounds

Decision Trees

- Decision trees try to construct small, consistent hypothesis
- Suppose our concept is "blue cube"



Facts About Decision Trees

- If the concept has d conjuncts, there will be a decision tree for this concept with depth d
- Decision trees very bad in some functions:
  - Parity function
  - Majority function
- For errorless data, can always construct decision tree that correctly labels entire training set, but may be exponential in size

Decision Tree Algorithms

- Aim for:
  - Small decision trees
  - Robustness to misclassification
- Constructing shortest decision tree is intractable
- Standard approaches are greedy
- Classical approach is to split tree using an information-theoretic criterion

Growing Decision Trees

```
Repeat until no good leaves
  Pick leaf
  Split = choose_variable(variabes – all_parents(leaf))
  For val in domain(split)
   new_leaf = new_leaf(split=val)
   new_leaf.instances=leaf.stances s.t. split=val

For leaf in tree
  classification(leaf)=majority_classification(leaf)
```

Information Theory

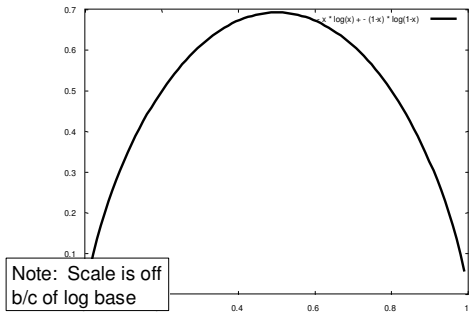
- Roughly speaking, information theory measures the expected number of bits needed to communicate information from one person to another
- Suppose person1 is flipping a coin with bias p
- Person1 wants to tell person2 the sequence of results
- What is the expected number of bits person 1 will send to person 2?
- Note relation to compression

Information Content

$$I(p_1,\ldots,p_n) = E(\#\text{bits}) = \sum_{i=1}^{n} - p_i \log_2(p_i)$$

For an unbiased coin, the information content is 1.
For a totally biased coin, the information content is 0.

Information Content



Note: Scale is off b/c of log base

Information Content of a Leaf

$$I(\frac{p}{p+n},\frac{n}{p+n}) = -\frac{p}{p+n}\log_2\frac{p}{p+n} - \frac{n}{p+n}\log_2\frac{n}{p+n}$$

Information gain of a split:

$$I(\frac{p}{p+n},\frac{n}{p+n}) - \sum_{i=1}^{v}\frac{p_i+n_i}{p+n} I(\frac{p_i}{p_i+n_i},\frac{n_i}{p_i+n_i})$$

Gain Example

- Suppose we have seen:
  - Red tetrahedron(f), Blue sphere(t), Blue cone(t), green cone(f)
- Is it better to split on shape or color?
- Information of original set is: 1
- Information gain of splitting on cone:
- Information gain of splitting on blue:

Favoring Small Examples

- Information gain (and other splitting criteria)
  - Are greedy
  - Favor small trees
- This makes representation an issue yet again
- Suppose you want to learn “parity(+) and blue”
- Hard to learn with decision trees, but
  - If we treat parity like a state variable, then it’s easy
  - Call these derived variables features or attribrutes

Decision Tree Conclusion

- Simple method
- Works surprisingly well in many cases
- Issues:
  - Continuous variables
  - Missing values
  - Expressive power