CS205L
Continuous Mathematical Methods
(emphasizing Machine Learning)

http://web.stanford.edu/class/cs205l/index.html
Knowledge Based Systems (KBS)

Contains two parts:

Knowledge Base
- Explicit knowledge or facts
- Often populated by an expert (expert systems)

Inference Engine
- Way of reasoning about the facts in order to generate new facts
- Typically follows the rules of Mathematic Logic

See Wikipedia for more details...
Machine Learning (ML)

Contains two parts:

Training Data
- Data Points typically as domain/range pairs
- Hand labeled by a user, measured from the environment, or generated procedurally

Model
- Derived from Training Data in order to estimate new data points minimizing errors
- Uses Algorithms, Statistical Reasoning, Rules, Networks, Etc....

See Wikipedia for more details...
KBS vs. ML

• Personally, I might argue that these are simply the **discrete math** and **continuous math** approaches to the same problem (respectively)

• ML’s Training Data is its Knowledge Base

• Logic is the algorithm used to discover new discrete facts for KBS, whereas many algorithms/methods are used to fill in continuous facts/data for ML
  • Logic happens to be especially useful for discrete facts
Data/Fact Collection

2 # 2 = 4

What operation does “#” stand for? Multiplication or Addition?

Suppose we also had 2 # 3 = 6? Multiplication!
Suppose we also had 2 # 3 = 5? Addition!
More data helps makes ambiguities clear

Suppose there is noise or error in the data and we have 2 # 3 = 5.5
Now what? Again, need more data to help combat errors

Goal in learning is to make a model for #, so we can predict a # b for any a and b
KBS Approach

• Rule: a and b commute
• Start with a and b as single digits, and record all a # b outcomes as facts (using addition/multiplication)
  • This is how/why kids memorize arithmetic tables
• Add rules to deal with numbers with more than one digit by pulling out powers of 10
  • That is, use the single digit rules column by column and learn to carry, etc.
  • Multiplication rules require the help of addition rules (carry, add, etc.)
• Mimics human learning (or at least human education)
• This is a discrete approach, and it has no inherent error!
ML Approach

• Make a 2D domain $\mathbb{R}^2$, and 1D range $\mathbb{R}^1$
• Let $(a,b)$ be a 2D input point in the domain
• For each input point $(a,b)$ evaluate $a \# b$ (using addition/multiplication)
• Then, the 3D point $(a, b, a\#b)$ is training data, so make a bunch of these
• Pass a smooth surface through these 3D points to get a model for $\#$
• Pros:
  • Get decimals, irrationals, etc. for free (all the continuous numbers)
  • Don’t need to create discrete rules for multiplying pi or root 2 etc. as in a KBS
• Cons:
  • Errors in the predictions, noisy
  • Cannot extrapolate outside where there is training data
  • Needs lots of training data (although, one could generate lots of sparse data using a KBS!)