Computer Discovery

machine learning in scientific domains

Easy versus Hard Inductive Generalization

- Easy means
  - Finite sample
  - Finite Test
  - Given classes to find

- Hard means
  - Finite sample
  - Infinite test or no classes.
    - The classic hard problem is acquisition of "infinite language" from finite exemplars
Learnability of Language

Studies of children learning language find amazing abilities:
- leap from 2 word sentences to 8 word sentences
- acquisition of 14 words per day
- rarely makes wrong generalization

Stages of Language Learning

- babbling
- signing and pointing
- listening vocabulary
- one word utterance
- two word utterances
- multiword
- drastic growth of lexicon
- Overgeneralization Errors
**Poverty of the Stimulus**

- Positive language data only
  - from several sources (variance in pitch, voice, etc)
  - Limited "motherese" (repetitive, high pitch, intoned)
- Prosody
- Semantic/Pragmatic
- Positive Feedback (limited)
  - "gimme cookie"  "Did you say give me cookie?"
  - Yes he said DADA!

**Supervised vs Unsupervised Learning**

- Classification is GIVEN categories as input.
  - What if you only have samples, but no categorization?
  - The learner has to discover a set of classes and determine how to divide new inputs into different classes.
Many clustering algorithms

- Large field of Statistical models of clustering
  - hierarchal clustering
  - principal components
  - Nearest Neighbor algorithms
- AI approaches to "concept formation"
  - Cobweb, AQ11
- NN approaches
  - Kohonen, Neocognitron, ART,
  - Competitive Learning
- EE Approaches
  - Data Compression via vector quantization

Where are Hard problems

- Mathematical Induction and Discovery
  - What is a number?
    - the concept, not the 32 bits!
  - What is infinity?
    - Why does pi have infinite digits?
- Metacognition
  - The Mutilated Checkerboard problem
- Scientific Discovery
  - Observe some behavior
  - Induce an explanation of the behavior
Discovery Problems

- Theory from data
- Finding Categories or taxonomies
- Induction of Laws
- Explanation
- Prediction
- Creative discovery
  - New emergent fields call "Knowledge Discovery" and "Data Mining" with application to e.g. bioinformatics

How can Machine Learning help?

- Scientific theories are a form of knowledge which is acquired
- Lots of data can be reduced to simple rules, as in decision-tree learning and this is often the goal of scientific analysis
  - However, the results of scientific discovery is not in decision-tree form!
- Discovery of clusters and taxonomies using contrasting information
- Induction of quantitative theories (equations)
- Drug Discovery (profitable branch of AI)
Two Famous AI Programs

- **BACON**
  - Heuristic construction of equations from data
- **Automated Mathematician**
  - Heuristic search of number theory

BACON (Langley & Simon)

- Subtask of science: Fitting Theories to Data
  - Equation Creation
  - Curve Fitting
    - But not a parameterized universal method like polynomials
Bacon

- **Input:** Tables of experimental data
  - masses, forces, accelerations
  - length, height, area of rectangles
  - distance and period of Planets
  - Volts, Ohms, Current
  - Pressure, Temperature
- **Output:**
  - Equations of the "laws" which are holding

---

How can a program do this?

- Searching among functional relationships using heuristics to find invariant relationships
  - If X and Y are linearly related with slope S and intersect I, hypothesize a linear relation
  - (xy>0) If X increases as Y decreases and X and Y are not linearly related then define new term as product of X and Y
  - (xy > 0) IF X increases as Y increases, define a new term as ratio of X and Y
  - (XY<0) If X increases as Y increases, look at ratio
  - (XY<0) if X and Y Increase, look at product
Bacon Example

- Both are going up, nonlinearly, look at ratio...

<table>
<thead>
<tr>
<th></th>
<th>distance</th>
<th>period</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mercury</td>
<td>0.382</td>
<td>0.241</td>
</tr>
<tr>
<td>Venus</td>
<td>0.724</td>
<td>0.616</td>
</tr>
<tr>
<td>Earth</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Mars</td>
<td>1.524</td>
<td>1.881</td>
</tr>
<tr>
<td>Jupiter</td>
<td>5.199</td>
<td>11.855</td>
</tr>
<tr>
<td>Saturn</td>
<td>9.539</td>
<td>29.459</td>
</tr>
</tbody>
</table>

- Now, d/p goes down as D goes up...

<table>
<thead>
<tr>
<th></th>
<th>distance</th>
<th>period</th>
<th>d/p</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mercury</td>
<td>0.382</td>
<td>0.241</td>
<td>1.607</td>
</tr>
<tr>
<td>Venus</td>
<td>0.724</td>
<td>0.616</td>
<td>1.175</td>
</tr>
<tr>
<td>Earth</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Mars</td>
<td>1.524</td>
<td>1.881</td>
<td>0.81</td>
</tr>
<tr>
<td>Jupiter</td>
<td>5.199</td>
<td>11.855</td>
<td>0.439</td>
</tr>
<tr>
<td>Saturn</td>
<td>9.539</td>
<td>29.459</td>
<td>0.324</td>
</tr>
</tbody>
</table>
D/P and DD/P move in opposite directions...

<table>
<thead>
<tr>
<th></th>
<th>distance</th>
<th>period</th>
<th>d/p</th>
<th>dd/p</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mercury</td>
<td>0.382</td>
<td>0.241</td>
<td>1.607</td>
<td>0.622</td>
</tr>
<tr>
<td>Venus</td>
<td>0.724</td>
<td>0.616</td>
<td>1.175</td>
<td>0.851</td>
</tr>
<tr>
<td>Earth</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Mars</td>
<td>1.524</td>
<td>1.881</td>
<td>0.81</td>
<td>1.234</td>
</tr>
<tr>
<td>Jupiter</td>
<td>5.199</td>
<td>11.855</td>
<td>0.439</td>
<td>2.28</td>
</tr>
<tr>
<td>Saturn</td>
<td>9.539</td>
<td>29.459</td>
<td>0.324</td>
<td>3.088</td>
</tr>
</tbody>
</table>

Our Program has "discovered" Kepler's Third Law!

<table>
<thead>
<tr>
<th></th>
<th>distance</th>
<th>period</th>
<th>d/p</th>
<th>dd/p</th>
<th>ddd/pp</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mercury</td>
<td>0.382</td>
<td>0.241</td>
<td>1.607</td>
<td>0.622</td>
<td>1</td>
</tr>
<tr>
<td>Venus</td>
<td>0.724</td>
<td>0.616</td>
<td>1.175</td>
<td>0.851</td>
<td>1</td>
</tr>
<tr>
<td>Earth</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Mars</td>
<td>1.524</td>
<td>1.881</td>
<td>0.81</td>
<td>1.234</td>
<td>1</td>
</tr>
<tr>
<td>Jupiter</td>
<td>5.199</td>
<td>11.855</td>
<td>0.439</td>
<td>2.28</td>
<td>1</td>
</tr>
<tr>
<td>Saturn</td>
<td>9.539</td>
<td>29.459</td>
<td>0.324</td>
<td>3.088</td>
<td>1</td>
</tr>
</tbody>
</table>
Laws that BACON has (re)discovered

- BACON.1: Kepler's law \( \frac{D_3}{P^2} = c \)
- BACON.2: Ideal gas law \( PV = aNT + bN \)
- Coulomb's law \( \frac{F D^2}{Q_1 Q_2} = c \)

AM (Lenat 1977)

- Represented mathematical concepts in a frame systems
- Used 250 hand-made Heuristics
- Mutated small lisp programs generating sets of numbers
  - select most interesting concept and generate examples
  - Look for regularities and create conjectures
  - propagate through existing knowledge
- Maintained an agenda of "interestiness"
- "Discovered" many concepts in number theory
AM Implementation Concepts

- Agenda (what to try next)
- Interestingness Heuristics
- Concept Representation

AM Representation like semantic net

- NAME: Prime
- ISA: set
- DEFN: Prime (x) if Z|x then z is element of \{1,x\}
- SPECIALIZATIONS: odd-primes
- GENERALIZATIONS: NUMBERS
- EXAMPLES
- INTERESTINGNESS: 100
- ORIGIN: 11-10-92 10:00
- SEE-ALSO: divisors-of
AM Heuristics

- A set of rules which helped to generate new concepts and to direct the search to more interesting paths
  - Consider Extreme elements
    - Factor sets -> prime numbers, Squares, Maximally divisable
  - Look at intersections of concepts
  - Generalize and specialize concepts

Results

- Discovered "Prime numbers"
- Discovered other mathematical conjectures
  - Goldbach's Conjecture
  - Unique Prime Factorization Thm
AM ultimately petered out into uselessness

- What was problem?
  - Heuristics need to be improved...
  - Lenat knew what he wanted to happen...

In Conclusions...

- AM was revealed as a bit bogus
  - Google for "micro-lenat"?
  - Code was not released
  - AM wasn't "discovering," but was "harvesting" low-hanging mathematical fruit intentionally placed in LISP by designers.
    - In other words, the inductive bias of AM matched the mathematical bias of LISP
- Bacon series worked on inducing laws, but started with "nice" data: Who chose the nice data?