Day 19: Scientific Discovery

- Computer Discovery: Machine Learning in Scientific Domains
  - It's a very difficult form of ML
- Inductive Bias: Any restriction on the universe, or bias on ways to choose between equally valid alternatives
  - Conundrum: limits possibilities, but necessary for learning
  - Ex: Occam's Meta Strategy: simpler descriptions are better
- Easy vs. Hard Generalization
  - Easy: Finite sample and finite test set, with a given set of classes
  - Hard: Finite sample, but infinite test set and/or no provided classes
    - Chomsky's problem: given a finite sample of language you need to be able to parse infinite sentences
- Supervised vs. Unsupervised Learning
  - If supervised, the training sample is already classified
  - If unsupervised, it's clustering
- What are Hard ML problems?
  - Mathematical induction/discovery
    - What is a number, at a conceptual level?
    - What is infinity? Why is π infinite?
  - Metacognition: how can an agent learn about the problem it's solving and change its behavior accordingly?
    - Ex: deduce that a problem can't be solved and give up without searching exhaustively (ex. even/odd parity)
    - Ex: realize that pi is infinite and give up
  - Scientific Discovery: observe variables of a system and produce an explanation that generalizes

        - Theory from data/Induction of laws
        - Categorizing/taxonomy
        - Explanation
        - Prediction
        - Creative discovery
        - "Find something interesting"
    - How can ML help science?
        - Scientific theories are a form of knowledge
        - It can often be reduced to simple rules (ex. decision trees)
            - These simple rules are often the goal of science, but usually not in decision tree form
        - Production/discovery of clusters, etc.
        - Induction of quantitative theory/equations
        - Drug discovery
- Three Famous Scientific AI Progroms:
    - Bacon: Heuristic construction of equations from data
        - Equation creation
        - Like curve fitting, but more advanced than just a polynomial
        - Takes experimental data in, outputs equations of "laws" that appear to be correct
        - How? Heuristics (ex. look for linear correlations, then look for ratios, multiply variables to try and find correlations, etc.)
        - Worked on inducing laws, but started with nice data, so who found the data?
    - Automated Mathematician (AM): heuristic search of number theory
        - Represented mathematical concept in a frame system

        - A frame represents a concepts, somewhat similar to a conceptual dependency script
    - 250 hand-made heuristics (ie. a way to mutate a heuristic)
        - Ex: consider extremes, intersections, and generalizations of concepts
    - Select most *interesting* concept and generate examples
    - Interestingness was a heuristic
    - Look for regularities, create conjectures, and propagate through existing knowledge
    - Had a list of things to tried next sorted by interestingness
    - Key challenge: how to figure out what's a concept?
    - Successfully discovered prime numbers, but discovered lots of useless things and petered out
        - Really was just building off of inductive bias via heuristics and the built-ins of Lisp
        - Considered a fraud ("micro-Lenat" is the measure of bogusness)
- Eureqa: extraction of equations of motion from lots of data of physical systems
    - Takes large, amorphous, engineering/scientific data
    - Produced complex equations/laws
    - Used evolution
- Symbolic Regression: what function describes this data?
    - Not just polynomial curve fitting
    - Uses evolution
    - Key improvement: use a random subset of the data for learning at any given time
    - This was productized/marketized as Eureqa