Information Technology in an Expanding Universe

Scott Kirkpatrick
Hebrew University,
Jerusalem

A Little History

- Evolving applications of physics (information theory, statistical mechanics of disorder) to computer science. Examples:
  - ’60s scaling
  - ’70s renormalization group
  - ’75 – ’85 spin glasses (and simulated annealing)
  - ’90s broader study of dynamical systems
  - ’85 – present neural networks → “machine learning”

- But computer science is defined by computing practice, and this has evolved dramatically in only 60 years.

Evolution of Computer Science

- ’40s to ’60s “unit record” era (showing roots in Hollerith cards and hand calculators)
  - Von Neumann self-repairing automata
  - Vannevar Bush Memex
  - Marvin Minsky’s apochryphal “vision summer project”
- ’70s optimization of polytime algorithms on random-access machines
  - Example, matrix multiply goes as N^2.7
    - Here worst case = average case = all cases
- ’75 – ’85 NP-Complete flowering

CS Evolution, ctd.

- ’90s Complexity bifurcates into “theory” and “heuristics”
  - Theory – “non-proliferation agreements” (STOC, FOCS)
    - Nature and methods of proof (e.g. “zero knowledge”)
    - Randomization of algorithms
  - Heuristics – compute cost of potentially exact algorithms (AAAI)
    - Depth-first search, with backtracking
    - Worst-case != typical case
    - Average case behavior sometimes not calculable
    - Increasing importance of computer experiments
- ‘00s Taking notice of Moore’s Law
  - Operating at both ends of the spectrum
  - Need for automation becomes critical as well as fashionable

Appreciation of phase transitions in CS

- Thresholds for properties on graphs – entirely parallel evolution
  - Which graphs?
    - Erdos-Renyi random regular graphs
    - Regular or random lattices in metric spaces (2d, 3D…)
    - Now scale-free graphs defined by growth policies
      - Sparse and dense at the same time
  - Percolation threshold at first considered unique
    - Erdos' "double jump" independent of Fisher, Temperley, …
    - K-core transition known 1^{st} order in RGs, may also occur in 3D
      - K-core rather different in scale-free networks
    - Other thresholds discovered to be sharp in the limit N→ infty.

Phase transitions in CS, ctd.

- Phase transitions (transitions which sharpen as N → infty and can be characterized by threshold functions) are now understood to be common in random graphs
  - Friedgut, Achlioptas, et al… (rigorous, and almost what you wanted to know)

    any monotonic transition not “captured” by a finite set of (cyclic) graphs will be sharp as N → infty
- Phase transitions on scale-free networks?
  - Because these combine dense and sparse parts, gradual or smeared transitions are the most likely outcome.

Understanding NP-Hard problems

- Efforts to apply physics of disordered materials still incomplete, and widely misunderstood by CS practitioners
- Fu-Anderson (1985) graph partitioning is a spin glass
  - Suggested extrapolation – NP Complete problems are spin glasses
  - FALSE – e.g. 2D Ising spin glass, no magnetic field
- Workers in SAT and scheduling problems ('91-94) identified "easy – hard – easy" problems, with the "hard" cases coming at phase boundaries.
  - On closer inspection, these are "easy – hard – less hard"
- Analyze these heuristics by addressing typical case != worst case. Average cost still not well controlled.

NP-Hard problems, ctd.

- 2 + p SAT example
  - NP to P boundary (worst case) occurs as $p > 0$
  - Exponential cost (typical case) starts at $p = 0.4$
  - Suggested extrapolation (by authors) 1st order transitions account for hardness
  - (FALSE – consider k-core)
  - Suggested extrapolation (not by the authors) 1st order transitions explain NP-Completeness (clearly false)
- Recent work on 1-step RSB in 3-SAT
  - Best current generalization – RSB accounts for typical case hardness of depth-first search based heuristics.
  - Note that the work also exposes new heuristics which do better

Where do we go next?

- Where do the three different types of networks occur?
  - Grids – dense computing, storage, and communications fabrics
  - Scale-free – the Internet and things in it
  - Random graphs – problems derived from other networks
- How big is the Internet, its information space, its underpinnings?
  - At least 20 TB, but > 100 TB of non robot-accessible content
  - No one search engine covers all the accessible material
- How fast is it growing?
  - Still doubling every year, changing at least every two months, but not observed on any shorter timescales.

Where do we go next?

- Many algorithms are developing in a way that makes them distributable
  - E.g. survey propagation, "belief propagation", turbo decoding
- What are the most important issues in managing it, or better, in managing organisms that live in it and grow with it?
  - Things "fail in place" leaving a family of percolation problems
  - Recovery speed more important than mean time to fail
  - You can't optimize a constantly evolving organism, but you can regulate its growth
  - Secret weapons – are there effects of RSB in the Web?