Statistical Inference (Re: Chess) and Computational Complexity

Kenneth W. Regan[1]
University at Buffalo (SUNY)

8 September, 2020

[1]Recent and Current Students: Tamal Biswas, Chaowen Guan; Marissa Dominijanni, Chen Xu

Includes work using the UB Center for Computational Research (CCR)

A Basic Inference Example

- Suppose positive *cancer* result from test that is **99.9%** accurate.
- Suppose the cancer affects **1 in 5,000** people.
- *What are the odds that you have the cancer?*
- Let's give 5,000 people the same test. We will expect to get:
  - 1 true positive from 1 person who has the cancer;
  - **5 false positives** from the other 4,999 people who don't have it.
- All you know is you are **1 of 6** positives, 5 of whom do not have it.
- So: **1 in 6**.
- **Now, however, suppose the test is for Covid-19.**
- Affects **1 in 50** people in US. (Local positivity rate even higher.)
- The 5,000 tests give same **5** false positives but **100** true positives.
- So odds are **20-to-1** in favor.

Chess Cheating Before 2020...

- I have been dealing this year with *essentially the same numbers*. I have a statistical test for cheating with computers (in human-only tournaments) that gives a **z-score** representing "face-value odds" against the *null hypothesis* of fair play.
- In **over-the-board chess**, the *prior probability* of a player cheating is about 1-in-5,000.
- Even if I have **99.999%** accuracy, meaning face-value odds of **100,000-to-1**, that becomes only **20-1** odds after the *prior*—not enough confidence for *comfortable satisfaction* (CAS criterion).
- Using a $\mathbf{z > 5.00}$ criterion (3.5 million-to-one face value) gives a **1-in-700** *case-error rate*.
- Tournaments recognized by the International Chess Federation (**FIDE**) comprise 50,000-to-100,000 players per year.
- At 10–20 cases a year for OTB chess (fewer than 5 coming to hearings), that projects an error once per 35 years at most.

...And Since Chess Went Online

- **But in online chess**, the observed rate is **above 2%**.
- Now sanctioning at 99.9% face-value confidence is 1-in-20 case-error (still too high).
- Sanctioning at 99.99% (**$z > 3.75$**) is about 1-in-200 case error.
- Sanctioning at 99.999% (**$z > 4.25$**) is 1-in-2,000 case error. **OK?**
- *However the* **rate** *of online play is also much higher.*
- **Over 30 million games per month** on each of several major chess servers. (How many in yea-recognized tournaments?)
- So 1-in-2,000 case error could mean errors every week...
- But are online sanctions themselves less serious?
- FIDE is 7th largest world sporting body by # of member federations (FIFA is 4th), higher by # of registered players.
- Have been in policy "summits"; recent word of thanks.

Predictive Analytics

A Predictive Analytic Model:

- Addresses events or decisions with possible outcomes $m_1, m_2, \ldots, m_j, \ldots$
- Assigns to each m_j a probability p_j.
- Projects risk/reward quantities associated to the outcomes.
- Should also assign *confidence intervals* for p_j and those quantities.

Examples of areas that use predictive models:

- Insurance
- Weather forecasting
- Investment managing
- Equity markets
- Betting—in particular, setting initial odds in horse racing etc.

In my model, the m_j are possible moves in a given chess position.

Decision Making in Chess... and Tests

The ____ of drug-resistant strains of bacteria and viruses has ____ researchers' hopes that permanent victories against many diseases have been achieved.

(a) vigor . . corroborated

(b) feebleness . . dashed

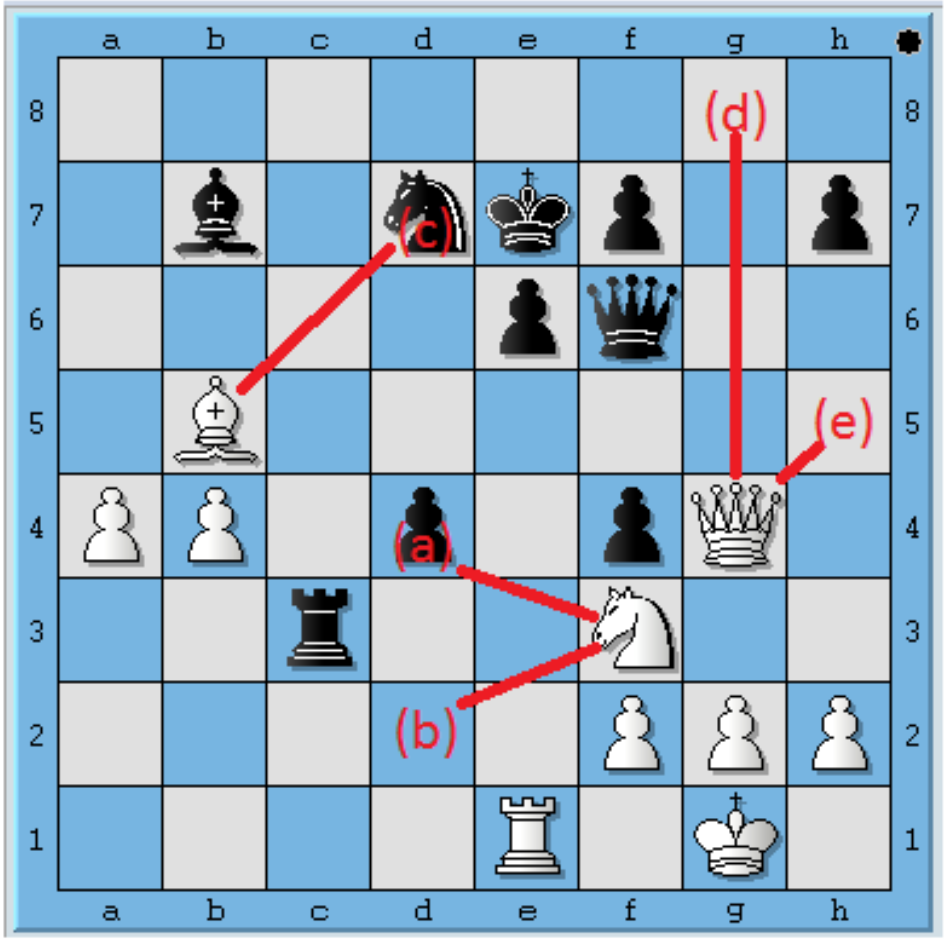
(c) proliferation . . blighted

(d) destruction . . disputed

(e) disappearance . . frustrated

(source: itunes.apple.com)

=



Advantages of Chess Model

1. **Large data**: tens of millions of moves in the public record of games.
2. **Known and Stable Standards**: Quality in chess measured by **Elo rating scale**.
3. **Depth** and **level** of thinking natural from structure of game.
4. **Intrinsic** formulation of **difficulty**.
5. **CSE735 in Fall 2019**: "Introspected" model's own predictive error.
6. Led to **new test** based on **Spiegelhalter's Z-test**.
7. Model update in April 2020 deployed it just in time...
8. **Discover new scientific regularities of human thought processes.**

Computational Complexity

- The study of the time *needed* to solve computational problems, and how much memory and other resources computers require.
- Largely independent of the computer model, beyond a fundamental divide into **serial**, **parallel**, and **quantum**.
- Main technical achievement: the relation of computational problems by **reducibility**.
- Main scientific surprise:

> The **many thousands** of computational problems that have been studied in many disciplines, some for centuries, cluster into **barely over a dozen** equivalence classes under reducibility.

- The biggest cluster is the class of **NP-complete** problems.

P=NP and Worse

- **P**: problems with algorithms that **solve** them in **polynomial time**:

> As the size of the data doubles, the time needed goes up by at most a **linear** factor: $t(n) = n^k \implies t(2n) \leq Kt(n)$, $K = 2^k$.

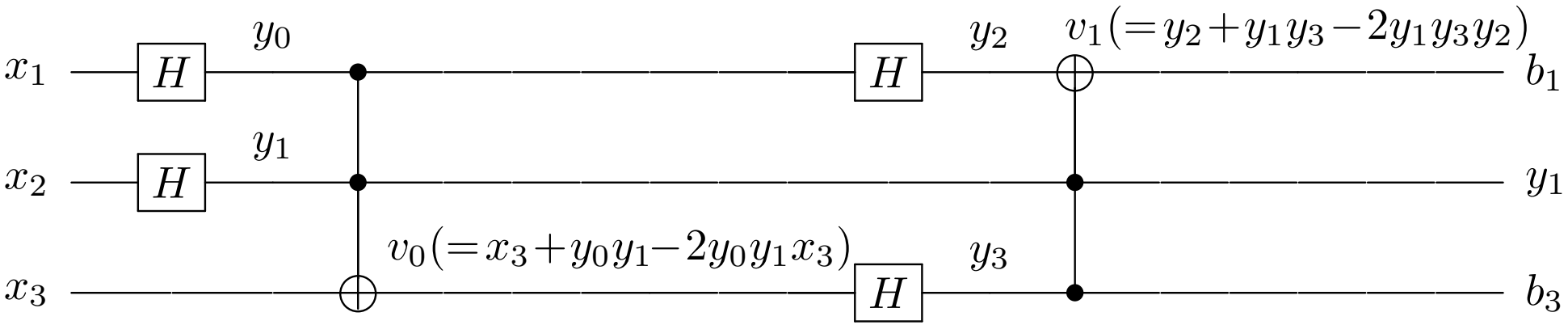
- **NP**: "Nondeterministic" Polynomial Time: If you know a secret fact or guess a good answer, you can verify and **teach** it to someone in polynomial time.
- Example: Given a Boolean formula f like

$$f = (x_1 \vee (\neg x_2)) \wedge ((\neg x_1) \vee x_2 \vee x_3) \wedge ((\neg x_2) \vee (\neg x_3)),$$

  is there a way to make f true?
- Called *Satisfiability* (SAT).
- Equivalent to $\neg f$ *not* being a **tautology**.
- Is NP-complete, so NP = P $\iff$ SAT belongs to P.
- We don't even know whether SAT can be solved in **linear** time!

Other Problems and Models

- **Factoring** is among a handful of problems in NP not known to be complete or in P.
- RSA security depends on it, so many want it to be *hard*.
- But solvable in polynomial time by a **quantum computer**.
- Textbook on quantum algorithms; blog series: Can QCs be Built?
- Research on simulating **quantum circuits** by logic and algebra:




```
p1 = p0 XOR (y0 & x1)   p2 = p1 XOR (y1 & x2)   v0 = x3 XOR (y0 & y1)
p3 = p2 XOR (y2 & y0)   p4 = p3 XOR (y3 & v0)   v1 = y2 XOR (y3 & y1)
```