Chess and Informatics

Kenneth W. Regan
University at Buffalo (SUNY)

CISIM 2017 Keynote

Chess and CS...

Chess and CS...

- Chess: "The *Drosophila* of AI" (Herbert Simon, John McCarthy, after Alexander Kronod)

Chess and CS...

- Chess: "The *Drosophila* of AI" (Herbert Simon, John McCarthy, after Alexander Kronod)
- Advice for AI grad students 10 years ago: "Don't do chess."

Chess and CS...

- Chess: "The *Drosophila* of AI" (Herbert Simon, John McCarthy, after Alexander Kronod)
- Advice for AI grad students 10 years ago: "Don't do chess." (I've lost my source but see Daniel Dennett, "Higher-order Truths About Chmess" [sic], 2006)

Chess and CS...

- Chess: "The *Drosophila* of AI" (Herbert Simon, John McCarthy, after Alexander Kronod)
- Advice for AI grad students 10 years ago: "Don't do chess." (I've lost my source but see Daniel Dennett, "Higher-order Truths About Chmess" [sic], 2006)
    - From 1986 to 2006, **I** followed this advice.

Chess and CS...

- Chess: "The *Drosophila* of AI" (Herbert Simon, John McCarthy, after Alexander Kronod)
- Advice for AI grad students 10 years ago: "Don't do chess." (I've lost my source but see Daniel Dennett, "Higher-order Truths About Chmess" [sic], 2006)
  - From 1986 to 2006, **I** followed this advice. Turned down many requests for what I saw as "me too" computer chess.

Chess and CS...

- Chess: "The *Drosophila* of AI" (Herbert Simon, John McCarthy, after Alexander Kronod)
- Advice for AI grad students 10 years ago: "Don't do chess." (I've lost my source but see Daniel Dennett, "Higher-order Truths About Chmess" [sic], 2006)
  - From 1986 to 2006, **I** followed this advice. Turned down many requests for what I saw as "me too" computer chess. Main area = computational complexity, in which I also partner Richard Lipton's popular blog.

Chess and CS...

- Chess: "The *Drosophila* of AI" (Herbert Simon, John McCarthy, after Alexander Kronod)
- Advice for AI grad students 10 years ago: "Don't do chess." (I've lost my source but see Daniel Dennett, "Higher-order Truths About Chmess" [sic], 2006)
  - From 1986 to 2006, **I** followed this advice. Turned down many requests for what I saw as "me too" computer chess. Main area = computational complexity, in which I also partner Richard Lipton's popular blog.
  - Then came the cheating accusations at the 2006 world championship match...

Chess and CS...

- Chess: "The *Drosophila* of AI" (Herbert Simon, John McCarthy, after Alexander Kronod)
- Advice for AI grad students 10 years ago: "Don't do chess." (I've lost my source but see Daniel Dennett, "Higher-order Truths About Chmess" [sic], 2006)
  - From 1986 to 2006, **I** followed this advice. Turned down many requests for what I saw as "me too" computer chess. Main area = computational complexity, in which I also partner Richard Lipton's popular blog.
  - Then came the cheating accusations at the 2006 world championship match...
- Now: chess gives a window on CS advances and data-science problems.

External History of Computer Chess: Part One

External History of Computer Chess: Part One

- **1950s:** Papers by Turing, Shannon, Newell-Simon-Shaw, others. . . , programs by Prinz, Bernstein, Russian BESM group.

External History of Computer Chess: Part One

- **1950s:** Papers by Turing, Shannon, Newell-Simon-Shaw, others. . . , programs by Prinz, Bernstein, Russian BESM group.
- **1960s:** First programs able to defeat club-level players.

External History of Computer Chess: Part One

- **1950s:** Papers by Turing, Shannon, Newell-Simon-Shaw, others. . . , programs by Prinz, Bernstein, Russian BESM group.
- **1960s:** First programs able to defeat club-level players.
- **1968:** David Levy, International Master (my rank) bet McCarthy and Newell $1,000 that no compter would defeat him by 1978.

External History of Computer Chess: Part One

- **1950s:** Papers by Turing, Shannon, Newell-Simon-Shaw, others. . . , programs by Prinz, Bernstein, Russian BESM group.
- **1960s:** First programs able to defeat club-level players.
- **1968:** David Levy, International Master (my rank) bet McCarthy and Newell $1,000 that no compter would defeat him by 1978.
- **1978:** Levy defeats CHESS 4.7 by 4.5–1.5 to win bet, but computer wins first ever game over master.

External History of Computer Chess: Part One

- **1950s:** Papers by Turing, Shannon, Newell-Simon-Shaw, others. . . , programs by Prinz, Bernstein, Russian BESM group.
- **1960s:** First programs able to defeat club-level players.
- **1968:** David Levy, International Master (my rank) bet McCarthy and Newell $1,000 that no compter would defeat him by 1978.
- **1978:** Levy defeats CHESS 4.7 by 4.5–1.5 to win bet, but computer wins first ever game over master.
- **1981:** CRAY BLITZ (software by Robert Hyatt) achieves first "Master" rating, followed soon by Ken Thompson's BELLE.

External History of Computer Chess: Part One

- **1950s:** Papers by Turing, Shannon, Newell-Simon-Shaw, others..., programs by Prinz, Bernstein, Russian BESM group.
- **1960s:** First programs able to defeat club-level players.
- **1968:** David Levy, International Master (my rank) bet McCarthy and Newell $1,000 that no compter would defeat him by 1978.
- **1978:** Levy defeats CHESS 4.7 by 4.5–1.5 to win bet, but computer wins first ever game over master.
- **1981:** CRAY BLITZ (software by Robert Hyatt) achieves first "Master" rating, followed soon by Ken Thompson's BELLE.
- **1988:** HITECH by Hans Berliner of CMU defeats grandmaster (GM) Arnold Denker in match; DEEP THOUGHT by another CMU group defeats GM and former world championship candidate Bent Larsen in a tournament game.

External History of Computer Chess: Part One

- **1950s:** Papers by Turing, Shannon, Newell-Simon-Shaw, others. . . , programs by Prinz, Bernstein, Russian BESM group.
- **1960s:** First programs able to defeat club-level players.
- **1968:** David Levy, International Master (my rank) bet McCarthy and Newell $1,000 that no compter would defeat him by 1978.
- **1978:** Levy defeats Chess 4.7 by 4.5–1.5 to win bet, but computer wins first ever game over master.
- **1981:** Cray Blitz (software by Robert Hyatt) achieves first "Master" rating, followed soon by Ken Thompson's Belle.
- **1988:** HiTech by Hans Berliner of CMU defeats grandmaster (GM) Arnold Denker in match; Deep Thought by another CMU group defeats GM and former world championship candidate Bent Larsen in a tournament game.
- **1997:** Deep Blue defeats Garry Kasparov 3.5–2.5 in match.

Internal Story of Computer Chess

Internal Story of Computer Chess

- Chess was microcosm of human thinking.

Internal Story of Computer Chess

- Chess was microcosm of human thinking.
- "Chess Knowledge" approach persisted into 1970s.

Internal Story of Computer Chess

- Chess was microcosm of human thinking.
- “Chess Knowledge” approach persisted into 1970s.
- “Brute Force” considered dominant by 1980.

Internal Story of Computer Chess

- Chess was microcosm of human thinking.
- "Chess Knowledge" approach persisted into 1970s.
- "Brute Force" considered dominant by 1980.
- Hsu et al. (1990): "emulation" and "engineering" camps.

Internal Story of Computer Chess

- Chess was microcosm of human thinking.
- "Chess Knowledge" approach persisted into 1970s.
- "Brute Force" considered dominant by 1980.
- Hsu et al. (1990): "emulation" and "engineering" camps.

"It may seem strange that our machine can incorporate relatively little knowledge of chess and yet outplay excellent human players. Yet one must remember that the computer does not mimic human thought—it reaches the same ends by different means."

Internal Story of Computer Chess

- Chess was microcosm of human thinking.
- "Chess Knowledge" approach persisted into 1970s.
- "Brute Force" considered dominant by 1980.
- Hsu et al. (1990): "emulation" and "engineering" camps.

> "It may seem strange that our machine can incorporate relatively little knowledge of chess and yet outplay excellent human players. Yet one must remember that the computer does not mimic human thought—it reaches the same ends by different means."

- Forecast that a *basic search depth* of 14–15 *plies* from raw speed of *1 billion positions per second* would give an *Elo Rating* of **3400**.

Internal Story of Computer Chess

- Chess was microcosm of human thinking.
- "Chess Knowledge" approach persisted into 1970s.
- "Brute Force" considered dominant by 1980.
- Hsu et al. (1990): "emulation" and "engineering" camps.

> "It may seem strange that our machine can incorporate relatively little knowledge of chess and yet outplay excellent human players. Yet one must remember that the computer does not mimic human thought—it reaches the same ends by different means."

- Forecast that a *basic search depth* of 14–15 *plies* from raw speed of *1 billion positions per second* would give an *Elo Rating* of **3400**.
- Real story IMHO is **benchmarking**: *How much measurable problem-solving power can we get out of a machine?*

Benchmarks and Ratings

Benchmarks and Ratings

- Famous benchmarks: *Whetstones*

Benchmarks and Ratings

- Famous benchmarks: *Whetstones*, *Dhrystones*, *mega/giga/tera/peta-FLOPS* via LINPACK, *IOzone*,. . .

Benchmarks and Ratings

- Famous benchmarks: *Whetstones*, *Dhrystones*, *mega/giga/tera/peta-FLOPS* via LINPACK, *IOzone*,...
- Other benchmarks across business suites, embedded computing functions...

Benchmarks and Ratings

- Famous benchmarks: *Whetstones*, *Dhrystones*, *mega/giga/tera/peta-FLOPS* via LINPACK, *IOzone*,...
- Other benchmarks across business suites, embedded computing functions...
- Whole-system benchmarking is harder.

Benchmarks and Ratings

- Famous benchmarks: *Whetstones*, *Dhrystones*, *mega/giga/tera/peta-FLOPS* via LINPACK, *IOzone*,. . .
- Other benchmarks across business suites, embedded computing functions. . .
- Whole-system benchmarking is harder.
- Do we include human software acumen?

Benchmarks and Ratings

- Famous benchmarks: *Whetstones*, *Dhrystones*, *mega/giga/tera/peta-FLOPS* via LINPACK, *IOzone*,...
- Other benchmarks across business suites, embedded computing functions...
- Whole-system benchmarking is harder.
- Do we include human software acumen?
- *Ratings* ground performance in human competitive arenas.

Benchmarks and Ratings

- Famous benchmarks: *Whetstones*, *Dhrystones*, *mega/giga/tera/peta-FLOPS* via LINPACK, *IOzone*,...
- Other benchmarks across business suites, embedded computing functions...
- Whole-system benchmarking is harder.
- Do we include human software acumen?
- *Ratings* ground performance in human competitive arenas.
- Personnel evaluation tests and other *psychometrics* are partial like course grades...

Benchmarks and Ratings

- Famous benchmarks: *Whetstones*, *Dhrystones*, *mega/giga/tera/peta-FLOPS* via LINPACK, *IOzone*,...
- Other benchmarks across business suites, embedded computing functions...
- Whole-system benchmarking is harder.
- Do we include human software acumen?
- *Ratings* ground performance in human competitive arenas.
- Personnel evaluation tests and other *psychometrics* are partial like course grades...
- **Elo Ratings** originated for chess by Arpad Elo in the US in the 1950s.

Benchmarks and Ratings

- Famous benchmarks: *Whetstones*, *Dhrystones*, *mega/giga/tera/peta-FLOPS* via LINPACK, *IOzone*,...
- Other benchmarks across business suites, embedded computing functions...
- Whole-system benchmarking is harder.
- Do we include human software acumen?
- *Ratings* ground performance in human competitive arenas.
- Personnel evaluation tests and other *psychometrics* are partial like course grades...
- **Elo Ratings** originated for chess by Arpad Elo in the US in the 1950s.
- Adopted by the World Chess Federation (FIDE) from 1971 on.

Benchmarks and Ratings

- Famous benchmarks: *Whetstones*, *Dhrystones*, *mega/giga/tera/peta-FLOPS* via LINPACK, *IOzone*,...
- Other benchmarks across business suites, embedded computing functions...
- Whole-system benchmarking is harder.
- Do we include human software acumen?
- *Ratings* ground performance in human competitive arenas.
- Personnel evaluation tests and other *psychometrics* are partial like course grades...
- **Elo Ratings** originated for chess by Arpad Elo in the US in the 1950s.
- Adopted by the World Chess Federation (FIDE) from 1971 on.
- Used by some other sporting bodies.

Benchmarks and Ratings

- Famous benchmarks: *Whetstones*, *Dhrystones*, *mega/giga/tera/peta-FLOPS* via LINPACK, *IOzone*,...
- Other benchmarks across business suites, embedded computing functions...
- Whole-system benchmarking is harder.
- Do we include human software acumen?
- *Ratings* ground performance in human competitive arenas.
- Personnel evaluation tests and other *psychometrics* are partial like course grades...
- **Elo Ratings** originated for chess by Arpad Elo in the US in the 1950s.
- Adopted by the World Chess Federation (FIDE) from 1971 on.
- Used by some other sporting bodies.
- Embraced by the politics and sports prediction website *FiveThirtyEight*.

Elo Ratings R_P for players P

Elo Ratings R_P for players P

- Based on idea that your *points expectation* e goes from 0.0 to 1.0 as a function of difference $x = R_P - R_O$ to your opponent's rating.

Elo Ratings R_P for players P

- Based on idea that your *points expectation* e goes from 0.0 to 1.0 as a function of difference $x = R_P - R_O$ to your opponent's rating.
- Most commonly based on the logistic curve

$$e = \frac{1}{1 + e^{-Bx}} \qquad \text{with} \qquad B = \ln(10)/400.$$

Elo Ratings R_P for players P

- Based on idea that your *points expectation* e goes from 0.0 to 1.0 as a function of difference $x = R_P - R_O$ to your opponent's rating.
- Most commonly based on the logistic curve

$$e = \frac{1}{1 + e^{-Bx}} \qquad \text{with} \qquad B = \ln(10)/400.$$

- Makes a 200-point difference == just over 75% expectation.

Elo Ratings R_P for players P

- Based on idea that your *points expectation* e goes from 0.0 to 1.0 as a function of difference $x = R_P - R_O$ to your opponent's rating.
- Most commonly based on the logistic curve

$$e = \frac{1}{1 + e^{-Bx}} \qquad \text{with} \qquad B = \ln(10)/400.$$

- Makes a 200-point difference == just over 75% expectation.
- Adding e over every game in a tournament yield expected score e_P.

Elo Ratings R_P for players P

- Based on idea that your *points expectation* e goes from 0.0 to 1.0 as a function of difference $x = R_P - R_O$ to your opponent's rating.
- Most commonly based on the logistic curve

$$e = \frac{1}{1 + e^{-Bx}} \qquad \text{with} \qquad B = \ln(10)/400.$$

- Makes a 200-point difference == just over 75% expectation.
- Adding e over every game in a tournament yield expected score e_P.
- New rating is $R'_P = R_P + K \cdot (s_P - e_P)$ where s_P is $P's$ actual score and the factor K is set by policy (e.g. $K = 10$ for established players but $K = 40$ for young/novice/rapidly improving ones).

Elo Ratings R_P for players P

- Based on idea that your *points expectation* e goes from 0.0 to 1.0 as a function of difference $x = R_P - R_O$ to your opponent's rating.
- Most commonly based on the logistic curve

$$e = \frac{1}{1 + e^{-Bx}} \qquad \text{with} \qquad B = \ln(10)/400.$$

- Makes a 200-point difference == just over 75% expectation.
- Adding e over every game in a tournament yield expected score e_P.
- New rating is $R'_P = R_P + K \cdot (s_P - e_P)$ where s_P is $P's$ actual score and the factor K is set by policy (e.g. $K = 10$ for established players but $K = 40$ for young/novice/rapidly improving ones).
- Since only differences matter, absolute rating numbers are arbitrary.

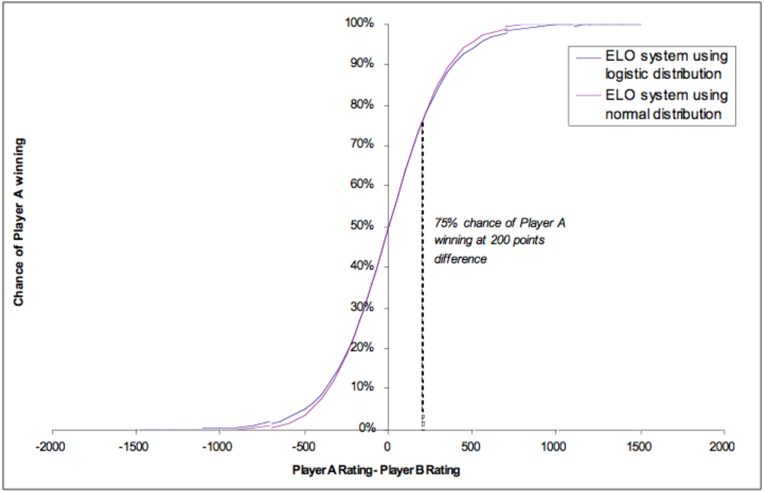
Elo Ratings R_P for players P

- Based on idea that your *points expectation* e goes from 0.0 to 1.0 as a function of difference $x = R_P - R_O$ to your opponent's rating.
- Most commonly based on the logistic curve

$$e = \frac{1}{1 + e^{-Bx}} \qquad \text{with} \qquad B = \ln(10)/400.$$

- Makes a 200-point difference == just over 75% expectation.
- Adding e over every game in a tournament yield expected score e_P.
- New rating is $R'_P = R_P + K \cdot (s_P - e_P)$ where s_P is $P's$ actual score and the factor K is set by policy (e.g. $K = 10$ for established players but $K = 40$ for young/novice/rapidly improving ones).
- Since only differences matter, absolute rating numbers are arbitrary.
- *FiveThirtyEight* centers on 1500 and rated Golden State at 1850, Cavaliers at 1691 before the NBA Finals began: 28.6% chance for Cavs per game, about 11% for 7-game series.

Expectation Curve for Elo Differences



Source: http://www.mrscienceshow.com/2009/06/sumo-vs-chess-how-their-ranking-systems.html

Chess Ratings and “Human Depth”

Chess Ratings and "Human Depth"

- **600**: Adult beginner (scholastics go under 100. . .)

Chess Ratings and "Human Depth"

- **600**: Adult beginner (scholastics go under 100. . .)
- **1000**: Minimum FIDE rating, beginning tournament player.

Chess Ratings and “Human Depth”

- **600**: Adult beginner (scholastics go under 100. . .)
- **1000**: Minimum FIDE rating, beginning tournament player.
- **1500**: Solid club player.

Chess Ratings and "Human Depth"

- **600**: Adult beginner (scholastics go under 100. . .)
- **1000**: Minimum FIDE rating, beginning tournament player.
- **1500**: Solid club player.
- **2000**: Expert.

Chess Ratings and "Human Depth"

- **600**: Adult beginner (scholastics go under 100...)
- **1000**: Minimum FIDE rating, beginning tournament player.
- **1500**: Solid club player.
- **2000**: Expert.
- **2200**: Master.

Chess Ratings and “Human Depth”

- **600**: Adult beginner (scholastics go under 100. . .)
- **1000**: Minimum FIDE rating, beginning tournament player.
- **1500**: Solid club player.
- **2000**: Expert.
- **2200**: Master.
- **2500**: Typical Grandmaster.

Chess Ratings and “Human Depth”

- **600**: Adult beginner (scholastics go under 100. . .)
- **1000**: Minimum FIDE rating, beginning tournament player.
- **1500**: Solid club player.
- **2000**: Expert.
- **2200**: Master.
- **2500**: Typical Grandmaster.
- **2800**: Human championship level.

Chess Ratings and "Human Depth"

- **600**: Adult beginner (scholastics go under 100. . .)
- **1000**: Minimum FIDE rating, beginning tournament player.
- **1500**: Solid club player.
- **2000**: Expert.
- **2200**: Master.
- **2500**: Typical Grandmaster.
- **2800**: Human championship level.
- **3200**: Exceeded by today's best programs on commodity PCs.

Chess Ratings and "Human Depth"

- **600**: Adult beginner (scholastics go under 100…)
- **1000**: Minimum FIDE rating, beginning tournament player.
- **1500**: Solid club player.
- **2000**: Expert.
- **2200**: Master.
- **2500**: Typical Grandmaster.
- **2800**: Human championship level.
- **3200**: Exceeded by today's best programs on commodity PCs.
- **3400-3500**: Ceiling of perfect play??

László Mérő, *Ways of Thinking* (1990): Chess has *human depth* of 11 (or 14) *class units* of 200 Elo, 14 (or 17) including computers.

Programs for Chess and Other Games

$\boxed{\text{Game Representation} + \text{Evaluation} + \text{Search}}$

Programs for Chess and Other Games

$$\boxed{\text{Game Representation} + \text{Evaluation} + \text{Search}}$$

- Game Rep.: Hardware advances and software tricks.

Programs for Chess and Other Games

$\boxed{\text{Game Representation} + \text{Evaluation} + \text{Search}}$

- Game Rep.: Hardware advances and software tricks.
- Base evaluation $e_0(p)$ for each position p.

Programs for Chess and Other Games

$\boxed{\text{Game Representation + Evaluation + Search}}$

- Game Rep.: Hardware advances and software tricks.
- Base evaluation $e_0(p)$ for each position p.
- Typically linear: $\sum_j w_j$(value of factor j).

Programs for Chess and Other Games

$\boxed{\text{Game Representation + Evaluation + Search}}$

- Game Rep.: Hardware advances and software tricks.
- Base evaluation $e_0(p)$ for each position p.
- Typically linear: $\sum_j w_j(\text{value of factor } j)$.
- Factors begin with 1 for each pawn, 3+ for Knight, 3++ for Bishop, 5 per Rook, 9 (or 10 or. . .) for the Queen, then go into many "positional" elements.

Programs for Chess and Other Games

$\boxed{\text{Game Representation} + \text{Evaluation} + \text{Search}}$

- Game Rep.: Hardware advances and software tricks.
- Base evaluation $e_0(p)$ for each position p.
- Typically linear: $\sum_j w_j$(value of factor j).
- Factors begin with 1 for each pawn, 3+ for Knight, 3++ for Bishop, 5 per Rook, 9 (or 10 or. . .) for the Queen, then go into many "positional" elements.
- Weights w_i now automatedly "tuned" by extensive game testing.

Programs for Chess and Other Games

$$\boxed{\text{Game Representation} + \text{Evaluation} + \text{Search}}$$

- Game Rep.: Hardware advances and software tricks.
- Base evaluation $e_0(p)$ for each position p.
- Typically linear: $\sum_j w_j$(value of factor j).
- Factors begin with 1 for each pawn, 3+ for Knight, 3++ for Bishop, 5 per Rook, 9 (or 10 or...) for the Queen, then go into many "positional" elements.
- Weights w_i now automatedly "tuned" by extensive game testing.
- Eval in **discrete** units of 0.01 called *centipawns*.

Programs for Chess and Other Games

$$\boxed{\text{Game Representation} + \text{Evaluation} + \text{Search}}$$

- Game Rep.: Hardware advances and software tricks.
- Base evaluation $e_0(p)$ for each position p.
- Typically linear: $\sum_j w_j(\text{value of factor } j)$.
- Factors begin with 1 for each pawn, 3+ for Knight, 3++ for Bishop, 5 per Rook, 9 (or 10 or. . .) for the Queen, then go into many "positional" elements.
- Weights w_i now automatedly "tuned" by extensive game testing.
- Eval in **discrete** units of 0.01 called *centipawns*.
- *Minimax* search: $e_d(p) = \max_{i \leq \ell(p)} e_{d-1}(p[m_i])$.

Programs for Chess and Other Games

$$\boxed{\text{Game Representation} + \text{Evaluation} + \text{Search}}$$

- Game Rep.: Hardware advances and software tricks.
- Base evaluation $e_0(p)$ for each position p.
- Typically linear: $\sum_j w_j(\text{value of factor } j)$.
- Factors begin with 1 for each pawn, 3+ for Knight, 3++ for Bishop, 5 per Rook, 9 (or 10 or. . .) for the Queen, then go into many "positional" elements.
- Weights w_i now automatedly "tuned" by extensive game testing.
- Eval in **discrete** units of 0.01 called *centipawns*.
- *Minimax* search: $e_d(p) = \max_{i \leq \ell(p)} e_{d-1}(p[m_i])$.
- Negate eval for opponent's view and recurse: *negamax* search.

Programs for Chess and Other Games

$\boxed{\text{Game Representation + Evaluation + Search}}$

- Game Rep.: Hardware advances and software tricks.
- Base evaluation $e_0(p)$ for each position p.
- Typically linear: $\sum_j w_j(\text{value of factor } j)$.
- Factors begin with 1 for each pawn, 3+ for Knight, 3++ for Bishop, 5 per Rook, 9 (or 10 or...) for the Queen, then go into many "positional" elements.
- Weights w_i now automatedly "tuned" by extensive game testing.
- Eval in **discrete** units of 0.01 called *centipawns*.
- *Minimax* search: $e_d(p) = \max_{i \leq \ell(p)} e_{d-1}(p[m_i])$.
- Negate eval for opponent's view and recurse: *negamax* search.
- *Basic branching factor* $\ell \approx 35$ legal moves on average.

Sound Search Principles

Sound Search Principles

- If we already know an opponent reply n_2 to move m_2 that makes $e_{d-1}(p[m_2]) < e_{d-1}(p[m_1])$, then no need to search any other replies to m_2.

Sound Search Principles

- If we already know an opponent reply n_2 to move m_2 that makes $e_{d-1}(p[m_2]) < e_{d-1}(p[m_1])$, then no need to search any other replies to m_2.
- We need not be precise about values far from $v = e_d(p)$.

Sound Search Principles

- If we already know an opponent reply n_2 to move m_2 that makes $e_{d-1}(p[m_2]) < e_{d-1}(p[m_1])$, then no need to search any other replies to m_2.
- We need not be precise about values far from $v = e_d(p)$.
- Hence we can save by guessing not just v but a *window* $\alpha < v < \beta$ around v, using "$< \alpha$" and "$> \beta$" as boundary "cutoff" values.

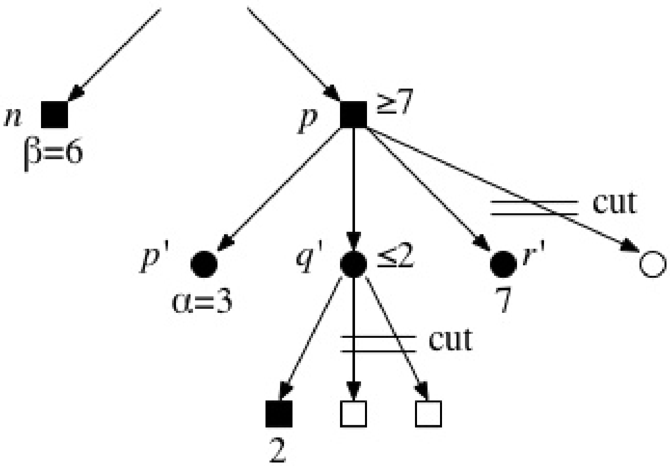
Sound Search Principles

- If we already know an opponent reply n_2 to move m_2 that makes $e_{d-1}(p[m_2]) < e_{d-1}(p[m_1])$, then no need to search any other replies to m_2.
- We need not be precise about values far from $v = e_d(p)$.
- Hence we can save by guessing not just v but a *window* $\alpha < v < \beta$ around v, using "$< \alpha$" and "$> \beta$" as boundary "cutoff" values.
- If we guess wrong and it appears $v < \alpha$ ("fail low") or $v > \beta$ ("fail high"), widen the window and start over.

Sound Search Principles

- If we already know an opponent reply n_2 to move m_2 that makes $e_{d-1}(p[m_2]) < e_{d-1}(p[m_1])$, then no need to search any other replies to m_2.
- We need not be precise about values far from $v = e_d(p)$.
- Hence we can save by guessing not just v but a *window* $\alpha < v < \beta$ around v, using "$< \alpha$" and "$> \beta$" as boundary "cutoff" values.
- If we guess wrong and it appears $v < \alpha$ ("fail low") or $v > \beta$ ("fail high"), widen the window and start over.
- Successful α-β *pruning* reduces branching factor to $\approx \sqrt{\ell}$.

Alpha-Beta Search—Diagram



Iterative Deepening

Iterative Deepening

- Work in *rounds of search* $d = 1, 2, 3, \ldots$

Iterative Deepening

- Work in *rounds of search* $d = 1, 2, 3, \ldots$
- Use *rankngs* of moves at $d - 1$ to optimize α-β pruning: "try the best moves first."

Iterative Deepening

- Work in *rounds of search* $d = 1, 2, 3, \ldots$
- Use *ranknngs* of moves at $d - 1$ to optimize α-β pruning: "try the best moves first."
- Use *value* v_{d-1} as best guess for v_d to center the window.

Iterative Deepening

- Work in *rounds of search* $d = 1, 2, 3, \ldots$
- Use *rankngs* of moves at $d - 1$ to optimize α-β pruning: "try the best moves first."
- Use *value* v_{d-1} as best guess for v_d to center the window.
- *Extend* search to depths $D > d$ along lines of play that have checks and captures and/or moves that are *singular* (meaning next-best move is much worse).

Iterative Deepening

- Work in *rounds of search* $d = 1, 2, 3, \ldots$
- Use *ranking*s of moves at $d - 1$ to optimize α-β pruning: "try the best moves first."
- Use *value* v_{d-1} as best guess for v_d to center the window.
- *Extend* search to depths $D > d$ along lines of play that have checks and captures and/or moves that are *singular* (meaning next-best move is much worse).
- Stop extending when line becomes *quiescent*.

Iterative Deepening

- Work in *rounds of search* $d = 1, 2, 3, \ldots$
- Use *ranknings* of moves at $d - 1$ to optimize α-β pruning: "try the best moves first."
- Use *value* v_{d-1} as best guess for v_d to center the window.
- *Extend* search to depths $D > d$ along lines of play that have checks and captures and/or moves that are *singular* (meaning next-best move is much worse).
- Stop extending when line becomes *quiescent*.
- Each stage yields a well-defined *principal variation* (PV) along which:

$$e_d(p) = e_{d-1}(p') = \cdots = e_0(p^{(D)}).$$

Iterative Deepening

- Work in *rounds of search* $d = 1, 2, 3, \ldots$
- Use *ranknsgs* of moves at $d - 1$ to optimize α-β pruning: "try the best moves first."
- Use *value* v_{d-1} as best guess for v_d to center the window.
- *Extend* search to depths $D > d$ along lines of play that have checks and captures and/or moves that are *singular* (meaning next-best move is much worse).
- Stop extending when line becomes *quiescent*.
- Each stage yields a well-defined *principal variation* (PV) along which:

$$e_d(p) = e_{d-1}(p') = \cdots = e_0(p^{(D)}).$$

- Stop when time budget dictates making a move.

Iterative Deepening

- Work in *rounds of search* $d = 1, 2, 3, \ldots$
- Use *ranknqs* of moves at $d - 1$ to optimize α-β pruning: "try the best moves first."
- Use *value* v_{d-1} as best guess for v_d to center the window.
- *Extend* search to depths $D > d$ along lines of play that have checks and captures and/or moves that are *singular* (meaning next-best move is much worse).
- Stop extending when line becomes *quiescent*.
- Each stage yields a well-defined *principal variation* (PV) along which:

$$e_d(p) = e_{d-1}(p') = \cdots = e_0(p^{(D)}).$$

- Stop when time budget dictates making a move.
- Values $v_1, v_2, v_3, \ldots, v_d, \ldots$ converge to "true value."

“Soundy” Search Principles

"Soundy" Search Principles

- Often one can "prove" cutoffs faster by letting the other player make two moves in a row.

"Soundy" Search Principles

- Often one can "prove" cutoffs faster by letting the other player make two moves in a row.
- Unsound for *Zugzwang* positions (where you want your opponent not you to have to move), but there are smart ways to avoid being fooled by them.

"Soundy" Search Principles

- Often one can "prove" cutoffs faster by letting the other player make two moves in a row.
- Unsound for *Zugzwang* positions (where you want your opponent not you to have to move), but there are smart ways to avoid being fooled by them.
- Evaluate inferior moves only to depth $c \ll d$.

"Soundy" Search Principles

- Often one can "prove" cutoffs faster by letting the other player make two moves in a row.
- Unsound for *Zugzwang* positions (where you want your opponent not you to have to move), but there are smart ways to avoid being fooled by them.
- Evaluate inferior moves only to depth $c \ll d$.
- These "Null Move" and "Late Move" reduction heuristics do the most to reduce the *operatioal branching factor* to about 1.5–1.6(!)

“Soundy” Search Principles

- Often one can “prove” cutoffs faster by letting the other player make two moves in a row.
- Unsound for *Zugzwang* positions (where you want your opponent not you to have to move), but there are smart ways to avoid being fooled by them.
- Evaluate inferior moves only to depth $c \ll d$.
- These “Null Move” and “Late Move” reduction heuristics do the most to reduce the *operatioal branching factor* to about 1.5–1.6(!)
- Note: $1.55^{40} \approx 6^{10} =$ only about 60 million(!)

"Soundy" Search Principles

- Often one can "prove" cutoffs faster by letting the other player make two moves in a row.
- Unsound for *Zugzwang* positions (where you want your opponent not you to have to move), but there are smart ways to avoid being fooled by them.
- Evaluate inferior moves only to depth $c \ll d$.
- These "Null Move" and "Late Move" reduction heuristics do the most to reduce the *operatioal branching factor* to about 1.5–1.6(!)
- Note: $1.55^{40} \approx 6^{10} =$ only about 60 million(!)
- The champion program Stockfish 8 reaches depth 40 within an hour on my laptop.

“Soundy” Search Principles

- Often one can “prove” cutoffs faster by letting the other player make two moves in a row.
- Unsound for *Zugzwang* positions (where you want your opponent not you to have to move), but there are smart ways to avoid being fooled by them.
- Evaluate inferior moves only to depth $c \ll d$.
- These “Null Move” and “Late Move” reduction heuristics do the most to reduce the *operatioal branching factor* to about 1.5–1.6(!)
- Note: $1.55^{40} \approx 6^{10}$ = only about 60 million(!)
- The champion program Stockfish 8 reaches depth 40 within an hour on my laptop.
- Nominal depth d really a mix of depth c and depth D; actual visited nodes are mostly wrapped around the PV. **How effective?**

The Logistic Law...

What percentage e of points do human players (of a given rating R) score from positions that a program gives value v?

Answer:

$$e \approx \frac{1}{1 + e^{-Bv}},$$

where B depends on R.

The Logistic Law...

What percentage e of points do human players (of a given rating R) score from positions that a program gives value v?

Answer:

$$e \approx \frac{1}{1 + e^{-Bv}},$$

where B depends on R.

Exact fit to $A + \frac{1-2A}{1+\exp(-Bv)}$ where A is small; A represents the chance of missing a checkmate or otherwise blowing a “completely winning”game.

The Logistic Law. . .

What percentage e of points do human players (of a given rating R) score from positions that a program gives value v?

Answer:

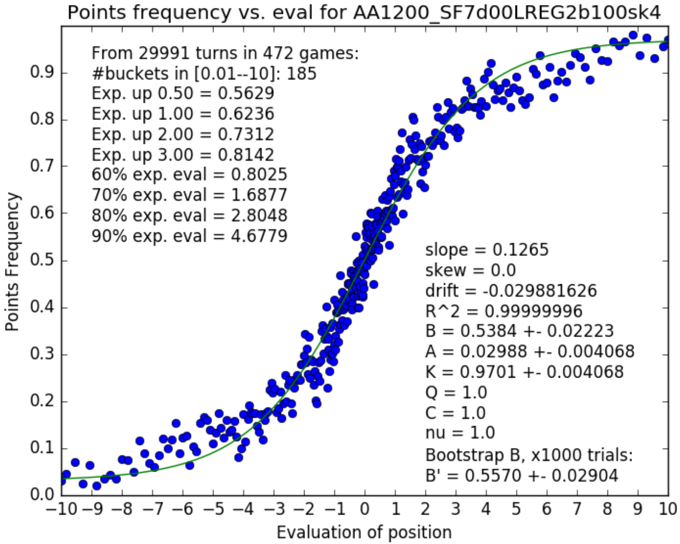
$$e \approx \frac{1}{1 + e^{-Bv}},$$

where B depends on R.

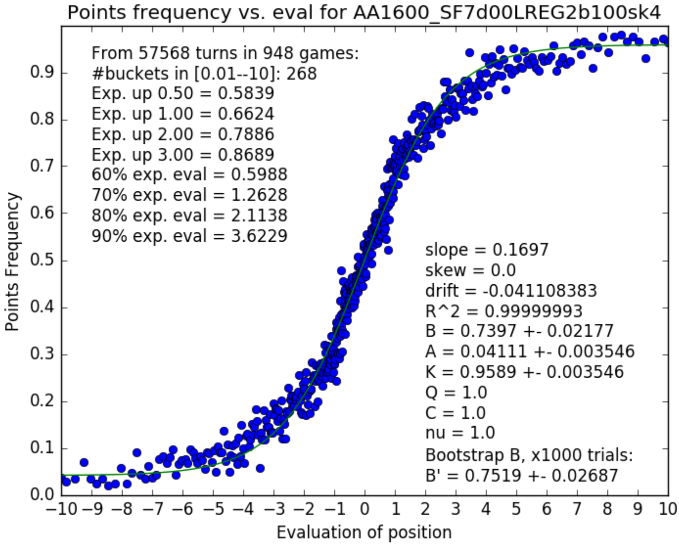
Exact fit to $A + \frac{1-2A}{1+\exp(-Bv)}$ where A is small; A represents the chance of missing a checkmate or otherwise blowing a "completely winning"game.

Data from all available games at standard time controls with both players rated within 10 (or 12) of an Elo quarter-century point **1025**, **1050**, **1075**, **1100**, . . . , **2800**. From 1,000s to 100,000s of positions in each group, just over 3 million positions total.

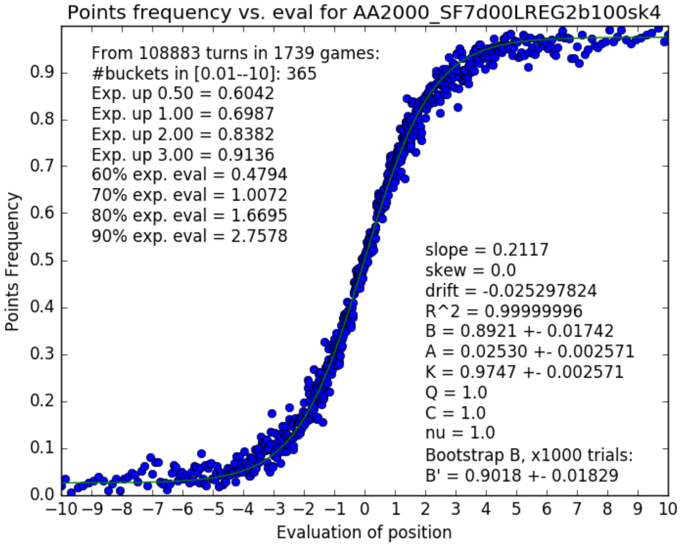
Example: Elo 1200



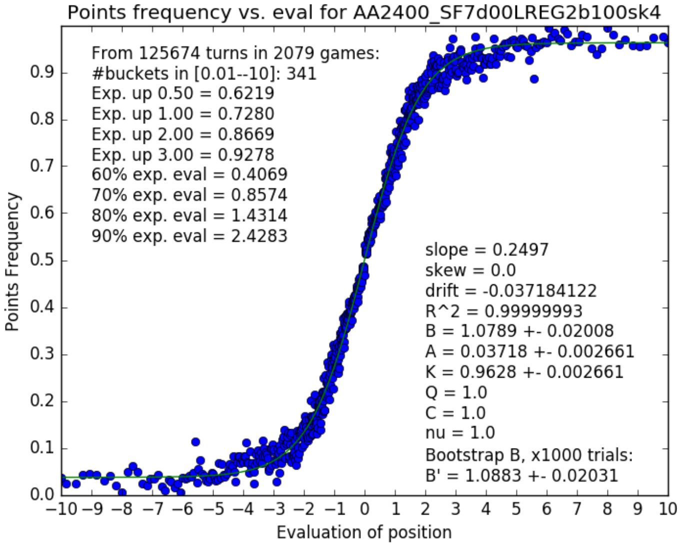
Example: Elo 1600



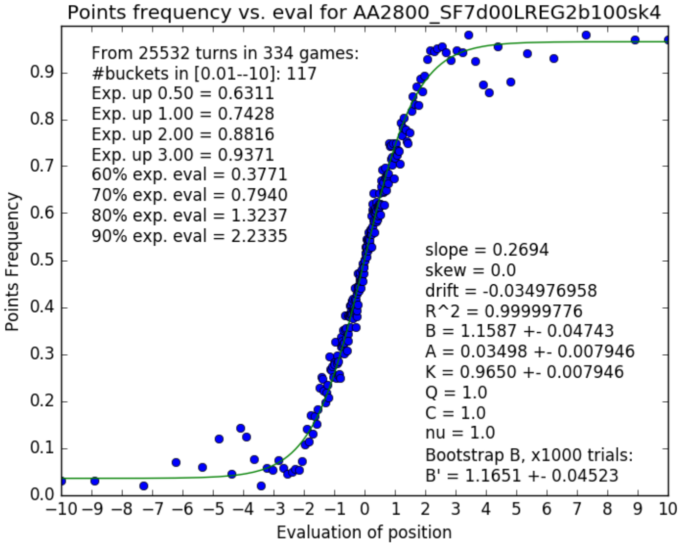
Example: Elo 2000



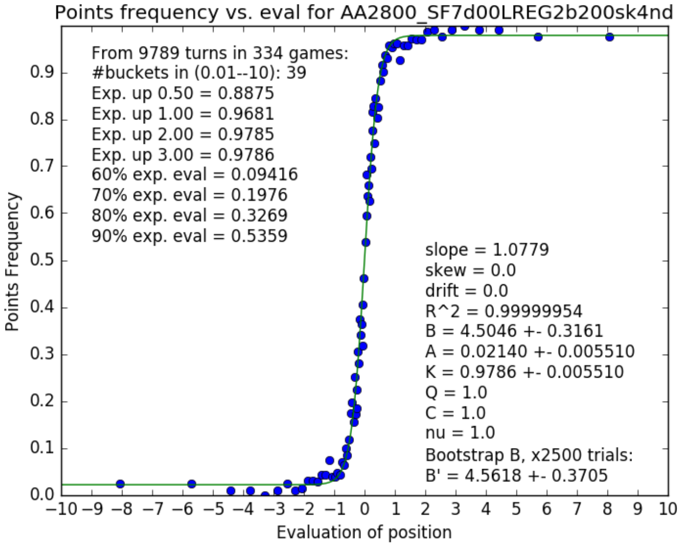
Example: Elo 2400



Example: Elo 2800



Example: Elo 2800 Ignoring Draws



Significances

Significances

1. Rated skill difference x and position value v occupy the same scale—both multiplied by B.

Significances

1. Rated skill difference x and position value v occupy the same scale—both multiplied by B.
2. For expert players, being rated 150 Elo higher is like having an extra Pawn.

Significances

1. Rated skill difference x and position value v occupy the same scale—both multiplied by B.
2. For expert players, being rated 150 Elo higher is like having an extra Pawn.
3. B has a third role as the conversion factor between engine scales.

Significances

1. Rated skill difference x and position value v occupy the same scale—both multiplied by B.
2. For expert players, being rated 150 Elo higher is like having an extra Pawn.
3. B has a third role as the conversion factor between engine scales.
    - That is, if one program values a Queen as 9 and another says 10, you might expect to convert the latter by 9/10.

Significances

1. Rated skill difference x and position value v occupy the same scale—both multiplied by B.
2. For expert players, being rated 150 Elo higher is like having an extra Pawn.
3. B has a third role as the conversion factor between engine scales.
   - That is, if one program values a Queen as 9 and another says 10, you might expect to convert the latter by 9/10.
4. Higher B for higher rating thus means we *perceive values more sharply*.

The Logistic Law . . . is Technically False

A program's behavior is unchanged under any transformation of values $e_d(m_i)$ that preserves the rank order of the moves m_i.

The Logistic Law . . . is Technically False

A program's behavior is unchanged under any transformation of values $e_d(m_i)$ that preserves the rank order of the moves m_i.

- Some commercial programs do such transformations after-the-fact.

The Logistic Law . . . is Technically False

A program's behavior is unchanged under any transformation of values $e_d(m_i)$ that preserves the rank order of the moves m_i.

- Some commercial programs do such transformations after-the-fact.
- The open-source Stockfish program does not.

The Logistic Law ... is Technically False

A program's behavior is unchanged under any transformation of values $e_d(m_i)$ that preserves the rank order of the moves m_i.

- Some commercial programs do such transformations after-the-fact.
- The open-source Stockfish program does not.
- Amir Ban, co-creator of both the chess program Deep Junior and the USB flash drive, attests that the law comes from doing things naturally and maximizes predictivity as well as playing strength for programs.

A Second Tweak to the Logistic Law

Conditioned on the position having value v from your point of view, would you rather have it be your turn to move or the opponent's?

A Second Tweak to the Logistic Law

Conditioned on the position having value v from your point of view, would you rather have it be your turn to move or the opponent's?

- The value $v ==$ the value of the best move, so it "prices in" your finding it.

A Second Tweak to the Logistic Law

Conditioned on the position having value v from your point of view, would you rather have it be your turn to move or the opponent's?

- The value $v ==$ the value of the best move, so it "prices in" your finding it.
- More crudely put, the player to move has the first chance to make a game-losing blunder.

A Second Tweak to the Logistic Law

> Conditioned on the position having value v from your point of view, would you rather have it be your turn to move or the opponent's?

- The value $v ==$ the value of the best move, so it "prices in" your finding it.
- More crudely put, the player to move has the first chance to make a game-losing blunder.
- Measured difference of 3–4% in expectation.

A Second Tweak to the Logistic Law

> Conditioned on the position having value v from your point of view, would you rather have it be your turn to move or the opponent's?

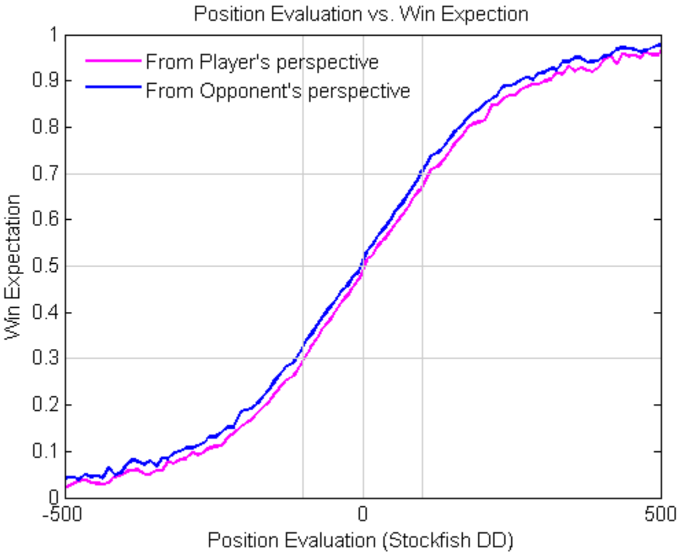
- The value $v ==$ the value of the best move, so it "prices in" your finding it.
- More crudely put, the player to move has the first chance to make a game-losing blunder.
- Measured difference of 3–4% in expectation.
- The curves you saw were symmetrized by including both player-to-move and opponent-to-move data points.

A Second Tweak to the Logistic Law

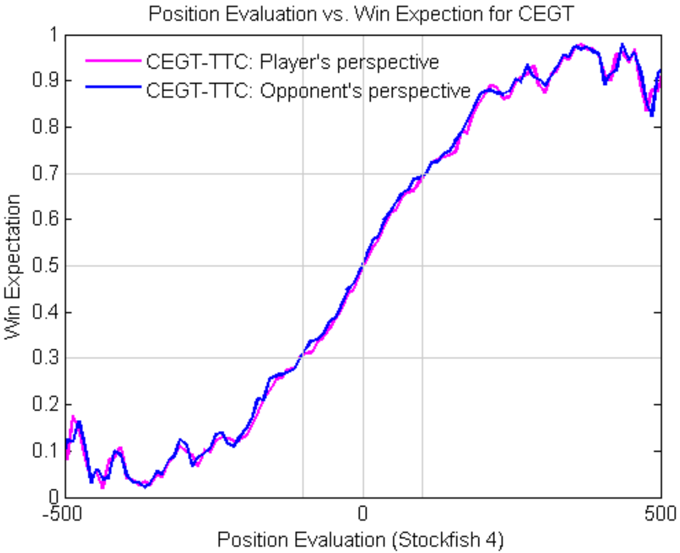
Conditioned on the position having value v from your point of view, would you rather have it be your turn to move or the opponent's?

- The value v == the value of the best move, so it "prices in" your finding it.
- More crudely put, the player to move has the first chance to make a game-losing blunder.
- Measured difference of 3–4% in expectation.
- The curves you saw were symmetrized by including both player-to-move and opponent-to-move data points.
- GM Savielly Tartakover (Polish: Ksawey Tartakower, born in Rostov-on-Don): "The game is won by the player who makes the next-to-last blunder."

Tartakover's Dictum. . .



. . . Is Not True for Computers



History of Computer Chess – Part 2

History of Computer Chess – Part 2

- **1997**: Deep Blue abruptly retires.

History of Computer Chess – Part 2

- **1997**: Deep Blue abruptly retires.
- **1998**: Kasparov says, “if you can’t beat ’em, join ’em” and promotes *Advanced Chess* where players team with one computer.

History of Computer Chess – Part 2

- **1997**: Deep Blue abruptly retires.
- **1998**: Kasparov says, “if you can’t beat ’em, join ’em” and promotes *Advanced Chess* where players team with one computer.
- (*Freestyle Chess* allows any number of computers; majpr events sponsored in 2005–2008 and 2014.)

History of Computer Chess – Part 2

- **1997**: Deep Blue abruptly retires.
- **1998**: Kasparov says, “if you can’t beat ’em, join ’em” and promotes *Advanced Chess* where players team with one computer.
- (*Freestyle Chess* allows any number of computers; majpr events sponsored in 2005–2008 and 2014.)
- **1999–2003**: Smaller systems beat GMs but only tie with Kasparov and later World Champion Viswanathan Anand.

History of Computer Chess – Part 2

- **1997**: Deep Blue abruptly retires.
- **1998**: Kasparov says, “if you can’t beat ’em, join ’em” and promotes *Advanced Chess* where players team with one computer.
- (*Freestyle Chess* allows any number of computers; majpr events sponsored in 2005–2008 and 2014.)
- **1999–2003**: Smaller systems beat GMs but only tie with Kasparov and later World Champion Viswanathan Anand.
- **2004-2005**: Fritz, Deep Junior, and massively parallel Hydra beat WC Challenger class players 16.5-7.5 in two Bilbao Human-Computer tournaments.

History of Computer Chess – Part 2

- **1997**: Deep Blue abruptly retires.
- **1998**: Kasparov says, "if you can't beat 'em, join 'em" and promotes *Advanced Chess* where players team with one computer.
- (*Freestyle Chess* allows any number of computers; majpr events sponsored in 2005–2008 and 2014.)
- **1999–2003**: Smaller systems beat GMs but only tie with Kasparov and later World Champion Viswanathan Anand.
- **2004-2005**: Fritz, Deep Junior, and massively parallel Hydra beat WC Challenger class players 16.5-7.5 in two Bilbao Human-Computer tournaments.
- **2005**: Souped-up Hydra crushes GM Michael Adams 5.5-0.5.

History of Computer Chess – Part 2

- **1997**: Deep Blue abruptly retires.
- **1998**: Kasparov says, “if you can’t beat ’em, join ’em” and promotes *Advanced Chess* where players team with one computer.
- (*Freestyle Chess* allows any number of computers; majpr events sponsored in 2005–2008 and 2014.)
- **1999–2003**: Smaller systems beat GMs but only tie with Kasparov and later World Champion Viswanathan Anand.
- **2004-2005**: Fritz, Deep Junior, and massively parallel Hydra beat WC Challenger class players 16.5-7.5 in two Bilbao Human-Computer tournaments.
- **2005**: Souped-up Hydra crushes GM Michael Adams 5.5-0.5.
- **2006**: WC Vladimir Kramnik loses to Deep Fritz 10 on ordinary quad-core PC by 4-2; he overlooks Mate-in-1 in one game.

No human GM has played a computer on even terms in a sponsored match since then.

History of Computer Chess – Part Deux

History of Computer Chess – Part Deux

- **2006**: GM Veselin Topalov accuses Kramnik of getting moves from Fritz 9 by Internet cable to his toilet—the only off-camera part of their 2006 WC match milieu.

History of Computer Chess – Part Deux

- **2006**: GM Veselin Topalov accuses Kramnik of getting moves from Fritz 9 by Internet cable to his toilet—the only off-camera part of their 2006 WC match milieu.
  - Only evidence given was alleged too-high “coincidence rates” of Kramnik’s moves with those liked by Fritz 9.

History of Computer Chess – Part Deux

- **2006**: GM Veselin Topalov accuses Kramnik of getting moves from Fritz 9 by Internet cable to his toilet—the only off-camera part of their 2006 WC match milieu.
  - Only evidence given was alleged too-high "coincidence rates" of Kramnik's moves with those liked by Fritz 9.
  - Frederic Friedel, co-founder of Fritz maker ChessBase: "Can anyone help us evaluate such statistical accusations?" $\rightarrow$ my involvement.

History of Computer Chess – Part Deux

- **2006**: GM Veselin Topalov accuses Kramnik of getting moves from Fritz 9 by Internet cable to his toilet—the only off-camera part of their 2006 WC match milieu.
    - Only evidence given was alleged too-high "coincidence rates" of Kramnik's moves with those liked by Fritz 9.
    - Frederic Friedel, co-founder of Fritz maker ChessBase: "Can anyone help us evaluate such statistical accusations?" → my involvement.
- **2009**: Smartphone "Pocket Fritz" measured at 2900+ performance crushing 2250-level human players 9.5–0.5.

History of Computer Chess – Part Deux

- **2006**: GM Veselin Topalov accuses Kramnik of getting moves from Fritz 9 by Internet cable to his toilet—the only off-camera part of their 2006 WC match milieu.
  - Only evidence given was alleged too-high "coincidence rates" of Kramnik's moves with those liked by Fritz 9.
  - Frederic Friedel, co-founder of Fritz maker ChessBase: "Can anyone help us evaluate such statistical accusations?" → my involvement.
- **2009**: Smartphone "Pocket Fritz" measured at 2900+ performance crushing 2250-level human players 9.5–0.5.
- **2010**: First later-proven case involving top-100 player.

History of Computer Chess – Part Deux

- **2006**: GM Veselin Topalov accuses Kramnik of getting moves from Fritz 9 by Internet cable to his toilet—the only off-camera part of their 2006 WC match milieu.
  - Only evidence given was alleged too-high "coincidence rates" of Kramnik's moves with those liked by Fritz 9.
  - Frederic Friedel, co-founder of Fritz maker ChessBase: "Can anyone help us evaluate such statistical accusations?" $\rightarrow$ my involvement.
- **2009**: Smartphone "Pocket Fritz" measured at 2900+ performance crushing 2250-level human players 9.5–0.5.
- **2010**: First later-proven case involving top-100 player.
- **2012-13**: Borislav Ivanov produced my first-ever **z-score** above 3.5. It was > 5.5. Higgs Boson declared discovered at $z = 5.1$.

History of Computer Chess – Part Deux

- **2006**: GM Veselin Topalov accuses Kramnik of getting moves from Fritz 9 by Internet cable to his toilet—the only off-camera part of their 2006 WC match milieu.
  - Only evidence given was alleged too-high "coincidence rates" of Kramnik's moves with those liked by Fritz 9.
  - Frederic Friedel, co-founder of Fritz maker ChessBase: "Can anyone help us evaluate such statistical accusations?" $\rightarrow$ my involvement.
- **2009**: Smartphone "Pocket Fritz" measured at 2900+ performance crushing 2250-level human players 9.5–0.5.
- **2010**: First later-proven case involving top-100 player.
- **2012-13**: Borislav Ivanov produced my first-ever **z-score** above 3.5. It was > 5.5. Higgs Boson declared discovered at $z = 5.1$.
- **2013**: FIDE formed Anti-Cheating Commission.

History of Computer Chess – Part Deux

- **2006**: GM Veselin Topalov accuses Kramnik of getting moves from Fritz 9 by Internet cable to his toilet—the only off-camera part of their 2006 WC match milieu.
  - Only evidence given was alleged too-high "coincidence rates" of Kramnik's moves with those liked by Fritz 9.
  - Frederic Friedel, co-founder of Fritz maker ChessBase: "Can anyone help us evaluate such statistical accusations?" $\rightarrow$ my involvement.
- **2009**: Smartphone "Pocket Fritz" measured at 2900+ performance crushing 2250-level human players 9.5–0.5.
- **2010**: First later-proven case involving top-100 player.
- **2012-13**: Borislav Ivanov produced my first-ever **z-score** above 3.5. It was > 5.5. Higgs Boson declared discovered at $z = 5.1$.
- **2013**: FIDE formed Anti-Cheating Commission.
- **2014–2017**: More cases, including players caught stashing smartphones in toilet stalls.

Fantastic CS Success Story

Fantastic CS Success Story

- Chess is a *hard problem*. Narrowly defined but needs broad resources.

Fantastic CS Success Story

- Chess is a *hard problem*. Narrowly defined but needs broad resources.
- Advances in hardware first.

Fantastic CS Success Story

- Chess is a *hard problem*. Narrowly defined but needs broad resources.
- Advances in hardware first.
- Later trumped by advances in software.

Fantastic CS Success Story

- Chess is a *hard problem*. Narrowly defined but needs broad resources.
- Advances in hardware first.
- Later trumped by advances in software.
    - Albert Silver 2014 experiment: Komodo 8 on smartphone trounced 2006 leader Shredder 9 on hardware 50 times faster.

Fantastic CS Success Story

- Chess is a *hard problem*. Narrowly defined but needs broad resources.
- Advances in hardware first.
- Later trumped by advances in software.
    - Albert Silver 2014 experiment: Komodo 8 on smartphone trounced 2006 leader Shredder 9 on hardware 50 times faster.
- Still not emulating the human mind...

Fantastic CS Success Story

- Chess is a *hard problem*. Narrowly defined but needs broad resources.
- Advances in hardware first.
- Later trumped by advances in software.
  - Albert Silver 2014 experiment: Komodo 8 on smartphone trounced 2006 leader Shredder 9 on hardware 50 times faster.
- Still not emulating the human mind...
- But powerful enough to "scope" players' minds...

Fantastic CS Success Story

- Chess is a *hard problem*. Narrowly defined but needs broad resources.
- Advances in hardware first.
- Later trumped by advances in software.
  - Albert Silver 2014 experiment: Komodo 8 on smartphone trounced 2006 leader Shredder 9 on hardware 50 times faster.
- Still not emulating the human mind. . .
- But powerful enough to "scope" players' minds. . .
- . . . aided by *acuity* in modeling.

Predictive Models

Given data and analysis on potential events $E_1, \ldots, E_L$ estimate probabilities $p_1, \ldots, p_L$ for them to occur.

Examples:

Predictive Models

Given data and analysis on potential events $E_1, \ldots, E_L$ estimate probabilities $p_1, \ldots, p_L$ for them to occur.

Examples:

- Some of the events $E_1, \ldots, E_m$ are natural disasters.

Predictive Models

Given data and analysis on potential events $E_1, \ldots, E_L$ estimate probabilities $p_1, \ldots, p_L$ for them to occur.

Examples:

- Some of the events $E_1, \ldots, E_m$ are natural disasters.
- $E_1, \ldots, E_L$ are potential courses that a disease can take.

Predictive Models

Given data and analysis on potential events $E_1, \ldots, E_L$ estimate probabilities $p_1, \ldots, p_L$ for them to occur.

Examples:

- Some of the events $E_1, \ldots, E_m$ are natural disasters.
- $E_1, \ldots, E_L$ are potential courses that a disease can take.
- The events are correct answers on an exam with L questions, and we want to estimate the distribution of results.

Predictive Models

Given data and analysis on potential events $E_1, \ldots, E_L$ estimate probabilities $p_1, \ldots, p_L$ for them to occur.

Examples:

- Some of the events $E_1, \ldots, E_m$ are natural disasters.
- $E_1, \ldots, E_L$ are potential courses that a disease can take.
- The events are correct answers on an exam with L questions, and we want to estimate the distribution of results.
- The events are the legal moves in a chess position.

Predictive Models

Given data and analysis on potential events $E_1, \ldots, E_L$ estimate probabilities $p_1, \ldots, p_L$ for them to occur.

Examples:

- Some of the events $E_1, \ldots, E_m$ are natural disasters.
- $E_1, \ldots, E_L$ are potential courses that a disease can take.
- The events are correct answers on an exam with L questions, and we want to estimate the distribution of results.
- The events are the legal moves in a chess position. They are *mutually exclusive* and (together with "draw" or "resign") *collectively exhaustive*: $\sum_i p_i = 1$.

Predictive Models

Given data and analysis on potential events $E_1, \ldots, E_L$ estimate probabilities $p_1, \ldots, p_L$ for them to occur.

Examples:

- Some of the events $E_1, \ldots, E_m$ are natural disasters.
- $E_1, \ldots, E_L$ are potential courses that a disease can take.
- The events are correct answers on an exam with L questions, and we want to estimate the distribution of results.
- The events are the legal moves in a chess position. They are *mutually exclusive* and (together with "draw" or "resign") *collectively exhaustive*: $\sum_i p_i = 1$.
- *Cost* of a (non-optimal) move m_i == its difference in value $\delta_i = \delta(v_1, v_i)$ to the first move m_1.

Predictive Models

Given data and analysis on potential events $E_1, \ldots, E_L$ estimate probabilities $p_1, \ldots, p_L$ for them to occur.

Examples:

- Some of the events $E_1, \ldots, E_m$ are natural disasters.
- $E_1, \ldots, E_L$ are potential courses that a disease can take.
- The events are correct answers on an exam with L questions, and we want to estimate the distribution of results.
- The events are the legal moves in a chess position. They are *mutually exclusive* and (together with "draw" or "resign") *collectively exhaustive*: $\sum_i p_i = 1$.
- *Cost* of a (non-optimal) move m_i == its difference in value $\delta_i = \delta(v_1, v_i)$ to the first move m_1.
- Predicted cost: $\sum_{i=1}^{\ell} p_i \delta_i$. *Scaled* down when $|v_1|$ is high.

Inputs and Outputs

Inputs and Outputs

1. Domain: A set T of decision-making situations t.
   Chess game turns

Inputs and Outputs

1. Domain: A set T of decision-making situations t.
   Chess game turns
2. Inputs: Values v_i for every option at turn t.

Inputs and Outputs

1. Domain: A set T of decision-making situations t.
   Chess game turns
2. Inputs: Values v_i for every option at turn t.
3. Parameters: $s, c, \ldots$ denoting skills and levels.

Inputs and Outputs

1. Domain: A set T of decision-making situations t.
   Chess game turns
2. Inputs: Values v_i for every option at turn t.
3. Parameters: $s, c, \ldots$ denoting skills and levels.
4. Defines *fallible agent* $P(s, c, \ldots)$.

Inputs and Outputs

1. Domain: A set T of decision-making situations t.
   Chess game turns
2. Inputs: Values v_i for every option at turn t.
3. Parameters: $s, c, \ldots$ denoting skills and levels.
4. Defines *fallible agent* $P(s, c, \ldots)$.
5. Main Output: Probabilities $p_{i,t}$ for $P(s, c, \ldots)$ to select option i at time t.

Inputs and Outputs

1. Domain: A set T of decision-making situations t.
   Chess game turns
2. Inputs: Values v_i for every option at turn t.
3. Parameters: $s, c, \ldots$ denoting skills and levels.
4. Defines *fallible agent* $P(s, c, \ldots)$.
5. Main Output: Probabilities $p_{i,t}$ for $P(s, c, \ldots)$ to select option i at time t.
6. Derived Outputs (*Aggregate Statistics*):

$$\begin{aligned}
\textbf{MM} &= \sum_t p_{1,t} && \textit{Move-Match} \\
\textbf{EV} &= \sum_t \sum_{i:\delta_{i,t}=0} p_{i,t} && \textit{Equal-top Value} \\
\textbf{ASD} &= \sum_t \sum_i p_{i,t}\delta_{i,t} && \textit{Average Scaled Difference.}
\end{aligned}$$

Obtaining the Proabilities

Obtaining the Proabilities

- Each move m_i is assigned a perceived inferiority $z_i \geq 0$.

Obtaining the Proabilities

- Each move m_i is assigned a perceived inferiority $z_i \geq 0$.
- Dimensionless, not in centipawn units like δ_i.

Obtaining the Proabilities

- Each move m_i is assigned a perceived inferiority $z_i \geq 0$.
- Dimensionless, not in centipawn units like δ_i.
- Exponential decay:

$$p_i = p_1^{g(z_i)},$$

where $g(0) = 1$, $u_i = g(z_i) \geq 1$ is the "utility share curve."

Obtaining the Proabilities

- Each move m_i is assigned a perceived inferiority $z_i \geq 0$.
- Dimensionless, not in centipawn units like δ_i.
- Exponential decay:
  $$p_i = p_1^{g(z_i)},$$
  where $g(0) = 1$, $u_i = g(z_i) \geq 1$ is the "utility share curve."
- Could be $g(z_i) = z_i + 1$ but a second layer of exponentiation works better (so far).

Obtaining the Proabilities

- Each move m_i is assigned a perceived inferiority $z_i \geq 0$.
- Dimensionless, not in centipawn units like δ_i.
- Exponential decay:

$$p_i = p_1^{g(z_i)},$$

  where $g(0) = 1$, $u_i = g(z_i) \geq 1$ is the "utility share curve."
- Could be $g(z_i) = z_i + 1$ but a second layer of exponentiation works better (so far).
- Have used $g(z) = e^z$ and $g(z) = \frac{e^z+1}{2}$; the latter makes $1/g(z)$ a "folded" logistic curve.

Obtaining the Proabilities

- Each move m_i is assigned a perceived inferiority $z_i \geq 0$.
- Dimensionless, not in centipawn units like δ_i.
- Exponential decay:
  $$p_i = p_1^{g(z_i)},$$
  where $g(0) = 1$, $u_i = g(z_i) \geq 1$ is the "utility share curve."
- Could be $g(z_i) = z_i + 1$ but a second layer of exponentiation works better (so far).
- Have used $g(z) = e^z$ and $g(z) = \frac{e^z+1}{2}$; the latter makes $1/g(z)$ a "folded" logistic curve.
- Then calculate p_1 to make $\sum_i p_i^{u_i} = 1$.

Given $u_1, \ldots, u_\ell \geq 1$, how to solve for p giving $p^{u_1} + \cdots + p^{u_\ell} = 1$? Better way than Newton?

Inferiority Main Equation

$$z_i = \left(\frac{\delta_i}{s}\right)^c$$

Inferiority Main Equation

$$z_i = \left(\frac{\delta_i}{s}\right)^c$$

- Parameters s for *sensitivity*, c for *consistency*.

Inferiority Main Equation

$$z_i = \left(\frac{\delta_i}{s}\right)^c$$

- Parameters s for *sensitivity*, c for *consistency*.
- ∂s greatest near $\delta_i = 0$; ∂c takes over for large mistakes.

Inferiority Main Equation

$$z_i = \left(\frac{\delta_i}{s} \right)^c$$

- Parameters s for *sensitivity*, c for *consistency*.
- ∂s greatest near $\delta_i = 0$; ∂c takes over for large mistakes.
- Given any sample of positions, fit s, c to make projected **MM** and **ASD** agree with the sample values.

Inferiority Main Equation

$$z_i = \left(\frac{\delta_i}{s}\right)^c$$

- Parameters s for *sensitivity*, c for *consistency*.
- ∂s greatest near $\delta_i = 0$; ∂c takes over for large mistakes.
- Given any sample of positions, fit s, c to make projected **MM** and **ASD** agree with the sample values.
- Makes **MM** and **ASD** into **unbiased estimators** (**EV** generally conservative).

Inferiority Main Equation

$$z_i = \left(\frac{\delta_i}{s}\right)^c$$

- Parameters s for *sensitivity*, c for *consistency*.
- ∂s greatest near $\delta_i = 0$; ∂c takes over for large mistakes.
- Given any sample of positions, fit s, c to make projected **MM** and **ASD** agree with the sample values.
- Makes **MM** and **ASD** into **unbiased estimators** (**EV** generally conservative).
- *Monotone* in sense that better moves always get higher probability no matter how weak the player, and an uptick in the value of a move always increases its probability.

Inferiority Main Equation

$$z_i = \left(\frac{\delta_i}{s}\right)^c$$

- Parameters s for *sensitivity*, c for *consistency*.
- ∂s greatest near $\delta_i = 0$; ∂c takes over for large mistakes.
- Given any sample of positions, fit s, c to make projected **MM** and **ASD** agree with the sample values.
- Makes **MM** and **ASD** into **unbiased estimators** (**EV** generally conservative).
- *Monotone* in sense that better moves always get higher probability no matter how weak the player, and an uptick in the value of a move always increases its probability.
- Not only yields linear relation $E = \alpha s + \beta c$ to Elo rating, but the training gives good progressions $[s_E]$ and $[c_E]$ in each parameter.

Inferiority Main Equation

$$z_i = \left(\frac{\delta_i}{s}\right)^c$$

- Parameters s for *sensitivity*, c for *consistency*.
- ∂s greatest near $\delta_i = 0$; ∂c takes over for large mistakes.
- Given any sample of positions, fit s, c to make projected **MM** and **ASD** agree with the sample values.
- Makes **MM** and **ASD** into **unbiased estimators** (**EV** generally conservative).
- *Monotone* in sense that better moves always get higher probability no matter how weak the player, and an uptick in the value of a move always increases its probability.
- Not only yields linear relation $E = \alpha s + \beta c$ to Elo rating, but the training gives good progressions $[s_E]$ and $[c_E]$ in each parameter.
- Unique fit and *Intrinsic Performance Rating* (IPR) for any set of games.

How Sensitive Are We?

Conditioned on the best move m_1 being superior to m_2 by x and one of m_1 or m_2 being played, with what frequency f_1 do **2000**-rated players prefer m_1?

How Sensitive Are We?

Conditioned on the best move m_1 being superior to m_2 by x and one of m_1 or m_2 being played, with what frequency f_1 do **2000**-rated players prefer m_1?

- $x = 0.01$,

How Sensitive Are We?

Conditioned on the best move m_1 being superior to m_2 by x and one of m_1 or m_2 being played, with what frequency f_1 do **2000**-rated players prefer m_1?

- $x = 0.01$, $f_1 = 52.85\%$.

How Sensitive Are We?

Conditioned on the best move m_1 being superior to m_2 by x and one of m_1 or m_2 being played, with what frequency f_1 do **2000**-rated players prefer m_1?

- $x = 0.01$, $f_1 = 52.85\%$.
- $x = 0.02$,

How Sensitive Are We?

Conditioned on the best move m_1 being superior to m_2 by x and one of m_1 or m_2 being played, with what frequency f_1 do **2000**-rated players prefer m_1?

- $x = 0.01$, $f_1 = 52.85\%$.
- $x = 0.02$, $f_1 = 53.83\%$.

How Sensitive Are We?

Conditioned on the best move m_1 being superior to m_2 by x and one of m_1 or m_2 being played, with what frequency f_1 do **2000**-rated players prefer m_1?

- $x = 0.01$, $f_1 = 52.85\%$.
- $x = 0.02$, $f_1 = 53.83\%$.
- $x = 0.03$,

How Sensitive Are We?

Conditioned on the best move m_1 being superior to m_2 by x and one of m_1 or m_2 being played, with what frequency f_1 do **2000**-rated players prefer m_1?

- $x = 0.01$, $f_1 = 52.85\%$.
- $x = 0.02$, $f_1 = 53.83\%$.
- $x = 0.03$, $f_1 = 56.08\%$.

How Sensitive Are We?

Conditioned on the best move m_1 being superior to m_2 by x and one of m_1 or m_2 being played, with what frequency f_1 do **2000**-rated players prefer m_1?

- $x = 0.01$, $f_1 = 52.85\%$.
- $x = 0.02$, $f_1 = 53.83\%$.
- $x = 0.03$, $f_1 = 56.08\%$.
- $x = 0.04$,

How Sensitive Are We?

Conditioned on the best move m_1 being superior to m_2 by x and one of m_1 or m_2 being played, with what frequency f_1 do **2000**-rated players prefer m_1?

- $x = 0.01$, $f_1 = 52.85\%$.
- $x = 0.02$, $f_1 = 53.83\%$.
- $x = 0.03$, $f_1 = 56.08\%$.
- $x = 0.04$, $f_1 = 56.165\%$.

How Sensitive Are We?

Conditioned on the best move m_1 being superior to m_2 by x and one of m_1 or m_2 being played, with what frequency f_1 do **2000**-rated players prefer m_1?

- $x = 0.01$, $f_1 = 52.85\%$.
- $x = 0.02$, $f_1 = 53.83\%$.
- $x = 0.03$, $f_1 = 56.08\%$.
- $x = 0.04$, $f_1 = 56.165\%$.
- $x = 0.05$,

How Sensitive Are We?

Conditioned on the best move m_1 being superior to m_2 by x and one of m_1 or m_2 being played, with what frequency f_1 do **2000**-rated players prefer m_1?

- $x = 0.01$, $f_1 = 52.85\%$.
- $x = 0.02$, $f_1 = 53.83\%$.
- $x = 0.03$, $f_1 = 56.08\%$.
- $x = 0.04$, $f_1 = 56.165\%$.
- $x = 0.05$, $f_1 = 58.28\%$.

How Sensitive Are We?

Conditioned on the best move m_1 being superior to m_2 by x and one of m_1 or m_2 being played, with what frequency f_1 do **2000**-rated players prefer m_1?

- $x = 0.01$, $f_1 = 52.85\%$.
- $x = 0.02$, $f_1 = 53.83\%$.
- $x = 0.03$, $f_1 = 56.08\%$.
- $x = 0.04$, $f_1 = 56.165\%$.
- $x = 0.05$, $f_1 = 58.28\%$.
- $x = 0.00$,

How Sensitive Are We?

Conditioned on the best move m_1 being superior to m_2 by x and one of m_1 or m_2 being played, with what frequency f_1 do **2000**-rated players prefer m_1?

- $x = 0.01$, $f_1 = 52.85\%$.
- $x = 0.02$, $f_1 = 53.83\%$.
- $x = 0.03$, $f_1 = 56.08\%$.
- $x = 0.04$, $f_1 = 56.165\%$.
- $x = 0.05$, $f_1 = 58.28\%$.
- $x = 0.00$, $f_1 = 58.72\%$.

Co?

How Sensitive Are We?

Conditioned on the best move m_1 being superior to m_2 by x and one of m_1 or m_2 being played, with what frequency f_1 do **2000**-rated players prefer m_1?

- $x = 0.01$, $f_1 = 52.85\%$.
- $x = 0.02$, $f_1 = 53.83\%$.
- $x = 0.03$, $f_1 = 56.08\%$.
- $x = 0.04$, $f_1 = 56.165\%$.
- $x = 0.05$, $f_1 = 58.28\%$.
- $x = 0.00$, $f_1 = 58.72\%$.

Co? Note: Sample sizes are 2,605–7,701 positions each, out of 140,999 positions by 2000-rated players overall.

It's an ESP Test

Same thing for **2600**-rated players, 102,472 positions overall:

It's an ESP Test

Same thing for **2600**-rated players, 102,472 positions overall:

- $x = 0.01$,

It's an ESP Test

Same thing for **2600**-rated players, 102,472 positions overall:

- $x = 0.01$, $f_1 = 54.78\%$.

It's an ESP Test

Same thing for **2600**-rated players, 102,472 positions overall:

- $x = 0.01$, $f_1 = 54.78\%$.
- $x = 0.02$,

It's an ESP Test

Same thing for **2600**-rated players, 102,472 positions overall:

- $x = 0.01$, $f_1 = 54.78\%$.
- $x = 0.02$, $f_1 = 54.64\%$.

It's an ESP Test

Same thing for **2600**-rated players, 102,472 positions overall:

- $x = 0.01$, $f_1 = 54.78\%$.
- $x = 0.02$, $f_1 = 54.64\%$.
- $x = 0.03$,

It's an ESP Test

Same thing for **2600**-rated players, 102,472 positions overall:

- $x = 0.01$, $f_1 = 54.78\%$.
- $x = 0.02$, $f_1 = 54.64\%$.
- $x = 0.03$, $f_1 = 56.99\%$.

It's an ESP Test

Same thing for **2600**-rated players, 102,472 positions overall:

- $x = 0.01$, $f_1 = 54.78\%$.
- $x = 0.02$, $f_1 = 54.64\%$.
- $x = 0.03$, $f_1 = 56.99\%$.
- $x = 0.04$,

It's an ESP Test

Same thing for **2600**-rated players, 102,472 positions overall:

- $x = 0.01$, $f_1 = 54.78\%$.
- $x = 0.02$, $f_1 = 54.64\%$.
- $x = 0.03$, $f_1 = 56.99\%$.
- $x = 0.04$, $f_1 = 57.86\%$.

It's an ESP Test

Same thing for **2600**-rated players, 102,472 positions overall:

- $x = 0.01$, $f_1 = 54.78\%$.
- $x = 0.02$, $f_1 = 54.64\%$.
- $x = 0.03$, $f_1 = 56.99\%$.
- $x = 0.04$, $f_1 = 57.86\%$.
- $x = 0.05$,

It's an ESP Test

Same thing for **2600**-rated players, 102,472 positions overall:

- $x = 0.01$, $f_1 = 54.78\%$.
- $x = 0.02$, $f_1 = 54.64\%$.
- $x = 0.03$, $f_1 = 56.99\%$.
- $x = 0.04$, $f_1 = 57.86\%$.
- $x = 0.05$, $f_1 = 61.11\%$.

It's an ESP Test

Same thing for **2600**-rated players, 102,472 positions overall:

- $x = 0.01$, $f_1 = 54.78\%$.
- $x = 0.02$, $f_1 = 54.64\%$.
- $x = 0.03$, $f_1 = 56.99\%$.
- $x = 0.04$, $f_1 = 57.86\%$.
- $x = 0.05$, $f_1 = 61.11\%$.
- $x = 0.00$?

It's an ESP Test

Same thing for **2600**-rated players, 102,472 positions overall:

- $x = 0.01$, $f_1 = 54.78\%$.
- $x = 0.02$, $f_1 = 54.64\%$.
- $x = 0.03$, $f_1 = 56.99\%$.
- $x = 0.04$, $f_1 = 57.86\%$.
- $x = 0.05$, $f_1 = 61.11\%$.
- $x = 0.00$? $f_1 = 60.22\%$.

It's an ESP Test

Same thing for **2600**-rated players, 102,472 positions overall:

- $x = 0.01$, $f_1 = 54.78\%$.
- $x = 0.02$, $f_1 = 54.64\%$.
- $x = 0.03$, $f_1 = 56.99\%$.
- $x = 0.04$, $f_1 = 57.86\%$.
- $x = 0.05$, $f_1 = 61.11\%$.
- $x = 0.00$? $f_1 = 60.22\%$.
- Last dataset has 10,611 turns with tied-optimal moves.

It's an ESP Test

Same thing for **2600**-rated players, 102,472 positions overall:

- $x = 0.01$, $f_1 = 54.78\%$.
- $x = 0.02$, $f_1 = 54.64\%$.
- $x = 0.03$, $f_1 = 56.99\%$.
- $x = 0.04$, $f_1 = 57.86\%$.
- $x = 0.05$, $f_1 = 61.11\%$.
- $x = 0.00$? $f_1 = 60.22\%$.
- Last dataset has 10,611 turns with tied-optimal moves.
- Go back all the way to 1971—when there was no Stockfish 7 program.

It's an ESP Test

Same thing for **2600**-rated players, 102,472 positions overall:

- $x = 0.01$, $f_1 = 54.78\%$.
- $x = 0.02$, $f_1 = 54.64\%$.
- $x = 0.03$, $f_1 = 56.99\%$.
- $x = 0.04$, $f_1 = 57.86\%$.
- $x = 0.05$, $f_1 = 61.11\%$.
- $x = 0.00$? $f_1 = 60.22\%$.
- Last dataset has 10,611 turns with tied-optimal moves.
- Go back all the way to 1971—when there was no Stockfish 7 program.
- Stockfish 7 would not diminish in game-playing quality at all if m_1 and m_2 were switched in those situations. How can we "precognite" which one it will list first??? An ESP test that humans pass over **60%**.

Measuring "Swing" and Complexity and Difficulty

- Non-Parapsychological Explanation:

Measuring "Swing" and Complexity and Difficulty

- Non-Parapsychological Explanation: *Stable* Library Sorting.

Measuring "Swing" and Complexity and Difficulty

- Non-Parapsychological Explanation: *Stable* Library Sorting.
- Chess engines sort moves from last depth to schedule next round of search.

Measuring "Swing" and Complexity and Difficulty

- Non-Parapsychological Explanation: *Stable* Library Sorting.
- Chess engines sort moves from last depth to schedule next round of search.
- By stability, lower move can become 1st only with *strictly higher* value.
- Lead moves tend to have been higher at lower depths. Lower move "swings up."

Measuring "Swing" and Complexity and Difficulty

- Non-Parapsychological Explanation: *Stable* Library Sorting.
- Chess engines sort moves from last depth to schedule next round of search.
- By stability, lower move can become 1st only with *strictly higher* value.
- Lead moves tend to have been higher at lower depths. Lower move "swings up."
- Formulate numerical measure ρ_i of swing "up" and "down" (a trap).

Measuring "Swing" and Complexity and Difficulty

- Non-Parapsychological Explanation: *Stable* Library Sorting.
- Chess engines sort moves from last depth to schedule next round of search.
- By stability, lower move can become 1st only with *strictly higher* value.
- Lead moves tend to have been higher at lower depths. Lower move "swings up."
- Formulate numerical measure ρ_i of swing "up" and "down" (a trap).
- When best move swings up **4.0–5.0** versus **0.0–1.0**, players rated 2700+ find it only **30%** versus **70%**.

Measuring “Swing” and Complexity and Difficulty

- Non-Parapsychological Explanation: *Stable* Library Sorting.
- Chess engines sort moves from last depth to schedule next round of search.
- By stability, lower move can become 1st only with *strictly higher* value.
- Lead moves tend to have been higher at lower depths. Lower move “swings up.”
- Formulate numerical measure ρ_i of swing “up” and “down” (a trap).
- When best move swings up **4.0–5.0** versus **0.0–1.0**, players rated 2700+ find it only **30%** versus **70%**.
- Goal is to develop a **Challenge Quotient** based on how much trappy play a player sets for the opponent

Measuring "Swing" and Complexity and Difficulty

- Non-Parapsychological Explanation: *Stable* Library Sorting.
- Chess engines sort moves from last depth to schedule next round of search.
- By stability, lower move can become 1st only with *strictly higher* value.
- Lead moves tend to have been higher at lower depths. Lower move "swings up."
- Formulate numerical measure ρ_i of swing "up" and "down" (a trap).
- When best move swings up **4.0–5.0** versus **0.0–1.0**, players rated 2700+ find it only **30%** versus **70%**.
- Goal is to develop a **Challenge Quotient** based on how much trappy play a player sets for the opponent—and emself.

Measuring “Swing” and Complexity and Difficulty

- Non-Parapsychological Explanation: *Stable* Library Sorting.
- Chess engines sort moves from last depth to schedule next round of search.
- By stability, lower move can become 1st only with *strictly higher* value.
- Lead moves tend to have been higher at lower depths. Lower move “swings up.”
- Formulate numerical measure ρ_i of swing “up” and “down” (a trap).
- When best move swings up **4.0–5.0** versus **0.0–1.0**, players rated 2700+ find it only **30%** versus **70%**.
- Goal is to develop a **Challenge Quotient** based on how much trappy play a player sets for the opponent—and emself.
- Separates *performance* and *prediction* in the model.

Example of "Swing" over Increasing Depths

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

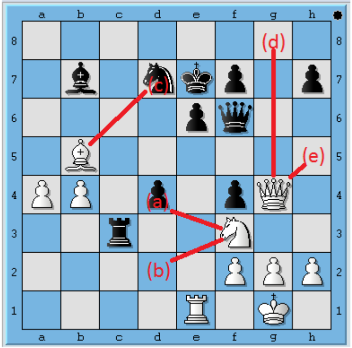
(c) proliferation . . blighted

(d) destruction . . disputed

(e) disappearance . . frustrated

(source: itunes.apple.com)

=



Move	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Nd2	103	093	087	093	027	028	000	000	056	-007	039	028	037	020	014	017	000	006	000
Bxd7	048	034	-033	-033	-013	-042	-039	-050	-025	-010	001	000	-009	-027	-018	000	000	000	000
Qg8	114	114	-037	-037	-014	-014	-022	-068	-008	-056	-042	-004	-032	000	-014	-025	-045	-045	-050
...			...			...			...			...			...			...	
Nxd4	-056	-056	-113	-071	-071	-145	-020	-006	077	052	066	040	050	051	-181	-181	-181	-213	-213

Modeling "Heave"

$$z_i' = \left(\frac{\delta_i}{s}\right)^c + \left(\frac{h \cdot \rho_i}{s}\right)^{a \cdot c}$$

Modeling “Heave”

$$z_i' = \left(\frac{\delta_i}{s}\right)^c + \left(\frac{h \cdot \rho_i}{s}\right)^{a \cdot c}$$

- Coupling h, a to s, c in the second term gives the interpretation

$$h, a > 1 \implies \rho_i \text{ is more significant than } \delta_i \ .$$

Modeling "Heave"

$$z_i' = \left(\frac{\delta_i}{s}\right)^c + \left(\frac{h \cdot \rho_i}{s}\right)^{a \cdot c}$$

- Coupling h, a to s, c in the second term gives the interpretation

$$h, a > 1 \implies \rho_i \text{ is more significant than } \delta_i \; .$$

- Often allows solving **EV** plus 1 more equation for improved fits.

Modeling "Heave"

$$z_i' = \left(\frac{\delta_i}{s}\right)^c + \left(\frac{h \cdot \rho_i}{s}\right)^{a \cdot c}$$

- Coupling h, a to s, c in the second term gives the interpretation

$$h, a > 1 \implies \rho_i \text{ is more significant than } \delta_i \,.$$

- Often allows solving **EV** plus 1 more equation for improved fits.
- But those fits usually give $h > 1.5$, **Uh-Oh!**

Big Wins for the New Model

Big Wins for the New Model

- Predicts tied-move frequencies without an *ad-hoc* patch.

Big Wins for the New Model

- Predicts tied-move frequencies without an *ad-hoc* patch.
- Fits with 4 equations often make 30 others follow…

Big Wins for the New Model

- Predicts tied-move frequencies without an *ad-hoc* patch.
- Fits with 4 equations often make 30 others follow...
- No longer strictly monotone: Weaker players may prefer weaker moves that look better at early depths, more so if they have higher h.

Big Wins for the New Model

- Predicts tied-move frequencies without an *ad-hoc* patch.
- Fits with 4 equations often make 30 others follow...
- No longer strictly monotone: Weaker players may prefer weaker moves that look better at early depths, more so if they have higher h.
- Separates prediction and performance-assessment components.

Big Wins for the New Model

- Predicts tied-move frequencies without an *ad-hoc* patch.
- Fits with 4 equations often make 30 others follow...
- No longer strictly monotone: Weaker players may prefer weaker moves that look better at early depths, more so if they have higher h.
- Separates prediction and performance-assessment components.
- Often accurately predicts inferior moves to be more likely, **But...**

Fine-Grained Trouble Under the Dial

Fine-Grained Trouble Under the Dial

- . . . **at the same time** it gives near-**zero** probability to reasonable moves **that were played.**

Fine-Grained Trouble Under the Dial

- **. . . at the same time** it gives near-**zero** probability to reasonable moves **that were played.**
- Even sometimes gives ϵ projection to the best move!

Fine-Grained Trouble Under the Dial

- **. . . at the same time** it gives near-**zero** probability to reasonable moves **that were played.**
- Even sometimes gives ϵ projection to the best move!
- [show examples from web article, "Stopped Watches and Data Analytics"]

Fine-Grained Trouble Under the Dial

- **. . . at the same time** it gives near-**zero** probability to reasonable moves **that were played.**
- Even sometimes gives ϵ projection to the best move!
- [show examples from web article, "Stopped Watches and Data Analytics"]
- So far the cause seems to be that the fit is latching on to features of ρ_i that allow it to be welded onto the frequency histogram $f_1, f_2, f_3, \ldots$.

From “Data Skeptic” to “Model Skeptic”

From "Data Skeptic" to "Model Skeptic"

- "Data Skeptic" is even the name of a podcast I once appeared on.

From “Data Skeptic” to “Model Skeptic”

- “Data Skeptic” is even the name of a podcast I once appeared on.
- Jaap van den Herik’s CISIM 2016 keynote gave a healthy dose of it.

From “Data Skeptic” to “Model Skeptic”

- “Data Skeptic” is even the name of a podcast I once appeared on.
- Jaap van den Herik’s CISIM 2016 keynote gave a healthy dose of it.
- “Model Skpetic” is represented by Cathy O’Neil’s book *Weapons of Math Destruction*.

From "Data Skeptic" to "Model Skeptic"

- "Data Skeptic" is even the name of a podcast I once appeared on.
- Jaap van den Herik's CISIM 2016 keynote gave a healthy dose of it.
- "Model Skpetic" is represented by Cathy O'Neil's book *Weapons of Math Destruction*.
- And by the University of Washington—Seatle course `http://callingbullshit.org/`.

From Jaap van den Herik's CISIM 2016 Keynote

"In data science we nowadays distinguish seven phases of activities:

From Jaap van den Herik's CISIM 2016 Keynote

"In data science we nowadays distinguish seven phases of activities:

1. collecting data,

From Jaap van den Herik's CISIM 2016 Keynote

"In data science we nowadays distinguish seven phases of activities:

1. collecting data,
2. cleaning data,

From Jaap van den Herik's CISIM 2016 Keynote

"In data science we nowadays distinguish seven phases of activities:

1. collecting data,
2. cleaning data,
3. interpreting data,

From Jaap van den Herik's CISIM 2016 Keynote

"In data science we nowadays distinguish seven phases of activities:

1. collecting data,
2. cleaning data,
3. interpreting data,
4. analyzing data,

From Jaap van den Herik's CISIM 2016 Keynote

"In data science we nowadays distinguish seven phases of activities:

1. collecting data,
2. cleaning data,
3. interpreting data,
4. analyzing data,
5. visualization of data,

From Jaap van den Herik's CISIM 2016 Keynote

"In data science we nowadays distinguish seven phases of activities:

1. collecting data,
2. cleaning data,
3. interpreting data,
4. analyzing data,
5. visualization of data,
6. narrative science, and

From Jaap van den Herik's CISIM 2016 Keynote

"In data science we nowadays distinguish seven phases of activities:

1. collecting data,
2. cleaning data,
3. interpreting data,
4. analyzing data,
5. visualization of data,
6. narrative science, and
7. the emergence of new paradigms.

These are our recommendations:

From Jaap van den Herik's CISIM 2016 Keynote

"In data science we nowadays distinguish seven phases of activities:

1. collecting data,
2. cleaning data,
3. interpreting data,
4. analyzing data,
5. visualization of data,
6. narrative science, and
7. the emergence of new paradigms.

These are our recommendations:

1. Increase research on AI systems for Big Data and Deep Learning with emphasis on moral constraints.

From Jaap van den Herik's CISIM 2016 Keynote

"In data science we nowadays distinguish seven phases of activities:

1. collecting data,
2. cleaning data,
3. interpreting data,
4. analyzing data,
5. visualization of data,
6. narrative science, and
7. the emergence of new paradigms.

These are our recommendations:

1. Increase research on AI systems for Big Data and Deep Learning with emphasis on moral constraints.
2. Increase research on AI systems for Big Data and Deep Learning with emphasis on the prevention of AI systems to be hacked.

From Jaap van den Herik's CISIM 2016 Keynote

"In data science we nowadays distinguish seven phases of activities:

1. collecting data,
2. cleaning data,
3. interpreting data,
4. analyzing data,
5. visualization of data,
6. narrative science, and
7. the emergence of new paradigms.

These are our recommendations:

1. Increase research on AI systems for Big Data and Deep Learning with emphasis on moral constraints.
2. Increase research on AI systems for Big Data and Deep Learning with emphasis on the prevention of AI systems to be hacked.
3. Establish (a) a committee of Data Authorities and (b) an ethical committee.

Adding a few more "Commandments"

Adding a few more “Commandments”

- Models should be “introspected” for meanings of their quantities. . .

Adding a few more "Commandments"

- Models should be "introspected" for meanings of their quantities...
- ...and for implications of those meanings.

Adding a few more “Commandments”

- Models should be “introspected” for meanings of their quantities. . .
- . . . and for implications of those meanings.
- Cross-validation not just on subsets of the data but also of models.

Adding a few more "Commandments"

- Models should be "introspected" for meanings of their quantities...
- ...and for implications of those meanings.
- Cross-validation not just on subsets of the data but also of models.
- Models should be "Good At Any Grain"—?

Adding a few more "Commandments"

- Models should be "introspected" for meanings of their quantities...
- ...and for implications of those meanings.
- Cross-validation not just on subsets of the data but also of models.
- Models should be "Good At Any Grain"—?
- [your reactions go here]

Adding a few more "Commandments"

- Models should be "introspected" for meanings of their quantities...
- ...and for implications of those meanings.
- Cross-validation not just on subsets of the data but also of models.
- Models should be "Good At Any Grain"—?
- [your reactions go here]
- Thank you very much for the invitation!