Computer and Human Preference Divergences at Chess

Southern Economic Association Annual Conference 2014, Atlanta, GA

Kenneth W. Regan\(^1\)
University at Buffalo (SUNY)

Nov. 24, 2014

\(^1\)Joint work with Tamal Biswas and Jason Zhou. Websites:
http://www.cse.buffalo.edu/~regan/
http://www.cse.buffalo.edu/~regan/chess/fidelity/
Computer chess *engines* running on mass-market hardware can slay even the best human players.
1. Computer chess *engines* running on mass-market hardware can slay even the best human players.

2. Hence **cheating** with computers during human games has emerged...
Outline

1 Computer chess *engines* running on mass-market hardware can slay even the best human players.

2 Hence **cheating** with computers during human games has emerged... and **cheating detection** has become a need.
Outline

1. Computer chess *engines* running on mass-market hardware can slay even the best human players.
2. Hence *cheating* with computers during human games has emerged... and *cheating detection* has become a need.
3. Desirable to measure more than raw performance.
Outline

1. Computer chess *engines* running on mass-market hardware can slay even the best human players.

2. Hence *cheating* with computers during human games has emerged... and *cheating detection* has become a need.

3. Desirable to measure more than raw performance.

4. Other axis is **Cognitive Style**—how do computers’ differ from humans’?
Computer and Human Preference Divergences at Chess

Outline

1. Computer chess engines running on mass-market hardware can slay even the best human players.

2. Hence cheating with computers during human games has emerged... and cheating detection has become a need.

3. Desirable to measure more than raw performance.

4. Other axis is Cognitive Style—how do computers’ differ from humans’?

5. Can we discriminate it? How much does it add to significance of cheating tests?
Outline

1. Computer chess *engines* running on mass-market hardware can slay even the best human players.
2. Hence *cheating* with computers during human games has emerged...and *cheating detection* has become a need.
3. Desirable to measure more than raw performance.
4. Other axis is *Cognitive Style*—how do computers’ differ from humans’?
5. Can we discriminate it? How much does it add to significance of cheating tests?
6. Implications for *Computerized Agents*, including PDAs.
Computer and Human Preference Divergences at Chess

Outline

1. Computer chess *engines* running on mass-market hardware can slay even the best human players.

2. Hence *cheating* with computers during human games has emerged... and *cheating detection* has become a need.

3. Desirable to measure more than raw performance.

4. Other axis is **Cognitive Style**—how do computers’ differ from humans’?

5. Can we discriminate it? How much does it add to significance of cheating tests?

6. Implications for **Computerized Agents**, including PDAs.

Outline

1. Computer chess *engines* running on mass-market hardware can slay even the best human players.
2. Hence **cheating** with computers during human games has emerged... and **cheating detection** has become a need.
3. Desirable to measure more than raw performance.
4. Other axis is **Cognitive Style**—how do computers’ differ from humans’?
5. Can we discriminate it? How much does it add to significance of cheating tests?
6. Implications for **Computerized Agents**, including PDAs.
8. Discussion and applicability of model to the other papers.
A Predictive Analytic Model

Domain: A set of decision-making situations $t$. Chess game turns
A Predictive Analytic Model

1 Domain: A set of decision-making situations \( t \).
   Chess game turns

2 Inputs: Values \( v_i \) for every option at turn \( t \).
   Computer values of moves \( m_i \)
A Predictive Analytic Model

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

2. Inputs: Values $v_i$ for every option at turn $t$. Computer values of moves $m_i$

3. Parameters: $s, c, \ldots$ denoting skills and levels. Trained correspondence to chess Elo rating $E$
A Predictive Analytic Model

1. Domain: A set of decision-making situations \( t \).
   Chess game turns

2. Inputs: Values \( v_i \) for every option at turn \( t \).
   Computer values of moves \( m_i \)

3. Parameters: \( s, c, \ldots \) denoting skills and levels.
   Trained correspondence to chess Elo rating \( E \)

4. Defines fallible agent \( P(s, c, \ldots) \).
A Predictive Analytic Model

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

2. Inputs: Values $v_i$ for every option at turn $t$. Computer values of moves $m_i$

3. Parameters: $s, c, \ldots$ denoting skills and levels. Trained correspondence to chess Elo rating $E$

4. Defines fallible agent $P(s, c, \ldots)$.

5. Main Output: Probabilities $p_{t,i}$ for $P(s, c, \ldots)$ to select option $i$ at time $t$. 

A Predictive Analytic Model

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

2. Inputs: Values $v_i$ for every option at turn $t$. Computer values of moves $m_i$

3. Parameters: $s, c, \ldots$ denoting skills and levels. Trained correspondence to chess Elo rating $E$

4. Defines *fallible agent* $P(s, c, \ldots)$.

5. Main Output: Probabilities $p_{t,i}$ for $P(s, c, \ldots)$ to select option $i$ at time $t$.

6. Derived Outputs:
   - Aggregate statistics: *move-match* MM, *average error* AE, \ldots
   - Projected confidence intervals for those statistics.
   - “Intrinsic Performance Ratings” (IPR’s).
Elo Rating System

- Points are (ideally) zero-sum: what $P$ gains $O$ loses.
Elo Rating System

- Points are (ideally) zero-sum: what $P$ gains $O$ loses.
- Only rating differences matter—absolute numbers have no intrinsic meaning.
Elo Rating System

- Points are (ideally) zero-sum: what P gains O loses.
- Only rating differences matter—absolute numbers have no intrinsic meaning. Yet my work argues no significant “inflation.”

2800 World champ: Carlsen peak 2881, now 2863
2700 “Super-GM”
2600 “Strong GM”
2500 Grandmaster (GM)
2400 International Master (IM) (KWR, D. Levy, H. Berliner)
2200 National Master, 30,000 worldwide
2000 Expert

... ... ... ...
1000 Class E, “bright beginner”

600? True beginner with “sight of the board”? 
Computer and Human Preference Divergences at Chess

The Data

- Over 2 million moves of 50-PV data: approaching 200GB
The Data

- Over 2 million moves of 50-PV data: approaching 200GB
- Over 30 million moves of Single-PV data: about 35 GB
The Data

- Over 2 million moves of 50-PV data: approaching 200 GB
- Over 30 million moves of Single-PV data: about 35 GB
- = over 100 million pages of text data at 2k/page.
- All taken on two quad-core home-style PC’s. Is this “Big Data”?
The Data

- Over 2 million moves of **50-PV** data: approaching **200GB**
- Over 30 million moves of **Single-PV** data: about **35 GB**
- = over 100 million pages of text data at 2k/page.
- All taken on two quad-core home-style PC’s. Is this “Big Data”?
Two Modes of Operation

1. To test games by player $P$ for cheating:

   - Regress on large data to set params $s_c, \ldots$ for Elo rating of $P$.
   - Use $s_c, \ldots$ to generate projections and confidence intervals for tests ("MM" and "AE" tests) from analysis of player's games.
   - So far independent of moves played.
   - Compare actual results from moves played.

2. To compute "Intrinsic Performance Rating" (IPR) for $P$:

   - Regress on $P$'s games — i.e. on small data — to get $s_P, c_P, \ldots$.
   - Apply $s, c, \ldots$ to "Virtual Standardized Test" (same 8,316 positions for everyone, results agree with whole training set to 4 places).
   - Score mapped to Elo scale, to get IPR $\pm$ error.
   - Error of measurement, not confidence test.
Two Modes of Operation

1. To test games by player $P$ for cheating:
   - Regress on large data to set params $s, c, \ldots$ for Elo rating of $P$. 

   - Score mapped to Elo scale, to get IPR ± error.

   - Error of measurement, not confidence test.
Two Modes of Operation

1. To test games by player $P$ for cheating:
   - Regress on large data to set params $s, c, \ldots$ for Elo rating of $P$.
   - Use $s, c, \ldots$ to generate projections and confidence intervals for tests ("MM" and "AE" tests) from analysis of player’s games.
Two Modes of Operation

1. To test games by player $P$ for cheating:
   - Regress on large data to set params $s, c, \ldots$ for Elo rating of $P$.
   - Use $s, c, \ldots$ to generate projections and confidence intervals for tests ("MM" and "AE" tests) from analysis of player’s games.
   - So far independent of moves played.
Two Modes of Operation

1. To test games by player \( P \) for cheating:
   - Regress **on large data** to set params \( s, c, \ldots \) for Elo rating of \( P \).
   - Use \( s, c, \ldots \) to generate projections and confidence intervals for tests ("MM" and "AE" tests) from analysis **of player’s games**.
   - So far independent of moves played.
   - Compare actual results from moves played.
Two Modes of Operation

1. To test games by player $P$ for cheating:
   - Regress **on large data** to set params $s, c, \ldots$ for Elo rating of $P$.
   - Use $s, c, \ldots$ to generate projections and confidence intervals for tests ("MM" and "AE" tests) from analysis of player’s games.
   - So far independent of moves played.
   - Compare actual results from moves played.

2. To compute "Intrinsic Performance Rating" (IPR) for $P$: 

Two Modes of Operation

1. To test games by player $P$ for cheating:
   - Regress on large data to set params $s, c, \ldots$ for Elo rating of $P$.
   - Use $s, c, \ldots$ to generate projections and confidence intervals for tests (“MM” and “AE” tests) from analysis of player’s games.
   - So far independent of moves played.
   - Compare actual results from moves played.

2. To compute “Intrinsic Performance Rating” (IPR) for $P$:
   - Regress on $P$’s games—i.e. on small data—to get $s_P, c_P, \ldots$
Two Modes of Operation

1. To test games by player $P$ for cheating:
   - Regress on large data to set params $s, c, \ldots$ for Elo rating of $P$.
   - Use $s, c, \ldots$ to generate projections and confidence intervals for tests ("MM" and "AE" tests) from analysis of player’s games.
   - So far independent of moves played.
   - Compare actual results from moves played.

2. To compute “Intrinsic Performance Rating” (IPR) for $P$:
   - Regress on $P$’s games—i.e. on small data—to get $s_P, c_P, \ldots$
   - Apply $s, c, \ldots$ to “Virtual Standardized Test” (same 8,316 positions for everyone, results agree with whole training set to 4 places).
Two Modes of Operation

1. To test games by player $P$ for cheating:
   - Regress on large data to set params $s, c, \ldots$ for Elo rating of $P$.
   - Use $s, c, \ldots$ to generate projections and confidence intervals for tests ("MM" and "AE" tests) from analysis of player's games.
   - So far independent of moves played.
   - Compare actual results from moves played.

2. To compute “Intrinsic Performance Rating” (IPR) for $P$:
   - Regress on $P$’s games—i.e. on small data—to get $s_P, c_P, \ldots$
   - Apply $s, c, \ldots$ to “Virtual Standardized Test” (same 8,316 positions for everyone, results agree with whole training set to 4 places).
   - Score mapped to Elo scale, to get IPR $\pm$ error.
Two Modes of Operation

1. To test games by player $P$ for cheating:
   - Regress **on large data** to set params $s, c, \ldots$ for Elo rating of $P$.
   - Use $s, c, \ldots$ to generate projections and confidence intervals for tests (“MM” and “AE” tests) from analysis of player’s games.
   - So far independent of moves played.
   - Compare actual results from moves played.

2. To compute “Intrinsic Performance Rating” (IPR) for $P$:
   - Regress **on $P$’s games**—i.e. **on small data**—to get $s_P, c_P, \ldots$
   - Apply $s, c, \ldots$ to “Virtual Standardized Test” (same 8,316 positions for everyone, results agree with whole training set to 4 places).
   - Score mapped to Elo scale, to get IPR ± error.
   - Error of measurement, not confidence test.
Examples

2014 World Championship Match

- Anand, 2785 ± 145
- Carlsen, 2920 ± 135
- Combined, 2850 ± 100,

Screening test:

<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>72</td>
<td>67.8%</td>
<td>0.055</td>
<td>202</td>
<td>Tal, Mihail</td>
<td>Marseille1989cat11</td>
<td></td>
</tr>
<tr>
<td>73</td>
<td>67.8%</td>
<td>0.078</td>
<td>121</td>
<td>Morphy, Paul</td>
<td>MorphyParisOffhand1858</td>
<td></td>
</tr>
<tr>
<td>74</td>
<td>67.7%</td>
<td>0.032</td>
<td>297</td>
<td>Shirov, A.</td>
<td>SofiaMTel2009cat21</td>
<td></td>
</tr>
</tbody>
</table>

Full test: Morphy at 2350 projected to match almost 60%, and full test actual is less ("regression to mean"), so not significant.
Engines work by iteratively deepened search. Some moves' values "swing" markedly down (a "trap") or up (a "hidden resource"). New "Depth" paper shows strong effect of swing on human probabilities. Computers largely immune to effect, especially in fixed-depth play. Explains 'strange' 58%–42% law for human preference of first-listed of two moves given equal value at highest depth, conditioned on one of them having been played. First-listed move higher-valued at lower depths; moves sort is stable. Use values at all depths to predict; use highest-depth values to assess.
Separating **Skill Assessment** From **Prediction**

- Engines work by *iteratively deepened search*. 

Some moves’ values "swing" markedly down (a "trap") or up (a "hidden resource"). New "Depth" paper shows strong effect of swing on human probabilities. Computers largely immune to effect, especially in fixed-depth play. Explains ‘strange’ 58%–42% law for human preference of first-listed of two moves given equal value at highest depth, conditioned on one of them having been played. First-listed move higher-valued at lower depths; moves sort is stable. Use values at all depths to predict; use highest-depth values to assess.
Separating **Skill Assessment** From **Prediction**

- Engines work by *iteratively deepened search*.
- Some moves’ values “*swing*” markedly down (a “*trap*”) or up (a “*hidden resource*”).
Separating **Skill Assessment** From **Prediction**

- Engines work by *iteratively deepened search*.
- Some moves’ values “*swing*” markedly down (a “trap”) or up (a “hidden resource”).
- New “Depth” paper shows strong effect of *swing* on human probabilities.
Separating Skill Assessment From Prediction

- Engines work by iteratively deepened search.
- Some moves’ values “swing” markedly down (a “trap”) or up (a “hidden resource”).
- New “Depth” paper shows strong effect of swing on human probabilities.
- Computers largely immune to effect, especially in fixed-depth play.
Separating **Skill Assessment** From **Prediction**

- Engines work by *iteratively deepened search*.  
- Some moves’ values “*swing*” markedly down (a “trap”) or up (a “hidden resource”).
- New “Depth” paper shows strong effect of *swing* on human probabilities.
- Computers largely immune to effect, especially in fixed-depth play.
- Explains ‘strange’ 58%–42% law for human preference of first-listed of two moves given equal value at highest depth, conditioned on one of them having been played.
Separating **Skill Assessment** From **Prediction**

- Engines work by *iteratively deepened search*.
- Some moves’ values “*swing*” markedly down (a “*trap*”) or up (a “*hidden resource*”).
- New “*Depth*” paper shows strong effect of *swing* on human probabilities.
- Computers largely immune to effect, especially in fixed-depth play.
- Explains ‘*strange*’ 58%–42% law for human preference of first-listed of two moves given equal value at highest depth, conditioned on one of them having been played.
- First-listed move higher-valued at lower depths; moves sort is *stable*. 
Separating **Skill Assessment** From **Prediction**

- Engines work by *iteratively deepened search*.
- Some moves’ values “*swing*” markedly down (a “*trap*”) or up (a “*hidden resource*”).
- New “Depth” paper shows strong effect of *swing* on human probabilities.
- Computers largely immune to effect, especially in fixed-depth play.
- Explains ‘strange’ 58%–42% law for human preference of first-listed of two moves given equal value at highest depth, conditioned on one of them having been played.
- First-listed move higher-valued at lower depths; moves sort is *stable*.

*Use values at all depths to *predict*; use highest-depth values to *assess*.**
And When You’re Higher Rated
Would You Like it to be Your Move?

Position Evaluation vs. Win Expectation

- From Player's perspective
- From Opponent's perspective

Win Expectation vs. Position Evaluation (Stockfish DD)
Effect Absent in Computer Play

Position Evaluation vs. Win Expectation for CEGT

CEGT-TTC: Player's perspective
CEGT-TTC: Opponent's perspective

Win Expectation

Position Evaluation (Stockfish 4)
Managing a Time Budget

![Graph showing Scaled and Unscaled data over Move Index]

- Error Per Move
- Time Control
Minding Nickels and Dimes

![Graph showing error versus advantage or disadvantage for different programs: Houdini, Komodo, Rybka, StockFish, and CEGT. The graph indicates that humans, checked with four programs, perform better than computers.](image)

Humans, checked with four programs.
Some Evidence for Psychological

Minima stay at 0.
Add Human-Computer Tandems

Forcing Index (2500 perspective)

- Computer (avg.): 49
- Computer+Human: 54.5
- Human: 53.3
Evidently the humans called the shots. How was the quality?
2007–08 Freestyle Performance

Forcing Index (2500 perspective)

- **Computer (avg.)**: 49
- **Computer+Human**: 54.5
- **Human**: 53.3

- **2895 in 2007-08**
- **3105 in 2007-08**

Adding 210 Elo was significant. Forcing but good teamwork.
2014 Freestyle Tournament Performance

Forcing Index (2500 perspective)

<table>
<thead>
<tr>
<th></th>
<th>45</th>
<th>50</th>
<th>55</th>
<th>60</th>
<th>65</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computer (avg.)</td>
<td>49</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Computer+Human</td>
<td>54.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Human</td>
<td>53.3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

2014: tandems marginally better W-L, but quality not clear...
Summary For Us and PDAs

PDAs pick up every little difference: "Forest and Trees"

We should avoid overconfidence... and take counsel when "down."

Look before we Leap... Don't rush in... Measure risks.

Even at a purely calculational pursuit like chess, our brains still contribute.

Main takeaway: It should be natural to program PDAs so they enhance our freedom rather than constrain it. This could be the beginning of a beautiful relationship.
Summary For Us and PDAs

1. PDAs pick up every little difference: “Forest and Trees”
Summary For Us and PDAs

1. PDAs pick up every little difference: “Forest and Trees”
2. We should avoid overconfidence...
Summary For Us and PDAs

1. PDAs pick up every little difference: “Forest and Trees”
2. We should avoid overconfidence... and take counsel when “down.”
Summary For Us and PDAs

1. PDAs pick up every little difference: “Forest and Trees”
2. We should avoid overconfidence... and take counsel when “down.”
3. Look before we Leap...
Summary For Us and PDAs

1. PDAs pick up every little difference: “Forest and Trees”
2. We should avoid overconfidence... and take counsel when “down.”
3. Look before we Leap... Don’t rush in...
Summary For Us and PDAs

1. PDAs pick up every little difference: “Forest and Trees”
2. We should avoid overconfidence...and take counsel when “down.”
3. Look before we Leap...Don’t rush in...Measure risks.
Summary For Us and PDAs

1. PDAs pick up every little difference: “Forest and Trees”
2. We should avoid overconfidence...and take counsel when “down.”
3. Look before we Leap...Don’t rush in...Measure risks.
4. Even at a purely calculational pursuit like chess, our brains still contribute.
Summary For Us and PDAs

1. PDAs pick up every little difference: “Forest and Trees”
2. We should avoid overconfidence... and take counsel when “down.”
3. Look before we Leap... Don’t rush in... Measure risks.
4. Even at a purely calculational pursuit like chess, our brains still contribute. (2014: maybe)
Summary For Us and PDAs

1. PDAs pick up every little difference: “Forest and Trees”
2. We should avoid overconfidence...and take counsel when “down.”
3. **Look before we Leap...Don’t rush in...Measure risks.**
5. **Main takeaway:**
Summary For Us and PDAs

1. PDAs pick up every little difference: “Forest and Trees”
2. We should avoid overconfidence... and take counsel when “down.”
3. Look before we Leap... Don’t rush in... Measure risks.
4. Even at a purely calculational pursuit like chess, our brains still contribute. (2014: maybe)
5. Main takeaway:

   It should be natural to program PDAs so they enhance our freedom rather than constrain it.
Summary For Us and PDAs

1. PDAs pick up every little difference: “Forest and Trees”
2. We should avoid overconfidence... and take counsel when “down.”
3. Look before we Leap... Don’t rush in... Measure risks.
4. Even at a purely calculational pursuit like chess, our brains still contribute. (2014: maybe)
5. Main takeaway:

   It should be natural to program PDAs so they enhance our freedom rather than constrain it.

This could be the beginning of a beautiful relationship...