Learning a Value Analysis Tool For Agent Evaluation

Martha White  Michael Bowling

Department of Computer Science
University of Alberta

International Joint Conference on Artificial Intelligence, 2009
Motivation: A Story

- Imagine that you have made the world’s best poker agent
- You’ve played millions of games against other bots and won!
- Now you want to pit the agent against the world’s best human players...
Problem

- Poker has a lot of luck
- In Texas hold’em two player-limit poker:
  - Standard deviation of winnings is 6.0 sb
  - Required precision to distinguish pro and amateur: 0.05 sb
- Need 50K hands for statistically significant results using average of winnings (Monte Carlo estimation)
- Humans play 500-1000 hands
Poker Example

Figure: Always Call versus Always Raise

Martha White, Michael Bowling
Learning a Value Analysis Tool For Agent Evaluation
First Man-Machine Poker Championship

Ali Eslami vs Phil Laak

Polaris
First Man-Machine Poker Championship

- Total winnings must exceed 25 bets
- Results:
  - Match 1: Polaris up by 7 bets - Draw
  - Match 2: Polaris up by 93 bets - Win
  - Match 3: Polaris up by 82 bets - Loss
  - Match 4: Polaris down by 57 bets - Loss
- None of the results were statistically significant
Approach: Remove Luck

- Monte Carlo approach only uses utilities (winnings)
- **Idea:** Look at information from the entire game to reduce variance of the performance estimate
- Separate value obtained from **luck** and from own **skill**
- New estimator called **DIVAT**
With DIVAT plus a few extra tricks:

- Can now estimate performance in 500 hands
- First-Man Machine Result: 2 statistically sig. wins, 2 Draws
With DIVAT plus a few extra tricks:

- Can now estimate performance in 500 hands
- First-Man Machine Result: 2 statistically sig. wins, 2 Draws
Results (cont...)

For the Second Man-Machine Poker Championship

- Switch between strategies depending on the human player
- Final Result: 3 statistically sig. wins, 0 Losses, 3 Draws
What about other competitions?

How statistically significant are the results in...

- The Trading Agent Competition
- The Annual Reinforcement Learning Competition
- The RoboCup Competition (Soccer Simulation League)
- Others?
What about other competitions?

How statistically significant are the results in...

- The Trading Agent Competition
- The Annual Reinforcement Learning Competition
- The RoboCup Competition (Soccer Simulation League)
- Others?
What about other competitions?

How statistically significant are the results in...

- The Trading Agent Competition
- The Annual Reinforcement Learning Competition
- The RoboCup Competition (Soccer Simulation League)
- Others?
What about other competitions?

How statistically significant are the results in...

- The Trading Agent Competition
- The Annual Reinforcement Learning Competition
- The RoboCup Competition (Soccer Simulation League)
- Others?
The [Second Place] team was quick to point out that the point spread was not statistically significant, while the [First Place] team was quick to point out that they won.

-Doug Bryan, Association for Trading Agent Research, 2000

In general during TAC01 no agent performed significantly better than all the others.

-A Statistical Analysis of the Trading Agent Competition 2001
Contribution

- The success in Poker springs from an advantage-sum technique for ‘removing the luck’ from the estimate
  - The technique requires an expert defined value function over the states of the system
- We propose to learn this value function from interactions between players
- This approach facilitates applying the advantage-sum technique to a host of other domains
Contribution

- The success in Poker springs from an advantage-sum technique for ‘removing the luck’ from the estimate
  - The technique requires an expert defined value function over the states of the system
- We propose to learn this value function from interactions between players
  - This approach facilitates applying the advantage-sum technique to a host of other domains
Contribution

- The success in Poker springs from an advantage-sum technique for ‘removing the luck’ from the estimate
  - The technique requires an expert defined value function over the states of the system
- We propose to learn this value function from interactions between players
- This approach facilitates applying the advantage-sum technique to a host of other domains
Intuitive Definition

- Finite horizon sequential decision-making tasks
- Domains where the history can be represented as 
  \( c_0 a_0 c_1 a_1 \ldots c_m a_m \)
- A utility function \( u_i : \mathbb{Z} \rightarrow \mathbb{R} \) for each player \( i \in \{1 \ldots N\} \)
Intuitive Definition

- Finite horizon sequential decision-making tasks
  - Domains where the history can be represented as $c_0 a_0 c_1 a_1 \ldots c_m a_m$
- A utility function $u_i : \mathbb{Z} \rightarrow \mathbb{R}$ for each player $i \in \{1 \ldots N\}$
Examples

- $n$-Player general-sum and zero-sum games
- Finite-horizon POMDPs/MDPs
Assumptions

- We assume we know the dynamics of the chance nodes: $P(c|h) =$ the probability that $c$ occurs given $h$
- No assumptions about the strategies, $\sigma$, of the players
Basic Approach

- Estimate the expectation with independent samples $Z_1, \ldots, Z_T$

$$\hat{U}_j = \frac{1}{T} \sum_t u_j(z_t)$$

- Estimate is unbiased

$$E[\hat{U}_j|\sigma] = E[u_j(z)|\sigma]$$
Improved Approach

- Identify an unbiased, **lower variance function** $\hat{u}_j : Z \rightarrow \mathbb{R}$

$$\forall \sigma \quad E_Z [\hat{u}_j(z) | \sigma] = E_Z [u_j(z) | \sigma]$$
Zinkevich et al. [2006] introduced a general approach to constructing low-variance estimators. Given a value function $V_j : H \rightarrow \mathbb{R}$ with $V_j(z) = u_j(z)$, separate utility into luck and skill.

$$S_{V_j}(z) = \sum_i V_j(c_0a_0...c_ia_i) - V_j(c_0a_0...c_i)$$

$$L_{V_j}(z) = \sum_i V_j(c_0a_0...c_ia_ia_{i+1}) - V_j(c_0a_0...c_ia_i)$$
Advantage Sum Estimators (cont...)

- Notice

\[ u_j(z) = S_{V_j}(z) + L_{V_j}(z) + P_{V_j} \]

\[ P_{V_j} = V_j(\emptyset) \]

- If \( V_j \) chosen carefully such that \( E \left[ L_{V_j}(z) | \sigma \right] = 0 \),
  then \( \hat{u}_{V_j} = S_{V_j}(z) + P_{V_j} \) unbiased

- This approach gives the minimum variance estimator if \( V_j \)
  exactly predicts the utility.
Advantage Sum Estimators (cont...)

- Notice

\[ u_j(z) = S_{V_j}(z) + L_{V_j}(z) + P_{V_j} \]

\[ P_{V_j} = V_j(\emptyset) \]

- If \( V_j \) chosen carefully such that \( E\left[ L_{V_j}(z) | \sigma \right] = 0 \),
then \( \hat{u}_{V_j} = S_{V_j}(z) + P_{V_j} \) unbiased

- This approach gives the minimum variance estimator if \( V_j \) exactly predicts the utility.
Advantage Sum Estimators (cont...)

- Notice

\[ u_j(z) = S_{V_j}(z) + L_{V_j}(z) + P_{V_j} \]
\[ P_{V_j} = V_j(\emptyset) \]

- If \( V_j \) chosen carefully such that \( E \left[ L_{V_j}(z) \mid \sigma \right] = 0 \), then \( \hat{u}_{V_j} = S_{V_j}(z) + P_{V_j} \) unbiased

- This approach gives the minimum variance estimator if \( V_j \) exactly predicts the utility.
DIVAT: Ignorant Value Assessment Tool

- Applied to two-player, limit Texas hold’em poker
- Uses a hand-designed function shown to produce an unbiased estimator
- Three-fold reduction (needs nine times fewer hands for statistical conclusions)
DIVAT: Example

- **Call**
  - Card: 10 of Hearts
  - Action: +4 sb

- **Raise**
  - Card: 10 of Diamonds
  - Action: +3 sb

- **Reraise**
  - Card: 10 of Clubs
  - Action: -8 sb

- **Bet**
  - Card: 10 of Spades
  - Action: -7 sb

- **Fold**
  - Card: 10 of Clubs
  - Action: -4 sb

---

Martha White, Michael Bowling

Learning a Value Analysis Tool For Agent Evaluation
MIVAT: Informed Value Assessment Tool

- Learns the value function $V_j$ from past interaction between players
- Main advantages:
  - Designing a value function can be difficult
  - Can tailor a function to a specific group of players
How do we learn a value function?

- Notice that \( \hat{u}_V(z) = u_j(z) - L_V(z) \)
- Define

\[
V_j(h_i = c_0a_0...c_ia_i) \equiv \sum_{c'} P(c'|h_i) V_j(h_i,c') \quad (= E[V_j(h_i,c)])
\]

- so then

\[
L_V(z) = \sum_i \left( V_j(h_ic_{i+1}) - \sum_{c'} P(c'|h_i) V_j(h_i,c') \right)
\]
How do we learn a value function? (cont...)

This reformulation simplifies the learning problem because:
- We need only define a value function for the histories directly following chance nodes.
- We are guaranteed unbiasedness.

\[
E[L_{V_j}] = \sum_i E[V_j(h_i c_{i+1})] - \sum_{c'} P(c'|h_i) V_j(h_i c')
\]

\[
= 0
\]

Martha White, Michael Bowling
Learning a Value Analysis Tool For Agent Evaluation
What value function should we learn?

Goal: minimize variance

Minimize: \( \sum_{t=1}^{T} \left( \hat{u}_{V_j}(z_t) - \frac{1}{T} \sum_{t'=1}^{T} \hat{u}_{V_j}(z_{t'}) \right)^2 \)
Learning a linear value function

- \( \phi : H \rightarrow \mathcal{R}^d \) a vector of \( d \) features on the histories
- We need to learn the weights, \( \theta_j \), on these features

\[
V_j(h) = \phi(h)^T \theta_j
\]
The optimization simplifies to

\[
\text{Minimize: } \theta_j \in \mathcal{R}^d \quad C(\theta_j) = \sum_{t=1}^{T} \left[ f(\phi(t)) \theta_j \right]^2
\]

We can obtain a closed-form solution for \( \theta_j^* \) for all players \( 1 \ldots N \) by optimizing this function.
MIVAT: Example

- \( c_1 \) with +3 sb
- \( a_1 \) with call
- \( c_2 \) with -8 sb
- \( a_2 \) with raise
- \( c_3 \) with -6 sb
- \( a_3 \) with reraise

Martha White, Michael Bowling

Learning a Value Analysis Tool For Agent Evaluation
Recap

- We have simplified learning the value function
- We have a closed-form solution for linear value functions
- The approach is within a well-justified theoretical framework (advantage-sum estimators)
Domain: Texas hold’em poker

Three Texas hold’em domains:
- Two-player limit poker
- Two-player no-limit poker
- Six-player limit poker
Datasets

- 2008 AAAI Computer Poker Competition (Bots-Bots data)
  - Two-player limit: 9 bots
  - Two-player no-limit: 4 bots
  - Six-player limit: 6 bots
- Strong poker program versus battery of weak to strong human players (Bots-Humans)
- 450,000 training samples and 50,000 testing samples
Feature Design

- **Features:**
  - Pot-equity: the probability your hand wins given the current history
  - Hand-strength: expectation (over the undealt public cards) of winning against a random hand
  - Pot-size: amount of money in the pot
- Used polynomials of the features (up to quads)
- For two-player limit poker, also used the DIVAT estimate as a feature
Results: Two-Player Limit

- **General estimator**

<table>
<thead>
<tr>
<th>Data</th>
<th>MIVAT</th>
<th>MIVAT+</th>
<th>DIVAT</th>
<th>Money</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bot-Humans</td>
<td>2.387</td>
<td>2.220</td>
<td>2.238</td>
<td>5.669</td>
</tr>
<tr>
<td>Bots-Bots</td>
<td>2.542</td>
<td>2.454</td>
<td>2.506</td>
<td>5.438</td>
</tr>
</tbody>
</table>

- **Tailored estimators**

<table>
<thead>
<tr>
<th>Data</th>
<th>MIVAT Bot-Humans</th>
<th>MIVAT Bots-Bots</th>
<th>DIVAT</th>
<th>Money</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bot-Humans</td>
<td>2.169</td>
<td>2.253</td>
<td>2.238</td>
<td>5.669</td>
</tr>
<tr>
<td>Bots-Bots</td>
<td>2.461</td>
<td>2.418</td>
<td>2.506</td>
<td>5.438</td>
</tr>
</tbody>
</table>
### Results: Two-Player Limit

#### General estimator

<table>
<thead>
<tr>
<th>Data</th>
<th>MIVAT</th>
<th>MIVAT+</th>
<th>DIVAT</th>
<th>Money</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bot-Humans</td>
<td>2.387</td>
<td>2.220</td>
<td>2.238</td>
<td>5.669</td>
</tr>
<tr>
<td>Bots-Bots</td>
<td>2.542</td>
<td>2.454</td>
<td>2.506</td>
<td>5.438</td>
</tr>
</tbody>
</table>

#### Tailored estimators

<table>
<thead>
<tr>
<th>Data</th>
<th>MIVAT Bot-Humans</th>
<th>MIVAT Bots-Bots</th>
<th>DIVAT</th>
<th>Money</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bot-Humans</td>
<td>2.169</td>
<td>2.253</td>
<td>2.238</td>
<td>5.669</td>
</tr>
<tr>
<td>Bots-Bots</td>
<td>2.461</td>
<td>2.418</td>
<td>2.506</td>
<td>5.438</td>
</tr>
</tbody>
</table>
### Results: Domains w/o Variance Reduction Functions

- **Two-player no-limit results (25% variance reduction)**
  
<table>
<thead>
<tr>
<th>Data</th>
<th>MIVAT</th>
<th>Money</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bots-Bots</td>
<td>32.407158</td>
<td>42.339743</td>
</tr>
</tbody>
</table>

- **Six-player limit results (20% variance reduction)**
  
<table>
<thead>
<tr>
<th>Data</th>
<th>MIVAT</th>
<th>Money</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bots-Bots</td>
<td>22.9234365</td>
<td>28.0091805</td>
</tr>
</tbody>
</table>
Results: Domains w/o Variance Reduction Functions

- **Two-player no-limit results (25% variance reduction)**
  
<table>
<thead>
<tr>
<th>Data</th>
<th>MIVAT</th>
<th>Money</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bots-Bots</td>
<td>32.407158</td>
<td>42.339743</td>
</tr>
</tbody>
</table>

- **Six-player limit results (20% variance reduction)**
  
<table>
<thead>
<tr>
<th>Data</th>
<th>MIVAT</th>
<th>Money</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bots-Bots</td>
<td>22.9234365</td>
<td>28.0091805</td>
</tr>
</tbody>
</table>
Results Summary

**Pros**
- Using simple features, MIVAT matched an expert defined value function in two-player limit poker
- MIVAT enabled us to find lower-variance estimators in domains with no previous ones

**Cons**
- Feature design remains an important issue for the success of the estimator
Agent evaluation is important for scientific evaluation and agent development. We help automate agent evaluation with:
- A generic framework that overcomes the need for hand-designed functions
- The flexibility to tailor functions to specific groups of players
- A closed-form solution for linear value-functions
Future Work

- Find closed-form solutions for
  - different loss functions
  - non-linear value functions
- Extend approach to complex settings where explicit game formulation unavailable
  - e.g. simulated settings such as the Trading-Agent Competition
Questions?