Finding Friend and Foe in Multi-agent Games

Jack Serrino*, Max Kleiman-Weiner*, David Parkes, Josh Tenenbaum

Harvard & MIT

Poster #197
The Resistance: Avalon as a testbed for multi-agent learning and thinking

- Recent progress has focused on games where teammates are known in advance.
- 5-10 player game
- Two teams: “Spy” and “Resistance”
  - Spies know who is Spy and who is Resistance
  - Resistance only know they are Resistance
- Each rounds players sequentially propose subsets of players to go on missions and vote.
- Resistance players want missions to succeed and Spies want missions to fail.
- $> 10^{56}$ distinct information sets.

(Eskridge, 2012)
The Resistance: Avalon game overview

Round 1:
- The round begins with players 1, 2, 3, 4, 5.
- The round ends with a success or failure.

Round 2:
- If the round ends with a success, go to Round 3.
- If the round ends with a failure, go to Round 4.

Round 3:
- The round begins with players 1, 2, 3, 4, 5.
- The round ends with a success or failure.

Round 4:
- The round begins with players 1, 2, 3, 4, 5.
- The round ends with a success or failure.

Round 5:
- The round begins with players 1, 2, 3, 4, 5.
- The round ends with a success or failure.

Proposal Process:
- A proposal is made by a player.
- Players vote on the proposal.
- A majority vote determines the outcome.
- If the proposal is approved, the game moves forward.
- If the proposal is not approved, the game resets.

Mission:
- Players have private actions.
- If the mission succeeds, the game ends.
- If the mission fails, the game resets.

Proposals:
- Proposal 1: [4, 3]
- Proposal 2: [2, 3]
Combining counter-factual regret minimization with deep value networks

- Approach follows DeepStack system developed for NL poker (Moravcik et al, 2017).

- Actions themselves are only partially observed:
  - Deduction required in the loop of learning

- Unconstrained value networks are slower and less interpretable:
  - Develop an interpretable win-probability layer with better sample efficiency.

![Proposal Regrets](image)

![Voting Regrets](image)

![Mission Regrets](image)

![Diagram](image)

(Johanson et al, 2012)
The Win Layer

\[ V(I, \pi^\sigma) \in \mathbb{R}^{n \times |P|} \]

\[ |P| := \text{number of assignments to roles}, \, \rho \]
\[ n := \text{number of players} \]

Previous approaches:

\[ NN(I, \pi^\sigma) \approx V(I, \pi^\sigma) \]
- In 5-player Avalon, **300 values to estimate!**
- Correlations are learned imperfectly.

Our approach:

\[ \tilde{w}(I, \pi^\sigma) = \begin{bmatrix} P(\text{good win}|I, \pi^\sigma, \rho_1) \\ \vdots \\ P(\text{good win}|I, \pi^\sigma, \rho_{|P|}) \end{bmatrix} \in [0, 1]^{|P|} \]

\[ V(I, \pi^\sigma) = f(\tilde{w}(I, \pi^\sigma)) \]

\[ NN(I, \pi^\sigma) \approx \tilde{w}(I, \pi^\sigma) \]
- **60 values to estimate (via sigmoid)**
- Correlations are exact.
Deep value network architecture.

One-hot encoding of proposer position
[0 1 0 0 0]

Public belief state
b = P(role (p))

<table>
<thead>
<tr>
<th>P1</th>
<th>P2</th>
<th>P3</th>
<th>P4</th>
<th>P5</th>
<th>Pr</th>
</tr>
</thead>
<tbody>
<tr>
<td>M</td>
<td>R</td>
<td>R</td>
<td>S</td>
<td>A</td>
<td>.15</td>
</tr>
<tr>
<td>M</td>
<td>R</td>
<td>S</td>
<td>R</td>
<td>A</td>
<td>.12</td>
</tr>
<tr>
<td>M</td>
<td>S</td>
<td>R</td>
<td>R</td>
<td>A</td>
<td>.00</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S</td>
<td>A</td>
<td>R</td>
<td>R</td>
<td>M</td>
<td>.00</td>
</tr>
<tr>
<td>R</td>
<td>S</td>
<td>A</td>
<td>R</td>
<td>M</td>
<td>.03</td>
</tr>
</tbody>
</table>

Dense (ReLU)

80 x 1

60 x 1

Win likelihood
P(win|role (p))

<table>
<thead>
<tr>
<th>P1</th>
<th>P2</th>
<th>P3</th>
<th>P4</th>
<th>P5</th>
<th>P(win)</th>
</tr>
</thead>
<tbody>
<tr>
<td>M</td>
<td>R</td>
<td>R</td>
<td>S</td>
<td>A</td>
<td>.96</td>
</tr>
<tr>
<td>M</td>
<td>R</td>
<td>S</td>
<td>R</td>
<td>A</td>
<td>.73</td>
</tr>
<tr>
<td>M</td>
<td>S</td>
<td>R</td>
<td>R</td>
<td>A</td>
<td>.55</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>...</td>
</tr>
<tr>
<td>S</td>
<td>A</td>
<td>R</td>
<td>R</td>
<td>M</td>
<td>.23</td>
</tr>
<tr>
<td>R</td>
<td>S</td>
<td>A</td>
<td>R</td>
<td>M</td>
<td>.41</td>
</tr>
</tbody>
</table>

Probability-weighted value of each information set
P(I_i)*V_i(I_i)

<table>
<thead>
<tr>
<th>R</th>
<th>M</th>
<th>S</th>
<th>A</th>
</tr>
</thead>
<tbody>
<tr>
<td>.5</td>
<td>-.2</td>
<td>.3</td>
<td>-.1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>P1</th>
<th>P2</th>
<th>P3</th>
<th>P4</th>
<th>P5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 x 15</td>
<td>1 x 15</td>
<td>1 x 15</td>
<td>1 x 15</td>
<td>1 x 15</td>
</tr>
</tbody>
</table>
The Win Layer enables faster + better NN training
Deductive reasoning enhances learning when actions are not fully public

\begin{algorithm}
\begin{algorithmic}
\Procedure{CalculateTerminalBelief}{h, b, \pi_1...p}
\For{$\rho \in b$}
\State $b_{\text{term}}[\rho] \leftarrow b[\rho] \prod_i \pi_i(I_i(h, \rho))$
\EndFor
\State $b_{\text{term}}[\rho] \leftarrow b_{\text{term}}[\rho](1 - 1\{h \vdash \neg \rho\})$ \Comment{Zero beliefs that are logically inconsistent}
\EndProcedure
\end{algorithmic}
\end{algorithm}

1. Calculate joint probability of assignment given the public game history

2. Zero out assignments that are impossible given the history.

\textit{2) is not necessary in games like Poker, with fully observable actions!}
DeepRole wins at higher rates than: vanilla-CFR, MCTS, heuristic algorithms
Different strategies play different equilibria

(Wellman, 2006; Tuyls et al 2018)
DeepRole lesions are less robust and dominated by the full model.

(Wellman, 2006; Tuyls et al 2018)
DeepRole played online in mixed teams of human and bot players w/o communication (1,500+ games)
DeepRole outperformed humans playing online as both a collaborator and competitor

<table>
<thead>
<tr>
<th></th>
<th>Adding DeepRole to 4 DeepRole</th>
<th>+Human</th>
<th></th>
<th>Adding DeepRole to 4 Human</th>
<th>+Human</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Win Rate (%)</td>
<td>(N)</td>
<td></td>
<td>Win Rate (%)</td>
<td>(N)</td>
</tr>
<tr>
<td>Overall</td>
<td>46.9 ± 0.6</td>
<td>(7500)</td>
<td></td>
<td>38.8 ± 1.3</td>
<td>(1451)</td>
</tr>
<tr>
<td>Resistance</td>
<td>34.4 ± 0.7</td>
<td>(4500)</td>
<td></td>
<td>25.6 ± 1.5</td>
<td>(856)</td>
</tr>
<tr>
<td>Spy</td>
<td>65.6 ± 0.9</td>
<td>(3000)</td>
<td></td>
<td>57.8 ± 2.0</td>
<td>(595)</td>
</tr>
<tr>
<td></td>
<td>60.0 ± 5.5</td>
<td>(80)</td>
<td></td>
<td>48.1 ± 1.2</td>
<td>(1675)</td>
</tr>
<tr>
<td></td>
<td>51.4 ± 8.2</td>
<td>(37)</td>
<td></td>
<td>40.3 ± 1.5</td>
<td>(1005)</td>
</tr>
<tr>
<td></td>
<td>67.4 ± 7.1</td>
<td>(43)</td>
<td></td>
<td>59.7 ± 1.9</td>
<td>(670)</td>
</tr>
</tbody>
</table>
DeepRole make rapid accurate inferences about human roles during play and observation.
Finding Friend and Foe in Multi-agent Games

Jack Serrino*, Max Kleiman-Weiner*, David Parkes, Josh Tenenbaum

Harvard & MIT

Poster #197