Learning, Equilibria, Limitations, and Robots*

Michael Bowling
Computer Science Department
Carnegie Mellon University

*Joint work with Manuela Veloso
Talk Outline

- Robots
  - A two robot, adversarial, concurrent learning problem.
  - The challenges for multiagent learning.

- Limitations and Equilibria

- Limitations and Learning
The Domain = CMDragons = 1

Global

5 robots per team
Color markers for ID

2.3m

2.8m
The Domain = CMDragons = 2

Vision/Tracking

Current State
\[ [x, y, \theta, v_x, v_y, \omega]^T \]

Tactics/Strategy

Final Target
\[ [x_t, y_t, \theta_t, v_{t_{\text{max}}}]^T \]

Nav2Point

Waypoint
\[ [x_t, y_t, \theta_t, v_{t_{\text{max}}}]^T \]

Go2Point

Robot Command
\[ [v_x, v_y, \omega]^T \]

30Hz, 100ms system latency

Robot Servo
The Task = Breakthrough
The Task = Breakthrough
The Task = Breakthrough
The Challenges
The Challenges

- Challenge #1: Continuous State and Action Spaces
  - Value function approximation, parameterized policies, state and temporal abstractions.
  - Limits agent behavior, sacrificing optimality.
The Challenges

- Challenge #1: Continuous State and Action Spaces
  - Value function approximation, parameterized policies, state and temporal abstractions.
  - Limits agent behavior, sacrificing optimality.

- Challenge #2: Fixed Behavioral Components
  - Don’t learn motion control or obstacle avoidance.
  - Limits agent behavior, sacrificing optimality.
The Challenges

• Challenge #1: Continuous State and Action Spaces
  – Value function approximation, parameterized policies, state and temporal abstractions.
  – Limits agent behavior, sacrificing optimality.

• Challenge #2: Fixed Behavioral Components
  – Don’t learn motion control or obstacle avoidance.
  – Limits agent behavior, sacrificing optimality.

• Challenge #3: Latency
  – Can predict our own state through latency, not others.
  – Asymmetric partial observability.
  – Limits agent behavior, sacrificing optimality.
The Challenges – 1

- Challenge #1: Continuous State and Action Spaces
- Challenge #2: Fixed Behavioral Components
- Challenge #3: Latency

All of these challenges involve agent limitations. . . . their own and other’s.
Talk Outline

- Robots
  - A two robot, adversarial, concurrent learning problem.
  - The challenges for multiagent learning.

- Limitations and Equilibria

- Limitations and Learning
Limitations Restrict Behavior

- **Restricted Policy Space** — $\overline{\Pi}_i \subseteq \Pi_i$
  
  Any subset of stochastic policies, $\pi : S \rightarrow PD(A_i)$.

- **Restricted Best-Response** — $\overline{BR}_i(\pi_{-i})$
  
  The set of all policies from $\overline{\Pi}_i$ that are optimal given the policies of the other players.

- **Restricted Equilibrium** — $\pi_{i=1...n}$
  
  $\pi_i \in \overline{BR}_i(\pi_{-i})$
  
  A strategy for each player, where no player can and wants to deviate given the other players continue to play the equilibrium.

Do Restricted Equilibria Exist?
**Do Restricted Equilibria Exist? = 1**

### Explicit Game

| Payoffs | \[
\begin{pmatrix}
0 & -1 & 1 \\
1 & 0 & -1 \\
-1 & 1 & 0
\end{pmatrix}
\] |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Equilibrium</td>
<td>[\left\langle \frac{1}{3}, \frac{1}{3}, \frac{1}{3} \right\rangle, \left\langle \frac{1}{3}, \frac{1}{3}, \frac{1}{3} \right\rangle]</td>
</tr>
</tbody>
</table>

### Implicit Game

| Payoffs | \[
\begin{pmatrix}
-\frac{1}{2} & 0 \\
\frac{1}{2} & -\frac{1}{2} \\
0 & \frac{1}{2}
\end{pmatrix}
\] |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Equilibrium</td>
<td>[\left\langle 0, \frac{1}{3}, \frac{2}{3} \right\rangle, \left\langle \frac{2}{3}, \frac{1}{3} \right\rangle]</td>
</tr>
</tbody>
</table>

**Restricted Equilibrium**

\[\left\langle 0, \frac{1}{3}, \frac{2}{3} \right\rangle, \left\langle \frac{1}{3}, \frac{1}{2}, \frac{1}{6} \right\rangle\]
Do Restricted Equilibria Exist? = 2

- Two-player, zero-sum stochastic game (Marty’s Game 2). ¹

- Players restricted to policies that play the same distribution over actions in all states.

This game has no restricted equilibria!

¹This counterexample is brought to you by Martin Zinkevich.
Do Restricted Equilibria Exist? = 3

- In matrix games, if $\overline{\Pi}_i$ is convex, then . . .

- If $\overline{\Pi}_i$ is statewise convex, then . . .

- In no-control stochastic games, if convex $\overline{\Pi}_i$, then . . .

- In single-controller stochastic games, if $\overline{\Pi}_1$ is statewise convex, and $\overline{\Pi}_{i \neq 1}$ is convex, then . . .

- In team games . . .
Do Restricted Equilibria Exist? — 3

- In matrix games, if $\Pi_i$ is convex, then . . .

- If $\Pi_i$ is statewise convex, then . . .

- In no-control stochastic games, if convex $\Pi_i$, then . . .

- In single-controller stochastic games, if $\Pi_1$ is statewise convex, and $\Pi_i \neq 1$ is convex, then . . .

- In team games . . .

  . . . there exists a restricted equilibrium.

Proofs. Uses Kakutani’s fixed point theorem after showing

$$\forall \pi_{-i} \quad BR_i(\pi_{-i}) \text{ is convex.}$$
The Challenges = 2

- Challenge #1: Continuous State and Action Spaces
- Challenge #2: Fixed Behavioral Components
- Challenge #3: Latency

None of these are nice enough to guarantee the existence of equilibria.
Talk Outline

- Robots
  - A two robot, adversarial, concurrent learning problem.
  - The challenges for multiagent learning.
- Limitations and Equilibria
- Limitations and Learning
Three Ideas = One Algorithm

- Idea #1: Policy Gradient Ascent
- Idea #2: WoLF Variable Learning Rate
  
  \textit{GräWoLF— Gradient-based WoLF}

- Idea #3: Tile Coding
• Policy Gradient Ascent (Sutton et al., 2000)
  - Policy improvement with parameterized policies.
  - Takes steps in direction of the gradient of the value.

$$\pi(s, a) = \frac{e^{\phi_{sa} \cdot \theta_k}}{\sum_{b \in A_i} e^{\phi_{sb} \cdot \theta_k}}$$

$$\theta_{k+1} = \theta_k + \alpha_k \sum_a \phi_{sa} \pi(s, a) f_k(s, a)$$

- $f_k$ is an approximation of the advantage function.

$$f_k(s, a) \approx Q(s, a) - V^\pi(s)$$

$$\approx Q(s, a) - \sum_b \pi(s, b) Q(s, b)$$
Idea #2

- Win or Learn Fast (WoLF)  (Bowling & Veloso, 2002)
  - Variable learning rate accounts for other agents.
    * Learn fast when losing.
    * Cautious when winning, since agents may adapt.
  - Theoretical and empirical evidence of convergence.

![RPS Without WoLF](image1)

![RPS With WoLF](image2)
Idea #2 = 2

- Nash Equilibrium
- Restricted Policy Space
- Restricted Equilibrium

**RPS Without WoLF**

- P(Rock)
- P(Paper)
- P(Scissors)

**RPS With WoLF**

- P(Rock)
- P(Paper)
- P(Scissors)
Idea #3

- Tile Coding (a.k.a. CMACS) (Sutton & Barto 1998)
  - Space covered by overlapping and offset tilings.
  - Maps continuous (or discrete) spaces to a vector of boolean values.
  - Provides discretization and generalization.
The Task
The Task = Goofspiel

The Task = Goofspiel

- Each player plays a full suit of cards.
- Each player uses their cards (without replacement) to bid on cards from another suit.
The Task = Goofspiel

- Each player plays a full suit of cards.
- Each player uses their cards (without replacement) to bid on cards from another suit.

| $n$ | $|S|$ | $|S \times A|$ | SIZEOF($\pi$ or $Q$) | VALUE(det) | VALUE(random) |
|-----|------|---------------|---------------------|------------|--------------|
| 4   | 692  | 15150         | $\sim 59$K          | −2         | −2.5         |
| 8   | $3 \times 10^6$ | $1 \times 10^7$ | $\sim 47$M          | −20        | −10.5        |
| 13  | $1 \times 10^{11}$ | $7 \times 10^{11}$ | $\sim 2.5$T         | −65        | −28          |

- The game is very large.
- Deterministic policies are very bad.
- The random policy isn’t too bad.
### The Task = Goofspiel = 2

<table>
<thead>
<tr>
<th>My Hand</th>
<th>1</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>8</th>
<th>11</th>
<th>13</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quartiles</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Opp Hand</td>
<td>4</td>
<td>5</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
</tr>
<tr>
<td>Quartiles</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Deck</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>5</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
</tr>
<tr>
<td>Quartiles</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Card</td>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Action</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\[ \langle 1, 4, 6, 8, 13 \rangle, \]
\[ \langle 4, 8, 10, 11, 13 \rangle, \]
\[ \langle 1, 3, 9, 10, 12 \rangle, \]
\[ 11, 3 \]

\( \text{(Tile Coding)} \)

\( \text{TILES} \in \{0, 1\}^{10^6} \)

- Gradient ascent on this parameterization.
- WoLF variable learning rate on the gradient step size.
Results - Worst-Case

4 Cards

Value v. Worst-Case Opponent

Number of Training Games

WoLF
Fast
Slow
Random

13 Cards

Value v. Worst-Case Opponent

Number of Training Games

WoLF
Fast
Slow
Random
Results - While Learning

Fast

Slow

WoLF
The Task = Breakthrough
The Task = Breakthrough = 2
WARNING!
Results – Breakthrough

WARNING!

- These results are preliminary… some are only hours old.
- They involve a single run of learning in a highly stochastic learning environment.
- More experiments in progress.
Results: “To the videotape...”

Playback of learned policies in simulation and on the robots.

The robot video can be downloaded from...  
http://www.cs.cmu.edu/~mhb/research/
Omni vs Omni: Learned Policies

- LR v R: 0.55
- LL v R: 0.45
- R vs R: 0.4
- R vs LL: 0.3
- R v RL: 0.35
Results = 4

Diff vs Omni: Learned Policies

Attacker's Expected Reward

LR v R    LL v R    R vs R    R vs LL    R v RL

0.1
0.2
0.3
0.4
0.5
0.6

LR v R    LL v R    R vs R    R vs LL    R v RL

0.1
0.2
0.3
0.4
0.5
0.6
Results = 5

Diff vs Diff: Learned Policies

Attacker's Expected Reward

LR v R  LL v R  R vs R  R vs LL  R v RL
Omni vs Omni: Worst-Case

Attacker’s Expected Reward

E(\(\text{LL}\))

<table>
<thead>
<tr>
<th>A: LL</th>
<th>A: R*</th>
<th>D: LL</th>
<th>D: R</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>0.35</td>
<td>0.3</td>
<td>0.6</td>
</tr>
</tbody>
</table>
WARNING!

- These results are preliminary.... some are only hours old.
- They involve a single run of learning in a highly stochastic learning environment.
- More experiments in progress.
Big Picture

- How do we scale our (collective) algorithms to large problems with limited agents?
How do we scale our (collective) algorithms to large problems with limited agents?

- Equilibrium learning may be in trouble.
- Lagoudakis and Parr’s approximation and minimax (NIPS ’02)
- Correlated equilibria?
Big Picture

- How do we scale our (collective) algorithms to large problems with limited agents?
  - Equilibrium learning may be in trouble.
  - Lagoudakis and Parr’s approximation and minimax.
    (NIPS ’02)
  - Correlated equilibria?

- What is the objective?
Big Picture

• How do we scale our (collective) algorithms to large problems with limited agents?
  – Equilibrium learning may be in trouble.
  – Lagoudakis and Parr’s approximation and minimax. (NIPS ’02)
  – Correlated equilibria?

• What is the objective?
  – Performance during learning.
  – Generality of learned policies.
    * How can I be exploited?
    * What if everyone played this policy?