On the Agenda(s) of Research on Multi-Agent Learning
by Yoav Shoham and Rob Powers and Trond Grenager

Learning against opponents with bounded memory
by Rob Powers and Yoav Shaham

Presented by:
Ece Kamar and Philip Hendrix
April 3, 2006 CS 286r
Summary

• Stochastic Game
  – Represented by a tuple: \((N, S, \vec{A}, \vec{R}, T)\)
    where
    • \(N\) is the set of agents
    • \(S\) is the set of \(n\)-agent stage games
    • \(\vec{A}=A_1, \ldots, A_n\) with \(A_i\) the set of actions of agent \(i\)
    • \(\vec{R}=R_1, \ldots, R_n\) with \(R_i: S \times A \rightarrow R\) reward function of agent \(i\)
    • \(T: S \times A \rightarrow \Pi(S)\) stochastic transition function
Bellman’s Heritage

• Single agent Q-learning

\[
Q(s, a) \leftarrow (1 - \alpha)Q(s, a) + \alpha[R(s, a) + \gamma V(s')]
\]
\[
V(s) \leftarrow \max_{a \in A} Q(s, a)
\]

converges to optimal value function \( V^* \)

• Simple extension to multi-agent SG setting

\[
Q_i(s, a_i) \leftarrow (1 - \alpha)Q_i(s, a_i) + \alpha[R_i(s, \bar{a}) + \gamma V_i(s')]
\]
\[
V_i(s) \leftarrow \max_{a_i \in A_i} Q_i(s, a_i)
\]

Q values updated without regard of opponents’ actions
Justified if opponents’ choice of actions are stationary
Bellman’s Heritage

• Cure: Define \( Q \)-values as a function of all agents’ actions

\[
Q_i(s, \vec{a}) \leftarrow (1 - \alpha)Q_i(s, \vec{a}) + \alpha[R_i(s, \vec{a}) + \gamma V_i(s')] 
\]

Problem: How to update \( V \)?

• Maximin Q-learning

\[
V_1(s) \leftarrow \max_{P_1 \in \Pi(A_1)} \min_{a_2 \in A_2} \sum_{a_1 \in A_1} P_1(a_1)Q_1(s, (a_1, a_2)) 
\]

Problem: Motivated only for zero-sum SG
Bellman’s Heritage

• Maintain belief about the likelihood of opponents’ policies
  Update $V$ based on expectation of $Q$ values

\[ V_i(s) \leftarrow \max_{a_i} \sum_{a_{-i} \in A_{-i}} P_i(s, a_{-i}) Q_i(s, (a_i, a_{-i})) \]

• Generalization of $Q$-learning to general-sum games: 
  $Nash-Q$ learning

\[ V_i(s) \leftarrow Nash_i(Q_1(s, \bar{a}), \ldots, Q_n(s, \bar{a})) \]

$CE-Q$ learning

\[ V_i(s) \leftarrow CE_i(Q_1(s, \bar{a}), \ldots, Q_n(s, \bar{a})) \]

Problem: What if equilibriums are not unique?
Bellman’s Heritage

- Two special class of SGs:
  - *Friend class*: Q values define a globally optimal action profile
  - *Foe class*: Q values define a game with a saddle point

\[
\begin{align*}
\text{Friend: } V_1(s) &\leftarrow \max_{a_1 \in A_1, a_2 \in A_2} Q_1(s, (a_1, a_2)) \\
\text{Foe: } V_1(s) &\leftarrow \max_{P_1 \in \pi(A_1)} \min_{a_2 \in A_2} \sum_{a_1 \in A_1} P_1(a_1) Q_1(s, (a_1, a_2))
\end{align*}
\]

- *Friend* Q updates V similar to regular Q learning
- *Foe* Q updates V similar to maximin
Convergence Results

- Ability to converge is main criteria for judging performance

- Maximin-Q learning converges in the limit to the correct Q-values for any zero-sum game with infinite exploration

- Q-learners and belief-based joint action learners converge to equilibrium in common payoff games under the condition of self play and decreasing exploration

- Nash-Q learning converges to the correct Q-values for Friend or Foe games.

- CE-Q converges to Nash equilibrium in some empirical experiments

- Result: Convergence results are limited special classes of games.
Why Focus on Equilibria?

- Equilibrium identifies conditions under which learning can or should stop
- Easier to play in equilibrium as opposed to continued computation

Why not to Focus on Equilibria

- Nash equilibrium strategy has no prescriptive force
- Multiple potential equilibria
- Use of an oracle to uniquely identify an equilibria is “cheating”
- Opponent may not wish to play an equilibria
- Calculating a Nash Equilibrium for a large game can be intractable
Criteria for Learning

- Use of convergence to NE as evaluation criteria is problematic
- Bowling & Veloso propose new criteria:
  - Converge to stationary policy
    - Not necessarily Nash
  - Only terminate once best response to play of other agents found
  - During self play, learning only terminate in a stationary Nash Equilibrium
Five Agendas in Multi-Agent Learning

Descriptive agenda:
How do humans learn?

3) Figure out how humans learn with other humans
   - Show experimentally that a certain formal model agrees with people’s behavior
Five Agendas (Cont.)

1) Learn through iteration
   
   - View learning as an iterative process to compute solution concepts
     
     • Ex: Fictitious Play

Limitation of 1st and 2nd agendas:

• No agreed upon objective criterion
Five Agendas (Cont.)

Prescriptive agendas:
How should agents learn?

3) Distribute control in dynamic systems
   – need to decentralize control
   – Too difficult to have centralized control over all aspects of a real world scenario
Five Agendas (Cont.)

• Equilibrium Agenda
  – When does a vector of learning strategies form an equilibrium?
  – What class of learning strategies form equilibrium for which class of stochastic games?
  – Find strategies s.t. an agent wouldn't want to change its learning algorithm.
Five Agendas (Cont.)

1) AI agenda
   - How to design an agent for an environment
   - Environment is defined by opponents
   - Find the best learning strategy (next paper)
   - Evaluation criteria for strategy is its payoff
   - Convergence to equilibrium is valuable if helps to maximize the payoff
   - Sets bounded rationality as the starting point, results greater applicability
   - Parameterize the environment:
     • Hard computationally
     • Place bounds on stuff like priors, memory, etc.
Proposed Criteria

• Targeted Optimality
  – Against any member of the target set of opponents, the algorithm achieves within $\epsilon$ of the expected value of the best response to the actual opponent.

• Compatibility
  – During self-play, the algorithm achieves at last within $\epsilon$ of the payoff of some Nash equilibrium that is not Pareto dominated by another Nash equilibrium.

• Safety
  – Against any opponent, the algorithm always receives at least within $\epsilon$ of the security value for the game.
Environment

- Two-Players
- Repeated games with average reward
- Simultaneous moves
- Each agent tries to maximize its *average* reward
- Full game structure and payoffs are known to both agents
Bounded Memory

- Limit the opponent’s capabilities
- If opponent consider complete history, can learn nothing in a single repeated game
- Limit the available history
- Opponents play conditional strategy where their action depend on $k$ most recent periods of history
Learning against adaptive opponents

- Opponent Agent has two possible strategies
  - Tit-for-tat
  - Always Cooperate
- Agent needs to explore
- New target: Highest average value after exploration: no discounting
- Makes use of the bounded memory

Prisoner’s Dilemma

<table>
<thead>
<tr>
<th></th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>3,3</td>
<td>0,4</td>
</tr>
<tr>
<td>D</td>
<td>4,0</td>
<td>1,1</td>
</tr>
</tbody>
</table>
Explain Algorithm

- Start with teaching strategy for coordination/exploration phase
- At the end of exploration, decide:
  - If opponent in target class
    - Adopt best response
  - If opponent adopted best response to teaching
    - Continue
  - Otherwise
    - Select default strategy
• MemBR calculates best response against target set
• Godfather is the teaching strategy
• Godfather is the self-play guarantee
• Minimax is the security level

```plaintext
Set strategy = StochGodfather
for \( \tau_1 \) time steps, Play strategy
for \( \tau_2 \) time steps
  if \( (AvgValue < V_{Godfather} - \epsilon_1) \)
    With probability \( p \),
    set strategy = MemBR
Play strategy
if opponentInTargetSet()
  for \( \tau_1 \) time steps, Play MemBR
if opponentInTargetSet()
  Set bestStrategy = MemBR
else set bestStrategy = strategy
else if (strategy == StochGodfather
  AND AvgValue > V_{Godfather} - \epsilon_1)
  Set bestStrategy = StochGodfather
else set bestStrategy = MemBR
while not end of game
  if AvgValue < V_{security} - \epsilon_0
    Play maximin strategy
  else
    Play bestStrategy
```
Coordination/ exploration phase

Set strategy = StochGodfather
for $\tau_1$ time steps, Play strategy
for $\tau_2$ time steps
  if ($AvgValue < V_{Godfather} - \epsilon_1$)
    With probability $p$,
    set strategy = MemBR
  Play strategy
if opponentInTargetSet() 
  for $\tau_1$ time steps, Play MemBR
  if opponentInTargetSet() 
    Set bestStrategy = MemBR
  else set bestStrategy = strategy
else if (strategy == StochGodfather
  AND $AvgValue > V_{Godfather} - \epsilon_1$)
  Set bestStrategy = StochGodfather
else set bestStrategy = MemBR
while not end of game
  if $AvgValue < V_{security} - \epsilon_0$
    Play maximin strategy
  else
    Play bestStrategy
Set strategy = StochGodfather
for $\tau_1$ time steps, Play strategy
for $\tau_2$ time steps
  if ($AvgValue < V_{Godfather} - \epsilon_1$)
    With probability $p$,
    set strategy = MemBR
  Play strategy
if opponentInTargetSet()
  for $\tau_1$ time steps, Play MemBR
if opponentInTargetSet()
  Set bestStrategy = MemBR
else set bestStrategy = strategy
else if (strategy == StochGodfather
  AND $AvgValue > V_{Godfather} - \epsilon_1$)
  Set bestStrategy = StochGodfather
else set bestStrategy = MemBR
while not end of game
  if $AvgValue < V_{security} - \epsilon_0$
    Play maximin strategy
  else
    Play bestStrategy
If opponent adopted best response to teaching, continue

Set strategy = StochGodfather
for $\tau_1$ time steps, Play strategy
for $\tau_2$ time steps
    if $(\text{AvgValue} < V_{\text{Godfather}} - \epsilon_1)$
        With probability $p$,
        set strategy = MemBR
    Play strategy
if opponentInTargetSet()
    for $\tau_1$ time steps, Play MemBR
if opponentInTargetSet()
    Set bestStrategy = MemBR
else set bestStrategy = strategy
else if (strategy == StochGodfather
    AND $\text{AvgValue} > V_{\text{Godfather}} - \epsilon_1$)
    Set bestStrategy = StochGodfather
else set bestStrategy = MemBR
while not end of game
    if $\text{AvgValue} < V_{\text{security}} - \epsilon_0$
        Play maximin strategy
    else
        Play bestStrategy
Otherwise, adopt the default strategy

Set strategy = StochGodfather
for \( \tau_1 \) time steps, Play strategy
for \( \tau_2 \) time steps
  if \( (AvgValue < V_{Godfather} - \epsilon_1) \)
    With probability \( p \),
    set strategy = MemBR
  Play strategy
if opponentInTargetSet()
  for \( \tau_1 \) time steps, Play MemBR
if opponentInTargetSet()
  Set bestStrategy = MemBR
else set bestStrategy = strategy
else if (strategy == StochGodfather
  AND AvgValue > V_{Godfather} - \epsilon_1)
  Set bestStrategy = StochGodfather
else set bestStrategy = MemBR
while not end of game
  if AvgValue < V_{security} - \epsilon_0
    Play maximin strategy
  else
    Play bestStrategy
If payoff is below security level, adopt security level strategy

Set strategy = StochGodfather
for $\tau_1$ time steps, Play strategy
for $\tau_2$ time steps
   if $(AvgValue < V_{Godfather} - \epsilon_1)$
      With probability $p$,
      set strategy = MemBR
   Play strategy
if opponentInTargetSet()
   for $\tau_1$ time steps, Play MemBR
if opponentInTargetSet()
   Set bestStrategy = MemBR
else set bestStrategy = strategy
else if (strategy == StochGodfather
   AND AvgValue > V_{Godfather} - \epsilon_1)
   Set bestStrategy = StochGodfather
else set bestStrategy = MemBR
while not end of game
   if AvgValue < V_{security} - \epsilon_0
      Play maximin strategy
   else
      Play bestStrategy
Proposed Criteria

• Targeted Optimality
  – Against any member of the target set of opponents, the algorithm achieves within $\epsilon$ of the expected value of the best response to the actual opponent.

• Compatibility
  – During self-play, the algorithm achieves at last within $\epsilon$ of the payoff of some Nash equilibrium that is not Pareto dominated by another Nash equilibrium.

• Safety
  – Against any opponent, the algorithm always receives at least within $\epsilon$ of the security value for the game.
Talk about thm 1

**Theorem 1** Our algorithm, Manipulator, satisfies the three properties stated in the introduction for the class of conditional strategies with bounded memory $k$, after a training period depending on $\frac{|A|^k}{\lambda}$, where $\lambda$ is the minimum probability the opponent assigns to any action, or $\lambda = 1$ for opponents that condition only on the agent’s actions.

- No proof, just like the algorithm
- Exploration grows exponentially in the size of the bounded memory
- Exploration becomes unbounded if added the requirement of a minimum probability of playing any given action
- Exploration can be limited for small memory and high $\lambda$

Potential discounted sum implementation
Empirical Results

Figure 3: Average value for last 20K rounds (of 200K) across selected games in GAMUT. Game payoffs range from -1 to 1.
Empirical Results

self play

Figure 3: Average value for last 20K rounds (of 200K) across selected games in GAMUT. Game payoffs range from -1 to 1.
Empirical Results

Figure 4: Percent of maximum value for last 20K rounds (of 200K) averaged across all opponents for selected games in GAMUT. The rewards were divided by the maximum reward achieved by any agent to make visual comparisons easier.
Conclusion

• Limitations (self criticism)
  – Criteria only defined for games with two players
  – Criteria are only defined for repeated games (rather than general stochastic games)
  – Criteria defined for games in which an agent only cares about its average reward (rather than discounted sum)
  – Agent needs perfect observations of opponent’s actions
  – The algorithm needs to know all of the payoffs for each agent from the beginning of the game.
Conclusion

• Achievements
  – Gives an algorithm for bounded agents
  – Considers adaptive opponents
  – Presents detailed empirical results and comparisons
  – Paper ends with paper good self criticism