Outline

A. Introduction

B. Single Agent Learning

C. Game Theory

D. Multiagent Learning

E. Future Issues and Open Problems
Algorithms for Multiagent Learning

- Equilibrium Learners

- Regret Minimizing Algorithms

- Best Response Learners
  - Q-Learning
  - Opponent Modeling Q-Learning
  - Gradient Ascent
  - WoLF

- Learning to Coordinate
What’s the Goal?

- Learn a best response, if one exists.
- Make some other guarantees. For example,
  - Convergence of payoffs or policies.
  - Low regret or at least minimax optimal.
- If best response learners converge against each other, then it must be to a Nash equilibrium.
Q-Learning

• ... or any MDP learning algorithm.

• The most commonly used approach to learning in multiagent systems.
Q-Learning

- ... or any MDP learning algorithm.

- The most commonly used approach to learning in multiagent systems. And, not without success.
Q-Learning

• ... or any MDP learning algorithm.

• The most commonly used approach to learning in multiagent systems. And, not without success.

• If it is the only learning agent...

  – Recall, if the other agents are using a stationary strategy, it becomes an MDP.

  – Q-learning will converge to a best-response.

• Otherwise, requires on-policy learning.
Q-Learning

- Dominance solvable games.
Q-Learning

- Dominance solvable games.

- It has also been successfully applied to...
  
  - Team games.
    (Sen et al. 1994; Claus & Boutilier, 1998)

  - Games with pure strategy equilibria.
    (Tan, 1993; Crites & Sandholm, 1995, Bowling, 2000)

  - Adversarial games.
    (Tesauro, 1995; Uther, 1997)

  * TD-Gammon remains one of the most convincing successes of reinforcement learning.
Opponent Modeling Q-Learning

(Uther, 1997) and others.

- Fictitious play in stochastic games using approximation.
- Choose action that maximizes,

\[
V(s) = \max_{a_i} \sum_{a_{-i}} \frac{C(s, a_{-i})}{C(s)} Q(s, \langle a_i, a_{-i} \rangle).
\]

- Update opponent model and Q-values,

\[
\begin{align*}
Q(s, a) & \leftarrow Q(s, a) + \alpha \left( R(s, a) + \gamma \sum_{s'} T(s, a, s') V(s') - Q(s, a) \right) \\
C(s, a_{-i}) & \leftarrow C(s, a_{-i}) + 1 \\
C(s) & \leftarrow C + 1.
\end{align*}
\]
Opponent Modeling Q-Learning

- Superficially less naive than Q-learning.
  - Recognizes the existence of other agents.
  - But assumes they use a stationary policy.

- Similar results to Q-learning, but faster approximation.

(Uther, 1997) — Hexcer

<table>
<thead>
<tr>
<th></th>
<th>First 50000 Games</th>
<th></th>
<th>Second 50000 Games</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MMQ</td>
<td>Q</td>
<td>OMQ</td>
</tr>
<tr>
<td>MMQ</td>
<td>—</td>
<td>27%</td>
<td>32%</td>
</tr>
<tr>
<td>Q</td>
<td>73%</td>
<td>—</td>
<td>40%</td>
</tr>
<tr>
<td>OMQ</td>
<td>68%</td>
<td>60%</td>
<td>—</td>
</tr>
</tbody>
</table>
Gradient Ascent

• Compute gradient of value with respect to the player’s strategy.

• Adjust policy to increase value.

• Single-agent learning (parameterized policies).
  (Williams, 1993; Sutton et al., 2000, Baxter & Bartlett, 2000)

• Multiagent Learning.
Infinitesimal Gradient Ascent

(Singh, Kearns, & Mansour, 2000)

\[ R = \begin{bmatrix} r_{11} & r_{12} \\ r_{21} & r_{22} \end{bmatrix} \quad C = \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix} \]

\[ V_r(\alpha, \beta) = \alpha \beta r_{11} + \alpha (1 - \beta) r_{12} + (1 - \alpha) \beta r_{21} + (1 - \alpha)(1 - \beta) r_{22} \]

\[ = u \alpha \beta + \alpha (r_{12} - r_{22}) + \beta (r_{21} - r_{22}) + r_{22} \]

where,

\[ u = r_{11} - r_{12} - r_{21} + r_{22} \]
\[
\frac{\partial V_r(\alpha, \beta)}{\partial \alpha} = \beta u - (r_{22} - r_{12}) \\
\frac{\partial V_c(\alpha, \beta)}{\partial \beta} = \beta u' - (c_{22} - c_{21})
\]

\[
\alpha^{k+1} = \alpha^k + \eta \frac{\partial V_r(\alpha^k, \beta^k)}{\partial \alpha^k} \\
\beta^{k+1} = \beta^k + \eta \frac{\partial V_r(\alpha^k, \beta^k)}{\partial \beta^k}
\]
IGA — Theorem

(Singh et al., 2000)

**Theorem.** If both players follow Infinitesimal Gradient Ascent (IGA), where $\eta \to 0$, then their strategies will converge to a Nash equilibrium OR the average payoffs over time will converge in the limit to the expected payoffs of a Nash equilibrium.
IGA — Proof

\[
\begin{bmatrix}
\frac{\partial \alpha}{\partial t} \\
\frac{\partial \beta}{\partial t}
\end{bmatrix} = \begin{bmatrix} 0 & u \\ u' & 0 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} + \begin{bmatrix} (r_{12} - r_{22}) \\ (c_{21} - c_{22}) \end{bmatrix}.
\]

\[
\begin{array}{c}
U \text{ is not invertible} \\
u = 0 \text{ or } u' = 0
\end{array}
\]

\[
\begin{array}{c}
U \text{ has real eigenvalues} \\
uu' < 0
\end{array}
\]

\[
\begin{array}{c}
U \text{ has imaginary eigenvalues} \\
uu' > 0
\end{array}
\]
IGA — Summary

- One of the first convergence proofs for a payoff maximizing multiagent learning algorithm.
- Expected payoffs do not necessarily converge.
(Zinkevich, 2003)

- Generalized Infinitesimal Gradient Ascent (GIGA).
  - At time $k$, select actions according to $\sigma_i^{k+1}$.
  - After observing others select $a_{-i}$,

\[
\sigma_i^{k+1} = \arg\min_{\sigma_i \in \mathcal{PD}(A_i)} \left\| \sigma_i^k + \eta R(\langle \cdot, a_{-i} \rangle) - \sigma_i \right\|
\]

i.e., step the probability distribution toward immediate reward, then project into a valid probability space.
• GIGA is identical to IGA for two-player, two-action games, while approximating the gradient.

\[ \alpha^{k+1} = \alpha^{k+1} + \eta (\beta u + r_{12} - r_{22}) \]

\[ \sigma_i^{k+1} = \arg\min_{\sigma_i \in PD(A_i)} ||\sigma_i^k + \eta R(\langle \cdot, a_{-i} \rangle) - \sigma_i|| \]

• GIGA is universally consistent!
• Assumption: Policy gradient is bounded.
GIGA — Intuition

- Assumption: Policy gradient is bounded.
GIGA — Intuition

- Assumption: Policy gradient is bounded.
• Assumption: Policy gradient is bounded.
WoLF

(Bowling & Veloso, 2002, 2003)

• Modify gradient ascent learning to converge.

• Vary the speed of learning: Win or Learn Fast.
  – If winning, learn cautiously.
  – If losing, learn quickly.

• Algorithms: WoLF-IGA, WoLF-PHC, GraWoLF.
\[
\alpha^{k+1} = \alpha^k + \eta \ell^k_r \frac{\partial V_r(\alpha^k, \beta^k)}{\partial \alpha} \\
\beta^{k+1} = \beta^k + \eta \ell^k_c \frac{\partial V_r(\alpha^k, \beta^k)}{\partial \beta}
\]

\[
\ell^k_{r,c} \in [\ell^\text{min}, \ell^\text{max}] > 0
\]
WoLF-IGA

\[ \alpha^{k+1} = \alpha^k + \eta \ell_r^k \frac{\partial V_r(\alpha^k, \beta^k)}{\partial \alpha} \]

\[ \beta^{k+1} = \beta^k + \eta \ell_c^k \frac{\partial V_r(\alpha^k, \beta^k)}{\partial \beta} \]

WoLF = Win or Learn Fast!

\[ \ell_r^k = \begin{cases} \ell_{\text{min}} & \text{WINNING if } V_r(\alpha^k, \beta^k) > V_r(\alpha^e, \beta^k) \\ \ell_{\text{max}} & \text{LOSING otherwise} \end{cases} \]

\[ \ell_c^k = \begin{cases} \ell_{\text{min}} & \text{WINNING if } V_c(\alpha^k, \beta^k) > V_c(\alpha^k, \beta^e) \\ \ell_{\text{max}} & \text{LOSING otherwise} \end{cases} \]
WoLF-IGA — Theorem

**Theorem.** If both players follow WoLF-IGA, where $\eta \to 0$, and $\ell^{\text{max}} > \ell^{\text{min}}$, then their strategies will converge to a Nash equilibrium.
WoLF-IGA — Proof

\[
\begin{bmatrix}
\frac{\partial \alpha}{\partial t} \\
\frac{\partial \beta}{\partial t}
\end{bmatrix} = \begin{bmatrix}
0 & u \ell_r(t) \\
u' \ell_c(t) & 0
\end{bmatrix} \begin{bmatrix}
\alpha \\
\beta
\end{bmatrix} + \begin{bmatrix}
\ell_r(t)(r_{12} - r_{22}) \\
\ell_c(t)(c_{21} - c_{22})
\end{bmatrix}.
\]

Lemma. Qualitative dynamics is unchanged.
Lemma. Sign of gradient is unchanged.
**WoLF-IGA — Proof**

**Lemma.** A player’s strategy is moving away from the equilibrium if and only if they are “winning”.

I.e., $V_r(\alpha, \beta) - V_r(\alpha^*, \beta) > 0 \iff (\alpha - \alpha^*)\frac{\partial V_r(\alpha, \beta)}{\partial \alpha} > 0$.

**Proof:**

\[
V_r(\alpha, \beta) - V_r(\alpha^*, \beta) \\
(\alpha \beta u + \alpha(r_{12} - r_{22}) + \beta(r_{21} - r_{22}) + r_{22}) - \\
(\alpha^* \beta u + \alpha^*(r_{12} - r_{22}) + \beta(r_{21} - r_{22}) + r_{22}) \\
(\alpha - \alpha^*)\beta u + (\alpha - \alpha^*)(r_{12} - r_{22}) \\
(\alpha - \alpha^*)(\beta u + (r_{12} - r_{22})) \\
(\alpha - \alpha^*)\frac{\partial V_r(\alpha, \beta)}{\partial \alpha}.
\]
WoLF-IGA — Proof
WoLF-IGA — Proof
WoLF-IGA — Proof
WoLF-IGA — Proof
WoLF-IGA — Proof — Summary

**Theorem.** If both players follow WoLF gradient ascent with $\ell_{\text{max}} > \ell_{\text{min}}$ then their strategies will converge to a Nash equilibrium.

\[ U \text{ is not invertible} \quad \quad u = 0 \text{ or } u' = 0 \]

\[ U \text{ has real eigenvalues} \quad \quad uu' < 0 \]

\[ U \text{ has imaginary eigenvalues} \quad \quad uu' > 0 \]
WoLF-IGA — Corollary

**Corollary.** If both players follow the WoLF-IGA algorithm but with different $\ell_{\text{min}}$ and $\ell_{\text{max}}$, then their strategies will converge to a Nash equilibrium if,

$$\frac{\ell_{\text{min}}}{\ell_{\text{max}}} < 1.$$ 

Specifically, WoLF-IGA (with $\ell_{\text{max}} > \ell_{\text{min}}$) versus IGA ($\ell_{\text{max}} = \ell_{\text{min}}$) will converge to a Nash equilibrium.
Practical Versions of WoLF

- WoLF Policy Hill-Climbing (WoLF-PHC)
  - Combines WoLF with a Q-learning like algorithm that can learn stochastic policies.
  - Shown empirically to converge in a variety of stochastic games.

- Gradient-Based WoLF (GraWoLF)
  - Combines WoLF with a policy gradient technique.
  - Learned policies in goofspiel and an adversarial robot task.
Algorithms for Multiagent Learning

• Equilibrium Learners

• Best Response Learners

• Learning to Coordinate
  – ILs and JALs
  – Brafman and Tennenholtz
  – Optimal Adaptive Learning
ILs and JALs

(Claus & Boutilier, 1998)

\[
\begin{align*}
\text{ILs} & = \text{Q-Learning} \\
\text{JALs} & = \text{Opponent Modelling Q-learning}
\end{align*}
\]

- Guaranteed to converge to a Nash equilibrium.
- Not necessarily an optimal Nash equilibrium.

\[
R_{\{1,2\}} = \begin{pmatrix}
A & B & C \\
A & 10 & 0 & -k \\
B & 0 & 2 & 0 \\
C & -k & 0 & 10
\end{pmatrix}
\]
Optimal Adaptive Learning

(Wang & Sandholm, 2002)

- Learn an optimal Nash equilibrium.

- Q-Learning plus a coordinating mechanism.
  - Learn Q-values.
  - Construct a per-state virtual game from Q-values.
  - Use biased adaptive play on the virtual games.
    - Adaptive play “fixes” fictitious play. (Young, 1993)
    - Biased adaptive play “fixes” adaptive play.
Optimal Adaptive Learning

- Virtual Game

\[
Q_{\{1,2\}} = \begin{pmatrix} A & B \\ A & B \end{pmatrix} = \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix} \Rightarrow Q_{\{1,2\}} = \begin{pmatrix} A & B \\ A & B \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}
\]

\[
Q_{\{1,2\}} = \begin{pmatrix} A & B \\ A & B \end{pmatrix} = \begin{pmatrix} 3 & 3 \\ 0 & 3 \end{pmatrix} \Rightarrow Q_{\{1,2\}} = \begin{pmatrix} A & B \\ A & B \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}
\]
Optimal Adaptive Learning

- Adaptive Play — Adds Randomness
  - Randomize among all best-responses.
  - Sample randomly from past history.

\[
Q_{\{1,2\}} = \begin{pmatrix} A & B \\ A & 0 \\ B & 1 \\ B & 0 \end{pmatrix}
\]

- Overcomes the pathological cases.
Optimal Adaptive Learning

• Biased Adaptive Play — Removes Randomness

\[ Q_{\{1,2\}} = \begin{pmatrix} A & B \\ A & \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} & B \end{pmatrix} \]

– Deterministically choose most recent best response. (under certain circumstances)

– Can converge to weak Nash equilibria.

• Guaranteed to converge to an optimal equilibrium.
Brafman and Tennenholtz

(Brafman & Tennenholtz, 2002, 2003)

- Learn an optimal Nash equilibrium.

- Learns it in polynomial time.
Brafman and Tennenholtz

- Normal-Form Games — Simple Solution
  - Randomize over all actions for $T$ steps.
  - Select the globally optimal joint action.
  - Key Fact: Choose $T$ large enough to make sure all joint actions are played with high probability $(1 - \delta)$.

$$T = O \left( k^4 \log \left( \frac{k}{\delta} \right) \right)$$
Brafman and Tennenholtz

• Stochastic Games — Less Simple

  – Relies on an MDP algorithm: \( R\text{-MAX} \)
    * Near optimal, polynomial time algorithm.
    * Deterministic.

  – Each player runs \( R\text{-MAX} \) on the joint action space.

  – The players then select their portion of the selected joint action.
Brafman and Tennenholtz

• Assumptions. . .

  – Agents and action sets are ordered and known.

  – Guarantees agents select the same joint action.

  – Assumption can be relaxed. . .
    
    • Loop over all possible action set sizes.
    • Guess $m$ random ordering of agents.
    • Run algorithm
    • Choose learned policy with best reward.
Outline

A. Introduction

B. Single Agent Learning

C. Game Theory

D. Multiagent Learning

E. Future Issues and Open Problems
   - Graphical Games
   - Equilibria as a Solution Concept
Algorithms in Equilibrium
(Brafman & Tennenholtz, 2003; Littman & Stone, 2003)

• Learning a Nash equilibrium is unimportant.
• Algorithms are themselves (non-Markovian) strategies.
• Algorithms themselves should be in equilibrium.
Algorithms in Equilibrium

(Brafman & Tennenholtz, 2003; Littman & Stone, 2003)

- Learning a Nash equilibrium is unimportant.
- Algorithms are themselves (non-Markovian) strategies.
- Algorithms themselves should be in equilibrium.
- Questions...
  - What about Folk Theorems?
  - What about learning a “learning” strategy through repeated play? Is this an infinite regress?
Acknowledgements

We wish to thank the following people whose insights, explanations, and/or slides have found their way into this tutorial.

- Michael Kearns
- Amy Greenwald
- Gerry Tesauro
- Will Uther
Questions and Discussion