A selection of MAS learning techniques based on RL

Ann Nowé
Content

Single stage setting
- Common interest (Claus & Boutilier, Kapetanakis & Kudenko)
- Conflicting interest (Based on LA)
Key questions

Are RL algorithms guaranteed to converge in MAS settings? If so, do they converge to (optimal) equilibria?

Are there differences between agents that learn as if there are no other agents (i.e. use single agents RL algorithms) and agents that attempt to learn both the values of specific joint actions and the strategies employed by other agents?

How are rates of convergence and limit points influenced by the system structure and action selection strategies?
Simple single stage
common deterministic interest game

<table>
<thead>
<tr>
<th></th>
<th>$a_0$</th>
<th>$a_1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$b_0$</td>
<td>$x$</td>
<td>0</td>
</tr>
<tr>
<td>$b_1$</td>
<td>0</td>
<td>$y$</td>
</tr>
</tbody>
</table>

If $x > y > 0$, $(a_0, b_0)$ and $(a_1, b_1)$ 2 equilibria
first one is optimal
If $x = y > 0$ equilibrium selection problem

Super RL agent (Q-values for joint actions and joint action selection)
No challenge, equivalent to single agent learning

Joint action learners (Q-values for joint actions, actions are selected independently)

Independent learners (Q-values for individual actions, actions are selected independently)
Simple single stage
common deterministic interest game

Joint action learners (Q-values for joint actions, actions are selected independently)

Use e.g. Q-learning to learn $Q(a_0, b_0)$, $Q(a_0, b_1)$, $Q(a_1, b_0)$ and $Q(a_1, b_1)$

Assumption: actions taken by the other agents can be observed.

Action selection for individual agents:

the quality of an individual action depends on the action taken by
the other agent-> maintain beliefs about strategies of other agents.

$$EV(a^i) = \sum_{a^{-i} \in A_{-i}} Q(a^{-i} \cup \{a^i\}) \prod_{j \neq i} \{Pr_{a^{-i}[j]}\}$$
Simple single stage
common deterministic interest game

Independent learners (Q-values for joint action, actions are selected independently)

Use e.g. Q-learning to learn $Q(a_0)$, $Q(a_1)$, $Q(b_0)$ and $Q(b_1)$
No need to observe actions taken by other agents.

Action selection for individual agents:
Exploration strategy is crucial
(Random not OK, Boltzmann with decreasing $T$ is Ok)
Simple single stage
Comparing Independent Learners and Joint action learners

<table>
<thead>
<tr>
<th></th>
<th>$a_0$</th>
<th>$a_1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$b_0$</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>$b_1$</td>
<td>0</td>
<td>10</td>
</tr>
</tbody>
</table>

Figure 1: Convergence of coordination for ILs and JALs (averaged over 100 trials).
The penalty game

<table>
<thead>
<tr>
<th></th>
<th>$a_0$</th>
<th>$a_1$</th>
<th>$a_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$b_0$</td>
<td>10</td>
<td>0</td>
<td>$k$</td>
</tr>
<tr>
<td>$b_1$</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>$b_2$</td>
<td>$k$</td>
<td>0</td>
<td>10</td>
</tr>
</tbody>
</table>

$k < 0$

3 Nash Equilibria, 2 optimal

![Graph showing likelihood of convergence to optimal joint actions in dependency of penalty $k$.](image)

Figure 2: Likelihood of convergence to opt. equilibrium as a function of penalty $k$ (averaged over 100 trials).

Similar results hold for IL with decreasing exploration.
Climbing game

<table>
<thead>
<tr>
<th></th>
<th>(a_0)</th>
<th>(a_1)</th>
<th>(a_2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(b_0)</td>
<td>11</td>
<td>-30</td>
<td>0</td>
</tr>
<tr>
<td>(b_1)</td>
<td>-30</td>
<td>7</td>
<td>6</td>
</tr>
<tr>
<td>(b_2)</td>
<td>0</td>
<td>0</td>
<td>5</td>
</tr>
</tbody>
</table>

2 Nash Equilibria, 1 optimal

Figure 3: \(A\)'s strategy in climbing game

Figure 4: \(B\)'s strategy in climbing game

initial temperature 10000 is decayed at rate 0.995
Climbing game

<table>
<thead>
<tr>
<th></th>
<th>$a_0$</th>
<th>$a_1$</th>
<th>$a_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$b_0$</td>
<td>11</td>
<td>-30</td>
<td>0</td>
</tr>
<tr>
<td>$b_1$</td>
<td>-30</td>
<td>7</td>
<td>6</td>
</tr>
<tr>
<td>$b_2$</td>
<td>0</td>
<td>0</td>
<td>5</td>
</tr>
</tbody>
</table>

2 Nash Equilibria, 1 optimal

Figure 5: Joint actions in climbing game

initial temperature 10000 is decayed at rate 0.995
**Optimistic Boltzmann (OB):** For agent $i$, action $a_i \in A_i$, let $MaxQ(a_i) = \max_{\Pi_i} Q(\Pi_i, a_i)$. Choose actions with Boltzmann exploration (another exploitive strategy would suffice) using $MaxQ(a_i)$ as the value of $a_i$.

**Weighted OB (WOB):** Explore using Boltzmann using factors $MaxQ(a_i) \cdot Pr_i$ (optimal match $\Pi_i$ for $a_i$).

**Combined:** Let $C(a_i) = \rho \cdot MaxQ(a_i) + (1 - \rho)EV(a_i)$, for some $0 \leq \rho \leq 1$. Choose actions using Boltzmann exploration with $C(a_i)$ as value of $a_i$.

<table>
<thead>
<tr>
<th></th>
<th>$a_0$</th>
<th>$a_1$</th>
<th>$a_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$b_0$</td>
<td>10</td>
<td>0</td>
<td>$k$</td>
</tr>
<tr>
<td>$b_1$</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>$b_2$</td>
<td>$k$</td>
<td>0</td>
<td>10</td>
</tr>
</tbody>
</table>

Figure 6: Sliding avg. reward in the penalty game
Content

Single stage setting
- Common interest (Claus & Boutilier, Kapetanakis & Kudenko)
- Conflicting interest (Based on LA)
Observation:
The setting of the temperature in the Boltzmann strategy for independent learners is crucial.
Converge to some equilibrium, but not necessarily the optimal.
FMQ : Frequency Maximum Q value heuristic

\[ EV(a) = Q(a) + c \times \text{freq}(\max R(a)) \times \max R(a) \]

- Controls weight of heuristic
- Fraction of time maxR(a)
- Max reward so far for action a

\[ p(a) = \frac{\frac{EV(a)}{T}}{\sum_{action' \in A_i} e^{\frac{EV(action')}{T}}} \]

\[ T(x) = e^{-sx} \times \max_{temp} + 1 \]

x number of iterations
s decay parameter
**FMQ Heuristic** (Kapetanakis & Kudenko)

<table>
<thead>
<tr>
<th></th>
<th>$a_0$</th>
<th>$a_1$</th>
<th>$a_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$b_0$</td>
<td>11</td>
<td>-30</td>
<td>0</td>
</tr>
<tr>
<td>$b_1$</td>
<td>-30</td>
<td>7</td>
<td>6</td>
</tr>
<tr>
<td>$b_2$</td>
<td>0</td>
<td>0</td>
<td>5</td>
</tr>
</tbody>
</table>

The climbing game

![Graph showing the probability of convergence to the optimal action](image)

Likelihood of convergence to the optimal joint action (average over 1000 trials)
FMQ Heuristic (Kapetanakis & Kudenko)

\[
\begin{array}{ccc}
  a_0 & a_1 & a_2 \\
  b_0 & 10 & 0 & k \\
  b_1 & 0 & 2 & 0 \\
  b_2 & k & 0 & 10 \\
\end{array}
\]

\( k < 0 \)

The penalty game

- Probability of convergence to the optimal action
- Number of interactions

Likelihood of convergence to the optimal joint action (average over 1000 trials), \( k = 0 \)
**FMQ Heuristic** (Kapetanakis & Kudenko)

\[ \begin{array}{ccc} 
  a_0 & a_1 & a_2 \\
  b_0 & 10 & 0 & k \\
  b_1 & 0 & 2 & 0 \\
  b_2 & k & 0 & 10 \\
\end{array} \]

- **k < 0**

The penalty game

![Graph showing probability of convergence to the optimal action vs. penalty k](image)

Likelihood of convergence to the optimal joint action (average over 1000 trials, in function of k)
The FMQ Heuristic is not very robust in stochastic reward games

<table>
<thead>
<tr>
<th></th>
<th>$a_0$</th>
<th>$a_1$</th>
<th>$a_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$b_0$</td>
<td>10/12</td>
<td>5/-65</td>
<td>8/-8</td>
</tr>
<tr>
<td>$b_1$</td>
<td>5/-65</td>
<td>14/0</td>
<td>12/0</td>
</tr>
<tr>
<td>$b_2$</td>
<td>5/-5</td>
<td>5/-5</td>
<td>10/0</td>
</tr>
</tbody>
</table>

GOAL is stochastic

Improvement: commitment sequences

The stochastic climbing game (50%)
Commitment Sequences (Kapetanakis & Kudenko)

- motivation: difficult to distinguish between the two sources of uncertainty (other agents, multiple rewards)
- definition: a commitment sequence is some list of time slots for which an agent is committed to taking the same action
- condition: an exponentially increasing time interval between successive time slots

Sequence 1: (1,3,6,10,15,22, …)
Sequence 2: (2,5,9,14,20,28, …)
Sequence 3: (4, …)

assumptions:
1. common global clock
2. common protocol for defining commitment sequences
Content

Single stage setting
- Common interest (Claus & Boutilier, Kapetanakis&Kudenko)
- Conflicting interest (Based on LA)
Learning Automata

Basic Definition
- Learning automaton as a policy iterator
- Overview of Learning Schemes
- Convergence issues

Automata Games
- Definition
- Analytical Results
- Dynamics
- ESRL + Examples
Learning automata

Single Stage, Single Agent

- Environment
- Learning Automaton
- Action
- Reinforcement
Learning automata

Single Stage, Single Agent
Assume binary feedback, and L actions
When feedback signal is positive,

\[ p_i(k + 1) = p_i(k) + a[1 - p_i(k)] \] if \( i \)th action is taken at time \( k \)
\[ p_j(k + 1) = (1 - a)p_j(k), \text{ for all } j \neq i \]
with \( a \) in \( ]0,1[ \)

When feedback signal is negative,

\[ p_i(k + 1) = (1 - b)p_i(k), \text{ if } i \text{th action is taken at time } k \]
\[ p_j(k + 1) = b/(l - 1) + (1 - b)p_j(k), \text{ for all } j \neq i \]
with \( b \) in \( ]0,1[ \)

**Reward-penalty,** \( L_{R-P} \)

**Reward-\( \varepsilon \) penalty,** \( L_{R-\varepsilon P} \) \( b << a \)
Learning automata, cont.

When updates only happen at positive feedback, (or \( b = 0 \))

\[
p_i(k+1) = p_i(k) + a[1 - p_i(k)] \quad \text{if } i^{th} \text{ action is taken at time } k
\]

\[
p_j(k+1) = (1-a)p_j(k), \text{ for all } j \neq i
\]

Reward-in-action, \( L_{R-I} \)

Some terminology:

- Binary feedback: P-model
- Discrete valued feedback: Q-model
- Continuous valued feedback: S-model
- Finite action Learning Automata: FALA
- Continuous action Learning Automata: CALA
General S-model

Reward penalty, $L_{R-P}$

\[ p_i(k + 1) = p_i(k) + a \cdot r(k)(1 - p_i(k)) - b \cdot (1 - r(k))p_i(k), \text{ with } i \text{ the action taken} \]
\[ p_j(k + 1) = p_j(k) - a \cdot r(k)p_j(k) + b \cdot (1 - r(k))\left[(1 - 1) - p_j(k)\right], \text{ for all } j \neq i \]

$r(k)$ real valued reward signal in $[0,1]$

If $b << a :$ Reward- $\epsilon$ penalty, $L_{R-\epsilon P}$

If $b = 0 :$ Reward-in-action, $L_{R-I}$
Learning automata, a simulation

Action selection for LA is implicit, based on the action probabilities

\[ L_{R-I} (a=0.1) \]
\[ L_{R-p} (a=0.1, b=0.005, (\gamma=20)) \]
\[ L_{R-p} (a=0.1, b=0.01, (\gamma=10)) \]
\[ L_{R-p} (a=0.1, b=0.05, (\gamma=5)) \]
\[ L_{R-p} (a=b=0.1, (\gamma=1)) \]
\[ L_{R-p} (a=b=0.01, (\gamma=1)) \]

(\gamma=a/b)

2 actions
reward probabilities:
c_1 = 0.6, c_2 = 0.2

Iteration steps

\[ \hat{P}_1 \]
Learning automata, a simulation

- 5 actions
- Reward probabilities:
  - $c_1 = 0.35$, $c_2 = 0.8$, $c_3 = 0.5$, $c_4 = 0.6$, $c_5 = 0.15$

Diagram shows:
- $L_{R-I}$ with $a = 0.02$
- $L_{R-p}$ with $a = 0.02$, $b = 0.002$, $\gamma = 10$
- $L_{R-p}$ with $a = b = 0.02$, $\gamma = 1$
Convergence properties of LA single state, single automaton

$L_{R-I}$ and $L_{R-\epsilon P}$ are $\epsilon$-optimal in stationary environments:

$$\liminf_{k \to \infty} p_i(k) > 1 - \epsilon', \quad w.p.1.$$  

We can make the probability of the best action converge arbitrarily close to 1

$$\lim_{k \to \infty} W(k) > d_i - \epsilon', \quad w.p.1.$$  

We can let the average reward converge arbitrarily close to the highest expected reward

LR-P is not $\epsilon$-optimal, but Expedient:

$$\lim_{k \to \infty} W(k) > W(0)$$  

Performs strictly better than a pure chance automaton

$W(K)$ is the average accumulated reward

$D_l$ the expected reward of the best action
Learning Automata

Basic Definition
- Learning automaton as a policy iterator
- Overview of Learning Schemes
- Convergence issues

Automata Games
- Definition
- Analytical Results
- Dynamics
- ESRL + Examples
Automata Games

Single Stage, Multi-Automata

Environment

a1, a2, a3,...

r1, r2, r3, r...

Learning Automaton 1

Learning Automaton 2

Learning Automaton 3

Learning Automaton...
Automata Games

(Narendra and Wheeler, 1989)

Players in an n-person non-zero sum game who use independently a reward-inaction update scheme with an arbitrarily small step size will always converge to a pure equilibrium point.

If the game has a pure NE, the equilibrium point will be one of the pure NE. Convergence to Pareto Optimal (Nash) Equilibrium not guaranteed.

=> Coordinated exploration will be necessary
Category 2: Battle of the sexes

Paths induced by a linear reward -inaction LA.
Starting points are chosen randomly
x-axis = prob. of the first player to play Bach
y-axis = prob. of the second player to play Bach

(Tuyts ’04)
Exploring selfish Reinforcement Learners ESRL

Basic idea: 2 phases

- Exploration: Be Selfish
  - Independent Learning
  - Convergence to different NE and Pareto optimal non-NE

- Synchronization: Be Social
  - Exclusion phase: shrink the action space by excluding an action

(Verbeeck ’04)
**ESRL and common interest games**

### The Penalty Game

<table>
<thead>
<tr>
<th></th>
<th>b₁</th>
<th>b₂</th>
<th>b₃</th>
</tr>
</thead>
<tbody>
<tr>
<td>a₁</td>
<td>10,10</td>
<td>0,0</td>
<td>k,k</td>
</tr>
<tr>
<td>a₂</td>
<td>0,0</td>
<td>2,2</td>
<td>0,0</td>
</tr>
<tr>
<td>a₃</td>
<td>k,k</td>
<td>0,0</td>
<td>10,10</td>
</tr>
</tbody>
</table>

#### Exploration:
- use $L_{RI} \rightarrow$ the agents converge to a pure (Nash) joint action

#### Synchronization:
- update average payoff for action $a$ converged to, optimistically
- exclude action $a$, and explore again if empty action set $\rightarrow$ RESET

If “done”: select BEST

---

**Exploration Phases**

- N
- 2N
- 3N

**Synchronization Phases**

- N

---

**With $k < 0$**
ESRL and common interest games

The Penalty Game

<table>
<thead>
<tr>
<th></th>
<th>b₁</th>
<th>b₂</th>
<th>b₃</th>
</tr>
</thead>
<tbody>
<tr>
<td>a₁</td>
<td>10,10</td>
<td>0,0</td>
<td>k,k</td>
</tr>
<tr>
<td>a₂</td>
<td>0,0</td>
<td>2,2</td>
<td>0,0</td>
</tr>
<tr>
<td>a₃</td>
<td>k,k</td>
<td>0,0</td>
<td>10,10</td>
</tr>
</tbody>
</table>

Exploration:
- use L_RI -> the agents converge to a pure (Nash) joint action

Synchronization:
- update average payoff for action \( a \) converged to, optimistically
- exclude action \( a \), and explore again if empty action set \( \rightarrow \) RESET

If “done”: select BEST
The Penalty Game

<table>
<thead>
<tr>
<th></th>
<th>a₁</th>
<th>a₂</th>
<th>a₃</th>
</tr>
</thead>
<tbody>
<tr>
<td>b₁</td>
<td>10,10</td>
<td>0,0</td>
<td>k,k</td>
</tr>
<tr>
<td>b₂</td>
<td>0,0</td>
<td>2,2</td>
<td>0,0</td>
</tr>
<tr>
<td>b₃</td>
<td>k,k</td>
<td>0,0</td>
<td>10,10</td>
</tr>
</tbody>
</table>

Exploration:
- use L_RI -> the agents converge to a pure (Nash) joint action

Synchronization:
- update average payoff for action a converged to, optimistically
- exclude action a, and explore again if empty action set -> RESET

If “done”: select BEST

ESRL and common interest games
The Penalty Game

Exploration:
- use L_RI -> the agents converge to a pure (Nash) joint action

Synchronization:
- update average payoff for action \( a \) converged to, optimistically
- exclude action \( a \), and explore again if empty action set \( \rightarrow \) RESET

If “done”: select BEST

ESRL and common interest games
ESRL and common interest games

The Penalty Game

<table>
<thead>
<tr>
<th></th>
<th>b₁</th>
<th>b₂</th>
<th>b₃</th>
</tr>
</thead>
<tbody>
<tr>
<td>a₁</td>
<td>10,10</td>
<td>0,0</td>
<td>k,k</td>
</tr>
<tr>
<td>a₂</td>
<td>0,0</td>
<td>2,2</td>
<td>0,0</td>
</tr>
<tr>
<td>a₃</td>
<td>k,k</td>
<td>0,0</td>
<td>10,10</td>
</tr>
</tbody>
</table>

Exploration:
- Use L_RI -> the agents converge to a pure (Nash) joint action

Synchronization:
- Update average payoff for action \( a \) converged to, optimistically
- Exclude action \( a \), and explore again if empty action set \( \rightarrow \) RESET

If “done”: select BEST

With \( k < 0 \)

Exploration Phases

Synchronization Phases
ESRL and common interest games

Exploration:
- use L_RI -> the agents converge to a pure (Nash) joint action

Synchronization:
- update average payoff for action \( a \) converged to, optimistically
- exclude action \( a \), and explore again if empty action set \( \rightarrow \) RESET

If “done”: select BEST

The Penalty Game

<table>
<thead>
<tr>
<th></th>
<th>( b_1 )</th>
<th>( b_2 )</th>
<th>( b_3 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( a_1 )</td>
<td>10,10</td>
<td>0,0</td>
<td>( k,k )</td>
</tr>
<tr>
<td>( a_2 )</td>
<td>0,0</td>
<td>2,2</td>
<td>0,0</td>
</tr>
<tr>
<td>( a_3 )</td>
<td>( k,k )</td>
<td>0,0</td>
<td>10,10</td>
</tr>
</tbody>
</table>

With \( k < 0 \)

Exploration Phases

<table>
<thead>
<tr>
<th>Time</th>
<th>Exploration Phases</th>
</tr>
</thead>
<tbody>
<tr>
<td>( N )</td>
<td></td>
</tr>
<tr>
<td>( 2N )</td>
<td></td>
</tr>
<tr>
<td>( 3N )</td>
<td></td>
</tr>
</tbody>
</table>
### The Penalty Game

<table>
<thead>
<tr>
<th></th>
<th>(a_1)</th>
<th>(a_2)</th>
<th>(a_3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(b_1)</td>
<td>10,10</td>
<td>0,0</td>
<td>(k,k)</td>
</tr>
<tr>
<td>(b_2)</td>
<td>0,0</td>
<td>2,2</td>
<td>0,0</td>
</tr>
<tr>
<td>(b_3)</td>
<td>(k,k)</td>
<td>0,0</td>
<td>10,10</td>
</tr>
</tbody>
</table>

**Exploration:**
- use L_RI -> the agents converge to a pure (Nash) joint action

**Synchronization:**
- update average payoff for action \(a\) converged to, optimistically
- exclude action \(a\), and explore again if empty action set → RESET

**If “done”: select BEST**

Note: in more than 2 agent games, at least 2 agents have to exclude an action in order to escape from an NE
ESRL and conflicting interest games

Exploration:
- use L_RI -> the agents converge to a (Nash) pure joint action

Synchronization:
- send and receive average payoff for joint action converged to (not the actions information)
- if best agent: excludes private action
- else RESET
Conflicting Interest games: periodical policies

Player 1

<table>
<thead>
<tr>
<th></th>
<th>B</th>
<th>S</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>2,1</td>
<td>0,0</td>
</tr>
<tr>
<td>S</td>
<td>0,0</td>
<td>1,2</td>
</tr>
</tbody>
</table>

Player 2
ESRL & Job Scheduling

\[ \mu_1 = \mu_2 = \mu_3 > \mu_C \]
ESRL & Job Scheduling
ESRL & Job Scheduling
More next week on graphical games

And interconnected automata
for solving multi-stage problems


