Dynamic Analysis of Multiagent Q-learning with $\varepsilon$-greedy Exploration

Eduardo R. Gomes
Ryszard Kowalczyk

Intelligent Agent Technology - CS3 - FICT
egomes@groupwise.swin.edu.au
Motivation

- Multiagent Learning (MAL) has become very active research area
- MAL-based systems are finding application in a wide variety of domains
- Tools to understand and model the expected dynamics are necessary
Motivation

> Multiagent Learning (MAL) has become very active research area
> MAL-based systems are finding application in a wide variety of domains
> Tools to understand and model the expected dynamics are necessary

Multiagent Q-learning with $\varepsilon$-greedy exploration
Motivation

- Multiagent Learning (MAL) has become very active research area
- MAL-based systems are finding application in a wide variety of domains
- Tools to understand and model the expected dynamics are necessary

Multiagent Q-learning with $\varepsilon$-greedy exploration

- Classic algorithm
- It has been applied with success in several domains
Motivation

Q-learning

- Most studied Reinforcement Learning algorithm
- Strong theoretical support and convergence guarantees
Motivation

Q-learning

> Most studied Reinforcement Learning algorithm
> Strong theoretical support and convergence guarantees
> ... only in the single-agent case
Motivation

Q-learning

> Most studied Reinforcement Learning algorithm
> Strong theoretical support and convergence guarantees
> ... only in the single-agent case
Motivation

**Q-learning**
- Most studied Reinforcement Learning algorithm
- Strong theoretical support and convergence guarantees
- ... only in the single-agent case

**Multiagent Q-learning**
- Lack of theoretical support and convergence guarantees
- Very dynamic environment
- Co-adaptation effect
- Rewards and state transitions depend on the joint actions
- Very hard to obtain the dynamics
Researchers have explored links between RL and EGT. The same principles apply: the growth in probability of one strategy's performance is directly proportional to its performance against the others. Model of Multiagent Q-learning with Boltzmann exploration cannot be applied because we have a semi-uniform distribution $\epsilon$-greedy mechanism. Selects the best action with probability $1-\epsilon$ and selects a random action with probability $\epsilon$. Dynamic Analysis of Multiagent Q-learning with $\epsilon$-greedy Exploration, ICML 2009 Eduardo R. Gomes
Researchers have explored links between RL and EGT

> Same principles
  - Growth in one strategy’s probability is directly proportional to its performance against the others

> Model of Multiagent Q-learning with Boltzmann exploration
RL and Evolutionary Game Theory

- Researchers have explored links between RL and EGT
- Same principles
  - Growth in one strategy’s probability is directly proportional to its performance against the others
- Model of Multiagent $Q$-learning with Boltzmann exploration
- Cannot be applied because we have a semi-uniform distribution
RL and Evolutionary Game Theory

- Researchers have explored links between RL and EGT
- Same principles
  - Growth in one strategy’s probability is directly proportional to its performance against the others
- Model of Multiagent Q-learning with Boltzmann exploration
- Cannot be applied because we have a semi-uniform distribution

$\epsilon$ – greedy mechanism
- Selects the best action with probability $1 - \epsilon$
- Selects a random action with probability $\epsilon$
Multiagent Q-learning

- Each agent applies the standard Q-learning algorithm
- The agents learn independently
- Rewards and state transitions depend on their joint strategies
Background

Multiagent Q-learning

- Each agent applies the standard Q-learning algorithm
- The agents learn independently
- Rewards and state transitions depend on their joint strategies

- Each agent maintains a table of Q-values
  - $Q(s, i)$ represents how good it is to take action $i$ at state $s$

- They update the Q-values as they gather experience in the environment
  $$Q(s, i) = Q(s, i) + \alpha (r(s, i) + \gamma \max_{i'} Q(s', i') - Q(s, i))$$
  - $r(s, i)$ is the reward for taking action $i$ at state $s$
  - $\alpha$ is the learning rate
  - $\gamma$ is the discount rate
Action-selection mechanism

Exploration - exploitation problem

- exploit actions known to be good
- explore new actions

$\epsilon$-greedy

- chose the currently best action with probability $1 - \epsilon$
- chose a random action with probability $\epsilon$
Action-selection mechanism

Exploration - exploitation problem

> exploit actions known to be good
> explore new actions

$\epsilon$-greedy

> chose the currently best action with probability $1 - \epsilon$
> chose a random action with probability $\epsilon$

$$x(s,i) = \begin{cases} (1 - \epsilon) + (\epsilon/n), & \text{if } Q(s,i) \text{ is currently the highest} \\ \epsilon/n, & \text{otherwise} \end{cases}$$
Modelling the algorithm

- Build a continuous-time version of the Q-learning update rule
- Analyse the limits of this equation for the single-learner case
- Show how they change dynamically in the multi-learner case
- Investigate how the $\varepsilon$-greedy affects the shape of the function
- Develop a system of difference equations to obtain the expected behaviour of the agents
Modelling the algorithm

> Build a continuous-time version of the Q-learning update rule
Modelling the algorithm

> Build a continuous-time version of the Q-learning update rule
> Analyse the limits of this equation for the single-learner case
Modelling the algorithm

> Build a continuous-time version of the Q-learning update rule
> Analyse the limits of this equation for the single-learner case
> Show how they change dynamically in the multi-learner case
Modelling the algorithm

> Build a continuous-time version of the Q-learning update rule
> Analyse the limits of this equation for the single-learner case
> Show how they change dynamically in the multi-learner case
> Investigate how the $\varepsilon$-greedy affects the shape of the function
Modelling the algorithm

> Build a continuous-time version of the Q-learning update rule
> Analyse the limits of this equation for the single-learner case
> Show how they change dynamically in the multi-learner case
> Investigate how the $\varepsilon$-greedy affects the shape of the function
> Develop a system of difference equations to obtain the expected behaviour of the agents
Notation

Single-state scenarios composed of 2 agents with 2 actions each.

The reward functions can be described as payoff tables:

\[ A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \]

\[ B = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} \]

The Q-learning rule can be simplified to:

\[ Q_{a_i} \leftarrow Q_{a_i} + \alpha (r_{a_i} - Q_{a_i}) \]

where:

- \( Q_{a_i} \) is the Q-value of agent \( a \) for action \( i \).
- \( r_{a_i} \) is the immediate reward that agent \( a \) receives for playing action \( i \).
Notation

Single-state scenarios composed of 2 agents with 2 actions each
Notation

Single-state scenarios composed of 2 agents with 2 actions each

The reward functions can be described as payoff tables

\[
A = \begin{bmatrix}
a_{11} & a_{12} \\
a_{21} & a_{22}
\end{bmatrix} \quad B = \begin{bmatrix}
b_{11} & b_{12} \\
b_{21} & b_{22}
\end{bmatrix}
\]
Notation

Single-state scenarios composed of 2 agents with 2 actions each

The reward functions can be described as payoff tables

\[
A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \quad B = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}
\]

Q-learning rule can be simplified to

\[
Q_{a_i} \leftarrow Q_{a_i} + \alpha (r_{a_i} - Q_{a_i})
\]

\(Q_{a_i}\) is the Q-value of agent \(a\) for action \(i\)

\(r_{a_i}\) is the immediate reward that agent \(a\) receives for playing action \(i\)
Continuous-time version

\[ Q_{ai} \leftarrow Q_{ai} + \alpha (r_{ai} - Q_{ai}) \]

Q-learning rule
Continuous-time version

\[ Q_{a_i} \leftarrow Q_{a_i} + \alpha (r_{a_i} - Q_{a_i}) \]  

Q-learning rule

\[ Q_{a_i}(k + 1) = Q_{a_i}(k) + \alpha (r_{a_i}(k + 1) - Q_{a_i}(k)) \]
Continuous-time version

\[ Q_{a_i} \leftarrow Q_{a_i} + \alpha (r_{a_i} - Q_{a_i}) \]

Q-learning rule

\[ Q_{a_i}(k + 1) = Q_{a_i}(k) + \alpha (r_{a_i}(k + 1) - Q_{a_i}(k)) \]

\[ Q_{a_i}(k + 1) - Q_{a_i}(k) = \alpha (r_{a_i}(k + 1) - Q_{a_i}(k)) \]

discrete
Continuous-time version

\[ Q_{a_i} \leftarrow Q_{a_i} + \alpha (r_{a_i} - Q_{a_i}) \]  
Q-learning rule

\[ Q_{a_i}(k + 1) = Q_{a_i}(k) + \alpha (r_{a_i}(k + 1) - Q_{a_i}(k)) \]

discrete

\[ Q_{a_i}(k + 1) - Q_{a_i}(k) = \alpha (r_{a_i}(k + 1) - Q_{a_i}(k)) \]

\[ Q_{a_i}(k + \Delta t) - Q_{a_i}(k) \approx \Delta t \times \alpha (r_{a_i}(k + \Delta t) - Q_{a_i}(k)) \]
Continuous-time version

\[ Q_{a_i} \leftarrow Q_{a_i} + \alpha (r_{a_i} - Q_{a_i}) \]  
\[ Q_{a_i}(k + 1) = Q_{a_i}(k) + \alpha (r_{a_i}(k + 1) - Q_{a_i}(k)) \]  
\[ Q_{a_i}(k + 1) - Q_{a_i}(k) = \alpha (r_{a_i}(k + 1) - Q_{a_i}(k)) \]  
\[ Q_{a_i}(k + \Delta t) - Q_{a_i}(k) \approx \Delta t \times \alpha (r_{a_i}(k + \Delta t) - Q_{a_i}(k)) \]  
\[ \lim_{\Delta t \to 0} \frac{Q_{a_i}(k + \Delta t) - Q_{a_i}(k)}{\Delta t} \approx \alpha (r_{a_i}(k) - Q_{a_i}(k)) \]
Continuous-time version

\[ Q_{a_i} \leftarrow Q_{a_i} + \alpha (r_{a_i} - Q_{a_i}) \] 

Q-learning rule

\[ Q_{a_i}(k + 1) = Q_{a_i}(k) + \alpha (r_{a_i}(k + 1) - Q_{a_i}(k)) \]

\[ Q_{a_i}(k + 1) - Q_{a_i}(k) = \alpha (r_{a_i}(k + 1) - Q_{a_i}(k)) \] 

discrete

\[ Q_{a_i}(k + \Delta t) - Q_{a_i}(k) \approx \Delta t \times \alpha (r_{a_i}(k + \Delta t) - Q_{a_i}(k)) \]

\[ \lim_{\Delta t \to 0} \frac{Q_{a_i}(k+\Delta t) - Q_{a_i}(k)}{\Delta t} \approx \alpha (r_{a_i}(k) - Q_{a_i}(k)) \]

\[ \frac{dQ_{a_i}(k)}{dt} \approx \alpha (r_{a_i}(k) - Q_{a_i}(k)) \] 

continuous
Limit of the equation

\[
\frac{dQ_{a_i}(k)}{dt} \approx \alpha (r_{a_i}(k) - Q_{a_i}(k)) \quad \text{continuous}
\]
Limit of the equation

\[ \frac{dQ_{a_i}(k)}{dt} \approx \alpha (r_{a_i}(k) - Q_{a_i}(k)) \quad \text{continuous} \]

\[ Q_{a_i}(k) = Ce^{-\alpha t} + r_{a_i} \quad \text{general solution} \]
Limit of the equation

\[ \frac{dQ_a(k)}{dt} \approx \alpha (r_a(k) - Q_a(k)) \]  
continuous

\[ Q_a(k) = Ce^{-\alpha t} + r_a \]  
general solution

\[ \lim_{t \to \infty} Q_a(k) = \lim_{t \to \infty} Ce^{-\alpha t} + \lim_{t \to \infty} r_a = r_a \]
Non-learning adversary with pure strategy

\[ Q_{a_i} \text{ will monotonically increase or decrease towards } r_{a_i} \]
Non-learning adversary with pure strategy

\( Q_{a_i} \) will monotonically increase or decrease towards \( r_{a_i} \)

\[ \alpha = 0.2 \text{ and } r_{a_i} = 5; \quad Q_{a_i}(0) \in \{0, 2, 8, 10\} \]
Non-learning adversary with mixed strategy.
Non-learning adversary with mixed strategy

\( r_{ai} \) can be replaced by \( E[r_{ai}] = \sum_j a_{ij} y_j \)

<table>
<thead>
<tr>
<th></th>
<th>0.8</th>
<th>0.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>5</td>
</tr>
<tr>
<td>0</td>
<td></td>
<td>3</td>
</tr>
</tbody>
</table>

\[
E[r_{a1}] = (0.8 \times 1) + (0.2 \times 5) = 1.8 \\
E[r_{a2}] = (0.8 \times 0) + (0.2 \times 3) = 0.6
\]

\[
\frac{dQ_{ai}(t)}{dt} \approx \alpha(E[r_{ai}(t)] - Q_{ai}(t))
\]
Non-learning adversary with mixed strategy

$r_{a_i}$ can be replaced by $E[r_{a_i}] = \sum_j a_{ij} y_j$

\[ E[r_{a_1}] = (0.8 \times 1) + (0.2 \times 5) = 1.8 \]
\[ E[r_{a_2}] = (0.8 \times 0) + (0.2 \times 3) = 0.6 \]

\[ \frac{dQ_{a_i}(t)}{dt} \approx \alpha (E[r_{a_i}(t)] - Q_{a_i}(t)) \]

\[ Q_{a_i}(t) = Ce^{-\alpha t} + E[r_{a_i}] \]

\[ \lim_{t \to \infty} Q_{a_i}(k) = \lim_{t \to \infty} Ce^{-\alpha t} + \lim_{t \to \infty} E[r_{a_i}] = E[r_{a_i}] \]

\[ \text{0} \quad \text{E}[r_{a_i}] \]
Non-learning adversary with mixed strategy

\[ r_{ai} \text{ can be replaced by } E[r_{ai}] = \sum_j a_{ij}y_j \]

\[
\begin{array}{|c|c|}
\hline
0.8 & 0.2 \\
1 & 5 \\
0 & 3 \\
\hline
\end{array}
\]

\[ E[r_{a1}] = (0.8 \times 1) + (0.2 \times 5) = 1.8 \]
\[ E[r_{a2}] = (0.8 \times 0) + (0.2 \times 3) = 0.6 \]

\[
\frac{dQ_{ai}(t)}{dt} \approx \alpha (E[r_{ai}(t)] - Q_{ai}(t))
\]

\[ Q_{ai}(t) = Ce^{-\alpha t} + E[r_{ai}] \]

\[ \lim_{t \to \infty} Q_{ai}(k) = \underbrace{\lim_{t \to \infty} Ce^{-\alpha t}}_{0} + \underbrace{\lim_{t \to \infty} E[r_{ai}]}_{E[r_{ai}]} = E[r_{ai}] \]

then \( Q_{ai} \) will move in expectation towards \( E[r_{ai}] \) in a monotonic fashion.
Learning adversary

Adversary can change its strategy during the learning, changing the expected rewards.

\[ E[r_{a1}] = (0.8 \times 1) + (0.2 \times 5) = 1.0 \]

\[ E[r_{a2}] = (0.2 \times 1) + (0.8 \times 5) = 4.2 \]

Each time the expected reward changes, it changes the limits and direction fields.
Learning adversary

Adversary can change its strategy during the learning changing the expected rewards
Learning adversary

Adversary can change its strategy during the learning changing the expected rewards

<table>
<thead>
<tr>
<th>0.8</th>
<th>0.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>0</td>
<td>3</td>
</tr>
</tbody>
</table>

\[ E[r_{a_1}] = (0.8 \times 1) + (0.2 \times 5) = 1.8 \]
Learning adversary

Adversary can change its strategy during the learning changing the expected rewards

\[
\begin{array}{cc}
0.8 & 0.2 \\
1 & 5 \\
0 & 3 \\
\end{array}
\]

\[
E[r_{a_1}] = (0.8 \times 1) + (0.2 \times 5) = 1.8
\]

\[
\begin{array}{cc}
0.2 & 0.8 \\
1 & 5 \\
0 & 3 \\
\end{array}
\]

\[
E[r_{a_1}] = (0.2 \times 1) + (0.8 \times 5) = 4.2
\]
Learning adversary

Adversary can change its strategy during the learning changing the expected rewards

\[
E[r_{a_1}] = (0.8 \times 1) + (0.2 \times 5) = 1.8
\]

<table>
<thead>
<tr>
<th>0.8</th>
<th>0.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>0</td>
<td>3</td>
</tr>
</tbody>
</table>

\[
E[r_{a_1}] = (0.2 \times 1) + (0.8 \times 5) = 4.2
\]

<table>
<thead>
<tr>
<th>0.2</th>
<th>0.8</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>0</td>
<td>3</td>
</tr>
</tbody>
</table>

Each time the expected reward changes, it changes the limits and direction fields
Learning adversary

Important to identify when the changes in the adversary’s strategy will occur
Learning adversary

Important to identify when the changes in the adversary’s strategy will occur

$\epsilon$-greedy updates the strategy whenever a new action becomes the one with highest $Q$-value

Need to find the intersection points in the adversary’s functions
Learning adversary

Important to identify when the changes in the adversary’s strategy will occur

\(\varepsilon\)-greedy updates the strategy whenever a new action becomes the one with highest \(Q\)-value

Need to find the intersection points in the adversary’s functions
The effects of the $\varepsilon$-greedy actions have different probabilities ($x_i$) of being played, e.g. if $\varepsilon = 0.2 \rightarrow x = [0.9, 0.1]$ or $x = [0.1, 0.9]$ they are updated at different speeds:

$$\frac{dQ_{ai}(t)}{dt} \approx x_i(t) \alpha \left( E[r_{ai}(t)] - Q_{ai}(t) \right)$$

$$Q_{ai}(t) = C e^{-x_i \alpha t} + E[r_{ai}]$$
The effects of the $\varepsilon$-greedy

Actions have different probabilities ($x_i$) of being played

e.g. if $\varepsilon = 0.2 \rightarrow x = [0.9, 0.1]$ or $x = [0.1, 0.9]

they are updated at different *speeds*
The effects of the $\varepsilon$-greedy

Actions have different probabilities ($x_i$) of being played

\[
e.g. \text{ if } \varepsilon = 0.2 \quad \rightarrow \quad x = [0.9, 0.1] \text{ or } x = [0.1, 0.9]
\]

they are updated at different speeds

\[
\frac{dQ_{a_i}(t)}{dt} \approx x_i(t) \alpha (E[r_{a_i}(t)] - Q_{a_i}(t))
\]
The effects of the $\varepsilon$-greedy

Actions have different probabilities ($x_i$) of being played

e.g. if $\varepsilon = 0.2 \rightarrow x = [0.9, 0.1]$ or $x = [0.1, 0.9]$

they are updated at different speeds

$$\frac{dQ_{ai}(t)}{dt} \approx x_i(t)\alpha(E[r_{ai}(t)] - Q_{ai}(t))$$

$$Q_{ai}(t) = Ce^{-x_i\alpha t} + E[r_{ai}]$$
The effects of the $\varepsilon$-greedy

It does not change the limits of the equation

$$\lim_{t \to \infty} Q_{a_i}(t) = \lim_{t \to \infty} Ce^{-x_i \alpha t} + \lim_{t \to \infty} E[r_{a_i}] = E[r_{a_i}]$$
The effects of the $\epsilon$-greedy

It does not change the limits of the equation

$$\lim_{t \to \infty} Q_{a_i}(t) = \lim_{t \to \infty} Ce^{-x_i\alpha t} + \lim_{t \to \infty} E[r_{a_i}] = E[r_{a_i}]$$

But changes the shape of the function and associated direction field
Summary of the analysis (roughly speaking)
Summary of the analysis (roughly speaking)

**Expected Rewards**

are the values to which the $Q$-values will converge to
Summary of the analysis (roughly speaking)

<table>
<thead>
<tr>
<th><strong>Expected Rewards</strong></th>
<th>are the values to which the (Q)-values will converge to</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Speeds</strong></td>
<td>determine the paths that the (Q)-values will follow to get there</td>
</tr>
</tbody>
</table>
Summary of the analysis (roughly speaking)

<table>
<thead>
<tr>
<th>Expected Rewards</th>
<th>are the values to which the $Q$-values will converge to</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Speeds</strong></td>
<td>determine the paths that the $Q$-values will follow to get there</td>
</tr>
<tr>
<td><strong>Intersection points</strong></td>
<td>define if the $Q$-values will ever get there</td>
</tr>
</tbody>
</table>
System of difference equations

\[ Q_{a_i}(t+1) = Q_{a_i}(t) + x_i(t) \alpha \left( \sum_j a_{ij} y_j(t) - Q_{a_i}(t) \right) \]

\[ Q_{b_i}(t+1) = Q_{b_i}(t) + y_i(t) \alpha \left( \sum_j b_{ij} x_j(t) - Q_{b_i}(t) \right) \]

\[ x_i(t) = \begin{cases} 
(1 - \varepsilon) + \left( \frac{\varepsilon}{n} \right), & \text{if } Q_{a_i}(t) \text{ is currently the highest} \\
\varepsilon/n, & \text{otherwise} 
\end{cases} \]

\[ y_i(t) = \begin{cases} 
(1 - \varepsilon) + \left( \frac{\varepsilon}{n} \right), & \text{if } Q_{b_i}(t) \text{ is currently the highest} \\
\varepsilon/n, & \text{otherwise} 
\end{cases} \]
System of difference equations

<table>
<thead>
<tr>
<th>A and B</th>
<th>X and Y</th>
<th>$Q_a$ and $Q_b$</th>
</tr>
</thead>
<tbody>
<tr>
<td>payoff tables</td>
<td>strategy vectors</td>
<td>Q-values vectors</td>
</tr>
</tbody>
</table>

$Q_a^i(t+1) = Q_a^i(t) + x^i(t) \alpha \left( \sum_j a_{ij} y^j(t) - Q_a^i(t) \right)$

$Q_b^i(t+1) = Q_b^i(t) + y^i(t) \alpha \left( \sum_j b_{ij} x^j(t) - Q_b^i(t) \right)$

$x^i(t) = \begin{cases} (1 - \varepsilon) + \left( \frac{\varepsilon}{n} \right), & \text{if } Q_a^i(t) \text{ is currently the highest} \\ \frac{\varepsilon}{n}, & \text{otherwise} \end{cases}$

$y^i(t) = \begin{cases} (1 - \varepsilon) + \left( \frac{\varepsilon}{n} \right), & \text{if } Q_b^i(t) \text{ is currently the highest} \\ \frac{\varepsilon}{n}, & \text{otherwise} \end{cases}$
System of difference equations

<table>
<thead>
<tr>
<th>A and B</th>
<th>X and Y</th>
<th>Qa and Qb</th>
</tr>
</thead>
<tbody>
<tr>
<td>payoff tables</td>
<td>strategy vectors</td>
<td>Q-values vectors</td>
</tr>
</tbody>
</table>

\[ Q_{a_i}(t+1) = Q_{a_i}(t) + x_i(t)\alpha(\sum_j a_{ij}y_j(t) - Q_{a_i}(t)) \]

\[ Q_{b_i}(t+1) = Q_{b_i}(t) + y_i(t)\alpha(\sum_j b_{ij}x_j(t) - Q_{b_i}(t)) \]

\[ x_i(t) = \begin{cases} (1 - \varepsilon) + (\varepsilon/n), & \text{if } Q_{a_i}(t) \text{ is currently the highest} \\ \varepsilon/n, & \text{otherwise} \end{cases} \]

\[ y_i(t) = \begin{cases} (1 - \varepsilon) + (\varepsilon/n), & \text{if } Q_{b_i}(t) \text{ is currently the highest} \\ \varepsilon/n, & \text{otherwise} \end{cases} \]
Prisoner’s Dilemma

\[
A = \begin{bmatrix} 1 & 5 \\ 0 & 3 \end{bmatrix} \quad B = \begin{bmatrix} 1 & 0 \\ 5 & 3 \end{bmatrix}
\]
Prisoner’s Dilemma

\[ A = \begin{bmatrix} 1 & 5 \\ 0 & 3 \end{bmatrix}, \quad B = \begin{bmatrix} 1 & 0 \\ 5 & 3 \end{bmatrix} \]

\( Q_a = [0, 1], \ Q_b = [1, 0], \ \alpha = 0.1, \ \epsilon = 0.4 \)

\( X = [0.2, 0.8], \ Y = [0.8, 0.2]. \)
Prisoner's Dilemma

\[ A = \begin{bmatrix} 1 & 5 \\ 0 & 3 \end{bmatrix} \quad B = \begin{bmatrix} 1 & 0 \\ 5 & 3 \end{bmatrix} \]

\[ Q_a = [0, 1], \quad Q_b = [1, 0], \quad \alpha = 0.1, \quad \epsilon = 0.4 \]

\[ X = [0.2, 0.8], \quad Y = [0.8, 0.2]. \]
Prisoner’s Dilemma

Dynamic Analysis of Multiagent Q-learning with $\varepsilon$-greedy Exploration, ICML 2009

Eduardo R. Gomes
Battle of the Sexes

\[
A = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix} \quad B = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}
\]
Battle of the Sexes

\[ A = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix} \quad B = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix} \]

\[ Q_a = [2, 1], \quad Q_b = [2, 4], \quad \alpha = 0.1, \quad \varepsilon = 0.1 \]
\[ X = [0.95, 0.05], \quad Y = [0.05, 0.95]. \]
Battle of the Sexes

\[ A = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix} \]

\[ Q_a = [2, 1], \quad Q_b = [2, 4], \quad \alpha = 0.1, \quad \varepsilon = 0.1 \]
\[ X = [0.95, 0.05], \quad Y = [0.05, 0.95]. \]
Battle of the Sexes

Dynamic Analysis of Multiagent Q-learning with $\epsilon$-greedy Exploration, ICML 2009

Eduardo R. Gomes
A game with no equilibrium

\[ A = \begin{bmatrix} 2 & 3 \\ 4 & 1 \end{bmatrix} \quad B = \begin{bmatrix} 3 & 1 \\ 2 & 4 \end{bmatrix} \]
A game with no equilibrium

\[ A = \begin{bmatrix} 2 & 3 \\ 4 & 1 \end{bmatrix} \quad B = \begin{bmatrix} 3 & 1 \\ 2 & 4 \end{bmatrix} \]

\[ Q_a = [0, 1], \ Q_b = [2, 3], \ \alpha = 0.1, \ \varepsilon = 0.1 \]
\[ X = [0.05, 0.95], \ Y = [0.05, 0.95]. \]
A game with no equilibrium

\[ A = \begin{bmatrix} 2 & 3 \\ 4 & 1 \end{bmatrix} \quad B = \begin{bmatrix} 3 & 1 \\ 2 & 4 \end{bmatrix} \]

\[ Q_a = [0, 1], \quad Q_b = [2, 3], \quad \alpha = 0.1, \quad \varepsilon = 0.1 \]

\[ X = [0.05, 0.95], \quad Y = [0.05, 0.95]. \]
A game with no equilibrium
Conclusions

- Presented a model for the dynamics of Multiagent Q-learning with $\varepsilon$-greedy exploration
  - Studied a continuous-time version of the Q-learning update rule
  - Investigated how the presence of other agents and the $\varepsilon$-greedy mechanism affect it
Conclusions

> Presented a model for the dynamics of Multiagent Q-learning with $\varepsilon$-greedy exploration
  - Studied a continuous-time version of the $Q$-learning update rule
  - Investigated how the presence of other agents and the $\varepsilon$-greedy mechanism affect it

> Defined a system of difference equations
  - Model the expected evolution of the $Q$-values
  - Derive the expected behaviour from the $Q$-values
Conclusions

> Presented a model for the dynamics of Multiagent Q-learning with $\varepsilon$-greedy exploration
  > Studied a continuous-time version of the $Q$-learning update rule
  > Investigated how the presence of other agents and the $\varepsilon$-greedy mechanism affect it

> Defined a system of difference equations
  > Model the expected evolution of the $Q$-values
  > Derive the expected behaviour from the $Q$-values

> The evaluation of the model in typical games has shown its feasibility
Future Works

> Extend the model to multi-state scenarios
> Develop techniques for the visualization of the agents’ behaviour
Dynamic Analysis of Multiagent Q-learning with $\varepsilon$-greedy Exploration

Eduardo R. Gomes
Ryszard Kowalczyk

Intelligent Agent Technology - CS3 - FICT
eyomes@groupwise.swin.edu.au