Partial Local FriendQ Multiagent Learning: Application to Team Automobile Coordination

Julien Laumônier, Brahim Chaib-draa
Université Laval, Québec

June 8, 2006
Motivation

- Multiagent real world problems are too complex to be solved optimally.
- We need approximation to learn the joint policy.
- In such problems, each agent has often a partial view of the environment.
- We propose to use this partial view to approximate the problem.
- Possible to control the degree of observability using communication.
- Find a compromise between increasing communication and partial view to calculate the multiagent policy.
Introduction

More specifically, we focus on teams where each agent has the same reward.

In this context, we base our work on optimal algorithms (Friend Q-learning).

We use the context of DEC-MDP where:

1. Agents have mutually exclusive observations,
2. Communication is implicit to construct the state of the environment,
3. Agents produce only negative interactions.
Introduction

- What is the performance of a multiagent reinforcement learning algorithm where agents have partial vision?
- How to compare the performance with the optimal joint policy?
- Which problem’s characteristics do influence performances using reinforcement learning with partial vision?
Plan

1. Introduction
2. Formal models
3. Problem Description
4. Partial View Algorithms
5. Results
6. Conclusion
A DEC-MDP is a tuple $\langle Ag, S, A^i, P, R, \Omega, O \rangle$ where
- $Ag$ is the set of agents where $\text{card}(Ag) = N$,
- $A$ the finite set of actions. $A^N$ is the joint action space,
- $S$ is a finite set of states. $S^N$ represents the set of joint states,
- $R$ is the immediate reward function,
- $P$ is the transition probability,
- $\Omega^i$ is the set of observation for agent $i$,
- $O$ is the observation probability table.

Friend Q-Learning

Friend Q-Learning built an optimal policy for fully observable problems.
Plan

1. Introduction
2. Formal models
3. **Problem Description**
4. Partial View Algorithms
5. Results
6. Conclusion
Problem

- Represents a vehicle coordination problem.
- Each vehicle is represented by a position and a velocity,
- At each step, a vehicle can stay in its lane, change lane to the right or to the left,
- Common Reward: average velocity over vehicles,
- Environment’s dynamic: \( x(t) = v \times t + x_0 \).
## Plan

1. Introduction
2. Formal models
3. Problem Description
4. **Partial View Algorithms**
5. Results
6. Conclusion
Partial View

Road direction

Ag1
Ag2
Ag3

Come Back
Partial View

Road direction

Agent 1 partial view
Partial View

Road direction

Agent 1 partial view

Agent 2 partial view
Partial View

Agent 1 partial view

Agent 2 partial view

Agent 3 partial view
Local view with distance $d$

- Let $d$ the visibility distance around each agent.
- The main goal is to reduce the number of state and/or the number of joint actions.

<table>
<thead>
<tr>
<th>Actions</th>
<th>Reduction</th>
<th>States</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reduction</td>
<td>PJA Q-learning</td>
<td>No reduction</td>
</tr>
<tr>
<td>No reduction</td>
<td>TJA Q-learning</td>
<td>Friend Q-learning</td>
</tr>
</tbody>
</table>
Algorithms TJA et PJA

**TJA Q-learning**

Update function:

\[
Q(f^i_d(s), a) = (1 - \alpha)Q(f^i_d(s), a) + \alpha[r + \gamma \max_{a' \in A} Q(f^i_d(s'), a)]
\]

**PJA Q-learning**

Update function:

\[
Q(s^i_d, a^i_d) = (1 - \alpha)Q(s^i_d, a^i_d) + \alpha[r + \gamma \max_{a' \in G^i_d(A, S)} Q(s'^i_d, a')] \\
\text{with } s^i_d = f^i_d(s) \text{ and } a^i_d = g^i_d(a, s)
\]
Set of states

State Space Reduction

- Total view: $O((X \times Y \times |V|)^N)$
- $O(((2d + 1)^2 \times V)^N)$.
- The number of states is divided by $(Y/(2d + 1))^N$
Plan

1. Introduction
2. Formal models
3. Problem Description
4. Partial View Algorithms
5. Results
6. Conclusion
We test on different problems $P_1 \ldots P_4$ which have the following parameters:

1. $P_1$: Size $3 \times 7$; $N_{b\text{Agents}} = 3$
2. $P_2$: Size $5 \times 20$; $N_{b\text{Agents}} = 3$
3. $P_3$: Size $5 \times 20$; $N_{b\text{Agents}} = 5$
4. $P_4$: Size $6 \times 28$; $N_{b\text{Agents}} = 4$
Comparing TJA and PJA

- Convergence to local maximum for $d = 0 \ldots 2$ for both algorithms TJA and PJA.
- The more $d$ increases, the more the policy is efficient.
- It is possible for each $d$ to measure the optimal approximation.
PJA Q-learning on problem $P_2$
Let $d_{app}$, a distance for each problem which give a policy value close to optimal value.

- For problem $P_1$, $d_{app} = 3$
- For problem $P_2$, $d_{app} = 4$

Drawback: necessary to calculate the optimal policy to know $d_{app}$ in each problem.
Generalization

- To generalize problems, we calculate the Degree of Space for each agent in each problem.

\[ DS = \frac{XY}{N} \]

- We compare PJA algorithm on different problems with different \( DS : DS = \{7, 20, 33, 40\} \).

- We measure \( d_{app} \) such that \( 1 - \left( \frac{R_{d_{app}}}{R_{friendQ}} \right) < \epsilon \) with \( \epsilon = 0.01 \).
Degree of Observability and Degree of Space

- Let $\frac{d_{app}}{d_{total}}$, the degree of observability.
- We measure the ratio between observability and space for 4 problems then we interpolate the curve.
Conclusion

- Introduction of degree of observability using implicit communication.
- It is possible to obtain a good approximation of the optimal policy with Degree of Observability less than 100%.
- We obtained an evaluation of the degree of observability required for a good approximation according to some characteristics of the problem.
- Drawback : Linked to the studied problem (only negative interactions) and only empirical results.
- Future Works :
  - Theoretical Study of the degree of observability
  - Explicit communication.
  - Extend to non cooperative cases and to positive/negative interactions cases.
Questions

Thank you for your attention