2011 International Symposium on Nonlinear Theory and its Applications
NOLTA2011, Kobe, Japan, September 4-7, 2011

A Multi-agent Reinforcement Learning Method for Acquiring the Sociality

Yasuo Nagayuki

Department of Social and Management Studies, Otemae University
2-2-2 Inano, Itami, Hyogo 664-0861 Japan
Email: nagayuki@otemae.ac.jp

Abstract– In multi-agent environments, it is important that the agents have the "sociality". In this article, I propose a reinforcement learning framework, which is based on Q-learning, that the agent is able to learn the "sociality" in a multi-agent environment. In this framework, the agent learns to ignore the near goal, which is left for the other agent, and go toward the farther goal, if the agent judges that the decision is effective from the social viewpoint, but not from the agent's greedy viewpoint.

1. Introduction

It is hard that the agent, which learns by using a reinforcement learning [1] algorithm, learns to go toward the farther goal rather than the near goal. If the values of the reward obtained by reaching each goal are not different, it is all the more.

In the case that the environment the agent faces is a single-agent environment, the agent should not ignore the near goal because the agent should be greedy for the environment, that is, the near goal. If the agent learns to go toward the farther goal rather than the near goal in this case, the reinforcement learning algorithm must have some defect.

In the case that the environment the agent faces is a multi-agent environment [2], on the other hand, there are the situations that the agent should ignore the near goal, which is left for the other agent, and go toward the farther goal. If the agent's decision is not effective from the agent's greedy (selfish) viewpoint, but effective from the social (global) viewpoint, it is called that the agent has the "sociality". In multi-agent environments, it is important that the agents have the "sociality".

However, it is difficult that the agent applied the reinforcement learning algorithms proposed so far learn the "sociality" in multi-agent environments, because the reinforcement learning algorithms are designed that the agent learns greedy policies for the environment.

In this article, I propose a reinforcement learning framework, which is based on Q-learning [3], that the agents learn the "sociality" in a multi-agent environment. In this framework, the agents learn to ignore the near goal and go toward the farther goal, if the agent judges that the decision is effective from the social viewpoint.

2. Learning Task

In this study, I prepared a two-agents problem as a task. This task is designed that one agent should learn to ignore the near goal and go toward the farther goal from the social viewpoint, and the other agent should learn to go toward the near goal from the social and greedy viewpoint.

- In a 4×7 grid space, two agents exist and there are two goal positions as shown in Fig.1.
- At each episode, the initial positions of the agents are the positions shown in Fig.1.
- At a discrete time step in an episode, the agents synchronously execute one out of five actions: moving up, down, left, or right from the current position, or staying in the current position. But, the agents are not able to execute the actions that the agents go out of the grid space. For example, in Fig.1, the agent A1 is not able to execute the two actions: moving down and right.
- The agents are able to look over the whole grid space.
- When an agent reaches a goal position (G1 or G2), the agent obtains a positive reward. The value of the reward is 1.0 regardless of G1 or G2.
- The reward of each goal position disappears when an agent arrives at the goal position. Therefore, in each goal position, the reward is given for only one agent.
- The global goal as this task is that each of two agents reaches either of the two goal positions. But, the goal positions two agents reach must be different. That is, there are two goal situations in this task as shown in Fig.2 (a) or Fig.2 (b). When a goal situation is achieved, the episode ends.

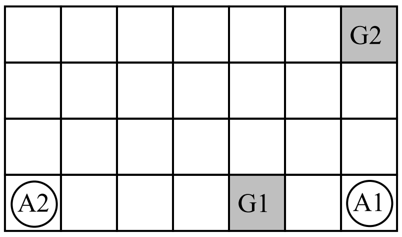
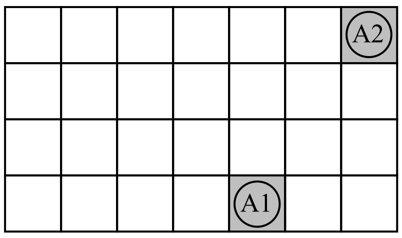
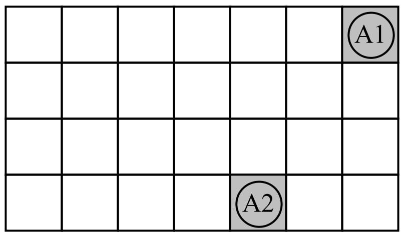


Fig.1. Grid space of learning task: A1 and A2 are the agents, and G1 and G2 are the goal positions for the agents.

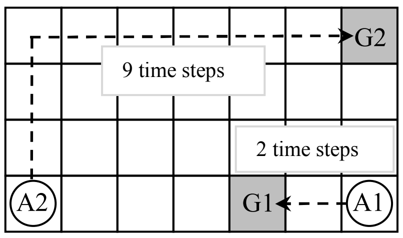


(a) Goal situation-1: the agent A1 reaches the goal position G1 and the agent A2 reaches the goal position G2.

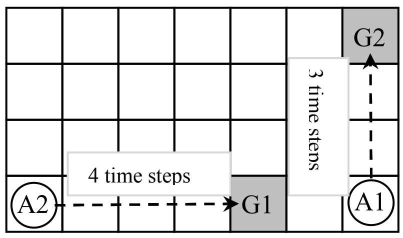


(b) Goal situation-2: the agent A1 reaches the goal position G2 and the agent A2 reaches the goal position G1.

Fig.2. Goal situations of this learning task



(a) An example of shortest passes for the agents until achieving the goal situation-1.



(b) An example of shortest passes for the agents until achieving the goal situation-2.

Fig.3. Shortest passes until achieving each goal situation for each agent from the initial positions.

Here, I consider which goal situations are more effective if each agent goes toward its goal position from its initial position with the shortest pass in this task. In the case of the goal situation-1 shown in Fig.2 (a), the agent A1 is able to reach the goal position G1 with 2 time steps, and the agent A2 is able to reach the goal position G2 with 9 time steps as shown in Fig.3 (a). Therefore, the number of the shortest time step that the goal situation-1 is achieved is 9. It is remarked that the agent A1 is waited until the agent A2 reaches the goal position G2. In the case of the goal situation-2 shown in Fig.2 (b), on the other hand, the agent A1 is able to reach the goal position G2 with 3 time steps, and the agent A2 is able to reach the goal position G1 with 4 time steps as shown in Fig.3 (b). Therefore, the number of the shortest time step that the goal situation-2 is achieved is 4.

As above-mentioned, if each agent goes toward its goal position with shortest pass, the goal situation-2 is more effective than the goal situation-1 because the number of the time step until achieving the goal situation is shorter. However, in order to achieve the goal situation-2, the agent A1 must give up going toward the near goal position G1, which is left for the agent A2.

3. MDP and Q-learning

The reinforcement learning framework proposed in this article is based on Q-learning [3]. Q-learning was originally defined to deal with Markov decision processes (MDPs) [4].

3.1. MDP

An MDP is defined by a set of finite states of the environment, S, and a set of finite actions, A. At each time step, an agent observes a state s ($\in S$), executes an action a ($\in A$) and receives a reward r from the environment. The state transition of the environment is defined by a transition probability function:

$$P_s{}^a{}_{s'} = \Pr(s' | s, a) \qquad (1),$$

which is the probability that the environment changes to a new state s' ($\in S$) when the agent executes the action a in the state s. The reward r is a variable, possibly probabilistic, which depends only upon the current state s and action a.

3.2. Q-learning

Q-learning [3] is an incremental reinforcement learning method. According to Q-learning, the agent selects an action based on an action-value function (called the Q-function), $Q(s, a)$,which is a scalar function and represents the value that the agent executes the action a at the state s. The Q-function is updated using the agent's experiences. The learning flow is as follows:

1. For the current state s, the agent executes an action a_i ($\in A$) with the probability:

$$\pi(a_i \mid s) = \begin{cases} \frac{1-\varepsilon}{h} + \frac{\varepsilon}{|A|} & (a_i = \mathrm{argmax}_b Q(s,b)) \\ \frac{\varepsilon}{|A|} & (\text{otherwise}) \end{cases} \quad (2),$$

where $\pi(a_i \mid s)$ represents the probability that the agent executes the action a_i in the state s, and $|A|$ is the number of the element of the action set A, and ε ($\in [0,1]$) is a parameter for determining the randomness of the action selection, and h is the number of the action $b(\in A)$, with which the value of the Q-function $Q(s,b)$ is max.

2. The agent executes the action a selected in step 1. The environment changes to a new state s' according to $P_{s}{}^{a}{}_{s'}$, and a reward r is given from the environment. The Q-function $Q(s,a)$ is updated as follows:

$$Q(s,a) \leftarrow (1-\alpha)Q(s,a) + \alpha(r + \gamma \max_{a' \in A} Q(s',a')) \quad (3),$$

where α $(0 \leq \alpha < 1)$ is a parameter called the learning rate, and γ $(0 \leq \gamma \leq 1)$ is a parameter called the discount factor.

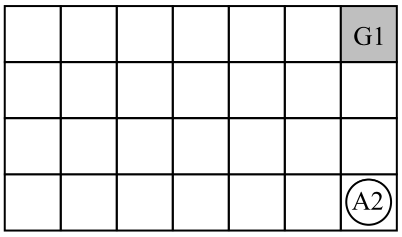
3. If the new state s' satisfies a terminal condition, then the episode ends. Otherwise, let $s' \rightarrow s$ and go back to step 1.

4. Multi-agent Reinforcement Learning

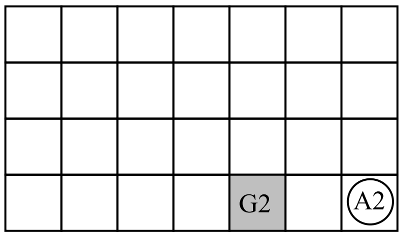
4.1. Point of Problem

Here, I consider that each agent in the task shown in section 2 learns its policies by using reinforcement learning. It is remarked that the goal situation-2 (shown in Fig.2 (b) or Fig.3 (b)) is more effective than the goal situation-1 (shown in Fig.2 (a) or Fig.3 (a)) from the global (social) view point.

If the agents learn its policy by using the general reinforcement learning algorithms proposed so far (like Q-learning), it is unfortunately expected that the leaning converges to the goal situation-1. The reason is that the agent applied the reinforcement learning algorithms proposed so far is greedy for the nearer goal. Therefore, it is expected that the agent A1 does not give up going toward the near goal position G1. Even if the agent A2 aims at reaching the nearer goal position G1 with shortest time step (4 time steps), the agent A1 reaches the goal position G1 (with 2 time steps) earlier than the agent A2. Furthermore, because the positive reward of the goal position G1 disappears after the agent A2 reaches the goal position G1, the agent A2 inevitably goes toward another goal position G2. That is, it is difficult that the agent applied the reinforcement learning algorithms proposed so far learns the "sociality" in a multi-agent environment as shown in section 2.



(a) Task space-1.



(b) Task space-2.

Fig.4. Virtual task spaces from the agent A2's viewpoint.

In this article, I propose a reinforcement learning framework, which is based on Q-learning [3], that the agents are able to learn the "sociality" in a multi-agent environment as shown in section 2.

4.2. Learning Framework

First, the grid space shown in Fig.1 is divided into two virtual task spaces form each agent's view point. In each virtual task space, there is only one goal position. For example, the two virtual task spaces from the agent A2's view point are as shown in Fig.4 (a) and (b).

Each agent has two Q-functions $Q^1(c_1,a)$, $Q^2(c_2,a)$ corresponding to the two virtual task spaces c_1, c_2, respectively. Here, c_1, c_2 are the virtual task spaces corresponding to the goal positions G1, G2, respectively. Each agent does Q-learning with each virtual task space (that is, each Q-function), separately and simultaneously. At the beginning of each learning episode, the agent decides which of Q-function should be used by using the value of w^i_j. Here, w^i_j, which is a variable, is updated when either of the goal situations is satisfied. The update rule is as follows:

$$w^i_j \leftarrow \theta w^i_j + (1-\theta) y^i_j \quad (4),$$

where θ ($\in [0,1]$) is a parameter, and y^i_j is the number of the time steps from the initial state (as shown in Fig.1) until a goal situation, in which the agent Ai reaches the goal position Gj, is satisfied. The measure of choosing the Q-function, which is used when the agent selects the action during the episode, is as follows. If $w^i_1 > w^i_2$, then

$Q^2(c_2,a)$ is chosen with probability $\omega\,(0 \leqslant \omega \leqslant 1)$, or $Q^1(c_1,a)$ is chosen with probability $1-\omega$. If $w^i_1 \leqq w^i_2$, then $Q^1(c_1,a)$ is chosen with probability $\omega\,(0 \leqslant \omega \leqslant 1)$, or $Q^2(c_2,a)$ is chosen with probability $1-\omega$.

The learning flow is as follows:

1. At the beginning of each episode, the agent decides which of Q-function should be used during the episode based on the value of w^i_j as above mentioned. Here, it assumes that $Q^1(c_1,a)$ was chosen.
2. For the current state s (the virtual task spaces c_1, c_2), the agent executes an action a_i ($\in A$) with the probability:

$$\pi(a_i \mid s) = \begin{cases} \frac{1-\varepsilon}{h} + \frac{\varepsilon}{|A|} & (a_i = \operatorname{argmax}_b Q^1(c_1,b)) \\ \frac{\varepsilon}{|A|} & (\text{otherwise}) \end{cases} \quad (5).$$

3. The agent executes the action a selected in step 2. The environment changes to a new state s' (the virtual task spaces c_1', c_2') and a reward r is given from the environment. The two Q-function is updated as follows:

$$Q^1(c_1,a) \leftarrow (1-\alpha)Q^1(c_1,a) + \alpha(r + \gamma \max_{a' \in A} Q(c_1', a')) \quad (6),$$

$$Q^2(c_2,a) \leftarrow (1-\alpha)Q^2(c_2,a) + \alpha(r + \gamma \max_{a' \in A} Q(c_2', a')) \quad (7).$$

4. If the new state s' satisfies a terminal condition, then the episode ends. Otherwise, let $s' \rightarrow s$ and go back to step 2.

5. Experiments and Results

Here, I did two experiments for the task shown in section 2. One is the experiment that the usual Q-learning (as shown in subsection 3.2) is applied to the agents, the other is the experiment that proposed reinforcement learning framework (as shown in subsection 4.2) is applied to the agents. The parameters are: α =0.3, γ =0.9, ε =0.3 × 0.9977$^{\text{num\_ep}}$, θ =0.3 × 0.9977$^{\text{num\_ep}}$, ω =0.5 × 0.9977$^{\text{num\_ep}}$, where num_ep is the number of the learning episodes.

In the case that the usual Q-learning is applied, the learning converges to the goal situation-1 at the 1000 learning episodes. In the case that the proposed learning framework is applied, on the other hand, the learning converges to the goal situation-2 at the 1000 learning episodes.

6. Conclusions

In this article, I propose a reinforcement learning framework, which is based on Q-learning, that the agent is able to learn "sociality" in a multi-agent environment. In this framework, the agent learned to ignore the near goal, which is left for the other agent, and go toward the farther goal.

References

[1] R. S. Sutton, and A. G. Barto, *Reinforcement Learning: An Introduction*, MIT Press, 1998.
[2] M. Wooldridge, *An Introduction to Multi Agent Systems*, John Wiley & Sons, 2009.
[3] C. J. C. H. Watkin, and P. Dayan, Technical note Q-learning. *Machine Learning* 8(3):279-292, 1992.
[4] M. L. Puterman, *Markov Decision Processes*. Wiley Interscience, 1994.