View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Elsevier - Publisher Connector

Procedia Computer Science

Volume 88, 2016, Pages 94–101

7th Annual International Conference on Biologically Inspired Cognitive Architectures, BICA 2016

A Study of an Indirect Reward on Multi-agent Environments

Kazuteru Miyazaki

National Institution for Academic Degrees and Quality Enhancement of Higher Education, Kodaira, Tokyo, Japan. teru@niad.ac.jp

Abstract

In a multi-agent learning where multiple agents are learning, there is a problem about *an indirect reward* that is how to distribute a reward to an agent that does not obtain a reward directly.We have shown the theorem [3] about "negative effect" of an indirect reward. This paper focuses on the "positive effect" of an indirect reward such as an elimination of *the perceptual aliasing problem* [1]. First, we describe the relationship the theorem [3] and the "positive effect" of the indirect reward. Next, we propose a method to eliminate the perceptual aliasing problem and show the effectiveness of the proposed method by numerical examples.

Keywords: Reinforcement Learning, Multi-agent Learning, Indirect Reward, Direct Reward, Perceptual Aliasing Problem, Profit Sharing

1 Introduction

Among machine-learning approaches, reinforcement learning (RL) focuses most on goal-directed learning from interaction [10]. It is very attractive because it can use dynamic programming (DP) to analyze behavior. RL generally treats, rewards and penalties as teaching signals in learning. DP-based RL involves optimizing behavior under reward and penalties signals designed by RL users on the Markov Decision Processes (MDPs).

RL is difficult to design to fit real-world problems because, first, interaction requires too many trial-and-error searches and, second, no guidelines exist on how to design values of reward and penalty signals. While these are essentially neglected in theoretical researches,they become serious issues in real-world applications, e.g., unexpected results arise if inappropriate values are assigned to reward and penalty signals [4].

We are interested in approaches treating reward and penalty signals independently. We also want to reduce the number of trial-and-error searches by strongly enhancing successful experience — a process known as exploitation-oriented learning (XoL) [4].XoL has four features.(1) XoL learns more quickly by strongly tracing successful experiences. (2) XoL treats, rewards and penalties as independent signals, letting these signals be handled more intuitively and easily than the handling of concrete values. (3) XoL does not pursue optimality efficiently, which can

Selection and peer-review under responsibility of the Scientific Programme Committee of BICA 2016

© The Authors. Published by Elsevier B.V.

doi:10.1016/j.procs.2016.07.411

be acquired by multi-start resetting all memory to get a better policy. (4) XoL is strong in the class that exceeds MDPs because it is a Bellman-free method. An example of XoL learning methods for a type of a reward includes Profit Sharing (PS) [2].

In this paper, we focus on a multi-agent learning [7, 3, 9] where multiple agents are learning. In the multi-agent learning, there is a problem about an indirect reward that is how to distribute a reward to an agent that does not obtain a reward directly. We have been shown the theorem about "negative effect" of an indirect reward in the paper [3] in order to avoid to obtain no reward in the multi-agent system.

This paper focuses on the "positive effect" of the indirect reward such as an elimination of *the perceptual aliasing problem* [1]. First, we describe the relationship the theorem [3] and the "positive effect" of the indirect reward. Next, we propose a method to eliminate the perceptual aliasing problem and show the effectiveness of the proposed method by numerical examples.

2 The Domain

Consider an agent in an unknown environment. After perceiving sensory input from the environment, the agent selects and executes an action. Time is discretized by one input-action cycle. *An action* is selected from among the discrete types.Input from the environment is called *a state*. The discrete types of action is called *the number of actions*. The pair consisting of the state and an action selected in a state is called *a rule*. Rewards and penalties based on a series of actions are provided from the environment, and a reward is given to a state or an action causing transition to a state in which our purpose is achieved, whereas a penalty given to a state or corresponding action in which our purpose is not achieved. In this paper, we consider the cast that there is no penalty.

A rule series that begins from a reward/penalty state or an initial state and ends with the next reward/penalty state is called *an episode*. If an episode contains rules of the same state, but paired with different actions, the partial series from one state to the next is called *a detour*. A rule always existing on a detour is called *an irrational rule*, and otherwise called *a rational rule*. A function that maps states to actions is called *a policy*. The policy with a positive amount of reward acquisition expectations is called *a rational policy*. *The optimal policy* is a policy that can maximize the amount of a reward.

We call indistinction of state values a *type 1 confusion*. Furthermore, we call indistinction of rational and irrational rules a *type 2 confusion*. In general, if there is a type 2 confusion in some sensory input, there is a type 1 confusion in it. By these confusions, we can classify environments. *Q-learning* (QL) [10], that guarantees the acquisition of an optimal policy in MDPs, is deceived by the type 1 confusion since it uses state values to make a policy. PS is not deceived by the confusion since it does not use state values. On the other hand, learning systems that use the weight (including QL and PS) are deceived by the type 2 confusion. If the perceptual aliasing problem occurs, the type 2 confusion may occur. Though there are many researches about the perceptual aliasing problem in *Partially Observable Markov Decision Processes* (POMDPs) [8] it is often eliminated by enriching the sensory input in real applications [6].

An environment in which multiple agents are present is referred to as *a multi-agent environment*. The learning of a multi-agent environment is referred to as *a multi-agent learning*.

This paper focuses on a multi-agent environment that only one type of reward is present and assumes *a synchronous* environment in which only one agent is performing in each time as same as the the paper [3]. Though the paper also assume that there is no type 2 confusion, we do

not require the assumption. The condition that there is no type 2 confusion in the multi-agent learning is more strict condition than the case of single agent learning in general.

3 Multi-agent Learning

3.1 Expectations for Multi-agent Learning

In a multi-agent learning, it may be possible to solve the perceptual aliasing problem. For example, if only the agent A is present in the environment in Fig 1 and each agent only perceives the vertical and horizontal states themselves, the perceptual aliasing problem arises, since the agent perceives a hatched state as the same state. On the other hand, such the problem could be solved when the agent B is present and is performing the appropriate motion such as approaching to the agent A. In this case, the field of view of the agent A is possible to capture the agent B.

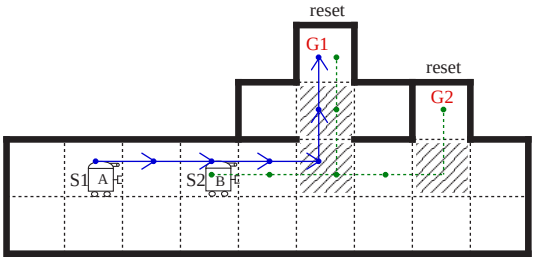

Figure 1: An environment that is used to show the effectiveness of multi-agent learning.

Such the behavior of the agent B is likely to be derived by an indirect reward. The authors has been analyzed the "negative effect" of an indirect reward in the paper [3]. In contrast, in this paper, we consider the "positive effect" such as the elimination of the perceptual aliasing problem. Next, we introduce *the indirect reward theorem* [3] that is a typical theorem in multi-agent learning.

3.2 The Indirect Reward Theorem on Multi-agent Learning

We use PS. When the agents obtain a reward, the learning of PS proceeds by updating the weight of the rules that have been utilized to obtain a reward. We have proved how to distribute that guarantees the acquisition of rational policy in the environment where there is no type 2 confusion in the paper [2]. It is called *the Rationality Theorem of PS*. We use the following equation satisfying the rationality theorem of PS.

$$f_n = \frac{1}{M} f_{n-1}, \quad n = 1, 2, \cdots, W_a - 1. \tag{1}$$

where there are M types of action and the agent obtains a direct reward for the value of R ($R > 0$). The episode in which weights are updated is referred to as *a reinforcement interval*.

If we introduce an indirect reward μR ($\mu \geq 0$), there is a possibility that a part of an effective rule changes to an ineffective rule. As a result, here is a possibility that the reward acquisition of the entire system becomes zero. Theorem regarding the range of the value of μ in order to prevent this is the following [3].

Theorem 1 (Rationality Theorem of in Multi-agent Learning)**.**
In the multi-agent learning that has an indirect reward μR ($\mu \geq 0$), the necessary and sufficient condition of μ in order to avoid obtaining no reward in the entire system is the following.

$$0 \leq \mu < \frac{M-1}{M^W(1-(\frac{1}{M})^{W_0})(n-1)L}. \tag{2}$$

where R is the value of a direct reward, M is the maximum number of conflicting rules in the same sensory input, L is the maximum number of conflicting rational rules, W is the maximum episode length of a direct-reward agent, W_0 is the reinforcement interval of indirect-reward agents and n is the number of agents.

4 A Study of an Indirect Reward on Multi-agent Learning

4.1 About the Indirect Reward Theorem

Theorem 1 is always necessary to be satisfied when we use an indirect reward. Furthermore, there is a need to satisfy the theorem even when conditions such as "it is required for all learning agents to obtain a reward."

Theorem 1 assumes the learning in the class where there is no type 2 confusion, since it is based on the rationality theorem of PS. In a multi-agent learning, the learning result becomes unstable since there is a possibility that *the concurrent learning problem* [7] occurs. Therefore, the condition, that there is no type 2 confusion, becomes more strict constraint in a multi-agent learning in comparison with a single-agent learning.

The purpose of this paper is to eliminate the perceptual aliasing problem in multi-agent learning. Furthermore, we aim to obtain a reward as uniformly as possible among all learning agents. This purpose is considered to be those that also contribute to the realization of a sustainable society through altruistic agent. This is a way of thinking that leads to the realization of a society in harmony where everyone is aiming to become as much as possible equally happy instead of only one person can be a winner.

By reviewing the theorem 1 in the above aspect, we can derive the following theorem.

Theorem 2 (The necessary condition to be obtained a reward by all learning agents)**.**
When an indirect and direct reward is updated in the same weight table, the necessary condition to be obtained a reward by all learning agents is the inequality (2).

4.2 Relaxation of the Indirect Reward Theorem

By introducing an indirect reward, it may be possible to eliminate the perceptual aliasing problem as shown in Fig 1. We should use more indirect rewards in order to achieve the elimination of the perceptual aliasing problem. However, it is not permitted to grant a value that exceeds the scope of Theorem 2 in the case where indirect and direct rewards is given for the same weight table.

In particular, from Theorem 2, even in the case to achieve the object of this paper, that "all of the learning agents aim to obtain a reward as uniformly as possible," we are restricted by the constraints of Theorem 2. It means that it is not allowed to give a lot of indirect rewards. Therefore, in order to enhance the effect of an indirect reward, we have to loosen the conditions that are assumed in the theorem.

As a specific relaxation method, for example, it can be considered the following methods.

1. **A Reward is given when all of the learning agents took the ideal behavior**
   The ideal behavior requires to obtain a reward as uniformly as possible among learning agents. It will be unlikely occurred through trial-and-error searches. A reward, therefore will be difficult to be obtained with this method.

2. **Reward is given independently for each agent and initialize the weights table at the time a reward is no longer obtained**

Though it is likely to be easier to obtain a reward than (1), in general, it will be frequently happened to initialize the weight table. It may therefore take many actions for learning.

3. **Every learning agent has weight tables for each learning agent, and they are switched in the action selection**
   If there is an agent that can no longer be obtained a reward, the other agents take altruistic actions utilizing the weight table which is updated when the agent had obtained a reward in the past. When the agent can obtain a reward, agents with altruistic behavior return to the action selection using a weight table themselves. Remark that this method requires the broadcast of a reward as with the case of using an indirect reward.

These methods from 1 to 3 are sufficient condition in which all of the agent are able to obtain a reward. Especially, though the method 3 that prepares weight tables for the number of learning agents is disadvantageous in terms of memory, it is expected to be excellent in terms of learning speeds. This paper therefore uses the method of 3.

5 Elimination of the Perceptual Aliasing Problem by Multiple Weight Tables

5.1 Basic Concept

In this section, we describe a specific learning method using multiple weight tables. Each agent has a weight table in a number of equal to the number of agents. Therefore, in selecting the actual action, it is necessary to some conflict resolution between them.

As a conflict resolution method, though we can consider to integrate multiple weight tables to one weight table, we propose a method of selecting only one weight table from multiple weight tables.

Section 5.2 describes the method of determining the agent, that is called *the altruistic agent*, to perform the action selection using the weight tables that had been enhanced by the rewards that were obtained by the other agents, rather than the weight table that had been strengthened by the reward for oneself. Section 5.3 describes the procedure after a reward acquisition.

Learning of altruistic behavior is an important issue in the multi-agent learning, though previous papers pay attention to the fact that altruistic behavior has been acquired as a result of learning, this paper focuses on providing a framework for the learning.

5.2 How to Determine the Altruistic Agent

If there is no altruistic agent, we make a selection and determination of an altruistic agent. Though we can consider the case that multiple agents perform the altruistic behavior for each other agent, this paper treats that only one agent is a target for an altruistic agent in order to deal with a more simpler case.

Specifically, determine the altruistic agent by the following method. First, find an agent with less number of times of reward acquisition. This agent is regarded as a target of "altruistic". If there are several agents where the number of times of reward acquisition is the minimum, the agent that has been found in the first is regarded as the altruistic target.

All agents other than the agent that has been regarded as the altruistic target become the agent to perform the altruistic behavior. In other words, all agents other than the agent in which the number of reward acquisition times is the minimum select an action using the weight table that is updated when the agent of the minimum of the number of reward acquisition times

had been rewarded. Though it can be also conceivable that only agent that has the maximum number of times of reward acquisition helps the agent that is the minimum number of times of reward acquisition, it will be discussed in numerical experiments.

5.3 Procedure after Reward Acquisition

If the agent that carried out the action obtains a reward, all weight tables corresponding to the agent that had obtained a reward are updated not only the agent to obtain a direct reward.

We use PS to update the weight table. Reward value is assigned using the same values for all agents. Reinforcement interval is the same as the value of the agent that had been obtained a direct reward. Initialization of the episode is carried out only for the agent that had been obtained a direct reward.

Finally, if the agent that had been obtained a direct reward is coincident to an altruistic target, altruistic behavior to help the agent is finished.

6 Numerical Experiments

6.1 Setting

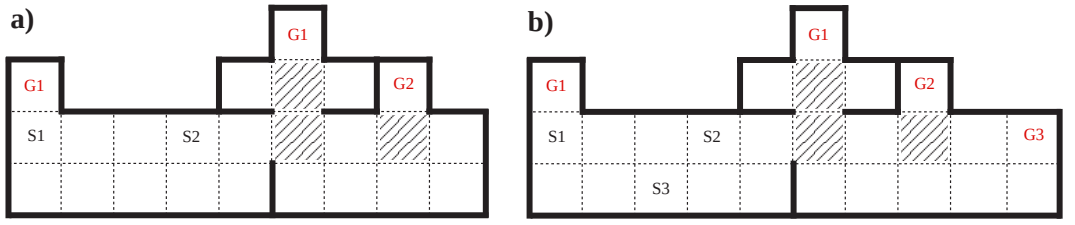

Figure 2: a) 2 agents environment that is used to show the effectiveness of the proposed method. b) 3 agents environment that is used to show the effectiveness of the proposed method.

We aim to obtain a reward as uniformly as possible in the case where more than one agent performs learning. The effectiveness of the proposed method in Section 5 is evaluated by using the environment as shown in Fig. 2 a) and b). Fig. 2 a) and b) are cases where the number of agents is 2 and 3, respectively.

Each agent is located in any one of the squares, and can perceive the vertical and horizontal states of the square that it is located. Thick line is a wall, preventing the perception of the square of the other side of the wall. As a result, the hatched squares are perceived as the same square for an agent. The perception is one of $\{there\ is\ nothing, there\ is\ the\ other\ agent, there\ is\ a\ wall\}$ on each square. That is, the agent cannot be distinguished each agent other than itself. Each agent $(i = 1, 2, 3)$ is located in $S_i(i = 1, 2, 3)$ initially.

The agent selects an action from $\{up, down, left, right\}$ movement after receiving the sensory input. That multiple agents can occupy the same square. Each agent $(i = 1, 2, 3)$ aims to move to the target state $G_i(i = 1, 2, 3)$. A reward is given, when an agent transit to the target state $G_i(i = 1, 2, 3)$, and returns to the square of the initial position $S_i(i = 1, 2, 3)$. Transition to the wall is not allowed, it will remain in the original state.

We use the ϵ-roulette strategy as an action selection method depending on the current weight of the rule. The upper limit of the action is 100,000 in this paper. After ϵ values was calculated

by $\epsilon = 1.0 - \frac{The\ number\ of\ times\ of\ action\ selection}{50000.0}$, generating a random number between 0.0 to 1.0. If the value of ϵ is larger than the random number, we make a roulette selected in proportion to the values of weight tables, otherwise, we make a random selection of the action selection. The ratio of the roulette and random selection has been reversed in 50,000 actions. If the ϵ value is zero or less, we set ϵ=0.0.

We compare the proposed method with SARSA [10], normal PS, random selection, and *max help* that has changed the method of determining the agent to perform the altruistic behavior.

6.2 Preliminary Experiment: The Results of SARSA

SARSA has two important parameters such as learning rate α and the discount rate γ. In general, these values are a significant impact on the learning result. Therefore, we have changed the values of these parameters for preliminary experiment. The results are shown in Table 1. The experiment was carried out 100 times by changing random seeds in the case of the two agents (Fig. 2 a)). The table shows the average number of reward acquisition times at the top and the standard deviation at the bottom.

Table 1: Results of SARSA

	$\alpha = 0.5$ $\gamma = 0.5$	$\alpha = 0.5$ $\gamma = 0.9$	$\alpha = 0.8$ $\gamma = 0.9$	$\alpha = 0.95$ $\gamma = 0.95$
agent 1	2352.1	**991.3**	978.4	817.6
	76.5	43.2	45.5	44.4
agent 2	711.2	**821.8**	633.8	484.9
	65.0	35.0	31.1	24.1

In Table 1, we can confirm that the case of α=0.5 and γ=0.9 is the most closest to our expected behavior where the two agents has become a more uniform number of times of reward acquisition. We therefore use the parameter set in the experiment using SARSA.

Table 2: Results of 2 and 3 agents.

	Proposed	PS	SARSA	random	max help
agent 1	**2547.9**	69310.6	991.3	2529.1	2776.0
	1219.4	139.5	43.2	73.8	1480.6
agent 2	**2259.4**	669.5	821.8	230.5	2487.5
	1501.1	37.3	35.0	14.4	1757.8
agent 1	**2461.6**	69507.3	2655.0	2583.5	5770.5
	789.8	142.8	64.6	70.0	442.0
agent 2	**2439.9**	660.6	664.9	188.7	1133.8
	834.3	35.3	51.2	13.8	413.1
agent 3	**2439.6**	64707.0	912.8	75.9	5392.3
	834.3	274.4	41.6	7.63	67.1

6.3 Results and Discussion

The results of the case of two and three agents are shown in the upper part of the lower part of Table 2, respectively. Construction of these tables is the same as Table 1. We can confirm that the proposed method is the most closest to our expectations in these tables. In particular, the result of the three agents is more pronounced. We can confirm the effectiveness of the proposed method through these results.

As a method for determining the altruistic agent, we can consider that the agent that obtained the most reward helps the agent that is the minimum number of times of reward acquisition. As a result of implementation of this method is the part that is displayed with the *max help*.

In the case of two agents, the proposed method and the max help should show the same behavior. As expected, there is no big difference among them in the case of two agents. On the

other hand, in the case of three agents, the number of times of reward acquisition of the agent 2 in max help has deteriorated. As in this environment if there is an agent that is clearly difficult to obtain a reward, to help the agent was particularly effective for using all the other agents. In general, since the structure of the environment is unknown, full utilizing of all agents would not always be required. It is considered to be effective that dynamically determine the number of agents to carry out altruistic behavior. A Specific way to do this is one of our future works.

7 Conclusions

In a multi-agent learning in which multiple agents are learning, there is the indirect reward problem how to distribute a reward to the agent other than the agent to obtain a reward directly. This paper focuses on the "positive effect" such as the elimination of the perceptual aliasing problem, though we have proven the theorem of the analysis of the "negative effect" of an indirect reward [3]. On which to organize the relationship between our previous theorem, we proposed a method to eliminate the perceptual aliasing problem, and showed the effectiveness of the proposed method by numerical experiments. In the future, we make our method to apply to several areas [5, 6].

Acknowledgment

This work was supported by JSPS KAKENHI Grant Number 26330267.

References

[1] Chrisman, L.: Reinforcement learning with perceptual aliasing: The perceptual distinctions approach, Proc. of the 10th National Conf. on Artificial Intelligence, pp.183-188 (1992)

[2] Miyazaki, K., Yamamura, M. & Kobayashi, S: On the Rationality of Prot Sharing in Reinforcement Learning, Proc. of the 3rd International Conference on Fuzzy Logic, Neural Nets and Soft Computing, pp.285-288 (1994)

[3] Miyazaki, K. & Kobayashi, S.: Rationality of Reward Sharing in Multi-agent Reinforcement Learning, New Generation Computing, 19, 2, pp.157-172 (2001)

[4] Miyazaki, K. & Kobayashi, S.: Exploitation-Oriented Learning PS-r$^{\#}$. J. of Advanced Computational Intelligence and Intelligent Informatics, 13, 6, pp.624-630 (2009)

[5] Miyazaki, K. & Takeno, J.: The Necessity of a Secondary System in Machine Consciousness, Procedia Computer Science, 41, pp.15-22 (2014)

[6] Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D. & Riedmiller, M.: Playing Atari with Deep Reinforcement Learning, NIPS Deep Learning Workshop 2013 (2013)

[7] Sen, S. & Sekaran, M.: Multiagent Coordination with Learning Classifier Systems, Adaptation and Learning in Multi-agent systems, pp.218-233 (1995)

[8] Spaan, M. T.: Partially Observable Markov Decision Processes, in M. Wiering and M. van Otterlo eds., Reinforcement Learning, Springer-Verlag Berlin Heidelberg, chapter 12, pp.387-414 (2012)

[9] Stone, P., Sutton, R. S. & Kuhlamann, G.: Reinforcement Learning toward RoboCup Soccer Keepaway. Adaptive Behavior, 13, 3, pp.165-188 (2005)

[10] Sutton, R. S. & Barto, A. G.: Reinforcement Learning: An Introduction, A Bradford Book, MIT Press (1998)