A Theory of Profit Sharing in Dynamic Environment

Shingo Kato and Hiroshi Matsuo

Department of Electrical & Computer Engineering, Nagoya Institute of Technology,
Gokiso-cho, Showa-ku, NAGOYA, 466-8555, JAPAN
{katosin@mars., matsuo@}elcom.nitech.ac.jp

Abstract. Reinforcement learning is one of the most popular learning method for machine learning. Some reinforcement learning algorithms for adapting to the dynamic environment are proposed. In this paper, the number of episode to suppress the ineffective rule after the change of the environment was examined analytically. Afterwards, the forgettable profit sharing method to suppress the ineffective rule quickly is proposed, and the effectiveness was experimentally confirmed comparing the proposed method with conventional method.

Keywords: reinforcement learning, profit sharing, dynamic environment

1 Introduction

Reinforcement learning[1–3] is one of the most popular learning methods for machine learning. It aims to adapt a system to a given environment according to rewards. The application in a robot system is expected in order to realize the autonomous action in unknown environment and dynamically changing actual environment [4, 5]. The use of the routing in dynamic networks is proposed in [6, 7]. Reinforcement learning in the multi-agent system is recently advanced[8–10]. For example, the research for the speedup of the learning by transmitting the learning result between agents, and imitating other agent is proposed in [11]. At other, the theoretical consideration of reward allocation in profit sharing in the multi-agent reinforcement learning is also proposed in [12].

In the conventional reinforcement learning algorithm, static environment is assumed. However, it is important to consider changes in the environment (dynamics of the environment). Yamamoto proposed the detection algorithm to recognize environment changes using stochastic gradient method [13, 14]. However, learning method after environment changes is important.

In this paper, the number of episodes which needs to suppress the ineffective rule as the environment changes was examined analytically. And, the forgettable profit sharing which quickly suppresses the ineffective rule is proposed.

The paper is organized as follows: In Section 2, the rationality theorem of profit sharing is discussed as a preparation. Number of episodes necessary for suppression of the ineffective rule as the environment changes is also discussed.

In Section 3, the proposal technique is discussed. In Section 4, the experimental method and result are discussed. Finally, in Section 5 we conclude with a brief summary.

2 Profit Sharing in the environmental change

2.1 Preparation

Profit Sharing memorizes rule series, which consists of the pair of state x and action a in the episode, and the rule on the series are reinforced in the following equation when the reward was obtained.

$$w(x_i, a_i) \leftarrow w(x_i, a_i) + f(r, i) \tag{1}$$

where $w(x_i, a_i)$ is the weight of the rule of i on the episode series, r is reward value, and f is the reinforcement function. Afterwards, we describe the learning machine as an agent.

The rule "if x then a" which chooses action a of state x is described as $\overrightarrow{xa}$. Rule series of the interval is called detour, when the rule differs in some episode for the identical state has been chosen. For example, the detour $(\overrightarrow{xb} \cdot \overrightarrow{zb} \cdot \overrightarrow{yb})$ exists(Fig.2) for the episode $(\overrightarrow{yb} \cdot \overrightarrow{xb} \cdot \overrightarrow{zb} \cdot \overrightarrow{yb} \cdot \overrightarrow{xa})$ in the environment of Fig.1. The rule on the detour may not contribute to the acquisition of the reward. The rule in detour always is called ineffective rule in the episode, and the rule except for it is called effective rule. When an ineffective rule competes with an effective rule, it should reinforce clearly the effective rule.

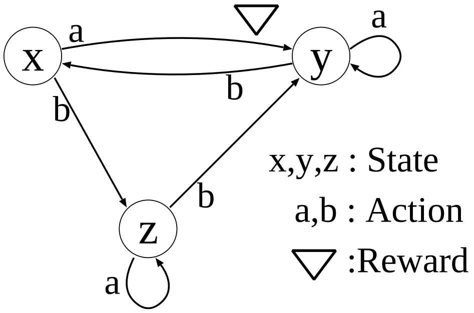

Fig. 1. Environment

The condition described in (2) which effective rule always suppresses the ineffective rule was derived in [15].

$$L \sum_{j=i}^{W} f_j < f_{i-1}, (\forall i = 1, 2, \cdots, W.) \tag{2}$$

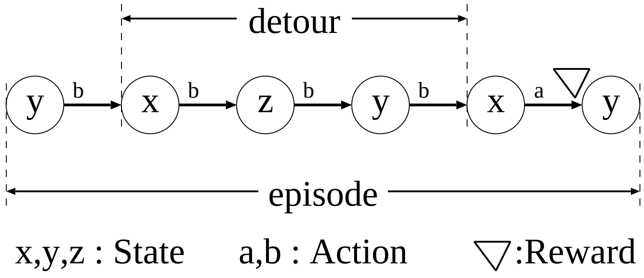

Fig. 2. Example of episode and detour

where W is the largest length of episode, and L is the maximum number of effective rules which exists under an identical state. Equal ratio decrease function shown next, is considered as the simplest reinforcement function which satisfies the equation (2),

$$f_n = \frac{1}{S} f_{n-1}, \quad n = 1, 2, \ldots, W - 1. (S \geq L + 1) \tag{3}$$

The ratio of weight of effective rule and weight of the ineffective rule which acquired the maximum reward is shown in (4).

$$P \times f_{i-1} = L \sum_{j=i}^{W} f_j, (P < 1) \tag{4}$$

For example, P is shown by the following equation, when the reinforcement function presented in equation (3) with $W = \infty$.

$$P = \frac{L}{S - 1} \tag{5}$$

2.2 The environment changes

The change of the environment is considered. The reward, which gives the effective rule of identical state, is considered to be constant for the simplification.

The most difficult condition in which the ineffective rule is suppressed in the new environment is considered. It is clear that to suppress only recursive and ineffective rule is the most difficult task in the static environment [15]. Recursive rule is the rule in which the state does not change as a result of action. According to the amount of the ineffective rule, the suppression of the ineffective rule becomes hard. Therefore, the most difficult condition is the case in which the only recursive and ineffective rule has got all reward of the effective rule in previous environment.

In such conditions, the reward after X episode learning step becomes $Rw \cdot \frac{X}{L}$, because the effective rule gets the reward in the every L episode. The initial value of the ineffective rule is the value obtained multiplying the number of episodes from previous environment by the reward which the effective rule gets before environment changing. And, the ineffective rule gets the reward obtained multiplying P by the reward which the effective rule gets in the new environment. Therefore, the weight of the ineffective rule is calculated as $G \cdot Rw + P \cdot Rw \cdot \frac{X}{L}$.

The necessary condition for suppression of the ineffective rule is shown by following equation.

$$G \cdot Rw + P \cdot Rw \cdot \frac{X}{L} < Rw \cdot \frac{X}{L} \tag{6}$$

$$\frac{L}{1-P} \cdot G < X \tag{7}$$

where G is the number of episodes of previous environment and Rw is the value of the reward in which the effective rule gets.

A lot of episodes X shown in equation (7) is needed to suppress ineffective rule in the most difficult condition. For example, number of episodes which needs to certainly suppression of the ineffective rule is 12th times as much as number of episode in the previous environmental change in $L = 3$ and $P = 0.75$.

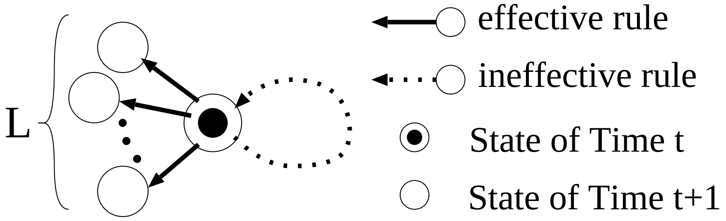

Fig. 3. Example of state with only recursive and ineffective rule.

3 Forgettable Profit Sharing

The reason why the conventional method could not quickly adapt for environmental changes is that the previous information is kept permanently. For adapting to the new environment, the adapting method which forget the disadvantageously rule in the new environment quickly is suitable. However, to recognize the environmental change and disadvantageous rule in the new environment is difficult. The forgettable profit sharing in which the weight of rule gradually decreases is proposed.

3.1 Profit sharing with queue

Adding queue with an agent was considered in order to eliminate gradually previous information. The agent puts the reward got in each episode in the queue(Fig.4).

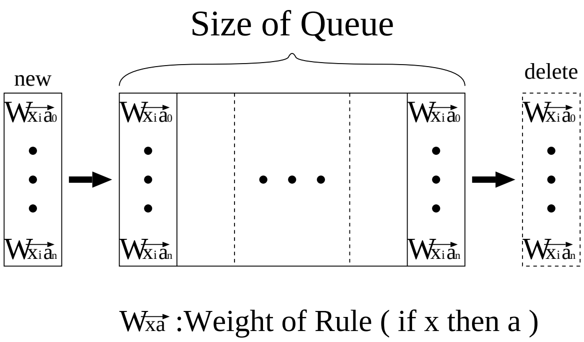

Fig. 4. Queue

Let's assume that the number of the episode from previous environment is larger than the length of the queue. The weight of the effective rule is $Rw \cdot \frac{X}{L}$, which is equal to the conventional method. The initial value of the ineffective rule is $(Q_{size} - X) \cdot Rw$, because to store the reward over the size of the queue is not possible. Therefore , the necessary condition for suppression of the ineffective rule is shown by following equation.

$$(Q_{size} - X) \cdot Rw + P \cdot Rw \cdot \frac{X}{L} < Rw \cdot \frac{X}{L} \tag{8}$$

$$\frac{Q_{size} \cdot L}{1 + L - P} < X \tag{9}$$

where Q_{size} is the size of queue.

3.2 Profit sharing with weighted virtual queue

The propose method in 3.1 adding queue with agent is surely effective. But to apply this method , a lot of queue(memory) is needed. Therefore it is impracticable to use generally. The new method named “weighted virtual queue” is proposed. This method decreases previous reward virtually. Therefore, “weighted virtual queue” is expressed in the following equation.

$$w(x,a) \leftarrow w(x,a) \times \tau + \sum_{k=0}^{W} g(x,a,k) \tag{10}$$

$$g(x,a,k) = \begin{cases} f(r,k) & (\text{if } x = x_k \text{ and } a = a_k) \\ 0 & (\text{else}) \end{cases} \tag{11}$$

where τ is the forgetting rate.

The number of episodes in previous environment G is set to infinity, because the suppression of the ineffective rule becomes difficult, as the initial value of the ineffective rule is larger. The weight of the effective rule is $\frac{1}{L}\sum_{i=1}^{X} Rw \cdot \tau^{i-1}$ which can be derived from equation (10). And, the initial value of the ineffective rule is $\sum_{i=X+1}^{\infty} Rw \cdot \tau^{i-1}$. Hence, the necessary condition for suppression of the ineffective rule is shown by following equation.

$$\sum_{i=X+1}^{\infty} Rw \cdot \tau^{i-1} + \frac{P}{L}\sum_{i=1}^{X} Rw \cdot \tau^{i-1} < \frac{1}{L}\sum_{i=1}^{X} Rw \cdot \tau^{i-1} \tag{12}$$

$$\frac{Rw \cdot \tau^{X}}{1-\tau} < \frac{Rw(1-P)(1-\tau^{X})}{L(1-\tau)} \tag{13}$$

$$\tau^{X} < \frac{1-P}{1+L-P} \tag{14}$$

4 Experiment

4.1 The experimental method.

The grid world in which the discrimination of effective rule and ineffective rule is easy is used for the experiment. Fig.5 shows a grid world in which each cell the agent has four actions(N,S,E,W) and transitions are made deterministically to an adjacent cell, unless there is a block, in which case no movement occurs. The agent gets the reward when it arrives at the goal from the start.

We take w, the initial weight, to be 0.1, and Rw, the reward, to be 1, and S, a parameter of reinforcement function in equation (3), to be 5. In the proposed method described in 3.1, the Q_{size} set at 5.0×10^5. Every 1.0×10^3 episodes were assigned to the same element of the queue. As a result each queue has 500 elements.

In the proposed method which use weighted virtual queue, we set τ to 0.999996.

As a result of the preliminary experiment, it is desirable that the environment was changed after 5.0×10^7 episodes, because 3.0×10^7 episodes are necessary for the convergence using conventional method. But due to the limitation of the accuracy of the floating point calculation, the environment changes after 5.0×10^5 episodes. The goal is located at G_1 in Fig.5, in the case of the environment was not changed. The artificiality environment changes are generated to move the goal from G_1 to G_2.

The experiment performed 10 times as changing seeds of random value, and the obtained mean value is used as final result.

4.2 Experimental result and consideration.

No artificial environment change The learning curve (the relationship between number of episodes and number of steps in which agent get reward) is

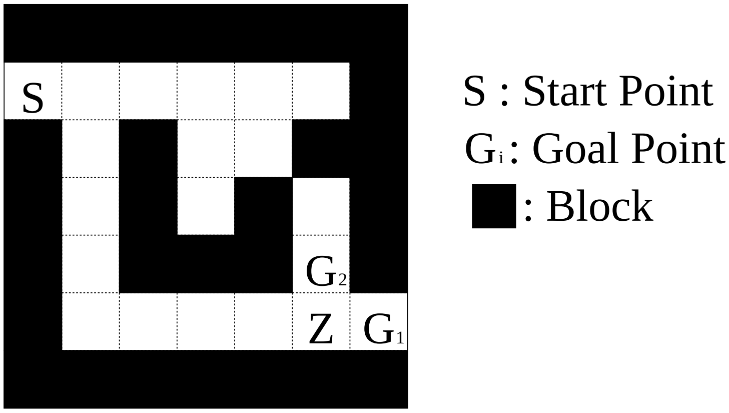

Fig. 5. Grid World for experiment.

shown in Fig.6. Converged numerical value of proposed method "weighted virtual queue" becomes worse compare with other methods. This is because the value of τ was set by adjusting as the environment is changed at 5.0×10^5 episodes. However, the same converged numerical value can be taken by substituting an appropriate value to τ(for example, $\tau = 0.99999993$).

Add artificial environment change The learning curve after artificial environmental change is shown in Fig.7. Fig.8 shows the transition of the weight in the position Z on Fig.5. The horizontal axis shows number of episodes (artificial environment change occurred at 5.0×10^5). The vertical axis shows the rate between weight of rule which moves to North and weight of rule which moves to East. When the rate exceed 1.0, the ineffective rule is suppressed. Fig.8 shows also the theoretical number of episodes necessary to suppress the ineffective rule.

Table.1 shows experimental and theoretical maximum and minimum[1] results which suppress the ineffective rule. This theoretical value of proposal methods in Table.1 is very small, because τ and Q_{size} were decided on the assumption of the most difficult condition. It is appropriate to set these value τ and Q_{size} at most difficult condition , because the knowledge of the largest number of the effective rule can not be informed in advance.

5 Conclusions

In this paper, the analytical consideration of the number of the necessary episode which suppresses the ineffective rule using profit sharing as the environment changes was performed, and the relation between number of episode before the

[1] the derivation procedure is eliminated because of page limitations

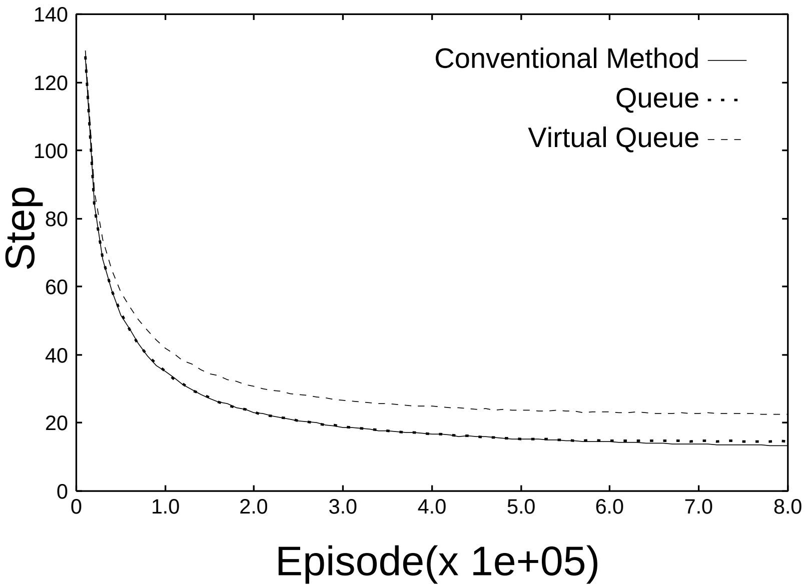

Fig. 6. Learning curve on static environment.

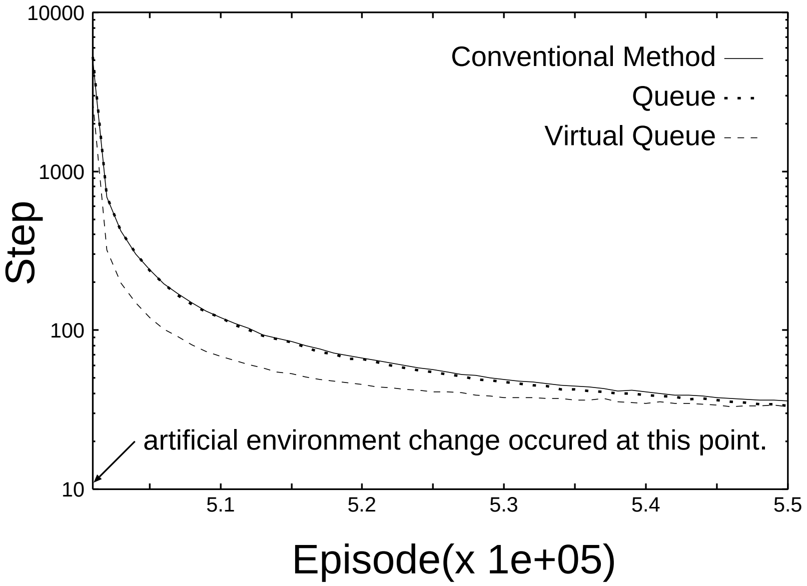

Fig. 7. Learning curve after artificial environment changes.

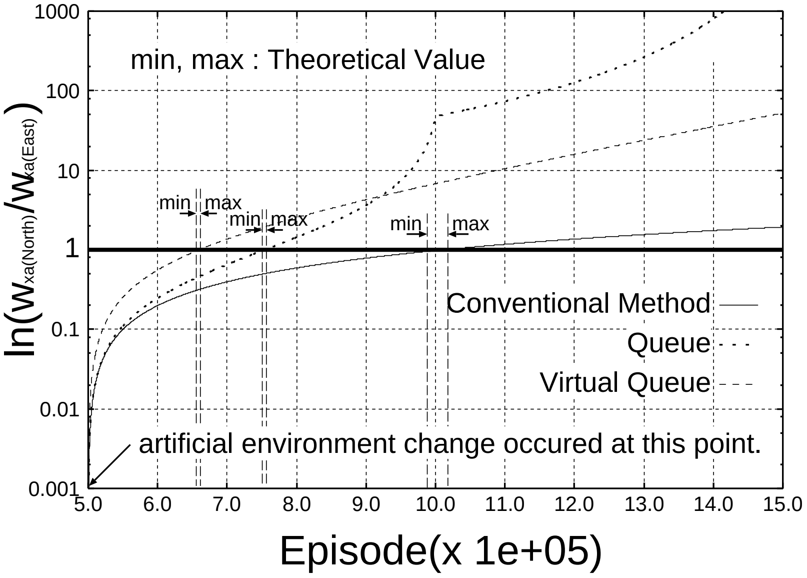

Fig. 8. Rate between weight of effective rule and weight of ineffective rule.

Table 1. Number of episodes in which the ineffective rule is suppressed.

Learning method	Experimental value	Theoretical maximum value(L=1)	Theoretical minimum value(L=1)
Conventional Method	5.12×10^5	5.21×10^5	4.80×10^5
Queue	2.54×10^5	2.55×10^5	2.45×10^5
Virtual Queue	1.60×10^5	1.61×10^5	1.51×10^5

environment changes and number of episode necessary for suppressing the ineffective rule was carried out. Forgettable profit sharing which suppress the ineffective rule after the environment change within the constant episodes was proposed, and the effectiveness of the proposed algorithm was confirmed by the experiment.

The proposed method needs the knowledge of the convergence, but generally this knowledge can not be informed in advance. Future research will address the development of this algorithm to apply it in various practical cases.

References

1. Leslie Pack Kealbling, Michael L. Littman, and Andrew W. Moore: Reinforcement Learning: A Survey, Journal of Artificial Intelligence Research 4, pp237-285(1996)
2. Miyazaki, K., and Kobayashi, S.: Reinforcement Learning Systems for Discrete Markov Decision Processes, JJSAI,Vol.12,No.6,pp811-821(1997)

3. Miyazaki, K., Yamamura, M., and Kobayashi, S.: MarcoPolo : A Reinforcement Learning System Considering Tradeoff Exploitation and Exploration under Markovian Environments, JSSAI, Vol.12, No.1, pp78-88(1997)
4. Yamaguchi, T., Masubuchi, M., Fujihara, K., and Yachida, M.: Accelerating Reinforcement Learning for a Real Robot with Automated Abstract Sub-Rewards Generation, JSSAI, Vol.12, No.5, pp712-723(1997)
5. Minoru A.: Research Issues on Real Robot Reinforcement Learning, JSSAI, Vol.12, No.6, pp.831-836(1997)
6. Devika Subramanian, Peter Druschel, and Johnny Chen: Ants and reinforcement learning: A case study in routing in dynamic networks, In Proceedings of IJCAI-97, 1997.
7. Justin A. Boyan and Michael L. Littman. Packet routing in dynamically changing networks: A reinforcement learning approach. In Jack D. Cowan, Gerald Tesauro, and Joshua Alspector, editors, Advances in Neural Information Processing Systems, volume 6, pages 671–678. Morgan Kaufmann, San Francisco CA, 1993
8. Arai, S., Miyazaki, K., and Kobayashi, S.: Methodology in Multi-Agent Reinforcement Learning –Approaches by Q-learning and Profit Sharing–, JJSAI, Vol.13, No.4, pp609-617(1998)
9. Unemi, T.: Collective Behavior of Reinforcement Learning Agents, MACC, pp137-150(1993)
10. Yamamura, M., Miyazaki, K., and Kobayashi, S.: A Survey on Learning for Agents, JSSAI, Vol.10, No.5, pp683-689(1995)
11. Yamaguchi, T., Miura, M., and Yachida, M.: Multi-Agent Reinforcement Learning with Adaptive Mimetism, JSSAI, Vol.12, No.2, pp323-330(1997)
12. Miyazaki, K., Arai, S., and Kobayashi, S.: A Theory of Profit Sharing in Multi-Agent Reinforcement Learning, JSSAI, Vol.14, No.6, pp1156-1164(1999)
13. Kimura, H., Yamamura, M., and Kobayashi, S.: Reinforcement Learning in Partially Observable Markov Decision Processes: A Stochastic Gradient Method, JSSAI, Vol.11, No.5, pp761-768(1996)
14. Yamamoto, S., Yamaguchi, F., Saito, H., and Nakanishi, M.: A recognization method of environmental change on reinforcement learning, TECHNICAL REPORT OF Institute of Electronics Information and Communication Engineers, AI99-81, pp31-36(2000-01)
15. Miyazaki, K., Yamamura, M., and Kobayashi, S.: A Theory of Profit Sharing in Reinforcement Learning,JJSAI, Vol.9, No.4, pp.580-587(1994)