AUTOMATIC TRANSFER RATE ADJUSTMENT FOR TRANSFER REINFORCEMENT LEARNING

Hitoshi Kono[1], Yuto Sakamoto[2], Yonghoon Ji[3] and Hiromitsu Fujii[4]

[1]Department of Engineering, Tokyo Polytechnic University, Kanagawa, Japan
[2]Department of Electronics and Mechatronics,
Tokyo Polytechnic University, Kanagawa, Japan
[3]Graduate School for Advanced Science and Technology, Japan Advanced Institute of Science and Technology, Ishikawa, Japan
[4]Department of Advanced Robotics, Chiba Institute of Technology, Chiba, Japan

ABSTRACT

This paper proposes a novel parameter for transfer reinforcement learning to avoid over-fitting when an agent uses a transferred policy from a source task. Learning robot systems have recently been studied for many applications, such as home robots, communication robots, and warehouse robots. However, if the agent reuses the knowledge that has been sufficiently learned in the source task, deadlock may occur and appropriate transfer learning may not be realized. In the previous work, a parameter called transfer rate was proposed to adjust the ratio of transfer, and its contribution include avoiding dead lock in the target task. However, adjusting the parameter depends on human intuition and experiences. Furthermore, the method for deciding transfer rate has not discussed. Therefore, an automatic method for adjusting the transfer rate is proposed in this paper using a sigmoid function. Further, computer simulations are used to evaluate the effectiveness of the proposed method to improve the environmental adaptation performance in a target task, which refers to the situation of reusing knowledge.

KEYWORDS

Reinforcement Learning, Transfer Learning, Transfer rate, Overfitting, Overlearning

1. INTRODUCTION

Machine learning systems and intelligent robot systems are increasingly being deployed to solve practical problems, such as house cleaning and conveyance systems in warehouses [1] [2]. The reinforcement learning framework has been widely discussed in applications of machine learning, such as deep Q-networks [3] [4]. Basic reinforcement learning techniques are usually time consuming. Thus, they are disadvantageous for implementation in actual applications, such as robot systems. To address this problem, transfer learning has been proposed for reinforcement learning [5-8]. Transfer learning theory allows the application of prior knowledge to another similar task. In reinforcement learning, a learning agent is used to draw and transfer policies from previous tasks (source task) to current tasks (target task). Advantages of transfer learning in reinforcement learning include: improved learning speed, fast adaptation to environments, and exploration of more effective performances. Agent systems, including transfer learning, have been successful in some cases. However, transfer learning is not successful in all applications, and in most cases, it is necessary to adjust learning conditions and avoid negative transfer situations, such as deadlock. In the previous years, adjusting the ratio of reusing policy had been proposed as a method to increase the environmental adaptation performance in the target task. The method of adjusting the transfer rate is a mechanism that determines the action value of the

DOI: 10.5121/ijaia.2020.11605

reusing policy, and then uses it in the target task [9] [10]. However, the transfer rate decision depends on human intuition and experience, and the method to determine the transfer rate has not been discussed. In addition, the transfer rate is a fixed value, and it is desirable to adjust it adaptively according to the environment and behavioral conditions. Therefore, an adjusting method of transfer rate using a sigmoid function is proposed in this paper. The proposed method automatically adjusts the transfer rate, and the method adjusting the value has to be discounted when triggered by a bad situation of learning in a target task owing to the reuse of knowledge, such as collision with an obstacle. Computer simulation is used to confirm whether the proposed method for transfer learning in reinforcement learning can adjust the transfer rate. The knowledge obtained is not reused in the case of a negative transfer situation, such as a deadlock; however, it can be reused in other cases. In particular, in recent years, transfer in reinforcement learning in multi-agent systems and human robot teams has been discussed, and it is considered essential to avoid deadlock, that is, negative transfer by transfer learning [11-13].

The remainder of this paper is organized as follows. Section 2 provides an overview of the reinforcement learning and transfer learning method, and discusses the previous method. It adjusts the transferring ratio of reusing knowledge. Section 3 discusses the adjusting method for transferring ratio with the sigmoid function as the proposed method. Section 4 presents the computer simulation experiments and results to evaluate the performance of the proposed method compared with previous methods. Section 5 presents the concluding remarks.

2. Previous Works

2.1. Reinforcement Learning

Reinforcement learning is a machine learning algorithm [3]. The reinforcement learning agent explores the optimal solution via trial-and-error and creates its own knowledge as policy. Thus, reinforcement learning does not require training datasets, unlike supervised learning. Many types of reinforcement learning algorithms have been developed in the past decades. In this study, Q-learning was adopted as the learning algorithm [14]. The Q-learning is defined by

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \left\{ r_{t+1} + \gamma \max_{a \in A} Q(s_{t+1}, a) - Q(s_t, a_t) \right\}, \tag{1}$$

where $s \in \boldsymbol{S}$ is the state of environments in a state space, $a \in \boldsymbol{A}$ is the action of agent in an action space, α is the learning rate $(0 < \alpha \leq 1)$, γ is the discount rate $(0 < \gamma \leq 1)$, and r denotes the reward. $Q(s, a)$, also known as Q-table, contains all states of the environment and each action value pair.

To select action by the agent, an action selection method was employed in previous reinforcement learning research. In this study, the Boltzmann distribution model is adopted as an action selection function [3]. The action selection probability adopted from the Boltzmann distribution is defined by

$$\mathrm{P}(a_i | s_t) = \frac{\exp\left\{ {Q(s_t, a_i)}/{T} \right\}}{\sum_{a \in A} \exp\left\{ {Q(s_t, a)}/{T} \right\}}, \tag{2}$$

where T is a parameter that determines the randomness of action selection. The following discussion is based on Watkins's Q-learning and action selection function is derived from the Boltzmann distribution model.

2.2. Policy Copy in Transfer Learning

For traditional transfer learning in reinforcement learning, Tyalor *et al.* defined a method for transferring the learned action-value function [5] [6]. This method is referred to as policy copy in transfer learning, and is defined as follows:

$$Q_t(s,a) \leftarrow Q_t(s,a) + Q_s(s,a). \tag{3}$$

Here, $Q_t(s,a)$ is the policy that includes the initial value in the target task. It is used for learning and action selection. $Q_s(s,a)$ is reusing policy from the source task. Reusing policy has state-action values obtained from the source task. In Taylor's method, the agent in the initial state of the target task sums the initial value of $Q_t(s,a)$ and transferring policy $Q_s(s,a)$. The agent is learned using the assimilated policy $Q_t(s,a)$ in the target task. Simply, the agent is reusing the transferred policy and overwriting the policy through trial-and-error simultaneously.

Originally, Taylor's method includes inter-task mapping, which defines the mapping between the agent's observable environmental state $\boldsymbol{S}$ and the executable action $\boldsymbol{A}$. Inter-task mapping is not mentioned in this paper owing to the description of simplicity.

In Taylor's method, if the transferred policy is sufficiently learned in the source task, it will behave according to the policy when used in different environments, and deadlock, such as collision with obstacles, may occur. This phenomenon is called over-fitting and over learning. The agent can overwrite the policy through behavior in the target task, and some learning time is required.

2.3. Adjusting of Transfer Ratio

Takano et al. proposed a method to adjust the action value of the transferred policy [9] by using parameters ζ in the policy assimilation equation. One of the methods in Takano's study is defined by

$$Q_c(s,a) \leftarrow \frac{1}{2}(1-\zeta)\, Q_t(s,a) + \zeta Q_s(s,a), \tag{4}$$

Where ζ is the adjusting parameter $(0 < \zeta < 1)$. $Q_c(s,a)$ is the assimilation policy, and is used for action selection. Learning in the target task is used as a policy $Q_t(s,a)$. This method discounts $Q_s(s,a)$ by ζ , and conversely uses $Q_t(s,a)$ by the amount of $1-\zeta$. Thus, it is possible to reduce the effect on the target ask even if a policy $Q_s(s,a)$with a high action value is reused. However, Takano's method requires the value of ζ to be determined before operation and depends on human intuition and experience. Furthermore, the policy $Q_t(s,a)$that should be trusted is discounted.

2.4. Transfer Rate

Kono et al. proposed the transfer rate to adjust the ratio of transferring the obtained policy [10]. The method is based on Takano's method, and it is similar and simpler than Takano's method. Transferring method with transfer rate can be defined as

$$Q_c(s,a) \leftarrow Q_t(s,a) + \tau Q_s(s,a), \tag{5}$$

where $\tau, (0 < \tau \leq 1)$ is transfer rate, and $Q_c(s,a)$ is an assimilated policy, which is used for action selection in the target task. $Q_t(s,a)$ is the policy in the target task., and is used for learning in the target task. $Q_s(s,a)$ is a reusing policy that was learned in the source task. Kono's method is also used to determine the parameters τ before the learning operation, and the value setting depends on human intuition and experience. In this paper, apart from Taylor's method, Kono's and Takano's methods are collectively called Takano's method.

3. Proposed Method

In previous study, transferring policy method overwrote transferred policy and fixed discount parameters. An automatic method for adjusting the parameters is proposed in this section. When considering the situation of negative transfer, the action of the agent that can be originally executed cannot be executed by the target task because the policy is reused, and thus, a collision with an obstacle occurs. Therefore, to avoid the above situation, if the agent becomes unable to act, the transfer rate value should be lowered. To lower the value of the transfer rate, the value is adjusted using a sigmoid function according to the number of times that the agent cannot act. The sigmoid function is defined as:

$$\frac{1}{1+e^{-\sigma g}}, \tag{6}$$

where g is the gain value, which is determined by an arbitrary value, and σ is the input value of the sigmoid function. The transfer method is reformulated using the sigmoid function as defined by

$$Q_c(s,a) \leftarrow Q_t(s,a) + \frac{1}{1+e^{-\sigma g}} Q_s(s,a), \tag{7}$$

σ is the increment or decrement with state $s_t = s_{t+1}$ and $s_t \neq s_{t+1}$, respectively. The inability of the agent to act means that the current environmental state s_t and the next environmental state s_{t+1} are the same; thus, σ is subtracted by an arbitrary constant value d. Conversely, if the action is performed by reusing the policy, only an arbitrary constant d is added for each action. This mechanism is defined as follows:

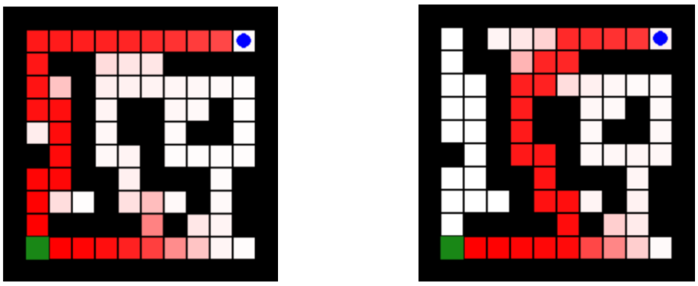
$$\sigma = \begin{cases} \sigma - d \ (s_t = s_{t+1}) \\ \sigma + d \ (s_t \neq s_{t+1}) \end{cases}. \tag{8}$$

The above function is calculated for each action, such as a step of the agent. It is possible to adjust the rate of reuse of the observes by comparing the environmental condition at each time step in a generalized form, regardless of actual situations, such as collisions with obstacles.

4. Evaluation with Computer Simulation

4.1. Experimental Setup

This experiment aims to confirm the performance of an agent's environmental adaptation using the proposed method. Furthermore, the shortest path problem is adopted in this study as the basic evaluation with a single agent. The learning environment is shown in Figure 1 in the computer simulation. Figures 1 (a) and (b) show the source task environment and target task environment, respectively. In this environment, if the agent achieves the goal, the agent can obtain a reward $r = 1$ from the environment. An agent can observe self-localization. The reinforcement learning parameter is set as $\alpha = 0.1$ and $\gamma = 0.99$. The parameter T of the Boltzmann distribution model is set as $T = 0.01$. These parameters are common to all conditions. In each experiment, 300 episodes were conducted for the source and target tasks. The agent can move to 1 grid in each step, and there are four directions that can be moved: front, right, back, and left. The time when the agent reaches the goal from the start is referred to as a single 1episode.



(a) Source task

(b) Target task.

Figure 1. Grid world of shortest path problem. Circle mark is agent and initial position in each simulation. Green square is set as goal position. In each sub-figure, red is shortest path between start position and goal position with reinforcement learning.

For experimental conditions, the learning results are compared using Taylor, Takano, and the proposed methods. In Taylor's method, reinforcement learning is executed, as shown in Figure 1 (a), and the acquired policy is transferred to the agent in Figure 1 (b) for transfer learning. In Takano's method, Kono's method was adopted for the implementation of this experiment. It is clear that Takano's method is overfitting, and it assumes a considerable amount of time to learn with the target task. Therefore, the agent is set up to obtain a negative reward $r = -1$ in the event of a collision with a wall. In addition, it was confirmed in advance that the transfer learning was possible, and the transfer rate was set to $\tau = 0.1$. In the proposed method, the parameter d is set to 0.1, and the gain g is set to 9.0.

4.2. Results

In this section, the experimental results under the three conditions are presented. The obtained learning curves are shown in Figure 2. The learning curve represents the performance related to the learning progress, whereby the horizontal axis is the number of learning episodes and the vertical axis is the number of steps required from the start to the goal. Each learning curve had 10 trial averages. In Figure 2, the learning curve of reinforcement learning that does not reuse the policy in the target task is the standard for performance evaluation. The black line in Figure 2 represents the learning curve of reinforcement learning.

4.2.1. Result of Taylor's Method

Taylor's method is believed to have a high number of steps in the early stage of learning, and it is considered that an overfitting state emerges. However, because the target task is relearned, the optimal solution, i.e., the convergence to the shortest path, appears from the learning curve of reinforcement learning.

4.2.2. Result of Takano's Method

The learning curve of Tekano has a relatively small number of steps from the early state of learning compared with the learning curve of reinforcement learning, and its convergence is considered to be equivalent to the learning curve of reinforcement learning. High performance in the early stages of the learning curve is called the jump start, and is one of the basic profits of transfer reinforcement learning. It has been confirmed that relearning in the target task is not possible when the transfer rate $\tau = 1.0$ because if the learning in the target task progresses, the value of the update is small and it is time-consuming to cancel the action value of the transferred policy.

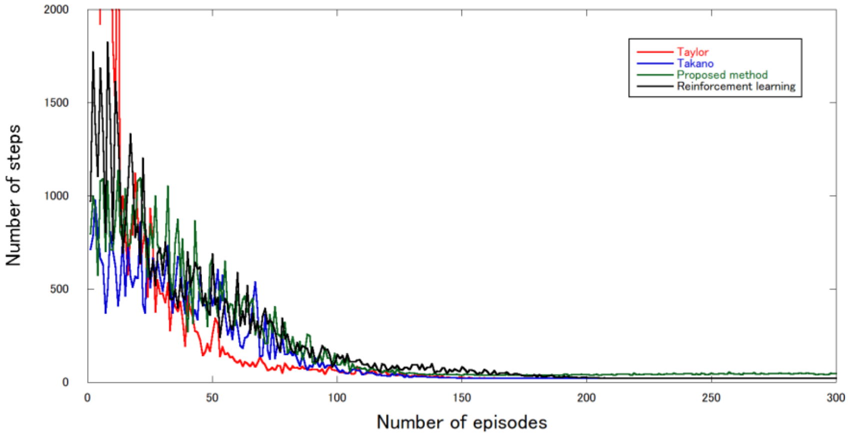


Figure 2. Learning curves

4.2.3. Result of the Proposed Method

The learning curve of the proposed method also has jumpstarted. The speed of convergence is slower than that of Takano's method and is close to the learning curve of reinforcement learning. From the figure, it can be observed that the proposed method performs better than reinforcement learning. In the initial state, the proposed method is equivalent to the transfer rate $\tau = 1.0$ of Takano's method, which normally requires time for learning. However, to adaptively change the transfer rate, the performance of the proposed method is verified by Takano's method with a transfer rate $\tau = 0.1$.

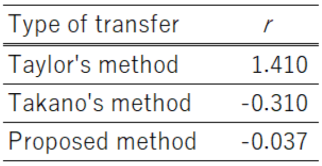
4.2.4. Discussion

The transfer ratio was used to quantitatively compare the performance. The transfer ratio can be evaluated by improving the learning time compared with the basic learning curve, such as reinforcement learning. Transfer ratio r can be defined as follows:

$$r = \frac{\sum L_t(t) - \sum L_{wt}(t)}{\sum L_{wt}(t)}, \tag{9}$$

where $L_t(t)$ is the learning curve with transfer and t is the number of episodes. Therefore, $\sum L_t(t)$is calculated as the learning curve area. In addition, $L_{wt}(t)$ is the learning curve without transfer, which indicates reinforcement learning in the target task. $\sum L_{wt}(t)$ is the calculated learning curve area of without transfer. Table 1 presents the transfer ratios in Taylor, Takano, and the proposed methods after evaluation.

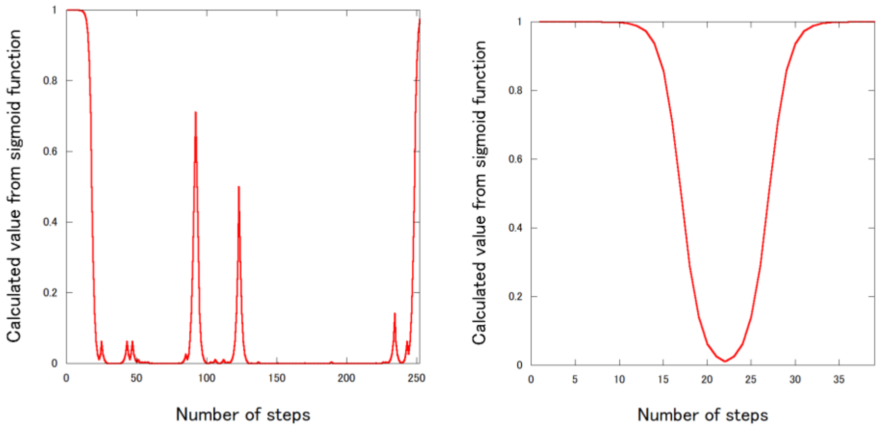
Table 1. Comparison of transfer ratio in each conditions



Type of transfer	r
Taylor's method	1.410
Takano's method	-0.310
Proposed method	-0.037

In Table 1, it can be observed that only Taylor's method increases the learning time by approximately 1.41 times. Takano's method and the proposed method decrease learning time. Figure 3 shows the transition of the output value of the sigmoid function at the beginning and end of learning. In the early stage of learning, the value corresponding to the transfer rate is adjusted by interaction with the environment. Sometimes, the transferred policy is used by the agent moving to the goal. At the end of learning, the reusing policy may not be used once. It time-consuming to use the policy at the end of learning than in the early stages of learning because the agent learns a new policy and obtains a high action value; therefore, it becomes possible to reach the goal without being affected by the reusing policy. However, if the agent obtains sufficient policy, there is no need to adjust the ratio of reusing policy. The learning curve of the proposed method does not converge because the ratio is still adjusted, even at the end of learning.

Therefore, compared to Takano's method, the result suggested that the proposed method allows the agent to interact with the environment to automatically adjust the ratio of reuse of the policy without manual adjustment or determining the transfer rate. Moreover, compared with the previous method, it was shown that the agent can improve the environmental adaptability while maintaining the characteristic of not overwriting the reusing policy from the source task.



(a) Early stage of learning (1 episode)

(b) End of learning (300 episodes)

Figure 3. Transition of the value output from the sigmoid function

5. Conclusions

This paper proposed a novel parameter for transfer reinforcement learning to avoid over-fitting in the relearning process of the target task. In the proposed method, the ratio of reusing policy from the source task is adjusted by the sigmoid function and input value, such as collision with obstacle. Basic experiments were performed with the shortest path problem with transfer reinforcement learning based on Taylor's method without inter-task mapping. The result suggests that the learning agent can be adapted to the environment. Moreover, compared with the previous method, it was shown that the agent can improve the environmental adaptability while maintaining the characteristic of not overwriting the reuse policy from the source task. This result suggests that it can be applied to reusing policy selection in future applications of reinforcement learning agents.

Our proposed method does not show convergence in the target task compared with the previous method. For future work, it is necessary to discuss the adjusting method of reusing the ratio of transferring policy at the end of learning. In the proposed method, the ratio of reuse was adjusted adaptively. The effect of the transferring policy cannot be ignored even at the end of learning.

Moreover, the proposed method is evaluated in a simple task and environment in this study. The computer simulation environment is a Markov decision process (MDP), which is different from the real-world environment. To demonstrate the effectiveness of the proposed method, it is necessary to carry out various types of more complex experiments and evaluations. In addition, the proposed method should be evaluated not only in various environments but also in non-MDP environments, such as multi-agent systems.

ACKNOWLEDGEMENTS

This study was partially supported by JSPS KAKENHI Grant number JP18K18133, and SUZUKI Foundation. This work was founded by The Japan Atomic Energy Agency (JAEA) FY2020 Center of World Intelligence Project for Nuclear Science/Technology and Human Resource Development (Grant no. R02I015).

REFERENCES

[1] B. Ramalingam, A. K. Lakshmanan, M. Ilyas, A. V. Le, and M. R. Elara (2018). “Cascaded Machine-Learning Technique for Debris Classification in Floor-Cleaning Robot Application.” Applied Sciences 8(12): 1-19.

[2] R. D. Andrea (2012). “Guest Editorial: A Revolution in the Warehouse: A Retrospective on Kiva Systems and the Grand Challenges Ahead.”IEEE Transactions on Automation Science and Engineering 9(4): 638-639.

[3] R. S. Sutton and A. G. Barto (1998). “Reinforcement learning: An introduction.” MIT press.

[4] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G.Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S.Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis (2015). “Human-level control through deep reinforcement learning.” Nature 518: 529-533.

[5] M. E. Taylor and P. Stone (2009). “Transfer learning for reinforcement learning domains: A survey.” Journal of Machine Learning Research 10(Jul): 1633-1685.

[6] M. E. Taylor (2009). “Transfer in Reinforcement Learning Domains.”Studies in Computational Intelligence 216: Springer.

[7] A. Lazaric (2012). “Transfer in Reinforcement Learning: A Framework and a Survey. Reinforcement Learning. Adaptation, Learning, and Optimization.” Berlin, Heidelberg, Springer. 12: 143-173.

[8] Q. Yang, Y. Zhang, W. Dai, and S. J. Pan (2020) “Transfer learning.”Cambridge University Press.

[9] T. Takano, H. Takase, H. Kawanaka, H. Kita, T. Hayashi and S. Tsuruoka (2011). “Transfer Learning based on Forbidden Rule Set in Actor-Critic Method.” International Journal of Innovative Computing, Information and Control 7(5(B)).

[10] H. Kono, A. Kamimura, K. Tomita, Y. Murata, and T. Suzuki (2014) “Transfer Learning Method Using Ontology for Heterogeneous Multi-agent Reinforcement Learning.” International Journal of Advanced Computer Science and Application 5(10): pp.156-164.

[11] R. Ramakrishnan, C. Zhang, and J. Shah (2017). “Perturbation Training for Human-Robot Teams.” Journal of Artificial Intelligence Research 59: pp.495-541.

[12] F. L. Da Silva and A. H. R. Costa (2019). “A Survey on Transfer Learning for Multiagent Reinforcement Learning Systems.” Journal of Artificial Intelligence Research 69: pp.645-703.

[13] F. L. Da Silva, G. Warnell, A.H.R.Costa, and P. Stone (2020). “Agents teaching agents: a survey on inter-agent transfer learning.”Autonomous Agents and Multi-Agent Systems 34 (9).

[14] C. J. C. H. Watkins and P. Dayan (1992). “Q-Learning.” Machine Learning 8: pp.279-292