Accelerate Learning Processes by Avoiding Inappropriate Rules in Transfer Learning for Actor-Critic

Toshiaki TAKANO†, Haruhiko TAKASE†, Hiroharu KAWANAKA† and Shinji TSURUOKA‡
†Graduate School of Engineering, Mie University, Japan
‡Graduate School of Regional Innovation Studies, Mie University, Japan
takano@ip.elec.mie-u.ac.jp

Abstruct—*This paper aims to accelerate processes of actor-critic method, which is one of major reinforcement learning algorithms, by a transfer learning. In general, reinforcement learning is used to solve optimization problems. Learning agents acquire a policy to accomplish the target task autonomously. To solve the problems, agents require long learning processes for trial and error. Transfer learning is one of effective methods to accelerate learning processes of machine learning algorithms. It accelerates learning processes by using prior knowledge from a policy for a source task. We propose an effective transfer learning algorithm for actor-critic method. Two basic issues for the transfer learning are method to select an effective source policy and method to reuse without negative transfer. In this paper, we mainly discuss the latter. We proposed the reuse method which based on the selection method that uses the forbidden rule set. Forbidden rule set is the set of rules that cause immediate failure of tasks. It is used to foresee similarity between a source policy and the target policy. Agents should not transfer the inappropriate rules in the selected policy. In actor-critic, a policy is constructed by two parameter sets: action preferences and state values. To avoid inappropriate rules, agents reuse only reliable action preferences and state values that imply preferred actions. We perform simple experiments to show the effectiveness of the proposed method. In conclusion, the proposed method accelerates learning processes for the target tasks.*

Keywords: *Reinforcement learning, actor-critic method, Transfer learning*

1 Introduction

Acceleration of learning processes is one of important issues in machine learning, especially reinforcement learning[1, 2]. Reinforcement leanring make agent's decision rules for its action suitable for a given environment. Since they have no information to solve a target task at the begining of learning, they should get information by trial and error. It requires long learning processes to acquire enough information. Therefore, many researchers try to accelerate learning processes[3, 4, 5].

Transfer learning[6] is one of effective methods to accelerate learning processes in some machine learning algorithms. It is based on the ideas that knowledge to solve source tasks, which are called as source policies, accelerate learning processes of a target task. Important processes in transfer learning for reinforcement learning are selection of effective source policies and reusing the selected policies, we focus on the latter.

In this paper, we aims to propose effective reuse method for selected policies which is decided by using our previous proposed method[7]. In detail, agents reuse it each parameter of reinforcement learning in a selected policy. Here, we treat actor-critic method that is one of major reinforcement learning algorithms.

2 Acceleration a Learning Process by Transfer Learning

In this section, we simply explain actor-critic method and framework of transfer learning.

2.1 Actor-critic Method

Actor-critic is one of popular reinforcement learning algorithms[1]. It finds a policy Π that maximizes the quantity R_t,

$$R_t = \sum_{\tau} \gamma^{\tau} r_{t+\tau} \tag{1}$$

for given tasks. Here, R_t is a stochastic reward function $R : S \times A \rightarrow \Re$, and γ is a predefined parameter, which called as discount rate. S is a finite set of states. A is a finite set of actions.

Actor-critic method is separated structure of actor and critic(See Fig.1). Actor decides an action according to action preferences. An action preference $p(s, a)$ is a parameter that is defined as preference of the action $a \in A$ at the state $s \in S$. Critic evaluates the action based on the reward r and state values. A state value $v(s)$ represents the inference of the state s. Each state values is modified according to a reward, and each action preference is modified according to state values, repeatedly.

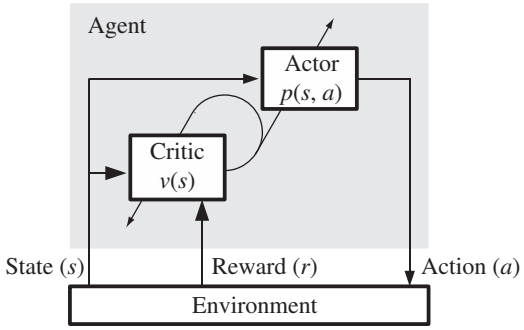

Fig. 1: Framework of actor-critic method

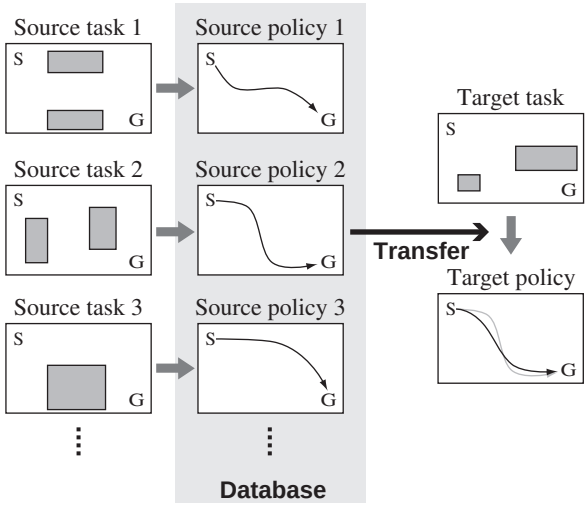

Fig. 2: Framework of Transfer Learning

2.2 Transfer Learning

In this paper, we discuss a transfer learning in actor-critic method. Figure 2 illustrates the framework of it. First, agents learn various source tasks and construct a database of policies. Second, an agent for the target task refers the database and selects a similar source policy to the optimal policy for the target task. Finally, the agent trains the target task based on the selected policy. Since the selected policy would contain effective information for the target task, the learning process of the target task would be accelerated. Transfer learning reuses a source policy which has same domain in the database to the target task. We define the domain as follows.

Definition 1. *Domain D is a tuple $< S, A >$. Task Ω is a tuple $< D, T, R >$. T is a stochastic state transition function $T : S \times A \times S \to \Re$, which is the probability that the action a in the state s_1 will lead the state s_2.*

The method accepts source tasks that have a same size of the state value table and the action preference table with ones for the target task. The domain is defined by many researchers independently. For example, Fernández defined a domain as a tuple $< S, A, T >$[8]. We intend the definition 2 to keep wide application of the proposed method.

2.3 Our Previous Work for Transfer Learning

We proposed the selection method for transfer learning in the previous work[7]. In [7], we introduced two concepts: forbidden rule set and concordance rate. The former is a set of rules that cause immediate failure of a task. The latter is defined as follows.

Definition 2. *The state s is an equivalent state, if all source fobidden rules related to the state s are agreed with ones for the target task. The concordance rate of the source forbidden rule set is a rate of equivalent states against all state.*

High concordance rate of a source forbidden rule set means that the corresponding policy is effective for the target task.

Since the complete forbidden rule set for the target task is unknown during the training phase, agents compute the concordance rate based on an incomplete forbidden rule set, which is found by the instant. They select the knowledge that has the highest concordance rate from the database, if its concordance rate is greater than the given transfer threshold θ. Here, a high threshold brings precise similarity and less transfer, and a low threshold brings opposite.

3 Proposal

In this section, we propose a reuse method in actor-critic method.

3.1 Reuse method based on the selected policy

Agents cannot completely foresee the optimal target policy by using our selection method. Therefore, the selected policy may be include inappropriate rules which cause decelerate learning process for the target task.

We discuss a method that reuse action preferences and state values instead of a policy in the form of the set of rules. Since function of each parameter is different, they should be reused in consideration of their characteristics.

Action preferences should be transferred carefully, since they are directly used to decide agent's action. Only reliable action preferences should be reused. Rules that related to an equivalent state would be reliable, since all forbidden rules are agreed. The agent merges reliable source action preferences into

current action preferences by the equation 2,

$$p_t(s,a) \leftarrow p_t(s,a) + \zeta p_s(s,a), \\ \forall s \in \text{equivalent states}, \forall a \in A. \quad (2)$$

Here, subscript t and s mean target and source, respectively. Transfer efficiency ζ is a fixed parameter that controls effects of the reused action preferences. To prevent negative transfer, the transfer efficiency is defined as $0 < \zeta < 1$.

State values can be reused aggressively. State values have less impact for the negative transfer than action preferences, since they affect agent's decision indirectly. Agents reuse only reliable action preferences, which are selected according to forbidden rules. It implies that reliable action preferences would not contain information related to preferred actions. To compensate it, preferred actions are reused with state values. Agents transfer only positive state values, because agents tend to move to states which have higher state values. They merge source state values into its state values by the equation 3,

$$v_t(s) \leftarrow v_t(s) + \eta v_s(s), \\ \forall s \in \{s | v_s(s) > 0, s \in S\}. \quad (3)$$

Here, transfer efficiency η is a fixed parameter that controls effects of reused state values. As well as transfer efficiency ζ, η is defined as $0 < \eta < 1$.

```
initialize parameters P and V.
φ → forbidden rule set F
( ) → the latest transferred item (P_p, V_p, F_p).
while( agent does not satisfy termination conditions ) {
    observe state s.
    decide action a.
    receive reword r.
    if( a is a forbidden action ) {
        add (s, a) into F.
    }
    ( ) → the most effective item (P_e, V_e, F_e).
    0 → the highest concordance rate C_e.
    foreach( (P_d, V_d, F_d) in database D ) {
        concordance rate for F_d to F → C.
        if( C > C_e ) {
            (P_d, V_d, F_d) → (P_e, V_e, F_e).
            C → C_e.
        }
    }
    if( C_e > θ && (P_e, V_e, F_e) != (P_p, V_p, F_p) ) {
        merge P_e to P according to equation (2).
        merge V_e to V according to equation (3).
        (P_e, V_e, F_e) → (P_p, V_p, F_p).
    } else {
        update P and V (actor-critic method).
    }
}
```

Fig. 3: Pseudo code to learn the target task

3.2 Whole Algorithm Flow

In this section, we show the complete transfer algorithm. In the training phase, an agent learns the target task Ω_t. It searches a policy to transfer, every time it receives a reward. It transfers the policy, if the policy is different from the last selected policy. Figure 3 shows pseudo code of this phase. We get the optimal policy from the database L, and the target task Ω_t. Here, the optimal policy is represented as the final action preferences P.

4 Experiments

In this section, we perform simple experiments to show the effectiveness of the proposed method. We perform the effectiveness of proposed method by comparing it with π-reuse[8].

4.1 Experiments Setting

We use simple maze tasks for our experiments. Each maze consists of 7×7 cells. Each cell is a coordinate or a pit. An agent moves from the start cell to the goal cell through only coordinates. The agent moves 4-way one-by-one, and decides its action by sensing its location. It repeats observation, decision, and action, every time it moves one cell. Here, the domain D is defined with $S = \{S_1, S_2, ..., S_{49}\}$ and $A = \{\text{up}, \text{down}, \text{left}, \text{right}\}$. State labels are arranged in a row major way from the left upper corner to the right bottom corner. The state S_9 is the start cell and S_{41} is the goal cell for all tasks. Rewards are defined as follows: $r = -50$ for actions to get out of coordinates, $r = 100$ for actions to reach the goal, and $r = -25$ for actions every 100th move. State transition T is defined as follows. For all moves that are same to agent's actions, transition rate is 0.9. Agents turn right against their actions by the transition rate 0.05, turn left in the same manner. They never move to opposite to agent's actions and remain stationary.

We prepare three mazes for target tasks (see figure 4) and 24 mazes for source tasks. In figure 4, white cells are coordinates, and black cells are pits. First, we prepare a database by training an agent for each source task. The database is commonly used for following experiments.

An agent finishes its learning process, when it reaches to the goal cell for ten episodes in a row. Each episode is a subsequence of the learning process while the agent moves from the start to a pit or the

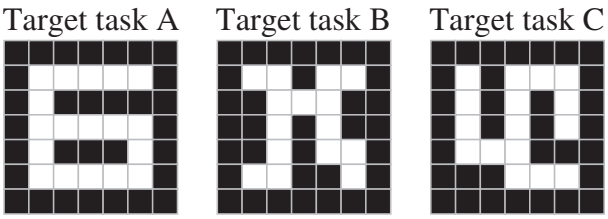

Fig. 4: Maze of target tasks

Table 1: Number of episodes for each transfer method

	Original	Proposed	π-reuse
Ω_A	250.4(38)	221.2(16)	255.9(41)
Ω_B	231.1(66)	195.7(31)	228.9(67)
Ω_C	281.1(147)	281.7(142)	271.0(149)

goal. Parameters of actor-critic method are as follows: discount rate $\gamma = 0.95$, learning rate $\alpha = 0.05$, step size parameter $\beta = 0.05$. The agent decides its action by soft-max method during its learning. The transfer threshold θ is 0.2. The fixed transfer efficiency ζ and η is 0.5, 0.05, respectively. Their experiments iterated 2000 trials.

4.2 Acceleration of Learning Processes

In this section, we discuss the effect of the proposed method. Agents learn each target task by three methods: original actor-critic method, proposed method, and π-reuse method.

In Table 1, Ω_A, Ω_B and Ω_C show the result for the target task A, B and C, respectively. Each value represents the average number of episodes, and each value in parentheses represents the number of failure of training. Grayed cells mean results that shows significant differences ($p < 0.05$) from the original method (left row).

The learning cycles of π-reuse tend to hardly differ from the learning cycles of original actor-critic method, and the learning cycles of proposed method tend to accelerate learning processes from original ones. From the result, proposed method reuses the selected policy avoiding inappropriate rules, and accelerate learning process.

5 Conclusion

In this paper, we proposed reuse method for transfer learning in actor-critic. The method allows learning agent to avoid inappropriate rules for current task. In detail, it merges action preferences and state values of the selected policy to the current parameters. We perform simple experiments to show the effectiveness of the proposed method. As the result, our proposed method accelerate learning process for the current task.

References

[1] Richard S. Sutton and Andrew G. Barto, *Reinforcement Learning*, MIT Press, Cambridge, MA, 1998.

[2] Leslie Pack Kaelbling, Michael L. Littman and Andrew W. Moore, Reinforcement Learning — A Survey, *Journal of Artificial Intelligence Research*, vol.4, pp.237–285, 1996.

[3] Marco Wiering and Jürgen Schmidhuber, Fast Online Q(λ), *Machine Learning*, vol.33, pp.105–115, 1998.

[4] Arthur Plínio de S. Braga and Aluízio F. R. Araújo, Influence zones — A strategy to enhance reinforcement learning, *Neurocomputing*, vol.70, pp.21–34, 2006.

[5] Laëtitia Matignon, Guillaume J. Laurent and Nadine Le Fort-Piat, Reward Function and Initial Values — Better Choices for Accelerated Goal-Directed, *Lecture Notes in Computer Science*, vol.4131, pp.840–849, 2006.

[6] Sinno Jialin Pan and Qiang Yang, A Survey on Transfer Learning, *Technical Report, Dept. of Computer Science and Engineering, Hong Kong Univ. of Science and Technology*, HKUST-CS08-08, 2008.

[7] Toshiaki Takano, Haruhiko Takase, Hiroharu Kawanaka, Hidehiko Kita, Terumine Hayashi, Shinji Tsuruoka: Detection of the effective knowledge for knowledge reuse in Actor-Critic, *Proceedings of the 19th Intelligent System Symposium and the 1st International Workshop on Aware Computing*, pp.624–627, 2009.

[8] Fernando Fernández and Manuela Veloso, Probabilistic Policy Reuse in a Reinforcement Learning Agent, *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, pp.720–727, 2006.