A Study on Efficient Transfer Learning for Reinforcement Learning Using Sparse Coding

Midori Saito and Ichiro Kobayashi
Advanced Sciences, Graduate School of Humanities and Sciences, Ochanomizu University, Tokyo, Japan
Email: {saito.midori, koba}@is.ocha.ac.jp

***Abstract*—By applying the knowledge previously obtained by reinforcement learning to new tasks, transfer learning has been successful in achieving efficient learning, rather than re-learning knowledge about action policies from scratch. However, in the case of applying transfer learning to reinforcement learning, it is not easy to determine which and how much the obtained knowledge should be transferred. With this background, in this study, we propose a novel method that enables to decide the knowledge and to determine the ratio of transference by adopting sparse coding in transfer learning. The transferred knowledge is represented as a linear combination of the accumulated knowledge by means of sparse coding. In the experiments, we have adopted colored mazes as tasks and confirmed that our proposed method significantly improved in terms of jumpstart and of the reduction of the total learning cost, compared with normal Q-learning.**

***Index Terms*—sparse coding, transfer learning, reinforcement learning, maze**

I. Introduction

In reinforcement learning [1], an agent explores a target environment repeatedly, performing given tasks by trial and error to obtain optimal action policies in the environment. However, the way of learning requires quite a number of random explores until satisfied action policies are obtained, so this is regarded as a big problem with the method [2]. To solve this problem, many approaches have been studied to aim to reduce the number of exploration steps. Among such approaches, it has been reported that transfer learning is especially useful and succeeds in achieving efficient reinforcement learning by reusing the previously learned action policies in similar environments in the target environment [3][4]. However, because the similarities among task environments are not clearly defined in the framework of transfer learning, the similarity is required to be calculated in response to each task [5]. If the number of the states of a task and the number of agents' actions are a quite a few, the information to be transferred shall be a huge amount, and then the calculation shall be so much complicated. With respect to this problem, in this paper, we employ sparse coding to calculate the similarities between the environments of a source task and a target task, and determine which and how much action policies are transferred. By this, we aim to make possible to efficiently transfer action policies to new target tasks.

Manuscript received June 11, 2015; revised October 2, 2015.

II. Related Studies

This section presents the related studies to our study, in particular, the ones focus on transfer learning employed in reinforcement learning and also sparse coding employed in transfer learning. First, as for the transfer learning employed in reinforcement learning, it has been successful in generalizing information across multiple tasks. Even though between two different tasks, transferring the knowledge an agent has learned in a source task is useful in a target task [6]. Fernando *et al.* [3] employed Policy Reuse as a technique to improve transfer efficiency in reinforcement learning by reusing similar policies leaned in a past. The technique improves its exploration in target environments by probabilistically including the exploitation of those past policies. Then, they succeeded to improve the learning performance over different strategies with policy reuse, and contributed policy reuse as transfer learning among different domains [4]. Trung *et al.* [7] proposed a method to transfer old knowledge, and evaluated new options to see if they worked better. They succeeded in achieving efficient and online transfer, and improved jumpstart and faster convergence to near optimum policy. Furthermore, they proposed model-based reinforcement learning that supports efficient online-learning of the relevant features and introduced an online sparse coding learning technique for feature selection in high-dimensional spaces [8]. Then, they demonstrated practicality of their proposed model in both simulated and real robotics domains.

Next, as for sparse coding, sparse coding was originally developed to achieve a good result in the field of signal processing, i.e., audio and natural images. It can approximately represent the input signal expressed by a vector with the linear combination of a few bases of the vector well [9]. For instance, in the field of image processing, Kai *et al*. [10] presented a method for earning image representations using a two-layer sparse coding scheme. The algorithm provided excellent results for hand-written digit recognition and object recognition, and achieved to automatically learn the features of the target. In this research, they proposed an approach that accounts for high-order dependency among patterns in a local image neighborhood. Then, as an example of applying

doi: 10.18178/joace.4.4.324-330

sparse coding to transfer learning, Haithman *et al.* [5][11] proposed a novel transfer learning for reinforcement learning method capable of autonomously creating an inter-task mapping by using a novel combination of sparse coding. They succeeded to show not only transfer of information between similar tasks, but also between two very different domains in their experiments. Then, they illustrated that the learned inter-task mapping can be successfully used to improve the performance of a learned policy, reduce the learning times, and converge faster to a near-optimal policy.

In this study, we propose a novel online transfer method making use of sparse coding in reinforcement learning, based on those state-of-the-art researches mentioned above.

III. Reinforcement Learning and Transfer Learning

A. Reinforcement Learning

The reinforcement learning [1] is a machine learning method to obtain optimal action policies by making an agent repeatedly search in a target environment. In concrete, the learning process is shown as the following three steps:

1. An agent observes the states of an environment.
2. An agent selects and performs an action among the possible actions in the current environment.
3. The action performed at an environment is evaluated by being given reward or a penalty.

The reinforcement learning is defined as a Markov Decision Processes (MDPs), and its state is represented as a tuple, $<S, A, P, R>$. Here, S is a set of states; A is a set of actions; P is the transition probability expressed as $P=Pr\{s_{t+1}=s' \mid s_t=s, a_t=a\}$; and R is reward given to an agent from the environment. An agent's action policy is expressed as $\pi(s, a)=Pr\{a_t=a \mid s_t=s\}$. The reinforcement learning aims to acquire the optimal action policies that maximize the total expectation value of the reward given from an environment as in (1).

$$V^{\pi}(s) = E_{\pi}\{R_t \mid s_t = s\} = E_{\pi}\{\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \mid s_t = s\} \quad (1)$$

here, $V^{\pi}(s)$is called state-value function for policyπ. γ *indic*ates the discount ratio.

B. Q-learning

We employ Q-learning [14] as a reinforcement learning algorithm. It is a kind of Temporal Differential learning, and aims to maximize the evaluation value of actions, called Q-value. The equation for updating Q-value is shown in (2).

$$Q(s_t,a_t) = Q(s_t,a_t) + \alpha(r + \gamma \max_a Q(s_{t+1},a) - Q(s_t,a_t)) \quad (2)$$

here, $Q(s,a)=E[R|s_t=s,a_t=a]$, called action-value function, which expresses the value obtained from the action a at the state s. αindicates the learning ratio and γ indicates the discount ratio. In this study, we employ ε-greedy algorithm in deciding agent's action. In the action selection by ε-greedy algorithm, the actions are randomly selected with the probability of ε, and are selected so as maximum Q-value with the probability of $1-\varepsilon$.

C. Transfer Learning

The transfer learning employed in the framework for reinforcement learning aims to reduce the number of random exploration in a new task. First, we obtain knowledge as action policies or Q-values by executing reinforcement learning in a source task, and then apply the obtained knowledge to a target task. Thereby, the efficiency of learning the target task will be improved, even if the target task is not the same as the source task. However, if the environment of the target task is considerably different from that of the source task, the transferred knowledge is useless at the target task. So, we should correctly choose the knowledge to be transferred in accordance with the situation of the target task. Therefore, we introduce sparse coding into transfer learning.

IV. Sparse Coding

A. Sparse Coding

In this study, we propose a method to realize efficient transfer learning in the framework for reinforcement learning by introducing sparse coding [13]. Sparse coding is a method of the signal processing. It selects the number of the basis vectors as small as possible for a signal, and expresses input signal with linear combination of the basis vectors. We show an equation representing the idea of sparse coding below.

$$y = Dx \quad (3)$$

here, y indicates an input signal, D is a set of basis vectors called dictionary, and x is an activation matrix which indicates a set of coefficient corresponding to each of the base. Like this, in sparse coding, y is represented in two matrices, D and x. The objective function to be optimized for sparse coding is defined in (4).

$$x^* = \min_x \frac{1}{2}\|y - Dx\|_2^2 + \lambda\|x\|_1 \quad (4)$$

here, the first right term of (4) indicates the term for minimizing the square error between the original information y and the restored information Dx, and the second term is for regularization which provides a constraint on deriving x in a sparse condition. λ is the parameter for regularization. By (4), we obtain an optimal sparse activation matrix x.

B. Sparse Coding for Transfer Learning

Now we explain an overview of how to apply sparse coding to transfer learning. In the case of reusing the knowledge, i.e., action policies, obtained by reinforcement learning in a new target task environment, the accuracy of transfer learning will be considerably different depending on which pieces of knowledge are chosen from a huge quantity of knowledge. Moreover, only a part of the whole accumulated knowledge is used to search an appropriate action policy for a target task in short term, though a huge quantity of knowledge is

necessary for the whole transfer. Therefore, it is thought that sparse coding works well in transfer learning by regarding one piece of knowledge obtained in a source task environment as one basic vector of dictionary matrix, namely, all accumulated knowledge is regarded as a set of basis vectors. Then, by regarding the state *s* observed in a target task as input vector, it can be represented with linear combination of knowledge in the dictionary matrix.

C. Proposed Method

As seen in Fig. 1, at first, in the target task, the costs of the target cell and its surrounding 5-5 square cells are observed and compiled as an input vector (step 1). Secondary, sparse coding is applied to the input vector with the dictionary built by source task, and an activation matrix is calculated (step 2). Then, the Q-value obtained in the source tasks which corresponds to the index of the non-zero elements of the activation matrix is multiplied with the value of the non-zero elements of the activation matrix (step 3). Lastly, the Q-value calculated in step 3 is returned to the current target cell (step4). In this way, in the target task, by repeating the above 4 steps and sequentially providing the Q-value obtained in the source tasks as the value of the target cell, an efficient transfer learning is achieved. The detail of this our proposed method is described in section V.

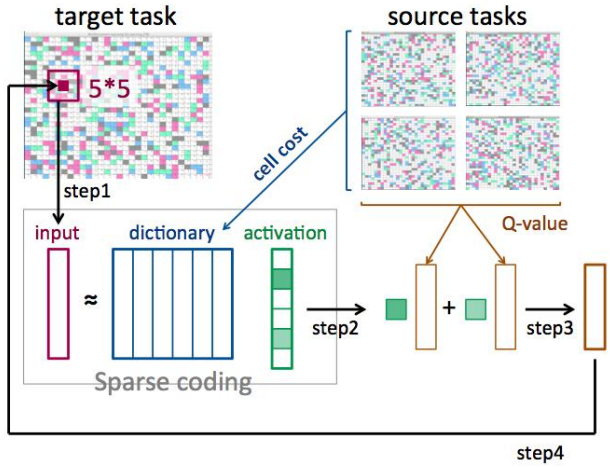

Figure 1. Overview of proposed method.

V. Experiments

A. Source Task

At first, in order to construct a dictionary by means of sparse cording, we prepared 4 different source tasks (see, Fig. 2).We adopted colored mazes as tasks in [14], these mazes have 5 different cell colors in themselves, and the size of all 4 source tasks are the same, the height is 30 cells and the width is 30 cells. Each color has its own cost: i.e., white: 0.0, blue : -2.0, green : -3.0, red : -5.0, and black : -10.0. We regard the cell costs as the reward, and +100 is given when an agent arrives at goal. In each cell, the agent can select an action among 4 actions, i.e., moving 1 cell in either up, down, right, or left direction. The task we adopt in our experiments is moving to the goal at the right corner of the bottom of the maze from the start point at the left corner of the top of the maze. As for the parameter settings for Q-learning, α is set as 0.1 and γ is set as 0.9. Moreover, as the algorithm of selecting actions, we employ ε-greedy algorithm in which it takes random action with 20% probability and the action based on maximum Q-value with 80% probability, aiming that the agent can learn action policies to find the optimal route to the goal with total cell costs and total steps as low as possible. With these settings, the agent executed Q-learning and learned the optimal route to the goal. We took reinforcement learning for 1000000 episodes repeatedly on each maze, and after 1000000 episodes simulation, we recorded the information about cost and Q-value of each cell as the knowledge to be used for transfer learning.

B. Construction of a Dictionay Matrix

As mentioned in V-A, a dictionary matrix is constructed based on the information obtained from the result of 1000000 episodes Q-learning.

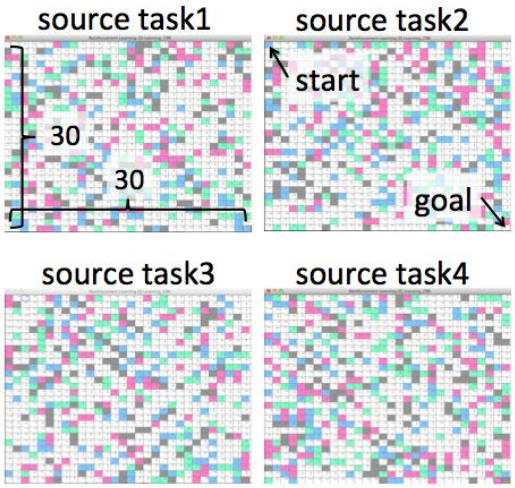

Figure 2. 4 different source tasks.

We regard the information in 5-5 square cell costs of the source tasks as a basis vector in a dictionary and extract possible information in 5-5 square cells from 30-30 square cells. By this, the number of basis vectors in each task is 676 (=26*26) and then becomes 2704 in total (=676*4 tasks). So, the size of the dictionary matrix becomes 25*2704. The image of how to make a dictionary is shown in Fig. 3.

C. Target Task

As for target tasks, we use the same size square colored mazes as the source tasks. As well as the source tasks, the start point is at the left corner of the top of the maze and the goal point is at the right corner of the bottom of the maze (Fig. 4). As explained at step 1 in IV-C, the agent obtained the cell cost data of the currently exploring cell and its 5-5 square surroundings, and regarded as an input vector. Then, sparse coding chose some environment in 4 source tasks (the bases of the dictionary shown in V-B) that are similar to the target 5-5 square cell cost information, and calculated how similar those chosen environment (the results of the activations). Next, we extracted the index of non-zero activations from

the activation matrix, and the 5-5 cells' Q-value data (gained in V-A) corresponding to that index, is multiplied by the corresponding activation values, and lastly took these linear combination. In that calculated Q-value data, the Q-value data for 4 action policies: up, down, right and left, corresponding to the middle cell of 5-5 square was extracted, and it returned to the target cell as its Q-value. While exploring the target task, if it is the first time for the agent to visit the cell, because there is no prior knowledge about a proper action at the cell, Q-value is transferred through the execution of sparse coding. On the other hand, if the agent visits the cell more than twice, normal Q-learning is applied to update the Q-value of the cell because there already exists prior knowledge about actions.

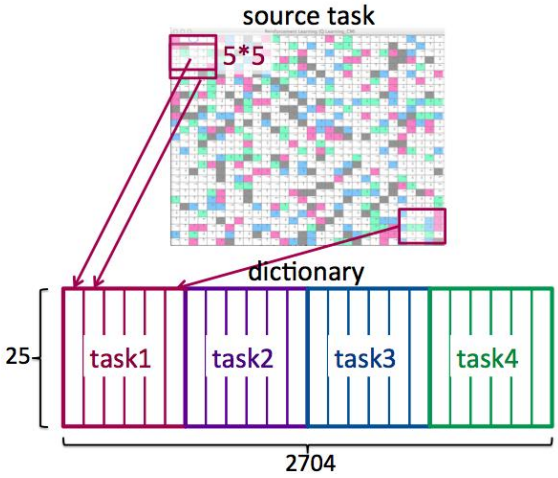

Figure 3. How to construct a dictionary matrix.

D. Experimental Settings

In this paper, we examined our proposed method with 5 different target tasks (see, Fig. 4). In order to show the effectiveness of our proposed method, we employed a normal Q-learning as a baseline method to compare. The Q-learning parameters of the proposed method and the base line method are α=0.1,γ=0.9, and ε =0.2 of ε-greedy. We set the total value of the elements in the activation matrix to be 10.0 for a sparse coding setting. These parameter settings are empirically decided. Fig. 5 shows the relation between bases in the dictionary matrix and the elements in the activation matrix at a particular input cell when the total value of the elements of the activation matrix changes in the range of [0,100].For example, when the total activation value gets 40 in horizontal axis, we see that 3 bases in the dictionary are used to represent the input and the sum of the activation values of those 3 bases gets 40. Fig. 6 shows the result that activation value when the total value of activations changed to 1, 10, and 50. As seen from this figure, a few bases contain non-zero value in their activation, other bases have 0 activation value by sparse coding. Through these experiments, we evaluated 2 points: how the proposed method could reduce the total cell cost and supported jumpstart which means the reduction of number of steps necessary to reach the goal.

Figure 4. 5 different target tasks.

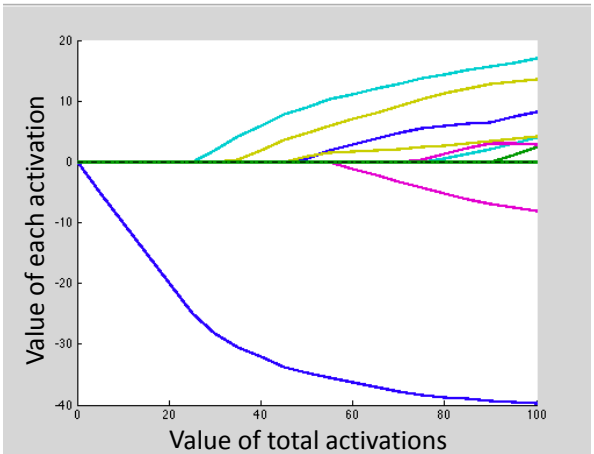

Figure 5. The trace of activations.

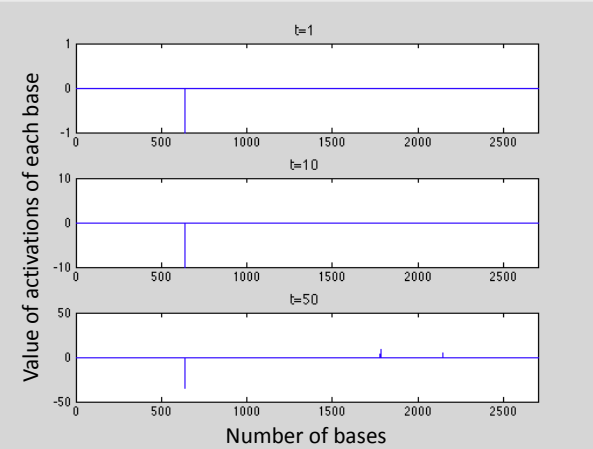

Figure 6. The value of activations.

E. Result

Fig. 7-Fig. 11 represent the experimental results of the number of total steps and the values of total cell costs obtained in 5 target tasks. In the experiments, each task is repeatedly performed 100 times. In all the figures, the red lines indicate the proposed method and the blue lines are normal Q-learning. And these figures show the number of episodes in the horizontal-axis and the number of total steps or the value of total cell costs in the vertical-axis.

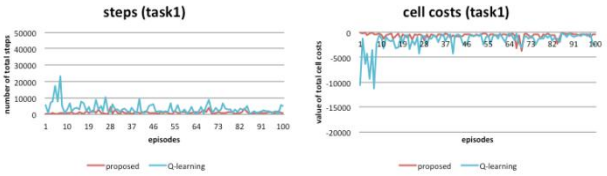

Figure 7. The result of target task 1.

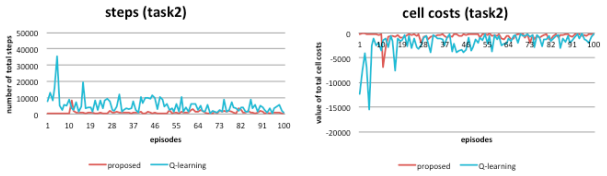

Figure 8. The result of target task 2.

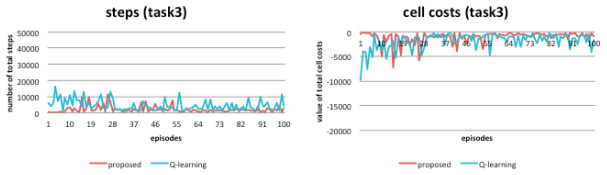

Figure 9. The result of target task 3.

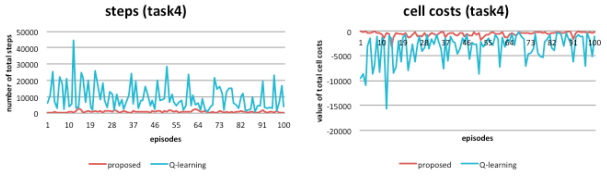

Figure 10. The result of target task 4.

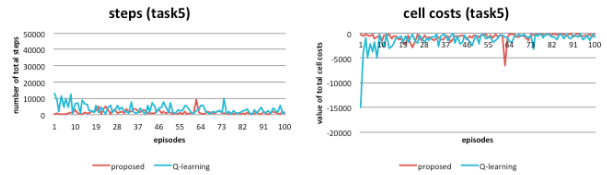

Figure 11. The result of target task 5.

F. Discussions

Table I shows the number of total steps and total cell costs when the agent first arrived at the goal in 5 target tasks. As this table shows, the proposed method could decrease the number of necessary steps taken to reach the goal. We see clearly from the results that the jumpstart was achieved. We also see that the total costs decreased.

On the whole, the proposed method, which transfers the Q-values obtained in the source tasks to the target task, was more efficiently than the normal Q-learning that learns Q-value of the target task from scratch. However, as seen in Fig. 7-Fig. 11, there are some points where the proposed method did not work well than the normal Q-learning, especially in task 3 and 5. As the reason for this, it can be thought that the negative transfer has arisen at some cells in the target task and then more steps were necessary to get recovered from the problem. That also led the value of total cell cost to be much worse. Depending on task environment, there is some difference among the effect of transfer via sparse coding. We have verified the assumption that negative transfer happened in the target tasks. As clear examples, we focus on the 61st episode that took the largest number of steps to reach the goal and the 2nd episode that took the smallest number of steps in target task 5. Fig. 12 shows how many times the agent arrived at each cell of target task 5 in the 61st episode. In the figure, the horizontal-axis is the x-coordinate of the target task5, the depth-axis is the y-coordinate and vertical-axis is the number of steps at each cell. We see that the top left corner of the x-y plane is the start point and the front of bottom right corner is the goal point. As this graph shows, there are many useless steps observed at around the bottom left. Likewise, it is observed that the number of steps increased around the bottom left in 2nd episode, though the total number of steps is considerably lower than that of 61stepisode (see, Fig. 13).

TABLE I. THE RESULT OF THE FIRST EPISODE

		Task1	Task2	Task3	Task4	Task5
steps	**proposed**	**208**	**196**	**480**	**166**	**506**
	Q-learinig	5728	7878	5910	5972	13238
costs	**proposed**	**-125**	**-196**	**-476**	**-157**	**-208**
	Q-learning	-10522	-12391	-9783	-9516	-15016

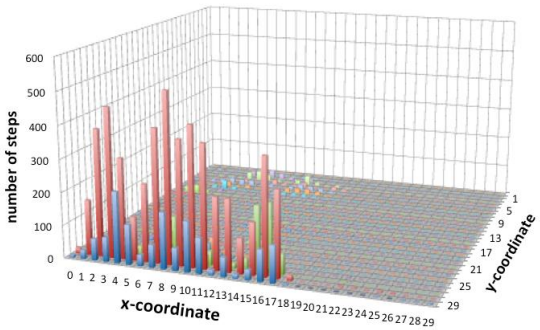

Figure 12. Number of steps in 61st Episode of target task 5.

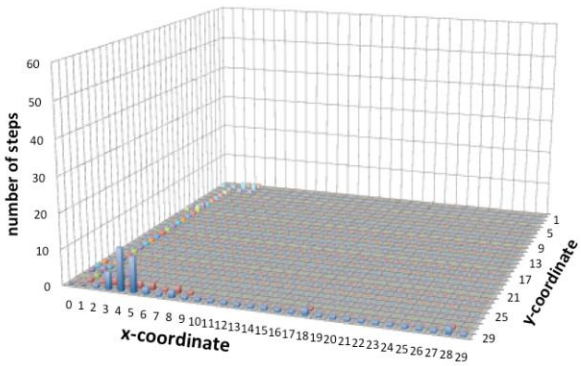

Figure 13. Number of steps in 2nd Episode of target task5

Among the steps each cell took, the number of steps of the cell at (8, 28) in 61st episode took 516 steps, the largest steps in the episode. Therefore, we investigated which piece of knowledge was transferred to the cell at (8,

28). Fig. 14 shows knowledge transfer happened in the cell at (8, 28) of target task 5. The left 5-5 square cell of Fig. 14 indicates the cell cost, obtained when the agent steps into the cell at (8, 28), which is used as input information for sparse coding. In the case of the 5-5 square, we treat the cost of the bottom row as -100, because of the out of search range. Depending on the input information, sparse coding is executed. As the result of execution of sparse coding, we confirmed that 100% of the Q-value of the cell at (23, 24) in source task 3 was transferred to the target cell. However, even though compared the left target circumstance to the right source circumstance, there is less similar between the two environments. It can therefore be thought that the transfer was influenced by the cost at the bottom row of the target task. In addition, Fig. 15 shows a result of the case where the input information for sparse coding does not include the cost at the bottom row. Here, the figure shows the case where knowledge is transferred to the cell at (10,6) of the target task 5. As a result of this case, there were several bases employed as the knowledge to be transferred. Among the bases, the base of the cell at (23, 14) in the source task 1 had the largest transfer ratio, which was about 55.8%. In this example, we see that the transferred Q-value was successfully represented as linear combination of the Q-values of several bases. To consider the difference between these two transfers, because there were lack of the bases which correspond to the out of range, that led transfer accuracy to get worse. In this study, we think if the bases including the out of range cost, the influence by those bases might cause imbalance knowledge transfer, so, we did not take those bases in the dictionary. Thereby, we think the reason why knowledge transfer did not work well at some points as the dictionary was not well prepared.

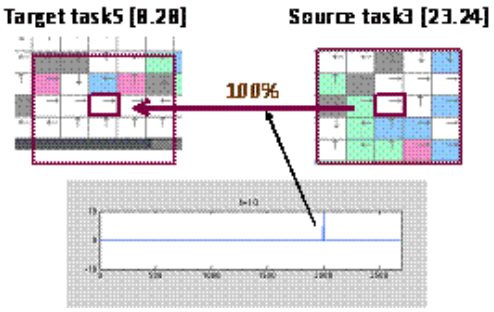

Figure 14. Bad transfer.

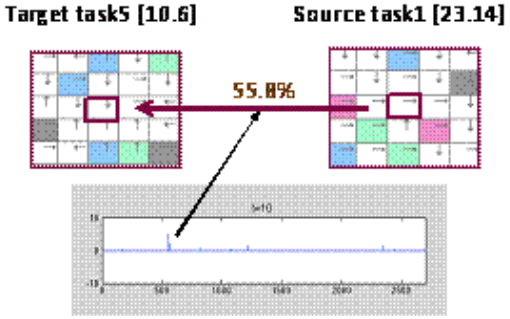

Figure 15. Good transfer.

VI. Conclusions

In this study, we have proposed a novel technique to employ sparse coding in transfer learning for reinforcement learning. By applying the knowledge to the bases of sparse coding dictionary matrix, we succeeded to choose the correct knowledge that should be transferred to new environment target tasks. We experimented the proposed method on the 5 target tasks and confirmed that the method provided jumpstart and reduced the total cell costs. By the result of these experiments, our proposed method was able to contribute significant improvement on the effective online transfer learning for reinforcement learning. However, as mentioned in section VII-E, negative transfer happened on some cells of the target task.

For future work, to solve this problem we would like to reconsider the number of bases and the quality of the dictionary, and then aim to improve the accuracy of transfer learning. In concrete, we will increase the number of source tasks to obtain a larger dictionary and obtain a large number of high diversity bases in the dictionary. On the other hand, we also have to take the processing time into account. It will become unrealistic if the number of bases becomes enormous. So, in practice we have to limit the maximum number of bases. In this point, to deal with this problem, we are going to take the methods employed by prior studies such as [9] in our study and aim to construct an ideal dictionary within the upper limit of the number of bases.

References

[1] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, The MIT Press, 1998.
[2] T. Takano, H. Takase, H. Kawanaka, and S. Tsuruoka, "A study on selection method of transfer knowledge in same transition model for reinforcement learning (in Japanese)," in *Proc. 27th Fuzzy System Symposium*, MB1-3, 2011.
[3] F. Fernandez and M. Veloso, "Probabilistic policy reuse in a reinforcement learning agent," *AAMAS'06*, May 8–12, 2006.
[4] F. Fernández, J. García, and M. Veloso, "Probabilistic policy reuse for inter-task transfer learning," *Robotics and Autonomous Systems 58*, pp. 866–871, 2010.
[5] H. B. Ammar, K. Tuyls, M. E. Taylor, K. Driessens, and G. Weiss, "Reinforcement learning transfer via sparse coding," in *Proc. the 11th International Conference on Autonomous Agents and Multiagent Systems*, 2012, pp. 4-8.
[6] M. E. Taylor and P. Stone, "An introduction to inter-task transfer for reinforcement learning," *Association for the Advancement of Artificial Intelligence*, 2011.
[7] T. T. Nguyen, T. Silander, and T. Y. Leong, "Transferring expectations in model-based reinforcement learning," *NIPS*, 2012.
[8] T. T. Nguyen, Z. Li, T. Silander, and T. Y. Leong, "Online feature selection for model-based reinforcement learning," *ICML*, 2013.
[9] I. Ramirez, P. Sprechmann, and G. Sapiro, "Classification and clustering via dictionary learning with structured incoherence and shared features," *IEEE*, 2010.
[10] K. Yu, Y. Q. Lin, and J. Lafferty, "Learning image representations from the pixel level via hierarchical sparse coding," in *Proc. 2011 IEEE Conference on Computer Vision and Pattern Recognition*, 2011, pp. 1713-1720.
[11] H. B. Ammar, M. E. Taylor, K. Tuyls, and G. Weiss, "Reinforcement learning transfer using a sparse coded inter-task mapping," *EUMAS, Lecture Notes in Computer Science*, Springer, vol. 7541, pp. 1-16, 2011.

[12] B. A. Olshausen and D. J. Field, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," *Nature*, vol. 381, pp. 607-609, 1996.
[13] A. Wilson, A. Fern, and P. Tadepalli, "Transfer learning in sequential decision problems: a hierarchical bayesian approach," *JMLR: Workshop and Conference Proceedings*, vol. 27, pp. 217–227, 2012.
[14] C. J. C. H. Watkins, "Learning from delayed rewards," PhD thesis, King's College, Cambridge, UK, 1989.

Midori Saito (Aichi, Mar. 16th, 1991), Master student at Advanced Sciences, Graduated School of Humanities and Sciences, Ochanomizu University. She graduated from Dept. of Information Sciences, Faculty of Sciences, Ochanomizu University in 2013.

Ichiro Kobayashi (Tokyo, Aug. 2nd, 1965), 2011- Professor, Advanced Sciences, Graduated School of Humanities and Sciences, Ochanomizu University. 2003-2010 Associate Professor, Advanced Sciences, Graduated School of Humanities and Sciences, Ochanomizu University.
1996-2003 Associate Professor, Faculty of Economics, Hosei University. 1995 Assistant Professor, Faculty of Economics, Hosei University.