Journal of Computer and Communications, 2014, 2, 210-219
Published Online March 2014 in SciRes. http://www.scirp.org/journal/jcc
http://dx.doi.org/10.4236/jcc.2014.24028

Scientific Research

A Policy-Improving System for Adaptability to Dynamic Environments Using Mixture Probability and Clustering Distribution

Uthai Phommasak[1], Daisuke Kitakoshi[2], Jun Mao[1], Hiroyuki Shioya[1]

[1]Division of Information and Electronic Engineering, Graduate School of Engineering, Muroran Institute of Technology, Hokkaido, Japan
[2]Department of Information Engineering, Tokyo National College of Technology, Tokyo, Japan
Email: s1823204@mmm.muroran-it.ac.jp

Received October 2013

Abstract

Along with the increasing need for rescue robots in disasters such as earthquakes and tsunami, there is an urgent need to develop robotics software for learning and adapting to any environment. A reinforcement learning (RL) system that improves agents' policies for dynamic environments by using a mixture model of Bayesian networks has been proposed, and is effective in quickly adapting to a changing environment. However, the increase in computational complexity requires the use of a high-performance computer for simulated experiments and in the case of limited calculation resources, it becomes necessary to control the computational complexity. In this study, we used an RL profit-sharing method for the agent to learn its policy, and introduced a mixture probability into the RL system to recognize changes in the environment and appropriately improve the agent's policy to adjust to a changing environment. We also introduced a clustering distribution that enables a smaller, suitable selection, while maintaining a variety of mixture probability elements in order to reduce the computational complexity and simultaneously maintain the system's performance. Using our proposed system, the agent successfully learned the policy and efficiently adjusted to the changing environment. Finally, control of the computational complexity was effective, and the decline in effectiveness of the policy improvement was controlled by using our proposed system.

Keywords

Reinforcement Learning; Profit-Sharing Method; Mixture Probability; Clustering

1. Introduction

Reinforcement learning (RL) is an area of machine learning within the computer science domain, and many RL methods have recently been proposed and applied to a variety of problems [1-3], where agents learn the policies to maximize the total number of rewards decided according to specific rules. In the process whereby agents obtain rewards, data consisting of state-action pairs is generated. The agents' policies are effectively improved by a

How to cite this paper: Phommasak, U., *et al*. (2014) A Policy-Improving System for Adaptability to Dynamic Environments Using Mixture Probability and Clustering Distribution. *Journal of Computer and Communications*, **2**, 210-219.
http://dx.doi.org/10.4236/jcc.2014.24028

supervised learning mechanism using the sequential expression of the stored data series and rewards.

Normally, RL agents initialize the policies when they are placed in a new environment and the learning process starts afresh each time. Effective adjustment to an unknown environment becomes possible by using statistical methods, such as a Bayesian network model [4,5], mixture probability and clustering distribution [6], etc., which consist of observational data on multiple environments that the agents have learned in the past [7,8]. However, the use of a mixture model of Bayesian networks increases the system's calculation time. Also, when there are limited processing resources, it becomes necessary to control the computational complexity. On the other hand, by using mixture probability and clustering distribution, even though the computational complexity was controlled and the system's performance simultaneously maintained, the experiments were only conducted on fixed obstacle environments. Therefore, examination of the computational complexity load and the adaptation performance in dynamic and 3D environments is required.

In this paper, we describe a mixture probability consisting of the integration of observational data on environments that an agent learned in the past within the framework of RL, which provides initial knowledge to the agent and enables efficient adjustment to a changing environment. We also describe a novel clustering method that makes it possible to select fewer mixture probability elements for a significant reduction in the computational complexity while retaining the system's performance.

The paper is organized as follows. Section 2 briefly explains the profit-sharing method, the mixture probability, the clustering distribution, and the flow system. The experimental setup and procedure as well as the presentation of results are described in Section 3. Finally, Section 4 summarizes the key points and mentions our future work.

2. Preparation

The RL method makes it possible for agents to learn new behaviors for profit-sharing, mixture probability, and clustering, which are the three principal components of the proposed system. This section describes the three principal components and the flow system.

2.1. Profit-Sharing

Profit-sharing is an RL method that is used as a policy learning mechanism in our proposed system. RL agents learn their own policies through "rewards" received from an environment.

The policy is given by the following function:

$$w: S \times A \to R \tag{1}$$

where S and Adenote a set of state and action, respectively. Pair $(s,a)(\forall s \in S, \forall a \in A)$ is referred to as a rule. $w(s,a)$ is used as the weight of the rule ($w(s,a)$ is positive in this paper).When state s^0 is observed, a rule is selected in proportion to the weight of rule $w(s^0,a^0)$. The agent selects a single rule corresponding to given state s^0 using the following probability:

$$P(s^0,a^0) = \frac{w(s^0,a^0)}{\sum_{s' \in S, a' \in A} w(s',a')} \tag{2}$$

The agent stores the sequence of all rules that were selected until the agent reaches the target as an episode.

$$L = \{(s_1,a_1),\ldots,(s_L,a_L)\} \tag{3}$$

where L is the length of the episode. When the agent selects rule (s_L,a_L) and requires reward r, the weight of each rule in the episode is reinforced as in **Figure 1** by

$$w(s_i,a_i) \leftarrow w(s_i,a_i) + r\gamma^{L-i} \tag{4}$$

where $\gamma(\in (0,1])$ is the "learning rate". In this paper, the following nonfixed reward is used:

$$r = r_0 + (t-n) \tag{5}$$

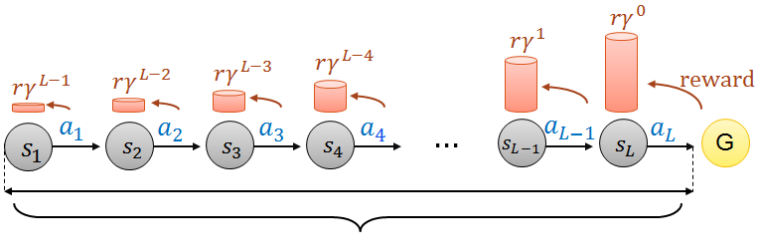

Figure 1. Reward sharing.

where r_0 is the initial reward, t is the action number limit in one trial and n is the real action number until the agent reaches the target. We expect that the agent can choose a more suitable rule to reach the target in a dynamic environment by using this nonfixed reward.

2.2. Mixture Probability

Mixture probability is a mechanism for recognizing changes in the environment and consequently improving the agent's policy to adjust to those changes.

The joint distribution [9] $P(s,a)$, consisting of the episode observed while learning an agent's policy, is probabilistic knowledge about the environment. Furthermore, the policy acquired by the agent is improved by using the mixture probability of $P_i(i=1,\ldots,m)$ obtained in multiple known environments. The mixing distribution is given by the following function:

$$P_{mix}(s,a)=\sum_{i=1}^{m}\beta_i P_i(s,a) \tag{6}$$

where m denotes the number of joint distributions, and β_i is the mixing parameter $\left(\sum_i \beta_i = 1, \beta_i \geq 0\right)$. By adjusting the environment subject to this mixing parameter, we expect appropriate improvement of the policy on the unknown dynamic environment.

In this paper, we use the following Hellinger distance [10] function to fix the mixing parameter:

$$D_H(P_i,Q)=\left\{\sum_x\left[P_i(x)^{\frac{1}{2}}-Q(x)^{\frac{1}{2}}\right]^2\right\}^{\frac{1}{2}} \tag{7}$$

where D_H is the distance between P_i and Q, and D_H is set to 0 when P_i and Q are the same. P_i is joint distributions obtained in m different environments that an agent has learned in the past, Q is the sample distribution obtained from the successful trial of τ times in an unknown environment, and x is the total number of rules. Given that $D_H(P_i,Q)\leq\sqrt{2}$ is established, the mixing parameter can be fixed by the following function:

$$\beta_i=\frac{\sqrt{2}-D_H(P_i,Q)}{\sum_{j=1}^{m}\left[\sqrt{2}-D_H(P_j,Q)\right]} \tag{8}$$

However, when $\sum_{j=1}^{m}\left[\sqrt{2}-D_H(P_j,Q)\right]=0$, $\beta_i=\frac{1}{m}$, and when all distributions are equal, the mixing parameter is evenly allotted.

2.3. Clustering Distributions

In this study, we used the group average method as opposed to the clustering method. The distance between the clusters can be determined by the following function:

$$D\left(Cl_i, Cl_j\right) = \frac{1}{n_i n_j} \sum_{P_i \in Cl_i, P_j \in Cl_j} D_H\left(P_i, P_j\right) \tag{9}$$

where $n_i; n_j$ are the number of joint distributions contained in Cl_i and Cl_j, respectively. In this study, we used the Hellinger distance function $D_H\left(P_i, P_j\right)$. After completing the clustering, element P_i having the minimum $D_H\left(P_i, Q\right)$ will be selected as the mixture probability element from each cluster.

As shown in **Figure 2**, we expect that the computational complexity of the system can be controlled and it will be possible to maintain the effectiveness of policy learning by selecting only the suitable joint distributions as the mixture probability elements based on this clustering method.

2.4. Flow System

The system framework is shown in **Figure 3**. A case involving the application of mixture probability to improve the agent's policy is explained in the following procedure:

Step 1 Learn the policy in m environments by using the profit-sharing method to make the joint distributions $P_i = \left(i = 1, \ldots, m\right)$

Step 2 Cluster$m$ distributions into n clusters

Step 3 Calculate the Hellinger distance D_H of distributions P_i and sample distribution Q

Step 4 Select the element having the minimum $D_H\left(P_i, Q\right)$ from each cluster

Step 5 Calculate the mixing parameter β_i

Step 6 Calculate the mixture probability P_{mix}

Step 7 Update the weight of all rules by using the following function:

$$w\left(s,a\right)^{new} \rightarrow w\left(s,a\right)^{old} + w\left(s,a\right)^{old} \times P_{mix}\left(s,a\right)$$

And then continue learning the updated weight by using the profit-sharing method.

3. Experiment

We performed an experiment to demonstrate the agent navigation problem and to illustrate the applied improvement in the RL agent's policy through the modification of parameters of the profit-sharing method and using the mixture probability scheme. The purpose of this experiment was to evaluate the adjustment performance in the unknown dynamic environment by applying the policy improvement, and to evaluate its effectiveness by using mixture probability.

3.1. Experimental Setup

The aim in the agent navigation problem is to arrive at the target from the default position of the environment where the agent is placed. In the experiment, the reward is obtained when the agent reaches the target by avoiding the obstacle in the environment, as shown in **Figure 4**.

The types of state and action are shown in **Table 1** and **Table 2**, respectively. **Table 1** shows the output actions of an agent in 8 directions and **Table 2** shows 256 types of the total input states coming from the combination of existing obstacles in 8 directions. The 8 directions are the top left, top, top right, left, right, bottom left, bottom, and bottom right. The agent has 2048 (8 actions × 512 states) rules in total that result from a combination of input states and output actions. The size of agent, target, and environment are 1 × 1, 5 × 5, and 50 × 50,

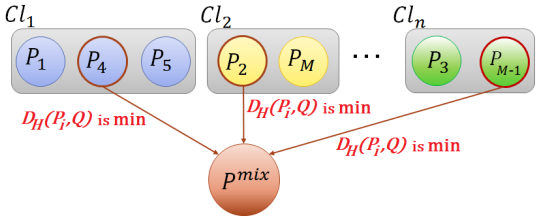

Figure 2. Element selection.

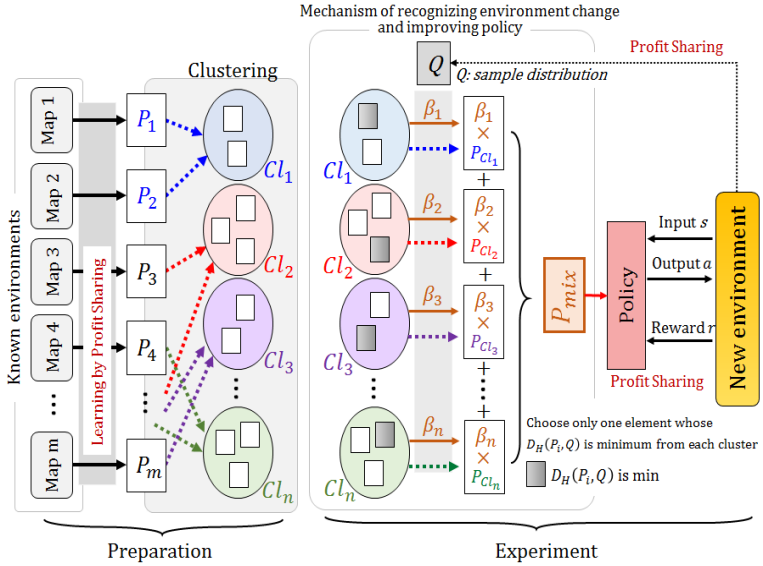

Figure 3. System framework.

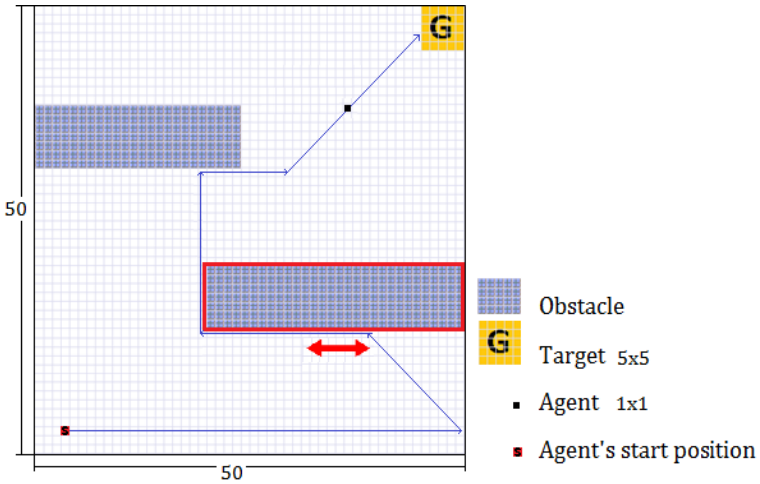

Figure 4. Environment of agent navigation problem.

Table 1. Types of action.

Direction of action			Value		
↖	↑	↗	0	1	2
←	Ⓐ	→	3	Ⓐ	4
↙	↓	↘	5	6	7

Table 2. Some types of state.

Position of obstacle and value						
			…		…	
0	1	2		111		255

respectively. Some of the known environments that became mixture probability elements, and the unknown dynamic environments (E_A, E_B, E_C) used to evaluate the policy improvement are shown in **Figure 5** and **Figure 6**, respectively.

3.2. Experimental Procedure

The agent learns the policy by using the profit-sharing method. A trial is considered to be successful if an agent reaches the target at least once out of 300 action attempts. The action is selected by randomization and that action continues until the state is changed.

The purpose of the experiment is to learn the policy in unknown dynamic environments E_A, E_B and E_C in three cases (fixed obstacle, periodic dynamic and nonperiodic dynamic environments), by employing only the profit-sharing method and the mixture probability scheme (elements are *m* and *n*); the evaluation is based on the success rate of 2000 trials. The experimental parameters are shown in **Table 3**.

3.3. Results and Discussion

The success rate of policy improvement in E_A, E_B and E_C by using mixture probability and clustering in

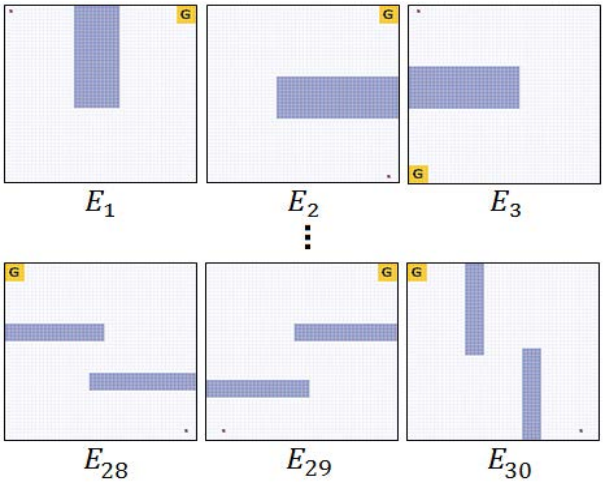

Figure 5. Some of the known environments.

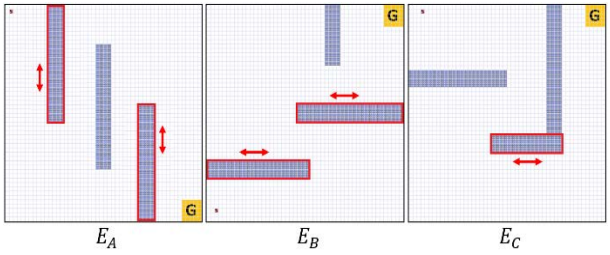

Figure 6. Unknown environments.

Table 3. Experimental parameters.

Variable	Value	Variable	Value
t	300	τ	20
γ	0.8	w_0	10.0
r_0	nonfixed	n	10, 15, 20
r_0	100	m	30

fixed-obstacle cases, and the processing time from **Step 3** (system flow) to the experiment conclusion are shown in **Figure 7** and **Table 4**, respectively.

Figure 7 shows that the success rate by using mixture probability is clearly higher than when using only the profit-sharing method in all environments. Even the success rate by using only 10 elements is also higher than that using only the profit-sharing method, but is still lower compared to the results using 15 and 20 elements in E_A and E_B. Hence, we can say that the influence on policy improvement by reducing the number of elements is apparent in some environments.

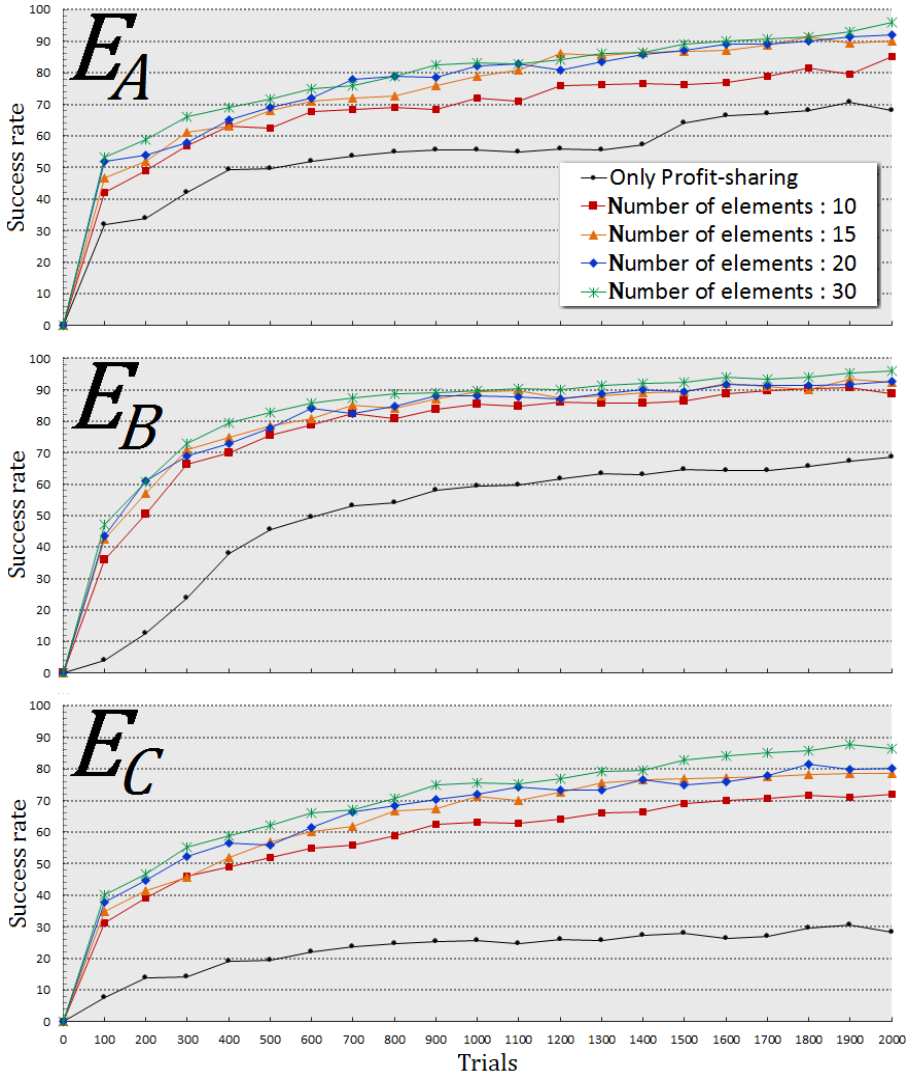

Figure 7. Transition of success rate (fixed-obstacle).

Table 4. Processing time.

	Element number and processing time (s)			
	30	20	**15**	10
E_A	19.622	15.537	**12.240**	9.356
E_B	22.107	17.963	**14.578**	10.125
E_C	18.250	14.641	**11.894**	9.087

We can confirm that the success rate is higher when using more mixture probability elements in any environment. The success rate using all 30 elements was the highest, but that obtained using 15 elements was almost the same as that using all the elements in this result.

Furthermore, from the results in **Table 4**, we can confirm that by reducing the number of elements, the processing time was reduced considerably.

From these results, it can be seen that the immediate success rate obtained by policy improvement is higher than that obtained by only the profit-sharing method in all environments, and the higher success rate continues until the experiments end. Furthermore, the decline in effectiveness can be controlled even if the number of mixture probability elements is reduced by half based on the use of clustering.

The results of policy improvement by using 15 elements of mixture probabilities in three cases are shown in **Figure 8**, and the results of five sets of experiments in periodic and nonperiodic dynamic movement in E_A and E_B are shown in **Figure 9**, respectively.

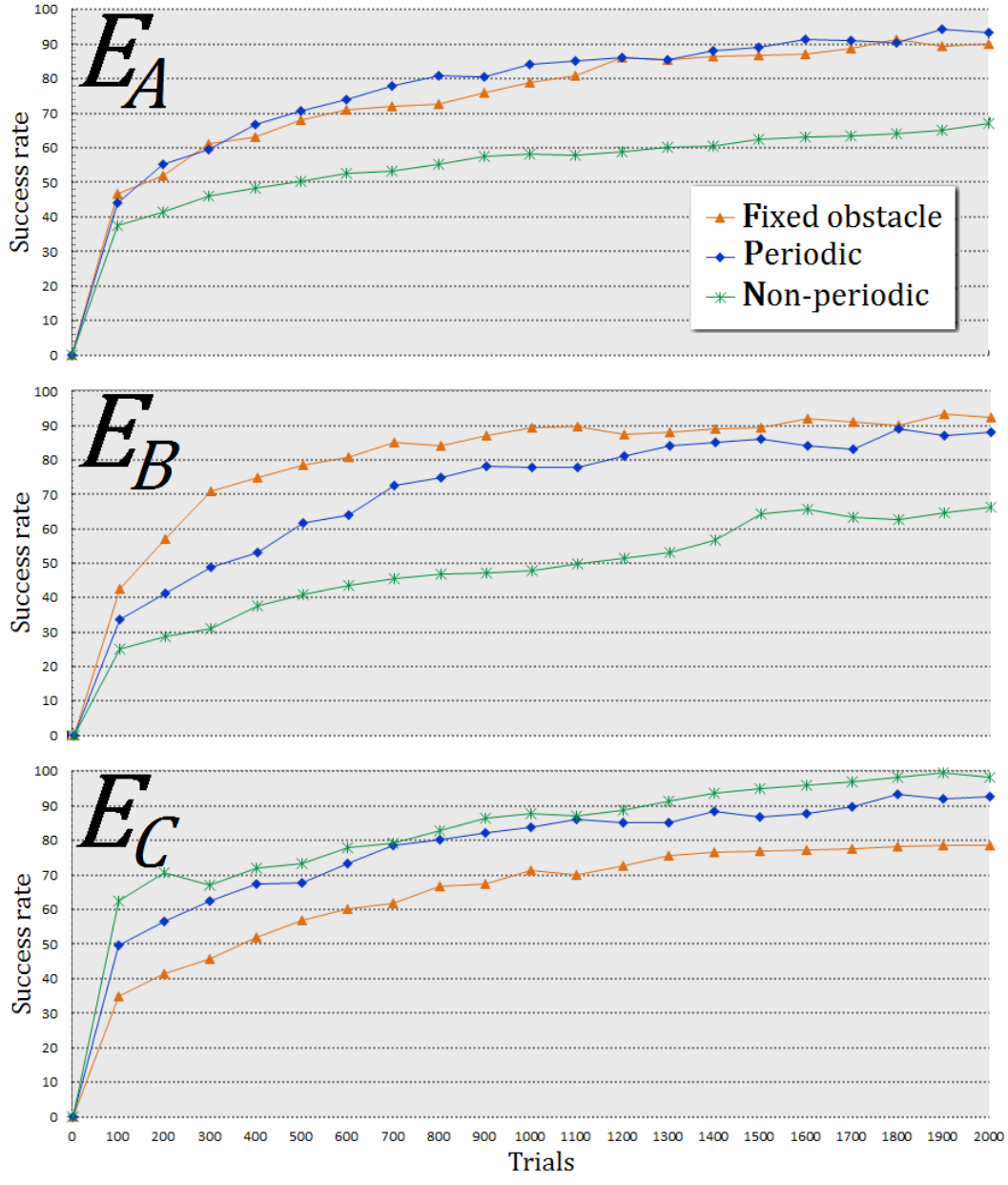

Figure 8. Transition of success rate (3 cases).

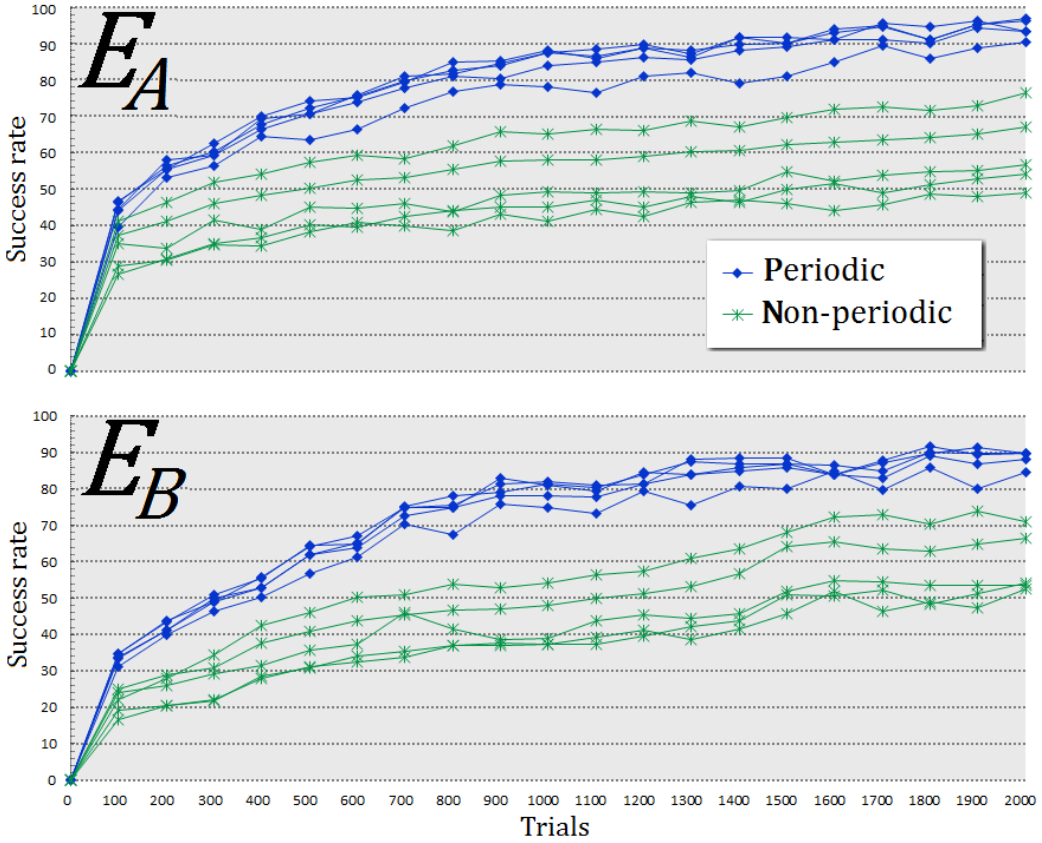

Figure 9. Five sets of experiments on E_A and E_B.

Figure 8 shows that the success rate in E_A and E_B in the case of periodic dynamic movement was slightly lower in the early period compared with the fixed obstacle case, but there was almost no difference finally. However, the success rate in the case of periodic dynamic movement was conversely higher in E_C. On the other hand, in the case of nonperiodic dynamic movement, even the success rate in E_C was higher compared with the fixed obstacle case, but the results in E_A and E_B were quite low and unstable compared with the other cases, as shown in **Figure 9**.

From these results, we can deduce that the agent successfully learns the policy in the periodic dynamic movement environment and can more easily reach the target when the obstacle moves out from the trajectory as in E_C. On the contrary, when the obstacle moves into the trajectory, it will be more difficult for the agent to reach the target.

4. Conclusions

Humans can visually judge a new environment and easily select the appropriate rule to reach the target. However, this is not so for robots. A robot cannot judge a new environment by sight, and so it is necessary to select various rules to make a robot reach the target. In addition, to be more efficient, the robot needs to learn the policy by using knowledge obtained from prior target arrivals. In this paper, we used the joint distributions $P(s,a)$ as the knowledge and the sample distribution Q to find the degree of similarity between the unknown and each known environment. We then used this as the basis to update the initial knowledge as being very useful for the agent to learn the policy in a changing environment. Even if obtaining the sample distribution is time consuming, it is still worthwhile if the agent can efficiently learn the policy in an unknown dynamic environment.

Also, by using the clustering method to collect similar elements and then selecting just one suitable joint distribution as the mixture probability elements from each cluster, we can avoid using similar elements to maintain a variety of elements when we reduce their number.

From the results of the computer experiment as an example application in the agent navigation problem, we can confirm the following:

- The policy improvement in unknown dynamic environments is effective.
- The decline in effectiveness of the policy improvement can be controlled by using the clustering method.

We conclude that the improvement of stability and speed in policy learning, and the control of computational complexity are effective by using our proposed system.

Examination of the computational complexity load and adaptation performance in a dynamic 3D environment is necessary. Improvement of the RL policy is also required by using mixture probability with a positive and negative weight value for making the system adaptable to unknown environments that are not similar to any known environments. Finally, a new reward process is needed as well as a new mixing parameter for the agent to adjust to a changing environment more efficiently and to be able to work well in 3Denvironments in future work.

References

[1] Sutton, R.S. and Barto, A.G. (1998) Reinforcement Learning: An Introduction. MIR Press, Cambridge.

[2] Croonenborghs, T., Ramon, J., Blockeel, H. and Bruynooghe, M. (2006) Model-Assisted Approaches for Relational Reinforcement Learning: Some Challenges for the SRL Community. *Proceedings of the ICML*-2006 *Workshop on Open Problems in Statistical Relational Learning*, Pittsburgh.

[3] Fernandez, F. and Veloso, M. (2006) Probabilistic Policy Reuse in a Reinforcement Learning Agent. *Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multi-Agent Systems*, New York, 720-727. http://dx.doi.org/10.1145/1160633.1160762

[4] Kitakoshi, D., Shioya, H. and Nakano, R. (2004) Adaptation of the Online Policy-Improving System by Using a Mixture Model of Bayesian Networks to Dynamic Environments. *Electronics, Information and Communication Engineers*, **104**, 15-20.

[5] Kitakoshi, D., Shioya, H. and Nakano, R. (2010) Empirical Analysis of an On-Line Adaptive System Using a Mixture of Bayesian Networks. *Information Science*, **180**, 2856-2874. http://dx.doi.org/10.1016/j.ins.2010.04.001

[6] Phommasak, U., Kitakoshi, D. and Shioya, H. (2012) An Adaptation System in Unknown Environments Using a Mixture Probability Model and Clustering Distributions. *Journal of Advanced Computational Intelligence and Intelligent Informatics*, **16**, 733-740.

[7] Tanaka, F. and Yamamura, M. (1997) An Approach to Lifelong Reinforcement Learning through Multiple Environments. *Proceedings of the Sixth European Workshop on Learning Robots*, EWLR-6, Brighton, 93-99.

[8] Minato, T. and Asada, M. (1998) Environmental Change Adaptation for Mobile Robot Navigation. *Proceedings of IEEE/RSJ International Joint Conference on Intelligent Robots and Systems*, IROS'98, Victoria, 1859-1864.

[9] Pearl, J. (1988) Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann Pub. Inc., San Francisco.

[10] Hellinger, E. (1909) Neue Begründung der Theorie quadratischer Formen von unendlichvielen Veränderlichen. *Journal für die Reine und Angewandte Mathematik*, **136**, 210-271.