Australian Journal of Basic and Applied Sciences, 8(4) Special 2014, Pages: 455-458

AENSI Journals

Australian Journal of Basic and Applied Sciences

ISSN:1991-8178

Journal home page: www.ajbasweb.com

Investigation of Reinforcement Learning With Multiplex Learning Spaces

C. Nishizawa, H. Matsui, Y. Nomura

Graduate School of Engineering, Mie University, Kurima Machiya 1577, Z, 514-8507, JAPAN.

ARTICLE INFO

Article history:
Received 20 November 2013
Received in revised form 24 January 2014
Accepted 29 January 2014
Available online 5 April 2014

Key words:
Reinforcement Learning With Multiplex Learning Spaces

ABSTRACT

In this paper, we propose a Q-learning method with multiplex Q-tables. In the method, an agent has some Q-tables. The agent evaluates all the Q-tables based on information entropy at each step, and selects an action by the best Q-table. In some of ordinary methods, an agent has plural Q-tables, but doesn't apply plural Q-tables for the same task, while comparing one Q-table with others at each step. We confirmed in experimental simulations that the proposed learning was earlier converging on a better policy than ordinary simple methods.

To Cite This Article: C. Nishizawa, H. Matsui, Y. Nomura, Investigation of Reinforcement Learning With Multiplex Learning Spaces. ***Aust. J. Basic & Appl. Sci.,*** *8(4): 455-458, 2014*

INTRODUCTION

Animals including human being can get a food of reward in an efficient way, but is not taught how to act for getting the reward by others. An animal recognizes environmental changes to be caused by the own actions and learns appropriate actions through trial and error. Reinforcement learning is modeling of the animal learning process. Reinforcement learning is easy to apply to a robot action learning, since a robot designer only sets the rewards at ultimate goals. Reinforcement learning has an advantage that the designing is simple, but has a disadvantage that the learning needs many trial and errors. The trial frequency increases exponentially by increasing the states to represent uncertain environment in detail, such as actual environment.

In an ordinary Q-learning method, an agent has only one Q-table a policy, which has some Q-values. A policy is a function to represent which action should be selected at each state. Each Q-value is the evaluating value of a state-action pair. Each Q-value is updated by the value of the next state leaded by the action. The value of a state is decided by the highest value among all the Q-values related to the state. If an agent acquires a good policy, it is same that it finds a good way to reach the goal. In general, a rougher Q-table is converging earlier in learning, but the agent cannot learn an optimal policy. Since a rough Q-table cannot represent exactly the learned environment by too rough states. An agent with a more detailed Q-table can learn an optimal policy, but it is converging more slowly in learning, since a detailed Q-table has much inexperienced state-action pairs early in learning, increasing the number of inexperienced pair's cause's exponentially slower learning. However, none knows which Q-table has optimal size for an environment before the trials.

As methods to let a learning be more efficient, a multi-layered reinforcement learning (Yasutake *et al*., 2003; Eiji and Kenji, 2004) split and arranged hierarchically learning spaces. But it needs a priori knowledge for the learning spaces to switch from one to another. The method (Tomoli *et al*., 2003) adaptively segments states to prevent increasing the states in a robot task. But the result of segments depends on initial segments.

In contrast, we proposed the learning method (Osamu *et al*., 2006) to decrease the trial frequency prepared not only a whole learning space but a part learning space for an environment, that are applied at the same time, without a priori knowledge. However the learning speed depends on selecting a part learning space, the result of learning is independent of selecting a part learning space. In this paper, we extend the proposed method to multiplexing learning spaces.

Proposed Method:

Here, we describe the proposed Q-learning method with multiplex Q-tables. It applies multiplex Q-tables that represent the same learned environment from in detail to roughly, relatively, at the same learning step. It selects the best Q-table based on information entropy among all the Q-tables at the state of each step, and updates the Q-table based on the result. We consider Q-table to be more effective, that has lower information

Corresponding Author: C. Nishizawa, Graduate School of Engineering, Mie University, KurimaMachiya 1577, Z, 514-8507, JAPAN.

entropy at a state, since the efficiency of a Q-table at a state is indicated by the difference of Q-values at a state in a Q-table. Some of ordinary Q-learning methods have plural Q-tables, but they don't apply plural Q-tables for the same task, while comparing one Q-table with others at each step.

We design the proposed method to compensate that disadvantage with this advantage and this disadvantage with that advantage, each other. Applying multiplex Q-tables means that the environment is observed from multi-viewpoint. In other words, the proposed method evaluates a state from multi-viewpoints at the same step, and applies viewpoints in order of high probability of success. We show the learning algorithm for the multiplex learning spaces in Fig. 1.

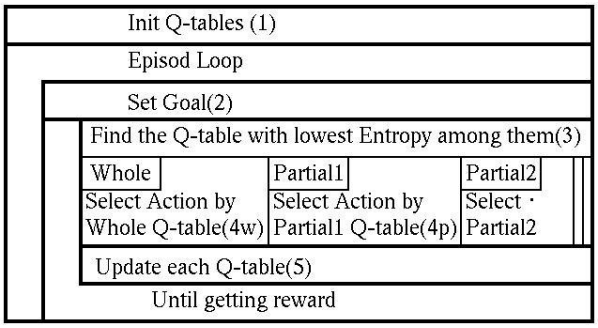

Fig. 1: NS chart of the reinforcement learning with multiplex Q-tables.

1. Init Q-tables:

A learning agent initializes all the multi-Q-tables to set each Q-value to an init value.

2. Set goal:

The designer decides state of goal where the agent gets the reward.

3. Select Q-table:

The agent selects the Q-table with lowest information entropy H(s), among all the Q-tables at states, that are calculated by Eq. (1).

$$Entropy = \sum_{a \in actions} p(a|s^k) \, log_2 \frac{1}{p(a|s^k)} \tag{1}$$

where $p(a \mid s^k)$ is a probability of selecting action a at the state s^k, that is defined by the following Eq. (2).

4. Select Action:

The agent decides an action by the Boltzmann selection applied generally on Q-learning. The selection probability of the actions is shown by Eq. (2).

$$p(a_i|s^k) = \frac{exp\left(\frac{Q(s^k,a_i)}{T}\right)}{\sum_{a \in actions} exp\left(\frac{Q(s^k,a_i)}{T}\right)} \tag{2}$$

where $p(a \mid s^k)$ is probability of selecting action at states^k, k is time, T is temperature.

5. Update each Q-table:

Q-value is updated by Eq. (3), (4).

$$Q(s^k, a) \leftarrow (1-\alpha)Q(s^k, a) + \alpha r + \gamma V(s^{k+1}) \tag{3}$$

$$V(s^{k+1}) = \max_{a \in actions} Q(s^{k+1}, a) \tag{4}$$

wheres^k is the state at time k, s^{k+1} is the next state, a is selected action, r is reward, α is learning rate ($0 < \alpha < 1$),γ is discount rate($0 < \gamma < 1$).

3. Simulation:

Australian Journal of Basic and Applied Sciences, 8(4) Special 2014, Pages: 455-458

Here, we confirm that the proposed method is effective in a case of simulation with three Q-tables.

Experiment Condition:

We simulate the following conditions: In the experimental field, a mobile robot and an object exist as shown Fig. 2. The robot has a camera, and it moves according to information from the camera. The object has single color. The color is changed, when the object moves. The robot gets reward, if it finds the colored object at the determined position on the image plane ((a) in Fig.2). In this experiment, the different goal position is determined by each color. After the robot gets reward, the object moves to a random position in the field, and the object changes the color at random ((b) in Fig.2).

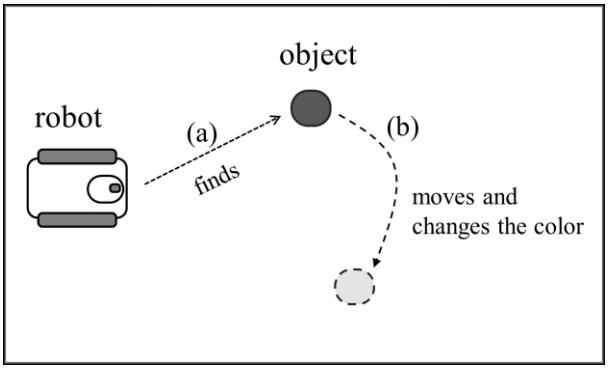

Fig. 2: Experimental Field of Simulation

The robot learns a policy of the set of state-action pairs to get more rewards. We show the learning's with three Q-tables in Fig. 3. Q-table W (enough detailed), that has 3 axes: position axis, color axis and action axis. Q-table Pp (rough one), that has 2 axes: position axis and action axis. Q-table Pc (the rough other), that has 2 axes: color axis and action axis.

The color axis has 4 states, the position axis has 50 states, the action axis has 8 states. All the Q-values are initialized to 0.0, the reward r is set to ±1.0, the learning rate α is set to 0.08, the discount rate γ is set to 0.8, the temperature T of Boltzmann selection is set to 0.1.

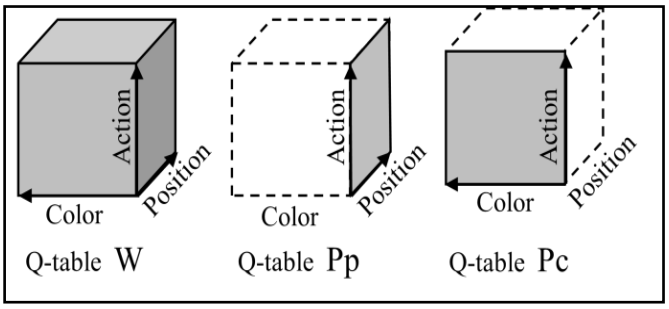

Fig. 3: Enough detailed and rough Q-tables.

RESULTS AND DISCUSSION

Here, we simulated some combinations of the three Q-tables. The results of simulations are shown in Fig. 4. The horizontal axis indicates the episode number; an episode is a period between a goal and the previous goal. The vertical axis indicates the total number of steps, a step is a period of doing an action. A gradient of each graph indicates a number of steps per episode. If the gradient becomes constant, the learning has converged. If the gradient is smaller, the learning acquires a better policy. If learning is efficient, the learning acquires early a good policy. An ordinary method is learning with each Q-table, one the detailed and two the rough Q-tables, W, Pp, Pc.

The learning with Q-table Ppwas converging earlier than W early in the learning as shown Fig. 4. It shows that the rough Q-table Ppacquired a better policy early in the learning. However, the learning with Q-table Pp had converged on a worse policy than W, late in the learning as shown in Fig. 4. It shows that the Q-table Ppacquired a worse policy than W at last.

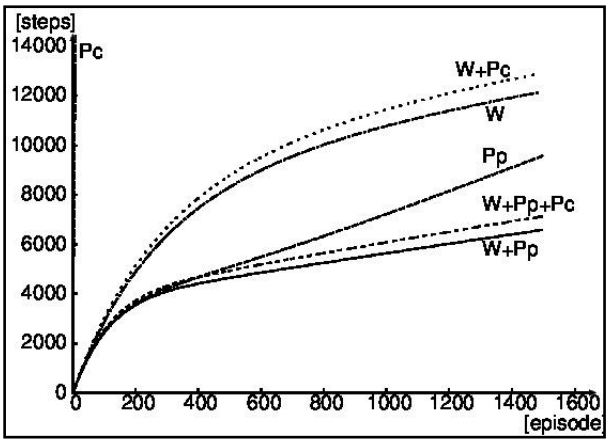

Fig. 4: Simulation Results of Some Combinations of the Three Q-Tables

The gradient of Pc graph was too large. It shows that the rough Q-table Pc could not acquire a policy for goal achievement at all.

The rough Q-table Ppwas available. The learning with Q-tables W+Pp converged earlier than that only with Q-table W, and acquired as efficient a policy as W at last. It shows that the robot acquired an efficient policy early in the learning by adding available Q-table.

The rough Q-table Pc wasn't available at all. But the learning with Q-tables W+Pc converged slightly as late as that only with Q-table W, and it acquired as efficient a policy as W. It shows that the robot could learn an efficient policy with a little bit disturbances, even if the learning was disturbed by adding unavailable Q-table.

The learning with three Q-tables W+Pp+Pc converged earlier than W, and it acquired as efficient a policy as W. It shows that it was effective that the proposed method is extended from dual Q-tables to multiplex Q-tables. However, the learning with Q-tables W+Pp+Pc converges slightly as late as that with Q-tables W+Pp, and it acquired as efficient a policy as W+Pp. The results show that the learning with multiplex Q-tables was more efficient in the case of adding available Q-tables and was slightly less efficient in the case of adding unavailable Q-tables.

Conclusion:

In this paper, we extended the proposed method to multiplexing learning spaces. We confirmed that it makes the learning more efficient with available spaces, and it makes the learning slightly inefficient with unavailable spaces, with multiplex learning spaces, too. We can anticipate efficiency by further multiplexing, whereas those show the possibility that multiplexing has large inefficiency by accumulating slight inefficiency with unavailable spaces. In the case, we must consider how to pick up available learning spaces as multiplex learning spaces.

REFERENCES

Eiji, U. and D. Kenji, 2004. "Hierarchical Reinforce-ment Learning for Multiple Reward Functions," Journal of the Robotics Society of Japan, 22(1): 120-129.

Osamu, N., M. Hirokazu, H. Chieko, N. Yoshihiko, 2006. " Reinforcement Learning with Self-Instruction by using dual Q-tables," AROB 11th.

Tomoki, H., K. Seiichi, H. Hironori, 2003. "An Adjustment Method of the Number of States on Q-Learning Segmenting State Space Adaptively," Journal of the Institute of Electronics, Information, and Communication Engineers, J86-D-I(7): 490-499.

Yasutake, T., A. Minoru, 2003. "State-Action Space Construction for Multi-Layered Learning System," Journal of the Robotics Society of Japan, 21(2): 164-171.