Multi-Module Learning System for Behavior Acquisition in Multi-Agent Environment

Yasutake Takahashi*†, Kazuhiro Edazawa*, and Minoru Asada*†

*Department of Adaptive Machine Systems,
† HANDAI Frontier Research Center,
Graduate School of Engineering, Osaka University, Japan

Abstract

The conventional reinforcement learning approaches have difficulties in handling the policy alternation of the opponents because it may cause dynamic changes of state transition probabilities of which stability is necessary for the learning to converge. A multiple learning module approach would provide one solution for this problem. If we can assign multiple learning modules to different situations in which the each module can regard the state transition probabilities as consistent, then the system would provide reasonable performance. This paper presents a method of multi-module reinforcement learning in a multiagent environment, by which the learning agent can adapt its behaviors to the situations as results of the other agent's behaviors. We show a preliminary result of a simple soccer situation.

1 Introduction

There have been an increasing number of approaches to robot behavior acquisition based on reinforcement learning methods [1, 2]. The conventional approaches need an assumption that the environment is almost fixed or changing slowly so that the learning agent can regard the state transition probabilities as consistent during its learning. Therefore, it seems difficult to apply the reinforcement learning method to a multiagent system because a policy alteration of the other agents may occur, which dynamically changes the state transition probabilities from the viewpoint of the learning agent.

There are a number of works on reinforcement learning systems in a multiagent environment. Asada et al. [3] proposed a method which estimates the state vectors representing the relationship between the learner's behavior and those of other agents in the environment using a technique from system identification, then reinforcement learning based on the estimated state vectors is applied to obtain the cooporative behavior. However, this method requires a global learning schedule in which only one agent is specified as a learner and the rest of agents have a fixed policies. Therefore, the method cannot handle the alternative of the opponents policies. This problem happens because one learning module can maintain only one policy.

A multiple learning module approach would provide one solution for this problem. If we can assign multiple learning modules to different situations in which the each module can regard the state transition probabilities as consistent, then the system would provide reasonable performance.

Singh [4, 5] has proposed compositional Q-learning in which an agent learns multiple sequential decision tasks with multi learning modules. Each module learns its own elemental task while the system has a gating module for the sequential task, and this module learns to select one of the elemental task modules. Takahashi and Asada [6] proposed a method by which a hierarchical structure for behavior learning is self-organized. The modules in the lower networks are organized as experts to move to different categories of sensor value regions and learn lower level behaviors using motor commands. In the meantime, the modules in the higher networks are organized as experts which learn higher level behavior using lower modules. Each module assigns its own goal state by itself. However, there are no such measure to identify the situation that the agent can change modules corresponding to the current situation.

Sutton [7] has proposed DYNA-architectures which integrate world model learning and execution-time planning. Singh [8] has proposed a method of learning a hierarchy of models of the DYNA-architectures. The world model is not for the identification of the situations, but only for improving the scalability of reinforcement learning algorithms.

Doya et al. [9] have proposed MOdular Selection and Identification for Control (MOSAIC), which is a modular reinforcement learning architecture for non-linear, non-stationary control tasks. The basic idea is to decompose a complex task into multiple domains in space and time based on the predictability of the environmental dynamics. Each module has a state prediction model and a reinforcement learning controller. The models have limited capabilities of state prediction as linear predictors, therefore the multiple prediction models are required for a non-linear task. A domain is specified as a region in which one linear predictor can estimate sensor outputs based on its own prediction capability. The responsibility signal is defined by a function of the prediction errors, and the signals of the modules define the outputs of the reinforcement learning controllers.

We adopt the basic idea of combination of a forward model and a reinforcement learning into an architecture of behavior acquisition in the multi-agent environment. In this paper, we propose a method by which multiple modules are assigned to different situations and learn purposive behaviors for the specified situations as results of the other agent's behaviors. We show a preliminary result of a simple soccer situation.

2 A Basic Idea and An Assumption

The basic idea is that the learning agent could assign one reinforcement learning module to each situation which is caused by the other agents and the learning module would acquire a purposive behavior under the situation if the agent can distinguish a number of situations in which the state transition probabilities are consistent. We introduce a multiple learning module approach to realize this idea. A module consists of learning component which models the world and an execution-time planning one. The whole system performs these procedure simultaneously.

- find a model which represents the best estimation among the modules,
- calculate action values to accomplish a given task based on dynamic programming (DP).

As a preliminary experiment, we prepare a case of ball chasing behavior with collision avoidance. In the environment there are a learning agent, a ball, and an opponent which moves at random. The problem here is to find the model which can most accurately describe the opponent's behavior from the view point of the learning agent and to execute the policy which is calculated under the estimated model. It may take a time to distinguish the situation, therefore, we put an assumption.

- The policy of the opponent might change after a fixed period.

3 A Multi-Module Learning System

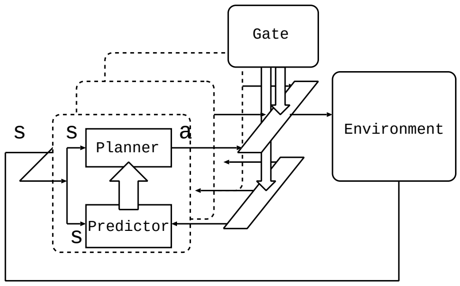


Figure 1: *A multi-module learning system*

Figure 1 shows a basic architecture of the proposed system, that is, a multi-module reinforcement learning one. Each module has a forward model (predictor) which represents the state transition model, and a behavior learner (policy planner) which estimates the state-action value function based on the forward model in the reinforcement learning manner. This idea of combination of a forward model and a reinforcement learning system is similar to the H-DYNA architecture [8] or MOSAIC [9]. In other words, we extend such architectures to an application of behavior acquisition in the multi-agent environment.

The system selects one module which has the best estimation of the state transition sequence by activating a gate signal corresponding to a module and by deactivating the gate signals of other modules, and the selected module sends action commands based on its policy.

3.1 Predictor

In this experiment, the agent recognizes a ball and the opponent in the environment. The state space of the planner consists of features of all objects in order to calculate state value (discounted sum of the reward received over time) for each pair of a state and an action. However, it is impractical to maintain a full size state transition model for real robot applications because the size of state-action space becomes easily huge and it is really rare to experience all state transitions within the reasonable learning time.

In general, the motion of the ball depends on the opponent because there are interactions between the ball and the opponent. However, the proportion of the interaction time is much shorter than that of non-interaction time. Therefore, we assume that the ball motion is independent from the opponent. Further, we assume that the opponent motion from the viewpoint of the agent seems independent from the ball positions and to depend only on the learning agent's behavior even if the opponent's decision may depend on the ball. If the system maintains the forward models of the ball and the opponent separately, each model can be much more compact and it is easy to experience almost all state transition within reasonable learning time.

As mentioned above, the module has two forward models for the ball and the opponents. We estimate the state transition probability $\hat{\mathcal{P}}^a_{ss'}$ for the triplet of state s, action a, and next state s' using the following equation:

$$\hat{\mathcal{P}}^a_{ss'} = \hat{\mathcal{P}}^a_{{}^bs{}^bs'} \cdot \hat{\mathcal{P}}^a_{{}^os{}^os'} \ , \qquad (1)$$

where the state $s \in S$ is a combination of two states in the ball state space ${}^bs \in {}^bS$ and the opponent state space ${}^os \in {}^oS$. The system has not only the state transition model but also a reward model $\hat{\mathcal{R}}^a_{ss'}$.

We simply store all experiences (state-action-next state sequences) to estimate these models. According to the assumption mentioned in **2**, we share the state transition models of the ball and the reward model among the modules, and each module has its own opponent model. This leads further compact model representation.

3.2 Planner

Now we have the estimated state transition probabilities $\hat{\mathcal{P}}^a_{ss'}$ and the expected rewards $\hat{\mathcal{R}}^a_{ss'}$, then, an approximated state-action value function $Q(s,a)$ for a state action pair s and a is given by

$$Q(s,a) = \sum_{s'} \hat{\mathcal{P}}^a_{ss'} \left[\hat{\mathcal{R}}^a_{ss'} + \gamma \max_{a'} Q(s',a') \right] \ , \qquad (2)$$

where $\hat{\mathcal{P}}^a_{ss'}$ and $\hat{\mathcal{R}}^a_{ss'}$ are the state-transition probabilities and expected rewards, respectively, and the γ is the discount rate.

3.3 Gating Signals

The basic idea of gating signals is similar to Tani and Nolfi's work [10] and the MOSAIC architecture [9]. The gating signal of the module becomes larger if the module does better state transition prediction during a certain period, else it becomes smaller. We assume that the module which does best state transition prediction has the best policy against the current situation because the planner of the module is based on the model which describes the situation best. In our proposed architecture, the gating signal is used for gating the action outputs from modules. We calculate the gating signals g_i of the module i as follows:

$$g_i = \prod_{t=-T+1}^{0} \frac{e^{\lambda p_i^t}}{\sum_j e^{\lambda p_j^t}}$$

where p_i is the occurrence probability of the state transition from the previous $(t-1)$ state to the current (t) one according to the model i, and the λ is a scaling factor.

4 Experiments

We have studied preliminary experiments so far. The task of the learning agent is to catch the ball while it avoids the collision with the opponent.

4.1 Setting

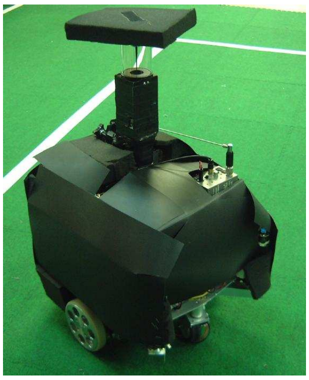
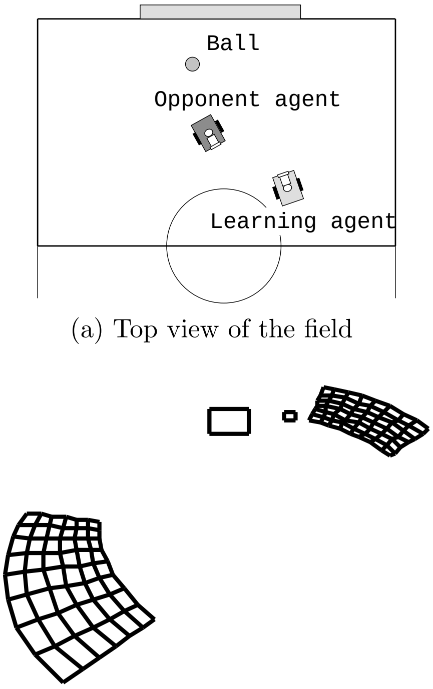


Figure 2: *Robot*

Figure 2 shows the mobile robot we have designed and built. The robot has an omni-directional camera system. A simple color image processing (Hitachi IP5000) is applied to detect the ball area and an opponent one in the image in real-time (every 33ms). Figure 3 (a) shows a situation which the learning agent can encounter and Figure 3 (b) shows the simulated image of the camera with the omni-directional mirror mounted on the robot. The larger box indicates the opponent and the smaller one indicates the ball.

The state space is constructed in terms of the centroid of the ball and the opponent on the image (Figure 4 (a)). The driving mechanism is PWS (Power Wheeled Steering) system, and the action space is constructed in terms of two torque values to be sent to two motors corresponding to two wheels (Figure 4 (b)). These parameters of the robot system are unknown to the robot, and it tries to estimate the mapping from sensory information to appropriate motor commands by the method.

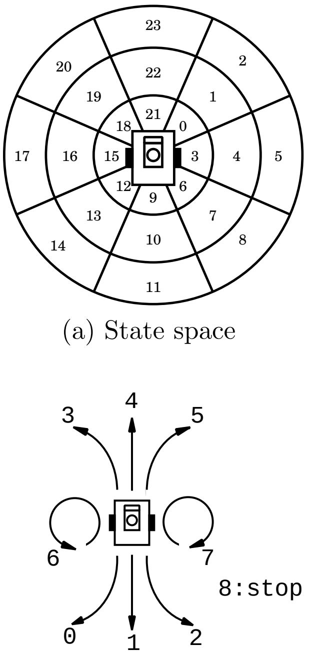
The opponent has a number of behaviors such as "stop", "move left", and "move right", and switch them randomly after a fixed period. The learning agent has models to those behaviors of the opponents. The learning agent behaves randomly while it



(a) Top view of the field

(b) Simulated image of the camera with a omni-directional mirror

Figure 3: *Simulation Environment*



(a) State space

(b) Action space

Figure 4: *State-action space*

gathers the data of the ball and the opponent image positions and builds up models for them.

4.2 Simulation Result

Table 1: *Comparison of the success rates between the agent with multi-module system and one with one-module system*

system	success rate
multi-module	61 %
one-module	50 %

We have applied the method to a learning agent and compared it with only one learning module. Table 1 shows the success rates of these two system after the learning. The success indicates that the learning agent successfully caught the ball with collision avoidance while the opponent moved randomly. The success rate indicates the number of successes in one hundred trials. The multi-module system shows better performance than the one-module system. Figure 5 shows an example sequence of the behavior when the agent executes its learned policy and the opponent behaves randomly after a fixed period. Figure 6 shows an example sequence of the gating signal (the opponent's behavior estimation) in the sequence. The arrows and alphabet indexes at the bottom correspond to the indexes of the figure 5. The agent seems to fail to estimate the opponent's behavior at the beginning and end periods, however, it accomplishes the given task. This means that even if the agent fails to estimates the other agent's behavior, there is no problem in some situations where the learning agents policy does not depend on the other agent's behavior. For example, the opponent's behavior does not depend on the agent's behavior when the ball is near and the opponent is far from the agent. In such a case, the agent does not have to estimate the other's behaviors correctly.

5 Conclusion and Future Work

In this paper, we proposed a method by which multiple modules are assigned to different situations which are caused by the alternation of the other agent policy and learn purposive behaviors for the specified situations as results of the other agent's behaviors. We have shown a preliminary result of a simple soccer situation.

Currently, we have a fixed number of learning modules and assigned modules to specific situations. As a future work, we are planning to develop a mechanism of self module assignment. We expect we can

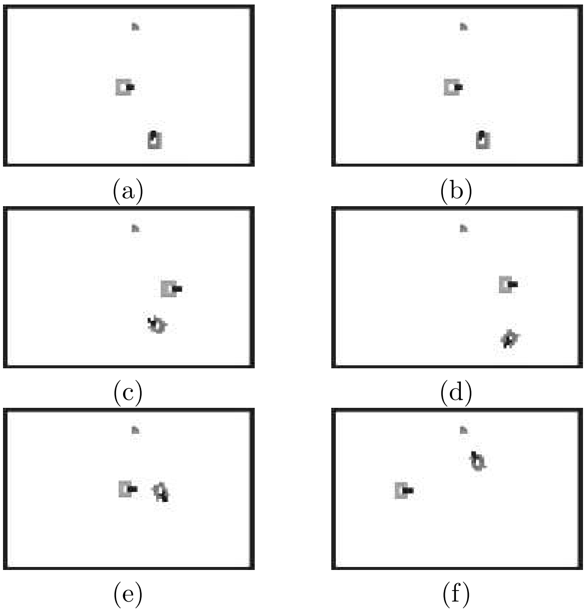

(a) (b) (c) (d) (e) (f)

Figure 5: *A sequence of a chasing behavior*

apply similar approach to the simultaneous learning problem in multi-agent system.

Acknowledgments

This research was partially supported by the Japan Science and Technology Corporation, in Research for the the Core Research for the Evolutional Science and Technology Program (CREST) titled Robot Brain Project in the research area "Creating a brain."

References

[1] M. Asada, S. Noda, S. Tawaratumida, and K. Hosoda. Purposive behavior acquisition for a real robot by vision-based reinforcement learning. *Machine Learning*, 23:279–303, 1996.

[2] Jonalthan H. Connell and Sridhar Mahadevan. *ROBOT LEARNING*. Kluwer Academic Publishers, 1993.

[3] M. Asada, E. Uchibe, and K. Hosoda. Cooperative behavior acquisition for mobile robots in dynamically changing real worlds via vision-based reinforcement learning and development. *Artificial Intelligence*, 110:275–292, 1999.

[4] Satinder Pal Singh. Transfer of learning by composing solutions of elemental sequential tasks. *Machine Learning*, 8:323–339, 1992.

[5] Satinder P. Singh. The effeicient learnig of multiple task sequences. In *Neural Information Processing Systems 4*, pages 251–258, 1992.

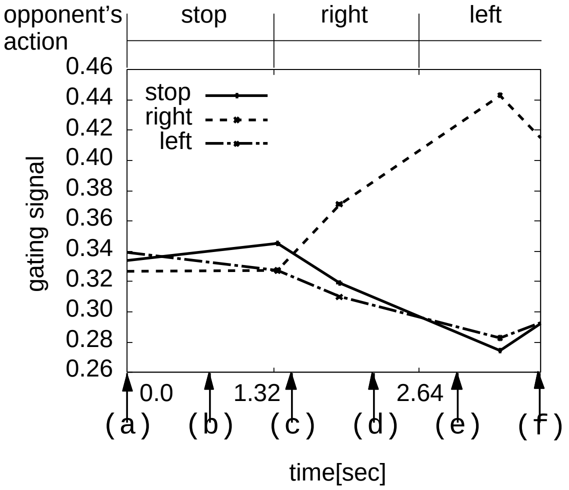


Figure 6: *A sequence of gating signal while the agent executes its learned policy*

[6] Y. Takahashi and M. Asada. Vision-guided behavior acquisition of a mobile robot by multi-layered reinforcement learning. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 1, pages 395–402, 2000.

[7] Richard S. Sutton. Integrated modeling and control based on reinforcement learning and dynamic programming. *Advances in Neural Information Processing Systems 3*, pages 471–478, 1991.

[8] Satinder P. Singh. Reinforcement learning with a hierarchy of abstract models. In *National Conference on Artificial Intelligence*, pages 202–207, 1992.

[9] Kenji Doya, Kazuyuki Samejima, Ken ichi Katagiri, and Mitsuo Kawato. Multiple model-based reinforcement learning. Technical report, Kawato Dynamic Brain Project Technical Report, KDB-TR-08, Japan Science and Technology Corporation, June 2000.

[10] Jun Tani and Stefano Nolfi. Self-organization of modules and their hierarchy in robot learning problems: A dynamical systems approach. Technical report, Technical Report: SCSL-TR-97-008, 1997.