Modular Learning System and Scheduling for Behavior Acquisition in Multi-agent Environment

Yasutake Takahashi[1], Kazuhiro Edazawa[2], and Minoru Asada[1]

[1] Emergent Robotics Area, Dept. of Adaptive Machine Systems,
Graduate School of Engineering, Osaka University,
Yamadaoka 2-1, Suita, Osaka 565-0871, Japan
{yasutake, asada}@ams.eng.osaka-u.ac.jp
[2] eda@er.ams.eng.osaka-u.ac.jp

Abstract. The existing reinforcement learning approaches have been suffering from the policy alternation of others in multiagent dynamic environments such as RoboCup competitions since other agent behaviors may cause sudden changes of state transition probabilities of which constancy is necessary for the learning to converge. A modular learning approach would be able to solve this problem if a learning agent can assign each module to one situation in which the module can regard the state transition probabilities as constant. This paper presents a method of modular 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. Scheduling for learning is introduced to avoid the complexity in autonomous situation assignment.

1 Introduction

There have been an increasing number of work to robot behavior acquisition based on reinforcement learning methods [1, 2]. The conventional approaches need an assumption that the environment is almost stationary or changing slowly so that the learning agent can regard the state transition probabilities as constant during its learning. Therefore, it seems difficult to apply the reinforcement learning method to a multiagent system because a policy alteration of other agents may occur, which dynamically changes the state transition probabilities from the viewpoint of the learning agent. RoboCup provides such a typical situation, that is, a highly dynamic, hostile environment, in which an agent has to obtain purposive behaviors.

There are a number of studies 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 of system identification, then reinforcement learning based on the estimated state vectors is applied to obtain a cooperative behavior. However, this method requires a global learning

D. Nardi et al. (Eds.): RoboCup 2004, LNAI 3276, pp. 548–555, 2005.

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 alternation of the opponents policies. This problem happens because one learning module can maintain only one policy. A modular learning approach would provide one solution to this problem. If we can assign multiple learning modules to different situations in each of which module can regard the state transition probabilities as constant, then the system could show a reasonable performance.

Jacobs and Jordan [4] proposed the mixture of experts, in which a set of the expert modules learn and the gating system weights the output of the each expert module for the final system output. This idea is very general and has wide applications. Singh [5, 6] has proposed compositional Q-learning in which an agent learns multiple sequential decision tasks with a number of learning modules. Each module learns its own elemental task while the system has a gating module which learns to select one of the elemental task modules. However, there are no such measure to identify the situation that the agent can switch modules corresponding to the change of the situation. Tani and Nolfi [7, 8] extended the idea to mixture of recurrent neural network and introduced it to predict sensory flow pattern under a navigation task. Their scheme, however, doesn't have any control learning structure, which makes it difficult to acquire a purposive behavior by itself. 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. Their idea was applied to relatively simple tasks/dynamic environment, however, it is uncertain that it is possible to assign modules automatically in the multi-agent system that has highly dynamic ones.

We adopt the basic idea of the mixture of experts 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 which are expected as the result of other agent's behavior under different policies. Takahashi et al. [10] have shown preliminary experimental results under same domain, however, the learning modules were assigned by the human designer. In this paper, scheduling for learning is introduced to avoid the complexity in autonomous situation assignment.

2 A Basic Idea and an Assumption

The basic idea is that the learning agent could assign one behavior 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 constant. We introduce a modular learning approach to realize this idea. A module consists of learning component that models the world and an execution-time planning component. The whole system performs these procedures simultaneously.

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

As a experimental task, we prepare a case of ball passing behavior without interception by the opponent player (Figs. 3,5). In the environment there are a learning agent (passer), a ball, an opponent, and two teammates (receivers). The problem here is to find the model which can most accurately describe the opponent's behavior from the viewpoint 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 opponent continues the one of its policies during one trial and changes after the trial.

3 A Multi-module Learning System

Fig. 1 shows a basic architecture of the proposed system, that is, a multi-module reinforcement learning system. 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 a reinforcement learning manner. This idea of combination of a forward model and a reinforcement learning system is similar to the H-DYNA architecture [11] or MOSAIC [9]. The system selects one module which has the best estimation of a state transition sequence by activating a gate signal corresponding to a module while deactivating the gate signals of other modules, and the selected module sends action commands based on its policy.

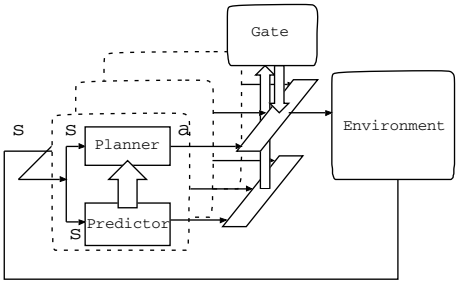


Fig. 1. A multi-module learning system

3.1 Predictor

Each learning module has its own state transition model. This model estimates the state transition probability $\hat{\mathcal{P}}^a_{ss'}$ for the triplet of state s, action a, and next state s':

$$\hat{\mathcal{P}}^a_{ss'} = Pr\{s_{t+1} = s' | s_t = s, a_t = a\} \tag{1}$$

Each module has a reward model $\hat{\mathcal{R}}^a_{ss'}$:

$$\hat{\mathcal{R}}^a_{ss'} = E\{r_{t+1}|s_t = s, a_t = a, s_{t+1} = s'\} \tag{2}$$

We simply store all experiences (sequences of state-action-next state and reward) to estimate these models.

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

$$Q(s,a) = \sum_{s'} \hat{\mathcal{P}}^a_{ss'} \left[\hat{\mathcal{R}}^a_{ss'} + \gamma \max_{a'} Q(s', a')\right] , \tag{3}$$

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

3.3 Module Selection

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 the 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:

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

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

3.4 New Module Assignment

If all modules show worse prediction of state transition, that means all gating signals g_i of the modules become small, the system add one learning module and feed data of sensory-motor sequence to this modules for a while.

4 Task and Assumption

The task of the learning agent is to pass the ball to one of the teammates while it avoids interception by the opponent. The game is like a three on one; there are

Fig. 2. A real robot

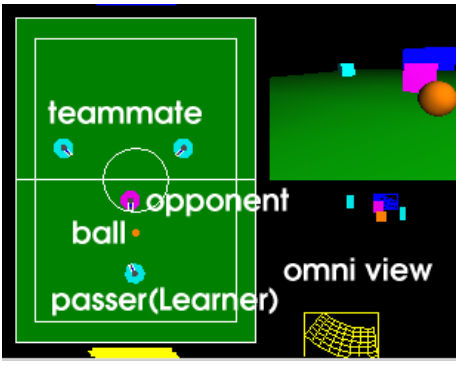


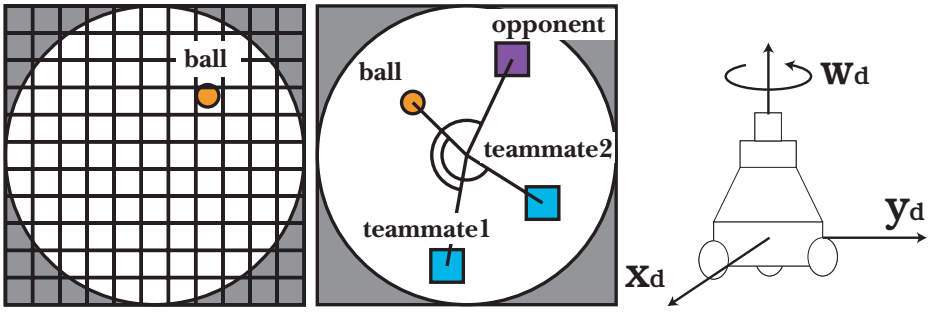
Fig. 3. A simulation environment

one opponent and other three players. The player nearest to the ball becomes to a passer and passes the ball to one of the teammates while the opponent tries to intercept it.

Fig. 2 shows a mobile robot we have designed and built. Fig. 3 shows the simulator of our robots and the environment. The robot has an omni-directional camera system. A simple color image processing is applied to detect the ball area and opponent ones in the image in real-time (every 33ms). The left of Fig. 3 shows a situation in which the agent can encounter and the bottom right shows the simulated image of the camera with the omni-directional mirror mounted on the robot. The robot consists of an omni-directional vehicle of which motion (any translation and rotation on the plane) can be controlled.

The state space is constructed in terms of the centroid of the ball on the image, the angle between the ball and the opponent, and the angles between the ball and the teammates (see Fig. 4 (a) and (b)). We quantized the ball position space 11 by 11 as shown in Fig. 4 (a) and the each angle into 8. As a result, the number of state becomes $11^2 \times 8 \times 8 \times 8 = 61952$. The action space is constructed in terms of desired three velocity values (x_d, y_d, w_d) to be sent to the motor controller (Fig. 4 (b)). Each value is quantized into three, then the number of action is $3^3 = 27$. The robot has a pinball like kick device, and it automatically kicks the ball whenever the ball comes to the region to be kicked. It tries to estimate the mapping from sensory information to appropriate motor commands by the proposed method.

The initial positions of the ball, the passer, the opponent, and teammates are shown in Figs. 5. The opponent has two kinds of behaviors; it defend the left side, or right side. The passer agent has to estimate which direction the opponent will defend and go to the position in order to kick the ball to the direction the opponent does not defend. From a viewpoint of the multi-module learning system, the passer agent will estimate which situation of the module is going on, select the most appropriate module to behave. The passer agent acquires a positive reward when it approach to the ball and kicks it to one of the teammate dodging the opponent.



(a) state variables (position) (b) state variables (angle) (c) action variables

Fig. 4. A state-action space

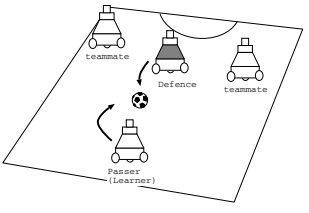
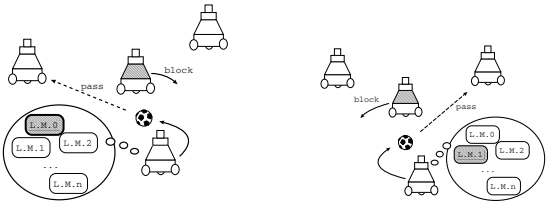


Fig. 5. Task : 3 on 1



(a) right block (b) left block

Fig. 6. Module switching

4.1 Learning Scheduling

We prepare a learning schedule composed of three stage to show its validity. The opponent fixes its defending policy as right side block at the first stage. After 250 trials, the opponent changes the policy to block the left side at the second stage and continues this for another 250 trials. Then, the opponent changes the defending policy randomly after one trial.

4.2 Simulation Result

We have applied the method to a learning agent and compared it with one module learning system. We have also compared the performances between the methods with and without the learning scheduling. Fig. 7 shows the success rates of those during the learning. The success indicates that the learning agent successfully kick the ball without interception by the opponent. The success rate indicates the rate of the number of successes in 50 trials. The multi-module system with scheduling shows better performance than the one-module system. The "mono. module" in the figure indicates "monolithic module" system and it tries to acquire a behavior for both policies of the opponent with one learning module. The monolithic module with scheduling means that we applied learning scheduling mentioned in **4.1** even though the system has only one learning module. The

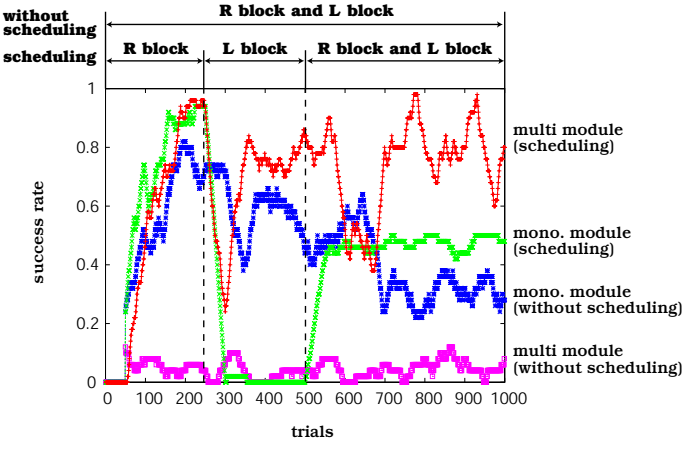


Fig. 7. Success rate during the learning

performance of this system is similar with multi-module system until the end of first stage (250 trials), however, it goes down at the second stage because the obtained policy is biased against the experiences at the fist stage and cannot follow the policy change of the opponent. Since the opponent takes one of the policies at random at the third stage, the learning agent obtains about 50% of success rate. "without scheduling" means that we do not applied learning scheduling and the opponent changes its policy at random from the start. Somehow the performance of the monolithic module system without learning scheduling is getting worse after the 200 trials. The multi-module system without learning schedule shows the worst performance in our experiments. This result indicates that it is very difficult to recognize the situation at the early stage of the learning because the modules has too few experiences to evaluate their fitness, then the system tends to select the module without any consistency. As a result, the system cannot acquires any valid policies at all.

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 consequences of the other agent's behaviors. We have shown reffectiveness of the proposed method with a simple soccer situation and the importance of the learning scheduling.

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. R. Jacobs, M. Jordan, Nowlan S, and G. Hinton. Adaptive mixture of local expoerts. *Neural Computation*, 3:79–87, 1991.
5. Satinder Pal Singh. Transfer of learning by composing solutions of elemental sequential tasks. *Machine Learning*, 8:323–339, 1992.
6. Satinder P. Singh. The effeicient learnig of multiple task sequences. In *Neural Information Processing Systems 4*, pages 251–258, 1992.
7. 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.
8. J. Tani and S. Nolfi. Self-organization of modules and their hierarchy in robot learning problems: A dynamical systems approach. Technical report, Sony CSL Technical Report, SCSL-TR-97-008, 1997.
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. Yasutake Takahashi, Kazuhiro Edazawa, and Minoru Asada. Multi-module learning system for behavior acquisition in multi-agent environment. In *Proceedings of 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages CD–ROM 927–931, October 2002.
11. Satinder P. Singh. Reinforcement learning with a hierarchy of abstract models. In *National Conference on Artificial Intelligence*, pages 202–207, 1992.