SUPERVISED LEARNING FOR AGENT POSITIONING BY USING SELF-ORGANIZING MAP

Kazuma Moriyasu, Takeshi Yoshikawa and Hidetoshi Nonaka
Graduate School of Information and Technology, Hokkaido University, Sapporo 060 0814, Japan

Keywords: Supervised learning, Self-organizing map, Multi-agent.

Abstract: We propose a multi-agent cooperative method that helps each agent to cope with partial observation and reduces the number of teaching data. It learns cooperative actions between agents by using the Self-Organizing Map as supervised learning. Input Vectors of the Self-Organizing Map are the data that reflects the operator's intention. We show that our proposed method can acquire cooperative actions between agents and reduce the number of teaching data by two evaluation experiments using the pursuit problem that is one of multi-agent system.

1 INTRODUCTION

Recently, multi-agent system is one challenge in the field of artificial intelligence. Autonomous positioning is one of main topics of multi-agent system. There are some problems to achieve the optimal positioning, for example, the perceptual aliasing problem and the concurrent learning problem. To solve such problems of multi-agent system, it will be effective to adopt reinforcement learning. But it demands a large number of trials in the early stage, because it is the approach acquiring knowledge of achieving the goal by trial and error. As another approach, it will be effective to adopt supervised learning based on operator's intuitive teaching (Akiyama and Noda, 2008). But it is difficult to give desirable actions and position of each agent for any state beforehand in the multi-agent environment where there are many agents and states. The larger the number of training data is, the more desirable results we can get. However, it is desirable that the number of training data is fewer for reducing the operator's work.

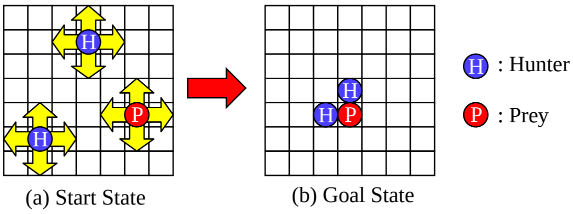
In this paper, we propose a multi-agent cooperative method where each agent can cope with partial observation. It can interpolate between teaching data and reduce the number of teaching data by using the Self-Organizing Map as supervised learning. The teaching data is made by adding operator's intuitive teaching that is each hunter agent's x,y-desirable coordinate to x,y-coordinate data of all agents and helps each hunter agent to act cooperatively. For evaluating our proposed method, we did two evaluation experiments using the pursuit problem. By the result of experiments, we show that our proposed method is effective for reduction of the number of teaching data and acquiring cooperative actions between agents in the partially observable environment.

2 PURSUIT PROBLEM

In our study, we take up the pursuit problem (Ono and Fukumoto, 1997) as a task of multi-agent system that cooperative actions between agents involve complicated processes. Many researchers have treated the pursuit problem as a benchmark problem of multi-agent reinforcement learning. In this paper, we consider the pursuit problem defined as follows: In an $n \times n$ non-toroidal grid world, a single prey agent and m hunter agents are placed randomly, as shown in Figure 1(a). The purpose of m hunter agents that we control is surrounding the prey agent by all hunter agents (Figure 1(b)). The neighboring positions are four directions (up, down, right, and left) except four corners. At every step, each agent selects an action. We call surrounding the prey agent by all hunter agents from the state of random initial position an episode. Each hunter agent moves to one of four directions (up, down, right, and left) only a square, or alternatively remains at the current position as the action. The prey agent selects among these actions randomly every step. It is prohibited for each agent to come in the same grid. The limited visual field of each hunter agent is $l \times l$ that the center is the current hunter

Moriyasu K., Yoshikawa T. and Nonaka H. (2010).
SUPERVISED LEARNING FOR AGENT POSITIONING BY USING SELF-ORGANIZING MAP.
In *Proceedings of the 12th International Conference on Enterprise Information Systems - Artificial Intelligence and Decision Support Systems*, pages 368-372
DOI: 10.5220/0003018603680372

agent's position.



(a) Start State (b) Goal State

Figure 1: Pursuit problem.

3 PROPOSED METHOD

3.1 Summary

We propose *Supervised Learning for Agent Positioning by Using Self-Organizing Map* (SLAPSOM), which is a supervised learning method using the data that the operator adds his/her intuitive teaching to x,y-coordinate data of all agents as input vectors of the Self-Organizing Map. SLAPSOM helps each hunter agent to acquire the cooperative position in the partially observable environment by learning of the Self-Organizing Map based on these data. Each hunter agent has own Self-Organizing Map. The procedure of SLAPSOM is as follows: First, the operator makes its input vector (hereafter teaching data) by adding the operator's intuitive teaching that expresses each hunter agent's x,y-desirable coordinate to x,y-coordinate data of all agents. The operator makes enough teaching data for each hunter agent. Second, the hunter agents learn the relationship between these teaching data by using the Self-Organizing Map. Last, they acquire appropriate x,y-coordinate that reflect operator's intention from own learned Self-Organizing Map for each input vector. The Self-Organizing Map is essentially unsupervised learning, but we use it as supervised learning.

3.2 Self-organizing Map

The Self-Organizing Map (SOM) is one of the neural network based on unsupervised competitive learning (Kohonen, 2001). It is composed of the input layer and the output layer, and each input unit is connected to all output units by the weight vectors (Figure 2). This structure makes multi-dimensional input vectors low level expression and the more resemble features of input vectors are, the nearer their position is on the output layer. It has turned out that the SOM is a very robust algorithm and has superior performance of interpolation, compared with many other neural models. The SOM is used in various field such as image analysis, sound analysis, and data mining, because it has these superior features.

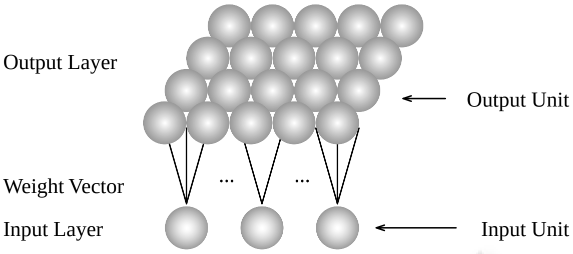


Figure 2: A basic structure of Self-organizing Map.

There are two phases called the learning phase and the judgment phase in the SOM. Learning algorithm of the SOM is as follows:

1. Set the initial value of weight vectors $m_1, \cdots, m_M$ randomly
2. Determine the winner unit c for input vector x_n by the smallest *Euclidean Distance* by:
$$c = \arg\min_i \|x_n - m_i\| \quad (1)$$
3. Update weight vectors of the winner unit c and units neighboring to c as follows:
$$m_i \leftarrow m_i + h_{ci}(t)[x_n - m_i] \quad (2)$$
where, $h_{ci}(t)$ is a Gaussian neighborhood function.
4. Repeat steps 2 to 3 until the rule number of times.

We use the fixed value of weight vectors of the learned SOM in the judgment phase. Therefore, it is possible to determine the winner unit for each input vector. The winner unit is calculated by Eq.(1) as well as the learning phase.

3.3 Implementation

SLAPSOM is composed of two phases. One is the phase that the operator makes a teaching data set composed of enough teaching data for each hunter agent as input vector of the SOM and they learn relationship between these teaching data by using the SOM. Another is the phase that each hunter agent acquires appropriate x,y-coordinate by using its own SOM. We call each phase of SLAPSOM *the learning phase* and *the execution phase*. Each phase is a counterpart of two phases of the SOM. We explain each phase in the following section.

3.3.1 Learning Phase

Teaching data sets of each hunter agent are composed of the perception part that is x,y-coordinate data of all agents and the action part that is x,y-coordinate data of own desirable x,y-coordinate (Figure 3). These teaching data sets are made by subject of the operator that overlooks the field. They help each agent to act cooperatively considering the position of each agent.

The operator makes a teaching data set for each hunter agent. Hunters learn by the SOM using own teaching data set as input vectors of the SOM. In other words, they have the SOM for exclusive use of oneself. The SOM has a superior performance of interpolation, so that it is possible to interpolate between teaching data. Therefore, learning based on a teaching data set by the SOM helps each hunter agent to acquire own appropriate x,y-coordinate and reduces the number of required teaching data.

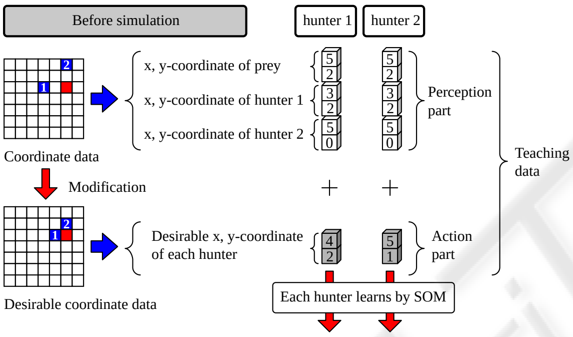


Figure 3: An example of making the teaching data set and learning by the SOM.

3.3.2 Execution Phase

Each hunter agent gives the perception information that they could get at the step in the multi-agent environment as the input vector of the SOM to own learned SOM. And they calculate *Euclidean Distance* between the weight vector of each output unit and perception information. It is the data that they could get, so that it may be partial data potentially. They determine the winner unit for the input vector of the smallest *Euclidean Distance*. They get x,y-coordinate that reflect the operator's intention from the action part of the winner unit and moves based on it at the next step (Figure 4). SLAPSOM can cope with a partial data such as perception information potentially, because the SOM is a very robust algorithm, compared with many other neural models.

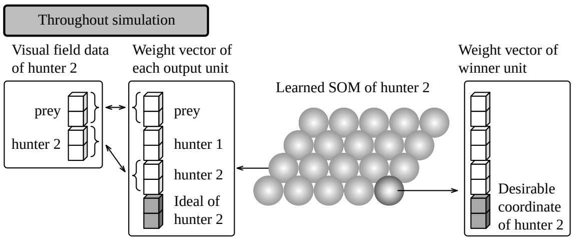

Throughout simulation
Visual field data of hunter 2
Weight vector of each output unit
Learned SOM of hunter 2
Weight vector of winner unit
prey
hunter 2
prey
hunter 1
hunter 2
Ideal of hunter 2
Desirable coordinate of hunter 2

Figure 4: An example of the calculation of execution phase.

4 EVALUATION EXPERIMENT

For evaluating SLAPSOM, we have done two evaluation experiments using the pursuit problem. In the experiment 1, we change the number of the teaching data in order to verify the learning performance of SLAPSOM for each setting of the number of the teaching data. In the experiment 2, we change the range of the visual field of the hunter agent in order to verify performance in the partially observable environment.

The common settings of two evaluation experiments about the pursuit problem are as follows: the number of hunter agents is 2, field is 7×7 non-toroidal grid world, the information that each hunter agent gets is the absolute coordinate of the agent that comes into the visual field of the hunter including itself. The settings of the SOM are as follows: map size of output layer is 30×30, the number of learning is 100000 times.

4.1 Experiment 1

In the experiment 1, we compare SLAPSOM with the neural network that is one of the most general methods as supervised learning for verifying the learning performance because SLAPSOM uses supervised learning. We change the number of the teaching data $(20, 25, 30, 35, 40, 45, 50)$ and inspect for each case. We give same teaching data set to SLAPSOM and the neural network. In our study, it is made by the operator's intuitive teaching, so that it is better that the number of teaching data is fewer.

We use three-layered neural network based on back-propagation learning algorithm (Rumelhart and McClelland, 1986). Settings of it are as follows: The number of hidden layer's units is 20. The value of learning coefficient is 0.3. The moment coefficient is 0.7. The threshold of errors is 0.08. The value of output is calculated by using perception information as well as SLAPSOM.

Figure 5 shows the result of the experiment 1. The average number of steps to the goal is the average of

10000 episode's steps. The range of the visual field of each hunter agent is 7×7. We confirmed that the average number of steps of SLAPSOM is significantly superior to that of the neural network in all cases. Reduction of the average number of steps equals to the acquirement of cooperative actions between agents effectively. We also confirmed that SLAPSOM helps reduction of the number of teaching data significantly. For example, the learning performance of SLAPSOM given 20 teaching data corresponds to that of the neural network given 40 teaching data. These results show that SLAPSOM helps acquirement of cooperative actions between hunter agents and reduction of the number of teaching data significantly as compared with the neural network.

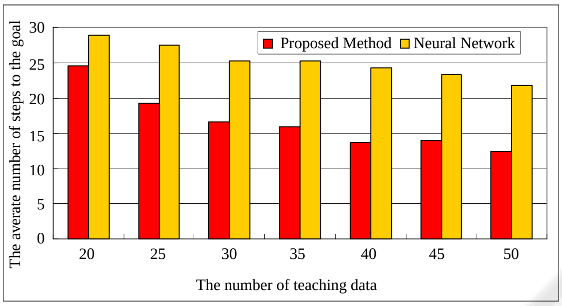


Figure 5: The result of experiment 1.

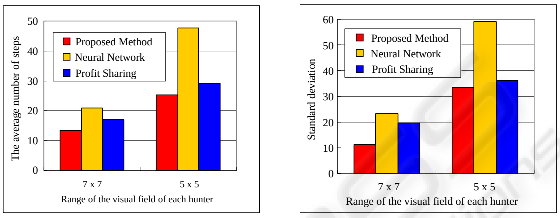
4.2 Experiment 2

In the experiment 2, we compare two cases that the ranges of the visual fields of hunter agents are 5 × 5 and 7 × 7. For evaluating SLAPSOM, we prepared three learning model: SLAPSOM, the neural network, and the profit sharing. We compare it with the profit sharing in addition to the neural network because the profit sharing is one of the most suitable methods for multi-agent reinforcement learning. Hunter agents can't get the information about other hunter agents and a prey agent that is not within their visual fields. We need to verify whether hunter agents can cope with the case that the ranges of their visual fields are narrow so that the perception information that hunter agents could get are fewer in such partially observable environment.

In this paper, we implement coarse-graining method for the profit sharing, because it demands a large number of trials in the case that the amount of perception information increases (Ito and Kanabuchi, 2001). The coarse-graining used in our experiment is the method that each hunter agent treats other agents as a part of environment. The position of the prey agent is represented by eight directions as perception information. It is hopeful that convergence of learning is much faster by treating perception information as this. Then the weight of action rule is updated as the normal profit sharing algorithm by:

$$w(s_i,a_i) \leftarrow w(s_i,a_i) + f(r,i) \qquad (3)$$

where, $w(s_i,a_i)$ is the weight of i-th action rule on a series of rule, s_i is the state, a_i is the action, r is the value of reward, and f is reinforcement function.



(a) The average number of steps.

(b) Standard deviation.

Figure 6: The result of experiment 2.

Figure 6(a) shows the result of the average number of steps to the goal for each setting. The profit sharing is the reinforcement learning, so that we use the results from 100001 to 100100 episodes after 100000 learning. The settings of the SOM and the neural network are equal to them of the experiment 1. And we gave 50 teaching data to SLAPSOM and the neural network. As the result, we confirmed that SLAPSOM was superior to the neural network and the profit sharing that is one of the most suitable method for multi-agent reinforcement learning. In the case of 5 × 5, we also confirmed that the result of the neural network gets worse significantly as compared with the case of 7 × 7. On the other hand, the result of SLAPSOM does not have a big change between 5 × 5 and 7 × 7 , as compared with the result of the neural network. The result of standard deviation is similar to the result of the average number of steps to the goal (Figure 6(b)). Therefore, it is shown that SLAPSOM helps each hunter agent to acquire cooperative actions in the partially observable environment such as the case that the range of the visual filed of the hunter agent is narrow.

5 RELATED RESEARCH

As related researches, we cite a teaching method by using Self-Organizing Map for reinforcement learning (Tateyama et al., 2004), and reinforcement learning agents with analytic hierarchy process (Katayama et al., 2004).

The former's method uses the SOM for acquiring appropriate actions of agents as well as SLAPSOM. They apply their method to a task of mobile robot navigation in addition to the pursuit problem that we take up as evaluation experiment. Their method differs from SLAPSOM in that each hunter agent excludes other hunter agents in the learning phase. They have done a simulation of mobile robot navigation with an only agent. And, in the simulation of the pursuit problem each hunter agent reinforces their actions that they approach a prey agent, so that they exclude other hunter agents. On the other hand, each hunter agent learns by the SOM with including the position of other hunter agents in SLAPSOM, so that it is possible to acquire advanced cooperative actions as ambush in addition to the action that is approach to a prey agent.

The latter's method is similar except the reinforcement learning to the pursuit problem. They also propose a combination method between analytic hierarchy process (AHP) and the profit sharing. They show that their method based on AHP that is superior in the result of the early learning stage and the profit sharing that is superior in the result of the later learning stage help each other. Their method differs from SLAPSOM in excluding other hunter agents as well as the former's method. SLAPSOM can give operator's intuitive teaching that the operator overlooks the field by using the coordinate data, and help acquiring cooperative actions between hunter agents. As another different point, their method can get only the direction that hunter agent will move, on the other hand SLAPSOM can get the coordinate that hunter agent will move at the next step. It is possible for SLAPSOM to cope with the real number environment and acquire detailed cooperative actions potentially.

6 CONCLUSIONS

In our study, we proposed a multi-agent cooperative method where each agent can cope with partial observation, interpolate between teaching data, and reduce the number of them by using the Self-Organizing Map as supervised learning.

For evaluating our proposed method, we did two experiments using the pursuit problem. As the results, our proposed method helped reduction of the number of teaching data significantly as compared with the neural network and acquiring cooperative actions between hunter agents in the partially observable environment.

In our future work, we have to do more complex experiments, because the settings of this paper's experiment were relatively simple as the field size is 7×7 and the number of hunter agents is 2. We aim to implement our proposed method for more complicated tasks of multi-agent system such as RoboCup Soccer Simulation. We consider making the GUI tools that the operator can make teaching data sets more conveniently, because we cite increase of the operator's work as a current problem of our proposed method.

REFERENCES

Akiyama, H. and Noda, I. (2008). Triangulation based approximation model for agent positioning problem. *Transactions of the Japanese Society for Artificial Intelligence*, 23(4):255–267.

Ito, A. and Kanabuchi, M. (2001). Speeding up multi-agent reinforcement learning by coarse-graining of perception –hunter game as an example–. *The transactions of the Institute of Electronics, Information and Communication Engineers*, J84-D-1(3):285–293.

Katayama, K., Koshiishi, T., and Narihisa, H. (2004). Reinforcement learning agents with analytic hierarchy process: a case study of pursuit problem. *Transactions of the Japanese Society for Artificial Intelligence*, 19(4):279–291.

Kohonen, T. (2001). *Self-Organizing Maps(3rd ed.)*. Springer.

Ono, N. and Fukumoto, K. (1997). *A modular approach to multi-agent reinforcement learning*. Springer-Verlag.

Rumelhart, D. and McClelland, J. (1986). Learning internal representation by error propagation. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, 1:318–362.

Tateyama, T., Kawate, S., and Oguchi, T. (2004). A teaching method by using self-organizing map for reinforcement learning. *Transactions of the Japan Society of Mechanical Engineers*, 70:1722–1729.