A Reinforcement Learning Method Supported by a Bayesian Network

Daisuke Kitakoshi Hiroyuki Shioya Tsutomu Da-te
Division of Systems and Information Engineering, Hokkaido University
Kita 13, Nishi 8, Kita-ku, Sapporo, 060-8628, Japan.
Fax: +81-706-7833 / E-mail: kitakosi@main.eng.hokudai.ac.jp

Abstract

A reinforcement learning (RL) is known as one of the machine learning methods, and has been applied to multi-agent problems. In this paper, we propose a new RL method using a Bayesian network (BN), which is a stochastic model and plays a role of the supervised learning procedure. An agent learns how to move under certain circumstances by an original RL method, and then the strategy is improved by using BN. We verify the effectiveness of our method by carrying out simulations for a certain multi-agent problem, and show that an agent learns its appropriate strategy for complicated tasks more effectively by using our method.

Keywords: reinforcement learning, multi-agent, supervised learning, Bayesian network, minimum description length principle

1 Introduction

In recent years, many RL methods have been suggested, refined and applied to the multi-agent problems, where agents learn their strategies to maximize the total amount of rewards decided according to a certain rule (Kondo et al., 1999; Mikami, 1997). However, it is still difficult for them to learn the optimal strategies.

We sometimes adjust our strategies, in our daily life, through each trial-and-error interaction, and then improve the strategies by using the knowledge obtained through the interactions. This adjustment and the improvement can be regarded, respectively, as RL and the supervised learning in the multi-agent problems. Therefore, we consider that combining RL with the supervised learning is effective for learning strategies in the multi-agent problems.

In this paper, we propose a new RL method using a Bayesian network (BN), which is a stochastic model and plays a role of the supervised learning procedure. BN is represented as a directed acyclic graph that expresses the relation among random variables. We assign the states (sensory inputs of an agent) and the reward to the nodes of BN, so that the knowledge of an agent may be represented in the structure of BN. In our proposed method, an agent learns its strategy through applying an original RL method, and then the strategy is improved by using the knowledge represented in the structure of BN. The structure of BN is decided by the minimum description length

International Journal of Computing Anticipatory Systems, Volume 12, 2002
Edited by D. M. Dubois, CHAOS, Liège, Belgium, ISSN 1373-5411 ISBN 2-9600262-6-8

criterion. We verify the effectiveness of our method by carrying out simulations in the pursuit problem, which is a typical multi-agent problem. The experimental results show that the agent learns its appropriate strategy more effectively by using our method.

2 Framework of a reinforcement learning method supported by a Bayesian network

This section describes about the framework of our proposed method. Each agent has its own set of state-action pairs consisting of the following two parts. State part contains information about a sensory input, and action part contains an action corresponding to the input. Each state-action pair has its own strength S which is a real value in an interval $[S_{min}, S_{max}]$. Initial value of S equals S_0.

An agent selects an action as follows. When an agent receives inputs, one of the state-action pairs whose state parts coincide with one of the inputs is selected by the roulette selection. The selection rate of the state-action pair is in proportion to its current strength. Then the agent selects an action described in action part of the selected state-action pair. Strength of a state-action pair is varied through applying profit sharing (Miyazaki et al., 1994) known as one of the RL methods. In profit sharing, a memory called "episode", reserves a series of state-action pairs. A reward, which an agent obtains, is shared with all state-action pairs in the episode. When an agent obtains a reward r at time t, strength S_i of the i-th last state-action pair in the episode is calculated as follows.

$$S_i(t+1) = f(i) \tag{1}$$

$$f(i) = S_i(t) + r\gamma^{(i-1)} \qquad (i = 1, \ldots, C) \tag{2}$$

where γ is a constant $(0 < \gamma \leq 1)$ and C is the capacity of the episode. The contents of the episode are reset after a reward is shared. In this paper, agents obtain a positive reward when it detects or captures a target mentioned later, and obtain a negative reward when it touches a wall in a simulation environment or loses sight of a target. Profit sharing is known to be effective against such complicated problem as multi-agent problem because it can learn quickly. However, it may increase the computational cost. Moreover, since we can not know the appropriate values of reward, it is still difficult for agents to learn the optimal strategies.

In order to solve the above problems, we attempt to improve the strategy in which an agent learns through applying RL methods. We show the following example of improving the strategies. In our daily life, we sometimes adjust our strategies through each trial-and-error interaction, and then improve the behaviors by using the knowledge obtained through the interactions. This adjustment and the improvement can be regarded, respectively, as RL and the supervised learning. Therefore, it is considered that combining RL with the supervised learning is effective for improving agents' strategies in multi-agent problems.

In this paper, we use a Bayesian network (BN) for the supervised learning procedure. BN, which is one of the stochastic models, is represented as a directed acyclic graph expressing the relation among random variables. Fig. 1 shows an example of BN's structure. In this figure, arcs represent stochastic relations between nodes. We assign the states and the reward to the nodes of BN, so that the knowledge of an agent may be represented in the structure of BN. Values of a random variable denote actions of agent or values of reward. Thus, each random variable (i.e. each node of BN) is corresponding to the set of state-action pairs in each state. The structure of BN is decided by the minimum description length (MDL) criterion. MDL is calculated as follows.

$$MDL = -\log P_{\hat{\theta}}^{N}(D) + \frac{d \log N}{2} \qquad (3)$$

where N is the number of sample data, $\hat{\theta}$ is the maximum likelihood estimator using the sample data D, and d is the dimension of the probability model $P_{\hat{\theta}}$. In our method, when an agent learns its strategy through applying profit sharing, sample data for deciding the structure of BN is stored in its own memory.

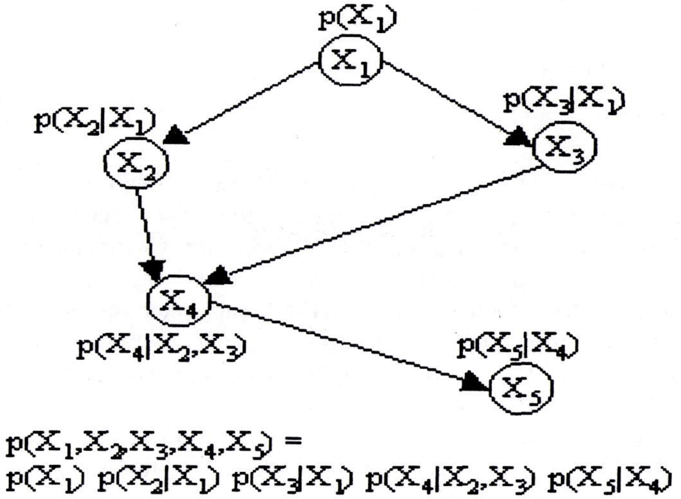

Fig. 1: an example of BN's structure

The structure of BN expresses stochastic relations between the state-action pairs, or between the state-action pair and the reward. In our method, the strategy of agent is improved according to the following procedures. First, a set M of the nodes connected with the node of reward is formed. Second, one action x_i' satisfying the following equation is selected.

$$x_i' = \arg \max_{x \in X_i} \; p(X_r = positive \quad | X_i = x) \qquad (4)$$

where X_r and X_i denote the node of reward and the i-th node of the set M, respectively. Finally, the strength of state-action pair $S_t(X_i, x_i')$ is varied according to eq. 5.

$$S_{t+1}(X_i, x_i') = S_t(X_i, x_i') + r_{im} \qquad (5)$$

where r_{im} is a constant.

While agents learn through applying a reinforcement learning method, sample data can be stored in their own memory. Moreover, it is expected that the strategy of agent is improved by using the knowledge represented in the structure of BN.

3 Experiments

This section describes simulations in the pursuit problem, which is a typical multi-agent problem, in order to verify our method. In this problem, two kinds of agents are employed. One agent called a chasing agent (CA), aims to capture a target (i.e. escaping agent mentioned later) in a simulation environment, and the other, called an escaping agent (EA), behaves according to a certain strategy. For CA, to capture EA means to touch EA, in our simulations. We use the term "energy" for quantities exchanged between CA and the simulation environment. CA has its own energy that varies in accordance with exchanges. The further assumptions about the agents are shown as follows:

- CA and EA can detect the other agent or a wall within distance Vr
- CA has initial energy of E_{ca0}
- CA and EA can select one action of "moving at high-speed or low-speed" or "staying"
- CA consumes energy E_{run}, E_{walk}, E_{tired}, or E_{-} in case of moving at high-speed, moving at low-speed, staying, or touching the walls, respectively
- CA can not select actions if its energy is equal to 0;

According to the above assumptions, tasks are carried out in three cases of one CA and one EA which behaves according to one of the following strategies: (a) it does not move; (b) it escapes from CA if it detects CA, and does not move otherwise; (c) it selects actions randomly. For CA, "to succeed the task", means, "to capture EA in the environment". We call the steps needed for CA to succeed the task "time steps" and count one step when every agent selects its own action. It is desirable to minimize the number of time steps. While the inputs of an agent at time step t ($inputs(t)$) coincide with $inputs(t-1)$, each CA and EA continues to select an $action(t)$ ($action(t)$ coincides with $action(t-1)$). In this paper, one unit denotes the quantity that an agent moves at low-speed at one time step. A body size of CA or EA is indicated by circle with radius 5 units, and the environment adopted in the simulations is indicated by square of size 500 × 500 units surrounded by the walls. The agents can move toward 8 directions in the environment. We count one trial of the simulations when CA succeeds the task, or all of its energy is consumed. The initial position of each CA and EA is fixed through all trials of the simulations. The number of trials equals 2000 in the simulations.

During the trials, EA can not change its own strategies. CA has one episode that reserves a series of 5 state-action pairs. The contents of the episode and inputs are stored, and used as sample data to decide the structure of BN. The learning of network is converged by decreasing r to 0 as CA succeeds the tasks successively. We compare two types of learning for CA:

1. CA learns only through applying profit sharing for 2000 trials (previous method, or PM)
2. CA learns only through applying profit sharing from 1 to 1000 trials, and it learns again from 1001 to 2000 trials after the improvement by using the structure of BN is done (proposed method, or PM1)

Table 1 shows the setting of the experiment.

Table 1: setting of the experiment.

S_{min}	1	Vr	140
S_{max}	20000	E_{ca0}	2000
S_0	200	E_{run}	3
γ	0.9	E_{walk}	1
Positive reward	5	E_{tired}	1
Negative reward	-5	E_{-}	3
r_{im}	100		

4 Results and discussions

Fig. 2, 3 and 5 show the rate to succeed the tasks from 1 to 20 sets, where 1 set is equivalent to 100 trials of simulation. We call the rate "success rate". In this section, each task is called task (a), (b) or (c) based on the EA's strategies, respectively. Fig. 2 shows the success rate on PM1 was similar to that on PM in each set of simulation in task (a).

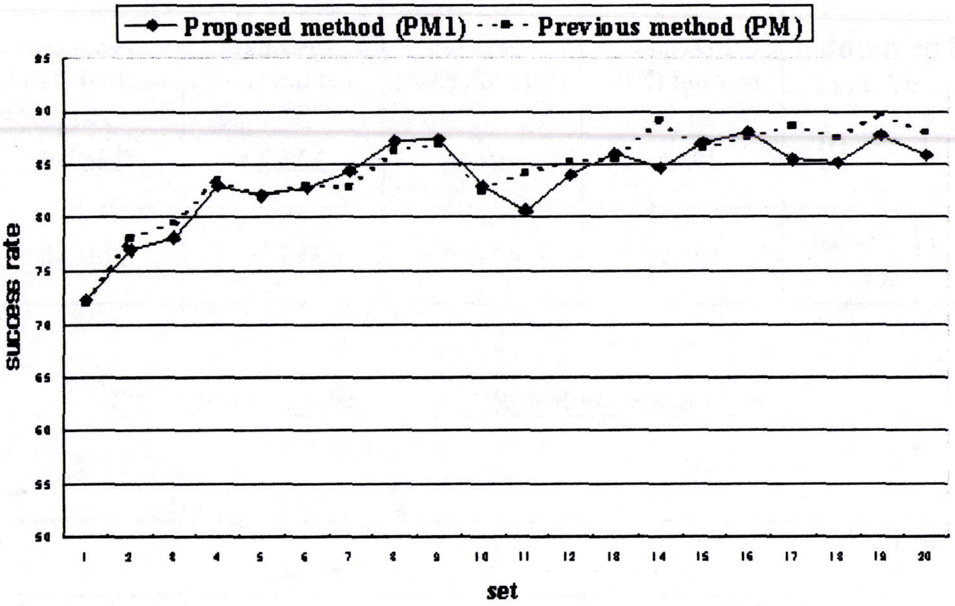

Fig. 2: relationship between the number of sets and the success rate for task (a)

Table 2 represents average number of time steps in task (a) and (b). As shown in this table, the average number of time steps on both of two methods decreased with the number of sets, similarly in task (a). It is considered that the strategy of CA is not improved effectively by using the structure of BN, in task (a). This illustrates complexities of problems affect the experimental results. In simple problems such as task (a), the agent with both of two methods learned an appropriate strategy to succeed the tasks before the first 1000 trials. Thus, it is difficult to improve the strategy of agent. In addition, appropriate sample data to decide the structure of BN is not stored effectively because agent often continues to select only specific actions to obtain positive rewards since the initial stage of learning.

Let us now compare PM1 with PM according to the result of task (b). In Fig. 3, the success rate on PM1 increased with the number of sets after the strategy was improved at 10 sets. In contrast, the rate on PM did not vary almost from 10 sets to the end of simulation. In Table 2, the number of time steps on both of two methods increased with the number of sets in task (b) because CA learned the strategy to capture EA escaping from it more frequently.

Table 2: comparison of two methods in task (a) and (b)

	Average number of time steps in task (a)		Average number of time steps in task (b)	
The number of sets	**Previous method (PM)**	**Proposed method (PM1)**	**Previous method (PM)**	**Proposed method (PM1)**
1 ~ 10 sets	**492.93**	**494.05**	**235.44**	**236.75**
11 ~ 20 sets	**475.95**	**473.20**	**315.07**	**280.38**

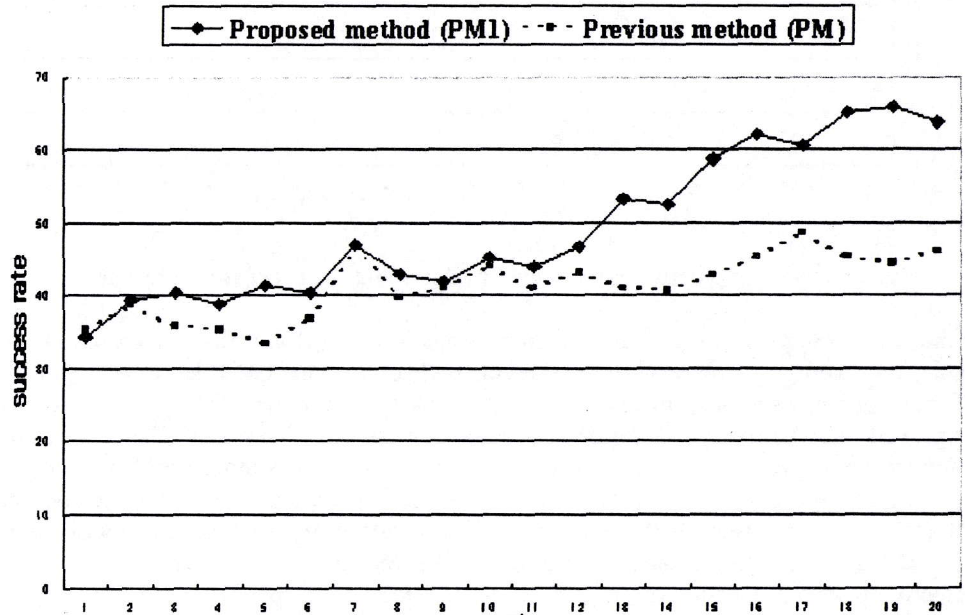

Fig.3: relationship between the number of sets and the success rate for task (b)

These results about task (b) show that the improvement of the strategy by using our method is effective in complicated problems. We show a typical example of the stochastic relations in the structure of BN, in task (b) (Fig. 4). In Fig. 4, since EA selects actions stochastically in proportion to the strength of each state-action pair to escape from CA, stochastic relations between the reward and some of the state-action pairs are represented in the structure of BN. The stochastic relations can be regarded as the knowledge of CA obtaining positive rewards, or succeeding the tasks. It is considered that the strategy of CA learning through applying profit sharing is improved by using the knowledge represented in the structure of BN because the selection rate (i.e. strength) of state-action pair, in which the probability for CA obtaining positive rewards is high, are increased by using our method.

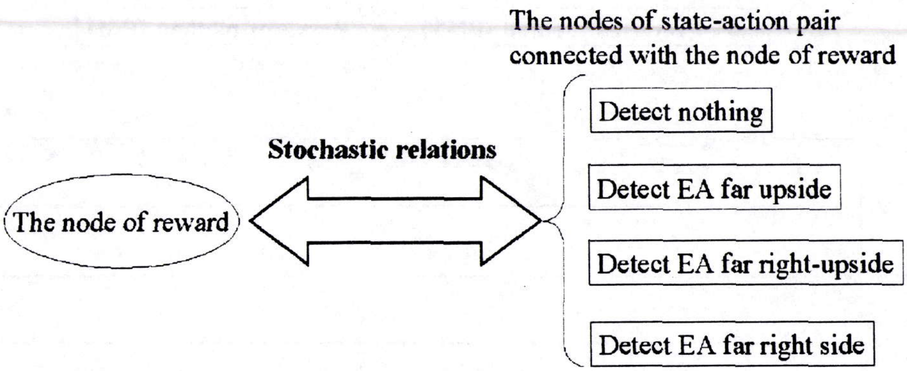

Fig. 4: an example of stochastic relations represented in the structure of BN (task (b))

Table 3: comparison of two methods in task (c)

The number of sets	**Average number of time steps in task (c)**	
	Previous method (PM)	**Proposed method (PM1)**
1 ~ 10 sets	**281.93**	**285.81**
11 ~ 20 sets	**272.60**	**271.90**

Finally, let us discuss about the result of task (c). In Fig. 5 and Table 3, the results were similar to that in task (a), however, the factors causing the results are different from that in the case of task (a). In task (c), since EA selects actions randomly, the stochastic relations between the reward and the state-action pairs are not represented in the structure of BN, and the number of them is less than that in task (a) (Table 4). Therefore, in our method, it is difficult to improve the strategy in such problems as stochastic relations between the reward and the state-action pairs can not be represented.

We considered that task (c) is complicated similarly to task (b), but the structures of BN decided (constructed) in these two tasks were quite different. It is expected that we can compare the "complexity" of problems through investigating stochastically the structures of BN representing the strategies of agents.

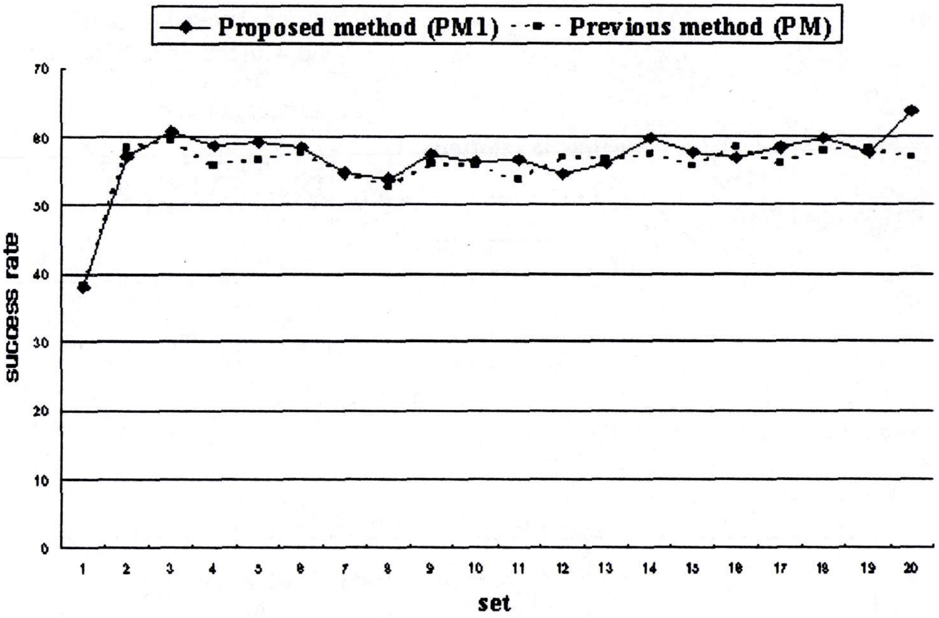

Fig 5: relationship between the number of sets and the success rate for task (c)

Table 4: comparison of the number of stochastic relations between the reward and the state-action pairs in task (a) and (c)

(Total number of nodes = 26)	task (a)	task (c)
The number of stochastic relations	**8**	**1**

5 Conclusion

In this paper, we have proposed a new reinforcement learning method using a Bayesian network. It has been confirmed that the strategy of agent through applying profit sharing are improved by using the knowledge (i.e. stochastic relations) represented in the structure of BN. In addition, sample data for deciding the structure of BN has been stored while the agent learns its strategy. However, such sample data is not enough to decide the appropriate structure in some situations. Therefore, we need to prepare new methods to store sample data effectively, and to improve the strategy of agent adaptively to the knowledge of agents represented in the structure of BN. In this paper, we adopted only profit sharing as an original RL method, but the other RL methods are available for our proposed method. Thus, it is required to investigate the effect of our method in the case of adopting the other RL methods. In addition, we must verify the effectiveness of our method in more complicated environments than those used in this research. We have represented the agent's strategy as the stochastic relations between the reward and the state-action pairs. We have to develop a method comparing the strategies stochastically by using these stochastic relations.

Acknowledgements: This research is supported by the Grant-In-Aid of The Telecommunications Advancement Foundation.

References

Ito, A. (1997). How do Selfish Agents Learn to Cooperate. Artificial Life V, pp. 142-149.

Kitakoshi, D., Nonaka, H. and Da-te, T. (2000). The Analysis of Learning and Adaptation in Autonomous Mobile Agent with Network Based Immune System. Proc. of the Society of Instrument and Control Engineers, Hokkaido Branch, pp. 127-128 (in Japanese).

Kondo, T., Ishiguro, A. and Uchikawa, Y. (1999). An Emergent Approach to Construct Behavior Arbitration Mechanism for Autonomous Mobile Robot. Transactions of the Society of Instrument and Control Engineers, Vol. 35, No. 2, pp. 262-270 (in Japanese).

Mikami, S. (1997). Reinforcement Learning for Multi-Agent Systems. Journal of Japanese Society for Artificial Intelligence, Vol. 11, No. 6, pp. 845-849 (in Japanese).

Miyazaki, K., Yamamura, M. and Kobayashi, S. (1994). A Theory of Profit Sharing in Reinforcement Learning. Journal of Japanese Society for Artificial Intelligence, Vol. 9, No. 4, pp. 580-587 (in Japanese).

Nakamura, M. and Kurumatani, K. (1997). Formation Mechanism of Pheromone Pattern and Control of Foraging Behaviour in an Ant Colony Model. Artificial Life V, pp. 48-55.

Saito, K., Shioya, H. and Da-te, T. (1999). A Treatment of Usefulness of keywords in

Fuzzy Requests for an Information Retrieval System with Bayesian networks. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, Vol. 7, No. 4, pp. 399-406.

Takadama, K., Hajiri, K., Nomura, T., Nakasuka, S. and Shimohara, K. (1997). Learning Model for Organizational Learning in Coexistent Sub Groups of Swarm Robots. The Fourth European Conference on Artificial Life, web site (http://www.cogs.susx.ac.uk/ecal97/present.html).

Unemi, T. and Nagayoshi, M. (1997). Evolution of Learning Robot Team via Local Mating Strategy. The Fourth European Conference on Artificial Life, web site (http://www.cogs.susx.ac.uk/ecal97/present.html).