Improvement of the Performance Using Received Message on Learning of Communication Codes

(Extended Abstract)

Tatsuya Kasai Hayato Kobayashi Ayumi Shinohara
{kasai, kobayashi}@shino.ecei.tohoku.ac.jp ayumi@ecei.tohoku.ac.jp
Graduate School of Information Sciences, Tohoku University, Japan

ABSTRACT

Communication is a key for facilitating multi-agent coordination on cooperative problems. On unknown problems, however, it is hard to construct beneficial communication codes. In order to tackle such problems, we focus on a method that allows agents to learn communication codes autonomously. Kasai et al. [2] proposed *Signal Learning*, by which agents learn policies of communication and action concurrently in multi-agent reinforcement learning framework. In this paper, we extend the existing signal learning and apply the extended method to an example problem, where agents can observe only partial information, for verifying the power of communication. We show that the performance of the proposed method is better than that of the existing method, and agents can obtain optimal policies on the applied problem by using the proposed method.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Multiagent systems*; I.2.11 [**Artificial Intelligence**]: Learning—*Language acquisition*

General Terms

Experimentation, Verification

Keywords

Communication, Multi-agent reinforcement learning

1. INTRODUCTION

In Multi-Agent Reinforcement Learning (MARL), each agent learns a cooperative policy $\pi : S \to A$, where S and A are a set of states and a set of actions, respectively. If we utilize communication to facilitate multi-agent coordination, we must construct communication codes so that agents can communicate with each other. However, it is a hard task since we usually do not know workable communication codes and/or information on unknown problems.

Kasai et al. [2] proposed *Signal Learning* (SL), which allows agents to learn communication codes autonomously. By using SL, agents can learn communication policy ($\pi_c : S \to M$) and action policy ($\pi_a : S \times M \to A$) concurrently in MARL framework, where M is a set of messages whose meanings are not predetermined explicitly, e.g., $M = \{1, 2, 3\}$. In addition, they showed that the performance becomes better as $|M|$ increases. It should be noted that the messages in M have no meaning in the initial phase of learning. Their results suggest that some beneficial meaning can emerge through learning process in SL, and SL is extremely helpful for unknown problems. Although there are several related studies of communication [1], it is rare to utilize meaningless messages to enhance multi-agent coordination.

Cite as: Improvement of the Performance Using Received Message on Learning of Communication Codes (Extended Abstract), Tatsuya Kasai, Hayato Kobayashi, Ayumi Shinohara, *Proc. of 8th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2009)*, Decker, Sichman, Sierra and Castelfranchi (eds.), May, 10–15, 2009, Budapest, Hungary, pp. 1229–1230

2. EXTENSION OF SIGNAL LEARNING

Our extension is just the change of communication policy from $\pi_c : S \to M$ to $\pi_c : S \times M \to M$. We call this method *SL with Messages* (SLM) to distinguish from the existing one. Algorithm 1 shows the one-step dynamics of each agent in MARL with two agents. In SL, an agent decides on a certain message to send depending only on an observed state, while in SLM, the agent decides depending on both an observed state and a received message. We expect that the performance of SLM will be better than that of SL, since much more information is available in SLM.

Algorithm 1 One-step dynamics of each agent

```
1: observe a state $s \in S$ from the environment
2: receive a message $m \in M$ sent by the other agent
3: perform the action $a = \pi_a(s, m)$ in the environment
4: send the message $m' = \pi_c(s, m)$ to the other agent
5: observe a reward $r \in \boldsymbol{R}$ from the environment
6: update $\pi_c$ and $\pi_a$ based on the reward $r$
```

3. EXAMPLE PROBLEM

We consider an example problem suitable for characterizing the difference between SL and SLM as shown in Fig. 1. The goal of the problem is that both agents, starting from their own Start/Goal (SG) states, go back to the SG states after *activation*. In order to activate the goal, both agents must occupy their Button (B) states at the same time. We call the other states Center (C). In the problem, each agent can perceive only its own state, i.e., $S = \{\text{SG, C, B}\}$, and

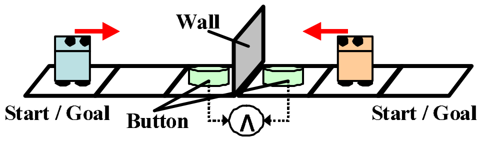

Figure 1: Example problem

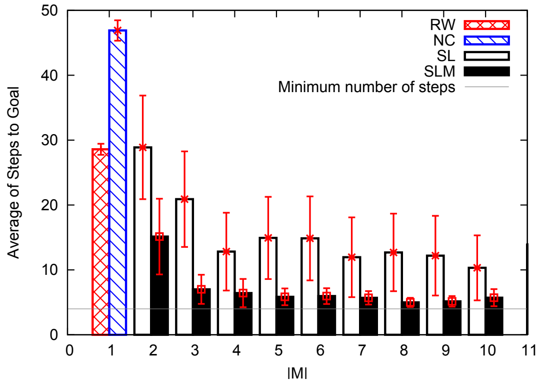

Figure 2: Comparisons of RW, NC, SL and SLM.

move forward or backward, i.e., A = {Fore, Back}. Note that each agent can neither know the state of the other agent by the wall nor remember whether the goal has been activated since the agent is oblivious. This problem is quite difficult to learn without communication, and also SL cannot yield deterministic optimal policies.

4. EXPERIMENTS

We carried out experiments for comparing SL and SLM, where $|M|$ is varied from 2 to 10. When $|M|$=1, since SL=SLM, we identify them as No Communication (NC). To verify the difficulty of our problem, we added the result of Random Walk (RW), which selects one action randomly in each time step.

We adopted Profit Sharing (PS) which has robustness on non-MDPs as a learning algorithm, and roulette strategy as action selection strategy. In PS, Q-value is updated by $Q(s_t, a_t) \leftarrow Q(s_t, a_t) + f(t, r, T)$ for each $s_t \in S$, $a_t \in A$ in a batch manner at the end of an episode (i.e., when both agents reach the goal), where r is a constant reward received only at the end, T is the final time step of the episode, and f is a *credit assignment function*. In these experiments, we use $f(t, r, T) = r \cdot \gamma^{T-t-1} / \log(t)$, where γ is a *discount rate parameter*. Here, we set $\gamma = 0.5$ and $r = 100$. The number of steps is limited to 100, i.e., if agents cannot reach the goal within 100 steps, the episode is regarded as invalid and restarted without updating Q-value.

We estimated the average number of steps to reach the goal in the last 100 episodes in 10,000 episodes in one trial. Fig. 2 shows the results averaged in 100 trials, as a bar chart with error bars, where each error bar represents the standard deviation of the corresponding bar.

5. DISCUSSION

Table 1: Percentage of successful trials

$\|M\|$	2	3	4	5	6	7	8	9	10
SL (%)	0	1	1	0	0	0	0	0	2
SLM (%)	31	34	42	47	41	45	60	54	48

Table 2: Deterministic optimal policy (M={1,2})

$S \times M$	(SG,1)	(SG,2)	(C,1)	(C,2)	(B,1)	(B,2)
π_a	Fore	Fore	Fore	Back	Back	Back
π_c	1	1	1	2	2	2

By comparing NC with SL, SL is clearly better than NC. It seems to be almost impossible to learn by NC. This shows that some beneficial meaning emerges in messages in M through the learning processes in SL. In other words, in SL, $\pi_c : S \to M$ probably allows each agent to include its own state in a message, while in NC, each agent can know neither the state of the other one nor the status of the button. We observed a strange fact that RW performed better than NC. The reason is that NC tends to reinforce only the action Back in the state C. By comparing SLM with SL, SLM is clearly better and more robust than SL. This means that SLM can allow each agent to include much more information in a message than SL.

In SLM, the messages should contain the information of the activation status of the goal. To verify this hypothesis, let us consider the experiments from the viewpoint of *optimal policy*. By using an optimal policy, both agents reach the goal with the minimum number of steps, which is 4 in our problem. We say a trial is *successful* if both agents reach the goal in 4 steps in the last episode. Table 1 shows the percentage of the successful trials in all 100 trials. As shown in the table, SLM has the ability to acquire an optimal policy. Actually, SLM can allow the agent to get a deterministic optimal policy. Table 2 shows a simplest example of the acquired optimal policies ($|M|$=2). The table shows that $\pi_c : S \times M \to M$ obviously allows each agent to include the activation status in a message, i.e., $1 \in M$ as inactivated and $2 \in M$ as activated.

The process to acquire an optimal policy in SLM can be regarded as the decomposition of POMDPs. In related work of POMDPs [3], the main approach is decomposition via *belief states*. Each belief state is a probability distribution of where the agent is. Our study leads to another meaningful approach in MARL, which is essentially different from the belief method, in the sense that in SLM, agents cooperate to solve unknown problems in limited information, while in the belief method, they do not cooperate explicitly.

6. CONCLUSION

In this paper, we empirically showed that the performance was improved dominantly by using SLM, which is an extension of SL. In addition, we confirmed that SLM has the ability to acquire a deterministic optimal policy, which cannot be achieved by SL.

7. REFERENCES

[1] D. Chakraborty and S. Sen. Computing effective communication policies in multiagent systems. In *AAMAS'07*, pages 153–155, 2007.

[2] T. Kasai, H. Tenmoto, and A. Kamiya. Learning of Communication Codes in Multi-Agent Reinforcement Learning Problem. In *Proc. of the 2008 IEEE Conference on Soft Computing in Industrial Applications (SMCia/08)*, pages 1–6, 2008.

[3] M. T. J. Spaan and N. Vlassis. Perseus: Randomized Point-based Value Iteration for POMDPs. *Journal of Artificial Intelligence Research*, 24:195–220, 2005.