We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

4,800
Open access books available

122,000
International authors and editors

135M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

CLARIVATE ANALYTICS BOOK CITATION INDEX INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

19

Implicit Estimation of Another's Intention Based on Modular Reinforcement Learning

Tadahiro Taniguchi[1], Kenji Ogawa[2] and Tetsuo Sawaragi[3]
[1]*College of Information Science and Engineering,*
[2]*Matsushita Electoronic Industrial Co.*
[3]*Graduate School of Engineering and Science,*
Japan

1. Introduction

When we try to accomplish a collaborative task, e.g., playing football or carrying large tables, we have to share a goal and a way of achieving the goal. Although people accomplish such tasks, achieveing such cooperation is not so easy in the context of a computational multi-agent learning system because participating agents cannot observe another person's intention directly. We cannot know directly what other participants intend to do and how they intend to achieve that. Therefore, we have to notice another participant's intention by utilizing other hints or information. In other words, we have to estimate another's intention to accomplish collaborative tasks.

In particular, in multi-agent reinforcement learning tasks, when another's intention is unobservable the learning process is fatally harmed. When a participating agent of a collaborative task changes its intention and switches or modifies its controller, system dynamics for each agent will inevitably change. If other agents learn on the basis of simple reinforcement learning architecture, they cannot keep up with changes in the task environment because most reinforcement learning architectures assume that environmental dynamics are fixed. To overcome the problem, each agent must have a simple reinforcement learning architecture and some additional capability, which solves the problem. We take the capability of "estimation of another's intention" as an example of such a capability.

Human beings can perform several kinds of collaborative tasks. This means that we have some computational skills, which enable us to estimate another's intention to some extent even if we cannot observe another's intention directly.

The computational model for implicit communication is described in this chapter on the basis of a framework of modular reinforcement learning. The computational model is called situation-sensitive reinforcement learning (SSRL), which is a type of modular reinforcement learning architecture. We assumed that such a distributed learning architecture would be essential for an autonomous agent to cope with a physically dynamic environment and a socially dynamic environment that included changes in another agent's intentions. The skill, estimation of another's intention, seems to be a social skill. However, human adaptability, which we believe our selves to be equipped with to deal with a physically dynamic environment, enables an agent to deal with such a dynamic social environment, including

Source: Machine Learning, Book edited by: Abdelhamid Mellouk and Abdennacer Chebira,
ISBN 978-3-902613-56-1, pp. 450, February 2009, I-Tech, Vienna, Austria

intentional changes of collaborators. Determining clearlify the computational relationship between the two skills is also a purpose of this study.
The mathematical basis for the implicit estimation of another's intention based on the framework of reinforcement learning is also provided. Furthermore, a simple truck-pushing task performed by a pair of agents is presented to evaluate the learning architecture.

2. Communication and estimation of another's intention

Communicating one's intention to another person enables the other person to estimate one's intention. Therefore, communication and estimation of another's intention are different aspects of the same phenomenon. Implicit estimation is a key idea to supplement the classical communication model, i.e., Shannon-Weaver communication model. Additionally, it is also important to understand a computational mechanism of emergence of communication.

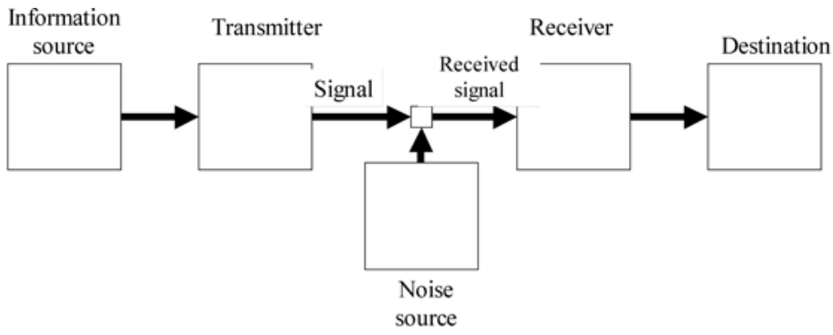


Fig. 1. Schematic diagram of general communication system

We describe the background in this section. In addition to that, an abstract mechanism of the implicit estimation is described on the basis of the notion of multiple internal models.

2.1 Communication models

Shannon formulated "communication" in mathematical terms [5]. In Shannon's communication model, a sender's messages encapsulated in signals or signs are carried through an information channel to a receiver. An encoder owned by the sender encodes the message to the signal by referring to its code table. When a receiver receives the signal, the receiver's decoder decodes the signal back to a message by referring to its code table. After that, the receiver understands the sender's intention and determines what to do. The general communication system described by Shannon is shown in Fig. 1 schematically.
In contrast to Shannon, Peirce, who started "semiotics," insisted that the basis of communication is symbols, and he defined a symbol as a triadic relationship among "sign," "object," and "interpretant"[2]. A "sign" is a signal that represents something to an interpreter. An "object" is something that is represented by the sign, and an "interpretant" is something that relates the sign to the object. In other words, an "interpretant" is a mediator between a "sign" and an "object." The words "sign" and "object" are easy for most people to understand. However, "interpretant" may be difficult to understand. An "interpretant" is sometimes a concept an interpreter comes up with, an action the interpreter takes, or culture in which people consider the sign and object to be related. The important point of Peirce's

semiotics is that the relationship between "sign" and "object" is not fixed. The relationship can be dynamically changing. The relationship simply depends on the "interpretant." The dynamic process by which a sign represents an object mediated by an interpretant is called "semiosis." Peirce's semiotics is thoroughly constructed from the viewpoint of an interpreter. In the framework of Peirce's semiotics, the third element, "interpretant," plays an essential role in communication. In Shannon's communication model, one premise is that a shared code table is required. However, an autonomous agent cannot observe other agents' internal goals or code table. In contrast, Peirce's semiosis does not require such a premise. Semiosis is a phenomenon that emerges inside of an autonomous agent. The participants in a communication must create meaning from incoming signs based on their physical and social experience. Such an individual learning process is considered to supplement symbolic communication. However, semiosis requires autonomous agents to have sufficient adaptability and capability to create meanings from superficial meaningless signs.

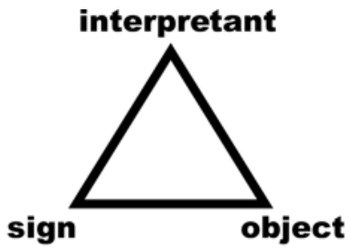


Fig. 2. Semiotic triad

In a human collaborative task, a human participant becomes able to distinguish several situations, which are modified by another's changing intentions. In such a case, the kind of policy the participant should follow in each situation is not clear beforehand. However, if the team continues to collaborate through trial and error, some kind of shared rules will be formed as a kind of habit of the team, and a follwer on the team becomes able to perform adequately by referring to the situation and the habit. This process corresponds to "semiosis" in Peirce's semiotics. Here, "sign," "object," and "interpretant" correspond to a "situation," "the leader's intention," and "acquired rule" or "the follower's action," respectively.

An important point in this scenario is that the "situation" has no meaning before the follower distinguishes situation, performs adequately, and a tacit rule is established between the two agents.

In this chapter, we describe candidates for computational communication models, which are based on Peirce's semiosis.

2.2 Estimation of another's intention

Roughly speaking, we assume there are two ways in which we estimate another's intention. Here, we explain the difference between the two ways of estimating another's intention.

For illustrative purposes, we assume that there is a leader in an organization who makes decisions. The leader makes decisions to direct the team, and followers play their roles based on the decision.

In such a case, the leader communicates his/her intention to the follwers, and followers in the organization have to estimate a leading agent's intention to cope with cooperative tasks.

The communication and the estimation of another's intention are different aspects of the same phenomenon, as we described above. How can followers members estimate the leader's intention? This is the problem.
Here, we take two kinds of estimation of another's intention into consideration. One is "explicit estimation," and the other is "implicit estimation."

2.2.1 Explicit estimation of another's intention

One solution for communicating one's intention to another person is to express one's intention directly with predefined signals, e.g., by pointing to the goal and by commanding the other person to act. The method of communication requires a shared symbolic system as a basic premise. The symbolic system is often called a code table. If the symbolic system used in this communication must be completely shared by the participants in the cooperativetask environment, a participant who receives a message understands exactly what the person transmitting the message wants to do. The receiver of the message can estimate the sender's intentions based on externalized signs. We call this process the "explicit estimation" because the intention of the leader is explicitly expressed as externalized signals. In this communication model, both agents have to share a predefined code table before the tasks. In the explicit estimation model, the accuracy of the communication is measured by the coincidence between the transmitted message and the receiver's interpretation of the sender's message, which is obtained by decoding the incoming signal utilizing the shared code table. The process of estimating another's intention in a collaborative task is shown in Figure 3 schematically. A leader and a follower carry a truck collaboratively. How can the follower estimate the leader's goal using explicit estimation when the leader changes his goal?

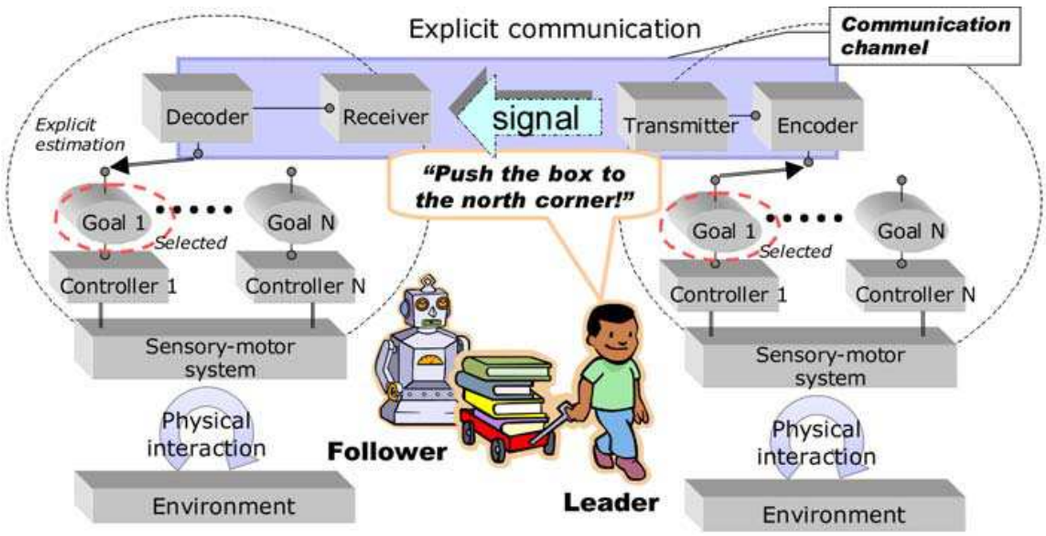


Fig. 3. explicit estimation

First, the leader agent changes his goal. In the explicit estimation scheme, this seems like a natural framework of communication. After Shannon formulated "communication" mathematically, many sociologists and computer scientists have described "communication" as above. However, the communication model based on explicit estimation of another's

intention has two shortcomings. One is that the method of sharing the code table between the two agetns is unknown. If we consider the two agents to be autonomous, neither agent can observe the other agent's internal goals and code table. Therefore, neither agent can utilize a "teacher signal" as feedback of its interpretation to upgrade its code table. The second shortcoming is that the leader agent has to display his intention whenever he changes his goal. These are two problems of explicit estimation of another's intention.

In contrast, when we review what we do in collaborative tasks, we find that we do not always send verbal messages representing our intention to our collaborators. We sometimes execute a collaborative task without saying anything. In this case, the leader's intention is not transmitted to the follower by sending the explicit linguistic sign but through the shared environmental dynamics implicitly. Explicit estimation of another's intention is not the only way of communication. To complement or to support the explicit estimation, implicit estimation is necessary.

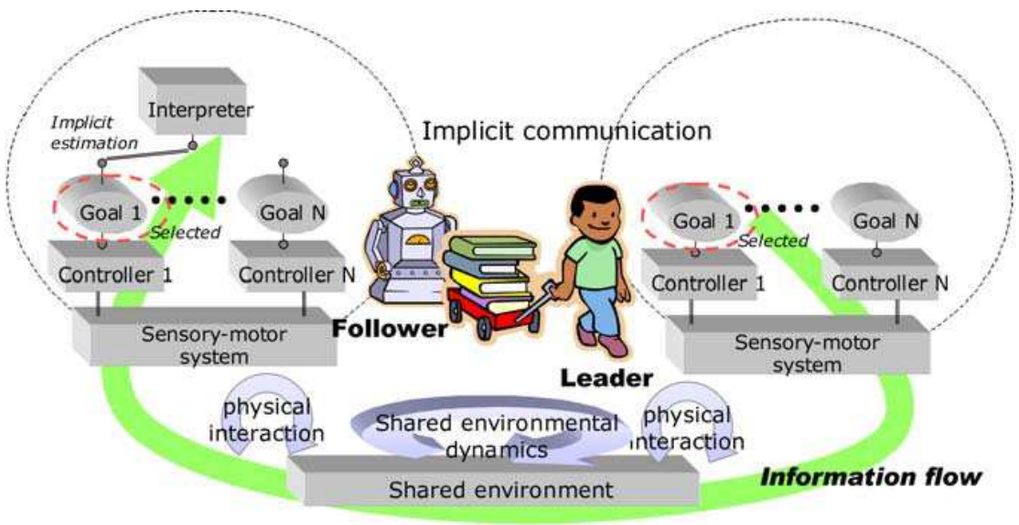


Fig. 4. Implicit communication

2.2.2 Implicit estimation of another's intention

People occasionally undertake collaborative tasks without saying anything. Even if a leader says nothing to members of his organization, they can often perform the task by estimating the leader's intentions on the basis of their observation. We call such an estimation process "implicit estimation" of another's intention. However, if there were no pathways through which information about the leader's intention goes to the followers, the followers could never estimate the leader's intention. One reason followers can estimate the leader's intention is that the action and sensation of the followers are causally related to the leader's intentions.

In other words, sensations a participating agent has after he/she performs actions are affected by the leader's way of acting and another agents' ways of acting. Therefore, subjective environmental dynamics for a participating agent are causally affected by the leader's intention because other agents are assumed to behave based on the leader's intention.

We assume none of the members can observe any information except for their own sensory-motor information. However, they can estimate the leader's intentions. We call this process "implicit estimation." Implicit estimation is achieved by watching how the agent's sensation changes. In control tasks, an agent usually observes state variables.
In what follows, we assume that an agent obtains state variables, e.g., position, velocity, and angle. State variables are usually considered to be objectives to be controlled in many control tasks. However, in implicit communication, state variables also become information media of another agent's intention. An participating agent can estimate another's intention by observing changes in state variables. The information goes through their shared dynamics.
The process of implicit estimation of another's intention is showen in Figure 4, schematically. First, the leader changes his goal. When the leader's goal has changed, his controller, which produces his behavior, is switched. That, of course, affects physical dynamics of the dynamical system shared between the leader and the follower. If a participating agent has a state predictor, he will become aware of the qualitative change in the shared dynamics because his prediction of the state value collapses If physical dynamics are stable, he can predict his state variables consistently. If the follower agent notices the change in subjective physical dynamics, the follower can notice the change in the leader's intention based on the causal relationship between the leader's intention and his facing dynamical system.
Therefore, the capability to predict state variables seems to be required for physical skills and social skills. This scenario suggests the process of learning physical skills to control the target system and the method to communicate with the partner agent might be quite similar in such cooperative tasks.

3. Multiple internal models

Our computational model of implicit estimation of another's intention is based on modular reinforcement learning architecture including multiple internal models. To achieve implicit estimation of another's intention described in the previous section, an agent must have a learning architecture that includes state predictors. We focus on multiple internal models as neural architectures that achieve such an adaptive capability.

3.1 Multiple internal models and social adaptability

Relationships between the human brain's social capability and physical capability are commanding interest. From the viewpoint of computational neuroscience, Wolpert et al. [17, 3] suggested that MOSAIC, which is a modular learning architecture representing a part of the human central nervous system (CNS), acquires multiple internal models that play an essential role in adapting to the physical dynamic environment as well as other roles. We regard this as a candidate for a brain function that connects human physical capability and social capability. An internal model is a learning architecture that predicts the state transition of the environment or other target system. This is a belief that a person can operate his/her body and his/her grasping tool by utilizing an obtained internal model[16]. The internal model is acquired in the cerebellum through interactions. The learning system of internal models is considered to be a kind of schema that assimilates exterior dynamics and accommodates the internal memory system, i.e., internal model. If a person encounters various kinds of environments and/or tools, which have different dynamical properties, the

human brain needs to differentiate them and acquire several internal models. However, segmentation of dynamics is not given a priori. Therefore, a learning architecture representing multiple internal models should generate and learn internal models, and recognize changes in physical dynamics in its facing environment at the same time. To describe such a learning system, several computational models have been proposed, e.g., MPFIM [17], the mixture of RNNs[10], RNNPB[9], and the schema model [13]. Most of them are comprised of several learning predictors. The learning architecture switches the predictors and accommodates them through interactions with the environment. Such a learning architecture is often called a modular learning system. The RNNPB is not a modular learning system. Tani insisted internal models should be obtained in a single neural network in a distributed way[9]. In most modular learning architecture, a Bayesian rule is used to calculate the posterior probability in which a current predictor is selected. In contrast, the schema model [13] is a modular learning architecture that does not use a Bayesian rule but hypothesis-testing theory. At the moment, multiple internal models are usually considered to be a learning system for an autonomous system to cope with a physically dynamic environment. Meanwhile, Wolpert et al. addressed a hypothesis that a person utilizes multiple internal models to estimate another's intention from the observation of another's movement. Although these internal models described in the hypothesis seem to add a slightly different feature to the original definition of an internal model, interestingly, the hypothesis tries to connect neural architectures for physical adaptability and social adaptability. Doya et al. [1] proposed a modular learning architecture that enables robots to estimate another's intention and to communicate with each other in a reinforcement learning task.

In addition, when a person performs a collaborative task with others, one can notice changes in another agent's intention by recognizing the change in his/her facing dynamical system without any direct observation of the other agent's movement. This means multiple internal models enable an agent to notice changes in another agent's intention. This usage of multiple internal models does not require adding any features to the original definition of multiple internal models.

3.2 Implicit estimation of another's intention based on multiple internal models

"Intention" in everyday language denotes a number of meanings. Therefore, a perfect computational definition of "intention" is impossible. In this chapter, we simply consider an "intention" as a goal the agent is trying to achieve. In the framework of reinforcement learning, an agent's goal is represented by a reward function. Therefore, an agent who has several intentions has several internal goals, i.e., several internal reward functions, G^m. If an internal reward function, G^m, is selected, a policy, u^m, is selected and modified to maximize the cumulative future internal reward through interactions with the task environment.

In the following, we assume that the collaborative task involves two agents. The system is described as

$$y = f(x, u_1, u_2^m) + n, \tag{1}$$

$$= f(x, u_1, u_2^m(x)) + n, and \tag{2}$$

$$= F^m(x, u_1) + n. \tag{3}$$

Here, x is a state variable, u_i is the i-th agent's motor output, and n is a noise term. We assumed that an agent would not be able to observe another agent's motor output directly. In such cases, environmental dynamics seem to be Eq. 3 to the first agent. If the second agent changes its policy, environmental dynamics for the first agent change. Therefore, in a physically stationary environment, the first agent can establish that the second agent has changed its intention by noticing changes in environmental dynamics.

The discussion can be summarized as follows. If physical environmental dynamics, f, is fixed, agents who have multiple internal models can detect changes in another agent's intentions by detecting changes in subjective environmental dynamics, F. The computational process is equal to the process by which an agent detects changes in the original physical dynamics.

We define "situation" as "how state variable x and motor output u change observed output y." In this case, a change in an agent's intentions leads to a change in the subjective situation of another agent. By utilizing multiple internal models, an agent is expected to differentiate situations and execute adequate actions. In the next section, we describe a concrete modular reinforcement learning architecture named Situation-Sensitive Reinforcement Learning (SSRL).

4. Situation-sensitive reinforcement learning architecture

It is important for autonomous agents to accumulate the results of adaptation to various environments to cope with dynamically changing environments. Acquired concepts, models, and policies should be stored for similar situations that are expected to occur in the near future. Not only learning a certain behavior and/or a certain model, but also the obtained behaviors, policies, and models is essential to describe such a learning process. Many modular learning architectures [7, 4] and hierarchical learning architectures [10, 8] have been proposed to describe this kind of learning process. This section introduces such a modular-learning architecture called the situation-sensitive reinforcement learning architecture (SSRL). This enables an autonomous agent to distinguish changes the agent is facing in situations, and to infer the partner agent's intentions without any teacher signals from the partner.

4.1 Discrimination of intentions based on changes in dynamics

Fig. 5 is an overview of SSRL. SSRL has several state predictors, F^m, representing situations and internal goals, G^m, representing intentions. Each state predictor F^m corresponds to each situation.

$$e_t^j = ||y_t - F^j(x_t, u_t)||^2, \tag{4}$$

$$P(j|\bar{e}_t^j) = \exp(-\frac{\bar{e}_t^j}{2\sigma^2}) / \sum_{k=1}^{p} \exp(-\frac{\bar{e}_t^k}{2\sigma^2}), and \tag{5}$$

$$j^* = \arg\min_j P(j|\bar{e}_t^j), \tag{6}$$

where $\bar{e}_t^j$ is the temporal average of the prediction error, e_t^j, of the j-th state predictor, F_j. If averaged error $\bar{e}_t^j$ has a normal distribution and the system dynamics is F_j, the posterior probability, $P(j \mid \bar{e}_t^j)$, can be defined based on the Bayesian framework above under the condition that there is no other information. If there are no adequate state predictors in SSRL, the SSRL allocates one more state predictor based on hypothesis-testing theory [13].

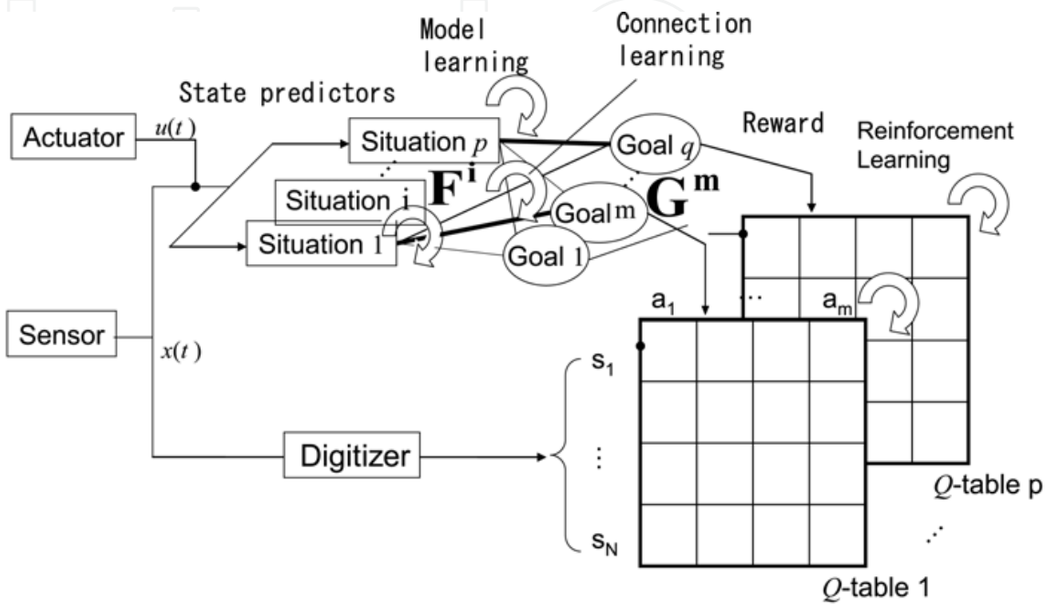


Fig. 5. Situation-Sensitive Reinforcement Learning architecture

We model the state predictors by using locally linear predictors, and we don't estimate the standard deviation σ. The updating rule are switched based on hypothesis testing.

Case 1: $\bar{e}_t^{j^*} < \delta_l$

In this case, the learning system considers that incoming sample data are normal samples for the existing predictors, decides the curret situation j^*, and update the corresponding function F^{j^*} by using assimilated samples.

Case 2: $\bar{e}_t^{j^*} > \delta_u$

In this case, the learning system considers that incoming sample data are outliers for the existing predictors, and prepare a new fnction F^{p+1}. It decides the curret situation j^{p+1}. However, the new predictor is considered as a exeptional state predictor until $\bar{e}_t^{p+1} < \delta_l$. If the predictor's averaged error reaches under δ_l, the function F^{p+1} is taken into a list of existing predictors, and $p \leftarrow p + 1$.

Case 3: $\delta_l < \bar{e}_t^{j^*} < \delta_u$

In this case, the system take no account of the incoming sample.

This is an intermediate method for the MOSAIC model [17, 15], which is based on the Basian framework, and the schema model [13], which is based on hypothesis-testing theory. SSRL detects the current situation based on Eq. 6. During this an adequate state predictor is selected and assimilates the incoming experiences; SSRL acquires the state predictors by ridge regression based on the assimilated experiences.

4.2 Reinforcement learning

Each policy corresponding to a goal is acquired by using reinforcement learning [6]. SSRL uses *Q*-Learning [14] in this paper. This method can be used to estimate the state-action value function, $Q(s, a)$, through interactions with the agent's environment. The optimal state-action value function directly gives the optimal policy. When we define $\mathcal{S}$ as a set of state variables and $\mathcal{A}$ as a set of motor outputs, and we assume the environment consists of a Markov decision process, the algorithm for *Q*-learning is described as

$$\begin{aligned} Q &\leftarrow Q(s,a) + \alpha(r + \gamma V(s') - Q(s,a)), \\ V(s') &= \max_{a' \in \mathcal{A}} Q(s', a'), and \end{aligned} \tag{7}$$

$$u(s) = \underset{a' \in \mathcal{A}}{\operatorname{argmax}}\, Q(s,a), \tag{8}$$

where $s \in \mathcal{S}$ is a state variable, $a \in \mathcal{A}$ is a motor output, $r(s, a)$ is a reward, and s' is a state variable at the next time step. In these equations, α is the learning rate and γ is a discount factor. After an adequate Q is acquired,the agent can utilize an optimal policy, u, as in Eq. 8. Boltzmann selection is employed during the learning phase.

$$p(a|s) = exp(\beta Q(s,a)) / \sum_{a}^{\prime} (\beta Q(s,a')) \tag{9}$$

4.3 Switching architecture of internal goals

An agent can detect changes in the other agent's intentions by distinguishing between situations he/she faces. However, the goals themselves cannot be estimated even if switching between several goals can be detected. Here, we describe a learning method, which enables an agent to estimate the another's intentions implicitly. The method requires three assumptions to be made.

A1 Physical environmental dynamics *f* do not change.
A2 Every internal goal is equally difficult to achieve.
A3 The leader agent always selects each optimal policy for each intention.

The mathematical explanation for these assumptions will be described in the next section. We employes Boltzmann selection for internal goal switch. The rule to select the internal goals are described as

$$p(m|j) = \exp(Bw_{jm}) / \sum_{i=1}^{q} \exp(Bw_{ji}), \tag{10}$$

where $p(m|j)$ is the probability that G^m will be selected under situation, F^j, and B is the inverse temperature. The network connection, w_{jm}, between the current situation, F^j, and the current internal goal, G^m, is modified by the sum of the obtained reward, R_t^{jm}, during a certain period during the *t*-th trial, i.e.,

$$w_{jm} = \nu R_t^{jm} + (1-\nu) w_{jm} \tag{11}$$

Here, ν is the learning rate of the internal goal switching module. Eq. 11 shows that connection w_{jm} becomes strong if internal goal G^m is more easy to accomplish when the situation is F^j. Eq. 10 shows that an internal goal is more likely to be selected if its network connection is stronger than the other's. The abstract figure for the switching module is shown in Fig. 6. If the learning process for the switching architecture of internal goals is preceded and converged, a certain internal goal corresponding to a situation is selected.

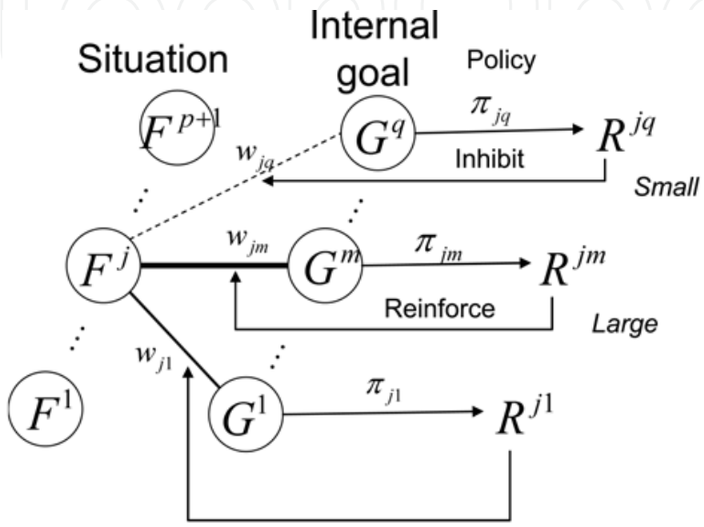


Fig. 6. Internal goal switching module

4.4 Mathematical basis for internal goal switching module

This section provides the mathematical basis for the learning rule ofr the implicit communication. First, the Bellman equation for the *i*-th (*i* = 1, 2) agent of a system involving two agents are described as[1].

$$V_i^{\lambda}(x)|_{u_j} = \max_{u_i \in U_i} \sum_{x'} P(x'|x, u_i, u_j) \times [G^{\lambda}(x, x') + \gamma V_i^{\lambda}(x')], \tag{12}$$

where G^{λ} is a reward function for the λ-th goal, u_i is the *i*-th agent's motor output, and x' is the x in the next step. G^{λ} in this framework is not assumed to have motor outputs as variables of the function. The optimal value function for the *i*-th agent depends on the other agent's policy, u_j. Here, we define u_i^{λ} as the *i*-th agent's policy that maximizes the *j*-th agent's maximized value function whose goal is G^{λ}.

$$u_i^{\lambda} = \operatorname*{argmax}_{u_i \in U_i} \max_{u_j \in U_j} \sum_{x'} P(x'|x, u_i, u_j) \times [G^{\lambda}(x, x') + \gamma V_i^{\lambda}(x')] and \tag{13}$$

$$V_i^{\lambda|\nu} \equiv V_i^{\lambda}|_{u_j^{\nu}}. \tag{14}$$

[1] In this section, we have assumed $i \neq j$ without making any remarks.

The assumptions, A2 and A3, we made in the previous section can be translated into the following,

A'2 : We assumed the *j*-th agent would use the controller, u_j^{λ}, and

A'3 : $V_i^{\lambda|\lambda}(x_0) = V_i^{\nu|\nu}(x_0),$

where x_0 is the initial point of the task. The following relationship can easily be derived from the definition.

$$V_i^{\lambda|\lambda} = V_i^{\nu|\nu} \geq V_i^{\nu|\lambda}. \tag{15}$$

Therefore, the *i*-th agent's internal goal becomes the same as *j*-th agent's goal, if the *i*-th agent select a reward function that maximizes the value function under the condition that the *j*-th agent uses controller u_j^{λ}. When the initial point is not fixed, $V_i(x_0)$ is substituted by the averaged cumulative sum of rewards the *i−th* agent obtains, who starts the task around the initial point, x_0. This leads us to the algorithm eq.11.

5. Experiment

We evaluate SSRL in this section. To fulfill all the assumptions made in Section 4 completely is difficult in a realistic task environment. The task described in this section roughly satisfies the assumptions, A'2 and A'3.

5.1 Conditions

We applied the proposed method to the truck-pushing task shown in Fig. 7. Two agents in the task environment, "Leader" and "Follower,"cooperatively push a truck to various locations. Both agents can adjust the truck's velocity and the angle of the handle. However, a single agent cannot achieve the task alone because its control force is limited. In addition, the Leader has all fixed policies for all sub-goals beforehand, and holds a stake in deciding the next goal. However, the agents cannot communicate with each other. Therefore, the agents cannot "explicitly" communicate their intentions. The Follower perceives situation F^j by using SSRL, changes its internal goal G^m based on the situation, and learns how to achieve the collaborative task. The two agents output the angle of the handle, θ_L, θ_F, and the wheel's rotating speed, ω_L, ω_F. Here , the final motor output to the truck, θ, ω, is defined as

$$\theta = K_\theta(\theta_L + \theta_F) and \tag{16}$$

$$\omega = K_\omega(\omega_L + \omega_F), \tag{17}$$

where K_θ and K_ω are the gain parameters of the truck. K_θ and K_ω were set to 0.5 in this experiment. The Leader's controller was designed to approximately satisfy the assumptions in Section 3. The controller in this experiment was a simple PD controller. The Follower's state, *s*, was defined as $s = [\rho, \alpha]$. The state space was digitized into 10 × 8 parts. The action space was defined as $\theta_F = \{-\pi/4, -\pi/8, 0, \pi/8, \pi/4\}$ and $\omega_F = \{0.0, 3.0\}$. As a result of the two agents' actions, the truck's angular velocity, Ω, was observed by the Follower agent. Ω, θ, and ω have a relationship of

$$\Omega \propto \omega \tan\theta. \tag{18}$$

The agents can carry the truck to a certain goal by cooperatively controlling Ω. The main state variables are shown in Fig. 8. Internal reward function G^m is defined as

$$G^m(x) = \begin{cases} 5 & \text{if } ||C - Goal_m|| < 1 \\ \kappa(1 - ||C - Goal_m||) & \text{otherwise,} \end{cases} \tag{19}$$

where C is the position of the truck, and $Goal_m$ is the position of the m-th goal.

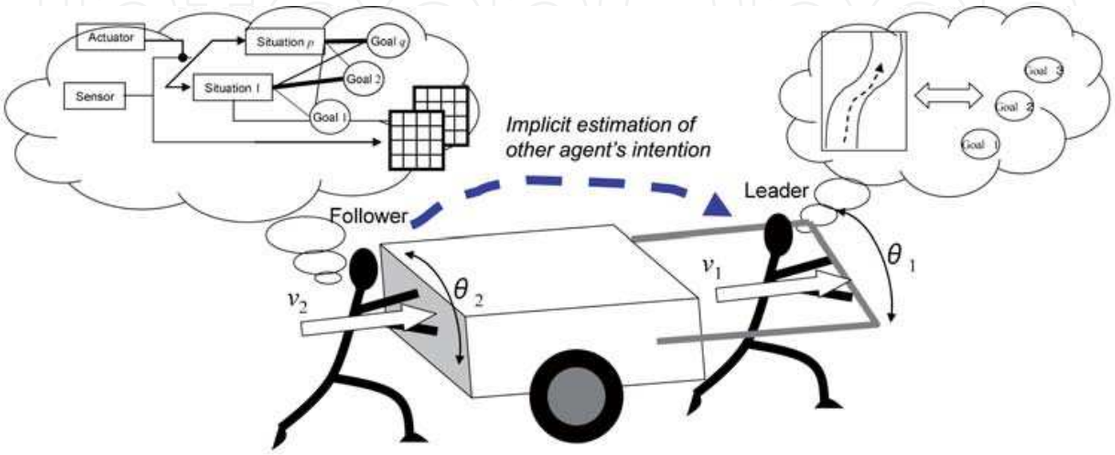


Fig. 7. Simple truck-pushing task by pair of agents

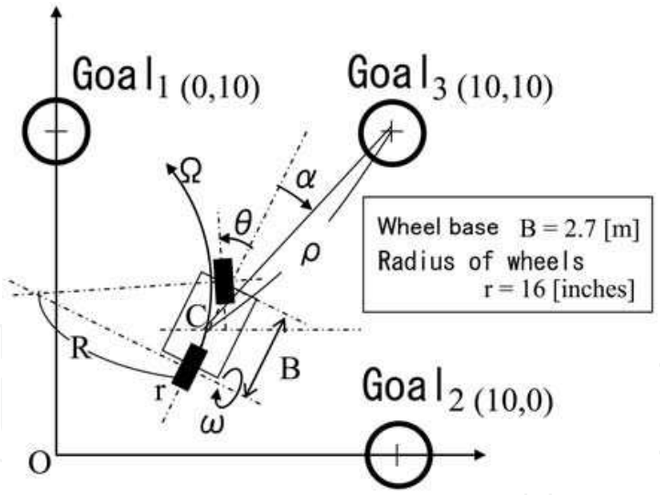


Fig. 8. State variables and parameters in task environment

5.2 Experiment 1: implicit estimation of another's intention

Wwe fisrt conducted an experiment in which the Follower estimated the Leader's goal, where the Leader selected one of three sub-goals, and learned how to achieve the collaborative task (Fig. 9, top). There were three goals, and the Leader changed its goals from $G_1 -> G_2 -> G_3$ alternately every 1000 trials.

In contrast to simple reinforcement learning, the Follower agent not only has to learn the policies for the goals but also the state for predictors the relationship between the current situation and the internal goal by updating these parameters.

The 1000 trajectories of the truck corresponding to all 1000 trials in this experiment are shown in Figs. 10 and 11. Simple Q-learning with explicitly given internal goals and SSRL are compared. Fig. 10 shows the results obtained from the experiment using Q-learning, and Fig. 11 shows those from the experiment using SSRL. The task success rate is indicated in each figure. The red curves represent the trajectories for the team that reached the goal, and the gray curves represent the trajectories for the team that did not reach the goal. This shows that simple Q-learning achieves a single task. However, the Follower could not coordinate with the Leader agent after it had changed its goal because it could not discover the Leader agent's intentions. SSRL performs better when the Leader changes its intentions. Fig.13 shows that three predictors were generated that discover the Leader's intentions. Furthermore, Fig. 12 shows that appropriate internal goals were selected inside the Follower agent.

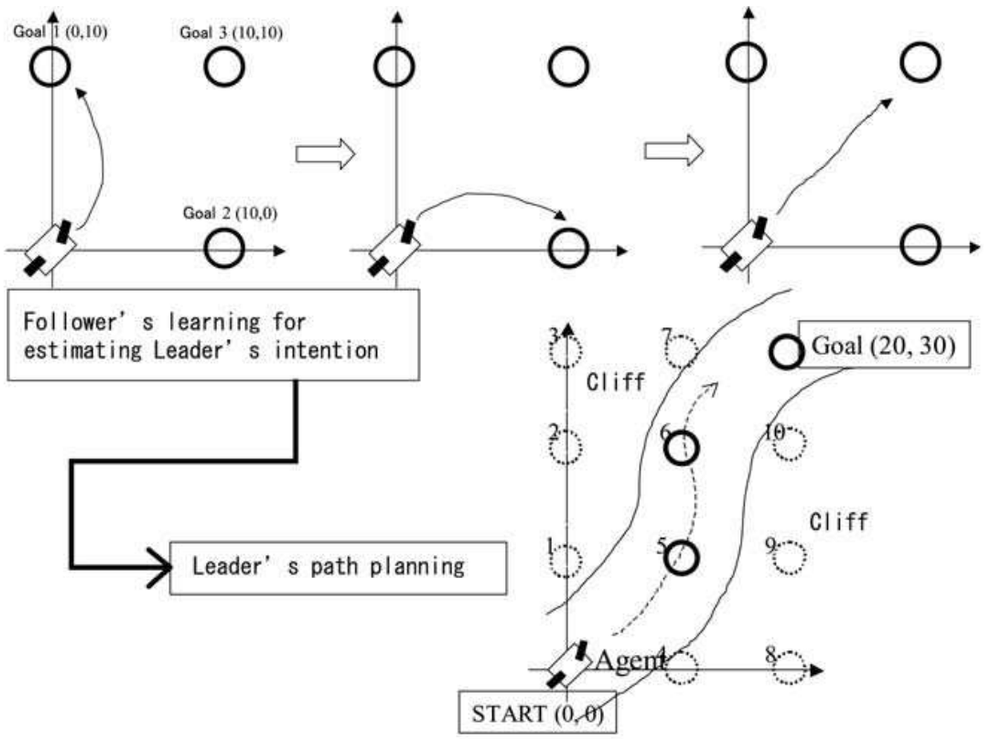


Fig. 9. Top: cooperative action is acquired by Follower, bottom: plan toward the goal is acquired by Leader

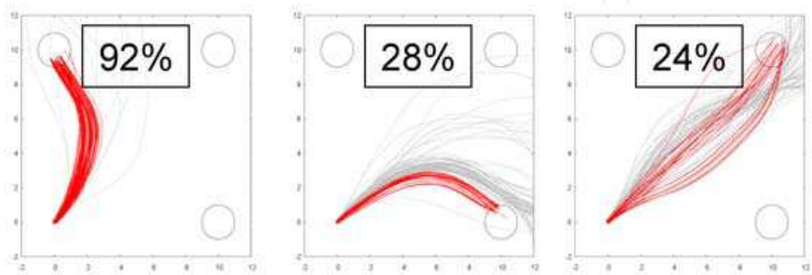


Fig. 10. Behaviors of truck at Follower's learning stage with single *Q*-table and internal goal-switching module without state predictors

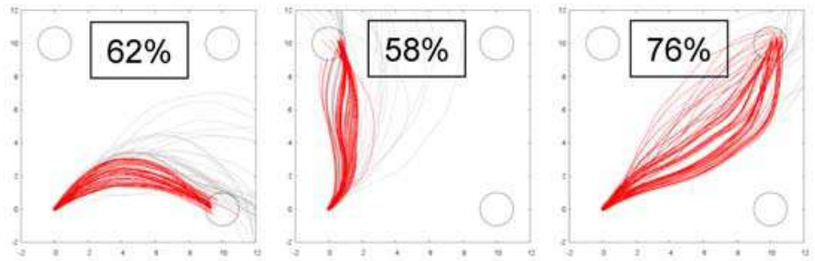


Fig. 11. Behaviors of truck at Follower's learning stage with SSRL

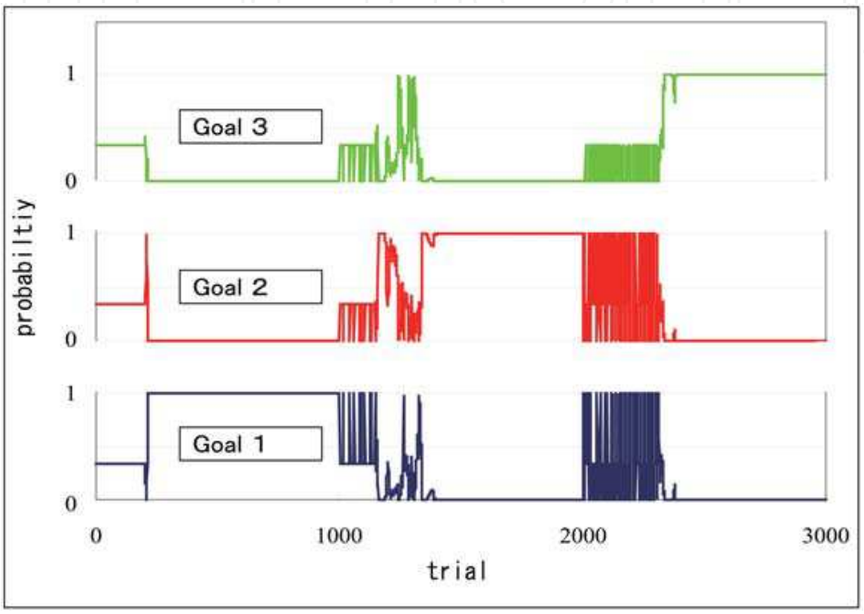


Fig. 12. Time course of probabilities where *m*-th internal goal is selected

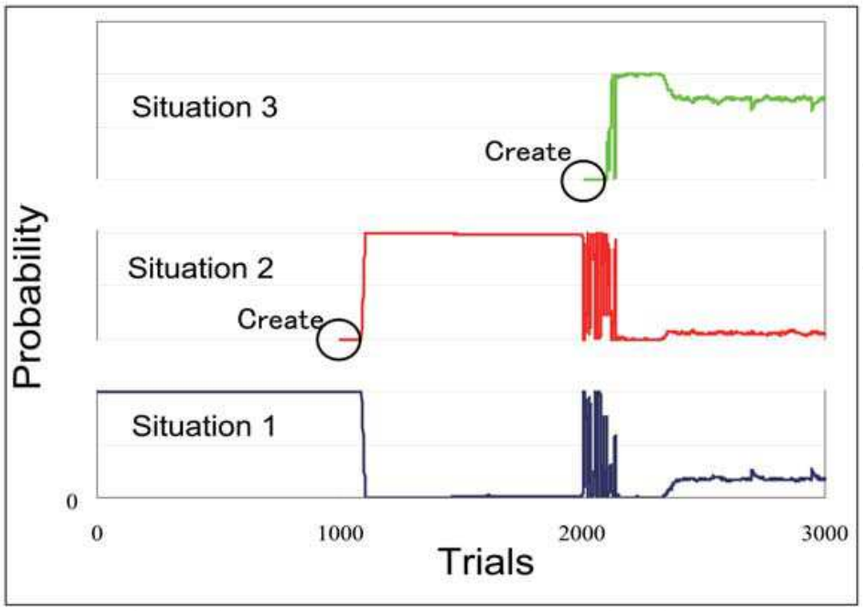


Fig. 13. Time course of probabilities that environment being faced is the *i*-th situation

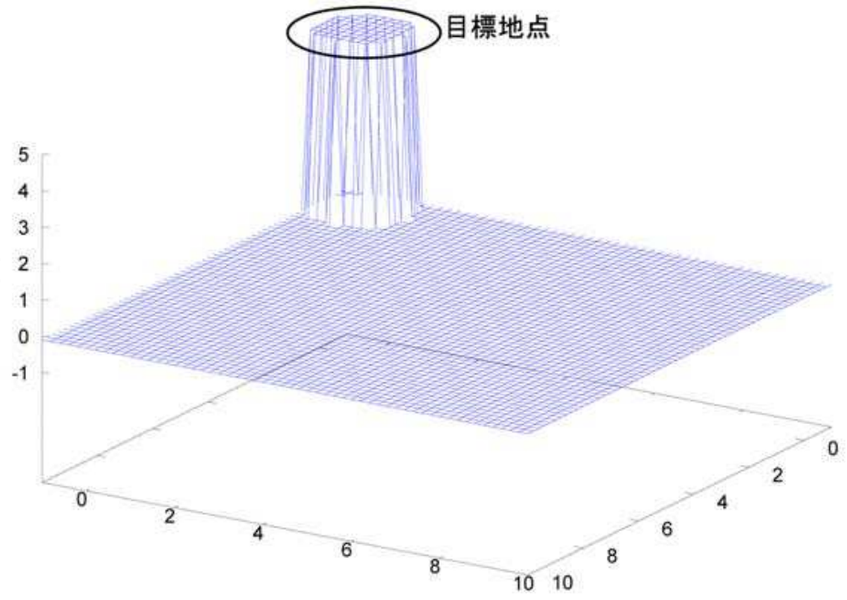


Fig. 14. Reward function for Follower's internal goals

These results show that SSRL enabled the Follower to implicitly estimate the Leader's intention.

5.3 Experiment 2: sequential collaborative task

After the follower had acquired the ability to implicitly estimate the leader's intentions, the next experiment was carried out. The experimental environment is shown at the bottom of Fig. 9. The task required the agents to go through several checkpoints (sub-goals), and reach the final goal. The Follower in the next experiment exploited the SSRL acquired through Experiment 1, and the Leader explored and planed the path to the final goal. The Leader agent can chose the next sub-goal out of three check points that correspond to three goals in Experiment 1, i.e., "up," "upper right," and "right," from the current checkpoint as shown in Fig. 9. There are also two "cliffs" in this task environment. If the truck enters the cliffs, it can no longer move. The Leader learned the path to the final goal by using a simple *Q*-learning. The reward function for the Leader is shown in Fig. 15. Two kinds of Follower agents are compared in this experiment. The first has a single Q-learning architecture and a perfect internal goal switch. The second has SSRL.

Fig. 16 shows the results for the experiment using simple Q-learning. Fig. 17 shows the results for the experiment using SSRL. Fig. 18 shows the success rate representing the probability that the team will finally reach the final goal. The results reveal that the team whose Follower agent could not discriminate the Leader's intentions performed worse than the team whose Follower agent could distinguish the Leader's intentions. Without such a distributed memory system like SSRL, the Follower would not be able to up with in the Leader's intentions. In addition to disadvantage, the poor performance of the Follower agent adversely affects the Leader's learning process. However, the Follower with SSRL could estimate the Leader's intentions and keep up with the Leader's plans although there was no explicit communication between the two agents.

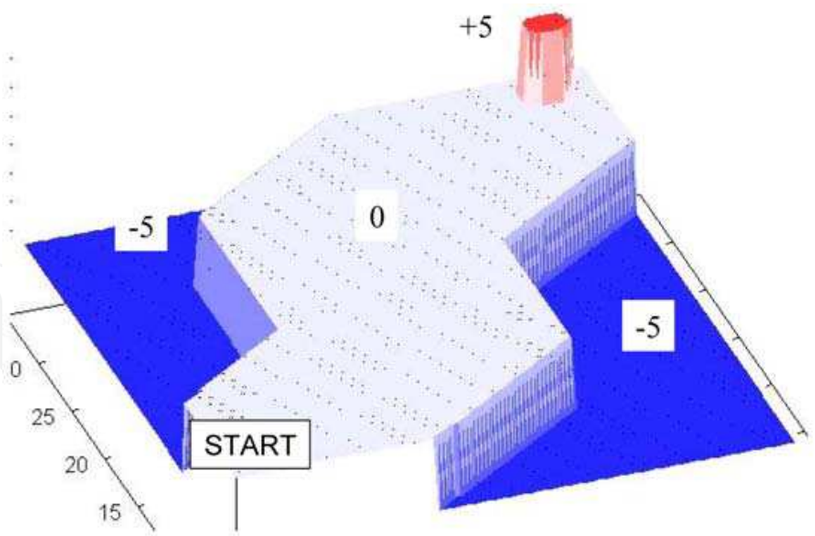


Fig. 15. Reward function for Leader agent for planning path

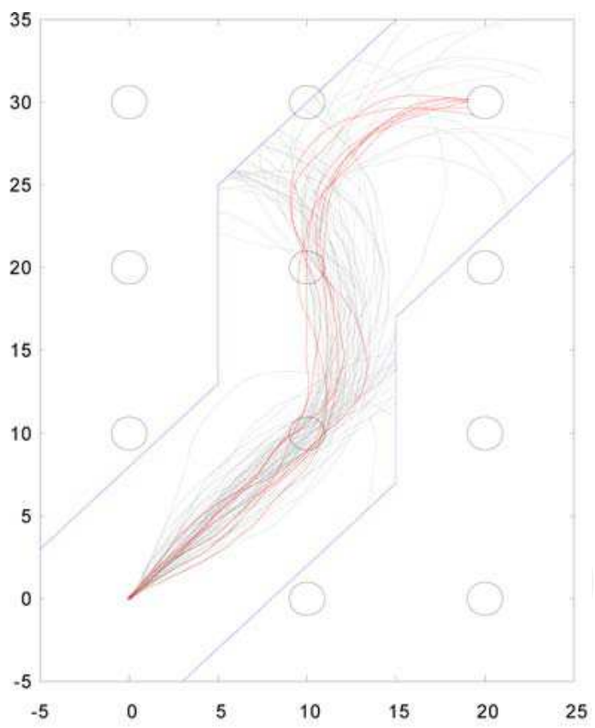


Fig. 16. Behaviors of truck at Leader's learning stage with single *Q*-table and internal goal switching module without Situation Recognizer

However, the success rate for the collaborative task saturated at about 40%. The reason for this is that the Follower notices changes in the Leader's intentions after these changes have sufficiently affected the state variables. The delay until the Follower becomes aware of the changes is sometimes critical, and the truck occasionally fell into the cliffs. To estimate the other's intentions without any explicit signs outside the state variables, the information has

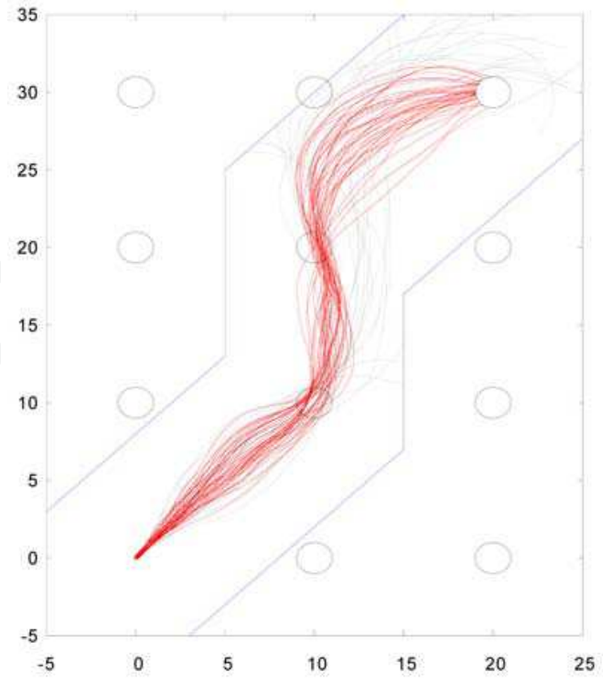


Fig. 17. Behaviors of truck at Leader's learning stage with SSRL

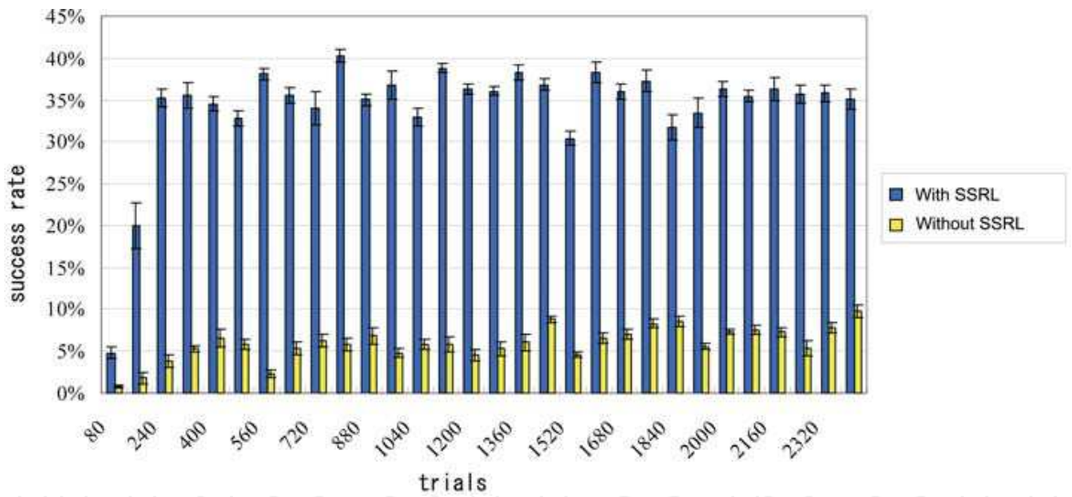


Fig. 18. Success rate for cooperative task

to be embedded in the state variables, which are the objectives of the team's control task. Our results suggest that it is not impossible to implicitly estimate the other's intentions, but it is important to have a communication channel whose variables are not related to the state variables, which are the objectives of the task, e.g., voice, colar sign, or marker. This must be the reason why we use explicit sign in collaborative tasks. As we mentioned, the "implicit estimation" must back up and complement "explicit estimation." "Explicit estimation" must be faster and better than "implicit estimation" as far as a code table was shared in a team. However, this does not mean "explicit estimation" is superior to"implicit estimation." They are complementary architectures.

6. Conclusion

We described a framework for implicitly estimating another's intentions based on modular reinforcement learning. We applied the framework to a truckpushing task by two agents as a concrete example. In the experiment, the Follower agent could perceive changes in the Leader's intentions and estimate his intentions without observing any explicit signs on any action outputs from the Leader. This demonstrated that autonomous agents can cooperatively achieve a task without any explicit communication. Self-enclosed autonomous agents can indirectly perceive the other's changes in intentions from changes in their surrounding environment. It is revealed that multiple internal models help an autonomous agent to achieve collaborative task.

In the context of artificial intelligence, "symbol grounding problem" is considered as an important problem. The problem deals with how robots and people can relate their symbolic system to their physical and embodied experiences. The symbolic system mentioned here is also used in communication, usually. Takamuku et al. presented a system for lexicon acquisition through behavior learning which is based on a modified multi-module reinforcement. The robot in their work is able to automatically associate words to objects with various visual features based on similarities in features of dynamics[8]. At the same time, Taniguchi et al. described an integrative learning architecture for spike timing-dependent plasticity (STDP) and the reinforcement learning schemata model (RLSM) [12, 11]. The learning architecture enables an autonomous robot to acquire behavioral concepts and signs representing the situation where the robot should initiate the behavior. They called this process "symbol emergence." The symbolic system plays a important role in human social communication.They also utilize modular learning architecture to describe the process of symbol organization. However, they treat bottomup organization of "explicit symbols," which is assumed to be used explicit communication.

In many researches, "symbolic communication" means exchanging discrete signals. However, the essential point of symbolic communication is not such an externalized signs, but an adaptive formation of "interpretant" from the viewpoint of Peirce's semiotics. Therefore, we focus on the implicit communication and its bottom-up process of organization.

However, the system we treated in this chapter is constrained to some extent. This framework for implicit estimates does not always work well. If the system does not satisfy the assumptions made in Section 4, the framework is not guaranteed to work. The Leader's policies are fixed when the Follower agent is learning its policies, predictors, and network connections in our framework. The model described in this chapter may not work in the simultaneous multi-agent reinforcement learning environment. We intend to take these into account in future work.

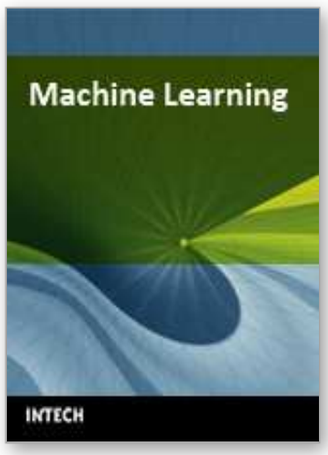
7. References

[1] K. Doya, N. Sugimoto, D. Wolpert, and M. Kawato. Selecting optimal behaviors based on context. *International symposium on emergent mechanisms of communication*, 2003.

[2] Charles Hartshorne, Paul Weiss, and Arthur W. Burks, editors. *Collected Papers of Charles Sanders Peirce*. Thoemmes Pr, 4 1997.

[3] M. Haruno, D.M.Wolpert, and M. Kawato. Mosaic model for sensorimotor learning and control. *Neural Computation*, 13:2201–2220, 2001.

[4] K. Murphy. Learning switching kalman-filter models. *Compaq Cambridge Research Lab Tech Report*, pages 98–10, 1998.
[5] C.E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*,, 27:379–423, and 623–656, 1948.
[6] R. Sutton and A.G. Barto. *Reinforcement Learning : An Introduction*. The MIT Press, 1998.
[7] Y. Takahashi et al. Modular learning syatem and scheduling for behavior acquisition in multi-agent environment. In *RoboCup 2004 Symposium papers and team description papers, CD-ROM*, 2004.
[8] Shinya Takamuku, Yasutake Takahashi, and Minoru Asada. Lexicon acquisition based on object-oriented behavior learning. *Advanced Robotics*, 20(10):1127–1145, 2006.
[9] J. Tani, M. Ito, and Y. Sugita. Seif-organization of distributedly represented mulyiple behavior schemata in a mirror system:reviews of robots using rnnpb. *Neural Networks*,, 17:1273–1289, 2004.
[10] J. Tani and S. Nolfi. Learning to perceive the world as articulated: an approach for hierarchical learning in sensory-motor systems. *Neural Networks*, 12:1131–1141, 1999.
[11] T. Taniguchi and T. Sawaragi. Symbol emergence by combining a reinforcement learning schema model with asymmetric synaptic plasticity. In *5th International Conference on Development and Learning*, 2006.
[12] T. Taniguchi and T. Sawaragi. Incremental acquisition of behaviors and signs based on a reinforcement learning schemata model and a spike timing-dependent plasticity network. *Advanced Robotics*, 21(10):1177–1199, 2007.
[13] T. Taniguchi and T. Sawaragi. Incremental acquisition of multiple nonlinear forward models based on differentiation process of schema model. *Neural Networks*, 21(1):13–27, 2008.
[14] C. Watkins and P. Dayan. Technical note: Q-learning. *Machine Learning*, 8:279–292, 1992.
[15] D.M. Wolpert, K. Doya, and M. Kawato. A unifying comuputational framework for motor control and social interaction. *Phil Trans R Soc Lond B*, 358:593–602, 2003.
[16] D.M. Wolpert, Z. Ghahramani, and M. I. Jordan. An internal model for sensorimotor integration. *science*, 269:1880–1882, 1995.
[17] D.M. Wolpert and M. Kawato. Multiple paired forward and inverse models for motor control. *Neural Networks*, 11:1317–1329, 1998.



Machine Learning

Edited by Abdelhamid Mellouk and Abdennacer Chebira

ISBN 978-953-7619-56-1
Hard cover, 450 pages
Publisher InTech
Published online 01, January, 2009
Published in print edition January, 2009

Machine Learning can be defined in various ways related to a scientific domain concerned with the design and development of theoretical and implementation tools that allow building systems with some Human Like intelligent behavior. Machine learning addresses more specifically the ability to improve automatically through experience.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Tadahiro Taniguchi, Kenji Ogawa and Tetsuo Sawaragi (2009). Implicit Estimation of Another's Intention Based on Modular Reinforcement Learning, Machine Learning, Abdelhamid Mellouk and Abdennacer Chebira (Ed.), ISBN: 978-953-7619-56-1, InTech, Available from: http://www.intechopen.com/books/machine_learning/implicit_estimation_of_anothers_intention_based_on_modular_reinforcement_learning

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen