Electronic Preprint for Journal of Information Processing Vol.22 No.2

Regular Paper

# Encouragement of Right Social Norms by Inverse Reinforcement Learning

Sachiyo Arai[1,a)] Kanako Suzuki[2]

Received: July 7, 2013, Accepted: January 8, 2014

**Abstract:** This study is intended to encourage appropriate social norms among multiple agents. Effective norms, such as those emerging from sustained individual interactions over time, can make agents act cooperatively to optimize their performance. We introduce a "social learning" model in which agents mutually interact under a framework of the coordination game. Because coordination games have dual equilibria, social norms are necessary to make agents converge to a unique equilibrium. As described in this paper, we present the emergence of a right social norm by inverse reinforcement learning, which is an approach for extracting a reward function from the observation of optimal behaviors. First, we let a mediator agent estimate the reward function by inverse reinforcement learning from the observation of a master's behavior. Secondly, we introduce agents who act according to an estimated reward function in the multiagent world in which most agents, called citizens, have no way to act. Finally, we evaluate the effectiveness of introducing inverse reinforcement learning.

**Keywords:** inverse reinforcement learning, social norms

## 1. Introduction

For a modern society with extended diversification of individual values and globalization in progress, the necessity for social norms is increasing as a common action agenda. A social dilemma is considered cancelable by sharing appropriate social norms among agents, so that its property and generation mechanism attract attention as study subjects. A social norm here implies an action strategy that arises spontaneously in an iterative process of interactions among agents, and which promotes agents' coordination to improve the performance of the entire system if the norm is common and an effective one. We have been involved in urban planning where there are many cases, such as a little common rule which makes people's flow smoothly. For example, Sen et al. [1] takes the case: i.e., which side of a road a car should move along when two cars meet on a road in a coordination game context. Therefore, in this study, it is assumed that the agents in society learn based on rewards, and we aim at urging its generation in response to the demand of an appropriate social norm as described above. This paper adopts Q-learning, one method of reinforcement learning for designing an agent's rules, to generate a social norm from the bottom-up. Specifically, a reward function is inferred by inverse reinforcement learning from the optimal action sequence.

We focus on an environment in which multiple agents iterate interactions locally, and formulate such an interaction among agents by a coordination game. Multiple Nash equilibria might exist in a coordination game, where each agent chooses a mutually differing equilibrium point and the performance, from a global perspective, drops.

Methods for converging such equilibria into one are divisible roughly into the following two approaches: establishing direct communications among agents, such as the introduction of the mechanism of planning negotiation and consensus building among agents, and generating an appropriate social norm that leads the whole system to one balance with no direct consensus building. This study specifically addresses the latter, then we define the urge to generate of a social norm such as "encouragement."

To encourage of the right social norm, three kinds of agents: *master*, *mediator*, and *citizen*, are introduced. Master takes action in alignment with the social norm to be achieved. Master knows the ideal situation of the environment, and then pursues a vision. However, he does not know how other people are made to follow. Then, the mediator will learn an incentive which is required to spread the master's policy throughout the system. Here, an incentive is calculated in terms of reward by inverse reinforcement learning. The reward function, which is estimated by inverse reinforcement learning, will be effective not only to make the mediator achieve the master's policy but also to make other people, we call them citizen, behave the same way as the master's.

In Section 2, our problem domain is defined and the related work is introduced. In Section 3, our proposed method, which is to encourage a common social norm by inverse reinforcement learning, is explained. The estimated reward function by mediator is shown in Section 4, and in Section 5, it is verified whether a social norm is encouraged by comparing with an environment only with agents who carry out Q-learning. Finally, in Section 6, conclusions and directions for future work are presented.

[1] Chiba University, Inage, Chiba 263–8522, Japan
[2] Japan Maritime Self-Defense Force, Meguro, Tokyo 153–8933, Japan
[a)] arai@tu.chiba-u.ac.jp

## 2. Problem Domain

### 2.1 Coordination Game

Agents $i$, $j$ ($i \neq j$) have an identical strategy set $\{S_1, S_2\}$ in a coordination game. Each agent chooses one strategy simultaneously, and acquires payoffs based on **Table 1**. Table 1 is a regularized payoff matrix of a coordination game with the off-diagonal element as 0, and $v_1 > 0$, $v_2 > 0$ are realized. The Nash equilibria of a coordination game are expressed as $(S_1, S_1)$ and $(S_2, S_2)$.

**Case (1)** $v_1 > v_2$ $(S_1, S_1)$ is Pareto optimal and designated as a Pareto superior solution. A rational choice is $S_1$. The greatest payoff $v_1$ is acquired if an adversary's action is $S_1$. However, a Pareto superior solution is not necessarily implemented, because it is favorable to choose $S_2$ when an adversary chooses $S_2$.

An appropriate social norm for choosing $S_1$ is necessary to implement a Pareto superior solution in this type of game.

**Case (2)** $v_1 = v_2$ Two Nash equilibria are identical. There exists neither temptation nor risk when choosing the action. Accordingly, a problem exists by which the optimal action cannot be specified from a payoff matrix. A social norm as an action agenda common to all the agents for choosing either $S_1$ or $S_2$ is required in this type of game. This paper specifically examines this latter case.

### 2.2 Learning a Social Norm

#### 2.2.1 Environmental Model

This paper follows the setting of multiagent system environments [1], [2].

Spatially distributed multiple agents iterate coordination games locally. An adversary of each agent is chosen randomly from among agents positioned in the neighborhood each time. Then each agent learns by obtaining payoff in the game as a reward. The agent cannot distinguish itself as an adversary at this time, but can sense his action. Sen et al. [1] modeled this situation as "learning the rule of which side on the road to drive on," and defined it as "Social Learning," meaning to learn which side of the road a car should move along when two cars meet on a road in a coordination game.

A two-dimensional torus grid shown in **Fig. 1** is used as a space structure, in which the location of agents is fixed, but where each agent plays a game with an agent positioned in the neighborhood (hereinafter designated as a neighboring agent), where a neighboring agent means an agent positioned within distance $D$ from the agent concerned, and where $D$ is expressed in Manhattan distance. The number of neighboring agents $n_D$ is given as shown in Eq. (1).

**Table 1** Normalized payoff matrix.

| $i \setminus j$ | $S_1$ | $S_2$ |
|---|---|---|
| $S_1$ | $v_1, v_1$ | $0, 0$ |
| $S_2$ | $0, 0$ | $v_2, v_2$ |

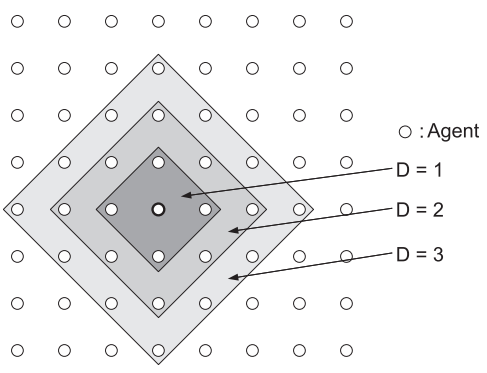


**Fig. 1** Location of agent in torus-grid graph.

$$n_D = \begin{cases} 4 \ (D = 1) \\ 4D + n_{D-1} \ (D \geq 2) \end{cases} \quad (1)$$

**Table 2** shows the payoff matrix of the coordination game used in this paper. Let the action set for agents $i$, $j$ ($i \neq j$) be $\mathcal{A} = \{\mathrm{L}, \mathrm{R}\}$. A social norm is regarded as generated when the whole system converges to L or R.

#### 2.2.2 Agent Model

Q-learning [3], one method of reinforcement learning, is used to design an agent's rules. An agent senses state $s \in \mathcal{S}$, and chooses action $a \in \mathcal{A}(s)$ based on policy $\pi$, where $\mathcal{S}$ is a set of states to which an environment can transit, and $\mathcal{A}(s)$ is a set of actions selectable at state $s$. An agent receives reward $r$ after a choice of action, and senses a new state $s'$. Then Q-learning updates $Q(s, a)$, the value of state $s$ and action $a$, with Eq. (2), where $\alpha$ ($0 < \alpha \leq 1$) is a learning rate, $\gamma$ ($0 \leq \gamma \leq 1$) is a discount rate, and $k$ is the number of times when $a$ is chosen at $s$ and $Q(s, a)$ is updated :

$$Q_{k+1}(s, a) = Q_k(s, a) + \alpha \left\{ r + \gamma \max_{a' \in \mathcal{A}(s')} Q_k(s', a') - Q_k(s, a) \right\}. \quad (2)$$

Setting the state and the reward is shown below.

**State** Let the combination of one's own action $a^i$ and an adversary's action $a^j$ during past $l$ steps be $(a^i_{t-l}, a^j_{t-l}, a^i_{t-l+1}, a^j_{t-l+1}, \ldots, a^i_{t-1}, a^j_{t-1})$. The number of states is $2^{2l}$. The state set $\mathcal{S}$ is presented in **Table 3**.

**Reward** Payoff defined by Table 2 is given as reward $r$.

Let one step be an update of Q-value by all the agents once. One trial consists of $T$ steps. The procedure of one step is shown in **Fig. 2**, where $n$ is the number of agents and $n > 3$.

### 2.3 Preliminary Experiment

The influence of the distance from a neighboring agent $D$

**Table 2** Coordination game of our setting.

| $i \setminus j$ | L | R |
|---|---|---|
| L | $1, 1$ | $-1, -1$ |
| R | $-1, -1$ | $1, 1$ |

**Table 3** $\mathcal{S}$: Set of states.

| | | | | |
|---|---|---|---|---|
| 0 | – | | | |
| 1 | $s_1$: LL | $s_2$: LR | $s_3$: RL | $s_4$: RR |
| 2 | $s_1$: LLLL | $s_2$: LLLR | $s_3$: LLRL | $s_4$: LLRR |
| | $s_5$: LRLL | $s_6$: LRLR | $s_7$: LRRL | $s_8$: LRRR |
| | $s_9$: RLLL | $s_{10}$: RLLR | $s_{11}$: RLRL | $s_{12}$: RLRR |
| | $s_{13}$: RRLL | $s_{14}$: RRLR | $s_{15}$: RRRL | $s_{16}$: RRRR |
| ⋮ | ⋮ | | | |
| $l$ | $a^i_{t-l} a^j_{t-l} a^i_{t-l+1} a^j_{t-l+1} \cdots a^i_{t-1} a^j_{t-1}$ | | | |
| ⋮ | ⋮ | | | |

Iteration for $T$ steps (One trial):
(1) All agents choose an action based on their own policy.
(2) Agent $i$ ($= 1, 2, \ldots, n$) chooses adversary $j$ ($\neq i$) at random out of all neighboring agents.
(3) Agent $i$ plays a game with agent $j$, and updates the Q-value based on the acquired payoff $r_i$.

**Fig. 2** Update procedure of Q-value (one trial).

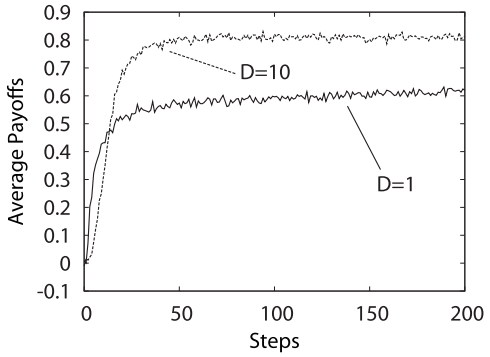


**Fig. 3** Transition of average payoff $\bar{r}$ ($\epsilon = 0.0$).

on the generation of a social norm was verified using the two-dimensional lattice of $N = 100$, or $10 \times 10$ agents. This experiment compared cases $D = 1$ and $D = 10$, in which the numbers of neighboring agents $n_D$ were $n_1 = 4$ and $n_{10} = 99$, respectively. As for the parameter setting in Q-learning, the state variable was not examined ($l = 0$), learning rate $\alpha = 0.1$, discount rate $\gamma = 0.9$, and the $\epsilon$-greedy method of $\epsilon = 0.1$ was used for action selection. 100 trial experiments were conducted with a different random number seed. The average payoff of these 100 trials was evaluated as a result.

The transition of the global mean acquired payoff at each step $\bar{r}$ is shown in **Fig. 3**. Each value of Fig. 3 is computed as follows. After updating the Q-value with $\epsilon = 0.1$ at each time step, each agent selects an action based on this Q-table with $\epsilon = 0.0$. Then we check the payoffs $r$ of agents'. We repeat this process 500 times, and calculate their average as payoffs $\bar{r}$ of each step. If $\bar{r} = 1.0$, it means that a social norm has emerged in the system. In each case of $D = 1$ and $D = 10$, the percentage where the social norm emerged was 58% and 80%, respectively. That is, the intended social norm is not always generated. When the social norm occurs in $D = 1$ and $D = 10$, the required numbers of average steps are 2,938.0 and 22.0, respectively.

Figure 3 demonstrates that $\bar{r}$ rises earlier in the case of $D = 1$ compared with $D = 10$, but a social norm is generated quickly in the case of $D = 10$. Because the number of neighboring agents $n_D$ is small in the case of $D = 1$, there is little influence on the change to each agent's adversary (strategy) each time. Accordingly, learning progresses more quickly than $D = 10$, so that $\bar{r}$ rises earlier. However, each agent is involved in such a narrow region that a local equilibrium is apt to be implemented. It takes time until it is eliminated by either Nash equilibria. Nevertheless, because the spatial relation does not limit adversary agents in $D = 10$, the whole system tends to converge as one of the Nash equilibria, so that a social norm is generated more quickly than in $D = 1$. Although not shown in the result, a social norm was confirmed to be generated more quickly in $D = 10$ in the cases of state representation $l = 1, 2$.

The $D = 1$ and $D = 10$ cases can be expressed respectively as a two-dimensional grid and a complete graph by tying agents to play a game with a link. That is, the generation rate of a social norm is strongly affected by a network structure. However, various network structures in the real world demand that social norms be generated as early as possible, irrespective of the network structure. Consequently, an approach for encouraging a social norm in the environment of $D = 1$ of slow social norm generation is discussed henceforth in response to this result.

### 2.4 Related Work

Studies that elucidate the generation mechanism of social norms by a mathematical model are attracting attention [4], [5].

Oura [6] classified problems of how to establish an orderly system according to game theory, and analyzed factors for generating social norms and sustaining the order from the perspective of evolutionary game theory, which analyzes the transition of a social state expressed by strategy distribution, and which can elucidate emergence phenomena occurring in the process of strategy evolution. However, discussion about social norms taking the selfishness of an agent into consideration presumably requires a bottom-up approach using reinforcement learning in which an agent has adapted to an environment from past experience as a lesson, thereby learning an action rule directly from a reward. Moreover, the viewpoint of multiple robotic control anticipates that the norm generation method using reinforcement learning in which an agent can acquire an action rule autonomously is applicable to the acquisition of coordinate action for a robot.

There have been previous studies of social norm generation by reinforcement learning [1], [2]. Sen et al. [1] demonstrated that a social norm is generated while each agent learns from past experience and is adapted for environment, in an environment where multiple agents iterate coordination games locally. Mukherjee et al. [2] showed that the spatial relations among agents affect the generation rate of social norms. These studies are centered on the discussion of phenomena, such as analyses of parameters related to social norm generation, whereas this study aims at encouraging an appropriate social norm.

Tuyls et al. [7] presented that an agent's individuality and sociality emerge in an environment where a payoff matrix varies by reinforcement learning. Although an approach by which sociality without consideration of communication among agents is shared by this study, this paper does not take an agent's individuality into consideration, and assumes that a payoff matrix is invariant with time.

## 3. Encouraging Social Norms

An approach to encourage social norms using inverse reinforcement learning (IRL) is proposed. This chapter presents a description of inverse reinforcement learning.

### 3.1 Inverse Reinforcement Learning

Reinforcement learning is a method by which an agent acquires an action rule autonomously based on a reward provided from the

environment. It is advantageous because it anticipates the discovery of a solution that is superior to both automation and simplification of a control program, and hand coding [8]. However, it entails problems by which learning performance is strongly dependent on the mode of providing a reward, and that it is unknown how to provide an effective reward for implementing an expert's skill in a robot. Consequently, establishment of a reward design problem remains as a subject for further investigation [9].

Inverse reinforcement learning was defined originally by Russel [9] as a problem that determines a reward function given the optimal action sequence and environment model. Various approaches have been proposed [10], [11], [12]. Ng et al. [10] reported an approach for estimating a reward function using linear programming for an environment with finite state space, and the Monte Carlo method for an environment with an infinite state space. Abbeel et al. [11] reported the approach of "Apprenticeship learning" in which the optimal policy is acquired in the process of presuming a reward function. Natarajan et al. [12] presumed multiple reward functions in a multiagent environment, and proposed an approach for controlling the behaviors from a global perspective.

This study adopts the inverse reinforcement learning method of a finite state space proposed by Ng et al. [10] as described in Section 3.2. This study, in spite of the multiagent environment, adopts a bottom-up approach by which each agent estimates a reward function and ignores global control. Therefore the approach of Natarajan et al. is not adopted.

Syed et al. [13], who introduced inverse reinforcement learning into game theory, presumed the reward function of an adversary as unknown by inverse reinforcement learning, and proposed a policy acquisition method with the policy of an adversary examined. This paper also introduces inverse reinforcement learning into game theory, where inverse reinforcement learning is used as a reward design problem that estimates a reward from the action sequence of an agent who knows the optimal action.

### 3.2 Inverse Reinforcement Learning in Finite State Space

Let $a_1$ be the optimal action at state $s_m$ ($m = 1, 2, \ldots, M$). The reward function $R'$ is inferred by solving the linear programming problem of Eq. (3). Reward function vector $\boldsymbol{R'}$ in Eq. (3) is given by reward $r'_{s_m}$ of state $s_m$, expressed by Eq. (4). State transition matrix $\boldsymbol{P}_a$ is an $M \times M$ matrix given by state transition probability $P^a_{ss'}$ of action $a$. $\boldsymbol{P}_a$ is expressed by Eq. (5), where the probability of taking action $a$ to transit from state $s_m$ to $s_{m'}$ is $P^a_{mm'}$.

$\boldsymbol{P}_a(m)$ represents the $m$-th row vector of $\boldsymbol{P}_a$. $\lambda$ is a penalty coefficient. A state of high value can be extracted by enhancing $\lambda$. $R'_{\max}$ ($> 0$) is a value set as restrictions of a reward.

$$\begin{aligned} \text{maximize: } & \sum_{m=1}^{M} \min_{a \in \mathcal{A} \backslash a_1} \{(\boldsymbol{P}_{a_1}(m) - \boldsymbol{P}_a(m)) \\ & (\boldsymbol{I} - \gamma \boldsymbol{P}_{a_1})^{-1} \boldsymbol{R'}\} - \lambda \|\boldsymbol{R'}\|_1 \end{aligned} \tag{3}$$

$$\begin{aligned} \text{s. t.: } & (\boldsymbol{P}_{a_1} - \boldsymbol{P}_a)(\boldsymbol{I} - \gamma \boldsymbol{P}_{a_1})^{-1} \boldsymbol{R'} \geq 0 \quad \forall a \in \mathcal{A} \backslash a_1 \\ & |r_{s_m}| \leq R'_{\max} \quad m = 1, \ldots, M \end{aligned}$$

$$\boldsymbol{R'} = (r_{s_1}, \ldots, r_{s_m}, \ldots, r_{s_M})^{\mathrm{T}} \ (M \times 1 \text{Vector}) \tag{4}$$

$$\therefore \|\boldsymbol{R'}\|_1 = \max_{1 \leq j \leq n} \sum_{i=1}^{M} |r_{ji}| = \sum_{i=1}^{M} |r_{si}|$$

$$\boldsymbol{P}_a = \begin{pmatrix} P^a_{11} & P^a_{12} & \cdots & P^a_{1m} & \cdots & P^a_{1M} \\ P^a_{21} & P^a_{22} & \cdots & P^a_{2m} & \cdots & P^a_{2M} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ P^a_{m1} & P^a_{m2} & \cdots & P^a_{mm} & \cdots & P^a_{mM} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ P^a_{M1} & P^a_{M2} & \cdots & P^a_{Mm} & \cdots & P^a_{MM} \end{pmatrix} \tag{5}$$

The inverse reinforcement learning method is used on the premise that environment models such as a state transition probability are known. However, because the state transition probability $P^a_{ss'}$ is often unknown in the multiagent environment described in this paper, application of the inverse reinforcement learning method requires the presumption of a state transition probability by observation.

Herein, we present two computing approaches used for estimating a state transition probability.

**(1)** Approach based on the Bayesian estimation [14]

Let $\hat{P}^a_{ss'}$ be the estimate of state transition probability $P^a_{ss'}$, which is determined by Eq. (6).

$$\hat{P}^a_{ss'} = \frac{C_{s'} + 1}{C_a + M_s} \tag{6}$$

Therein, $C_{s'}$ denotes the number of times the transition to state $s'$ from state $s$ by taking action $a$, $C_a$ is the number of times action $a$ at state $s$ is taken, and $M_s$ is the number of states that can be transited to form state $s$, in Eq. (6). This approach is used under the assumption that $\hat{P}^a_{ss'}$ is random when $C_a$ is small. However, when $C_a$ is large, $\hat{P}^a_{ss'}$ tends to the fraction of cases of taking action $a$ at state $s$ to transit to state $s'$. Thereby the result is reflected by the amount of knowledge obtained by experience.

**(2)** Approach determined by the fraction of observational data

The estimate $\hat{P}^a_{ss'}$ of state transition probability $P^a_{ss'}$ is determined according to Eq. (7).

$$\hat{P}^a_{ss'} = \begin{cases} \frac{C_{s'}}{C_a} \ (C_a \neq 0) \\ 0 \ (C_a = 0) \end{cases} \tag{7}$$

Equation (7) nulls all the estimates $\hat{P}^a_{ss'}$ of the state transition probability not observed by this approach, so that the presumption of a reward function by inverse reinforcement learning is unaffected. This approach is adopted in this paper.

### 3.3 Proposed Model

#### 3.3.1 Definition of an Agent

Agents of three types, *master*, *mediator*, and *citizen*, are set up to introduce inverse reinforcement learning. The definition of each agent is described below. $\mathcal{M}_a$, $\mathcal{M}_e$, $C$ respectively represent a set of master, mediator, and citizen that comprises elements *ma*, *me*, and *c*.

- *master* ($ma \in \mathcal{M}_a$)
  Agents who take optimal action $a_1$.
- *mediator* ($me \in \mathcal{M}_e$)
  Agents who act based on state value $V(s)$ obtained from $R'$, as estimated by inverse reinforcement learning by observing of the action sequence of the master.
- *citizen* ($c \in C$)
  Agents who carry out Q-learning based on reward $r$ defined

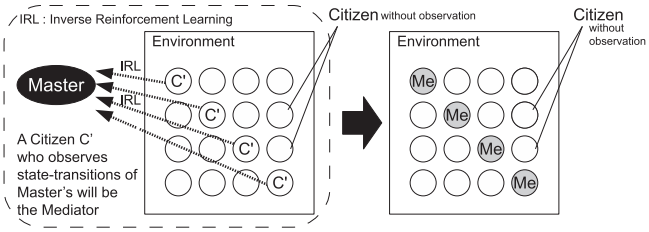


**Fig. 4** Encouragement of a social norm via inverse reinforcement learning.

in Table 2.

As mentioned in Section 1, a master knows the ideal situation of the environment and acts optimally. However, he does not (or cannot) explain the reason why his actions are desirable in a quantitative way. For example, the principal of a school is regarded as a master who walks along the right side of a passage. However, there are few opportunities to show quantitatively why right-hand traffic is beneficial to a student from the viewpoint of an interaction with other pedestrians. On the other hand, teachers should explain right-hand traffic to the students through interaction. They do not know the value of each state-action before applying inverse reinforcement learning. As for the students, whichever side is not a problem, but they will opt for the side which is the safest based on their experience. They only learn from the environmental reward. Therefore, when a teacher walks on the right-hand side, the reward obtained by the student walking along the right becomes high, making the student walk along the right as well.

#### 3.3.2 Introduction of Inverse Reinforcement Learning

**Figure 4** indicates the flow of the introduction of inverse reinforcement learning. Encouragement of a social norm by inverse reinforcement learning is conducted in the following two steps.

(1) $c' \in C'$ ($\subseteq C$) of a fraction of $p$ ($0 \leq p \leq 1$) is chosen from existing agents [*1]. Each chosen $c'$ plays a coordination game with *ma* iteratively. $c'$ estimates the reward function $R'$ from the observed action sequence of *ma*.

(2) After $c'$ estimated reward function $R'$, $c'$ turns into *me*, which acts based on the state value $V(s)$ obtained from $R'$, and affects the learning of $c$.

## 4. Estimate of Master's Reward Function $R'$

### 4.1 Experimental Settings

An experiment is conducted in which $c'$ chosen from the environment with a fraction of $p$ plays a coordination game of 10,000 iteration steps with *ma*, and $c'$ estimates the reward function $R'$ of *ma*.

Let the discount rate $\gamma = 0.9$ and $R'_{\max} = 1.0$. Let the penalty coefficient $\lambda = 0$ or $\lambda = 4$. Setting of the action criteria and states are as follows.

**Criteria of Master's Action** *ma* chooses the action of the adversary 1 step before the present while $c'$ acts at random. *ma* takes the action following that of $c'$ with no information related to the environment.

**State Space of Master's** The state representation of Table 3 is obeyed, where agents $i$ and $j$ are replaced respectively by *ma* and $c'$. This experiment uses $l = 1$ of $l = 2$, and aims at acquiring a reward function that takes iteration into consideration by adopting the action sequence of the past game as a state. Moreover, an initial state is generated at random for every step because it is necessary to observe a transition among all the states.

[*1] Initially all agents are citizen.

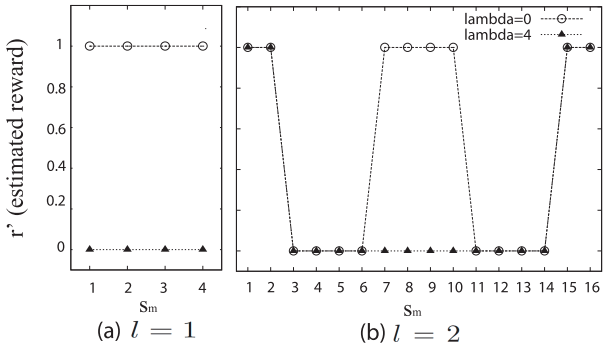


**Fig. 5** $R'$: Estimated reward function of master's.

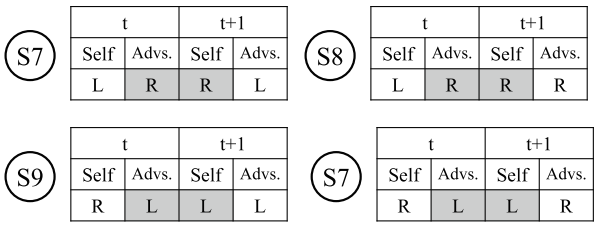


**Fig. 6** Set of states of which has higher reward at $\lambda = 0$.

### 4.2 Experimental Results

Estimated reward function $R'$ is shown in **Fig. 5**, where state $s_m$ is along the horizontal axis and estimated reward $r'_{s_m}$ along the vertical axis. Figure 5 (a) shows that all the $r'_{s_m}$ are identical in the case of $l = 1$. This is true because no deviation occurs in the observed state transition, so that no valuable state is detected even if $\lambda$ is increased. Table 2 implies that $s_2$, $s_3$ are states in which the payoff of $-1$ is presumed be acquired. Therefore, misunderstandings have arisen. Consequently, this result suggests that $l = 1$ is an inappropriate state representation.

However, Fig. 5 (b) shows that $r'_{s_m}$ of states that can be transitioned from no state $\{s_3, \ldots, s_6, s_{11}, \ldots, s_{14}\}$ is 0, and $r'_{s_m}$ of states in which the transition observed $\{s_1, s_2, s_7, \ldots, s_{10}, s_{15}, s_{16}\}$ is 1, in the case of $l = 2$, $\lambda = 0$. Furthermore, when a reward of 1 is obtained according to Table 2 at time $t-2$, then states that take the same action at time $t-1$ $\{s_1, s_2, s_{15}, s_{16}\}$ are detected as valuable states in the cases of $\gamma = 0.9$, $\lambda = 4$ of Fig. 5 (b).

An example of state transition observed in the case of $l = 2$ is portrayed in **Fig. 7**, which demonstrates transition from state $s_t$ to the next state $s_{t+1}$, as indicated by an arrow. States in gray $\{s_1, s_2, s_7, \ldots, s_{10}, s_{15}, s_{16}\}$, at which transition is observed, are detected as $r'_{s_m} = 1$ at $\lambda = 0$. The discount rate $\gamma$ in reinforcement learning can be treated as a parameter that determines the value of a state to be transited next in inverse reinforcement learning, as well as a parameter that determines the current value of a reward expected to be obtained in the future. Accordingly, states $s_1$ and $s_{16}$ that loop into the same state shown by the frame in

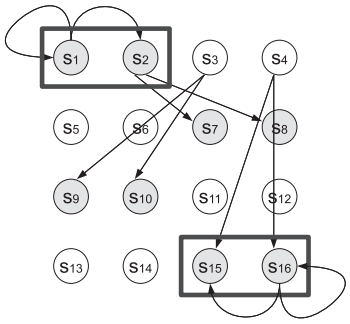


**Fig. 7** Example of state transition in the case of $l = 2$.

Fig. 7 and states $s_2$, and $s_{15}$ which are observed to transit from $s_1$ and $s_{16}$ are presumed to be detected as states with a high value, under the influence of $\gamma = 0.9$ in the case of $\lambda = 4$.

## 5. Mediator and Citizen

### 5.1 Experimental Settings

$C'$ with reward function $R'$ estimated in Section 4 is introduced into the environment as Mediator, and it is verified whether a social norm is encouraged. An environment with a number of agents $N = 100$ and $D = 1$ is used. Mediator's arrangements of four types shown in **Fig. 8** are adopted, and Mediator's fraction $p$ is $p = 0.00, 0.04, 0.25, 0.50$, and $1.00$. Settings of action criteria and states are as follows.

**Criteria of Mediator's Action** Mediator ($me$) with reward function $R'$ estimated from a master has a value function $V(s_m)$ of state $s_m$ as $V(s_m) = r'_{s_m}$, and acts greedy according to $V(s_m)$. Citizen ($c$) with learning rate $\alpha = 0.1$ and discount rate $\gamma = 0.9$ conducts Q-learning with the $\epsilon$-greedy method of $\epsilon = 0.1$ for action selection.

**State Space of Mediator** Both $me$ and $c$ obey Table 3. For $l = 1$, because all values of the state value function of Mediator $V(s)$ are identical, it is confirmed by experimentation that random action is implemented and that there is no effective influence on the learning of $c$. Consequently, $l = 2$ and $\lambda = 0, 4$ are adopted according to the result presented above.

100 trial experiments with a different random number seeds were conducted. As we explained in section 2.3, we checked the payoffs $r$ of agents' 500 times with $\epsilon = 0.0$ by using a different random number seeds after updating the Q-value with $\epsilon = 0.1$. The reason why we take the average over 1,000 times is to examine the influence of the different combinations of the adversary agents. When the averaged value $\bar{r}$ becomes 1.0 at $t$ steps, a social norm is regarded as generated at step $t$ which is called $T_e$.

In the case of $p = 1.0$, because all the agents are acting greedy as Mediator, when $\bar{r}$ is 1.0 during the 500 steps after $T_e$, a social norm is regarded as generated at $T_e$. When the arrangement of Mediator is random, a total of 100 trial experiments is conducted with 10 kinds of different arrangements and 10 trials at each arrangement.

### 5.2 Experimental Results

The average of step $T_e$ at which a social norm is generated in 100 trials is shown in **Fig. 9**, where the fraction of Mediator $p$ is along the horizontal axis and $T_e$ is along the vertical axis. The scale of the vertical axes is not the same. Trials were excluded from which no social norm occurred through 1,000,000 steps, which include "45 trials at $\lambda = 0$ and $p = 1.00$," and "4 trials at $\lambda = 0$, uniformity arrangement, and $p = 0.50$."

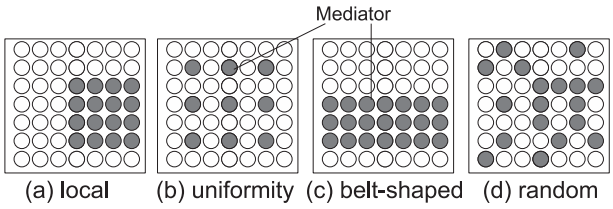


**Fig. 8** Variety of Mediator arrangements.

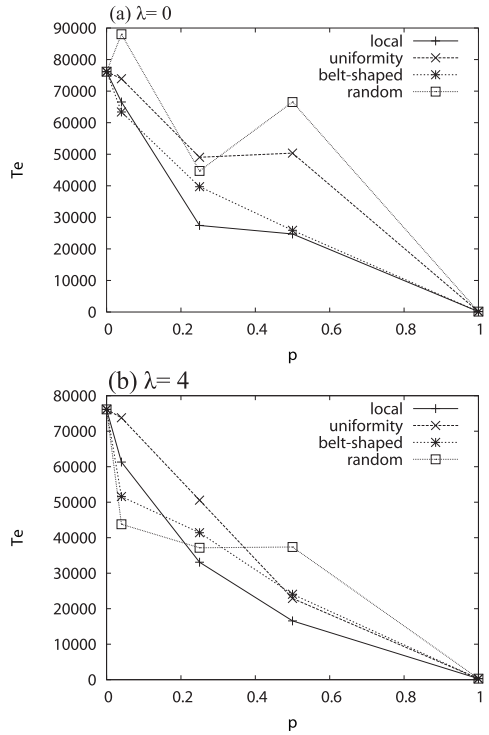


**Fig. 9** Average number of required steps for emerged social norm.

### 5.3 Discussion

The results demonstrate that the reward function $R'$ obtained at $\lambda = 0$ is ineffective at any trial. Here, we discuss the typical drawback of the $\lambda = 0$. In the case of $\lambda = 0$, the estimated reward function is shown in Fig. 5 (b). Comparing with the case of $\lambda = 4$, there are additional states, $S_7$, $S_8$, $S_9$, $S_{10}$, which are regarded as the states which bring about a higher reward. In **Fig. 6**, the above four states representations are shown to examine this drawback. All of these four states have the rule - If adversary's action at time $t$ is different from my action, then I change my action to meet the adversary's. This rule causes fluctuation in the agents' behavior. Especially, in the case of "random" action is changed in each time step because there are few opportunities for $m_e$ adjoin mutually. The reason why $T_e$ of the "local" and "belt" arrangement are comparatively small is that $m_e$ adjoins mutually in these cases. The following cases are mostly brought about because the high reward were assigned to the four states shown in Fig. 6.

In the case of $p = 0.04$ and $p = 0.5$, $T_e$ is higher than the

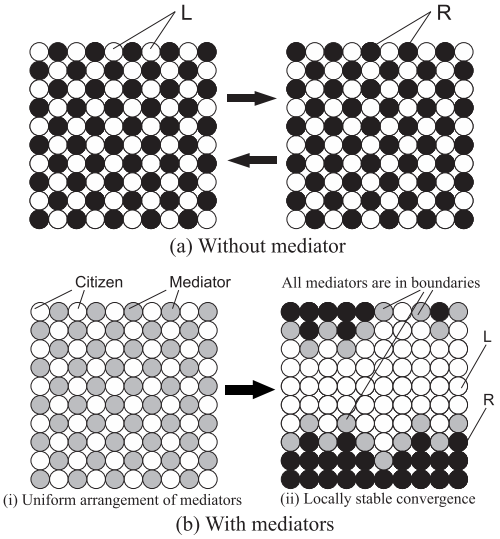


**Fig. 10** The examples of social norm creation.

case of smaller value of $p$ in the random arrangement. Not only for the both cases, the fluctuation of agents' behavior in the random arrangement takes place frequently. Therefore, the standard deviation of $T_e$ is larger than other cases'.

In the case of $\lambda = 0$ and $p = 1.0$, the whole system converges to a state in which adjoining agents take either L or R alternately in a trial where no social norm is generated, as shown in **Fig. 10** (a). Filled and open circles in Fig. 10 (a) respectively denote that an agent takes L and R action.

In the case of $\lambda = 0$, uniformity arrangement, and $p = 0.50$, Fig. 10 (b)-(i) presents the arrangement of Mediators. When the whole system converges to neither equilibrium but falls into a local equilibrium, *me* tends to be a boundary, as shown in Fig. 10 (b)-(ii), where filled and open circles represent an agent who takes L and R action, respectively, and gray circles represent *me* positioned at a boundary. Because such *me* positioned at a boundary adjusts itself with each equilibrium, its action is undetermined. Consequently, it is confirmed that no social norm is generated as a result.

On the other hand, Fig. 9 (b) has verified that a reward function obtained at $\lambda = 4$ generates a social norm more quickly than $p = 0.00$ with no Mediator in any combination of Mediator arrangements and fraction $p$. This result demonstrates that the proper setting of $\lambda$ implements an effective social norm encouraging method by introducing of inverse reinforcement learning. An arrangement for quick social norm generation varies with Mediator fraction $p$. Accordingly, it is necessary to examine the influence of the arrangement of a Mediator that promotes social norm generation.

## 6. Conclusion and Future Work

This study specifically examined an environment in which spatially distributed agents interact locally and aims to generate and encourage an appropriate social norm.

First, the influence of an agent's spatial relation on the generation rate of a social norm was observed using Q-learning: one method of reinforcement learning to design an agent's rules. Then the necessity for encouraging a social norm was described based on the result.

Next, agents acting according to a state value determined using a reward function estimated by inverse reinforcement learning was introduced into the environment. It was confirmed that a social norm was encouraged by providing state representation and a penalty coefficient $\lambda$ properly. It was demonstrated that inverse reinforcement learning can be introduced to design a reward for encouragement of a social norm based on this result.

Our future work shall include investigation of the effective arrangement of Mediators and the effective mode of giving a penalty coefficient verified experimentally in the present study. Moreover, we will examine whether the social norm encouragement method by the introduction of inverse reinforcement learning is effective in other game environments or with other network structures, such as a small world network.

### References

[1] Sen, S. and Airiau, S.: Emergence of norms through social learning, *Proc. 20th International Joint Conference on Artificial Intelligence*, pp.507–512 (2007).
[2] Mukherjee, P., Sen, S. and Airiau, S.: Norm Emergence Under Constrained Interactions in Diverse Societies, *Proc. 7th International Joint Conference on Autonomous Agents and Multiagent Systems*, Vol.2, pp.779–786 (2008).
[3] Watkins, C.J.C.H. and Dayan, P.D.: Q-learning, *Machine Learning*, Vol.8, pp.279–292 (1992).
[4] Kobayashi, J.: Unanimous Opinions in Social Influence Networks, *Journal of Mathematical Sociology*, Vol.25, pp.285–297 (2001).
[5] Axelrod, R.: An Evolutionary Approach to Norms, *American Political Science Review*, Vol.80, pp.1095–1111 (1986).
[6] Oura, H.: Evolutionary Game Theoretical Approach for Order Problem, *Sociological Theory and Methods*, Vol.18, No.2, pp.132–152, in Japanese (2003).
[7] Tuyls, K. and Nowe, A.: Evolutionary game theory and multi-agent reinforcement learning, *The Knowledge Engineering Review*, Vol.20, No.1, pp.63–90 (2006).
[8] Mahadevan, S. and Connell, J.: Automatic programming of behavior-based robots using reinforcement learning, *Artificial Intelligence*, Vol.55, pp.311–365 (1992).
[9] Russell, S.: Learning agents for uncertain environments (extended abstract), *Proc. 16th International Conference on Machine Learning*, pp.278–287 (1998).
[10] Ng, A. and Russell, S.: Algorithms for inverse reinforcement learning, *Proc. 17th International Conference on Machine Learning*, pp.663–670 (2000).
[11] Abbeel, P. and Ng, A.: Apprenticeship learning via inverse reinforcement learning, *Proc. 21st International Conference on Machine Learning*, pp.1–8 (2004).
[12] Natarajan, S., Kunapuli, G., Judah, K., Tadepalli, P., Kersting, K. and Shavlik, J.: Multi-agent inverse reinforcement learning, *Proc. International Conference on Machine Learning and Applications*, pp.395–400 (2010).
[13] Syed, U. and Schapire, R.E.: A game-theoretic approach to apprenticeship learning, *Advances in Neural Information Processing Systems*, Vol.20, pp.1449–1456 (2007).
[14] Yukinawa, N., Yoshimoto, J., Oba, S. and Ishii, S.: System identification of gene expression time-series based on a linear dynamical system model with variational Bayesian estimation, *IPSJ Trans. Mathematical Modeling and Its Applications*, Vol.46, No.10, pp.57–65 (2005).

**Sachiyo Arai** received her B.S. and M.S. degrees in control engineering, and Ph.D. degree in artificial intelligence, from Tokyo Institute of Technology in 1998. After receiving the Ph.D. degree, she worked at the Robotics Institute in Carnegie Mellon University 1999–2001, a visiting Associate Professor at the department of Social Informatics in Kyoto University 2001–2003. Currently, an Associate Professor of Urban Environment Systems, Faculty of Engineering, Chiba University. She is involved in realizing system consists of autonomous and multiple agents.

**Kanako Suzuki** received her B.S. and M.S. degrees in engineering for the urban environment systems from Chiba University in 2010 and 2012, respectively. She is interested in optimization, decision science, and game theory. She has been a student of Martime Officer Candidate School since 2013, and will work for Japan Maritime Self-Defence Force from 2014.