A New Criterion Using Information Gain for Action Selection Strategy in Reinforcement Learning

Kazunori Iwata, *Student Member, IEEE*, Kazushi Ikeda, *Member, IEEE*, and Hideaki Sakai, *Senior Member, IEEE*

***Abstract*—In this paper, we regard the sequence of returns as outputs from a parametric compound source. Utilizing the fact that the coding rate of the source shows the amount of information about the return, we describe ℓ-learning algorithms based on the predictive coding idea for estimating an expected information gain concerning future information and give a convergence proof of the information gain. Using the information gain, we propose the ratio w of return loss to information gain as a new criterion to be used in probabilistic action-selection strategies. In experimental results, we found that our w-based strategy performs well compared with the conventional Q-based strategy.**

***Index Terms*—Information gain, predictive coding, probabilistic action selection strategy, reinforcement learning.**

I. Introduction

Considering an agent that learns a policy for optimizing systems, we are interested in how the agent chooses an action that maximizes future rewards in an unknown environment without a supervisor's support. Examples of an agent include an autonomous robot, a control device, and so on. Reinforcement learning [1] is an effective framework to mathematically describe a general process that consists of interactions between an agent and an environment. The framework has been applied in the fields of online clustering, task scheduling, and financial engineering [2]–[4], for example.

In reinforcement learning, the agent maximizes the return (the discounted sum of future rewards) by exploiting the knowledge of its environment precedently explored by itself. Hence, it is important to know how accurate the knowledge is, or more concretely, how well the expected return termed "Q-function" [1], [5] is estimated for switching its strategies from exploration to exploitation at an appropriate time step. Accordingly, we often try to know how much taking an action contributes for estimating the Q-functions. An effective and viable method is to work out the coding rate of the return that corresponds to the mean of the codeword length when the observed return is encoded; since the coding rate is written as the sum of the essential uncertainty (entropy rate) and the distance between the true and the estimated distributions (redundancy). In other words, the coding rate shows the amount of information on the return, so the "information gain" concerning future information is given by the discounted sum of the coding rates to be observed in future. We accordingly formulate a temporal difference (TD) learning for estimating the expected information gain and prove the convergence of the information gain under certain conditions.

As an example of applications, we propose a new criterion to be used in probabilistic action-selection strategies. Some typical strategies have simply utilized the estimates of the Q-function. Although the estimate is an experience-intensive value for exploitative strategies, it is insufficient for exploration because it does not include factors evaluating the uncertainty and the accuracy of the estimate. This is one reason why controlling the tradeoff between exploration and exploitation is difficult. Hence, we propose the ratio w of return loss to information gain as a criterion for making action selection strategies more efficient. We apply it to a typical probabilistic strategy and show in experiments that the w-based strategy performs well compared with the conventional Q-based strategy.

The organization of this paper is as follows. We begin with the encoding of return sources by the TD learning in Section II. In Section III, we apply the proposed criterion w to the softmax method and show the experimental results comparing the w-based strategy with the conventional Q-based strategy. Finally, we discuss the question of model selection and give some conclusions in Section IV.

II. Source Coding for Return Sequence

We first review the framework of discrete-time reinforcement learning with discrete states and actions, where stochastic processes are Markovian. Let $\mathcal{T} = \{t \mid t = 0, 1, 2, \ldots\}$ denote the set of time steps. Let $\mathcal{S}$ be the finite set of states of the environment, $\mathcal{A}$ be the finite set of actions, and $\Re$ be the set of real numbers. At each step t, an agent senses a current state $s_t \in \mathcal{S}$ and chooses an action $a_t \in \mathcal{A}(s_t)$, where $\mathcal{A}(s_t)$ denotes the set of actions available in the state s_t. The selected action a_t changes the current state s_t to a subsequent state $s_{t+1} \in \mathcal{S}$. The environment yields a scalar reward $r_{t+1} \in \Re$ according to the state transition. The interaction between the agent and the environment produces a sequence of states, actions, and rewards, $s_1, a_1, r_2, s_2, a_2, r_3, \ldots$ The goal of the agent is to learn the optimal policy $\pi^*\colon \mathcal{S} \rightarrow \mathcal{A}$, that maximizes the return over time

$$x(s_t, a_t) = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots = \sum_{i=0}^{\infty} \gamma^i r_{t+i+1} \quad (1)$$

where r_{t+1} is called an immediate reward, whereas $r_{t+2}, r_{t+3}, \ldots$ are called delayed rewards. The parameter γ, where $0 \leq \gamma \leq 1$, is the discount factor that controls the

Manuscript received March 15, 2003; revised October 21, 2003. This work was supported in part by Grant 14003714 for scientific research from the Ministry of Education, Culture, Sports, Science, and Technology, Japan.

The authors are with the Graduate School of Informatics, Kyoto University, Kyoto 606-8501, Japan (e-mail: kiwata@sys.i.kyoto-u.ac.jp).

Digital Object Identifier 10.1109/TNN.2004.828760

relative importance of the immediate reward and the delayed rewards.

A. Return Source

Suppose that the agent chooses an action $a \in \mathcal{A}$ at a state $s \in \mathcal{S}$ n times in the experience. Let $x_i(s,a)$ be the return given by (1) in the ith trial for "fixed" state-action pair (s,a). We regard the returns in n trials as a sequence of length n and denote the "return sequence" by

$$x^n(s,a) = x_1(s,a), x_2(s,a), \dots, x_n(s,a). \tag{2}$$

We will make the following three assumptions regarding return sources. First, for every (s,a), the return source $\boldsymbol{X}(s,a) = \{X_i(s,a) | i = 1,2,\dots\}$ is drawn independently according to parametric probability distribution

$$p_{\theta(s,a)}\left(x_i(s,a)\right) = \Pr\left(X_i(s,a) = x_i(s,a)\right) \tag{3}$$

where $\theta(s,a) = (\theta_1(s,a), \theta_2(s,a), \dots, \theta_k(s,a))$ denotes the k-dimensional parameter vector of the distribution $p_{\theta(s,a)}$ in a compact set $\Theta \subset \Re^k$. Second, the model set $\mathcal{M}_k = \{p_{\theta(s,a)} | \theta(s,a) \in \Theta\}$ of probability distributions includes the true probability distribution. Note that the discussion in this section similarly holds even if we do not assume it. Third, the return source satisfies the ergodic theorem due to Birkhoff [6]. In short, this means that it is possible to estimate the true parameter from a large number of trials. Otherwise, we cannot gather sufficient information to identify the parameter, no matter how many returns are observed. For notational simplicity, we drop (s,a), henceforth, for example we use x, X, $\boldsymbol{X}$, and θ.

B. TD Learning for Information Gain Estimation

To acquire the optimal policy to maximize the return, the agent has to accurately estimate the Q-functions before exploitation. The information gain to be received in future is an important value for estimation, particularly in early stages, because it tells us how taking an action contributes to refining the current estimates. We will define the information gain as the discounted sum of the coding rates later.

Consider a coding algorithm for the return source $\boldsymbol{X}$, so that we can obtain the coding rate that means the amount of information on the return. In order for the algorithm to apply to the framework of reinforcement learning, it should work online and its coding rate should asymptotically converge to the entropy rate. We accordingly employ Rissanen's predictive coding [7] for calculating the coding rate and give a TD learning for estimating the information gain expressed by the discounted sum of the coding rates.

The predictive coding algorithm sequentially encodes one-by-one each return x_i in the sequence x^n for any fixed state-action (s,a). For $i \geq 1$, the algorithm finds the maximum-likelihood (ML) estimate $\hat{\theta}^{(i-1)}$ from the observed return sequence x^{i-1} and calculates the conditional probability distribution

$$p_{\hat{\theta}^{(i-1)}}(x_i | x^{i-1}) = \frac{p_{\hat{\theta}^{(i-1)}}(x^i)}{p_{\hat{\theta}^{(i-1)}}(x^{i-1})} \tag{4}$$

using $\hat{\theta}^{(i-1)}$. Since the return source is independently distributed, the probability distribution is rewritten as

$$p_{\hat{\theta}^{(i-1)}}(x_i | x^{i-1}) = p_{\hat{\theta}^{(i-1)}}(x_i). \tag{5}$$

We will use the convention that $\log y \overset{\text{def}}{=} \log_2 y$ for arbitrary nonnegative y. The codeword length of the ith return x_i is then

$$l(x_i) = -\log p_{\hat{\theta}^{(i-1)}}(x_i). \tag{6}$$

Therefore, the total codeword length of the sequence is written as $l(x^n) = \sum_{i=1}^n l(x_i)$. By taking its expectation, we have

$$L(X^n) = \sum_{i=1}^n E\left[l(X_i)\right]. \tag{7}$$

Under the assumptions, the total codeword length is asymptotically equal to what is called the stochastic complexity given by

$$L(X^n) = \mathrm{H}(X^n) + \frac{k}{2}\log n + o(1) \tag{8}$$

where $\mathrm{H}(\cdot)$ denotes the entropy. For the proof, see [8, pp. 231–233]. We see that the coding rate $L(X^n)/n$ converges to the entropy rate of the return source as $n \to \infty$. Note that instead of the above predictive (one-step) coding we can employ the two-step coding [8, Ch. 7] composed of the two scans for computing the ML estimate $\hat{\theta}$ of the whole sequence x^n and again for computing $p_{\hat{\theta}}(x^n)$. However, such operations take unnecessarily too much time and memory for storing the whole sequence when n is large. Accordingly, we extend the predictive coding form to TD methods.

Using the above predictive coding idea, let us formulate TD-learning algorithms, called "ℓ learning" in this paper, for the purpose of approximating the mean of the information gain. Since we cannot directly observe the return x in practice, we encode the return estimate $\hat{x}$ instead of x. The parameter estimate $\hat{\theta}$ is also calculated using TD methods. We denote the estimate of the Q-function by Q. Let γ_Q be the discount factor for the value of Q and γ_ℓ be the discount factor for the value ℓ of the information gain. The information gain $\ell(s_t,a_t)$ is expressed as the discounted sum of the amount of information

$$\ell(s_t,a_t) = -\sum_{i=0}^{\infty} \gamma_\ell^i \log p_{\hat{\theta}_{t+i}}\left(\hat{x}(s_{t+i}, a_{t+i})\right) \tag{9}$$

that is expected to be received in the future. We describe the ℓ-learning algorithms under the one-step versions of two typical TD methods, Q-learning [9] and Sarsa [1, Ch. 6]. The ℓ-learning algorithms take a similar approach to the Q-learning. The algorithms can be readily extended to two-step or more versions.

1) ℓ-Learning Under Q-Learning: For each time step t, given a one-step episode $(s_t, a_t, r_{t+1}, s_{t+1})$, Q-learning has the update form

$$Q(s_t,a_t) \leftarrow Q(s_t,a_t) + \alpha_t \delta Q_t \tag{10}$$

where the learning rate α_t is set within [0,1] and

$$\delta Q_t = r_{t+1} + \gamma_Q \max_{a' \in \mathcal{A}(s_{t+1})} Q(s_{t+1}, a') - Q(s_t,a_t). \tag{11}$$

With the estimate $\hat{\theta}_t = \hat{\theta}(s_t, a_t)$ of the parameter vector at time step t, the information gain is updated according to the rule

$$\ell(s_t, a_t) \leftarrow \ell(s_t, a_t) + \alpha_t \delta\ell_t \tag{12}$$

where

$$\begin{aligned}\delta\ell_t =& -\log p_{\hat{\theta}_t}\left(\hat{x}(s_t, a_t)\right)\\ &+ \gamma_\ell \max_{a' \in \mathcal{A}(s_{t+1})} \ell(s_{t+1}, a') - \ell(s_t, a_t) \qquad (13)\\ \hat{x}(s_t, a_t) =& r_{t+1} + \gamma_Q \max_{a' \in \mathcal{A}(s_{t+1})} Q(s_{t+1}, a'). \qquad (14)\end{aligned}$$

For the asymptotic behavior, see Appendix I. Under some conditions of ℓ and the convergence conditions of Q-learning, the value of ℓ converges to the expected value (see Appendix II). If $\hat{\theta}$ is the true parameter and $\gamma_\ell = 0$, then the information gain converges to the entropy rate of the return source.

2) ℓ-Learning Under Sarsa: For each time step t, given a one-step episode $(s_t, a_t, r_{t+1}, s_{t+1}, a_{t+1})$, Sarsa under a policy π has the update form

$$Q^\pi(s_t, a_t) \leftarrow Q^\pi(s_t, a_t) + \alpha_t \delta Q_t^\pi \tag{15}$$

where

$$\delta Q_t^\pi = r_{t+1} + \gamma_Q Q^\pi(s_{t+1}, a_{t+1}) - Q^\pi(s_t, a_t). \tag{16}$$

With the estimate $\hat{\theta}_t = \hat{\theta}(s_t, a_t)$ of the parameter vector at time step t, the information gain is updated according to the rule

$$\ell(s_t, a_t) \leftarrow \ell(s_t, a_t) + \alpha_t \delta\ell_t \tag{17}$$

where

$$\begin{aligned}\delta\ell_t =& -\log p_{\hat{\theta}_t}\left(\hat{x}(s_t, a_t)\right)\\ &+ \gamma_\ell \max_{a' \in \mathcal{A}(s_{t+1})} \ell(s_{t+1}, a') - \ell(s_t, a_t) \qquad (18)\\ \hat{x}(s_t, a_t) =& r_{t+1} + \gamma_Q Q^\pi(s_{t+1}, a_{t+1}). \qquad (19)\end{aligned}$$

In general, the convergence is dependent on the policy π. The value of Q^π converges to the expected value under certain conditions and policies due to Singh *et al.* [10]. The value of ℓ also converges to the expected value under the same conditions and some conditions of ℓ discussed in Appendix II, because the value of ℓ is derived from the value of Q^π. Given the true parameter and $\gamma_\ell = 0$, ℓ converges to the entropy rate of the return source.

III. An Example of Applications

In this section, we consider a criterion useful for probabilistic action-selection strategies using the information gain. We begin with the review of dilemma between exploration and exploitation in action-selection strategies.

A. Exploration-Exploitation Dilemma

Within the framework of reinforcement learning, an agent learns a policy based only on the immediate reward and the subsequent state. This means that the learning is influenced by the distribution of episodes that have been observed. If the agent always selects actions that maximize current estimates of the Q-function, then the agent often favors actions with high-return estimates in early stages of learning, while failing to notice other actions that may exhibit even higher returns. This leads us to the question of what strategy of action selection is most effective in learning. In other words, the agent faces a tradeoff in choosing which should be favored, "exploration" to gather new information of the environment or "exploitation" to maximize the return using the knowledge already collected. This problem is well known as the exploration–exploitation dilemma [1, Sec. 1.1]. The subject has been widely studied in the community of reinforcement learning. See [11], [12], for example. It is common to try and control the dilemma by using a probabilistic approach for action selection. Here we propose a new criterion w for more efficient probabilistic action-selection strategies as an example of using the information gain. As a typical strategy, the following softmax method is widely known and has been used in many cases.

1) (Q-Based) Softmax Method: Let $\pi(s, a)$ be the probability that the agent chooses an action a in a state s. The softmax selection is written as

$$\pi(s, a) \overset{\text{def}}{=} \frac{\mathrm{g}\left(\frac{Q(s,a)}{\tau}\right)}{\sum_{a' \in \mathcal{A}(s)} \mathrm{g}\left(\frac{Q(s,a')}{\tau}\right)} \tag{20}$$

where g is a nonnegative and monotone increasing function. When $\mathrm{g}(\cdot) = \exp(\cdot)$, it is called the "Boltzmann selection." The temperature parameter τ is gradually decreased over time for promoting an exploitation strategy. In practice, it is difficult to tune the temperature parameter without any prior knowledge of the values of Q.

B. Criterion w for Action Selection Strategy

Typical "Q-based" selections refer only to the estimates of the Q-function as (20) indicates. The estimate is informative for exploitative strategies, but not for exploratory strategies. Hence, we introduce a new criterion effective for both strategies utilizing the fact that the coding rate is written as the sum of the essential uncertainty and the distance between the true and the estimated distributions. This is based on the idea that the strategy should make decisions taking into account the long-run return loss and the information gain. Recall that we can get the information gain concerning future information using the ℓ-learning algorithm. We find the optimal policy via a strategy based on the neat ratio of return loss to information gain, for any state-action pair

$$w(s, a) \overset{\text{def}}{=} \frac{\eta(s, a)}{\ell(s, a)} \tag{21}$$

where the loss function is

$$\eta(s, a) \overset{\text{def}}{=} \max_{a' \in \mathcal{A}(s)} Q(s, a') - Q(s, a). \tag{22}$$

Note that $w(s, a) = 0$ for action $a = \arg\max_{a' \in \mathcal{A}(s)} Q(s, a')$. The smaller the criterion $w(s, a)$ is, the better the state-action pair (s, a) is in both exploration and exploitation. By setting a large value as the initial value of ℓ, during early stages the information gain ℓ is large compared to the loss function η since the

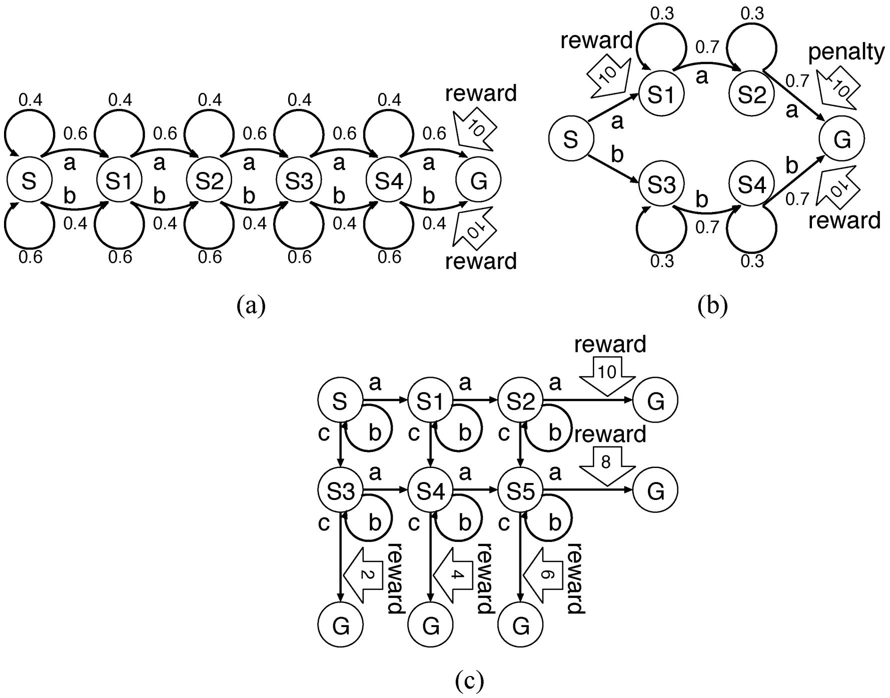


Fig. 1. Each domain is governed by a Markov decision process. Each circle expresses a state and each narrow arrow represents a state transition. The number associated with each narrow arrow is the probability of the state transition. The letters "a", "b", and "c" denote the action available in each state. The reward and the penalty are indicated by the value of wide arrow. (a) Shortcut domain. (b) Misleading domain. (c) Deterministic domain.

estimated parameter is far from the true parameter, that is, the redundancy of the coding rate is large. A large initial value works only to prevent from dividing zero and to give priority to choice of actions which have never been selected. Hence, taking action $a \neq \arg\max_{a' \in \mathcal{A}(s)} Q(s, a')$ that exhibits the smaller value of $w(s, a)$, where $w(s, a) > 0$ is a more efficient exploration so that the agent get a larger amount of information for estimating the Q-functions. As the estimated parameter tends to the true parameter the information gain ℓ goes to a constant value and the value of w is determined mainly by the loss function value. Therefore, taking action a that shows the smaller value of $w(s, a)$ is better because it yields a smaller loss, in other words, a higher return. Again, we see that for any state the action which gives the minimum value $w(s, a) = 0$ is always consistent with the best action $a = \arg\max_{a' \in \mathcal{A}(s)} Q(s, a')$ at each stage. Accordingly, by assigning higher probabilities to actions with the smaller value of w we perform an efficient exploration in early stages and a good exploitation in later stages.

Let us confirm how the criterion w behaves as the number of time steps increases. For $t \in \mathcal{T}$ and any pair (s, a), we use $w_t(s, a)$, $\eta_t(s, a)$, and $\ell_t(s, a)$ to denote the values at time step t. Let $I_t(s, a)$ be the event indicator function that the pair (s, a) occurs at time step t. For any (s, a) the evolution has the form

$$\begin{aligned} w_{t+1}(s,a) &= \frac{\eta_{t+1}(s,a)}{\ell_{t+1}(s,a)} I_t(s,a) + \frac{\eta_t(s,a)}{\ell_t(s,a)} \left(1 - I_t(s,a)\right) \\ &= \frac{\eta_t(s,a) + \delta\eta_t(s,a)}{\ell_t(s,a) + \delta\ell_t(s,a)} I_t(s,a) \\ &\quad + \frac{\eta_t(s,a)}{\ell_t(s,a)} \left(1 - I_t(s,a)\right). \end{aligned} \tag{23}$$

Define $\varepsilon_t \overset{\text{def}}{=} 1/\ell_t(s, a)$. Then the form is rewritten as

$$w_{t+1}(s,a) = w_t(s,a) + \varepsilon_t \frac{\delta\eta_t(s,a) - w_t(s,a)\delta\ell_t(s,a)}{1 + \varepsilon_t \delta\ell_t(s,a)} I_t(s,a). \tag{24}$$

The numerator $\delta\eta_t(s, a) - w_t(s, a)\delta\ell_t(s, a)$ in the second term characterizes the behavior of time evolution. If $\delta\eta_t(s, a) > w_t(s, a)\delta\ell_t(s, a)$, then $w_{t+1}(s, a)$ becomes larger. This means that the action a is penalized since the obtained information gain at time step t is few compared to the payed return loss judged from the current ratio $w_t(s, a)$. Otherwise, $w_{t+1}(s, a)$ becomes smaller so that taking the action a is encouraged for a good information gain in efficiency. Hence, while t is still small, taking actions which contribute to estimating the Q-functions is encouraged and as t increases the best action a which $w(s, a) = 0$ holds becomes preferred.

Now, taking the minus value of w and then applying it to the form of the softmax method, we have

$$\pi(s,a) \overset{\text{def}}{=} \frac{\mathrm{g}\left(\frac{-w(s,a)}{\tau}\right)}{\sum_{a' \in \mathcal{A}(s)} \mathrm{g}\left(\frac{-w(s,a')}{\tau}\right)}. \tag{25}$$

Notice that smaller values of w assign higher probabilities to actions. We call this the "Boltzmann selection" when $\mathrm{g}(\cdot) = \exp(\cdot)$. Since the values of w are automatically tuned during the learning process, this alleviates some troubles associated with tuning τ.

C. Experiments

We have examined the performance of the Q-based and the w-based Boltzmann selections. For simplicity, we tested them on three domains of a Markov decision process (see Fig. 1).

TABLE I
PARAMETERS m IN EACH DOMAIN

Strategy	Shortcut	Misleading	Deterministic
Q-based Boltzmann Selection	$m = 50$	$m = 50$	$m = 30$
w-based Boltzmann Selection	$m = 1$	$m = 1$	$m = 1$

In these figures, the circle and the narrow arrow are the state of the environment and the state transition, respectively. The number associated with each narrow arrow represents the probability of state transition, and the letters "a", "b", and "c" denote the action available in each state. Each wide arrow represents a scalar reward or a penalty. During each episode, the agent begins at the initial state "S", and is allowed to perform actions until it reaches the goal state "G". For every state-action pair (s, a), we used the normal distribution $p_{\theta(s,a)}$ with the parameter $\theta(s, a) = (Q(s, a), v(s, a))^T$ where $Q(s, a)$ and $v(s, a)$ denote the mean return and its variance, respectively, and initialized as $\theta(s, a) = (0, 1)^T$. Here, T denotes transposition. Let the integer $n(s, a)$ denote the number of times that the state-action pair (s, a) has been tried. The agent learns Q, v, and ℓ by the tabular versions of the one-step Q-learning, variance learning [13], and ℓ-learning, respectively

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha_{n(s,a)} \delta Q_t \tag{26}$$

$$\delta Q_t = r_{t+1} + \gamma_Q \max_{a' \in \mathcal{A}(s_{t+1})} Q(s_{t+1}, a') - Q(s_t, a_t) \tag{27}$$

$$v(s_t, a_t) \leftarrow v(s_t, a_t) + \alpha_{n(s,a)} \delta v_t \tag{28}$$

$$\delta v_t = (\delta Q_t)^2 + \gamma_Q^2 v\left(s_{t+1}, \arg \max_{a' \in \mathcal{A}(s_{t+1})} Q(s_{t+1}, a')\right) - v(s_t, a_t) \tag{29}$$

$$\ell(s_t, a_t) \leftarrow \ell(s_t, a_t) + \alpha_{n(s,a)} \delta \ell_t \tag{30}$$

$$\delta \ell_t = - \log p_{\hat{\theta}_t} \left(\hat{x}(s_t, a_t)\right) + \gamma_\ell \max_{a' \in \mathcal{A}(s_{t+1})} \ell(s_{t+1}, a') - \ell(s_t, a_t) \tag{31}$$

where the learning rate is $\alpha_{n(s,a)} = 20/(100 + n(s, a))$, the discount factors are $\gamma_Q = \gamma_\ell = 0.95$, and $\hat{x}(s_t, a_t)$ is given by (14). For every state-action pair (s, a), the initial information gain was set as $\ell(s, a) = 50$ to prevent from dividing by zero in the calculation of $w(s, a)$ during early phases of the learning process.[1] We applied the function $\mathrm{g}(\cdot) = \exp(\cdot)$ to each softmax method, namely, we used the Boltzmann selection. In order to smoothly shift the strategy from exploration to exploitation, the temperature of each strategy was decreased as $\tau_{n(s)} = m \times 100/(n(s)+1)^2$ where the integer $n(s) = \sum_{a \in \mathcal{A}(s)} n(s, a)$ is the number of times that the state s has been visited and the parameter m was tuned as appropriately for each domain as possible. The values of m are shown in Table I.

Fig. 1(a) shows a shortcut task that consists of five states; two actions for each state and one goal. Each action has a similar value so that it is difficult to decide which action is better. The optimal policy for this domain is to choose action "a" everywhere. In this domain, the return of each episode is a constant value "10" regardless of the agent's strategy. The key point here is to find the most efficient policy that allows the agent to reach the goal as quickly as possible. The next domain, as shown in Fig. 1(b), is a misleading domain again composed of five states; two actions for state S and one goal. This has a suboptimal policy that the agent tends to accept. At first sight choosing action "a" looks better because of the reward at the start of the episode, but the reward is finally offset by the penalty. The point is to avoid the suboptimal policy as soon as possible so that the optimal policy taking "b" is encouraged. Finally, the largest domain shown in Fig. 1(c) is a deterministic domain where state transitions are deterministic. This consists of six states; three actions for each state and five goals each with a different reward. Here the problem is that the agent attempts to select the best of the goals without performing sufficient exploration. The optimal policy is to select action "a" everywhere. There are several ways for measuring the learning performance of a particular strategy. To measure the efficiency of a strategy during a fixed number of episodes, we evaluate the learning performance by three measures, collected total return, its standard deviation, and return per episode. However these measures are not suitable for the shortcut domain because the agent always receives the same return. For this reason, we evaluate the performance in this domain by measuring the return "per step," using what is called the "return for effort" principle. The return per step/episode also yields an analysis of how the efficiency of each strategy changes as the number of episodes increases.

The results of the total return and its standard deviation are shown in Table II. Fig. 2 shows the results of the return per step/episode measure. In addition, in the shortcut domain, the Q-based and the w-based strategies took 930.4 steps and 912.72 steps per trial, respectively. These results are averaged over 10 000 trials and each trial consists of 100 episodes. From the total steps and return results, we find that the w-based strategy is better than the conventional Q-based strategy in terms of policy optimization. The results of the standard deviation, especially in deterministic domain, show that the w-based strategy also has a superiority in stability. From the results of the return per step/episode, we see that the w-based strategy is more efficient without unnecessary excessive exploration in early stages. This suggests that our criterion plays a role for avoiding fruitless exploration. The point here is that the information gain becomes small according to the complexity of the probabilistic structure of the domain. If the probabilistic structure is simple, then the information gain decreases quickly and vice versa. Therefore, the agent can efficiently explore the domain. Furthermore, in Table I, we confirm that some troubles regarding tuning are alleviated because the values of m in the w-based strategy are constant regardless of the value of return given in each domain. Thus, the proposed criterion w is a good criterion for strategies of a probabilistic action-selection, yet it is simple.

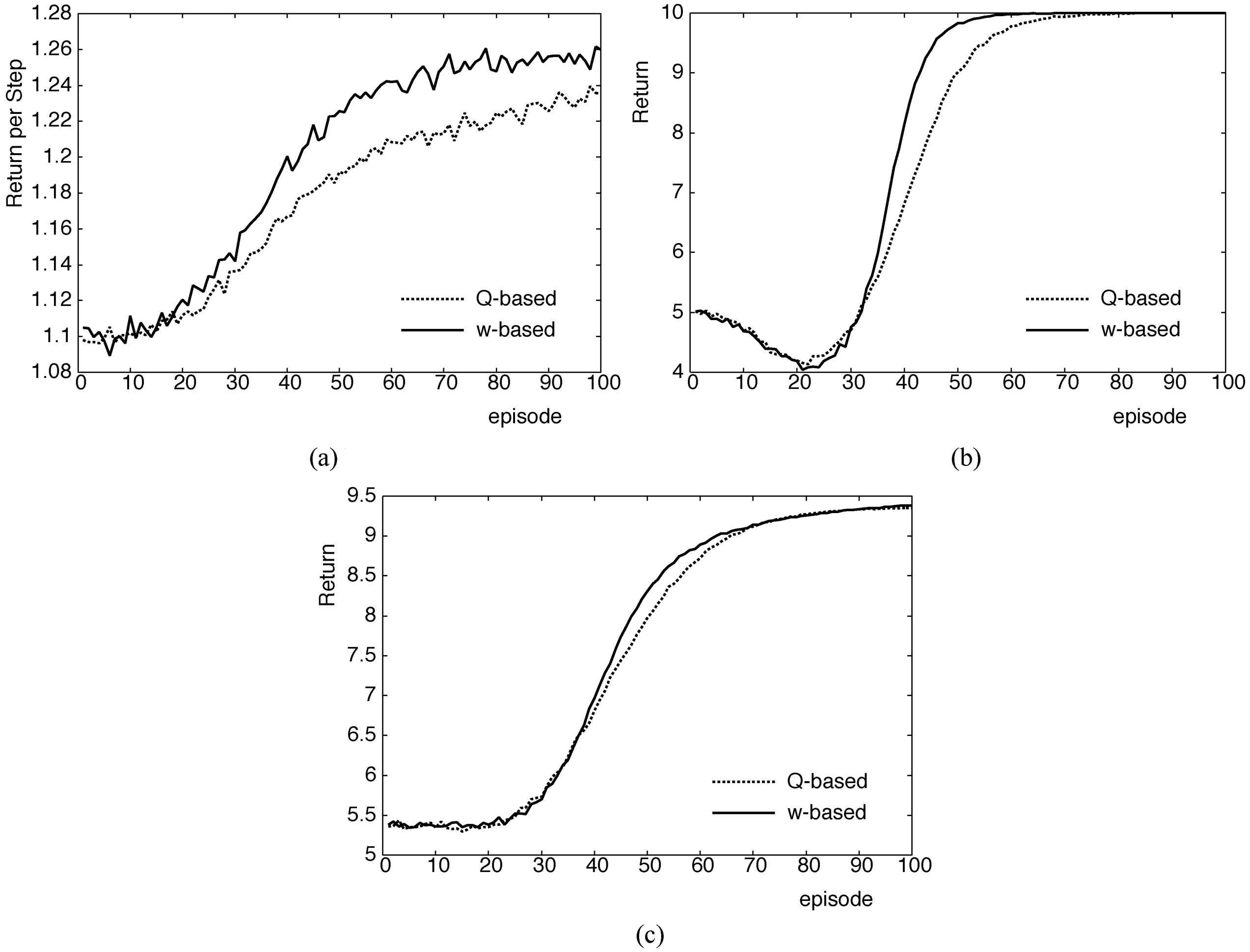
IV. DISCUSSIONS AND CONCLUSION

Our idea is based on the assumption that the return source is described by a parametric distribution. It is possible that by assuming a proper distribution we can achieve more efficient learning, because the prior knowledge about the distribution can

[1] Of course, any large value can be set as the initial value.

TABLE II
TOTAL RETURN PER TRIAL (MEAN AND STANDARD DEVIATION)

Strategy	Shortcut		Misleading		Deterministic	
	mean	s.d.	mean	s.d.	mean	s.d.
Q-based Boltzmann Selection	1000	0	769.46	37.38	753.26	99.23
w-based Boltzmann Selection	1000	0	791.59	31.84	759.8	64.07



(a)

(b)

(c)

Fig. 2. These are simulation results of return (or per step) measure, averaged over 10 000 trials. Each trial consists of 100 episodes. The x-axis and the y-axis are the number of episodes and the collected return (or per step) during each episode, respectively. These represent a change of the efficiency of each strategy in the run. (a) Return per step in shortcut domain. (b) Return in misleading domain. (c) Return in deterministic domain.

be incorporated into the learning process. At the same time, we need to examine the case where there is no knowledge of the return source. The key problem is deciding what distribution is appropriate to express the return source. This is well-known as the model selection problem in the field of statistics and a great deal of controversy surrounds this problem [7], [14]–[16].

In the experiments, we assumed that the return source obeys a normal distribution. Given a large amount of both memory and time, it is possible that we can select the distribution more effectively. From the view point of the minimum description length principle, the distribution that minimizes the information gain is the best distribution for describing the source, because minimizing of the information gain corresponds to shortening the total description length of the source. Thus, a better way is to apply a distribution that gives the minimum information gain, by updating the parameter vectors of several distributions. In addition, if the model set $\mathcal{M}$ does not include the true distribution q, the coding rate of the information gain converges to a positive constant value determined by the closest point in $\mathcal{M}$, that is, the point $p_\theta \in \mathcal{M}$ minimizing the divergence $\mathrm{D}(p_\theta \| q) > 0$ as is well-known in information theory. See [8, Ch. 7], for example.

In this paper, we regarded the sequence of returns as outputs from a parametric compound source. We then described the ℓ-learning algorithms based on the predictive coding idea for estimating the expected information gain and gave the convergence proof. As an example of applications, we proposed the ratio w of return loss to information gain as a new criterion for action-selection, and applied it to the softmax strategy. In experimental results, we found that our w-based strategy performs well compared with the conventional Q-based strategy. Finally, it is likely that our ℓ-learning can be applied to a wide area of applications including the strategy of action-selection, and generalization, for example. We would like to consider other applications such as these in future work.

APPENDIX I
ASYMPTOTIC BEHAVIOR OF ℓ-LEARNING UNDER Q-LEARNING

The behavior of the information gain ℓ is not simple, since for any state-action pair (s,a) the time evolution of the sequence $\{\ell_t(s,a)|t \in \mathcal{T}\}$ depends on the time evolution of the sequence $\{\hat{\theta}_t(s,a)|t \in \mathcal{T}\}$ of the parameter vector estimate. However, if the learning rate of the parameter is small, the parameter changes slowly, roughly speaking, we can assume that the parameter is almost constant. We hence introduce the fixed-θ process in order to study an asymptotic behavior of the information gain. The analysis of ℓ-learning under Sarsa is virtually the same.

For simplicity of notation, let Q^* be the expected return (Q-function) and θ be the fixed parameter vector, hereafter. Let $\mathrm{P}(s'|s,a)$ denote the transition probability that taking action a in state s produces a subsequent state s'. Define

$$\ell^*(s,a) \overset{\text{def}}{=} l^*(s,a) + \gamma_\ell \sum_{s' \in \mathcal{S}} \mathrm{P}(s'|s,a) \max_{a' \in \mathcal{A}(s')} \ell^*(s',a') \tag{32}$$

where

$$l^*(s,a) \overset{\text{def}}{=} -E\left[\log p_\theta\left(r_{t+1} + \gamma_Q \max_{a' \in \mathcal{A}(s_{t+1})} Q^*(s_{t+1},a')\right) \middle| s_t = s, a_t = a\right]. \tag{33}$$

For fixed θ, define the expectations, which are conditioned on the minimal σ-algebra $\mathcal{F}_t$ created by the set $\{s_i, a_i, s_t | i = 0,1,\ldots,t-1\}$, by

$$\mathrm{T}\left(\hat{x}(s_t,a_t)\right) \overset{\text{def}}{=} E\left[r_{t+1} + \gamma_Q \max_{a' \in \mathcal{A}(s_{t+1})} Q(s_{t+1},a') | \mathcal{F}_t, a_t\right] \tag{34}$$

and

$$\mathrm{T}(\ell(s_t,a_t)) \overset{\text{def}}{=} E\left[-\log p_\theta\left(r_{t+1} + \gamma_Q \max_{a' \in \mathcal{A}(s_{t+1})} Q(s_{t+1},a')\right) + \gamma_\ell \max_{a' \in \mathcal{A}(s_{t+1})} \ell(s_{t+1},a') | \mathcal{F}_t, a_t\right]. \tag{35}$$

By the Markov property, we can rewrite this as

$$\mathrm{T}\left(\hat{x}(s_t,a_t)\right) = E[r_{t+1}|s_t,a_t] + \gamma_Q \sum_{s' \in \mathcal{S}} \mathrm{P}(s'|s_t,a_t) \max_{a' \in \mathcal{A}(s')} Q(s',a') \tag{36}$$

and

$$\mathrm{T}(\ell(s_t,a_t)) = -E\left[\log p_\theta\left(r_{t+1} + \gamma_Q \max_{a' \in \mathcal{A}(s_{t+1})} Q(s_{t+1},a')\right) | s_t,a_t\right] + \gamma_\ell \sum_{s' \in \mathcal{S}} \mathrm{P}(s'|s_t,a_t) \max_{a' \in \mathcal{A}(s')} \ell(s',a') \tag{37}$$

respectively. Define the noise of $\hat{x}$ by

$$\delta \mathrm{M}_t\left(\hat{x}(s_t,a_t)\right) \overset{\text{def}}{=} \hat{x}(s_t,a_t) - \mathrm{T}\left(\hat{x}(s_t,a_t)\right). \tag{38}$$

With the definition of the noise term

$$\delta \mathrm{M}_t\left(\ell(s_t,a_t)\right) \overset{\text{def}}{=} -\log p_\theta\left(\mathrm{T}\left(\hat{x}(s_t,a_t)\right) + \delta \mathrm{M}_t\left(\hat{x}(s_t,a_t)\right)\right) + \gamma_\ell \max_{a' \in \mathcal{A}(s_{t+1})} \ell(s_{t+1},a') - \mathrm{T}\left(\ell(s_t,a_t)\right) \tag{39}$$

(13) is rewritten as

$$\delta \ell_t = \mathrm{T}\left(\ell(s_t,a_t)\right) + \delta \mathrm{M}_t\left(\ell(s_t,a_t)\right) - \ell(s_t,a_t). \tag{40}$$

Note that $\delta \mathrm{M}_t(\cdot)$ is the martingale difference and the conditioned variance is bounded uniformly in t, namely

$$E\left[\delta \mathrm{M}_t(\cdot)|\mathcal{F}_t, a_t\right] = 0 \tag{41}$$

$$E\left[(\delta \mathrm{M}_t)^2(\cdot)|\mathcal{F}_t, a_t\right] < \infty. \tag{42}$$

As written in [17, Ch. 2], the map T is a Lipschitz continuous contraction with respect to the supremum norm and $\ell^*(s,a)$ is a unique fixed point. Equation (39) suggests that the performance of the ℓ-learning is at most the Robbins–Monro procedure performance [18], because the evolution of the information gain is available only after some updates due to delayed rewards. The convergence speed depends on the propagation delay from occurrence points of reward. Under the convergence conditions of the above and the Q-learning, for every pair (s,a) the value of $\ell(s,a)$ converges to the value of $\ell^*(s,a)$ with probability one, as will be seen in Appendix II.

For any pair (s,a), let $\ell(s,a,\cdot)$ be the piecewise interpolated continuous function of the sequence $\{\ell_t(s,a)|t \in \mathcal{T}\}$ in continuous time. There is also a value of $d_{s,a}(t)$ lying in the interval $[1/\bar{u}_{s,a}, 1]$, where the value of $\bar{u}_{s,a}$ bounds the time interval between occurrences of the pair (s,a). For any pair (s,a), the mean ordinary differential equation that characterizes the limit point is given by

$$\dot{\ell}(s,a,t) = d_{s,a}(t)\left(\mathrm{T}\left(\ell(s,a)\right) - \ell(s,a,t)\right) + z_{s,a} \tag{43}$$

where $z_{s,a}$ works only to hold $|\dot{\ell}(s,a,t)| \le B$ for a large B.

APPENDIX II
ROUGH PROOF OF INFORMATION GAIN CONVERGENCE

The proof that we discuss below is based on the manner due to Kushner and Yin [17, Ch. 12]. Let us show the convergence by describing that the theorem [17, Ch. 12, Th. 3.5] that all $\ell(s,a)$ converge to the limit point $\ell^*(s,a)$ holds under the following conditions. The other conditions that we do not write are either obvious or not applicable for the convergence theorem.

We deal with a practical constrained algorithm in which the value of ℓ is truncated at $\mathcal{H} = [-B, B]$ for large B, and assume a constant learning rate for simplicity of development. The proof for a decreasing learning rate is virtually the same. Define $C = \max_{(s,a) \in \mathcal{S} \times \mathcal{A}} \ell(s,a)$. Let $B > C/(1-\gamma_\ell)$. Recall that $\ell_t(s,a)$ denotes the information gain $\ell(s,a)$ at time step t. The dimension of the problem is determined by the number $b = \|\mathcal{S} \times \mathcal{A}\|$ of state-action pairs and $\{\ell_t(s,a)|t \in \mathcal{T}\}$ existing in $\mathcal{H}^b$.

Let α be a small real constant value and $I_t(s,a)$ be the event indicator function that the pair (s,a) is observed at time step t. Recall that for any (s,a) ℓ-learning algorithm with truncation has the form

$$\ell_{t+1}(s,a) \leftarrow \Pi_{\mathcal{H}}\left[\ell_t(s,a) + \alpha \delta \ell_t I_t(s,a)\right] \tag{44}$$

where $\delta\ell_t$ is given by (40) and $\Pi_{\mathcal{H}}[\cdot]$ denotes the truncation which brings $\cdot$ to the closest point in $\mathcal{H}$ if $\cdot$ goes out of $\mathcal{H}$. Suppose that the state transition process is reducible with probability one. Let $t_{s,a}(n+1)$ be the time step that the $(n+1)$st update for the pair (s,a) is done, and let $\tau_{s,a}(n)$ denote the time interval between the nth and the $(n+1)$st occurrences of the pair (s,a). Define the expectation of the time interval

$$u_{s,a}(n+1) = E\left[\tau_{s,a}(n+1)|\mathcal{F}_{t_{s,a}(n+1)}\right]. \tag{45}$$

We assume that for any nonnegative n the value of $u_{s,a}(n)$ is uniformly bounded by a real number $\bar{u}_{s,a} \geq 1$ and that the time evolution of the sequence $\{\tau_{s,a}(n)|n\}$ is uniformly integrable. Let $\ell(s,a,\cdot)$ denote the piecewise constant interpolation [17, Ch. 4] of the sequence $\{\ell_t(s,a)|t \in \mathcal{T}\}$ in "scaled real" time, that is, with interpolation intervals of width α. Under the given conditions, $\{\ell(s,a,T_\alpha+\cdot)|T_\alpha \in \Re\}$ is tight [17, Ch. 7 and 8] for any sequence of real numbers T_α, so under Q-learning convergence conditions [9] we can show that if T_α tends to infinity, then as α tends to zero it exhibits a weak convergence to the process with constant value defined by (32). The process is written as the ordinary differential equation given by (43). Now we will show that all solutions of (43) tend to the unique limit point given by (32). Suppose that $\ell(s,a,t) = B$ for some t and pair (s,a). By the bound on B

$$\begin{aligned}\sup_{\ell(s,a)=B} & \left[\mathrm{T}\left(\ell(s,a)\right) - \ell(s,a)\right] \\ & \leq C + \gamma_\ell \sum_{s' \in \mathcal{S}} \mathrm{P}(s'|s,a)B - B \\ & = C - (1-\gamma_\ell)B < 0. \end{aligned} \tag{46}$$

This means

$$\begin{cases} \dot{\ell}(s,a,t) < 0, & \text{if } \ell(s,a,t) = B \\ \dot{\ell}(s,a,t) > 0, & \text{if } \ell(s,a,t) = -B \end{cases}. \tag{47}$$

Hence, the boundary of $\mathcal{H}^b$ is not accessible by a trajectory of the ordinary differential equation given by (43) from any interior point. From the contraction property of T and by neglecting $z_{s,a}$, for every (s,a) the value of $\ell^*(s,a)$ is the unique limit point of (43). Taking into account that (43) is the limit mean ordinary differential equation, all the conditions for the convergence theorem is satisfied. Accordingly, the convergence proof is complete. ■

REFERENCES

[1] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, ser. Adaptive Computat. Mach. Learn. Cambridge, MA: MIT Press, Mar. 1998.

[2] A. Likas, "A reinforcement learning approach to online clustering," *Neural Computat.*, vol. 11, no. 8, pp. 1915–1932, 1999.

[3] W. Zhang and T. G. Dietterich, "A reinforcement learning approach to job-stop scheduling," in *Proc. 14th Int. Joint Conf. Artificial Intelligence*, C. S. Mellish, Ed., Montreal, Canada, 1995, pp. 1114–1120.

[4] M. Sato and S. Kobayashi, "Variance-penalized reinforcement learning for risk-averse asset allocation," in *Proc. 2nd Int. Conf. Intelligent Data Engineering Automated Learning*, vol. 1983, K. S. Leung, L.-W. Chan, and H. Meng, Eds., Hong Kong, China, 2000, pp. 244–249.

[5] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement learning: A survey," *J. Artif. Intell. Res.*, vol. 4, pp. 237–285, Jan. 1996.

[6] P. Billingsley, *Probability and Measure*, 3rd ed, ser. Wiley Ser. Probability Math. Statist. New York: Wiley, Apr. 1995.

[7] J. Rissanen, "Stochastic complexity and modeling," *Ann. Statist.*, vol. 14, no. 3, pp. 1080–1100, 1986.

[8] T. S. Han and K. Kobayashi, *Mathematics of Information and Coding*, ser. Translations of Mathematical Monographs. Providence, RI: Amer. Math. Soc., 2002, vol. 203.

[9] C. J. C. H. Watkins and P. Dayan, "Technical note: Q-learning," *Mach. Learning*, vol. 8, pp. 279–292, 1992.

[10] S. Singh, T. Jaakkola, M. L. Littman, and C. Szepesvári, "Convergence results for single-step on-policy reinforcement-learning algorithms," *Mach. Learning*, vol. 39, pp. 287–308, 2000.

[11] L. P. Kaelbling, *Learning in Embedded Systems*. Cambridge, MA: MIT Press, 1993.

[12] R. Dearden, N. Friedman, and S. Russell, "Bayesian Q-learning," in *Proc. 15th Nat. Conf. Artificial Intelligence*, Madison, WI, July 1998, pp. 761–768.

[13] M. Sato and S. Kobayashi, "Average-reward reinforcement learning for variance penalized Markov decision problems," in *Proc. 18th Int. Conf. Machine Learning*, C. E. Brodley and A. P. Danyluk, Eds., San Francisco, CA, June 2001, pp. 473–480.

[14] H. Akaike, "A new look at the statistical model identification," *IEEE Trans. Automat. Contr.* , vol. AC-19, pp. 716–723, Dec. 1974.

[15] J. Rissanen, "Modeling by shortest data description," *Automatica*, vol. 14, pp. 465–471, 1978.

[16] ——, "A universal prior for integers and estimation by minimum description length," *Ann. Statist.*, vol. 11, pp. 416–431, 1983.

[17] H. J. Kushner and G. G. Yin, *Stochastic Approximation Algorithms and Applications*, ser. Applications of Mathematics. New York: Springer-Verlag, 1997, vol. 35.

[18] H. Robbins and S. Monro, "A stochastic approximation method," *Ann. Math. Statist.*, vol. 22, pp. 400–407, 1951.

Kazunori Iwata (S'04) received the B.E. and M.E. degrees from Nagoya Institute of Technology, Nagoya, Japan, in 2000 and 2002, respectively. He is currently working toward the Ph.D. degree from the Graduate School of Informatics, Kyoto University, Japan.

His research interests include machine learning, statistical inference, and information theory.

Mr. Iwata is currently a Fellow of the Japan Society for the Promotion of Science (JSPS).

Kazushi Ikeda (M'96) was born in Shizuoka, Japan, in 1966. He received the B.E., M.E., and Ph.D. degrees in mathematical engineering and information physics from the University of Tokyo, Tokyo, Japan, in 1989, 1991, and 1994, respectively.

From 1994 to 1998, he was with the Department of Electrical and Computer Engineering, Kanazawa University, Kanazawa, Japan. Since 1998, he has been with the Department of Systems Science, Kyoto University, Kyoto, Japan. His research interests are focused on adaptive and learning systems, including neural networks, adaptive filters, and machine learning.

Hideaki Sakai (M'78–SM'02) received the B.E. and Dr.Eng. degrees in applied mathematics and physics from Kyoto University, Kyoto, Japan, in 1972 and 1981, respectively.

From 1975 to 1978 he was with Tokushima University. He spent six months from 1987 to 1988 at Stanford University, CA, as a Visiting Scholar. He was an Associate Editor of *IEICE Transactions Fundamentals of Electronics, Communications, and Computer Sciences* from 1996 to 2000 and is currently on the editorial board of *EURASIP Journal of Applied Signal Processing*. He is currently a Professor in the Department of Systems Science, Graduate School of Informatics, Kyoto University. His research interests are in the area of statistical and adaptive signal processing.

Dr. Sakai was an Associate Editor of IEEE TRANSACTIONS ON SIGNAL PROCESSING from 1999 to 2001.