Learning Automata with 2-State Bayesian Estimators

Mothoshi Hara †, Wataru Aoto ††, Noriyo Kanayama †, Toru Watanabe †, Hiroyuki Kamaya ‡

† National Institute of Technology, Matsue College
†† Internet Initiative Japan Inc.
‡ National Institute of Technology, Hachinohe College

† 14-4 Nishi-ikuma, Matsue, Shimane 690-8518, Japan
E-mail: hara@matsue-ct.jp

Abstract

This paper presents a novel learning automaton, β-type, which consists of 2-state Bayesian estimators. The β-type learning automaton is presently among the fastest learning automata known, which was proposed in our earlier works. However, compared with the β-type learning automaton and the conventional learning automata, the β-type learning automaton deteriorates from the viewpoint of memory usage and other resources. For example, since computational and energy resources of some applications are limited, such as the wireless sensor networks, reducing memory footprint and performance optimization are very important issues. So, in this study, we propose the β-type learning automaton with minimum resources, 2-state Bayesian estimators. Then, the efficiency of proposed β-type learning automaton is shown through several simulation results under some random environments.

1 Introduction

In recent years, the field of the artificial intelligence study, which is including the deep learning, game AI and so on shows remarkable progress by the spread of high-performance computers[1],[2]. Especially, the deep Q-Networks(DQN), which is classified into the deep learning, combines Q-learning with a deep neural network and suffers from substantial overestimations in some computer games. As is generally known, the theory of learning automata forms the basis of the reinforcement learning including such as the Q-learning[3].

The theory of stochastic learning automaton(LA) is yet another AI-based methodology, which is the origin from the M.L.Tsetlin’s pioneering work in 1962[4]. The LA, which operates in random environments, have been extensively studied over past 4 decades[5],[6]. We also have proposed a novel family of LA termed β-type, which consists of several Bayesian estimators, indicates the property of conditional optimal under some stationary random environments[7].

The β-type LA is presently among the fastest learning automata known, which have some reinforcement schemes, such as $L_{R-1}$, $L_{R-P}$, $N_{R-P}$, $HL_{R-1}$, $SE_{R-1}$ and so on. However, compared with the β-type LA and the conventional LAs, the β-type one deteriorates from the viewpoint of memory usage and other resources. For example, since computational and energy resources of sensor nodes are limited in the wireless sensor networks(WSNs), reducing memory footprint and performance optimization are very important issues[8],[9].

So, in this study, we propose the β-type LA with minimum resources, 2-state Bayesian estimators. Then, the efficiency of proposed β-type LA is shown through several simulation results under some random environments.

2 Learning Automaton

Learning automaton(LA) operating in an unknown random environment have been proposed earlier as learning models of living organisms[6]. The LA is a kind of stochastic variable structure automaton, it updates its action probabilities in accordance with the environment responses derived from the random environment by using reinforcement scheme(learning algorithm). If the reinforcement scheme is selected properly, the LA can asymptotically chose the optimal action in the random environment. A model of random environment and learning automaton is shown as the Figure.1. In this chapter, we describe briefly the LA as follows.

![Automaton-Environment Configuration](Fig. 1: Automaton-Environment Configuration)
(1) The Random Environment

Mathematically, a random environment \(RE\) is defined by a 3-tuple \(RE = \langle \omega, X, f \rangle\). The \(P\)-model or \(Q\)-model \(RE\) has a finite action set \(\omega = \{\alpha_1, \alpha_2, \ldots, \alpha_r\}\) and a response set \(X = \{x_1, x_2, \ldots, x_n\}\). Here, we assume that \(X\) is a set of distinct real numbers which indicates the degree of failure, unfavorable response or penalty.

The action \(\alpha(t)\) at discrete time \(t\) is defined by \(\alpha(t) = \alpha_i \in \alpha\). Where, \(f_i = (f_{i1}, f_{i2}, \ldots, f_{ir})\) is a set of corresponding action. Furthermore, \(f\) indicates the degree of failure, unfavorable response or penalty. The learning automaton determines its internal state \(\omega'(t)\) at time \(t\), which is defined as:

\[
\omega_i(t) = \text{Pr}[\omega'(t) = \omega_i], \quad \sum_{j=1}^{s} \pi_j(t) = 1. \tag{1}
\]

Furthermore, expanding both the output function \(G\) and the state probability vector \(\pi(t)\), a vector of action probabilities \(\phi(t) = (\phi_1(t), \phi_2(t), \ldots, \phi_r(t))\) can be defined. Here, the \(\phi_i(t)\) means

\[
\phi_i(t) = \text{Pr}[\alpha(t) = \alpha_i], \quad \sum_{i=1}^{r} \phi_i(t) = 1. \tag{2}
\]

(2) The Learning Automaton

The learning automaton can be represented by a 6-tuple \(\langle X, \Omega', \omega_i, \pi(t), T, G \rangle\). Where, \(X\) is a response set, \(\omega\) is a action set, \(\Omega' = \{\omega_1', \omega_2', \ldots, \omega_s'\}\) is a set of internal states of learning automaton. \(\pi(t)\) is a state probability vector, which is defined by

\[
\pi_j(t) = \text{Pr}[\omega'(t) = \omega_j'], \quad \sum_{j=1}^{s} \pi_j(t) = 1. \tag{3}
\]

At any time \(t\), a internal state \(\omega'(t)\) is determined by the probability vector \(\pi(t)\). \(T\) is a reinforcement scheme, which updates \(\pi(t)\) to \(\pi(t+1)\) by using \(\alpha(t)\) and \(x(t)\) as follows:

\[
\pi(t+1) = T(\pi(t), \alpha(t), x(t)). \tag{4}
\]

Then, \(G\) is an output function, which can be ether deterministic or stochastic,

\[
\alpha(t) = G[\pi(t)]. \tag{5}
\]

Furthermore, expanding both the output function \(G\) and the state probability vector \(\pi(t)\), a vector of action probabilities \(\phi(t) = (\phi_1(t), \phi_2(t), \ldots, \phi_r(t))\) can be defined. Here, the \(\phi_i(t)\) means

\[
\phi_i(t) = \text{Pr}[\alpha(t) = \alpha_i], \quad \sum_{i=1}^{r} \phi_i(t) = 1. \tag{6}
\]

(3) Learning Automaton and Environment Interaction

An interaction of the learning automaton and random environment is described as follows.

Step 0: At time \(t = 0\), initialize \(\pi(t)\) as follows.

\[
\pi_j(0) = \frac{1}{s}, \quad (j = 1, 2, \ldots, s) \tag{7}
\]

Step 1: The learning automaton determines its internal state \(\omega'(t)\) according to the state probability vector \(\pi(t)\) at time \(t\). Then, it chooses and outputs the action \(\alpha(t)\) by using the output function \(G\).

Step 2: The random environment returns the response \(x(t)\) to the learning automaton as evaluation value for a chosen action \(\alpha(t)\).

Step 3: If the learning automaton satisfies some convergence condition then go to END, else \(t \leftarrow t + 1\) and go to Step 1.

END

In the above algorithm, the process of Step 1 and Step 2 is called “trial”. Generally, in Step 0 at time \(t = 0\), the action probability vector \(\phi(t)\) is initialized as:

\[
\phi_i(0) = \frac{1}{r}, \quad (i = 1, 2, \ldots, r). \tag{8}
\]

If the learning automaton chooses an action \(\alpha(t) = \alpha_i\) at time \(t\), the random environment return the response \(x(t) \in X\) which is belongs to unknown probability distribution \(f_i \in f\). Then, the expected value of \(x(t)\) is represented by

\[
c_i = E[x(t)|\alpha(t) = \alpha_i] = \sum_{j=1}^{n} x_j f_{ij}. \tag{9}
\]

The value \(c_i\) is the evaluation of the action \(\alpha_i\) and it is assumed that a unique minimum of \(c_i (i = 1, 2, \ldots, r)\) exists. The ultimate goal of the learning automaton is to find the action that produces the most lowly expected response (penalty) in iterative manner under the unknown random environment. Furthermore, an optimal action set is denoted by

\[
\alpha_{opt} = \{\alpha_i | c_{i_{opt}} \leq c_i, \forall i\}. \tag{10}
\]
(4) Norms of Behavior
In evaluating the performance of learning automaton, the concepts of optimality and ε-optimality are very important. The optimality means that the learning automaton becomes to asymptotically choose the optimal action or actions with probability one when time goes infinity, and the ε-optimality is a weaker requirement than the optimality. The optimality and ε-optimality are defined as follows.

**Definition 2.1.** A learning automaton is optimal\(^3\), if:

\[
\lim_{t \to \infty} \sum_{\alpha_i \in \Omega_{opt}} \phi_i(t) = 1 \quad \text{a.s.} \quad (8)
\]

**Definition 2.2.** A learning automaton is ε-optimal , if:

\[
\lim_{t \to \infty} E\left[ \sum_{\alpha_i \in \Omega_{opt}} \phi_i(t) \right] \geq 1 - \varepsilon \quad (9)
\]

In the equation (9), \(\varepsilon\) is arbitrarily small positive number.

(5) \(L_{R-I}\) Reinforcement Scheme
In general term, a reinforcement scheme can be represented in the formula (2). If the internal state number \(s\) of the learning automaton is equal to the action number \(r\), the state \(\omega_i^t \in \Omega\) corresponds to the action \(\alpha_i \in \alpha\), \(\forall i\), then the structure of learning automaton becomes very simple. When \(s = r\), the famous reinforcement scheme \(L_{R-I}\) in the P-model environment, which has ε-optimal property, describes as follows\([6]\).

When \(\alpha(t) = \alpha_i \in \alpha\), if \(x(t) = 0\) then
\[
\phi_i(t + 1) = \phi_i(t) + a(1 - \phi_i(t)) \quad (10)
\]
\[
\phi_j(t + 1) = (1 - a)\phi_j(t) \quad (i \neq j). \quad (11)
\]

If \(x(t) = 1\) then
\[
\phi_k(t + 1) = \phi_k(t), \quad \forall k \quad (12)
\]

Where, \(a\) means a reword parameter \(0 < a < 1\).

3 β-type Learning Automaton
In this study, we use the learning automaton termed β-type with the reinforcement scheme which is similar to the Bayesian learning scheme. A learning automaton is connected in a feedback system with an unknown random environment, which produces environmental responses(See Figure 1). We describe briefly the β-type learning automaton model as follows\([7]\).

\[\text{Fig. 2: Construction of β-Type Learning Automaton}\]

A β-type learning automaton consists of \(r\) Bayesian estimators (BE) (See Figure 2), and each \(BE_i(i = 1, 2, \ldots, r)\) corresponds to an action \(\alpha_i \in \alpha\). The \(BE_i(i = 1, 2, \ldots, r)\) is described by a 6-tuple \(\text{<X, Ω, } \lambda_i(t), T, \Omega >\), where \(\Omega = \{\omega_1, \omega_2, \ldots, \omega_m\}\) is the set of its states and \(m\) is state number; \(\lambda_i(t) = (\lambda_{i1}(t), \lambda_{i2}(t), \ldots, \lambda_{im}(t))\) is the state probability vector at time \(t\); \(T\) is the reinforcement scheme defined as below.

The state probability vector satisfies the following conditions:

\[0 \leq \lambda_{ij}(t) \leq 1, \quad \sum_{j=1}^{m} \lambda_{ij}(t) = 1, \quad \forall i, j. \quad (13)\]

For each \(BE_i\), further, we assign a real number \(\mu_k\) to each state \(\omega_k\). These real numbers \(\mu_k(k = 1, 2, \ldots, m)\) are taken to satisfy the following conditions:

\[x < \mu_1 < \mu_2 < \ldots < \mu_m < \bar{x}, \quad (13)\]
\[x = \min_{i}(x_i), \quad \bar{x} = \max_{i}(x_i). \quad (14)\]

The interaction between β-type learning automaton and Q-model stationary environment is explained as follows.

At the start time \(t = 0\), every state probability vectors \(\lambda_i(0)(i = 1, 2, \ldots, r)\) are set as

\[\lambda_{i1}(0) = \lambda_{i2}(0) = \cdots = \lambda_{im}(0) = \frac{1}{m}, \quad \forall i \quad (15)\]

At time \(t\), each \(BE_i(i = 1, 2, \ldots, r)\) chooses randomly its state \(\omega_{ki} \in \Omega\) according to its state probability vector \(\lambda_i(t)\), and generates the output \(\mu_k(t) = \mu_{\omega_{ki}}\).

The β-type learning automaton chooses an action \(\alpha(t) = \alpha_i \in \alpha\) corresponding to the \(BE_i\) which generates the most lowest output and determines \(\mu(t) = \mu_i(t)\) as the output of the β-type learning automaton.

Thus, the β-type one performs action \(\alpha(t)\) to the environment and subsequently receives the penalty \(x(t)\) corresponding to \(\alpha(t)\) as a response from the environment. Then the state probability vector \(\lambda_i(t)\) of the
$BE_i$ corresponding to the chosen action $\alpha(t) = \alpha_i$ is updated by

$$\lambda_i(t+1) = T'(\lambda_i(t), x(t)).$$  \hfill (16)

In above scheme, the $\beta$-type learning automaton changes its probabilistic structure in order to adjust itself to the environment in iterative manner.

The reinforcement scheme $T'$ is described as follows.

$$\lambda_k(t+1) = c\lambda_k(t)\{\mu_k^x(t)(1-\mu_k)^{1-x'}(t)\}^\gamma(k = 1, 2, \cdots, n)$$

where $c$ is the normalizing constant; $\gamma$ is the parameter which dominates the speed of convergence, $x'(t)$ is the normalized response from the environment which lies in the interval $[0,1]$ and is defined by

$$x'(t) = \frac{x(t) - x}{x - 1}. \hfill (17)$$

Particularly when $\gamma$ is equal to 1 the scheme is the same form as the Bayesian learning scheme.

**Theorem 3.1.** Suppose there exists an optimal action $\alpha_{i_0} \in \mathcal{A}_{\text{opt}}$ such that (7) and an integer number $k_0 \in \{1, 2, \cdots, m-1\}$ between $c_{i_0}$ and $c_i$. Then if an inequality is established as

$$c_{i_0} \leq \mu_{i_0} \text{ and } c_i > \mu_{i_0+1} \ (i \neq i_0),$$ \hfill (18)

the $\beta$-type LA is optimal. That is,

$$\lim_{t \to \infty} \phi_{i_0}(t) = 1 \ a.s..$$ \hfill (19)

(The proof is omitted. See [7].)

Where, if we set $\mu_0 = 0.0$ and $\mu_m = 1.0$, then $\mu_k$ is denoted by

$$\mu_k = \frac{\log(\frac{1-\mu_{k+1}}{1-\mu_k})}{\log(\frac{1-\mu_{k+1}}{1-\mu_k}) - \log(\frac{\mu_{k+1}}{\mu_k})} \ (k = 1, 2, \cdots, m-1).$$ \hfill (20)

In the above theorem, the $\beta$-type learning automaton has optimal property if the state number $m$ of the Bayesian estimator takes an enough large, because the difference between $\mu_{k+1}$ and $\mu_k$ can be as small as possible. Therefore, the $\beta$-type LA is classified as conditionally optimal. Furthermore, the $\beta$-type LA is presently among the fastest learning automata known. For example, $\beta$-type LA converges almost an order of magnitude quicker comparing with other conventional reinforcement schemes such as $L_{R-1}$.

However the $\beta$-type LA deteriorates in terms of memory usage and processing power comparing with the other conventional LA. So, in this study, we propose the $\beta$-type LA with 2-state Bayesian estimators, which satisfies the minimum configuration requirements.

### 4 The $\beta$-Type Learning Automaton with 2-state Bayesian Estimators

Here, we consider about the $\beta$-type learning automaton with 2-state Bayesian estimators. The 2-state Bayesian estimator has following properties.

**[Properties of Bayesian Estimator]**

We suppose that the response $x(t)$ at time $t$ lies in the interval $[\underline{x}, \overline{x}]$ and $x(t)$ is normalized into $[0,1]$ by the formula (17). In the Bayesian estimator $BE_i$ with $t$ has the state number $m = 2$, which is corresponding to the action $\alpha_i$, the properties (21) and (22) are described as bellow.

- if $c_i < \underline{x}$, then
  $$\lim_{t \to \infty} \lambda_{i1}(t) = 1 \hfill (21)$$

- if $c_i > \overline{x}$, then
  $$\lim_{t \to \infty} \lambda_{i2}(t) = 1 \hfill (22)$$

According to above properties, we can consider about $\beta$-type LA algorithm with 2-state Bayesian estimators.

**Step 0:** $[\underline{x}, \overline{x}]$ of $BE$ is set arbitrary, then 2-parameters $\varepsilon$ and $\delta$ are introduced. Where, $0 < \varepsilon < 1$, $0 < \delta < \overline{x} - \underline{x}$.

**Step 1:** Each $A_k(t)$ of $BE_k$ is initialized as $\lambda_{k1}(t) = 1/2 \ (t = 1, 2)$.

**Step 2:** Achieve the trial experiment under unknown random environments.

(i) If two or more of $BE_i$ is $\lambda_{i1}(t) > (1 - \varepsilon)$, then
  $$[\underline{x}, \overline{x}] \leftarrow [\underline{x}, \overline{x} - \delta]$$ \hfill (23)

  go to Step 1.

(ii) If all of $BE_i$ are $\lambda_{i2}(t) > (1 - \varepsilon)$, then
  $$[\underline{x}, \overline{x}] \leftarrow [\underline{x} + \delta, \overline{x}]$$ \hfill (24)

  go to Step 1.

**Step 3:** If the learning automaton satisfies some convergence conditions then go to END, else $t \leftarrow t + 1$ and go to Step 1.

**END**

In the above algorithm, we focuses on the properties of (21) and (22) because the $\beta$-type LA can choose asymptotically an optimal action or optimal actions such as (7), if the Bayesian estimators of the $\beta$-type LA are able to distinguish only the difference between the optimal action and second optimal action in the meaning of the expected response value from the environment.
5 Experiments

5.1 Simulation Results

Simulation experiments have been conducted using the model of $\beta$-type LA in Chap.4 to investigate the performance of it.

The experiments were run under the $P$-model environment with action number $r = 10$ which is represented by Table.1.

<table>
<thead>
<tr>
<th>Action $a_i$</th>
<th>$a_1$</th>
<th>$a_2$</th>
<th>$a_3$</th>
<th>$a_4$</th>
<th>$a_5$</th>
<th>$a_6$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Failure Prob. $c_i$</td>
<td>0.3</td>
<td>0.2</td>
<td>0.5</td>
<td>0.8</td>
<td>0.9</td>
<td>0.4</td>
</tr>
</tbody>
</table>

First, the Figure.3 shows the learning curves of $L_{R-I}$ reinforcement scheme with several parameters $a$ which were mentioned in the formula from (10) to (12) and the normal $\beta$-type LA with the state number $m = 50$ and the parameter $\gamma = 1.0$ of the Bayesian estimators. Although the proposed $\beta$-type LA’s result is bit worse than the $SE_{R-I}$, this result shows that the proposed $\beta$-type LA performs better than the $L_{R-I}$ and it is considered that enough performance is obtained in practical use.

![Graph]

Fig. 3: Learning Curves of $L_{R-I}$ and normal $\beta$-type LA

In this Figure 3, the y-axis shows the optimal action probability and the x-axis shows the trial number $t = 0, 1, 2, \cdots$. This result shows that the $\beta$-type LA performs much superior than the $L_{R-I}$ reinforcement scheme, and the $L_{R-I}$ has wrong convergence when the value of parameter $a$ becomes 0.01 or more. Especially, the $L_{R-I}$ has about 15% of wrong converges in the parameter $a = 0.03$.

Then, Figure 4 shows the learning curves of the proposed $\beta$-type LA with 2-state Bayesian estimators in the parameters $\epsilon = 1.0E - 5$ and $\delta = 1.0$, the $L_{R-I}$ in the best parameter $a = 0.01$ and $SE_{R-I}$ (Stochastic Estimators Reward-Inaction Algorithm) by G.I.Papadimitriou, et al.[11],[12] which has the fastest property among the conventional LAs except for the $\beta$-type LA.

![Graph]

Fig. 4: Learning Curves of Proposed $\beta$-LA

In addition, Table.2 shows the comparison of memory usage about the $L_{R-I}$ scheme, $SE_{R-I}$ scheme, $\beta$-type LA with 2-state $BE$(proposed) and normal $\beta$-type LA. Compared with the memory usage of the $SE_{R-I}$ and

![Table]

Table 2: Comparison of Memory Usage

<table>
<thead>
<tr>
<th>LA type</th>
<th>memory usage (floating-point data)</th>
<th>learning parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>$L_{R-I}$</td>
<td>$r$</td>
<td>1</td>
</tr>
<tr>
<td>$SE_{R-I}$</td>
<td>$4 \times r$</td>
<td>2</td>
</tr>
<tr>
<td>$\beta$-type LA</td>
<td>$2 \times r$</td>
<td>3</td>
</tr>
<tr>
<td>(proposed, $m = 2$)</td>
<td>$m \times r$</td>
<td>2</td>
</tr>
</tbody>
</table>

![Table]

Table 1: $P$-model Environment ($r=10$)
proposed the $\beta$-type learning automaton, the usage of the proposed one is $1/2$ of the $SE_{R-1}$. Furthermore, Compared with the memory usage of the normal $\beta$-type and proposed $\beta$-type LA, the usage of the proposed one is $2/m$ of the normal $\beta$-type LA’s usage.

It is clearly that the proposed $\beta$-type LA’s memory usage is the smallest next to the $LR-I$ scheme among these LAs. Since resource requirements in such as WSNs (Wireless Sensor Networks) design, memory, energy minimization and performance optimization are very important issues. Therefore, the $\beta$-LA with 2-state Bayesian estimators is considered to be more ideal than other LAs.

6 Conclusion

In this research, the efficiency of $\beta$-LA with 2-state Bayesian estimators was shown through some simulation results. The $\beta$-type LA employs the minimum configured Bayesian estimators, and it shows appropriate property. Compared with the proposed $\beta$-type LA and the conventional LAs, it achieved well balanced performance under some simulation results.

However, we don’t consider about computational efficiency of the proposed system. Therefore, this results lead us to analyze our system from both points of view of practicality and theoretical explanation.

Acknowledgment

I would like to give sincere thanks to Prof. Kenichi Abe(deceased), who is a honorary professor of Tohoku University. Without his guidance, this paper would not have materialized.

References