Conditional Optimality of Learning Automata with 2-state Bayesian Estimators

Motoshi Hara†, Noriyo Kanayama†, Toru Watanabe†, Satoru Kato†, and Hiroyuki Kamaya‡
†National Institute of Technology, Matsue College,
‡National Institute of Technology, Hachinohe College
† 14-4 Nishi-ikuma, Matsue, Shimane 690-8518, Japan
E-mail: hara@matsue-ct.jp

Abstract

A novel learning automaton with 2 state Bayesian estimators, which is based on β-type learning automaton, is considered. The β-type learning automaton with multiple Bayesian estimators is our original work and the scheme of it has been proven to be conditionally optimal under some environments. Furthermore it was shown that β-type learning automaton exhibits superior learning behavior compared to conventional learning automata through several experimental results. However, the β-type one is inferior from the viewpoints of memory usage and computational efficiency. So, the β-type learning automaton which consists of 2-state Bayesian estimators as minimum resources has been proposed in our previous work. In this study, we show the sufficient condition for conditional optimality of the β-type LA which consists of 2-state Bayesian estimators.

1 Introduction

Recently, the field of the Artificial Intelligence(AI) shows remarkable progress through the wider spread of high-performance computing[1],[2]. Although the AI and Machine Learning(ML), especially its subfield of Deep Learning, are not quite the same thing, Machine Learning is considered as a current application of AI. The technology of ML is based on the idea that we should really just be able to give machines access to data, and let them learn for themselves. In recent years, the ML had many advances and important research papers may lead to breakthroughs in technology. For example, the deep Q-Networks(DQN), which is classified into the deep learning, combines Q-learning and a deep neural network and suffers from substantial overestimations in some computer games[1].

As is generally known, the Q-learning is a value-based reinforcement learning. The theory of stochastic learning automata forms the basis of the reinforcement learning, such as 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, has 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 conditionally 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 \( L_{R-I} \), \( L_{R-P} \), \( N_{R-P} \), \( H_{L_{R-I}} \), \( S_{E_{R-I}} \) 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 node are limited in the wireless sensor networks(WSNs), reducing memory usage and performance optimization are very important issues[8],[9].

For the above reasons, we have proposed the β-type LA with minimum resources, 2-state Bayesian estimators[10]. In this study, we show the sufficient condition for conditional optimality of the β-type LA with 2-state Bayesian estimators.

2 Learning Automaton

Learning automaton(LA) operating in an unknown random environment has 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 environmental response derived from the random environment by using reinforcement scheme.

A model of random environment and learning automaton is shown as Fig.1. In this chapter, we describe briefly the LA as follows.

Fig. 1: Learning Automaton-Environment Model
(1) 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 $\Omega = \{\alpha_1, \alpha_2, \cdots, \alpha_r\}$ and a response set $X = \{x_1, x_2, \cdots, x_n\}$. Here, we assume that $X$ is a set of distinct real numbers which indicates the degree of failure, unfavorable response or penalty. Furthermore, $f = \{f_1, f_2, \cdots, f_r\}$ is a set of unknown probability distributions.

The action $\alpha(t)$ belongs to the set $\Omega$ at discrete time $t = 0, 1, 2, \cdots$. The response $x(t)$ belongs to $X$ and obeys unknown probability distribution $f_i$ on $X$, which is corresponding to an action $\alpha_i \in \Omega$. Where, $f_i = (f_{i1}, f_{i2}, \cdots, f_{in})$, $(n = 1, 2, \cdots, r)$, which satisfies the following conditions,

$$0 \leq f_{ij} = Pr[x(t) = x_j|\alpha(t) = \alpha_i] \leq 1$$

$$\sum_{j=1}^{n} f_{ij} = 1, \forall i.$$ 

In the case that $n = 2$, $X = \{0, 1\}$ and $f_i = \{(1 - c_i), c_i\}$, the environment is called $P$-model\(^1\). And in the case that $2 < n < \infty$, $X = \{x_1, x_2, \cdots, x_n\} (-\infty < x_1 < \cdots < x_n < \infty)$ and $f_i = (f_{i1}, f_{i2}, \cdots, f_{in})$ $(n = 1, 2, \cdots, r)$, the environment is called $Q$-model\(^2\).

(2) Learning Automaton

The learning automaton can be represented by a 6-tuple $\langle X, \Omega', a, \pi(t), T, G \rangle$. Where, $X$ is a response set, $a$ is a action set, $\Omega' = \{\omega'_1, \omega'_2, \cdots, \omega'_r\}$ is a set of internal states of learning automaton. $\pi(t)$ is a state probability vector, which is defined by

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

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

$$\pi(t + 1) = T(\pi(t), \alpha(t), x(t)). \quad (2)$$

Then, $G$ is a output function, which can be ether deterministic or stochastic,

$$\alpha(t) = G[\pi(t)]. \quad (3)$$

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

$$\phi_i(t) = \sum_{j=1}^{s} \pi_j(t)g_{ij}. \quad (4)$$

Where, $g_j$ is $r$-dimension probability vector

$$g_j = (g_{j1}, g_{j2}, \cdots, g_{jr}) \quad (j = 1, 2, \cdots, s). \quad (5)$$

Furthermore $\phi_i(t)$ means

$$\phi_i(t) = Pr[\alpha(t) = \alpha_i] \quad (6)$$

and satisfies $\sum_{i=1}^{r} \phi_i(t) = 1$. For example, the LA chooses an action $\alpha(t) = \alpha_i \in \Omega$ with probability $g_{ij}$ at internal state $\omega'(t) = \omega'_j \in \Omega'.

(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.

$$\pi_j(0) = \frac{1}{s} \quad (j = 1, 2, \cdots, s)$$

**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 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 an evaluation value of the action $\alpha(t)$. Then, the learning automaton updates from $\pi(t)$ to $\pi(t + 1)$ by using the reinforcement scheme $T$.

**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

$$\phi_i(0) = \frac{1}{r} \quad (i = 1, 2, \cdots, r). \quad (7)$$

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

$$c_i = E[x(t)|\alpha(t) = \alpha_i] = \sum_{j=1}^{n} x_jf_{ij}. \quad (8)$$

The value $c_i$ is the evaluation of the action $\alpha_i$ and it is assumed that a unique minimum of $c_i(i = 1, 2, \cdots, r)$ exists. The ultimate goal of the learning automaton is to find the action that produces the lowest expected

---

\(^1\)Where, the $c_i$ is called “failure probability” under the action $\alpha(t) = \alpha_i$ in the $P$-model environment.

\(^2\)In the case that $X = \{x, \tau\} (-\infty < x < \tau < \infty)$, $f = \{f_1(x), f_2(x), \cdots, f_r(x)\}$ is a set of unknown probability densities, the environment is called $S$-model.
A learning automaton is optimal if:
\[
\lim_{t \to \infty} \sum_{\alpha_i \in \Omega_{opt}} \phi_i(t) = 1 \quad a.s.
\]
(10)

Definition 2.2. A learning automaton is \(\varepsilon\)-optimal, if:
\[
\lim_{t \to \infty} E[\sum_{\alpha_i \in \Omega_{opt}} \phi_i(t)] \geq 1 - \varepsilon
\]
(11)

In the equation (11), \(\varepsilon\) is arbitrarily small positive number.

3 \(\beta\)-Type Learning Automaton with 2-state Bayesian Estimators

A \(\beta\)-type learning automaton consists of \(r\) Bayesian estimators (BE) (See Fig.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 4-tuple \(X, \Omega, \alpha_i(t), T\), where \(\Omega = \{\omega_1, \omega_2\}\) is the set of its states; \(\alpha_i(t) = (\alpha_{i1}(t), \alpha_{i2}(t))\) is the state probability vector at time \(t\); \(T\) is the updating algorithm of \(BE\).

The state probability vector satisfies the following conditions
\[
0 \leq \lambda_{ij}(t) \leq 1,
\]
\[
2 \sum_{j=1}^{2} \lambda_{ij}(t) = 1, \forall i, j.
\]
(12)

Then, the state probability vector \(\pi(t)\) contains \(2^r\) components, and each component is defined by
\[
\pi_k(t) = \lambda_{1k_1}(t) \cdot \lambda_{2k_2}(t) \cdot \cdots \lambda_{rk_r}(t) \quad (k = 0, 1, 2, \ldots, 2^r).
\]
(13)

For each \(BE_i\), we assign a real number \(\mu_j\) to each state \(\omega_j\). These real numbers \(\mu_j(j = 1, 2)\) are taken to satisfy the following conditions:
\[
x < \mu_1 < \mu_2 < x,
\]
(14)
\[
x = \min_i(x_i), x = \max_i(x_i).
\]
(15)

The interaction between \(\beta\)-type learning automaton and Q-model stationary environment is explained as follows.

Step 0: \([x, \overline{x}]\) of \(BE\) is set arbitrary, then 2-parameters \(\varepsilon\) and \(\delta\) are introduced. Where, \(0 < \varepsilon < 1, 0 < \delta < (\overline{x} - x)\). Furthermore, every state probability vectors \(\lambda_i(0)(i = 0, 1, 2, \ldots, r)\) are initialized as \(\lambda_{i1}(0) = \lambda_{i2}(0) = 1/2\).

Step 1: At time \(t\), each \(BE_i(i = 1, 2, \ldots, r)\) chooses randomly its state \(\omega_{k_i} \in \Omega\) according to its state probability vector \(\lambda_i(t)\), and generates the output \(\mu_i(t) = \mu_{k_i}\). Then, the \(\beta\)-type learning automaton chooses an action \(\alpha(t) = \alpha_i \in \alpha\) corresponding to the \(BE_i\) which generates the lowest output and determines \(\mu(t) = \mu_i(t)\) as the output of the \(\beta\)-type learning automaton.

Step 2: After Step 1, the \(\beta\)-type LA subsequently receives the response \(x(t)\) corresponding to \(\alpha(t)\) 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
\[
\lambda_i(t + 1) = T^\prime(\lambda_i(t), x(t)).
\]
(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 updating algorithm \(T^\prime\) is described as follows.
\[
\lambda_i(t + 1) = c\lambda_i(t)(q_i^k)^\gamma
\]
(17)

Where, the finite model \(Q_i^k = (q_i^k, q_i^\overline{k})\) \((i = 0, 1, 2, \ldots, r)\) is denoted by
\[
q_i^k = \mu_j^{x(t)}(1 - \mu_j)^{1-x(t)} \quad (j = 1, 2)
\]
(18)
and c is the normalizing constant; γ is the parameter which determines the speed of convergence, $x'(t)$ is the normalized response from the environment which lies in the interval $[0,1]$ and is defined by

$$x'(t) = N(x(t)) = \frac{x(t) - x}{x - x}. \quad (19)$$

Particularly, when γ is equal to 1, the updating algorithm is the same form as Bayesian learning.

Theorem 4.1. Under the updating algorithm $(17)$ and $(19)$, if the random sequence $\{x(t)\}$ is equally and independently distributed with the probability distribution $(18)$, then the state probability vector $\lambda_i(t)$ asymptotically carried on the set $M_i$. That is

$$\lim_{t \to \infty} \sum_{k \in M_i} \lambda_{ik}(t) = 1 \quad a.s. \quad (20)$$

**Proof of Theorem 4.1.**

If $|M_i| = |C(P_i)| = 2$ then $\lambda_{ik}(t)(k \in M_i)$ takes the value from 0 to 1. Furthermore, the state probability vectors $\lambda_i(t)$ ($i = 1, 2, \ldots, r$) carried on $M_1 \times M_2 \times \cdots \times M_r$, which is the direct product of asymptotic carriers, almost surely at $t \to \infty$.

**Theorem 4.2.** If there exists a unique optimal action $a_{i_0}$ (that is $\epsilon_i < \epsilon_i$ ($i \neq i_0$)) and

$$N(c_{i_0}) < \hat{\mu} < N(c_i) \quad (i \neq i_0) \quad (25)$$

is satisfied, the β-type LA with 2-state Bayesian estimators becomes optimal as the formula

$$\lim_{t \to \infty} \sum_{a_i \in \Omega_{2\beta}} \phi_i(t) = 1 \quad a.s. \quad (26)$$

Where, $\hat{\mu}$ is defined by

$$\hat{\mu} = \frac{\log(\frac{1-\mu_1}{\mu_2})}{\log(\frac{1-\mu_{\alpha}}{\mu_{\beta}}) - \log(\frac{\mu_1}{\mu_2})}. \quad (27)$$

**Definition 4.3.** Under $k_i \in M_i$, $\forall i$, let Likelihood ratios $L_{ik}(t)$ be

$$L_{ik}(t) = \frac{L_{ik}(t+1)}{1 + \sum_{k' \neq k} L_{ik'}(t+1)}, \quad k \notin M_i. \quad (28)$$

Then, each $\lambda_i(t)$ is rewritten to

$$\lambda_{ik}(t) = \frac{\lambda_{ik}(t)}{1 + \sum_{k' \neq k} L_{ik'}(t-1)}, \quad k \notin k_i. \quad (29)$$

If $\lim_{t \to \infty} L_{ik}(t) = 0$ for each $k_i \in M_i$ then the conditional optimality is proved by $(29)$. However each action $a(t)$ of LA depends on its action probability vector $\phi(t)$, therefore it must be proved by another approach.

**Lemma 4.1.** Let $T_i(t)$ be the number of times the action $a_i \in A$ has been chosen up to time $t$, we have

$$\lim_{t \to \infty} T_i(t) = \infty \quad a.s. \quad (30)$$

**[Proof of Lemma 4.1]**

For all $i$, we consider

$$\varphi_{ik} = \sum_{j=1}^{n} p_{ij} \log_{\beta_j} k, \quad k \notin M_i. \quad (31)$$

Where, $\gamma$ is the parameter that appears in the equation $(17)$. Here, $\beta_{ik} = (q^k_j / q^k_{i})$. From this definition, $\varphi_{ik}(1) = 1$ and the differential $\varphi_{ik}'(1)$ at $u = 1$ is

$$\varphi_{ik}'(1) = -\gamma [D(P_i, Q^k) - D(P_i, Q^{k_i})]. \quad (32)$$
Since $k \notin M_i$ and $\gamma > 0$ are established, $\varphi_i'k(1)$ is negative. Therefore, if the value $u_0$ is greater than 1 and sufficiently close to 1,
\[ \varphi_i'k(u_0) < 1 \quad \tag{33} \]
is established for all $k \notin M_i$.

We assume that $\mathcal{F}_t$ is $\sigma$-algebra on \{ $\alpha(t), x(t)$ \}, $t = 0, 1, 2, \cdots, T$, then
\[
\begin{align*}
E[u_0^\log L_i(t) | \mathcal{F}_{t-1}] &= u_0^\log L_i(t-1) \{(1 - \phi(t)) + \varphi_i'k(u_0) \phi_i(t)\} \\
&\leq u_0^\log L_i(t-1). \quad \tag{34}
\end{align*}
\]

So, $u_0^\log L_i(t)$ is a non-negative super martingale.

Furthermore, from the definition of $\lambda_i(t)$, it is easily to derive
\[ \phi_i(t) \geq \frac{1}{r} \cdot \frac{\varepsilon}{1 + (r - 1)\varepsilon}, \quad k_i = 2 \]
\[ \phi_i(t) \geq \frac{1}{r} \cdot \frac{1}{1 + (r - 1)\varepsilon}, \quad k_i = 1 \quad \tag{40} \]

That is, the $\phi_i(t)$ remains at a certain value at $t \to \infty$.

Therefore, the probability of (37) is 0, then Lemma 4.1 is established.

[Proof of Theorem 4.1]

For $u > 0$, define $\Psi_i'k(t)$ by
\[
\begin{align*}
\Psi_i'k(t) &= \frac{u_0^\log L_i(t)}{\varphi_i'k(u)T_i(t)} \\
&= \left[ \frac{u_0^\log L_i(t)/T_i(t)}{\varphi_i'k(u)} \right]^{T_i(t)}. \quad \tag{41}
\end{align*}
\]

Where $E[\Psi_i'k(t) | \mathcal{F}_1] = \Psi_i'k(t - 1)$ can be easily derived for $k$. That is, $\Psi_i'k(t)$ is a non-negative super martingale and $E[\Psi_i'k(t)] = 1$. So, $\Psi_i'k(t)$ must converges to non-negative value at $t \to \infty$. For arbitrarily small positive number $\varepsilon$, we set
\[ s = \varphi_i'k(1) + \varepsilon \quad \tag{42} \]
and $g(u) = u^\varepsilon / \varphi_i'k(u)$, then we get $g(1) = 1$ and $g'(1) = \varepsilon > 0$. Therefore if
\[ \frac{\log L_i(t)}{T_i(t)} \geq s \quad \tag{43} \]
then
\[ u_0^\log L_i(t)/T_i(t) \geq g(u_0) > 1 \quad \tag{44} \]
is derived. If (43) is established for all $t$, $\Psi_i'k(t)$ includes a partial sequence diverging to infinity from (43) and Lemma 4.1. It is contradiction to the fact that $\Psi_i'k(t)$ converges. So
\[ \lim_{t \to \infty} \sup_{t \to \infty} \frac{\log L_i(t)}{T_i(t)} \leq s = \varphi_i'k(1) + \varepsilon. \quad \tag{45} \]

Similarly, we can arbitrarily determine $u_0 < 1$ and derive
\[ \lim_{t \to \infty} \inf_{t \to \infty} \frac{\log L_i(t)}{T_i(t)} \geq s = \varphi_i'k(1) - \varepsilon. \quad \tag{46} \]

Because the $\varepsilon$ is arbitrarily determined number, we derive from (45) and (46) that
\[ \lim_{t \to \infty} \log L_i(t) \geq \varphi_i'k(1) = -\gamma[D(P, Q^k) - D(P, Q^{k_1})]. \quad \tag{47} \]

Due to $\varphi_i'k(1)$ is negative for $k \notin M_i$, $L_i(t)$ ($k \notin M_i$) must converges to 0 from (47) and Lemma 4.1. Thus the state probability vector $\lambda_i(t)$ asymptotically carried on the set $M_i$ almost surely.
5 Discussion

In this result, the proposed $\beta$-type LA with 2 state Bayesian estimators searches for the appropriate interval $X = \{0, 1\}$ which satisfies the condition of (25). If the parameter $\delta$ isn’t appropriate, the search period will be longer in the $\beta$-type LA’s learning process, otherwise, the $\beta$-type LA can’t satisfy the condition of (25) by using (20) and (21). In our proposed algorithm, the Bayesian estimators focus on only the optimal and second optimal actions and ignore the other actions. Therefore, it is seemed that they achieved to implement “Select and Concentration Strategy”.

6 Conclusion

In this paper, we shown the sufficient condition for conditional optimality of $\beta$-LA with 2-state Bayesian estimators. The $\beta$-type LA employs the minimum configured Bayesian estimators, and shows appropriate property. Compared with the proposed $\beta$-type LA and the conventional LAs, the $\beta$-type one achieve well balanced performance through some simulation results.[10]

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

Acknowledgment

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

References

[1] Y.Gendio : “Learning Deep Architectures of AI”, Foundations and Trends® in Machine Learning, Now Publishers(2009).
[2] W.Liu,Z.Wang, et.al. : “A survey of deep neural network architectures and their applications”, Nurocomputing, Vol.234, pp.11-26(2017).
[3] C.Watkins, P.Dayan: “Q-Learning”, Machine Learning, Vol.8, pp.279-292(1992).
[4] M.L.Tsetlin: “On the Behavior of Stochastic Automata with a Variable Structure”, Automation and Remote Control, Vol.24, pp.1345-1354(1962).
[5] K.S.Narendra, M.A.L.Thathachar: “ Learning Automata -A Survey”, IEEE trans. SMC, Vol.SMC-4, No.4, pp.323-334(1974).
[6] K.S.Narendra,M.A.L.Thathachar: “Learning Automata: An Introduction”, Prentice Hall(1989).
[7] M.Hara, K.Abe: “On a Conditionally Optimal Learning Automaton”, Trans. the Society of Instrument and Control Engineers, Vol.28, No.6, pp.742-749(1992)[in Japanese].
[8] P.Harrop, R.Das: ”Wireless Sensor Networks”, pp.2011-2021, IDTechEx(2011).
[9] M.Hara, W.Aoto, A.Iwata, N.Kanayama, T.Watanabe, H.Kamaya : ”Geographic Routing for 3-D Wireless Sensor Networks with Stochastic Learning Automata”, Proceedings of The 48th ISCIE International Symposium on Stochastic Systems Theory and Its Applications , pp.153-159(2017).
[10] M.Hara, W.Aoto, N.Kanayama, T.Watanabe, H.Kamaya: ” Learning Automata with 2-State Bayesian Estimators ”, Proceedings of The 49th ISCIE International Symposium on Stochastic Systems Theory and Its Applications , pp.40-45(2018).
[11] K.Abe: ” On Conditional Optimality of a Class of Learning Automata in Random Environments”, Information Sciences, Vol.31, No.3, pp.243-263(1983).