Neurocomputing 137 (2014) 150–156

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

A general method for P-model FSSA learning in triple level environment

Wen Jiang, Sheng-Hong Li *

Department of Electronic Engineering, Shanghai Jiao Tong University, 800 Dongchuan Road, Min Hang, Shanghai 200240, PR China

ARTICLE INFO

Article history:
Received 29 November 2012
Received in revised form
27 February 2013
Accepted 25 March 2013
Available online 16 February 2014

Keywords:
P-model
Q-model
Learning Automata
ε-Optimal
Triple level environment
Compatible environment

ABSTRACT

Within the context of Learning Automata (LA), great efforts have been made to deliver P-model automaton which accepts two values from the stochastic environment, i.e., reward or penalty. However, in many real life problems, the stochastic environments have more than two feedback choices. This paper aims at proposing a general approach for extending P-model Fixed Structure Stochastic Automata (FSSA) to learn the optimal action in a triple level environment which provides three choices of feedback, i.e., reward, small scale penalty and large scale penalty while keeping the existing P-model FSSA algorithms intact in order not to impose much impact on existing systems. For performance evaluation, we introduce compatible stochastic environment, incompatible environment and boundary environment to assess the performance of the proposed method. Simulation results demonstrate that the proposed method is expedient in dealing with triple level environment.

1. Introduction

Learning Automata (LA) is one of the most powerful tools in the field of Reinforcement Learning which itself is a branch of Artificial Intelligence. The goal of Learning Automata is to learn the one sole optimal action among a list of candidate ones via interacting with stochastic environment in terms of a sequence of repetitive feedback cycles. As shown in Fig. 1, during a cycle, Learning Automata selects an action $\alpha(n)$ at the time instant n, to execute in the environment which provides a feedback $\beta(n)$ (e.g., reward or penalty) to this action. Learning Automata uses this feedback response and the selected action to update its internal state which determines the next action to pick up. This cycle repeats until the Learning Automata learns the optimal action. Learning Automata has a myriad of applications in network optimization problem [24–29], capacity assignment problem [17,22], modeling Tutorial-like System [9], shortest path problem [18], stochastic point location problem [10], training Hidden Markov Models [20], knapsack problem [14], cooperative spectrum sensing [16], guard channel algorithm [23], constraint satisfaction problem [30] and so on [13,15,19,21].

In general, Learning Automata can be classified into two types: Finite Action set Learning Automata (FALA) [1–8] and Continuous Action set Learning Automata (CALA) [12]. The former is characterized by selecting one action from a countable action set, while the latter chooses an action from a predefined interval. Among these two types, FALA is more widely studied and is recognized as the main form of Learning Automata. FALA can be further classified into Fixed Structure Stochastic Automata (FSSA) [1,3–5,11] and Variable Structure Stochastic Automata (VSSA) [2,7,8]. FSSA, whose transition and output functions are time invariant, is easier to implement and requires less computation per time step compared with VSSA. Some well-known FSSA algorithms include Krylov, Krinsky, IJA, TFSLA and STARs [1,3–5,11]. On the other hand, VSSA is easier to adapt to stochastic environment and has been widely studied [2,7,8].

Most of these Learning Automata algorithms belong to P-model Learning Automata which accepts two types of feedbacks from environment, namely, reward or penalty. However, a great many of real life problems cannot be solved within this framework. For example, the feedback of the environment may include not only reward and penalty but also the severity of penalty.

In order to solve this kind of problems, Jamalian et al. [11] introduced the concept of the triple level stochastic environment which has three kinds of responses (i.e., reward, small scale penalty and large scale penalty), and then proposed a new algorithm, TILA, to interact with the triple level environment. However, as far as we known, there is no other algorithms designed for this triple level environment. In this paper, we will go one step further and focus on proposing a general paradigm to extend the existing P-model FSSAs to deal with triple level environment and, hopefully, shed some light on learning under

* Corresponding author.
E-mail addresses: wenjiang@sjtu.edu.cn (W. Jiang), shli@sjtu.edu.cn (S.-H. Li).

http://dx.doi.org/10.1016/j.neucom.2013.03.071
0925-2312

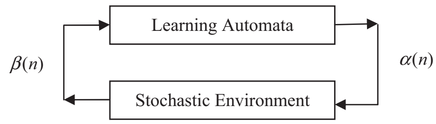

Fig. 1. Learning Automata interaction with stochastic environment.

the triple level environment. Note that this triple level Learning Automata belongs to Q-model Learning Automata which allows finite kinds of responses from environment.

Contributions of this paper are listed below.

(1) We propose a general method for extending the current P-model FSSAs to deal with triple level environment;
(2) As far as we know, this is the first paper that introduces the concept of compatible triple level stochastic environment, incompatible triple level environment, and boundary environment to evaluate the performance of Learning Automata;
(3) Simulation results demonstrate that our proposed method leads to experimentally expedient triple level automaton.

The rest of this paper is organized as follows. Section 2 introduces basic concepts of FSSA. In Section 3, a general method is proposed to extend existing P-model FSSA to deal with triple level environment. Extensive simulations have been conducted to demonstrate the effectiveness of the proposed method in Section 4. The last Section concludes this paper.

2. Fundamentals

In this section, we introduce the basic concepts of the FSSA which include the definition of automaton, the stochastic environment with which the automaton interacts, and the norms of automaton's learning behavior.

2.1. Fixed Structure Stochastic Automata

Stochastic automata is considered as a finite state automata which can be represented by

$$\langle A, B, \Phi, T, G \rangle \tag{1}$$

where $A = \{\alpha_1, \alpha_2, \ldots, \alpha_r\}$ is the set of actions of stochastic automaton; $B = \{\beta_1, \beta_2, \ldots, \beta_m\}$ is the set of responses from stochastic environment. As mentioned before, the stochastic automaton is a Q-model automaton if $m > 2$. In this paper, we consider $B = \{0, 1, D\}$ where "1" means small scale penalty and "D" means large scale penalty. $\Phi = \{\phi_1, \phi_2, \ldots, \phi_s\}$ is the set of internal states of stochastic automaton; $T : \Phi \times B \rightarrow \Phi$ is the function that maps current state to the next state when the response from environment is received; $G : \Phi \rightarrow A$ is the output function of the automaton which maps the current state to one of the actions of the stochastic automaton.

2.2. Triple level environment

A triple level environment [11] can be mathematically represented by

$$\langle A, B, C, E \rangle \tag{2}$$

where $A = \{\alpha_1, \alpha_2, \ldots, \alpha_r\}$ is the input set of the triple level environment and is the set of actions of stochastic automaton; $B = \{0, 1, D\}$ is the set of outputs of the triple level environment; if $B = \{0,1\}$, the stochastic environment takes one of these two types of response to Learning Automata and thus named as double level environment. $C = \{c_1, c_2, \ldots, c_r\}$ is the set of penalty probabilities for each action $\alpha_i, i = 1, 2, \ldots, r$

$$c_i = pr\{\beta(n) \neq 0 | \alpha(n) = \alpha_i\}, \quad 0 < c_i. \tag{3}$$

$E = \{e_1, e_2, \ldots, e_r\}$ is the set of probabilities for environment which feedbacks a small scale penalty under the condition that no reward response is received

$$e_i = pr\{\beta(n) = 1 | \alpha(n) = \alpha_i, \quad \beta(n) \neq 0\}. \tag{4}$$

An environment is called stationary environment if both c_i and e_i are time invariant. This paper will concentrate on designing a general method for updating P-model FSSA to interact in this stationary triple level environment.

2.3. Norms of behaviors

One of the widely used performance evaluation metrics for automaton is the average penalty received by automaton at time instant n:

$$M(n) = E[\beta(n) \neq 0]. \tag{5}$$

For P-model Learning Automata, the average penalty at time instant n, denoted as $M^p(n)$, can be obtained by

$$\begin{aligned} M^p(n) &= E[\beta(n) \neq 0] = \Pr[\beta(n) = 1] \\ &= \sum_{i=1}^{r} \Pr[\beta(n) = 1, \alpha(n)] = \sum_{i=1}^{r} pr[\alpha(n) = \alpha_i] \cdot c_i. \end{aligned} \tag{6}$$

Then the average penalty for a pure chance P-model Learning Automaton which selects each action equally likely is

$$M_0^p = E[\beta(n) \neq 0] = \sum_{i=1}^{r} \frac{1}{r} \cdot c_i. \tag{7}$$

However, as indicated in [11], average penalty, denoted as $M^{tri}(n)$, received by triple level Learning Automata at time instant n is

$$M^{tri}(n) = E[\beta(n) \neq 0] = \sum_{i=1}^{r} pr[\alpha(n) = \alpha_i] \cdot [c_i e_i + c_i(1-e_i)D]. \tag{8}$$

where $c_i e_i + c_i(1-e_i)D$ is the average penalty for α_i.

Then, the average penalty received by pure chance triple level Learning Automata is

$$M_0 = E[\beta(n) \neq 0] = \sum_{i=1}^{r} \frac{1}{r} \cdot [c_i e_i + \ c_i(1-e_i)D]. \tag{9}$$

As mentioned in [11], the following Lemma shows that the double level environment is a special case of the triple level environment.

Lemma 1. *The average penalty received by triple level Learning Automata is the same as that received by a P-model Learning Automata only if $D = 1$ or $e_i = 1$.*

The proof of this Lemma is presented in [11], and is summarized as follows.

Comparing Eqs. (6) and (8), it can be derived

$$\begin{aligned} M^{tri}(n) = M^p(n) &\Rightarrow c_i = c_i e_i + c_i(1-e_i)D \\ &\Rightarrow c_i(1-e_i) = c_i(1-e_i)D \\ &\Rightarrow c_i(1-e_i)(1-D) = 0. \end{aligned} \tag{10}$$

Then, it is easy to understand that P-model Learning Automata can be seen as a type of triple level Learning Automata since Eq. (9) holds only for $e_i = 1$ or $D = 1$ while $0 < c_i$ is defined by stochastic environment. That is to say, if $e_i = 1$ or $D = 1$, triple level Learning Automata converts to P-model one and hence the learning process can be achieved by employing P-model Learning Automata algorithm. Thus, in this paper we shall pay more attention to the case where $e_i \neq 1$.