Neurocomputing xx (xxxx) xxxx–xxxx

Contents lists available at ScienceDirect

Neurocomputing

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

A novel parallel framework for pursuit learning schemes

Hao Ge, Jianhua Li, Shenghong Li*, Wen Jiang, Yifan Wang

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

ARTICLE INFO

Keywords:
Learning automata
Parallel framework
Decentralized learning
Centralized fusion
Learning speed

ABSTRACT

Parallel operation of learning automata (LA), which is proposed by Thathachar and Arvind, is a promising mechanism that can reduce the computational burden without compromising accuracy. However, as far as we know, this parallel mechanism has not been widely used due to two reasons: one is the fact that the environment can response to multi-actions simultaneously are few, the other is the relatively slow speed of the learning process.

In this paper, a novel parallel framework is presented to reduce the number of required interactions between the incorporated pursuit LA and the environment by introducing decentralized learning and centralized fusion. The philosophy is to learn various aspects of the problem at hand by taking advantage of the diverse exploration of decentralized learning and summarize the common knowledge learned by centralized fusion. Simulations are conducted to verify the effectiveness of our framework and demonstrate its outperforming. The proposed framework is further applied to the stochastic point location problem and obtains an attractive performance.

1. Introduction

Learning Automaton (LA), an important research area of Artificial Intelligence (AI), is a self-adaptive machine that can learn the optimal action from a random environment. LA was first investigated by Tsetlin to model the behavior of biological learning systems [1,2] and by now the study of LA has reached a relatively high level of maturity. Various successful applications utilizing LA have been reported in areas such as community detection [3], cooperative spectrum sensing [4], clustered wireless ad-hoc networks [5], tutorial-like systems [6,7], on-line event pattern tracking [8] and multi-class classification problems [9]. One intriguing property that popularize the learning automata based approaches in engineering is that LA can learn the stochastic characteristics of the external environment it interacts with, and maximize the long term reward it obtains through interacting with the environment. When the environment in which they operate provides noisy and incomplete information, LA's performance is significantly better than other methods.

In theoretical field, networks of LA are committed to solve the problems that are difficult for single LA to handle. By the synthesis of complex learning structures from simple learning automaton, networks demonstrate some new features.

The study of networks of LA was pioneered by Thathachar [10]. Through his efforts, systems consisting of several learning automaton such as hierarchical structure, games and parallel operation are constructed.

1. Hierarchical structure looks at larger aggregations of LA so that more complex learning problems can be handled. Such a system has several levels, each of whom is comprised of several LA. The automaton in upper level selects an action, which activates the corresponding automaton in the lower level. This procedure repeats from the top to the bottom. Thus, the hierarchy system has a tree like structure. Only those LA that at the bottom level (corresponds to leaf nodes in the tree) can interact with environment directly. Some new features are demonstrated by this new structure. Poznyak and Najim [11] showed that the use of hierarchical structure LA (HSLA) accelerates the learning process. And stochastic point location (SPL) can be solved by using hierarchical learning automata [12]. Meanwhile, [13] demonstrates the hierarchical structure can cope with problems of general non-stationary multi-teacher environment (NME).
2. “Games of LA” are multi-automata system constructed to overcome the high dimensionality of the decision space. In the case where the objective function has N variables, then the point where a maximum is attained would be a vector of N components. Using a single LA whose action set corresponds to points in R^N is unreasonable because the number of actions would be unacceptably large. It would be better to use one automaton for learning one component of the maximum point. These N LA constitute a multi-automata system, where each automaton would be viewed as a player involved in a game. In such a game, multiple automata are able to control a

* Corresponding author.
E-mail addresses: sjtu_gehao@sjtu.edu.cn (H. Ge), lijh888@sjtu.edu.cn (J. Li), shli@sjtu.edu.cn (S. Li), wenjiang@sjtu.edu.cn (W. Jiang), wangyifan_1123@sjtu.edu.cn (Y. Wang).

http://dx.doi.org/10.1016/j.neucom.2016.09.082
Received 18 February 2016; Received in revised form 20 July 2016; Accepted 3 September 2016
Available online xxxx
0925-2312/

finite Markov chain with unknown transition probabilities and rewards [14]. The collective wisdom of these N LA are utilized to locate the optimum point in solution space R^N. In pattern recognition field, [15] shows an example that N LA constitute a common payoff game to learn the underlying separating hyperplane, and each LA corresponds to one dimension of the hyperplane.

3. Parallel operation of LA [10] is presented with the objective of improving the speed of convergence via utilizing the parallel nature of the environment. The learning process of single LA can be viewed as essentially sequential, at a time only one action is selected and one feedback is elicited from the environment. In some cases where the environment could response to multi-actions simultaneously, several actions could be sent collectively as an input and all feedback signals can be used collectively to update the action probability. The philosophy of this parallel mechanism is to trade space for time.

However, unfortunately, despite the parallel operation looks promising, there are few literatures that exist in the field of LA for parallel applications. The reason is two-fold: 1) The classic parallel framework does not intend to decrease the number of interactions with environment, but only to reduce computational burden. Computation power is no longer the bottleneck for most scenarios nowadays, but getting a response from the environment may be time-consuming or energy-consuming sometimes. So a framework that can reduce the required number of interactions is desired. 2) In practical applications, the environment could response to multi-actions simultaneously are few.

In this paper, a novel parallel framework is presented. The parallel operation is divided into two steps, decentralized learning and centralized fusion. Several numerical simulations are also carried out to verify the effectiveness of the proposed framework. The results demonstrate this new framework can outperform the classic one, and reduce the number of interactions with environment. The contributions of this work are highlighted as follows:

1. Classic parallel framework is presented with the objective of reducing computational burden. The total number of interactions required does not vary with parallel scale. While the goal of our framework is to learn in parallel more efficiently, i.e., requires fewer interactions with the environment.
2. Two novel mechanisms are employed in our framework. One is the decentralized learning, which can take advantage of diverse exploration to learn different aspects of the problem at hand. The other is centralized fusion which will summarize the learned information into common knowledge.
3. The ε-optimality of the proposed framework is derived and simulation results demonstrate its superiority to the classic one.

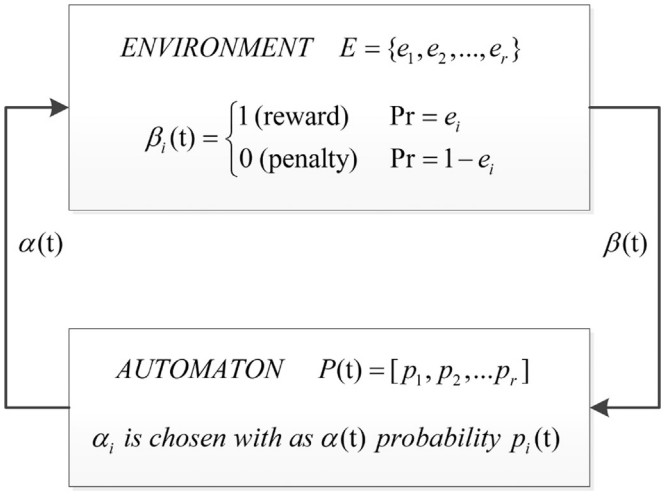

Fig. 1. Block diagram of a learning automaton.

2. Related works

2.1. Deterministic estimator based LA

In the history of single LA, various approaches have been proposed to speed up the learning process, among which discretization [16] and estimation [17] are two epoch-making concepts. The former is implemented by restricting the probability of choosing an action within a finite number of values in the interval (0, 1), and the latter are modules that gather history information to estimate the reward probability of each possible action, in order to update action probability vector purposefully.

Deterministic estimator based LA, such as DP_{ri} [18], DGPA [19] and the newly presented LELA [20], DGCPA [21], are the major family of LA. SE_{ri} [22], a very fast LA scheme, has an extra tunable parameter to control the randomness imposed to the deterministic estimates. Its training stage need to traverse a 2-dimensional parameter space. After training, this extra parameter carries extra information about the environment. It is not fair to compare it with deterministic estimator based LA. Hence we only take deterministic estimator based LA into consideration in this paper.

As pursuit schemes are the most fundamental one of deterministic estimator based LA, we take the classic DP_{ri} as an example to describe the main features of a learning automaton. A block diagram depicting the automaton – environment interaction is shown in Fig. 1.

Environment, the aggregate of external influences of learning process, can be depicted as $\{A, B, E\}$. Automaton, the learning module, is defined as $\{A, B, P, T, D\}$. Among them, A and B are interaction information exchanged between automaton and the environment.

$A = \{\alpha_1, \alpha_2, \ldots, \alpha_r\}$, a finite set of r actions. $\alpha(t) \in A$ is the output of the automaton and the input of the environment at time t.

B is a feedback set. $\beta(t) \in B$ denotes the reaction from the environment at time t. If B is a binary output set, e.g. $\{0, 1\}$, the environment is referred to as a P-model environment. All the schemes discussed within this paper are restricted to interacting with a P-model stationary environment.

$E = \{e_1, e_2, \ldots, e_r\}$ is the set of reward probabilities. The feedback in response to each action α_i is modeled as a Bernoulli distribution over B, that is $e_i = Prob[\beta(t) = 1 | \alpha(t) = \alpha_i]$. The challenge of the learning problem is that the reward probabilities are unknown to the automaton. The only information that can be utilized by the automaton is the stochastic reinforcement signal in response to each action choice made.

$P(t) = [p_1(t), p_2(t), \ldots, p_r(t)]$ is the action probability vector, where $p_i(t) = Prob[\alpha(t) = \alpha_i | P(t)]$, $i = 1, \ldots, r$.

$D(t) = [d_1(t), d_2(t), \ldots, d_r(t)]$ is the deterministic estimator vector. $d_i(t)$ is the current deterministic estimates of e_i and is calculated as formula (1)[1]. Where $Z_i(t)$ is the number of times action α_i was selected up to t, and $W_i(t)$ is the number of times action α_i was rewarded during the same period.

$$d_i(t) = \frac{W_i(t)}{Z_i(t)}, \ \forall\, i \in \{1, \ldots, r\} \tag{1}$$

T is the updating rule so that $P(t+1) = T(P(t), \cdot)$. During a cycle, LA chooses an action $\alpha(t)$ and then receives a stochastic response $\beta(t)$ from the environment.

According to pursuit scheme with reward-inaction philosophy, the updating rule T is: If $\beta(t) = 1$ then

$$p_j(t+1) = \max\{p_j(t) - \Delta, 0\}, \ \forall\, j \neq m \tag{2}$$

$$p_m(t+1) = 1 - \sum_{j \neq m} p_j(t+1) \tag{3}$$

[1] This kind of estimator is called Maximum Likelihood Estimator (MLE), there are also other kinds of estimators, such as Confidence Interval Estimator (CIE) proposed in [21].