Neurocomputing 188 (2016) 311–318

Contents lists available at ScienceDirect

Neurocomputing

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

A cooperative framework of learning automata and its application in tutorial-like system

Hao Ge [a], Yifan Wang [a], Shenghong Li [a,*], Chun Lung Philip Chen [b], Ying Guo [a]

[a] Department of Electronic Engineering, Shanghai Jiao Tong University, 800 Dongchuan Road, Shanghai 200240, China
[b] Faculty of Science and Technology, University of Macau Macau, China

ARTICLE INFO

Article history:
Received 23 January 2015
Received in revised form
9 April 2015
Accepted 16 April 2015
Available online 2 December 2015

Keywords:
Tutorial-like system
Learning automata (LA)
Cooperative method
Current State of Learning (SoL)

ABSTRACT

A novel cooperative framework of learning automata (LA) is presented in this paper. In a system, where different LA can be integrated via the proposed framework, current *State of Learning* (SoL) index is advocated to evaluate the learning status of individual LA. Based on that learning status, individual LA will adaptively choose an appropriate interaction strategy at cooperative learning phase. Theoretical analysis demonstrated that this index is able to preserve the ε-optimal feature of independent learning automata. Experimental simulations validated that this cooperative framework is effective in improving learning speed of a variety of LA embedded in this framework. Then, we also present an application of the proposed cooperative framework in tutorial-like systems. Compared with existing method, our cooperative framework based method outperforms in both speed and accuracy.

1. Introduction

Learning automaton (LA), a promising field of artificial intelligence, is a kind of self-adaptive finite state machine that interacts with a stochastic environment. LA is able to track the optimal action even provided with probabilistic wrong hints. Learning automata are very suitable for learning to control an agent by letting it interact with a well-defined environment. When the information is incomplete or the environment is noisy, LA is significantly superior to other methods. A wide range of learning automata applications are published in areas and some latest such as dynamic resource allocation [1], event patterns tracking [2], recommender systems [3], complex networks [4], function optimization [5], and so on [6–14].

While applications of LA are flourishing, the relatively slow rate of convergence become the bottleneck of LA applications been further investigated. On improving the convergence rate of LA, a plenty of techniques have been introduced: *estimator* [15], *discretization* [16], *generalization* [17] and *stochastic estimator* [18]. There are a number of different learning automata algorithms that incorporating these techniques: DPri [19], DGPA [17], SEri [18], LELA [20] and DGCPA [21] to name but a few. The maturity of single LA theory provides a solid basis for aforementioned applications.

Recent years, however, a number of ensemble methods, such as bagging, boosting, random forests, mixtures of experts, and swarm theory have been proposed in the field of machine learning, known as ensemble learning. Ensemble learning is a aggregation of base learners with the goal of improving accuracy and speed. Likewise, in some specific applications, a team of LA should be employed to solve one learning problem. These LA can be organized in a myriad of structures, such as parallel structure [22], hierarchical structure [23–26] and even networks of LA [27]. Frameworks that combine multiple LA together also show new features and potentials to speed up learning process. In [22], the author proposed a general procedure that suitable for parallelizing a large class of sequential learning algorithms on a shared memory system(demonstrated in Fig. 1). A variety of learning algorithms have shown speed improvement through parallelization. Hierarchical sturcture(demonstrated in Fig. 2) was first introduced in [23] to reduce the number of updatings to be made at each instant, which will lead to slow convergence on obsolete computers when the number of actions is large. Later it is further developed by incorporating estimator-based automaton in [25]. Thathachar show us the way how learning automata can constitute a feedforward network in his book [27], just like a nerual network. All of these structures are utilizing collaborative intelligence to solve sophisticated problems more effectively and more efficiently. This motivate us to explore the possibility for a prospective framework for ensemble of different LA algorithms. Therefore, a cooperative framework that combines multiple LA algorithms, which allows each learning member communicate with others to enhance its own performance, is proposed in this paper. The ultimate goal is to

[*] Corresponding author.
E-mail addresses: sjtu_gehao@sjtu.edu.cn (H. Ge), wangyifan_1123@sjtu.edu.cn (Y. Wang), shli@sjtu.edu.cn (S. Li), philipchen@umac.mo (C.L.P. Chen), guoying2014vip@sjtu.edu.cn (Y. Guo).

http://dx.doi.org/10.1016/j.neucom.2015.04.122
0925-2312/© 2015 Elsevier B.V. All rights reserved.

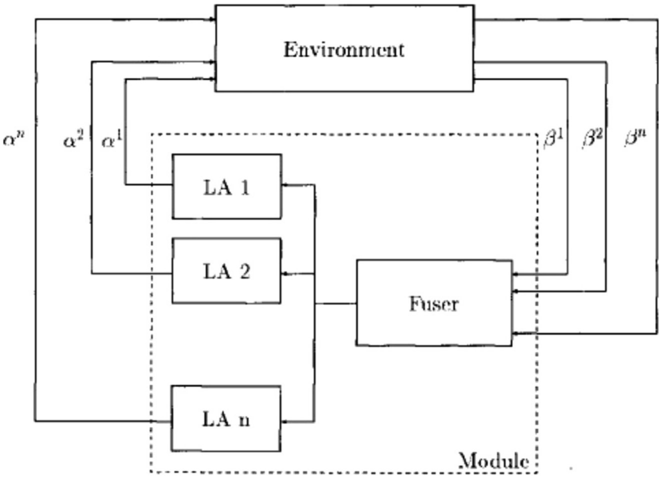


Fig. 1. module of learning automata [22].

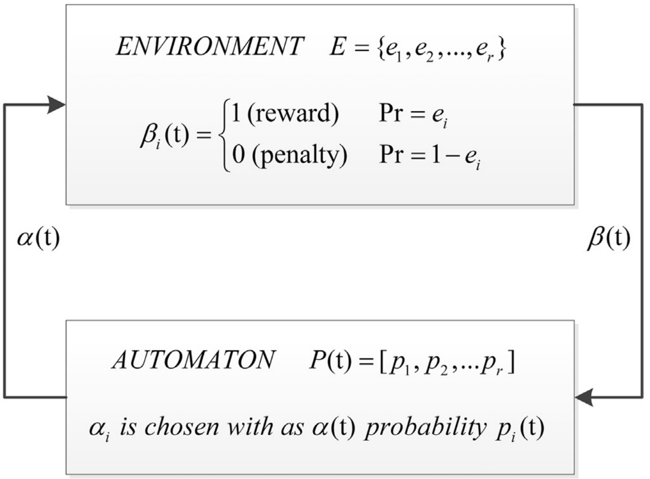

ENVIRONMENT $E = \{e_1, e_2, \ldots, e_r\}$
$\beta_i(t) = \begin{cases} 1 \text{ (reward)} & \Pr = e_i \\ 0 \text{ (penalty)} & \Pr = 1 - e_i \end{cases}$
$\alpha(t)$
$\beta(t)$
AUTOMATON $P(t) = [p_1, p_2, \ldots p_r]$
α_i is chosen with as $\alpha(t)$ probability $p_i(t)$

Fig. 3. Block diagram of *learning automaton and environment* interactions.

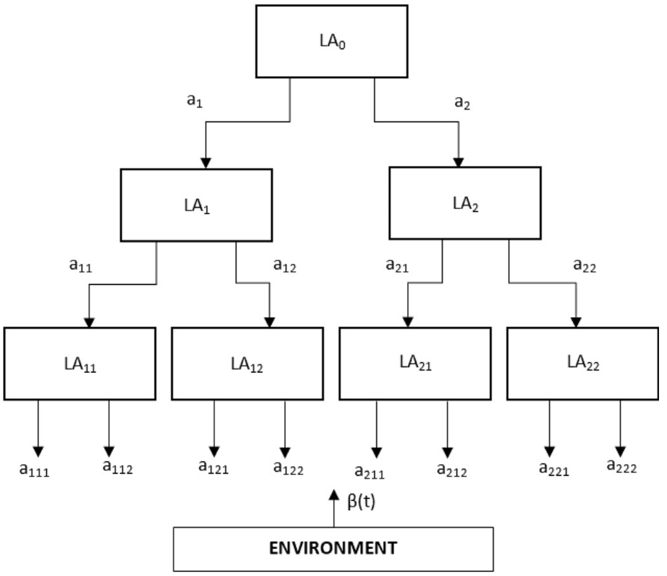


Fig. 2. A hierarchical learning automaton interacting with a stochastic environment [23].

accelerate learning speed without compromising accuracy by the cooperation of LA.

From another perspective, the combination of tutorial-like systems [28] and learning automata (LA) has been a new study direction in the recent decade. Researchers endeavor to simulate components in tutorial-like systems using appropriate learning models. What to teach (domain model [29,30]), who to teach (student model [31,28]) and how to teach (teacher model [32,33]) are the main concerns in tutorial-like systems. Oommen and Hashem have done some excellent pioneering works [31,28–30,32,33] in using LA to model different components of tutorial-like sysyem.Oommen and Hashem [28] first present the student model using LA, then a student–classroom interaction model [34] is built upon it. LA are used as student simulators in [28] which attempt to model the behavior of real-life student in tutorial system. The long-term goal of the literature is that if the tutorial-like system can understand how the student perceives and processes knowledge, it will be able to customize the way by which it communicates the knowledge to the student to attain an optimal teaching strategy. In [34], Oommen and Hashem further developed the paradigm of tutorial-like system, by allowing students to be a member of a classroom of students, which implies that the student of the classroom can not only learn from the teacher(s) but also learn from any of his fellow students. Experimental simulation shows that the new philosophy can improve the learning speed of a weak student up to 73%. However, this model enhances speed at the cost of accuracy. In this paper, we use the proposed cooperative LA model to simulate the interaction model among students.

The contributions of this paper are listed as follows:

1. This paper proposed a new cooperative framework of learning automata. A system consists of different LA can benefit from this framework.
2. Current State of Learning (SoL) index is presented to evaluate the learning performance of individual LA. Theoritic analysis demonstrated that this index is able to preserve the ε-optimal feature of learning automata.
3. Experimental simulations verified that the proposed framework is effective in improving learning speed of a variety of LA.
4. Modeling student–classroom interaction in tutorial-like system is presented as a successful application of the cooperative framework.

The rest of this paper is organized as follows. Section 2 introduces basic concepts of LA, as well as background of tutorial-like systems. In Section 3, a cooperative framework of learning automata is proposed, analyzed and experimentaly verified. Application in tutorial-like system has been conducted to further demonstrate the significance of the proposed cooperative framework in Section 4. The last section concludes this paper.

2. Fundamentals

In this section, we would like to introduce the basic concepts of the learning automata firstly, including the definition of automaton and the stochastic environment with which the automaton interacts, and a brief introduction of tutorial-like systems is given then.

2.1. Learning automaton (LA)

Learning automaton is an autonomous system that interacts with the environment and adaptively adjusts it behavior [16] towards being maximal rewarded. A block diagram illustrating how automaton interacts with environment is given in Fig. 3.

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

A mathematical definition of each symbol are listed as below:

- $A = \alpha_1, \alpha_2, \ldots, \alpha_r$, a finite set of r actions. $\alpha(t) \in A$ is the output of the automaton (input of the environment) at the time t.
- B is the set of environment feedbacks. $\beta(t) \in B$ represents the reaction from the environment, indicating how much the environment favors action $\alpha(t)$. If B is a binary output set, for example {0,1}, where 0 stands for a penalty and 1 for a reward, the environment is called a P-model environment. All the algorithms discussed throught this paper are restricted to a P-model stationary environment.
- $E = [e_1, e_2 ,...., e_r]$ is the vector of reward probabilities. Each action is associated with α_i a probability distribution over B, that is $e_i = Prob[\beta(t) = 1 \mid \alpha(t) = \alpha_i]$. The chanllenge of the learning problem is that this reward probabilities E are unknown to the automaton, the only information the automaton can acquire is the stochastic reinforcement signal (feedback) in response to every action choice made.
- $P(t) = [p_1(t), p_2(t) ,...., p_r(t)]$ is the action probability vector, where $p_i(t) = Prob[\alpha(t) = \alpha_i], i = 1, \ldots, r$. Vector $P(t)$ time instant t,
- $D(t) = [d_1(t), d_2(t) ,...., d_r(t)]$ is the deterministic estimator vector. $d_i(t)$ is the current deterministic estimates of e_i . Maximum likelihood estimate (MLE) is a widely used method [35] for estimating e_i, which is calculated as the following formula (1).

$$d_i(t) = \frac{W_i(t)}{Z_i(t)}, \forall i \in \{1, 2, ..., r\} \tag{1}$$

where $Z_i(t)$ is the number of times action α_i was selected up to time instant t, and $W_i(t)$ is the number of times action α_i was rewarded during the same period.
- T is the learning algorithm to update P, that is $P(t+1) = T(P(t), \bullet)$. T can be categoried as linear or non-linear, continous or discretized according to different ways of updating P.

In the history of single LA, various approaches have been proposed to speed up the learning process. *Discretization* [19] and *Estimation* [36] are two epoch-making concepts. Discretization is implemented by restricting the probability of choosing an action within a finite number of values in the interval (0,1). Estimators, in the context of learning automata, are the techniques that store history information and estimate the reward probability of each possible action, in order to leverage those estimates to guide the action probability updating.

MLE based LA has a number of members, such as DPri [19], DGPA [17], SEri [18] and the newly presented LELA [20], DGCPA [21]. Among them, SEri and DGCPA are the two fastest LA algorithms. SEri, which takes the MLE as deterministic estimate value and impose a random perturbation with zero mean to the deterministic estimates. SEri uses a pertubation parameter to control the stochasitc estimator. Usually, a grid search is performed to tune the parameter for learning in a particular environment. It is worth noting that, once the grid search is done, the tuned best parameter can be regard as some degree of prior information about the environment that obtained during the tuning process. However, DGCPA has a different philosophy. The upper bound of a 99% confidence interval, rather than MLE, is used as estimate of each action's reward probability. DGCPA reaches a comparative performance without the process of tuning an extra parameter.

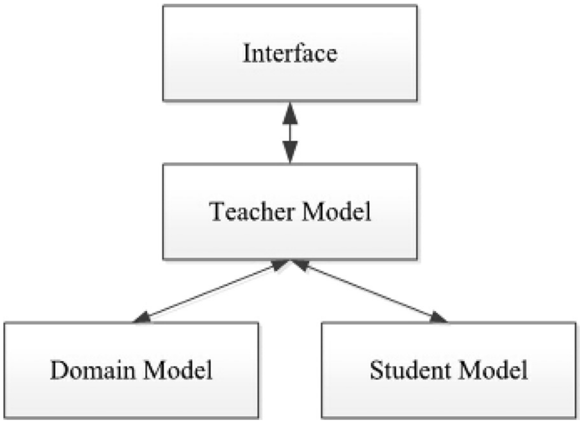


Fig. 4. Structure of Intelligent Tutorial Systems.

2.2. Tutorial-like system

A tutorial system is a program that can provcide specific materials that help you to learn a specific domain of knowledge. It is comprised of three parts: the teaching machine, the man–machine interface and the student. A tutorial system can be called a tutorial-like system if we replace the real-life students with some intelligent agents.

In tutorial-like system, domain model, teacher model and student model are three important parts. As shown in Fig. 4, these three parts constitute the backbone of a tutorial-like system.

Domain knowledge [30] is presented using a Socratic model via multiple choice questions. For each question, every choice is associated with a reward probability. The domain knowledge is analogous to the environment $E = [e_1, e_2, \ldots, e_r]$ in LA.

Teacher [32] attempts to present the domain material to students. Stochastic is the nature of this teacher model. That is to say, teacher doesn not pass on his knowledge directly to students, but replies to the students' choices through a heuristic approach. The option gets a “yes” or “no” response from teacher with a certain probability, which is equal to this choice's reward probability. Teacher's feedback has the same nature as the environmental response in LA.

Let β^i denotes the teacher's feedback to option i, and

$$\beta^i = \begin{cases} 1 & \text{with probability } e_i \\ 0 & \text{with probability } 1 - e_i \end{cases} \tag{2}$$

Student [37] learns by the philosophy of “trial and error”. At every time instant, student interacts with teacher by choosing one option from the candidate option set. Then teacher gives him a response of this choice, the response is generated in the aforementioned way. This feedback helps the student to refine his decision strategy. This process can be well modeled by learning automata. Different learning algorithms demonstrates different learning speed, which coincide with students with different learning speed. LA algorithms can be ranked according to their speed of convergence: Stochastic Estimator Algorithm, Generalized Pursuit Algorithm (GPA), Pursuit Algorithm, Non-estimator Variable Structure Stochastic Learning Automaton (VSSA) and Fixed Structure Stochastic Automaton (FSSA).

A student–classroom interaction in tutorial-like system is proposed in [34] by Oommen and Hashem. This classroom consists of three types of students, who have different learning speed. Knowledge provider, knowledge seeker and independent learner are three possible statuses of a student. One of them is selected as a student's interaction status by *Tactic-LA*. If a student is a knowledge seeker, he can pick an interaction strategy to communicate with a knowledge provider. Such strategies include *transferring no knowledge*, *embracing provider's knowledge*, *incorporating reliable knowledge* and *unlearning useless information after probation* [34].

However, there are some unreasonable setups in this system. The preset reward probability leads the *Tactic-LA* to converge to certain action, which is totally controlled by this manually preprogrammed probability rather than the current state of learning (see Section 3). Besides, this setting does not take each student's individual learning style into account. Furthermore, during every interaction, each student can only communicate with one other student.

3. Cooperative framework of learning automata

In this section, we propose a novel cooperative framework of learning automata, which is aimed to accelerate the learning speed of each automaton.

3.1. Model description

As mentioned before, learning automaton is defined by probability vector $P(t)$, estimator $D(t)$, learning algorithm $T(\bullet)$. As the learning algorithms $T(\bullet)$ are of different natures, it is impossible to combine these algorithms directly. Estimator $D(t)$ has been used for relating actions with feedbacks and storing the information of environment. This acts as the memory system and it is not appropriate to be the cooperation information. Probability vector $P(t)$ is the current decision-making basis and convergence is determined according to $P(t)$. Therefore, in our cooperative framework, we combine the different policies derived from the real-time probability vectors learned by the LA algorithms.

Next, we depict the model that we use for cooperation. Denote the number of learning automata that consisted in a cooperative learning system is N and the cooperation frequence is M. The learning process can be divided into two phases: *independent learning phase* and *cooperative learning phase*(shown in Fig. 5). An cooperation frequence M indicates the cooperative learning phase takes place every M instants. The whole learning process starts with independent learning phase. At independent learning phase, each member learner learns independently, recieving and updating themselves according to their own learning algorithm. After M-1 independent learning instants, a cooperative learning phase occurs. While at cooperative learning phase, the control center collects the probability vectors of all LA and calculates the average vector of them. Then, we calculate the SoL index of all learners and the SoL index of average probability $AvgP(t)$. The average SoL index is used as an standard to evaluate each learner's learning status and each learner will choose a interaction strategy based on this evaluation. Then each learner update themselves according to their interaction strategy and the system turns to independent learning phase again. This two phases interchanges until every learner have converged.

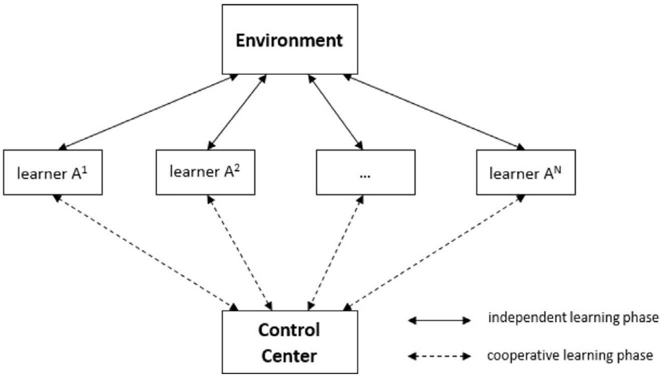


Fig. 5. Cooperative learning interaction model.

It is noted that, different from existing method, all learns in the proposed framework are forced to contribute to the collective intelligence (represented by avgeraged probability vector) and then decide whether draw lesson from the collective intelligence or not.

The SoL (current state of learning) index of each individual learning automaton A^i is defined as the difference between the two largest values of elements of probability vector. That is

$$SoL\{P^i(t)\} = \max_j\{P^i_j(t)\} - sub\max_j\{P^i_j(t)\} \tag{3}$$

where $P^i_j(t)$ denotes the probability of aciton a_j of automaton A^i being selected at time instant t.

Then the average probability can be written as:

$$AvgP(t) = AvgP_1(t), AvgP_2(t), \ldots, AvgP_r(t) \tag{4}$$

$$AvgP_i(t) = \frac{1}{N}\sum_{j=1}^{N} p^j_i(t) \tag{5}$$

At every cooperative learning phase, each learner A^i will choose an interaction strategy according to $SoL\{P^i(t)\}$ and $SoL\{AvgP(t)\}$ by the following rules:

1. If $SoL\{P^i(t)\} < SoL\{AvgP(t)\}$, which means the current performance of ith LA is below average standard. Thus, learner A^i become a *knowledge seeker* and he will use collective intelligence $AvgP(t)$ to fully replace its own decision-making tactic, i.e. $P^i(t) = AvgP(t)$.
2. If $SoL\{P^i(t)\} \geq SoL\{AvgP(t)\}$, which means learner A^i tend to be simply a *knowledge provider*. He is very confident about himself and would not change his decision-making tactic. Thus, we leave $P^i(t)$ unchanged.

3.2. Analysis of SoL index

As defined previously, SoL uses the difference value between $max\{P(t)\}$ and $submax\{P(t)\}$. This is analogous to the definition of the size of the learning problem in [38].

$$w = max\{E\} - submax\{E\} \tag{6}$$

To the author's perspective, if the difference w is large, automaton can distinguish the best action among the actions clearly. If the difference w is small, which means the reward probabilities are roughly equal, thus the automaton needs more time to pick the optimal one. Similarily, SoL index is related to the current learning state. At initialization step, each action share the equal probability value, thus $SoL = 0$. When the automaton is converged, the maximum probability is very close to 1. While others are reduced to 0, and $SoL = 1$. It can be seen that in the process of convergence, SoL value of each individual LA approaching 1 from initial value 0. Meanwhile, the action with the second largest probability is the most competitive candidate to the action with the largest value. If SoL index is large, action with largest value can be chosen at high probability, automaton is close to convergence and its learning achievements is remarkable. On the other hand, if the SoL index is small, the reward probabilities are roughly equal, thus the convergence trend is not obvious.

Let us take ith LA as example, cooperation step is happened every M instants.

Table 1
The number of iterations until convergence in five benchmark environments when SEri, DPri and DGPA algorithms working independently.

Enviroment	SEri				DPri			DGPA	
	Parameter[2]	Iteration	Accuracy	Parameter	Iteration	Accuracy	Parameter	Iteration	Accuracy
E_1	(16,8)	426	0.997	298	1086	0.995	33	880	0.997
E_2	(32,12)	834	0.996	653	2500	0.994	65	1677	0.996
E_3	(105,25)	2540	0.995	2356	9613	0.993	A 204	5191	0.995
E_4	(13,6)	325	0.998	216	783	0.996	28	754	0.997
E_5	(33,12)	729	0.997	881	2363	0.994	55	1445	0.997

[2] Parameter of SEri in is a two-tuples, where the first element is the resolution parameter n and the second element is γ [18].

Table 2
Iterations and improvement in five benchmark environments when SEri, DPri and DGPA algorithms working within cooperative framework.

Enviroment	LA	Cooperative framework		
		Iteration	Accuracy	Improvement(%)
	SEri	416	0.999	2.34
E_1	DPri	438	0.999	59.7
	DGPA	440	0.999	50.0
	SEri	749	0.999	10.2
E_2	DPri	788	0.999	68.5
	DGPA	792	0.999	52.8
	SEri	2038	0.999	19.8
E_3	DPri	2044	0.999	78.7
	DGPA	2097	0.999	59.6
	SEri	323	0.999	0.615
E_4	DPri	345	0.999	55.9
	DGPA	345	0.999	54.2
	SEri	673	0.999	7.68
E_5	DPri	709	0.999	70.0
	DGPA	710	0.999	50.9

1. If $mod(t, M) \neq 0$, LA are in the independent learning phase. ε-optimal feature is maintained in this phase. Based on Lemma 1 presented in [19], we have

   $$p_j^i(0) - t/(r \cdot n) < p_j^i(t) \le 1 \tag{7}$$

   where n is the resolution parameter of the algorithm and r in the number of candidate actions. $Q^i(t)$ is the state that there exists action α_i, time $t_i < \infty$ and for all $j \neq l, t > t_i$, we have $d_l^i(t) > d_j^i(t)$. When $Q^i(t)$ satisfied, we can obtain:

   $$p_l^i(t+1) - p_l^i(t) \ge 0 \tag{8}$$

2. If $mod(t, M) = 0$, indicating LA are in the cooperative learning phase, then
   (a) If $SoL\{P^i(t)\} \ge SoL\{AvgP(t)\}$, $P^i(t)$ remains unchanged, Eqs. (7) and (8) hold.
   (b) If $SoL\{P^i(t)\} < SoL\{AvgP(t)\}$, $P^i(t) \leftarrow AvgP(t)$,

   $$AvgP_j(t+1) > AvgP_j(0) - \frac{t}{r \cdot n} \tag{9}$$

   $$AvgP_j(t+1) - AvgP_i(t+1) \ge 0 \tag{10}$$

   After $P^i(t) = AvgP(t)$, we still have Eq. Eqs. (7) and (8).

From the above analysis, cooperation step will maintain the ε-*optimal* feature of learning automata.

3.3. Experimental simulation

In this section, the proposed cooperative framework is compared to single operated LA. Different algorithms such as DPri, DGPA and SEri are performed for a comprehensive comparison. Among them, DPri is a classic estimator algorithm and the most common method used in practice. DGPA is a generalization of DPri algorithm and faster than DPri [17]. SEri is the fastest algorithm in MLE-based LA family[1] reported to date.

We follow the same experimental configuration given in [18]. Threshold T=0.999, number of experiments NE=250,000 and cooperative frequency M=10. Five ten-action ($r = 10$) benchmarks are considered for simulation studies. And all LA involved in this cooperative learning framework use their corresponding "best" parameter when performing independently (given in Table1).

Indicators for performance comparison are defined as below. Accuracy is calculated as *correctly convergence/NE*. The improvement is obtained by calculating:

$$(Iteration_{\{newmodel\}} - Iteration_{\{independent\}})/Iteration_{\{independent\}} \tag{11}$$

The first comparison is made between cooperative framework and independent model. These results are shown in Tables 1 and 2. On convergence precision, compared to the member learning independently, all members improve the performance, which improved to 0.999. On the iteration numbers, the environment E_3 of DPri algorithm improves largest. In the corresponding case of independent learning, the automata needs 9613 iterations for convergence and the convergence precision is 0.993. In the cooperative learning automata which contain DPri, DGPA and SEri, the automata only need 2044 iterations and the convergence precision is 0.999, which increases the speed of 78.7%. And in the environment E_1, DPri algorithm needs 880 iterations and the

[1] Hao et al. proposed a confidence interval estimator based learning autoamata algorithm which claims a faster convergence than SEri. [21]

convergence precision is 0.997, while the cooperative framework only needs 440 iterations for convergence and convergence precision is 0.999, which increases the speed twice.

Then the comparision is made between cooperative framework with and without SoL. This comparison is intended to show the effectiveness of SoL index. Cooperative framework without SoL always execute $P^i(t) \leftarrow AvgP(t)$. From Table 3, it is clearly that the system consisting of DPri, DGPA and SEri, increases the speed of DPri and DGPA, but decreases the speed of SEri significantly. In Table 3, SEri improves the iterations by -17.45%, -20.74%, -37.00%, -18.87% and -24.55%. However, it is worth noting that the cooperative framework with SoL shows a win–win state. The cooperative framework with SoL comprised of three models, DPri and DGPA do not slow down SEri's speed. As is shown in the Table 2, SEri improves the iterations by 2.34%, 10.2%, 19.8%, 0.615% and 7.68% in environment E_1 to E_5, respectively. And in the harder environments, such as E_3, SEri need less iterations, so we can draw the conclusion safely that the cooperative framework with SoL has a better performance.

4. Application in tutorial-like system

In this section, the proposed cooperative framework is applied to simulate the students' interactions in tutorial-like system. The student, modeled by LA, is able to interact with the teacher and cooperate with other fellow team-workers.

Table 3
Interactions and improvement in five benchmarks when SEri, DPri and DGPA algorithms working in cooperative framework without SoL.

Enviroment	LA	cooperative framework without SoL		
		Iteration	Accuracy	Improvement(%)
E_1	SEri	500	0.999	-17.45
	DPri	533	0.999	50.89
	DGPA	529	0.999	39.87
E_2	SEri	1007	0.999	-20.74
	DPri	1058	0.999	57.65
	DGPA	1053	0.999	37.20
E_3	SEri	3479	0.999	-37.00
	DPri	3690	0.999	61.61
	DGPA	3673	0.999	29.23
E_4	SEri	386	0.999	-18.87
	DPri	427	0.999	45.43
	DGPA	424	0.999	43.75
E_5	SEri	907	0.999	-24.55
	DPri	951	0.999	59.72
	DGPA	946	0.999	34.53

4.1. Cooperation method in student-team

Cooperation occurs every M iterations. At every cooperative learning phase, all students calculate SoL index as self-examination indicator. In order to provide a standard to evaluate the SoL value of a student is small or large, the average of the team's probability vector is adopted. Average probability vector reveals the collective wisdom of the student-team.

If $SoL\{P^i(t)\} < SoL\{AvgP(t)\}$, which means the current observed index SoL of the ith student is lower than the SoL of average. The performance of the simulator is below normal standard. In our proposed cooperative method, this type of student is becoming a knowledge seeker. He uses the average probability vector to fully replace his knowledge.

If $SoL\{P^i(t)\} \geq SoL\{AvgP(t)\}$, which means the current observed index SoL of the ith student is equal or greater than the SoL of average. The performance of the simulator is above normal standard. In our proposed cooperative method, this type of student is becoming a knowledge provider. He offers help to others in an implicit way. He contributes to the team through affecting the SoL of average probability.

Here gives a explanation of the difference between interaction with teacher and the cooperation among student-team.

Teacher helps student to refine his decision strategy to achieve the ultimate goal. All the students can learn the correct answer from teacher, but the learning speed differs from others.

However, the cooperation among student-team is to accelerate the learning process. It could be possible that a student has a strong preference of bad action, and SoL of this student is higher than the average baseline. He refuse to accep any collective knowledge in cooperative learning phase, but after sufficie interaction with teacher, student can gradually change this preference and benefit from the collective in the future cooperation.

4.2. Experimental results

In this section, experimental results are presented to test the implementation of cooperative student-team. These results were obtained by performing numerous simulation experiments.

Table 4 shows a comparison between scenario 1 and 2. Scenario 1 is our proposed cooperative student-team system. Scenario 2 is the best situation in the student–classroom interaction system published in [34]. We adopt exactly the same environment setting and parameters as [34], i.e.,three types of LA, to simualte fast-learning student, normal-learning student and below-normal-learning student respectively, and there are three students of

Table 4
Interactions and improvement in four benchmarks when SEri, DPri and DGPA algorithms working in cooperative framework with and without SoL.

Environment	Student Type	Single student		Scenario 1			Scenario 2		
		Iterations	Wrong	Iterations	Improvement(%)	Wrong	Iterations	Improvement(%)	Wrong
$E_{4,A}$	Fast	572	0	549	4	0	492	14	0
	Normal	996	0	535	46	0	476	52	0
	Below	1382	0	549	60	0	507	63	0
$E_{4,B}$	Fast	1482	0	731	51	0	972	34	0
	Normal	2201	0	683	69	0	879	60	9
	Below	2633	0	697	73	0	758	71	9
$E_{10,A}$	Fast	686	1	614	10	0	585	15	0
	Normal	1297	1	612	53	0	570	56	1
	Below	1804	0	629	65	0	586	68	1
$E_{10,B}$	Fast	1655	4	876	47	1	1053	36	1
	Normal	2114	4	783	63	1	780	63	13
	Below	2859	4	810	72	1	784	73	13

each type. Cooperative learning was performed under four existing benchmark environments, cooperative frequency M is set to be 10, the threshold T was set to be 0.99 and number of experiments NE was 75.

The results of the comparisons are given in Table 4. In scenario 1, all types of students benefited from the proposed strategy. The convergence speed is comparative to the scenario 2 and even better when the environment is becoming harder. For example, in environment $E_{4,B}$, all types of students got more improvement than what they got in scenario 2. Moreover, the accuracy is also improved. In scenario 2, nine out of NE times normal and below normal students got the wrong answer. While in our proposed system, all types of students converge correctly.

From above data analysis, the cooperative student-team does well. Learning speed and accuracy are both benefited from the proposed strategy. This novel idea of cooperation provides another reference for the real-life students' learning, as well as explores a new research field in the intelligent tutorial systems (ITSs).

5. Conclusion

This paper presents a brand new cooperative framework of learning automata. Different types of learning automata are embedded in the proposed cooperative framework, which is proved to be able to improve the learning speed of all LA. SoL is advocated as a self-exam indicator, which is defined as the difference between the two largest probability values. Theoretical analysis indicated that this index is able to preserve the ε-optimal feature. Experiments verified that this cooperative framework is effective in improving learning speed of various LA embedded in this framework. An application in tutorial-like system is presented as a successful case applying the cooperative framework. The simulation evidently showed that the convergence speed of proposed methods is comparative to the student–classroom interaction method and even better when the environment is becoming harder. The accuracy is clearly improved compared to the previous interaction method and the single operated student.

However, this paper does not aim to outperform all the proposed method, but to show the cooperative interaction can effectively serve as an attractive method to modeling in tutorial-like system. For future work, we believe that a cooperative method can be used in the LA theoretical field to enhance learning speed without compromising accuracy. And the study of student simulators in tutorial-like system is still open.

Acknowledgement

This research work is funded by the National natural Science Foundation of China (61271316), 973 Program of China (2013CB329605), Key Laboratory for Shanghai Integrated Information Security Management Technology Research, and Chinese National Engineering Laboratory for Information Content Analysis Technology.

References

[1] W. Zhong, Y. Xu, J. Wang, D. Li, H. Tianfield, Adaptive mechanism design and game theoretic analysis of auction-driven dynamic spectrum access in cognitive radio networks, EURASIP Wirel Commun Netw 1 (2014) (2014) 44.

[2] W. Jiang, C.-L. Zhao, S.-H. Li, L. Chen, A new learning automata based approach for online tracking of event patterns, Neurocomputing 137 (2014) 205–211.

[3] PV. Krishna, S. Misra, D. Joshi, MS. Obaidat, Learning automata based sentiment analysis for recommender system on cloud, in: 2013 International Conference on Computer, Information and Telecommunication Systems (CITS), IEEE, 2013, pp. 1–5.

[4] A. Rezvanian, M. Rahmati, M.R. Meybodi, Sampling from complex networks using distributed learning automata, Physica A: Stat. Mech. Appl. 396 (2014) 224–234.

[5] B. Moradabadi, H. Beigy, A new real-coded bayesian optimization algorithm based on a team of learning automata for continuous optimization, Genet. Program. Evolvable Mach. (2013) 1–25.

[6] W. Yuan, H. Leung, W. Cheng, S. Chen, Optimizing voting rule for cooperative spectrum sensing through learning automata, IEEE Trans. Veh. Technol. 60 (7) (2011) 3253–3264.

[7] J.A. Torkestani, Laap: a learning automata-based adaptive polling scheme for clustered wireless ad-hoc networks, wirel. Personal Commun. 69 (2) (2013) 841–855.

[8] S. Misra, B.J. Oommen, S. Yanamandra, M.S. Obaidat, Random early detection for congestion avoidance in wired networks: a discretized pursuit learning-automata-like solution, IEEE Trans. Syst. Man Cybern., Part B 40 (1) (2010) 66–76.

[9] G. Horn, B.J. Oommen, Solving multiconstraint assignment problems using learning automata, IEEE Trans. Syst. Man Cybern. part B 40 (1) (2010) 6–18.

[10] S. Misra, P. Krishna, K. Kalaiselvan, V. Saritha, M. Obaidat, Learning automata-based qos framework for cloud iaas, IEEE Trans. Netw. Serv. Manag. 99 (2014) 1–10.

[11] F. Fathy, N. Salek, Y. Masoudi, E. Laleh, Distributing of patterns in cutter machines boards using learning automata, in: Communication Systems and Network Technologies (CSNT), 2013, pp. 774–777.

[12] S. Misra, P. Krishna, V. Saritha, M. Obaidat, Learning automata as a utility for power management in smart grids, IEEE Commun. Mag. 51 (1) (2013) 98–104.

[13] A. Yazidi, O.-C. Granmo, B. Oommen, Learning-automaton-based online discovery and tracking of spatiotemporal event patterns, IEEE Trans. Cybern. 43 (3) (2013) 1118–1130.

[14] N. Rasouli, M. Meybodi, H. Morshedlou, Virtual machine placement in cloud systems using learning automata, in: Fuzzy Systems (IFSC), 2013, pp. 1–5.

[15] M. Thathachar, P. Sastry, A new approach to the design of reinforcement schemes for learning automata,, IEEE Trans. Syst. Man Cybern. 1 (1985) 168–175.

[16] BJ. Oommen, Recent advances in learning automata systems, in: Computer Engineering and Technology (ICCET), vol. 1, IEEE, 2010, pp. V1–724.

[17] M. Agache, B.J. Oommen, Generalized pursuit learning schemes: new families of continuous and discretized learning automata, IEEE Trans. Syst. Man Cybern. Part B: Cybern. 32 (6) (2002) 738–749.

[18] G.I. Papadimitriou, M. Sklira, A.S. Pomportsis, A new class of ε-optimal learning automata, IEEE Trans. Syst. Man Cybern. Part B: Cybern. 34 (1) (2004) 246–254, http://dx.doi.org/10.1109/TSMCB.2003.811117.

[19] B.J. Oommen, J.K. Lanctôt, Discretized pursuit learning automata, IEEE Trans. Syst. Man Cybern. 20 (4) (1990) 931–938.

[20] J. Zhang, C. Wang, M. Zhou, Last-position elimination-based learning automata, IEEE Trans. Cybern. 44 (12) (2014) 2484–2492, http://dx.doi.org/10.1109/TCYB.2014.2309478.

[21] H. Ge, W. Jiang, S. Li, J. Li, Y. Wang, Y. Jing, A novel estimator based learning automata algorithm, Appl. Intell. 42 (2) (2015) 262–275, http://dx.doi.org/10.1007/s10489-014-0594-1.

[22] M. Thathachar, M. Arvind, Parallel algorithms for modules of learning automata, IEEE Trans. Syst. Man Cybern. Part B: Cybern. 28 (1) (1998) 24–33.

[23] M. Thathachar, K. Ramakrishnan, A hierarchical system of learning automata, IEEE Trans. Syst. Man Cybern. 11 (3) (1981) 236–241.

[24] M. Thathachar, P.S. Sastry, A hierarchical system of learning automata that can learn die globally optimal path, Inf. Sci. 42 (2) (1987) 143–166.

[25] G.I. Papadimitriou, Hierarchical discretized pursuit nonlinear learning automata with rapid convergence and high accuracy, IEEE Trans. Knowl. Data Eng. 6 (4) (1994) 654–659.

[26] N. Baba, Y. Mogami, A relative reward-strength algorithm for the hierarchical structure learning automata operating in the general nonstationary multi-teacher environment, IEEE Trans. Syst. Man Cybern. Part B: Cybern. 36 (4) (2006) 781–794.

[27] M. A. Thathachar, P. S. Sastry, Networks of Learning automata: Techniques for online stochastic optimization, Springer Science& Business Media, 2003.

[28] B.J. Oommen, M.K. Hashem, Modeling a student's behavior in a tutorial-like system using learning automata, IEEE Trans. Syst. Man Cybern. Part B: Cybern. 40 (2) (2010) 481–492.

[29] K. Hashem, B. J. Oommen, Using learning automata to model a domain in a tutorial-like system, in:2007 International Conference on Machine Learning and Cybernetics, vol. 1, IEEE, 2007, pp. 112–118.

[30] B.J. Oommen, M.K. Hashem, Modeling a domain in a tutorial-like system using learning automata, Acta Cybern 19 (3) (2010) 635–653.

[31] K. Hashem, BJ. Oommen, On using learning automata to model a student's behavior in a tutorial-like system, in: New Trends in Applied Artificial Intelligence, Springer, 2007, pp. 813–822.

[32] K. Hashem, BJ. Oommen, Using learning automata to model the behavior of a teacher in a tutorial-like system, in: Systems, Man and Cybernetics, 2007. ISIC. IEEE International Conference on, IEEE, 2007, pp. 76–82.

[33] B.J. Oommen, M.K. Hashem, Modeling the learning process of the teacher in a tutorial-like system using learning automata, IEEE Trans. Cybern. 43 (6) (2013) 2020–2031.

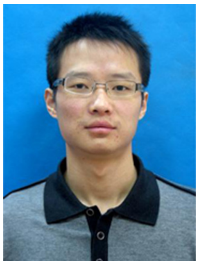
[34] B.J. Oommen, M.K. Hashem, Modeling a student–classroom interaction in a tutorial-like system using learning automata, IEEE Trans. Syst. Man Cybern. Part B: Cybern. 40 (1) (2010) 29–42.

[35] B.J. Oommen, M. Agache, Continuous and discretized pursuit learning schemes: various algorithms and their comparison, IEEE Trans. Syst. Man Cybern. Part B: Cybern. 31 (3) (2001) 277–287.

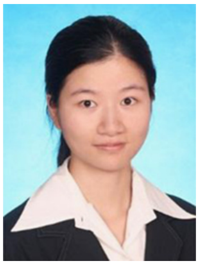
[36] M. Thathachar, P.S. Sastry, Varieties of learning automata: an overview, IEEE Trans. Syst. Man Cybern. Part B: Cybern. 32 (6) (2002) 711–722.

[37] K. Hashem, BJ. Oommen, Using learning automata to model a student–classroom interaction in a tutorial-like system, in: Systems, Man and Cybernetics, 2007. ISIC. IEEE, 2007, pp. 1177–1182.

[38] K. Rajaraman, P. Sastry, Finite time analysis of the pursuit algorithm for learning automata, IEEE Trans Syst. Man Cybern. Part B: Cybern. 26 (4) (1996) 590–598.



Hao Ge received B.E. degree School of Information Science and Engineering at Southeast University, Nanjing, China in 2010. He is currently working toward the Ph.D. degree with School of Electronic, Information and Electrical Engineering, Shanghai Jiao Tong University. His research interests include learning automata and their applications, artificial neutral network, computer communication network and data mining.



Yifan Wang received the B.E. degree in automation from Nanjing University of Posts and Telecommunications, Jiangsu, China, in 2012. She is currently working toward the M.E. Degree in information and communication engineering at the Shanghai Jiao Tong University, Shanghai, China. She was selected as outstanding student leader of Jiangsu province and won the National Scholarship and twice the First Class Scholarship of Nanjing University of Posts and Telecommunications. She is also in Information Security Institute of Shanghai Jiao Tong University, Shanghai, China. Her research interest include networks of learning automata and its applications, recommendation system.

Shenghong Li received the B.S. and the M.S. degrees in electrical engineering from Jilin University of Technology, China, in 1993 and 1996 respectively, and received the Ph.D. degree in radio engineering from Beijing University of Posts and Telecommunications, China, in 1999. Since September 1999, he has been working in Shanghai Jiaotong University, China, as research fellow, associate professor and professor, successively. In 2010, he worked as visiting scholar in Nanyang Technological University, Singapore. His research interests include information security, signal and information processing, artificial intelligence. He published more than 80 papers, co-authored four books, and holds ten granted patents. In 2003, he received the 1st Prize of Shanghai Science and Technology Progress in China. In 2006 and 2007, he was elected for New century talent of Chinese Education Ministry and Shanghai dawn scholar.