Absorbing Stochastic Estimator Learning Algorithms with High Accuracy and Rapid Convergence

G.I.Papadimitriou A.S.Pomportsis S.Kiritsi E.Talahoupi

Department of Informatics, Aristotle University, Box 888, 54006 Thessaloniki, Greece.
email:gp@csd.auth.gr

Abstract

An absorbing learning automaton which is based on the use of a stochastic estimator is introduced. According to the proposed stochastic estimator scheme, the estimates of the reward probabilities are computed stochastically. Actions that have not been selected many times have the opportunity to be estimated as optimal, to increase their choice probabilities, and consequently, to be selected. In this way, the automaton's accuracy is significantly improved. This proposed automaton is proven to be absolutely expedient in all stationary environments, while the simulation results demonstrate that the proposed scheme achieves a significantly higher performance in comparison with the deterministic estimator based schemes.

I. Introduction

Learning automata (LA) [2],[14] have attracted considerable interest in the last decades due to their potential usefulness in a variety of engineering problems that are characterized by nonlinearity and a high level of uncertainty. They have the property of progressively improving their performance. A learning system is connected in a feedback loop to the environment where it operates.

The environment communicates with the learning system and supplies it with information. Every time that the LA selects one from a set of actions the feedback from the environment tells the LA whether the chosen action was rewarded or penaltized. The automaton uses this information to find which action is optimal. A learning automaton is one that adapts itself to the environment by learning the optimal action and that ultimately chooses this action more frequently than other actions.

The environment is said to be stationary if the reward probabilities are not dependent on the time, otherwise it is said to be nonstationary.

Learning systems can be compared to adaptive systems and also they are closely linked to artificial intelligence. The applications of learning automata cover a wide range of problems: bioreactors, computers, image processing, optimization, pattern recognition, robots, thermal reactors, neural network synthesis [2] and communication networks [15]-[17].

Different kinds of learning systems have been proposed. According to the nature of its input a learning automaton can be characterized as a P, Q or S-model. A learning automaton is a Q-model automaton [11],[12] if the input set is a finite set of distinct symbols. if the input set is binary ({0,1}), it is called a P-model automaton [7],[11],[12]. Finally, if the automaton's input can take any real value in the [0,1] range, the automaton is called an S-model one [1],[8],[9],[11],[12].

With respect to their Markovian representation, learning automata are classified into two main categories: ergodic or automata-possessing absorbing barriers [11],[12].[14]. The ergodic automata converge with a distribution independent of the initial state. On the other hand, the automata with absorbing states, after a number of a finite steps, get locked (converge) into a particular action. If the reward probabilities of the actions are stable (stationary environment), absorbing automata are preferred.

Another way to classify learning automata is according to the values that the action probabilities can take. By this view, a learning automaton can be characterized as a continuous or a discretized one [3]. In the former case, the action probabilities can take any value in the [0,1] interval. In the latter, the action probabilities can take values from a finite set only. In other words, the [0,1] interval is divided into a finite number of subintervals. If these subintervals are all

0-7695-1165-1/01 $10.00 © 2001 IEEE

of equal length, the automaton is characterized as a linear one; if not, it is nonlinear one.

Non-estimator learning algorithms update the probability vector based directly on the environment' s feedback. If the selected action is rewarded, then the automaton increases the probability of choosing that action. If the answer is penalty, then the probability of choosing that action at the next time instant is decreased. On the other hand, estimator algorithms [6] are characterized by the use of a running estimate of the reward probability of each action. The change in the probability of choosing an action is based on the running estimates of the probability of reward rather than on the feedback from the environment. That means that even when an action is rewarded it is possible that the probability of choosing another action is increased. These algorithms, at every time instant, increase the probability of choosing the action with the maximum current estimate of reward probability. Simulation results have demonstrated the superiority of the estimator algorithms over the traditional learning algorithms [6], [7].

Stochastic Estimator Learning Algorithms scheme were introduced in [1] as an effort to achieve a high adaptivity when the automaton operates in rapidly switching non-stationary environments.

In this paper we present a new stochastic-estimator-based scheme, which is capable of achieving a high accuracy and a rapid convergence when operating in stationary environments.

Section II introduces the reader to the basic concepts of the stochastic estimator. The presentation of the proposed Absorbing Stochastic Estimator Learning Automaton (ASELA) scheme in section III is followed by the proof of its absolute expediency in section IV. In section V extensive simulation results are presented, which indicate that the proposed scheme achieves a high performance when operating in stationary environments. Finally, concluding remarks are given in section VI.

I. The stochastic estimator

A stationary environment is considered. In case of an S-model envrironment, the learning automaton keeps running estimates of the actions' mean rewards. When the environment is a P-model one, estimates of the actions' reward probabilities are used. The estimates are computed stochastically, so they are not strictly dependent on the environmental responses. A zero mean normally distributed random variable is added to the deterministic estimates in order to take the stochastic ones. The variance of the normally distributed random variable, differs from action to action and is proportional to the reverse value of the number of times that each action was selected. The learning automaton gives to actions that have been selected only a few times - and so their estimates are considered as unreliable - the opportunity to be estimated as "optimal", to increase their choice probability and consequently, to be selected.

This kind of estimator, which determines the estimates of actions in a nondeterministic way, is called a stochastic estimator. The use of stochastic estimator in nonstationary environments has been studied in [1]. In the present paper we are going to study the use of stochastic estimator in stationary environments.

II. The Absorbing Stochastic Esrimator Learning Automaton (ASELA)

The ASELA learning automaton is defined by the quintuple {A, B, P, E, T}.

A is the set of r actions that the automaton can choose from. The various actions are then elements of $A=\{a_1,\ldots,a_r\}$. The automaton is allowed one choice at each time instant t and its choice is denoted as a(t). This choice is constrained by the requirements that $a(t)\epsilon A$ for all t.

The set of possible responses from the environment is denoted by B. When an S-model environment is considered, then B=[0,1]. Otherwise, when a P-model environment is considered, then B={0,1}. The environmental response at time instant t is denoted by b(t).

When an S-model environment is considered, the mean reward of action a_i at time instant t is denoted by $d_i(t)$. Since the environment is stationary, $d_i(t)$ is constant for all time t and so the index of time is dropped and the quantity is denoted as d_i. The set of the actions' mean rewards is defined as: $D=\{d_1,\ldots,d_r\}$. When a P-model environment is

considered, D denotes the set of the actions' reward probabilities.

It is assumed that there is a unique maximum for the set D called d_b, where $d_b=\max_{1\le i\le r}\{d_i\}$. The action possessing d_b, namely a_b, is referred to as the best action. The value of each $d_i \in D$ is unknown to the automaton, and so its task is to decide which action is the best. It bases its decision on the information gained by selecting actions, and seeing the environmental feedback. This cycle continues until the learning process is terminated.

P is a probability distribution over the set of actions. We have $P(t)=\{p_1(t),\ldots,p_r(t)\}$, where $p_i(t)$ is the probability of selecting action a_i at time instant t. In discretized automata there are only finitely many vales for $p_i(t)$, namely $p_i(t)$ is one of $\{0, \Delta, 2\Delta, 3\Delta, \ldots, 1\}$ for all t. Here Δ is referred to as the smallest step size and is inversely proportional to the total number of subdivisions of the probability space [0,1]. This parameter Δ is defined by $\Delta=1/N$, where $N=rn$, r is the number of actions and n is the resolution parameter.

E is the estimator that at any time instant contains the estimated environmental characteristics. we define $E(t)=(D'(t), M(t), U(t))$, where $D'(t)=\{d'_1(t),\ldots,d'_r(t)\}$ is the Deterministic Estimator Vector, which contains the current deterministic estimates of the mean rewards of the actions as shown below (for i=1,…,r):

$$d'_i(t) = \frac{\text{The total reward received by the automaton up to time instant t for the selections of actions } a_i}{\text{The number of times the action } a_i \text{ has been selected up to time t}}$$

$$= \frac{\sum_{k=1}^{T_i} Q_i^k(t)}{m_i(t)} \qquad (1)$$

where $Q_i^k(t)$ for k=1,…,T_i are the rewards received at each time k that action a_i was selected and T_i is the last time that a_i was selected. $M(t)=\{m_1(t),\ldots, m_r(t)\}$ where $m_i(t)$ is the number of times that the action a_i has been selected up to time instant t $U(t)=\{u_1(t),\ldots,u_r(t)\}$ is the Stochastic Estimator Vector which at any time instant t, contains the current stochastic estimates of the mean rewards of the actions. The current stochastic estimate $u_i(t)$ of the mean reward of action a_i is defined as follows:

$$u_i(t) = d'_i(t) + N\left(0, s_i^2(t)\right) \qquad (2)$$

where $s_i(t) = \min\left\{a \cdot \frac{1}{m_i(t)}, s_{\max}\right\}$.

$N(0, s_i^2(t))$ denotes a random number selected with a normal probability distribution, with a mean equal to 0 and a variance equal to $s_i^2(t)$. a is an internal automaton's parameter that determines how rapidly the stochastic estimates become independent from the deterministic ones. When a=0, no noise is added to the deterministic estimates. s_{max} is the maximum permitted value of $s_i(t)$ (i=1,…,r). It limits the variance of the stochastic estimates in order not to increase infinitely.

Finally, T is the learning algorithm which is presented below:

STEP 1: Select an action $a_i(t)=a_k$ according to the probability vector.

STEP 2: Receive the feedback $b(t) \in [0,1]$ from the environment.

STEP 3: Update M(t) by setting $m_k(t+1)=m_k(t)+1$ and $m_i(t+1) = m_i(t)$ for all $i\neq k$.

STEP 4: Compute the new deterministic estimate $d'_k(t)$ as it is given by relation (1).

STEP 5: For every action a_i (i=1,..,r) compute the new stochastic estimate $u_i(t)$ as it is given by relation (2).

STEP 6: Select the "optimal" action a_m that has the highest stochastic estimate of mean reward. Thus $u_m(t)=\max_i\{u_i(t)\}$.

STEP 7: Update the probability vector in the following way:

for every action a_i (i=1,…,m-1,m+1,…,r), with $p_i(t)\ge 1/N$, set the following condition:

$p_i(t+1)=p_i(t)-1/N$

For the "optimal" action a_m set

$$p_m(t+1) = 1 - \sum_{i\neq m} p_i(t+1)$$

STEP 8: If $p_m(t)<1$ then GOTO to Step 1

else CONVERGE to action a_m

The above algorithm can be applied in both S-model and P-model environments.

An S-model environment consists of three components denoted by (A,L,B), where A and B are as defined above and L=(D,F). $D=\{d'_1, \ldots, d'_r\}$ is the set that contains the mean rewards of the actions at any time instant. $F(t)=\{f_1^t(x), \ldots, f_r^t(x)\}$ is the set that contains the probability density functions of the actions rewards at every time instant t. $f_i^t(x)$ is symmetric about the line $x=d_i^j=d_i-d_j$. As our automaton operates in a stationary environment the means and the density functions of the actions rewards are time-invariant.

A P-model environment is defined in the same way, except that L=D, where D is the set of the reward probabilities of actions.

III. Analysis

Theorem 1: In every S-model or P-model stationary random environment that offers symmetrically distributed noise, the ASELA learning automaton is absolutely expedient. Thus,

$$\text{if } R(t)=\sum_{i=1}^{r} d_i\, p_i(t) \text{ then } E\left[R(t+1)\mid P(t)\right] > R(t)$$

for all t, for all $p_i(t) \in (0,1)$, i=1,2,...,r and for all possible values of d_i, i=1,2,...,r, assuming that the maximum reward (let d_i) is unique.

Proof: For the sake of brevity the proof is omitted.

IV. Simulation results

Simulation was performed to demonstrate the superiority of the proposed stochastic estimator scheme toward the deterministic estimator one [3]. Both schemes were simulated to operate in P-model and Q-model stationary environments. The speed and accuracy of convergence were used as performance metrics, in order to evaluate the two scheme which are under comparison. For both automata was said to have converged when the probability of choosing an action was exactly unity. The two automata that are under comparison were placed in a ten- action environment. The reward probability (or the mean reward) of the optimal action was fixed at 0.85 for all simulations, while the reward probabilities of the other actions were equally spaced in the interval [0.1, 0.5], as in the r-action case we have $d_1=0.85$ and $d_i=0.5-(i-2)\delta$ for i=2,3,...,r where $\delta=(0.5-0.1)/(r-2)$.

Before starting the algorithm initial estimates for D' were obtained by selecting each action once. This extra iteration was then included in the total number of iterations until the algorithm converged. The average results are shown at Tables I to VI. The value of the resolution parameter appears in the first column of each table. In each case the resolution parameter N is defined by relation $\Delta=1/N$, N=rn, where as referred earlier r is the number of actions and Δ is the step size of the probability vector. The second column contains the accuracy that corresponds to the resolution parameter of the first column. The automaton is said to have converged accurately, if it converged to the best action. The mean number of iterations required for convergence is appeared at the third column.

The noise in the environment was simulated by taking truncated samples from normal (Gaussian) distributions with the mean rewards and a variance σ^2. Figures 1 to 6 represent the simulation results for the two automata for various values of Δ and the variance σ^2 of the Gaussian environmental noise and the internal automaton's parameter a. As a result of our experiments we observe that for a given accuracy, in both versions (S-model and P-model) the stochastic automaton is faster than the deterministic one. For example for the same accuracy (90.3) the stochastic S-model automaton requires almost only 50% of the iterations (16.78) that corresponding deterministic automaton requires for convergence (30.89). Furthermore, it becomes clear that the accuracy of the scheme is remarkably high.

V. Conclusion

An absorbing learning automaton that uses a stochastic estimator in order to achieve a high accuracy and a high speed of convergence in stationary environments is introduced. Extensive simulation results are presented that indicate that the proposed ASELA scheme achieves a superior performance over the well-known deterministic-estimator-based absorbing schemes when they operate in stationary environments. Furthermore, it is proved that the proposed ASELA learning automaton is absolutely expedient in every stationary random environment.

The stochastic estimator innovation can be the base of a new generation of powerful learning automata with a broad range of applications. We are currently working in this direction.

References

[1] G.I.Papadimitriou "A New Approach to the Design of Reinforcement Schemes for Learning Automata: Stochastic Estimator Learning Algorithms" IEEE Transactions on Knowledge and Data Engineering, vol.6, no.4, August 1994.

[2] K.Najim and A.S.Poznyak "Learning Automata: Theory and applications".

[3] B.J.Oommen and J.K.Lanctot, "Discretized pursuit learning automata," IEEE Trans. Syst., Man, Cybernetics, vol. SMC-20, no. 4, pp.931-938,July/Aug.1990.

[4] B.J.Oommen and J.P.R.Christensen. "Epsilon optimal discretized linear reward-penalty learning automata," IEEE Trans. Syst., Man, Cybernetics, vol. SMC-18, no. 3, pp. 451-458, May/June 1988.

[5] S.Lakshmivarahan and M.A.L.Thathachar." Absolutely expedient learning algorithms for stochastic automata," IEEE Trans. Syst., Man, Cybernetics, vol. SMC-3, pp. 281-286. May 1973.

[6] M.A.L.Thathachar and P.S.Sastry, "A class of rapidly converging algorithms for learning automata," IEEE Trans. Syst., Man, Cybernetics, vol. SMC-15, no. 1, pp. 168-175, Jan./Feb. 1985.

[7] G.I.Papadimitriou, "Hierarchical discretized pursuit nonlinear learning automata with rapid convergence and high accuracy," IEEE Transactions on Knowledge and Data Engineering, vol.6, no.4, August 1994.

[8] R.Viswanathan and K.S.Narendra," Stochastic automata models with applications to learning systems, "IEEE Trans. Syst. Man, Cybernetics, vol. SMC-3, pp. 107-111, Jan. 1973.

[9] R.Simha and J.F Kurose," Relative reward strength algorithms for learning automata," IEEE Transactions on Systems, Man and Cybernetics, vol. 19, no. 1, pp. 388-398, Mar./April 1989.

[10] G. I. Papadimitriou and D.G. Maritsas, "WDM passive star networks: Receiver collision avoidance algorithms using multifeedback learning automata," in 17th IEEE Conf. Local Comput. Networks, Minneapolis, MN, USA, 13-16 Sept. 1992.

[11] K.S.Narendra and M.A.L.Thathachar, "Learning automata: A survey," IEEE Trans. Syst., Man, Cybernetics, vol. SMC-4 no. 4, pp. 323-334, July 1974.

[12] K.S.Narendra and S.Lakshmivarahan, "Learning automata: A critique," J. Cybernetics and Inform, Sci., vol. 1, pp. 53-66, 1977.

[13] O.V.Nedzelnitski and K.S.Narendra, "Nonstati-onary models of learning automata routing in data communication networks."IEEE Transactions on Systems., Man and Cybernetics, vol. SMC-17, no. 6, pp. 1004-1015, Nov./Dec. 1987.

[14] K.S.Narendra and M.A.L.Thathachar, "Learning Automata: An Introduction", Prentice Hall, New Jersey, 1989.

[15] G.I.Papadimitriou and D.G.Maritsas, "Learning Automata-Based Receiver Conflict Avoidance Algorithms for WDM Broadcast-and-Select Star Networks", IEEE/ACM Transactions on Networking, vol.4, no.3, June 1996.

[16] G.I.Papadimitriou and A.S.Pomportsis, "Learning-Automata-Based TDMA Protocols for Broadcast Communication Systems with Bursty Traffic", IEEE Communications Letters, vol.4, no.3, March 2000.

[17] G.I.Papadimitriou and A.S.Pomportsis, "Self-Adaptive TDMA Protocols for WDM Star Networks: A Learning-Automata-Based Approach", IEEE Photonics Technology Letters, vol.11, no.10, pp.1322-1324, October 1999.

	USING STOCHASTIC ESTIMATOR		USING DETERMINISTIC ESTIMATOR ONLY	
	Accuracy	Speed	Accuracy	Speed
N=25	95.48	11.95	86.62	4.95
N=50	98.58	18.67	91.42	9.72
N=100	99.51	33.21	96.33	17.7
N=200	99.71	55.19	97.97	37.01
N=300			98.96	45.79

Table I

Stochastic vs. Deterministic Estimator in an S-model automaton with α=0.25, in a 10-Action Environment with σ^2=0.25.

	USING STOCHASTIC ESTIMATOR		USING DETERMINISTIC ESTIMATOR ONLY	
	Accuracy	Speed	Accuracy	Speed
N=25	97.2	16,66	4.95	86.62
N=50	99.41	25.7	9.72	91.42
N=100	99.8	39.28	17.7	96.33
N=200	99.89	61.7	37.01	97.97
N=300			45.79	98.96

Table II

Stochastic vs. Deterministic Estimator in an S-model automaton with α=0.35, in a 10-Action Environment with σ^2=0.25.

	USING STOCHASTIC ESTIMATOR		USING DETERMINISTIC ESTIMATOR ONLY	
	Accuracy	Speed	Accuracy	Speed
N=25	90.39	16.78	75.23	8.72
N=50	96.27	29.58	82.36	16.13
N=100	97.93	49.03	90.6	30.89
N=200	98.6	76.51	95.01	48.5
N=300			96.75	66.16
N=400			97.82	79.3

Table III

Stochastic vs. Deterministic Estimator in an S-model automaton with α=0.35, in a 10-Action Environment with σ^2=0.35.

	USING STOCHASTIC ESTIMATOR		USING DETERMINISTIC ESTIMATOR ONLY	
	Accuracy	Speed	Accuracy	Speed
N=25	88.44	30.44	85.4	8.54
N=100	97.63	68.59	93.23	25.86
N=200	98.76	91.49	94.97	55.37
N=300	99.09	115.09	96.29	81.42
N=400	99.24	138.84	96.99	97.42
N=500			97.48	136.3

Table IV

Stochastic vs. Deterministic Estimator in a P-model automaton with a=0.65, in a 10-Action Environment.

	USING STOCHASTIC ESTIMATOR		USING DETERMINISTIC ESTIMATOR ONLY	
	Accuracy	Speed	Accuracy	Speed
N=25	90.55	35.78	85.4	8.54
N=100	98.74	66.66	93.23	25.86
N=200	99.14	94.53	94.97	55.37
N=300	99.35	117.3	96.29	81.42
N=400	99.45	141.5	96.99	97.42
N=500			97.48	136.3

Table V

Stochastic vs. Deterministic Estimator in a P-model automaton with α=0.75, in a 10-Action Environment.

	USING STOCHASTIC ESTIMATOR		USING DETERMINISTIC ESTIMATOR ONLY	
	Accuracy	Speed	Accuracy	Speed
N=25	92.22	40.93	85.4	8.54
N=100	99.08	70.61	93.23	25.86
N=200	99.38	97.16	94.97	55.37
N=300	99.51	121.3	96.29	81.42
N=400	99.59	144.2	96.99	97.42
N=500			97.48	136.3

Table VI

Stochastic vs. Deterministic Estimator in a P-model automaton with α=0.85, in a 10-Action Environment.

FIGURES

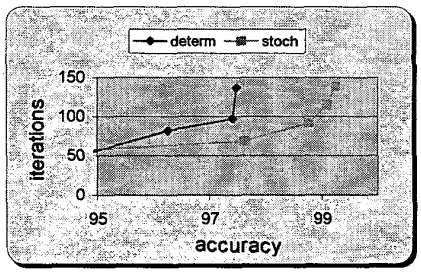

Fig.1 Iterations vs. accuracy in a P-model automaton with a=0,65.

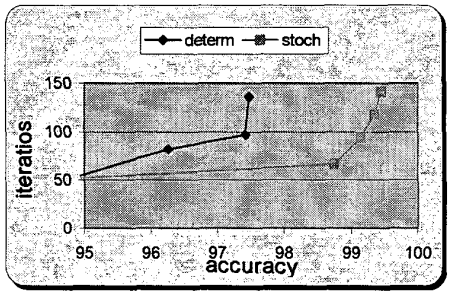

Fig.2 Iterations vs. accuracy in a P-model automaton with α=0.75.

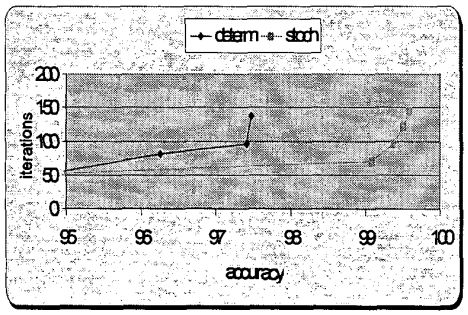

Fig.3 Iterations vs. accuracy in a P-model automaton with α=0.85.

a

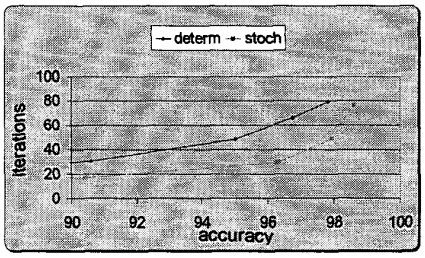

Fig.4 Iterations vs. accuracy in an S-model automaton with σ^2=0.35 and α=0.35.

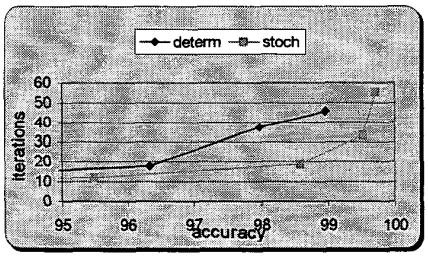

Fig.5 Iterations vs. accuracy in an S-model automaton with σ^2=0.25 and α=0.25.

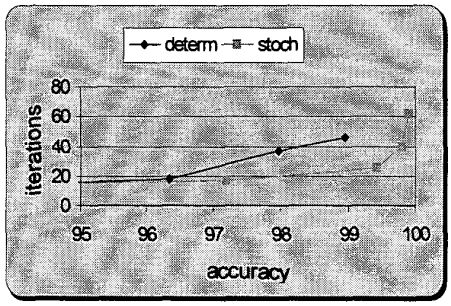

Fig.6 Iterations vs. accuracy in an S-model automaton with σ^2=0.25 and α=0.35.