Vol. 02, No. 2 ,(2018), 067–074 , **ISSN:2581-3242**

IJMLNCE
JOURNAL

International Journal of Machine Learning and Networked Collaborative Engineering

Journal Homepage: http://www.mlnce.net/home/index.html

DOI : https://doi.org/10.30991/IJMLNCE.2018v02i02.004

Improvement of Automated Learning Methods based on Linear Learning Algorithms

Farzad Kiani

Department of Computer Engineering, Engineering and Natural Sciences Faculty,

Istanbul Sabahattin Zaim University, 34303, Kucukcekmece, Istanbul/Turkey

Farzad.kiani@izu.edu.tr, https://orcid.org/0000-0002-0354-9344

Abstract

In recent years, the learning methods are converted to one of the new research area. These researches are divided into two general categories. The first category recognizes the principles of learning the living entities and its stages. The second is learning based methodology to any machines that the proposed method of this paper is based on it. Learning is defined as changes made in the performance of a system based on experiences. An important feature of learning systems is the ability to improve their efficiency over time. In mathematical terms, it can be stated that the purpose of a learning system is to optimize a task that is not well-known. Therefore, an approach to this problem is to reduce the goals of the learning system to an optimization problem. So, it is defined on a set of parameters and its purpose is to find the optimal set of parameters. In many of the issues raised, there is no knowledge of the correct answers to the problem in supervised learning based methods especially. For this reason, the use of a learning method called reinforcement learning has been considered. The main advantage of this technique over other learning methods is the need for no information from the environment (except amplification signal). The other learning methods as supervised or unsupervised are not appropriate to these problems. In this method, each agent decides the next its actions based on current k-actions instead of one action. In this paper is proposed a new approach based on the reinforcement learning technique that has three versions in order to implementation in different areas. It behaviors based on reward and penalty model. The effectiveness of these interactions with the environment is evaluated by the maximum and minimum of the number of rewards and penalties that are taken from the environment. The three versions are simple, sequential and unstructured linear learning methods so they evaluated in different possibilities to get the appropriate responses. Depending on the needs of any system, they can be used. The mode of convergence of actions in the proposed automaton (machine) in six different scenarios is examined.

Keywords

Automated learning,

linear learning,

smart systems,

reinforcement learning

1 Introduction

In recent years, the process of learning creatures is converted to one of the new research area [1]. These researches are divided into two general categories. The first category introduces learning the live stage presence and recognizes their principles. Aim of the second group is provide a methodology for placing these principles in a machine. Learning is defined as changes made in the performance of a system based on experiences [2]. An important feature of learning systems is the ability to improve their efficiency over time. The most prominent features of learning-based systems are that they improve themselves over time. In mathematical terms, it can be stated that the purpose of a learning system is to optimize a task that is not well known [3, 4]. Therefore, an approach to this problem is to reduce the goals of the learning system to an optimization problem, which is defined on a set of parameters and aims to find a set of optimal (appropriate) parameters. In many of the issues raised, there is no knowledge of the correct answers to the problem, which is required by supervised supervision. For this reason, the use of a learning method called Reinforcement Learning (RL) has been considered. This category of learning method is an orthogonal approach to solving different and more difficult problems. It uses a combination of dynamic programming and supervisory learning to achieve a powerful machine learning system. In the RL is defined a goal for the learning agent so the agent must be achieve it. Consequently, the agent learns how to achieve the target by different tests in various environment [5, 6]. In the RL, a learner's factor in learning through repeated interactions with the environment leads to an optimal control policy. The effectiveness of these interactions with the environment is evaluated by the maximum (minimum) of the number of rewards (penalty) taken from the environment. The main advantage of the RL over other learning methods is the need for no information from the environment (except amplification signal) [2].

The first attempts to use the learning automata in control applications were carried out by Fu et al [7]. In these years, this field has been very intensive and relevant studies. Among these research studies, we can mention the uses of learning automata in the estimation of parameters, pattern recognition, linear update methods, game theory [8, 9], McLaren [10], etc. One of the learning based approaches that can be useful in different environments is Bayesian networks. Bayesian methods provide a standard for optimal decision-making, although in many cases it is incalculable. The [11] proposes A novel decentralized decision making scheme was proposed in [11] that it is based on the Goore Game. The nature of each decision-maker is naturally Bayesian and avoids the difficulty of updating hyperactive parameters of the sibling conjugate primers and calculating them from random sampling from these posterior. Learning-based systems have become very popular and are nowadays used in many business and scientific fields. On the other hand, they have the ability to integrate with todays technologies and research area. On many occasions, the combination of highly valuable products and research is also achieved as multidisciplinary.

A learning automaton consists of two main parts. One of them is stochastic learning automaton that has limited actions number and a random environment in which the automaton is associated with it. The other is the learning algorithm that the automata learns the optimal operation using it. A random automaton is defined as $SA \equiv \{\alpha, \beta, F, G, \phi\}$. Where, α shows set of automata actions. α is defined from index one to r, so, r is number of automatic actions. $\alpha \equiv \{\alpha_1, \alpha_2, ..., \alpha_r\}$. β is input sets of automata that its domain is assumed as m so $\beta \equiv \{\beta_1, \beta_2, ..., \beta_m\}$. F is new status generator function that is defined as $F \equiv \phi \times \beta \rightarrow \phi$. G is output function that the output function that maps the current state to the next output ($G \equiv \phi \rightarrow \alpha$). ϕ presents the input states set of the automata is at the n moment as $\phi(n) \equiv \{\phi_1, \phi_2, ..., \phi_k\}$. The set α includes automated outputs (actions), in which the automation in each step chooses an operation of r for this set to apply to the environment. If the mapping F and G are definite, the automata is called deterministic automata. When the F and G maps are random, the automata is called non-deterministic automata.

Learning automata are divided into two groups of fixed and variable structure automata. In stochastic automata with a fixed structure, the probabilities of automated operations are constant. While in stochastic automata with a variable structure, the probabilities of automated operations are updated in each repetition. In this structure, changing the likelihood of actions is done based on the learning algorithm and the internal state of the automata is represented by the probabilities of the operation of the automata. In fact, each automaton machine has some states as input and output that they are equivalent in almost times. The action probability vector of the operation that is defined at the follow equation defines the internal state of the automaton at the instant n.

$$P(n) = \{p_1(n), p_2(n), ..., p_r(n)\} \tag{1}$$

So that, at the beginning of the activity of the automaton, the probability of its operation is equal and equal (r is the number of automatic operations).

$$\sum_{i=1}^{r} p_i(n) = 1, \ \forall n, \ p_i(n) = Prob[\alpha(n) = \alpha_i] \qquad (2)$$

The environment can be represented as $E \equiv \{\alpha, \beta, c\}$. Where, α shows the input sets of environment, β presents the output sets of environment, and c introduces set of penalty probabilities. $\alpha=\{\alpha_1, \alpha_2, \dots, \alpha_r\}$, $\beta=\{\beta_1, \beta_2, \dots, \beta_m\}$, $c=\{c_1, c_2, \dots, c_r\}$.

The input of the environment is one of the r automata actions. β specifies the output (response) of the environment to each action. For example, the system is called P-Model system if the response of the β is binary. In such an environment, $\beta_i(n)=1$ is as an unfavorable or failure response. The favorable or successful answer is when the $\beta_i(n)$ is zero. Other environment model is Q-Model. In this model, $\beta_i(n)$ contains a limited number of values in the interval [0, 1]. In addition, in S-Model, $\beta_i(n)$ is a random variable in the interval [0, 1]. As mentioned above, c specifies the probabilities of penalty (failures) of environmental responses and is defined as equation 3.

$$C_i = Prob\{\beta(n) = 1 \mid \alpha(n) = \alpha_i\}, \ \ i = \{1, 2, \dots, r\} \qquad (3)$$

It shows that α_i may receive an undesirable response from the environment. The values of α_i are unspecified and is assumed the C_i have at least one unique value. The same way, the environment can be represented by the set of reward probabilities (success). So, it is shown by d_i. The d_i indicates the probability of receiving the desired response to the action of α_i. In static environments, the probability of penalty of α_i are constant. While in non-stationary environments, the probabilities of fines change over time. The connection of random automata with the environment is shown in figure 1.

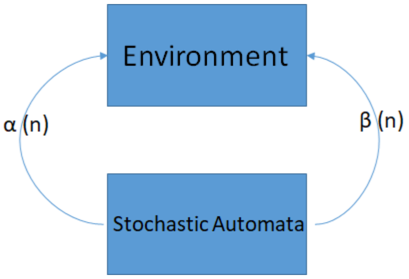

Fig. 1: SLA Schema

The figure 1 refers a learning algorithm that is called the Stochastic Learning Automata (SLA) [10]. Similarly, SLA can be defined as LA={α,β, p, T}. Where, α is defined from index one to r, so, r is number of automatic actions. $\alpha=\{\alpha_1, \alpha_2, \dots, \alpha_r\}$. β is input sets of automata that its domain is assumed as r so $\beta=\{\beta_1, \beta_2, \dots, \beta_r\}$. P shows action probability vector and is represented as $p=\{p_1, p_2, \dots, p_r\}$. In this case, our learning algorithm is:

$$T = p(n+1) = T[\alpha(n), \beta(n), p(n)] \qquad (4)$$

If T is a linear operator, the reinforcement learning algorithm is called linear. Otherwise, it is called Non-linear. If the learning automata in repeat n, chooses a sample action such α_i and receive an appropriate response from the environment, the likelihood of α_i action increases and the likelihood of other actions decreases. Conversely, if the response of the environment is undesirable, the likelihood of action of α_i decreases and the probability of other automated actions increases. We can increase the performance of the system by selecting more than one action at the same time while in the previous methods was selected a single action at each stage.

The learning based machines can be applied on the various application area such as smart systems, wireless ad-hoc networks, pervasive systems, IoT, emotional analysis systems etc. [12-14]. In the literature, many learning based techniques are applied on the above applications especially on wireless sensor networks. One the popular used models is cellular automata. It was introduced by Von Neumann at 1940 and after that, he proposed a mathematician named Olam Modelli to study the complex systems behavior. The cellular automata are, in fact, discrete dynamic systems whose behavior is based entirely on local communication. From the point of view of pure mathematics, they can be considered as a branch of topological dynamics and from the point of view of electrical engineering, iterative arrays. In cellular automation, space is defined as a network that is called a cell to each house. Time runs out discretely and its rules are global. Through each cell, each cell acquires its new status by considering its neighbors. Cellular automata can also be considered as computational systems that process the information they encode on their own. In addition, a cellular automaton with its control unit can be interpreted as a SIMD machine. The rules of cellular automation explains how to affect the cell's adherence to its neighboring cells. We call a cell is a neighbor of another cell, if it can affect it in one-step and in accordance with the rules. The rules are generally divided into three categories in this automaton. Firstly is general rules; in this rule, the amount of a cell in the next stage depends on the amount of single neighboring cells in the current state. In the second rule (totalistic Rules), the amount of a cell in the next step depends on the number of neighboring cells that are in different states. In this type of rule, unlike general rule, attention is not paid to single cells. In the last rule (outer totalistic rules), determining the next state of the cell is also effective in the current state.

In the section 2 of the paper is introduced a multi actions and learning based machine that has three versions. So, the evaluation results of them are discussed in the section 3. Finally, the conclusion of the paper is explained in the section 4.

2 Proposing Linear Learning Algorithms

In the standard learning automaton, an action is selected at each step. Then, the probability vector of the actions is updated with considering the appropriate or unfavorable environmental responses. If the appropriate or inappropriate environmental response is not based on the effect of an action, gives unsuccessful results. In this paper is proposed a new type of learning automaton that in this type of machine, instead of choosing one action, k-actions are chosen and then the environment response is received. It is an approach based on reward and penalty system so it is based on reinforcement learning and game theory. The environmental response can be determined by different methods. In the majority voting technique, the optimal appropriate environment response is appropriate when the effect of k/(2 + 1) of the selected action in the environment has the same environment response. Otherwise, the environmental response is inappropriate. All selected actions will be penalized if the response is inappropriate. If the response is desirable, the actions whose effects on the environment have the same environmental response (effective actions in giving the desired environment response) are rewarded and other actions are penalized. The initialized parameters of proposed machine is defined as SLA and is based on Eq. 4.

In this machine, in each step k actions are selected and applied in the environment. The acceptable answer sets of automata to choice of h^th^ action in step of s is calculated based on Equation 5.

$$\alpha^h(s) \sqsubseteq \alpha, r_s^h \leq r \;\; \alpha^h(s) = \left\{\alpha 1, \ldots, \alpha_{r_s^h}\right\}, \quad \alpha^1(s) = \alpha \tag{5}$$

In this status, the h^th^ selected answer is deleted from the next actions of the related answer sets. In this machine, three types of ***linear learning algorithms*** are suggested for the automata.

The <u>*first learning algorithm*</u>, which we call the simple linear learning algorithm and is calculated by Equations 6 and 7. So, The Eq. 6 is used for finding appropriate answers and the Eq. 7 is applied to find the inappropriate answers. In here is assumed the h^th^ selected action in n^th^ step is action α_i.

$$p_i^{h+1}(n) = p_i^h(n) + \alpha[1 - p_i^h(n)] \quad and \quad p_j^{h+1}(n) = (1-\alpha)p_j^h(n) \tag{6}$$

$$p_i^{h+1}(n) = (1-b)p_i^h(n) \quad and \quad p_j^{h+1}(n) = \left(\frac{b}{r-1}\right) + (1-b)p_j^h(n), \forall j \;\; j \neq i \tag{7}$$

The relationship between these two equations with the probability vector is expressed by equation 8.

$$P_i^1(n) = p_i(n) \quad \forall i \quad such\ that\ p_i \in p(n)$$

$$p_i(n+1) = p_i^{k+1}(n) \quad \forall i \quad such\ that\ p_i \in p(n+1) \tag{8}$$

As with the relationships of the first learning algorithm, in this type of learning algorithm, at each stage, all probability vector members are updated to a number of k times.

The *second learning algorithm* is called a sequential linear learning algorithm and is formulized by Equations 9 and 10. So, The Eq. 9 is used for finding appropriate answers and the Eq. 10 is applied to find the inappropriate answers. In here is assumed the hth selected action in nth step is action α_i.

$$\hat{p}_i^{h+1}(n) = \hat{p}_i^{h}(n) + \alpha\left[1 - \hat{p}_i^{h}(n)\right] \quad and \quad \hat{p}_j^{h+1}(n) = (1-\alpha)\hat{p}_j^{h}(n) \quad \forall j \;\; j \neq i \tag{9}$$

$$\hat{p}_i^{h+1}(n) = (1-b)\hat{p}_i^{h}(n) \quad and \quad \hat{p}_j^{h+1}(n) = \left(\frac{b}{r-1}\right) + (1-b)\hat{p}_j^{h}(n) \quad \forall j \;\; j \neq i \tag{10}$$

The value of $\hat{p}$ is calculated by Equation 11.

$$\hat{p}_i^{h}(s) = \frac{p_i^h(s)}{M} \quad \forall i \;\; such\ that \;\; \alpha_i \in \alpha^h(s), \quad M = \sum_{j=1}^{r_s^h} p_j^h(s) \quad \forall j \;\; such\ that \;\; \alpha_j \in \alpha^h(s) \tag{11}$$

In the Eq. 9 and 10, α presents reward parameter and b shows the penalty parameter. The probability vector of the automata is updated according to the Equation 12 in order to return the machine to normal conditions and select the (h+1)th action.

$$p_i^{h+1}(s) = \hat{p}_i^{h+1}(s).M \quad \forall i \;\; such\ that \;\; \alpha_i \in \alpha^h(s)$$

$$p_i^{h+1}(s) = p_i^{h}(s) \quad \forall i \;\; such\ that \;\; \alpha_i \notin \alpha^h(s) \tag{12}$$

As it is known from the relations of the second learning algorithm, at each step of in this type, the probability of the action of the hth selected action will be updated for h times, and actions that are not selected are updated to k times.

The *third learning algorithm* is called unstructured linear learning algorithm and is formulized by Equations 13 and 14. So, The Eq. 13 is used for finding appropriate answers and the Eq. 14 is applied to find the inappropriate answers. In here is assumed the hth selected action in nth step is action α_i.

$$\hat{p}_i^{h+1}(n) = \hat{p}_i^{h}(n) + \alpha\left[1 - \hat{p}_i^{h}(n)\right] \quad and \quad \hat{p}_j^{h+1}(n) = (1-\alpha)\hat{p}_j^{h}(n) \quad \forall j \;\; j \neq i \tag{13}$$

$$\hat{p}_i^{h+1}(n) = (1-b)\hat{p}_i^{h}(n) \quad and \quad \hat{p}_j^{h+1}(n) = \left(\frac{b}{r_s^h-1}\right) + (1-b)\hat{p}_j^{h}(n) \quad \forall j \;\; j \neq i \tag{14}$$

The value of $\hat{p}$ is calculated by Equation 15. Where, $\alpha_k(s)$ presents k actions selected in s step. In addition, α presents reward parameter and b shows the penalty parameter.

$$\hat{p}_i^{h}(s) = \frac{p_i^h(s)}{M} \quad \forall i \;\; such\ that \;\; \alpha_i \in \{\alpha - \alpha_k(s) + \alpha_i\}, \quad M = \sum_{j=1}^{r_s^h} p_j^h(s) \quad \forall j \;\; such\ that \;\; \alpha_j \in \{\alpha - \alpha_k(s) + \alpha_i\} \tag{15}$$

The probability vector of the automata is updated according to the Equation 16 in order to return the machine to normal conditions and select the (h+1)th action.

$$p_i^{h+1}(s) = \hat{p}_i^{h+1}(s).M \quad \forall i \;\; such\ that \;\; \alpha_i \in \{\alpha - \alpha_k(s) + \alpha_i\}$$

$$p_i^{h+1}(s) = p_i^{h}(s) \quad \forall i \;\; such\ that \;\; \alpha_i \notin \{\alpha - \alpha_k(s) + \alpha_i\} \tag{16}$$

As it is known from the relations of the third learning algorithm, at each step of in this type, the probability of the selected actions will be updated for one time, and actions that are not selected are updated to k times.

3 Evaluation

With a sample of automata, we examine the mode of convergence of operations in this type of automaton (machine) in MATLAB software simulation. This sample machine has 6 actions, and each step is selected 3 steps. By responding to the environmental responses (responses from the effects of each action on the environment), the desired response is detected. Given the appropriate response, the related actions are rewarded and penalized. The probability of receiving the appropriate response from the environment for each operation is shown in Table 1.

Table 1. Possibility to receive the optimal appropriate responses from the environment for each of the actions in the proposed learning machine.

Action #	1	2	3	4	5	6
Possibility to get the appropriate response	0.1	0.9	0.1	0.7	0.7	0.5

The reward and penalty parameters are set at 0.01. The mode of convergence of automated actions for different learning algorithms is seen in Figures 2,3 and 4.

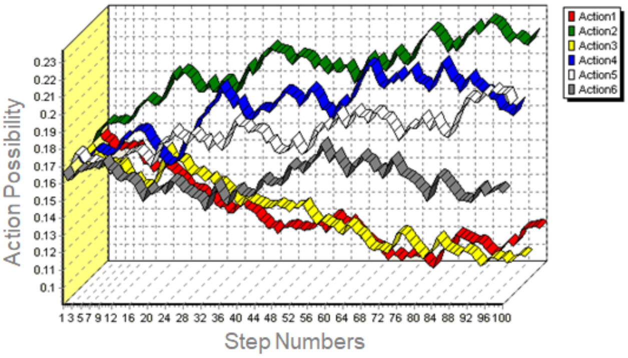

Fig. 2: Results of convergence of automated actions in simple linear learning algorithm.

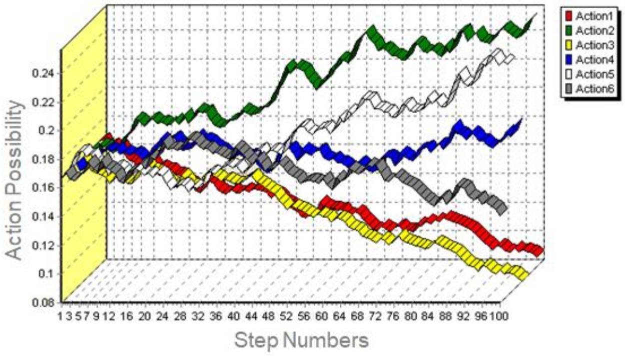

Fig. 3: Results of convergence of automated actions in sequential linear learning algorithm.

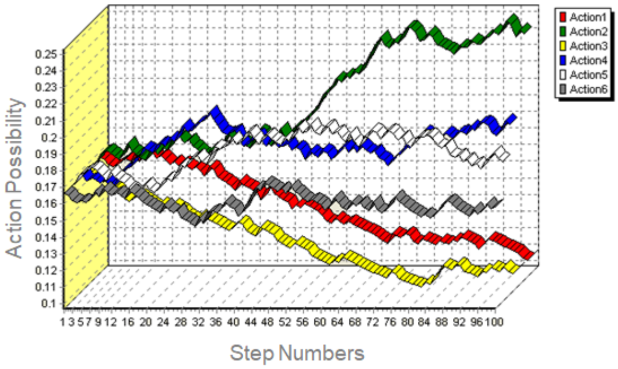

Fig. 4: Results of convergence of automated actions in unstructured linear learning algorithm.

It is possible to use this method in various fields such as wireless ad-hoc networks and IoT based on the results. Each proposed algorithm can be useful in the related environment application based on system requirements. In additional, this method can be used as a mixed approach in different learning based algorithms as Bayesian networks, game theory techniques, Q-learning, Sarsa, neural network and deep learning.

4 Conclusion and Future Works

In this paper was proposed a new model of learning automaton machine that it worked based on K-actions contrary to existing methods in the literature. It affects the response of the environment based on selecting and analyzing the k-actions. In this paper was introduced three different linear learning methods as simple, sequential and unstructured. The Analysis results on the six samples of actions, which the k was supposed 1 to 6, are shown the each method’s performance. They can be used in various application areas that we will apply it on the wireless Ad-hoc networks and IoT research area. The third method seems to be high performance on wireless networks that it is one of the future works. It is possible to use this method in various fields such as wireless ad-hoc networks and IoT based on the results. Each proposed algorithm can be useful in the related environment application based on system requirements. In additional, this method can be used as a mixed approach in different learning based algorithms as Bayesian networks, game theory techniques, Q-learning, Sarsa, neural network and deep learning. In future work, we have a plan in research study on wireless sensor networks based on the proposed algorithms. We will analyze the simulation results and will compare them with the relevant methods in the literature.

References

[1]. V. Saritha, P. V. Krishna, S. Misra and M.S. Obaidat, “Learning Automata based Optimized Multipath routing using Leapfrog Algorithm for VANETs”, IEEE ICC 2017 Mobile and Wireless Networking, 1-5, 2017. https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7997401

[2]. K. S. Narendra and M. A. L. Thathachar, “Learning automata: An introduction”, *Proceedings of the Prentice Hall*, 1989. https://dl.acm.org/citation.cfm?id=64802

[3]. Z. Shariyat, A. Movaghar, M. Hoseinzadeh, “A learning automata and clustering-based routing protocol for named data networking”, *Telecommunications Systems*, 65(1), 9-29, 2017. https://link.springer.com/article/10.1007/s11235-016-0209-8

[4]. H. Ge and Sh. Li, “A Parameter-Free Learning Automaton Scheme”, *Cornell University Library*, 1-13, 2017. https://arxiv.org/pdf/1711.10111.pdf

[5]. E. Mance and S. H. Stephanie, “Reinforcement learning: A tutorial”, *Proceedings of the Wright Laboratory*, 1996. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.33.2480&rep=rep1&type=pdf

[6]. K. Arulkuraman, M. Peter, M. Brundage and A. Bharath, “Deep Reinforcement Learning: A brief survey”, *IEEE Signal Processing Magazine*, 26-38, 2017. https://ieeexplore.ieee.org/document/8103164/

[7]. K. S. Fu and T. J. Li, “Formulation of learning automata and automata games”, Proceedings of the Information Science, 1(3), 237-256, 1969. https://www.sciencedirect.com/science/article/pii/S0020025569800101

[8]. C. S. Chasparis, “Stochastic Stability of Perturbed Learning Automata in Positive-Utility Games”, 1-16, 2018, https://arxiv.org/pdf/1709.05859.pdf

[9]. A. Jitpattanakul, “Learning k-edge Deterministic Finite Automata in the Framework of Active Learning”, *International Journal of Applied Engineering Research*, *International Journal of Applied Engineering Research*, 12(6), 6050-6054, 2017. https://pdfs.semanticscholar.org/ef37/f0bf558148dd2b830abdd4dfbb406b50e2f7.pdf

[10]. R. W. McLaren, “A stochastic automaton model for synthesis of learning systems”, Proceedings of the IEEE Transactions on System Science and Cybernetics, 2, 109-114, 1966. https://link.springer.com/chapter/10.1007/978-1-4615-9050-7_5

[11]. O. Christoffer, S. Glimsdal, “Accelerated Bayesian learning for decentralized two-armed bandit based decision making with applications to the Goore Game”, *Appl Intell, Springer Science and Business Media, LLC 2012,* 1-10, 2012. https://brage.bibsys.no/xmlui/bitstream/handle/11250/137969/Granmo_2012_Accelerated.pdf?sequence=1

[12]. S. Tanwer et al., “LA-MHR: Learning Automata Based Multilevel Heterogeneous Routing for Opportunistic Shared Spectrum Access to Enhance Lifetime of WSN”, *IEEE Systems Journal*, 1-11, 2018. https://ieeexplore.ieee.org/document/8351993/

[13]. D. Mendez, I. Papapanagiotou, B. Yang, “Internet of Things: Survey on Security and Privacy”, 1-15, 2017, https://arxiv.org/pdf/1707.01879.pdf

[14]. F. Kiani, “Reinforcement Learning Based Routing Protocol for Wireless Body Sensor Networks”, International Symposium on Cloud and Service Computing (SC2), 2017 IEEE 7th, 71-78, 2017. https://ieeexplore.ieee.org/abstract/document/8315358/

Author’s Biography

Farzad Kiani: Farzad Kiani has PhD degree and he is Assistant Prof. in computer engineering dept. at Istanbul Sabahattin Zaim University (IZU) from 2014 to now. In addition, he is deputy head of computer engineering and head of wireless sensor networks and IoT Labs at IZU from 2016 and 2017 to now, respectively. His current research interests include wireless sensor networks, Vehicle Ad-hoc networks, flying Ad-hoc networks, machine learning, game theory and IoT. He is also working on optimization algorithms and security.

How to Cite

Kiani, Farzad, “Improvement of Automated Learning Methods based on Linear Learning Algorithms*”, International Journal of Machine Learning and Networked Collaborative Engineering,* Vol. 02 No. 02, 2018, pp. 67-74. doi: https://doi.org/10.30991/IJMLNCE.2018v02i02.004.