Journal of AI and Data Mining
Vol 8, No 4, 2020, 491-514. *Original Research Paper* DOI: 10.22044/JADM.2020.8928.2029

Optimizing Membership Functions using Learning Automata for Fuzzy Association Rule Mining

Z. Anari[1], A. Hatamlou [2*] B. Anari[3] and M. Masdari[4]

1. Department of Computer Engineering and Information Technology, Payame Noor University (PNU), P. OBox,19395-4697 Tehran, Iran.
2. Department of Computer Engineering, Khoy Branch, Islamic Azad University, Khoy, Iran.
3. Department of Computer Engineering, Shabestar Branch, Islamic Azad University, Shabestar, Iran.
4. Department of Computer Engineering, Urmia Branch, Islamic Azad University, Urmia, Iran.

Received 30 September 2019; Revised 25 March 2020; Accepted 06 June 2020
**Corresponding author: rezahatamloo@gmail.com (A. Hatamlou).*

Abstract

The transactions in web data often consist of quantitative data, suggesting that the fuzzy set theory can be used to represent such data. The time spent by the users on each web page is one type of web data, regarded as a trapezoidal membership function (TMF), and can be used to evaluate the user browsing behavior. Quality of the mining fuzzy association rules depends on the membership functions, and since the membership functions of each web page are different from those for the other web pages, the automatic finding of the number and position of TMF is significant. In this paper, a different reinforcement-based optimization approach called LA-OMF is proposed to find both the number and position of TMFs for the fuzzy association rules. In the proposed algorithm, the centers and spreads of TMFs are considered as the parameters of the search space, and a new representation using learning automata (LA) is proposed to optimize these parameters. The performance of the proposed approach is evaluated, and the results obtained are compared with the results of the other algorithms on a real dataset. Experiments on the datasets with different sizes confirm that the proposed LA-OMF approach improves the efficiency of mining fuzzy association rules by extracting the optimized membership functions.

Keywords: *Web Usage Mining, Learning Automata, Fuzzy Set, Membership Function, Fuzzy Association Rule.*

1. Introduction

With increase in the growth of data over the internet, discovering and extracting the useful information from the vast amount of data on the Web is difficult and time-consuming [1]. Thus, the effective techniques are required to discover the relevant information from the Web data. Web mining plays a significant role in discovering such data [2]. Web mining is the application of the data mining methodologies and techniques to find and extract useful information from the Web data [3]. Web mining can be divided into three classes, namely Web usage mining, Web content mining, and Web structure mining. Web usage mining utilizes the data mining techniques for discovering meaningful patterns from the Web access logs as the user's interaction with the Web sites. Web content mining is the process of discovering useful information from the contents of Web documents and services. Web structure mining is used to discover the knowledge from the Web sites and link structures of the Web in order to categorize the Web pages. The knowledge extracted from Web data can be used in many fields such as the Web personalization [4], recommender systems [5, 6], clustering [8, 68], sequential pattern mining [7], and association rule mining [8, 9]. Extracting the association rules from the Web usage data is one of the important data mining techniques, which can be used for the Web log records [7-10].

The association rule mining is one of the important techniques used in the pattern discovery. An association rule is described as $X \Rightarrow Y$, where X and Y are the itemsets and

$X \cap Y = \varnothing$. It means that the transactions that include the itemsets in X may also include the itemsets in Y [11, 12]. The association rule mining is used in the Web usage data to find groups of pages that are frequently accessed in the user's sessions. For example, if a transaction contains a set of Web pages $A.html$, $B.html$, and $C.html$, then an extracted rules such as $A.html, B.html \Rightarrow C.html$ means that if the user observes the $A.html$ and $B.html$ pages, he is most likely to meet the page $C.html$ at the same session.

The simplicity of using the fuzzy sets and their similarity to the human inference make them easy to use in a variety of fields such as the sequential pattern mining, rule extraction from datasets, biological knowledge elicitation, intrusion detection, Ad-hoc Networks, engineering and diagnosis [13-20, 69, 71, 72]. The Web datasets often consist of quantitative, vague, and linguistic datasets. Therefore, fuzzy logic can be used in several Web mining algorithms such as [21-24].

Many fuzzy web mining algorithms have been proposed to extract the association rules. Most of the fuzzy Web mining algorithms proposed to show the importance of each Web page and extract rules use the time duration spent by the users on each web page as a fuzzy linguistic variable [21, 25-29]. The time duration of each Web page is one of the most important parameters that is saved in Web logs and can be used as an important factor for analyzing the user's browsing behavior. The time duration of each Web page can be considered as a TMF.

The main problem in all the previous studies conducted on fuzzy Web mining algorithms that use the time duration parameter as TMF is that the number of membership functions and their parameters for each linguistic variable are specified by the experts' ideas. In other words, they assume that the number and parameters of the membership functions are fixed and pre-defined. If this is the case, some mining results may not be suitable in some real-world applications. Moreover, different membership functions may infer different knowledges. Therefore, finding the suitable membership functions to extract fuzzy associations is a significant task. The reason is two-fold: (a) it is not easy to have a prior knowledge in obtaining the suitable membership functions that cover all the domain variables, and (b) data that is appropriate for one website may not be appropriate for another one.

In this paper, we address the above issue, and propose a new algorithm called learning automata-based algorithm for finding the optimized membership functions (LA-OMF). To the best of our knowledge, this is the first attempt that simultaneously finds the optimal number and parameters of the trapezoidal membership functions (TMFs) for mining fuzzy association rules.

The proposed LA-OMF algorithm takes a number of TMFs as the inputs and uses a team of LA to optimize its parameters. Learning automata (LA) is a mathematical model aiming to optimize a function. In this model, each learning automaton attempts to find its optimal action. In order to do this, each automaton selects an action using its action-set, and examines the environmental response by applying it to a random environment. Depending on the environmental response, the automaton corrects its behavior to select the next action. Accordingly, each LA learns how to find its optimal action [30-32]. The proposed LA-OMF algorithm is composed of three main steps. In step 1, a new representation of LA for each TMF is performed. In step 2, we propose a new approach to reduce the search space and eliminate the inappropriate membership functions. Additionally, two constraints are used for their implementation. In step 3, the learning algorithm is used to evaluate the objective function. Finally, using the LA-OMF algorithm, we develop an algorithm to optimize the number and positions of the trapezoidal membership functions in fuzzy association rule mining.

The contributions of this research work can be briefly described as follow:

- We proposed a new algorithm called LA-OMF to optimized the TMF parameters. Then using the LA-OMF algorithm, another approach was developed to find both the number and positions of TMFs simultaneously.
- In the proposed approach, in order to improve the speed of convergence and reduce the search space, we proposed several constraints to eliminate the inappropriate membership functions during the learning process. Additionally, we developed a new approach to implement the constraints.
- We modified the objective function by considering TMFs.
- We validated the high efficiency of the proposed algorithm on fuzzy association rule mining for Web usage data based on the experiments performed on the datasets with different sizes.

The remaining sections of this paper are organized as what follows. Section 2 focuses on a review of the related works. Section 3 provides the necessary background. Section 4 introduces the proposed LA-OMF algorithm. Section 5 presents the dataset used and the experimental results. Finally, conclusions and directions for future works are described in Section 6.

2. Related Works

Optimization of the membership functions has a significant effect on the fuzzy association rule mining. Several fuzzy mining algorithms have been suggested for deriving both the appropriate membership functions and the fuzzy association rules from quantitative transactions in databases. These algorithms can dynamically tune membership functions by the meta-heuristic methods and use them to transform the quantitative transactions to fuzzy values.

Hong *et al*. [33] have proposed a fuzzy data mining approach using a genetic algorithm to simultaneously derive the suitable membership functions and the fuzzy association rules. Their approach was a population-based method containing a population of membership functions. Their algorithm dynamically extracted the membership functions. In the first step, each set of membership functions was transformed into a constant length string. Then in the second step, suitable strings were chosen to construct a proper membership function set under the evolution procedure. Chen *et al*. [34] have later enhanced the previous approach [33], and using the genetic algorithm, fuzzy theory and clustering concepts, presented a genetic algorithm-based fuzzy mining algorithm to find both the appropriate membership functions and the fuzzy association rules. In their algorithm, each chromosome showed a set of membership functions. At first, each set of membership functions was transformed into a string with a fixed length. Afterwards, the k-means clustering algorithm was used to divide the chromosomes into a population consisting of k-clusters. Finally, the best set of membership functions was obtained by evaluating the fitness value for each cluster and applying the genetic operator.

Moreover, Alcala-fdez [35] has presented a fuzzy mining algorithm using a genetic algorithm to derive the fuzzy association rules and membership functions from transactions in the log data. A model based on a genetic algorithm was used to construct the learning process of membership functions. In their algorithm, in order to reduce the search space and remove the unsuitable membership functions, a 2-tuple linguistic representation scheme was developed. Chen *et al*. [36] have introduced a fuzzy mining algorithm using a genetic algorithm to obtain the suitable membership functions for the data items and mining the concept-drift patterns. They used a 2-tuple representation scheme for linguistic terms to encode the membership functions of items into chromosomes. They claimed that their method could derive more concept-drift patterns compared to the other methods that used pre-defined membership functions.

Later, Chen *et al*. [37] have developed a fuzzy mining approach, presenting a novel chromosome representation to find a suitable number of membership functions and their optimized parameters. Each chromosome showed a set of membership functions and included two sections. The first section determined the activation of each membership function, and was represented by binary strings. The second section specified the related optimized parameters of the membership functions with real number schema. They obtained the best set of membership functions by applying the genetic operators and checking different states of active or inactive membership functions.

Palacios *et al*. [38] have presented a two-step approach to automatically find the suitable membership functions and fuzzy association rules from imprecise data. In the first step, they extracted the best set of membership functions using a genetic algorithm based on a 3-tuple representation scheme. In the second step, they applied a fuzzy frequent pattern-growth mining algorithm to extract the fuzzy association rules. They showed that their algorithm could reduce both the search space and the execution time.

Hong *et al*. [39] have proposed a new framework using the ant colony systems to discover the suitable membership functions and fuzzy association rules in data mining problems. In their approach, at first, the membership functions were represented as a binary code for each item. Then the initial graph was constructed by ants and the final membership function set was determined by moving ants in the graph. Finally, the best set of membership function set was used to derive the fuzzy association rules. Using an improved ant colony system, Wu *et al*. [40] have presented a fuzzy mining method to find the suitable membership functions from quantitative items in data mining problems. They designed an encoding representation, and by introducing certain operators, searched the actual global optimal solution in the continuous space. Their proposed approach did not have fixed edges and nodes in the search process. They dynamically generated search edges in the graph using the distribution of pheromones in the solution space. Thus, their method could obtain a superior nearly global optimal solution than the other ant-based fuzzy mining approaches.

Moreover, Ting *et al*. [41] have presented a memetic algorithm to find the appropriate membership functions for the fuzzy association rule mining. Their algorithm considered the structure type of membership functions for representation of chromosomes. They managed to remove the unsuitable membership functions by considering the structure type in their approach and thus reduced the search space. They also simplified the design of heuristics for a proper overlap and coverage. Moreover, using the structure type, they extended a local search operator to reduce the search space in finding the suitable membership functions.

Similar to [42], another approach has been proposed by Ting *et al*. using a genetic algorithm. In their approach, each triangle membership function was represented by three parameters. All the parameters were encoded as the chromosomes. Each parameter of the triangular membership functions corresponded to one vertex of a triangle. Then the best set of membership functions was extracted by applying the genetic operators. They developed two heuristics in the shape of membership functions. The heuristics eliminated the unsuitable shapes of the membership functions and also improved the search space to find the suitable membership functions.

Numerous algorithms have used multi-objective optimization in addition to single-objective optimization to optimize the membership functions in order to mine the fuzzy association rules. Radzinski *et al*. [43] have presented a multi-objective optimization approach for the fuzzy-rule-based classifier systems using the genetic algorithms. They used two benchmarks to evaluate the accuracy and interpretability. The benchmarks were considered as the objectives in the multi-objective genetic optimization methods. The learning process of their approach could determine both the parameters of the membership functions and the structure of the fuzzy rule-based systems. Using a multi-objective evolutionary learning model, Antonelli *et al*. [44] have proposed a learning scheme to extract the rules for the fuzzy rule-based classifiers. Their approach included two phases. The first phase was rule-based learning, and the second one was database learning. In the rule-based learning phase, the rules were learned, and in the database learning phase, the parameters of membership functions were learned from a set of fuzzy rules. Another approach was proposed by Minaei-Bidgoli [45] using a multi-objective genetic algorithm to mine the numerical association rules. They described three criteria, namely confidence, comprehensibility, and interestingness, as an objective of the multi-objective optimization approach. In their approach, one rule was used in each chromosome to represent the rules in the chromosome, and a concept of rough patterns was defined to represent a range or set of values.

In addition, Song *et al*. [46] have proposed a fuzzy decimal bat algorithm to mine the association rules so that they could dynamically find the membership functions from quantitative values. Their algorithm increased the local and global search capacity. Moreover, in order to evaluate the fuzzy set of membership functions, a new fitness function was proposed. The fitness function considered more factors; therefore, the number of association rules obtained was evaluated more accurately.

Chamazi *et al*. [47] have introduced a fuzzy temporal mining algorithm by combining a bee algorithm and a fuzzy temporal data mining approach. The purpose was to extract the appropriate membership functions and derive the fuzzy temporal association rules. Alikhodemi [48], using the fuzzy sets and the particle swarm optimization, presented a framework to extract the suitable membership functions to discover the fuzzy association rules, applying a master-slave parallel processing technique and a genetic algorithm. Hong *et al*. [49] have presented a genetic algorithm-based fuzzy mining algorithm to find the appropriate membership functions and fuzzy association rules. The master processor, like a genetic algorithm, divided the task of the fitness value among the slave processors. Each slave processor evaluated the fitness function allocated by the master processor and sent its own result to the master processor. Then the master processor evaluated the fitness results obtained from all the processors and selected the best fitness value as an appropriate membership function. Finally, the best set of membership functions obtained by the master processor was applied in the fuzzy data mining algorithm to extract the useful rules.

3. Preliminaries

In this section, in order to provide a background information for the remainder of the paper, we present a brief overview of the mining fuzzy association rules, learning automata, and objective function.

3.1. Mining Fuzzy Association Rules

The mining association rules is an important technique in data mining, utilized to generate interesting relationships between the data items in transaction databases [12]. This technique can also be applied in Web usage mining to find interesting patterns of Web pages visited by the users. The extracting information of such association rules can be applied to improve the structure of websites, predict the next pages to be visited by the users, rank the websites, and recommend Web pages.

A fuzzy aassociation rule in Web usage mining is defined as follows:

Let $P = \{p_1, p_2, \ldots, p_n\}$ represent a set of pages, and $S = \{s_1, s_2, \ldots, s_n\}$ represent a set of sessions in the Web browsing transaction records. Each session $s_i, (1 \le i \le n)$ contains a subset of the pages in P such that $S \subseteq P$. An association rule is represented as $X \rightarrow Y$, where $X \subseteq P$, $y \subseteq P$ and $X \cap Y = \emptyset$. Intuitively, a rule $X \rightarrow Y$ means that the sessions containing X tend to contain Y.

For example, the association rule $A.html, B.html \rightarrow C.html$ indicates the visitors who viewed the $A.html$ and $B.html$ pages; they would most probably view $C.html$. The purpose of the association rule mining is to extract all the association rules in a dataset of sessions with a support and confidence greater than or equal to the user-specified minimum support (called min support) and minimum confidence (called min confidence).

Definition 1. The support of X with respect to S is defined as the frequency of sessions $s \in S$ in the dataset that contain the itemset X.

$$\text{support}(x) = \frac{\left|\{s \in S; X \subseteq s\}\right|}{|S|} \tag{1}$$

Definition 2. The confidence of the association rule $X \rightarrow Y$ is described by the percentage of the sessions containing both X and Y, i.e.

$$Confidence(X \rightarrow Y) = \frac{sup(X \cup Y)}{sup(X)} \tag{2}$$

Agrawal *et al*. [50] have introduced an algorithm called the Apriori algorithm to mine the association rules in a set of transaction databases. The Apriori algorithm finds large item sets in a set of transactions *D*. Suppose that the minimum support is given; the Apriori algorithm attempts to find all the frequent item sets $L = \{L_1, \ldots, L_k\}$ from all the candidate item sets $C = \{C_1, \ldots, C_k\}$ whose support is greater than the user-specified mininmum support. Then the items in the frequent item sets are acquired and the minimum confidence constraints are used to form the association rules.

Definition 3. Fuzzy support is described as follows:

$$Fuzzy\ Support(\gamma_{j,k}) = \frac{\sum_{i=1}^{n} f_{j,k}^{i}}{n} \tag{3}$$

where $\gamma_{j,k}$ represents a fuzzy region of a page (denoted by $R_{j,k}$), $f_{j,k}^{i}$ shows the fuzzy membership value of $\gamma_{j,k}$ in the *i*-th session, and n is the number of sessions. If a fuzzy region $\gamma_{j,k}$ has a fuzzy support larger than the specified minimum support, like the Apriori algorithm, it is added to the large 1-sequences in L_1.

3.2. Overview of Learning Automata

Learning automata (LA) is a mathematical model used to optimize the functions. In this model, each learning automaton attempts to find its optimal action from the action-set using repetitive interaction with a random environment [3, 51]. In order to do this, each automaton selects an action using its action-set, and examines the environmental response by applying it to the random environment. Depending on the environment response, the automaton corrects its behavior to select the next action. Accordingly, each learning automaton learns how to find its optimal action [30, 32]. The interaction between the learning automaton and its random environment is presented in figure 1.

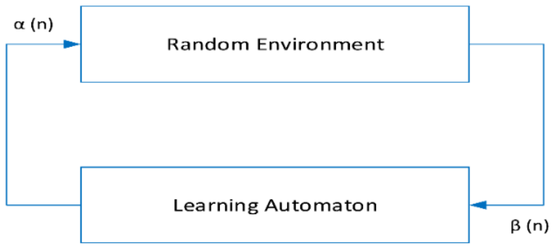

Figure 1. Interaction between the learning automata and the random environment.

In LA, the random environment is defined by the triple $< \alpha, \beta, c >$, where $\alpha = \{\alpha_1, \alpha_2, \ldots, \alpha_r\}$ shows the set of actions, $\beta = \{\beta_1, \beta_2, \ldots, \beta_m\}$ is the environment response (reinforcement signal) and $c = \{c_1, c_2, \ldots, c_r\}$ is a set of penalty probabilities; here, the penalty probability of each action α_i is represented by c_i. Each learning automaton such as LA_i at the instant n performs the following steps: 1) selects an action $\alpha_i(n) \in \alpha$ randomly based on the action probability distribution; 2) applies $\alpha_i(n)$ to the random environment and obtains the reinforcement signal $\beta(n)$; 3) uses $\beta(n)$ and a learning algorithm to check whether the selected action is to be rewarded or penalized; and 4) updates its action-set probability distribution based on the learning algorithm. These steps are repeated for each learning automaton to find the optimal action receiving the minimum penalty.

The learning automata can be classified into two main categories: fixed-structure learning automata (FSLA) and variable-structure learning automata (VSLA) [52-55]. In the following, VSLA, which is used in this paper, is described. VSLA is defined by $<\alpha, \beta, p, T>$, where $\alpha = \{\alpha_1, \alpha_2, \ldots, \alpha_r\}$ is the set of actions, $\beta = \{\beta_1, \beta_2, \ldots, \beta_m\}$ is the set of inputs, $p = \{p_1, p_2, \ldots, p_r\}$ is the action probability distribution over the action set, and T is the learning algorithm that updates the action probability vector of the automaton according to the response received from the environment. Also, r is the number of actions that can be chosen by the automaton. Let $\alpha_i(n) \epsilon \alpha$ and $p(n)$ denote the action selected by the learning automaton at time stamp t and the action probability vector over the action set, respectively. At each instant t, the action probability vector $p(n)$ is updated by the linear learning algorithm given in Equation (4) when the selected action $\alpha_i(n)$ is rewarded by the random environment, and it is updated as given in Equation (5) when the taken action is penalized. In Equations (4) and (5), $a \geq 0$ and $b \geq 0$ are called the reward and penalty parameters, which determine the amount of increases and decreases of the action probabilities, respectively.

$$p_j(n+1) = \begin{cases} p_j(n) + a\left[1 - p_j(n)\right] & j = i \\ (1-a)\, p_j(n) & \forall j \neq i \end{cases} \tag{4}$$

$$p_j(n+1) = \begin{cases} (1-b)\, p_j(n) & j = i \\ \left(\frac{b}{r-1}\right) + (1-b)\, p_j(n) & \forall j \neq i \end{cases} \tag{5}$$

For $a = b$, the recurrence Equations (1) and (2) are called linear reward-penalty (L_{R-P}) algorithm; when $a \gg b$, the given equations are called the linear reward-epsilon penalty ($L_{R-\epsilon p}$); and when $b = 0$, they are called linear reward-inaction (L_{R-I}). In the latter case, the action probability vector remains unchanged when the taken action is penalized by the environment.

LA has been used as the optimization tools in a variety of applications such as computer networks [56], image processing [57], fuzzy membership function optimization [58], speech recognition [59], signal analysis [60], information retrieval [61], cloud computing [62], dynamic optimization, [63], and data clustering [64], social networks[70].

3.3. The Objective Function

For the mining fuzzy association rules, in order to search for the optimized membership functions, an evaluation function, namely the objective function, was used to evaluate the quality of the membership function sets. In this work, we modified and used the objective function proposed by [65] in order to obtain a proper set of membership functions. The objective function was composed of the fuzzy support as well as the suitability. For each Web page, the objective function is defined as:

$$f = \frac{\sum_{X \in L_1} fuzzy\ support\left(\gamma_{j,k}\right)}{Suitability} \tag{6}$$

$$Objectve\ function = \frac{1}{f} \tag{7}$$

where, L_1 is the set of large 1-sequences obtained from the membership functions and *fuzzy support($\gamma_{j,k}$)* is the fuzzy support of the large 1-sequences from the given Web log records.

The suitability of the membership functions is described as the sum of the overlap and coverage factors. The suitability applied in the fitness function decreases the event of the two unsuitable types of the membership functions, as represented in figure 2, where the first one is too separate and the second one is too redundant. The suitability value of the membership functions of the Web page p_i is defined as [65]:

$$Suitability = Overlap\ factor + Covrage\ factor \tag{8}$$

where, the *overlap factor* is the overlap factor of the membership functions of the Web page p_i modified according to the trapezoidal membership function and defined as:

$$Overlap\ factor = \sum_{i<j}\left(\max\left(overlap\ ratio\left(R_i, R_j\right), 1\right) - 1\right) \tag{9}$$

with:

$$overlap\ ratio\left(R_i, R_j\right) = \frac{The\ area\ covered\ by\ both\ R_i\ and\ R_j}{\min\left(R_{i,4} - R_{i,3}, R_{j,2} - R_{j,1}\right)} \tag{10}$$

The *overlap ratio* of the two membership functions R_1 and R_2 is defined as the area covered by both membership functions divided by the minimum span of the right half of the left membership function and the left half of the right membership function. The overlap factor is used to avoid the excessive overlap of the membership functions and is non-negative with a best value of zero. For fuzzy web mining, we changed the coverage factor formula proposed by [65]. Therefore, the coverage factor of membership functions for each Web page p_i is defined as:

$$Coverage\ factor = \frac{\max(t_i)}{range(R_1, R_2, \ldots, R_m)} \quad (11)$$

where, the range (R_1, R_2, …, R_m) is the length of the horizontal axis covered by the membership functions of Web page p_i , $max(t_i)$ is the maximum browsing duration of p_i in the Web log dataset, and m is the number of fuzzy regions (membership functions) of p_i. The factor has the best value of 1 for a full coverage of the Web page's browsing duration.

Example 1: Suppose that the membership functions for the time duration of the Web page A are presented in figure 3.

Based on this figure, Web page A has three fuzzy regions: R_A^1 , R_A^2, and R_A^3 . In this figure, the overlap (R_A^1, R_A^2) = 22, the overlap (R_A^2, R_A^3) = 27, the overlap (R_A^1, R_A^3) = 0, the minimum spread (R_A^1, R_A^2) = min((34-0), (45-12)) = 33, the minimum spread (R_A^2, R_A^3) = min((82-45),(130-55)) = 37, and the minimum spread (R_A^1, R_A^3) = min((34-0),(130-55)) = 34. Therefore, the overlap factor of Web page A is calculated as:

overlap factor = $[\max((\frac{22}{33}, 1) - 1] + [\max\left(\left(\frac{22}{37}, 1\right) - 1\right] + [\max\left(\left(\frac{0}{34}, 1\right) - 1\right] = 0 + 0 + 0 = 0.$

In this example, the range (R_A^1, R_A^2, R_A^3) = 130 because the area covered by these three fuzzy regions is from 0 to 130. Suppose that the maximum time duration of Web page A in the sessions is 130, max (t_i) = 130; then the coverage factor of the membership functions of page A is computed as follows: coverage factor = $\frac{130}{130} = 1$. Therefore, in example 1, the suitability value of the membership functions of page A is computed as follows: Suitability = 0 +1 = 1. For the other web pages in the sessions, the suitability value of the membership functions was also calculated in the same way. Table 1 shows the time duration of all Web pages browsed by each user. Each row in the table is shown by Web page and time. Suppose that Web page A exists in five browsing sequences with time durations of 30, 8, 92, 20, and 110. The fuzzy support of each region (R_A^1, R_A^2, R_A^3) was calculated as follows:

Fuzzy Support $(\alpha_1) = \frac{1}{6} \times \left(\frac{34-30}{34-20} + 1 + 0 + 1 + 0\right) =$ 0.38

Fuzzy Support $(\alpha_2) = \frac{1}{6} \times \left(1 + 0 + 0 + \frac{20-12}{30-12} + 0\right) =$ 0.24

Fuzzy Support $(\alpha_3) = \frac{1}{6} \times \left(0 + 0 + \frac{92-55}{100-55} + 0 + 1\right) =$ 0.3

Assume that the minimum support is at 0.28; the larger 1-sequences are $L_1 = \{\alpha_1, \alpha_3\}$, in which α_1 and α_3 have a fuzzy support value larger than 0.28. Therefore, the total fuzzy support value was computed as: Fuzzy Support (X) = 0.38 + 0.3 = 0.68. Consequently, the objective function = $\frac{1}{0.68}$ = 1.47.

Table 1. Browsing sequences by each user.

User ID	Browsing sequences
1	(A,30) (E,42) (D,98) (C,1)
2	(D,62) (A,8) (D,102)
3	(A,92) (D,89)
4	(A,20) (C,101) (E,118) (B,11) (C,42)
5	(D,64) (A,110) (C,74)
6	(D,80) (C,61) (E,122) (B,17)

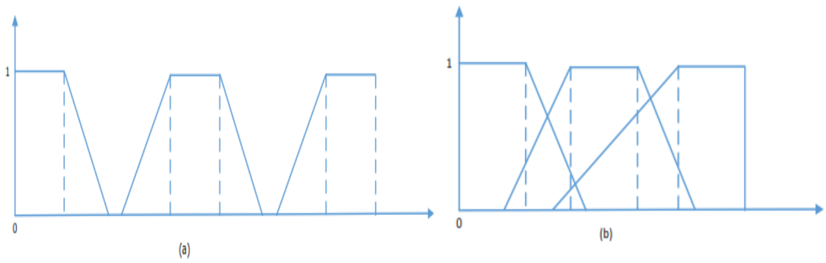

Figure 2. Two types of unsuitable membership functions.

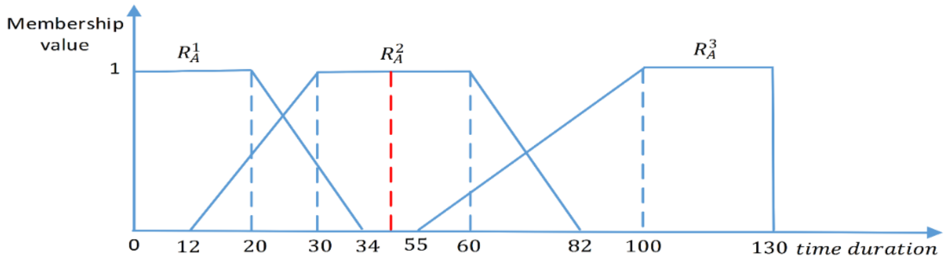

Figure 3. Membership functions for the time duration of Web page *A*.

4. The proposed Algorithm (LA-OMF)

In this section, the main concepts of the LA-OMF algorithm are described.

4.1. The Proposed Framework

The centers and spreads of TMFs are the parameters of the search space, and finding suitable TMFs is considered as an optimization problem. The proposed framework was utilized to find the optimal values of these parameters for the given membership function. In the proposed framework, the number of membership functions is given as an input, and a team of LA is built. After determining a team of LA, the learning process was repeated until the minimum value of the objective function was found. Additionally, during this learning process, the proposed heuristics help eliminate the inappropriate membership functions. Finally, the optimal parameters were obtained for the given membership function. The proposed framework is depicted in figure 4 and consists of the following steps: 1) representing the membership functions using LA; 2) generating the appropriate membership functions; 3) generating actions; 4) evaluating the objective function; and 5) updating the parameters of LA. Each step is explained in details below.

4.2. Representing Membership Functions using Learning Automata (LA)

The time spent by the users on each Web page is one of the parameters of Web usage data, which can be used to analyse the users' browsing behavior. The time spent by the users on each Web page is one of the parameters of Web usage data, which can be used to analyse the users' browsing behavior. This parameter is shown as TMFs. This section describes how to display TMFs using LA. Let $R = \{R_1, R_2, \ldots, R_m\}$ - represent the set of TMFs and *m* be the number of membership functions. Each R_i, $(1 \le i \le m)$ shows the membership function of the *i*-th fuzzy region. R_1 is the membership function of the first fuzzy region and R_m is the membership function of the last fuzzy region. Each R_i is depicted by a quadruple $(wl_{i,j}, cl_{i,j}, cr_{i,j}, wr_{i,j})$, where the indices *i* and *j* indicate the *j*-th parameter of the *i*-th membership function. For example, in figure 5, R_2 is indicated by a quadruple $(wl_{2,1}, cl_{2,2}, cr_{2,3}, wr_{2,4})$. Additionally, for R_i, we assume that $wl_{i,j}$ represents the left spread of the fuzzy region R_i, $cl_{i,j}$ demonstrates the left center of the fuzzy region R_i, $cr_{i,j}$ stands for the right center of fuzzy region R_i, and $wr_{i,j}$ indicates the right spread of the fuzzy region R_i. In the proposed approach, one learning automaton is assigned to each parameter of quadruple $(wl_{i,1}, cl_{i,2}, cr_{i,3}, wr_{i,4})$. These LA are labeled as $A^i_{wl_{i,1}}$, $A^i_{cl_{i,2}}$, $A^i_{cr_{i,3}}$ and $A^i_{wr_{i,4}}$, respectively. Thus, each membership function R_i requires four LA to optimize its parameters. Each automaton has a length of *r*, where *r* shows the number of actions of each automaton. For each automaton, *r* is defined and the action probability vector of each automaton is initialized. For the first and last membership functions, which are characterized by two parameters, only two automata are required. Let K_{max} be the number of membership functions taken from the user as the input. In order to optimize all of these parameters, a team of LA with $K_{max} \times 4 - 4$ LA is built. In this work, the LA team includes the following learning automata:

$$LA = \left\{ \begin{array}{l} A^1_{cr1,3}, A^1_{wr1,4}, \ldots, A^i_{wli,1}, A^i_{cli,2}, A^i_{cri,3}, A^i_{wri,4}, \ldots, \\ A^{m-1}_{wlm-1,1}, A^{m-1}_{clm-1,2}, A^{m-1}_{crm-1,3}, A^{m-1}_{wrm-1,4}, A^m_{wlm,1}, A^m_{clm,2} \end{array} \right\}$$

4.3. Generating Appropriate Membership Functions

The fuzzy region of TMFs is specified by four parameters. TMFs must convince two rules. These rules adjust the parameters in accordance with the form of the membership functions and are described as Equations 12 and 13, respectively.

$$cr_{i,3} \le cl_{i+1,2} \le cr_{i+1,3} \le \ldots \le cl_{m-1,2} \le cr_{m-1,3} \le cl_{m,2} \quad (12)$$

$$wl_{i,1} \le cl_{i,2} \le cr_{i,3} \le wr_{i,4} \quad (13)$$

The first limitation adjusts the arrangement of the centers of the membership functions and the second limitation maintains the trapezoidal shape.
Figure 6 presents a trapezoidal membership function (TMF) with three regions fulfilling the above rules.
The suitability is an important factor in the optimization of membership functions [65]. The suitability value of the membership functions consists of two sections: overlap and coverage. Overlap shows the area covered by two membership functions and coverage shows the area covered by all membership functions. The search space is reduced by a considerable amount by considering the two constraints on overlap and coverage in the membership functions. Equations 14 and 15 satisfy the full coverage and Equation 16 satisfies the suitable overlap.

$$wl_{i-1,1} \le wl_{i,1} \le wr_{i-1,4} \tag{14}$$

$$wl_{i+1,1} \le wr_{i,4} \le wr_{i+1,4} \tag{15}$$

$$wr_{i,4} \le wl_{i+2,1} \tag{16}$$

In this research work, a new approach is proposed to decrease the search space and use the constraints mentioned above. This approach eliminated the inappropriate membership functions and searched the optimal parameter values among the appropriate membership functions. The proposed approach can be described as follows: Each R_i ,$(1 \le i \le m)$ is demonstrated by a quadruple $(wl_{i,j}, cl_{i,j}, cr_{i,j}, wr_{i,j})$. For the first ($R_1$) and last membership function (R_m), only two parameters $(cr_{1,3}, wr_{1,4})$ and $(wl_{m,1}, cl_{m,2})$ are required.
Let

$$P = \left\{ \begin{array}{c} cr_{1,3}, wl_{2,1}, wr_{1,4}, \ldots, wl_{i,1}, wr_{i-1,4}, cl_{i,2}, cr_{i,3}, \\ wr_{i,4}, \ldots, wl_{m-1,1}, \\ cl_{m-1,2}, cr_{m-1,3}, wl_{m,1}, wr_{m-1,4}, cl_{m,2} \end{array} \right\}$$

be the set of all parameters of TMFs. Any permutation of this set can represent a TMFs; however, many of these permutations are invalid. For each TMF of R_i, any non-repetitive sequence of the Cartesian product $\{wr_{i-1,4}, cl_{i,2}\} \times \{wr_{i-1,4}, cl_{i,2}, cr_{i,3}, wl_{i+1,1}\} \times \{wr_{i-1,4}, cl_{i,2}, cr_{i,3}, wl_{i+1,1}\} \times \{cr_{i,3}, wl_{i+1,1}\}$
is considered as a valid sequence (appropriate membership functions). After computing each R_i, any non-repetitive sequence of the Cartesian product $R_1 \times R_2 \times \ldots \times R_m$ is considered as the final appropriate membership function.
For example, for three TMFs, if we do not use the proposed approach, we have the 12! = 479, 001,600 sequences. Among these sequences, the proposed approach considers only 24 membership functions as the appropriate ones. All the appropriate membership functions for three TMFs are represented as $L_3 = \{v_1, v_2, \ldots, v_{24}\}$, where each v_i shows a sequence of valid TMFs points. Table 2 presents 24 appropriate membership functions for three TMFs fulfilling the above rules, i.e. full coverage and appropriate overlap.

4.4. Generating Actions

After determining the set of required learning automata and generating the appropriate membership functions, we describe the learning algorithm used by each automaton. Let LA be a set of the required learning automata. Additionally, let i represent the i-th membership function, where $(1 \le i \le m)$, and j∈ $\{wl_{i,j}, cl_{i,j}, cr_{i,j}, wr_{i,j}\}$ indicate the j-th parameter of the i-th TMFs. Let the selected action by the learning automaton LA_j^i at instant n be represented by $\alpha_{j,k}^i(n)$, where $k(1 \le k \le r)$ represents the action number. Suppose t_{max} is the maximum time duration among all Web pages. The action of each automaton at instant n can be selected as $\alpha_{j,k}^i(n)$. Using the selected action and the actions of other LA in step $n-1$, the selected action vector such as $A(n)$ was build. In order to build an action vector $A(n)$, we used the selected action $\alpha_{j,k}^i(n)$ and the actions of other $LA_p^i \in LA, p \ne j$ such as $\alpha_{p,k'}^i(n-1)$,$1 \le k' \le r$ at the instant $n-1$.
By considering the union of all of these actions, the selected action vector $A(n)$ can be determined using Eq. 17.

$$A(n) = \alpha_{j,k}^i(n) \cup_{p=1}^{|LA|} \alpha_{p,k'}^i(n-1) \tag{17}$$

Suppose that $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_r)$ represents the action set and $p(n) = \{p_1, p_2, \ldots, p_r\}$ denotes the action probability distribution of LA_j^i at instant n. The method used to select an action in LA_j^i was as follows:

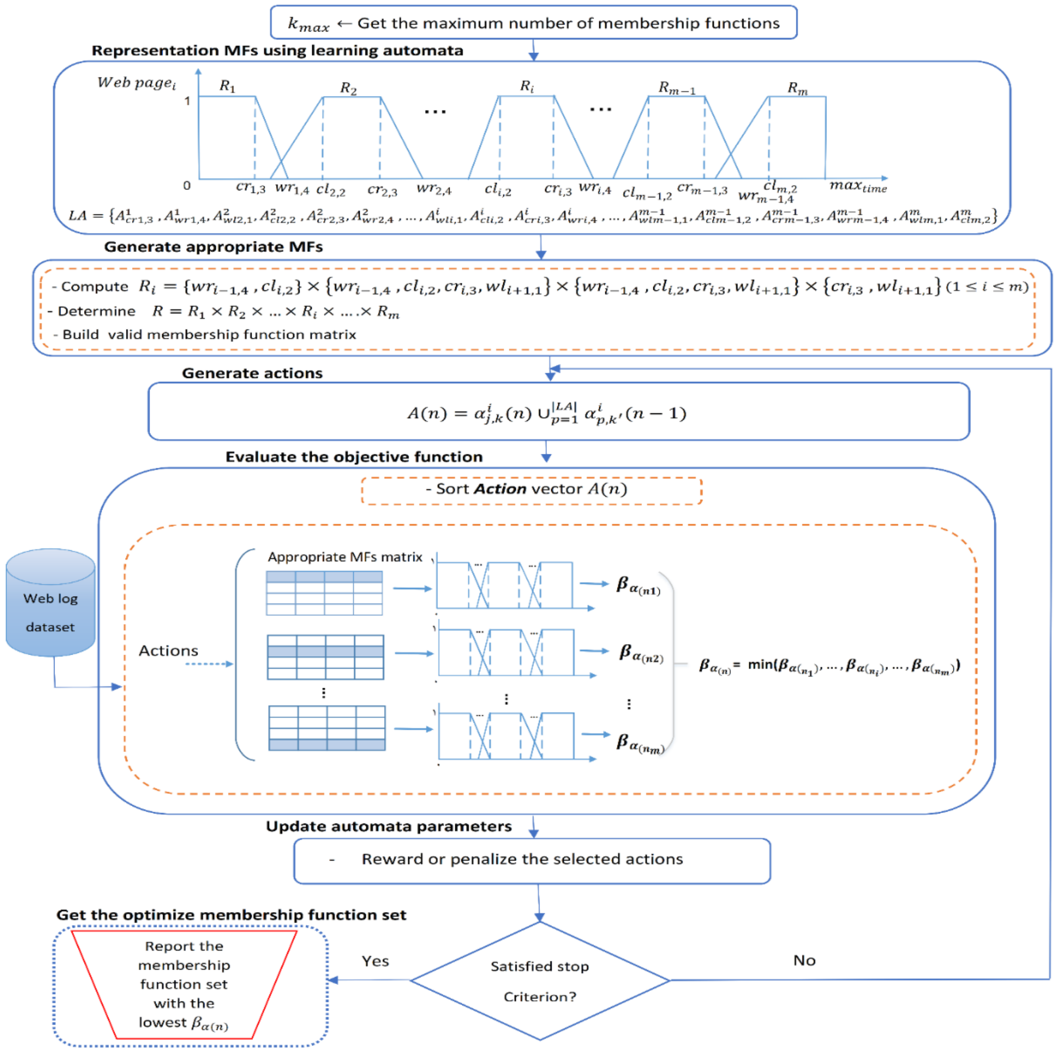

Figure 4. The proposed framework.

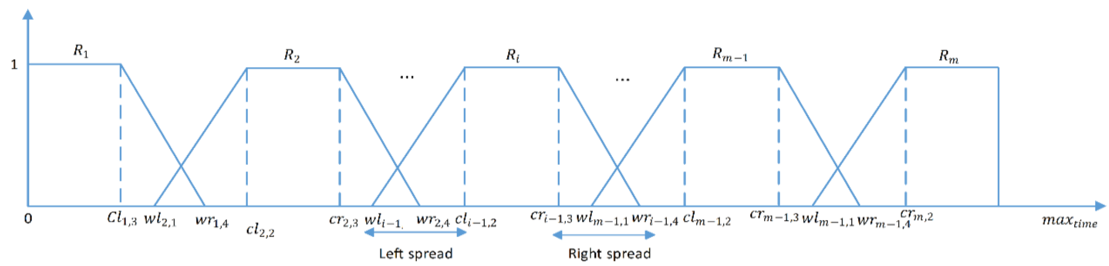

Figure 5. Time duration of each web page as the trapezoidal membership functions.

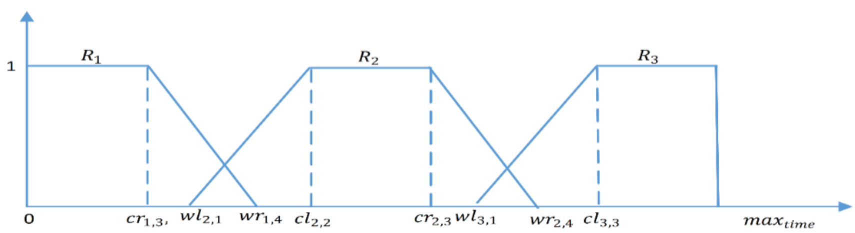

Figure 6. A trapezoidal membership function with three regions.

Table 2. Number of appropriate membership functions for the three trapezoidal membership functions.

V_i	Valid Sequence V_i	V_i	Valid Sequence V_i
V_1	$(cr_{1,3}\ wl_{2,1}\ wr_{1,4}\ cl_{2,2}\ cr_{2,3}\ wl_{3,1}\ wr_{2,4}\ cl_{3,3})$	V_{13}	$(wl_{2,1}\ cr_{1,3}\ wr_{1,4}\ cl_{2,2}\ cr_{2,3}\ wl_{3,1}\ wr_{2,4}\ cl_{3,3})$
V_2	$(cr_{1,3}\ wl_{2,1}\ wr_{1,4}\ cl_{2,2}\ cr_{2,3}\ wl_{3,1}\ cl_{3,3}\ wr_{2,4})$	V_{14}	$(wl_{2,1}\ cr_{1,3}\ wr_{1,4}\ cl_{2,2}\ cr_{2,3}\ wl_{3,1}\ cl_{3,3}\ wr_{2,4})$
V_3	$(cr_{1,3}\ wl_{2,1}\ wr_{1,4}\ cl_{2,2}\ wl_{3,1}\ cr_{2,3}\ wr_{1,4}\ cl_{3,3})$	V_{15}	$(wl_{2,1}\ cr_{1,3}\ wr_{1,4}\ cl_{2,2}\ wl_{3,1}\ cr_{2,3}\ wr_{2,4}\ cl_{3,3})$
V_4	$(cr_{1,3}\ wl_{2,1}\ wr_{1,4}\ cl_{2,2}\ wl_{3,1}\ cr_{2,3}\ cl_{3,3}\ wr_{2,4})$	V_{16}	$(wl_{2,1}\ cr_{1,3}\ wr_{1,4}\ cl_{2,2}\ wl_{3,1}\ cr_{2,3}\ cl_{3,3}\ wr_{2,4})$
V_5	$(cr_{1,3}\ wl_{2,1}\ wr_{1,4}\ wl_{3,1}\ cl_{2,2}\ cr_{2,3}\ wr_{2,4}\ cl_{3,3})$	V_{17}	$(wl_{2,1}\ cr_{1,3}\ wr_{1,4}\ wl_{3,1}\ cl_{2,2}\ cr_{2,3}\ wr_{2,4}\ cl_{3,3})$
V_6	$(cr_{1,3}\ wl_{2,1}\ wr_{1,4}\ wl_{3,1}\ cl_{2,2}\ cr_{2,3}\ cl_{3,3}\ wr_{2,4})$	V_{18}	$(wl_{2,1}\ cr_{1,3}\ wr_{1,4}\ wl_{3,1}\ cl_{2,2}\ cr_{2,3}\ cl_{3,3}\ wr_{2,4})$
V_7	$(cr_{1,3}\ wl_{2,1}\ cl_{2,2}\ wr_{1,4}\ cr_{2,3}\ wl_{3,1}\ wr_{2,4}\ cl_{3,3})$	V_{19}	$(wl_{2,1}\ cr_{1,3}\ cl_{2,2}\ wr_{1,4}\ cr_{2,3}\ wl_{3,1}\ wr_{2,4}\ cl_{3,3})$
V_8	$(cr_{1,3}\ wl_{2,1}\ cl_{2,2}\ wr_{1,4}\ cr_{2,3}\ wl_{3,1}\ cl_{3,3}\ wr_{2,4})$	V_{20}	$(wl_{2,1}\ cr_{1,3}\ cl_{2,2}\ wr_{1,4}\ cr_{2,3}\ wl_{3,1}\ cl_{3,3}\ wr_{2,4})$
V_9	$(cr_{1,3}\ wl_{2,1}\ cl_{2,2}\ wr_{1,4}\ wl_{3,1}\ cr_{2,3}\ wr_{2,4}\ cl_{3,3})$	V_{21}	$(wl_{2,1}\ cr_{1,3}\ cl_{2,2}\ wr_{1,4}\ wl_{3,1}\ cr_{2,3}\ wr_{2,4}\ cl_{3,3})$
V_{10}	$(cr_{1,3}\ wl_{2,1}\ cl_{2,2}\ wr_{1,4}\ wl_{3,1}\ cr_{2,3}\ cl_{3,3}\ wr_{2,4})$	V_{22}	$(wl_{2,1}\ cr_{1,3}\ cl_{2,2}\ wr_{1,4}\ wl_{3,1}\ cr_{2,3}\ cl_{3,3}\ wr_{2,4})$
V_{11}	$(cr_{1,3}\ wl_{2,1}\ cl_{2,2}\ cr_{2,3}\ wr_{1,4}\ wl_{3,1}\ wr_{2,4}\ cl_{3,3})$	V_{23}	$(wl_{2,1}\ cr_{1,3}\ cl_{2,2}\ cr_{2,3}\ wr_{1,4}\ wl_{3,1}\ wr_{2,4}\ cl_{3,3})$
V_{12}	$(cr_{1,3}\ wl_{2,1}\ cl_{2,2}\ cr_{2,3}\ wr_{1,4}\ wl_{3,1}\ cl_{3,3}\ wr_{2,4})$	V_{24}	$(wl_{2,1}\ cr_{1,3}\ cl_{2,2}\ cr_{2,3}\ wr_{1,4}\ wl_{3,1}\ cl_{3,3}\ wr_{2,4})$

1) Using the probability vector, the interval length for each action of LA_j^i was determined by Equation 18.

$$
\begin{aligned}
&\alpha_1 \in [0\ , 0+p_1)\\
&\alpha_2 \in [0+p_1, 0+p_1+p_2)\\
&\alpha_3 \in [0+p_1+p_2\ , 0+p_1+p_2+p_3)\\
&\quad .\\
&\quad .\\
&\quad .\\
&\alpha_r \in \left[\begin{matrix}0+p_1+p_2+\ldots+p_{L-1}\ , 0+p_1\\ +p_2+\ldots+p_r\end{matrix}\right)
\end{aligned}
\tag{18}
$$

2) A random number was selected between zero and one.
3) If the selected random number belongs to kth interval, then this shows that the kth action can be selected.

Example 2:
Suppose that $LA_j^i \in LA$ is a learning automaton and includes four actions. Additionally, we assume that the action probability of LA_j^i at instant n is $p(n) = (0.2,0.3,0.4,0.1)$. In order to choose an action, we generate a random number between 0 and 1. For example, if the random number is 0.6, the selected action according to Equation 18 will be equal to 3. Thus the selected action is α_3. Each selected action corresponds to one optimized parameter value. In this case, the optimal parameter values are obtained using the selected action according to Eq. 19. In this example, the optimal parameter value for the chosen action α_3 using Eq. 19 is 0.66.

$$
\textit{Optimize parameter value} = \frac{i-1}{r-1},\ 1 \le i \le r \tag{19}
$$

4.5. Evaluating Objective Function

After generating all the appropriate membership functions and sorting the action vector $A(n)$, the next step is to evaluate the objective function to find the value of $\beta_\alpha(n)$. The appropriate membership functions can be represented by a 2D vector, known as the valid membership function matrix. Each row of this matrix denotes an appropriate sequence (valid membership function). The number of rows in this matrix shows the number of valid membership functions. The proposed algorithm uses a function assessment called $\beta_\alpha(n)$. In order to compute $\beta_\alpha(n)$ for each row of this matrix, the membership function was first constructed, and then the objective function was evaluated according to Equation 7. By considering the minimum value of the objective function value among all the valid membership functions, the value of $\beta_\alpha(n)$ was determined.

Example 3: Suppose that the number of membership functions is 2 and the action vector is A(n) = (0.6, 0.3, 0.8, 0.2). For two membership functions, we need four valid sequences, as depicted in the form of a matrix in figure 7. The *Action* vector is sorted in an ascending order using α(n) to build the membership functions. In this case, four valid membership functions were built. After performing the mentioned steps, the value of $\beta_\alpha(n)$ was determined as 0.57. As a result, the membership function set $\{wl_{2,1} = 0.2\,, cr_{1,3} = 0.3\,, cl_{2,2} = 0.8\,, wr_{1,4} = 0.6\}$ associated with $\beta_\alpha(n)$ = 0.57 was the best result.

4.6. Updating Automaton Parameters

Using the value of the objective function and the global best value, we decided whether the selected action should be rewarded or penalized. Let $Global\,\beta_\alpha(n)$ show a minimum value of the objective function obtained so far. After finding the value of $\beta_\alpha(n)$ as a common reinforcement signal, each learning automaton decides whether to give reward or penalty to its selected action.

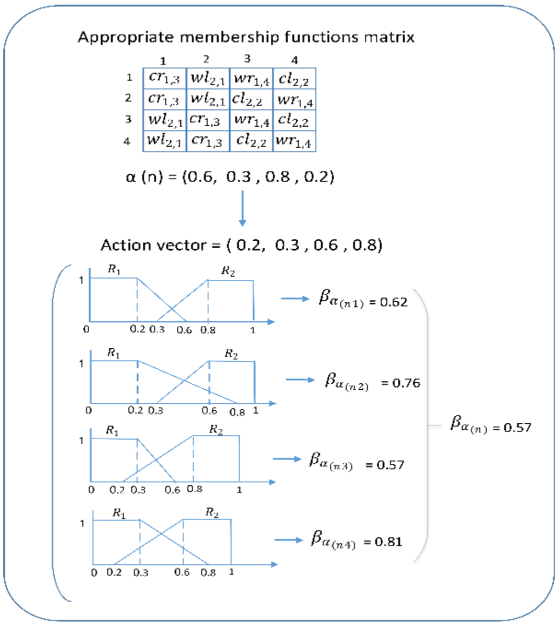

Figure 7. An example of finding objective function values.

After obtaining the value of $\beta_\alpha(n)$, it was compared with $Global\,\beta_\alpha(n)$. If the value of $\beta_\alpha(n)$ is less than the value of $Global\,\beta_\alpha(n)$, then the number of positions of the membership functions are saved. Additionally, the selected action $\alpha^i_{j,k}(n)$ is rewarded according to Equation 4; otherwise, only the selected action is penalized according to Equation 5.

4.7. Extracting the Optimal Number of Membership Functions

After determining the proposed framework, we developed the LA-OMF algorithm to automatically find the optimal number and positions of the trapezoidal membership functions. Let K_{max} be the optimal number of the membership functions and represent the maximum number of membership functions specified by the user. In order to determine the value of the optimized membership function, the proposed algorithm was used as follows: For each membership function, $2 \leq MFs \leq K_{max}$, the value of $\beta_\alpha(n)$, using the proposed framework, was determined. Finally, the lowest $\beta_\alpha(n)$ was considered as an optimal number of the membership functions. Figure 8 demonstrates how to find the optimal number of the membership functions. Additionally, the pseudo-code for the LA-OMF algorithm is shown below.

4.8. Complexity Analysis

In this section, both the time and space complexity of the proposed LA-OMF algorithm in comparison to the meta-heuristic algorithms is presented.

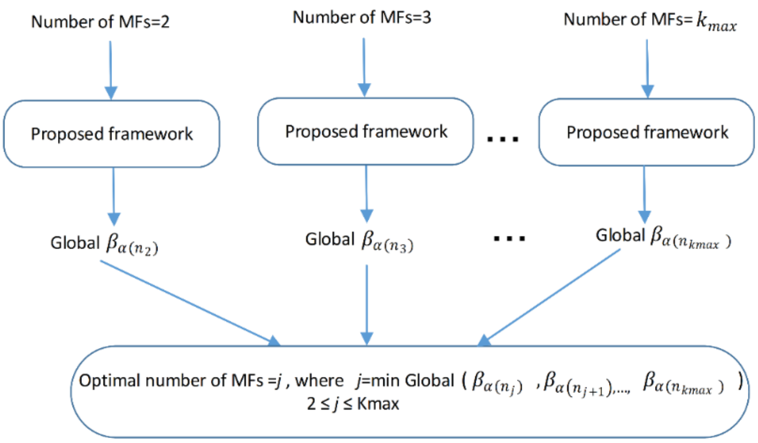

Figure 8. Finding the optimal number of trapezoidal membership functions.

Proposed LA-OMF Algorithm

// ***n***: number of Web log records; ***m***: number of Web pages; $\boldsymbol{K_{max}}$: maximum number of membership functions; **r**: number of actions; **Itr**: number of iterations

Input: n, m, K_{max},r

Output: optimal number of membership functions and its associated set of membership functions

Step 1: Find appropriate membership functions for each MF_i , $2 \le i \le K_{max}$.

Step 2: $Global\ \beta_\alpha(n) \leftarrow \infty$

Step 3: For each MF_i , $2 \le i \le K_{max}$ **do**

Step 3.1: LA←Create a team of LA with the size of K_{max} × 4 - 4) automata.

Step 3.2: Initialize parameters of each automaton. (see Section 3.2).

Step 3.3: Define the action-set of each automaton (see Section 4.4).

Step 3.4: n←0

Step 3.5: Repeat

Step 3.5.1: For each automaton $A_j^i \in LA$ ($1 \le i \le m$, $j \in \{wl_{i,1}, cl_{i,2}, cr_{i,3}, wr_{i,4}\}$) at instant *n* **do**:

Step 3.5.1.1: selects an action $\alpha_{j,k}^i(n)$ (see Section 4.3).

Step 3.5.1.2: Action← $\alpha_{j,k}^i(n) \cup_{p=1}^{|LA|} \alpha_{j,k'}^i(n-1), where\ p \neq j, and\ k' \in Action - set$

Step 3.5.1.3: Action←Sort (Action)

Step 3.5.1.4: $\beta_\alpha(n) \leftarrow$ Evaluate the objective function using Action and appropriate membership functions (see Section 4.5).

Step 3.5.1.5: If $\beta_\alpha(n) < Global\beta_\alpha(n)$,

Step 3.5.1.5.1: $Global\beta_\alpha(n) \leftarrow \beta_\alpha(n)$

Step 3.5.1.5.2: Save the number (MF_i) and its associated set of membership functions

Step 3.5.1.5.3: Reward the selected action $\alpha_{j,k}^i(n)$

else

Step 3.5.1.5.4: Penalize the selected action $\alpha_{j,k}^i(n)$

End if.

End foreach

Step 3.5.1.6: n←n+1

Until learning automata converges to the best actions

End for

4.8.1. Time Complexity

Assume that the maximum number of membership functions for each Web page is k_{max}, the number of actions is r, the number of iterations that it takes for an LA for converge is Itr, the number of transactions is n, the number of items (Web pages) is m, and the number of large 1-sequences is $|L_1|$. Then the time complexity of each step of the proposed algorithm LA-OMF can be described as follows: Step 1 requires O (k_{max}); steps 2 requires O (1); step 3 requires O (k_{max}); step 3.1 requires O (k_{max}); step 3.2 requires O (r); step 3.3 requires O (r); step 3.4 requires O (1); step 3.5 requires O (Itr); step 3.5.1 requires O (k_{max}); step 3.5.1.1 requires O (r); step 3.5.1.2 requires O (r); step 3.5.1.3 requires O ($rlogr$); and step 3.5.1.4 computes the objective function. Thus, the actual execution time of the proposed algorithm depends on the adopted dataset and the objective function. The objective function consists of two parts, namely the suitability of the membership functions and the number of large 1-sequences (L_1). The computational complexity of calculating the number of L_1 is thus O (n × m × k_{max}). In order to evaluate the suitability factor, the proposed approach requires to check the values of the overlap and coverage factors. The computational complexity of finding the suitability values is thus O (m× k_{max}). Hence, in the worst case, the computational complexity of evaluating an automaton using the objective function is [$n \times m \times k_{max}$ + m× k_{max}], which is O (n × m× k_{max}). Step 3.5.1.5 requires O (k_{max}) + O (r), and step 3.5.1.6 requires O (1). Finally, the fuzzy association rules are mined from the given database. Consequently, it requires O(s). Considering all the above steps, the total time complexity is:

O (k_{max})+$\mathrm{O}(k_{max})$ × [$\mathrm{O}(k_{max})$ + O (r) + O(ir)× [O(k_{max})× [O($rlogr$) + O(n× m × k_{max})]]] +O(s), Which is simplified as follows: O(ir)× [O($rlogr$)+O (n× m × k_{max})] +O(s).

We showed that the time complexity of the proposed approach was more efficient than that of meta-heuristic algorithms. The time complexity of the nature-inspired meta-heuristic algorithms such as evolutionary algorithms was analyzed as follows. Assume that the number of generations (iterations) is *itr*, the population size is *Psize*, and the time for evaluating a chromosome is *eTimeChro*.

The execution time of the meta-heuristic algorithm is O (*itr* × *Psize* ×*eTimeChro*) . The *eTimeChro* value depends on the fitness function used. Thus, the actual execution time of the genetic algorithm depends on the adopted dataset and the selected fitness function. Consider the fitness function.

Since it consists of two parts, namely the suitability of the membership functions and the number of large 1-itemsets ($|L_1|$), the computational complexity of a chromosome can be analyzed as follows. Assume that the number of transactions log is n, the number of items (web pages) is m, and the maximum possible number of membership functions (linguistic terms) for an item (Web page) is k_{max}. The computational complexity of calculating the number of large 1-itemsets is thus O (n × m × k_{max}).

In order to evaluate the suitability factor, the genetic algorithm is required to check the chromosomes to calculate the values of the overlap and coverage factors. The length of a chromosome can be found by O (m× k_{max}). If we assume that the length of the chromosome is r, the time required to arrange the chromosome with the length r will be $\mathrm{O}(rlogr)$. The computational complexity of finding the suitability values is thus O (m × k_{max}).

Hence, in the worst case, the computational complexity of evaluating a chromosome with a fitness function is O($n \times m \times k_{max}$ + $m \times k_{max}$), which is $\mathrm{O}(n \times m \times k_{max})$.

In the final step, the fuzzy association rules are mined from the given database. Therefore, it requires O(s). Considering all the above steps, the total time complexity of the genetic algorithm is: O (*itr* × *Psize*) × [$\mathrm{O}(rlogr)$ + O (n × m × k_{max})] + O(s).

4.8.2. Space Complexity

Assume that the maximum number of membership functions is k_{max} and the number of actions is r. The proposed algorithm uses $K_{max} \times r$ automata to find the centers and spreads of the membership functions. Hence, the total space complexity of the proposed algorithm is O(K_{max} ×r). Other algorithms such as the nature-inspired meta-heuristic algorithms use another factor called the number of agents (for example, the number of chromosomes) to calculate the space complexity. Assume that the number of agents (chromosomes) is n, the maximum number of membership functions is k_{max}, and the length of each chromosome is r. Then the space complexity of these algorithms is O ($n \times k_{max} \times r$). Thus, the space efficiency of the proposed algorithm in comparison with that of nature inspired meta-heuristic algorithms is efficient.

5. Experiments and Analysis of Results

In this section, several experiments were performed to demonstrate the effectiveness of the proposed LA-OMF algorithm. We implemented all the experiments using the java language on a computer with Intel core i7 at 1.80GHz and 16GB RAM running windows 10. In this work, the LA-OMF results were compared with the fuzzy web mining algorithm (FWMA) [66], FTARM+PFGM[47], and MAOMF [41]. The FWMA algorithm was applied to the fuzzy Web mining algorithms, and pre-defined TMFs were used. The FTARM+PFGM and MAOMF algorithms used the optimization of the triangular membership functions. Thus the time parameter for the FTARM+PFGM and MAOMF algorithms were considered as triangular membership functions.

In order to analyze the experimental results, the criteria such as the fuzzy support, overlap, suitability, coverage, average value of objective function ($\beta_\alpha(n)$), execution time, and number of large 1-sequences L_1 were considered. In the experiments, the minimum and maximum possible numbers of the linguistic terms were set to 2 and 6, respectively. In all the experiments, each algorithm was independently run for 30 times, and the mean value of the 30 runs was considered. The parameters used by LA-OMF are presented in table 3. The parameters of the other compared algorithms according to the literature were also taken into account. Figure 9 shows the pre-defined membership functions used in the experiments.

Table 3. Parameters used by LA-OMF.

Parameter	Value
Maximum number of MFs	6
Reward rate (a)	0.3
Penalty rate(b)	0.003
Number of actions (r)	20
Minimum support	0.002
Minimum confidence	0.1

5.1. Preparation of Datasets

In order to evaluate and analyze the proposed LA-OMF, we considered the DePaul CTI dataset [67]. The data was collected during a two-week period in April 2002 from a sample of users who viewed this website. The cleaned data was obtained by deleting the page views with a low support and removing the sessions of size one. The cleaned data included 13745 user sessions and 683-page views.

The proposed algorithm specified the appropriate membership functions for each Web page of the CTI dataset [67]. In all the experiments, the Web page with the number of 387 (/news/default.asp) was selected among 683 pages. In this dataset, each distinct page view was represented by a unique number called page view IDs. Table 4 shows a part of the log data in the CTI dataset. In this table, the list and ID of each page view are given.

In table 5, the browsing sequences are depicted for each client/user. Each client had a unique number called the client ID. Each browsing sequence showed the list of the page view IDs along with the amount of time each user spent on a page view during one session. The maximum page view time was 999 s.

5.2. Experimental Evaluations

Experiment 1:

This experiment aimed at finding a suitable number of membership functions and their optimized parameters. In this experiment, the values of 2, 3, 4, 5, and 6 were used for the number of membership functions. Figure 10 depicts the optimal number of membership functions for LA-OMF on the CTI dataset. Additionally, we determined the optimal number of membership functions for other algorithms. Then using the obtained membership functions, the values of the fuzzy support, overlap, suiability, coverage, and average value of objective function ($\beta_\alpha(n)$) were determined. The results obtained for the compared algorithms are shown in table 6. In this table, the optimal number of membership functions, taking into account the lowest value of the objective function ($\beta_\alpha(n)$), was three for the proposed algorithm, four for the FTARM + PFGM algorithm, two for FWMA, and two for the MAOMF algorithm, respectively.

The results of table 6 demonstrate that for the number of membership functions 2, 3, 4, 5, and 6, the average value of the objective function using the LA-OMF algorithm increased by 20%, 23%, 18%, 17%, and 15%, respectively. Thus, the average efficiency of the objective function in comparison to other algorithms increased by 20%. In a similar way, LA-OMF increased the average efficiency of the fuzzy support by 17%. According to the results in table 6, LA-OMF obtained the minimum value of the objective function for each number of membership functions.

Figures $11a_1 - e_1$ show the initial membership functions for the membership functions 2, 3, 4, 5, and 6, respectively. Figure 8 shows that the membership functions generated in the first iteration of the proposed algorithm have a large overlap or are very separate. After executing the proposed algorithm, the membership functions have a suitable shape. The optimized membership functions are presented in figure $12a_2 - e_2$, respectively.

Experiment 2:

The purpose of this experiment was to examine the effect of different dataset sizes on the efficiency of the proposed LA-OMF algorithm. The datasets with different sizes including the 50K, 100K, 150K, 200K, and 250K sessions were tested in this experiment. The results obtained are shown in table 7.

The results show that the LA-OMF algorithm produces better results for the parameters of $\beta_\alpha(n)$, fuzzy support, and suitability. Thus by increasing the dataset sizes, the performance of the proposed LA-OMF algorithm with respect to the other algorithms was a significant improvement.

Table 4. List of the page view IDs along with a page view in the CTI dataset.

Page view id	Page view
0	/admissions/
1	/admissions/career.asp
2	/admissions/checklist.asp
3	/admissions/costs.asp
681	/shared/404.asp?404; http://www.cs.depaul.edu/msoffice/cltreq.asp
682	/shared/404.asp?404; http://www.cs.depaul.edu/resources/grad_scholarships.asp

Table 5. List of client IDs along with browsing sequences by each user in the CTI dataset.

Client ID	Browsing sequence
1	(679, 2) (574, 7) (585, 5) (604, 4)
2	(387, 37) (558, 20)
3	(387, 24) (400, 125) (71, 26) (228, 34)
…	…
13563	(54, 11) (358, 55)

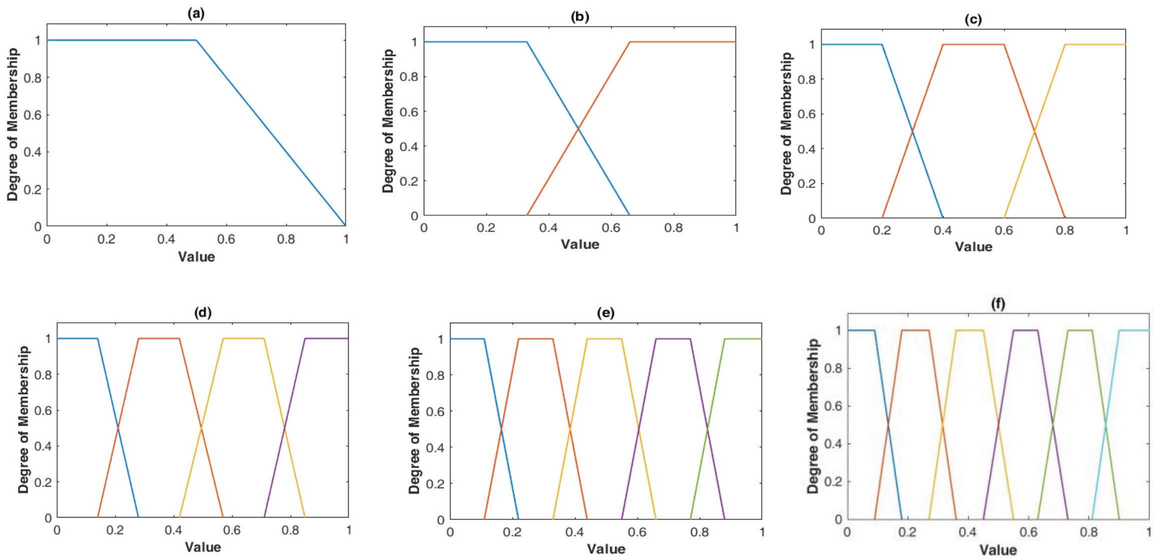

Figure 9. The pre-defined membership functions with different numbers of linguistic terms used in the experiments.

Experiment 3:

In order to show the change of $\beta_\alpha(n)$, the fuzzy support and overlap along with different numbers of iterations, another experiment was performed. Figs. 12, 13, and 14 show the change of $\beta_\alpha(n)$, fuzzy support, and overlap along with different numbers of iterations on the datasets of the 50k and 250k sessions, respectively. As it could be seen in these figures, the LA-OMF algorithm for all the dataset sizes produced the minimum value of $\beta_\alpha(n)$ in comparison to the other algorithms. The values of $\beta_\alpha(n)$, fuzzy support, and overlap for the FWMA algorithm were plotted as a straight line because FWMA produced a constant value at each iteration. Figures 14 and 15 also show that LA-OMF obtained the membership functions with a higher fuzzy support and a better overlap than the other algorithms. In addition, using the constraints introduced in the proposed algorithm made the proposed algorithm more efficient than the other algorithms.

Table 6. Comparison of the results between LA-OMF and other algorithms.

Number of membership functions	Algorithm	Overlap	Coverage	Suitability	Fuzzy support	$\beta_\alpha(n)$
# MFs = 2	LA-FOMF	0.0000	1.0000±0.0000	1.0000	**1.3541**	**0.7368±0.0243**
	FWMA	0.0000	1.0000±0.0000	1.0000	0.8216	1.2175±0.0000
	FTARM+PFGM	0.0018	1.0000±0.0000	1.0018	1.2805	0.7823±0.0342
	MAOMF	0.0512	1.0000±0.0000	1.0512	1.3101	0.8024±0.0580
# MFs = 3	LA-FOMF	0.0026	1.0000±0.0000	1.0026	**1.4771**	**0.6842±0.0092**
	FWMA	0.0000	1.0000±0.0000	1.0000	0.8285	1.2070±0.0000
	FTARM+PFGM	0.0186	1.0000±0.0000	1.0186	1.1951	0.8523±0.0672
	MAOMF	0.0245	1.0000±0.0000	1.0245	1.2166	0.8421±0.0652
# MFs = 4	LA-FOMF	0.0073	1.0000±0.0000	1.0073	**1.2674**	**0.7895±0.0072**
	FWMA	0.0000	1.0000±0.0000	1.0000	0.8261	1.2105±0.0000
	FTARM+PFGM	0.0235	1.0000±0.0000	1.0235	1.3714	0.7463±0.0161
	MAOMF	0.0362	1.0000±0.0000	1.0362	1.1727	0.8836±0.1139
# MFs = 5	LA-FOMF	0.0140	1.0000±0.0000	1.0140	**1.2024**	**0.8421±0.0058**
	FWMA	0.0000	1.0000±0.0000	1.0000	0.8173	1.2235± 0.0000
	FTARM+PFGM	0.0476	1.0000±0.0000	1.0476	1.1455	0.9145±0.0524
	MAOMF	0.0752	1.0000±0.0000	1.0752	1.1369	0.9457±0.0943
# MFs = 6	LA-FOMF	0.0230	1.0000±0.0000	1.0230	**1.1424**	**0.8947±0.0063**
	FWMA	0.0000	1.0000±0.0000	1.0000	0.8055	1.2414±0.0000
	FTARM+PFGM	0.0674	1.0000±0.0000	1.0674	1.1174	0.9552±0.0389
	MAOMF	0.0923	1.0000±0.0000	1.0923	1.0987	0.9941±0.0873

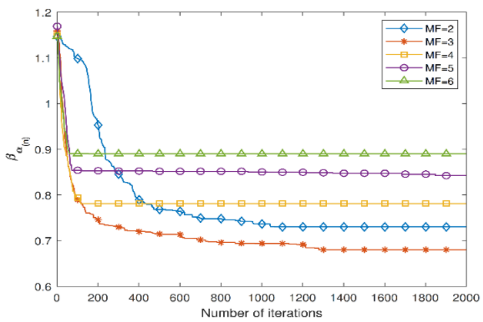

Figure 10. Value of $\beta_\alpha(n)$ obtained by LA-OMF for the membership functions 2, 3, 4, 5, and 6.

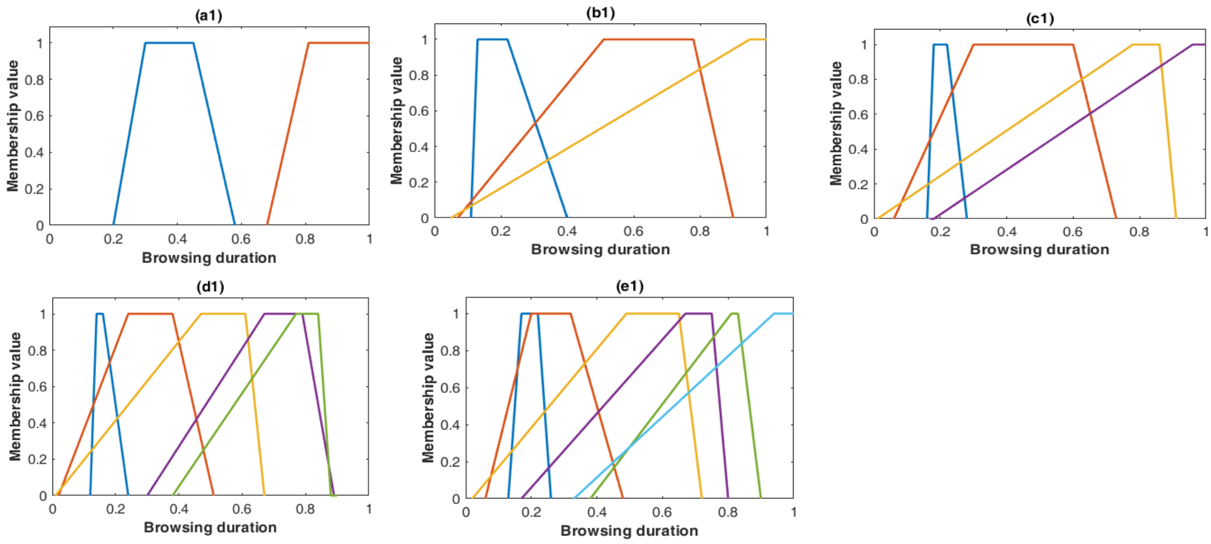

Figure 11. Initial membership functions before executing LA-OMF in the CTI dataset.

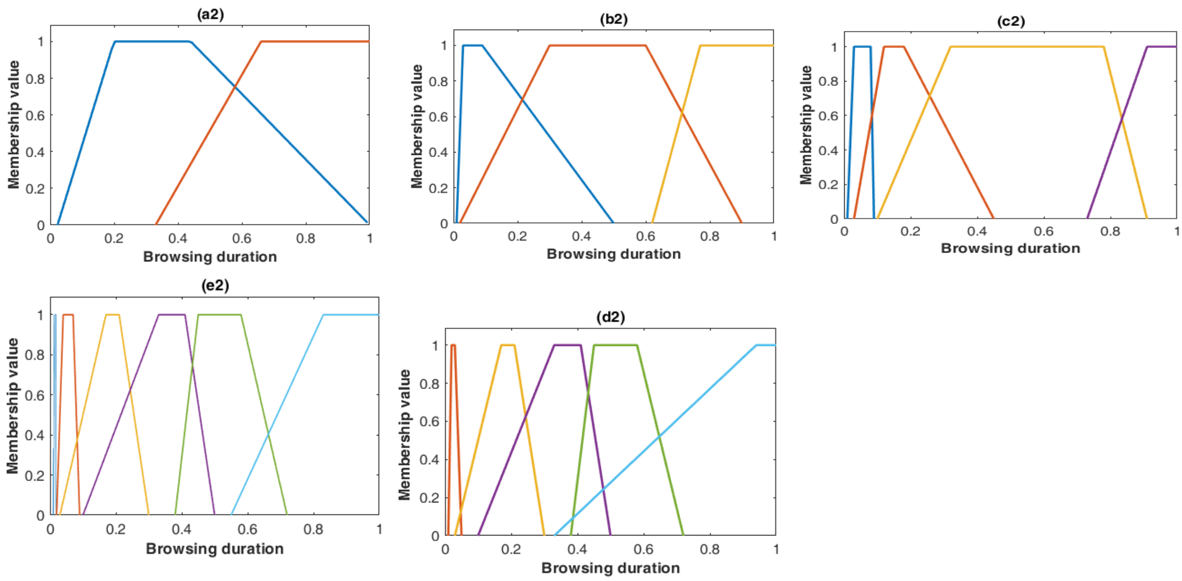

Figure 12. Optimized membership functions after executing LA-OMF in the CTI dataset.

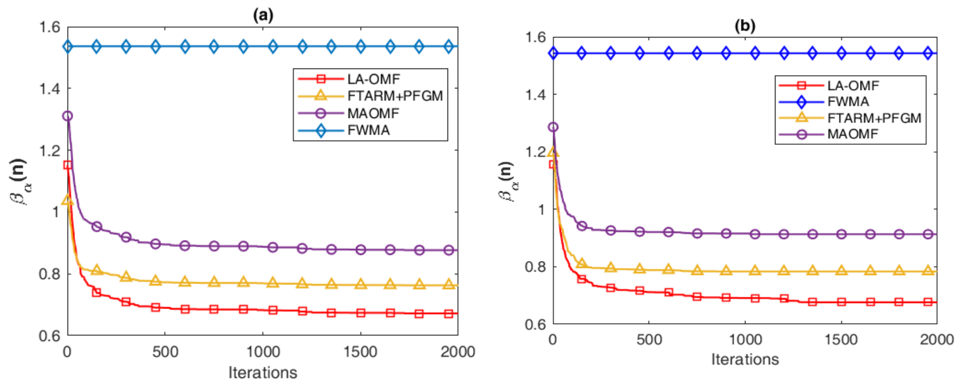

Figure 13. Change of $\beta_\alpha(n)$ along with different numbers of iterations for four test algorithms on different datasets. (a) 50k sessions and (b) 250k sessions.

Table 7. Comparison of the results acquired from LA-OMF and FWMA on the data of different sizes.

Data	Algorithm	Overlap	Coverage	Suitability	Fuzzy support	$\beta_\alpha(n)$
50K	LA-OMF	0.0018	1.0000	1.0018	**1.4918**	**0.6715**
	FWMA	0.0000	1.0000	1.0000	0.6509	1.5362
	FTARM+PFGM	0.0064	1.0000	1.0064	1.3191	0.7629
	MAOMF	0.0723	1.0000	1.0723	1.2238	0.8762
100K	LA-OMF	0.0025	1.0000	1.0025	**1.5075**	**0.6650**
	FWMA	0.0000	1.0000	1.0000	0.6594	1.5164
	FTARM+PFGM	0.0058	1.0000	1.0058	1.3710	0.7336
	MAOMF	0.0535	1.0000	1.0535	1.2375	0.8513
150K	LA-OMF	0.0043	1.0000	1.0043	**1.6156**	**0.6216**
	FWMA	0.0000	1.0000	1.0000	0.6656	1.5023
	FTARM+PFGM	0.0081	1.0000	1.0081	1.4093	0.7153
	MAOMF	0.1763	1.0000	1.1763	1.3565	0.8671
200K	LA-OMF	0.0039	1.0000	1.0039	**1.5826**	**0.6343**
	FWMA	0.0000	1.0000	1.0000	0.6632	1.5075
	FTARM+PFGM	0.0072	1.0000	1.0072	1.3688	0.7358
	MAOMF	0.1636	1.0000	1.1636	1.3109	0.8876
250K	LA-OMF	0.0050	1.0000	1.0050	**1.4761**	**0.6763**
	FWMA	0.0000	1.0000	1.0000	0.6524	1.5428
	FTARM+PFGM	0.0413	1.0000	1.0413	1.3288	0.7836
	MAOMF	0.1483	1.0000	1.1483	1.2574	0.9132

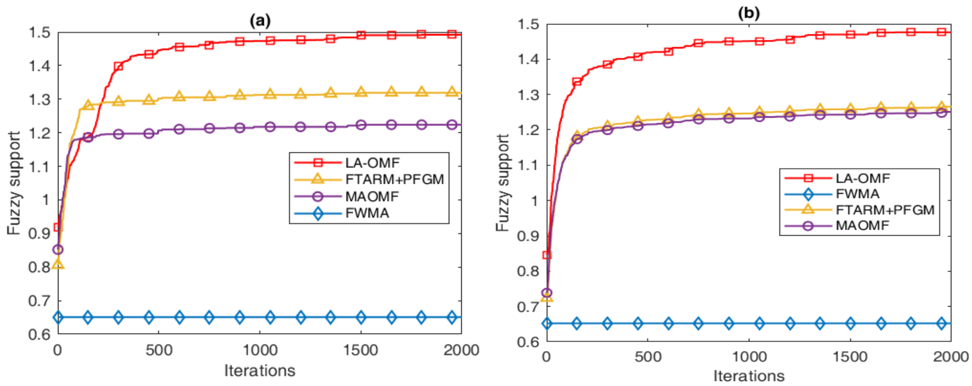

Figure 14. Change of fuzzy support along with different numbers of iterations for four test algorithms on different datasets. (a) 50k sessions and (b) 250k sessions.

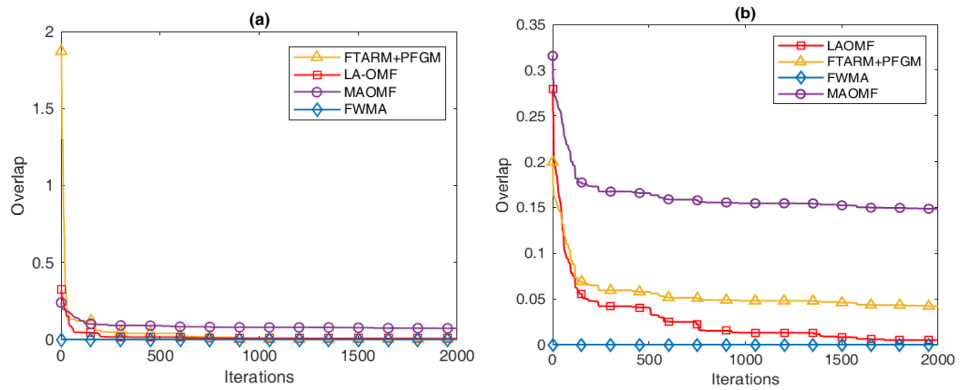

Figure 15. Change of overlap along with different numbers of iterations for four test algorithms on different datasets. (a) 50k sessions and (b) 250k sessions.

Experiment 4:
Another experiment was performed to show the effect of the number of large 1-sequences and the number of rules with different numbers of minimum supports for LA-OMF and other algorithms. The relationship between the number of large 1-sequences and the mininmum supports for four test algorithms is shown in figure 16. It could be seen in figure 16 that the number of large 1-sequences extracted by the proposed algorithm was larger than the other algorithms. Figure 17 shows the relationship between the numbers of association rules extracted by LA-OTMF along with different minimum supports. It could be seen in figure 17 that the number of extracted rules by LA-OMF was also higher than the other algorithms.

Experiment 5:
The purpose of this experiment was to compare the average execution time of the LA-OMF algorithm with other algorithms. The total running time of LA-OMF included two steps. In Step 1, the running time was calculated in order to find the optimized membership functions. In Step 2, the running time was calculated in order to mine large 1-sequences using the obtained optimized membership functions. Thus, the total running time was computed as follows:

$$Total\ runinng\ time = \left(\sum_{i=2}^{k_{max}} running\ time\ for\ MF_i \right) + \left(\begin{array}{l} running\ time\ for \\ mining\ large\ 1-sequences \end{array} \right) \quad (20)$$

Figure 18 gives the total running time for LA-OMF and other algorithms for the CTI dataset. The number of membership functions and the running time also increased. It can also be observed that with increase in the dataset size, the average execution time in the proposed algorithm and other algorithms increases linearly.

Experiment 6:
The purpose of carrying out this experiment was to investigate the effect of different values of the parameters *a* and *b* on the LA-OMF performance. We considered two cases: in the first case, the value of the parameter *a* was assumed to be constant and the value of the parameter *b* was tested with different values. In the second case, the value of the parameter *b* was assumed to be constant and the value of the parameter *a* was tested with different values. The appropriate values for the parameters were obtained when the proposed algorithm reached the minimum value of $\beta_\alpha(n)$. Tables 10 and 11 show the change in $\beta_\alpha(n)$ for different values of the parameters. The results of table 10 and 11 show that the appropriate value for $\beta_\alpha(n)$ was obtained when the parameters *a* and *b* were set at 0.3 and 0.003, respectively.

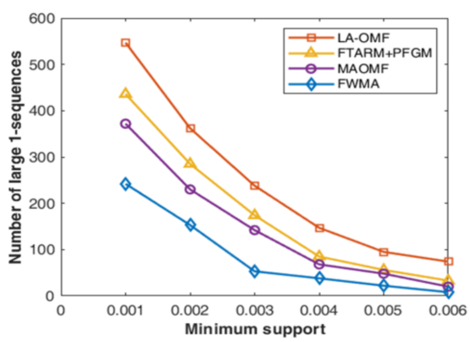

Figure 16. Relationship between the number of large 1-sequences and different minimum supports.

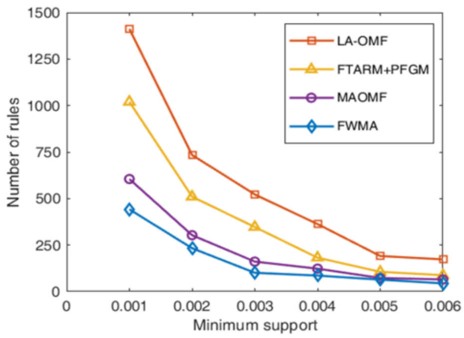

Figure 17. Relationship between the number of rules and different minimum supports.

Table 10. Change of $\beta_\alpha(n)$ to increase *b*.

Reward rate (a)	Penalty rate (b)	$\beta_\alpha(n)$
0.3	0.003	**0.6763**
0.3	0.03	0.7943
0.3	0.3	0.8520

Table 11. Change of $\beta_\alpha(n)$ to increase *a*.

Reward rate (a)	Penalty rate (b)	$\beta_\alpha(n)$
0.1	0.003	0.8663
0.2	0.003	0.8421
0.3	0. 003	**0.6763**

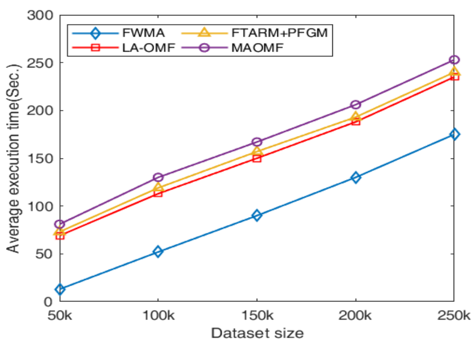

Figure 18. Average execution time with increasing dataset sizes for the CTI dataset.

6. Conclusion

In this work, the time spent by the users on each Web page was considered as a trapezoidal membership function; an LA-based approach called the LA-OMF was proposed to optimize the centers and spreads of each trapezoidal membership function. The proposed LA-OMF algorithm takes the number of TMFs as the input, and optimizes the TMF parameters using a team of LA. This algorithm is composed of two steps. In Step 1, a new representation of LA for each TMF was introduced. In Step 2, we developed a new approach to find both the number and the position of the membership functions for the mining fuzzy association rules in the Web usage data. In the proposed LA-OMF, in order to reduce the search space and eliminate the inappropriate membership functions, two constraints were used, and a new approach was proposed for their implementation. Several experiments were also conducted in order to examine the efficiency of LA-OMF on the datasets with different sizes. The experimental results showed that the proposed algorithm could reach a high fuzzy support, suitable overlap, and coverage. The results obtained also revealed that the proposed algorithm obtained a significantly higher objective function than the other algorithms. In the future, we will focus on two topics: a) enhancing the application of the proposed framework on multi-objective association rule mining and b) using the proposed framework to optimize membership functions in other fuzzy systems.

References

[1] Etzioni, O. (1996). The world wide web: Quagmire or gold mine? Communications of the ACM, vol. 39, no. 11.

[2] Cooley, R., Mobasher, B., and Srivastava, J. (1997). Web Mining: Information and Pattern Discovery on the World Wide Web. In: ictai, pp. 558-567.

[3] Kosala, R., & Blockeel, H. (2000). Web mining research: A survey. ACM Sigkdd Explorations Newsletter, vol. 2, pp. 1-15.

[4] Mobasher, B., Dai, H., Luo, T., Sun, Y., & Zhu, J. (2000). Integrating web usage and content mining for more effective personalization. In: International Conference on Electronic Commerce and Web Technologies: Springer, pp. 165-176.

[5] Cho, Y. H., Kim, J. K., and Kim, S.H. (2002). A personalized recommender system based on web usage mining and decision tree induction. Expert systems with Applications, vol. 23, pp. 329-342.

[6] Eirinaki, M., and Vazirgiannis, M. (2003). Web mining for web personalization. ACM Transactions on Internet Technology (TOIT), vol. 3, pp. 1-27.

[7] Pei, J., Han, J., Mortazavi-Asl, B., & Zhu, H. (2000). Mining access patterns efficiently from web logs. In: Pacific-Asia Conference on Knowledge Discovery and Data Mining: Springer, pp. 396-407.

[8] Castellano, G., Fanelli, A., and Torsello, M. (2007). LODAP: a log data preprocessor for mining web browsing patterns. In: Proceedings of the 6th Conference on 6th WSEAS Int. Conf. on Artificial Intelligence, Knowledge Engineering and Data Bases: Citeseer, pp. 12-17.

[9] Sisodia, D. S., Khandal, V., & Singhal, R. (2018). Fast prediction of web user browsing behaviours using

most interesting patterns. Journal of Information Science, vol. 44, pp. 74-90.

[10] Malarvizhi, S., & Sathiyabhama, B. (2016). Frequent pagesets from web log by enhanced weighted association rule mining. Cluster Computing, vol. 19, pp. 269-277.

[11] Agrawal, R., Imieliński, T., & Swami, A. (1993). Mining association rules between sets of items in large databases. In: Proceedings of the 1993 ACM SIGMOD international conference on Management of data, pp. 207-216.

[12] Agrawal, R., Mannila, H., Srikant, R., Toivonen, H., & Verkamo, A.I. (1996). Fast discovery of association rules. Advances in knowledge discovery and data mining, vol. 12, pp. 307-328.

[13] Zadeh, L. A. (1965). Fuzzy sets. Information and control, vol. 8, pp. 338-353.

[14] Lopez, F. J., Blanco, A., Garcia, F., & Marin, A. (2007). Extracting biological knowledge by fuzzy association rule mining. In: 2007 IEEE International Fuzzy Systems Conference: IEEE, pp. 1-6.

[15] Mamdani, E. H. (1974). Application of fuzzy algorithms for control of simple dynamic plant. In: Proceedings of the institution of electrical engineers: IET, pp. 1585-1588.

[16] Tajbakhsh, A., Rahmati, M., & Mirzaei, A. (2009). Intrusion detection using fuzzy association rules. Applied Soft Computing, vol. 9, pp. 462-469.

[17] Wang, M., Su, X., Liu, F., & Cai, R. (2012). A cancer classification method based on association rules. In: 2012 9th International Conference on Fuzzy Systems and Knowledge Discovery: IEEE, pp. 1094-1098.

[18] Watanabe, T., & Fujioka, R. (2012). Fuzzy association rules mining algorithm based on equivalence redundancy of items. In: 2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC): IEEE, pp. 1960-1965.

[19] Weber, R. (1992). A class of methods for automatic knowledge acquisition. In: Proc. Of the 2nd International Conference on Fuzzy Logic and Neural Networks, 1992.

[20] Kudłacik, P., Porwik, P., & Wesołowski, T. (2016). Fuzzy approach for intrusion detection based on user's commands. Soft Computing, vol. 20, pp. 2705-2719.

[21] Wu, R., Tang, W., & Zhao, R. (2005). Web mining of preferred traversal patterns in fuzzy environments. In: International Workshop on Rough Sets, Fuzzy Sets, Data Mining, and Granular-Soft Computing: Springer, pp. 456-465.

[22] Lin, C. W., & Hong, T. P. (2013). A survey of fuzzy web mining. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, vol. 3, pp. 190-199.

[23] Ansari, Z. A., & Syed, A. S. (2016). Discovery of web usage patterns using fuzzy mountain clustering. International Journal of Business Intelligence and Data Mining, vol. 11, pp. 1-18.

[24] Ansari, Z. A., Sattar, S. A., & Babu, A. V. (2017). A fuzzy neural network based framework to discover user access patterns from web log data. Advances in Data Analysis and Classification, vol. 11, pp. 519-546.

[25] Hong, T.-P., Huang, C.-M., & Horng, S.-J. (2008). Linguistic object-oriented web-usage mining. International journal of approximate reasoning, vol. 48, pp. 47-61.

[26] Hong, T.-P., Chiang, M.-J., & Wang, S.-L. (2002). Mining weighted browsing patterns with linguistic minimum supports. In: IEEE International Conference on Systems, Man and Cybernetics: IEEE, vol. 4, pp. 5-pp. IEEE.

[27] Hong, T.-P., Chiang, M.-J., & Wang, S.-L. (2008). Mining fuzzy weighted browsing patterns from time duration and with linguistic thresholds.

[28] Wang, S.-L., Lo, W.-S., & Hong, T.-P. (2005) Discovery of fuzzy multiple-level Web browsing patterns. In: Classification and Clustering for Knowledge Discovery: Springer, pp. 251-266.

[29] Wu, R. (2010). Mining generalized fuzzy association rules from Web logs. In: 2010 Seventh International Conference on Fuzzy Systems and Knowledge Discovery: IEEE, pp. 2474-2477.

[30] Narendra, K. S., & Thathachar, M. A. (2012). Learning automata: an introduction. Courier Corporation.

[31] Thathachar, M. A., & Sastry, P. S. (2011). Networks of learning automata: Techniques for online stochastic optimization. Springer Science & Business Media.

[32] Thathachar, M. A., & Sastry, P. S. (2002). Varieties of learning automata: an overview. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), vol. 32, pp. 711-722.

[33] Hong, T.-P., Chen, C.-H., Wu, Y.-L., & Lee, Y.-C. (2006). A GA-based fuzzy mining approach to achieve a trade-off between number of rules and suitability of membership functions. Soft Computing, vol. 10, pp. 1091-1101.

[34] Chen, C.-H., Tseng, V. S., &Hong, T.-P. (2008) Cluster-based evaluation in fuzzy-genetic data mining. IEEE transactions on fuzzy systems, vol. 16, pp. 249-262.

[35] Alcalá-Fdez, J., Alcalá, R., Gacto, M. J., & Herrera, F. (2009). Learning the membership function contexts for mining fuzzy association rules by using genetic algorithms. Fuzzy Sets and Systems, vol. 160, pp. 905-921.

[36] Chen, C.-H., Li, Y., Hong, T.-P., Li, Y.-K., & Lu, E.H.-C. (2015). A GA-based approach for mining

membership functions and concept-drift patterns. In: 2015 IEEE Congress on Evolutionary Computation (CEC): IEEE, pp. 2961-2965.

[37] Chen, C.-H., Hong, T.-P., Lee, Y.-C., & Tseng, V.S. (2015). Finding active membership functions for genetic-fuzzy data mining. International Journal of Information Technology & Decision Making, vol. 295, pp. 358-378.

[39] Hong, T.-P., Tung, Y.-F., Wang, S.-L., Wu, M.-T., and Wu, Y.-L. (2009). An ACS-based framework for fuzzy data mining. Expert Systems with Applications, vol. 36, pp. 11844-11852.

[40] Wu, M.-T., Hong, T.-P., & Lee, C.-N. (2012). A continuous ant colony system framework for fuzzy data mining. Soft Computing, vol. 16, pp. 2071-2082.

[41] Ting, C.-K., Liaw, R.-T., Wang, T.-C., & Hong, T.-P. (2018). Mining fuzzy association rules using a memetic algorithm based on structure representation. Memetic Computing, vol. 10, pp. 15-28.

[42] Ting, C.-K., Wang, T.-C., Liaw, R.-T., & Hong, T.-P. (2017). Genetic algorithm with a structure-based representation for genetic-fuzzy data mining. Soft Computing, vol. 21, pp. 2871-2882.

[43] Rudziński, F. (2016). A multi-objective genetic optimization of interpretability-oriented fuzzy rule-based classifiers. Applied Soft Computing, vol. 38, pp. 118-133.

[44] Antonelli, M., Ducange, P., & Marcelloni, F. (2014). A fast and efficient multi-objective evolutionary learning scheme for fuzzy rule-based classifiers. Information Sciences, vol. 283, pp. 36-54.

[45] Minaei-Bidgoli, B., Barmaki, R., & Nasiri, M. (2013). Mining numerical association rules via multi-objective genetic algorithms. Information Sciences, vol. 233, pp. 15-24.

[46] Song, A., Song, J., Ding, X., Xu, G., & Chen, J. (2017). Utilizing bat algorithm to optimize membership functions for fuzzy association rules mining. In: International Conference on Database and Expert Systems Applications: Springer, pp. 496-504.

[47] Chamazi, M.A., & Motameni, H. (2019) Finding suitable membership functions for fuzzy temporal mining problems using fuzzy temporal bees method. Soft Computing, vol. 23, pp. 3501-3518.

[48] Alikhademi, F., & Zainudin, S. (2014). Generating of derivative membership functions for fuzzy association rule mining by Particle Swarm Optimization. In: 2014 International Conference on Computational Science and Technology (ICCST): IEEE, pp. 1-6.

[49] Hong, T.-P., Lee, Y.-C., & Wu, M.-T. (2014). An effective parallel approach for genetic-fuzzy data mining. Expert Systems with Applications, vol. 41, pp. 655-662.

[50] Agrawal, R., Imieliński, T., & Swami, A. (1993). Mining association rules between sets of items in large databases. In: Acm sigmod record: ACM, pp. 207-216.

[51] TSetlin, M., & TSetlin, M. (1973). Automaton theory and modeling of biological systems.

[52] Lakshmivarahan, S. (2012). Learning Algorithms Theory and Applications: Theory and Applications. Springer Science & Business Media.

[53] Meybodi, M., & Lakshmivarahan, S. (1984). On a class of learning algorithms which have a symmetric behavior under success and failure. Lecture Notes in Statistics, Berlin: SpringerVerlag, pp. 145-155.

[54] Meybodi, M.R., & Beigy, H. (2002). New learning automata based algorithms for adaptation of backpropagation algorithm parameters. International Journal of Neural Systems, vol. 12, pp. 45-67.

[55] Ghavipour, M., & Meybodi, M.R. (2018). A streaming sampling algorithm for social activity networks using fixed structure learning automata. Applied Intelligence, vol. 48, pp. 1054-1081.

[56] Narendra, K.S., & Thathachar, M.A. (1980). On the behavior of a learning automaton in a changing environment with application to telephone traffic routing. IEEE Transactions on Systems, Man, and Cybernetics, vol. 10, pp. 262-269.

[57] Anari, B., Torkestani, J. A., & Rahmani, A. M. (2017). Automatic data clustering using continuous action-set learning automata and its application in segmentation of images. Applied Soft Computing, vol. 51, pp. 253-265.

[58] Ghavipour, M., & Meybodi, M.R. (2016). An adaptive fuzzy recommender system based on learning automata. Electronic Commerce Research and Applications, vol. 20, pp. 105-115.

[59] Kumar, N., Lee, J.-H., & Rodrigues, J. J. (2014). Intelligent mobile video surveillance system as a Bayesian coalition game in vehicular sensor networks: Learning automata approach. IEEE Transactions on Intelligent Transportation Systems, vol. 16, pp. 1148-1161.

[60] Helmzadeh, A., & Kouhsari, S. M. (2016). Calibration of erroneous branch parameters utilising learning automata theory. IET Generation, Transmission & Distribution, vol. 10, pp. 3142-3151.

[61]Torkestani, J.A. (2012). An adaptive learning automata-based ranking function discovery algorithm. Journal of intelligent information systems, vol. 39, pp. 441-459.

[62] Morshedlou, H., & Meybodi, M. R. (2014). Decreasing impact of sla violations: a proactive resource allocation approachfor cloud computing environments. IEEE Transactions on Cloud Computing, vol. 2, pp. 156-167.

[63] Rezvanian, A., & Meybodi, M.R. (2010). Tracking extrema in dynamic environments using a

learning automata-based immune algorithm. In: Grid and Distributed Computing, Control and Automation: Springer, pp. 216-225.

[64] Anari, B., Akbari Torkestani, J., & Rahmani, A.M. (2018). A learning automata-based clustering algorithm using ant swarm intelligence. Expert systems, vol. 35, no. 6, e12310.

[65] Hong, T.-P., Chen, C.-H., Lee, Y.-C., & Wu, Y.-L. (2008). Genetic-fuzzy data mining with divide-and-conquer strategy. IEEE Transactions on Evolutionary Computation, vol. 12, pp. 252-265.

[66] Tao, Y.-H., Hong, T.-P., Lin, W.-Y., &Chiu, W.-Y. (2009). A practical extension of web usage mining with intentional browsing data toward usage. Expert Systems with Applications, vol. 36, pp. 3937-3945.

[67] http://www.cs.depaul.edu.

[68] Nosratian, F., Nematzadeh, H., & Motameni, H. (2019). A Technique for improving Web mining using enhanced genetic algorithm. Journal of AI and Data Mining, vol. 7, no. 4, pp. 597-606.

[69] Azimi Kashani, A., Ghanbari, M., & Rahmani, A. M. (2020). Improving performance of opportunistic routing protocol using fuzzy logic for vehicular ad-hoc networks in highways. Journal of AI and Data Mining, vol. 8 , no. 2, pp. 213-226.

[70] Roohollahi, S., Khatibi Bardsiri, A., & Keynia, F. (2020). Using an evaluator fixed structure learning automata in sampling of social networks. Journal of AI and Data Mining, vol. 8, no. 1, pp. 127-148.

[71] Vaghei, Y., & Farshidianfar, A. (2016). Trajectory tracking of under-actuated nonlinear dynamic robots: adaptive fuzzy hierarchical terminal sliding-mode control. Journal of AI and Data Mining, vol. 4, no. 1, pp. 93-102.

[72] Hatamlou, A. R., & Deljavan, M. (2019). Forecasting gold price using data mining techniques by considering new factors. Journal of AI and Data Mining, vol. 7, no. 3, pp. 411-420.

نشریه هوش مصنوعی و داده کاوی

بهینه سازی توابع عضویت با استفاده از اتوماتای یادگیر برای کاوش قوانین انجمنی فازی

زهره اناری[۱]، عبدالرضا حاتم لو[۲*]، بابک اناری[۳] و محمد مصدری[۴]

[۱] گروه مهندسی کامپیوتر و فناوری اطلاعات، دانشگاه پیام نور، صندوق پستی ۴۶۹۷-۱۹۳۹۵ ، تهران، ایران.

[۲] گروه مهندسی کامپیوتر، واحد خوی، دانشگاه آزاد اسلامی، خوی، ایران.

[۳] گروه مهندسی کامپیوتر، واحد شبستر، دانشگاه آزاد اسلامی، شبستر، ایران.

[۴] گروه مهندسی کامپیوتر، واحد ارومیه، دانشگاه آزاد اسلامی، ارومیه، ایران.

ارسال ۲۰۱۹/۰۹/۳۰؛ بازنگری۲۰۲۰/۰۳/۲۵؛ پذیرش ۲۰۲۰/۰۶/۰۶

چکیده:

تراکنش‌ها در داده‌های وب اغلب از داده های کمّی تشکیل شده و نشان می‌دهد که تئوری مجموعه‌های فازی می‌تواند برای نشان دادن چنین داده‌هایی استفاده شود. زمان سپری شده توسط کاربران در هر صفحه وب، یکی از انواع داده‌ی وب بوده که به عنوان یک تابع عضویت ذوزنقه‌ای در نظر گرفته شده و می‌تواند برای ارزیابی رفتار حرکتی کاربران استفاده شود. کیفیت کاوش قوانین انجمنی فازی وابسته به توابع عضویت بوده و از آنجایی که توابع عضویت هر صفحه‌ی وب با سایر صفحات وب متفاوت است، یافتن خودکار تعداد و موقعیت توابع عضویت ذوزنقه‌ای قابل توجه است. در این مقاله، یک روش بهینه سازی مبتنی بر یادگیری تقویتی متفاوت با نام LA-OMF پیشنهاد شده تا تعداد و موقعیت توابع عضویت را برای قوانین انجمنی فازی پیدا کند. در الگوریتم پیشنهادی، مراکز و پهنای توابع عضویت ذوزنقه‌ای به عنوان پارامترهای فضای جستجو در نظر گرفته شده و نمایشی جدید با استفاده از اتوماتای یادگیر برای بهینه‌سازی این پارامترها پیشنهاد شده است. کارایی روش پیشنهادی ارزیابی شده و نتایج به دست آمده با نتایج سایر الگوریتم-های موجود در یک مجموعه داده واقعی مقایسه شده است. آزمایش‌ها بر روی مجموعه‌ی داده‌ها با اندازه‌های مختلف تأیید می‌کند که روش پیشنهادی LA-OMFبا استخراج توابع عضویت بهینه، کارآیی کاوش قوانین انجمنی فازی را بهبود می‌بخشد.

کلمات کلیدی: کاوش استفاده از وب، اتوماتای یادگیر، مجموعه فازی، توابع عضویت، قوانین انجمنی فازی.