Cairo University

Journal of Advanced Research

ORIGINAL ARTICLE

Multi-agents and learning: Implications for Webusage mining

CrossMark

Hewayda M.S. Lotfy [a,*], Soheir M.S. Khamis [a], Maie M. Aboghazalah [b]

[a] *Computer Science, Mathematics Department, Faculty of Science, AinShams University, Cairo, Egypt*
[b] *Pure Math. & Computer Science, Menoufia University, Menoufia, Egypt*

ARTICLE INFO

ABSTRACT

Characterization of user activities is an important issue in the design and maintenance of websites. Server weblog files have abundant information about the user's current interests. This information can be mined and analyzed therefore the administrators may be able to guide the users in their browsing activity so they may obtain relevant information in a shorter span of time to obtain user satisfaction. Web-based technology facilitates the creation of personally meaningful and socially useful knowledge through supportive interactions, communication and collaboration among educators, learners and information. This paper suggests a new methodology based on learning techniques for a Web-based Multiagent-based application to discover the hidden patterns in the user's visited links. It presents a new approach that involves unsupervised, reinforcement learning, and cooperation between agents. It is utilized to discover patterns that represent the user's profiles in a sample website into specific categories of materials using significance percentages. These profiles are used to make recommendations of interesting links and categories to the user. The experimental results of the approach showed successful user pattern recognition, and cooperative learning among agents to obtain user profiles. It indicates that combining different learning algorithms is capable of improving user satisfaction indicated by the percentage of precision, recall, the progressive category weight and F_1-measure.

© 2015 Production and hosting by Elsevier B.V. on behalf of Cairo University.

Introduction

Web user drowns to huge information and faces the problem of being overloaded with information due to the exponential growth for both the number of online available Web applications and the number of their users. This growth has generated huge quantities of data related to user interactions with the Websites, stored by the servers in log files. On the other hand, the degree of personalization that a Website is able to offer in presenting its services to users represents an important attribute contributing to the site's success. Hence, the need for a Website that understands the interests of its users is becoming a fundamental issue. If properly exploited, log files can reveal useful information about user preferences.

Reinforcement learning is the name of a set of algorithms for control systems that automatically improve their behaviors

by trying to maximize the rewards received from an environment. Q-Learning is an example of reinforcement learning. Fuzzy C Means (FCM) is an unsupervised learning technique that became a good candidate method to handle ambiguity in the data, since it enables the creation of overlapping clusters and introduces a degree of item-membership in each cluster. A multi-agent system (MAS) is a system composed of multiple interacting intelligent agents within an environment. MASs can be used to solve problems that are difficult or impossible for an individual agent. There are few related studies regarding utilizing the combination of FCM and Q-learning for MAS in Webusage mining field. Kaya et al. [1] have introduced an approach based on utilizing the mining process for modular cooperative learning systems. It incorporates fuzziness and online analytical processing (OLAP) based mining to effectively process the information reported by agents. A fundamentally different approach have been proposed by Tesauro [2] introduced "Hyper-Q" Learning, in which values of mixed strategies rather than base actions are learned and in which other agents' strategies are estimated from observed actions via Bayesian inference. Tuyls et al. [3] discussed the use of traditional Reinforcement Learning (RL) algorithms in MAS and utilized in games using the replicator equations and dynamical equations. Matignon et al. [4] were interested in learning in MAS especially RL methods, where an agent learns by interacting with its environment, using a scalar reward signal as performance feedback. Li [5] has considered a channel selection scheme without negotiation for multi-user and multi-channel cognitive radio systems. To avoid collision incurred by non-coordination, each user secondary learns how to select channels according to its experience. Multi-agent RL is applied in the framework of Q-learning by considering the opponent secondary users as a part of the environment. Tan [6] has used reinforcement learning to study intelligent agents in which each agent can incrementally learn an efficient decision policy over a state space by trial and error. When the only input from the environment is a delayed scalar reward, the task of each agent is to maximize the long term discounted reward per action.

Web Usage Mining (WUM) can be broadly defined as preprocessing, pattern discovery then the analysis of useful information from the World Wide Web data based on the different emphasis and ways to obtain information. Lakheyan and Kaur [7] have presented a survey on WUM along with its functionalities and FCM algorithm for the retrieval of data from the search engine. Castellano et al. [8] have presented an approach for clustering Website users into different groups to generate common user profiles. These profiles are intended to be used to make recommendations by suggesting interesting links to the user via using FCM and directing users toward the items that best meet their needs and interests. Few works have been reported in Web-based MAS directed approaches integrating FCM and RL. For Example, Taghipour et al. [9] have proposed a novel machine learning perspective toward the problem, based on RL. It models the problem as Q-learning employing concepts and techniques commonly applied in the WUM. Web personalization technology enables the dynamic insertion, customization, or suggestion of content in any format that is relevant to the individual user. Birukov et al. [10] suggested that the Web developer needs to know what the user want and her/his interest to customize the web pages via learning her/his navigational pattern, based on the user's implicit behavior and preferences and explicitly given details. Various approaches have been defined to discover applicative techniques to get higher and corrective recommendations for user surfing. Reddy et al. [11] claimed that the Website structure and the users' profiles may constitute supplementary data for such a process while the Weblog files are the input data in a WUM process. The paper introduces a methodology for Learning in Web-Based Education System (LWBES) in two phases, the FCM to categorize user behavior into user interest category-list and the reinforcement learning to categorize user behavior into user interest link-list inside the category-list. The paper is organized into four sections. The second section introduces the description of the LWBES methodologies, the third section presents the experimental results, its evaluation, and discussion, and finally the conclusion and the future work.

LWBES methodology

A model of the website in which this methodology should be investigated on contains categories of downloadable materials. Each category is represented by collection of materials and each material is represented by a URL. The primary objective of LWBES can be stated as follows. Suppose a set $R = \{R_i| i$ is the number of the webpage R in a category$\}$ of URLs composing a Website and u is a user interactively navigating the Website. The problem is to obtain a personal-list (or recommendation-list) for u, $R_u \subseteq R$, which is a set of URLs that are ranked based on u's interests. In general, to acquire a personal-list for a user, the process goes through four phases which are given in the following:

1. Webusage: Data about user perceptions such as navigation behaviors are collected.
2. Obtaining user insights: Usually this data require further processing for inferring information which is used in the later phases.
3. Ranking the items: The inferred user interests are utilized to provide the predicted user personal-list utilizing offline and online processes.
4. Adjusting user settings: LWBES obtains the resulted navigation behaviors from the user and employs it to refine the user settings based on the user perceptions.

LWBES consists of one interface with two kinds of users which are student and admin. The user logs into LWBES by providing user name and password. The user searches it by entering a keyword and the results of the search are ordered according to two main coordinates based on categories and links. The knowledge base of LWBES is based on a database model that appears as a star schema in which materials are in the center of the graph. The study is centered on the user and materials, therefore the duration in which the user stays in a material Webpage is an important consideration. As user server log file is tracked, the user satisfaction is needed to be captured as the user spends more time in a Webpage which affect the Webpage category weight. Therefore user "satisfaction" can be deduced from the user behavior while surfing. Sen and Weiss [12] presented a useful distinction between requirements for learning about passive components (such as databases), active components (such as agents), and learning about interactive components (such as organizational
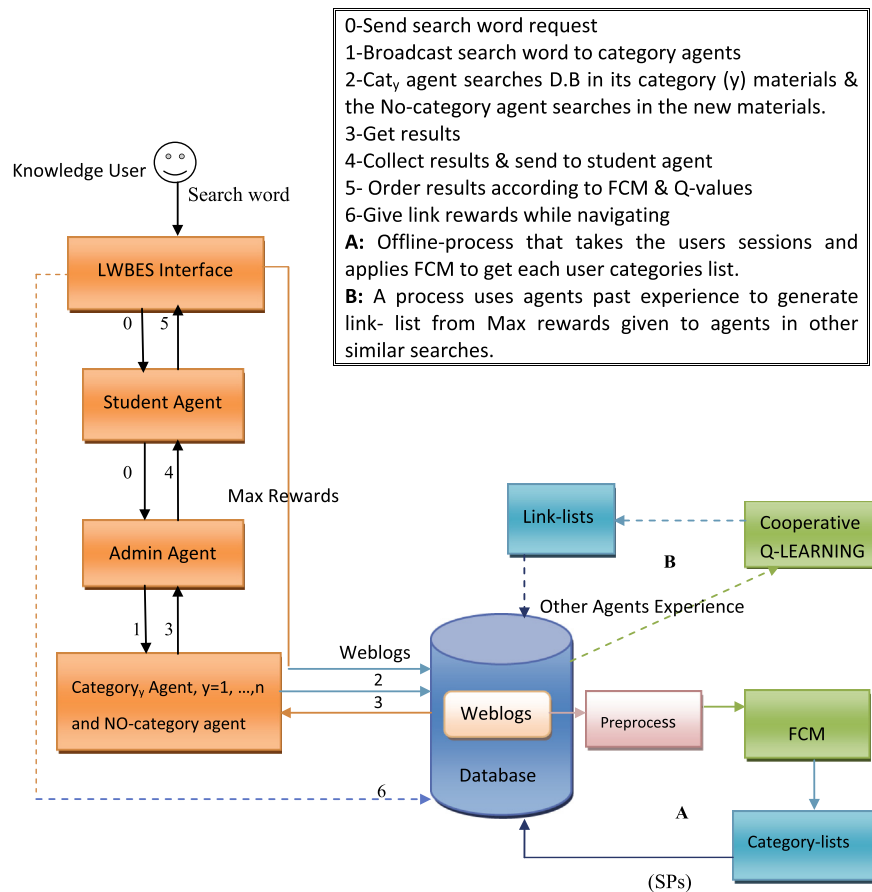
Fig. 1 LWBES general process flow.

structures). A database structure is necessary and more precisely, maybe a data warehouse as its characteristics such as orientation, subject, integration, history, and non-volatility are advantageous. The database dimensions are relations between tables. It can be split into six dimensions where each of the following dimensions is a relation between two tables:

1. Types and Persons. Where "types" is viewed as the types of LWBES users which are student or admin.
2. Category and Materials. Each material has to belong to one category.
3. Materials and Ranks. Stores each material reward which given by the user according to Q-Learning.
4. Persons and Sessions. Related the user with his/her sessions.
5. Materials and Sessions. Related the materials with sessions (which materials are visited in that session).
6. Persons and Materials. Related each user by materials added by that user (admin only can add materials).

The LWBES is a multi-agent recommender that accepts inputs from the users as keywords to search for in the website provided materials. The output is a personal-list consisting of category-list and link-list according to user patterns discovered in the user's previous logs. Fig. 1 illustrates the process flow of LWBES that provide each active user who is using it a satisfactory-and-customized personal-list by analyzing user navigation behaviors. During an offline-process, it clusters

the collected Webusage data from the Weblog and generates the corresponding personal categories-list and the centroid for each navigation-pattern cluster. During the online-process, LWBES maintains rewards for the user URLs to generate personal link-list based on the individual visited URL. If the user is a first user at all then there are no histories for the agents in the database. If any other user or the same user makes another search, then its agent will search first in the history table in the database to minimize the search time and get the most rewarded links in the top of the search results.

The personal list is a consolidation of the category- and link-list. When a user logs into LWBES and starts a search, the application calls the student agent and passes the word to the admin agent which acts as LWBES center agent. Next, the admin agent sends the word to another kind of agents called categories agents and also to an agent that is of kind No-category agent. Each category agent searches in database within the previous searches made by other users' agents for material of the same kind of its category and the No-category agent searches in the new materials that were not visited before for not neglecting the updates of LWBES materials. If it is found, then the category agent selects the most expert agents with the most rewards and at that point each category agent passes its results to the admin agent which collect all results. Therefore, the result is sent to the student agent who orders the results according to materials category based on FCM and material links based on rewards previously given

Table 1 Scheme of interactions between the system actors within the search session.

Actor1	Actor2	Action	Target	Parameters	Tools of communication
User	User-agent	Send request	Resource-links	Searchword	Browser, servlets
User-agent	Admin-agent	Send request	Resource-links	Searchword	Java class method call
Admin-agent	Cat-agent	Send request	Resource-links	Searchword	Java class method call
Cat-agent	D.B.	Search DB	Resource-links	Searchword	Java class method call
Cat-agent	Admin-agent	Send reply	Resource-links	–	Feedback Protocol
Admin-agent	User-agent	Send reply	Resource-links	–	Feedback Protocol
User-agent	User	Send reply	Accepted resource-links	–	Browser, servlets

to the agents. Retrieval results are introduced to the user agent is based on older agents' experience that is fed by these agents' previous searches and their Q-values. If category agents did not find similar searches saved in the database, then the No-category agent search in the materials saved in database and gets the result. A general scheme of interactions and their tools among the system actors within the search session is shown in Table 1. Actor1 communicates to Actor2 performing the communication act Action; Actor1 would like to obtain Target as a result of communication; Actor1 provides Parameters to Actor2; the last column represents tool within the communication act. From this point of view, LWBES has centralized agent architecture similar to as illustrated in Arnoux et al. [13] and is demonstrated in Fig. 2.

Phase one: Webusage data collection and preprocessing

The Weblog files are the input data to a WUM process. Weblog files are obtained from web servers' database which consists of user sessions that describe the user behavior by the most visited links and the time spent in each visit.

Data transformation

According to LWBES, the user accessing a resource link will send a HTTP request to the server that containing this resource, GET http://localhost:8084/JadeWeb/show.jsp?result_id = 26 as an example. Therefore, the server interprets this request, accesses the requested resource and delivers it to the user. As most of the software programs, these operations of all users are saved in database which we call log file of the user. The log file allows us to have a detailed trace of the Web server activity. All the requests made by a single user during the period of browsing constitute the user sessions. In LWBES, a session is split into several navigations where each one represents a single visit to the Web page. The navigation ends when a time threshold of at least 30 min exists between two consecutive requests. Identifying users from the log file is a simple task because each user has user name and password and hence each user has a user ID.

Session identification

For a specific user, the log is investigated and processed to obtain user session. A user session can be defined as a limited set of pages accessed by the same username and password within a particular visit. Assume that the Website is composed of N pages, each page URL is assigned to a unique number

$n = 1, \ldots, N$. Formally, the ith user sessions are represented by a vector $s^i = (s_1^i, s_2^i, \ldots, s_N^i)$. All vectors s^i, $i = 1, \ldots, L$ constitute a feature matrix S of dimension $L \times N$ (where L represents number of sessions) in which each $s_n^i \in s^i$ for $n = 1, \ldots, N$, is defined as:

$$s_n^i = \begin{cases} f_{i,n} \sum_{i,n}^t & user\ visit\ n\text{th}\ url \\ 0 & otherwise \end{cases} \quad (1)$$

where $f_{i,n}$ and $t_{i,n}$ are the access frequency and the total time spent by the user on the nth URL only during the ith session. Furthermore, $f_{i,n} \sum_{i,n}^t$ defines the nth URL weight in session i. Summarizing, after this preprocessing phase, a collection of L sessions is identified from the log data.

Phase two: pattern discovery and recognition

This phase is concerned with obtaining user insights to reach from system input to output and depends on two techniques, FCM that is used to capture user's patterns of behavior and interests by classifying the user's sessions into categories. Then, it presents the search results to the user according to the recognized pattern of behavior. The second technique is RL, where each user agent can get rewards which are saved in the database to be used later by other agents. Webusage clustering for recommendation reduces problem space and increases the efficiency of generating recommendations and filtering based on the distance of the active users' sessions to the centroid of the clustering. An algorithm of the centralized architecture of LWBES is demonstrated in Listing 1. The FCM is an extension of classical C-Means algorithm for fuzzy applications [8]. It uses fuzzy techniques to obtain the fuzzy c-partition and is based on an objective function where a data item may belong to more than one partition which compatible with the status of real data. Once user sessions have been identified, it is arranged in a feature matrix S of size $L \times N$ then a FCM is applied on S in order to group similar sessions in a cluster. Hence, the identified sessions represent the different user profiles that will be successively exploited for suggesting links to pages considered interesting for a current user.

Session categorization by FCM

Given feature matrix $S = \{s_1, \ldots, s_L\}$ which represent the data set, the FCM [14] goal is to partition S into C homogeneous fuzzy clusters by minimizing the objective function T_α using the Euclidean distance metric. The LWBES-FCM algorithm is presented in Listing 2. It starts with an initial guess for the
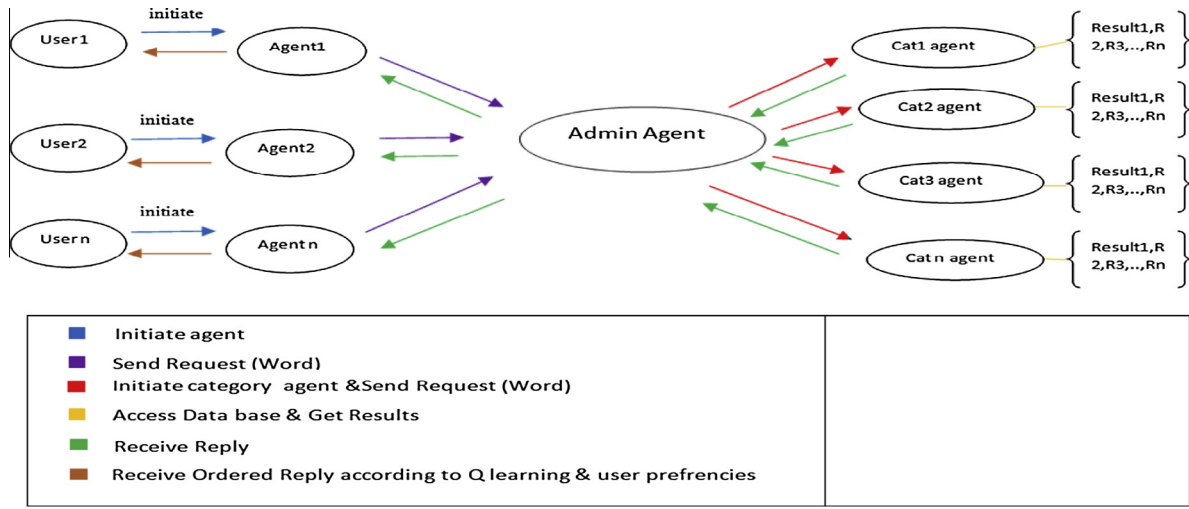
Fig. 2   The centralized view multi-agent architecture.

```
If (user.login == student) then
  load.student.interface
  if (student.search) then
    student-agent (keyword)
    send message to admin-agent (keyword)
    admin-agent divide task on category-agents
      category-agents : DB-search (query.keyword, result.links)
      NO-category-agent : DB-search (query.keyword, result.links)
    Inform (admin, results.links)
    admin-agent send results to student-agent
    student-agent.orderResult (results)
        if (result.links.click) then
          send message to admin-agent (link.id, user.id)
            admin-agent.LWBES-QLearning (link.id, agent.id)
        end if
    student-agent.save (user.id, session)
  end if
end if
If (user.login == admin) then
  admin-agent.LWBES-FCM (S, C, ε, α, J)
  admin-agent.save (user.id.categorized-sessions)
end if
```

Listing 1   LWBES algorithm in a centralized architecture.

cluster centers, which are intended to mark the mean location of each cluster. The initial guess for these cluster centers is most likely incorrect. The FCM assigns every session s^i a membership in each cluster. For each iteration j ($j = 1, \ldots, J$) where J is the number of iterations, it updates the cluster centers and the membership for each s^i as well as moving the cluster centers to the correct location within S. This process is based on minimizing an objective function that represents the distance from any s^i to a cluster center c weighted by s^i membership.

$$T_\alpha = \sum_{i=1}^{L}\sum_{c=1}^{C} m_{ic}^\alpha ||s^i - v_c||^2 \quad 1 \leqslant \alpha < \infty \qquad (2)$$

where L is the number of sessions, $i = 1, \ldots, L$, s^i is a row in S of N-dimension, C is the number of centers and $c = 1, \ldots, C$, m_{ic} is the degree of member ship of session s^i in cluster c, $\alpha > 1$ is a weighting exponent that controls the fuzziness of membership of sessions, v_c is the centroid of cluster c with N dimension, i.e., $v_c = (v_{1c}, \ldots, v_{nc})$ and $||s^i - v_c||^2$ is the Euclidean distance between session s^i and cluster centroid v_c.

Summarizing, the clustering phase mines a collection of C session categories from session data and provides profiles of the users; a general algorithm for LWBES is listed in Listing 2.

Link rewards by reinforcement learning

It refers to a framework for learning optimal decision making from rewards or punishment as has been illustrated in McCallum et al. [15]. It differs from supervised learning in that the learning agent is never told the correct action for a particular state, but is simply told how well or bad the selected

```
LWBES-FCM (S, C, ε, α, J)

  1-Initialize membership matrix M^j at iteration j=0, M^(0) = [m_ic^(0)]_{i=1...L, c=1...C}

  2-At j^th iteration with M^(j) calculate the clusters center vectors V^(j) :

          V^(j) = [v_c^(j)]_{c=1}^{C},  where  v_c^(j) = (Σ_{i=1}^{L}(m_ic^(j))^α s_i) / (Σ_{i=1}^{L}(m_ic^(j))^α)

  3-Update M^(j) with M^(j+1)   m_ic^(j+1) = [Σ_{c=1}^{C}((||s^i - v_c^(j)||^2)/(||s^j - v_c^(j)||^2))^(2/(α-1))]^(-1)

  4-At ||M^(j+1) - M^(j)|| < ε with 0 < ε < 1 STOP;  Otherwise return to step 2 until reach J.
```

Listing 2   LWBES-FCM.

```
Set the γ and β parameters, and webpage links rewards for their actions in table r.

In each training session: Observe the current state, st

LWBES-agentQlearning (γ, β, r, st):

Repeat:

        Initialize the Q-values table, Q(st, a) arbitrarily

        Choose action 'a' (either download, read more, or show video links), for st and execute it

        Observe and receive an immediate  preset reward r(st, a) out of three available actions

        Observe a new state, st' (the new state resulted after action 'a').

        Update the Q-value for the state using the observed reward and the maximum reward

        possible for the next state. The updating is done according to the formula:.

                Q(st, a) ← Q(st, a) + γ*TDerror,

                TDerror = βmax_a' {Q(st', a' )}+ r (st, a) - Q(st, a),

        Set the current state to the new state st'

Until a goal state is reached.
```

Listing 3   LWBES-Qlearning for a (link-id, agent-id) pair.

action was, expressed in the form of a scalar\reward. The reinforcement learning is used to define the optimal behavior of the user agent in order to enforce the user preferences. Q-learning is the most common and well-studied variant of temporal difference learning. Essentially, a table of Q-values is maintained with an entry for each state/action pair. A Q-value, $Q(s, a)$, is an estimate of the expected sum of future rewards that the agent is likely to encounter when starting in state s and initially selecting action a. This sum includes not only the immediate reward signal but also all the other rewards accumulated on the way to the goal state. The purpose of reinforcement learning is to discover these Q-values empirically. If the agent has a complete table, then the agent may interact with the environment optimally by searching through the set of available actions for the current state and selecting the table entry $Q(s, a)$ with the maximum value as have been explained in Kretchmar [16]. The basic algorithm for Q-learning is given by Listing 3, where *TDerror* is the Temporal Difference error which contains an estimate of optimal future value plus the reward observed minus the old value. The learning rate $\gamma \in [0, 1]$ determines to what extent the newly acquired information overrides the old information, while the discount factor $\beta[0, 1]$ is a measure of the importance of future rewards.

The user of LWBES, can download the document in a material URL, see more or see the video of that material. Each choice has a different reward, for example in Fig. 3 if the user choose to download the article then LWBES assigns value x to its own agent and if the user choose to read further then it is assigned another value y. Those values are then saved in database so that later any other user agent can search in the most rewarded results given by other agents. Fulda and Ventura [17] showed many benefits of Multi-agent reinforcement learning systems as they are interesting because they share many benefits of distributed artificial intelligence, including parallel execution, increased autonomy, and simplicity of individual agent design. The Q-learning is a natural choice for studying such systems because of its simplicity and its convergence guarantee.

Phase three: ranking

Ranking is used to obtain the category- and link-lists therefore the search result is based on the resulted category and page

effectiveness from FCM and Q-learning. The category effectiveness in the user profile is measured by estimating user's interest in the cluster. After clustering, the Significance Percentage (SP_{dc}) of a category d in a cluster c which is the ratio of the number of appearance of category to the total number of sessions in the cluster and computed as follows.

$$SP_{dc} = \frac{\sum_{d=1}^{D} occ(d, c)}{\sum_{i=1}^{L} s^i} \quad (3)$$

where D is the number of categories in LWBES and L is the number of sessions in a cluster c. The function $occ(d, c)$ computes the occurrence of category d in cluster c. The category CAT_c is the category with the highest SP in the cluster c and determined by the equation:

$$CAT_c = max(SP_{dc} : 1 \leqslant d \leqslant D) \quad (4)$$

This maximization function is used to recommend the winning category to the user profile in which CAT_c will be in the top of the resulted category-list. In LWBES, the individual page effectiveness in the user profile is measured using Q-learning where each link has three different possible rewards as shown in Fig. 3. The assigned agent of the user is rewarded by LWBES according to user selection. Finally, the most rewarded links by the user appear at the top of the resulted link-list.

Phase four: adjust user settings

After ranking, LWBES saves the SPs in the database then whenever the user searches LWBES the user gets a webpage customized with own preferences according to what LWBES saved before in the database. The user search page should contain resulted link-lists that is categorized where the top category is the most visited as seen in Fig. 4.

Experimental results and discussion

To test the proposed approach for mining usage profiles, a simulation was performed. A sample Website is considered in order to carry out the experiments. The website contains five educational categories (i.e., $D = 5$) which are Database, Network, Management, Data structures, and Economics, each

Title	Computer Network
Brief	Data communications refers to the transmission of this digital data between two or more computers and a computer network or data network is a telecommunications network that allows computers to exchange data. The physical connection between networked computing devices is established using either cable media or wireless media. The best-known computer network is the Internet.
Link	Download
Date	2013-06-06 23:31:17.65
Topic	Read Topic
Video	Show Video

Fig. 3 Resulted search link details and chance to give rewards.

Fig. 4 Keyword search results.

with its uploaded materials. The website installed on a computer with 4 GB of RAM and Core 2 Duo CPU processor. The Java Agent Development framework (JADE) was used which supports the development of complete agent-based applications by means of a run-time environment implementing the life-cycle support features required by agents, JADE is written completely in Java. The programming Language Java and the Java Servlets Pages JSP are used to code LWBES, and SQL Server 2008 is the database engine.

Table 2 shows a session-page view from feature matrix S which is a session profile of a user requests for each page for particular sessions. A row represents a session, every column represents the time of each page that is visited in that session, and each cell represents webpage weight in that session. Each session s^i is modeled as a vector over the N-dimensional space of page views, where $N = 10$. A filtering process is applied to select user sessions to get the mostly visited Web pages and categories. During the experiment, a total number of user sessions $L = 100$ were identified in period of 1 of December 2014 to 15 of the same month. Next, the FCM Algorithm was applied where the number of clusters $C = 5$. The progress of the objective function of FCM clustering is shown in the plot in Fig. 5, it is obvious that after the 30th iteration it receives its

minimum value. Table 3 shows the aggregate usage profiles for 5 clusters under 5 distinct categories of page views URL categories. The categories with highest rate of interest are indicated. If two or more categories have same percentage then it is ordered according to the order of user browsing such as in case of database and data structure. It is noted that some categories for example, category Network has max percentages. Hence, the resulted category-list of this user is stated as:

1. Database
2. Data structure
3. Management
4. Network
5. Economics

Table 4 shows the category visits and frequency of visited categories in the browser window. The 1st column refers to session number, the 2nd column refers to category number visited in that session, and 3rd refers to the page number that the user visited in each category and finally the 4th is the active page visited in seconds. As an example, the user with s^1 opened (4) pages of materials belonging to category with $id = 1$ in the database and opened (2) pages of materials belonging to

Table 2 A sample matrix for user session identification.

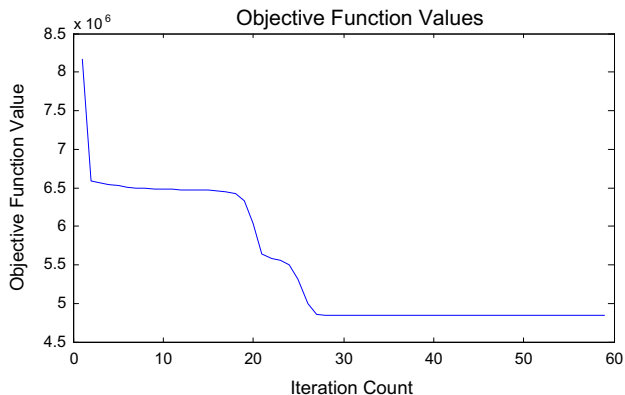|	url1	url2	url3	url4	url5	url6	url7	url8	url9	url10
s^1	62	15	20	13	10	10	17	0	0	5
s^2	26	16	12	13	14	1800	10	90	100	130
s^3	16	0	0	100	30	1800	50	43	17	23
s^4	180	27	20	30	12	15	37	16	20	12
s^5	139	134	16	57	17	145	45	25	17	24

Fig. 5 The progress of the objective function of FCM w.r.t. the number of iterations.

Table 3 The aggregate usage profiles (SP values).

	$c1$	$c2$	$c3$	$c4$	$c5$
Network	1.0	0.0	0.7	1.0	0.9
Database	1.0	0.0	1.0	1.0	1.0
Data structure	1.0	0.0	1.0	1.0	1.0
Management	1.0	0.0	0.8	1.0	1.0
Economics	0.03	0.0	0.5	0.8	1.0

category $id = 4$ in the database. The 4th column shows a window of size six column and four rows of the feature vector S containing session s^1–s^4 of Table 2. It states that the weight of the category is evaluated by the importance of a page in each category in terms of the ratio of the frequency of visits to the category with respect to the overall page visits in the active session. Finally to adjust user settings, categories in the user webpage are ordered according to their SP and the

links in each category are ordered according to each link rank in database the higher rank link shown at the top.

Evaluating metrics for LWBES performance and user satisfaction

An evaluation of LWBES performance in retrieving related results for a fixed keyword of a specific user was an indication of user satisfaction. Some metrics used were the precision, and recall which are defined as follows. Precision is the ratio of the number of relevant records retrieved to the total number of irrelevant and relevant records retrieved. Precision is an important measure of search effectiveness. It is the ability to filter out irrelevant hits and focus on potentially useful information. Recall is the ratio of the number of relevant records retrieved to the total number of relevant records in the database. Recall measures how well a search finds every possible document that could be of interest to the searcher. Both measurements are usually expressed as a percentage. Poor precision damages the reputation of a search system and discourages its use. High precision generally impresses search users and average quality of the recommendation. Recall has less influence on user satisfaction than precision. Many searchers, especially on the Web, are satisfied by precision results, even when recall is low. While these two measures are sometimes conflicting, another metric called *F*-measure [18,19], combines both of them with equal weights. Its general formula (for non-negative real, α) is:

$$F_\alpha = \frac{(1 + \alpha^2) \times Precision \times Recall}{\alpha^2 \times Precision + Recall}$$

When $\alpha = 1$, it is known as F_1 measure and represents the weighted harmonic mean of precision and recall giving equal weights to them where higher values of F_1 indicate a more balanced combination between recall and precision.

$F_1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$.

Table 4 Page visits in the sliding window of size 6.

Session no.	The visited categories	No. of visited pages in category	4 Sessions (s)					
1	Cat 1	4 pages						
1	Cat 4	2 pages						
2	Cat 4	3 pages	62	15	20	13	10	10
2	Cat 1	3 pages	26	16	12	13	14	1800
3	Cat 1	3 pages	16	0	0	100	30	1800
3	Cat 4	1 page	180	27	20	30	12	15
4	Cat 4	3 pages						
4	Cat 1	3 pages						

(a) Average Recall

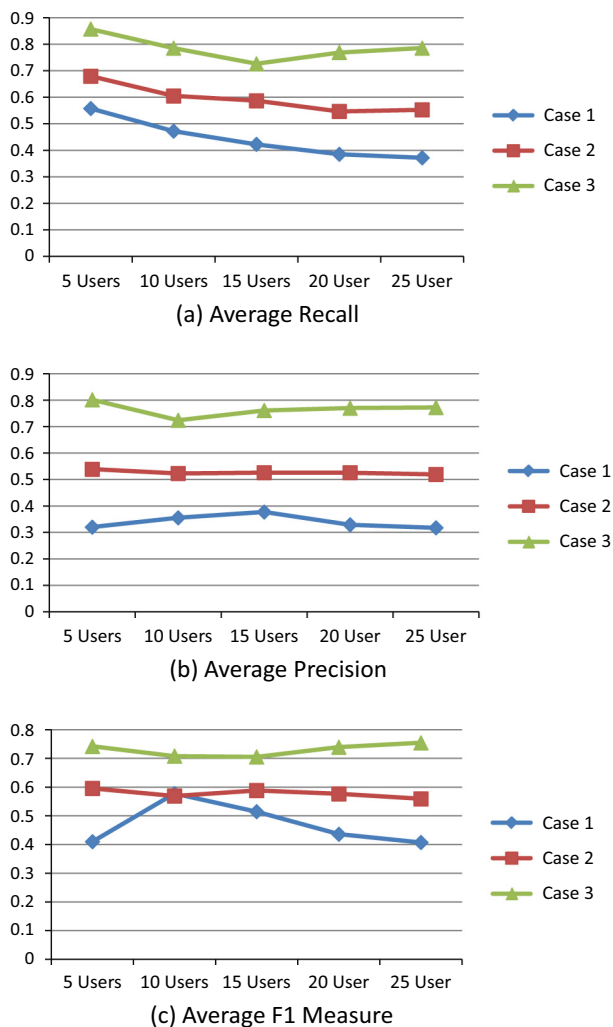(b) Average Precision

(c) Average F1 Measure

Fig. 6 LWBES average precision, recall, and F_1-measure.

For the performance test of LWBES, three test queries were used using the same keyword search and same user. The test queries are generated as follows, the first query (case 1) is performed before the FCM or Q-Learning is ever applied in LWBES. The second query (case 2) is performed when as only the Q-Learning applied. The third query (case 3) is performed after LWBES applied both FCM and Q-Learning. Fig. 6 shows the results of the average of the precision, recall, and F_1-measure for user groups consisting of 5, 10, 15, 20, and 25 users according to the previous considerations and formulas. The precision, recall and F_1-measure curve increased when the author applied Q-Learning only in LWBES and got even further higher values when both FCM and Q-Learning were applied. It is concluded that applying the LWBES approach improves the retrieval quality of the query and hence user satisfaction.

Another measure for user satisfaction is by examining the progressive category weight of the visited categories after recommendation and if the weight increases when applying Q-Learning and FCM it means that the system is successful to satisfy the user. Therefore, 150 randomly chosen sessions of a user were divided into three groups. The first group consisted of 50 sessions were without applying any techniques on them.

The second group consisted of another 50 sessions on which the Q-learning was applied. The third group consisted of 50 sessions as well were examined after applying FCM on the second group. The third group was divided into five groups each consists of 10 sessions, the user behavior is examined in each group. Eq. (1) was used in which $f_{i,n} \sum_{i,n}^{t}$ is considered the weight of nth URL in session s_n^i. Summing weights of all the URL's belonging to that category in session s^i then for all sessions of the group. This is called the progressive category weight, it how frequently the user visits that category in a group of sessions, and how much these visits give that category a weight:

$$ProgressiveCatWeight = \sum_{i=1}^{10} \sum_{n \in Cat} s_n^i$$

Fig. 7 shows that the category weight decreases in the first case in which there are none of our techniques were applied while in the second case, the category weight increased monotonically. In third case when applying the two techniques together the category weight is higher than both cases. This means that the user satisfaction increases by applying the two techniques which means that the system is satisfactory.

Comparison with other approaches

LWBES is a webusage learning system based on combination of FCM, Qlearning, and MAS. It is hard to compare our approach to other approaches since most of them use different measures and methodologies.

Our approach in LWBES relies on FCM as well as the system discussed in Castellano et al. [8]. The main idea of their approach is to cluster the Website users into different groups and generating common user profiles. These profiles are intended to be used to make recommendations by suggesting interesting links to the user. In that approach, by using a fuzzy clustering algorithm, they claim to enable the generation of overlapping clusters that can capture the uncertainty among Web user's navigation behavior. A sample Website was considered in order to carry out the experiments. During the log data preprocessing step, a filtering process is applied to select the mostly visited Web pages. The selected pages are indicated through filtering process by the letters A, B, C, D, E, F, G, H, I and L. In the experiments, the server log files contain the user accesses to the sample Website covering a time period of two weeks. Starting from these data, a total number of 62 user sessions were identified. Next, FCM was applied in order to obtain clusters of users with similar navigational behavior corresponding to the user profiles. Carrying out different tests, the best number of user profiles is determined setting the number of clusters $C = 6$. It was observed that setting a higher number of clusters (i.e., $C = 8$ or $C = 10$) then various prototype vectors with similar values were obtained. This demonstrated that a lower number of clusters were enough to model all the existing profiles.

LWEBS also relies on Q-learning similar as the system mentioned in Taghipour et al. [9] which shows that the reinforcement learning paradigm is an appropriate model for the recommendation problem from a framework in which the system constantly interacts with the user and learns from the user behavior. The data set is log data from web traffic
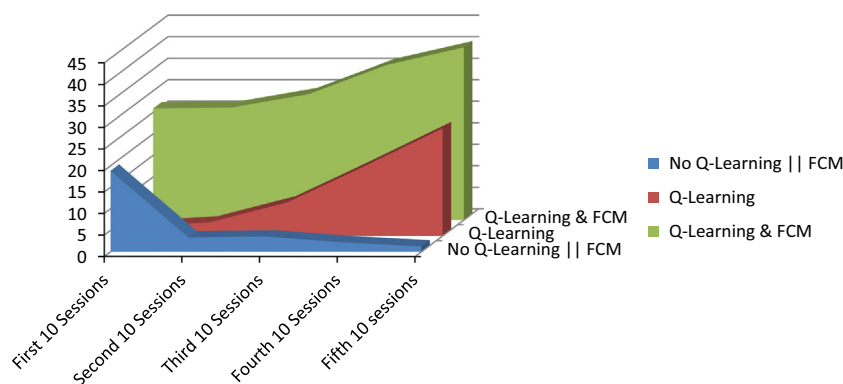
Fig. 7 Progressive category weight measure.

simulator containing 700 pages. User Sessions were of length 5 where 70% of data were used as training set and the rest is used to test the system. Their experiments varied the window size of user sessions and showed that the result is sensitive to it and best result achieved with sessions of window size 3. Their system achieves maximum 80% accuracy and 60% shortcut gain. LWBES also uses Q-Learning to rank links according to reward given by the users which discussed with diffusion in section ''Link Rewards by Reinforcement Learning''. This situation is actually compared to case 2 of our experimental results where precision ranged from 70% to 80%, recall from 70% to 90% for 25 users, and F_1-measure ranges from 70% to 80%.

LWBES goal is similar to the system mentioned in Birukov et al. [10] which is an agent-based recommendation system for supporting communities of people in searching the web by means of a popular search engine. Agents use data mining techniques in order to learn and discover users' behaviors, and interact with each other to share knowledge about their corresponding users. LWBES and this system face the fact that the increase in number of agents increases the system effectiveness. After computing precision and recall of the links proposed by the agents, it is noted that the increase of community members causes the increase of the agents' recall. It is probably conditioned by the fact that having more agents, means having more interactions among them. The agents provide each other only one link then with the growth of the number of links provided by the agents during the search, there is an increase of the percentage of relevant links proposed by the agents and therefore increase of recall. Precision ranges from 0.63 to 0.75 and the value of recall ranges from 0.09 to 0.23. Those three systems individually share the base techniques of LWBES. There is no such system that follows the approach of combining these different methods in Webusage mining.

Conclusions

Web server logs have abundant information about the nature of users accessing it. The analysis of the user current interest based on the navigational behavior may help societies to guide the users in their browsing activity and obtain information in a shorter span of time. In this paper, the new approach of LWBES that first takes the concept of cooperative agents

which gave higher results than an individual agent. Second it uses FCM for clustering user sessions in order to divide users' interests into categories. Third, it uses Q-Learning to order the category links according to rewards given by user to its own agent so that other or new agents can use those agents history to give more related links to users. LWBES helps users to get their preferred categories and favored links in short time and accurately. Based on experimental results and the evaluation of the application, it shows a high percentage for precision, recall, F_1-measure and the progressive category weight of query retrieval which provides more confidence in the system hence better user satisfaction. In the future work, additional learning techniques can be applied that may lead to even better LWBES performance. Another addition to this approach is to use intervened reinforcement learning and FCM which can be obtained by adding the rewards as a part of the feature vector S.

Conflict of Interest

The authors have declared no conflict of interest.

Compliance with Ethics Requirements

This article does not contain any studies with human or animal subjects.

References

[1] Kaya M, Alhajj R. Fuzzy OLAP association rules mining-based modular reinforcement learning approach for multi-agent system. IEEE Trans Syst Man Cybern—B: Cybern 2005;35(2).

[2] Tesauro G. Extending Q-learning to general adaptive multi-agent systems. neural information processing systems conference; 2003.

[3] Tuyls K, Verbeeck K, Lenaerts T. A selection-mutation model for Q learning in multi-agent systems. In: Proceedings of the 2nd international joint conference on Autonomous agents and multiagent systems Proceeding (AAMAS03); 2003. p. 693–700 [ACM 1-58113-683].

[4] Matignon L, Laurent GJ, Le Fort-Piat N. Hysteretic Q-learning: an algorithm for decentralized reinforcement learning in cooperative multi-agent team. In: Intelligent robots and systems (IROS 07) IEEE/RSJ international conference; 2007.

[5] Li H. Multi-agent Q-learning of channel selection in multi-user cognitive radio systems: a two by two case. In: IEEE conference on systems, man, and cybernetics, San Antonio, Texas, USA; 2009.

[6] Tan M. Multi-agent reinforcement independent vs. cooperative. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.; 1998.

[7] Lakheyan C, Kaur U. A survey on webusage mining with fuzzy C-means clustering algorithm. Int J Comput Sci Mobile Comput (IJCSMC) 2013;2(4):160–3.

[8] Castellano G, Fanelli AM, Torsello MA. Mining usage profiles from access data using fuzzy clustering. In: The 6th WSEAS international conference on simulation, modelling and optimization, Portugal; 2006.

[9] Taghipour N, Kardan A, Ghidary SS. Usage-based web recommendations: a reinforcement learning approach. In: Proceedings of the ACM conference on recommender systems; 2007.

[10] Birukov A, Blanzieri E, Giorgini P. Implicit: a multi-agent recommendation system for web search. J Auton Agents Multi-Agent Syst 2012;24(1).

[11] Reddy SK, Reddy MK, Sitaramulu V. An effective data preprocessing method for web usage mining. In: International conference on information communication and embedded systems (ICICES). IEEE Publisher; 2013. p. 7–10.

[12] Sen S, Weiss G. Learning in multi-agent systems. In: MultiAgent systems. MIT Press; 1999. p. 259–98 [chapter 6].

[13] Arnoux M, Lechevallier Y, Tanasa D, Trousse B, Verde R. Automatic clustering for the web usage mining. In: Mirton Timisoara, editor. Proceedings of the 5th international workshop on symbolic and numeric algorithms for scientific computation (SYNASC03); 2003. p. 54–66.

[14] Abdelghaffar Nashwa M, Lotfy Hewayda MS, Khamis Soheir M. A multi-agent-based approach for fuzzy clustering of large image data. J Real-Time Image Process 2014.

[15] McCallum AK, Nigam K, Rennie J, Seymore K. Automating the construction of internet portals with machine learning. Inform Retrieval J 2000;3:127–63.

[16] Kretchmar RM. Reinforcement learning algorithms for homogenous multi-agent systems. In: The 49th IEEE international Midwest symposium on circuits and systems (MWSCAS '06); 2006.

[17] Fulda N, Ventura D. Predicting and preventing coordination problems in cooperative Q-learning systems. In: Proceedings of the international joint conference on artificial intelligence (IJCAI); 2007. p. 780–5.

[18] Bedi P, Sharma R, Kaur H. Recommender system based on collaborative behavior of ants. J Artif Intell 2009:40–55, ISSN: 1994-5450.

[19] Nadi Shiva, Saraee Mohammad Hossein, Bagheri Ayoub. A hybrid recommender system for dynamic web users. Int J Multimedia Image Process 2011;1(1).