

A Hybrid Web Recommender System Based on Cellular Learning Automata

Mojdeh Talabeigi

Department of Computer Engineering
Islamic Azad University, Qazvin Branch
Qazvin, Iran
Mojde.talabeigi@gmail.com

Rana Forsati

Electrical & Computer Engineering
Dept., Shahid Beheshti University
Tehran, Iran
r_forsati@sbu.ac.ir

Mohammad Reza Meybodi

Department of Computer Engineering,
Amirkabir University of Technology
Tehran, Iran
mmeybodi@aut.ac.ir

Abstract-With the rapid growth of the World Wide Web (WWW), finding useful information from the Internet has become a critical issue. Web recommender systems help users make decisions in this complex information space where the volume of information available to them is huge. Recently, a number of web page recommender systems have been developed to anticipate the information needs of on-line users and provide them with recommendations to facilitate and personalize their navigation. Recent studies show that a web usage recommender system which focuses solely on access history has some problems because sometimes this information is incomplete or incorrect. One common solution to this problem is to incorporate some semantic knowledge about pages being recommended into system. In this paper we exploit this idea to improve the dynamic web recommender system which primarily devised for web recommendation based on web usage and structure data. We propose a hybrid web page recommender system based on asynchronous cellular learning automata with multiple learning automata in each cell which try to identify user's multiple information needs and then assist them to recommend pages to users. The proposed system use web usage data, content and structure of the web site to learn user information needs and predicting user's future requests. Our experiments show that incorporating conceptual relationship of pages with usage data can significantly enhance the quality of recommendations.

Keywords- Asynchronous cellular learning automata, web recommendation, web mining.

I. INTRODUCTION

With the explosive growth of knowledge available on the World Wide Web, which lacks an integrated structure or schema, it becomes much more difficult for users to access relevant information efficiently. Most web users complain about finding useful information on web sites. Recommender systems have been introduced to solve these problems. Web page recommender systems predict the information needs of users and provide them with recommendations to facilitate their navigation. Given a user's current actions, the goal is to determine which web pages will be accessed next. Many web sites on Internet use web page recommender systems to increase their usability and user satisfaction.

One research area that has recently contributed greatly to this problem is web mining. Recently, there has been an increasing interest in applying web usage mining techniques to build web recommender systems [1,2,3,4,5,10,22,23]. Web usage recommender systems take web server access logs as input, and make use of data mining techniques such as association rule and clustering to extract navigational patterns,

which are then used to provide recommendations. Web server access logs record user browsing history, which contains plenty of hidden information regarding users and their navigation. In web usage recommender systems, navigational patterns are generally derived as an off-line process.

A web usage recommender system which focuses solely on access history has its own problems. One restriction with web server logs is that the information in them is very limited. Thus, a number of heuristic assumptions have to be made to identify individual users, visit sessions and transactions in order to apply any data mining algorithm. One such assumption is that user information needs are fulfilled sequentially while in practice they are often in parallel. Pursuing more than one goal simultaneously (i.e. concurrent information needs) is fairly common for on-line users, but this fact has so far been ignored by web usage-based recommender systems and the research community [6]. Another problem is that when web site visitors are lost, the clicks made by them are recorded in the log, and may mislead future recommendations. This becomes more problematic when a web site is badly designed and more people end up visiting unsolicited pages, making them seem popular.

One common solution to this problem is exploiting other source of information such as content of web pages for recommendation process. According to the definitions given in [7], there are at least two ways to integrate content features of web pages into usage mining results: pre-mining integration, which involves the transformation of normal user access sessions into “content-enhanced” sessions containing the semantic features of the web page contents, and post-mining integration, which denotes performing usage mining and content mining independently and then combining their mining results. In [8], both usage and content attributes of a site were integrated into a web mining framework for web personalization. A “post-mining” type approach was implemented to obtain the uniform representation for both site usage and site content profiles to facilitate the real-time personalization. However, the techniques proposed in [8] were limited to the use of clustering to separately build site usage and content profiles.

An experimental system was designed in [9] to investigate combined mining of both web logs and web contents. The methods of document clustering are first applied to the website contents for grouping web pages into a certain number of clusters. Then, the representative information of page clusters is integrated into original web log entries as the content indicator. Finally, the association rule mining

algorithm is applied to the "content-enhanced" data source. The system demonstrates that novel association rules can be discovered from the integrated log data, while they could not be mined from original log entries. Furthermore, Li et al. [6] proposed a web recommendation system which combines not only usage data, content data, but structure data in a web site to generate user navigational models. These models are then fed back into the system to recommend users shortcuts or page resources. Similar to the method in [9], normal user access sessions are first transferred into so-called "missions", namely "content-enhanced" sessions. The structure data is later used to provide connectivity information among web pages to improve the navigational patterns obtained by performing web usage mining on derived "missions". Preliminary experiments in the paper prove that the system can provide recommendations of good quality.

In this paper, we made use of the conceptual relationships among web page to enhance the Asynchronous Cellular Learning automata (ACLA) based model which we have proposed to dynamically recommend pages to users [15]. Since we aim to design a dynamic system which does not have any offline process, we do not need to transform the original log. In this paper, we use another way for associating different channel of information. Similar to [6], we use content of web pages to identify concurrent information needs of a user in a session. In other word, we made use of the conceptual relationships among web page and derived a novel ACLA based model, enriched with semantic knowledge about the usage behavior. Our hybrid model for the web page recommendation problem shows the flexibility of the CLA for the recommendation problem and how it can be extended to incorporate other sources of information. We evaluate our method under different settings and show how this method can improve the overall quality of web recommendations.

The organization of the paper is as follows: in section 2, asynchronous cellular learning automata with multiple learning automata in each cell and our previous ACLA based model for recommendation problem are briefly introduced. Section 3 describes our proposed system for improving recommendation process. The overall performance of the system is evaluated in section 4. Finally, section 5 summarizes the paper and introduces future work.

II. BACKGROUND

In this section, we will introduce asynchronous cellular learning automata with multiple learning automata in each cell and our dynamic web recommender system based on simple asynchronous cellular learning automata.

A. Asynchronous Cellular Learning Automata with multiple LA

Cellular Learning Automata (CLA), which is a combination of Cellular Automata (CA) and Learning Automata (LA), is a powerful mathematical model for many decentralized problems and phenomena. CLA is a mathematical model for dynamical complex systems that

consist of a large number of simple components. The simple components which have learning capability, act together to produce complex emergent global behavior. The basic idea of CLA is to use LA to adjust the state transition probability of stochastic CA. A simple CLA is a CA in which a learning automaton is assigned to every cell. For more information about CLA the reader may refer to [11,12]. A CLA is called synchronous if all LAs are activated at the same time in parallel. A CLA is called asynchronous if at a given time only some LAs are activated independently from each other, rather than all together in parallel [13].

All previously mentioned models of CLA use one LA per cell. In some applications, there is a need for a model of CLA in which each cell is equipped with several LAs, for instance, the channel assignment in cellular mobile networks for which we need to have several decision variables, each of which can be adapted by a LA [14]. Such a CLA is called *CLA with multiple LAs in each cell*. In this model of CLA, LAs may be activated synchronously or asynchronously.

In ACLA with multiple LAs in each cell, several LAs are assigned to each cell of CLA, which are activated asynchronously. Without loss of generality and for the sake of simplicity, assume that each cell contains s LA. Formally, an d -dimensional step-driven ACLA with n cells is a structure $(Zd, \Phi, A, N, F, \rho)$, where Zd is a lattice of d -tuples of integer numbers, Φ is a finite set of states, A is the set of LAs assigned to cells, $N = \{x_1, x_2, \dots, x_m\}$ is the neighborhood vector, $F : \Phi^s \times m \rightarrow \beta$ is the local rule, and ρ is an $n \times s$ -dimensional vector called activation probability vector, where ρ_{ij} is the probability that the LA j in cell i (for $i = 1, \dots, n$ and $j = 1, \dots, s$) is to be activated in each stage.

The operation of an ACLA with multiple LAs in each cell can be described as follows: At iteration k , each LA A_{ij} is activated with probability ρ_{ij} , and the activated LAs choose one of their actions. The activated automata use their current actions to execute the rule (computing the reinforcement signal). The actions of neighboring cells of an activated cell are their most recently selected actions. The reinforcement signal is produced by the application of local rule. Finally, activated LAs update their action probability vectors, and the process repeats [14].

B. Dynamic Web Recommender System based on Web Usage Data

In [15], we propose a dynamic web recommendation system which we believe is suitable to the nature of recommendation problem and has some intrinsic advantages over previous dynamic method. We use an asynchronous cellular learning automaton for learning user navigation behavior and predicting useful and interesting pages.

We represent each web page by a cell and equip each cell with learning automata. Each cell in proposed ACLA has $n-1$ neighbors and each learning automaton has $n-1$ actions where n is the number of web pages. The task of each learning automata is to learn how making better recommendation. Obviously the goal of the system would be to make the most successful recommendations.

Using the notions of N-Grams, knowing only the last w page visits of the user, gives us enough information to predict his future page requests. We use a sliding window of size w on user's visited pages called "visit window", resulting in considering only the last w pages requested by the user. Also we put a sliding window of size w' on recommended pages called "recommendation window", allows only the last w' pages recommended by the system, have influence on rewarding or penalizing in each step.

For every user session in the log file, system begins with the first page and activates the cell corresponding to the page and its residing LA. Active LA recommends a page to user by choosing one of its actions with respect to its probability vector. After choosing an action, then the cell corresponds to the next page of the session and its residing LA will be activated.

In each activated cell, the visit and recommendation window will be compared and if they have similar page, the LA of the cell which has been chosen that page will be rewarded. In other word, an action should be rewarded if it recommends a page that will be visited in one of the w' next step, of course not necessarily the immediate next step, otherwise it should be penalized. The action probability vector of the learning automata is updated according to L_{ReP} learning algorithm with time varying learning parameters a and b [15]. We try to use all factors which have influence on quality of recommendations in rewarding and penalizing functions. One of these factors is importance of page which is calculated according to the time which user has spent on page p and frequency of page p in user sessions [16].

III. HYBRID WEB RECOMMENDER SYSTEM BASED ON ACLA WITH MULTIPLE LA IN EACH CELL

In this work, we have designed an experimental system to assist our investigation on whether using content of web pages with usage data could result in a more accurate prediction of users' future requests. In this section, we present a hybrid web page recommendation system based on asynchronous cellular learning automata with multiple learning automata in each cell. The proposed algorithm employs the web usage data, content of page and underlying site structure for recommending pages to the current user.

We extend the previous ACLA based system by extracting concepts of a page and assigning learning automata to each concept. The learning automata is used to learn the importance of the concept of a page to users, and try to give more consideration to topics which are more interesting and useful for the user, in order to identify the user's information need more precisely and recommend more useful pages to the user. Similar to previous ACLA based model, the proposed system is also a dynamic system which can easily adapt itself to change in website structure and content and new trend in user behavior.

In the following, we present a hybrid web recommender system based on ACLA with multiple LA in each cell. In this algorithm, a website with n page and m concept or topic is

modeled with an ACLA with n cells, where each cell is equipped with s LAs of L_{ReP} type. So we represent each web page by a cell and each concept of a page with a LA. Each cell in proposed ACLA has $n-1$ neighbors where n is the number of web pages. In each cell, the k th LA specifies that the page corresponding to that cell contains k th concept. The action set of LA j reside in cell i is pages which are in the cluster related to concept j . The task of each learning automata corresponds to concept j is to learn how making better recommendation which fulfill information needs correspond to concept j . Obviously the goal of the system would be to make the most successful recommendations related to user's information needs. In what follows, we study how the proposed system is mapped to ACLA with multiple LAs in each cell.

A. Concept Extraction and clustering

In order to determine the concurrent information need of users, we must extract the concept of pages. In other word, first we must identify web pages contain which concepts or topics. Then by using access log we can detect users are looking for what information when they traverse web pages. Concept extraction consists of two components, keyword extraction and word sense disambiguation. Keyword extraction module identifies the most important words and phrases in the page. Keyword extraction is the task of automatic identification of keywords in a page. The task of word sense disambiguation module is ascribing the relevant concept to each keyword. In order to map keywords to concepts (Wikipedia article titles), we must first disambiguate them in the context. Word sense disambiguation is the task of automatic assignment of the most appropriate sense to a polysemous word in the page. We use Wikipedia disambiguation pages for this task. After the disambiguation phase each keyword in the page is mapped to its corresponding Wikipedia article.

After the process of concept extraction, each page of the web site is transformed into a vector of most pertinent Wikipedia articles, each having a weight expressing its importance in the page. The importance of each concept in a page is obtained using well known TF-IDF measure:

$$w(\text{page}, \text{concept}) = \text{freq} \times \log_2 \frac{|W|}{df_{\text{concept}}} \quad (1)$$

Freq is the frequency of concept in page. $|W|$ is total number of pages in the web site. df_{concept} is the number of pages containing concept. As a result, each page P_i can be represented with a vector of concept-weight pairs:

$$P_i = \{ < c_{il}, w_{il} >, \dots, < c_{in_i}, w_{in_i} > \} \quad (2)$$

For determining the action set of each LA, we must cluster web pages based on their content and obtain semantic clusters. For this purpose, we use Hierarchical Unsupervised Fuzzy Clustering (HUFC). Fuzzy clustering algorithms [18] are effective methods for exploring the structure of complex

real data where grouping of overlapping and vague elements is necessary [17].

B. Activation Vector

After extracting the concepts and clustering of web pages in the hierarchy, we represent each concept of a page with learning automata. The activation probability vector of ACLA is obtained by taking expectation from the product of an n -dimensional vector π_1 and an $n \times nm$ -dimensional matrix π_2 . Vector π_1 is called *cell activation vector* and determines when a given cell is activated [14]. It is apparent that, when the page corresponds to a cell i is visited by current user, it will be activated, i.e., $\pi_1(i) = 1$. Thus, $E[\pi_1(i)]$ is the probability that the page corresponds to a cell i is visited by current user. Matrix π_2 is called *LA activation matrix* and determines when an LA in an activated cell is triggered. Element $\pi_2(i, j)$ becomes 1 when the j th LA in cell i is activated [14]. Thus, $E[\pi_2(i, j)|\pi_1(i) = 1]$ is equal to the probability of triggering the j th LA in cell i given that the page corresponds to cell i is visited in order to fulfill information need corresponds to concept j . The operation of this system can be described as follows: When current user visits page i , Cell i will be activated. Then some of its residing LAs will be triggered according to Matrix π_2 . Vector π_1 is determined by page navigation rate, while matrix π_2 is obtained from similarity matrix which is described hereinafter.

Similarity matrix which contains the similarity of nodes in a concept hierarchy proposed in [19]. In this method a probability $p(c)$ is assigned to each concept node c which is proportional to the frequency of pages belonging to this node and its descendants in user sessions. The information content of each node is then defined as:

$$I(c) = -\log p(c) \quad (3)$$

Then a set LCA is found which contain the least common ancestors, those occurring at the deepest level, of the pair of concept nodes and the similarity score between those are computed as:

$$\text{Sim}(c_1, c_2) = \max_{a \in \text{LCA}} \{I(a)\} \quad (4)$$

When a page is visited by a user, the cell correspond to that page will be activated. Then the LAs correspond to similar concept of this page and concept of visit window's pages will be triggered based on similarity matrix of concepts. In other word, system identifies the interesting concepts for users in visit window and recommends pages related to these concepts. If active cell and the visited pages do not have any similar concepts, then all LA residing in active cell will be triggered.

C. Reward and penalty Parameter

After triggering of learning automaton of a cell, Each LA must choose one of its actions. In other word, LA must recommend a page to user by choosing one of its actions with

respect to its probability vector. At beginning of training phase, the probabilities of all actions are equal, so LA chooses an action randomly. But after that the probability vector change according to reinforcement signal and LA chooses the action with highest probability. In this case, we represent each concept of a page with a LA, so when LA choose an action it is equal to recommend a page which can fulfill the information need related to its corresponding concept. Since we equip each cell with more than one LA, each cell has ability to recommend more than one page. The recommended pages are arranged according to the probability of corresponding actions. After choosing an action by each LA, then the action is rewarded or penalized like the previous dynamic web recommender based on simple ACLA. For enhancing the quality of recommendations, we decide to make use of content information of web pages in rewarding function. If concept corresponds to LA which makes a correct recommendation have larger weight in a page, then we must reward that action more. Because that concept is more important in the page and the pages related to selected actions are arranged according to their probabilities.

In this system, also the action probability vector of the learning automaton is updated according to L_{RP} learning algorithm with time varying learning parameters a and b . In order to adapting to this case, we just change the parameter a and use importance of concept instead of importance of page.

$$\text{importance}(c) = \frac{w(p, cn)}{\max\{w(p, c)\}} \times \text{duration} \times \text{frequency} \quad (5)$$

Where $w(p, cn)$ is the weight of concept n in page p and $\max\{w(p, c)\}$ is the largest weight of a concept on that page.

The action probability vector of the learning automaton is updated and ACLA continues its work until the end of access log.

IV. EXPERIMENTAL EVALUATION

In this section we present a set of experiments that we performed for evaluating the impact of our proposed technique on the prediction process. Our overall experiments have verified the effective of our proposed techniques in web page recommendation.

In [21], Lui and his colleagues have modeled the regularities of users behavior in the web by an agent based model and validated their model by usage data of several large sites such as Microsoft. This model provides an environment containing web pages and users. It can be used instead of using real web pages and web usage data. The advantage of using this model is that identification of users and their page visits are more precise and no data analysis is required. However, the parameters which are introduced in this model should be set carefully so that the result will be similar to real environment. The log files were simulated for a website containing 2000 web pages and 10 topics. We pruned user sessions with a length smaller than 5 and were provided with 35000 user sessions with average length of 8. We split the data sets in two non-overlapping time windows to form training and a test data set. 60% of the data set was used as the training

set and the remaining was used to test the system. For our evaluation we presented each user session to the system, and recorded the recommendations it made after seeing each page the user had visited. As noted before, this system is designed so that it can learn in both training and testing phase.

A. Evaluation metrics

In order to evaluate the recommendation effectiveness for our method, we measured the performance of proposed method using two different standard measures, namely, Accuracy, Coverage [10]. Recommendation accuracy and coverage are two metrics quite similar to the precision and recall metrics commonly used in information retrieval literature. Recommendation accuracy measures the ratio of correct, where correct recommendations are the ones that appear in the remaining of the user session. Recommendation coverage on the other hand shows the ratio of the pages in the user session that the system is able to predict before the user visits them.

As is the case with precision and recall, these metrics can be useful indicators of the system performance only when used in accordance to each other and lose their credibility when used individually. As an example, consider a system that recommends all the pages in each step, this system will gain 100% coverage, of course in the price of very low accuracy.

B. Experimental results

We evaluate our method under different settings. In all experiments we measured accuracy against coverage. The first experiments were performed to evaluate system sensitivity to the size of visit window and recommendation window. We show the affect of them on efficiency of our system.

To consider the impact of visit window's size (w), we vary visit window's size from 1 to 4 while the size of recommendation window is constant and equal to 3 (similar to previous ACLA based system's experiment). Fig. 1 shows the accuracy of the system against its coverage while the size of visit window is changed.

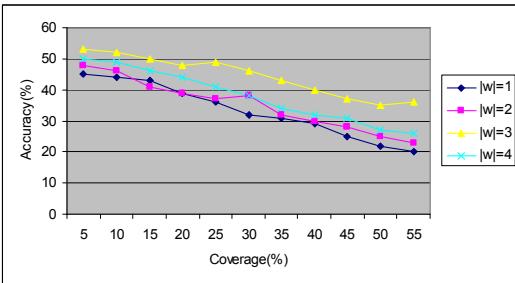


Figure 1. System performance with various visited window size

Our evaluations indicate the best performances are achieved when using window sizes of 3. This is due to the fact that smaller values of w keep insufficient information about navigation history and larger values of w keep un-necessary information regard to previous information need and trend of users. In this case the accuracy is relatively low. It seems to be due to the fact that in new ACLA based model, each cell of ACLA have more than one LA. So each cell can recommend

more than one page in each step. In order to improve accuracy, we think it is better to choose a larger size for recommendation window.

So this time, we vary recommendation window's size and consider visit window's size is constant and equal to 3. Fig.2 shows the result of this experiment. The results show that accuracy increases when we consider the system recommendations of neighbor cells. But when we used recommendation window size larger than 5 results in lower accuracy, it probably occurs because in each step system recommends one or two pages in average and the recommendations provided by the system can be useful for the user only few next step. After few step the information needs of the user will be changed.

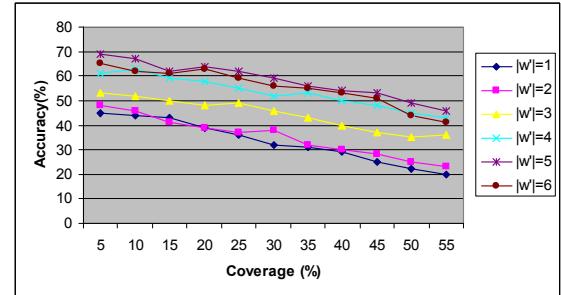


Figure 2. System performance with various recommendation window sizes

In the last part of our experiment, we consider performance of our system in comparison with previous ACLA based model [15] and traditional association rule mining approach [4]. Fig.3 has shown the comparison of our system's performance with ACLA based and AR method in the sense of their accuracy against coverage. In this experiment, we consider recommendation window size is fixed and equal to 5. As the coverage increases, naturally accuracy decreases in all systems, but it can be seen the rate in which accuracy decreases in new system is lower than previous ACLA based model. It seems due to the fact that the system is designed so that it can identify information needs and trends of users and recommend more useful pages which fulfill this information needs.

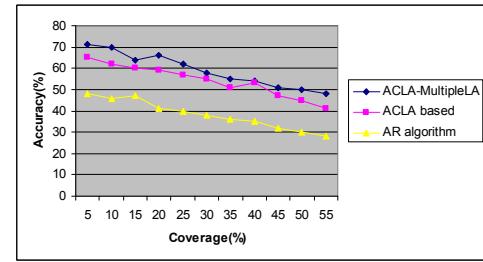


Figure 3. Comparing system performance with ACLA based model and AR algorithm

Experimental results show that the extended ACLA based system improves performance significantly and our system gains much better results than previous method. Also the system keeps its dynamic property so it can interact with the user and learns from his behavior. By extracting concept and topic of each page and assigning learning automata to each

concept, the system can learn that in each page which topic is more interesting and useful for users. So it can be concluded that new approach is capable of making more useful recommendation.

V. CONCLUSION

In this paper, we proposed a new web page recommendation system based on asynchronous cellular learning automata with multiple learning automata in each cell. We showed the restrictions that a usage-based system inherently suffers from and demonstrated how combining conceptual information regarding the web pages can improve the system.

We extend the previous ACLA based system by extracting concepts of a page and assigning learning automata to each concept. The learning automata is used to learn the importance of the concept of a page to users, and try to give more consideration to topics which are more useful to the user, in order to capture the user's information need more precisely and recommend more useful pages to the user. We use web usage data and content of the web page to learn user information needs and predicting user's future requests. Also the system keeps its dynamic property so it can interact with the user and learns from his behavior. This system constantly learns the user information needs, so accuracy of the system will remain high over time. System performance was evaluated under different settings and in comparison with previous ACLA based approach and association rule based model. Our evaluation results show the flexibility of the ACLA model to incorporate different sources of information to improve the quality of recommendations. Also results show that ACLA based model could significantly improve the recommendation effectiveness.

In future work, we aim to use more sophisticated methods for organizing a website into a concept hierarchy. More accurate methods of assessing implicit feed-back can also be used to derive a more precise reward function. Integration of other sources of domain knowledge e.g. a domain-ontology into the model can also be another future work for this paper.

REFERENCES

- [1] X. Fu, J. Budzik, and K.J. Hammond, "Mining navigation history for recommendation," In Intelligent User Interfaces, pp.106–112, 2000.
- [2] J. Srivastava, R. Cooley, M. Deshpande, P.N. Tan, "Web usage mining: discovery and applications of usage patterns from web data," SIGKDD Explorations, Vol. 1, No. 2, pp. 12–23, 2000.
- [3] A.L.C. Yi-Hung Wu, Y. Chen, "Enabling personalized recommendation on the web based on user interests and behaviors," Proc. 11th International Workshop on Research Issues in Data Engineering, 2001.
- [4] B. Mobasher, R. Cooley, J. Srivastava, "Automatic personalization based on web usage mining," Communications of the ACM, pp. 142-151, 2000.
- [5] B. Mobasher, H. Dai, T. Luo, M. Nakagawa, "Effective personalization based on association rule discovery from web usage data," Proc. 3rd ACM Workshop on Web Information and Data Management, 2001.
- [6] R. Zaiane, J. Li, R. Hayward, "Mission-based navigational behavior modeling for web recommender system," Proc. 5th International Conference on Electronic Commerce and Web Technologies, 2006.
- [7] H. Dai, B. Mobasher, "A Road Map to More Effective Web Personalization: Integrating Domain Knowledge with Web Usage Mining", Proc. International Conference on Internet Computing, pp. 58–64, 2003.
- [8] B. Mobasher, H. Dai, T. Luo, Y. Sun, J. Zhu, "Integrating web usage and content mining for more effective personalization," In EC-Web, pp. 165–176, 2000.
- [9] J. Guo, V. Kes'elj, Q. Gao, "Integrating web content clustering into web log association rule mining", Proc. Canadian AI'2005, Victoria, BC, Canada, May 2005.
- [10] R. Forsati, A. Rahbar, M. Mahdavi, "Effective page recommendation algorithms based on distributed learning automata," Proc. 4th International Multi-Conference on Computing in the Global Information Technology, France, 2009.
- [11] H. Beigy, M.R. Meybodi, "A Mathematical Framework for Cellular Learning Automata," Advances on Complex Systems, Vol.7, No.3-4, pp. 295-320, 2004.
- [12] M.R. Meybodi, H. Beigy, M. Taherkhani, "Cellular learning automata and its applications," Journal of Science and Technology, No. 25, pp.54-77, 2004.
- [13] H. Beigy, M.R. Meybodi, "Asynchronous cellular learning automata", Automatica, Vol. 44, No. 5, 2008.
- [14] H. Beigy, M.R. Meybodi, "Cellular Learning Automata With Multiple Learning Automata in Each Cell and Its Applications", IEEE Transaction on Systems, Man, and Cybernetics, Part B: Cybernetics, Vol. 40, No. 1, pp. 54-66, 2010.
- [15] M. Talabeigi, R. Forsati, M.R. Meybodi, "A dynamic web recommender system based on cellular learning automata", in press of 2nd International conference on Computer Engineering and Technology, 2010.
- [16] R. Forsati, M.R. Meybodi, "Effective web page recommendation algorithms based on distributed learning automata and weighted association rules," Journal of Expert Systems with Applications, Vol. 37, No. 2, pp. 1316-1330, 2010.
- [17] B. Safarkhani, M. Talabeigi, M. Mohsenzade, M.R. Meybodi, "Deriving Semantic Sessions from Semantic Clusters", Proc. 1st International conference on Information Management and Engineering, 2009.
- [18] A. B. Geva, "Hierarchical Unsupervised Fuzzy Clustering", in proceeding of the IEEE transaction on fuzzy systems, 1999.
- [19] A. Bose, K. Beemanapalli, J. Srivastava, S. Sahar, "Incorporating concept hierarchies into usage mining based recommendations," Proc. 8th WEBKDD workshop, 2006.
- [20] M. Morita, Y. Shinoda, "Information filtering based on user behavior analysis and best match text retrieval," Proc. of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Springer-Verlag New York, Inc., Dublin, Ireland, pp. 272–281, 1994.
- [21] J. Liu, S. Zhang, J. Yang, "Characterizing Web usage regularities with information foraging agents", IEEE Transactions on Knowledge and Data Engineering, pp. 566-584, 2004.
- [22] R. Forsati, M. R. Meybodi, A. Rahbar, "An efficient algorithm for web recommendation systems," Computer Systems and Applications, Proceedings of the Seventh ACS/IEEE International Conference on Computer Systems and Applications, 2009.
- [23] R. Forsati, M. R. Meybodi, A. Ghari Neiat, "Web page personalization based on weighted associative rules," Proceedings of International Conference on Electronic Computer Conference (ICECT 2009), pp. 13-135, Macao, 2009.