An Efficient Algorithm for Web Recommendation Systems

Rana Forsati
Department of Computer Engineering,
Islamic Azad University, Karaj Branch
Karaj, Iran
forsati@kiau.ac.ir

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

Afsaneh Rahbar
Department of Computer Engineering
Islamic Azad University, North Branch of Tehran
Tehran, Iran
afsanerahbar@ieee.org

Abstract— **Different efforts have been made to address the problem of information overload on the Internet. Web recommendation systems based on web usage mining try to mine users' behavior patterns from web access logs, and recommend pages to the online user by matching the user's browsing behavior with the mined historical behavior patterns. In this paper we propose effective and scalable technique to solve the web page recommendation problem. We use distributed learning automata to learn the behavior of previous users' and cluster pages based on learned pattern. One of the challenging problems in recommendation systems is dealing with unvisited or newly added pages. As they would never be recommended, we need to provide an opportunity for these rarely visited or newly added pages to be included in the recommendation set. By considering this problem, and introducing a novel Weighted Association Rule mining algorithm, we present an algorithm for recommendation purpose. We employ the HITS algorithm to extend the recommendation set. We evaluate proposed algorithm under different settings and show how this method can improve the overall quality of web recommendations.**

Keywords- web recommender system; association rules; data mining, learning automata, web minig

I. INTRODUCTION

The volume of information available on the internet is increasing rapidly with the explosive growth of the World Wide Web and the advent of e-Commerce. While users are provided with more information and service options, it has become more difficult for them to find the "right" or "interesting" information, the problem commonly known as information overload.

Recommender systems [1] are alternative, user-centric, promising approaches to tackle the problem of information overload by adapting the content and structure of websites to the needs of the users by taking advantage of the knowledge acquired from the analysis of the users' access behaviors. They can be generally defined as systems that guide users toward interesting or useful objects in a large space of possible options [2].

In recent years there has been an increasing interest in applying web usage mining techniques to build web recommender systems [3,4,6]. 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 implicit, and potentially useful 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. They could, therefore, be a good alternative to the explicit user rating or feedback in deriving user models. Unlike traditional techniques, which mainly recommend a set (referred to as the recommendation set) of items deemed to be of interest to the user base their decisions on user ratings on different items or other explicit feedbacks provided by the user [7,8]. These techniques discover user preferences from their implicit feedbacks, namely the web pages they have visited. Clustering and collaborative filtering approaches are ready to incorporate both binary and non-binary weights of pages, although binary weights are usually used for computing efficiency [9]. Association Rule (AR) mining [11] can lead to higher recommendation precision [9], and are easy to scale to large datasets, but how to incorporate page weight into the AR models has not been explored in previous studies.

In this paper, we focus on the association-mining method, which is a widely used data analysis method in web usage mining [12,13]. Association rule mining has been successfully applied in the pages recommendation systems, web page personalization and is easy to scale to large data datasets [14, 16], but how to incorporate page weight into the AR in the pages recommendation system has not been explored in previous studies.

Weighted Association Rule (WAR) mining allows different weights to be assigned to different items, and is a possible approach to improving the AR model in the web personalization process. Cai et al. [18] proposed assigning different weights to items to reflect their different importance. In their framework, two ways are proposed to calculate itemset weight: total weight and average weight.

Weighted support of an itemset is defined as the product of the itemset support and the itemset weight. Tao et al. [19] also proposed assigning different weights to items, the itemset/transaction weight is defined as the average weight of the items in the set/transaction, and weighted support of an itemset is the fraction of the weight of the transactions containing the itemset relative to the weight of all transactions. Both models attempt to give greater weights to more important items, facilitating the discovery of important but less frequent itemsets and association rules. However, both models assume a fixed weight for each item while in the context of web usage mining and page recommendation systems a page might have different importance in different sessions.

In this paper, we try to assign a quantitative weight to each page, taking into account the degree of interest. We extend the traditional association rule mining algorithm by allowing that a weight to be associated with each item in a transaction for reflecting the interest of each item within the transaction. In the proposed weighted association rule miner, the time spent by each user on each page and visiting frequency of each page are used to assign a quantitative weight to the pages instead of traditional binary weights. The intuition behind this idea is that the time spent on pages [20] and visiting frequency are good implicit interest indicator of a user on those pages. One of the challenging problems in recommendation systems is dealing with unvisited or newly added pages. It is conceivable that there are pages not yet visited, even though they are relevant and could be interesting to have in the recommendation list. Such resources could be, for instance, newly added web pages or pages that have links to them not evidently presented due to bad design. Thus, these pages or resources are never presented in the sessions previously discovered. So we need to provide an opportunity for these rarely visited or newly added pages to be included in the recommendation list. Otherwise, they would never be recommended. To alleviate this problem, we present an efficient recommendation algorithm based on proposed weighted association rule mining algorithm and distributed learning automata [22]. In the proposed algorithm we employ the HITS[23] algorithm to extend the recommendation set. The proposed algorithm solves the problem of recommending rarely visited or newly added pages. The methodology is like this: first, cluster the pages based on users' usage pattern. Second, the weighted association rules of each URL will be extracted from the web log data and similarity between active user sessions will be calculated upon the weighted rules instead of an exact match for finding the best rule. The recommendation engine will then find the most similar rules to the active user session with the highest weighted confidence by scoring each rule in terms of both its similarity to the active session and its weighted confidence. In this phase, the top-*n* most similar pages generate the seed recommendation set. Finally, we apply the HITS algorithm to rank the candidate set and generate final recommendation set.

We have applied proposed algorithm on standard data set and got very good results compared to the association rules, which is commonly known as one of the most successful approaches in web mining based recommender systems. The evaluation of the experimental results shows considerable improvements.

The organization of the paper is as follows: in section 2 we introduce our weighting schema. The proposed algorithm present in section 3. Section 4 gives the performance evaluation of the proposed algorithm compared to association rule based method. Section 5 concludes the paper.

II. INCORPORATING PAGE WEIGHT

Let $P=\{p_1,p_2,...,p_m\}$ denote the set of web pages accessed by users in web server logs after the preprocessing phase[24], each of them is uniquely represented by its associated URL. Also let $T=\{t_1,t_2,...,t_n\}$ be the set of user transactions where each $t_i \in T$ is a subset of *P*. To facilitate the high quality recommendation, we represent each transaction t as an m-dimensional vector over the space of web pages, $t=<(p_1,w_1),(p_2,w_2),...,(p_m,w_m)>$, where w_i denotes the weight with the i^{th} web page $(1\le i\le m)$ visited in a transaction t. The weight w_i in transaction *t* needs to be appropriately determined to capture a user's interest in i^{th} web page.

Since the recommendation process is based on the behavior of previous users, so the weighting schema must precisely model the user's interest. Recommendation approaches proposed in previous works; however, do not distinguish the importance of different pages and all the visited pages are treated equally whatever their usefulness to the user. They neglect the difference in the importance of the pages and degree of interest in a users' session. It is quite probable that not all the pages visited by the user are of interest to him/her. A user might get into a page only to find it is of no value to him/her, causing irrelevant page accesses to be recorded into the log file. Therefore, it is imperfect to use all the visited pages equally to capture user interest and predict user behavior. Although in usage-based recommendation systems we can't expect users to express likes or dislikes explicitly, we need a weight measure for approximating the interest degree of a web page to a user.

Inspired by Chan and coworkers [25,26], we propose a weighting measure which is calculated from web logs to extract the interest of page for the visitor. In our weighting schema, both of time length of a page and visiting frequency of a page are used to estimate its importance in a transaction, in order to capture the user's interest more precisely instead of binary which is typically used in other researches. This approach try to give more consideration to more useful pages, in order to better capturing the user's information need and recommend more useful pages to the user.

Several reasons validate the idea of using pages visit duration as one of the weighting parameters. First, it reflects the relative importance of each page, because a user generally spend more time on a more useful page [20,28], because if a user is not interested in a page, he/she do not spend much time on viewing the page and usually jumps to another page quickly [27]. However, a quick jump might also occur due to the short length of a web page so the size of a page may affect the actual visiting time. Hence, it is more appropriate to accordingly normalize duration by the length of the web page, that is, the total bytes of the page. The formula of duration is given in Equation (1). Second, the rates of most human beings getting information from web pages should not differ greatly [28]. If we assume a similar rate of acquiring information from pages for each user, the time a user spends on a page is proportional to the volume of information useful to him/her. As page duration can be

calculated from web logs, it is a good choice for inferring user interest.

Frequency is the number of times that a page is accessed by different users. It seems natural to assume that web pages with a higher frequency are of stronger interest to users. A parameter that must be considered in the calculating the frequency of a page is the in-degree of that page (e.g. the number of incoming links to the page). It is obvious that a page with large in-degree has more probability to be visited by a user than a page with small one. Specially, in comparing two pages with same visiting rate, the page with small in-degree is more interesting. The formula of frequency is given in Equation (2).

We use time spent by a user for viewing a page and frequency of visiting as two very important pieces of information in measuring the user's interest on the page, so we assign a significant weight to each page in a transaction according to these definitions as Equation (3).

$$Duration(p) = \frac{\frac{Total\ Duration(p)}{Size(p)}}{\max_{Q \in T}\left(\frac{Total\ Duration(p)}{Size(p)}\right)} \quad (1)$$

$$Frequency(p) = \frac{Number\ of\ visit(p)}{\sum_{Q \in T} Number\ of\ visit(Q)} * \frac{1}{\text{Indegree}(p)} \quad (2)$$

$$Weight(p) = \frac{2 * Frequency(p) * Duration(p)}{Frequency(p) + Duration(p)} \quad (3)$$

At the end, every user transaction is successfully transformed into a m-dimensional vector of weights of web pages, i.e., $t = <(p_1, w_1), (p_2, w_2), ..., (p_m, w_m)>$, where m is the number of web pages visited in all users' sessions.

III. The Proposed Recommendation Algorithm

In this section we present an efficient algorithm based on distributed learning automata and proposed weighted association rule algorithm. The algorithm solves the problem of recommending rarely visited or newly added pages and provides an opportunity for these rarely visited or newly added pages to be included in the recommendation list. The steps in the algorithms could be briefly summarized as follows:

Step 1: Cluster the pages based on users' usage pattern.

Step 2: Generate the seed recommendation set based on Weighted Association Rules Mining.

Step 3: Extend the seed set by clusters to generate the candidate set and apply the HITS algorithm to rank the candidate set and generate final recommendation set.

A general view of the proposed algorithm is depicted in Fig. 1. These steps are described in the next subsections.

A. Cluster the Pages Based on Users' Usage Pattern

We propose an algorithm to cluster web pages not from the content of the pages but from the pattern of their usage, assuming that users have an intuitive grasp of what a page is about and how valuable it is, and this intuition guides their actions. In the next subsections we employ the result of this algorithm to extend the recommendation set.

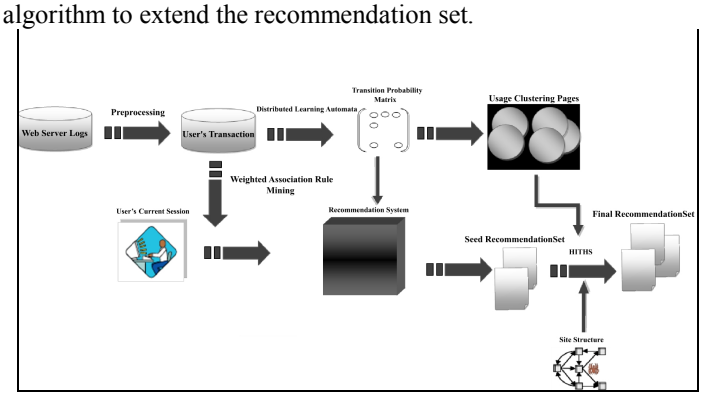

Figure 1. The framework of the proposed algorithm

The method clusters the pages based on how often they occur together across user sessions. On the other hand, page clusters tend to group together frequently co-occurring items across sessions, even if these items are themselves not deemed to be similar. This allows us to obtain clusters that potentially capture overlapping interests of different types of users.

The idea of clustering based on usage data is inspired by the functioning of the brain. In the brain, concepts that are activated simultaneously (co-activation) become more strongly associated. Since, users visiting a web site can be assumed to be looking for mutually relevant pages rather than a random assortment of unrelated pages, pages which are consulted by same user, are co-activated and have association with each other. In other words, documents develop stronger associations as they are more frequently co-activated. It is noticeable that this method is particularly useful for multimedia documents, which do not contain any searchable keywords.

To learn the associations implicitly exists between pages based on usage data; we use a distributed learning automaton (DLA) with n learning automata (LAs) [22] with variable number of actions. A distributed learning automata with n LAs with variable number of actions learns the association between pages using log data. For each page in the site a LA with $n-1$ actions is added to the DLA. Each action corresponds to following a page. For each LA[29,30] at each time a subset of its actions is active. The number of actions in the LA assigned to page i is equal to the number of pages that a user at page i can follow from that page. In the beginning, all of actions are inactive. When a user at page i go to page j, the action corresponds to page j is activated and awarded based on Equation (4).

$$\hat{p}_i(n+1) = \hat{p}_i(n) + a.(1 - \hat{p}_i(n)) \quad (4)$$
$$\hat{p}_j(n+1) = \hat{p}_j(n) - a.\hat{p}_i(n) \qquad \forall j \ \ j \neq i$$

For every user session in the log file, we begin with the first page. For each pair of consecutive pages in the session, the LA corresponding to the first page is used to update its probabilities if the action is already active; otherwise activates it. We assume that any consecutive pages' repetitions have been removed from the user sessions; on the

other hand, we keep any pages that have been visited more than once, but not consecutively. This process is repeated till reaching the latest page in the session. In the DLA, the probability of action j in LA_i represents the association between i^{th} and j^{th} page. We create the association matrix P from the actions probability in DLA as follows. We set the a_{ij} to the probability of action j in LA_i. Since the learning process assumes ordered page access, so the learning process yields to an asymmetric association matrix $(p_{ij} \neq p_{ji})$. By multiplying the (asymmetric) matrix P with its transpose we can create a new, symmetric matrix:

$$S = P \cdot P^T \quad (5)$$

$$s_{ij} = \sum_k a_{ik} a_{kj}$$

Where s_{ij} represents the degree of similarity between the pages i and j. Indeed, s_{ij} is the dot product between the all the associations that the documents i and j have with other documents. The more the association vectors overlap, and thus the more i and j resemble each other in the way they relate to other documents, the larger the dot product, and therefore s_{ij}. This similarity measure can now be used as an input to a variety of clustering algorithms that put documents together in classes depending on how similar/dissimilar they are from each other.

Having the symmetric association matrix, the clustering phase is conducted in the following steps:

1. We create a similarity matrix between web pages where the distance (similarity) between pages is either zero, if the two pages are directly linked in the web site structure (i.e. there is a hyperlink from one to the other) or set to the co-occurrence frequency between the two pages in matrix S otherwise.

2. A graph *G* is created in which each page is a node and each nonzero cell in the similarity matrix is an edge. In order to reduce noise, we apply a threshold to remove edges corresponding to low co-occurrence frequency.

3. The graph created in previous step is partitioned using graph partitioning tool MeTiS for minimizing the number of cut edges. The generated clusters will be used to extend the recommendation set.

B. Generating the Seed Recommendation Set

This phase consists of two parts. First, the weighted association rules of each URL will be extracted from the web log data, the rules produced is representing the behavior of user's navigation on the web site. Secondly, the recommendation engine will search the top-*n* most similar weighted rules to the active user session before generating recommendation for the user. During the second part instead of exact match between the active user and rules, we use a similarity measure for finding the most similar rules.

1) Mining Weighted Association Rules

Given a set of transactions where each transaction is a set of items (pages), an association rule implies the form $X \Rightarrow Y, where X \subset I, Y \subset I, X \cap Y = \phi$, where X and Y are two sets of items; X is the body and Y is the head of the rule.

Association rules capture the relationships among items based on their patterns of co-occurrence across transactions. In the case of web transactions, association rules capture relationships among pages based on the navigational patterns of users. Each web page can be viewed as an item, and the set of web pages accessed by a user within a short period of time can be treated as a transaction so the purpose of mining association rules is to find out which web pages are usually visited together in different sessions.

However, the traditional association rules (ARM) model focus on binary attributes. In other words, this approach only considers whether an item is present in a transaction or not. Also it is supposed that all items have the same significance and does not take into account the weight of an item within a transaction and all pages in a transaction are treated uniformly. Also, in most previous approaches of applying ARM to web usage personalization they ignore the difference in the importance of the pages in a user session.

It is quite probable that not all the pages visited by the user are of interest to him/her. As mentioned before, we first extend the traditional association rule problem by allowing a weight to be associated with each item in a transaction to reflect interest of each item within the transaction. In turn, this provides us with an opportunity to associate a weight parameter with each item in a resulting association rule, which called a weighted association rule (WAR).

In the following we describe weighted rules with the definition of associated parameters. We extend the Apriori[32] by adapting its parameters based on weighted items.

2) Weight Settings

Given the transformation of user transactions into m - dimensional space as vectors of weights of web pages, $t = <(p_1, w_1), (p_2, w_2), ..., (p_m, w_m)>$ where each $p_i \in P$, the weight w_i associated to page p_i is a non-negative real number to reflect the importance of page p_i in transaction t according to Equation (3). Inspired by Tao[19], we modify the measures exist in Apriori algorithm in the following definitions to reflect the weighting schema.

Definition-1. Weight of an itemset in a transaction:

Based on the item weight $w(p_i)$, the weight of an itemset X, denoted as $w(X,t)$, can be derived from the weights of its enclosing items. One simple way is to use the minimum weight of the all items in the itemset as the weight of whole itemset as shown in Equation (6).

$$w(X,t) = \begin{cases} \min(w(p_1, p_2, ..., p_k) & X \subseteq t \\ 0 & X \not\subset t \end{cases} \quad (6)$$

Where k is the number of items in the itemset.

Alternatively, we can use the average weights of its enclosing items as the itemset weight. Our experiments show that the minimum weight has better quality.

Definition-2. Transaction weight:

By assigning a weight to each item and itemset, we also assign a weight to each transaction to be used in the calculation of the support of each itemset. Assigning weight to transactions gives us the possibility to distinguish between different transactions. Usually the higher a transaction weight, the more it contributes to the mining result. One simple way is to calculate the average weights of all items that enclosed in each transaction. The weight of each transaction $\mathbf{w(t_k)}$ is calculated as shown in Equation (7).

$$\mathrm{w}(\mathrm{t_k}) = \frac{\sum_{i=1}^{|t_k|} w(p_i)}{|t_k|} \tag{7}$$

Definition-3. Weighted support of an itemset across all transaction:

We modify the support of an itemset, Weighted support $wsp(X)$ of an itemset X across all transactions is defined as follows:

$$wsp(X) = \frac{\sum_{t_i \in T} w(t_i) * w(X, t_i)}{\overline{w} * \sum_{k=1}^{|T|} w(t_k)} \tag{8}$$

Where $\overline{w}$ is the average weight of all the items across all transactions, and T is the set of all transactions.

3) Weighted Frequent Itemset

The problem of frequent pattern mining in the traditional association rule mining framework is to find the complete set of itemset satisfying a minimum support threshold in the database. In our model, we say an itemset is frequent if its weighted support is above a predefined weighted support threshold. Our approach to mining frequent itemsets is based on the Apriori [32] algorithm. To prune infrequent patterns, frequent pattern mining uses the downward closure property (anti-mono-tone property) [31]. That is, any subset of a significant itemset is also significant or if a pattern is infrequent pattern, all super patterns must be infrequent patterns. Using the downward closure property, infrequent patterns can be easily pruned. By our definition of weighted support and frequent itemsets, there is a property that any subset of a frequent itemset is also frequent, here called a weighted downward closure property [32]. The downward closure property of the support measure in the unweighted case longer exists. Therefore, the candidate itemsets having k items can be generated by joining large itemsets having k-1 items. This can result in much smaller number of candidate itemsets. For example, if we are looking for pairs of items with minsup, we can only consider those items that appear in the database having minsup. Provided minsup is high enough, the number of items for the next joining step will be small enough to speed up the computation significantly. Following theorem shows that our weighting schema holds the downward closure property.

Theorem: The proposed weighting schema holds the downward closure property and for any candidate itemset, all of its subitems also are candidate itemset.

Proof. Let I_1 and I_2 be two itemsets. Also suppose that $I_1 \subset I_2$, i.e. I_2 be a superset of I_1. For proving the validity of downward closure property in the proposed algorithm, we suppose that I_1 is not a significant itemset over all the transactions but I_2 is a significant itemset. Let T_1 denote a set of transactions which contains all the items in I_1 and similarly T_2 denote the set for I_2. Since I_2 is superset of I_1, so $T_2 \subset T_1$. Therefore $\sum_{t \in T_1} w(t) \geq \sum_{t \in T_2} w(t)$. According to the definition of weighted support of an itemset we have $wsp(I_1) = \frac{\sum_{t \in T_1} w(t) * w(I_1, t)}{\overline{w} * \sum_{t \in T_1} w(t)}$ and $wsp(I_2) = \frac{\sum_{t \in T_2} w(t) * w(I_1, t)}{\overline{w} * \sum_{t \in T_2} w(t)}$. By comparing $wsp(I_1)$ and $wsp(I_2)$ and considering the fact that $\sum_{t \in T_1} w(t) \geq \sum_{t \in T_2} w(t)$ we have that $wsp(I_1) \geq wsp(I_2)$. Because I_1 is not a significant itemset, its weighted support is less than the minimum threshold and since $wsp(I_1) \geq wsp(I_2)$ so the $wsp(I_2)$ is also less than the minimum support threshold and I_2 is not a significant itemset. In conclusion, if an itemset is a significant itemset, its subsets also are significant itemset and it proves that the downward closure property always valid in the proposed algorithm.

Definition-4. Weighted confidence of the weighted association rule:

We define the weighted confidence of association rule for weighted rules as follows:

$$wconf(X \Rightarrow Y) = \frac{wsp(X \cup Y)}{wsp(X)} \tag{9}$$

Definition-5. Weighted rules:

For each rule, besides the weighted confidence and weighted support, we also add the weight of each page. The result of weighted association rule mining conceptually described as follows:

$r = <(p_1, p_2, ..., p_k), (q_{k+1}, q_{k+2}, ..., q_{k+m}), (w_1, w_2, ..., w_{k+m}), \delta, \alpha> \in R$,

where $(p_1, p_2, ..., p_k), (q_{k+1}, q_2, ..., q_{k+m})$ present the body and head of the weighted rule respectively, w_i represent the weight of i^{th} page in the rule, δ represent the weighted support and α represent the weighted confidence of the rule.

Each of the weighted association rules $r = <(p_1, p_2, ..., p_k), (q_{k+1}, q_{k+2}, ..., q_{k+m}), (w_1, w_2, ..., w_{k+m}), \delta, \alpha> \in R$ obtained in the mining stage described in the previous section, are represented as a set of page-weight pairs. This will allow for both the active session and the association rules to be treated as m-dimensional vectors over the space of page in the site. Thus, given a weighted association rule r, we can represent the left-hand side of the each rule r_L as a vector: $r_L = \{w_1, w_2, ..., w_m\}$, where

$$w_i = \begin{cases} weight(p_i, r_{Li}), & if \quad p_i \in r_L \\ 0, & \text{otherwise} \end{cases} \tag{10}$$

Similarly, the current user session is also represented as a vector $S = \{s_1, s_2, ...s_m\}$ where s_i is a significance weight

associated with the corresponding page reference, if the user has accessed p_i in this session, and $s_i = 0$, otherwise.

Then we compute the matching score between association rules that capture relationships among page based on their co-occurrence in navigational patterns of users and the current active session. The matching score between them is defined as:

$$\text{Dissimilarity}(S, r_L) = \sum_{i: r_{Li} > 0} \left(\frac{2*(w(s_i) - w(r_{Li}))}{w(s_i) + w(r_{Li})}\right)^2 \quad (11)$$

$$\text{Match Score}(S, r_L) = 1 - \frac{1}{4}\sqrt{\frac{\text{Dissimilarity}(S, r_{Li})}{\sum_{i: r_{Li} > 0} 1}} \quad (12)$$

S and r_L represent the active user and left hand side of weighted association rule, respectively.

As the algorithm tries to find rules that are similar to the active user session, the similarity measure between a rule and the active session is dependent on the magnitude of the left-hand side of the rule.

The recommendation engine is the online component of a usage-based personalization system in order to determine which items (not already visited by the user in the active session) are to be recommended, a recommendation score is computed for each page p_i . Two factors are used in determining this recommendation score: the overall matching score of the active session to the weighted rules as a whole, and the weighted confidence of the rule. The recommendation scores for the active user are computed by multiplying these factors. Given the weighted association rule and active session S , a recommendation scores for the active session, $Rec(S, X => p)$, is computed as follows:

$$Rec(S, X => p) = Match\ Score(S, X) * wconf\ (X \Rightarrow p) \quad (13)$$

Finally the top-*n* most similar pages (seed recommendation set) are sorted to use in the next phase.

C. Extending the Seed Set and Apply HITHS

The most of recommendation algorithms suffer from two major drawbacks. First, with increasing the size of recommendation set, the precision decreases significantly. Second, some resources such as rarely visited or newly added page are out of recommendation consideration. It is conceivable that there are other resources not yet visited, even though they are relevant and could be interesting to have in the recommendation list. Such resources could be, for instance, newly added web pages or pages that have links to them not evidently presented due to bad design. We need to provide an opportunity for these rarely visited or newly added pages to be included in the recommendation set. Otherwise, they would never be recommended. To alleviate these problems, we use the seed recommendation set generated in previous step as the input of this phase. We extend the seed set to generate a candidate recommendation set. initially; we put all of the pages in seed set in the candidate set. For each page p in the seed set, the candidate set is supplemented with pages that are in the same cluster with page p. The clusters generated in the subsection 3.1. Since the pages in each cluster have strong association based on users' behavior, this extension sounds good. We generate a graph from pages included in the candidate set by connecting them with links exist in the underlying site structure. The result is what is called a connectivity graph which now represents our augmented navigational pattern.

This process of obtaining the connectivity graph is similar to the process used by the HITS algorithm [10] to find the authority and hub pages. We take advantage of the built connectivity graph by clustering to apply the HITS algorithm in order to identify the authority and hub pages within a given cluster. These measures of authority and hub allow us to rank the pages within the cluster. This is important because at real time during the recommendation, it is crucial to rank recommendations, especially if they are numerous. Authority and hub are mutually reinforcing [10] concepts. Indeed, a good authority is a page pointed to by many good hub pages, and a good hub is a page that points to many good authority pages. Since we would like to be able to recommend pages newly added to the site, in our framework, we consider only the hub measure [5]. This is because a newly added page would be unlikely to be a good authoritative page, since not many pages are linked to it. However, a good new page would probably link to many authority pages, it would, therefore, have the chance to be a good hub page. Consequently, we use the hub value to rank the candidate recommendation pages in the on-line module to create the final recommendation set. Then the highest recommendation score choose as the recommendation to the active user.

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. Overall our experiments have verified the effective of our proposed techniques in web page recommendation.

As our evaluation data set we used the web logs of the DePaul University CTI Web server[1], based on a random sample of users visiting the site for a 2 week period during April 2002. We split the data sets in two non-overlapping time windows to form training and a test data set. 70% of the data set (9745 sessions) 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. The system was allowed to make n recommendations in each step with $n < 10$ *and* $n < \sqrt{l}$, where l is the number of outgoing links of the last page visited by the user.

This limitation on number of recommendations is adopted from [21].

A. Evaluation Metrics

In order to evaluate the recommendation effectiveness for our method, we measured the performance of proposed method using 2 different standard measures, namely, Precision, Coverage [15]. Recommendation precision

[1] http://maya.cs.depaul.edu/classes/ect584/data/cti-data.zip.

measures the ratio of correct recommendations (i.e., the proportion of relevant recommendations to the total number of recommendations), where correct recommendations are the ones that appear in the remaining of the user session. For each visit session after considering each page p the system generates a set of recommendations $R(p)$. To compute the Precision, $R(p)$ is compared with the rest of the session $T(p)$ as follows:

$$\Pr ecision = \frac{T(p) \cap R(p)}{R(p)} \quad (14)$$

Recommendation coverage on the other hand shows the ratio of the pages in the user session that the system is able to predict(i.e., the proportion of relevant recommendations to all pages that should be recommended) before the user visits them:

$$Coverage = \frac{T(p) \cap R(p)}{T(p)} \quad (15)$$

B. Experimental Results

In all experiments we measured both Precision and Coverage of recommendations against varying number of recommended pages. In the implication of using a sliding window of size w is that we base the prediction of user future visits on his w past visits. The choice of this sliding window size can affect the system in several ways. To consider the impact of window size (the portion of user histories used to produce recommendations) we also vary window sizes from 1 to 4. The impact of different window sizes on precision scores of recommendations against varying umber of recommended pages from 1 to 12 shows in Fig. 2. The results show clearly that precision increases as a larger portion of user's history is used to generate recommendations. It can be inferred form this diagram that a window of size 1 ($|w| = 1$)which considers only the user's last page visit does not hold enough information in memory to make the recommendation, the accuracy of recommendations improve with increasing the window size and the best results are achieved with a window size of 3($|w| = 3$). As shown in Fig. 2 using a window size larger than 3 results in weaker performance, it seems to be due to the fact that, sequences of page visit that occurring less frequently in the usage logs.

We used number of recommended pages varying from 1 to 11 to measure the precision and coverage of algorithms. We used a fixed window size of 3 on recommendation history. As our experiments show the best results are achieved when using a window of size 3($|w| = 3$).We observed our system performance in comparison with association rules, which is commonly known as one of the most successful approaches in web mining based recommender systems [14].

Fig. 3 and Fig. 4 have shown the comparison of our system's performance with AR method in the sense of their accuracy and coverage in different number of recommended pages on CTI dataset. As the number of recommendation page increases, naturally precision decreases in all systems,

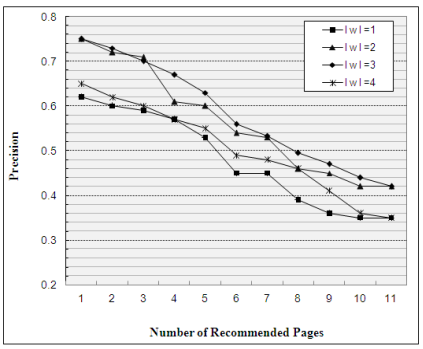

Figure 2. System performance with various windows size

but our system gains much better results than the association rule algorithm. It can be seen the rate in which precision decreases in our algorithm is lower than traditional association rule algorithm. Experimental results show that the proposed weighted association rule based model increases coverage and precision significantly and our system gains much better results than the traditional association rule algorithm.

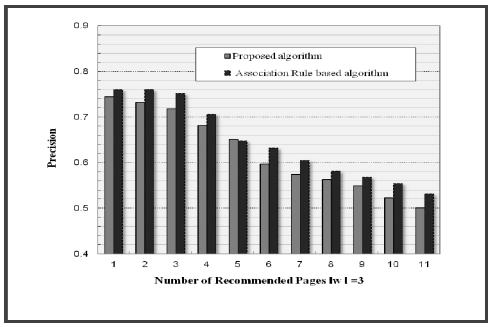

Figure 3. Precision of the AR and proposed Algorithm

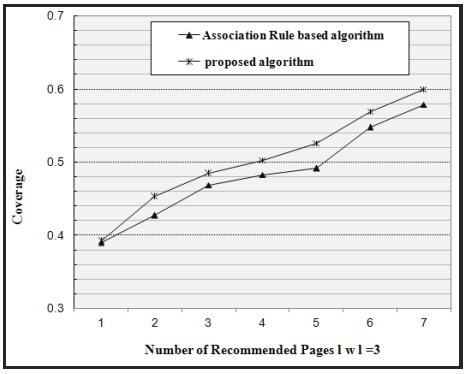

Figure 4. Coverage of the AR and proposed algorithm

It can be concluded that our approach is capable of making web recommendation more accurately and effectively against the conventional method. By combining similarity between rules and active user and confidence of the weighted rules, the recommendation engine has selected only the most relevant pages. Therefore, it increases the effectiveness of the recommendation engine.

V. CONCLUSION

This paper proposes a new web recommendation system based on the proposed Weighted Association Rule (WAR) model. We extend the association rule mining by assigning a significant weight to the pages based on time spent by each user on each page and visiting frequency of each page. The proposed weighting measure can be used to judge the importance of a page to a user, and try to give more consideration to pages which are more useful to the user. One of the challenging problems in recommendation systems is dealing with unvisited or newly added pages. As they would never be recommended, we need to provide an opportunity for these rarely visited or newly added pages to be included in the recommendation set. By considering this problem, we use distributed learning automata to learn the behavior of previous users' and cluster pages based on learned pattern and employ the HITS algorithm to extend the recommendation set. System performance was evaluated under different settings and in comparison with traditional Association Rule based model. The experimental results show that our method is better in precision and coverage rates than the conventional association rule based recommendation.

REFERENCES

[1] P. Resnick, H. R. Varian, "Recommender Systems", Communications of the ACM, 40 (3), 1997, pp. 56-58.

[2] P. Burke, "Hybrid Recommender Systems: Survey and Experiments", User Modeling and User-Adapted Interaction, 2002.

[3] X. Fu, J. Budzik, K. J. Hammond, "Mining Navigation History for Recommendation", In Intelligent User Interfaces, 2000, pp. 106–112.

[4] C. Lin, S. Alvarez, C. Ruiz, "Collaborative Recommendation via Adaptive Association Rule Mining, 2000.

[5] O. Za¨ıane, J. Li, R. Hayward, "Mission-Based Navigational Behavior Modeling for Web Recommender System", Springer-Verlag Berlin Heidelberg, 2007.

[6] A. L. C. Yi-Hung Wu, Yong-Chuan Chen, "Enabling Personalized Recommendation on the Web based on User Interests and Behaviors", In 11th International Workshop on research Issues in Data Engineering, 2001.

[7] M. Deshpande, G. Karypis, "Item-Based Top-N Recommendation Algorithms", ACM Transactions on Information Systems (TOIS), 2004.

[8] J. Herlocker, J. Konstan, A. Brochers, J. Riedel, "An Algorithmic Framework for Performing Collaborative Filtering", Proceedings of 200 Conference on Research and Development in Information Retrieval, 2000.

[9] B. Mobasher, "Web Usage Mining and Personalization", In Practical Handbook of Internet Computing, Munindar, P. Singh (ed.), CRC Press, 2005.

[10] J. M. Kleinberg," Authoritative Sources in a Hyperlinked Environment", Journal of The ACM, vol. 46, no. 5, 1999, pp. 604–632.

[11] M. Nakagawa, B. Mobasher, "A Hybrid Web Personalization Model Based on Site Connectivity", In The Fifth International WEBKDD Workshop: Web mining as a Premise to Effective and Intelligent Web Applications, 2003, pp. 59 – 70.

[12] F. H.Wang, S. M.Thao, "A Study on Personalized Web Browsing Recommendation based on Data Mining and Collaborative Filtering Technology", Proceedings of national computer symposium, Taiwan, 2003, pp. 18–25.

[13] M. Gery, H. Haddad, "Evaluation of Web Usage Mining Approaches for User's Next Request Prediction", Proceedings of the fifth ACM international workshop on Web information and data management, 2003, pp. 74–81.

[14] B. Mobasher, R. Cooley, J. Srivastava, "Automatic Personalization based on Web Usage Mining", Communications of the ACM, 43 (8), 2000, pp. 142-151.

[15] M. Gery, H. Haddad, "Evaluation of Web Usage Mining Approaches for User's Next Request Prediction", Proceedings of the fifth ACM international workshop on Web information and data management, 2003, pp. 74–81.

[16] B. Mobasher, H. Dai, T. Luo, M. Nakagawa, "Effective Personalization based on Association Rule Discovery from Web Usage Data", In Proceedings of the 3rd ACM Workshop on Web Information and Data Management (WIDM01), Atlanta, 2001.

[17] A. Demiriz, "Enhancing Product Recommender Systems on Sparse Binary Data", Data Mining and Knowledge Discovery, 2003.

[18] C.H. Cai, A.W.C. Fu, C.H. Cheng, W.W. Kwong," Mining Association Rules with Weighted Items", In Database Engineering and Applications Symposium, Proceedings IDEAS'98, July 1998, pp. 68 – 77.

[19] F. Tao, F. Murtagh, M. Farid," Weighted Association Rule Mining using Weighted Support and Significance Framework", In Proceedings of the 9th SIGKDD Conference, 2003.

[20] C. Shahabi, A. Zarkesh, J. Abidi, V. Shah, "Knowledge Discovery from User's Web-Page Navigation", In Proceedings of the 7th IEEE Intl, Workshop on Research Issues in Data Engineering, 1997.

[21] J. Li, O. R. Zaiane, " Combining Usage, Content and Structure Data to Improve Web Site Recommendation", 5th International Conference on Electronic Commerce and Web, 2004.

[22] M. R. Meybodi, H. Beigy, "Solving Stochastic Shortest Path Problem Using Monte Carlo Sampling Method: A Distributed Learning Automata Approach", Springer-Verlag Lecture Notes in Advances in Soft Computing, 2003, pp. 626-632.

[23] J. M. Kleinberg, "Authoritative Sources in a Hyperlinked Environment", Journal of the ACM, 46(5), 1999, pp. 604–632.

[24] R. Cooley, B. Mobasher, J. Srivastava, "Data Preparation for Mining World Wide Web Browsing Patterns", In Journal of Knowledge and Information Systems, 1(1), 1999, pp. 5–32.

[25] P.K. Chan, "A Non-Invasive Learning Approach to Building Web User Profiles", in: Workshop on Web usage analysis and user profiling, Fifth International Conference on Knowledge Discovery and Data Mining, San Diego, 1999.

[26] S. Dumais, T. Joachims, K. Bharat, A. Weigend, "Implicit Measures of User Interests and Preferences", 2003 workshop report: ACM SIGIR Forum, 2003.

[27] M. Morita, Y. Shinoda, "Information Filtering based on User Behavior Analysis and Best Match Text Retrieval", in: Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Springer-Verlag New York, Inc., Dublin, Ireland, 1994, pp. 272–281.

[28] Y. Liang, L. Chunping, " Incorporating Pageview Weight into an Association-Rule-Based Web Recommendation System", Springer-Verlag Berlin Heidelberg, AI 2006, LNAI 4304, 2006, pp. 577 – 586.

[29] K. Narendra, M. A. L. Thathachar, "Learning Automata: An Introduction", Prentice Hall, Englewood Cliffs, New Jersey, 1989.

[30] M. A. L. Thathachar, R. Harita Bhaskar, "Learning Automata with Changing Number of Actions", IEEE Transactions on Systems Man and Cybernetics, vol. 17, no. 6, Nov. 1987, pp. 1095-1100.

[31] J. Srivastava, R. Cooley, M. Deshpande, P. Tan, "Web Usage Mining: Discovery and Applications of Usage Patterns from Web Data", SIGKDD Explorations, 1(2), 20001, pp.2-23.

[32] R. Agrawal, R. Srikant, "Fast Algorithms for Mining Association Rules in Large Databases", In Proceedings of the 20th International Conference on Very Large Data Bases VLDB'94, Santiago, 1994, pp. 487-499.