WebPersonalizer: A Server-Side Recommender System Based on Web Usage Mining

Bamshad Mobasher
Department of Computer Science, DePaul University, Chicago, IL
mobasher@cs.depaul.edu

Abstract

Existing approaches to Web personalization often rely heavily on explicit and subjective user input resulting in static profiles which are prone to biases. In this paper we present a usage-based Web personalization system, called *WebPersonalizer*, drawing heavily upon Web mining techniques, making the personalization process automatic, and dynamic. The system architecture separates the offline tasks of data preparation and Web usage mining, and the online recommendation engine. At the heart of the system is a technique based on clustering of user transactions which allows for the discovery of effective aggregate usage profiles. We discuss how the discovered aggregate profiles can be used in conjunction with the current status of an ongoing Web activity to perform real-time personalization. The Web usage mining approach allows a site to provide effective personalization using anonymous and implicit user behavioral patterns without relying on subjective or personally identifying user input.

Keywords: Data Mining, Personalization, Web Usage Mining, Clustering.

1 Introduction

E-commerce activity that involves the end user is undergoing a significant revolution. The ability to track users' browsing behavior down to individual mouse clicks has brought the vendor and end customer closer than ever before. It is now possible for a vendor to personalize the product message for individual customers at a massive scale, a phenomenon that is being referred to as *mass customization.* This type of personalization is, in fact, applicable to any Web browsing activity. *Web personalization* can be described, as any action that makes the Web experience of a user personalized to the user's taste or preferences. The experience can be something as casual as browsing the Web or as (economically) significant as trading stocks or purchasing a car. The actions can range from simply making the presentation more pleasing to an individual to anticipating the needs of the user and providing the right information, as well as performing a set of routine book-keeping functions automatically. Principal elements of Web personalization include modeling of Web objects (pages, etc.) and subjects (users), categorization of objects and subjects, matching between and across objects and/or subjects, and determination of the set of actions to be recommended for personalization. User preferences may be obtained explicitly, or by passive observation of users over time as they interact with the system.

Current approaches to Web personalization generally fall into three major categories: manual decision rule systems, collaborative filtering systems, and content-based filtering agents. Manual decision rule systems, such as Broadvision (www.broadvision.com), allow Web site administrators to specify rules based on user demographics or static profiles (collected through a registration process). The rules are used to affect the content served to a particular user. Collaborative filtering systems such as Net Perceptions (www.netperceptions.com) typically take explicit information in the form of user ratings or preferences, and through a correlation engine, return information that is predicted to closely match the user's preferences. Content-based filtering systems such as those used by WebWatcher [8] and client-side agent

Letizia [10] generally rely on personal profiles and content similarity of Web documents to these profiles to make recommendations.

There are several well-known drawbacks to these traditional content-based or rule-based filtering techniques for personalization. The type of input is often a subjective description of the users by the users themselves, and thus is prone to biases. The profiles are often static, obtained through user registration, and thus the system performance degrades over time as the profiles age. Furthermore, using content similarity alone as a way to obtain aggregate profiles may result in missing important relationships among Web objects based on their usage. Collaborative filtering [9, 7, 19], has tried to address some of these issues, and, in fact, it is the predominant commercial approach in most successful e-commerce systems. These techniques generally involve matching, in real time, the ratings of a current user for objects (e.g., movies or products) with those of similar users (nearest neighbors) in order to produce recommendations on other objects not yet rated by the user. However, collaborative filtering techniques have their own potentially serious limitations. For instance, as noted in recent studies [13], it becomes hard to scale collaborative filtering techniques to a large number of items (e.g., pages or products), while maintaining reasonable prediction performance and accuracy. Part of this is due to the increasing sparsity in the ratings data as the number of items increase, as well as due to the increasing computational cost of determining user to user correlation in real time for a large number of items and users. Furthermore, collaborative filtering usually performs best when explicit non-binary user ratings for similar objects are available. In many Web sites, however, it may be desirable to integrate the personalization actions throughout the site involving different types of objects, including navigational and content pages, as well as implicit product-oriented user events such as shopping cart changes, or product information requests.

Several recent proposals have explored Web usage mining as an enabling mechanism to overcome some of the problems associated with more traditional techniques [11, 23, 12] or as a mechanism for improving and optimizing the structure of a site [14, 6, 16]. Data mining techniques, such as clustering, have also been shown to improve the scalability and performance of collaborative filtering techniques [13]. In general, Web usage mining systems [22, 5, 3, 15] run any number of data mining algorithms on usage or clickstream data gathered from one or more Web sites in order to discover interesting patterns in the navigational behavior of users. For an up-to-date survey of Web usage mining techniques and systems see [17].

However, the discovery of patterns from usage data, such as association rules, sequential patterns, and clusters of user sessions or pages, by itself is not sufficient for performing the personalization tasks. The critical step is the effective derivation of good quality and useful (i.e., actionable) "aggregate profiles" from these patterns. The discovery of aggregate usage profiles, through clustering as well as other Web mining techniques, has been explored by several research groups [21, 20, 18, 14, 12]. However, in all of these cases, the frameworks proposed for the discovery of profiles have not been extended to show how these profiles can be used as an integrated part of recommender systems for personalization. In the case of [14], aggregate usage profiles were used to automatically synthesize alternative static index pages for a site.

In this paper we describe the design and implementation of a usage-based Web personalization system, called WebPersonalizer, which takes into account the full spectrum of Web mining techniques and activities. The system heavily uses data mining techniques, thus making the personalization process both automatic and dynamic, and hence up-to-date. Specifically, we discuss the necessary data preparation tasks for preprocessing of Web usage logs and grouping URL references into units of semantic activity called *user transactions*. We then describe our technique for extracting aggregated usage knowledge, based on transaction clustering, that would be suitable for the purpose of Web personalization. We also describe how Web Personalizer combines this knowledge with the current status of an ongoing Web activity to perform real-time personalization. Finally, we provide an experimental evaluation of the system as a whole, and the underlying profile discovery technique, using real Web usage data.

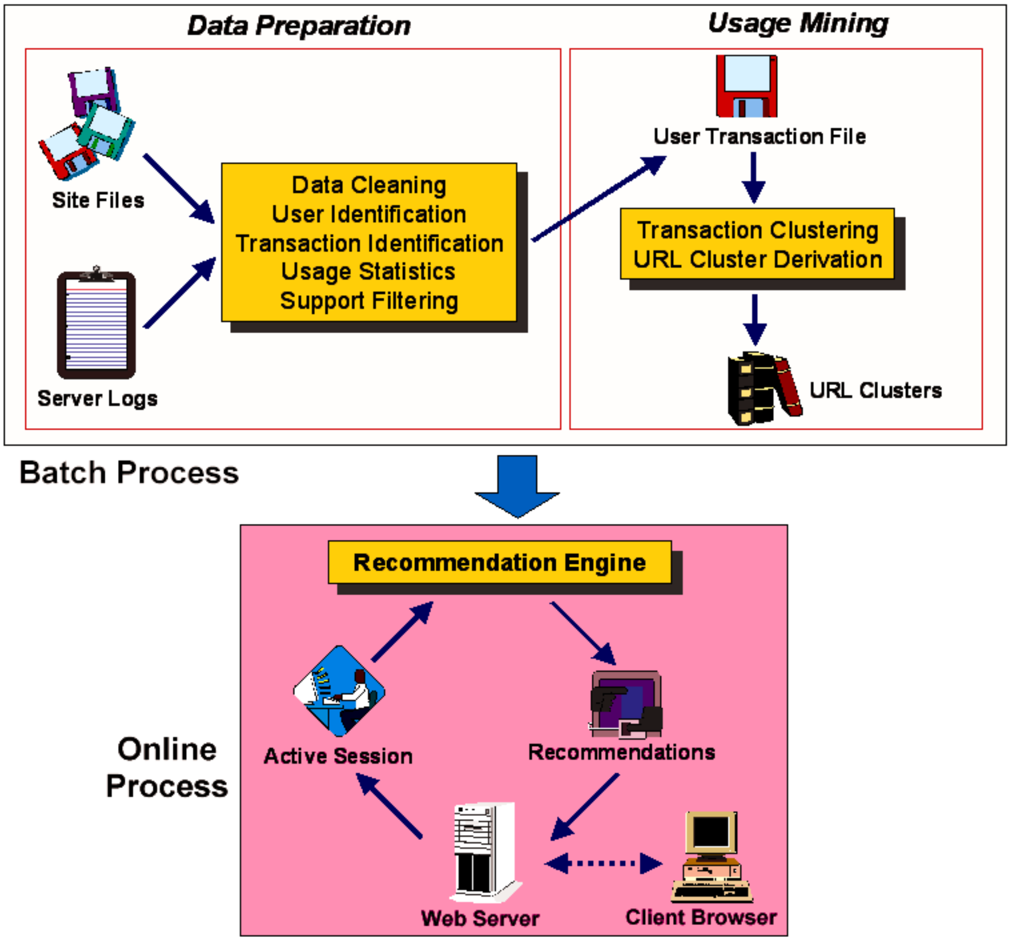


Figure 1. The Architecture of the WebPersonalizer System.

2 Mining Usage Data for Web Personalization

The overall process of usage-based Web personalization can be divided into two components. The offline component is comprised of the data preparation tasks resulting in a user transaction file, and the specific usage mining tasks, which in our case involves the discovery of clusters from user transactions and the derivation of *URL clusters* from the transaction clusters. The online component of the system provides dynamic recommendations to users based on their current navigational activity. In the online component, the Web server keeps track of the active session as the user browser makes HTTP requests, and the recommendation engine matches this active session with the URL clusters to compute a set of recommended URLs. The recommendation set is then added to the requested page as a set of links before the page is sent to the client browser. A generalized architecture for the WebPersonalizer system is depicted in Figure 1. In the rest of this section we discuss specific aspects of the offline components of the system, namely the preprocessing tasks and the details of deriving URL clusters from user transactions. In the next section we present recommendation engine.

2.1 Data Preparation Tasks

In the data preparation stage, initially, the raw server logs must be cleaned to remove redundant pages (e.g., image and sound files), leaving only one entry per *pageview* (the group of files that are referenced due to a single click of the user). This also includes handling pageviews that have multiple frames, and dynamic pages that have the same template name for multiple pageviews. The next critical step is the identification of a set of unique user sessions from the server log data. This is not a trivial task since log entries do not include unique user identifiers, and because of proxy servers and local browser caching, there may be missing references or references with the same identifying fields (IP address) that belong to multiple users. Furthermore, techniques such as client-side cookies are not always reliable and can be turned off by users. The WebPersonalizer system uses several simple heuristics, as detailed in [5], using the referrer and agent fields of a Server log, in order to identify user sessions and infer missing references with relative accuracy in the absence of other identifying information. It is also necessary to map references in the session files to the physical site topology in order to assure that we do not consider "out-of-date" or non-existent pages which may otherwise appear in the usage data.

Pageview identification is the task of determining which page file accesses contribute to a single browser display, and is heavily dependent on the intra-page structure, and hence requires detailed site structure information. Not all pageviews are relevant for specific mining tasks. Furthermore, among the relevant pageviews some may be more significant than others. The significance of a pageview may depend on usage, content and structural characteristics of the site, as well as prior domain knowledge specified by the site designer and the data analyst. For example, in an in an e-commerce site pageviews corresponding to product-oriented events (e.g., shopping cart changes or product information views) may be considered more significant than others. Similarly, in a site designed to provide content, content pages may be weighted higher than navigational pages. In order to provide a flexible framework for a variety of data mining activities a number of attributes must be recorded with each pageview. These attributes include the pageview id (normally a URL uniquely representing the pageview), average duration, static pageview type (e.g., content, navigational, product view, index page, etc.), and other meta-data.

Once unique user sessions have been identified, the usage data must be transformed into a set of user transactions, each containing a subset of references by a user to relevant pageviews. The goal of transaction identification is to dynamically create meaningful clusters of references for each user, based on an underlying model of the user's browsing behavior. This allows each page reference to be categorized as a *content* or *navigational* reference for a particular user. Content references can be further classified according to page types or the type of user activity (e.g., product purchases). Thus each pageview object will also have several associated attributes which will be set dynamically for each instance of that pageview during session or transaction identification. These include, the start time, the duration (for that particular user), and the usage type (e.g., content, navigational, or hybrid).

Finally, the transaction file can be further filtered by removing very low support or very high support pageview references (i.e., references to those pageviews which do not appear in a sufficient number of transactions, or those that are present in nearly all transactions). This type of *support filtering* can be useful in eliminating noise from the data, such as that generated by shallow navigational patterns of "non-active" users, and pageview references with minimal knowledge value for the purpose of personalization.

The above preprocessing tasks ultimately result in a set of n pageview records identified uniquely by their URL, $U = \{url_1, url_2, \ldots, url_n\}$; and a set of m user transactions $T = \{t_1, t_2, \ldots, t_m\}$, where each $t_i \in T$ is a non-empty subset of U. To facilitate various data mining operations such as clustering, we view each transaction t as an n-dimensional vector over the space of pageview references, i.e.,

$$t = \langle w(url_1, t), w(url_2, t), \ldots, w(url_n, t)\rangle,$$

where $w(url_i, t)$ is a weight, in the transaction t, associated with the pageview represented by $url_i \in U$. The weights can be determined in a number of ways, for example, binary weights can be used to represent existence or non-existence of a product-purchase or a documents access in the transaction. On the other hand, the weights can be a function of the duration of the associated pageview in order to capture the user's interest in a content page. The weights may also, in part, be based on domain-specific significance weights assigned by the analyst.

2.2 Transaction Clustering and Derivation of URL Clusters

The transaction file obtained in the data preparation stage can be used as the input to a variety of data mining algorithms such as the discovery of association rules [1] or sequential patterns [2], clustering, and classification. However, the discovery of patterns from usage data by itself is not sufficient for performing the personalization tasks. The critical step is the effective derivation of good quality and useful (i.e., actionable) "aggregate profiles" from these patterns.

In the WebPersonalizer system we represent usage profiles as a overlapping, weighted collections of pageview records. Each item in a usage profile represents a relevant pageview, and can have an associated weight representing its significance within the profile. Each pageview record in a profile is uniquely identified by a pageview URL. We call the resulting structures *URL clusters*. Ideally, URL clusters capture an aggregate view of the behavior of subsets of users based their interests and/or information needs. They are overlapping because many users may have common interests up to a point (in their navigational history) beyond which their interests diverge. The profiles represented as URL clusters can be viewed as an ordered collection (if the goal is to capture the navigational path profiles followed by users [15]), or as unordered (if the focus is on capturing associations among specified content or product pages). Based on the information contained in each pageview record discussed earlier, other types of constraints can also be imposed on profiles. Another advantage of this representation for usage profiles is that these profiles, themselves, can be viewed as pageview vectors, thus facilitating the task of matching a current user session with similar profiles using standard vector operations. In the rest of this section we present our technique for the derivation of URL clusters (representing aggregate usage profiles), based on clustering user transactions.

Given the mapping of user transactions into a multi-dimensional space as vectors of pageview URLs, standard clustering algorithms, such as *k-means*, generally partition this space into groups of transactions that are close to each other based on a measure of distance or similarity. In WebPersonalizer we use multivariate k-means clustering method for this task. Such a clustering will result in a set of transaction clusters, $TC = \{c_1, c_2, \ldots, c_k\}$, where each c_i is a subset of the set of transactions T. Dimensionality reduction techniques may be employed to focus only on relevant or significant features (which in this case are pageview URLs). For example, support filtering discussed earlier can provide an effective dimensionality reduction method while actually improving clustering results. Ideally, each cluster represents a group of users with similar navigational patterns. However, transaction clusters by themselves are not an effective means of capturing an aggregated view of common user profiles. Each transaction cluster may potentially contain thousands of user transactions involving hundreds of pageview references. Our ultimate goal in clustering user transactions is to reduce these clusters into URL clusters which, as note above, are weighted collections of pageviews URLs.

For each cluster $c \in TC$, we compute the mean vector m_c. The mean value for each pageview in the mean vector is computed by finding the ratio of the sum of the pageview weights across transactions to the total number of transactions in the cluster. To obtain the usage profile, the weights are normalized so that the maximum weight in each usage profile is 1, and low-support pageviews (i.e. those with mean value below a certain threshold μ) are filtered out. Thus, a usage profile associated with a transaction cluster c, is the set of all pageviews whose weight is greater than or equal to μ. In particular, if we simply use binary weights for pageviews, and the threshold μ is set at 0.5, then each profile will contain only those pageviews which appear in at least 50% of transactions within its associated transaction cluster.

To summarize, given a transaction cluster c, we construct a corresponding URL cluster, pr_c, representing an aggregate profile, as a set of pageview-weight pairs:

$$pr_c = \{\langle url, weight(p, pr_c)\rangle \mid url \in U, weight(url, pr_c) \geq \mu\},$$

where the significance weight, *weight*(*url*, pr_c), of the pageview represented by *url* within the usage profile pr_c is:

$$weight(url, pr_c) = \frac{1}{|c|} \cdot \sum_{t \in c} w(url, t)$$

and $w(url, t)$ is the weight of the pageview represented by *url* in transaction $t \in c$. Each profile, in turn, can be represented as vectors in the original *n*-dimensional space.

3 The Recommendation Engine

The recommendation engine is the online component of the WebPersonalizer system. Its task is to compute a *recommendation set* for the current session, consisting of links to pages that the current user may want to visit based on similar usage patterns. The recommendation set essentially represents a "short-term" view of potentially useful links based on the user's navigational activity through the site. These recommended links are then added to the last page in the session accessed by the user before that page is sent to the user browser.

In general there are several factors that we would like to consider in determining the recommendation set. These factors include:

1. the matching criteria for each cluster based on the its similarity to the current active session;
2. whether the candidate URLs for recommendation have already been visited in the current session;
3. a short-term history depth for the current user representing the portion of the user's activity history that we should consider relevant for the purpose of making recommendations; and
4. the length of the physical link path from the active session window to the candidate URL.

Maintaining a history depth may be important because most users navigate several paths leading to independent pieces of information within a session. In many cases these *sub-sessions* have a length of no more than 3 or 4 references. In such a situation, it may not be appropriate to use references a user made in a previous sub-session to make recommendations during the current sub-session. We capture the user history depth within a sliding window over the current session. The sliding window of size *n* over the active session allows only the last *n* visited pages to influence the recommendation value of items in the recommendation set. The notion of a sliding session window is similar to that of *N-grammars* discussed in [4]. Structural characteristics of the site or prior domain knowledge can also be used to associate an additional measure of significance with each pageview in the user's active session. For instance, the site owner or the site designer may wish to consider certain page types (e.g., content versus navigational) or product categories as having more significance in terms of their recommendation value. In this case, significance weights can be specified as part of the domain knowledge. The WebPersonalizer system automatically determines the optimum window size based on the average user transaction length identified during the preprocessing stage.

WebPersonalizer also weights a URL recommendation higher, if it is farther away form the current active session. To capture this notion, we maintain a directed graph, *G*, representing the topology of the site. The *physical link distance* between two URLs u_1 and u_2 is the length of a minimal path from u_1 to u_2 in this site graph. The physical link distance between the active session *s* and a URL $u \notin$ s is denoted by $dist(u,s,G)$, which is defined as the smallest physical link distance between *u* and any of the URLs in *s*. The *link distance* is defined as $ldf(u,s) = \log(dist(u,s,G))+1$. If the URL *u* is in the active session, then $ldf(u,s)$ is taken to be 0. We take the log of the link distance so that it does not count too heavily compared to item weights within clusters.

Each of the URL clusters obtained in the mining stage can be viewed as a virtual user profile indicating how various groups of users may access a set of links in the site within their respective user transactions. The representation of URL clusters as sets of pageview-weight pairs, allows us to treat the URL clusters *n*-dimensional vectors over the space of pageview URLs in the site. Thus, given a URL cluster *C*, we can represent *C* as a vector

$$C = \{w_1^C, w_2^C, \ldots, w_n^C\}$$

where

$$w_i^C = \begin{cases} weight(url_i, C), & \text{if } url_i \in C \\ 0, & \text{otherwise} \end{cases}$$

Similarly, the current active session S is also represented as a vector $S = \langle s_1, s_2, \ldots , s_n \rangle$, where s_i is a significance weight associated with the corresponding pageview reference, if the user has accessed url_i in this session, and $s_i = 0$, otherwise. In our experiments, discussed in the next section, we simply used binary weighting for the active session. We compute the *profile matching score* using the normalized cosine similarity measure for vectors:

$$match(S,C) = \frac{\sum_k w_k^C \cdot S_k}{\sqrt{\sum_k (S_k)^2 \times \sum_k (w_k^C)^2}}$$

Note that the matching score is normalized for the size of the clusters and the active session. This corresponds to the intuitive notion that we should see more of the user's active session before obtaining a better match with a larger cluster representing a user profile. Given a URL cluster C and an active session S, a recommendation score, $Rec(S, u)$, is computed for each URL u in C as follows:

$$Rec(S,u) = \sqrt{weight(u,C) \cdot match(S,C)} \times ldf(S,u)$$

If the URL u is in the current active session, then its recommendation value is set to zero. We obtain the usage recommendation set, $UREC(S)$, for current active session S by collecting from each URL cluster all URLs whose recommendation score satisfies a minimum recommendation threshold ρ, i.e.,

$$UREC(S) = \{u_i^C \mid C \in UC, u_i^C \in C, \text{ and } Rec(s, u_i^C) \geq \rho\},$$

where UC is the collection of all URL Clusters. Furthermore, for each pageview that is contributed by several usage profiles, we use its maximal recommendation score from all of the contributing profiles.

4 Experimental Results

We used the access logs from the Web site of the Association for Consumer Research (ACR) Newsletter (www.acr-news.org) for our experiments. The site includes a number of calls-for-papers for a variety of conferences and Journals related to consumer behavior and marketing, an archive of editorial articles, and a variety of pages related to organizational matters. After preprocessing and removing references by Web spiders, the initial log file (from June 1988 through June 1999), produced a total of 18342 transactions using the transaction identification process. The total number of URLs representing pageviews was 112. Support filtering was used to eliminate pageviews appearing in less than 0.5% or more than 80% of transactions (including the site entry page). Furthermore, for these experiments we eliminated short transactions, leaving only transactions with at least 5 references (which was the average transaction size in the whole data set). Approximately 25% of the transactions from the final set were randomly selected as the evaluation set, and the remaining portion was used as the training set to which we applied the profile generation methods described earlier. The total number of remaining pageview URLs in the training and the evaluation sets was 62.

Weight	Pageview ID
1.00	Conference Update
0.89	ACR 1999 Annual Conference
0.82	CFP: ACR 1999 Asia-Pacific Conference
0.83	CFP: ACR 1999 European Conference
0.56	ACR News Special Topics

Weight	Pageview ID
1.00	Call for Papers
1.00	CFP: Journal of Consumer Psychology I
0.72	CFP: Journal of Consumer Psychology II
0.61	CFP: Conf. on Gender, Marketing, Consumer Behavior
0.54	CFP: ACR 1999 Asia-Pacific Conference
0.50	Conference Update
0.50	Notes From the Editor

Weight	Pageview ID
1.00	President's Column - December, 1997
0.78	President's Column - March, 1998
0.62	Online Archives
0.50	ACR News Updates
0.50	ACR President's Column
0.50	From the Grapevine

Table 1. Examples of URL clusters representing aggregate usage profiles.

The profile derivation method discussed earlier resulted in a total of 28 URL clusters. Based on the average session size, the system automatically chose a session window of size of 3 references. However, we ran the recommendation engine using windows sizes of 2 and 3 in order to determine the impact of window size on recommendation accuracy. Table 1 show 3 of the discovered URL clusters for the site. Only pageview URLs with weights of at least 0.5 have been shown in each profile. The first URL cluster in Table 1 represents the profiles of users who are primarily interested in general ACR sponsored conferences. The second cluster, while containing some overlap with the first, seems to capture the activity of users whose interests are more focused on specific conferences or journal related to marketing and consumer behavior. Finally, the third cluster captures the activity of users interested in news items as well as specific columns that appear in the "Online Archives" section of the ACR site.

In order to evaluate the recommendation effectiveness we use the following basic methodology. For a given transaction t, and an active session window size n, we randomly chose $|t|-n+1$ groups of items from the transaction as the surrogate active session window. For each of these active sessions, we produced a recommendation set and compared the set to the remaining items in the transaction by computing the percentage of visited pages for which a recommendation was produced. The final score for transaction t is the mean score over all of the $|t|-n+1$ surrogate active sessions. Finally, the mean over all transactions in the training set was computed as the evaluation score.

To determine a recommendation set based on an active session, we varied the recommendation threshold from 0.2 to 0.9. A page is included in the recommendation set only if it has a recommendation score above this threshold. Clearly, fewer recommendations are produced at higher thresholds, while higher evaluation scores are achieved at lower thresholds (with larger recommendation sets). Ideally, we would like the recommendation engine to produce few but highly relevant recommendations. Table 2 shows the results produced by the recommendation engine using a session window size of 2. For example, at a threshold of 0.7, the system produced an evaluation score of 0.82 with an average recommendation set size of 10 over all trials. Roughly speaking, this means that on average 82% of unique pages actually visited by users in the evaluation set transactions matched the top 10 recommendations produced by the system.

Threshold	Eval. Score	Avg. Number of Recs.
0.9	0.54	3.8
0.8	0.75	7.0
0.7	0.82	10.0
0.6	0.88	13.3
0.5	0.92	16.7
0.4	0.95	21.6
0.3	0.98	25.8
0.2	0.98	28.7

Table 2. Evaluation scores and the average size of the recommendation set produced by the recommendation engine using a session window size of 2.

In order to compare the overall relative recommendation accuracy for different window sizes, the evaluation score percentage was divided by the size of the recommendation set. The overall score on this scale would be higher when the recommendation engine is able to produce relatively small recommendation set with higher average evaluation scores for that recommendation set. Thus, a higher number according to this measure corresponds to better overall performance by the recommendation engine. The results for session window sizes of 2 and 3 are depicted in Figures 2. As the results indicate, the overall performance or the recommendation engine increased as the window size wad enlarged (thus allowing the system to utilize a larger portion of the user's active session).

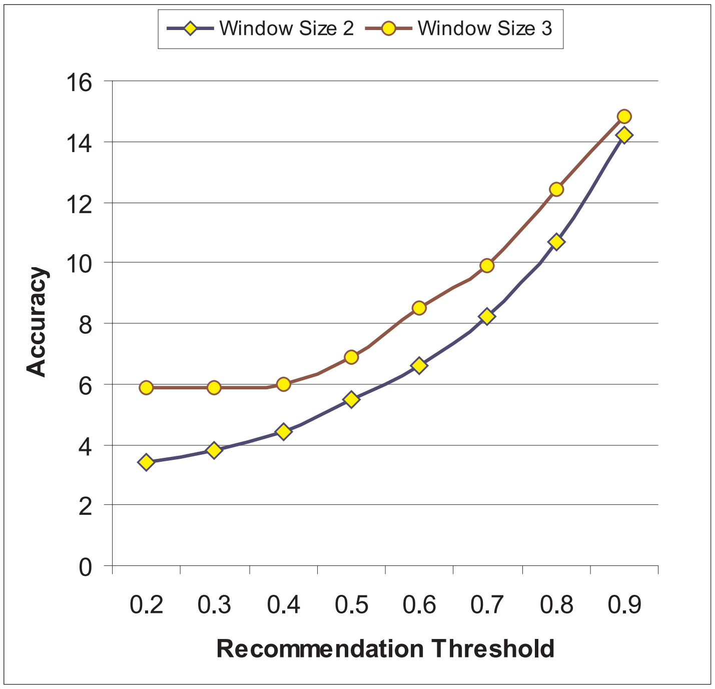
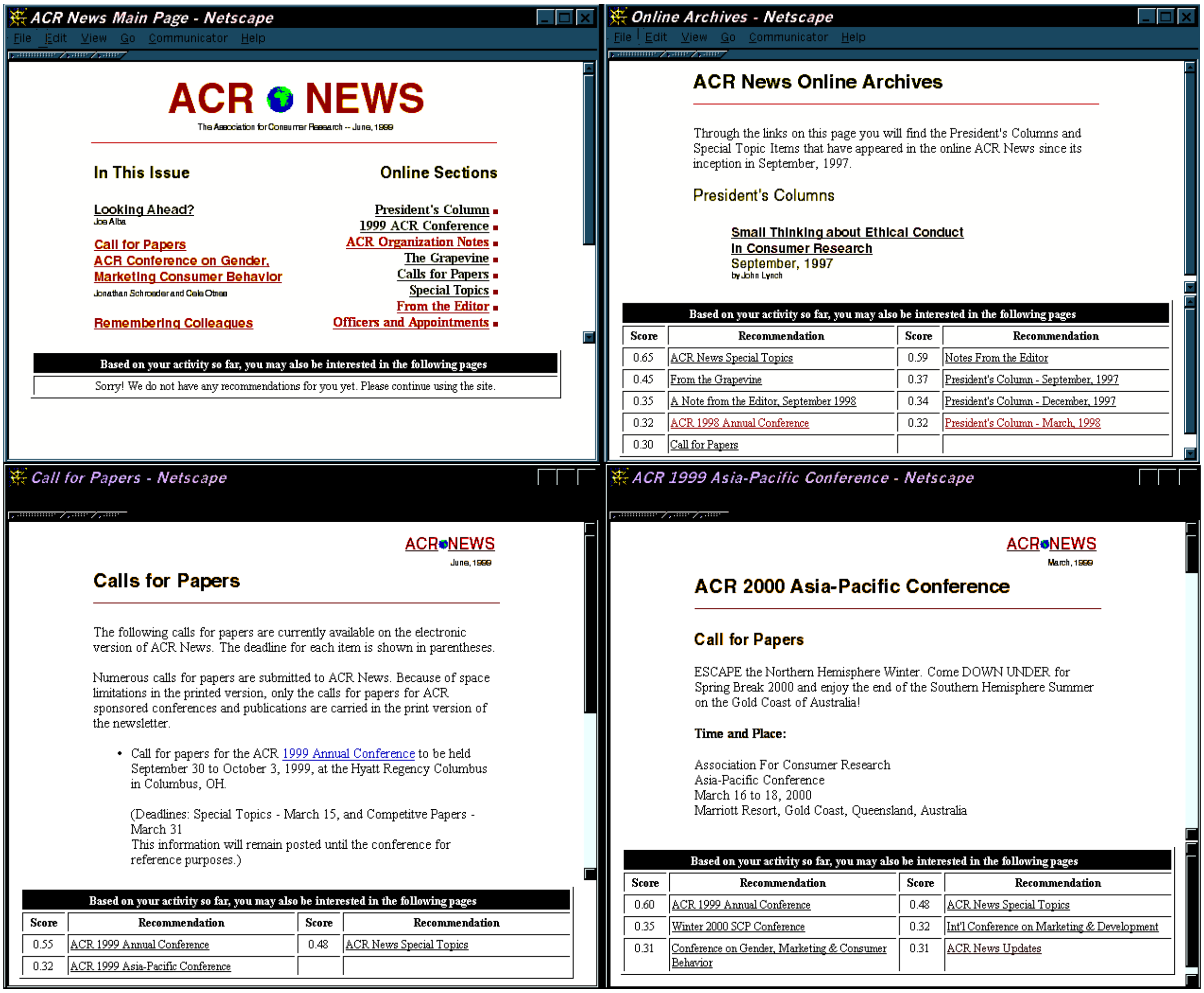


Figure 2. Comparison of recommendation accuracy with session window sizes 2 and 3.


ACR News Main Page - Netscape
File Edit View Go Communicator Help

ACR NEWS

The Association for Consumer Research -- June, 1999

In This Issue

Looking Ahead?
Joe Alba

Call for Papers
ACR Conference on Gender, Marketing Consumer Behavior
Jonathan Schroeder and Cele Otnes

Remembering Colleagues

Online Sections

President's Column ■
1999 ACR Conference ■
ACR Organization Notes ■
The Grapevine ■
Calls for Papers ■
Special Topics ■
From the Editor ■
Officers and Appointments ■

Based on your activity so far, you may also be interested in the following pages

Sorry! We do not have any recommendations for you yet. Please continue using the site.

Online Archives - Netscape
File Edit View Go Communicator Help

ACR News Online Archives

Through the links on this page you will find the President's Columns and Special Topic Items that have appeared in the online ACR News since its inception in September, 1997.

President's Columns

Small Thinking about Ethical Conduct In Consumer Research
September, 1997
by John Lynch

Based on your activity so far, you may also be interested in the following pages

Score	Recommendation	Score	Recommendation
0.65	ACR News Special Topics	0.59	Notes From the Editor
0.45	From the Grapevine	0.37	President's Column - September, 1997
0.35	A Note from the Editor, September 1998	0.34	President's Column - December, 1997
0.32	ACR 1998 Annual Conference	0.32	President's Column - March, 1998
0.30	Call for Papers		

ACR NEWS
June, 1999

Calls for Papers

The following calls for papers are currently available on the electronic version of ACR News. The deadline for each item is shown in parentheses.

Numerous calls for papers are submitted to ACR News. Because of space limitations in the printed version, only the calls for papers for ACR sponsored conferences and publications are carried in the print version of the newsletter.

- Call for papers for the ACR 1999 Annual Conference to be held September 30 to October 3, 1999, at the Hyatt Regency Columbus in Columbus, OH.

  (Deadlines: Special Topics - March 15, and Competitve Papers - March 31
  This information will remain posted until the conference for reference purposes.)

Based on your activity so far, you may also be interested in the following pages

Score	Recommendation	Score	Recommendation
0.55	ACR 1999 Annual Conference	0.48	ACR News Special Topics
0.32	ACR 1999 Asia-Pacific Conference		

ACR NEWS
March, 1999

ACR 2000 Asia-Pacific Conference

Call for Papers

ESCAPE the Northern Hemisphere Winter. Come DOWN UNDER for Spring Break 2000 and enjoy the end of the Southern Hemisphere Summer on the Gold Coast of Australia!

Time and Place:

Association For Consumer Research
Asia-Pacific Conference
March 16 to 18, 2000
Marriott Resort, Gold Coast, Queensland, Australia

Based on your activity so far, you may also be interested in the following pages

Score	Recommendation	Score	Recommendation
0.60	ACR 1999 Annual Conference	0.48	ACR News Special Topics
0.35	Winter 2000 SCP Conference	0.32	Int'l Conference on Marketing & Development
0.31	Conference on Gender, Marketing & Consumer Behavior	0.31	ACR News Updates

Figure 3. WebPersonalizer Recommendations from the Demonstration Site

Figure 3 depicts a typical interaction of user with the demonstration site using the WebPersonalizer recommendation engine. The top frame in each window contains the actual page contents from the site, while the bottom frame contains the recommended links. When the user clicks on a link in either frame, the top frame will display the content of the requested page, and the bottom frame is dynamically updated to include the new recommendations. Initially the system does not provide any recommendations until the user has navigated through at least two pages. The top-right panel in Figure 2 shows the recommendations resulting after the user has followed a path to "President's Column" and then to "Online Archives." The recommendations include past President Columns and Editor's Notes often visited by users who have shown similar access patterns. The bottom panels show the results of the user navigation through "Conference Update," "Call for Papers," (left), and then to "1999 Asia Pacific Conference" (right). As can be seen in the Figure, user's intention of looking for more specific information will result in more specific recommendations. For example, in the left panel general recommendations are provided guiding the user to upcoming conferences and news items. When the user accesses a specific conference page, other specific conference information is presented as potentially interesting (e.g., "Winter 2000 SCP Conference" and "Int'l Conference on Marketing and Development").

5 Conclusions and Future Work

The Web is providing a direct communication medium between the vendors of products and services, and their clients. Coupled with the ability to collect detailed data at the granularity of individual mouse clicks, this provides a tremendous opportunity for personalizing the Web experience for clients. In this paper we have presented a Web personalization system based on Web usage mining which can automatically provide effective navigational pointers to a user based on the user's active session and the aggregate usage patterns of other similar users. Our future work in this area involves incorporating other usage mining techniques, such as the discovery of association rules and sequential patterns, into the recommendation process. Furthermore, we plan on conducting experiments with various types of transactions derived from user sessions, for example, to isolate specific types of "content" pages in the recommendation process. The latter task is particularly important in the context of electronic commerce, since the system can automatically guide users to particular product pages based on matching the user's interests with other similar user access patterns.

References

1. R. Agrawal and R. Srikant. Fast algorithms for mining association rules. *In Proceedings of the 20th VLDB conference*, pp. 487-499, Santiago, Chile, 1994.
2. R. Agrawal and R. Srikant. Fast algorithms for mining association rules. *In Proceedings of the 20th VLDB conference*, pp. 487-499, Santiago, Chile, 1994.
3. A. Buchner and M. D. Mulvenna. Discovering internet marketing intelligence through online analytical Web usage mining. *SIGMOD Record*, (4) 27, 1999.
4. E. Charniak. *Statistical language learning*. MIT Press, 1996.
5. R. Cooley, B. Mobasher, and J. Srivastava. Data preparation for mining World Wide Web browsing patterns. *Journal of Knowledge and Information Systems*, (1) 1, 1999.
6. R. Cooley, P-T. Tan., and J. Srivastava. WebSIFT: The Web site information filter system. In *Workshop on Web Usage Analysis and User Profiling (WebKKD99)*, San Diego, August 1999.
7. J. Herlocker, J. Konstan, A. Borchers, and J. Riedl. An algorithmic framework for performing collaborative filtering. To appear in *Proceedings of the 1999 Conference on Research and Development in Information Retrieval*, August 1999.
8. T. Joachims, D. Freitag, and T. Mitchell. Webwatcher: A tour guide for the world wide web. In *the 15th International Conference on Artificial Intelligence*, Nagoya, Japan, 1997.
9. J. Konstan, B. Miller, D. Maltz, J. Herlocker, L. Gordon, and J. Riedl. GroupLens: applying collaborative filtering to usenet news. *Communications of the ACM* (40) 3, 1997.
10. H. Lieberman. Letizia: An agent that assists web browsing. In *Proc. of the 1995 International Joint Conference on Artificial Intelligence*, Montreal, Canada, 1995.
11. B. Mobasher, R. Cooley, and J. Srivastava. Creating adaptive web sites through usage-based clustering of urls. In *IEEE Knowledge and Data Engineering Workshop (KDEX'99),* 1999.
12. O. Nasraoui, H. Frigui, A. Joshi, R. Krishnapuram. Mining Web access logs using relational competitive fuzzy clustering. To appear in *the Proceedings of the Eight International Fuzzy Systems Association World Congress*, August 1999.
13. M. O'Conner, J. Herlocker. Clustering items for collaborative filtering. In *Proceedings of the ACM SIGIR Workshop on Recommender Systems*, Berkeley, CA, 1999.
14. M. Perkowitz and O. Etzioni. Adaptive Web sites: automatically synthesizing Web pages. In *Proceedings of Fifteenth National Conference on Artificial Intelligence*, Madison, WI, 1998.
15. M. Spiliopoulou and L. C. Faulstich. WUM: A Web Utilization Miner. *In Proceedings of EDBT Workshop WebDB98*, Valencia, Spain, LNCS 1590, Springer Verlag, 1999.
16. M. Spiliopoulou, C. Pohle, and L. C. Faulstich. Improving the effectiveness of a Web site with Web usage mining. In *Workshop on Web Usage Analysis and User Profiling (WebKKD99)*, San Diego, August 1999.

17. J. Srivastava, R. Cooley, M. Deshpande, P-T. Tan. Web Usage Mining: Discovery and Applications of Usage Patterns from Web Data. To appear in *SIGKDD Explorations*, (1) 2, 2000.
18. S. Schechter, M. Krishnan, and M. D. Smith. Using path profiles to predict HTTP requests. In *Proceedings of 7th International World Wide Web Conference*, Brisbane, Australia, 1998.
19. U. Shardanand, P. Maes. Social information filtering: algorithms for automating "word of mouth." In *Proceedings of the ACM CHI Conference*, 1995.
20. C. Shahabi, A. Zarkesh, J. Adibi, and V. Shah. Knowledge discovery from users Web-page navigation. In *Proceedings of Workshop on Research Issues in Data Engineering*, Birmingham, England, 1997.
21. T. Yan, M. Jacobsen, H. Garcia-Molina, and U. Dayal. From user access patterns to dynamic hypertext linking. In Proceedings of the 5th International World Wide Web Conference, Paris, France, 1996.
22. O. R. Zaiane, M. Xin, and J. Han. Discovering web access patterns and trends by applying OLAP and data mining technology on web logs. In *Advances in Digital Libraries*, pp. 19-29, Santa Barbara, 1998.
23. P. S. Yu. Data mining and personalization technologies. In Int'l Conference on Database Systems for Advanced Applications (DASFAA99), April 1999, Hsinchu, Taiwan.