Creating Adaptive Web Sites Through Usage-Based Clustering of URLs

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

Robert Cooley, Jaideep Srivastava
Dept. of Computer Science, University of Minnesota, Minneapolis, MN
cooley@cs.umn.edu, srivasta@cs.umn.edu

1 Introduction

Web personalization has quickly moved from an added value feature to a necessity, particularly for large information services and sites that generate revenue by selling products. Web personalization can be viewed as using user preferences profiles to dynamically serve customized content to particular users. User preferences may be obtained explicitly, or by passive observation of users over time as they interact with the system. Principal elements of Web personalization include modeling of Web objects (pages, etc.) and subjects (users), matching between and across objects and/or subjects, and determination of the set of actions to be recommended for personalization. Existing approaches used by many Web-based companies, as well as approaches based on collaborative filtering (e.g., GroupLens [HKBR99] and Firefly [SM95]), rely heavily on human input for determining the personalization actions. This type of input is often a subjective description of the users by the users themselves, and thus prone to biases. Furthermore, the profile is static, and its performance degrades over time as the profile ages.

Recently, a number of approaches have been developed dealing with specific aspects of Web usage mining for the purpose of automatically discovering user profiles. For example, Perkowitz and Etzioni [PE98] proposed the idea of *optimizing* the structure of Web sites based co-occurrence patterns of pages within usage data for the site. Schechter et al [SKS98] have developed techniques for using path profiles of users to predict future HTTP requests, which can be used for network and proxy caching. Spiliopoulou et al [SF99], Cooley et al [CMS99], and Buchner and Mulvenna [BM99] have applied data mining techniques to extract usage patterns from Web logs, for the purpose of deriving marketing intelligence. Shahabi et al [SZA97], Yan et al [YJGD96], and Nasraoui et al [NFJK99] have proposed clustering of user sessions to predict future user behavior.

In this paper we describe an approach to usage-based Web personalization taking into account both the offline tasks related to the mining of usage data, and the online process of automatic Web page customization based on the mined knowledge. Specifically, we propose an effective technique for capturing common user profiles based on association-rule discovery and usage-based clustering. We also propose techniques for combining this knowledge with the current status of an ongoing Web activity to perform real-time personalization. Finally, we provide an experimental evaluation of the proposed techniques using real Web usage data.

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 include the discovery of association rules and the derivation of URL clusters based on user access patterns. Once the mining tasks are accomplished, the frequent itemsets and the URL clusters are used by the online component of the architecture to provide dynamic recommendations to users based on their current navigational activity. The Web server keeps track of the active user session as the user browser makes HTTP requests. The recommendation engine considers the active user session in conjunction with the URL clusters to compute a set of recommended URLs. The recommendation set is then added to the last requested page as a set of links before the page is sent to the client browser. A generalized architecture for

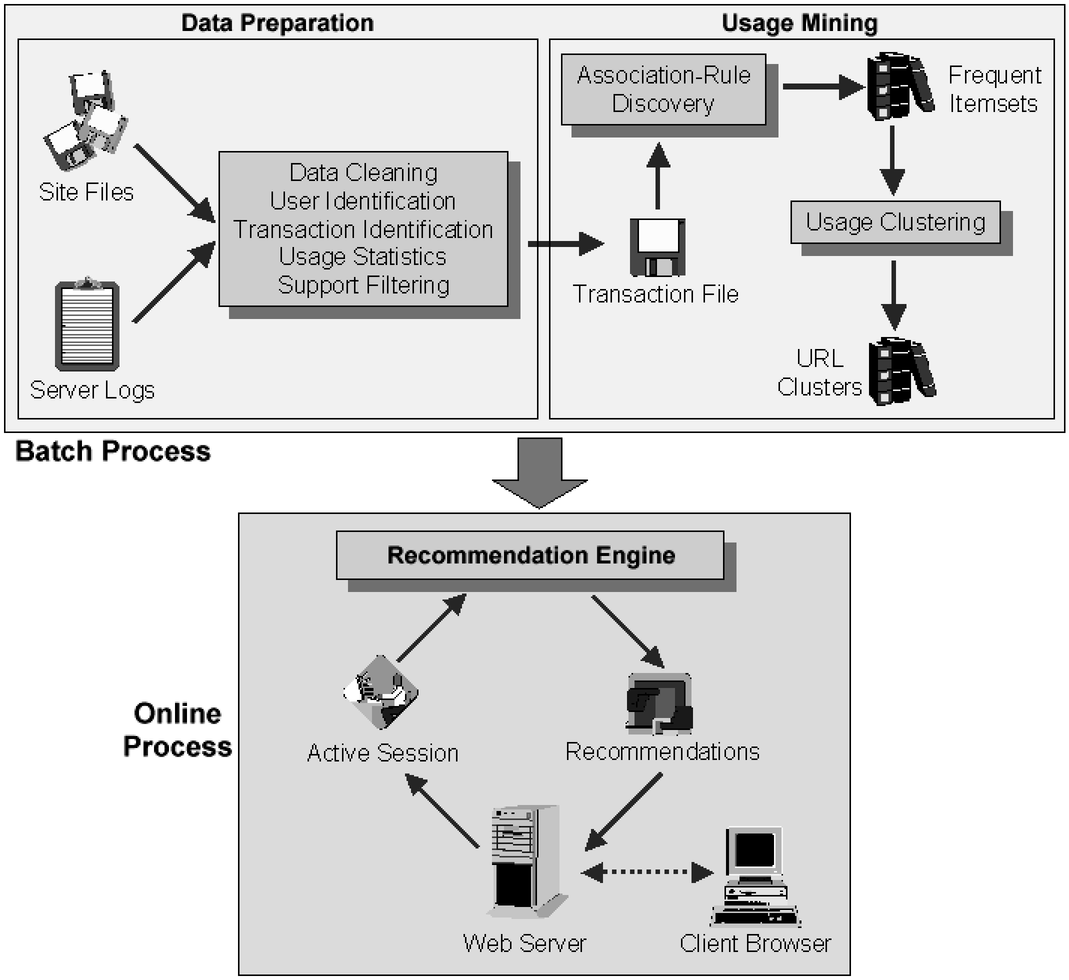

Figure 1. General Architecture for Usage-Based Web Personalization

usage-based personalization is depicted in Figure 1. In this section, we discuss the offline components of this architecture. The online customization process is presented in the next section.

2.1 Data Preparation Tasks

A critical step in effectively mining usage data for Web personalization is the cleaning and transformation of access log data, and the identification of a set of user sessions. Cleaning the server logs involves removing redundant references (e.g., image and sound files, multiple frames, and dynamic pages that have the same template), leaving only one entry per *page view*. It is also necessary to filter the log files by mapping the references to the site topology induced by physical links between pages. Client-side and proxy level caching often create impediments to the identification of unique user sessions. For example, in a Web server log, all requests from a proxy server have the same identifier, even though the requests potentially represent more than one user. Techniques such as the use of client-side cookies for user identification are not always practical due to privacy concerns of the users, or limitations of the capabilities of the Web server. In [CMS99] we proposed several simple heuristics using the referrer and agent fields of a Server log to identify user sessions and infer missing references with relative accuracy in the absence of additional information such as cookies.

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, each page reference can be categorized as a *content* reference, *auxiliary* (or *navigational*) reference, or *hybrid*. In this way different types of transactions can be obtained from the user session file, including content-only transactions involving references to content pages, and navigation-content transactions involving a mix of pages types. The details of methods for transaction identification are discussed in [CMS99]. Finally, the session file may be filtered to remove small transactions and low support references (i.e., URL references that are not supported by a specified number of user transactions). This type of *support filtering* is important in removing noise from the data, such as transactions corresponding to users who do not traverse the site.

Given the preprocessing steps outlined above, for the rest of this paper we assume that there is a set of n unique URLs $U = \{url_1, url_2, \ldots, url_n\}$, appearing in the preprocessed log, 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.

2.2 Usage Clustering Based on Frequent Itemsets

Traditional collaborative filtering techniques are often based on matching the current user's profile against clusters of similar profiles obtained by the system over time. Similar techniques have been proposed for obtaining user profiles from Web usage data by clustering user sessions [SZAS97, YJGD96]. However, session clusters by themselves are not an effective means of capturing an aggregated view of common user access patterns. Each session cluster may potentially contain thousands of user sessions involving hundreds of URL references. Furthermore, the computation of similarity or distance between user session vectors in the context of Web usage mining is not a trivial task, since feature weights based on the amount of time spent by a particular user on a page, or the number of references to a page within a session, are generally not good behavioral indicators in Web transactions.

In contrast to clustering user sessions, we directly compute overlapping clusters of URL references based on their co-occurrence patterns across user transactions. We call the URL clusters obtained in this way, *usage clusters*. Our experiments suggest that usage clusters tend to group together related items (based on user access patterns) across transactions, even if these transactions are themselves not deemed to be similar. This allows us to obtain clusters that potentially capture overlapping interests of different types of users. However, traditional clustering techniques, such as distance-based methods generally cannot handle this type clustering. The reason is that instead of using URLs as features, the transactions must be used as features, whose number is in tens to hundreds of thousands in a typical application. Furthermore, dimensionality reduction in this context may not be appropriate, as removing a significant number of transactions as features will result in losing too much information. *Association Rule Hypergraph Partitioning* (ARHP) technique [HKKM97, HKKM98] is well-suited for this task, since it provides automatic filtering capabilities, does not require distance computations, and can be used in high-dimensional data sets without requiring dimensionality reduction. The ARHP technique has been used successfully in a variety of domains, such as content-based categorization of Web documents [HBG+99].

Association rules capture the relationships among items based on their patterns of co-occurrence across transactions. In Web transactions, association rules capture relationships among URL references based on the navigational patterns of users. The association rule discovery methods, such as the Apriori algorithm [AS94], initially find groups of items (which in this case are the URLs appearing in the preprocessed log) occurring frequently together in many transactions. Such groups of items are referred to as *frequent item sets*. Given a set $I = \{I_1, I_2, \ldots, I_k\}$ of frequent itemsets, the *support* of I_i is the fraction of transactions containing I_i, and is denoted by $\sigma(I_i)$. Generally, a support threshold is specified before mining and is used by the algorithm for pruning the search space. An association rule r is an expression of the form $\langle X \Rightarrow Y, \sigma_r, \alpha_r \rangle$, where X and Y are sets of items, σ_r is the support of $X \cup Y$, and α_r is the confidence for the rule r given by $\sigma(X \cup Y) / \sigma(X)$.

The frequent itemsets are used as hyperedges to form a hypergraph $H = \langle V, E \rangle$, where $V \subseteq U$ and $E \subseteq I$. A hypergraph is an extension of a graph where each hyperedge can connect more than two vertices. The weights for hyperedges are computed based on the confidence of the association rules involving the items in the frequent itemset. The hypergraph H is then partitioned into a set of clusters $C = \{c_1, c_2, \ldots, c_h\}$. The similarity among items is captured implicitly by the association rules. Each cluster is examined to filter out vertices that are not highly connected to the rest of the vertices of the partition. The connectivity of vertex v (a URL appearing in the frequent itemset) with respect to a cluster c is defined as follows:

$$conn(v,c) = \frac{|\{e \mid e \subseteq c, v \in e\}|}{|\{e \mid e \subseteq c\}|}$$

Connectivity measures the percentage of edges with which a vertex is associated. A high connectivity value suggests that the vertex has many edges, connecting a good proportion of the vertices in the partition. The vertices with connectivity measure greater than a given threshold value are considered to belong to the cluster. Additional filtering of non-relevant items can be achieved using the support criteria in the association rule discovery components of the algorithm.

Once the URL clusters have been computed, a partial session for the current user can be assigned to a matching cluster. The connectivity value of an item (URL) defined above is important because it is used as the weight associated with that item for the cluster. These weights are used as part of the recommendation process when clusters are matched against an active user session. The details of the matching and recommendation process are discussed in the next section.

3 Automatic Customization Based on Usage Clusters

The online component of the Web personalization system involves the computation of a *recommendation set* for the current session, consisting of links to pages that the 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 browser.

We use a fixed-size sliding window over the active session to capture the current user's history depth. A sliding window of size n over the active session allows only the last n visited pages to influence the computation of the recommendation set. This makes sense in the context of personalization since most users go back and forth while navigating a site to find independent pieces of information. In many cases these *sub-sessions* have a length of no more than 2 or 3 references. The notion of a sliding session window is similar to the notion of *N-grammars* discussed in [Cha96]. We can determine the optimum window size based on the average user transaction length identified during the preprocessing stage.

We also want to weight 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. Now, 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 factor* of u with respect to s 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 URL cluster can be viewed as virtual user profile representing common access patterns. Once a new user starts a session, our goal is to match, at each step, the partial user session with the usage clusters, and provide dynamic recommendations to the user. Recall that the weight of URL u in a cluster c is the connectivity value of the item within the cluster, denoted by $conn(u, c)$, as defined in the previous section. We can therefore represent each cluster $c \in C$, as a vector $\vec{c} = \left\langle u_1^c, u_2^c, \ldots, u_n^c \right\rangle$, where

$$u_i^c = \begin{cases} conn(url_i, c), & \text{if } url_i \in c \\ 0, & \text{otherwise} \end{cases}$$

The current active session s is also represented as a (binary) vector $<s_1, s_2, \ldots, s_n>$, where $s_i = 1$, if the user has accessed URL_i in this session, and $s_i = 0$, otherwise. The cluster matching score is now computed as follows:

$$match(s,c) = \frac{\sum_k u_k^c \cdot s_k}{|s| \cdot \sqrt{\sum_k (u_k^c)^2}}$$

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 cluster c and an active session s, a recommendation score, $Rec(s,u)$, is computed for each URL u in c *as follows*:

$$Rec(s,u) = \sqrt{conn(u,c) \cdot match(s,c)} \cdot ldf\,(s,u).$$

If the URL *u* is in the current active session, then its recommendation value is zero because of the link distance factor. Finally, we can compute the recommendation set, *Recommend*(*s*), for current active session *s* by collecting from each cluster all URLs whose recommendation score satisfies a minimum recommendation threshold ρ, i.e., $Recommend(s) = \{u_i^c \mid c \in C, \text{ and } Rec(s,u_i^c) \geq \rho\}$. For each URL that is contributed by several clusters, we use its maximal recommendation score from all of the contributing clusters.

4 Experimental Results

We used the access logs from the University of Minnesota Computer Science Web server to evaluate our personalization technique. The preprocessed log (for February of 1999) was converted into a session file comprising 14294 user transactions and a total of 4001 unique URLs (before support filtering). We used the hypergraph partitioning algorithm as modified in [CC99] in order to take frequent itemsets as the input, and performed the clustering of URLs. The frequent itemsets were found using the tree projection algorithm described in [AAP99]. Each URL serves as a vertex in the hypergraph, and each edge represents a frequent itemset with the weight of the edge taken as the interest for the set. Since interest increases dramatically with the number of items in a rule, the log of the interest is taken in order to prevent the larger rules from completely dominating the clustering process. For the recommendation process we chose a session window size of 2, since the average session size was 2.4. The recommendation results are given for the sample path `/research=>/grad-info=>/registration-info`.

A summary of the results are given in Figure 2; additional experimental results and a demonstration site using the techniques discussed in this paper can be found at http://aztec.cs.depaul.edu/scripts/acr2. Each table in Figure 2 corresponds to one step in the user navigation through the path. In each case the current active session window is given along with the top recommendations. A cut-off value of 0.30 was used for the recommendation score. Each row shows two consecutive recommendations.

Note that in many cases the obvious recommendations associated with the URLs in the active session window rank lower than URLs for pages that are farther away in the site graph. This variation is mainly due to the link distance factor discussed earlier. In each case the recommendation set is composed of URLs from a number of matching clusters. When `/research` page is requested, the URLs for a number of popular research groups are included. But, note that pages from graduate information and registrations sections are also included, since many users (mainly graduate students) often have overlapping interests across these sections. When `/grad-info` is requested, some of the frequently visited URLs associated with that page as well as related class registration pages rank higher. Finally, when `/registration-info` is requested, a more focused set of recommendations results, not involving research related pages.

5 Conclusions

The ability to collect detailed usage data at the level of individual mouse clicks, provides Web-based companies with a tremendous opportunity for personalizing the Web experience of clients. In e-commerce parlance this is being termed *mass customization*. Most current approaches to personalization by various Web-based companies rely heavily on human participation to collect profile information about users. This suffers from the problems of the profile data being subjective, as well getting out of date as the user preferences change over time. In this paper we have presented an architecture for automatic Web personalization based on Web usage data. We introduced an effective clustering technique using association rule mining to learn overlapping user profiles, and discussed how the extracted knowledge can be used in real-time to provide navigational pointers for users. Our experimental results indicate that the techniques discussed here are promising, and bear further investigation and development. Our future work in this area involves conducting experiments with various types of transactions derived from user sessions, for example, to isolate specific types of "content" pages in the recommendation process. We also plan on incorporating client-side agents to provide an additional level of personalization based on user preferences.

Session Window: /research			
Recommendation	**Score**	**Recommendation**	**Score**
/newsletter/newfaculty.html	0.73	/newsletter	0.65
/faculty	0.55	/research/cnmrg	0.55
/research/softeng	0.55	/research/airvl	0.51
/research/mmdbms	0.48	/research/agassiz	0.47
/personal-pages	0.37	/registration-info	0.35
/registration-info/spring99.html	0.32	/grad-info	0.30
/registration-info/sch99-00.html	0.30	/grad-research	0.30

Session Window: /research => /grad-info			
Recommendation	**Score**	**Recommendation**	**Score**
/faculty	0.59	/personal-pages	0.52
/newsletter/newfac.html	0.52	/newsletter	0.46
/grad-info/grad-handbook.html	0.45	/grad-info/course-guide.html	0.45
/grad-info/prospective-grads.html	0.40	/registration-info	0.39
/research/cnmrg	0.39	/research/softeng	0.39
/research/airvl	0.36	/registration-info/spring99.html	0.35
/research/mmdbms	0.34	/research/agassiz	0.33

Session Window: /grad-info => /registration-info			
Recommendation	**Score**	**Recommendation**	**Score**
/faculty	0.51	/personal-pages	0.45
/grad-info/grad-handbook.html	0.45	/grad-info/course-guide.html	0.45
/grad-info/prospective-grads.html	0.40	/registration-info/spring99.html	0.36
/registration-info/sch99-00.html	0.34		

Figure 2. Recommendation Results for a Typical User Navigation Path

References

[AAP99] Agarwal, R., Aggarwal, C., and Prasad, V., A tree projection algorithm for generation of frequent itemsets. In *Proceedings of High Performance Data Mining Workshop*, Puerto Rico, 1999.

[AS94] Agrawal, R. and Srikant, R., Fast algorithms for mining association rules. *In Proceedings of the 20th VLDB conference*, pp. 487-499, Santiago, Chile, 1994.

[BM99] Buchner, A. and Mulvenna, M. D., Discovering internet marketing intelligence through online analytical Web usage mining. *SIGMOD Record*, (4) 27, 1999.

[Cha96] Charniak, E., *Statistical language learning*. MIT Press, 1996.

[CC99] Clifton, C. and Cooley, R., TopCat: data mining for topic identification in a text corpus. In *Proceedings of the 3rd European Conference of Principles and Practice of Knowledge Discovery in Databases*, Prague, Czech Republic, 1999.

[CMS99] Cooley, R., Mobasher, B., and Srivastava, J., Data preparation for mining World Wide Web browsing patterns. *Journal of Knowledge and Information Systems*, (1) 1, 1999.

[CPY96] Chen, M. S., Park, J. S., and Yu, P. S., Data mining for path traversal patterns in a Web environment. *In Proceedings of 16th International Conference on Distributed Computing Systems*, 1996.

[HBG+99] Han, E-H, Boley, D., Gini, M., Gross, R., Hastings, K., Karypis, G., Kumar, V., and Mobasher, B., More, J., Document categorization and query generation on the World Wide Web using WebACE. *Journal of Artificial Intelligence Review*, January 1999.

[HKBR99] Herlocker, J., Konstan, J., Borchers, A., Riedl, J.. An algorithmic framework for performing collaborative filtering. To appear in *Proceedings of the 1999 Conference on Research and Development in Information Retrieval*, August 1999.

[HKKM97] Han, E-H, Karypis, G., Kumar, V., and Mobasher, B., Clustering based on association rule hypergraphs. In *Proccedings of SIGMOD'97 Workshop on Research Issues in Data Mining and Knowledge Discovery (DMKD'97),* May 1997.

[HKKM98] Han, E-H, Karypis, G., Kumar, V., and Mobasher, B., Hypergraph based clustering in high-dimensional data sets: a summary of results. *IEEE Bulletin of the Technical Committee on Data Engineering*, (21) 1, March 1998.

[NFJK99] Nasraoui, O., Frigui, H., Joshi, A., Krishnapuram, R., 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.

[PE98] Perkowitz, M. and Etzioni, O., Adaptive Web sites: automaticlly synthesizing Web pages. In *Proceedings of Fifteenth National Conference on Artificial Intelligence*, Madison, WI, 1998.

[SF99] Spiliopoulou, M. and Faulstich, L. C., WUM: A Web Utilization Miner. *In Proceedings of EDBT Workshop WebDB98*, Valencia, Spain, LNCS 1590, Springer Verlag, 1999.

[SKS98] Schechter, S., Krishnan, M., and Smith, M. D., Using path profiles to predict HTTP requests. In *Proceedings of 7th International World Wide Web Conference*, Brisbane, Australia, 1998.

[SM95] Shardanand, U., Maes, P., Social information filtering: algorithms for automating "word of mouth." In *Proceedings of the ACM CHI Conference*, 1995.

[SZAS97] Shahabi, C., Zarkesh, A. M., Adibi, J., and Shah, V., Knowledge discovery from users Web-page navigation. In *Proceedings of Workshop on Research Issues in Data Engineering*, Birmingham, England, 1997.

[YJGD96] Yan, T., Jacobsen, M., Garcia-Molina, H., Dayal, U., From user access patterns to dynamic hypertext linking. In *Proceedings of the 5th International WWW Conference*, Paris, 1996.