Discovery and Evaluation of Aggregate Usage Profiles for Web Personalization

Bamshad Mobasher*, Honghua Dai, Tao Luo and Miki Nakagawa
School of Computer Science, Telecommunication, and Information Systems, DePaul University, Chicago, Illinois, USA

Abstract. Web usage mining, possibly used in conjunction with standard approaches to personalization such as collaborative filtering, can help address some of the shortcomings of these techniques, including reliance on subjective user ratings, lack of scalability, and poor performance in the face of high-dimensional and sparse data. 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 usage profiles" from these patterns. In this paper we present and experimentally evaluate two techniques, based on clustering of user transactions and clustering of pageviews, in order to discover overlapping aggregate profiles that can be effectively used by recommender systems for real-time Web personalization. We evaluate these techniques both in terms of the quality of the individual profiles generated, as well as in the context of providing recommendations as an integrated part of a personalization engine. In particular, our results indicate that using the generated aggregate profiles, we can achieve effective personalization at early stages of users' visits to a site, based only on anonymous clickstream data and without the benefit of explicit input by these users or deeper knowledge about them.

1. Introduction

Today many of the successful e-commerce systems that provide server-directed automatic Web personalization are based on collaborative filtering. Collaborative filtering technology [11, 15, 28], generally involves 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. There are, however, some well-known limitations to this type of approach. For instance, as noted in recent studies [21, 26], 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

* Please direct correspondence to *mobasher@cs.depaul.edu*.

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 [17, 19, 20, 31] or as a mechanism for improving and optimizing the structure of a site [9, 22, 24]. Data mining techniques, such as clustering, have also been shown to improve the scalability and performance of collaborative filtering techniques [21]. In general, Web usage mining systems [5, 8, 23, 32] 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 [25].

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 or patterns through clustering, as well as other Web mining techniques, have been explored by several research groups [4, 20, 22, 27, 29, 30]. 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. In the case of [22], aggregate usage profiles were discovered using an algorithm called PageGather which uses as its basis clustering of pages based the Clique (complete link) clustering technique. While the generated profiles were not integrated as part of a recommender system, they were used to automatically synthesize alternative static index pages for a site.

In this paper we present and experimentally evaluate two Web usage mining techniques, each with its own characteristics, for the discovery of aggregate usage profiles that can be effective in Web personalization. The first technique, called PACT (Profile Aggregations based on Clustering Transactions), is based on the derivation of overlapping profiles from user transactions clusters. A preliminary version of this technique was first introduced in the context of a generalized framework for usage-based Web personalization in [19]. The second technique, originally introduced in [17], uses Association Rule Hypergraph Partitioning [12, 13] to directly derive overlapping aggregate profiles from pageviews (rather than from user transactions). Each of these techniques generates

overlapping profiles which capture aggregate views of the behavior of subsets of site users based their interests and/or information needs.

Our primary focus in this paper is the experimental evaluation of the profile discovery techniques discussed above based on real usage data. To this end, we compare and evaluate both the quality of generated profiles, as well as the effectiveness of the techniques when used as part of a recommender system for Web personalization. We also compare our techniques with the Clique-based clustering technique used in [22], described above. Finally, based on the experimental results we draw some conclusions as to the circumstances under which each technique is most appropriately used. In particular, our evaluation suggests that, when applied in the context of anonymous clickstream data, these techniques show promise in creating effective personalization solutions that can potentially help retain and convert unidentified visitors based on their activities in the early stages of their visits when deeper knowledge about these visitors is not yet available.

2. Mining Web Usage Data for Personalization

2.1. General Framework for Usage-Based Personalization

A general framework for personalization based on aggregate usage profiles is depicted in Figure 1 [18]. This framework distinguishes between the offline tasks of data preparation and usage mining and the online personalization components. The data preparation tasks result in aggregate structures such as a user transaction file capturing meaningful semantic units of user activity to be used in the mining stage. Given the preprocessed data, a variety of data mining tasks can be performed. For example, the usage mining tasks can involve the discovery of association rules [1, 2], sequential patterns [3], pageview clusters, transaction clusters, or any other pattern discovery method from user transactions. Our attention in this paper is focused on specific clustering techniques for the discovery of aggregate usage profiles, and hence, only the relevant components are shown in Figure 1. In the online component of the system, the Web server keeps track of the active server session as the user's browser makes HTTP requests. The recommendation engine considers the active server session in conjunction with the discovered patterns to provide personalized content. The personalized content can take the form of recommended links and products, or targeted advertisements and textual content.

In the data preparation stage, we use the heuristics proposed in [8] to identify unique user sessions form anonymous usage data and

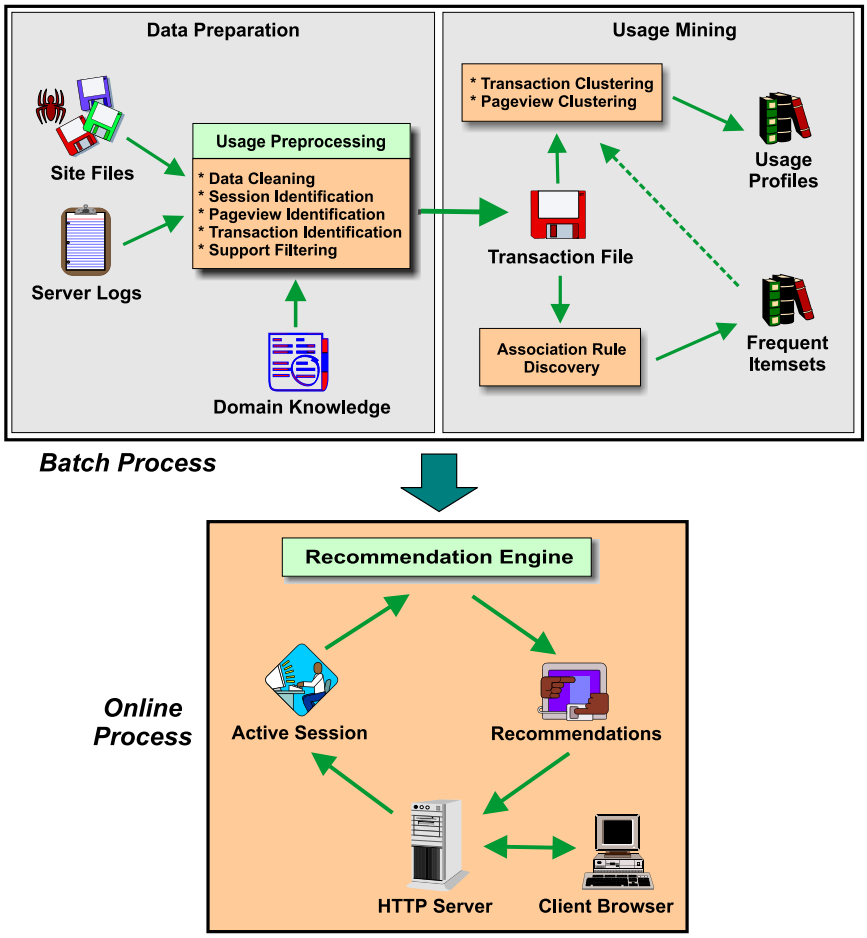

Figure 1. A General Framework for Personalization Based on Web Usage Mining

to infer cached references (path completion). In this stage, the data cleaning tasks involve the removal of erroneous or redundant references, as well as the detection and removal of robot navigation references. *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. Only relevant pageviews are included in the transaction file. Furthermore, among the relevant pageviews some may be more significant than others. For example, 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." A further level of

granularity is obtained by identifying transactions within the sessions [8]. 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.

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 result in a set of n pageviews, $P = \{p_1, p_2, \cdots, p_n\}$, appearing in the transaction file with each pageview uniquely represented by its associated URL; and a set of m user transactions, $T = \{t_1, t_2, \cdots, t_m\}$, where each $t_i \in T$ is a subset of P. To facilitate various data mining operations such as clustering, we view each transaction t as an n-dimensional vector over the space of all pageview references, i.e.,

$$t = \langle w(p_1, t), w(p_2, t), \ldots, w(p_n, t) \rangle ,$$

where $w(p_i, t)$ is a weight, in the transaction t, associated with the pageview represented by $p_i \in P$. 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 document 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 (for example navigational pages may be weighted less heavily than content or product-oriented pageviews).

The transaction file obtained in the data preparation stage is used as the input to the profile generation methods. Ideally, profiles capture an aggregate view of the behavior of subsets of users based their interests or information needs. In particular, to be effective for personalization, aggregate profiles must exhibit three important characteristics:

1. they should capture possibly overlapping interests of users, since many users may have common interests up to a point (in their navigational history) beyond which their interests diverge;

2. they should provide the capability to distinguish among pageviews in terms of their significance within the profile; and

3. they should have a uniform representation which allows the recommendation engine to easily integrate different kinds of profiles, independently of the profile generation method used.

Given these requirements, we have found that representing usage profiles as weighted collections of pageview objects provides a great deal of flexibility. Each item in a usage profile is a URL representing a relevant pageview object, and can have an associated weight representing its significance within the profile. The profile can be viewed as an ordered collection (if the goal is to capture the navigational path profiles followed by users [23]), or as unordered (if the focus is on capturing associations among specified content or product pages). 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 following sections, we present two techniques for discovering overlapping usage profiles based on clustering of transactions and clustering of pageviews, respectively. We also discuss how these profiles are used by the recommendation engine to provide real-time personalization.

2.2. Discovery of Aggregate Profiles Based on Transaction Clustering

Given the mapping of user transactions into a multi-dimensional space as vectors of pageviews, 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. Such a clustering will result in a set $TC = \{c_1, c_2, \cdots, c_k\}$ of transaction clusters, 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. 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 weighted collections of pageviews which, as noted earlier, represent aggregate profiles.

Preliminary results [19] have identified one potentially effective method for the derivation of profiles from transaction clusters. To obtain aggregate profiles from transaction clusters, we employ a technique analogous to concept indexing methods used to extract document

cluster summaries in information retrieval and filtering [14]. We call this method PACT (Profile Aggregations based on Clustering Transactions). In the simplest case, PACT generates aggregate profiles based on the centroids of each transaction cluster. In general, however, PACT can consider a number of other factors in determining the item weights within each profile. These additional factors may include the link distance of pageviews to the current user location within the site or the rank of the profile in terms of its significance. The primary difference between PACT and the concept indexing method proposed in [14] is that we start with clusters of transactions (rather than document clusters), and that the weights associated with items (in this case pageviews) are obtained differently.

In this paper we restrict the item weights to be the mean feature values of the transaction cluster centroids. For each transaction 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 in c to the total number of transactions in the cluster. The weight of each pageview within a profile is a function of this quantity thus obtained. In generating the usage profiles, 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 usage profile pr_c as a set of pageview-weight pairs:

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

where the significance weight, $weight(p, pr_c)$, of the pageview p within the usage profile pr_c is given by

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

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

2.3. Discovery of Aggregate Profiles Based on Pageview Clustering

The second profile generation method we consider is to directly compute clusters of pageview references based on how often they occur together across user transactions (rather than clustering transactions, themselves). In general, this technique will result in a different type of usage profiles compared to the transaction clustering technique. The profiles obtained by reducing transaction clusters group together pages that co-occur commonly across "similar" transactions. On the other hand, pageview clusters tend to group together frequently co-occurring items 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 pageviews 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 may result in losing too much information.

We have found that the Association Rule Hypergraph Partitioning (ARHP) technique [12, 13] is well-suited for this task since it can efficiently cluster high-dimensional data sets without requiring dimensionality reduction as a preprocessing step. Furthermore, the ARHP provides automatic filtering capabilities, and does not require distance computations. The ARHP has been used successfully in a variety of domains, including the categorization of Web documents [10].

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 pageviews based on their co-occurrence in navigational patterns of users. The association rule discovery methods such as the Apriori algorithm [2], initially find groups of items (which in this case are the URLs appearing in the transaction file) occurring frequently together in many transactions. Such groups of items are referred to as *frequent itemsets*.

Given a set $IS = \{I_1, I_2, \cdots, I_k\}$ of frequent itemsets, the support of I_i is defined as

$$\sigma(I_i) = \frac{|\{t \in T : I_i \subseteq t\}|}{|T|}$$

Generally, a support threshold is specified before mining and is used by the algorithm for pruning the search space. The itemsets returned by the algorithm satisfy this minimum support threshold.

In the ARHP, the set IS of large frequent itemsets are used as hyperedges to form a hypergraph $H = \langle V, E \rangle$, where $V \subseteq P$ and $E \subseteq IS$. A hypergraph is an extension of a graph in the sense that each hyperedge can connect more than two vertices. The weights associated with each hyperedge can be computed based on a variety of criteria such as the confidence of the association rules involving the items in the frequent itemset, the support of the itemset, or the "interest" of the itemset. In our experiments, we weight each hyperedge using a function of the interest of the itemset which is defined as:

$$Interest(I) = \frac{\sigma(I)}{\prod_{i \in I} \sigma(i)}.$$

Essentially, the interest measure [6] considers the support of a frequent itemset relative to the probability that all of the items appearing together in a set if the items are randomly distributed, assuming conditional independence. An interest measure that is greater than one indicates that the items in the itemset appear together more often than what would be expected through a random distribution.

The hypergraph H is then partitioned into a set of clusters C. Each partition 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 pageview appearing in the frequent itemset) with respect to a cluster c is defined as:

$$conn(v, c) = \frac{\sum_{e \subseteq c, v \in e} weight(e)}{\sum_{e \subseteq c} weight(e)}.$$

A high connectivity value suggests that the vertex has strong edges connecting it to other vertices in the partition. The vertices with connectivity measure greater than a given threshold value are considered to belong to the partition, and the remaining vertices are dropped from the partition.

The hypergraph is recursively partitioned until a stopping criterion for each partition is reached. The stopping criterion is determined according to a threshold on the ratio of the weights of the cut edges to the weights of uncut edges in the partition. Once the partitioning is completed, vertices can be "added back in" to clusters depending on the user defined overlap parameter. For each partial edge that is left in a cluster, if the percentage of vertices from the original edge that are still in the cluster exceed the overlap percentage, the removed vertices are added back in. This will allow some vertices to belong to more than

one cluster. In the ARHP method, additional filtering of non-relevant items can be achieved using the support criteria in the association rule discovery components of the algorithm.

The connectivity value of an item (pageviews) defined above is important also because it is used as the primary factor in determining the weight associated with that item within the profile. As noted earlier, the weights associated with pageviews in each profile are used as part of the recommendation process when profiles are matched against an active user session (see below).

2.4. A Recommendation Engine Using Aggregate Profiles

The goal of personalization based on anonymous Web usage data is to compute a *recommendation set* for the current (active) user session, consisting of the objects (links, ads, text, products, etc.) that most closely match the current user profile. The recommendation engine is the online component of a usage-based personalization system. If the data collection procedures in the system include the capability to track users across visits, then the recommendation set can represent a longer term view of potentially useful links based on the user's activity history within the site. If, on the other hand, profiles are derived from anonymous user sessions contained in log files, then the recommendations provide a "short-term" view of user's navigational history. These recommended objects are added to the last page in the active session accessed by the user before that page is sent to the browser.

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 [7]. 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.

Usage profiles, obtained using any of the techniques described in the previous section, are represented as sets of pageview-weight pairs. This will allow for both the active session and the profiles to be treated as n-dimensional vectors over the space of pageviews in the site. Thus, given a usage profile C, we can represent C as a vector:

$$C = \left\langle w_1^C, w_2^C, \ldots, w_n^C \right\rangle$$

where

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

Similarly, the current active session S is also represented as a vector $S = \langle s_1, s_2, \cdots, s_n \rangle$, where s_i is a significance weight associated with the corresponding pageview reference, if the user has accessed p_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.

In computing the matching score between aggregate profiles and the current active session, a variety of similarity measures can be used. In traditional collaborative filtering domains (where feature weights are item ratings on a discrete scale), the *Pearson r* correlation coefficient is commonly used. This measure is based on the deviations of users' ratings on various items from their mean ratings on all rated items. However, this measure is not appropriate in the context of anonymous personalization based on clickstream data (particularly in the case of binary weights). Instead other distance or similarity measures which are applicable in the context of a vector space model can be used. In our case, we use the *cosine coefficient*, commonly used in information retrieval, which measures the cosine of the angle between two vectors. The cosine coefficient can be computed by normalizing the dot product of two vectors with respect to their vector norms.:

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

Note that the matching score is normalized for the size of the profiles 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.

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 pageview p_i based on the matching aggregate profiles. Two factors are used in determining this recommendation score: the overall similarity of the active session to the aggregate profile as

a whole, and the average weight of each item in the profile. Given a profile C and an active session S, a recommendation score, $Rec(S, p)$, is computed for each pageview $p \in C$ as follows:

$$Rec(S, p) = \sqrt{weight(p, C) \cdot match(S, C)}.$$

The square root in the above function is to compensate for the impact of counting the weight of pageview $p \in C$ twice (both directly and in computing the profile matching score), and results in a normalized value between 0 and 1. If the pageview p 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 usage profile all pageviews whose recommendation score satisfies a minimum recommendation threshold ρ, i.e.,

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

where UP is the collection of all usage profiles. Furthermore, for each pageview that is contributed by several usage profiles, we use its maximal recommendation score from all of the contributing profiles. The reason for allowing recommendations to be contributed by multiple profiles is that, in most cases, users' activities tend to fit several different aggregate profiles during their visit to the site to various degrees. Considering multiple matching profiles has the effect of improving the "coverage" of the recommendation engine. In this paper, however, we are interested in the quality and effectiveness of individual profiles produced by our profile generation methods. Hence, in our experimental evaluation, discussed below, we limit the recommendation sets to those contributed only by the top matching profile.

3. Experimental Evaluation

In this section we provide a detailed experimental evaluation of the profile generation techniques presented above. Specifically, we discuss our experimental setting and the implementation details for each of the profile generation techniques. We then evaluate the effectiveness of the generated profiles in the context of Web personalization using several performance measures.

3.1. Experimental Setup

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 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 these transactions were randomly selected as the evaluation set, and the remaining portion was used as the training set for profile generation. The total number of remaining pageview URLs in the training and the evaluation sets was 62.

For the PACT method, we used multivariate k-means clustering to partition the transaction file. Overlapping aggregate profiles were generated from transaction clusters using the method described earlier. For Association Rule Hypergraph Partitioning, the frequent itemsets were found using the Apriori algorithm [2]. Each pageview 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 comparison purposes, we also generated usage profiles using the Clique-based clustering technique used in [22]. We used a similarity threshold of 0.5 to form the similarity graph among pairs of pageviews. Profiles were then generated from the completely connected components of the graph. The weight of items in each Clique profile was determined by measuring the similarity of the item vector (a vector of whose dimensions are transactions) to the cluster centroid.

In all cases, the weights of pageviews were normalized so that the maximum weight in each profile would be 1. In the case of PACT and Hypergraph, the maximum overlap among any pairs of profiles was already less than 50%, however, the Clique method tends to generate a large number of highly overlapping clusters, often differing by only 1 or 2 items. In order to rectify this situation we employed the overlap

Table I. Examples of Aggregate Usage Profiles Obtained Using the PACT Method

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 - Dec. 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

reduction method discussed in [22]. The profiles were ranked according to average similarity of items within the profiles, and then the lower ranking profiles which had more than 50% overlap with a previous profile were eliminated.

Table I depicts 3 partial profiles generated using the PACT method for the ACR site. The first profile in Table I represents the activity of users who are primarily interested in general ACR sponsored conferences. The second profile, while containing some overlap with the first, seems to capture the activity of users whose interests are more focused on specific conferences or journals related to marketing and consumer behavior. Finally, the third profile captures the activity of users interested in news items as well as specific columns that appear in the "Online Archives" section .

3.2. Evaluation of Individual Profile Effectiveness

As a first step in our evaluation, we computed the average visit percentage for the top ranking profiles generated by each method. This evaluation method, introduced by Perkowitz and Etzioni [22], allows us to evaluate each profile individually according to the likelihood that a user who visits any page in the profile will visit the rest of the pages in that profile during the same session. However, we modified the original algorithm to take the weights of items within the profiles into account.

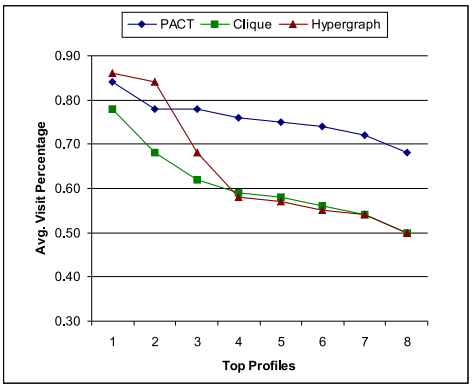

Figure 2. Comparison of top ranking usage profiles for the three profile generation methods based on their weighted average visit percentage.

Specifically, let T be the set of transactions in the evaluation set, and for a profile pr, let T_{pr} denote a subset of T whose elements contain at least one page from pr. Now, the weighted average similarity to the profile pr over all transactions is computed (taking both the transactions and the profile as vectors of pageviews) as:

$$\sum_{t \in T_{pr}} \left(\vec{t} \cdot \vec{pr} \right) / |t|.$$

The *weighted average visit percentage* (WAVP) is this average divided by the total weight of items within the profile pr:

$$\left(\sum_{t \in T_{pr}} \frac{\vec{t} \cdot \vec{pr}}{|t|} \right) \Big/ \left(\sum_{p \in pr} weight(p, pr) \right).$$

Profiles generated by each method were ranked according to their WAVP. Figure 2 depicts the comparison of top ranking profiles.

The top ranking profiles generated by the Hypergraph method perform quite well under this measure, however, beyond the top 2 or 3 profiles, both Hypergraph and the Clique methods seem to perform similarly. On the other hand the PACT method, overall, performs consistently better than the other techniques. It should be noted that, while WAVP provides a measure of the predictive power of individual profiles, it does not necessarily measure the "usefulness" of the profiles. For instance, the Hypergraph method tends to produce highly cohesive clusters in which potentially "interesting" items, such as pageviews that occur more deeply within the site graph, dominate. This is verified by our experiments on the recommendation accuracy of the method as a whole, discussed below.

3.3. Evaluation of Recommendation Effectiveness

The average visit percentage, while a good indication of the quality of individual profiles produced by the profile generation methods, is not by itself sufficient to measure the effectiveness of a recommender system based on these profiles as a whole. The recommendation accuracy may be affected by a other factors such as the size of the active session window and the recommendation threshold that filters out low scoring pages. For these reasons, it is important to evaluate the effectiveness of the aggregate usage profiles in the context of the recommendation process. In this section we present several measures for evaluating the recommendation effectiveness and discuss our experimental results based on these measures.

3.3.1. *Performance Measures*

In order to evaluate the recommendation effectiveness for each method, we measured the performance of each method using 3 different standard measures, namely, precision, coverage, and the $F1$ measure; and a new measure which we call the R measure.

Assume that we have transaction t (taken from the evaluation set) viewed as a set of pageviews, and that we use a window $w \subseteq t$ (of size $|w|$) to produce a recommendation set R using the recommendation engine. Then the *precision* of R with respect to t is defined as:

$$precision(R, t) = \frac{|R \cap (t - w)|}{|R|},$$

and the coverage of R with respect to t is defined as:

$$coverage(R, t) = \frac{|R \cap (t - w)|}{|t - w|}.$$

These measures are adaptations of the standard measures, precision and recall, often used in information retrieval. In this context, precision measures the degree to which the recommendation engine produces accurate recommendations (i.e., the proportion of relevant recommendations to the total number of recommendations). On the other hand, coverage measures the ability of the recommendation engine to produce all of the pageviews that are likely to be visited by the user (proportion of relevant recommendations to all pageviews that should be recommended). Neither of these measures individually are sufficient to evaluate the performance of the recommendation engine, however, they are both critical. This is particularly true in the context of e-commerce were recommendations are products. A low precision in this context will likely result in angry customers who are not interested in the

recommended items, while low coverage will result in the inability of the site to produce relevant cross-sell recommendations at critical points in user's interaction with the site. Both of these negative phenomena are characteristics of standard collaborative filtering techniques in the face of very sparse ratings data as the number of items that can potentially be rated by users increases.

Ideally, one would like high precision and high coverage. A single measure that captures this is the $F1$ measure [16]:

$$F1(R,t) = \frac{2 \times precision(R,t) \times coverage(R,t)}{precision(R,t) + coverage(R,t)}.$$

The $F1$ measure attains its maximum value when both precision and coverage are maximized. One might observe that, using the notation introduced above, the $F1$ measure can be reduced to an application of *Dice's coefficient* to the recommendation set R and the remaining portion of the user's session (i.e., $t - w$). Thus, $F1$ can be viewed as a measure of similarity between these two sets of pageviews.

When usage profiles and recommendation sets contain pageviews appearing in users' clickstreams, the recommendation engine tends to achieve much better coverage than when the focus is only on (a much smaller set of) products. This is because it is likely that many users visit substantial portions of the site, resulting in a much higher data density than exists in the typical collaborative filtering domains. In this context we may wish to have much smaller recommendation sets (while still maintaining the accuracy and coverage of the recommendations). To capture this notion, we introduce another hybrid measure, which we call the R *measure*. The R measure is obtained by dividing the coverage by the size of the recommendation set. This is a much more stringent measure than $F1$ and it produces higher values when a smaller recommendation set can cover the remaining portion of a (small) session.

Part of the motivation behind introducing the R measure is that it is better able to capture changes in the performance of the algorithms with varying window sizes. As detailed below, we evaluate the recommendations using a fixed set of user transactions as our evaluation set. If the window size used in producing the recommendations is increased, a smaller portion of the evaluation transactions are available to match against the recommendation set (thus the number of matches also decrease accordingly). This will negatively impact the precision scores, even though, generally, the recommendations are of better quality when larger portions of user's clickstream are taken into account. Our experiments show that the R measure helps capture the improvements in the quality of the recommendations when the window size is increased.

Table II. Coverage and Average Number of Recommendations Using Window Size 2

Rec. Threshold	Clique		PACT		Hypergraph	
	Coverage	Avg. No. of Recs.	Coverage	Avg. No. of Recs.	Coverage	Avg. No. of Recs.
0.3	0.35	5.29	0.37	5.55	0.55	7.3
0.4	0.35	5.17	0.33	4.61	0.55	7.3
0.5	0.35	4.92	0.31	3.84	0.53	7.07
0.6	0.34	3.65	0.28	2.94	0.52	6.97
0.7	0.33	3.33	0.25	2.15	0.51	6.91
0.8	0.33	3.08	0.21	1.82	0.48	5.58
0.9	0.31	2.56	0.18	1.41	0.44	4.52

3.3.2. *Evaluation Methodology*

The basic methodology used is as follows. For a given transaction t in the evaluation set, and an active session window size n, we randomly chose $|t| - n + 1$ groups of items from the transaction as the surrogate active session windows (this is the set denoted by w in the above discussion), each having size n. For each of these active sessions, we produced a recommendation set based on aggregate profiles and compared the set to the remaining items in the transaction (i.e., $t-w$) in order to compute the precision, coverage, $F1$, and R scores. For each of these measures, the final score for the transaction t was the mean score over all of the $|t| - n + 1$ surrogate active sessions associated with t. Finally, the mean over all transactions in the evaluation set was computed as the overall evaluation score for each measure. To determine a recommendation set based on an active session, we varied the recommendation threshold from 0.1 to 1.0. A page is included in the recommendation set only if it has a recommendation score greater than or equal to this threshold.

Clearly, fewer recommendations are produced at higher thresholds, while higher coverage 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 II shows a portion of the results produced by the recommendation engine for the 3 profile generation methods using a session window size of 2. For example, at a threshold of 0.6, the Hypergraph method produced coverage of 0.52 with an average recommendation set size of 6.97 over all trials. Roughly speaking, this means that on average 52% of unique pages actually visited by users in the (remaining portion of the) evaluation set transactions matched the top 7 recommendations produced by the system.

The evaluation results for an active session window size of 2 are depicted in Figure 3. In terms of precision, the PACT method clearly outperforms the other two methods, especially for higher threshold val-

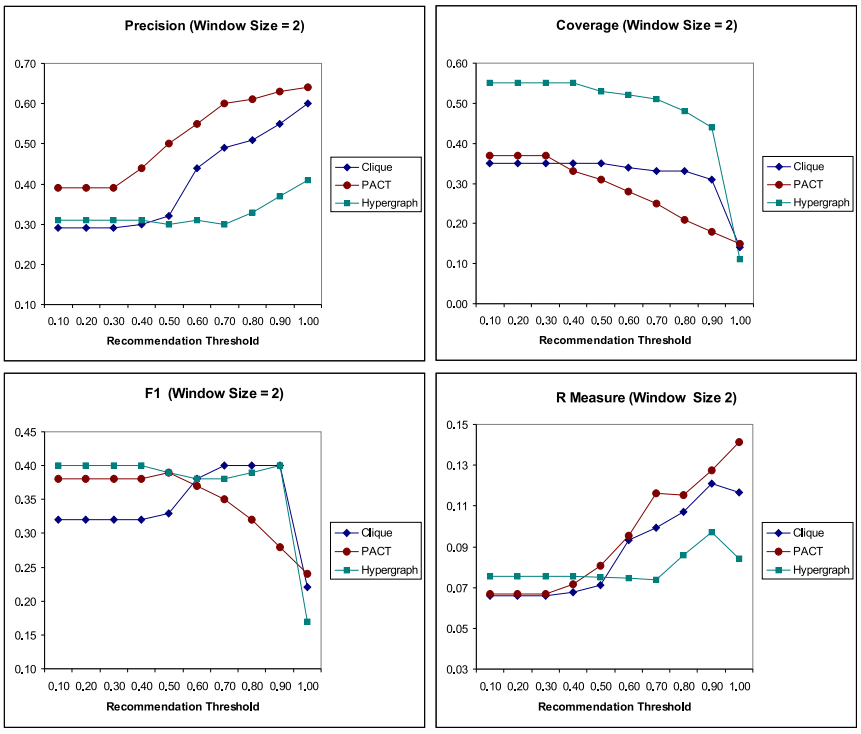

Figure 3. Comparison of recommendation effectiveness for an active session window of size 2 based on four performance measures.

ues. While the Hypergraph method showed a rather poor performance in terms of precision, it attained a much higher overall recommendation coverage leading to relatively good $F1$ scores. The R measure also verifies the PACT method as the clear winner in terms of recommendation accuracy at high recommendation thresholds where a very small number of recommendations are produced.

It should be emphasized that the scores achieved based on these measures are only based on simple anonymous clickstream data with very few pageviews (in this case 2) used to produce recommendations. In the case of PACT, these results show that it may be an effective technique for personalization based solely on the users' anonymous clickstreams, particularly at the early stages of these users' interaction with the site and before identifying or deeper information about these users are available (e.g., before registration).

Figure 4 shows the impact of an increase in the window size in terms of the two hybrid measures, i.e., the $F1$ and the R measures. All three techniques achieved an overall performance gain when the window size was increased from 2 to 3. However, the improved performance due to

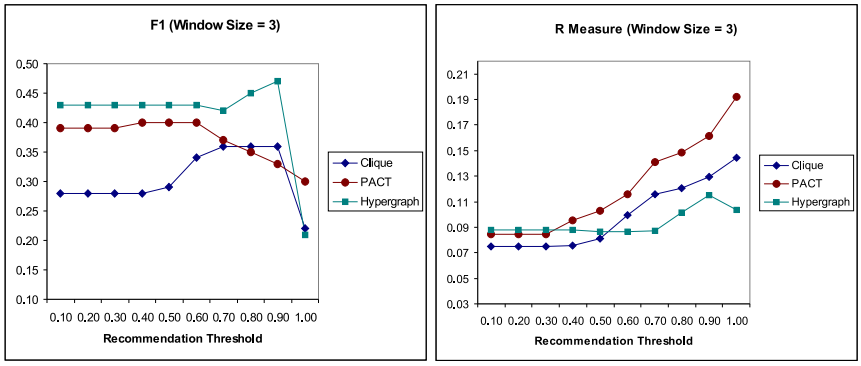

Figure 4. The impact of increase in active session window size (from 2 to 3) on recommendation effectiveness based on the $F1$ and R performance measures.

larger window size was more dramatic for PACT than the other two methods, especially as indicated by the R measure. The Hypergraph method still had the best $F1$ score since it produced dramatically higher recommendation coverage as compared to the other two methods.

Despite the fact that the Hypergraph method scored lower than PACT or Clique in terms of recommendation accuracy, casual observation of the recommendation results showed that the Hypergraph method tends to produce more "interesting" recommendations. In particular, this method often gives recommended pages that occur more deeply in the site graph as compared to top level navigational pages. This is in part due to the fact that interest of the itemsets was used to compute the weights for the hyperedges. Intuitively, we may consider a recommended object (e.g., a page or a product) more interesting or useful if a larger amount of user navigational activity is required to reach the object without the recommendation engine. In our experimental data set, these objects correspond to "content pages" that are located deeper in the site graph as opposed to top level navigational pages (these were primarily pages for specific conference calls or archived columns and articles).

In order to evaluate the effectiveness of the 3 profile generation methods in this context, we filtered out the top-level navigational pages in both the training and the evaluation sets and regenerated the aggregate profiles from the filtered data set. All other parameters for profile generation and the recommendation engine were kept constant. Figure 5 depicts the relative performance of the 3 methods on the filtered evaluation set based on an active session window of size 2. We only show the results for precision and $F1$; the improvements for the other measures are also consistent with these results.

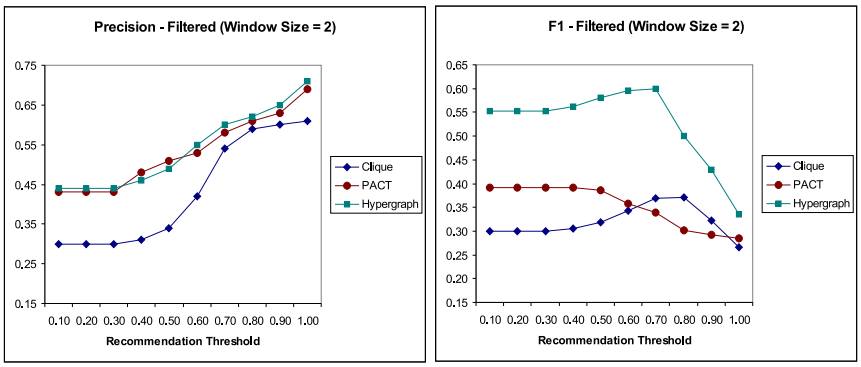

Figure 5. The impact of filtering on recommendation effectiveness based on precision and $F1$ performance measures. The results are shown only for an active session window of size 2.

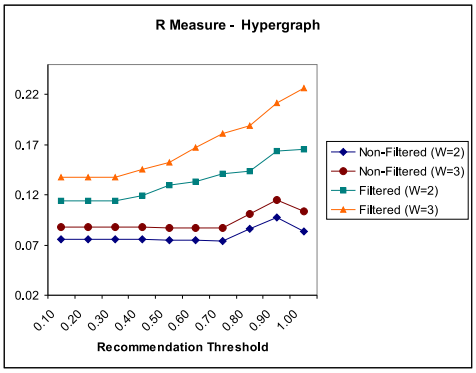

Figure 6. The performance improvements achieved by the Hypergraph method due to filtering and increased window sizes.

As these results indicate, filtering the data set resulted in better performance for all 3 methods. There was moderate improvement for Clique, while the improvement was much more dramatic for Hypergraph and (to a lesser degree) PACT. In particular, the Hypergraph method performed consistently better that the other two methods in these experiments, supporting our conjecture that it tends to produce more interesting recommendations. Particularly noteworthy is Hypergraph's improvement in terms of precision, now even surpassing PACT. To see the impact of filtering more clearly for the Hypergraph method, Figure 6 depicts its relative improvement, in terms of the R measure, when comparing the results for filtered and unfiltered data sets with window sizes of 2 and 3.

3.3.3. *Discussion*

We conclude this section by summarizing some of our observations based on the above experimental results. It should be noted that we have performed a similar set of experiments using the data from another site (a departmental Web server at a university) resulting in similar and consistent conclusions. Experiments indicate that, while specific values of performance measures differs across various data sets, the relative performance of different algorithms remains consistent with the results presented in this paper.

We used the Clique method, as used by Perkowitz and Etzioni [22] in their PageGather algorithm, for comparative purposes. In general, this technique for profile generation is not as useful as our two proposed methods, partly due to the prohibitive cost of computing a distance or similarity matrix for all pairs of pageviews and the discovery of maximal cliques in the associated similarity graph. The computation involved in this case quickly becomes unmanageable when dealing with a large, high traffic site with many unique pageviews. Furthermore, the overall performance of PACT and Hypergraph methods (in the filtered data set) is better both when considering individual profiles as well as in their use as part of the recommender system.

In comparing PACT and Hypergraph, it is clear that PACT emerges as the overall winner in terms of recommendation accuracy on the unrestricted data. However, as noted above, Hypergraph does dramatically better when we focus on more "interesting" objects (e.g., content pages that are situated more deeply within the site).

In general, the Hypergraph method seems to produce a smaller set of high quality, and more specialized, recommendations, even when a small portion of the user's clickstream is used by the recommendation engine. On the other hand, PACT provides a clear performance advantage when dealing with all the relevant pageviews in the site, particularly as the session window size is increased.

Whether PACT or Hypergraph methods should be used in a given site depends, in part, on the goals of personalization. Based on the above observations, we conclude that, if the goal is to provide a smaller number of highly focused recommendations, then Hypergraph may be a more appropriate method. This is particularly the case if only specific portions of the site (such as product-related or content pages) are to be personalized. On the other hand, if the goal is to provide a more generalized personalization solution integrating both content and navigational pages throughout the whole site, then using PACT as the underlying aggregate profile generation method seems to provide clear advantages.

The results suggest that in the contexts discussed above, PACT and Hypergraph methods may be used effectively for the purpose of anonymous personalization based on clickstream data at very early stages of a user's interaction with the site. This is particularly important in e-commerce since effective personalization at this level can lead to higher visitor retention and a higher conversion ratios (i.e., the conversion of casual browsers to potential customers).

4. Conclusions

The practicality of employing Web usage mining techniques for personalization is directly related to the discovery of effective aggregate profiles that can successfully capture relevant user navigational patterns. Once such profiles are identified, they can be used as part of usage-based recommender system, such as the one presented in this paper, to provide real-time personalization. The discovered profiles can also be used to enhance the accuracy and scalability of more traditional personalization technologies such as collaborative filtering. We have presented two effective techniques, based on clustering of transactions and clustering of pageviews, in which the aggregate user profiles are automatically learned from Web usage data. This has the potential of eliminating subjectivity from profile data as well as keeping it up-to-date. We have extensively evaluated these techniques both in terms of the quality of the individual profiles generated, as well as in the context of providing recommendations as an integrated part of a personalization engine.

Our evaluation results suggest that each of these techniques exhibits characteristics that make it a suitable enabling mechanism for different types of Web personalization tasks. But, in the particular context of anonymous usage data, these techniques show promise in creating effective personalization solutions that can help retain and convert unidentified visitors based on their activities in the early stages of their visits. This latter observation also indicates another advantage of usage-based Web personalization over traditional collaborative filtering techniques which must rely on deeper knowledge of users or on subjective input from users (such as book or music ratings).

References

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

2. R. Agrawal and Ramakrishnan Srikant. Fast algorithms for mining association rules. In *Proc. 20th Int. Conference on Very Large Data Bases, VLDB94*, 1994.
3. R. Agrawal and R. Srikant. Mining Sequential Patterns. In *Proceedings of the Int'l Conference on Data Engineering (ICDE)*, Taipei, Taiwan, March 1995.
4. A. Banerjee and J. Ghosh. Clickstream clustering using weighted longest common subsequences. In *Proceedings of the Web Mining Workshop at the 1st SIAM Conference on Data Mining*, Chicago, April 2001.
5. A. Buchner and M. D. Mulvenna. Discovering internet marketing intelligence through online analytical Web usage mining. *SIGMOD Record*, (4) 27, 1999.
6. S. Brin, R. Motwani, and C. Silverstein. Beyond market baskets: generalizing association rules to correlations. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, 1997.
7. E. Charniak. *Statistical language learning*. MIT Press, 1996.
8. 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.
9. R. Cooley, P-T. Tan., and J. Srivastava. WebSIFT: The Web site information filter system. In *Proceedings of the Workshop on Web Usage Analysis and User Profiling (WebKKD99)*, San Diego, August 1999.
10. E-H. Han, D. Boley, M. Gini, R. Gross, K. Hastings, G. Karypis, V. Kumar, B. Mobasher, and J. More. Document categorization and query generation on the World Wide Web using WebACE. *Journal of Artificial Intelligence Review*, Vol. 13, No. 5-6, pp. 365-391, 1999.
11. J. Herlocker, J. Konstan, A. Borchers, and J. Riedl. An algorithmic framework for performing collaborative filtering. In *Proceedings of the 1999 Conference on Research and Development in Information Retrieval*, August 1999.
12. E-H. Han, G. Karypis, V. Kumar, and B. Mobasher. 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.
13. E-H. Han, G. Karypis, V. Kumar, and B. Mobasher. 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.
14. G. Karypis, E-H. Han. Concept indexing: a fast dimensionality reduction algorithm with applications to document retrieval and categorization. *Technical Report #00-016*, Department of Computer Science and Engineering, University of Minnesota, March 2000.
15. 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.
16. D. Lewis, W. A. Gale. A sequential algorithm for training text classifiers. In *Proceedings of the 17th Annual ACM-SIGIR Conference*, (3) 12, London, UK, Springer-Verlag, 1994.
17. 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)*, November 1999.
18. B. Mobasher, R. Cooley, and J. Srivastava. Automatic personalization based on Web usage mining. In *Communications of the ACM*, (43) 8, August 2000.
19. B. Mobasher. A Web personalization engine based on user transaction clustering. In *Proceedings of the 9th Workshop on Information Technologies and Systems (WITS'99)*, December 1999.

20. O. Nasraoui, H. Frigui, A. Joshi, R. Krishnapuram. Mining Web access logs using relational competitive fuzzy clustering. In *Proceedings of the Eight International Fuzzy Systems Association World Congress*, August 1999.
21. M. O'Conner, J. Herlocker. Clustering items for collaborative filtering. In *Proceedings of the ACM SIGIR Workshop on Recommender Systems*, Berkeley, CA, 1999.
22. M. Perkowitz and O. Etzioni. Adaptive Web sites: automatically synthesizing Web pages. In *Proceedings of Fifteenth National Conference on Artificial Intelligence*, Madison, WI, 1998.
23. M. Spiliopoulou and L. C. Faulstich. WUM: A Web Utilization Miner. In *Proceedings of EDBT Workshop WebDB98*, Valencia, Spain, *LNCS 1590*, Springer Verlag, 1999.
24. M. Spiliopoulou, C. Pohle, and L. C. Faulstich. Improving the effectiveness of a Web site with Web usage mining. In *Proceedings of the Workshop on Web Usage Analysis and User Profiling (WebKKD99)*, San Diego, August 1999.
25. J. Srivastava, R. Cooley, M. Deshpande, P-T. Tan. Web usage mining: discovery and applications of usage patterns from Web data. *SIGKDD Explorations*, (1) 2, 2000.
26. B. M. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Analysis of recommender algorithms for e-commerce. In *Proceedings of the 2nd ACM E-Commerce Conference (EC'00)*, October 2000, Minneapolis.
27. S. Schechter, M. Krishnan, and M. D. Smith. Using path profiles to predict http requests. *In Proc. 7th International World Wide Web Conference*, April 1998, Brisbane, Australia.
28. U. Shardanand, P. Maes. Social information filtering: algorithms for automating "word of mouth." In *Proceedings of the ACM CHI Conference (CHI95)*, 1995.
29. 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.
30. 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.
31. P. S. Yu. Data mining and personalization technologies. In *Proceedings of the Int'l Conference on Database Systems for Advanced Applications (DASFAA99)*, April 1999, Hsinchu, Taiwan.
32. 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.