A Hybrid Web Personalization Model Based on Site Connectivity

Miki Nakagawa, Bamshad Mobasher
{mnakagawa,mobasher}@cs.depaul.edu
School of Computer Science, Telecommunication, and Information Systems
DePaul University, Chicago, Illinois, USA

ABSTRACT
Web usage mining has been used effectively as an underlying mechanism for Web personalization and recommender systems. A variety of recommendation frameworks have been proposed, including some based on non-sequential models, such as association rules and clusters, and some based on sequential models, such as sequential or navigational patterns. Our recent studies have suggested that the structural characteristics of Web sites, such as the site topology and the degree of connectivity, have a significant impact on the relative performance of recommendation models based on association rules, contiguous and non-contiguous sequential patterns. In this paper, we present a framework for a hybrid Web personalization system that can intelligently switch among different recommendation models, based on the degree of connectivity and the current location of the user within the site. We have conducted a detailed evaluation based on real Web usage data from three sites with different structural characteristics. Our results show that the hybrid system selects less constrained models such as frequent itemsets when the user is navigating portions of the site with a higher degree of connectivity, while sequential recommendation models are chosen for deeper navigational depths and lower degrees of connectivity. The comparative evaluation also indicates that the overall performance of hybrid system in terms of precision and coverage is better than the recommendation systems based on any of the individual models.

1. Introduction

Development of effective techniques for Web personalization is an important area of research with many immediate applications in e-commerce Web information systems. Understanding the users' navigational preferences and behavior is an essential step in studying the effectiveness of a Web site. For example, the discovery of most likely access patterns empowers e-business providers to measure and improve the quality of a site structure by modifying the hyperlinks or by dynamically customizing a Web page for guiding the user to interesting information. Web usage mining [18, 5] is a widely used approach to capture and model Web user behavioral patterns from the log data generated by Web and application server.

Various Web usage mining techniques have been used to develop efficient and effective recommendation systems to provide individualized content to users based on their preferences and past behavior. For example, association rule mining has recently been studied as an approach to discover models for recommendation systems [8, 10, 14]. It is a non-sequential mining technique that does not preserve the ordering information among pageviews in user sessions. Sequential pattern mining and the discovery of frequent navigational paths (contiguous sequential patterns) take into account the ordering constraints inherent in navigational patterns. The use of navigational and sequential patterns for predictive user modeling has been extensively studied [7, 13, 15]. However, the primary focus of these studies has been on prefetching of Web pages (i.e., predicting a user's immediate next access) to improve server performance or network latency. In [9, 19] a unified formal framework is presented to capture various navigational substructures in the usage data, including frequent itemsets and sequential patterns, and a generalized approach to personalization is proposed using navigational path fragments.

All recommender systems based on Web usage mining techniques have strength and weaknesses. For instance, sequential models generally produce accurate recommendations, but such models are often to selective resulting in unacceptably low coverage. In contrast, recommendation models based on less constrained patterns such as clustering and association rules can capture a broader range of recommendations, though this is sometimes at the cost of lower prediction accuracy. The common wisdom suggests that using more information about users' behavior results in more accurate models. In general, however, unnecessary constraints imposed during the pattern discovery process may result in missing important relationships in the mining results. Thus, two important questions that need to be answered in the context of personalization are:

- What are the conditions under which one model

is more effective?

- Can the mitigating factors be quantified in a way to allow the right model to be selected on-the-fly during users interaction?

In this paper we consider one such factor, namely, the degree of hyperlink connectivity within the site. In a previous study [12], we conducted a detailed comparative evaluation of non-sequential and sequential pattern recommendation models and considered their effectiveness and suitability. The study showed that the performance of each recommendation model depends, in part, on the structural characteristics of the Web site, in general, and on the degree of localized hyperlink connectivity, in particular. For example, in a highly connected Web site with short navigational paths, non-sequential models perform well by achieving higher overall precision and coverage than sequential pattern models. The study indicates that the degree of hyperlink connectivity may be a key element that determines the performance of the recommendation model.

In this paper, we present a framework for a hybrid Web personalization system that can intelligently switch among different recommendation models, based on a localized degree of hyperlink connectivity with respect to a user's current location within the site. The hybrid system selects less constrained models such as frequent itemsets when the user is navigating portions of the site with a higher degree of connectivity, and it selects sequential recommendation models for deeper navigational depths and lower degrees of connectivity. We perform a detailed evaluation of this proposed model using real Web usage data from three sites with different structural characteristics and show that the overall performance of hybrid system is better than the recommendation systems based on any of the individual models.

In the next section, we provide some background information and present the details of our recommendation models using association rule mining (AR), sequential pattern mining (SP), and contiguous sequential mining (CSP). In section 3, we briefly review the results of our previous experiments in [12], and provide the motivation for the proposed hybrid framework. In the rest of the paper we introduce our hybrid recommendation model and present detailed experimental results assessing the performance of the hybrid model when compared to the individual models.

2. Background: Personalization Based on Web Usage Mining

The overall process of Web personalization, generally consists of three phases: data preparation and transformation, pattern discovery, and recommendation. In traditional collaborative filtering approaches, the pattern discovery phase (e.g., neighborhood formation in the k-nearest-neighbor) as well as the recommendation phase are performed in real time. In contrast, personalization systems based on Web usage mining [11], perform the pattern discovery phase offline. Data preparation phase transforms raw web log files into clickstream data that can be processed by data mining tasks. A variety of data mining techniques can be applied to the clickstream or Web application data in the pattern discovery phase, such as clustering, association rule mining [1, 2, 17], and sequential pattern discovery [3]. The recommendation engine considers the active user session in conjunction with the discovered patterns to provide personalized content. The personalized content can take the form of recommended links or products, or targeted advertisements tailored to the user's perceived preferences as determined by the matching usage patterns. In this paper, our focus is specifically on association rule mining and sequential pattern discovery, and the suitability of the resulting patterns for personalization.

2.1 Pattern Discovery from Web Transactions

The required high-level tasks in the data preparation phase are data cleaning, user identification, session identification, pageview identification, and the inference of missing references due to caching. Pageview identification is the task of determining which page file accesses contribute to a single browser display. Transaction identification can be performed as a final preprocessing step prior to pattern discovery in order to focus on the relevant subsets of pageviews in each user session. In the present work we rely on the preprocessing techniques discussed in [6, 4] to perform the data preparation tasks on our experimental data sets.

The output of the data preparation phase is a set of n pageviews, $P = \{p_1, p_2, \cdots, p_n\}$, 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. Conceptually, we view each transaction t as an l-length sequence of ordered pairs:

$$t = \left\langle (p_1^t, w(p_1^t)), (p_2^t, w(p_2^t)), \cdots, (p_l^t, w(p_l^t)) \right\rangle,$$

where each $p_i^t = p_j$ for some $j \in \{1, \cdots, n\}$, and $w(p_i^t)$ is the weight associated with pageview p_i^t in the transaction t. Weights can be binary, representing the existence or non-existence of a product-purchase or a documents access in the transaction; or they can be a function of the duration of the associated pageview in the user's session. In this paper, our focus is on association rule and sequential pattern discovery, thus we only consider binary weights on pageviews within user transactions. In the case of association rule discovery, we ignore the ordering among the pageviews. In that

case, a transaction can be viewed as a set of pageviews $s_t = \{p_i^t \mid 1 \leq i \leq l \text{ and } w(p_i^t) = 1\}$. In the case of sequential (and contiguous sequential) patterns, however, we preserve the ordering relationship among the pageviews in the transactions.

Given a set of transactions as described above, a variety of unsupervised knowledge discovery techniques can be applied to obtain patterns. In the present work, we focus on three data mining techniques: Association Rule mining (AR), Sequential Pattern (SP), and Contiguous Sequential Pattern (CSP) discovery. CSP's are a special form of sequential patterns in which the items appearing in the sequence must be adjacent with respect to the underlying ordering. In the context of Web usage data, CSP's can be used to capture *frequent navigational paths* among user trails [16, 15]. In contrast, items appearing in SP's, while preserving the underlying ordering, need not be adjacent, and thus they represent more general navigational patterns within the site. Frequent item sets, discovered as part of association rule mining, represent the least restrictive type of navigational patterns, since they focus on the presence of items rather than the order in which they occur within user session.

Association rules capture the relationships among items based on their patterns of co-occurrence across transactions (without considering the ordering of items). In the case of Web transactions, association rules capture relationships among pageviews based on the navigational patterns of users. For the current paper we have used the Apriori algorithm [2, 17] that follows a generate-and-test methodology. This algorithm finds groups of items (in this case the pageviews appearing in the preprocessed log) occurring frequently together in many transactions (i.e., satisfying a user specified minimum support threshold). Such groups of items are referred to as *frequent item sets*.

Given a transaction T and a set $I = \{I_1, I_2, \ldots, I_k\}$ of frequent itemsets over T. The *support* of an itemset $I_i \in I$ is defined as $\sigma(I_i) = |\{t \in T : I_i \subseteq t\}|/|T|$.

Association rules which satisfy a minimum *confidence* threshold are then generated from the frequent itemsets. An association rule r is an expression of the form $X \Rightarrow Y(\sigma_r, \alpha_r)$, where X and Y are itemsets, σ_r is the support of $X \cup Y$, and α_r is the confidence for the rule r given by $\sigma(X \cup Y)/\sigma(X)$.

Sequential patterns in Web usage data capture the Web page trails that are often visited by users, in the order that they were visited. Sequential patterns are those sequences of items that frequently occur in a sufficiently large proportion of transactions. A *sequence* $< s_1, s_2, \ldots, s_n >$ occurs in a transaction $t = \langle p_1, p_2, \ldots, p_m \rangle$ (where $n \leq m$) if there exist n positive integers $1 \leq a_1 < a_2 < \ldots < a_n \leq m$, and $s_i = p_{a_i}$ for all i. We say that $\langle cs_1, cs_2, \ldots, cs_n \rangle$ is a *contiguous sequence* in t if there exists an integer $0 \leq b \leq m - n$, and $cs_i = p_{b+i}$ for all $i = 1$ to n. In a contiguous sequential pattern, each consecutive pair of elements, s_i and s_{i+1}, must appear consecutively in a transaction t which supports the pattern, while sequential pattern can represent non-contiguous frequent sequences in the underlying set of transactions.

Given a transaction set T and a set $S = \{S_1, S_2, \ldots, S_n\}$ of frequent sequential (respectively, contiguous sequential) pattern over T, the support of each S_i is defined as follows:

$$\sigma(S_i) = \frac{|\{t \in T : S_i \text{ is (contiguous) subsequence of } t\}|}{|T|}$$

The confidence of the rule $X \Rightarrow Y$, where X and Y are (contiguous) sequential patterns, is defined as $\alpha(X \Rightarrow Y) = \sigma(X \circ Y)/\sigma(X)$, where $\circ$ represents the concatenation operator on sequences. The Apriori algorithm used in association rule mining can also be adopted to discover sequential and contiguous sequential patterns. This is normally accomplished by changing the definition of support to be based on the frequency of occurrences of subsequences of items rather than subsets of items.

2.2 Using the Discovered Patterns for Personalization

The recommendation engine is the online component of the personalization process. In standard collaborative filtering, the recommendation engine is integrated with the "neighborhood formation" phase. In our context, the recommendation engine takes a collection of frequent itemsets or (contiguous) sequential patterns as input and generates a recommendation set by matching the current user's activity against the discovered patterns. In this section, we present two efficient recommendation generation algorithms based on frequent itemsets and sequential (resp. contiguous sequential) patterns, which use the discovered patterns together with the user's activity history to produce recommendations. We use a fixed-size sliding window over the current 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 recommendation value of items in the recommendation set. We call this sliding window, the user's *active session window*.

The recommendation engine based on association rules matches the current user session window with frequent itemsets to find candidate pageviews for giving recommendations. Given an active session window w and a group of frequent itemsets, we only consider all the frequent itemsets of size $|w| + 1$ containing the current session window. The recommendation value of each candidate pageview is based on the confidence of the corresponding association rule whose consequent

T1: $\{ABDE\}$
T2: $\{ABECD\}$
T3: $\{ABEC\}$
T4: $\{BEBAC\}$
T5: $\{DABEC\}$

Size 1	Size 2	Size 3	Size 4
$\{A\}(5)$	$\{A,B\}(5)$	$\{A,B,C\}(4)$	$\{A,B,C,E\}(4)$
$\{B\}(6)$	$\{A,C\}(4)$	$\{A,B,E\}(5)$	
$\{C\}(4)$	$\{A,E\}(5)$	$\{A,C,E\}(4)$	
$\{E\}(5)$	$\{B,C\}(4)$	$\{B,C,E\}(4)$	
	$\{B,E\}(5)$		
	$\{C,E\}(4)$		

Table 1. Sample Web Transactions and Frequent Itemsets generated by Apriori algorithm

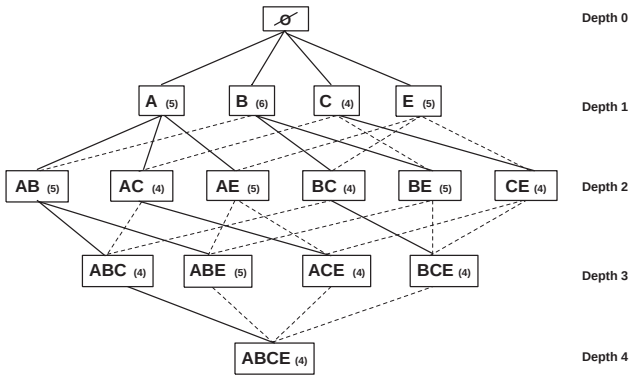


Figure 1. The Frequent itemsets Graph for example

is the singleton containing the pageview to be recommended.

In order to facilitate the search for itemsets (of size $|w|+1$) containing the current session window w, the frequent itemsets are stored in a directed acyclic graph, here called a *Frequent Itemset Graph*. The Frequent Itemset Graph is an extension of the lexicographic tree used in the tree projection algorithm of [1]. The graph is organized into levels from 0 to k, where k is the maximum size among all frequent itemsets. Each node at depth d in the graph corresponds to an itemset, I, of size d and is linked to itemsets of size $d+1$ that contain I at level $d+1$. The single root node at level 0 corresponds to the empty itemset. To be able to match different orderings of an active session with frequent itemsets, all itemsets are sorted in lexicographic order before being inserted into the graph. The user's active session is also sorted in the same manner before matching with patterns.

Given an active user session window w, sorted in lexicographic order, a depth-first search of the Frequent Itemset Graph is performed to level $|w|$. If a match is found, then the children of the matching node n containing w are used to generate candidate recommendations. Each child node of n corresponds to a frequent itemset $w \cup \{p\}$. In each case, the pageview p is added to the recommendation set if the support ratio $\sigma(w \cup \{p\})/\sigma(w)$ is greater than or equal to α, where α is a minimum confidence threshold. Note that $\sigma(w \cup \{p\})/\sigma(w)$ is the confidence of the association rule $w \Rightarrow \{p\}$. The confidence of this rule is also used as the recommendation score for pageview p. It is easy to observe that in this algorithm the search process requires only $O(|w|)$ time given active session window w.

To illustrate the process, consider the example transaction set given in Table 1 (top). Using these transactions, the Apriori algorithm with a frequency threshold of 4 (minimum support of 0.8) generates the itemsets shown in the bottom table. Figure 1 shows the Frequent Itemsets Graph constructed based on the frequent itemsets in Table 1. Now, given user active session window $< B, E >$, the recommendation generation algorithm finds items A and C as candidate recommendations. The recommendation scores of item A and C are 1 and 4/5, corresponding to the confidences of the rules $\{B,E\} \rightarrow \{A\}$ and $\{B,E\} \rightarrow \{C\}$, respectively.

The recommendation algorithm based on association rules can be adopted to work also with sequential (respectively, contiguous sequential) patterns. In this case, we focus on frequent (contiguous) sequences of size $|w|+1$ whose prefix contains an active user session w. The candidate pageviews to be recommended are the last items in all such sequences. The recommendation values are based on the confidence of the patterns. If the confidence satisfies a threshold requirement, then the candidate pageviews are added to the recommendation set.

A simple trie, an efficient data structure for storing strings in which there is one node for every common prefix, is used to store both the SP and CSP discovered during the pattern discovery phase. The *Frequent Sequence Trie* (FST) is organized into levels from 0 to k, where k is the maximal size among all SPs or CSPs. There is the single root node at depth 0 containing the empty sequence. Each non-root node N at depth d contains an item s_d and representing a frequent sequence $\langle s_1, s_2, \ldots, s_{d-1}, s_d \rangle$ whose prefix $\langle s_1, s_2, \ldots, s_{d-1} \rangle$ is the pattern represented by the parent node of N at depth $d-1$. Furthermore, along with each node we store the support value of the corresponding pattern. The confidence of each pattern (represented by a non-root node in the FST) is obtained by dividing the support of the current node by the support of its parent node.

The recommendation algorithm based on sequential and contiguous sequential patterns has a similar structure as the algorithm based on association rules. For each active session window $w = \langle w_1, w_2, \ldots, w_n \rangle$, we perform a depth-first search of the FST to level n. If a match is found, then the children of the matching

Size 1	Size 2	Size 3
$< A > (5)$	$< A, B > (4)$	$< A, B, E > (4)$
$< B > (6)$	$< A, C > (4)$	$< A, E, C > (4)$
$< C > (4)$	$< A, E > (4)$	
$< E > (5)$	$< B, C > (4)$	
	$< B, E > (5)$	
	$< C, E > (4)$	

Size 1	Size 2
$< A > (5)$	$< A, B > (4)$
$< B > (6)$	$< B, E > (4)$
$< C > (4)$	
$< E > (5)$	

Table 2. Frequent Sequential Patterns (top) and Contiguous Sequential Patterns (bottom)

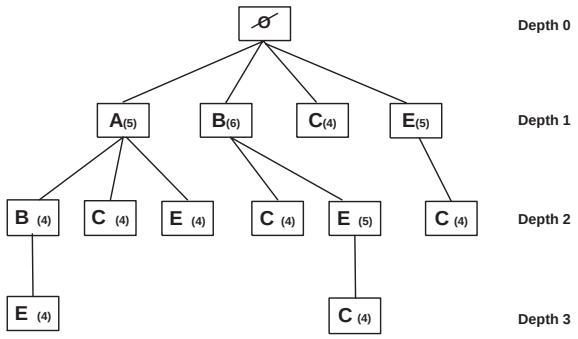


Figure 2. The Frequent Sequential Pattern Trie Structure for the Example

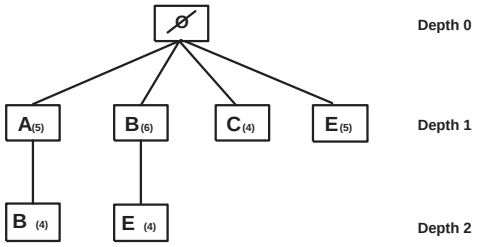


Figure 3. The Frequent Contiguous Sequential Pattern Trie Structure for the Example

node N are used to generate candidate recommendations. Given a sequence $S = \langle w_1, w_2, \ldots, w_n, p \rangle$ represented by a child node of N, the item p is then added to the recommendation set as long as the confidence of S is greater than or equal to the confidence threshold. As in the case of frequent itemset graph, the search process requires $O(|w|)$ time given active session window size $|w|$.

To continue our example, Table 2 shows the frequent sequential patterns and frequent contiguous sequential patterns with a frequency threshold of 4 over the example transaction set given in Table 1. Figures 2 and 3 show the trie representations of the sequential and contiguous sequential patterns listed in the Table 2, respectively. The sequential pattern $< A, B, E >$ appears in the figure 2 because it is the subsequence of 4 transactions: T_1, T_2, T_3 and T_5. However, $< A, B, E >$ is not a frequent contiguous sequential pattern since only 3 transactions (T_2, T_3 and T_5) contain the contiguous sequence $< A, B, E >$. Give a user's active session window $< A, B >$, the recommendation engine using sequential patterns finds item E as a candidate recommendation. The recommendation score of item E is 1, corresponding to the rule $< A, B > \Rightarrow < E >$. On the other hand, the recommendation engine using contiguous sequential patterns will, in this case, fails to give any recommendations.

Depending on the specified support threshold, it might be difficult to find large enough itemsets or sequential patterns that could be used for providing recommendations, leading to reduced coverage. This is particularly true for sites with very small average session sizes. In order to overcome this problem, we use *all-kth-order* method proposed in [13] in the context of *Markov chain models*. The *order* of the Markov model corresponds to the number of prior events used in predicting a future event. So, a kth-Order Markov model predicts user's next action by looking the past k actions. Higher-order Markov models generally provide a higher prediction accuracy. However, this is usually at the cost of lower coverage and much higher model complexity due to the larger number of states. The use of all-kth-order Markov models generally requires the generation of separate models for each of the k orders: if the model cannot make a prediction using the kth order, it will attempt to make a prediction by incrementally decreasing the model order.

Our recommendation framework for contiguous sequential patterns is essentially equivalent to kth-order Markov models, however, rather than storing all navigational sequences, we only store frequent sequences resulting from the sequential pattern mining process. In this sense, our method is similar to support pruned models described in [7], except that the support pruning is performed by the Apriori algorithm in the mining phase. The notion of all-kth-order models and also easily be extended to the context of general sequential patterns and association rule. We extend our recommendation algorithms to generate all-kth-order recommendations as follows. First, the recommendation engine uses the largest possible active session window as an input for recommendation engine. If the engine cannot generate any recommendations, the size of active session window is iteratively decreased until a recommendation is generated or the window size becomes 0. We use this extended recommendation framework for all 3 approaches in our experiments discussed in the next section.

2.3 Evaluating Usage-Based Recommendations

To evaluate the effectiveness of each recommendation model, the transaction data set is divided into training and evaluation sets. The training set is used to generate the models based on AR, SP, and CSP, while the evaluation set is used to test the generated model. Each transaction t in the evaluation set is divided into two parts. The first n pageviews in t are used for generating recommendations, whereas, the remaining portion of t is used to evaluate the generated recommendations. The active session window is the portion of the user's clickstream used by the recommendation engine in order to produce a recommendation set. We call this portion of the transaction t the *active session with respect to* t, denoted by as_t. The recommendation engine takes as_t and a recommendation threshold τ as inputs and produce a set of pageviews as recommendations. We denote this recommendation set by $R(as_t, \tau)$.

The set of pageviews $R(as_t, \tau)$ can now be compared with the remaining $|t| - n$, pageviews in t. We denote this portion of t by $eval_t$. Our comparison of these sets is based on 2 different metrics, namely, precision and coverage. The *precision* of $R(as_t, \tau)$ is defined as:

$$precision(R(as_t, \tau)) = \frac{|R(as_t, \tau) \cap eval_t|}{|R(as_t, \tau)|},$$

and the *coverage* of $R(as_t, \tau)$ is defined as:

$$coverage(R(as_t, \tau)) = \frac{|R(as_t, \tau) \cap eval_t|}{|eval_t|}.$$

Precision measures the degree to which the recommendation engine produces accurate recommendations while coverage measures the ability of the recommendation engine to produce all of the pageviews that are likely to be visited by the user. Finally, for a given recommendation threshold τ, the mean over all transactions in the evaluation set is computed as the overall evaluation score for each measure. We ran each set of experiments for thresholds ranging from 0.1 to 1.0.

3. The Impact of Site Characteristics on Recommendation Models

In [12], we performed a comparative evaluation of recommendation models based on association rules (AR), sequential patterns (SP) and contiguous sequential patterns (CSP). In this section, we briefly review the observed experimental results that have motivated the development of a hybrid recommender system based on the hyperlink connectivity.

Data Sets and Site Characteristics

We used the server logs from three different Web sites, each with its own structural and domain characteristics. The server logs used for the current experiments include the Association for Consumer Research (ACR) Newsletter (www.acr-news.org), the School of Computer Science, Telecommunication, and Information Systems (CTI) at DePaul University (www.cs.depaul.edu), and Network Chicago (NC) which combines the programs and activities associated with the Chicago Public Television and Radio (www.networkchicago.com). We performed 10-fold cross-validation using each of the three data sets. In each of the 10 iterations, the data set was divided into training (90%) and evaluation (10%) data sets. The training set was used to generate the models based on AR, SP, and CSP, and the evaluation set is used to test the generated models using the evaluation method described earlier.

The visitors to the ACR Web site tend to have a focused set of interests in specialized topics related to consumer psychology and marketing. The site is highly connected, but relatively shallow (maximum depth of 4) with all pages at levels 1 and 2 linked to all other pages at those levels. Furthermore, all pages in the site include navigation links to the top level pages as well as back links to pages in the parent level. The site is fairly small with close to 100 unique pageviews. The usage characteristics of the site reflect relatively short user sessions.

The CTI site, on the other hand, has a much broader audience. The visitors to this site include a variety of external users interested in a number of academic and research programs, as well as thousands of students, faculty, and staff, utilizing resources and applications available in the intranet portion of the site. The site is highly dynamic with many dynamically generated pages at deeper levels in the site. The site usage is characterized by fairly long navigational trails often reaching up to 10 or more levels deep.

The NC site also has a broad audience and is characterized by long navigational paths. Certain portions of this site (representing individual programs) are highly connected, but relatively deep navigational sequences are required to arrive at these connected components.

Figures 4 and 5 depict the underlying hyperlink structure for the ACR and the NC sites.

Observations and Discussion

Our results, reported in [12], suggest that in sites with a highly connected site structure, generally, AR and SP models, which capture less constrained navigational patterns, are better choices for personalization.

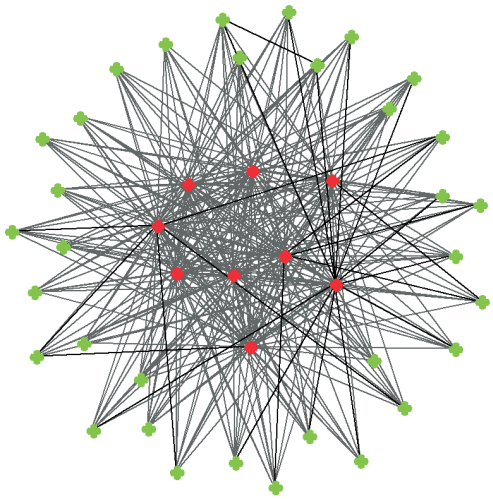


Figure 4. The Hyperlink Graph for the ACR site

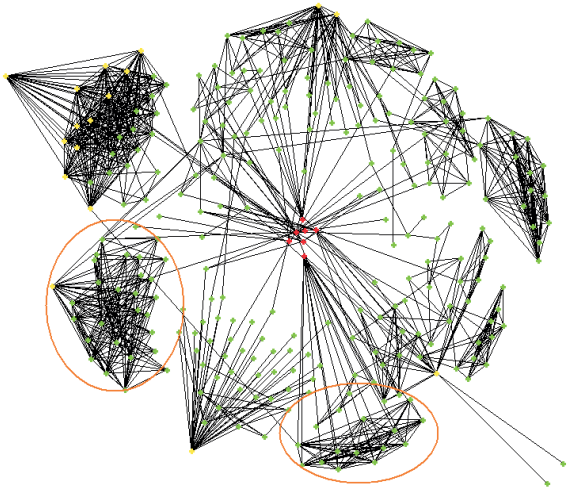


Figure 5. The Hyperlink Graph for the NC site

the precision and coverage results for the ACR data set support this observation. On the other hand, In a sites (or portions of a site), involving deeper substructures and longer paths (particularly sites with dynamically generated pages where the a pageview often depends on the results of previous actions by users), CSP models generally provide more accurate predictions of user interests. For example, the sequential models performed better in overall precision in the NC and CTI sites while the AR model did not provide a significant advantage in terms of coverage. In the NC site, where the degree of connectivity falls in the midpoint of the range relative to the ACR and CTI sites, the SP model seems to provide the best overall performance. Another experiment [12] on a highly connected subgraph of the NC site (the circled parts of Figure 5 verified that in that case the AR recommendation model achieved both high precision and coverage, very similar to that of ACR site.

These results lead to the hypothesis that the performance difference among these models is related, in part, to the structural characteristics of the site and the hyperlink connectivity in particular. Specifically, in highly connected sites (or a in a highly connected subgraph within the site), it is likely that a narrow focus on contiguous sequences (i.e., navigational paths) may not provide an advantage for personalization. On the contrary these more constrained models may lead to unacceptably low coverage without providing an advantage in terms of precision.

The above observations motivate us to develop a hybrid recommender system which can automatically switch among recommendation models based on the "localized" hyperlink connectivity of the site and the location of the user. We present such a framework in the next Section.

4. The Hybrid Recommendation Framework Based in Site Connectivity

In this section, we present an algorithm for a hybrid Web personalization system, which optimizes its recommendations by automatically switching recommendation models based on the characteristics of Web site.

4.1 A Localized Connectivity Measure

The proposed hybrid recommendation model uses the degree of hyperlink connectivity, within the "neighborhood" of a user's current location, as a switching criterion to select the best recommendation model. More specifically, the model uses the a *localized hyperlink connectivity* measure. To compute this quantity, we start by representing the Web pages as a collection of interconnected documents and use a directed graph to model the Web site. Let $G = (V, E)$ be a directed hyperlink graph, where V is a set of nodes and E is a set of edges. In G, a pageview is represented as a node, and a hyperlink is represented as an edge. Each edge (u, v) is an ordered pair of nodes, where u and v represent a directed hyperlink connection from a pageview u to another pageview v, with $u, v \in V$ and $u \neq v$.

To measure the local connectivity of an individual pageview, we first generate an induced subgraph from the site graph $G = (V, E)$. Any given node u induces a subgraph $G'(u) = (V', E')$ that contains the node u itself and its child nodes. This represents a local hyperlink structure of a pageview u and its neighborhood. In other words, we first specify a set of nodes $V' = \{u \cup V''\}$, where $V'' = \{v | \exists (u, v) \in E \text{ and } v \in V\}$. Next, we select all of the edges $E' = \{(u, v) \in E | u, v \in V'\}$ from the original graph that connect two nodes in V'. Given an induced subgraph $G'(u) = (V', E')$, we com-

pute the *localized connectivity measure with respect to* u as

$$LCM(u) = \frac{|E'|}{|V'| * (|V'| - 1)},$$

where $|E'|$ and $|V'|$ are the number of hyperlinks and pageviews in the induced subgraph $G'(u)$, respectively. The denominator of the equation represents the maximum possible directed links in $G'(u)$, which increases in proportion to the square of the number of nodes.

In the above definition, the LCM measure was computed using only one level of connectivity (representing the local connectivity among children of a pageview u). We can also compute the LCM at deeper levels. For example, at level 2, the LCM measures the localized connectivity in a neighborhood of a pagview u containing u as well as its children and grandchildren. In our experiments, going deepr than one level in computing the LCM did not seem to provide sigificant advantages. However, this is an areas that, we believe, requires further study.

4.2 Hybrid Recommendation Engine

We use the *localized connectivity measure* as a switching criterion to develop a hybrid recommendation engine, which automatically select one of three recommendation models based on Association Rules, Sequential Patterns and Contiguous Sequential Pattern.

To accomplish this, we employ logistic regression analysis to predict the best recommendation model. First, using the training set we apply the three recommendation models (i.e., AR, SP, CSP) and compute the average precision and coverage of each model. Next, we assign binary digits to each model: 1 for the best model with the highest precision and coverage and 0 for the other two. We also compute the LCM of each pageview in the active session. For each recommendation model, we use logistic regression to generate a model that predicts the probability of being selected as the best recommendation model as a function of the LCM.

Figures 6 and 7 show the logistic regression functions for each of recommendation models in the CTI and the NC sites. The AR model has the highest probability of being the best model at the high hyperlink connectivity values. The CSP model, on the other hand, has the highest probability of being a best model at the lowest connectivity values. The probability of selection for the SP model is relatively flat across the entire range of the hyperlink connectivity values.

The hybrid recommendation engine consists of two parts. The recommendation engine first finds the best recommendation model based on the results of logistic regression analysis and then uses the selected model to generates a recommendation set for current user. Note that the computation of the LCM values for

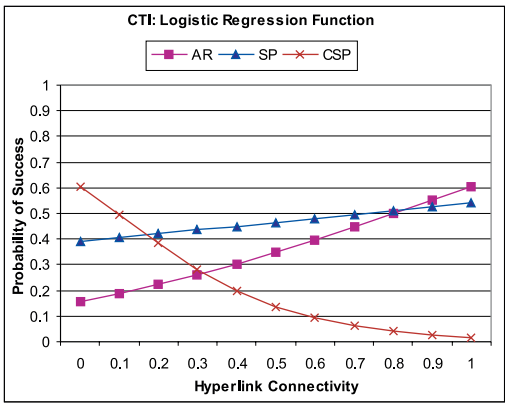


Figure 6. The Logistic Regression Functions for the CTI Site

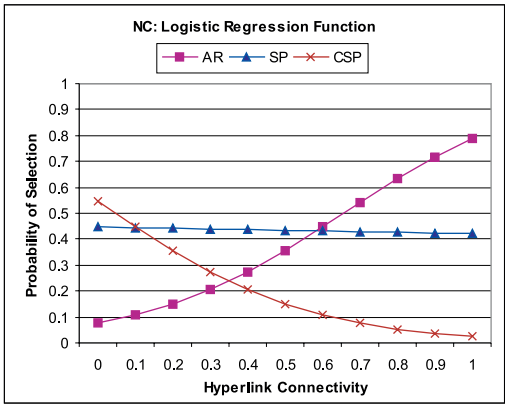


Figure 7. The Logistic Regression Functions for the NC site

all pageviews, as well as the logistic regression analysis are performed offline, and not during the real-time recommendation process.

More specifically, the recommendation engine takes an active user session as an input to the recommendation model with highest probability of selection according to the local connectivity measure of the last page in the active session. For example, if the active user session is $\langle A, B, C \rangle$, the hybrid recommendation engine calculates the probability of success for each recommendation model using the connectivity of pageview C. Then it finds all recommendation candidates by matching discovered frequent patterns against an active user session using the selected recommendation model. A candidate item satisfying the minimum confidence threshold is added to the recommendation set for the user.

4.3 Performance Evaluation of the Hybrid Model

The primary goal of this section is to evaluate the relative performance of a hybrid recommendation system to recommendation models based on Association Rules (AR), Sequential Patterns (SP), and Contiguous Sequential Patterns (CSP). In order to accurately compare the effectiveness of personalization, we used precision and coverage as performance measures. For the performance evaluation of the hybrid model, we use datasets used in the previous experiments, namely the Association for Consumer Research (ACR), the School of CTI at DePaul University (CTI), and Network Chicago (NC). A summary of the server logs for each dataset is shown in Table 3, including the average length of transactions. The usage characteristics of the ACR site reflect relatively short user transactions, while the CTI site usage is characterized by fairly long navigational trails often reaching up to 10 or more levels deep. The usage characteristics of the NC site is a mixture of short and long user transactions.

Dataset	Transactions	Pageviews	Avg. Length
ACR	7,832	40	2.945
CTI	3,283	299	8.006
NC	44,582	372	3.270

Table 3. Summary of Usage Data for ACR, CTI and NC datasets

Figure 8 shows the precision and coverage of AR, SP, CSP and the hybrid recommendation models on the NC dataset. We find that the hybrid model achieves higher precision than both SP and AR models across different recommendation thresholds and becomes comparable to the highest precision of the CSP model, particularly at the high recommendation thresholds. The hybrid model also achieves higher coverage than the models based on the sequential pattern mining (i.e., SP and CSP) and in par with the highest coverage levels of the non-sequential pattern mining technique (i.e., AR) at high recommendation thresholds. Although there exists a trade off between precision and coverage, the analysis demonstrates that the hybrid recommendation model seems to optimize both precision and coverage.

Figure 9 shows the performance evaluation of recommendation models on the CTI dataset. These results show that both the hybrid and SP models achieve relatively high precision and coverage, particularly at high recommendation thresholds. The CSP model achieves the highest precision, but the model suffers from the unacceptably low coverage. The AR model, on the other hand, has good coverage but it has the lowest precision. It is also interesting to note that the performance of the hybrid model is very similar to that

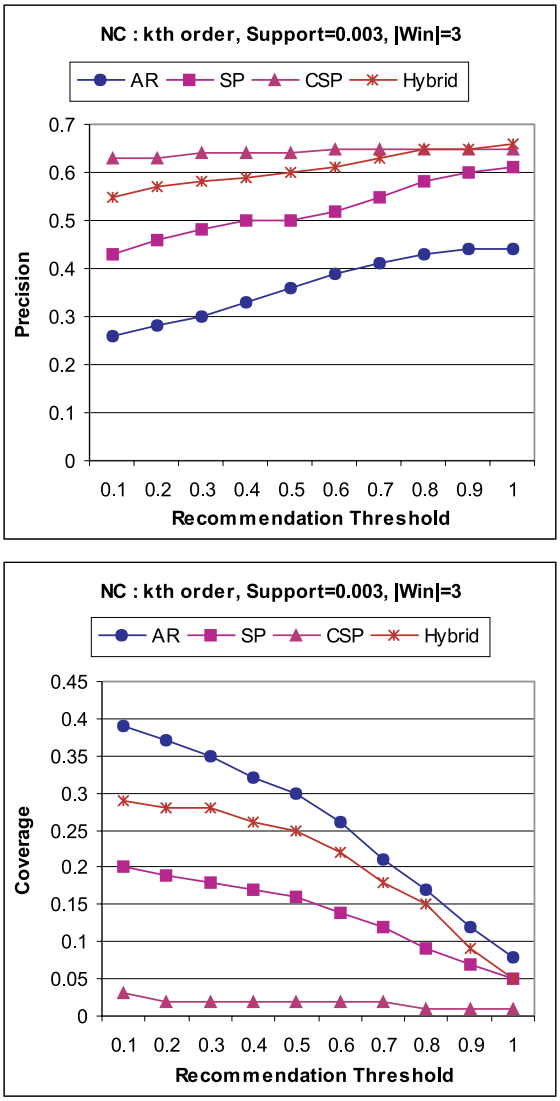
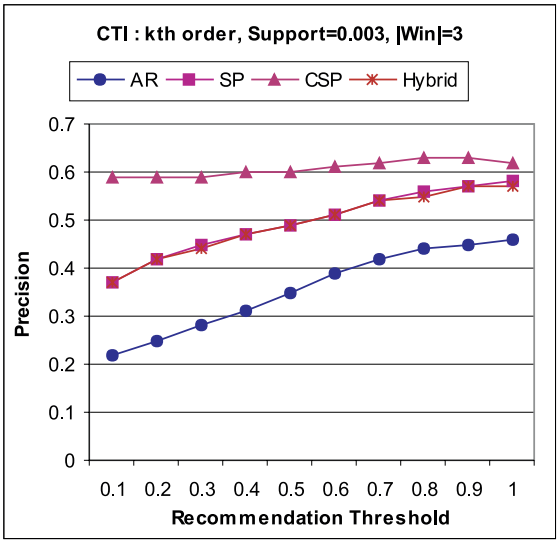


Figure 8. Performance of Hybrid Recommender System Compared to Association Rule and Sequential Pattern Models on the NC data set.)

of the SP model. This is because the hybrid model selected the SP model as an optimal recommendation model for more than 75% of user sessions in the evaluation sets.

The performance evaluation on the ACR site (not shown) indicated that the hybrid model has a similar performance to the association rule model for that site in terms of coverage and precision. Indeed, the hybrid framework selected the AR model in more than 80% of sessions in the ACR site, and provided a slight improvement in precision over the AR model.

Note that, in contrast to the results presented earlier, we have used the *kth-order* method for all analyses presented in this section. In the previous section, we applied the *all-kth-order* method to evaluate the per-



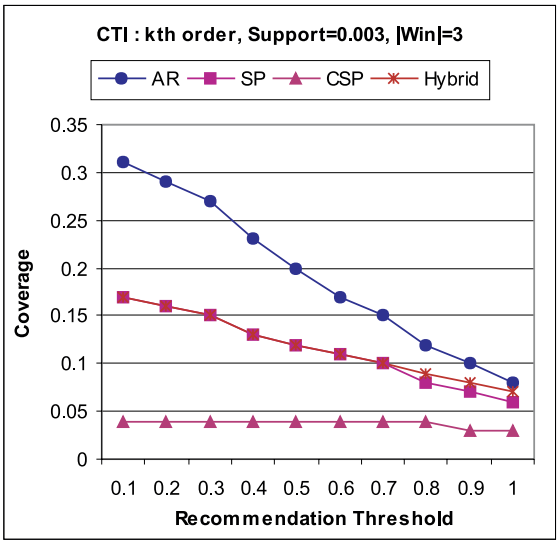


Figure 9. Performance of Hybrid Recommender System Compared to Association Rule and Sequential Pattern Models on the CTI data set

formance differences between recommendation models. Although the CSP results with *all-kth-order* improved the coverage by 0.17, the precision has decreased by 0.2. On the other hand, the hybrid model with *kth-order* improved the coverage by 0.25 without significantly degrading precision.

5. Conclusions

Generally speaking, sequential recommendation models (such as those based on sequential navigational patterns) produce fairly accurate recommendations, but such models do not generate enough recommendations and often result in unacceptably low coverage. In contrast, recommendation models based on less constrained patterns such as clustering and association rules can capture broader range of recommendations, but they often lack in accuracy when compared to sequential models. Our previous studies have shown that the performance of each recommendation model depends, in part, on the structural characteristics of the Web site and the degree of hyperlink connectivity, in particular.

In this paper, we have presented a framework for a hybrid Web personalization framework that can intelligently switch among different recommendation models, based on a localized connectivity measure. Our studies shows that the hybrid system selects less constrained models such as frequent itemsets when the user is navigating portions of the site with a higher degree of connectivity, and it selects sequential recommendation models for deeper navigational paths and lower degrees of connectivity. Overall, the results presented here show that the hybrid model can be used to develop a more effective and intelligent personalization framework when compared with any of the individual sequential or non-sequential models. In particular, our hybrid recommender system can generate not only accurate but also a wider range of recommendations.

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, April 1999.

[2] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In *Proceedings of the 20th International Conference on Very Large Data Bases (VLDB'94)*, Santiago, Chile, Sept 1994.

[3] R. Agrawal and R. Srikant. Mining sequential patterns. In *Proceedings of the International Conference on Data Engineering (ICDE'95)*, Taipei, Taiwan, March 1995.

[4] B. Berendt, B. Mobasher, M. Nakagawa, and M. Spiliopoulou. The impact of site structure and user environment on session reconstruction in web usage analysis. In *Proceedings of the 4th WebKDD 2002 Workshop, at the ACM-SIGKDD Conference on Knowledge Discovery in Databases (KDD'2000)*, Edmonton, Canada, July 2002.

[5] R. Cooley, B. Mobasher, and J. Srivastava. Web mining: Information and pattern discovery on the world wide web. In *Proceedings of the 9th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'97)*, Newport Beach, CA, November 1997.

[6] 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.

[7] M. Deshpande and G. Karypis. Selective markov models for predicting web-page accesses. In *Proceedings of the First International SIAM Conference on Data Mining*, Chicago, April 2001.

[8] X. Fu, J. Budzik, and K. J. Hammond. Mining navigation history for recommendation. In *Proceedings of the 2000 International Conference on Intelligent User Interfaces*, New Orleans, LA, January 2000. ACM Press.

[9] W. Gaul L. Schmidt-Thieme. Frequent substructures in web usage data: A unified approach. In *Proceedings of Web Mining Workshop, First SIAM International Conference on Data Mining 2001 (ICDM)*, Chicago, April 2001.

[10] W. Lin, S. A. Alvarez, and C. Ruiz. Efficient adaptive-support association rule mining for recommender systems. *Data Mining and Knowledge Discovery*, 6:83–105, 2002.

[11] B. Mobasher, R. Cooley, and J. Srivastava. Automatic personalization based on web usage mining. *Communications of the ACM*, 43(8):142–151, 2000.

[12] M. Nakagawa and B. Mobasher. Impact of site characteristics on recommendation models based on association rules and sequential patterns. In *Proceedings of the IJCAI 2003 Workshop on Intelligent Techniques for Web Personalization*, Acapulco, Mexico, August 2003.

[13] J. Pitkow and P. Pirolli. Mining longest repeating subsequences to predict www surfing. In *Proceedings of the 2nd USENIX Symposium on Internet Technologies and Systems*, Boulder, Colorado, October 1999.

[14] 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)*, Minneapolis, MN, October 2000.

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

[16] M. Spiliopoulou and L. Faulstich. Wum: A tool for web utilization analysis. In *Proceedings of EDBT Workshop at WebDB'98*, LNCS 1590, pages 184–203. Springer Verlag, 1999.

[17] R. Srikant and R. Agrawal. Mining generalized association rules. In *Proceedings of the 21st International Conference on Very Large Databases (VLDB'95)*, Zurich, Switzerland, September 1995.

[18] J. Srivastava, R. Cooley, M. Deshpande, and P. Tan. Web usage mining: Discovery and applications of usage patterns from web data. *SIGKDD Explorations*, 1(2):12–23, 2000.

[19] L. Schmidt-Thieme W. Gaul. Recommender systems based on navigation path features. In *Proceedings of WebKDD Workshop at the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD01)*, San Francisco, August 2001.