Impact of Site Characteristics on Recommendation Models Based On Association Rules and Sequential Patterns

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

Abstract

A number of studies have suggested the use of discovered Web usage patterns such as association rules, general sequential patterns, and contiguous sequential patterns (frequent navigational paths) for generating recommendations in personalization systems. To-date, however, no studies have considered the conditions under which recommendation models based on sequential patterns may be more appropriate for personalization as compared to those based on non-sequential patterns (such as frequent itemsets). We conjecture 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 these recommendation models. We present a framework for Web personalization based on association rules, contiguous and non-contiguous sequential patterns discovered from Web usage data. We then conduct a detailed comparative evaluation based on real Web usage data from three sites with different structural characteristics. Our results suggest that less constrained patterns, such as frequent itemsets, are better suited for personalization in sites with a higher degree of connectivity and shorter navigational depth, while the sequential recommendation models may be more suitable in sites with deeper navigational depth or in sites relying on many dynamically generated pages.

1 Introduction

One of the most important applications of Web usage mining [20] is predictive user modeling for the purpose of personalization. The goal of Web personalization is to provide users with dynamic content tailored to their individual interests. The personalization task generally takes the form of recommending one or more items or pages to a current user, possibly based on the patterns of past visitors who have similar profiles. Standard collaborative filtering (CF) [8] techniques used in personalization systems, such as k-Nearest-Neighbor, suffer from some well-known limitations [16]. For the most part these limitations are related to the scalability and efficiency of the kNN approach. Essentially, kNN requires that the neighborhood formation phase be performed as an online process, and for very large data sets this may lead to unacceptable latency for providing recommendations.

In contrast to standard CF-based approaches, Web usage mining techniques, that rely on offline pattern discovery from user transactions, can be used to improve the scalability of personalization systems. For example, previous work such as [10; 12] have considered automatic personalization based on clustering of user transactions and pageviews. The improvements in the scalability of collaborative filtering through clustering, though, is sometimes offset by reduced recommendation accuracy [13]. In general, the view of user sessions in a Web site as a sequence of pageviews also allows for the application of a variety of other data mining techniques in order to discover usage patterns. Some of these techniques, such as association rule discovery, generally ignore the inherent ordering relation among the items, and focus on non-sequential patterns based on co-occurrences of these items within sessions. On the other hand, techniques such as sequential pattern discovery or the discovery of frequent navigational patterns [18] take into account the ordering constraints among pageviews in user sessions.

Some recent studies have considered the use of association rule mining [2; 19] in recommender systems [7; 9; 16]. In [11], we proposed a framework for personalization based on association rule mining, utilizing an efficient data structure for storing the discovered frequent itemsets which are especially suitable for real-time recommender systems. In contrast to non-sequential patterns, such as association rules, sequential patterns [19] contain more precise information about user's navigational behavior. The use of navigational sequential patterns for predictive user modeling has been extensively studied [6; 15]. The primary focus of all of these studies has been on prefetching of Web pages (i.e., predicting a user's next access to a page) to improve server performance or network latency. In the context of personalization, however, the narrow focus on navigational sequences often leads

to very low recommendation coverage making such techniques less effective for generating a broad set of relevant recommendations.

In general using more fine-grained information about users' navigational histories as part of pattern discovery does not necessarily translate to more effective personalization. In particular, we conjecture that the structural characteristics of Web sites, such as the site topology, degree of connectivity, and the degree to which pages are dynamically generated, have a significant impact on the performance of recommendation models based on sequential patterns versus those based on non-sequential patterns. To this date, there have been no comprehensive studies comparing an evaluating the effectiveness of sequential and non-sequential pattern discovery techniques for the purpose of personalization. Furthermore, there have been no studies considering which site-relates factors might make the use of sequential patterns more appropriate than non-sequential patterns in personalization. Our goal in this paper is to provide such a study.

We first provide a framework for Web personalization based on association rules, contiguous and non-contiguous sequential patterns discovered from Web usage data. The framework integrates these different mining algorithms with efficient data structures and recommendation algorithms which are especially tailored for online generation of recommendations without the need for *a priori* rule generation from frequent itemsets or sequences. We then conduct a detailed comparative evaluation of sequential and non-sequential patterns in terms of their effectiveness and suitability for personalization tasks. The evaluation is performed based on real Web usage data from 3 different sites with different structural characteristics. Our results show that less constrained patterns, such as frequent itemsets or general sequential patterns are better suited for personalization in sites with a higher degree of connectivity and shorter navigational depth. On the other hand, more restrictive patterns, such as contiguous sequential patterns (e.g., frequent navigational paths) may be suitable for sites with substructure at deeper levels of navigation (including those resulting from dynamic pages generated through sequences of user interactions with applications).

2 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 [10], 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; 19], 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.

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 [5; 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, since our focus is on association rule and sequential pattern discovery, 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 [18; 17]. In contrast, items appearing in SP's, while preserv-

ing 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; 19] 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.

3 Recommendation Models Based on Sequential and Non-Sequential Patterns

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 represent efficient and scalable data structures for storing frequent itemset and sequential patterns, as well as a recommendation generation algorithms that use these data structures to directly produce real-time recommendations.

We use a fixed-size sliding window over the current active session to capture the current user's history depth. For example, if the current session (with a window size of 3) is $< A, B, C >$, and the user accesses the pageview D, then the new active session becomes $< B, C, D >$. Thus, 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. We call this sliding window, the user's *active session window*.

3.1 Recommendation Engine Based on Association Rules

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 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

Input: Session window $w = \langle p_1, \ldots, p_n \rangle$ in lexicographic order
Minimum confidence threshold α
Output: Recommendation set REC

$REC = \emptyset$
$currentNode = root$; $depth = 0$;
repeat
  $depth$++;
  if $\exists X \in currentNode.children$ with $X.itemset \subseteq \{p_1, \ldots, p_{depth}\}$
    $currentNode = X$;
  else
    $currentNode$ = NULL; Break;
until $depth > |w|$
if $currentNode \neq$ NULL then
  for each node $N \in currentNode.children$ do
    let $c = N.support/currentNode.support$;
    if $c \geq \alpha$
      $p = (N.itemset - w)$
      $p.rec\_score = c$
      $REC = REC \cup \{p\}$
    end if
  end for
end if

Figure 1: Recommendation Algorithm Based on Association Rules

is also sorted in the same manner before matching with patterns.

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

Table 1: Sample Web Transactions involving pageviews A, B, C, D and E

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. The details of the algorithm for the association-based recommendation engine are given in Figure 1.

To illustrate the process, consider the example transaction set given in Table 1. Using these transactions, the Apriori algorithm with a frequency threshold of 4

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 2: Frequent Itemsets generated by Apriori algorithm

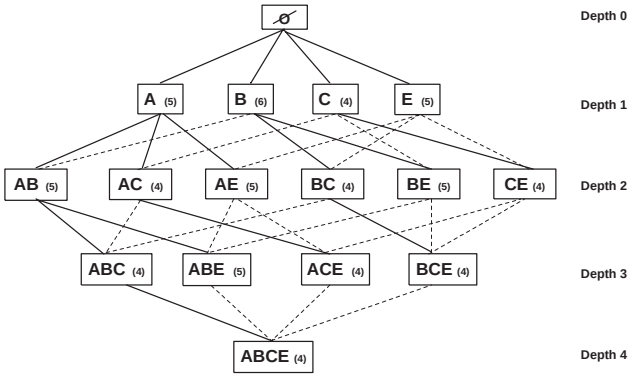


Figure 2: The Frequent itemsets Graph for example

(minimum support of 0.8) generates the itemsets given in Table 2. Figure 2 shows the Frequent Itemsets Graph constructed based on the frequent itemsets in Table 2. 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.

3.2 Recommendation Engine Based on Sequential Patterns

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 structure, which we call *Frequent Sequence Trie* (FST), is used to store both the sequential and contiguous sequential patterns discovered during the pattern discovery phase. The FST is organized into levels from 0 to k, where k is the maximal size among all sequential (respectively, contiguous sequential) patterns. 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 $< s_1, s_2, \ldots, s_{d-1}, s_d >$ whose prefix $< s_1, s_2, \ldots, s_{d-1} >$

Input: Session window $w = \langle p_1, \ldots, p_n \rangle$ in the original order
Minimum confidence threshold α
Output: Recommendation set REC

$REC = \emptyset$
$currentNode = root$; $depth = 0$;
repeat
  $depth$++;
  if $\exists X \in currentNode.children$ with $X.item = p_{depth}$
    $currentNode = X$;
  else
    $currentNode$ = NULL; Break;
until $depth > |w|$
if $currentNode \neq$ NULL then
  for each node $N \in currentNode.children$ do
    let $c = N.support/currentNode.support$;
    if $c \geq \alpha$ and $p \neq p_i$ for $i = 1, \ldots, n$
      $p = N.item$
      $p.rec\_score = c$
      $REC = REC \cup \{p\}$
    end if
  end for
end if

Figure 3: Recommendation Algorithm Based on Sequential or Contiguous Sequential Patterns

is the pattern represented by the parent node of N at depth $d-1$. Furthermore, along with each node we store the support (or frequency) 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.

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

Table 3: Frequent Sequential Patterns

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 $w =< w_1, w_2, \ldots, w_n >$, we perform a depth-first search of the FST to level n. If a match is found, then the children of the matching node N are used to generate candidate recommendations. Given a sequence $S =< w_1, w_2, \ldots, w_n, p >$ 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 $|w|$. The details of this algorithm are given in Figure 3.

To continue our example, Table 3 and 4 show 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 4 and 5 show the trie representation of the sequential and contiguous sequential patterns listed in the Table 3 and 4, respectively. The sequential pattern $< A, B, E >$ appears in the figure 4 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.

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

Table 4: Frequent Contiguous Sequential Patterns

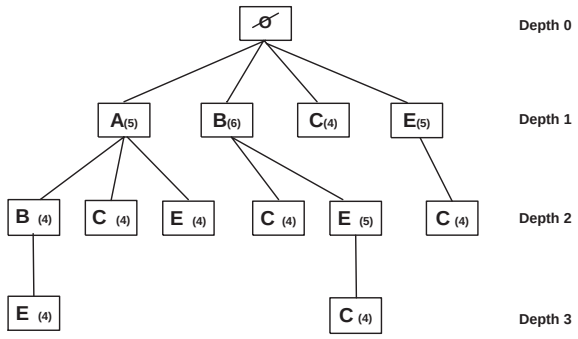


Figure 4: The Frequent Sequential Pattern Trie for example

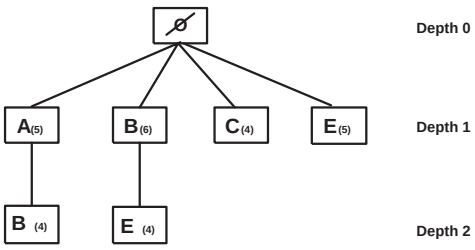


Figure 5: The Frequent Contiguous Sequential Pattern Trie for example

3.3 All-kth-Order Models

It should be noted that, 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. An alternative to reducing the support threshold in such cases would be to reduce the session window size. This latter choice may itself lead to some undesired effects since we may not be taking enough of the user's activity history into account. Generally, in the context of recommendation systems, using a larger window size over the active session can achieve better prediction accuracy. But, as in the case of higher support threshold, larger window sizes also lead to lower recommendation coverage.

In order to overcome this problem, we use *all-kth-order* method proposed in [15] in the context of *Markov chain models*. Markov models are especially suited for predictive modeling based on contiguous sequences of events. In Web usage analysis, they have been proposed as the underlying modeling machinery for Web prefetching applications or to minimize system latencies [6; 14; 15]. Such systems are designed to predict the *next* user action based on a user's previous surfing behavior. In general, Markov models generate state-to-state transition probabilities from Web navigational sequences and predict user's next access based on these transition probabilities. In this context, each state represents a subsequence of a current user's trail through the site. 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. In order to remedy the coverage and space complexity problems, [15] proposed All-kth-Order Markov models (for coverage improvement) and a new state reduction technique, called longest repeating subsequences (LRS) (for reducing model size). 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. This scheme can easily lead to even higher space complexity since it requires the representation of all possible states for each k.

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 [6], except that the support pruning is performed by the Apriori algorithm in the mining phase. Furthermore, in contrast to standard all-kth-order Markov models, our framework does not require additional storage since all the necessary information (for all values of k) is captured by Frequent Sequence Trie structure described above.

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.

4 Experimental Evaluation

The effectiveness of personalization must be measured in terms of both coverage and precision of the produced recommendations. Precision measures the degree to which the recommendation engine produces accurate recommendations. On the other hand, coverage measures the ability of the recommendation engine to produce all of the items that are likely to be visited by the user. Both of these measures are essential in evaluating the effectiveness of recommender systems. For example, in the e-commerce domain, low precision can easily lead to angry or frustrated users (who receive inaccurate recommendations) while low coverage will result in the site missing cross-sell or up-sell recommendations at critical junctures in user's navigation through the site.

We systematically evaluate the performance of recommendation engines using Association Rules (AR), Sequential Patterns (SP), and Contiguous Sequential Patterns (CSP), in terms of coverage and precision. Our goal is to determine the conditions (e.g., characteristics of a site) which contribute to one model performing better than others. In order to facilitate such an evaluation, we use real usage data from 3 different site each with its own characteristics. In general, our results indicate that the structure of a Web site characteristics, such as the underlying hyperlink structure, have a significant impact on the performance of the models.

4.1 Experimental Data Sets and Site Characteristics

For our experiments, we used the server logs from the 3 different Web sites each with its own structural and domain characteristics. The server logs used for the current experiments belong to of 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 associate with the Chicago Public Television and Radio (www.networkchicago.com).

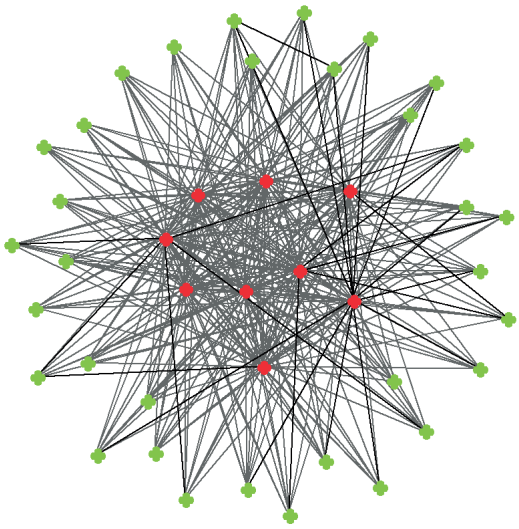


Figure 6: The Hyperlink Graph for the ACR Site

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 graph in Figure 6 depicts the underlying hyperlink structure for the ACR site. In particular, this graph illustrates the highly connected nature of the site.

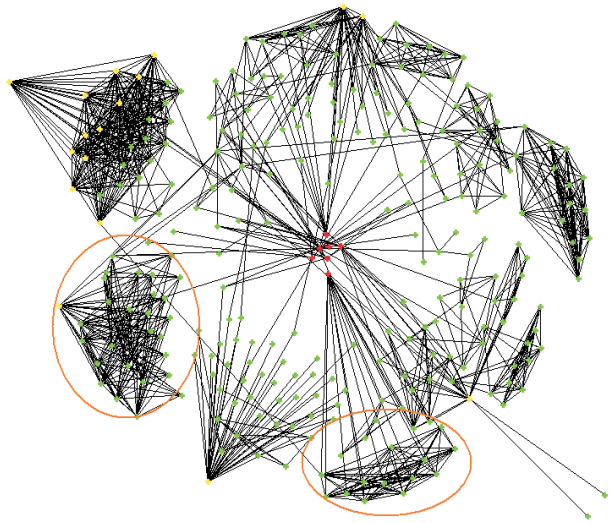


Figure 7: The Hyperlink Graph for the NC Site

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 student, 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. Figure 7 depicts the hyperlink graph for the NC site.

Our results, discussed in the next section, suggest that with a highly connected site structure, generally, AR and SP models, which capture less constrained navigational patterns, are better choices for personalization. On the other hand, In a sites (or portions of a site), involving deeper substructure 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.

4.2 Evaluation Methodology

We perform 10-fold cross-validation using each of the 3 data sets. In each of the 10 iterations, the data set divided into training (90%) and evaluation (10%) data 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.

Our evaluation methodology is as follows. 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 value n reflects the maximum allowable window size for the experiments (in our case 4). Given a window size $w \leq n$, we select a subset of the first n pageviews as the surrogate for a user's *active session window*. 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 $R(as_t, \tau)$. Note that $R(as_t, \tau)$ contains all pageviews whose recommendation score is at least τ (in particular, if $\tau = 0$, then $R(as_t, \tau) = P$, where P is the set of all pageviews).

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 (i.e., the proportion of relevant recommendations to the total number of recommendations). Coverage measures the ability of the recommendation engine to produce all of the pageviews that are likely to be visited by the user (i.e., the proportion of relevant recommendations to all relevant pageviews).

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. The results of these experiments are presented below.

4.3 Experimental Results

Figures 8, 9, and 10 depict the evaluation results for the ACR, CTI, and NC data sets. In each case, the performance of the recommendation frameworks based on association rules (AR), sequential patterns (SP), and contiguous sequential patterns (CSP) is measured in terms of precision and coverage of recommendations. In general, the sequential models (particularly CSP) often produce unacceptable coverage when used for recommendations. Thus, we have present only the all-kth-order models for each framework, as discussed in Section 3.3

As can be observed from these results, in the case of sites characterized by longer hyperlink paths and deeper substructure (e.g., CTI and NC sites), the sequential models such as CSP and SP tend to produce more accurate recommendations overall. The coverage of the CSP model, in general, is quote low. This may suggest that contiguous sequential patterns are better suited for applications such as Web pre-fetching where the goal is to predict a user's immediate *next* actions rather than producing a broader set of recommendations (as in Web personalization). On the other hand, the SP model in the case on the CTI and NC sites, produce more precise recommendations than the AR model while maintaining coverage levels in par with the AR model.

In contrast, as observed in Figure 8, in the case of the highly-connected ACR site, the AR model produces recommendations at a similar precision levels as the SP model, while achieving much higher coverage levels. These results indicate that in highly connected sites the use of sequential patterns for personalization may result in overly narrow recommendations sets, thus leading to lower coverage, while not providing any advantage in terms of higher precision.

To further verify the above observations, we selected two highly connected components of the NC site graph (the circled portions of the graph depicted in Figure 7), and filtered the NC data set with the resulting user transaction only containing pageviews from the selected

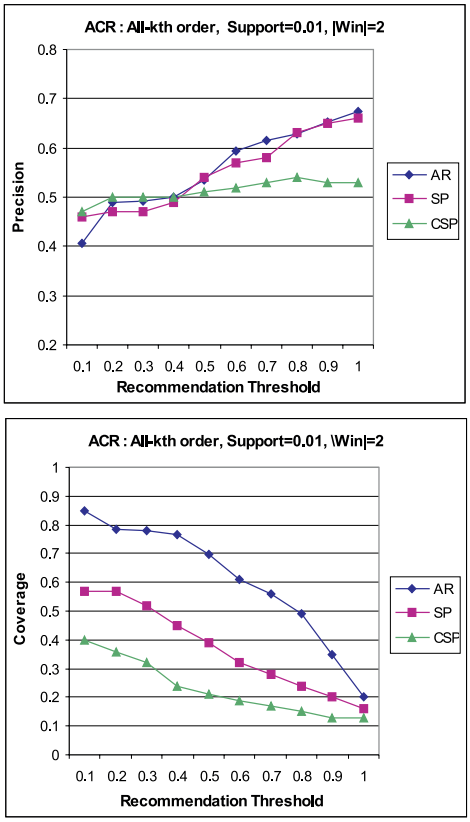


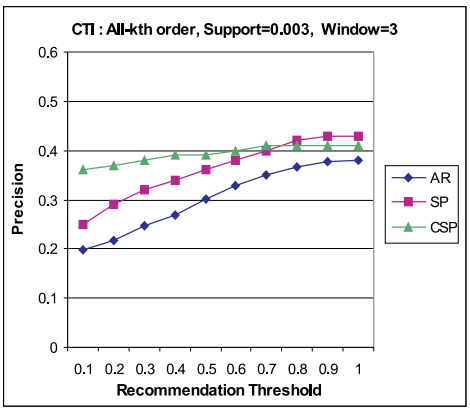
Figure 8: Performance of recommendation frameworks based on Association Rules and Sequential Patterns in the ACR data set.

components. We then performed our evaluation on this filtered data set. The results are depicted in Figure 11.

These results show a strong resemblance to those obtained for the ACR site. In this case, also, the AR model provides much better recommendation coverage, while producing precision levels in par with, or better than, the SP and CSP models. Indeed, in both cases, the SP and AR models provide significantly higher precision at higher recommendation thresholds, suggesting that for highly connected sites (or portions of sites), less constrained patterns are generally better suited for use in personalization tasks.

5 Conclusions and Future Work

The effectiveness of a personalization framework must be evaluated in terms on two important factors, namely the precision and the coverage of the generated recommendations. In general using more fine-grained information about users' navigational histories (such as the ordering information among pageviews in user transactions) as part of pattern discovery does not necessarily translate to more effective personalization. A number of factors, such as the structural characteristics of a site, have a significant impact on the performance of sequential recommendation models (based on sequential or contiguous



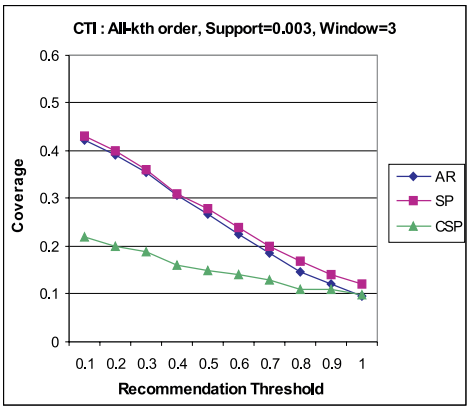
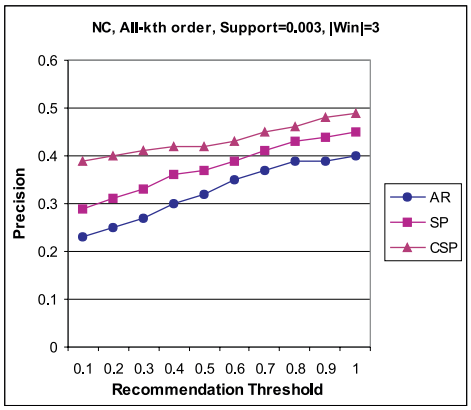


Figure 9: Performance of recommendation frameworks based on Association Rules and Sequential Patterns in the CTI data set.



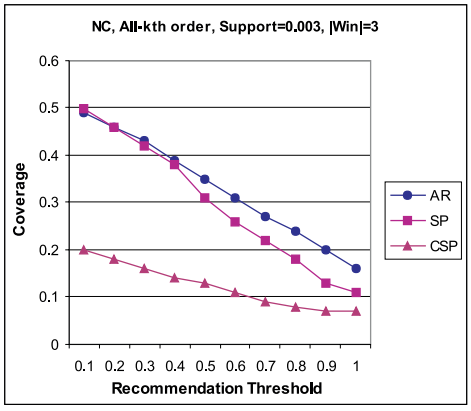


Figure 10: Performance of recommendation frameworks based on Association Rules and Sequential Patterns in the NC data set.

sequential patterns) versus those of non-sequential models (based on frequent itemsets or association rules).

In this paper We have conducted a detailed comparative evaluation of sequential and non-sequential patterns in terms of their effectiveness and suitability for personalization tasks. In particular, we have focused on the impact of a site's topology and its degree of connectivity on the precision and coverage of recommendations. Our results show that less constrained patterns, such as frequent itemsets or general sequential patterns are better suited for personalization in sites with a higher degree of connectivity and shorter navigational depth, while more restrictive patterns, such as contiguous sequential patterns may be suitable for sites with deeper navigational depths (including those resulting from dynamic pages generated through sequences of user interactions).

An interesting area of further study is the development of a hybrid recommendation framework that can automatically switch between recommendation models during user navigation based on the degree of connectivity of at different "neighborhoods" within the site.

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, Alberta, Canada, July 2002.

[5] R. Cooley, B. Mobasher, and J. Srivastava. Data preparation for mining world wide web browsing patterns. *Journal of Knowledge and Information Systems*, 1(1), 1999.

[6] M. Deshpande and G. Karypis. Selective markov models for predicting web-page accesses. In *Proceed-*

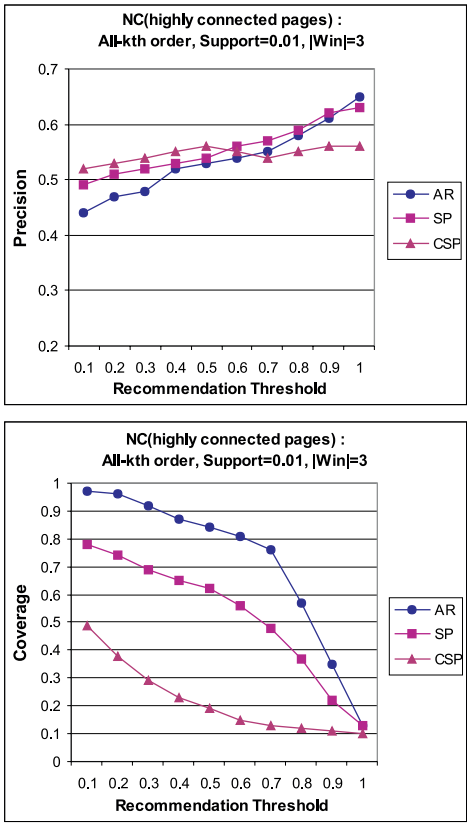


Figure 11: Performance of recommendation frameworks on a highly connected subgraphs of the NC site.

ings of the First International SIAM Conference on Data Mining, Chicago, April 2001.

[7] 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.

[8] J. Herlocker, J. Konstan, A. Borchers, and J. Riedl. An algorithmic framework for performing collaborative filtering. In *Proceedings of the 22nd ACM Conference on Research and Development in Information Retrieval (SIGIR'99)*, Berkeley, CA, August 1999.

[9] 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.

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

[11] B. Mobasher, H. Dai, T. Luo, and M. Nakagawa. Effective personalization based on association rule discovery from web usage data. In *Proceedings of the 3rd ACM Workshop on Web Information and Data Management (WIDM01)*, Atlanta, Georgia, November 2001.

[12] B. Mobasher, H. Dai, and M. Nakagawa T. Luo. Discovery and evaluation of aggregate usage profiles for web personalization. *Data Mining and Knowledge Discovery*, 6:61–82, 2002.

[13] M. O'Conner and J. Herlocker. Clustering items for collaborative filtering. In *Proceedings of the ACM SIGIR Workshop on Recommender Systems*, Berkeley, CA, August 1999.

[14] T. Palpanas and A. Mendelzon. Web prefetching using partial match prediction. In *Proceedings of the 4th International Web Caching Workshop (WCW99)*, San Diego, CA, March 1999.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] 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.