Using Sequential and Non-Sequential Patterns in Predictive Web Usage Mining Tasks

Bamshad Mobasher, Honghua Dai, Tao Luo, Miki Nakagawa
{mobasher,hdai,tluo,miki}@cs.depaul.edu
School of Computer Science, Telecommunication, and Information Systems
DePaul University, Chicago, Illinois, USA

Abstract

We describe an efficient framework for Web personalization based on sequential and non-sequential pattern discovery from usage data. Our experimental results performed on real usage data indicate that more restrictive patterns, such as contiguous sequential patterns (e.g., frequent navigational paths) are more suitable for predictive tasks, such as Web prefetching, which involve predicting which item is accessed next by a user), while less constrained patterns, such as frequent itemsets or general sequential patterns are more effective alternatives in the context of Web personalization and recommender systems.

1. Introduction

Web usage mining techniques [9], that rely on offline pattern discovery from user transactions, can be used to solve scalability problems associated with personalization systems based on standard collaborative filtering. In addition, user models discovered through data mining, can capture more fine-grained information, such as the inherent ordering among accessed pages, than standard techniques afford.

However, using more fine-grained information about users' navigational histories as part of pattern discovery does not necessarily translate to more effective personalization. Furthermore, techniques that may prove effective for predictive tasks such as prefetching, may not necessarily be appropriate in the context of personalization.

In this paper we present a scalable framework for Web personalization based on both sequential and non-sequential pattern mining from clickstream data. Our framework includes efficient data structures for storing frequent itemsets or sequential patterns combined with algorithms which allow for effective real-time generation of recommendations.

We have conducted a detailed comparative evaluation, based on real usage data, of both sequential and non-sequential patterns in terms of their effectiveness and suitability for personalization tasks. We distinguish between two different evaluation methodologies, one suited for evaluation of personalization effectiveness, and the other designed for evaluating the performance of predictive tasks such as Web prefetching (which involve predicting which item the user will access next during his/her navigation).

Our empirical results show that more restrictive patterns, such as contiguous sequential patterns (e.g., frequent navigational paths) are more suitable for predictive tasks such as Web prefetching (which involve predicting which item the user will access next during his/her navigation). On the other hand, less constrained patterns, such as frequent itemsets or general sequential patterns are more effective alternatives in the context of Web personalization.

2. Preprocessing and Pattern Discovery

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 [6, 7], 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. The recommendation engine considers the active user session in conjunction with the discovered patterns to provide personalized content.

Web usage preprocessing [5] ultimately result in 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:

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

0-7695-1754-4/02 $17.00 © 2002 IEEE

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 (usually binary weights are used in the context of association rule and sequential pattern discovery).

We focus on three data mining techniques: Association Rule mining (AR) [1, 2], Sequential Pattern (SP) [3], 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 [10]. 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.

3. Personalization With Sequential and Non-Sequential Patterns

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. We use a fixed-size sliding window over the current active session to capture the current user's history depth. 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*.

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 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.

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. A simple trie structure is used to store both the sequential and contiguous sequential patterns discovered during the pattern discovery phase.

Depending on the specified support threshold and window size, 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 [8] 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.

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. The notion of all-kth-order models can 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.

4. Experimental Evaluation

We conjecture that more restrictive patterns, e.g., CSP, may be better suited for predictive applications such as Web prefetching which involve predicting a user's *next* immediate access to a page. This is in contrast to personalization tasks which involve prediction a (broader) set of pages to be recommended to the user based on his/her previous access patterns. Thus, we propose two different evaluation methodologies, which here we shall call *NEXT* and *ALL*, respectively. The NEXT method evaluates recommendation effectiveness by comparing the system's predictions with a user's immediate *next* action, while the ALL method compares the predictions against *all* of the user's remaining actions (accesses to pageviews) in the duration of a session.

The experiments were performed on real Web usage data from 3 different commercial and non-commercial sites. The results shown below represent selected experiments from only one of these data set; the full set of results and experiments are included in the full paper and available upon request.

For all experiments, we performed 10-fold cross-validation. In each iteration, each transaction t in the evaluation set was divided into two parts. The first n pageviews in t were used for generating recommendations, whereas, the remaining portion of t (*target set*) was used to evaluate the generated recommendations. Given a window size $w \leq n$, we select a subset (or a subsequence in the case of SP or CSP) of the first n pageviews as the surrogate for a user's *active session window*, denoted by as_t.

Both All and NEXT evaluation methods take as_t and a recommendation threshold τ as inputs and produce a set of pageviews as recommendations. The recommendation set contains all pageviews whose recommendation score is at least τ. The NEXT method compares the generated recommendations with the immediate next pageview in the remaining portion of the transaction t. On the other hand, the ALL method compares the recommendation set to all of the pageviews in the remaining portion of t. The *precision* measure represents the ratio of matches between the recommendation set and the target set to the size of recommendation set. The *coverage* measure represents the ratio of matches to the size of the target set. 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 of the measures in both evaluation methods.

Figure 1 depicts the results for the all-kth-order versions of the three recommendation methods for both data sets. In these comparisons, we also included the precision and coverage of the standard k-Nearest-Neighbor (kNN) technique for standard collaborative filtering. The value of k was chosen based sensitivity analysis for the best performance in terms of coverage and precision.

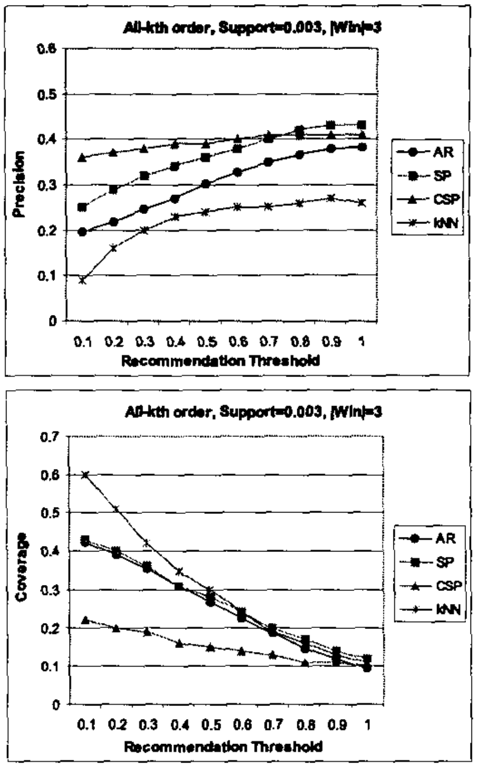

Figure 1: Recommendation Effectiveness Using Varying-Sized User Histories Based on the ALL Method

The results show that, in this case, SP will have similar (or better) performance than CSP in terms of both coverage and precision. This phenomenon is likely due to the fact that varying window sizes has a more dramatic impact on the precision of CSP than on SP or AR. The AR models also performs well in the context of personalization. In general, the precision of AR model is lower than SP, but it does provide better overall coverage. The comparison to kNN shows that all of the techniques presented in this paper, outperform kNN in terms of precision. In general, kNN provides good coverage (usually in par with the AR model), but the difference in coverage is diminished if we insist on higher recommendation thresholds (and thus more accurate recommendations).

We also compared the precision and coverage of AR, SP, and CSP based on the NEXT evaluation method. To provide a better basis for comparison of these results to those based on the ALL method, we use the same support threshold and window size parameters as those used in Figure 1. The results are shown in Figure 2.

The results show that, in this context, the CSP model provides much higher precision levels that both SP and AR,

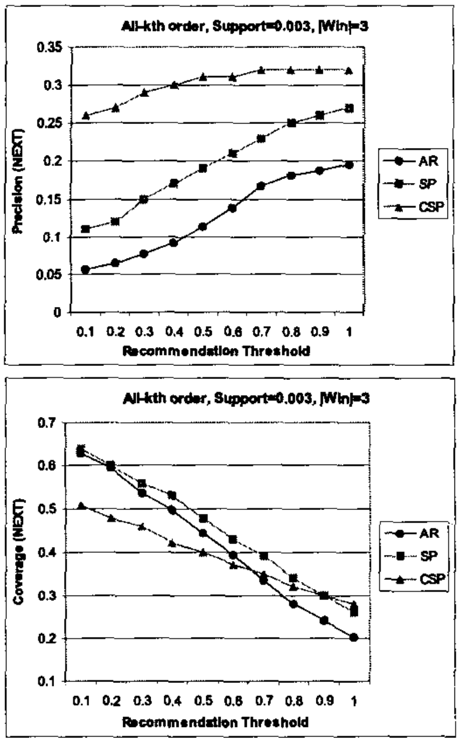

Figure 2: Comparison of Precision and Coverage Based on the NEXT Evaluation Method

while achieving coverage levels that are in par with the SP model. Indeed, at high recommendation thresholds, the coverage of CSP is similar to better than that of the SP model. The precision levels of the AR model are too low to make it a reasonable candidate for this type of application.

5. Discussion and Conclusions

Our overall conclusion based on the aforementioned results is that the SP and the AR models, generally provide the best choices for personalization applications. The CSP model can do better in terms of precision, but the coverage levels, in general, may be too low when the goal is to generate as many good recommendations as possible. On the other hand, when dealing with applications such as Web prefetching in which the primary goal is to predict the user's immediate next actions (rather than providing a broader set of recomendations), the CSP model provides the best choice. This is particularly true in sites with many dynamically generated pages (such as the one used in these experiments), where often a contiguous navigational path represents a semantically meaningful sequence of user actions each depending on the previous actions.

References

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

[2] R. Agrawal and R. Srikant. Fast Algorithms for Mining Association Rules. Proc. *20th Int. Conf. Very Large Data Bases, VLDB*, 1994.

[3] R. Agrawal and R. Srikant. Mining Sequential Patterns. *In Proc. of the 11th Int'l Conference on Data Engineering*, Taipei, Taiwan, March 1995.

[4] B. Berendt, B. Mobasher, M. Spiliopoulou, J. Wiltshire. Measuring the accuracy of sessionizers for Web usage analysis. In *Proceedings of the Web Mining Workshop* at the *First SIAM International Conference on Data Mining, SDM 2001*, Chicago, April 2001.

[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] B. Mobasher, R. Cooley and J. Srivastava. Automatic personalization based on Web usage mining. In *Communications of the ACM*, (43) 8, August 2000.

[7] B. Mobasher, H. Dai, T. Luo, 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, November 2001.

[8] J. Pitkow and P. Pirolli. Mining Longest Repeating Subsequences to Predict WWW Surfing. *Proceedings of the 1999 USENIX Annual Technical Conference*, 1999.

[9] J. Srivastava, R. Cooley, M. Deshpande, P-T. Tan. Web usage mining: discovery and applications of usage patterns from Web data. *SIGKDD Explorations*, (1) 2, 2000.

[10] M. Spiliopoulou and L.C. Faulstich. WUM: A Tool for Web Utilization Analysis. *In Proc. of EDBT Workshop WebDB'98*, Valencia, Spain, Mar. 1998.