Understanding Web Usage for Dynamic Web-Site Adaptation: A Case Study

Nan Niu, Eleni Stroulia and Mohammad El-Ramly
Department of Computing Science, University of Alberta
221 Athabasca Hall, Edmonton, Alberta Canada T6G 2E8
{nan, stroulia, mramly}@cs.ualberta.ca

Abstract

Every day, new information, products and services are being offered by providers on the World Wide Web. At the same time, the number of consumers and the diversity of their interests increase. As a result, providers are seeking ways to infer the customers' interests and to adapt their web sites to make the content of interest more easily accessible. Pattern mining is a promising approach in support of this goal. Assuming that past navigation behavior is an indicator of the users' interests, then, the records of this behavior, kept in the form of the web-server logs, can be mined to infer what the users are interested in. On that basis, recommendations can be dynamically generated, to help new web-site visitors find the information of interest faster. In this paper, we discuss our experience with pattern mining for dynamic web-site adaptation. Our particular approach is tailored to "focused" web sites that offer information on a well-defined subject, such as, for example, the web site of an undergraduate course. Visitors of such focused sites exhibit similar types of navigation behavior, corresponding to the services offered by the web site; therefore, page recommendation based on usage-pattern mining can be quite effective.

Keywords

Sequential pattern mining, Web usage mining, Dynamic web-site adaptation, Web page recommendation.

1. Introduction and Motivation

The World Wide Web contains an enormous amount of information in the form of a rather unstructured collection of hyperlinked documents, which increasingly makes finding relevant documents with useful information a challenge [9]. At any point in time, each web site is visited by many users, with different goals, who are interested in different information content or types of presentations [7]. Even the same user may visit the same web site for different purposes at different times. Furthermore, as the content published on the web site evolves, the users' interests also evolve, and so do their navigations of the site and the way they access its content. As a result, no single organization of the content of a web site can be satisfactory for all these varied needs [13]. Therefore, the problem of dynamically adapting the web site to better fit the individual visitor's preferences has become a great challenge for content providers, and consequently, a very interesting research problem.

Web-site designers want to increase the number of visitors and the time that these visitors spend on their web site. To accomplish that, they have to supply attractive content. And to make their content attractive, web-site designers and content providers need to know what their potential visitors want, in order to organize their content according to their visitors' needs, and, if possible, according to individual preferences.

Traditional methods for collecting data on software users, such as questionnaires and surveys, for example, are not applicable to understanding web-site users. The size of the potential user population is too large and varied and people usually visit the web site before they actually become "regular" users. The alternative is to collect data from these visits and to analyze them in order to understand what the visitors expect from the web site, so as to adapt the web site to deliver the desired content in a simpler more easily accessible manner. The most common type of web-site adaptation is enabling "cross selling", i.e., adapting pages describing one product to include links to other related products that previous customers may have bought together with the current product. Several different technologies may be used to support this (and other related) feature(s) – we review the state-of-the-art in industrial practices and research in Section 2.

In this paper, we present our initial work and experience with web-usage pattern mining in support of dynamic page recommendation. Our approach is tailored to "focused" web sites that offer well-defined types of information to customers who visit the site to access this information. For our experiments, we used the web site of an undergraduate course at the Department of Computing Science, at the University of Alberta. The basic intuition underlying our method is that users of such a focused site share a common purpose in accessing the site in question, which defines, to a great extent, their navigation and access behavior. Furthermore, the nature of the web-site content evolves in a regular way. For example, in our experiment, the users of the course web site are the course students, who are interested in accessing the information posted to the web site by the instructor team. The navigation behavior of the students is not simply browsing or exploring the web site; it is defined by their purposes, such as to retrieve lecture notes or to view their marks. At the same time, the content of the course web site is also updated in a regular manner, dictated by the timetable of delivering an undergraduate course during an academic term.

We believe that the pattern-mining approach is especially promising in the case of such focused web sites. Strong regularities in the structure and the evolution process of the web site and highly common visitor purposes are bound to result in consistent navigation behavior. Such consistent behavior should consequently result in frequently occurring patterns, corresponding to the purposes of the visitors. If this is the case, these patterns can also be effectively used to infer the purpose of future visits, to generate, at run time, recommendations so that information of interest to many early visitors could become more easily accessible to subsequent visitors. The second implication of the fact that a web site is focused is that personalization becomes less important. In the case of exploratory browsing, individual user differences are bound to have a more pronounced effect in the visitor's navigation behavior; when the visitors share well-defined common purposes, then their navigation behavior should be similar. Thus, a focused web site may become adaptive by monitoring the page-access behavior of its visitors, inferring their purposes as frequently followed patterns, and then using the extracted patterns to dynamically adapt its content for subsequent visitors that exhibit similar behavior.

Our approach is novel in several ways. First, we use a new pattern-mining algorithm, IPM [4, 5], for efficiently extracting approximate behavior patterns so that slight navigation variations can be ignored when extracting frequently occurring patterns. Second, the usage patterns, on the basis of which the web-site pages are adapted to include recommendations, are updated frequently and are sensitive both to changes in the web-site material and in the behavior of its visitors. Finally, our approach is fairly lightweight; it assumes that, because the web site is focused, there is a fairly homogeneous visitor type accessing it and it does not attempt to distinguish among different visitor groups.

The rest of this paper is organized as follows. In Section 2, we review the state-of-the-art in web usage monitoring and mining and web-site adaptation. In Section 3, we describe in detail our approach. In Section 4, we present our experimental results and we reflect on the effectiveness and efficiency of our method. Finally, we conclude, in Section 5, with some lessons we have learned and our plans for future work.

2. Related Practices and Research

To better understand the demographic profile of its users, some web sites require their visitors to register themselves. The registration process usually involves providing information about their person, their interests and needs. The collected profiles of the registered users are then clustered to identify demographic constituencies in the visitor population; new users are then classified according to their demographics and their behavior is predicted to be similar to the behavior of the other members of their group. Although explicit user registration is effective in collecting rich information on the preferences of users, it has non-trivial disadvantages. The registration procedure is often time-consuming, and although it is similar across web sites, it is rarely shared; as a result, users find themselves providing the same information repeatedly. Furthermore, the information collected at the time a user is registered is assumed to be static and is not updated, which is clearly a seriously flawed assumption.

Explicit user feedback is another, often employed, solution to the problem of understanding web-site visitors' preferences. Many web sites gather feedback on the quality of the products they offer and the design of the site itself, by explicitly requiring their users to fill out questionnaire forms. However, visitors feel often annoyed and tend to ignore the questions or provide false information. Furthermore, the feedback collected through a general questionnaire is often too "shallow" to help the web-site designers to adapt their site to improve the navigation experiences of its users.

2.1 Web-Usage Mining

Given the disadvantages of these explicit information-collection approaches, recent research has focused on deploying data-mining methods for understanding and predicting web-site visitor behavior. Research in Web mining spans three areas: Web-content mining, Web-structure mining and Web-usage mining. Web-content mining refers to the mining of structured content from unstructured web pages. Applications include customer support, automated e-mail routing and reply, and knowledge management, such as document clustering, content categorization, and keyword extraction and associations [11]. Web-structure mining focuses on analyzing the link structure of the Web to identify interesting relationships and patterns describing the connectivity of documents in the Web [6]. Such relationships are then used to retrieve relevant documents in response to user requests. Finally, Web-usage mining focuses on analyzing the visitor's navigation of a web site to assess problems with its organization, such as long traversal paths for example, or to identify paths that lead to sales and cross-sales [11].

The work we discuss in this paper focuses solely on Web-usage mining. Web-usage mining aims at the development of techniques and tools to study the navigation behavior and access patterns of web-site visitors. These methods examine the data related to the usage of the pages of a web site, such as IP addresses, page references, and the date and time of access [3], to extract frequent access patterns. Access data can be obtained from the logs kept by the web-site server. An access pattern is a recurring sequential pattern among the entries in the web-server log [12]. For example, if various users repeatedly access the same series of pages, a corresponding series of log entries will appear in the log file, and this series can be considered as an access pattern.

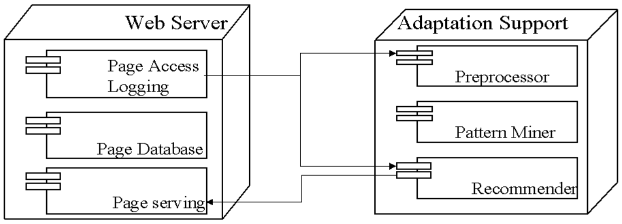

Figure 1: Data Flow Diagram of Web-Usage Mining.

Figure 1 depicts the architecture of an adaptive web site, based on web-usage mining. Such a web site, in addition to the regular database-supported web server, includes an additional component aimed at providing support for its adaptation capability. The "Adaptation Support" component consists of three sub-components: a preprocessor and a pattern miner to analyze the logs collected by the web server, and a recommender, to dynamically adapt the pages served by the web server with recommendations as additional links on these pages.

The preprocessing component translates the raw web-server logs into a set of "visitor sessions", which is the input necessary to the pattern-discovery process. The pattern-discovery component employs data-mining algorithms to discover regularities in the visitor navigation behavior, as captured in these sessions. This is the most crucial step of the whole adaptation-support component. The effectiveness of the web-site adaptation depends on the quality of the discovered patterns, which, in turns, depends on a predefined criterion of interestingness. Finally, the recommender component monitors the navigation behavior of the site visitors, at run time. When this behavior matches a prefix of some of the discovered patterns, the recommendation component informs the page server to dynamically add appropriate links on the page to recommend the subsequent pages of the pattern.

2.2 Sequential Pattern Mining

Sequential pattern functions, which are also known as temporal pattern functions, analyze a collection of items over a period of time. Sequential pattern mining (SPM) aims at extracting inter-session patterns such that the presence of a set of items is followed by another item in a time-ordered set of sessions or episodes [3]. The objective is, given a set of items, with each item associated with its own timeline of events, to find rules that predict strong sequential dependencies among different events. For example, when the identity of a customer who made a purchase is known, the collection of items bought can be analyzed. A sequential pattern function analyzes such collections of related items and detects frequently occurring patterns of products bought over time. By using this approach, marketers can predict future purchase patterns that may be helpful in placing advertisements aimed at certain user groups.

Similarly, SPM can be used to discover time ordered sequences of URLs visited by web-site users, in order to predict future user behavior and offer the predictions as recommendations. Sequential patterns could reveal temporal relationships such as: *"70% of web users who visited /assignment1.html and then /assignment1_hints.html, also accessed afterwards in the same session /lecturenots.html"*. Formally, a rule R generated from sequential patterns is a tuple $R = <<a_{1,}\ a_2\ _{\dots}\ a_i>, <c_{1,}\ c_2\ _{\dots}\ cj>>$ where, for $1 \leq k \leq i$, a_k stands for a set of URLs in the antecedent part and, for $1 \leq k \leq j$, c_k stands for a set of URLs in the consequent part. Both the antecedent and consequent parts take time constraints into consideration. Other types of temporal analysis that can be performed on sequential patterns include trend analysis, change point detection, or similarity analysis [3].

Another line of related research is association analysis, which discovers association rules showing attribute-value conditions that occur frequently together in a given set of data. Association analysis is widely used for market basket or transaction data analysis. These methods aim at finding intra-transaction patterns, whereas the problem of finding sequential patterns concerns the discovery of inter-transaction patterns. A pattern in the first problem consists of an unordered set of items whereas a pattern in the latter case is an ordered list of sets of items [1]. A typical application of association-rule mining in web-site evolution can help designers restructure their web sites. This kind of restructuring does not take temporal properties into consideration.

Sequential pattern mining methods have been applied to problems in a variety of domains. In bio-informatics, SPM is applied to discover frequently occurring subsequences of amino acids that may uniquely define a specific aspect of the biological function of a cell [2]. In the area of systems monitoring, such as network monitoring, SPM is used to predict faults and abnormal performance patterns [14].

IPM [4, 5] is a sequential pattern-mining algorithm, designed to discover patterns in recorded traces of interaction between a legacy software system and its users, in order to discover models of frequent user tasks for legacy software reengineering purposes. In our work, we have adopted this algorithm to discover usage patterns in the navigation behaviors of web-site users. Most previous work in web-site recommendation based on mined patterns of usage behavior has relied on the Apriori algorithm [1]. Apriori addresses the similar problem of discovering that "if a user visits two documents in a site, then he will most likely also visit another document within a time period". Instead, IPM focuses on understanding the likely navigational sequences of the web-site visitors within a single session.

Analyzing web-server logs for exhibiting useful access patterns has already been the subject of extensive research. A flexible architecture for Web mining, called WEBMINER, and several data mining function are proposed in [3]. In [8], the authors present an architectural framework for Web-usage mining; they show that association rules and sequential patterns extracted from web-server logs enable prediction of users visit patterns, as well as a dynamic hypertext organization.

The web-usage mining system proposed in [10] is designed to support automatic personalization, i.e. taking into account the user's taste to provide automatically the right information. Traditional web mining patterns are of interest in those projects, such as: association rules, clustering, and so forth.

In [13], the author proposes an intelligent agent that supports the user in navigating a web site. The employed LCSA algorithm analyzes overall usage, web page content, web site structure and current user's actions. However, LCSA does not allow for accesses to spurious pages of the web site, which is possible with the approximate-pattern discovery method of IPM. The application developed in [13] mainly discusses patterns extracted based on association rules between URLs and web users, without considering the evolution of these relationships over time. In our case study, we have explicitly studied the evolution of the patterns' relevance and the corresponding effectives of the pattern-based adaptation in time.

3. Dynamic, Run-time Page Recommendation

Our approach to dynamic, web-site adaptation follows the general process discussed in Section 2, consisting of the preprocessing, pattern mining, and pattern recommendation steps.

In the preprocessing phase, the raw web-server log file is cleaned, i.e., all image, audio and video files' accesses are eliminated, and the distinct sessions it contains are identified. In many cases, preprocessing also involves the identification of the individual users accessing the web site. This is necessary when the adaptation process assumes the existence of different types of visitors; then, identifying the individual users is necessary for inferring different user profiles that can be used for profile-based adaptation at run-time. Our approach is tailored to "focused" web sites, for which, we assume, the navigation behavior of all users is dictated by the purposes supported by the web site and does not vary much according to individual user preferences. This assumption results in the simplification of the preprocessing phase, thus making the overall approach simpler.

Next, for the pattern-mining phase, the IPM [4, 5] algorithm is applied to the retrieved sessions, to identify frequently occurring navigational patterns. IPM is designed to recognize approximate patterns, i.e., patterns that may include spurious steps in addition to the essential pattern steps. Thus it is robust to noisy data, which is very likely to be the case with web-site navigation data.

Finally, the extracted patterns are used to generate page recommendations at run time. In our adaptive web-site prototype, currently under implementation, we plan to employ a simple user-authentication process that will simplify the run-time user-navigation monitoring and the recommendation of more relevant pages, when the extracted patterns become applicable.

3.1 Web-site Usage-trace Collection and Preprocessing

The preprocessing phase consists of operations that process the available sources of information (HTTP server and auxiliary ones) and lead to the creation of an appropriately formatted data set to be used for knowledge discovery with the IPM algorithm. On an average size web server, access log files easily reach tens of megabytes per day, which causes the analysis process to be really slow and inefficient without an initial cleaning task.

Like most log analysis tools, our method employs a cleaning step to perform the following tasks. First, requests for URLs containing graphics, sounds, or video files are filtered out. Each time a browser accesses an HTML document, the images included are also requested, causing a corresponding number of accesses to be recorded in the log file of the server (unless images automatic load option of the client is switched off). These implicit accesses do not provide any information regarding the user's interests and can be eliminated from the log. They are identified based on the suffix of the URL name in the log file. For instance, all log entries with filename suffixes such as "gif", "jpg", "jpeg", "wav", "au", "ai", and "map" are removed following the predefined cleaning rules. The set of suffixes could be adjusted as needed for particular web sites, by appropriately configuring the log-cleaning criteria. Additional accesses may be filtered out, such as entries with script files such as "counter.cgi", records with particular code such as "HTTP status code equals to 404", which means resource is not found on the server, and accesses performed by agents such as crawlers, robots, or spiders.

Next, the names of the accessed URLs in the trace are standardized. A trailing slash may optionally be present in a URL name. In addition, the "www" in front of a URL is optional. This means that requests for "www.cs.ualberta.ca/~stroulia", "http://www.cs.ualberta.ca/~stroulia/", and "http://cs.ualberta.ca/~stroulia" all refer to the same file, and they are all transformed to "http://www.cs.ualberta.ca/~stroulia".

The next preprocessing task is session identification. A server session is identified as a set of subsequent accesses to the web site by a client with the same IP, the same type of browser on the same operating system, within a particular time length. The timeout method assumes that the user is starting a new session if the time between page requests exceeds a certain threshold. If requests from a user appear over long periods of time in the web-server log file, it is very likely that the user has visited the web site more than once. Usually a 30 minutes timeout between sequential requests from the same user is taken in order to close a session [3].

Finally, the identified sessions are rewritten in run-length encoding, to satisfy the preconditions of the IPM algorithm. The sessions, originally represented as sequences of standardized URLs, may contain repetitions, for example, from "refresh" requests. These repetitions may result in missing useful patterns, since they will differentiate otherwise similar navigation behavior. In run-length encoding, immediate repetitions of the same URL are substituted with the count of the repetitions followed by the URL being repeated.

3.2 Web-site Usage-Pattern Discovery

The IPM algorithm discovers approximate patterns with insertion errors. It does not assume that the patterns of interest appear always the same; instead it allows for a maximum number of extra web pages to be included in the recorded pattern instances. Allowing insertion errors enables IPM to tolerate minor differences in the users' navigation of the web site; navigation segments with such noise will still be discovered as instances of the original pattern.

The input data to the algorithm are sequences of IDs drawn from an alphabet. In our case, the alphabet consists of the URLs accessed and recorded in the web-server logs after preprocessing. Additionally, the algorithm needs its user to input a pattern interestingness criterion. This criterion depends on five user-configurable parameters and defines what patterns to look for in the input traces. The parameters are as follows:

1. the minimum and maximum pattern length;
2. the minimum number of occurrences (support) of a pattern to be considered interesting;
3. the maximum number of insertion errors allowed in the instances of the discovered patterns, i.e. the number of spurious URLs that IPM should tolerate in these pattern instances; and
4. the minimum score of a pattern.

The scoring function used, given a pattern *p*, is *score* (*p*) = $\log_2 |p| * \log_2 support(p) * density(p)$, where |*p*| is the length of *p*, *support*(*p*) is the support of *p* and *density*(*p*) is the ratio of |*p*| to the average length of the instances of *p*. Since these instances may include some noise, they might be longer than their pattern. For example, {2,4,3,4}, {2,4,3,2,4} and {2,3,4} are the available instances of a pattern, *p*1 = {2,3,4} with at most 2 insertions allowed. Hence, *density*(*p*1) = 0.75.

After preprocessing the web-server logs, URLs are given unique integer IDs to suite the input format needed for IPM. IPM retrieves all maximal patterns that meet the user criterion. A maximal pattern is a pattern that is not a sub-pattern of any other pattern with the same support.

IPM adopts a fairly standard search strategy in data mining literature. This strategy is to discover short or less ambiguous patterns using exhaustive search, possibly with pruning. Then the patterns that have enough support are extended to form longer or more ambiguous patterns. This process continues until no more patterns can be discovered. For a detailed description of the IPM algorithm and its variant, please refer to [4, 5].

3.3 Pattern-based Recommendation

The IPM algorithm extracts a set of sequential navigation patterns from the preprocessed web-server logs. Our method uses these patterns at run time, to generate recommendations for pages that new users of the web site may want to visit. To be able to do so, user session tracking and relevant pattern selection are the two necessary subtasks.

The HTTP protocol is, by design, stateless: it does not provide any support for establishing long-term connections between the web-site server and the user. Therefore, in order to track the user's navigation path in the web site, it is necessary to build an additional infrastructure. In our prototype adaptive web site, we plan to adopt the technique of dynamic page rewriting with hidden fields. When the user first submits a request to the web site, the server returns the requested page rewritten to include a hidden field with a session-specific ID. Each subsequent request of the user to the server will supply this ID to the server, thus enabling the server to maintain the user's navigation history. This session-tracking method does not require any information on the client side and can therefore be always employed, independently of any user-defined browser settings.

Since, at any point in time, the web server knows the recent navigation behavior of the user, it can examine it to see whether this behavior is the prefix of any of the collected patterns. If yes, then the suffixes of the relevant patterns may be offered as recommendations for subsequent navigation. The web-site recommendation generation is based on pattern selection. There might be several sequential patterns that the current visiting path satisfies. Among these potential useful patterns, we choose the patterns with highest scores, as defined by the IPM algorithm, as the basis for recommending subsequent navigation. Page rewriting is used to dynamically adapt the pages requested by the web-site users with the recommendations on new potential places to visit.

When a pattern-based generated link is added to a web page, the server increments the counter of how often the pattern in question has been instantiated; and when the user adopts the recommendation of a pattern, by invoking the corresponding link, the server increments that pattern's counter of how often it has been followed. By calculating the ratio of how many times a pattern has been followed to the number of times that the pattern has been generated, we can evaluate whether the pattern is effective in recommendation generation or not.

Consider, as a very simple example, a case where patterns $p1 = \{1,2,3,4\}$ and $p2 = \{1,2,5,6\}$ were discovered in recent server logs, and a new web-site user has visited pages 1, 2 (page 2 is the current page shown on the user's browser). If $score(p1)$ is greater than $score(p2)$, then $p1$ will be chosen to generate the runtime recommendation. The URLs corresponding to pages 3 and 4, will be included as hyperlinks at the top of page 2, offering a shortcut to page 4. Hence, if one of these recommendations is followed, it saves the user's time and eliminates some unnecessary navigation.

4. Experimental Evaluation

We have evaluated our approach with an experiment on a real web site (http://ugweb.cs.ualberta.ca/~c301) of an undergraduate course at the Computing Science Department, University of Alberta.

At that time, the architecture described in Figure 1 was not yet completely in place; the preprocessing and pattern-mining components were fully implemented, but the run-time monitoring and page adaptation were not. The experiment was conducted on the logs collected by the non-adaptive web-server for the Winter 2002 academic term: from January 7th to April 14th, 2002. We first used our pattern discovery and analysis methods to extract patterns in the students' behavior, for each week of the term. Then, we evaluated the quality of the recommendations that these patterns would have generated as follows. First, for each week, we identified, in the logs of the four subsequent weeks, prefixes of this week's patterns; they constitute the cases where the adaptive web site would have made a recommendation. If the navigation behavior, immediately following the identified pattern prefix, was to visit the recommended page, we inferred that the recommendation would have been successful.

4.1 Preprocessing

The first step of the pattern-mining process was to preprocess the web-server logs. The result of preprocessing was the reduction of the average size of each day's log by 75.15%, and the average number of log entries goes down by 57.00% after removal of trivial records and run-length encoding of the identified sessions in the log file.

4.2 Pattern Discovery

We processed the web server logs on a weekly basis, to coincide with the regular update cycle of the web site. Each weekend, instructors made certain that the lecture notes for the past week's lectures and the subsequent week's labs were posted. The first week is the week of Jan 7th, and the last week is the week of April 8th, 2002.

The graph of Figure 2 shows the numbers of discovered patterns when the pattern-qualification criterion in IPM is configured with different allowable insertion-error values. The minimum pattern length was set to be at least 5, and the minimum number of occurrences of a pattern was dynamically set to be 0.01% of the value of the size of processed log. The maximum pattern length was not bounded in this experiment.

As can be seen from Figure 2, not surprisingly, the higher the number of insertion errors allowed, the more patterns were discovered. In fact, during weeks 3 and 11, almost no patterns were discovered when the error threshold was set to 0.

It is interesting to note in Figure 2 that, irrespective of the chosen insertion-error threshold, there are two peaks in the number of discovered patterns at weeks 5 and 14. One explanation for this effect is that both these weeks are followed by examinations: the midterm examination was on week 6 and the final examination followed week 14, which was the last week of the term. At these two points, the purpose of the navigation activities of all the visits to the web site would be more focused on preparing for the examination. This overall purpose makes the navigation more consistent, and as a result, more qualifying patterns are discovered. Similarly, week 11 shows a substantially lower number of patterns. This is probably because during week 12, the most important deliverable of the students' coursework was due and as a result, students did not visit the web site as frequently, since they were working almost exclusively on project development.

One surprising result in the graph of Figure 2, is the lack of any interesting effect during week 7. This was the "Reading week" of the Winter 2002 term, during which there are no classes or labs. In spite of the web site's "inactivity", the students seem to have visited in a manner similar to the past week, albeit much less frequently, as can be seen from Figure 6, which displays the number of sessions recorded each week.

To better understand how the students' navigation through the course web site evolved through the term, we also calculated how patterns "expired", i.e., we calculated how many of the patterns discovered in some week, were still valid patterns in the subsequent week. These results are shown in the graph of Figure 3. The histogram bars in this graph reflect the number of patterns (of length 3, 4 5, and 6) that were discovered each week. The plotted lines indicate the number of patterns, discovered during the week before, which were still valid patterns during the current week. Irrespective of length, there are substantially more new patterns than are old, which implies that navigation changes substantially on a weekly basis. This is a result we expected to find, since the web site is updated with new information on a weekly basis, which the students need to access.

4.3 Recommendation Generation

To evaluate the quality of the recommendations generated based on the patterns discovered by IPM, we examined how often the recommendations that would have been generated at run-time, based on the discovered patterns, would have been followed by the web-site users. To that end, we discovered patterns in the navigation behavior of the web-site users during each of the fourteen term weeks. Then we computed the following two metrics: (a) how often these patterns would have been relevant in the four subsequent weeks, i.e., how many prefixes of these patterns exist in the web-server logs for these weeks, and (b) how often the recommendation generated based on them would have been followed, i.e., how often the users' behavior following the patterns' prefixes would have completed the pattern.

Our rationale for this design is that the content of the examined web site changes substantially on a weekly basis. Although changes, such as new lecture notes and announcement, are usually posted each lecture day, more substantial changes, such as assignments and grades, are updated weekly. Therefore, we want to exploit the natural update cycle of the examined site, to focus our pattern extraction in relatively stable time periods in the web-site evolution and our recommendation generation in periods following the time of pattern extraction. The graph in Figure 4 shows the relevance of the patterns discovered in week 1 during weeks 2, 3, 4, and 5. Shorter patterns are more relevant than longer patterns. Patterns of all lengths become decreasingly relevant as time passes.

The graph in Figure 5 shows the effectiveness of the recommendations based on the patterns discovered in week 1 during weeks 2, 3, 4, and 5. By effectiveness, we mean the percentage of times that recommendations were followed. Recommendations based on longer patterns were followed more often than recommendations based on short patterns. This effect is not surprising, since recommendations based on longer patterns are issued after a longer segment of the user's navigation has been matched to a pattern. There is not a very clear trend on whether or not the recommendation effectiveness drops as time passes. In fact, in Figure 5, one can see that, the effectiveness of the patterns discovered in week 1 is higher during week 5 than during week 2. However, there is no trend consistent across all weeks.

The product of pattern relevance multiplied by pattern effectiveness follows a more consistent trend through the term: it generally takes higher values for shorter patterns than for longer patterns and, as time passes, it decreases. More analysis is necessary in order to understand the usefulness of the pattern-based recommendations better.

5. Conclusions and Future Work

In this paper, we presented a pattern-mining approach to dynamic adaptation of focused web sites. We defined “focused web sites” to be web sites that provide information in support of well-defined purposes and are therefore accessed by their users in patterns consistent with these purposes, and not in an exploratory browsing mode. We believe that these web sites are the most likely to benefit from pattern mining as a dynamic recommendation and adaptation mechanism.

Our method for pattern mining in support of focused web-site adaptation involves three steps. First, the web-server access logs are compacted to eliminate implicit accesses and the user sessions are identified. Next, the IPM algorithm is used to extract sequential patterns of the desired occurrence frequency, length and insertion errors. Finally, the extracted patterns are used to generate recommendations at run time for the web-site users who follow navigation paths similar to the pattern prefixes.

Our case study using a course web site indicates that, indeed, interesting and useful trends can be discovered in the navigation behavior of the web-site users, by applying IPM in the web-server logs. Our work is still in a preliminary phase and initial results shows that useful runtime recommendations can be generated using the discovered patterns. We have partially built the infrastructure necessary to support such a pattern-mining based capability for web-site adaptation, and we plan to use it to collect more data and evaluate our approach more extensively in the short-term future.

Acknowledgements

This work was supported by an IRIS-4 grant on “Knowledge management and service provision for mobile users” and ASERC, the Alberta Software Engineering Research Consortium.

References

1. Agrawal, R., Srikant, R., *Mining Sequential Patterns*, In Proceedings of the 11th International Conference on Data Engineering, ICDE, pp. 3-14, 1995.
2. Brejova, B., DiMarco, C., Vinar, T., Hidalgo, S. R., Holguin, G. and Patten, C, *Finding Patterns in Biological Sequences*, Unpublished project report for CS798G, University of Waterloo, Fall 2000 (http://monod.uwaterloo.ca/cs798/pattern.pdf).
3. Cooley, R., Mobasher, B. and Srivastava, J., *Web Mining: Information and Pattern Discovery on the World Wide Web*, In Proceedings of the 9th IEEE International Conference on Tools with Artificial Intelligence, November 1997.
4. El-Ramly, M., Stroulia E. and Sorenson, P., *Recovering Software Requirements from System-user Interaction Traces*, In Proc. of 14th International Conference on Software Engineering and Knowledge Engineering (SEKE’02), 2002.
5. El-Ramly, M., Stroulia, E. and Sorenson, P., *From Run-time Behavior to Usage scenarios: An Interaction-pattern Mining Approach*, In the Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, July 2002, (to appear).
6. Kautz, H., Selman, B. and Shah, M., *The Hidden Web*, AI Magazine, 18(2) pp. 27-36, 1997.
7. Lavoie, B. and Nielsen, H. F., *Web Characterization Terminology & Definitions Sheet*, W3C Working Draft (http://www.w3.org/1999/05/WCA-terms/), 1999.
8. Masseglia, F., Poncelet, P. and Teisseire, M., *Using Data Mining Techniques on Web Access Logs to Dynamically Improve Hypertext Structure*, In ACM SigWeb Letters, 8(3) pp. 13-19, October 1999.
9. Mobasher, B., Jain, N., Han, E. and Srivastava, J., *Web Mining: Pattern Discovery from World Wide Transactions*, Technical Report TR 96-050, Department of Computer Science, University of Minnesota, 1996.
10. Mobasher, B., Cooley, R. and Srivastava, J., *Automatic Personalization Based on Web Usage Mining*, Communications of the ACM, 43(8) pp. 142-151, 2000.
11. Parsa, I., *Web-Mining: New Data Tools to Manage Web Strategy*, A Report from the Direct Marketing Association, (http://www.the-dma.org/cgi/dispnewsstand?article=244)
12. Srivastava, J., Cooley, R., Deshpande, M. and Tan, P. N., *Web Usage Mining: Discovery and Applications of Usage Patterns from Web Data*, SIGKDD Explorations, 1(2) pp. 12-23, 2000.
13. Wang, X., *PagePrompter: An Intelligent Agent for Web Navigation Created Using Data Mining Techniques*, Technical report CS-2000-08, Department of Computer Science, University of Regina, 2000.
14. Weiss, G., *Predicting Telecommunication Equipment Failures from Sequences of Network Alarms*, In W. Kloesgen and J. Zytkow (eds.), Handbook of Knowledge Discovery and Data Mining, Oxford University Press, June 2002.

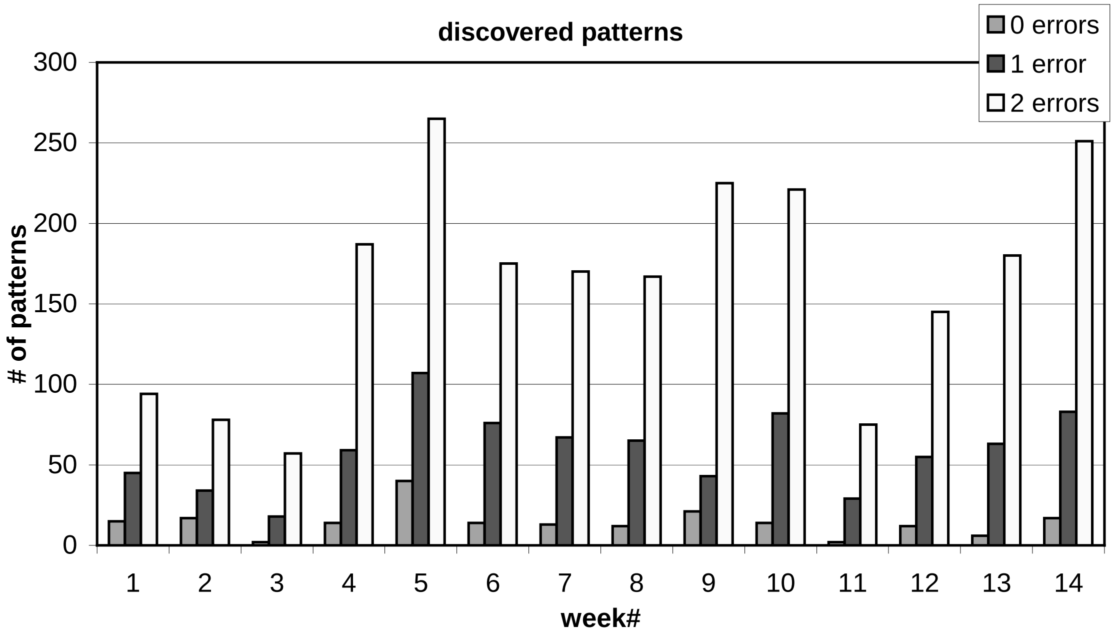

Figure 2: The patterns, of length >4 and with 0, 1, and 2 insertion errors, discovered each of the 14 weeks of the term.

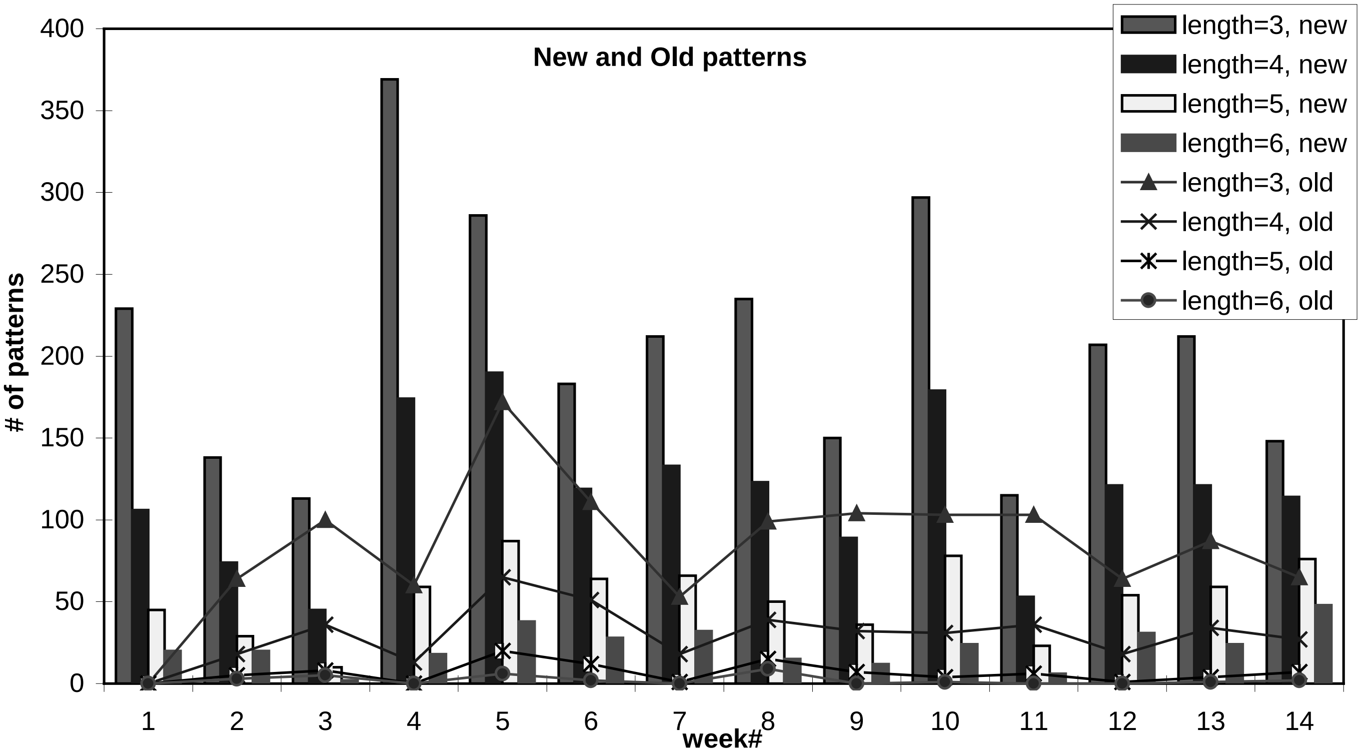

Figure 3: The new patterns, of length 3, 4, 5 and 6, discovered each week and the valid patterns of the week before.

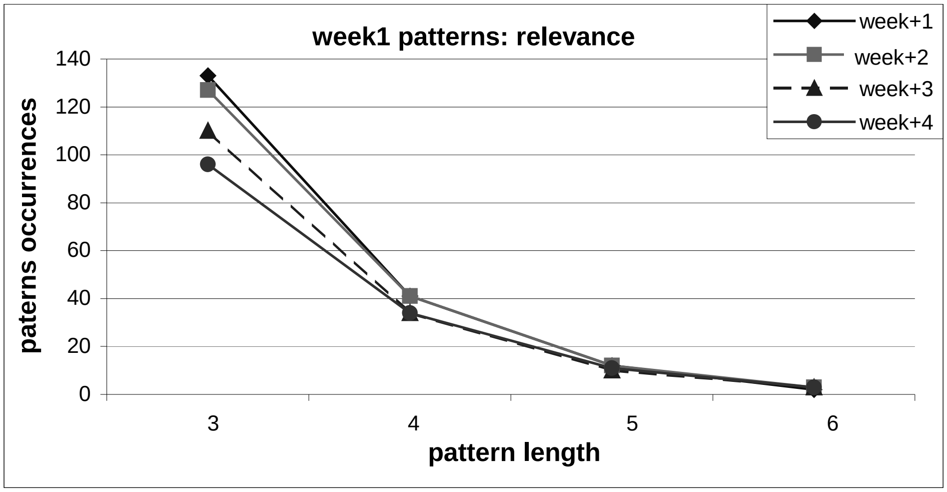

Figure 4: The number of cases in which the patterns of week1, of length 3, 4, 5 and 6, were used to generate recommendations in weeks 2, 3, 4 and 5.

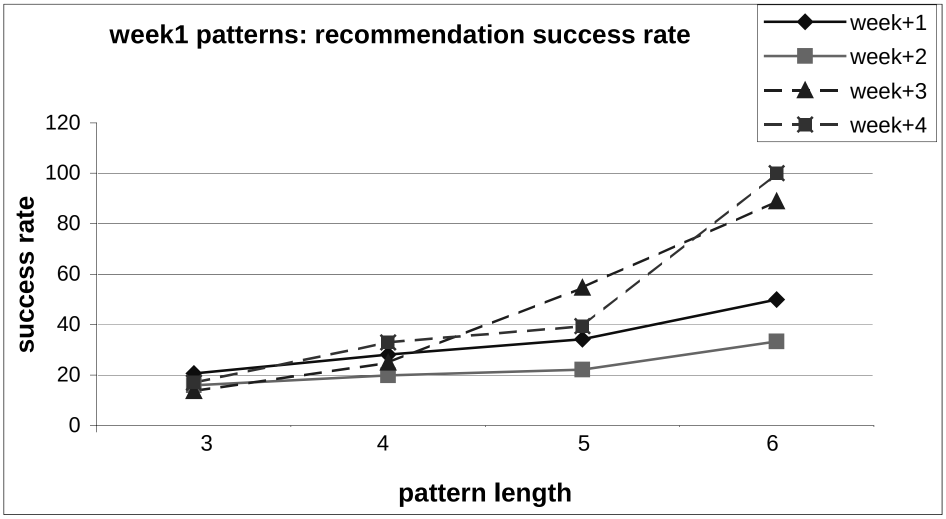

Figure 5: The success rate of the recommendations given based on the patterns of week1, of length 3, 4, 5 and 6, in weeks 2, 3, 4 and 5.

# of Sessions	Week#
1000	Week02
1275	Week03
969	Week04
948	Week05
683	Week06
452	Week07
781	Week08
1099	Week09
650	Week10
725	Week11
550	Week12
780	Week13
755	Week14

Figure 6: The number of user sessions each of the 14 weeks of the term.