

Research

Analysis of Web-usage Behavior for Focused Web Sites: A Case Study



Mohammad El-Ramly^{†‡}, Eleni Stroulia[§]

Department of Computing Science, University of Alberta, 221 Athabasca Hall, Edmonton, Alberta, Canada T6G 2E8

SUMMARY

The number of web users and the diversity of their interests increase continuously; web-content providers seek to infer these interests and to adapt their web sites to improve accessibility of the offered content. Usage-pattern mining is a promising approach in support of this goal. Assuming that past navigation behavior is an indicator of the users' interests, then, web-server logs can be mined to infer what the users are interested in. On that basis, the web site may be reorganized to make the interesting content more easily accessible or recommendations can be dynamically generated to help new visitors find information of interest faster. In this paper, we discuss a case study examining the effectiveness of sequential-pattern mining for understanding the users' navigation behavior in *focused* web sites. This study examines the web site of an undergraduate course, as an example of a web site, that offers information intrinsically related to a process and closely reflects the workflow of this underlying process. We found that in such focused sites, indeed, visitor behavior reflects the process supported by the web site and that sequential-pattern mining can effectively predict web-usage behavior in these sites. Copyright © 2000 John Wiley & Sons, Ltd.

KEY WORDS: H.2.8 Data mining, H.3.5 Online Information Services, Web-based services

*Correspondence to: Department of Computing Science, University of Alberta, 221 Athabasca Hall, Edmonton, Alberta, Canada T6G 2E8

[†]This work was conducted while the first author was a Doctoral candidate at the University of Alberta. He has since assumed a Lecturer position with the Department of Mathematics and Computer Science, University of Leicester, University Road Leicester, LE1 7RH, UK.

[‡]E-mail: mramly@cs.ualberta.ca

[§]E-mail: stroulia@cs.ualberta.ca

Contract/grant sponsor: IRIS; contract/grant number: Knowledge Management and Service Provision for Mobile Users

Contract/grant sponsor: ASERC; contract/grant number: N/A



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. This loose and dynamic organization structure makes finding relevant documents with useful information a challenge. At any point in time, each web site is visited by many users, with different goals, who are interested in different content and prefer different types of presentations [12]. 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, it is highly unlikely that any single organization of the content of a web site will be satisfactory for all these varied needs. Therefore, dynamically adapting the web site to better fit the individual visitor's preferences has become a great challenge for content providers and, at the same time, 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 their 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 recommendation for "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. We are particularly interested in *focused* web sites, that is, web sites whose purpose is to accompany and to support an on-going process by providing information intrinsically related to the process in question. It is already well known [3, 18] that web-user behavior differs depending on whether the user is surfing in an exploratory mode or searching for specific information. We are interested in investigating a related, more specific hypothesis: users of a focused site share a common purpose in accessing the site, which defines, to a great extent, their navigation and access behavior. Furthermore, the nature of the web-site content evolves in a regular way and the content-evolution patterns are reflected in the user navigation behavior as well.

We have used the web site of an undergraduate course at the Department of Computing Science, at the University of Alberta, as an example of a focused web site. The users of the course web site are primarily the students enrolled in the course, 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 and lab notes, to keep abreast with important announcements, 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 should be reflected 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, and 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 is less differentiated by their demographics and personality. 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.

The work we report in this paper is novel in several ways. First, we use a new sequential-pattern mining algorithm, IPM2 [9], for efficiently extracting approximate behavior patterns so that slight navigation variations can be ignored when extracting frequently occurring patterns. Second, we examine the benefits of frequently updating the usage patterns, on the basis of which the web-site documents are recommended. 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 PRACTICE AND RESEARCH

The problem of web-site adaptation for the purpose of improving the user experience with the site has been recently receiving increased attention. There is a single fundamental hypothesis underlying current practices and research: that understanding the needs of the site visitors can be used to reorganize the structure of the site and to filter its content so that these needs are better met and the overall user experience of the site is improved. In this section, we first review the overall web-site adaptation issues and then focus specifically on pattern-mining research.



2.1. Web-site adaptation

An established practice of most commercial web sites is to explicitly require their visitors to provide information regarding themselves, their interests and their opinions on the site content. The collected profiles of the registered users are then clustered to identify demographic constituencies in the visitor population. Such constituencies can then be used as a basis of a collaborative-filtering method, whereby, new users are classified according to their demographics and they receive content that their peers have judged of high quality.

There are several fundamental problems with explicit information-collection approaches. First, it infringes upon the user's privacy, and often visitors prefer to ignore a web site rather than provide their personal information. Furthermore, it often is time consuming, especially as users find that they have to provide the same information repeatedly. In addition, the collected information is assumed not to expire, which is unrealistic in most cases. Finally, the quality of this information is not necessarily high; the collected data can be rather superficial, since the collection process has to be short, and it is not necessarily a good indicator of the user behavior. In fact, recent research [4] has shown that implicit metrics, such as time spent on a page and amount of scrolling performed on a page, are strongly correlated with explicit interest indicators. These implicit metrics could be equally good indicators of the users' interests and have the advantage that they can be unobtrusively collected by specially instrumented browsers.

Indeed, web-usage logs collected at the client side, either by specially instrumented browsers or by personalized proxies or even using cookies, have been used to mine association rules between documents in a web site [10]. Then, a recommender system proposes to the web-site visitors documents that might be interesting to them by applying the mined rules to the recent visitors' navigation history. Such collaborative recommender systems have been recently moving from research to practical implementations; among the most well-known examples is Alexa <http://www.alexa.com>. These techniques can also be criticized as invading the privacy of the web users, since using the cookies each user's navigation is recorded exactly.

Web-usage data can also be collected at the server side: servers are usually configured to keep logs of all the requests they receive. This is the most readily available source of information and recent research has focused on deploying machine-learning and data-mining methods for automatically adapting the web site based on the navigation activity recorded in these logs. However, in order to produce useful-to-mine data server logs have to undergo substantial pre-processing and clean-up [7]. First, requests not explicitly issued by users are ignored. Such requests are, for example, the image-download requests for the images included in a requested page and requests by crawlers. Then the cleaned log is segmented to identify individual visit sessions. The collected sessions are examples of the site usage by its visitors and can be mined to extract frequent usage patterns. An access pattern is a recurring sequential pattern among the entries in the web-server log [19], indicating that many users repeatedly access the same series of pages. These patterns are usually assumed to be indicative of the user's needs and can then be analyzed to infer adaptations to the web site that would make the pages included in the patterns more easily accessible to the site visitors.

In general, adaptation based on server-log data can be static, i.e., reorganization of the site structure, or dynamic, i.e., run-time adaptation of the served documents to include



recommendations of possibly relevant pages. An example of the former type is the synthesis of new index pages organizing the documents that are frequently visited together within a single session [16]. The implied objective of such web-site organization change is to simplify the navigation through the documents that are visited together, by introducing extra links that constitute shortcuts to cliques of related pages.

Such web-site structure changes implicitly assume that the web-site visitors share a fairly uniform behavior that does not change much over time. A more realistic set of assumptions underlies the dynamic recommendation research. Instead of introducing new pages and links to the web site, it assumes that a run-time component monitors the visitor session and dynamically adapts the requested documents to include links to pages that the visitor has not yet seen but might likely be interested in, since they are correlated with the recent pages in the navigation path.

The central issue of this work is, of course, “what constitutes a meaningful usage pattern” and to address it, research has examined improvements to all phases of the log preprocessing, pattern mining, recommendation generation process [13]. To extract meaningful patterns that can effectively predict navigation behavior, the original usage examples have to be of high quality. Several heuristics have been examined for extracting sessions from the server log. Time-based heuristics segment the log based on how much time has lapsed between subsequent requests from the same browser. Other, more sophisticated heuristics identify consecutive request sequences from a page to the pages linked by it, without however producing substantially better results [2]. Once the user sessions have been extracted, they can be treated as a single uniform body or they can be clustered to infer user profiles [15]. Finally, several mining algorithms may be used for the crucial step of navigation pattern extraction, and they are reviewed in the next subsection 2.2.

Finally, it is important to note that a lot of effort has been invested in understanding the nature of web navigation as a function of the user’s purpose, although these results have not directly been used to inform the web-usage mining and recommendation process. Early on, analysis of server logs indicated that web-site visitors may be purposeful information searchers or unfocused browsers and, depending on their type, their navigation patterns are distinct [3]. Later, a user study, investigating web usability issues [18], also concluded that there are substantial differences in the way users experience a web site, depending on whether they are searching for specific information to accomplish a task, or whether they are simply browsing.

2.2. Pattern Mining

The majority of web-usage mining research has viewed the problem of navigation-pattern extraction as an instance of association mining. Association analysis has been widely used for market basket or transaction data analysis. These methods aim at finding patterns within a single transaction, ignoring the relative order of the transaction steps. A substantial body of this work has relied on the Apriori algorithm [1]. Apriori addresses the similar problem of discovering sets of URLs that co-occur frequently within a user’s session, irrespective of their relative order in the session. The intuition behind the application of apriori in web-usage mining is that “if two documents are frequently visited together in a site, then they must



contain related information; therefore if a user visits any one of them, s/he he will most likely also visit the other soon”.

Formally, given A , an alphabet of unique URL IDs, i.e., the set of IDs corresponding to the web-site URLs, and $S = \{s_1, s_2, \dots, s_n\}$ a set of sessions, where each session $s_i, 1 \leq i \leq n$ is a sequence of URL IDs from A that represents the navigation behavior of a web-site user, the objective of Apriori frequent-itemset mining is to identify all *sets of URLs* $fs = \{u_1, u_2, \dots, u_m\}, u_{i, 1 \leq i \leq m} \in A$, such that the URL IDs of fs exist in every session of a subset S' of S , which is called the *support set* of fs . A rule R generated from such a frequent itemset is a tuple $R = (\{a_1, a_2 \dots a_i\}, \{a_{i+1}, a_{i+2} \dots a_m\}), a_{i, 1 \leq i \leq m} \in A$ where, a subset of the URLs in fs form the antecedent part of the rule and the difference of fs minus the antecedent set of URL IDs stands for the consequent part. Such a rule implies that if a user visits the combination of URLs in the antecedent part, then s/he will also visit the URLs in the consequent part.

The WEBMINER (now WebSIFT) project [5, 13, 19, 6] is among the most mature projects in this area, and it has resulted in an elaborate and flexible framework for web-usage analysis. It has been designed to support collection of multiple types of data, including user-registration profiles and server logs, and its analysis for extraction of statistical information and association rules. The architecture diagrams of WebSIFT also include references to a “sequential pattern mining” component, but to our knowledge no results have been reported on sequential-pattern mining for web-usage logs.

SpeedTracer [23] adopts a slightly different approach to the problem; in addition to frequently co-occurring page requests, it also extracts frequently traversed path segments. Essentially, it uses the structure of the web site to identify paths of consecutively linked pages that are frequently part of user sessions.

WUM [17] adopts a top-down approach to pattern mining: it provides a query language that can be used by site designers and maintainers to examine their hypotheses regarding their visitors behavior. The underlying assumption is that web-site designers organize web sites according to an implicit model of the expected navigation through the site, and they need to validate, or possibly revise, this model against the actual behavior captured in the server logs.

In our work, we have been investigating whether frequently occurring sequential patterns of URL accesses may provide good-quality run-time recommendations. Sequential pattern functions, which are also known as temporal pattern functions, analyze a collection of items over a period of time. Sequential pattern mining 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 [5]. 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, sequential-pattern mining 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. This is particularly useful in case of focused web sites, where user



navigation usually depends on the workflow of the activity supported by the web site. 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 the document `/lecturenotes.html`, within their following 5 requests to the server”.

Formally, given A and $S = \{s_1, s_2, \dots, s_n\}$ as above, the objective of sequential-pattern mining is to identify all *sequential patterns of URL IDs* $p = \langle a_1, a_2, \dots, a_m \rangle, a_{i,1 \leq i \leq m} \in A$, such that p is a *subsequence* of every session of $S' \subset S$, which is called the *support set* of p . Different sequential-pattern mining algorithms adopt different operational definitions of what it means for the pattern sequence of IDs to be a *subsequence* of its support-set sessions. Irrespective of the definition adopted, however, once a set of patterns has been identified, they can be used as the basis for producing rules, predicting usage behavior. A rule R generated from a sequential pattern $p = \langle a_1, a_2, \dots, a_m \rangle$ is a tuple $R = (\langle a_1, a_2 \dots a_i \rangle, \langle a_{i+1}, a_{i+2} \dots a_m \rangle)$ where, a prefix of p of length i is the antecedent and the corresponding suffix of p of length $m - i$ is the consequent part. Such a rule implies that if a user visits the URL sequence in the antecedent part, then s/he will also visit the URLs in the consequent part. Note that, unlike Apriori-based rules, both the antecedent and consequent parts take temporal order into consideration. The implication of this difference is that given a frequent URL itemset many more rules can be generated than given a sequential pattern, and the focus of this paper is to examine whether the more conservative predictions of sequential-pattern mining are more effective in predicting web-usage behavior, in case of focused web sites.

Sequential pattern mining methods have been applied to problems in a variety of domains. In bio-informatics, for example, the task is to discover frequently occurring subsequences of amino acids that may uniquely define a specific aspect of the biological function of a cell [8]. In the area of systems monitoring, such as network monitoring, the task is to predict faults and abnormal performance patterns [22].

In [20], the authors describe a dynamic recommendation case study, where an intelligent agent supports user navigation through a course web site. The agent analyzes overall usage, web-page content, web-site structure and current user’s actions. The agent’s recommendations patterns extracted based on association rules between URLs and web users, without considering the evolution of these relationships over time. Our case study explicitly studies the evolution of the patterns’ relevance and the corresponding effectiveness of the pattern-based adaptation in time.

IPM2 [9] 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.

3. DYNAMIC, RUN-TIME PAGE RECOMMENDATION

Our approach to dynamic, web-site adaptation follows a process, generally adopted by work in this area, consisting of preprocessing, session extraction, pattern mining, and pattern recommendation steps.



In the preprocessing phase, the raw web-server log file is cleaned of all requests not explicitly issued by users. Next, the distinct sessions contained in the log are identified.

Next, for the pattern-mining phase, we applied the IPM2 [9] algorithm to the extracted sessions to identify frequently occurring navigational patterns. IPM2 is designed to recognize approximate patterns, i.e., patterns that may include spurious steps in addition to the essential pattern steps. More specifically, according to the IPM2 operational definition of *subsequence*, each session in the support set $s \in S'$ of a pattern p contains at least one subsequence sub that starts with the first URL of p , ends with the last URL of p , and all IDs in p should exist in the same order in sub , which may also contain other additional URL IDs. Formally, $\forall s \in S'$, the support set of p , there exists at least one subsequence sub such that, $p[1] = sub[1], p[|p|] = sub[|sub|], |p| \leq |sub|$, and if $i < |p|, j \leq |p|, i < j \implies sub[k] = p[i] \wedge sub[l] = p[j] \wedge k < l$. The above predicate defines the class of patterns discovered by IPM2, namely, patterns with at most a preset number of additional URL insertions. The flexibility of this *subsequence* definition enables some robustness to noisy data, which is very likely to be the case with web-site navigation data. The *support* of a pattern p is the number of its subsequences in S' .

Finally, the patterns extracted with IPM2 can be 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. In this case study, we simply looked in the “future” sessions of the server logs to identify opportunities of recommendation and whether they might have been followed or not.

3.1. Usage-trace Preprocessing and Session Identification

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. Each time a Web browser downloads an HTML document, the images included in the document generate corresponding requests, causing each of those accesses to be stored in the log file of the server (unless the “images automatic load” option of the client is switched off). Thus, to reduce the amount of data to be processed, the original log is cleaned and standardized.

Like most Web log analysis tools, the cleaning process employed in our method performs the following tasks. First, requests for URLs containing graphics, sounds, or video files are filtered out automatically. Irrelevant items could be 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, gif, jpg, jpeg, wav, au, ai, ico and map are removed following the predefined cleaning rules. The set of suffixes could be adjusted as needed for particular web sites by making changes of the cleaning criteria. Second, other known useless data are automatically filtered out like entries with some script files such as `counter.cgi`, records with particular code such as HTTP status code 404, which means resource is not found on the server, and accesses performed by agents such as crawlers, robots, or spiders. Third, some mistyped URLs and automatically created links that are created by CVS (Concurrent Versions System) or other tools may also exist. These ones are identified and then filtered out.



The objective of the second process is to eliminate from the log duplicate URLs referring to the same directory. First, all directory URLs that include sorting preferences are standardized. For example, when a directory URL ends with a suffix of the form $?C_1 = C_2$ where C_1, C_2 are characters, such as $?S = D$ indicating that the directory files are listed in descending size order, it is replaced by its prefix, up to and excluding the sorting-preference description. Second, all directory URLs are standardized to start with `http://www.` and to end with no trailing `/`. This means that requests for any of the links `www.cs.ualberta.ca/~stroulia`, `http://www.cs.ualberta.ca/~stroulia/`, or `http://cs.ualberta.ca/~stroulia` all refer to the same file, and they are all transformed to one standard format: `http://www.cs.ualberta.ca/~stroulia`.

After the server log has been cleaned, session identification is performed. For our case study, we used a time-based session identification method. All requests from the same IP address on the same day, occurring within 30 minutes from the first request from that IP, are considered one session. The first request from the same IP after this 30 minutes period is considered the beginning of a new session, which includes all the requests occurring within the next 30 minutes, and so on. For example, assume the following requests from the same IP, where each request is represented by the time and a URL, represented by an ID: (9:04, 219), (9:08, 230), (9:10, 227), (10:16, 1), (10:16, 319), (10:16, 404). The first three requests will form one session and the rest will form another.

Finally, the identified sessions are rewritten in run-length encoding [21], to satisfy the input-data requirements of IPM2. The sessions, originally represented as sequences of standardized URLs, may contain repetitions, resulting from pressing the “refresh” browser button for example. These repetitions may result in missing useful patterns, since they will differentiate otherwise similar navigation behavior. The run-length encoding algorithm replaces immediate repetitions with a count followed by the URL being repeated. In our experiment, this count is ignored as consecutive requests for the same page does not say much. Note that, sessions consisting of a single URL are removed as they cannot yield any meaningful patterns.

3.2. Web-site Usage-Pattern Discovery

The IPM2 algorithm discovers patterns with noise in the form of insertion errors, i.e., additional URLs not belonging in the pattern. Allowing insertion errors allows IPM2 to tolerate minor differences in navigating a web site, in which the user accessed some extra web pages while completing a navigation sequence. Navigation segments with such noise will still be discovered as instances of the original pattern.

The algorithm takes as input user sessions represented as sequences of URL IDs drawn from an alphabet corresponding to the URLs of the web site of interest. Additionally, the algorithm requires a “pattern interestingness” criterion, that can be tailored according to the problem in hand. This criterion is a function of five parameters, as follows:

1. the minimum pattern length (optional);
2. the maximum pattern length (optional);
3. the minimum support, i.e., number of pattern occurrences, required for a pattern to be considered interesting (required);



4. the maximum number of insertion errors allowed in instances of the discovered patterns, i.e., the number of spurious web pages that IPM2 should tolerate in these pattern instances (required); and
5. the minimum score of a pattern (optional).

If no minimum or maximum pattern lengths are specified, the algorithm will report all patterns of length 2 and above, fulfilling the other specified parameters. The minimum support defines how many instances are required in order for this pattern to be interesting. It can be set to a specific number or to a percentage of the input data size. The default value for the minimum support is 2. The default value for the maximum number of insertion errors is 0. However, usually a higher value would be tried to see how sensitive the pattern discovery process is to noise. The function used to assess the score of a pattern p is as follows: $score(p) = \log_2 |p| * \log_2 support(p) * density(p)$ where $|p|$ is the length of the pattern, $support(p)$ is its support and $density(p)$ is the ratio of $|p|$ to the average length of the pattern instances. Since these instances may include some noise, they might be longer than their pattern. For example, let $\langle 2, 4, 3, 4 \rangle$, $\langle 2, 4, 3, 2, 4 \rangle$ and $\langle 2, 3, 4 \rangle$ are the available instances of a pattern, $p_1 = \langle 2, 3, 4 \rangle$ with at most 2 insertions allowed. Hence, $density(p_1) = 0.75$. Typically, an experiment would require trying different values for the five parameters above to see how sensitive the results are to the variability of different parameters and to reach a reasonable results set.

After preprocessing the web server logs, URLs are given unique integer IDs to suit the input format needed for IPM2. IPM2 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. So, if $p_1 = \langle 2, 3, 4 \rangle$ and its support is 30, $p_2 = \langle 2, 3 \rangle$ and its support is 30 and $p_3 = \langle 3, 4 \rangle$ and its support is 35, then the final result set will include p_1 and p_3 only.

IPM2 follows a typical data-mining search strategy. 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 algorithm, please refer to [9].

3.3. Pattern-based recommendation

IPM2 extracts a set of sequential navigation patterns from the preprocessed web-server logs. These patterns can be used in a variety of ways.

First, they can be used as a means of understanding the navigation behavior of the web-site users. For example, the existence of many long patterns might imply usability problems: in principle, users want to be able to accomplish their tasks in shorter rather than longer visits. Alternatively, patterns can be used as the basis for generating run-time recommendations for pages that new users of the web site may want to visit, as discussed in Section 2.2.

A run-time infrastructure is required to enable the later use, capable of user session tracking and relevant-pattern selection. 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's browser. One technique for addressing this problem is dynamic page rewriting with hidden



fields: when the user first submits a request, 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 supplies 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.

The result is that, at any point in time, the web server knows the user's recent navigation history, and can therefore examine it to identify whether it includes the prefix of any of the collected patterns. If so, then the suffixes of the relevant patterns may be offered as recommendations for subsequent navigation.

The page-rewriting technique can easily support the dynamic adaptation of the pages requested by the web-site users with the recommendations on new potential places to visit. As there might be several sequential patterns that the current visiting path satisfies, the web-site recommendation generation can be based on pattern selection or combination. In the first case, the score provided by IPM2 for each extracted pattern could be used to prioritize the patterns for the purpose of recommendation generation. Each time a pattern-based generated link is added to a requested document, the server can keep track whether the link was followed. By calculating the ratio of how many times the recommendation of a pattern has been followed to the number of times that it has been generated, the server could evaluate whether the pattern is effective in recommendation generation or not. The case of combining patterns is possible when few sequential patterns' prefixes match the navigation behavior of the user. In this case, all IDs in the suffixes of the relevant patterns may be offered, regardless of patterns' scores.

Consider the following simple example of the case of pattern selection for recommendation. Assume that two patterns were discovered in recent server logs, $p_1 = \langle 1, 2, 3, 4 \rangle$ and $p_2 = \langle 1, 2, 5, 6 \rangle$, and a new web-site user has visited pages 1, 2 (page 2 is the current page shown on the user's browser). If $score(p_1) > score(p_2)$, then p_1 will be chosen to generate the run-time recommendation. The URLs corresponding to pages 3 and 4, will be included as hyperlinks at the top of page 2, offering shortcuts to pages 3 and 4. Hence, if one of these recommendations is followed, it saves the user's time and eliminates some unnecessary navigation. Note that although page 3 comes after page 2 in one of the patterns, this does not mean that there is a direct link in page 2 to 3. This is because navigation using the *back* and *forward* buttons of the browser is not recorded at the server side. If pattern combination is chosen, then the URLs corresponding to pages 3, 4, 5 and 6 will be included as hyperlinks at the top of page 2.

There are several interesting issues that arise by deploying such an infrastructure. For example, "to what extent do the recommended URLs change the original organization of the web site and affect user behavior?" In principle, some of the recommended URLs, dynamically added to the web-site documents, may already exist as links in these documents. In such cases, these recommended URLs would not change the original web-site organization, although they would probably increase the saliency of the existing links, thus further encouraging their access. More specifically, in the case of recommendations generated based on patterns discovered with IPM2, it is quite likely that the recommended links are not already included in the original page since these patterns often exclude intermediate pages found in their supporting sessions.

A related, and even more important, issue is whether or not it is desirable to recommend to new users the navigation behavior of their predecessors. The fact that many users followed



similar navigation paths through the web site does not necessarily imply that these paths lead to desirable documents; one could possibly argue that they might all lead to dead ends. The implicit assumption of all research in dynamic run-time recommendation generation is that the behavior of a sizable population of web-site users should be appealing to their peers. The validity of this assumption can be neither ascertained nor refuted without explicit user feedback, and to our knowledge, no such study has been carried out. However, for focused web sites, this assumption is supported by the fact that user navigation usually follows the workflow of some underlying process supported by the web site.

4. EXPERIMENTAL EVALUATION

We have experimented with our approach on logs collected from a real web site of an undergraduate course (CMPUT 301) at the Computing Science Department, University of Alberta. We examined the web-server logs for two academic terms, Fall 2001 (from September to December 2001) and Winter 2002 (from January to April 2002), and we evaluated our preprocessing and pattern discovery and analysis methods, to see what kind and quality of usage patterns, and hence, run-time recommendation our method can generate.

4.1. Usage-trace preprocessing

The collected server logs were preprocessed as described earlier in subsection 3.1. Then, log entries were replaced by integer IDs as required by IPM2.

4.2. The evolution of the web-site visitor sessions

First we examined the nature of the traffic to the web site. The objective of the web site was to communicate information relevant to the course and, as a result, its evolution followed the “workflow” of the course. For example, the instructors updated the lecture schedule with new material just before each lecture (on Mondays, Wednesdays and Fridays). All assignments were due on a Monday and usually the week before related announcements would be posted to the course whiteboard. We expected that the traffic to the web site would also reflect the course workflow.

Figures 1(a1,a2) and 1(b1,b2) depict the number of visits to the course web site during the Fall 2001 and Winter 2002 terms, respectively; the granularity of the former is daily and that of the latter is weekly. These figures validate our original expectations.

In Figures 1(a1,a2), it is very clear that the web-site traffic is periodical in nature; the period is a week, coinciding with the intrinsic period of the course-delivery process that this web site was designed to support. In Figure 1(a1), we can see that the second peak occurred on day 49, which is the Sunday right before the midterm exam of the Fall 2001 term, which was on the subsequent Monday. In Figure 1(b1), we can see that the peak of user sessions per week was in week 4 of the Fall 2001 term. This week was the week before the first “Demo week” and students worked hard to prepare their demos. In Figure 1(a2), we can see that the number of visits during the weekend is consistently and substantially below the weekly average, except when



a deliverable is due on the Monday after. Furthermore, for most weekends, Saturday's traffic is less than Sunday's. In Figure 1(b2), we also notice that the traffic is lowest during weeks 7, 12 and 14. Week 7 was the Winter term "Reading Week", during which many students take vacations. Weeks 12 and 14 were "Demo weeks", when the students take an informal break, after having worked hard on their projects the weeks before.

4.3. Recommendations based on IPM2 sequential-pattern mining

Our main objective in this case study was to examine how the underlying process workflow affects its corresponding web-site usage, and to investigate whether understanding this relation could be used to inform the recommendation-generation process. Given the highly periodic nature of the course web-site traffic, we designed an experiment to investigate whether traffic early in the week, i.e., during the first three working days, could be used to guide the navigation of web-site users in the later part of the week. First, we discovered usage patterns in the logs of the first three working days of the week (Monday to Wednesday). Next, we analyzed the users' navigation behavior in the last two working days (Thursday and Friday) to identify recommendation opportunities as follows: when a prefix of length 3 of a usage pattern was discovered in a user session, we considered that an opportunity for recommending all documents in the pattern suffix. Finally, we assessed the effectiveness of these recommendations by comparing the recommended URLs with the user's navigation behavior subsequently.

This experiment assumes that due to the nature of course web sites, students tend to exhibit patterns of usage of the site during each week. This is because there are different lecture and lab sessions and assignments every week. So, we expected that most students focus on accessing this new material in a "consistent way". This consistent way can be quantified in the form of sequential patterns of users' navigation of the web site. These patterns are collected by mining the data of the first 3 days of the week, and are used for generating potential recommendations for the students who navigate the site during the rest of the week. A recommendation is only possible if the user's navigation recent history matches the prefix of a discovered pattern. We used the three most recent accessed pages to represent this history. In the cases where this history matched the prefix of some patterns (i.e. it was exactly like the first three IDs of the pattern), pattern combination was used to generate a recommendation: all IDs in the suffixes of all the matched patterns were recommended to the user.

If there are many of such IDs, a criterion for choosing from them is needed; otherwise the recommendation will confuse the user and distract him instead of helping him. One possible criterion is the sum of the support of the patterns that include the ID. Then, the IDs with the highest sums are chosen.

The effectiveness (also usefulness) of a given recommendation is measured by counting how many times at least one of the URLs given in a recommendation exists among the 3 immediate URLs accessed by the user after the 3 history URLs. One could argue that this metric introduces a threat to the validity of the case-study results; its underlying assumption is that the actual navigation behavior of the web-site visitors' is a close approximation of their navigation in the presence of recommendations. Indeed, the actual dynamic rewrite of pages to include recommendations may affect the navigation behavior of visitors: we expect that visitors would be more likely to follow the recommended links, in which case, the case



study underestimates the effectiveness of the recommendation-generation process. The details of pattern discovery and run-time recommendation steps are included in the following.

Pattern Discovery First, for each week we combined the data of the first three working days (Monday to Wednesday). Then, we applied IPM2 to discover the frequent user activities of the students during these days. Different criteria were used with different parameters. The minimum pattern length was set to 4 and the maximum length to 6. This length range was found to be suitable for the type of student tasks, they performed through the web site. After some trials, minimum pattern support was chosen to be 0.2% of the number of entries of the preprocessed logs of these 3 days. This value gave sufficient patterns to generate recommendations and at the same time excluded random and less frequent navigation behavior. If 0.2% of the number of entries is less than 2, then support is set to 2. Different values were tried for the number of insertion errors allowed in a pattern instance in order to see whether allowing noise in the subsequences of the discovered patterns would give better results. Score was set to 0, meaning that it is not used in the discovery criterion. Figure 2 shows the number of patterns discovered using the qualification criterion mentioned, with 3 different values tried for the number of insertion errors: 0, 1 and 2.

Run-time Recommendation Generation In our off-line experiment for evaluating the effectiveness of the recommendations produced by IPM2, we first identified the sessions of Thursday and Friday of each week of length at least 4. Next, we slid a window of length 6 over the length of the session: as each new navigation segment became visible in the window, we evaluated whether any recommendations would have been generated based on the patterns discovered during the beginning of the same week, by matching the first 3 IDs of the segment with the first 3 IDs of the patterns. For each pattern matched, the IDs of the pattern suffix were added to the set of URLs that would have been recommended to the session's user. Given this would-be-recommended set of URLs, the recommendation was considered "effective" if one or more of these URLs were accessed by the user in one of the three pages accessed immediately after the segment that triggered the recommendation.

For example, consider the following pattern list and session taken from the data of week 7 of the Winter term of 2002:

- $Session = \langle \dots 54, 2, 55, 71, 80, 53, 70, 64, 49, 55, 113, 56, 50, 1, \dots \rangle$
- $Patterns = \{p_1, p_2, p_3, \dots\}$
 $= \{ \langle 54, 2, 55, 71, 53, 70 \rangle, \langle 54, 2, 55, 71, 53, 80 \rangle, \langle 2, 55, 71, 53, 70, 80 \rangle \dots \}$

If a sliding window of length six moves over the given session, then the following segments will appear in the window consecutively, one after another:

- $Segments = \{s_1, s_2, s_3, \dots\}$
 $= \{ \langle 54, 2, 55, 71, 80, 53 \rangle, \langle 2, 55, 71, 80, 53, 70 \rangle, \langle 55, 71, 80, 53, 70, 64 \rangle \dots \}$

Note that the prefix of s_1 matches the prefixes of p_1 and p_2 and the prefix of s_2 matches that of p_3 , while the prefix s_3 does not match the prefix of any of the discovered patterns. So, no recommendation can be made for the navigation sequence of the prefix of s_3 . The



recommendations corresponding to s_1 , s_2 and s_3 , respectively, based on the three patterns above only, were:

- $recommendation_1 = \{71, 53, 70, 80\}$
- $recommendation_2 = \{53, 70, 80\}$
- $recommendation_3 = \{\}$

The first two recommendations are considered effective since at least one URL in each of them appears in the suffix of the corresponding segment, i.e., it was accessed by the user in her/his subsequent navigation within three URLs. The percentage of segments, for which a recommendation was made, is calculated per week for the data of Thursday and Friday and is shown in Figure 3 for insertion errors of 0, 1 and 2. The percentage of effective recommendations is shown in Figure 4. Figure 5 shows the corresponding average number of URLs per recommendation.

As we have already discussed, this measure of effectiveness is not an objective measure of the “quality” of the recommendation. It rather measures how well the usage behavior of the future can be predicted given the recent usage behavior, and potentially, the extent to which usage behavior might be influenced by recommendations based on this recent history.

Discussion Given a specific number of insertion errors allowed, it is no surprise that when the number of patterns discovered increases, the percentage of cases when a recommendation is made increases, and the average number of URLs per recommendation increases too. As a result, the percentage of effective recommendations increases. As one can see in Figures 3, 4, 5 what is interesting is that with the increase of the number of insertion errors permitted, the number of URLs per recommendation increases significantly, the percentage of recommendations made increases moderately, but the percentage of accepted recommendations remains almost the same. This creates a trade off between discovering more patterns by allowing higher insertion error and getting recommendations more often with relatively more URLs on one side, and discovering less patterns using a smaller insertion error and getting fewer but focused recommendations, with fewer URLs, on the other side. Using fewer patterns to generate recommendations means faster processing at run-time when the user’s navigation history is checked against the discovered patterns for prefix match. But also, it means less chance of recommendation generation due to less chance of prefix match. In the data set used in this experiment, an insertion error of 1 was a good compromise. It almost gave the same percentage of recommendations and the same percentage of recommendation effectiveness as an insertion error of 2, but with about 1.5 URLs less on average in Fall 2001 and about 0.6 URLs less on average in Winter 2002.

Another interesting point to notice is that students were less active in navigating the course web site during the last third of both terms, as can be seen from the size of the processed logs in Figures 1(c1,c2). Consequently recommendations were made less often and were less effective during these periods as can be seen in Figures 3, 4.

The results shown in Figures 2, 3, 4, 5 are encouraging and promising. By averaging the data of the Fall 2001 term for an insertion error of 1, it was possible to make a recommendation for 68% of the pages the users accessed on Thursdays and Fridays. 84%



of these recommendations were effective. The average size of a recommendation is 4.4 URLs. This average recommendation size can be reduced to 3 by allowing 0 insertion errors. But this drops the average percentage of the pages getting recommendations to 55% with an effectiveness average of 82%. For the Winter 2002 term, student activity using the web site drops significantly in the second half of the term. The processed logs of the first half has 2.4 times the entries of the second half. For the first half of the term, an insertion error of 1 generated patterns that could generate recommendations for 23% of the pages accessed on Thursdays and Fridays, on average. 72% of these recommendations on average were effective and each had an average of 3 URLs. Using 0 insertion errors reduces this average to 2.5 URLs but recommendations are generated only for an average of 15% of the pages accessed.

5. CONCLUSIONS AND FUTURE WORK

In this paper, we presented a case study designed to investigate some issues related to web-usage mining and recommendation in the presence of regular web-content evolution. We defined *focused web sites* to be web sites designed to support an ongoing process that offer information essential to that process. These web sites evolve fairly continuously and regularly, as a result of the underlying process workflow. Furthermore, their organization usually reflects closely the underlying process and the nature of the users' navigation through them is more task-than data-driven. Users visit these web sites to retrieve specific information in support of well-defined 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, because we expect that the common users' tasks will result in stronger, more frequent navigation patterns.

Similar to the majority of the related research in the area, our web-usage mining and recommendation generation method involves three steps. First, the web-server access logs are compacted to eliminate implicit accesses and the distinct user sessions are identified. Next, navigation patterns of the desired quality are extracted. 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. A novelty in our method is the algorithm used to extract frequent navigation patterns: we use the IPM2 algorithm for extracting *sequential URL patterns* that meet a set of criteria regarding their length, error-density and support, as opposed to the majority of related research that uses apriori-based algorithms for identifying frequently co-occurring *sets of URLs*.

Our case used an undergraduate course companion web site, as an example focused web site. Our experimental results show that, in focused web sites, temporal attributes of the underlying process are, indeed, reflected in the users' navigation behavior. User sessions are periodical in nature; the period is a week, coinciding with the period of the course delivery. Moreover, tasks implied by course deliverables significantly impact the users' navigation behavior.

Patterns discovered by IPM2 can be quite effective in run-time document recommendation if enough navigation history is used to generate them. Our results show that the links offered based on these patterns are actually desired by the users and would be followed in an online environment in about 70% to 85% of the time.



Our work is, by no means, conclusive. Further experimentation is necessary to understand the types of recommendations generated on the basis of co-occurrence (such as using itemsets discovered by Apriori) vs. the types of recommendations generated on the basis of sequential patterns (such as using approximate patterns discovered by IPM2). In addition, we are interested in examining the qualitative differences in the recommendation effectiveness of the two approaches and the quantitative impact that their various input parameters may have. We are currently developing 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 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. The authors wish to thank Nan Niu who was involved with the earlier experiments, reported in the precursor of this paper and the anonymous reviewers whose constructive criticism and feedback helped to improve this manuscript.

REFERENCES

1. Agrawal R, Srikant R. *Mining Sequential Patterns*, In *Proceedings 11th International Conference on Data Engineering*. IEEE Computer Society Press: Los Alamitos CA, 1995;3-14.
2. Berendt B, Mobasher B, Spiliopoulou M, Wiltshire J. *Measuring the Accuracy of Sessionizers for Web Usage Analysis*, In *Workshop on Web Mining at the First SIAM International Conference on Data Mining*. 2001;7-14.
3. Catledge LD, Pitkow JE. *Characterizing Browsing Strategies in the World-Wide Web*. *Computer Networks and ISDN Systems* 1995; **26**(6):1065-1073.
4. Claypool M, Brown D, Phong L, Waseda M. *Inferring User Interests*. *IEEE Internet Computing* 2001; **5**(6):32-39.
5. Cooley R, Mobasher B, 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*, IEEE Computer Society Press: Los Alamitos CA, 1997;558-567.
6. Cooley R, Pang-Ning T, Srivastava J. *Discovery of Interesting Usage Patterns from Web Data*. In *Proceedings of the WEBKDD Workshop*, 1999;163-182.
7. Cooley R, Mobasher B, Srivastava J. *Data Preparation for Mining World Wide Web Browsing Patterns*. *Knowledge and Information Systems* 1999; **1**(1):5-32.
8. Durbin R, Eddy S, Krogh A, Mitchison G. *Biological Sequence Analysis: Probability Models of Proteins and Nucleic Acids*. Cambridge University Press, 1998.
9. El-Ramly M, Stroulia E, Sorenson P. *From Run-time Behavior to Usage scenarios: An Interaction-pattern Mining Approach*. In *Proceedings 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM Press, 2002;315-324.
10. Fu X, Budzik J, Hammond KJ. *Mining navigation history for recommendation*. In *Proceedings of the Intelligent User Interfaces Conference*. ACM Press, 2000;106-112.
11. Kautz H, Selman B, Shah M. *The Hidden Web*. *AI Magazine*, 1997, **18**(2):27-36.
12. Lavoie B, Nielsen HF. *Web Characterization Terminology and Definitions Sheet*. W3C Working Draft, 1997 (available at <http://www.w3.org/1999/05/WCA-terms/>)
13. Mobasher B, Cooley R, Srivastava J. *Automatic Personalization Based on Web Usage Mining*. *Communications of the ACM* 2000; **43**(8):142-151.
14. Mobasher B, Dai H, Luo T, Nakagawa M. *Effective Personalization Based on Association Rule Discovery from Web Usage Data* In *Proceedings Workshop on Web Information and Data Management*, 2001;9-15.



15. Paliouras G, Papatheodorou C, Karkaletsis V, Spyropoulos CD. *Clustering the Users of Large Web Sites into Communities* In *Proceedings of the International Conference on Machine Learning*. Morgan Kaufmann Publishers, 2000;719-726.
16. Perkwitz M, Etzioni O. *Adaptive Web Sites: Automatically Synthesizing Web Pages*, In *Proceedings 15th National Conference on Artificial Intelligence and 10th Innovative Applications of Artificial Intelligence Conference*. AAAI Press, 1998;727-732.
17. Spiliopoulou M, Faultich LC. *WUM: A Web Utilization Miner*. In *Proceedings of the International Workshop on the Web and Databases*, 1999;184-203.
18. Spool J, Scanlon T, Schroeder W, Snyder C, DeAngelo T. *Web Site Usability: A Designer's Guide*. Morgan Kaufmann Publishers, 1999.
19. Srivastava J, Cooley R., Deshpande M., Tan PN. *Web Usage Mining: Discovery and Applications of Usage Patterns from Web Data*. SIGKDD Explorations, 2000, 1(2):12-23.
20. Yao YY, Hamilton HJ, Wang X. *PagePrompter: An Intelligent Agent for Web Navigation Created Using Data Mining Techniques*. In *Proceedings 3rd International Conference on Rough Sets and Current Trends in Computing*. LNAI Springer Verlag, 2002;506-513.
21. Wayner P. *Compression Algorithms for Real Programmers*. Morgan Kaufmann Publishers, 1999.
22. 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, 2002.
23. Wu K, Yu PS, Ballman A. *SpeedTracer: A Web Usage Mining and Analysis Tool*. IBM Systems Journal. 1998, 37(1):89-105.

AUTHORS' BIOGRAPHIES

Mohammad El-Ramly is a lecturer at the Department of Mathematics and Computer Science, University of Leicester, UK. He earned his M.Sc. in operation Research from Cairo University, Egypt and his Ph.D. in software engineering from University of Alberta, Canada. His research interests include reverse engineering, legacy system wrapping and migration and mining sequential run-time data for patterns of user activity.

Eleni Stroulia (<http://www.cs.ualberta.ca/~stroulia>) obtained her B.Sc. degree from the University of Patras in Greece and her M.Sc. and Ph.D. degrees from the College of Computing, Georgia Institute of Technology. She is currently an Associate Professor with the Computer Science Department at the University of Alberta. Her current research focuses on reverse engineering and modeling software, developing tools for supporting collaborative software development, and using artificial-intelligence methods for wrapping and integrating existing web-based applications. She is a member of ACM, IEEE, and AAAI.

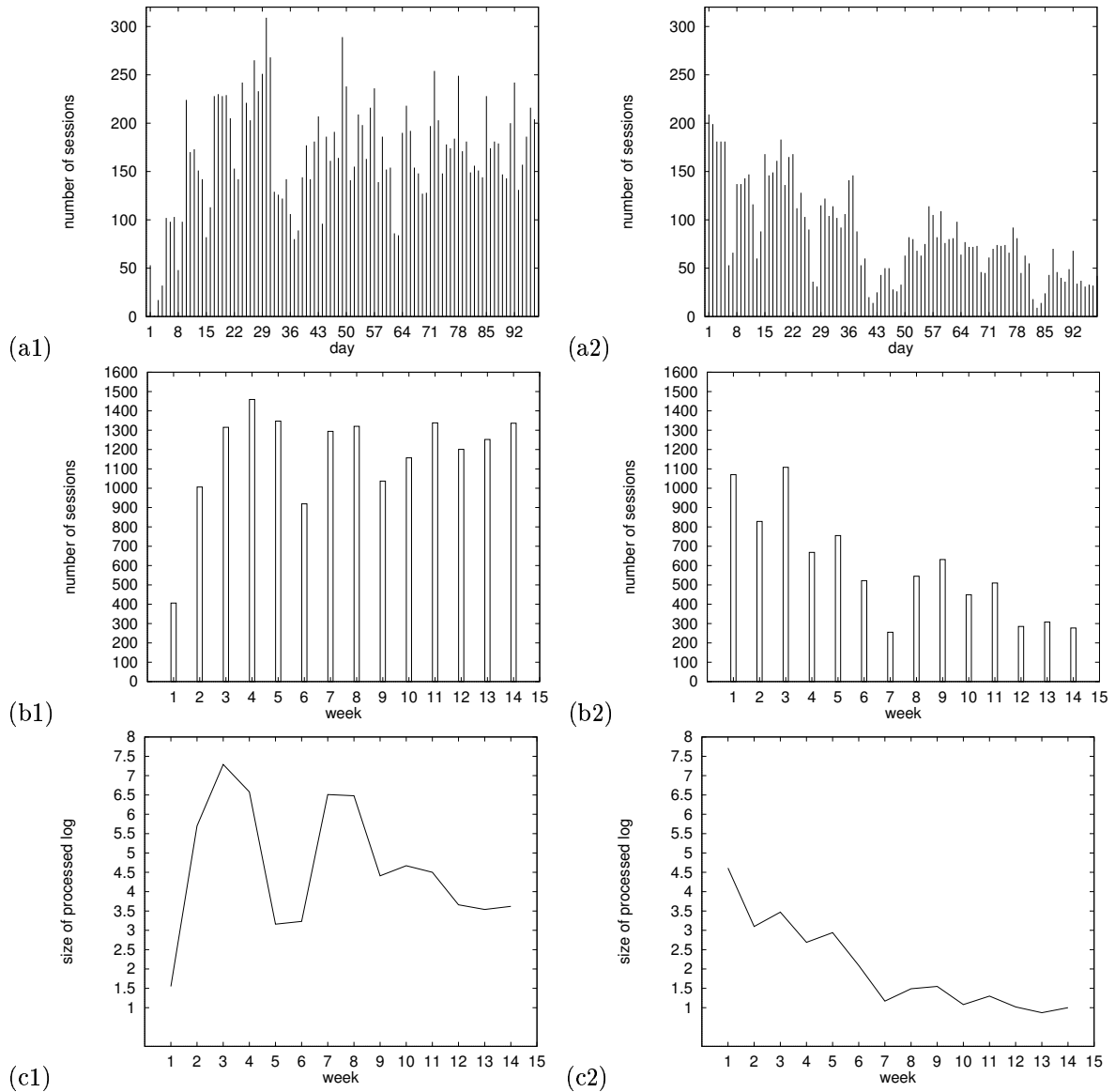


Figure 1. Some quantitative measures on the web-site traffic: (a1) Sessions during each day of the Fall 2001 term. (a2) Sessions during each day of the Winter 2002 term. (b1) Sessions during each week of the Fall 2001 term. (b2) Sessions during each week of the Winter 2002 term. (c1) Weekly server-log size during the Fall 2001 term (in 1000s of entries). (c2) Weekly server-log size during the Winter 2002 term (in 1000s of entries).

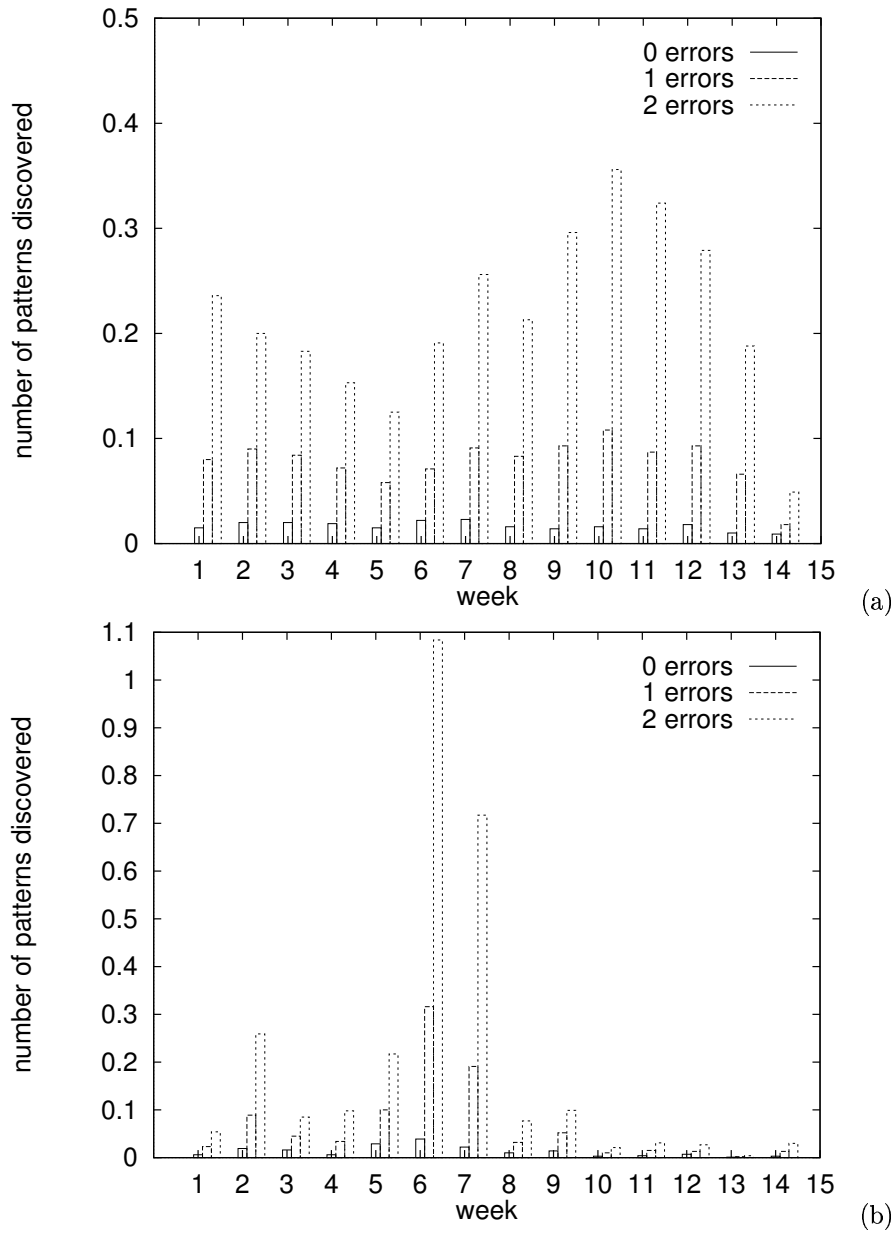


Figure 2. Number of sequential patterns discovered each week (in 1000s) during (a) Fall2001 and (b) Winter 2002.

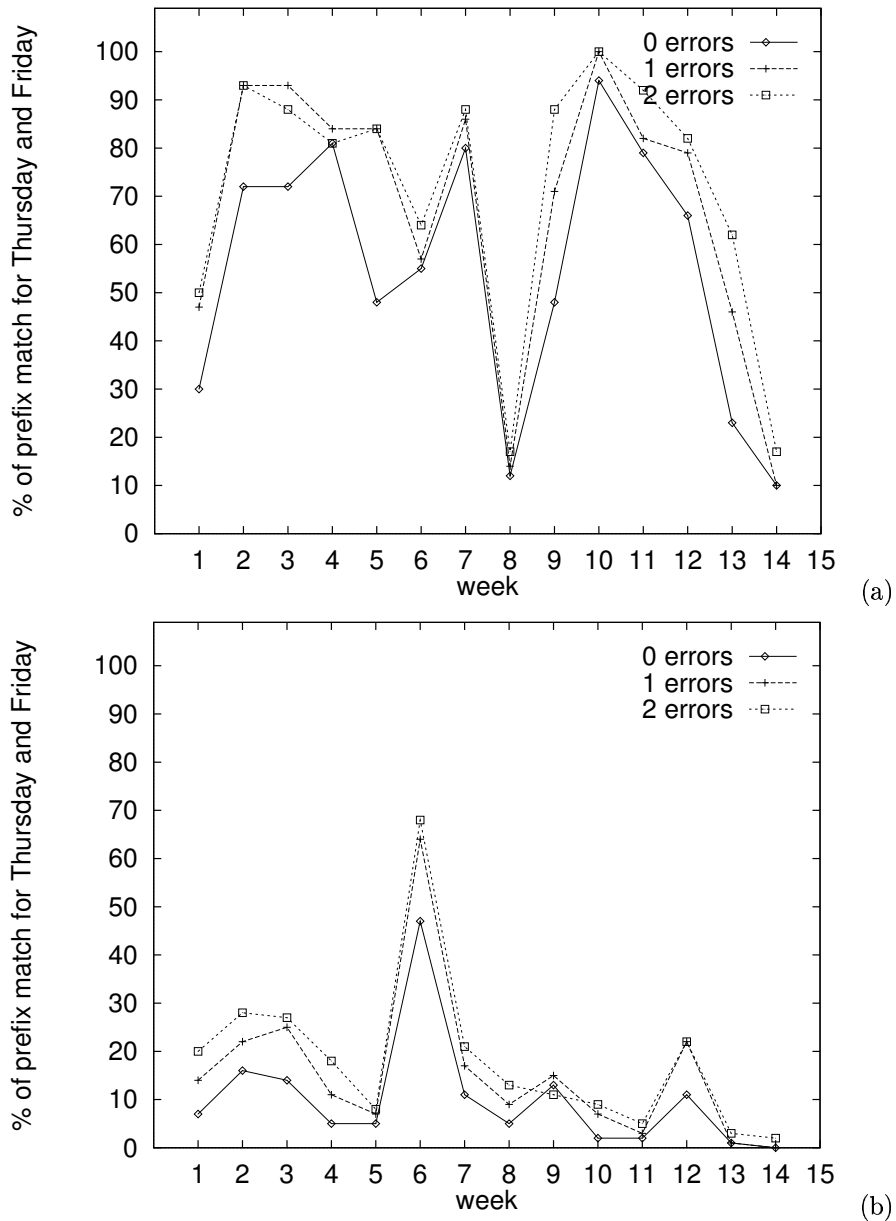


Figure 3. Percentage of prefix matches on Thursday and Friday of each week during (a) Fall2001 and (b) Winter 2002.

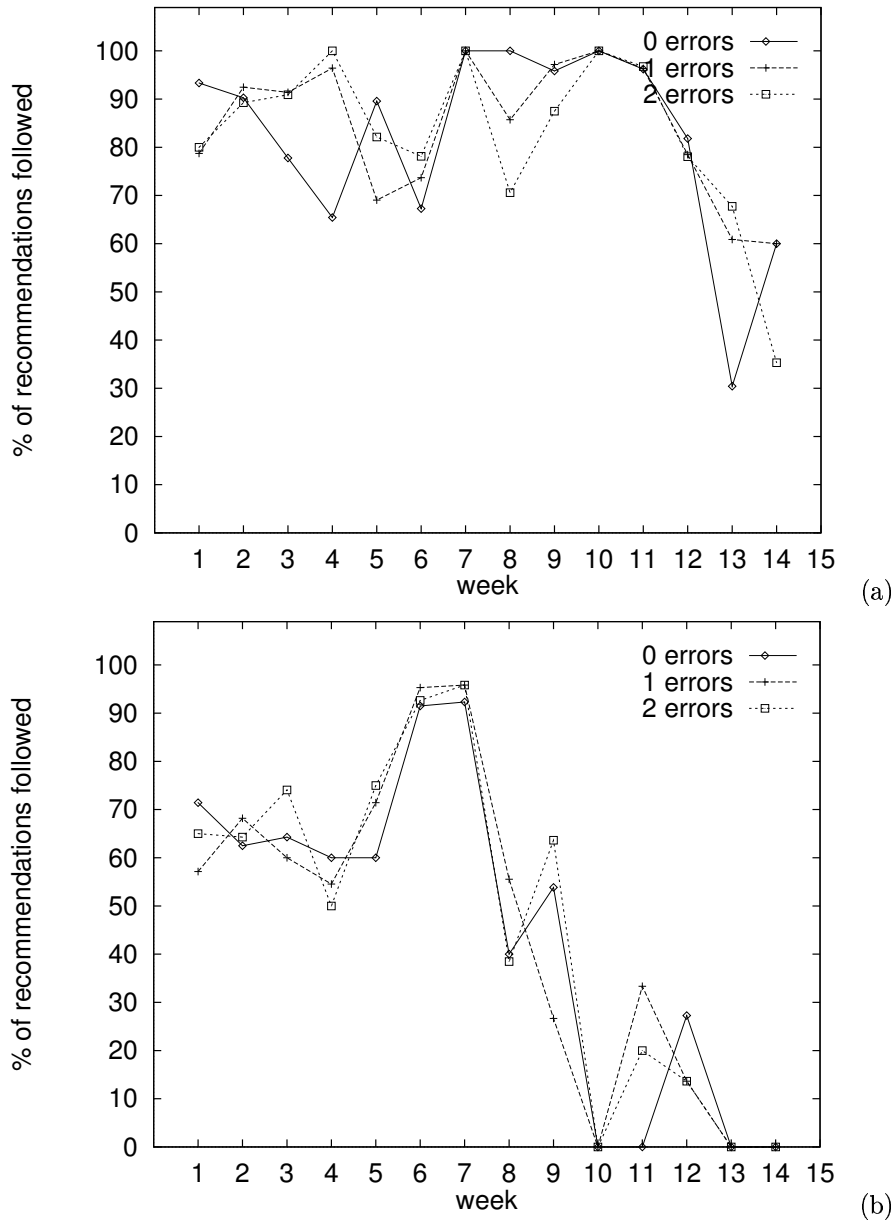


Figure 4. Percentage of effective recommendations during (a) Fall2001 and (b) Winter 2002.

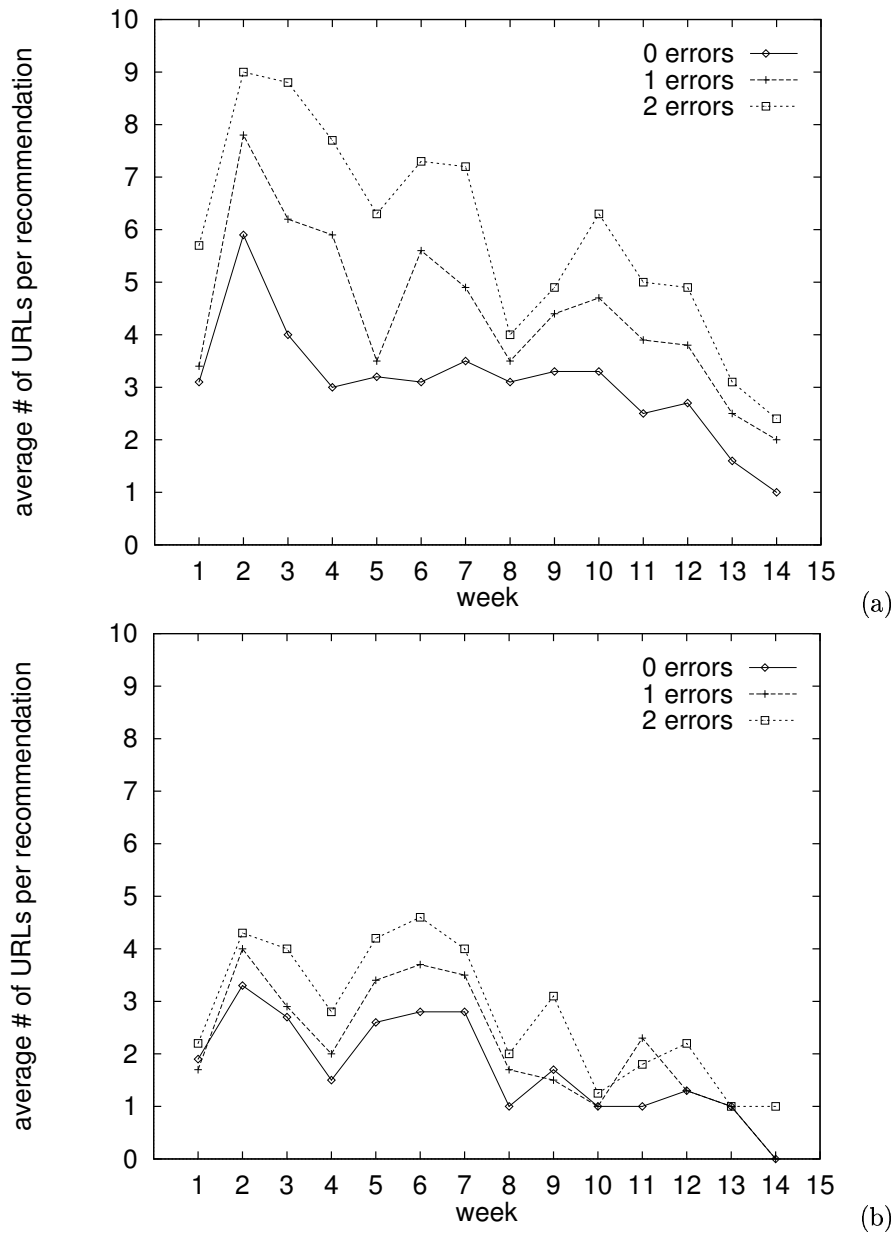


Figure 5. Average number of URLs per recommendation during (a) Fall2001 and (b) Winter 2002.