# Beyond the Usual Suspects: Context-Aware Revisitation Support

Ricardo Kawase, George Papadakis, Eelco Herder and Wolfgang Nejdl
L3S Research Center
Appelstrasse 9, 30167 Hannover, Germany
{kawase, papadakis, herder, nejdl}@L3S.de

## ABSTRACT

A considerable amount of our activities on the Web involves revisits to pages or sites. Reasons for revisiting include active monitoring of content, verification of information, regular use of online services, and reoccurring tasks. Browsers support for revisitation is mainly focused on frequently and recently visited pages. In this paper we present a dynamic browser toolbar that provides recommendations beyond these usual suspects, balancing diversity and relevance. The recommendation method used is a combination of ranking and propagation methods. Experimental outcomes show that this algorithm performs significantly better than the baseline method. Further experiments address the question whether it is more appropriate to recommend specific pages or rather (portal pages of) Web sites. We conducted two user studies with a dynamic toolbar that relies on our recommendation algorithm. In this context, the outcomes confirm that users appreciate and use the contextual recommendations provided by the toolbar.

## Categories and Subject Descriptors

H.5.4 [**Hypertext/Hypermedia**]: [User Issues]

## General Terms

Experimentation, Human Factors

## Keywords

Web behavior, Revisitation Prediction, Contextual Support

## 1. INTRODUCTION

The World Wide Web has become an important part of our lives. Search engines, travel planners, dictionaries and other online services have become essential for dealing with numerous task. News sites, portals, online games and streaming video are popular resources for information and entertainment. We communicate with our friends via email, social networking, forums, blogs and chat.

Many of these online activities are carried out on a hourly, daily, weekly or monthly basis. To facilitate them, we typically rely on known, trusted Websites that we have visited before. Web browsers support revisitation of pages and sites through mechanisms such as URL auto-completion, the forward and back buttons, bookmarks and the history sidebar. However, this support is found to be suboptimal and skewed toward a small set of frequently visited resources [26].

For this reason, the analysis and prediction of online browsing behavior and revisitation patterns has received much attention not only from the research community but also from the industry [1, 35, 20, 9, 28]. Academic research delivered several alternative history mechanisms, including gesture navigation [10], a SmartBack button that recognizes waypoints [22], a browsable SearchBar organized around a hierarchy of past queries [24] and many types of history visualizations: lists, hierarchies, trees, graphs, 2d and 3d stacks, footprints (see [21] for an overview). Browser add-ons that support users in revisiting pages and sites include Delicious[1] (social bookmarking), Infoaxe[2] and Hooeey (full-text history search), WebMynd[3] (history sidebar for search) and ThumbStrips (history visualization).

In this paper, we introduce SUPRA, a generic library for real-time, contextual prediction of navigational activity that encompasses a set of methods aligned in two tiers. The first tier ranks resources according to their likelihood of being used in the immediate future, as it is derived from their recency and/or frequency of use. The second tier, complements the ranking methods with propagation methods that identify resources that are commonly visited within the current user context.

The contextual prediction library is used as a basis for the PivotBar, a dynamic browser toolbar that recommends visited pages, relevant to the currently viewed page. The toolbar bears similarities to the concept of dynamic bookmarks [33, 25, 14]. In contrast to them, however, the recommendations of the PivotBar are contextualized and reflect the dynamics of user behavior, as they are encapsulated by the ranking methods.

We evaluated the prediction performance of the generic surfing prediction library with two datasets: one consisting of a detailed client-side log of 25 users, gathered over a period of six months, and another, more extensive one that contains anonymized usage logs that were recently collected through the Web History Repository project. The results


*HT'11,* June 6–9, 2011, Eindhoven, The Netherlands.


[1]See `http://www.delicious.com`.
[2]See `http://infoaxe.com`.
[3]See `http://www.webmynd.com`.

of the experiments indicate that taking the user context into account (i.e., combining ranking methods with propagation methods) drastically improves prediction performance. Moreover, the outcomes verify that predicting sites instead of individual pages is an easier task, thus exhibiting higher performance. The actual usage and appreciation of these recommendations has been evaluated in two user studies with the PivotBar browser add-on. The log data shows that a significant amount of revisits has taken place via the PivotBar.

The remainder of this paper is organized as follows. In the next section, we review related work on the analysis and prediction of revisitation patterns on the Web. In Section 3, we introduce the contextual browsing prediction library, together with an evaluation of its performance. The user evaluation of the actual usage and appreciation of the recommendations is discussed in Section 4. Based on the results and feedback of this user evaluation, we conducted a second experiment in which we compared the performance of page prediction with site prediction. The motivation for this experiment and the dataset used is described in Section 5; the experimental setup and results are discussed in Section 6. In Section 7, we compare the usage and appreciation of page versus site recommendations in the PivotBar. We conclude with a discussion of the results and pointers for future work.

## 2. RELATED WORK

In the first part of this section, we summarize the findings from several studies on how and why users revisit pages. In the second part, we discuss common approaches for predicting revisit patterns on the Web.

### 2.1 Studies on Web Usage and Revisitation

One of the first studies on Web usage behavior was carried out by Tauscher and Greenberg [34] in 1995. They recognized the fact that Web users often carried out recurrent tasks on the Web. Their empirical results confirmed Catledge and Pitkow's [8] finding that the links and the back button were the most frequently used methods for accessing a Web page; bookmarks and the temporally ordered history list were rarely used. They defined the *recurrence rate* to be the probability that any page visit is a repeat of a previous visit, expressing it as a percentage. An average recurrence rate of 58% was estimated for their participants; reanalysis of the data from the Catledge and Pitkow study yielded a recurrence rate or 61%.

The authors made some further characterizations of page revisits. It was found that the relation between the number of page requests and the number of unique pages visited thus far is roughly linear; the *URL vocabulary* grows linear with the number of page requests. Two important characteristics of revisited pages were described: first, most page revisits pertain to pages visited very *recently*; the probability for a page to be revisited decreases steeply with the number of page visits since the last visit. Second, there is a small number of highly *popular* pages that are visited very frequently; the probability for a page to be revisited decreases steeply with its popularity ranking.

Another long-term click-through study was carried out by Cockburn and McKenzie [10]. They observed that browsing is a *rapidly interactive* activity; the most frequently occurring time gap between subsequent page visits was around 1 second and gaps of more than 10 seconds are relatively rare. Analysis from the bookmark files revealed that most users have or will have problems with the size and the organization of their bookmark collections.

More recently, Weinreich et al. [36] carried out a long-term study in which they analyzed the interactions of 25 users with the Web browser during a period of four months and compared the results with the studies discussed above. They showed that the introduction of new browser features - such as tabbed browsing - and the change of the Web from a rather static hypermedia document repository to an interaction and transaction oriented platform, has a dramatic impact on the way users navigate the Web. Tabbed browsing has been established as a useful alternative for hub-and-spoke navigation that replaces backtracking to a significant extent.

Based on user action logs and interviews, Obendorf et al. [26] distinguished *short-term revisits* (backtrack or undo) from *medium-term* (re-utilize or observe) and *long-term revisits* (rediscover). For short-term revisits, the back button was found to be the most commonly used tool. For medium-term revisits, users normally type the page address directly into the address bar, making use of the URL completion. However, after a certain period the page is removed from the URL completion list. In these situations, if a user does not remember the exact address and if the address has not been bookmarked, she needs to rely on *waypoints*, from which a trail to the desired page can be followed. Further, the results showed that different categories of sites invite different revisit behavior: search engines and other portal sites typically have one page that users frequently return to, whereas institutional and project-related sites also comprise a long tail of pages visited several times.

Adar et al. [1] further investigated revisitation behavior, making use of a large user base collected via the Windows Live Toolbar. They found out that short-term revisits involve hub-and-spoke navigation, visiting shopping or reference sites or pages on which information was monitored. Medium-term revisits involve popular home pages, Web mail, forums, educational pages and the browser homepages. Long-term revisits involve the use of search engines for revisitation, as well as weekend activities, such as going to the cinema. A subsequent study was carried out [35], based on a merged dataset of search engine logs, Web browser logs and a large-scale Web crawl, comprising several millions of users. The results confirmed earlier findings: within-session refinding mainly involves continuing work on a task or a routine behavior, whereas cross-session revisits mainly involves re-evaluation (e.g., "Did I remember the information correctly?", "Did something change?" or "Has something new been added?").

The above observations were confirmed by Kumar et al. [20], who compared pageview categories for 'regular' revisits and long-term revisits, based on a random sample of users drawn from Yahoo! toolbar logs. The main finding of this study was that half of all pageviews are content (news, portals, games, multimedia), one-third are communication (email, social networking, forums, blog, chat) and the remaining one-sixth are search (including item search and multimedia search). Portal pages receive the largest percentage of revisits, which can be attributed to the promotion and use of homepages of - among others - Yahoo! and MSN as "entry points".

## 2.2 Prediction of Revisits

The problem of the next-page prediction has been extensively studied in the literature. The method that has prevailed in this field, at least in terms of popularity, is Association Rules Mining. *Association rules* (**AR**) constitute a well-established method for effectively identifying related resources without taking into account their order of appearance (e.g., pages that are typically visited together, in the same session, but not necessarily in the same order) [3, 4]. Numerous works have investigated the performance of different variations of AR [2, 23, 37, 13, 31]. A recent work by Kazienko [19] explores indirect AR for web recommendations, involving resources that are not 'hardly' connected, as in typical AR.

However, AR suffer from a variety of drawbacks: first, they rely on the most frequent patterns identified in the training set, thus misclassifying new patterns that are not included in it. Second, they fail to recommend rarely visited, and, thus, non-obvious and serendipitous items, since such resources never reach the minimum support limit. Third, disregarding the order of itemsets invariably leads to loss of information about the frequency of different patterns that involve the same resources (e.g., all six permutations of the itemset $I_1 = \{1, 2, 3\}$ are treated equally).

To overcome this last problem, *sequential patterns* have been employed in the context of prediction methods as well. Among them, state-based models like the Markov one, are particularly popular [39, 5, 38, 11, 12, 32, 6]. More recently, Chierichetti et al. [9] introduced a hybrid of a Markov process capturing the graph of web pages together with a branching process that captures the creation, splitting and closing of tabs. This model was then used to compare tabbed browsing with the simple PageRank model [7].

Slightly different from these models are sequence mining techniques that do not take into account the strict order between items [4, 29, 28]. A comparison of such techniques with AR was conducted by Géry and Haddad [16]. The authors evaluated AR against *Frequent Sequences* (which can be considered equivalent to association rule mining over temporal data sets) and *Frequent Generalized Sequences* (which constitute a more flexible form of the previous technique, involving wildcards [15]).

With the aim of introducing a prediction method that is equally effective with unseen data, Awad et al. [6] combined Markov Models with Support Vector Machines (SVM) under Dempster's rule. They compared experimentally their hybrid model with the individual methods comprising it, as well as with AR. The outcomes demonstrate the superiority of their model (especially when domain knowledge is incorporated into it). Although this is a considerable step toward a method with better generalization capabilities, it is far from being practical: it requires a different SVM classifier for each one of the available resources and a considerably high training time (in fact, their experimental study involved 5,430 classifiers and 26.3 hours of training for a single dataset).

In a more recent work by Parameswaran et al. [28], the authors coin *precedence mining* and build a suite of recommendation algorithms based on it. They model a users' history as a set of items having co-occurred in the past (without considering their order of appearance), and predict the set of items most likely to follow in no particular order and not necessarily in the next action of the user. Though quite interesting, their approach is not crafted to deal with the next-page prediction problem, as they explicitly point out.

# 3. CONTEXTUAL REVISIT PREDICTION

In this section, we explain the methods and algorithms used for generating contextual predictions of revisits on the Web. The prediction task can be more formally defined as follows:

**Problem** 1. *[Page Revisition Prediction] Given a collection of Web Pages, $P_u = \{p_1, p_2, ...\}$, that have been visited by a user, u, during her past n page requests, $R_u = \{r_1, r_2, , r_n\}$, rank them so that the ranking position of the* ***page*** *revisited in the next, $n + 1$, transaction is the highest possible.*

We developed a generic framework that consists of *two tiers of methods*. The first tier involves usage-based *ranking methods*, which estimate for each web page the likelihood that it will be accessed in the next request. The methods derive their estimate from evidence drawn from the surfing history of a web site or user, such as the recency and/or the frequency of accesses to each page. The second layer covers *propagation methods*; these are techniques that capture repetitiveness in the navigational activity of a Web user and identify groups of pages that are typically visited together (in the same session, but not necessarily in a specific order). Depending on the degree of connectivity between the associated Web pages, their values (assigned by the ranking methods) are then propagated to each other.

The implementation of the framework, *SUPRA*[4], is freely available at SourceForge[5] In this way, we encourage other researchers to experiment with them and to extend the library with new ranking and propagation methods. Special care has been taken to make this a straightforward procedure: any implementation complying with Definitions 2 and 3, which specify the minimal requirements for a ranking and a propagation method respectively, can be easily integrated into the library.

In the next subsections, we discuss the ranking and propagation methods we considered, and how they are combined. We conclude this section with the results of an experimental evaluation of the framework.

## 3.1 Ranking Methods

The aim of ranking methods is to provide for each item a numerical estimation of the likelihood that it will be accessed in the next transaction. After each page request, the selected ranking method goes through all visited items of interest (either pages or sites), estimates their value and sorts them in descending order of their expected value. The estimation is based on the access history of each item, represented by the indices of the related requests:

**Definition** 1. *[Item Request Indices] Given the page requests $R_u$ of a user u, the request indices of an item $m_i$, $I_{m_i}$, is the set of the serial numbers of those requests in $R_u$ that pertain to $m_i$. The serial number of the chronologically first request is 1 and is incremented by 1 for each of the subsequent page visits.*

[4]SUPRA stands for SUrfing PRediction frAmework.
[5]See `http://sourceforge.net/projects/supraproject`.

Given this definition, a ranking method is defined as follows:

**Definition** 2. *[Ranking Method] A ranking method is a function that takes as input the request indices of an item $m$, $I_{m_i} = \{i_1, i_2, \ldots, i_k\}$ together with the index of the latest request, $i_n$, of the respective user, and produces as output a value $v_{m_i} \in [0,1]$ that is proportional to the likelihood of $m_i$ being accessed at the next page request, $r_{n+1}$ (i.e., the closer $v_{p_i}$ is to 1, the higher this likelihood).*

In this work, we consider the following ranking methods (modified appropriately to be consistent with Definition 2):

1. Least Recently Used (**LRU**),

$$LRU(m_i, I_{m_i}, i_n) = \frac{1}{i_n - i_k + 1},$$

2. Most Frequently Used (**MFU**),

$$MFU(m_i, I_{m_i}, i_n) = \frac{|I_{m_i}|}{i_n},$$

3. Polynomial Decay (**PD**),

$$DEC(m_i, I_{m_i}, i_n) = \sum_{j=1}^{|I_{m_i}|} \frac{1}{1 + (i_n - i_j)^{\alpha}}, \ \alpha > 0$$

where $i_k$ is the index of the chronologically last transaction in $I_{m_i}$, $i_n$ is the index of the latest request of the system or user, and $|I_{m_i}|$ is the cardinality of $I_{m_i}$.

The first two methods, LRU and MFU, constitute well-established caching algorithms that are typically employed in prediction tasks. LRU is based on the assumption that the longer in the past a page was accessed, the less likely it is be accessed in the future. Similarly, MFU assumes that the more frequently a page is accessed, the more likely it is to be accessed in the next request. Thus, the former orders items according to the recency of their last request, whereas the latter sorts them in descending order of their popularity. PD, on the other hand, is based on the *decay ranking model* introduced by Papadakis et al. [27]. It incorporates recency and degree of usage into a single, comprehensive method, balancing them harmonically through the smooth decay of the contribution of each request to the total value of an item. Factor $a$ is available for tuning this equilibrium, by defining the intensity of the decay: values larger than 1 convey a steeper decay, which puts more emphasis on recency, while values close to 0 promote frequency of usage. In general, the best value for $a$ depends on the application at hand, but, as verified in [27], values between 1.0 and 2.0 provide performance close to the optimum, outperforming both LRU and MFU.

## 3.2 Propagation Methods

The purpose of propagation methods is to capture contextual information through the detection of patterns in the surfing activity of users. They identify those items that are commonly visited within *the same session* and associate them with each other. The 'links' created by these methods are used to propagate between the associated pages the values assigned to them by the ranking methods. In this way, the highest the value of a web page, the more the pages associated with it are boosted and the more their ranking is upgraded.

Sessions are transparently defined by browsers, and typically include all pages visited within the same tab of the browser for up to a specific time period. The temporal limit, though, can vary from browser to browser, and, thus, we do not provide a formal definition of a session. Instead, we consider a **session** $S$ to be a bag of visited items, defined by the browser, that are placed in chronological order from the earlier to the latest: $S = \{m_1, m_2, \ldots, m_k\}$.

Based on the above, propagation methods can be formally defined as follows:

**Definition** 3. *[Propagation Method] A propagation method is a function that takes as input the latest requested item, $m_i$, within a session $S$, and defines appropriately the degree of connection between $m_i$ and all other items visited during $S$. Hence, given two items, $X$ and $Y$, it returns a value $v_{XY} \in [0,1]$ that is proportional to the likelihood of $Y$ being accessed immediately after $X$ (i.e., the closer $v_{XY}$ is to 1, the more likely this transition is).*

We distinguish between two families of propagation methods: *order-neutral* methods, which disregard the order of transactions within a session and *order-preserving* methods, which take this order into account. For the former case, we examine association matrices. For the latter case, we consider transition matrices.

**Order-Neutral Propagation Methods.** Order-neutral methods are based on the idea that pages visited in the course of the same session should be equally connected with each other, regardless of their order and the number of transitions that intervene between them. The rationale behind this idea is that users may visit a group of pages $X, Y, Z$ on a regular basis, but not necessarily in that order.

We employed association matrices (**AM**) for order-neutral propagation. An AM is a matrix, whose rows and columns are the given set of web pages $P$. The AM is built by associating all pages visited in a single session with each other (i.e., each web page is connected not only with the pages preceding it, but also with those following it). Thus, an AM is always a symmetrical matrix ($\forall x \ AM(x,x) = 0$) and each cell $AM(x,y)$ expresses the number of sessions that involve both items $x$ and $y$.

**Order-Preserving Propagation Methods.** This category of propagation methods relies on the idea that pages are typically accessed in the same or similar order. Order-preserving methods build the associations between pages according to this ordering: each page is connected only with pages preceding it. To capture these transitions that form chronological patterns in the navigational activities of systems and users, we employ transition matrices.

In short, a transition matrix (**TM**) is a matrix with its rows and columns representing all pages visited by the user. Each cell $TM(x,y)$ expresses the number of times that a user visited item $y$ *directly after* $x$. Given that a transition matrix respects the order of accesses within a session, it is not symmetrical: $\exists x, y : \ TM(x,y) \neq TM(y,x)$. Moreover, its diagonal cells are all equal to 0: $\forall x \ TM(x,x) = 0$.

We conducted a series of experiments to identify which propagation method produces the best results for our problem [18]. Together with the order-neutral AM, we evaluated four kinds of order-preserving propagation methods. *Simple Connectivity TM* (**STM**): after each transition $x \rightarrow y$, only the value of the cell $TM(x,y)$ is incremented by one, thus functioning exactly like a first-order Markov model.

*Continuous Connectivity TM* (**CTM**): each web page visited within the current session is associated with all previously accessed pages. *Decreasing Continuous Connectivity TM*(**DTM**): this strategy lies in the middle of STM and CTM; the cell values are determined based on a decay parameter representing the distance between page visits. *Increasing Continuous Connectivity TM*(**ITM**): this strategy increases the value added to $TM(x, y)$ in proportion to the distance between pages visits.

Of the above methods, the *Simple Connectivity Transition Matrix* (**STM**) produced the best results, which provides support to the assumption that page revisits tend to occur in the same strict order. It is worth noting at this point that STM was also employed in Awad et al. [6], but its frequencies were merely used as features for a classification algorithm. It was also employed in [30] as a means to model the behaviour of individual users and to recommend relevant items to users by combining their matrices.

## 3.3 Combining Ranking with Propagation

To combine a raking method with one of the propagation techniques, we employ a simple, linear scheme: following the $i_n$-th page request, the value of all items is (re)computed, according to the selected ranking method. Then, for each non-zero cell of the TM (or AM) at hand, $TM(x,y)$ (or $AM(x,y)$), we increment the value assigned to page $y$ by the ranking method, $v_y$, as follows:

$$v_y + = p(x \to y) \cdot v_x, \text{ where}$$

- $p(x \to y)$ is the transition probability from item $x$ to item $y$, estimated by $p(x \to y) = \frac{TM(x,y)}{\sum_i^{in} TM(x,i)}$ (or $p(x \to y) = \frac{AM(x,y)}{\sum_i^{in} AM(x,i)}$), and
- $v_x$ is the value of $x$ estimated by the ranking method.

## 3.4 Experimental Study on Page Prediction

**Setup.** To evaluate our framework, we conducted an experimental study using data from a client-side Web usage log of 25 users with a total of 137,737 page requests, gathered in the course of 6 months [6]. The participant pool of the data set consists of 25 participants, 19 male and 6 female. Their average age is 30.5, ranging from 24 to 52 years.

We simulated the navigational activity of each user independently of the others. After each page request, the ranking of all visited pages was updated, and, in case the next access was a revisitation, the position of the corresponding web resource was recorded. Having all these ranking positions for all prediction methods, we derived the following metrics to evaluate their performance:

1. *Success at 1* (**S@1**). It denotes the portion of revisitation requests that involved the page placed at the first ranking position by the prediction method. The higher its value, the better the performance of the method. S@1 is interesting as it provides evidence for the accuracy of a prediction method in identifying the next revisited page.
2. *Success at 10* (**S@10**). It expresses the portion of revisits placed in one of the first 10 places. The higher its value, the better the performance of the method. S@10 expresses the actual usability of the prediction method, as users typically have a look only at the first 10 pages presented to them (just like they do with web search engine results [17]).
3. *Average Ranking Position* (**ARP**). It represents the average position of a revisited page in the ranking list that the prediction method produces. ARP provides, thus, an estimation of the overall performance of a prediction method, as it considers the performance over all the revisits in the navigational history of a system or user, and not only the top ranked ones. The lower its value, the better the performance of the prediction algorithm.

[6]This is the data set that was used in [26, 36]

| Method | ARP | S@1 | S@10 |
|---|---|---|---|
| **MFU** | 307 ($\sigma$=178) | 12.7 ($\sigma$=3.8) | 32.2 ($\sigma$=5.4) |
| **LRU** | 65 ($\sigma$=30) | 19.3 ($\sigma$=3.8) | 71.2 ($\sigma$=4.3) |
| **PD** | 60 ($\sigma$=27) | 19.3 ($\sigma$=3.8) | 71.7 ($\sigma$=4.2) |
| **MFU+STM** | 288 ($\sigma$=168) | 12.6 ($\sigma$=3.8) | 32.1 ($\sigma$=5.4) |
| **LRU+STM** | 32 ($\sigma$=14) | 23.8 ($\sigma$=3.7) | 81.5 ($\sigma$=2.8) |
| **PD+STM** | 31 ($\sigma$=14) | 22.7 ($\sigma$=3.3) | 81.8 ($\sigma$=2.8) |

**Table 1: Summary of Experimental Results**

**Results.** We compared the performance of the ranking methods LRU, MFU and PD by simulating these methods on the dataset described earlier in this section. Similarly, we evaluated the three ranking methods when combined with the propagation method STM[7]. We configured the steepness of the PD decay model with $\alpha = 1.5$. The results are summarized in table 1.

Of the ranking methods, MFU performs much worse than LRU and PD with respect to all metrics. This indicates that backtracking is more common than revisiting popular sites; moreover, frequent revisits to popular sites are largely covered by the list of recently used pages. LRU and PD have similar performance in terms of S@1 and S@10, but PD has a slightly better ARP, due to the incorporation of the frequency of usage. Their combination with the propagation method STM takes into account the current user context, as well, thus improving significantly the performance of LRU and PD ($t(24)=10.1$, $p<0.01$); combining MFU with STM hardly causes any change. This can be explained by the interaction between the recency effect and the current user context. The S@10 of the combined ranking and propagation methods performs up to 81.8%.

# 4. USER EVALUATION OF CONTEXTUAL RECOMMENDATIONS

To explore the actual usage and appreciation of our prediction framework, we developed the *PivotBar*, a browser toolbar that looks quite similar to the bookmark toolbar, containing favicons and links to already visited pages (see Figure 1). In contrast to the bookmark toolbar, however, PivotBar is dynamic, providing contextual recommendations; after each navigation action or tab change, the list of pages in the bar changes, containing the most relevant visited pages to the current one.

The design of the toolbar is kept minimalistic, in order to avoid occupying a large part of the browser's interface.

[7]The results using the other propagation methods were lower than the STM results, therefore we left them out of the discussion.

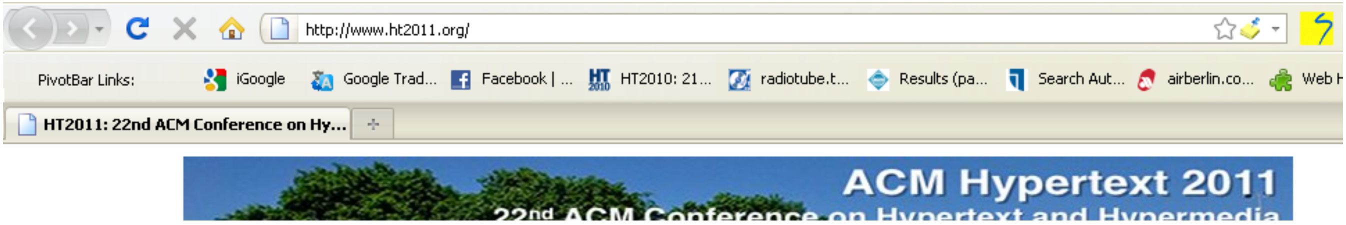


**Figure 1: PivotBar recommendations**

By placing it right under the URL field, we ensure that the dynamic character of the list catches user's attention only in the periphery and just for a short time period - unless the user chooses to follow a recommendation.

For the first implementation of the PivotBar, we chose Mozilla Firefox as the host browser, since it constitutes a freely available and platform-independent browser that provides developers with clear-cut documentation and transparent access to client data. The PivotBar Add-On makes use of the existing user history in the browser database and all computations take place on the client-side.

It is worth clarifying at this point that PivotBar is not designed for extensive search into the history - an activity that users hardly undertake anyway. Instead, it exclusively aims at reminding users of past visits that are judged relevant to the currently viewed page. For example, when planning a train-ride, the user will be prompted to visit his favorite hotel booking site, if she had done so in a similar situation in the past.

## 4.1 Diversity of Recommendations

At the core of our toolbar lies a composite prediction method that employs PD for ranking web pages and STM for propagating their values (see Section 3.1). The reason for this choice is twofold: first, these methods have exhibited the highest performance in their category, not only individually, but also in combination. Second, PD (and subsequently the propagation method on top of it) provides the best trade-off between the diversity and the relevance of the recommendation sites.

To verify the latter claim, we compared the average entropy of the top-10 recommendations, as generated by the ranking methods of Section 3.1, making use of a dataset consisting of 116 users with an average of 960 revisitations per person (see Section 5.2.1). The average entropy was estimated to be 4.2 for MFU, 7.9 for PD and 8.8 for LRU. This means that these methods recommend, on average, 18 (MFU), 240 (PD) and 445 (LRU) distinct pages. In contrast to the rather static nature MFU, LRU provides more diverse recommendations - but these pages are already accessible through the back button. In the middle of these two extremes lie the recommendations of PD.

## 4.2 Study Setup

The goal of our user study was to get an answer to the following questions: first, will users actually click on recommendations? In other words, will the toolbar be used? Second, what would be the user's appreciation of a dynamic toolbar? Third, which could be the directions for further improvement of the recommendations?

To this end, we asked 11 participants, aged 28 on average, to install the toolbar, either on their business computer or on their private one. Eight opted for the former choice, and the remaining three for the latter. Users were then provided with a brief introduction to the tool and some instructions for the experiment[8]. The participants were asked to keep the tool installed for a period of five working days. With the passage of this period, we collected the the quantitative results through the click-data of each participant, while qualitative feedback was elicited via an open-ended interview.

| User | Total Visits | Revisits | PivotBar | Percent |
|---|---|---|---|---|
| **1** | 541 | 264 | 104 | 39.4% |
| **2** | 596 | 248 | 38 | 15.3% |
| **3** | 352 | 147 | 49 | 33.3% |
| **4** | 828 | 424 | 49 | 11.6% |
| **5** | 321 | 63 | 10 | 15.9% |
| **6** | 567 | 283 | 39 | 13.8% |
| **7** | 259 | 137 | 20 | 14.6% |
| **8** | 179 | 102 | 40 | 39.2% |
| **9** | 183 | 75 | 19 | 25.3% |
| **10** | 312 | 149 | 14 | 9.4% |
| **11** | 423 | 145 | 46 | 31.7% |

**Table 2: Click data during the evaluation period.**

## 4.3 Results

All participants claimed to use the computer for about 6 to 8 hours per day. They all indicated that they typically use the auto-completion feature for revisitation, while half of them actively uses bookmarks, as well. Further, they acknowledged that they often use search engines to refind a known page. The recurrence rate during the evaluation period reached an average of 44.2% ($\sigma$=10.4), lying at the same levels indicated by previous studies [10, 26].

Table 2 summarizes the usage of the PivotBar for each participant. The second column indicates the total number of pages visited. The third column represents the number of revisits among the page requests (including requests for pages visited before the start of the evaluation period). The fourth column corresponds to the number of revisits that were initiated through the PivotBar. The fifth column shows the percentage of revisits covered by the PivotBar.

The average percentage of revisits through the PivotBar was 22.7% ($\sigma$=11.4), reaching a peak of 39.3% for participant 1. This number is surprisingly high - even if one takes the novelty effect into account. As a comparison, [26] observed that the back button covered 31% of all revisits, while bookmarks, the history list and the homepage button together were responsible for a mere 13.2% of all revisits.

[8]The exact instructions given to the participants were: "PivotBar automatically generates suggestions based on the current page you are accessing. You can use them simply by clicking on a link to be redirected to the target page. Feel free to use them or not."

Quite interesting was the qualitative feedback that we received via the open-ended interviews. When asked about the usage of the toolbar, one of the participants explicitly commented: "*I actually scan the shortcuts automatically when they change. The movement attracts my attention, without being distractive*". Another participant said: "*It's nice that I can see the pages that I usually access*". At the same time, though, he admitted that his routine behavior was hard to change: he still tended to automatically open a new tab and directly type the address of a page using auto-complete. This explains why for some users the usage percentage of PivotBar remains low, around 10%.

The participants also provided suggestions for further improvements. Some of them proposed to further reduce the influence of recency on the recommendations, favoring more serendipitous ones. Others thought that recommendations should be based on the currently visited site instead of the page (*site-level recommendations*). Finally, quite a few participants argued that the toolbar should recommend (portal pages of) sites instead of (specific) pages.

The comments about the preference for site-level recommendations can be explained by the growing importance of revisits to service-oriented sites and the monitoring of news sites [1]. However, site-level recommendations would ignore the informational value of specific news articles, blogs and other listings. News portals, on the other hand, continuously add new articles, which cannot be covered by a revisitation prediction method.

# 5. RECOMMENDING PAGES VS SITES

The results of the user evaluation in the previous section show that, when users are provided with relevant suggestions for page revisits, they will click on these suggestions. In the evaluation, the PivotBar recommended pages based on the currently visited page. Participant feedback suggested that it might be even more beneficial to provide suggestions for (portal pages of) Web sites instead of individual Web pages - or to use the currently visited site (not the specific page) as a basis for the prediction.

We address these suggestions with a second experiment and user evaluation in the following sections. In this section, we formalize the site prediction task and introduce the dataset used for the second experiment.

## 5.1 Site Revisitation Prediction

For clarity, we start the discussion of our experiments with a couple of definitions. We consider as a *Web Site* a domain that comprises a set of *Web Pages*. For instance, `http://www.ht2011.org/tracks.html` is a page under the site `http://www.ht2011.org`. In the following, we consider each page to contain in its description, the URL of the corresponding Web Site.

Similar to Problem 1, the task of recommending sites is defined as:

**Problem** 2. *[Site Revisition Prediction] Given a collection of Web Pages, $P_u = \{p_1, p_2, ...\}$, that have been visited by a user, u, during her past n page requests, $R_u = \{r_1, r_2, , r_n\}$, rank them so that the ranking position of the **site** revisited in the next, $n + 1$, transaction is the highest possible.*

In the following sections, we present two approaches to this problem, together with a new one for Problem 1.

## 5.2 Dataset

In order to verify the efficiency and performance of the new predicting methods, we started an effort to gather users' navigational data through the Web History Repository[9] (**WHR**).

### 5.2.1 Web History Repository

The Web History Repository Project aims to build a public repository of web usage data, which can be used by researchers to gain new insights in online browsing behavior. Using a Mozilla Firefox add-on[10], users can upload their anonymized usage data to the server. These data include the list of visited pages together with the timestamp and browser session of each request. A separate table stores for each visited page its (encrypted) URL and host, the total number of visits, the frecency and the last visit.

The Web History Repository was promoted through several targeted mailinglists, Facebook, Blogspot and Twitter. One month after the release of the add-on, more than 100 anonymous volunteers contributed over 1 million entries from their browser history. At this point, we considered the dataset large enough to give us significant results for our experiments. In contrast to the dataset of Section 3.4, the data of WHR are totally anonymized, and, thus, we do not have at our disposal any demographic information about the users.

### 5.2.2 Characteristics and Analysis

The dataset we used contained the navigational history of 116 users with a total of 1,006,941 page visits. The user with the largest history contributed exactly 6 months of data with 77,398 page visits. The average time period of the history for all users was 56 days. We pruned the data to remove users with less than one thousand visits. The remaining dataset consisted of 61 users and a total of 951,995 page visits, still representing 94.5% of the dataset.

For each of the selected users, the average number of page visits is 15,606 ($\sigma$=18,893), in a period of 87 days ($\sigma$=82). This corresponds to an average of 179 pageviews per day. The average recurrence rate is 34% ($\sigma$=14), slightly lower than the recurrence rate in other studies. By contrast, the recurrence rate per host (the relative number of visits to a site that constitute a revisit to this site) is astonishingly high at 92% ($\sigma$=5, min=69%, max=99,9%). This implies that only 8 out of every 100 pages we visit on the Web belong to new, unseen domains; in other words, Web use is mainly restricted to a more or less fixed set of sites that provide the services or information that the user needs. Figure 2 illustrates the linear growth of page visits, unique pages and hosts (domains) in the dataset. Table 3 provides a comparison with the datasets used in previous studies.

# 6. EXPERIMENTAL STUDY ON SITE PREDICTION

To re-evaluate our method, we ran a second experiment following the same procedure as in the first experiment, which we described in Section 3.4. We used the pruned dataset of the Web History Repository. For this experiment we did not vary the prediction method, but employed the

[9]See `http://webhistoryproject.blogspot.com`.
[10]https://addons.mozilla.org/en-US/firefox/addon/226419

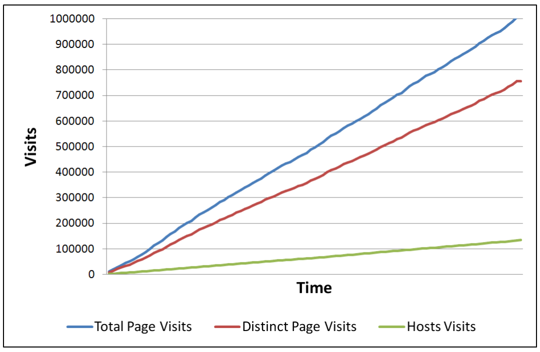


**Figure 2: Growth of page visits over time.**

| | Catledge & Pitkow | Tauscher & Greenberg | Dataset 1 [26, 36] | Dataset 2 (WHR) |
|---|---|---|---|---|
| Time of study | 1994 | 1995-1996 | 2004-2005 | 2010 |
| No. of users | 107 | 23 | 25 | 61 (of 116) |
| Length (days) | 21 | 35-42 | 52-195 | 1-385 |
| No. of visits | 31,134 | 84,841 | 137,272 | 951,995 |
| Recurrence | 61% | 58% | 45.6% | 35.9% |
| Back | 35.7% | 31.7% | 14.3% | ~7.5% |

**Table 3: Comparison of the datasets with previous studies.**

best-performing method from the first experiment: PD+STM. Instead, we varied the basis for the contextual prediction (page or site) and the type of suggestions (page or site). The following four strategies were considered:

- Page to Page recommendation (as in Experiment 1)
- Page to Site recommendation
- Site to Page recommendation
- Site to Site recommendation

## 6.1 Evaluation Measures

Similar to the first experiment, the evaluation measures used are S@1, S@10 and the Average Ranking Position. Further, in order to investigate differences in prediction performance between users, we used a number of measures to characterize their individual behavior. The first measure is the (page) recurrence rate [34]:

$$R = (1 - \frac{individual\ pages\ visited}{total\ page\ visits}) \times 100\%$$

The site recurrence rate is defined analogously to the page recurrence rate. A further measure we used was the *page* and *site entropy*, which characterizes the variance (or disorder) in the user's log:

$$E = \sum_i (p_i \times log_2(p_i)),$$

where $p_i$ is the probability of page/site i estimated as $p_i = (|I_{p_i}| - 1)/(\sum_i (|I_{p_i}| - 1))$.

The other measures we used are fairly straightforward: the average number of pages visited per site, per day and per session.

## 6.2 Results

The results of the four prediction strategies are summarized in Table 4. A first observation is that the page-to-page prediction results for this dataset are considerably lower

| Method | ARP | S@1 | S@10 |
|---|---|---|---|
| Site-to-Page | 285($\sigma$=166) | 5.0($\sigma$=2.1) | 46.2($\sigma$=8.0) |
| Page-to-Page | =80) | 15.3($\sigma$=6.9) | 61.6($\sigma$=5.9) |
| Site-to-Site | 22($\sigma$=12) | 20.9($\sigma$=4.0) | 78.0($\sigma$=4.1) |
| Page-to-Site | 23($\sigma$=12) | 33.9($\sigma$=5.6) | 79.4($\sigma$=4.2) |

**Table 4: Summary of Experimental Results**

(S@10=61.6) than for the dataset used in the first experiment (S@10=81.8). We attribute this to the larger variance in user behavior due to the way the dataset was created. Further, the S@k measures suggest that site predictions are more successful than page predictions. In addition, looking at the ARP, the average ranking position is much lower for sites than for pages. This effect can be explained by the fact that there are far less candidate sites to predict than candidate pages, which makes site prediction a safe fallback alternative for page prediction. It is also clear that page and site predictions alike perform better if they are based on the current page that the user visits instead of the current site. Finally, the differences in performance of the four strategies between individual users are highly correlated with p<0.01 (Pearson, 2-tailed), which implies that for users for whom one strategy performs well, other strategies will perform well too.

**Individual differences.** It is a likely assumption that individual differences in prediction performance are caused by differences in the user's online browsing behavior. For each user, we captured the browsing behavior in the measures introduced earlier in this section. We carried out stepwise linear regression to find out which aspects perform best in predicting the performance of page-to-page and page-to-site recommendation (in terms of S@10). The results indicate that the site entropy is the most important predictor, accounting for 22% of the variation in page prediction and 53% of the variation in site prediction; the page entropy explains another 9% of the variation in site prediction. Surprisingly, the page and site recurrence rates (which indicate to what extent a user revisits pages) are only weakly correlated to the prediction performance as well as to the entropy measures. In summary, the results indicate that it is not the amount of revisits, but the variance in revisit behavior that directly impacts the performance of any prediction algorithm.

# 7. SECOND USER EVALUATION

To evaluate the new methods with respect to real users, we carried out a second user evaluation with the PivotBar, as introduced in section 4. We modified the underlying methods so that users get a combined set of recommendations of pages and sites. For this, the following heuristic was used: if a recommended page has been visited less than 10 times before, the recommendation is replaced by the portal page of the recommended page's site, on the condition that this portal page has been visited before. The threshold of 10 is derived from the average distribution of page visits, which approximately defines the end of the head. In addition, following the suggestions of participants from the first evaluation, we added a new feature to the PivotBar that allows users to permanently hide a recommendation by adding a page or a site to a blacklist.

As in the first evaluation, our goal is to check the usability of the tool and whether the recommendations have an impact on users' navigational behavior. We evaluated this with respect to two evaluation measures: first, we observed the number of revisits triggered by clicks on the PivotBar. Second, we estimated the number of "*blind hits*"; that is, revisits that were not triggered by the PivotBar, but that were in the list of recommendations displayed in the toolbar.

## 7.1 Evaluation Setup

The setup for this evaluation was similar to the evaluation presented in Section 4. This time we had a total of 13 participants, aged 29 on average. Eight participants had the PivotBar installed at their work computers, the other five at their private computers. The instructions for using the tool were the same as before, with additional details about the functionality of the blacklist The participants were asked to keep the tool installed at least for a period of ten days. After this time period, we collected the click-data of each participant for the quantitative results; qualitative feedback was elicited through open-ended interviews.

## 7.2 Results

Table 5 summarizes the usage of the PivotBar for each participant. The second column indicates the total number of pages visited during the evaluation period. The third column represents the recurrence rate among the page requests (including revisits to pages visited before the evaluation). The fourth column shows the percentage of revisits triggered by the PivotBar and finally, the fifth column shows the percentage of blind hits.

| User | Total Visits | Revisit % | PivotBar % | BlindHits % |
|---|---|---|---|---|
| **1** | 603 | 50.1 | 30.8 | 22.8 |
| **2** | 535 | 45.0 | 19.5 | 51.0 |
| **3** | 445 | 39.6 | 15.9 | 8.5 |
| **4** | 578 | 51.2 | 15.9 | 15.9 |
| **5** | 1111 | 36.1 | 13.0 | 20.7 |
| **6** | 716 | 45.5 | 12.3 | 28.8 |
| **7** | 1219 | 49.1 | 8.8 | 18.0 |
| **8** | 899 | 41.7 | 8.8 | 8.5 |
| **9** | 379 | 56.2 | 7.0 | 11.7 |
| **10** | 1047 | 39.6 | 5.8 | 16.1 |
| **11** | 1089 | 43.3 | 4.7 | 7.6 |
| **12** | 674 | 29.4 | 11.1 | 6.6 |
| **13** | 896 | 34.6 | 3.9 | 19.0 |

**Table 5: Click data during the evaluation period.**

On average, 12.1% ($\sigma$=7.3) of all revisits resulted from a click on the PivotBar, reaching a peak of 30.8% for participant 1. The average percentage of blind hits was 18.1% ($\sigma$=12.0), meaning that these revisits were suggested in the PivotBar but not triggered by it. The strong correlation between the PivotBar clicks and blind hits (r=0.92, $p<0.01$) suggest a direct connection between the quality of recommendations and the take-up of the tool.

A further indicator of engagement is provided by the usage of the blacklist. The average number of removed pages per user was 7.2 ($\sigma$=14.3). Participant 10 had a total of 52 pages and hosts in her black list, while 3 other participants had an empty blacklist. However, the usage rate of the PivotBar for participant 4 (15.9%), who had an empty blacklist, was above the average and much higher than the engaged Participant 10 who pruned her results.

During the open interviews, all participants stated that the PivotBar was indeed useful, with few complains about visual issues due to compatibility with a specific operational system. When asked for what reasons they considered PivotBar to be useful, answers included the following: "*Because for the pages that were good suggestions, I didn't need to start typing the URL*", "*It was faster for reaching the pages I wanted*" and "*It was easier to remember pages that I have visited*".

None of the participants noticed that sometimes a specific page was recommended and sometimes the website. At the same time, we also did not receive any remarks that recommendations for very specific pages could better point to the associated site's portal page (which was one of the main remarks during the first experiment). We consider this lack of remarks as positive feedback.

# 8. CONCLUSION

In this paper, we introduced a generic framework for contextual prediction of revisits. The framework consists of two tiers of methods: ranking methods, which rank resources based on the recency and/or frequency of access to this resource, and propagation methods, which detect items that are commonly visited together with the currently visited resource.

Experimental evaluation shows that combining ranking methods with propagation ones drastically improves performance. In a second experiment, we found that site prediction is less complicated than page prediction, and that the performance of a prediction strategy mainly depends on variance in the users' online behavior (in particular, the page and site entropy). The best-performing prediction strategy has been put into practice in the context of a dynamic browser toolbar, the PivotBar. Two user studies with the PivotBar confirm that users appreciate and use the contextual recommendations provided by the toolbar. In addition, the log data shows that a significant amount of revisits has taken place via the PivotBar.

We see several directions for future work. First, the optimal value for $a$, which determines the balance between recency and frequency in the Polynomial Decay ranking method, was determined based on server-side data; verification on client-side user data might yield other results. Further, experimentation on the balance between recommendations for pages and sites may lead to better heuristics. Currently, the PivotBar only provides page and site recommendations. We plan to extend this functionality with contextual recommendations for past queries, tags and keywords.

# 9. ACKNOWLEDGEMENTS

This research has been co-funded by the European Commission within the eContentplus targeted project OpenScout, grant ECP 2008 EDU 428016, as well as the STELLAR NoE in Technology-Enhanced Learning.

# 10. REFERENCES

[1] E. Adar, J. Teevan, and S. T. Dumais. Large scale analysis of web revisitation patterns. In *CHI*, pages 1197–1206. ACM, 2008.

[2] G. Adomavicius and A. Tuzhilin. Using data mining methods to build customer profiles. *IEEE Computer*, 34(2):74–82, 2001.

[3] R. Agrawal, T. Imielinski, and A. N. Swami. Mining association rules between sets of items in large databases. In *SIGMOD Conference*, pages 207–216. ACM Press, 1993.

[4] R. Agrawal and R. Srikant. Mining sequential patterns. In *ICDE*, pages 3–14. IEEE Computer Society, 1995.

[5] D. W. Albrecht, I. Zukerman, and A. E. Nicholson. Pre-sending documents on the www: A comparative study. In *IJCAI*, pages 1274–1279, 1999.

[6] M. Awad, L. Khan, and B. M. Thuraisingham. Predicting www surfing using multiple evidence combination. *VLDB J.*, 17(3):401–417, 2008.

[7] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. *Computer Networks*, 30(1-7):107–117, 1998.

[8] L. D. Catledge and J. E. Pitkow. Characterizing browsing strategies in the world-wide web. *Computer Networks and ISDN Systems*, 27(6):1065–1073, 1995.

[9] F. Chierichetti, R. Kumar, and A. Tomkins. Stochastic models for tabbed browsing. In *Proc. WWW 2010*, 2010.

[10] A. Cockburn and B. J. McKenzie. What do web users do? an empirical analysis of web use. *Int. J. Hum.-Comput. Stud.*, 54(6):903–922, 2001.

[11] M. Deshpande and G. Karypis. Selective markov models for predicting web page accesses. *ACM Trans. Internet Techn.*, 4(2):163–184, 2004.

[12] M. El-Sayed, C. Ruiz, and E. A. Rundensteiner. Fs-miner: efficient and incremental mining of frequent sequence patterns in web logs. In *WIDM*, pages 128–135, New York, NY, USA, 2004. ACM.

[13] X. Fu, J. Budzik, and K. J. Hammond. Mining navigation history for recommendation. In *IUI*, pages 106–112, 2000.

[14] J. A. Gámez, J. L. Mateo, and J. M. Puerta. Improving revisitation browsers capability by using a dynamic bookmarks personal toolbar. In *WISE 2007*, pages 643–652, 2007.

[15] W. Gaul and L. Schmidt-Thieme. Mining generalized association rules for sequential and path data. In *ICDM*, pages 593–596. IEEE Computer Society, 2001.

[16] M. Géry and M. H. Haddad. Evaluation of web usage mining approaches for user's next request prediction. In *WIDM*, pages 74–81. ACM, 2003.

[17] D. Hawking, N. Craswell, P. Bailey, and K. Griffiths. Measuring search engine quality. *Inf. Retr.*, 4(1):33–59, 2001.

[18] R. Kawase, G. Papadakis, and E. Herder. How predictable are you? a comparison of prediction algorithms for web page revisitation. In *ABIS 2010*, 2010.

[19] P. Kazienko. Mining indirect association rules for web recommendation. *Applied Mathematics and Computer Science*, 19(1):165–186, 2009.

[20] R. Kumar and A. Tomkins. A characterization of online browsing behavior. In *Proc. WWW 2010*, 2010.

[21] M. Mayer. Web history tools and revisitation support: A survey of existing approaches and directions. *Foundations and Trends in Human-Computer Interaction*, 2 (3):173–278, 2009.

[22] N. Milic-Frayling, R. Jones, K. Rodden, G. Smyth, A. Blackwell, and R. Sommerer. Smartback: Supporting users in back navigation. In *WWW 2004*, 2004.

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

[24] D. Morris, M. R. Morris, and G. Venolia. Searchbar: a search-centric web history for task resumption and information re-finding. In M. Czerwinski, A. M. Lund, and D. S. Tan, editors, *CHI*, pages 1207–1216. ACM, 2008.

[25] T. Nagel and R. Sander. Hyperhistory. In S. Reich and M. Tzagarakis, editors, *Hypertext*, pages 276–277. ACM, 2005.

[26] H. Obendorf, H. Weinreich, E. Herder, and M. Mayer. Web page revisitation revisited: implications of a long-term click-stream study of browser usage. In *CHI*, pages 597–606. ACM, 2007.

[27] G. Papadakis, C. Niederee, and W. Nejdl. Decay-based ranking for social application content. In *WEBIST*, pages 276–282, 2010.

[28] A. G. Parameswaran, G. Koutrika, B. Bercovitz, and H. Garcia-Molina. Recsplorer: recommendation algorithms based on precedence mining. In *SIGMOD*, pages 87–98, 2010.

[29] J. Pei, J. Han, and W. Wang. Mining sequential patterns with constraints in large databases. In *CIKM*, pages 18–25, 2002.

[30] S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme. Factorizing personalized markov chains for next-basket recommendation. In *WWW*, pages 811–820, 2010.

[31] J. J. Sandvig, B. Mobasher, and R. Burke. Robustness of collaborative recommendation based on association rule mining. In *RecSys*, pages 105–112, 2007.

[32] G. Shani, D. Heckerman, and R. I. Brafman. An mdp-based recommender system. *J. Mach. Learn. Res.*, 6:1265–1295, 2005.

[33] H. Takano and T. Winograd. Dynamic bookmarks for the www. In *Hypertext 1998*, pages 297–298. ACM, 1998.

[34] L. Tauscher and S. Greenberg. How people revisit web pages: empirical findings and implications for the design of history systems. *Int. J. Hum.-Comput. Stud.*, 47(1):97–137, 1997.

[35] S. Tyler and J. Teevan. Large scale query log analysis of re-finding. In *WSDM 2010*, pages 191–200, 2010.

[36] H. Weinreich, H. Obendorf, E. Herder, and M. Mayer. Off the beaten tracks: exploring three aspects of web navigation. In *WWW*, pages 133–142. ACM, 2006.

[37] H. Yang and S. Parthasarathy. *On the Use of Temporally Constrained Associations for Web Log Mining*. Springer-Verlag, 2002.

[38] Y. Yao, L. Shi, and Z. Wang. A markov prediction model based on page hierarchical clustering. *Int. J. Distrib. Sen. Netw.*, 5(1):89–89, 2009.

[39] I. Zukerman, D. W. Albrecht, and A. E. Nicholson. Predicting users' requests on the www. In *UM*, pages 275–284, 1999.