How Predictable Are You?
A Comparison of Prediction Algorithms for Web Page Revisitation

Ricardo Kawase, George Papadakis, Eelco Herder
L3S Research Center
Hannover, Germany
{kawase, papadakis, herder}@L3S.de

Abstract

Users return to Web pages for various reasons. Apart from pages visited due to backtracking, users typically monitor a number of favorite pages, while dealing with tasks that reoccur on an infrequent basis. In this paper, we introduce a novel method for predicting the next revisited page in a certain user context that, unlike existing methods, doesn't rely on machine learning algorithms. We evaluate it over a large data set comprising the navigational activity of 25 users over a period of 6 months. The outcomes suggest a significant improvement over methods typically used in this context, thus paving the way for exploring new means of improving user's navigational support.

1 Introduction

Nowadays, millions of people browse the Web every second, navigating from site to site and producing massive amounts of navigational log data. These data have the intrinsic potential to provide a solid basis for understanding individual user's behavior. Modeling users is the first step towards this direction, serving as a foundation for developing recommendation and prediction techniques.

Many applications can benefit from effective methods of user modeling, like Web search and personalization/recommendation systems, to name but a few. For example, predictive models have improved the ranking of web search engine results, by computing the distribution of visits over all WWW pages and using it for re-weighting and re-ranking relevant web pages. Navigational information are actually considered more important than text keywords. Hence, the more accurate the predictive models, the better the search results [Brin and Page, 1998].

Several researchers have undertaken the task of understanding user's surfing behavior, by exploiting user data [Adar et al., 2008; Obendorf et al., 2007]. Some go further, using log data to improve algorithms that predict future requests [Awad et al., 2008 ; Gery and Haddad, 2003], while others apply these algorithms to provide users with improved tools for recommendations, bookmarks and history [LeeTiernan, 2003; Pedersen et al., 2010].

As a common practice of studying the past in order to define the future, in this paper we analyze the browser's log data of 25 users along 6 months with a total of 137,737 page requests. Our detailed, statistical analysis of the navigational data gives insights for our research as well as for future work in the area. In addition, we demonstrate a novel user modeling method for predicting the next page that will be revisited. We tested our model on the data set at hand, with the experimental results demonstrating a significant improvement in the support of Web page revisitation over existing methods, commonly used in this area.

2 Related Work

Several past works have explored surfing behaviors with respect to revisitation activity. Although they vary in their estimations, they all recognize that revisitation constitutes a large part of the Web activity. [Herder, 2005], for instance, quantifies it to 50% of the overall Web traffic, while [Cockburn and McKenzie, 2001] approximates it to 80%. They also noted that bookmarks, the most popular revisitation supporting tool, invariably involve managing and organizational problems due to the constantly increasing size of their collections.

Analysis of revisitation. [Tauscher and Greenberg, 1997] describes two important characteristics of revisitation: first, most page revisits pertain to pages accessed 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.

Revisitation behavior has been distinguished by [Obendorf *et al.*, 2007] into three different sets: *short-term* (i.e., backtrack or undo within the same session), *medium-term* (i.e., re-utilize or observe a resource in a period of time up to few weeks after the first encounter), *and long-term revisits* (i.e., rediscover a resource several months after the first encounter). The authors further argue that the back button is the most commonly used tool for short-term revisit. For medium-term revisits, the page address is directly typed into the address bar, making use of the automatic URL completion function. However, revisits to a broad range of pages that are accessed on a less frequent basis (i.e., long-term revisits) are poorly supported; users often do not remember the exact address, and ironically browsers do not 'remember' the address either.

[Adar *et al.*, 2008] demonstrates that short-term revisits involve hub-and-spoke navigation, visiting shopping or reference sites or pages on which information was monitored. Medium-term revisits pertain to popular home pages, Web mail, forums, educational pages and the browser homepages. As for long term revisits, they involve the use of search engines, as well as weekend activities (e.g., going to the cinema).

Revisitation Prediction. In [Gery and Haddad, 2003], the authors exploit three methods of Web usage mining: association rules, frequent sequences, and frequent generalized sequence. *Association Rules (AR)* are well documented in the literature as a method that effectively identifies pages that are typically visited together in a same session, but not necessarily in the same order. [Agrawal *et al.*, 1993; Agrawal and Srikant, 1995]. *Frequent Sequence Mining* can be considered as equivalent to association rule mining over temporal data sets, while *Frequent Generalized Sequence* introduces sequences that allow wildcards, constituting a more flexible means of modeling users's navigational activity [Gaul and Schmidt-Thieme, 2000]. Their evaluation shows that plain Frequent Sequence Mining performs better in revisitation prediction. However, their dataset consists of server side logs of 3 different websites, thus covering a limited number of possible revisited pages. Contrariwise, in our work we employ browsers's log data to analyze and predict users's, with the set of revisited web pages potentially involving the whole Web.

In [Awad *et al.*, 2008], the authors apply two well-established classification techniques in the context of Web surfing prediction: Markov model and Support Vector Machines (SVM). They also combine them in a hybrid method under Dempster's rule and the outcomes of their evaluation suggest that it outperforms the individual methods, especially when domain knowledge is incorporated into it.

3 Data Set and Revisitation Statistics

In this section we briefly introduce the data set that we used for our experiments. We also illuminate the most important aspects of users's revisit behavior – general characteristics as well as individual differences – which are used as a basis for the predictive methods that are evaluated in this paper.

3.1 Data set

The participant pool of our data set consists of 25 participants, 19 male and 6 female. Their average age is 30.5, ranging from 24 to 52 years. The participants were logged for some period between August, 2004 and March, 2005. The average time span of the actual logging periods was 104 days, with a minimum of 51 days and a maximum of 195 days. Participants were logged in their usual contexts - 17 at their workplace, 4 both at home and at work, and 4 just at home.

During the logging period, 152,737 page requests were recorded. 10.1% of them were removed, as they were artifacts (advertisements, reloads, redirects, frame sets). Hence, in total we have 137,737 page requests available for analysis.

3.2 Revisitation Statistics

We recorded an average revisitation rate of 45.6%. Note that this number is lower than in earlier studies, due to the fact that we took into account both GET and POST parameters. The wide range of individual revisitation range (between 17.4% and 61.4%) suggests that revisitation behavior is heavily influenced by personal habits, private interests and the sites visited (for more details, [Obendorf, 2008]). In this section we concentrate on individual ferences between users in their *revisitation profile*.

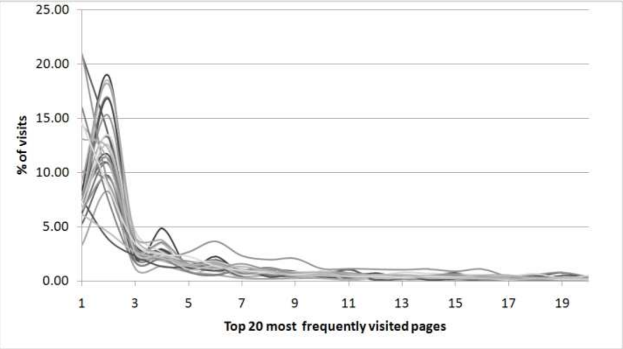


Figure 1: Distribution of most frequently visited pages for each user.

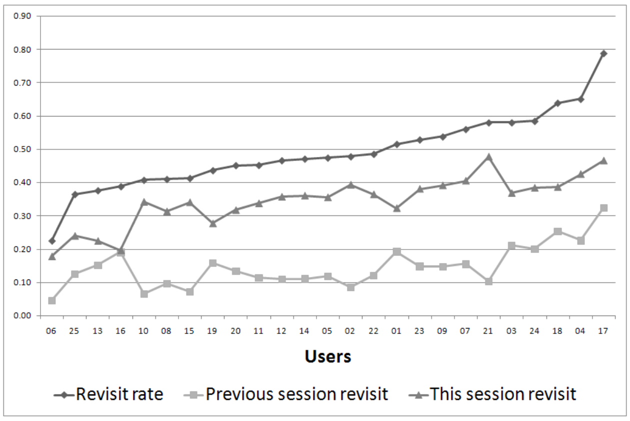


Figure 2: Backtracking and routine behavior plotted against the revisit rate (order by revisit rate).

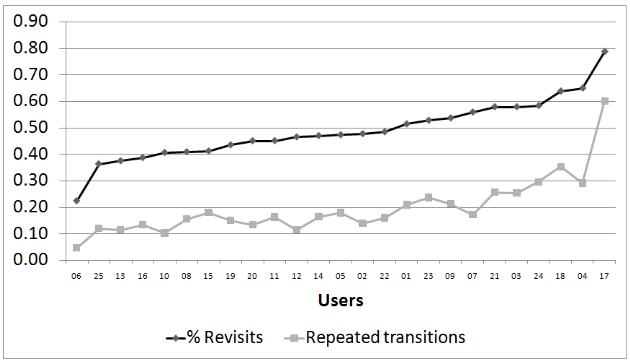


Figure 3: Repetitive behavior (% repeated actions) plotted against the revisit rate.

As discussed in Section 2, several studies have identified regularities in revisitation behavior. Users typically have a small set of frequently visited pages, including for example the browser's home page, search engines, favorite news sites, and social networking sites. As can be observed in **Figure 1**, the distribution of *most frequently used* pages clearly follows a power law for most of the users, but not for all – some have a large number of pages in their browsing routine.

The distribution of revisits to pages based on the number of pages between the last visit and the current visit does follow a power law distribution for all users. Consequently, the *backtracking activities* (revisits to pages in the current session) and *routine behavior* (revisits to pages in previous sessions) grow roughly linear with the revisit rate. This illustrated in **Figure 2** - note that despite the correlation there are still users that can be identified as predominantly backtrackers or predominantly routine revisitors. The average percentage of backtracking actions among revisits is 75%, with a minimum of 51% and a maximum of 84%.

Predictive models of Web navigation, such as Markov models, typically assume that users exhibit a rather large percentage of *repetitive behavior*, including sequences of pages that are regularly visited in the same order. In **Figure 3** we plot the users' repetitive behavior (based on the ratio between the number of unique pairs of pages that a user visited consecutively and the total number of transitions). The average percentage of repetitive transitions is 20%, with a minimum 5% and a maximum of 60%.

3.3 Discussion

Based on the statistics in the previous section, it becomes clear that Web page revisitation behavior follows sufficient regularities to be exploited for enhanced revisitation support – in a similar manner as is already common in recommender systems based on Web usage mining and collaborative filtering. Earlier work on revisitation (see Section 2) confirms this observation, but only to a limited extent.

In this following, we investigate, compare and combine the performance of several predictive methods for page revisitation. Our analysis attempts a general comparison of several prediction mechanisms, with the aim of identifying the best performing one, knowing though that their performance depends heavily on the regularities in the individual user's revisitation activities.

4 Prediction of Next Page Visits

The problem we are tackling in this paper can be formally defined as follows:

*Given a collection of Web Pages, **P = {p$_1$, p$_2$, ...}**, that have been visited by a user, **u**, during his past **n** transactions, **T$_u$={t$_1$, t$_2$,…, t$_N$}**, rank them so that the ranking position of the page re-visited in the next, **n+1**, transaction is the highest possible.*

The methods copying with this problem should exclusively try to facilitate the revisitation of already accessed pages , rather than trying to suggest to a user new pages that seem relevant to his surfing activity. The ranking of all web pages is updated after every transaction, and the higher the ranking position of the subsequently accessed page, the better. In fact, the lowest possible Average Ranking Position (**ARP**) of revisited pages, the higher the performance of the algorithm. This is in line with the intuition behind ranking search engine's query results: the higher the ranking of the desired resource, the better the performance of the search engine [Brin and Page, 1998].

To solve the aforementioned problem, we employ a framework combining two categories of methods. The first one involves *ranking methods*: they estimate for each web page the likelihood that it will be accessed in the next transaction based on some evidence, such as the recency or the frequency of earlier visits to this page. The second category covers *propagation methods*; these are techniques that capture repetitiveness in the surfing behavior of a user and identify groups of pages that are typically visited together, in the same session but not necessarily in a specific order.

In the following, we provide a brief outline of our framework that conglomerates these two categories of prediction methods. The implementation of the methods presented here is publicly available through the SUPRA project of SourceForge.net[1].

4.1 Ranking Methods

The aim of ranking methods is to provide for each web page a numerical estimate of the likelihood that it will be accessed in the next transaction. In this work, we consider the following ranking methods:

1. Least Recently Used (**LRU**)
2. Most Frequently Used (**MFU**)
3. Polynomial Decay (**PD**)

The first two methods, namely LRU and MFU, constitute well-established caching algorithms that are typically employed in prediction tasks. LRU is based on the idea that the more recently a web page was visited, the most likely that it will be re-visited in the immediate future. Hence, it assigns the highest ranking position to the latest accessed page. MFU, on the other hand, relies on the idea that the more often a web page is visited, the most likely it is to be revisited in the next transaction.

[Papadakis et al., 2010] demonstrated, though, that these methods are not adequate for effectively predicting future revisitations on server-side logs of closed corpus websites. Due to their unidimensionality, LRU produces a plainly chronological arrangement of web pages based on their recency, while MFU takes into account merely their *degree of usage*. More accurate predictions can be achieved when incorporating both evidences into a single, comprehensive method.

To this end, [Papadakis et al., 2010] introduced the decay ranking model for predicting the next revisited page. According to this model, the value $\boldsymbol{v_{in}}$ of a web page $\boldsymbol{w_i}$ after ***n*** transactions $\boldsymbol{T_u}$ of user ***u*** is derived from the following formula:

$$v_i = \sum_{k=0}^{n} d(t_k, w_i, n), \text{ where}$$

d(t$_k$,w$_i$, n) is a ***decay function*** that takes as an input the *k-th* transaction, t_k, of user, *u*, together with the index of the current transaction, *n*, and gives as output the value of this transactions for web page w_i . Every valid decay function should satisfy the following *properties* ([Cormode et al., 2009]):

1. $\mathrm{d}(t_k, w_i, n) = 1$ when k=n
2. $d(t_k, w_i, n) = 0$ if t_k doesn't pertain to web page w_i
3. $0 \le d(t_k, w_i, n) \le 1 \ \forall\ 0 \le \mathrm{k} \le \mathrm{n}$
4. d is monotone non-increasing as n increases (0≤k≤n):
$$n' \ge n \rightarrow d(t_k, w_i, n') \le d(t_k, w_i, n)$$

Among the various decay function families that satisfy these properties, the *polynomial decay functions* were found to outperform both the *exponential* and the *logarithmic* ones. The reason is that their smooth decay balances harmonically the recency and the degree of usage of web pages; in contrast to this, exponential functions convey a steep decay that puts more emphasis on the recency of usage, whereas the logarithmic functions promote excessively the degree of usage, due to their excessively slow decay.

The form of a ***polynomial decay function with exponent α*** is the following:

[1] http://sourceforge.net/projects/supraproject/

$$d(t_k, w_i, n) = \frac{b}{1+(n-k)^{\alpha}}, \text{ where}$$

b is equal to 1 if t_k pertains to w_i, and 0 otherwise.

4.2 Propagation Methods

Unlike ranking methods that produce an ordering of web pages, propagation methods aim at detecting and capturing patterns in the surfing activity of users. They identify those pages that are commonly visited within the same session and associate them with each other. The "links" created by these methods can be combined with a ranking method, so that the value of a web page is propagated to its relevant ones. In this way, the higher the value of a web page, the more the pages associated with it are boosted and the higher their ranking position.

In this work, we distinguish between two families of propagation methods: *those that take into account the order of the transactions within a session, and those who disregard this order*. For the former case, we consider transition matrices, whereas for the latter we examine association matrices.

4.2.1 Transition Matrix

Similar to a first-order Markov model, a transition matrix (**TM**) is a two dimensional structure with its row and columns representing the enumeration of web pages; each cell *TM(x,y)* expresses the number of times that a user visited page *y* after *x*. Given that a transition matrix respects the order of accesses within a session, it is not a symmetrical one: the value of *TM(x,y)* is not necessarily equal to that of *TM(y,x)*. Moreover, its diagonal cells are all equal to 0: $\mathrm{TM}(x,x) = 0\ \forall x$.

In the following, we introduce 4 different approaches to correlating web pages according to the past navigational activity in order to build the transition matrix. They can be intuitively illustrated through a simple walkthrough example. Given a set of 4 web pages – *A, B, C, D* - and the following set of transactions during a user session

A ⇨ B ⇨ C ⇨ D ⇨ A

we can associate these pages in four ways (taking into account the order of the accesses):

1. **Simple connectivity** – For each transition *x→y* in the given session, only the value of the cell *TM(x,y)* is incremented by one. **Figure 4a)** depicts the values of the transition matrix according to the simple connectivity rule after the last transition of the given session *D→A*.
2. **Continuous connectivity** – Each web page visited within the current session is associated with all the subsequently accessed pages. In our example, after transition *D→A*, *A* is associated with all other web pages (*B*,*C*,*D*) incrementing the corresponding cells by one, as shown in **Figure 4b)**.
3. **Decreasing continuous connectivity** – This strategy operates in a similar way as the previous one (i.e., connecting all the pages within a session) with the difference that it adds a decay parameter representing the distance (i.e., number of transitions) that intervene between two web pages. In our example, cell *(C,A)* is incremented by ½ after *D→A*, since page *C* is two steps away from the page A. **Figure 4c)** depicts the values of the transition matrix according to the decaying continuous connectivity rule after the transition *D→A*.
4. **Increasing continuous connectivity** – Is the inverted version of the previous strategy. Instead of decreasing the additional value of cell *TM(x,y)* according to the distance of pages *x* and *y*, it increases it proportionally. The outcomes of this rule after transition *D→A* are presented in **Figure 4d)**.

It is worth noting that the simple connectivity transition matrix was also used in [Awad *et al.*, 2008], but its frequencies were used as features of a classification algorithm instead.

$$\begin{array}{c|cccc} & A & B & C & D \\ \hline A & 0 & 1 & 0 & 0 \\ B & 0 & 0 & 1 & 0 \\ C & 0 & 0 & 0 & 1 \\ D & 1 & 0 & 0 & 0 \end{array} \qquad \begin{array}{c|cccc} & A & B & C & D \\ \hline A & 0 & 1 & 1 & 1 \\ B & 1 & 0 & 1 & 1 \\ C & 1 & 0 & 0 & 1 \\ D & 1 & 0 & 0 & 0 \end{array}$$

(a) (b)

$$\begin{array}{c|cccc} & A & B & C & D \\ \hline A & 0 & 1 & \frac{1}{2} & \frac{1}{4} \\ B & \frac{1}{4} & 0 & 1 & \frac{1}{2} \\ C & \frac{1}{2} & 0 & 0 & 1 \\ D & 1 & 0 & 0 & 0 \end{array} \qquad \begin{array}{c|cccc} & A & B & C & D \\ \hline A & 0 & 1 & 2 & 4 \\ B & 4 & 0 & 1 & 2 \\ C & 2 & 0 & 0 & 1 \\ D & 1 & 0 & 0 & 0 \end{array}$$

(c) (d)

Figure 4: Transition matrix example.

4.2.2 Association Matrix

In contrast with transition matrices, association matrices (**AM**) are based on the idea that the temporal order of transactions within a session is not important; pages that are 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 XYZ on a regular basis, but not necessarily in that order.

In this context, an association matrix is built simply by associating all the pages that are visited in a single session. Given the session presented above, the resulting AM has all non-diagonal cells equal to one, as all resources were accessed during this session (**Figure 5**).

A variation of the association matrix can be derived by normalizing its values with the help of the mutual information. More specifically, this involves the multiplication of each cell *TM(x,y)* of AM with the following mutual information factor (*mif*):

$$mif(x,y) = AM(x,y) \cdot \log \frac{p(x,y)}{p(x) \cdot p(y)}, \text{ where}$$

- *AM(x,y)* is the number of sessions containing both page x and y (i.e., the value of the cell *AM(x,y)*),
- *p(x,y)* is the probability of a session to contain pages *x* and *y* (i.e., the value of *AM(x,y)* divided by the number of sessions)

- *p(x)* (*p(y)*) is the probability that a session contains page *x* (*y*) (i.e., the number of sessions with x or y divided by the total number of sessions).

Without this smoothing factor, the values of AM are biased towards pairs of pages that have a high frequency of co-occurances, although they are not highly correlated.

$$\begin{array}{c c}
 & \begin{array}{cccc} A & B & C & D \end{array} \\
\begin{array}{c} A \\ B \\ C \\ D \end{array} & \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}
\end{array}$$

Figure 5: Association matrix example.

4.3 Combining Ranking with Propagation methods

To combine the available ranking methods with the variations of the propagation techniques, we employ a simple, linear scheme: following a transaction, the value of each web page is first (re)computed, according to the selected ranking method. Then, for each non-zero cell of the transition matrix at hand, *TM(x,y)*, we increase the value of page y, v_y, as follows:

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

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

In case an association matrix is used as a propagation method, the v_y is increased as follows:

1. $v_y += AM(x,y) \cdot v_x$ for the plain association matrix, or
2. $v_y += mif(x,y) \cdot v_x$ for the mutual-information-normalized association matrix.

All in all, considering the 3 ranking methods alone and in combination with the 4 variations of TM and the 2 variations of AM, we have 21 distinct ranking methods. Due to space limitation and for the sake of readability, the following, evaluation section focuses merely on the best performing ones.

5 Evaluation Setup and Discussion of Results

5.1 Setup

To evaluate experimentally our framework of methods, we employed the data set described in section 3, comprising 25 distinct users and 137,737 page requests in total (not evenly distributed among the users). In more detail, we simulated the navigational activity of each user, independently of the others. After each transaction, 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 places for each recommendation method, we derived the following metrics for evaluating its performance:

- *Precision at 10* (***P@10***): it expresses the percentage of revisitations that involved a web page ranked in some of the top 10 positions. The higher this percentage, the better the performance of the recommendation method. This metric provides evidence for the 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).
- *Average Ranking Position Reduction Ratio* (***RR***): it denotes the degree of improvement conveyed by the prediction method in comparison with the actual revisititation behavior of the user. More specifically, it is computed from the following formula:

$$RR = \frac{AcARP - PrARP}{AcARP} \cdot 100\%, \text{ where}$$

  - *AcARP* is the *Ac*tual *A*verage *R*anking *P*osition of the user, representing the average distance in terms of the number of page requests that intervene between the revisited web pages, and
  - *PrARP* is the *Pr*ediction *A*verage *R*anking *P*osition, expressing the place a revisited page is found on average in the ranking list that the prediction method produces.

The higher the value of RR, the better the performance of the recommendation algorithm, with negative values denoting that PrARP is lower than AcARP (i.e., no improvement with respect to the actual revisitation behavior of the user). RR provides, thus, an estimation of the overall performance of a prediction method, since it considers the performance over all the revisitations in the navigational history of a user, and not only the highest ranked ones.

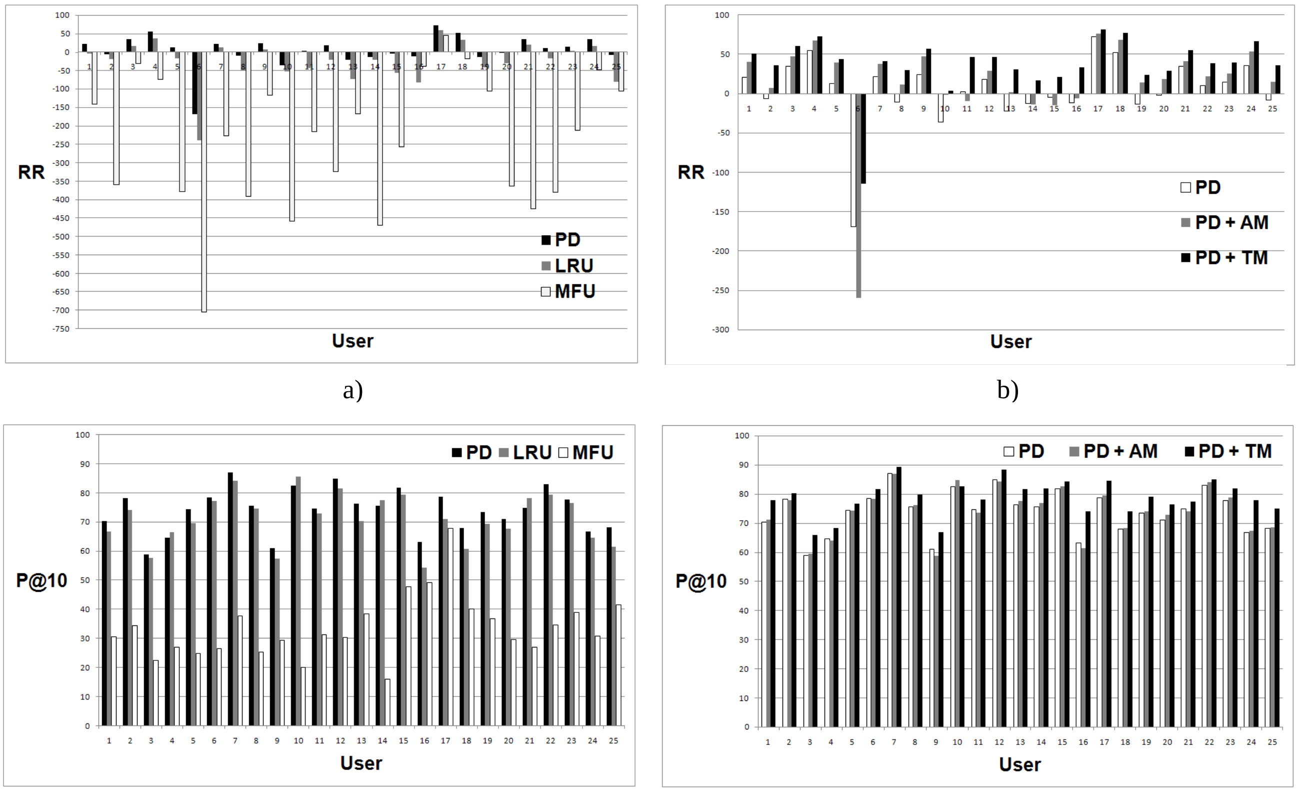
On the whole, the combination of these two metrics provides a comprehensive estimation of the effectiveness of a recommendation algorithm in predicting the next revisited page; they cover both the recommendations that are indeed useful for users as well as their performance in all the cases.

5.2 Results analysis

Regarding the performance of the ranking algorithms we are considering, it is summarized in **Figure 6**, with the RR depicted in **Figure 6a)** and the P@10 in **Figure 6c)**. It is evident that the baseline MFU performs much worse than the other methods. This is explained by the fact that backtracking (LRU) is more common than revisiting popular sites, thus ensuring much higher performance for LRU. Our proposed method, the Polynomial Decay, which is a combination of MFU and LRU, exhibits the best performance for all users, improving in each case that of LRU to a varying but considerable extent.

The performance of PD is significantly enhanced when combined with AM and TM, with TM accounting for a higher improvement. This is the case with respect to both metrics, as is clearly depicted in **Figure 6b)** for RR and **Figure 6d)** for P@10. Conversely, the combination of LRU and MFU with AM and TM results in a lower performance for both metrics (that's why their performance is not included in the figures). This suggests that users do not have many regular patterns in their page visit behavior (i.e. after having visited page X they do not always visit page Y). It is interesting to note, though, that PD achieves by far the best results in combination with the Simple TM, while LRU and MFU are better combined with the Increasing and the Decreasing TM, respectively.

Another observation is that, despite the different assumptions that lie behind the algorithms, there is a corre-



a) b)

c) d)

Figure 6: a) Reduction ratios of the average ranking position for LRU, MFU and PD. b) PD with AM and TM(simple. c) Precision at 10 for LRU, MFU and PD. d) PD with AM and TM(simple) on the bottom right.

lation between the performances of the algorithms per user. This can be observed in **Figure 7**, where the better the performance of the best-performing algorithm, PD+TM, the better the performances of PD, PD+AM and LRU. From the same figure it also becomes clear that there is no correlation at all with the revisit rate: one would expect that users who revisit pages more often – who are shown to have more frequent transitions – are more predictable in their behavior; this turns out not to be the case. Note also that the – poor – performance of MFU does not follow the pattern of the other algorithms and is not correlated with the revisit rate either.

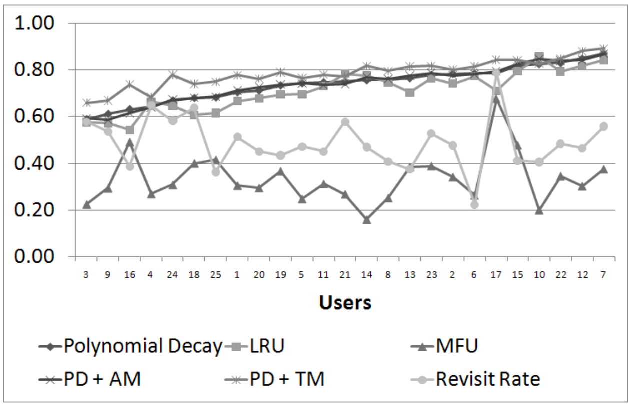


Figure 7: Performance (P@10) of the different algorithms per user. Users are ordered by the best-performing algorithm.

5.3 Discussion

In our analysis we compared various algorithms and combinations of algorithms for predicting which pages users will revisit in a session. These algorithms exploit the following characteristics of revisits:

- Revisits are typically focused on pages visited very recently.
- The more revisits, the more repetitive behavior in terms of transitions between pages.
- There is a small group of pages that is visited very frequently.

It turns out that, even though a small set of frequently visited pages covers the majority of revisits, the recency effect, as well as frequent transitions, plays a larger role in the prediction algorithms.

The evaluated algorithms, in particular Polynomial Decay in combination with the transition matrix, significantly improve upon the list of most recently used pages, in particular for users whose the list of LRU pages performs relatively bad. The differences become smaller together with the increase of the recency effect.

The counter-intuitive effect that a higher recency rate does *not* lead to better predictions can be explained by the many differences in individual behavior between users (such as the number of news sites or bulletin boards that a user actively follows, strategies for search and backtracking, the number of reoccurring activities).

From this we can conclude that revisiting behavior is mainly influenced by the recency effect, but it definitely makes sense to take the popularity of pages and the currently/last visited page (the user's current context) into account as well.

6 Conclusions

In this paper we studied the browsing behavior of 25 users during a period of approximately 6 months. We analyzed the data to build a comprehensive stereotype of users' behavior, focusing on their revisitation patterns. We also ran experiments applying a variety of methods to predict users's revisitation.

Our proposed Polynomial Decay algorithm in combination with users's navigational patterns as they are encapsulated by the Transition Matrix outperforms substantially existing methods commonly used for revisitation prediction.

In our previous work [Papadakis et al., 2010] we demonstrated the better performance of our methods on a server-side dataset. Combining with the results of the work presented here (on a client-side dataset) we can firmly claim that our proposed method is more effective than the baselines LRU and MFU for both cases.

Though the experiments presented here are on the field of Web Usage Mining, our real goal is to improve the support of revisitation by the means of intelligent user interfaces. Hence, this was the first step towards a more effective user modeling method. Our plan, as future work, is to implement a browser interface that allows users to interact with the output of our methods: a collection of related URLs that does not contain only the obvious selections, but also related websites that are usually overlooked between the head and the long tail.

References

[Adar *et al*., 2008] Adar, E., Teevan, J., and Dumais. S. T: Large scale analysis of Web revisitation patterns. In CHI, pages 1197-1206, 2008.

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

[Agrawal and Srikant, 1995] Agrawal, R., and Srikant, R.: Mining sequential patterns. In ICDE, pages 3–14, 1995.

[Awad *et al*., 2008] Awad, M., Khan, L., and Thuraisingham, B.:Predicting WWW surfing using multiple evidence combination. In The VLDB Journal 17, 3, pages 401-417, 2008.

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

[Cockburn and McKenzie, 2001] Cockburn, A., and McKenzie, B.: What do Web users do? An empirical analysis of Web use. In Int. J. of Human-Computer Studies, 54(6), pages 903-922, 2001.

[Cormode *et al.*, 2009] Cormode, G., Shkapenyuk, V., Srivastava, D., and Xu, B.: Forward Decay: A Practical Time Decay Model for Streaming Systems, In ICDE, pages 138-149, 2009.

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

[Gaul and Schmidt-Thieme, 2000] Gaul, W., and Schmidt-Thieme, L.: Mining web navigation path fragments. In WEBKDD, 2000.

[Herder, 2005] Herder, E.: Characterizations of user Web revisit behavior. In Proceedings of Workshop on Adaptivity and User Modeling in Interactive Systems, 2005.

[LeeTiernan, 2003] LeeTiernan, S., Farnham, S., and Cheng, L.: Two methods for auto-organizing personal web history. In CHI Extended Abstracts on Human Factors in Computing Systems, pages 814-815, 2003.

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

[Papadakis et al., 2010] Papadakis, G., Niederee, C., Nejdl, W.: Decay-based Ranking for social application content. In WEBIST, 2010.

[Pedersen et al., 2010] Pedersen, E. R., Gyllstrom, K., Gu, S., and Hong, P. J.:Automatic generation of research trails in web history. In IUI, pages 369-372, 2010.

[Tauscher and Greenberg, 1997] Tauscher, L., and Greenberg, S.: How people revisit Web pages: Empirical findings and implications for the design of history systems. InInternational Journal of Human-Computer Studies, v.47, n.1, pages 97-137, 1997.