10-1-2007

Web Site Personalization based on Link Analysis and Navigational Patterns

Magdalini Eirinaki
San Jose State University, magdalini.eirinaki@sjsu.edu

M. Varzirgiannis
Athens University of Economics and Business

Recommended Citation
Magdalini Eirinaki and M. Varzirgiannis. "Web Site Personalization based on Link Analysis and Navigational Patterns" *ACM Transaction on Internet Technology* (2007): 21:1-21:27. doi:10.1145/1278366.1278370

Web Site Personalization based on Link Analysis and Navigational Patterns

MAGDALINI EIRINAKI
Athens University of Economics and Business, Dept. of Informatics

MICHALIS VAZIRGIANNIS
Athens University of Economics and Business, Dept. of Informatics

---

The continuous growth in the size and use of the World Wide Web imposes new methods of design and development of on-line information services. The need for predicting the users' needs in order to improve the usability and user retention of a web site is more than evident and can be addressed by personalizing it. Recommendation algorithms aim at proposing "next" pages to users based on their current visit and the past users' navigational patterns. In the vast majority of related algorithms, however, only the usage data are used to produce recommendations, disregarding the structural properties of the web graph. Thus important – in terms of PageRank authority score – pages may be underrated. In this work we present *UPR*, a PageRank-style algorithm which combines usage data and link analysis techniques for assigning probabilities to the web pages based on their importance in the web site's navigational graph. We propose the application of a localized version of *UPR* (*l-UPR*) to personalized navigational sub-graphs for online web page ranking and recommendation. Moreover, we propose a hybrid probabilistic predictive model based on Markov models and link analysis for assigning prior probabilities in a hybrid probabilistic model. We prove, through experimentation, that this approach results in more objective and representative predictions than the ones produced from the pure usage-based approaches.

Categories and Subject Descriptors: H.2.8 [**Database Management**]: Database Applications – Data Mining; H.3.5 [**Information Storage and Retrieval**]: Online Information Services - Web-based services

General Terms: Algorithms, Experimentation
Additional Key Words and Phrases: Web Personalization, Recommendations, Link Analysis, Usage-based PageRank, Markov Models

---

1. INTRODUCTION

The evolution of World Wide Web as the main information source for millions of people nowadays has imposed the need for new methods and algorithms that are able to process efficiently the vast amounts of data that reside on it. Users become more and more demanding in terms of the quality of information provided to them when searching the web or browsing a web site. The area of web mining, including any method that utilizes data residing on the web, addresses this need. The most common applications involve the ranking of the web search engines results and the provision of recommendations to the users of – usually commercial – web sites, known as web personalization.

The connectivity features of the web graph play an important role in the process of web searching and navigating. Several link analysis techniques, based on the popular PageRank algorithm [Brin and

Authors' addresses: Dept. of Informatics, Athens University of Economics and Business, Patision 76, 10434, Athens, Greece

Page 1998], have been largely used in the context of web search engines. The underlying intuition of these techniques is that the importance of each page in a web graph is defined by the number and the importance of the pages linking to it. In the past many variations of this algorithm, aimed at improving the acquired results, have been proposed. Some of these approaches, make use of the so called "personalization vector" of PageRank in order to bias the results towards the individual needs of each user who is searching the web [Aktas et. al. 2004, Haveliwala 2002, Richardson and Domingos 2002].

In this work, we introduce link analysis in a new context, that of web personalization. Web personalization is defined as any action that adapts the information or services provided by a Web site to the needs of a user or a set of users, taking advantage of the knowledge gained from the users' navigational behaviour and individual interests, in combination with the content and the structure of the Web site [Eirinaki et. al. 2003]. Motivated by the fact that in the context of navigating a web site, a page is important if many users have visited it before, we propose a novel algorithm, named UPR (Usage-based PageRank), that assigns importance rankings (and therefore visit probabilities) to the web site's pages. UPR is a PageRank-style algorithm that is applied on an abstraction of the user sessions named the Navigational Graph (NG). We specialize this generalized personalization framework in two different contexts. We develop l-UPR, a recommendation algorithm based on a localized variant of UPR that is applied to the personalized navigational sub-graph of each user for providing fast, online recommendations. Moreover, we integrate UPR and its variations in a hybrid probabilistic predictive model (h-PPM) as a robust mechanism for determining prior probabilities of page visits. To the best of our knowledge, this is the first integrated solution addressing the problem of web personalization using a page ranking approach.

In a nutshell, our key contributions are:

- A unified personalization framework integrating web usage mining with link analysis techniques for assigning probabilities to the web pages based on their importance in the web site's navigational graph. We define *UPR*, a usage-based personalized PageRank-style algorithm used for ranking the web pages of a site based on the navigational behavior of previous users.
- The introduction of *l-UPR*, a localized version of *UPR* which is applied to personalized sub-graphs of the navigational graph in order to provide fast, online rankings of the most probable "next" pages of interest to the current users. We describe how these personalized sub-graphs are generated online, based on the current visit of each user.

- The application of *UPR* for extending and enhancing standard web usage mining and personalization probabilistic models such as Markov models. We present a hybrid probabilistic prediction framework (*h-PPM*) where *UPR*, as well as its variations, are used for assigning prior probabilities to the nodes (pages) of any Markov model based on the topology (structure) and the navigational patterns (usage) of the web site.
- An extensive set of experiments proving *UPR*'s effectiveness in both proposed frameworks. We apply *UPR* and its variations in order to assign prior probabilities of page visits. These priors probabilities are subsequently used by different order Markov models and show that the recommendation accuracy is better than pure-usage based approaches. Moreover, we apply *l-UPR* to localized navigational sub-graphs for generating online recommendations and again support our claim for the need of enhancing the prediction process with information based on the link structure in combination with the usage of a site.

The rest of the paper is organized as follows: In Section 2 we overview the related work. In Section 3 we present some preliminaries concerning the Navigational Graph and the Markov models. In Section 4 we provide the required theoretical background on link analysis and present *UPR*. We prove that this hybrid algorithm can be applied to any web site's navigational graph as long as the graph satisfies certain properties. The two proposed personalization frameworks in which *UPR* can be applied, namely, the localized personalized recommendations with *l-UPR* and the hybrid probabilistic predictive models (*h-PPM*), are described in Sections 5 and 6. Section 7 includes extensive experimental evaluation of both frameworks. Finally, we conclude with our plans for future work in Section 8.

2. RELATED WORK

Although the connectivity features of the web graph have been extensively used for personalizing web search results [Aktas et. al. 2004, Haveliwala 2002, Richardson and Domingos 2002], only a few approaches that take them into consideration in the web site personalization process exist. Zhu et. al. [Zhu et. al. 2002] use citation and coupling network analysis techniques in order to conceptually cluster the pages of a web site. The proposed recommendation system is based on Markov models. Nakagawa and Mobasher [2003] use the degree of connectivity between the pages of a web site as the determinant factor for switching among recommendation models based on either frequent itemset mining or

sequential pattern discovery. Nevertheless, none of the aforementioned approaches fully integrates link analysis techniques in the web personalization process by exploiting the notion of the authority or importance of a web page in the web graph.

In a very recent work, Huang et. al. [2005] address the data sparsity problem of collaborative filtering systems by creating a bipartite graph and calculating linkage measures between unconnected pairs for selecting candidates and make recommendations. In this study the graph nodes represent both users and rated/purchased items. Finally, subsequent to our original work, Borges and Levene [2006] proposed independently two link analysis ranking methods, SiteRank and PopularityRank which are in essence very much like the proposed variations of our *UPR* algorithm (*PR* and *SUPR* respectively). This work focuses on the comparison of the distributions and the rankings of the two methods rather than proposing a web personalization algorithm. The authors' concluding remarks, that the topology of the web site is very important and should be taken into consideration in the web personalization process, further support our claim.

In the second proposed framework, we extend Markov models by integrating link analysis techniques in the process of generating recommendations. In the past, many researches have proposed the use of 1st order (Markov Chains) [Borges and Levene 2000, Cadez et. al. 2000, Sarukkai 2000], higher-order [Levene and Loizou 2003], or hybrid [Cadez et. al. 2006, Deshpande and Karypis 2001, Manavoglu et. al. 2003, Sen and Hansen 2003] Markov models, based on the usage data of a web site. Apart from Markov models, there are many approaches that perform web usage mining for web personalization, employing association rules mining, clustering, sequential pattern discovery, frequent pattern discovery or collaborative filtering techniques. Some of these approaches model the user sessions with structures that resemble to the Navigational Graph presented in this work [El-Sayed et. al. 2004, Spiliopoulou and Faulstich 1998, Zhao and Bhowmick 2004]. These approaches are out of the scope of this paper and won't be further discussed. For an extensive overview of such approaches the reader may refer to [Eirinaki and Vazirgiannis 2003, Eirinaki 2004].

3. PRELIMINARIES

The input to our proposed algorithm is the Navigational Graph (*NG*). *NG* is a weighted directed graph representation of the user sessions. *NG* contains all the distinct user sessions, and is a full representation of the actual user paths followed in the past. Therefore it can be used in order to discover

page and path probabilities and support popular path prediction. This structure, however, can become large, especially when it represents the user sessions of big web sites. Therefore, the processing of *NG* may become very intensive computationally. The need for reduced complexity and online availability imposes the creation of approximations of the *NG*, referred to as *NG synopses*. An *NG synopsis* may be a Markov model of any order (depending on the desired simplicity/accuracy trade-off), or any other graph synopsis, such as those proposed in [Polyzotis and Garofalakis 2002, Polyzotis et. al. 2004]. We should stress at this point that our approach is orthogonal to the type of synopsis one may choose. In what follows we present in more detail the *NG* structure and its synopses, emphasizing on Markov models, since these are the *NG synopses* we are using in the second framework we propose in this paper as well as in the experimental study we performed.

3.1 The Navigational Graph

As already mentioned, the Navigational Graph (*NG*) is a weighted directed graph which represents the user sessions of a web site. In its simplest form, *NG* is a node- and edge-labeled tree, that has as root a special node *R* and the labels of the nodes identify the *N* web pages of the web site (*WS*). Another option would be to encode the data as a graph using a bisimulation of the tree-based representation. We stress that this choice is orthogonal to the techniques that we introduce. The edges of *NG* represent the links between the web pages (i.e. the paths followed by the users), and the labels (weights) on edges represent the number of link traversals. The weighted paths from the root towards the leaves represent all the user sessions' paths that are included in the web logs. All tree paths terminate in a special leaf-node *E* denoting the end of a path. The *NG* resembles to the web site's graph, it may, however, include page links that do not physically exist (if, for example a user jumps to a page from another following a bookmark). Since *NG* is a complete representation of the information residing on the web logs, there is a high degree of replication of states in different parts of this structure.

The *NG* creation algorithm is as follows: For every user session *US* in the web logs, we create a path starting from the root of the tree. If a subsequence of the session already exists we update the weights of the respective edges, otherwise we create a new branch, starting from the last visited common page in the path. We note that any consecutive pages' repetitions have been removed from the user sessions during the data cleaning process; on the other hand, we keep any pages that have been visited more than once, but not consecutively. As already mentioned, we denote the end of a session using a special leaf-node. The algorithm for creating the *NG* is detailed in Figure 1.

```
Procedure CreateTree(U)
Input: User Sessions U
Output: Navigational Tree *NG
1. root <- NG;
2. tmpP <- root;
3. for every US∈U do
4.  while US ≠ ∅ do
5.   s_i = first_state(US);
6.   if parent(tmpP,s_i) then
7.     w_tmpP,I = w_tmpP,I + 1;
8.     tmpP <- s_i;
9.     US <- remove(US, s_i);
10.  else
11.    addchild(tmpP,s_i);
12.    w_tmpP,I = 1;
13.    tmpP <- s_i;
14.    US <- remove(US, s_i);
15.  endif
16.  if parent(tmpP,E) then
17.     w_tmpP,E = w_tmpP,E + 1;
18.  else
19.    addchild(tmpP,E);
20.    w_tmpP,E = 1;
21.  endif
22. done
23. tmpP <- NG;
24.done
```

Fig. 1. *NG* Creation Algorithm

In order to make this process clearer, we present a simple example. Assume that the user sessions of a web site are those included in Table 1. The Navigational Graph created after applying the aforementioned algorithm is depicted in Figure 2.

Table 1. User Sessions

User Session #	Path
1	a → b → c → d
2	a → b → e → d
3	a → c → d → f
4	b → c → b → g
5	b → c → f → a

3.2 Markov Models

As already stated, *NG* can become large as it contains redundant information (such as recurring sub-paths). As a consequence, performing computations directly over the *NG* can become prohibitively expensive. The need for reduced complexity and online availability imposes the creation of *NG synopses*, for reducing the *NG* structure size. These synopses capture the sequential dependence

between visits up to some level, while preserving their most important statistical characteristics. The more detailed is the synopsis, the more accurate will be the representation of *NG*. On the other hand, the construction of a less detailed synopsis will save time and computational power. In this paper we elaborate on Markov models since these are the synopses used in our proposed frameworks and experimental study.

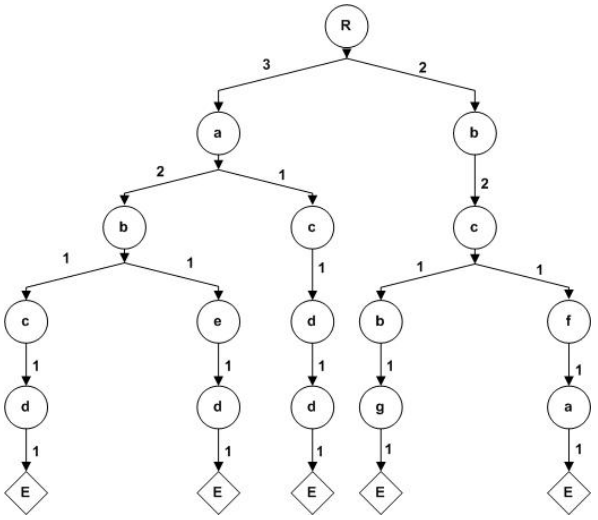

Fig. 2. Navigational Graph

The order of the Markov model defines the "memory" of the prediction, i.e. denotes the number of previous user steps that are taken into consideration in the probabilities' calculation. Therefore, in the case of Markov Chains, the visit to a page depends only on the previous one, in 2nd-order Markov models depends on the previous two, and so on. The selection of the order influences both the prediction accuracy and the complexity of the model while heavily depends on the application/data set. Since these issues fall out of the scope of this paper, we only provide an overview here. More on Markov model synopses can be found in [Eirinaki et. al. 2005].

In general, given that the user is currently at page x_i and has already visited pages $x_{i_{n-1}},...,x_{i_0}$, then, for an m^{th} – order Markov model, the probability of visiting page x_j, $P_{i,j}^{(m)}$ is based only on pages $x_i, x_{i_{n-1}},...,x_{i_{n-m+1}}$ and is given by Equation 1:

$$P_{i,j}^{(m)} = P\left(X_{n+1} = x_j \mid X_n = x_i, X_{n-1} = x_{i_{n-1}},...,X_0 = x_{i_0}\right) = P\left(X_{n+1} = x_j \mid X_n = x_i,...,X_{n-m+1} = x_{i_{n-m+1}}\right) \quad (1)$$

where the bounded probability of $\{X_{n+1}\}$, given all the previous events, is estimated by the bounded probability of $\{X_{n+1}\}$ given the m previous events. These transition probabilities are easily estimated using the information residing on *NG*. We define the one-step transition probability matrix *TP* as

follows: each item $TP_{i,j}$ represents the probability of transitioning from page(s) x_i to page x_j in one step. In other words,

$$TP_{i,j} = P(x_j \mid x_i) = \frac{w_{i \to j}}{w_i} \tag{2}$$

where w_i represents the total number of visits to page(s) x_i, and $w_{i \to j}$ represents the number of consecutive visits from x_i to x_j. Note that in case of paths having length l>1, we denote as x_i the prefix containing the first l-1 pages.

Table 2. Path Frequencies

l = 1		l = 2		l = 3	
x_i	w_i	$x_i \to x_j$	w_{ij}	$x_i \to x_j$	w_{ij}
a	4	a → b	2	a → b → c	1
b	5	a → c	1	a → b → e	1
c	4	b → c	3	a → c → d	1
d	3	b → e	1	b → c → b	1
e	1	b → g	1	b → c → d	1
f	2	c → b	1	b → c → f	1
g	1	c → d	1	b → e → d	1
		c → f	1	c → b → g	1
		d → f	1	c → d → f	1
		e → d	1	c → f → a	1
		f → a	1		

Table 2 includes the paths of length $l \leq 3$ corresponding to the user sessions included in Table 1. Using this information, and based on the previous analysis, we can compute the transition probabilities for the *NG* synopses based on 1st and 2nd-order Markov models. The respective 1st-order Markov model (Markov Chain) synopsis is depicted in Figure 3. The numbers in parentheses in the nodes denote the number of visits to a page whereas the edges' weights denote the number of times the respective link was followed. Nodes *S* and *E* represent the paths' start and end points respectively.

In the analysis that follows we use Markov models in two different frameworks. In the first, presented in Section 5, we use them in order to synopsize the *NG*, prior to applying the proposed localized personalized ranking algorithm *l-UPR*. In the second, presented in Section 6, we propose Markov model-based hybrid predictive models that incorporate link analysis techniques.

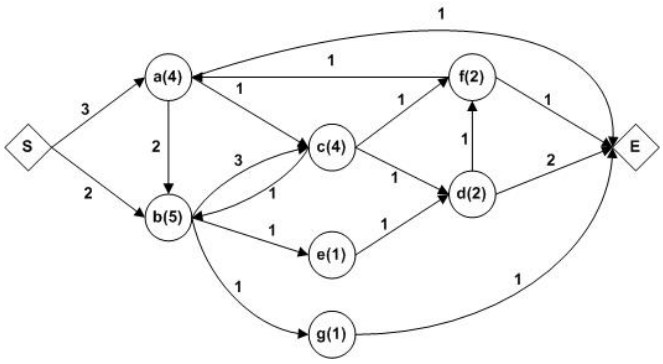

Fig. 3. *NG* synopsis (Markov Chain)

4. USAGE-BASED PAGERANK

So far, link analysis has been largely used in the context of web search. In this paper, we introduce link analysis techniques in the web personalization process. We propose *UPR*, a hybrid PageRank-style algorithm for ranking the pages of a web site based on its links' connectivity as well as its usage, in order to assist the recommendation process. In what follows we present the original PageRank algorithm as proposed by Brin and Page [1998]. We then provide the formal definition of the proposed algorithm, Usage-based PageRank (*UPR*).

4.1 PageRank

The PageRank algorithm is the most popular link analysis algorithm, used for assigning numerical weightings to web documents that are used from web search engines in order to rank the retrieved results. The algorithm models the behavior of a random surfer, who either chooses an outgoing link from the page he is currently visiting, or "jumps" to a random page. Each choice bears a probability. The PageRank of a page is defined as the probability of the random surfer visiting this page at some particular time step $k > K$ ($K \in Z^+$). This probability is correlated with the importance of this page, as it is defined based on the number and the importance of the pages linking to it. For sufficiently large K this probability is unique, as illustrated in what follows.

Consider the web as a directed graph G, where the N nodes represent the web pages and the edges represent the links between them. The random walk on G induces a Markov Chain where the states are given by the nodes in G, and M is the stochastic transition matrix with m_{ij} describing the one-step transition from page x_j to page x_i. The adjacency function m_{ij} is 0 if there is no direct link from x_j to x_i, and normalized such that, for each j:

$$\sum_{i=1}^{N} m_{ij} = 1 \tag{3}$$

As stated by the Perron-Frobenius theorem, if M is irreducible (i.e. G is strongly connected) and aperiodic, then M^k (i.e. the transition matrix for the k-step transition) converges to a matrix in which each column is the unique stationary distribution $P\vec{R}^*$, independent of the initial distribution $P\vec{R}$. The stationary distribution is the vector which satisfies the equation:

$$P\vec{R}^* = M \times P\vec{R}^* \tag{4}$$

in other words $P\vec{R}^*$ is the dominant eigenvector of the matrix M.

Since M is the stochastic transition matrix over the web graph G, PageRank is in essence the stationary probability distribution over pages induced by a random walk on G. As already implied, the convergence of PageRank is guaranteed only if M is irreducible and aperiodic [Motwani and Raghavan 1995]. The latter constraint is guaranteed in practice in the web context, since the visits to a web page do not usually follow a periodic pattern. The irreducibility is satisfied by adding a damping factor (1-ε) to the rank propagation (the damping factor is a very small number, usually set to 0.15), in order to limit the effect of rank sinks and guarantee convergence to a unique vector. We therefore define a new matrix M' by adding low-probability transition edges between every pair of nodes in G:

$$M' = (1-\varepsilon)M + \varepsilon U \tag{5}$$

In other words, the user may follow an outgoing link, or choose a random destination (usually referred to as random jump) based on the probability distribution of U. The latter process is also known as teleportation. PageRank can then be expressed as the unique solution to Equation 4, if we substitute M with M':

$$P\vec{R} = (1-\varepsilon)M \times P\vec{R} + \varepsilon\vec{p} \tag{6}$$

where $\vec{p}$ is a non-negative N-vector whose elements sum to 1.

Usually $m_{ij} = \frac{1}{\sum_{x_k \in Out(x_j)} |x_k|}$, where $Out(x_j)$ is the set of pages pointed to by x_j, and $U = \left[\frac{1}{N}\right]_{N\times N}$, i.e. the probability of randomly jumping to another page is uniform. In that case $\vec{p} = \left[\frac{1}{N}\right]_{N\times 1}$. By choosing, however, U, and consequently $\vec{p}$, to follow a non-uniform distribution, we can bias the PageRank vector computation to favor certain pages (therefore the "random" jump is no longer

random!) Thus, $\vec{p}$ is usually referred to as the personalization vector. This approach is largely used in the web search engines' context, where the ranking of the retrieved results are biased by favoring pages relevant to the query terms, or the user preferences to certain topic categories [Aktas et. al. 2004, Haveliwala 2002, Richardson and Domingos 2002]. In what follows, we present *UPR*, a usage-based personalized version of PageRank algorithm, used for ranking the pages of a web site based on the navigational behavior of previous visitors.

4.2 *UPR*: Link Analysis on the Navigational Graph

Based on the intuition that a page is important in a web site if many users have visited it before, we introduce the hybrid link analysis algorithm *UPR* (Usage-based PageRank). *UPR* extends the traditional link analysis algorithm PageRank, by biasing the page ranking with knowledge acquired from previous user visits, as they are recorded in the user sessions. In order to perform this, we define both the transition matrix *M* and the personalization vector $\vec{p}$ in such way that the final ranking of the web site's pages is strongly related to the frequency of visits to them.

Recapitulating from Section 3.1, we define the directed navigational graph *NG*, where the nodes represent the web pages of the web site *WS* and the edges represent the consecutive one-step paths followed by previous users. Both nodes and edges carry weights. The weight w_i on each node represents the number of times page x_i was visited and the weight $w_{j \to i}$ on each edge represents the number of times x_i was visited immediately after x_j. We denote the set of pages pointed to by x_j (outlinks) as $Out(x_j)$, and the set of pages pointing to x_j (inlinks) as $In(x_j)$.

Following the properties of the Markov theory and the PageRank computation, the Usage-based PageRank vector $\vec{UPR}$ is the solution to the following equation:

$$\vec{UPR} = (1-\varepsilon)M \times \vec{UPR} + \varepsilon\vec{p} \tag{7}$$

The transition matrix *M* on *NG* is defined as the square *N* x *N* matrix whose elements m_{ij} equal to 0 if there does not exist a link (i.e. visit) from page x_j to x_i and

$$m_{ij} = \frac{w_{j \to i}}{\sum_{x_k \in Out(x_j)} w_{j \to k}} \tag{8}$$

otherwise. The personalization vector $\vec{p}$ is defined as:

$$\vec{p} = \left[\frac{w_i}{\sum_{x_j \in WS} w_j} \right]_{N \times 1} \quad (9)$$

Using the aforementioned formulas, we bias the PageRank calculation to assign a higher rank to the pages that were visited more often by users in the past. We then use this hybrid ranking, combining the structure and the usage data of the site, to provide a ranked recommendation set to current users, as we describe in the subsequent sections.

Note that Equation 3 holds, that is, *M* is normalized such that the sum of each column equals to 1, therefore *M* is a stochastic transition matrix, as required for the convergence condition of the algorithm to hold. *M* is, as already mentioned, aperiodic in the web context and irreducible since we have included the damping factor *(1-ε)*. It is therefore guaranteed that Equation 7 will converge to a unique vector, $U\vec{P}R^*$.

Definition (*UPR*): We define the usage-based PageRank UPR_i of a web page x_i as the *n*-th iteration of the following recursive formula:

$$UPR_i^{\,n} = \varepsilon \sum_{x_j \in In(x_i)} \left(UPR_j^{\,n-1} \times \frac{w_{j \to i}}{\sum_{x_k \in Out(x_j)} w_{j \to k}} \right) + (1-\varepsilon) \frac{w_i}{\sum_{x_j \in WS} w_j} \quad (10)$$

Each iteration of *UPR* has complexity $O(n^2)$. The total complexity is thus determined by the number of iterations, which in turn depends on the size of the dataset. In practice, however, PageRank (and accordingly *UPR*) gives good approximations after 50 iterations for ε=0.85 (which is the most commonly used value, recommended in [Brin and Page 1998]). The computations can be accelerated by applying techniques such as those described in [Kamvar et. al. 2003a, Kamvar et. al. 2003b] even though it is not necessary in the proposed frameworks since *UPR* is applied to a single web site, therefore it converges after a few iterations.

In the Sections that follow, we present how *UPR* can be applied in different personalization frameworks in order to assist the recommendations process.

5. LOCALIZED UPR (*l*-UPR*)*

The *UPR* algorithm can be applied to a web site in order to rank its web pages taking into consideration both its link structure and the paths followed by users, as recorded in the web logs. This process results

in a "global" usage-based ranking of the web site's pages. In the context of web site personalization, however, we want to "bias" this algorithm further, focusing on the path the current visitor has followed and the most probable "next" pages he might visit, i.e. generating a "localized" personalized ranking. We select a small subset of the *NG* synopsis we have modeled the user sessions with, based on the current user's path. This sub-graph includes all the subsequent (to the current visit) pages visited by users with similar behavior in the past, until a predefined path depth d. Therefore, it includes all the potential "next" pages of the current user's visit. *l-UPR* (localized *UPR*) is in essence the application of *UPR* on this small, personalized fraction of the navigational graph. The resulting ranking is used in order to provide recommendations to the current visitor. This approach is much faster than applying *UPR* to the *NG* synopsis since the size of the graph is dramatically reduced, therefore enabling online computations. Moreover, the ranking results are personalized for each individual user, since they are based on the current user's visit and similar users' behavior in the past. We present the process of creating the personalized sub-graph, termed *prNG*, and the recommendation process in more detail below.

5.1 The Personalized Navigational Graph *(prNG)*

In short, the process of constructing the personalized sub-graph is as follows: We expand (part of) the path already visited by the user, including all the outgoing links (i.e. the pages and the respective weighted edges) existing in the *NG* synopsis. The length of the user path taken into consideration when expanding the graph depends on the *NG* synopsis we have used (in the case of Markov model synopses this represents the desired "memory" of the system). We subsequently perform this operation for the new pages (or paths), until we reach a predefined expansion depth. We then remove any pages that have already been visited by the user, since these don't need to be included in the generated recommendations. The children of the node (page) that is removed are linked to its parent. This ensures that all the previously visited pages by users having similar behavior will be kept in the final sub-graph, without including any higher-level pages they might have used as hubs for their navigation. After reaching the final set of nodes, we normalize each node's outgoing edge weights.

Before proceeding with the technical details of this algorithm, we illustrate its functionality using two examples, based on the sessions included in Table 1, and the respective path frequencies of Table 2. In both examples we create the *prNG*s for two user visits including the paths $\{a \rightarrow b\}$ and $\{b \rightarrow c\}$. In the first example, illustrated in Figure 4, we assume that the sessions are modeled using a Markov

Chain *NG* synopsis. Using the path frequencies for *l*=2 (i.e. the one-step transitions), we expand the two paths, $\{a \rightarrow b\}$ and $\{b \rightarrow c\}$, to create the respective *prNG*s. After expanding the sub-graph twice (depth = 2), we remove the pages previously visited by the users, i.e. page *b* from the first sub-graph, as shown in Figure 4a and pages *b* and *c* from the second sub-graph, as shown in Figure 4b. We observe that the removed node *b* (Figure 4b) has two children, nodes *g* and *e*. These nodes are linked to *b*'s parent, node *c* (highlighted edges). We finally normalize the outgoing edge weights of each node. The second example, illustrated in Figure 5, is based on a 2nd-order Markov model *NG* synopsis. Note that in this case we use the path frequencies for *l*=3. Based on these frequencies, we expand the sub-graph twice (depth = 2). For example, the path $\{a \rightarrow b\}$ (Figure 5a) has two outgoing links pointing to pages *c* and *e.* We add these nodes to the sub-graph and subsequently expand the generated paths $\{b \rightarrow c\}$ and $\{b \rightarrow e\}$ to point to pages *b, f, d,* and page *d,* respectively. We then remove all the pages that have been visited before and link their children to the remaining sub-graph. For example, after removing node *b* (Figure 5b), we link node *g* to node *c* (highlighted edge). Finally, the outgoing edge weights of each node are normalized so that they sum to 1. We observe that the nodes included in each *prNG* depend on the *NG* synopsis we choose to model the user sessions with.

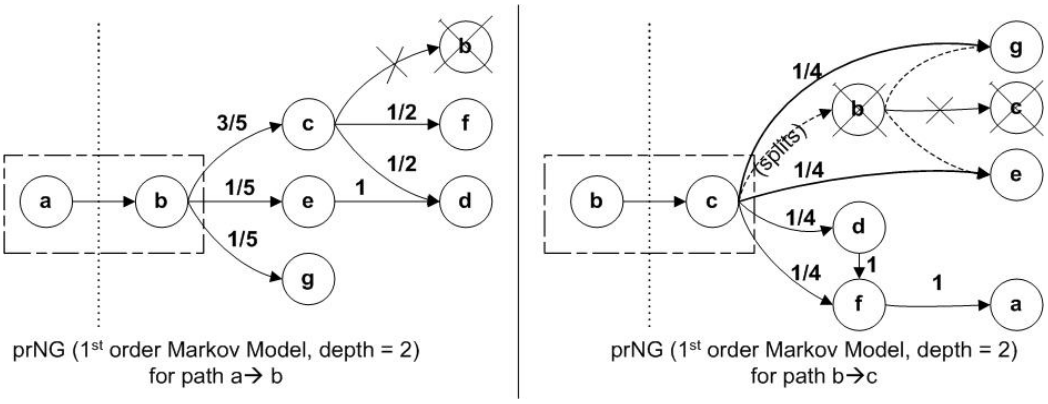

Fig.4a Fig. 4b

Fig. 4. *prNG* of Markov Chain *NG synopsis*

The *prNG* construction algorithm is presented in Figures 6 and 7. The algorithm complexity depends on the synopsis used, since the choice of the synopsis affects the time needed for locating the successive pages for expanding the current path. It also depends on the number of outgoing links of each sub-graph's page and the expansion depth, *d*. Therefore, if the complexity of locating successive pages in a synopsis is *k*, the complexity of the *prNG* creation algorithm is $O\left(k * fanout(NG)^d\right)$, where

fanout(NG) is the maximum number of a node's outgoing links in *NG*. In the case of Markov model synopses, *k*=1 since the process of locating the outgoing pages of a page or path reduces to the lookup in a hash table.

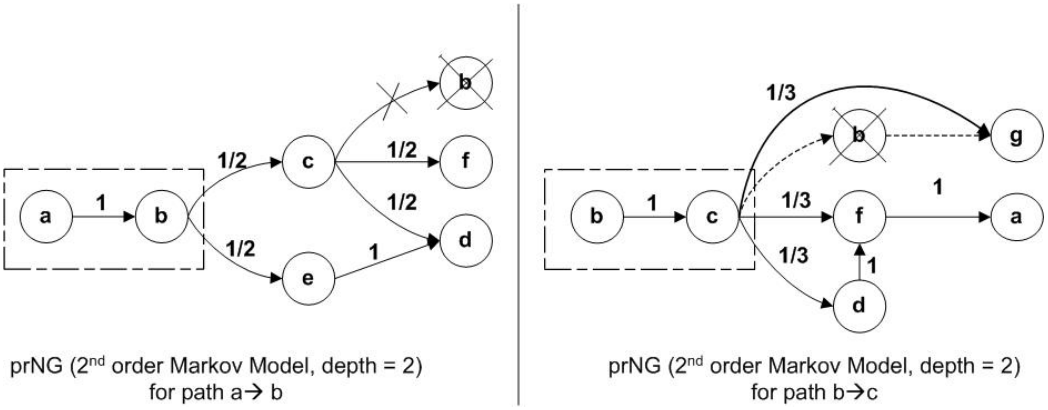

Fig. 5a Fig. 5b

Fig. 5. *prNG* of 2nd-order Markov model *NG synopsis*

```
Procedure Create_prNG(CV, NG)
Input: Current User Visit CV, Navigational
Graph NG
Output: Subset of NG prNG
1. start
2.  CV = {vp};
3.  cp = lastVisitedPath(CV);
4.  expand(cp, NG, depth, expNG);
5.  removeVisited(expNG, CV);
6.  updateEdges(expNG);
7.  prNG = normalize(expNG);
8. end
```

Fig. 6. Construction of *prNG*

```
Procedure expand(cp, NG, d, eNG)
Input: last page/path visited cp, navigational
graph synopsis NG, depth of expansion d
Output: expanded navigational graph eNG
1. start
2. P := cp;
3. R:= rootNode(eNG);
4. tempd = 0;
5. addNode(eNG, R, cp);
6. while (tempd <= d)do
7.  for every (p∈P of same level)do
8.   forevery np = linksto(NG, p, np, w)do
9.    addNode(enG, p, np, w);
10.   P += np;
11.  done;
12. done;
13. tempd +=1;
14.done;
15.end
```

Fig. 7. Path expansion subroutine

We finally apply the *UPR* algorithm to *prNG,* in order to rank all the possible "next" pages that are contained in this personalized navigational sub-graph. *prNG* should be built so as to retain the desirable attributes for *UPR* to converge. The irreducibility of the sub-graph is always satisfied since we have added the damping factor *(1-ε)* in the rank propagation. Moreover, Equation 3 which states that the sum of all outgoing edges' weights of every node in the sub-graph equals to 1, is satisfied since we normalize them. Note here that *prNG* does not include any previously visited pages.

Definition (*l-UPR*): We define $l\text{-}UPR_i$ of a page x_i as the *UPR* rank value of this page in the personalized sub-graph *prNG*.

These *l-UPR* rankings of the candidate pages are subsequently used to generate a personalized recommendation set to each user. This process is explained in more detail in the following Section.

5.2 *UPR*-based Personalized Recommendations

The application of *UPR* or *l-UPR* to the navigational graph results in a ranked set of pages which are subsequently used for recommendations. As already presented, the final set of candidate recommendation pages can be either personalized or global, depending on the combination of algorithm - navigational graph chosen:

1. Apply *l-UPR* to *prNG*. Since *prNG* is a personalized fraction of the *NG* synopsis, this approach results in a "personalized" usage-based ranking of the pages most likely to be visited next, based on the current user's path.
2. Apply *UPR* to *NG* synopsis. This approach results in a "global" usage-based ranking of all the web site's pages. This global ranking can be used as an alternative if the personalized ranking does not generate any recommendations. It can also be used for assigning page probabilities in the context of other probabilistic prediction frameworks, as we will describe in the Section that follows.

Finally, another consideration would be to have a pre-computed set of recommendations for all popular paths in the web site, in order to save time during the online computations of the final recommendation set.

6. WEB PATH PREDICTION USING HYBRID PROBABILISTIC PREDICTIVE MODELS

One of the most popular web usage mining methods is the use of probabilistic models. Such models represent the user sessions as a graph whose nodes are the web site's pages and edges are the hyperlinks between them. In essence, they are based in what we have already described as *NG* synopses. Using the transitional probabilities between pages as defined by the probabilistic model, a path prediction is made by selecting the most probable path among candidate paths, based on each user's visit. Such purely usage-based probabilistic models, however, present certain shortcomings. Since the prediction of users' navigational behavior is solely based on the usage data, the structural properties of the web graph are ignored. Thus important paths may be underrated. Moreover, as we will also see in the experimental study we performed, such models are often shown to be vulnerable to the training data set used.

In this Section we present a hybrid probabilistic predictive model (*h-PPM*) that extends Markov models by incorporating link analysis methods. More specifically, we choose the Markov models as *NG* synopses and use *UPR* and two more PageRank-style variations of it, for assigning prior probabilities to the web pages based on their importance in the web site's web and navigational graph.

6.1 Popular Path Prediction

As already presented in Section 3.2, Markov models provide a simple way to capture sequential dependence when modeling the navigational behavior of the users of a web site. After building the model, i.e. computing the transition probabilities, the path probabilities are estimated using the chain rule. More specifically, for an m^{th}-order Markov model, the path probability of following the path $x_1 \to x_2 \to ... \to x_k$ equals to:

$$P(x_1 \to x_2 \to ... \to x_k) = P(x_1) * \prod_{i=2}^{k} P(x_i \mid x_{i-m} ... x_{i-1}) \tag{11}$$

For example, using a Markov Chain as the prediction model, the probability of the path $\{a \to b \to c\}$ reduces to $P(a \to b \to c) = P(a)P(b \mid a)P(c \mid b) = P(a)\frac{P(a \to b)}{P(a)}\frac{P(b \to c)}{P(b)}$.

Equation 11 is used in order to predict the page that has higher probability of being visited by the user in the next step. Assuming that the current path of the user has length *l,* this is performed by estimating the probabilities of all the paths of length *l+1* that have the current user path as prefix, and choosing the suffix of the most probable path. Consider, for example, the user sessions of Table 1 and

the respective Markov Chain illustrated in Figure 3. Assuming that the user has already visited the path $\{a \rightarrow b\}$, we first estimate and compare the probabilities $P(a \rightarrow b \rightarrow c)$, $P(a \rightarrow b \rightarrow e)$, and $P(a \rightarrow b \rightarrow g)$ using Equation 11. Since $P(a \rightarrow b \rightarrow c)$ is higher than the other two, we predict that the next visit of the user will be page *c*. The bounded probabilities' computation is straightforward since it reduces to a lookup on the transition probability matrix *TP*. On the other hand, the prior probability assignment is an open issue, and we deal with it in the sequel.

6.2 Reconsidering Prior Probabilities' Computation

There are three approaches used commonly for assigning initial probabilities (priors) to the nodes of a Markov model. The first one assigns equal probabilities to all nodes (pages). The second estimates the initial probability of a page *p* as the ratio of the number of visits on *p* as a first page in a path, to the total number of user sessions. In the case of modeling web navigational behavior, however, neither of the aforementioned approaches provides accurate results. The first approach assumes a uniform distribution, favoring non-important web pages. On the other hand, the second does exactly the opposite: favors only top-level "entry" pages. Furthermore, in the case of a page that was never visited first, its prior probability equals to zero. The third approach is more "objective" with regards to the other two, since it assigns prior probabilities proportionally to the frequency of total visits to a page. This approach, however, does not handle important, yet new (i.e. not included in the web usage logs) pages. Finally, as shown in the experimental evaluation, all approaches are very vulnerable to the training data used for building the predictive model.

In the literature, a few approaches exist where the authors claim that these techniques are not accurate enough and define different priors. Sen and Hansen [2003] use Dirichlet priors, whereas Borges and Levene [2004] define a hybrid formula which combines the two options (taking into consideration the frequency of visits to a page as the first page, or the total number of visits to the page). For this purpose, they define the variable α, which ranges from 0 (for page requests as first page) to 1 (for total page requests). In their experimental study, however, they don't explicitly refer to the optimal value they used for α.

In this paper, we address such shortcomings following an alternative approach. Our motivation draws from the fact that the initial probability of a page should reflect the importance of this page in the web navigation. We propose the integration of the web site's topological characteristics, as represented by its link structure, with the navigational patterns of its visitors used for computing these probabilities.

More specifically, we propose the use of three PageRank-style ranking algorithms for assigning prior probabilities. The first (*PR*) is the PageRank algorithm applied on the web site's graph, and computes the page prior probabilities based solely on the link structure of the web site. The second is *UPR*, which, as already described, is applied on the web site's navigational graph and "favors" pages previously visited by many users. The third algorithm (*SUPR*) is a variation of *UPR*, which assigns uniform probabilities to the random jump instead of biasing it as well.

Definition (PageRank-based Prior Probability): We define the prior probability $P(x_i)$ of a page x_i as:

$$P(x_i) = P^{(n)}(x_i) = (1-\varepsilon) * p(x_i) + \varepsilon \sum_{x_k \in In(x_i)} \left(P^{(n-1)}(x_k) * p(x_k, x_i) \right) \quad (12)$$

with $(1-\varepsilon)$ being the damping factor (usually set to 0.15) and for

(i) ***PR* (PageRank):**

$$p(x_i) = \frac{1}{N} \quad \text{and} \quad p(x_k, x_i) = \frac{1}{\sum_{x_j \in Out(x_k)} x_j} \quad (13)$$

(ii) ***SUPR* (Semi-Usage PageRank):**

$$p(x_i) = \frac{1}{N} \text{ and } p(x_k, x_i) = \frac{w_{ki}}{\sum_{x_j \in Out(x_k)} w_{kj}} \quad (14)$$

(iii) ***UPR* (Usage PageRank):**

$$p(x_i) = \frac{w_i}{\sum_{x_j \in WS} w_j} \text{ and } p(x_k, x_i) = \frac{w_{ki}}{\sum_{x_j \in Out(x_k)} w_{kj}} \quad (15)$$

where N is the number of pages in the web site.

Any of the aforementioned ranking schemes can be applied on the web site's web or navigational graph (or its synopsis), resulting in a probability assignment for each one of its pages. These probabilities can be subsequently used instead of the commonly used priors for addressing the aforementioned problems. As we present in the experimental study we have performed, this approach provides more objective and precise predictions than the ones generated from the pure usage-based approaches.

7. EXPERIMENTAL EVALUATION

In this Section we present a set of experiments we performed in order to evaluate the performance of both recommendation frameworks proposed in this paper. In the case of *l-UPR*, since there is no previous related work to compare it with, we use two different setups of Markov Chains, which is the *NG synopsis* we used in *l-UPR* setup too. Using all three setups, we generate top-3 and top-5 recommendation sets for 10 representative user paths, and compare them to the actual paths the users followed. In order to evaluate the incorporation of page ranking in the hybrid probabilistic predictive models (*h-PPM*), we compare the top-*n* path rankings generated by five different setups with the n most frequent paths. For our experiments, we use two different data sets in order to examine how the proposed methods behave in various types of web sites.

7.1 Experimental Setup

In our experiments we used two publicly available data sets. The first one includes the page visits of users who visited the "msnbc.com" web site on 28/9/99 [MSNBC]. The visits are recorded at the level of URL category (for example sports, news, etc.). It includes visits to 17 categories (i.e. 17 distinct pageviews). We selected 96.000 distinct sessions including more than one and less than 50 page visits per session and split them in two non-overlapping time windows to form a training (65.000 sessions) and a test (31.000 sessions) data set. The second data set includes the sessionized data for the DePaul University CTI web server, based on a random sample of users visiting the site for a two week period during April 2002 [CTI]. The data set includes 683 distinct pageviews and 13.745 distinct user sessions of length more than one. We split the sessions in two non-overlapping time windows to form a training (9.745 sessions) and a test (4.000 sessions) data set. We will refer to these data sets as *msnbc* and *cti* data set respectively. We chose to use these two data sets since they present different characteristics in terms of web site context and number of pageviews. More specifically, *msnbc* includes the visits to a very big portal. That means that the number of sessions, as well as the length of paths is very large. This data set has, however, the characteristic of very few pageviews, since the visits are recorded at the level of page categories. We expect that the visits to this web site are almost homogeneously distributed among the 17 different categories. On the other hand, the *cti* data set refers to an academic web site. Visits to such sites are usually categorized in two main groups: visits from students looking for information concerning courses' or administrative material, and visits from researchers seeking

information on papers, research projects, etc. We expect that the recorded visits will imply this categorization.

Since in all the experiments we generate top-*n* rankings, in the evaluation step we used two metrics commonly used for comparing two top-n rankings r_1 and r_2. The first one, denoted as $OSim(r_1,r_2)$ [Haveliwala 2002] indicates the degree of overlap between the top-n elements of two sets *A* and *B* (each one of size *n*) to be:

$$OSim(r_1, r_2) = \frac{|A \cap B|}{n} \tag{16}$$

The second, $KSim(r_1,r_2)$ is based on Kendall's distance measure [Kendall and Gibbons 1990] and indicates the degree to which the relative orderings of two top-n lists are in agreement and is defined as:

$$KSim(r_1, r_2) = \frac{|(u,v) : r_1', r_2' \text{ have same ordering of } (u,v), u \neq v|}{|A \cap B|(|A \cap B| - 1)} \tag{17}$$

where r_1' is an extension of r_1, containing all elements included in r_2 but not r_1 at the end of the list (r_2' is defined analogously) [Haveliwala 2002]. In other words, *KSim* takes into consideration only the common items of the two lists, and computes how many pairs of them have the same relative ordering in both lists. It is obvious that *OSim* is more important (especially in small rankings) since it indicates the concurrence of predicted pages with the actual visited ones. On the other hand, *KSim* must be always evaluated in conjunction with the respective *OSim* since it can take high values even when only a few items are common in the two lists.

7.2 *l-UPR* Recommendations' Evaluation

As already mentioned, the choice of the *NG* synopsis we use to model the user sessions is orthogonal to the *l-UPR* framework. In this Section, we present results regarding the impact of using our proposed method instead of pure usage-based probabilistic models, focusing on Markov Chains.

We used 3 different setups for generating recommendations. The first two, referred to as *Start* and *Total*, are both based on Markov Chains. Their difference lies on the approach used in order to estimate the prior probabilities of the model. More specifically, *Total* assigns prior page probabilities proportional to the total page visits, whereas *Start* assigns prior page probabilities proportional to the visits beginning with this page. The third setup, referred to as *l-Upr*, is in essence our proposed

algorithm applied to a Markov Chain-based *prNG*. For the *l-Upr* setup, we set the damping factor *(1-ε)* to 0.15 and the number of iterations to 100 to ensure convergence. We expand each path to depth *d*=2.

The experimental scenario is as follows: We select the 10 most frequent paths comprising of two or more pages from the test data set. For each such path *p*, we make the assumption that it is the current path of the user and generate recommendations applying the aforementioned approaches to the training data set. Using the first two setups, we find the *n* pages having higher probability to be visited after *p*. On the other hand, using our approach, we expand *p* to create a localized sub-graph and then apply *l-UPR* to rank the pages included in it. We then select the top-*n* ranked pages. This process results in three recommendation sets for each path *p*. At the same time, we identify, in the test data set, the *n* most frequent paths that extend *p* by one more page. We finally compare, for each path *p*, the generated top-*n* page recommendations of each method (*Start, Total, l-Upr*) with the *n* most frequent "next" pages, using the *OSim* and *KSim* metrics. We should note that, even though we selected the 10 most representative paths for these experiments, our framework can generate recommendations for any given path.

We run the experiments generating top-3 and top-5 recommendation lists for each setup. We performed the experiments using small recommendation sets because this resembles more to what happens in reality, i.e. the system recommends only a few "next" pages to the user. The diagrams presented here, show the average *OSim* and *KSim* similarities over all 10 paths.

Figure 8 depicts the average *OSim* and *KSim* values for the top-3 and top-5 rankings generated for the *msnbc* data set. In the first case (top-3 page predictions) we observe that *l-Upr* behaves slightly worse in terms of prediction accuracy (*OSim*) but all methods achieve around 50% accuracy. The opposite result is observed in the second case (top-5 page predictions), where *l-Upr* behaves better in prediction accuracy than the other two methods, and the overall prediction accuracy is more than average. In both cases we observe a lower *KSim*, concluding that *l-Upr* managed to predict the "next" pages but not in the same order (as they were actually visited). As we mentioned earlier, however, the presentation order is not so important in such a small recommendation list. Overall, the differences between the three methods are insignificant. This can be justified if we take into account the nature of the data set used. As already mentioned, the number of distinct pageviews of the data set is very small and therefore the probability of coinciding in the predictions is the same, irrespective of the method used.

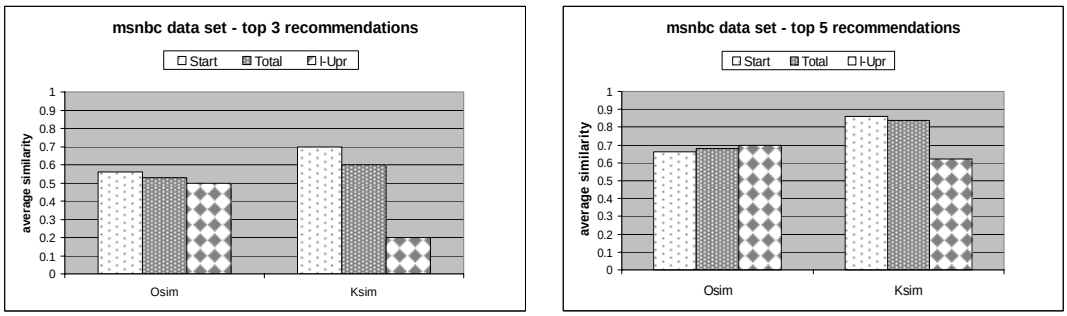

Fig. 8. Average *OSim* and *KSim* of top-*n* rankings for *msnbc* data set

In order to conclude on whether the number of distinct pageviews is the one affecting the prediction accuracy of the three methods, we performed the same experimental evaluation on the second data set, *cti*. Figure 9 depicts the average *OSim* and *KSim* values for the top-3 and top-5 rankings generated for the *cti* data set. We observe that in both cases *l-Upr* outperforms the other two methods both in terms of prediction accuracy (*OSim*) and relative ordering (*KSim*). This finding supports our intuition, that in the case of big web sites that have many pageviews, the incorporation of structure data in the prediction process enhances the accuracy of the recommendations.

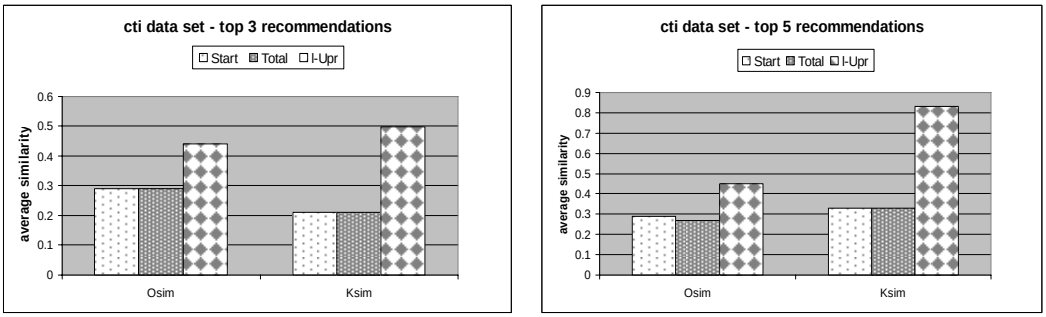

Fig. 9. Average *OSim* and *KSim* of top-n rankings for *cti* data set

Examining all findings in total, we verify our claim that *l-UPR* performs the same as, or better than commonly used probabilistic prediction methods. Even though the prediction accuracy in both experiments is around 50%, we should point out that this value represents the average *OSim* over 10 distinct top-*n* rankings. Examining the rankings individually, we observed a big variance in the findings, with some recommendation sets being very similar to the actually visited pages (*OSim* > 70%), whereas others being very dissimilar (*OSim* < 20%). The standard deviation of the experimental results is presented in Tables 3 and 4. We should also note that, the *NG* synopsis used in all three setups is the Markov Chain, which is the simplest synopsis model, yet the less accurate one. We expect better prediction accuracy if the algorithm is applied over a more accurate *NG* synopsis and leave this open for future work.

Table 3. *OSim* standard deviation

OSIM STDEV	*msnbc top-3*	*msnbc top-5*	*cti top-3*	*cti top-5*
Start	0.164	0.165	0.374	0.29
Total	0.176	0.169	0.374	0.289
l-Upr	0.179	0.17	0.236	0.2

Table 4. *KSim* standard deviation

KSIM STDEV	*msnbc top-3*	*msnbc top-5*	*cti top-3*	*cti top-5*
Start	0.483	0.163	0.395	0.47
Total	0.516	0.156	0.395	0.47
l-Upr	0.422	0.261	0.535	0.397

Overall, taking into consideration the low complexity of the proposed algorithm that enables the fast, online generation of personalized recommendations, we conclude that it is a very efficient alternative to pure usage-based methods.

7.3 *h-PPM* Recommendations' Evaluation

In order to evaluate the impact of incorporating link analysis methods in the probabilistic prediction process, we used 5 setups of the prediction model, differing in terms of the prior probabilities' computation. The first two setups, termed *Start* and *Total*, are the ones used in previous approaches for computing prior probabilities, as we already explained in the previous Section. More specifically, *Start* assigns probabilities proportional to the visits of a page in the beginning of the sessions, whereas *Total* assigns probabilities proportional to the total visits to a page. We do not include the approach of assigning uniform prior probabilities to all nodes, since it is shown to perform worse than the other two. The other three setups, termed *PR*, *SUPR*, and *UPR*, assign probabilities using the respective proposed algorithms defined in Section 6.2. We use two *NG* synopses for approximating the Navigational Graph *NG*, namely, the Markov Chain and the 2^{nd}-order Markov model. For the PageRank-style algorithms, the damping factor $(1-\varepsilon)$ was set to 0.15 and the number of iterations was set to 100.

Applying the five setups on the training data, we generated a list including the top-n most probable paths for $n \in \{3, 5, 10, 20\}$. We then compared these results with the top-n most frequent paths (i.e. the actual paths followed by the users), as derived from the test data.

The diagrams of Figures 10 and 11 depict the *OSim* and *KSim* similarities for the top 3, 5, 10, and 20 rankings of the *msnbc* data set, using a Markov Chain as *NG* synopsis and prediction model. We observe that *OSim* is around 60% for the two pure usage-based methods, *Start* and *Total*, whereas it is more than 80% for the three proposed methods. *KSim*, on the other hand, exceeds 90% for all rankings in the case of our proposed methods, whereas it is high only for the first three rankings for *Start* setup.

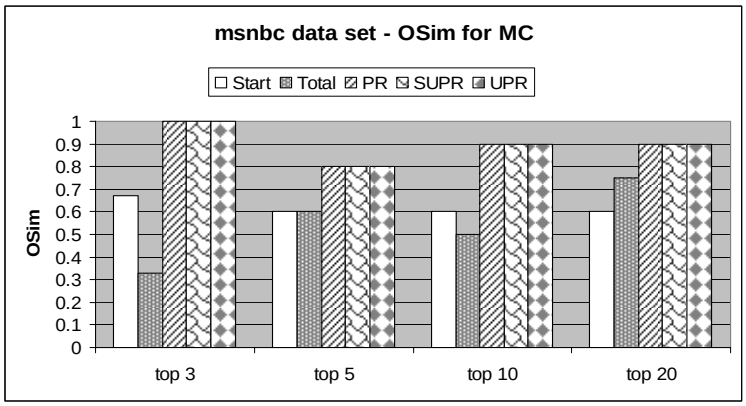

Fig. 10. *OSim* for *msnbc* data set, Markov Chain *NG synopsis*

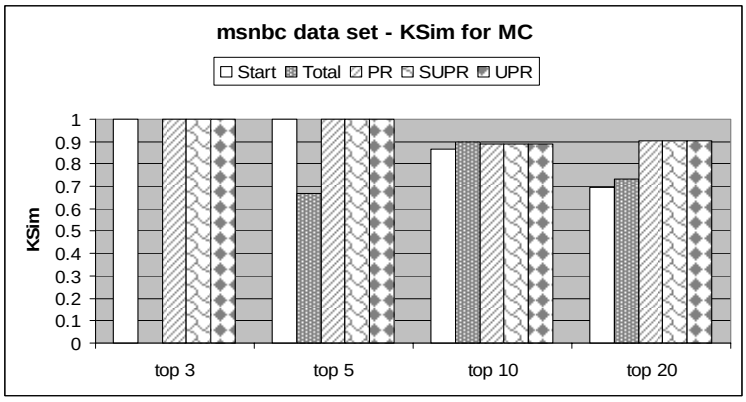

Fig. 11. *KSim* for *msnbc* data set, Markov Chain *NG synopsis*

The diagrams of Figures 12 and 13 depict the *OSim* and *KSim* similarities for the top 3, 5, 10, and 20 rankings of the *cti* data set. In this case, the rankings acquired by applying the two common methods did not match with the actual visits at all, giving a 0% *OSim* and *KSim* similarity! On the other hand, all three proposed methods reached an average of 80% *OSim* and 90% *KSim* in all setups, with *SUPR* slightly outperforming *PR* and *UPR*.

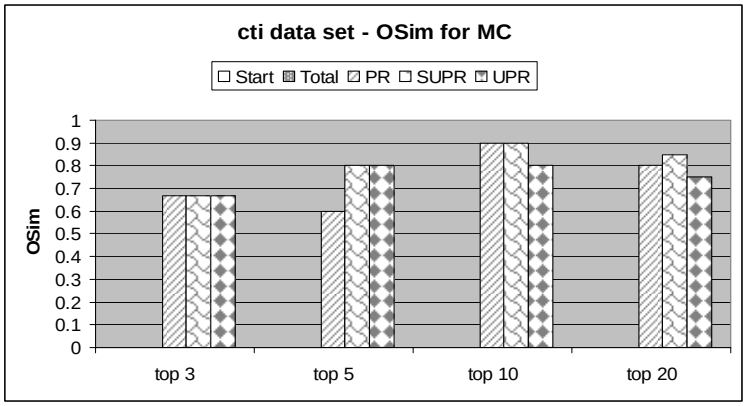

Fig. 12. *OSim* for *cti* data set, Markov Chain *NG synopsis*

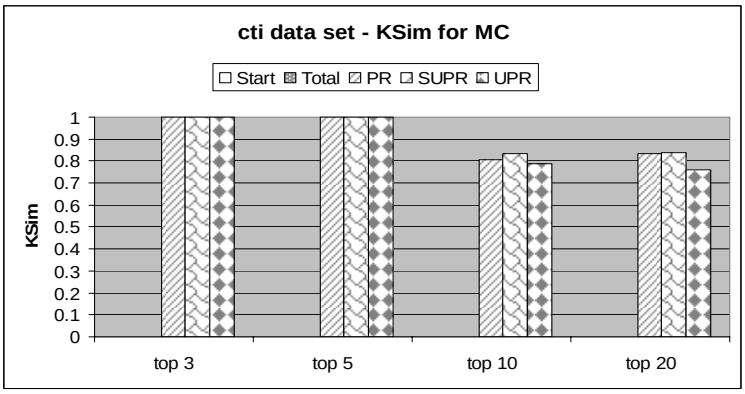

Fig. 13. *KSim* for *cti* data set, Markov Chain *NG synopsis*

At this point, we should analyze the behavior of the *Start* and *Total* setups, which represent the straightforward Markov model implementation. The outcomes of the experiments verify our claim that Markov models are very vulnerable to the training data used, and several pages may be overrated or underestimated in certain circumstances. In the case of the *msnbc* data set, where the number of distinct pages was very small and therefore the navigational paths were evenly distributed, the pure usage-based models seem to behave fairly (but, again, worse than the hybrid models). On the other hand, in the case of the *cti* data set, where hundreds of distinct pages (and therefore distinct paths) existed, the prediction accuracy of usage-based models was disappointing! We examined the produced top-*n* rankings of the two usage-based approaches, and observed that they include only the visits of students to course material. Since probably many students visited the same pages and paths in that period of time, accessing the pages directly (probably using a bookmark), these visits overlapped any other path visited by any other user. On the other hand, by taking into consideration the "objective" importance of a page, as conveyed by the link structure of the web site, such temporal influences are reduced.

The framework proposed in this section can be directly applied for computing the prior probabilities of visiting the pages of a web site. In other words, this framework can be directly applied to Markov Chain *NG* synopses. In the case of higher-order Markov models, however, our intuition was that this framework should be extended for supporting the computation of prior probabilities for path visits (up to some length, depending on the order). For instance, a 2^{nd}-order Markov model is based on the assumption that we have prior knowledge concerning the visit probabilities of all paths including up to 3 pages. Indeed, the results from applying the proposed algorithms to the *cti* dataset indicated the need for this model extension. In the case of the *msnbc* dataset, however, we did not observe any significant deviation of the results. This can be explained by the fact that *msnbc* has only a few distinct

nodes, hence a small number of different distinct paths a user can follow. As already mentioned, in this data set the users' visits were almost uniformly distributed across all web site's page categories. Therefore the probability of visiting two pages consecutively is very well approximated by the probability of visiting the last page (almost independent of the page the user was previously visiting). In what follows, we present the results of this experiment.

The results of the set of experiments we performed using the 2nd-order Markov models as *NG* synopsis on the *msnbc* data set are included in the diagrams of Figure 14 and 15.

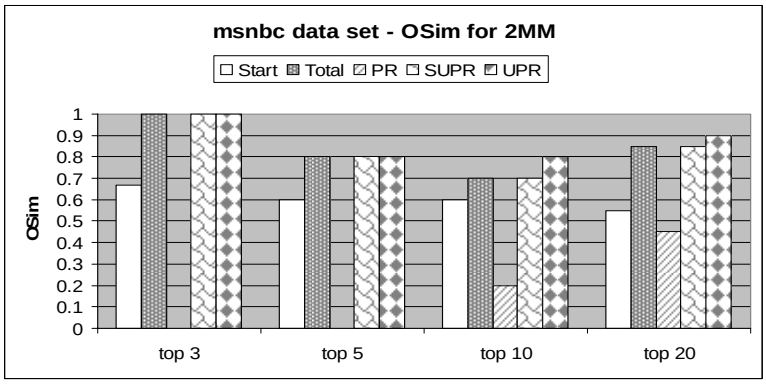

Fig. 14. *msnbc* data set, 2nd-order Markov model *NG synopsis*

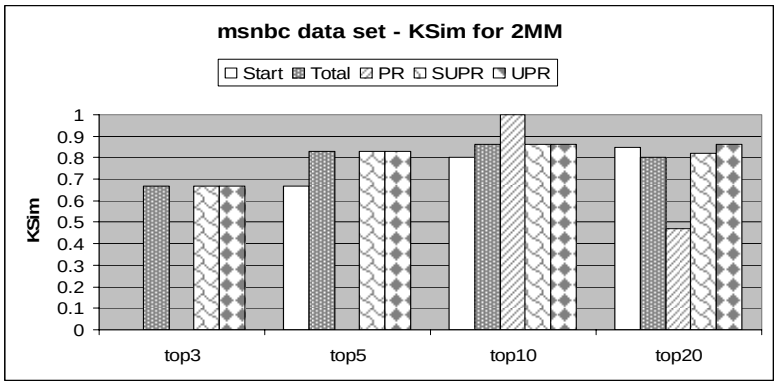

Fig. 15. *msnbc* data set, 2nd-order Markov model *NG synopsis*

We observe that in the case of 2nd-order Markov models the winner is *UPR* followed by *SUPR* and *Total* setups. A very interesting fact is that the pure link-based approach, *PR*, gives the worst results, having 0% *OSim* for the top-3 and top-5 rankings and only 20% *OSim* for the top-10 ranking. This can be explained by the fact that *PR*, which is in essence the application of PageRank algorithm on the web site's graph, represents the steady state vector of the Markov Chain, as it is defined on the web graph. Therefore, in the case of modeling the web graph as an *NG* synopsis other than the Markov Chain, it isn't as efficient. On the other hand, the hybrid usage/link ranking algorithms outperform the two commonly used usage-based approaches in most cases.

Overall, comparing the three proposed methods, we observe that, for the *msnbc* data set, all methods have the same *OSim* when a Markov Chain synopsis is used, whereas *UPR* outperforms the other two when a 2nd-order Markov model synopsis is used. On the other hand, in the case of the *cti* data set, we observe that *SUPR* outperforms the other two methods. Nevertheless, there is no prevalent underlying pattern between the number of recommendations and *OSim/KSim*. Therefore, we cannot conclude on the superiority of one of the proposed methods, other than that it strongly depends both on the data set and the *NG synopsis* used.

8. CONCLUSIONS

There is a wealth of recommendation models for personalizing a web site based on previous users' navigational behaviour. Most of the models, however, are solely based on usage data ignoring the link structure of the web graph visited. In this paper we study the integration of link analysis in the web personalization process. We propose a novel algorithm, *UPR*, which is applicable to any navigational graph synopsis, to provide ranked recommendations to the visitors of a web site, capitalizing on the structural properties of the navigation graph. We present *UPR* in the context of two different personalization frameworks. In the first a localized version of *UPR* is applied to a personalized sub-graph of the *NG synopsis* and is used to create online personalized recommendations to the visitors of the web site. The second approach addresses several shortcomings of pure usage-based probabilistic predictive models, by incorporating link analysis techniques in such models in order to support popular paths' prediction.

The experiments we have performed for both frameworks are more than promising, outperforming existing approaches. As we have already pointed out, the priors defined in *h-PPM* framework are directly applicable to Markov Chains, but do not always work for higher-order Markov models. We plan to extend the proposed framework on this regards. Our future plans also involve the application of *l-UPR* on different *NG synopses*. As shown in the experimental evaluation, *l-UPR* is a very promising recommendation algorithm. In our study we applied it on the Markov Chain *NG synopsis*. We expect better results in the case of more complex *NG synopses*, which approximate more accurately the navigational graph. Moreover, we plan to perform an experimental study of the two proposed frameworks with real users. Finally, we plan to investigate how this hybrid usage-structure ranking can be applied to a unified web/navigational graph which expands out of the limits of a single web site.

Such approach would enable a "global" importance ranking over the web, enhancing both web search results and the recommendation process.

REFERENCES

AKTAS, M.S., NACAR, AND M.A., MENCZER, F. 2004. Personalizing PageRank Based on Domain Profiles. In *Proceedings of WEBKDD 2004 Workshop*, Seattle, August 2004

BORGES, AND J., LEVENE, M. 2000. Data Mining of User Navigation Patterns. In *Revised Papers from the International Workshop on Web Usage Analysis and User Profiling*, LNCS Vol. 1836, 92-111

BORGES, J., AND LEVENE, M. 2004. A Dynamic Clustering-Based Markov Model for Web Usage Mining.*Technical Report*, available at http://xxx.arxiv.org/abs/cs.IR/0406032

BORGES, J., AND LEVENE, M. 2006. Ranking Pages by Topology and Popularity within Web Sites. Accepted for publication in *World Wide Web Journal*

BRIN, S., AND PAGE, L. 1998. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks*, 30(1-7): 107-117

CADEZ, I., HECKERMAN, D. MEEK, C., SMYTH, AND P., WHITE, S. 2000. Visualization of Navigation Patterns on a Web Site Using Model Based Clustering. In *Proceedings of ACM KDD2000 Conference*, Boston MA, 2000

CADEZ, I., GAFFNEY, AND S., SMYTH, P. 2006. A general probabilistic framework for clustering individuals and objects. In *Proceedings of ACM KDD2000 Conference*, Boston MA, 2000

CTI *DePaul web server data*, http://maya.cs.depaul.edu/~classes/ect584/data/cti-data.zip

DESHPANDE, M., AND KARYPIS, G. 2001. Selective Markov Models for Predicting Web-Page Accesses. In *Proceedings of the 1st SIAM International Conference on Data Mining*, 2001

EIRINAKI, M. 2004. Web Mining: A Roadmap. *Technical Report,* available at http://www.db-net.aueb.gr

EIRINAKI, M., AND VAZIRGIANNIS, M. 2003. Web Mining for Web Personalization. *ACM Transactions on Internet Technology (TOIT)*, 3(1), 1-29

EIRINAKI, M., VAZIRGIANNIS, M., AND KAPOGIANNIS, D. 2005. Web Path Recommendations based on Page Ranking and Markov Models. In *Proceedings of the 7th ACM International Workshop on Web Information and Data Management (WIDM 2005),* Bremen, Germany, November 2005

EIRINAKI, M., VAZIRGIANNIS, M., AND VARLAMIS, I. 2003. SEWeP: Using Site Semantics and a Taxonomy to Enhance the Web Personalization Process. In *Proceedings of ACM KDD2003 Conference*, Washington DC, August 2003

EL-SAYED, M., RUIZ, C., RUNDESTEINER, E.A. 2004. FS-Miner: Efficient and Incremental Mining of Frequent Sequence Patterns in Web Logs, *in Proceedings of the 6th ACM International Workshop on Web Information and Data Management (WIDM 2004)*, Washington DC, November 2004

HAVELIWALA, T. 2002. Topic-Sensitive PageRank. In *Proceedings of WWW2002 Conference*, Hawaii USA, May 2002

HUANG, Z., LI, X., AND CHEN, H. 2005. Link Prediction Approach to Collaborative Filtering. In *Proceedings of ACM JCDL'05,* Colorado, 2005

KENDALL, M., AND GIBBONS, J.D. 1990. *Rank Correlation Methods*, Oxford University Press

KAMVAR, S.D., HAVELIWALA, T.H., AND GOLUB, G.H. 2003a. Adaptive Methods for the Computation of PageRank. In *Proceedings of the International Conference on the Numerical Solution of Markov Chains*, September 2003

KAMVAR, S.D., HAVELIWALA, T.H., MANNING, C.D., AND GOLUB, G.H. 2003b. Extrapolation Methods for Accelerating PageRank Computations. In *Proceedings of the 12th International World Wide Web Conference (WWW2003),* May 2003.

LEVENE, M., AND LOIZOU, G. 2003. Computing the Entropy of User Navigation in the Web. *Intl. Journal of Information Technology and Decision Making*, 2, 459-476

MANAVOGLU, D., PAVLOV, D., AND GILES, C.L. 2003. Probabilistic User Behaviour Models. In *Proceedings of ICDM 2003*

MOTWANI, R., AND RAGHAVAN, P. 1995. Randomized Algorithms. *Cambridge University Press*, United Kingdom

MSNBC. *MSNBC.COM Web Log Data*, available from UCI KDD Archive, http://kdd.ics.uci.edu/databases/msnbc/msnbc.html

NAKAGAWA, M., AND MOBASHER, B. 2003. A Hybrid Web Personalization Model Based on Site Connectivity. In *Proceedings of the 5th WEBKDD Workshop*, Washington DC, 2003

POLYZOTIS, N., AND GAROFALAKIS, M. 2002. Structure and Value Synopses for XML Data Graphs. In *Proceedings of the 28th VLDB Conference*, 2002

POLYZOTIS, N., GAROFALAKIS, M., AND IOANNIDIS, Y. 2004. Approximate XML Query Answers. In *Proceedings of SIGMOD 2004*, Paris, France, June 2004

RICHARDSON, M., AND DOMINGOS, P. 2002. The Intelligent Surfer: Probabilistic Combination of Link and Content Information in PageRank. *Neural Information Processing Systems*, 14, 1441-1448

SARUKKAI, R.R. 2000. Link Prediction and Path Analysis Using Markov Chains. *Computer Networks*, 33(1-6): 337-386

SEN, R, AND HANSEN, M. 2003. Predicting a Web user's next access based on log data. *Journal of Computational Graphics and Statistics*, 12(1), 143-155

SPILIOPOULOU, M., AND FAULSTICH, L.C. 1998. WUM: A Web Utilization Miner. In *Proceedings of the 1st Intl. Workshop on the Web and Databases (WebDB 1998)*, Spain, March 1998

ZHAO, Q. AND BHOWMICK, S.S. 2004. Mining History of Changes to Web Access Patterns. In *Proceedings of PKDD 2004*, Italy, September 2004

ZHU, J., HONG, J., AND HUGHES, J.G. 2002. Using Markov Models for Web Site Link Prediction. In *Proceedings of ACM HT'02*, Maryland, 2002

Received November 2005; revised April 2006; accepted July 2006.