Web Path Recommendations based on Page Ranking and Markov Models

Magdalini Eirinaki, Michalis Vazirgiannis, Dimitris Kapogiannis

Athens University of Economics and Business
Department of Informatics
Patision 76, Athens, 10434, GREECE
(30210) 8203513
{eirinaki, mvazirg}@aueb.gr, p3010063@dias.aueb.gr

ABSTRACT

Markov models have been widely used for modelling users' navigational behaviour in the Web graph, using the transitional probabilities between web pages, as recorded in the web logs. The recorded users' navigation is used to extract popular web paths and predict current users' next steps. Such purely usage-based probabilistic models, however, present certain shortcomings. Since the prediction of users' navigational behaviour is based solely on the usage data, structural properties of the Web graph are ignored. Thus important - in terms of pagerank authority score - paths may be underrated. In this paper we present a hybrid probabilistic predictive model extending the properties of Markov models by incorporating link analysis methods. More specifically, we propose the use of a PageRank-style algorithm for assigning prior probabilities to the web pages based on their importance in the web site's graph. 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

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

Keywords

Web Personalization, Markov Models, Link Analysis, PageRank

1. INTRODUCTION

Markov models have been widely used for modeling and predicting the users' navigational behavior. This modeling is based on the transition probabilities between web pages, as recorded in the web logs. The proposed model can be a 1st order Markov model, if the probability of transition to a next state (i.e. web page) depends only on the previous state (i.e. the last visited web pages), or of a higher order, depending on the number of states taken into account when computing the transition probability. When all transition probabilities are computed from the training data, the model may be used for personalizing a web site, by matching the current user's navigational path to the model's paths, and recommending the most probable one.

The 1st-order Markov models (Markov Chains) provide a simple way to capture sequential dependence [5, 9, 27, 31], but do not take into consideration the "long-term memory" aspects of web surfing behavior since they are based on the assumption that the next state to be visited is only a function of the current one. Higher-order Markov models [21] are more accurate for predicting navigational paths, there exists, however, a trade-off between improved coverage and exponential increase in state-space complexity as the order increases. Moreover, such complex models often require inordinate amounts of training data, and the increase in the number of states may even have worse prediction accuracy and can significantly limit their applicability for applications requiring fast predictions, such as web personalization [11]. There have also been proposed some mixture models that combine Markov models of different orders [8, 11, 22, 28]. Such models, however, require much more resources in terms of preprocessing and training. It is therefore evident that the final choice that should be made concerning the kind of model that is to be used, depends on the trade-off between the required prediction accuracy and model's complexity/size.

All kinds of Markov models, however, present certain shortcomings when used for predicting users' navigational behavior. The initial probabilities assigned to each state (i.e. web page) in Markov models are either uniformly distributed to all pages of the web site, or proportional to the times each page was visited first in a user session. Both assumptions, however, are not accurate enough. The first one assigns uniform visit probabilities to all pages (consider for example a University Web site, then the main page of a department will be equally rated to the home page of a student). On the other hand, the second approach which is the most common, favors only the top-level pages of a site since users usually start their navigation from these, assigning zero or very small probabilities to others.

Furthermore, Markov models do not handle the case when a path is not included in the training data, or is included in low frequency, therefore cannot provide reliable estimates of the corresponding transition probabilities. Hence, the problem of predicting similar paths emerges. Moreover, they are very vulnerable to the training data used to construct the model. Consider, for example an academic web site. During certain periods of time there are recorded bursts of visits to particular web pages, e.g. pages including coursework or past exam papers. If

WIDM'05, November 5, 2005, Bremen, Germany

such log data are used for training the Markov model, then the predictions won't be representative for any other period of time. Another important issue to be dealt with is that pure Markov models enable us to predict only the next step of a user. This is very useful when trying to personalize a web site by providing dynamic links to the users, but a decision should be made in the case the user has already visited the page, or a link to it already exists. It is evident that the model should predict more than one steps ahead, in other words it should be able to predict popular paths of length more than the model's order.

PageRank is the most popular link analysis algorithm, used for ranking the results returned by a search engine after a user query. The ranking is performed by evaluating the importance of a page in terms of its connectivity to/from other important pages. Many variations of this algorithm have been proposed in the past, aiming at ranking the acquired results. In this work, we introduce PageRank-style algorithms in a totally different context, that of modeling the users' navigational behavior. Motivated by the problems of pure usage-based Markov models, as described above, and the fact that in the context of navigating a web site, a page/path is important if many users have visited it before, we propose a novel approach that integrates probabilistic web usage mining and link analysis. We introduce a set of PageRank-style algorithms, which integrate the graph structure with the usage of the web site, biasing the results of the ranking to "favor" pages and paths previously preferred by many users. We then integrate this knowledge in a probabilistic predictive model based on Markov models. More specifically, our contributions lie in:

- A novel PageRank-style set of algorithms 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.
- A hybrid probabilistic prediction framework extending Markov models, used for web usage mining and personalization. We represent the user sessions using a tree-like directed graph, and further process this graph to extract navigational patterns.
- A set of experimental results which align with our claim for the need for enhancing the prediction process with information based on the link structure in combination with the usage of a site.

It is important to point out that in this work we do not address the problem of selecting the optimal order Markov model. Rather, our model is orthogonal to such a choice. In the analysis that follows, we assume that the model is given, and we propose some extensions that enhance its predictive accuracy.

The rest of the paper is organized as follows: In Section 2 we briefly overview some related research efforts. In Section 3 we present the fundamentals of Markov models and PageRank, whereas in Section 4 we present in detail the proposed model. Section 5 includes extensive experimental evaluation of the proposed model. We conclude in Section 6 with insights to our plans for future work.

2. RELATED WORK

In most of the proposed models, the priors, i.e. the initial probabilities of the nodes (web pages) of the model, are computed as proportional to the number of times a page was visited, or visited first. There exist only a few approaches where the authors claim that these techniques are not accurate enough and define different priors. Sen and Hansen [28] use Dirichlet priors, whereas Borges and Levene [5] 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 setup, they don't explicitly refer to the value they used for α.

Many researchers have proposed methods for improving the accuracy of Markov models. Sarukkai [27] presents a study showing the utility of path analysis using Markov Chains for representing "user traversals" in the web space. He computes the probability of visiting each page in the *m*-th step as a weighted combination of the *n* previous visits of the user. Zhu et. al. [31] propose an improvement of the Sarukkai method, by computing the probability of the event of a user arriving in a state of the transition probability matrix *within* the next *m* steps. Even though both approaches address the problem of *m*-path prediction, the proposed methods are very demanding in terms of computational cost. Borges and Levene [4] propose the use of a Hypertext Probabilistic Grammar (HPG) that corresponds to a 1^{st}-order Markov model and makes use of the N-gram concept in order to achieve increased accuracy in modeling the user web navigation sessions. Levene and Loisou [21] propose a model that makes use of the state cloning operation to represent 2^{nd}-order transition probabilities, increasing the Markov model's accuracy.

Mixed Markov models are based on the selection of parts from Markov models of different order, so that the resulting model has reduced state complexity as well as increased precision in predicting the user's next step. Deshpande and Karypis [11] propose the All-Kth-Markov models, presenting 3 schemas for pruning the states of the All-K*th*-Order Markov model. Cadez et.al. [8] as well as Sen and Hansen [28] also proposed the use of mixed Markov models. A different approach is that of Acharyya and Ghosh [1], who use concepts, to describe the web site. Each visited page is mapped to a concept, imposing a tree hierarchy on these topics. A semi-Markov process is then defined on this tree based on the observed transitions among underlying visited pages. They prove that this approach is computationally much less demanding compared to using higher order Markov models.

There also exist a few approaches using link or citation analysis techniques in the context of web personalization. Wang et. al. [29] propose a link analysis algorithm that models the web pages and users of a site as nodes and the hyperlinks and visits as edges between them and use a version of HITS algorithm [20] to calculate their importance in the graph. The objective of this work is to measure the expertise of users and the importance of web pages of a web site. Zhu et. al. [32] proposed CitationCluster algorithm using co-citation and coupling similarity between web pages to conceptually cluster them. The algorithm is applied on a Markov model in order to construct a conceptual hierarchy and support link prediction. Huang et. al. [18] address the data sparsity problem of collaborative filtering systems by creating a bipartite user-item graph and calculating linkage measures between unconnected pairs for selecting candidates and make recommendations.

Apart from Markov models, there also exist many approaches that perform web usage mining for web personalization. The proposed systems are based on data mining algorithms such as association rules mining [14, 23], clustering [2, 25], sequential pattern discovery [3, 7], or frequent pattern discovery from a tree-like navigational graph [16, 30]. Since we are interested in approaches that are based on Markov models, these approaches are out of the scope of this paper and won't be further discussed. For an extensive overview of such approaches the user may refer to [12, 13].

3. PRELIMINARIES

3.1 Markov Models

Markov models provide a simple way to capture sequential dependence when modeling the navigational behavior of the users of a web site *WS*. The process of creating a Markov model for this purpose is rather simple. In this Section we provide a brief overview in order to familiarize the reader with this area.

After transforming the web usage logs to users' sessions, these sessions are mapped to a weighted directed graph *G* that has two special nodes, that denote the start (*S*) and end (*E*) states. All the other nodes (states) of the graph represent the web pages of *WS*, whereas the edges represent the transitions between them. The edges between nodes are weighted and represent the number of visits to the representing path.

After all user sessions are recorded, the graph will be an illustration of all the paths followed by the web site's visitors, along with the respective frequencies. This graph can then be transformed to a 1-step transition probability matrix *TP*. We define $TP_{i,j}$ as the probability of transitioning from page *i* (denoted as p_i) to page *j* in one step (hop).

The matrix TP^n represents the transition probabilities between two pages in *n* steps, regardless of the intermediate path. In the case of higher-order models, we should also compute the transition probabilities of a page given the past *m* pages visited (resulting in an $M^m \times M$ transition probability matrix).

After finding these transition probabilities, the chain rule is applied in order to compute all path probabilities $P(p_1 \to p_2 \to \ldots \to p_k)$ depending on the order of the Markov model we want to create. For an *m-th* order MM, (i.e. the predictions are based on the past *m* visits of the user), it equals to:

$$P(p_1 \to p_2 \to \ldots \to p_k) = P(p_1) * \prod_{i=2}^{k} P(p_i / p_{i-m} \cdots p_{i-1}) \quad (1)$$

If, for example, we want to predict the probability of the path *P(a→b→c)* using a Markov Chain, Equation 1 reduces to Equation 2:

$$P(a \to b \to c) = P(a)P(b/a)P(c/b) = P(a)\frac{P(a \to b)}{P(a)}\frac{P(b \to c)}{P(b)} \quad (2)$$

Then, whenever we want to predict the next page in a user's visit, in order to provide recommendations, we choose the path with the higher probability among all existing paths having the pages visited so far as prefix.

3.2 PageRank

The PageRank algorithm [6] is the most popular link analysis algorithm, used broadly for assigning numerical weightings to web documents and 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's currently at, or "jumps" to a random page after a few clicks. The PageRank of a page is defined as the probability of the random surfer being at some particular time step $k > K$ at this page. This probability is correlated with the *importance* of this page, as it is defined based on the number and the importance of the pages pointing 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 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 *j* to page *i.* The adjacency function m_{ij} is 0 if there is no direct link from p_j to p_i, and normalized such that, for each *j:*

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

(3).

PageRank is in essence the stationary probability distribution over pages induced by a random walk on the web. The convergence of PageRank is guaranteed only if *M* is irreducible and aperiodic [26]. The periodicity of *M* is guaranteed in practice in the web context, whereas the irreducibility is satisfied by adding a dumping factor *(1-ε)* to the rank propagation (ε 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. PageRank can then be expressed as the unique solution to Equation 4:

$$PR = \varepsilon \times M \times PR + (1 - \varepsilon)p$$

(4)

where *p* is a non-negative *N*-vector whose elements sum to Usually $m_{ij} = \frac{1}{\sum_{p_k \in Out(p_j)} |p_k|}$ and, $p = \left[\frac{1}{N}\right]_{N \times 1}$ i.e. the probability of randomly jumping to another page is uniform. By choosing, however, *p* to follow a non-uniform distribution, we essentially *bias* the resultant PageRank vector computation to favor certain pages.

4. THE PROPOSED MODEL

In our model, we represent the visitors' navigational behavior (i.e. the user sessions residing on the web logs) as a weighted tree-like structure *NtG*. This structure has as root the special node *R* and all the other nodes are instances of the *M* web pages of *WS*. The weighted paths from the root towards the leaves represent all user sessions' paths included in the web logs. All tree branches terminate in a special leaf-node *E* denoting the end of a path.

Definition 1: We define our model as a set of tuples <*S, TP, IP*> where *S* is the state space including the *M* distinct nodes *W* in *NtG*, *TP* is the *M*x*M* one-step transition probability matrix, and *IP* is the initial probability distribution regarding the states in *S.* We

also define w_i as the number of times page i was visited, and $w_{i,j}$ as the number of times pages i and j were visited consecutively.

Note that in our representation, there may exist several replications of the states in different parts of the tree-like structure. As we describe in what follows, this structure can be approximated by any tree-graph synopsis, for example a finite-state Markov Chain or an *m-th* order Markov model, based on the desired trade-off between the accuracy and complexity of the model. We should stress, however, that the Markov model selection is orthogonal to the proposed approach.

4.1 Construction of *NtG*

We assume that the log files have been preprocessed and separated into distinct user sessions.

Definition 2: We define as user session *US* a sequence of states $s \in S$, of length L.

The algorithm for creating the tree-like structure is depicted in Figure 1. Briefly, for every user session 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 common page visited in the path (we assume that any consecutive pages' repetitions have been removed from the user sessions; on the other hand, we keep any pages been visited twice, but not consecutively). We also denote the end of a session using a special exit node.

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

Figure 1. NtG creation algorithm

In order to illustrate this process, we use a sample scenario. Assume that the user sessions of a web site are those included in Table 1. Figure 2 depicts the Navigational Graph *NtG* created after applying the aforementioned algorithm as well as its respective Markov Chain synopsis.

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

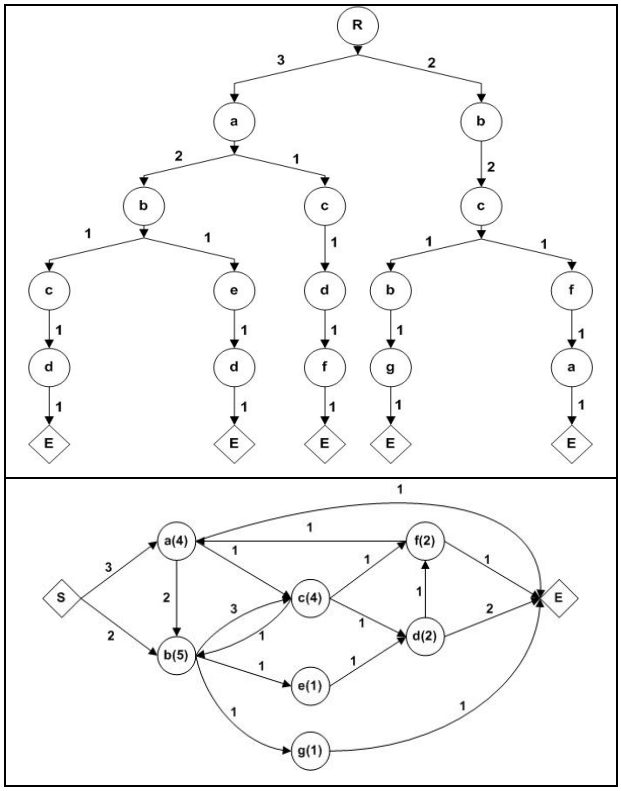

Figure 2. Navigational Graph NtG and the respective Markov Chain synopsis

4.2 Probabilities' Estimation

As already mentioned, the tree structure created using the algorithm described in the previous section, will have a root node and weighted edges connecting the rest tree nodes. Each web page in WS will appear many times as a node in the tree. The weighted paths are used for estimating the transition probabilities of our model.

4.2.1 Transitional Probabilities

As already mentioned, we define w_{ij} as the number of times pages p_i and p_j were consecutively visited. w_{ij} is therefore computed as the sum of the weights of all directed edges between nodes $p_i \rightarrow p_j$. We also compute w_i, the number of times a page was visited as the sum of all the weights of edges pointing to p_i:

$$w_i = \sum_{k \in In(p_i)} w_{ki} \qquad (5)$$

Using these weights, we can then estimate the prior probabilities of the web pages, as well as the transition probabilities between two pages.

Definition 3: The probability of a transition between two pages is estimated by the ratio of the number of times the sequence was visited to the number of times the anchor page was visited. We define:

$$TP_{i,j} = P(p_i \rightarrow p_j) = \frac{w_{ij}}{\sum_{k \in Out(p_i)} w_{ik}}, i \in \{WS \cup R\}, j,k \in \{WS \cup E\} \quad (6)$$

The one-step transition probabilities defined above can be used when we want to create a 1st-order Markov model. In case we want to create models of higher order, we should accordingly compute the respective path probabilities by dividing the number of times a path of certain length *l* (according to the model's order) was visited with the number of times all paths of length *l-1* starting with the same prefix were visited.

4.2.2 Prior Probabilities

There exist two broadly used approaches 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 as the ratio of the number of times this page has been visited as a first page to the total number of times a page was visited as a first page, i.e. 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 assumption 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 when a page was never visited first, its prior probability equals to zero. There also exists another approach which 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.

To handle such shortcomings, and motivated by the fact that the initial probability of a page should reflect the *importance* of this page in the web navigation, we instead propose the use of a hybrid algorithm based on the topology (link structure) of the web site, as well as the navigational patterns of its visitors for assigning prior probabilities. We introduce three ranking algorithms based on the well-known link analysis algorithm PageRank [6]. The first (*PR)* is the algorithm itself, and computes the page probabilities based on the link structure of the web site. The second is a usage-based personalized PageRank algorithm *(UPR),* which biases the algorithm to "prefer" pages previously visited by many users. The third is a variation of *UPR (SUPR)*, which assigns uniform probabilities to the random jump instead of biasing it as well.

Definition 4: We define the prior probability IP_i of a page p_i as:

$$IP_i = P(p_i) = \Pr^n(p_i) = (1-\varepsilon) * o(p_i) + \varepsilon \sum_{p_k \in In(p_i)} \left(\Pr^{n-1}(p_k) * o(p_k, p_i) \right) \quad (7)$$

with *(1-ε)* being the dumping factor (usually set to 0.15) and for

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

$$o(p_i) = \frac{1}{M} \text{ and } o(p_k, p_i) = 1 / |Out(p_k)| \quad (8)$$

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

$$o(p_i) = \frac{1}{M} \text{ and } o(p_k, p_i) = \frac{w_{ki}}{\sum_{p_j \in Out(p_k)} w_{kj}} \quad (9)$$

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

$$o(p_i) = \frac{w_i}{\sum_{p_j \in WS} w_j} \text{ and } o(p_k, p_i) = \frac{w_{ki}}{\sum_{p_j \in Out(p_k)} w_{kj}} \quad (10)$$

A more detailed description of *UPR*, as well as theoretical proof of its convergence can be found at [15].

4.3 Popular Path Prediction

Markov Models fail to predict directly more than one step ahead. In the case of web personalization this causes problems when, for example, we want to provide recommendations to a user, and the most probable next page already exists as a link on the page. In that case, there exist four different alternatives:

1. Predict one (or more) page ahead.
2. Recommend the next most probable page.
3. Predict popular paths (i.e. paths having the current visit as prefix and more than one pages as suffix).
4. Create a recommendation set by semantically expanding the page.

In the case of Markov Chains, the transition probability of visiting a page in 2 steps is given by the TP^2 matrix (similarly the probability of visiting a page in 3 steps is given by the TP^3 matrix, and so on). However, this does not hold for higher order models, whereas, even in the case of 1st-order Markov models, the transition probability matrices may become very large. It is therefore evident that we should choose one of the three other alternatives, or a combination of them. The second alternative is straightforward, since we choose to recommend the second most probable path having as prefix the path already visited by the user. If this page already exists as a link, move to the next and so on. This, however, would require multiple computations until we reach a page the user has not yet visited. On the other hand, we may pre-compute all popular paths visited by previous users offline and predict "next" paths instead of "next" pages. Finally, if there does not exist such a path in the model, we may expand the most probable next page, using the techniques introduced in [14]. Since this latter approach is out of the scope of this paper, we refer the reader to the related paper. In our experiments we deal with the third approach which is a generalization of the second.

5. EXPERIMENTAL EVALUATION

In this section we present a set of experiments that we performed for evaluating the impact of our proposed technique on the prediction process. Overall our experiments have verified the effectiveness of our proposed techniques in web path prediction for recommendations.

5.1 Experimental Setup

For our experiments we used two publicly available data sets. The first one, called *msnbc* [24], includes the page visits of users who visited msnbc.com on 28/9/99. The visits are recorded at the level of URL category (for example sports, news, etc.) and it includes visits to 17 categories (i.e. 17 distinct pageviews). We selected 96.000 distinct sessions each one containing at most 50 page visits per session. The second data set, called *cti* [10], includes the sessionized data for the main DePaul CTI web server, based on a random sample of users visiting the site for a two week period during April 2002. The data set includes 683 distinct pageviews and 13.745 distinct user sessions. 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 which essentially translates to a hight number of sessions with very long paths. On the other hand, this data set has 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 homogeneously distributed among the 17 different categories. The *cti* data set refers to an academic web site. Visits to such sites are usually categorized in two main groups: visits from students seeking for information concerning courses' or administrative material, and visits from researchers seeking for papers, research projects, etc. We expect that the recorded visits will imply this categorization.

We created 5 different setups of the prediction model, varying in terms of prior probabilities' estimation. The first two setups (further referred to as *Start* and *Total)* are the ones used in previous approaches for computing prior probabilities. More specifically, *Start* assigns probabilities proportional to the visits of a page as start page of the navigation, 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, as it has been shown that it typically performs worse than *Start*. The other three setups (*PR, SUPR, UPR)* assign probabilities using the respective proposed methods, as described in Section 4.2.2. The synopsis we used for approximating the Navigational Graph *NtG* is the Markov Chain. For the PageRank-style algorithms, the dumping factor *(1-ε)* was set to 0,15 and the number of iterations was set to 100.

We split the data sets in two non-overlapping time windows to form a training and a test data set (*msnbc{training:65.000, test:31.000}, cti{training:9.745, test:4.000}*). Applying the five setups on the training data, we produced a list including the top-*n* most probable paths for $n \in \{5, 10, 20\}$. 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. We used two metrics for comparing two top-*n* rankings r_1 and r_2. The first one, denoted as $OSim(r_1,r_2)$ indicates the degree of overlap between the top-*n* pages of two sets *A* and *B* (each one of size *n*) to be $OSim(r_1, r_2) = \frac{|A \cap B|}{n}$ (11) [17]. The second, $KSim(r_1,r_2)$ is based on Kendall's distance measure [19] and indicates the degree to which the relative orderings of two top-*n* lists are in agreement:

$$KSim(r_1, r_2) = \frac{|(u,v) : r_1', r_2' \text{ have same order of } (u,v), u \neq v|}{|A \cap B|(|A \cap B| - 1)} \quad (12)$$

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) [17].

5.2 Popular Path Ranking Evaluation

The diagrams of Figure 3 depict the *OSim* and *KSim* similarities for the top 5, 10, and 20 rankings of *msnbc* data set. We observe that the *OSim* is around 60% for the two pure usage-based methods, *Start* and *Total,* whereas it is reaches 90% for the three proposed methods. The *KSim*, on the other hand, is more than 90% for all rankings in the case of the proposed methods, whereas it is high only for the two first rankings and the second ranking for *Start* and *Total* setups respectively.

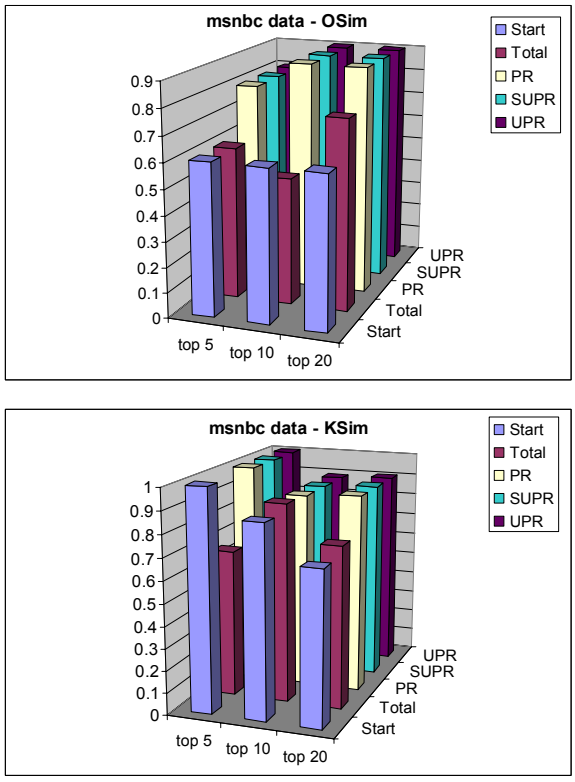

Figure 3. OSim and KSim similarities of top-n rankings for the *msnbc* data set

The diagrams of Figure 4 depict the *OSim* and *KSim* similarities for the top 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 returned more than 80% *OSim* and an average of 90% *KSim* in all setups.

What should be commended at this point is 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 important pages may be overrated or underestimated in certain circumstances. In the case of *msnbc* where the number of distinct pages was very small and therefore the paths were evenly distributed, the pure usage-based models seem to behave moderately (but, again, worse than the proposed hybrid models). On the other hand, in the case of the *cti* data set where there existed thousands of distinct pages (and therefore

distinct paths), the prediction accuracy of usage-based models was disappointing! In the Appendix we include the top-10 rankings of *Start* and *Total* setups, as well as the most frequent ones. The reader may observe that the top-10 rankings of the first two approaches represent 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 by a bookmarked page) 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 denoted by the link structure of the web site, such temporal behaviors' influence is reduced.

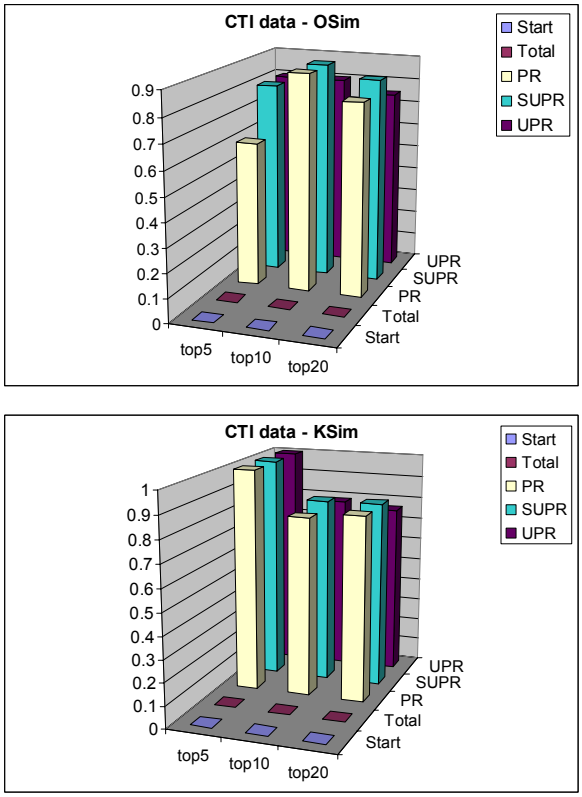

Figure 4. OSim and KSim similarities of top-n rankings for the *cti* data set

Finally, comparing the three proposed methods, we observe that in the case of the *msnbc* data set all methods have the same OSim, which increases analogously with the number of recommendations. In the case of the *cti* data set, on the other hand, we observe that SUPR outperforms the other two methods. There does not exist, however, a pattern underlying the relation between the number of recommendations and OSim/KSim. Therefore, we cannot conclude on the superiority of one of the proposed methods, since it depends on the data set used.

6. CONCLUSIONS

Markov models have been widely used for modeling and predicting the users' navigational behavior. In this paper, we address several shortcomings of these predictive models, which occur especially when such models are used for personalizing a web site by providing recommendations to its users. We propose a novel hybrid probabilistic predictive model, which addresses these problems, by combining link analysis and web usage mining techniques. We present three variations of a PageRank-style algorithm for assigning initial probabilities to the nodes of the model. Moreover, we introduce a framework, based on Markov models, that addresses issues such as popular path prediction.

The experiments we have performed are more than promising. We are currently performing experiments using higher-order Markov model synopses. Finally, we are currently working on using the *UPR* algorithm for ranking the pages of personalized web graphs.

7. REFERENCES

[1] S. Acharyya, J. Ghosh, Context-Sensitive Modeling of Web-Surfing Behaviour Using Concept Trees, *in Proc. of the 5th WEBKDD Workshop,* Washington DC, August 2003

[2] R. Baraglia, F. Silvestri, An Online Recommender System for Large Web Sites, *in Proc. of ACM/IEEE Web Intelligence Conference (WI'04),* China, September 2004

[3] B. Berendt, M. Spiliopoulou, *Analysing navigation behaviour in web sites integrating multiple information systems*, The VLDB Journal (2000), 9(1):56-75

[4] J. Borges, M. Levene, Data Mining of User Navigation Patterns*, in Revised Papers from the Intl. Workshop on Web Usage Analysis and User Profiling,* LNCS Vol. 1836, pp.92-111, 2000

[5] J. Borges, M. Levene, A Dynamic Clustering-Based Markov Model for Web Usage Mining*,* Submitted for review 2005

[6] S. Brin, L. Page, The anatomy of a large-scale hypertextual Web search engine, *in Proc. of the 7th International World Wide Web Conference (WWW7),* Brisbane, 1998

[7] A.G. Buchner, M. Baumgarten, S.S. Anand, M.D. Mulvenna, J.G. Hughes, Navigation pattern discovery from Internet data*, in Proc. of WEBKDD'99 Workshop,* August 1999, San Diego, CA

[8] I. Cadez, S. Gaffney, P. Smyth, A general probabilistic framework for clustering individuals and objects*, in Proc. of the 6th ACM SIGKDD Conference,* Boston, 2000

[9] I.Cadez, D.Heckerman, C.Meek, P. Smyth, S. White, Visualization of Navigation Patterns on a Web Site Using Model Based Clustering, *in Proc. of the 6th ACM SIGKDD Conference,* Boston MA, 2000, pp. 280-284

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

[11] M. Deshpande, G. Karypis, Selective Markov Models for Predicting Web-Page Accesses*, in Proc. of the 1st SIAM International Conference on Data Mining,* 2001

[12] M. Eirinaki, *Web Mining: A Roadmap,* Technical Report, DB-NET 2004, available at http://www.db-net.aueb.gr

[13] M. Eirinaki, M. Vazirgiannis, *Web Mining for Web Personalization*, in ACM Transactions on Internet Technology (TOIT), 3(1), February 2003, 1-29

[14] M. Eirinaki, M. Vazirgiannis, I. Varlamis, SEWeP: Using Site Semantics and a Taxonomy to Enhance the Web Personalization Process, *in Proc. of the 9th ACM SIGKDD Conference (KDD2003),* August 2003, Washington DC

[15] M. Eirinaki, M. Vazirgiannis, Usage-based PageRank for Web Personalization, *in Proc. of 5th IEEE International Conference on Data Mining (ICDM'05)*, Lousiana, November 2005

[16] M. El-Sayed, C. Ruiz, E.A. Rundensteiner, FS-Miner: Efficient and Incremental Mining of Frequent Sequence Patterns in Web Logs, *in Proc. of 6th ACM WIDM Workshop (WIDM04),* November 2004, Washington DC

[17] Taher Haveliwala, Topic-Sensitive PageRank, *in Proc. of WWW2002,* Hawaii USA, May 2002

[18] Z. Huang, X. Li, H. Chen, Link Prediction Approach to Collaborative Filtering, *in Proc. of ACM JCDL'05,* Colorado, 2005

[19] M. Kendall, J.D.Gibbons, *Rank Correlation Methods,* Oxford University Press, 1990

[20] J. Kleinberg, Authoritative Sources in a Hyperlinked Environment, *in Proc. of 13th AAAI Conference (AAAI-96),* Portland, 1996

[21] M. Levene, G. Loizou, *Computing the Entropy of User Navigation in the Web*, in Intl. Journal of Information Technology and Decision Making, 2:459-476, 2003

[22] E. Manavoglou, D. Pavlov, C.L. Giles, Probabilistic User Behaviour Models*, in Proc. of the 3rd IEEE International Conference on Data Mining (ICDM 2003),* 2003

[23] F. Masseglia, P. Poncelet, M. Teisseire, *Using Data Mining Techniques on Web Access Logs to Dynamically Improve Hypertext Structure*, in ACM SigWeb Letters, Vol. 8, N. 3, pp. 13-19, October 1999

[24] msnbc.com Web Log Data, available from *UCI KDD Archive,* http://kdd.ics.uci.edu/databases/msnbc/msnbc.html

[25] Mike Perkowitz, Oren Etzioni, *Towards Adaptive Web Sites: Conceptual Framework and Case Study,* in Artificial Intelligence 118[1-2] (2000), pp. 245-275

[26] R. Motwani and P. Raghavan. R*andomized Algorithms*, Cambridge University Press, United Kingdom, 1995.

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

[28] R. Sen, M. Hansen, *Predicting a Web user's next access based on log data*, in Journal of Computational Graphics and Statistics, 12(1):143-155, 2003

[29] J. Wang, Z. Chen, L. Tao, W. Ma, L. Wenyin, Ranking User's Relevance to a Topic through Link Analysis on Web Logs, *in Proc. of WIDM '02,* November 2002

[30] Q. Zhao, S.S. Bhowmick, Mining History of Changes to Web Access Patterns, *in Proc. of the 8th PKDD Conference (PKDD 2004),* Italy, September 2004

[31] J. Zhu, J. Hong, J.G.Hughes, Using Markov Chains for Link Prediction in Adaptive Web sites, *in Proc. of the Soft-Ware 2002: Computing in an Imperfect World, 2002*

[32] J. Zhu, J. Hong, J.G.Hughes, Using Markov Models for Web Site Link Prediction, *in Proc. of ACM HT'02,* Maryland, June 2002

APPENDIX

In this section we present the top-10 rankings of the *cti* data set for *Start* and *Total* setups, as well as the *Frequent paths* derived from the web usage data. We omit the top-10 rankings of *PR, SUPR* and *UPR* since they are very similar to the *Frequent paths* ranking.

Table 2. Top-10 Frequent Paths

/news/default.asp → /courses/
/authenticate/login.asp?section=mycti&title=mycti&urlahead=studentprofile/studentprofile → /cti/studentprofile/studentprofile.asp?section=mycti
/news/default.asp → /people/
/courses/ → finish
/news/default.asp → /courses/ → finish
/courses/ → /courses/syllabilist.asp
/cti/advising/login.asp -> /cti/advising/display.asp?page=intranetnews
/news/default.asp → /courses/ → /courses/syllabilist.asp
/news/default.asp → /programs/
/people/ → /people/search.asp

Table 3. Top-10 ranking for Start setup

/news/default.asp → /courses/syllabus.asp?course=250-97-802&q=2&y=2002&id=251
/news/default.asp → /courses/syllabus.asp?course=250-97-802&q=2&y=2002&id=251→ /courses/syllabilist.asp
/news/default.asp → /courses/syllabus.asp?course=312-99-601&q=3&y=2002&id=263
/news/default.asp → /courses/syllabus.asp?course=312-99-601&q=3&y=2002&id=263 → finish
/news/default.asp → /courses/syllabus.asp?course=318-21-601&q=3&y=2002&id=495
/news/default.asp → /courses/syllabus.asp?course=318-21-601&q=3&y=2002&id=495 → /news/
/news/default.asp → /courses/syllabus.asp?course=345-21-901&q=3&y=2002&id=351
/news/default.asp → /courses/syllabus.asp?course=345-21-901&q=3&y=2002&id=351 → finish
/news/default.asp → /courses/syllabus.asp?course=364-98-601&q=3&y=2002&id=921
/news/default.asp → /courses/syllabus.asp?course=364-98-601&q=3&y=2002&id=921 → /courses/syllabus.asp?course=463-98-301&q=3&y=2002&id=323

Table 4. Top-10 ranking for Total setup

/courses/ → /courses/syllabus.asp?course=224-21-601&q=3&y=2002&id=561
/courses/ → /courses/syllabus.asp?course=224-21-601&q=3&y=2002&id=561 → /courses/syllabus.asp?course=224-21-901&q=3&y=2002&id=214
/courses/ → /courses/syllabus.asp?course=224-21-601&q=3&y=2002&id=561 → /courses/syllabus.asp?course=224-21-901&q=3&y=2002&id=214 → /courses/syllabus.asp?course=224-21-902&q=3&y=2002&id=230
/courses/ → /courses/syllabus.asp?course=224-21-601&q=3&y=2002&id=561 → /courses/syllabus.asp?course=224-21-901&q=3&y=2002&id=214 → /courses/syllabus.asp?course=224-21-902&q=3&y=2002&id=230 → /courses/syllabus.asp?course=224-21-903&q=3&y=2002&id=250 → finish
/courses/ → /courses/syllabus.asp?course=224-21-903&q=3&y=2002&id=250
/courses/ → /courses/syllabus.asp?course=224-21-903&q=3&y=2002&id=250 → finish
/courses/ → /courses/syllabus.asp?course=309-21-903&q=3&y=2002&id=198
/courses/ →/courses/syllabus.asp?course=309-21-903&q=3&y=2002&id=198 → finish
/courses/ → /courses/syllabus.asp?course=311-98-601&q=3&y=2002&id=921
/courses/ → /courses/syllabus.asp?course=372-98-901&q=3&y=2002&id=326