Evaluating Variable-Length Markov Chain Models for Analysis of User Web Navigation Sessions

José Borges and Mark Levene

Abstract—Markov models have been widely used to represent and analyze user Web navigation data. In previous work, we have proposed a method to dynamically extend the order of a Markov chain model and a complimentary method for assessing the predictive power of such a variable-length Markov chain. Herein, we review these two methods and propose a novel method for measuring the ability of a variable-length Markov model to summarize user Web navigation sessions up to a given length. Although the summarization ability of a model is important to enable the identification of user navigation patterns, the ability to make predictions is important in order to foresee the next link choice of a user after following a given trail so as, for example, to personalize a Web site. We present an extensive experimental evaluation providing strong evidence that prediction accuracy increases linearly with summarization ability.

Index Terms—Web mining, navigation, Markov processes, modeling and prediction.

1 Introduction

Web usage mining has been defined as the research field focused on developing techniques to model and study users' Web navigation data [1]. Data characterizing navigation sessions can be collected from server log files or from a Web browser plug-in aimed at recording the user navigation options. From the collected data, it is possible to reconstruct users' Web navigation sessions [2], where a session, also referred to as a *trail*, consists of a sequence of Web pages viewed by a user within a given time window.

Several models have been proposed for modeling user Web data. Schechter et al. [3] utilized a tree-based data structure that represents the collection of paths inferred from the log data to predict the next page access. Dongshan and Junyi [4] proposed a hybrid-order tree-like Markov model, which provides good scalability and high coverage of the state space, also to predict the next page access. Chen and Zhang [5] utilized a Prediction by Partial Match forest that restricts the roots to popular nodes; assuming that most user sessions start in popular pages, the branches having a nonpopular page as their root are pruned. Deshpande and Karypis [6] proposed a technique that builds kth-order Markov models and combines them to include the highest order model covering each state; a technique to reduce the model complexity is also proposed. Moreover, Eirinaki et al. [7] propose a method that incorporates link analysis, such as the pagerank measure, into a Markov model in order to provide Web path recommendations.

In [8], we have proposed a first-order Markov model for a collection of user navigation sessions, and, more recently, we have extended the method to represent higher order conditional probabilities by making use of a cloning operation [9], [10]. In addition, we have proposed a method to evaluate the *predictive power* of a model that takes into account a variable-length history when estimating the probability of the next link choice of a user, given his or her navigation trail [11]. (We review these models in Sections 2.1 and 2.2.) Herein, we propose a new method to measure the *summarization ability* of a model, by which we mean the ability of a variable-length Markov model to summarize user trails up to a given length. Briefly, from a collection of trails a variable-length Markov model, up to order n, is built, and an algorithm is used to induce the trails having a high probability of being traversed. The set of trails is then ranked, from the most to the least probable, and the resulting ranking is compared with the ranking induced from the n-gram frequency count of the input trails. Thus, if a model accurately represents the collection of input navigation trails of length n, the two rankings should be identical. Note that although it is clear that an n-order Markov model accurately represents trails shorter or equal to n, it is important to understand how well such a model represents trails longer than n. To make the comparison between the rankings, we use the Spearman footrule [12] and percentage overlap metrics. We present the results of an extensive experimental evaluation conducted on three real-world data sets, which provide strong evidence that the measure of predictive power increases linearly with the measure for summarization ability.

Understanding the behavior of Web site visitors navigating through a site is an important step in the process of improving the quality of service of that site. A model of past user navigation behavior can be used to identify frequent usage patterns that can provide insights on how to improve the Web site design and structure in order to satisfy the

- J. Borges is with the School of Engineering, University of Porto, R. Dr. Roberto Frias, 4200-465 Porto, Portugal. E-mail: jlborges@fe.up.pt.
- M. Levene is with the School of Science and Information Systems, Birkbeck, University of London, Malet Street, London WC1E 7HX, UK. E-mail: mark@dcs.bbk.ac.uk.

Manuscript received 27 June 2006; revised 28 Nov. 2006; accepted 4 Dec. 2006; published online 24 Jan. 2007.
For information on obtaining reprints of this article, please send e-mail to: tkde@computer.org, and reference IEEECS Log Number TKDE-0320-0606.
Digital Object Identifier no. 10.1109/TKDE.2007.1012.

1041-4347/07/$25.00 © 2007 IEEE Published by the IEEE Computer Society

visitors needs. In addition, being able to predict the near-future navigation intentions of an individual user will enable the provision of pages adapted to the recent behavior of this user and the use of past behavior of other visitors to guide the user in the search to satisfy his information needs. Moreover, predicting the user's next choice enables the construction of dynamic pages in advance or the provision of a speculative prefetching service that sends, in addition to the requested document, a number of other documents that are expected to be requested in the near future.

In our opinion, Markov models are well suited for modeling user Web navigation data because they are compact, simple to understand and motivate, expressive, and based on a well-established theory. Commercial tools for log data analysis usually discard the information concerning the order in which page views occurred in a session, and the same can be said about the techniques using association rules methods. On the other hand, variable-length Markov chain (VLMC) models provide the probability of the next link chosen when viewing a Web page while taking into account the trail followed to reach that page.

Our measure of the summarization ability of the model answers a question we have often been asked about the adequacy of Markov models in representing user Web trails. The measure tells us how accurate the Markov model representing the users' trails is, and thus, when having a precise Markov representation, we are justified in using methods that extract the most probable trails from the model (see [13]). Moreover, Markov models can also be used to predict the user's next navigation step within the site, as shown in Section 2.4. Thus, the summarization ability of the model is important in enabling the identification of user navigation patterns, and the prediction ability is important for foreseeing the next link choice of a user after following a given trail.

We stress that there has been previous research on evaluating the ability of a Markov model to predict the next link choice of a user [6], [11]; however, to the best of our knowledge, there is a lack of publications on evaluating the ability of a Markov model to represent user sessions up to a given history length. Moreover, as far as we know, the relationship between summarization ability and prediction power has not been studied before in the context of Web mining.

The rest of the paper is organized as follows: In Section 2, we introduce the VLMC methods we make use of. More specifically, in Section 2.1, we introduce our method for building a first-order model from a collection of sessions, and in Section 2.2, its extension to higher order conditional probabilities; in Section 2.3, we present our new method for measuring the model's summarization ability; and in Section 2.4, we introduce our method for measuring the model's predictive power. In Section 3, we present an experimental evaluation of the methods, and, finally, in Section 4, we give our concluding remarks.

2 VLMC Methods

A user navigation session within a Web site can be represented by the sequence of pages requested by the user. First-order Markov models have been widely used to model a collection of user sessions. In such context, each Web page in the site corresponds to a state in the model, and each pair of pages viewed in sequence corresponds to a state transition in the model. A transition probability is estimated by the ratio of the number of times the transition was traversed to the number of times the first state in the pair was visited. Usually, artificial states are appended to every navigation session to denote the start and finish of the session.

A first-order Markov model is a compact way of representing a collection of sessions, but in most cases, its accuracy is low [14], which is why extensions to higher order models are necessary. In a (nonvariable) higher order Markov model, a state corresponds to a fixed sequence of pages, and a transition between states represents a higher order conditional probability [8]. For example, in a second-order model, each state corresponds to a sequence of two-page views. The serious drawback of fixed higher order Markov models is their exponentially large state space compared to lower order models.

A VLMC is a model extension that allows variable length history to be captured [15]. In [10], we proposed a method that transforms a first-order model into a VLMC so that each transition probability between two states takes into account the path a user followed to reach the first state prior to choosing the outlink corresponding to the transition to the second state. The method makes use of state cloning (where states are duplicated to distinguish between different paths leading to the same state) together with the K-Means clustering technique that separates paths revealing differences in their conditional probabilities.

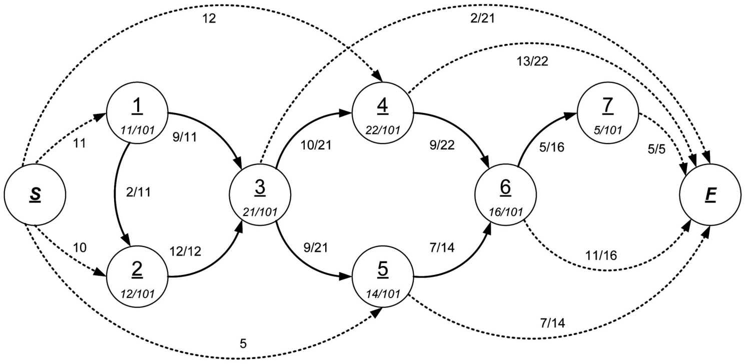
Herein, we make use of a VLMC model to summarize the navigation behavior of the users visiting a Web site. We note that the analysis could be refined by making use of user profile data to group users with similar interests and using an individual VLMC model for each group of users. We also note that it is possible to enhance the model in order to take into account Web page content. For example, each state definition can include a vector of keywords representing the contents of the corresponding Web page. As a result, it would be possible to identify high probability trails that are composed of pages that are relevant to a given topic. We will now introduce our method by means of an illustrative example; we refer readers looking for a formal description of the method to [10].

2.1 First-Order Model Construction

Fig. 1a shows an example of a collection of navigation sessions. We let a session start and finish at an artificial state; *freq.* denotes the number of times the corresponding sequence of pages was traversed. Fig. 1b presents the first-order model for these sessions. There is a state corresponding to each Web page and a link connecting every two pages viewed in sequence. For each state that corresponds to a Web page, we give the page identifier and the number of times the page was viewed divided by

session	*freq.*
S,1,2,3,4,F	2
S,1,3,5,F	7
S,1,3,4,F	2
S,2,3,5,6,F	2
S,2,3,4,F	3
S,4,6,7,F	1
S,4,6,F	5
S,4,F	6
S,5,6,7,F	4
S,5,6,F	1
S,2,3,4,6,F	3
S,2,3,F	2

(a)



(b)

Fig. 1. An example of (a) a collection of sessions and (b) the corresponding first-order model.

the total number of page views. This ratio is a probability estimate for a user choosing the corresponding page from the set of all pages in the site. For example, page 4 has 22 page views from a total of 101 page views. For each link, we indicate the proportion of times it was followed after viewing the anchor page. For example, page 5 was viewed 14 times, five of which were at the beginning of a navigation session (the weight of the link from the artificial state S indicates the number of sessions that started in that page). After viewing page 5, the user moved to page 6 in 7 of the 14 times and terminated the session seven times. The probability estimate of a trail is given by the product of the probability of the first state in the trail (that is, the initial probability) and the probabilities of the traversed links (that is, the transition probabilities). For example, the probability estimate for trail $(3,4)$ is $21/101 \cdot 10/21 = 0.099$, and for trail $(1,3,5)$, it is $11/101 \cdot 9/12 \cdot 9/21 = 0.035$.

2.2 Higher Order Model Construction

The first-order model does not accurately represent all second-order conditional probabilities. For example, according to the input data, the sequence $(1,3)$ was followed nine times, that is, $\#(1,3) = 9$, and sequence $(1,3,4)$ was followed twice, that is, $\#(1,3,4) = 2$. Therefore, the probability estimate for viewing page 4 after viewing 1 and 3 in sequence is $p(4|1,3) = \#(1,3,4)/\#(1,3) = 2/9$. The error of a first-order model in representing second-order probabilities can be measured by the absolute difference between the corresponding first- and second-order probabilities. For example, for state 3, we have that $|p(4|1,3) - p(4|3)| = |2/9 - 10/21| = 0.254$ and

$$|p(4|2,3) - p(4|3)| = |8/12 - 10/21| = 0.190;$$

thus, state 3 is not accurately representing second-order conditional probabilities. The accuracy of transition probabilities from a state can be increased by separating the in-paths to it that correspond to different conditional probabilities. We increase the accuracy in the example by cloning state 3 (that is, creating a duplicate state $3'$) and redirecting the link $(2,3)$ to state $3'$. The weights of the outlinks from states 3 and $3'$ are updated according to the number of times the sequence of three states was followed. For example, since $\#(1,3,4) = 2$ and $\#(1,3,5) = 7$ in the second-order model, the weight of the link $(3,4)$ is 2 and the weight of $(3,5)$ is 7. (Note that, according to the input data, no session terminates at page 3 when the user has navigated to it from page 1.) The same method is applied to update the outlinks from the clone state $3'$. Fig. 2 shows the resulting second-order model after cloning four states in order to accurately represent all second-order conditional probabilities.

In the extended model given in Fig. 2, all the outlinks represent accurate second-order probability estimates. The probability estimate of the trail $(1,3,5)$ is now $11/101 \cdot 9/11 \cdot 7/9 = 0.069$. The probability estimate for trail (3,4) is $(9/101 \cdot 2/9) + (12/101 \cdot 8/12) = 0.099$, which is equal to the first-order estimate. Therefore, the second-order model accurately models the conditional second-order probability estimates while keeping the correct first-order probability estimates.

In order to provide control over the number of additional states created by the method, we make use of a parameter γ that sets the highest admissible difference between a first-order and the corresponding second-order probability estimate. In a first-order model, a state is cloned if there is a second-order probability whose difference from the corresponding first-order probability is greater than γ. Alternatively, we interpret γ as a threshold for the average difference between the first-order and the corresponding second-order probabilities for a given state. In the latter, the state is cloned if the average difference between the first- and second-order conditional probabilities surpasses γ. Moreover, if we set $\gamma > 0$ and the state has three or more inlinks, we make use of the K-Means clustering algorithm to identify inlinks inducing identical conditional probabilities. When γ is measuring the maximum probability of divergence, we will denote it by γ_m, and when it is measuring the average probability divergence, we will denote it by γ_a.

We now give an example to illustrate the role of the γ parameter. Fig. 3a shows a collection of sessions and Fig. 3b the corresponding first-order model. In Fig. 4, the second-order model with one additional clone is shown (states S and F are omitted for the sake of simplicity). In Table 1, the accuracy of both models is measured. The conditional probabilities estimated from the input data are

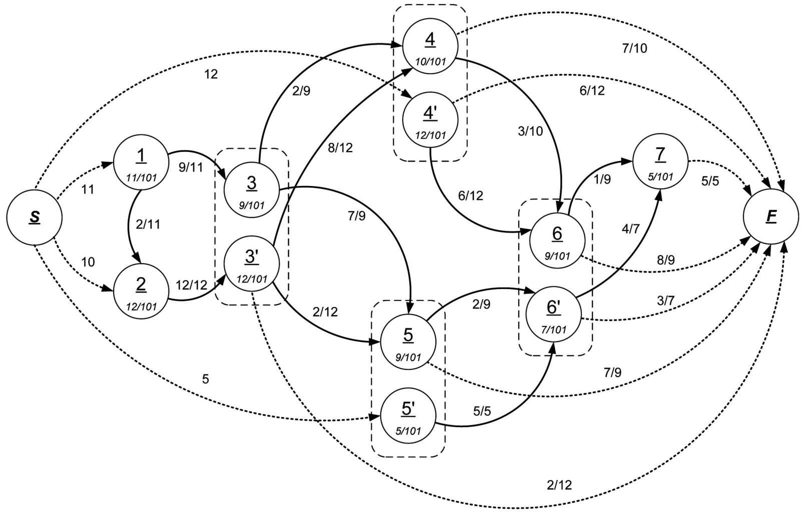


Fig. 2. The second-order model for the example given in Fig. 1.

given together with the maximum and average error attained by the first- and second-order models when representing the conditional probabilities. To create the second-order model, the method uses the K-Means clustering algorithm to group inlinks that induce similar conditional probabilities, in this case, links $(1, 2)$ and $(4, 2)$. Such links are assigned to the same clone. Depending on the γ value, the method decides if it is necessary to create an additional clone. As referred above, γ can be set to regulate the maximum error (γ_m) or, alternatively, the average error (γ_a). In the example, the model in Fig. 4 is considered accurate for $\gamma_a = 0.04$; however, for $\gamma_a = 0.02$, one more clone of state 2 is needed to increase second-order accuracy.

The method to extend a model to higher orders is identical. N-order conditional probability estimates are compared to the corresponding lower order estimates, and cloning is applied to states that are not accurate in order to separate their n-state length in-paths. A formal description of the method is given in [10]. We have shown experimentally in [10] that the running time is approximately linear with respect to the order of the model.

We note that the first-order model is incremental [8]; however, a given higher order model cannot be easily incremented within the current framework, since the decision of cloning or not cloning a state, and the corresponding clustering method, is based on the data available at the time.

2.3 Summarization Ability Evaluation

From the model definition, it follows that a first-order model accurately represents the probability estimates of all trails of length 2, that is, trails that visit two pages. Similarly, a second-order model accurately represents the probability estimates for all trails of length up to 3, and so on for higher order models. In practice, we will only build models up to a fixed order, say n, and so we would like to measure the ability of this n-order model to provide probability estimates for trails that are longer than n; we call the output of such a measure the *summarization ability* of a model.

We make use of two metrics to measure the summarization ability of a model: 1) the Spearman footrule with a location parameter [12], which measures the proximity between two top_m lists, and 2) the overlap between two top_m lists. In this context, each element of a list is a trail of a given length, and the trails are ordered by probability.

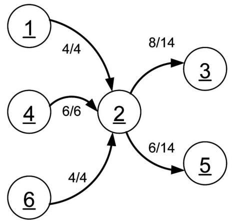
The metrics are defined as follows: Given two top_m lists L_1 and L_2, each with m elements, we let L be the union of the two lists and the location parameter be $m + 1$. In addition, we let $f(i)$ be a function that returns the ranking of any element $i \in L$ in L_1, and when $i \notin L_i$, we let $f(i) = m + 1$; $g(i)$ is the equivalent function for L_2.

The footrule metric is now defined as

$$F(L_1, L_2) = 1 - \frac{\sum_{i \in L} |f(i) - g(i)|}{MAX},$$

session	*freq.*
1,2,3	3
1,2,5	1
4,2,3	4
4,2,5	2
6,2,3	1
6,2,5	3

(a)



(b)

Fig. 3. An example of (a) a collection of sessions and (b) the corresponding first-order model.

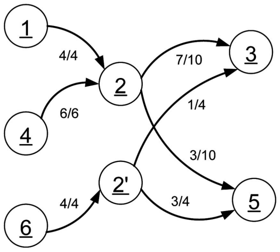


Fig. 4. The second-order model corresponding to the collection of sessions given in Fig. 3a.

TABLE 1
The Accuracy Assessment for the Models Given in Figs. 3 and 4

probability	**input data**	**1st order**		**2nd order**	
	prob.	*prob.*	*diff.*	*prob*	*diff*
$p(3\|1,2)$	3/4	8/14	0.18	7/10	0.05
$p(5\|1,2)$	1/4	6/14	0.18	3/10	0.05
$p(3\|4,2)$	4/6	8/14	0.10	7/10	0.03
$p(5\|4,2)$	2/6	6/14	0.10	3/10	0.03
$p(3\|6,2)$	1/4	8/14	0.32	1/4	0.00
$p(5\|6,2)$	3/4	6/14	0.32	3/4	0.00
	max		0.32		0.05
	avg		0.20		0.03

TABLE 2
The Ranking of 3-Grams as Given by the Input Data (L_1), the First-Order Model (L_2), and the Second-Order Model (L_3)

	L_1		L_2		L_3	
rank	*3-gram*	*freq*	*trail*	*prob*	*trail*	*prob*
1	2,3,4	8	4,6,F	0.061	2,3,4	0.0792
2	4,6,F	8	3,4,F	0.059	4,6,F	0.0792
3	1,3,5	7	2,3,4	0.057	1,3,5	0.0693
4	3,4,F	7	2,3,5	0.051	3,4,F	0.0693
5	3,5,F	7	6,7,F	0.050	3,5,F	0.0693
		footrule	0.60		1.00	
		overlap	0.60		1.00	

where MAX is a normalization constant, which for a top_m list is $m \cdot (m+1)$, corresponding to the case when there is no overlap between the two lists. We subtract 1 from the fraction to measure proximity rather than distance. In addition, we use list overlap to measure the proportion of elements the model identified from the reference set, which is the ranking that is considered to be correct. Given, the two top_m lists L_1 and L_2 and assuming that L_1 is the reference set, the overlap is defined as the percentage of elements in L_2 that occur in L_1. We note that although the overlap provides a simple measure of the L_2 quality, the footrule metric has the advantage of taking into account the relative ranking of the elements occurring in both lists.

We now illustrate the method using an example. We will measure the accuracy with which the first-order model, shown in Fig. 1, represents the trails having length 3 that are given in the input data. First, from the collection of sessions, we induce all sequences of three pages, the 3-grams, and rank them by frequency count. The top_m 3-grams, with $m = 5$, will constitute the reference set L_1. Second, we use a Breadth-First-Search (BFS) algorithm to infer the set of trails induced by the first-order model and each trail probability estimate (see [13] for details on the algorithm and its average linear time complexity). The top_m trails induced by the model form the second list L_2. To ensure consistency between the rankings, we lexicographically sort the 3-grams having the same frequency count and the trails having the same probability. Similarly, L_3 and L_4 are the lists of the top_m trails induced by the second- and third-order models.

Table 2 presents the results for the first- and second-order models. For the first-order model, three of the top_5 trails are in the top_5 3-grams; therefore, the overlap between L_2 and the reference set L_1 is $3/5 = 0.60$. The union of the two lists has seven elements, and for each of its elements, we compute the rank absolute difference as given by the two lists. For example, the ranking given by the reference set L_1 for trail $(2,3,4)$ is $f(2,3,4) = 1$ and by the set being assessed L_2 is $g(2,3,4) = 3$; therefore, the trail's contribution for the overall metric is $|f(2,3,4) - g(2,3,4)| = |1 - 3| = 2$. The footrule metric has the value $F(L_1, L_2) = 1 - 12/30$, which in this case is 0.60. As expected, the second-order model induces a ranking that is exactly the same as the one given by the 3-gram frequency counts.

In Table 3, we present the top_3 results for models up to the third-order when analyzing their ability to represent trails having length 5. (We consider only the top_3, since in the input data, there are only three distinct 5-grams.) It is interesting to note that the first-order model identifies two of the top three trails; however, the missing trail is ranked in the 11th place by the model (not shown in the table). As expected, the accuracy of the results improves as the order of the model increases.

2.4 Predictive Power Evaluation

In previous work, we have presented a method to evaluate a model's ability to predict the last page of a session based on the preceding sequence of pages viewed [11]. We will now review this method.

To assess a model's prediction ability, we randomly split the set of input trails into a training and a test set and then induce the model from the training set. For each test trail of length *l*, we call its last state the prediction target (tg), and we use the $l - 1$ trail prefix to predict it. The induced model gives the set of reachable pages (rp) from the tip of the prefix, assuming that the user has followed the sequence of pages defined by the prefix. The reachable pages are ranked by probability, with the most probable page having $rank = 1$. In order to measure the prediction accuracy of the model, we let the *Absolute Error* (AE) measure the prediction strength by setting $\text{AE} = rank - 1$; that is, the closer AE is to zero, the better the prediction is. The overall prediction accuracy metric is given by the *Mean Absolute Error* (MAE) that is defined as the sum of AE over all the test trails divided by the number of test trails.

TABLE 3
The Ranking of 5-Grams as Given by the Input Data and by Models of up to the Third Order

	L_1		L_2		L_3		L_4	
rank	*5-gram*	*freq*	*trail*	*prob*	*trail*	*prob*	*trail*	*prob*
1	2,3,4,6,F	3	2,3,5,6,F	0.0175	2,3,4,6,F	0.0211	2,3,4,6,F	0.0297
2	1,2,3,4,F	2	2,3,4,6,F	0.0159	3,5,6,7,F	0.0113	2,3,5,6,F	0.0198
3	2,3,5,6,F	2	3,5,6,7,F	0.0139	1,2,3,4,F	0.0092	1,2,3,4,F	0.0124
		footrule	0.50		0.67		0.83	
		overlap	0.67		0.67		1.00	

TABLE 4
The Prediction Results with the First-Order Model

trail	*tg*	*rp*	*prob.*	*rank*	*AE*
1,3,5	5	4	10/21	1	
		5	9/21	2	1
		F	2/21	3	
2,3,5,6	6	6	7/14	1	0
		F	7/14	1	
1,3,5,6,7	7	F	11/16	1	
		7	5/16	2	1
					MAE = 0.667

We illustrate the method by assuming that the trails given in Fig. 1a constitute the training set and that the test set is composed of three trails: {(1, 3, 5), (2, 3, 5, 6), and (1, 3, 5, 6, 7)}. According to the first-order model, after following the prefix $(1, 3)$, there are three reachable pages (see Fig. 1b), the pages corresponding to states 4, 5, and F (the latter corresponds to terminating the navigation session). In this case, the prediction target (state 5) has rank 2 among the reachable pages resulting in an absolute error $\mathrm{AE} = 2 - 1 = 1$. According to the second-order model (see Fig. 2), after following $(1, 3)$, the most probable choice is the link to page 5. Therefore, for the first trail, the second-order model provides a more accurate prediction than the first-order model. For the second test trail, the opposite occurs, although the overall MAE metric for the example confirms that the second-order model provides better predictions; see Tables 4 and 5 for details.

3 Experimental Evaluation

3.1 Data Sets Description

We conducted experiments with three real data sets. The first data set (CS) was made available by the authors of [2]. It originates from the DePaul University CTI Web site (www.cs.depaul.edu) and corresponds to two weeks of site usage during 2002; sessions were inferred using cookies. The second data set (MM) corresponds to two weeks of site usage from the Music Machines site (machines.hyperreal .org) during 1999 and was made available by the authors of [16]. The third data set (LTM) represents a month of site usage from the London Transport Museum Web site (www.ltmuseum.co.uk) during January 2003. In the CS data set, the sessions were already identified, and for the other two data sets, a session was defined as a sequence of requests from the same IP address with a time limit of 30 minutes between consecutive requests. Erroneous and image requests were eliminated, although for the MM data set, .jpg requests were left in, since in that specific site, they correspond to page views. When preprocessing the data sets, we set a session length limit of 15 requests, and therefore, very long sessions were split into two or more shorter sessions.

TABLE 5
The Prediction Results with the Second-Order Model

trail	*tg*	*rp*	*prob.*	*rank*	*AE*
1,3,5	5	5	7/9	1	0
		4	2/9	2	
2,3,5,6	6	F	7/9	1	
		6	2/9	2	1
1,3,5,6,7	7	7	4/7	1	0
		F	3/7	2	
					MAE = 0.333

TABLE 6
Summary Characteristics of the Three Real Data Sets Used

data set	*pages*	*requests*	*sessions*	$l = 1$	$l = 2$	$l = 3$
CS	547	115448	24548	7148	3474	2202
LTM	1362	372434	47021	13489	3428	1893
MM	8237	303186	50192	12644	7891	4925

Table 6 summarizes the characteristics of the data sets. For each data set, we indicate the number of pages occurring in the log file and the total number of requests recorded. We also give the total number of sessions derived from each data set and the number of sessions of lengths 1 ($l = 1$), 2 ($l = 2$), and 3 ($l = 3$); session length is measured by the number of requests a session is composed of.

From a collection of sessions, we infer the corresponding n-grams. An n-gram is defined as a sequence of n consecutive requests. Fig. 5 shows the distribution of the n-gram frequency counts for the three data sets. For example, the frequency count of the 2-grams gives the number of distinct sequences of two pages that occur in the collection of sessions. Each user session was modified so that it starts and finishes at a fixed artificial page, resulting in sequences of at most 17 requests as shown in the plot. As expected, there is a higher variety of shorter sequences of pages implying that long sequences of page views are generally rare.

From the collection of sessions, a first-order model was inferred. This model was then evaluated for second and higher order conditional probabilities and, if needed, a state was cloned to separate the in-paths due to differences in the conditional probabilities. As described above, the γ parameter sets the tolerance allowed on representing the conditional probabilities. In addition, there is a parameter that specifies the minimum number of times a page has to

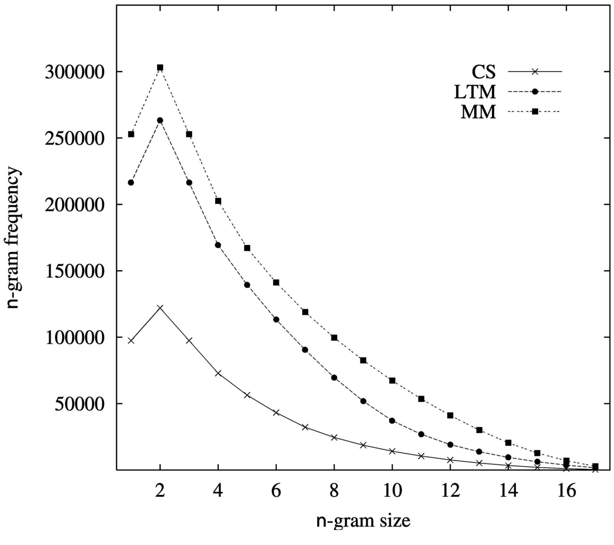


Fig. 5. The n-grams' distribution.

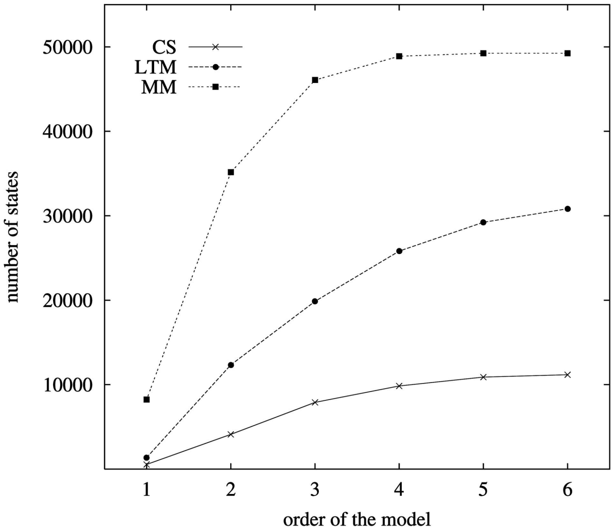


Fig. 6. The distribution of the number of states with the model order when $\gamma = 0$.

be requested in order to be considered for cloning; in these experiments, we set $num\_visits \geq 30$. Fig. 6 shows the variation of the model number of states when the order of the model increases while having the accuracy threshold set to $\gamma = 0$. The figure shows that there is a fast increase in the number of states for the second- and third-order models. For higher orders, the number of states increases at a slower rate, which is an indication that the gain in accuracy when using higher order models is smaller.

3.2 Model Summarization Ability Evaluation

As discussed in Section 2.2, when $\gamma = 0$, a model of a given order accurately represents trails of length $\leq$ order + 1 (the length of a trail is measured by the number of page views). For example, a first-order model accurately represents the probability of a transition between any two states, and a second-order model accurately represents the probability of trails composed of three consecutive states. In the following, we make use of the method described in Section 2.3 to assess the ability of a model to represent long trails. Briefly, the method compares the ranking produced by the BFS algorithm that infers the trails' probability estimates with the ranking induced by the n-gram frequency count. When the model is accurate, the two resulting rankings are identical.

For this purpose, we consider three parameters: 1) the cut point (λ), that is, only trails having probability above λ are considered, 2) the maximum length of a trail induced by the algorithm (mtl), and 3) the size of the ranked lists (top_m). We consider two variations for the mtl parameter: 1) a strict definition ($mtl_=$) that takes into account only trails with the specified length and 2) a nonstrict definition ($mtl_{<=}$) that takes into account trails with length less than or equal to the specified limit. In the second variation, we filter out subtrails, that is, trails that are prefixes of longer trails whose probability is also above the cut point since, by definition of trail probability, all trail prefixes have probability greater or equal to that of the full trail. As mentioned in Section 2.3, in order to compare the rankings, we make use of two metrics: 1) *overlap* and 2) Spearman's *footrule* metric. The overlap measures the percentage of trails that occur in both rankings, whereas the footrule takes into account not only the overlap, but also the differences of rank ordering between the lists. We note that although the cut point is essential to reduce the search space for the BFS algorithm, we will not emphasize it in our presentation, since the mtl is the parameter that directly influences the top_m ranking list. In fact, the cut point was necessary for large values of mtl due to the very large number of trails to assess; however, its value did not have any impact on the resulting top_m ranking.

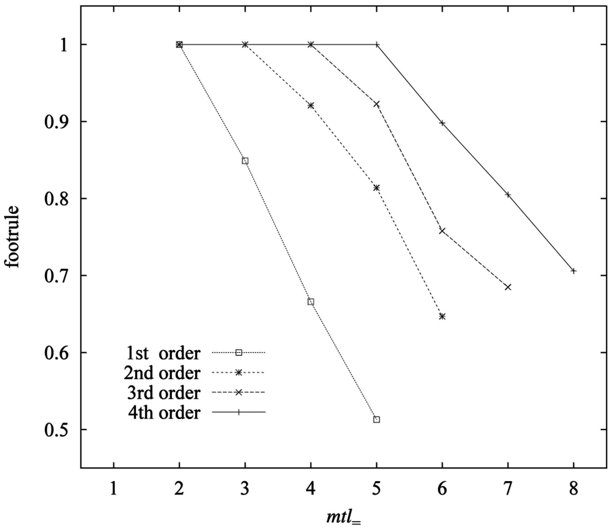


Fig. 7. The footrule measure for the top_250 trails on the LTM data set when using the maximum trail length strict definition ($mtl_=$).

Fig. 7 shows, for the LTM data set, the footrule value for the top_250 trails when using the strict trail length definition $mtl_=$. The results show that a first-order model accurately ranks trails composed of one and two requests. For trails of length 3, the value of the footrule is 0.85 revealing a close to linear decrease for longer trails. We point out that the

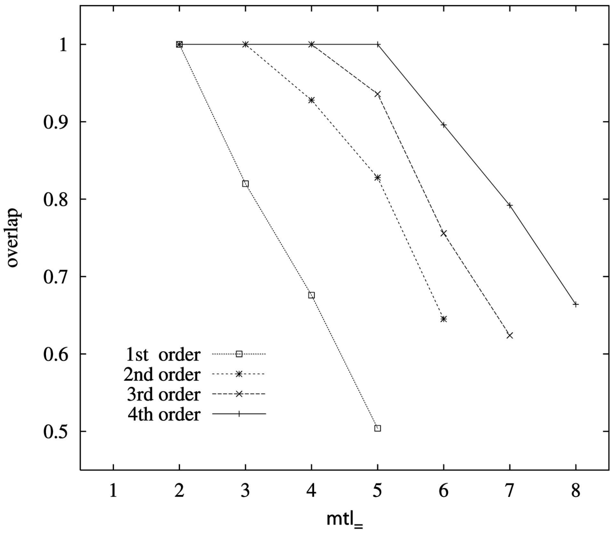


Fig. 8. The overlap measure for the top_250 trails on the LTM data set when using the maximum trail length strict definition ($mtl_=$).

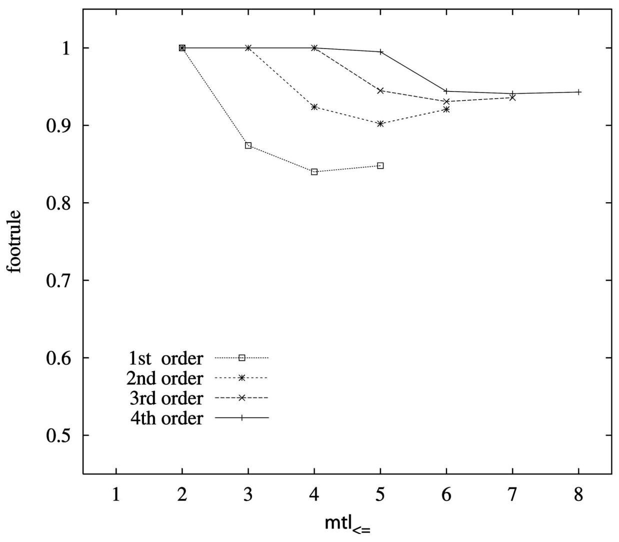


Fig. 9. The footrule measure for the top_250 trails on the CS data set when using the maximum trail nonstrict definition $(mtl_{<=})$.

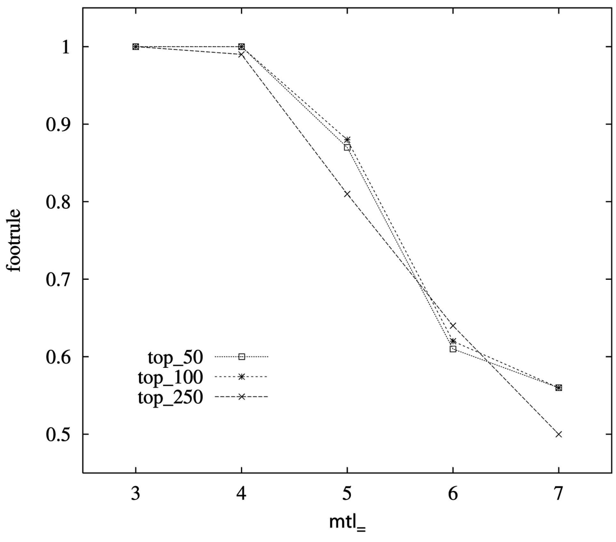


Fig. 11. Analysis of the footrule measure sensitivity to the size of the top_m rankings for the MM data set.

second-order model shows a footrule value of 0.92 for trails of length 4. Fig. 8 shows the overlap measure whose results are very similar to the footrule metric. The overlap value shows a close to linear decrease as $mtl_=$ increases. It is interesting to note that a first-order model achieves an overlap of 0.50 for trails having length 5, which means that 50 percent of the top_250 trails are identified in the list although probably not in the correct rank order. A second-order model achieves an overlap of 0.64 for trails of length 6. Similar results were obtained for the other two data sets.

Fig. 9 shows, for the CS data set, the footrule results when the nonstrict definition of the trail length limit parameter $(mtl_{<=})$ is used. Again, the results given by the overlap metric show a similar pattern. In general, the nonstrict definition gives better results for both the footrule and the overlap metrics, especially for longer trails. For example, for the CS data set and trails having length 5, the second-order model achieves $\mathrm{footrule} = 0.9$ for the nonstrict definition (as seen in Fig. 9) and $\mathrm{footrule} = 0.78$ for the strict definition. Whereas the former is evaluating the ability to rank trails having length equal to or shorter than 5, the latter is assessing the ability of ranking trails of length 5 only. We note that shorter trails are, in general, more probable, and when using the nonstrict definition, if $mtl_{<=}$ increases from 4 to 5, very few additional trails with length 5 are included in the updated top_250. Therefore, the strict definition for the mtl parameter is more adequate for evaluating a model's ability to represent long trails.

Fig. 10 shows, for a third-order model of the CS data set, the variation of the footrule measure for different sizes of the top_m ranking. The footrule is smaller for larger m, but the differences are not very significant for trails of length up to 6. Similar results were obtained for the MM data set as shown in Fig. 11. Therefore, from now on, we restrict our analysis to the top_250 rankings.

Figs. 12 and 13 show the number of states variation with the γ parameter. For the results in Fig. 12, the γ parameter

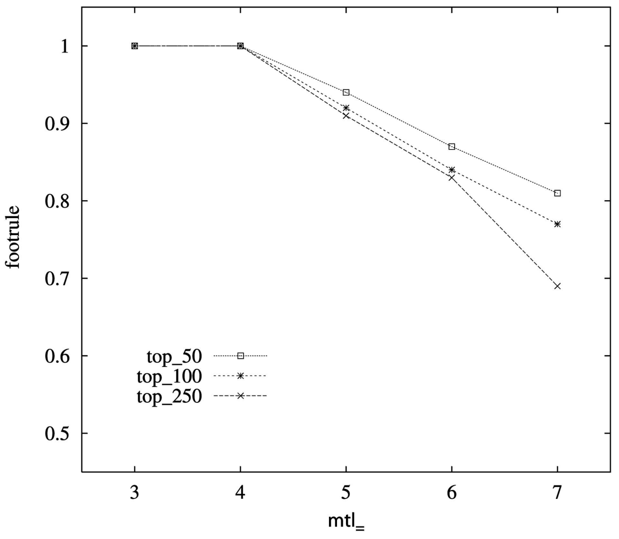


Fig. 10. Analysis of the footrule measure sensitivity to the size of the top_m rankings for the CS data set.

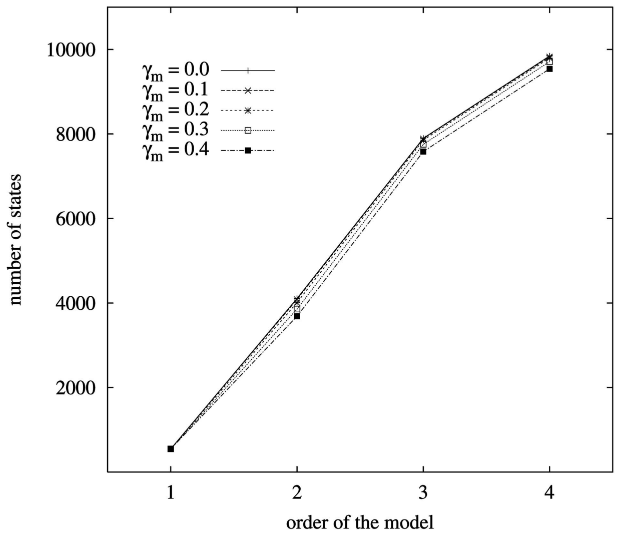


Fig. 12. The variation of the number of states with the γ_m version of the accuracy threshold parameter.

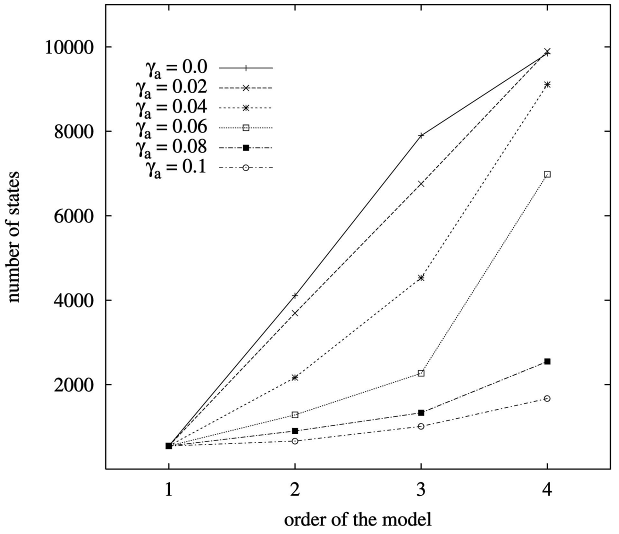


Fig. 13. The variation of the number of states with the γ_a version of the accuracy threshold parameter.

measures the maximum divergence between a conditional probability and the corresponding lower order probability; in this case, the parameter is denoted by γ_m. For the results in Fig. 13, γ measures the average difference between the in-paths conditional probabilities and the corresponding link probabilities. In this case, we will denote the parameter by γ_a. The results suggest that γ_a is better since it provides more control over the resulting number of states in the model.

Fig. 14 shows the footrule variation with γ_a for a third-order model on the CS data set taking the top_250 rankings. It is interesting to note that the plotted lines are close to parallel, which indicates that the decrease in the footrule value for an increment of $mtl_=$ is close to being independent of the value of γ_a. Also, when $mtl_=$ is set to 5 and γ_a increases from 0 to 0.1, the footrule decreases from 0.91 to 0.65, that is, it decreases by 29 percent. However, the number of states decreases from 7,897 to 1,011, which corresponds to a decrease of about 87 percent.

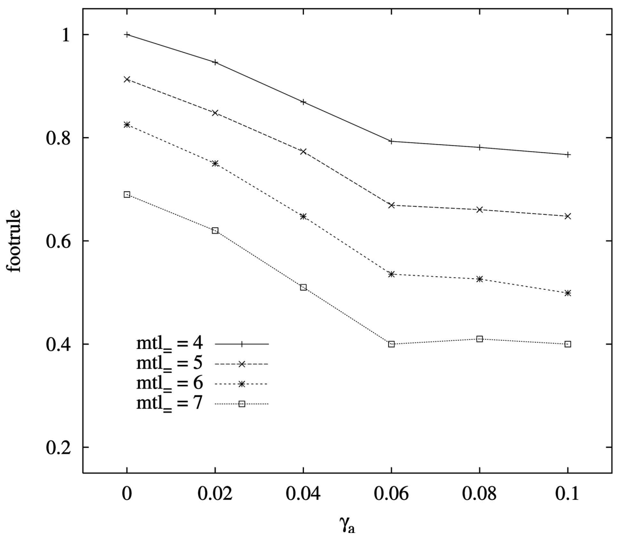


Fig. 14. The variation of the footrule measure with γ_a for third-order models on the CS data set.

TABLE 7
The MAE and st_MAE Measures for the Temporal-Based Fivefold Cross Validation on the CS Data Set When $\gamma_a = 0.0$

	k=2			**k=3**			**k=4**		
order	*states*	*MAE*	*st_MAE*	*states*	*MAE*	*st_MAE*	*states*	*MAE*	*st_MAE*
1st	411	4.415	0.099	464	5.227	0.105	511	4.944	0.083
2nd	2315	3.298	0.074	3052	3.710	0.075	3631	3.576	0.060
3rd	3727	3.087	0.069	5320	3.392	0.068	6661	3.290	0.055
4th	4439	3.030	0.068	6472	3.296	0.066	8262	3.217	0.054

3.3 Model Predictive Power Evaluation

In the following, we have decided to use the CS data set for the analysis of the predictive power of a model and its relationship to summarization ability, since its sessions are more reliable than those inferred from the other data sets due to the use of cookies to identify the user. In fact, the use of cookies enables us to identify the user more precisely when placing a page request, and therefore, utilizing such information results in more accurate inference of user navigation sessions.

We use cross validation in order to assess the predictive power of a model; that is, we split the data set into k folds and use $k-1$ folds as the training set and the remaining fold as the test set. In these experiments, we will split the collection of trails in such a way that all trails in the ith subset temporally precede the trails in the $(i+1)$th subset. For $k=i$, we infer the model from all the trail subsets from 1 to i and measure the prediction accuracy by using the $(i+1)$th subset as our test set. By splitting the data in this way, we maintain the temporal order among the partitions. We make use of the MAE metric to measure the model's prediction accuracy. In addition, we define the complement of MAE (MAE_c), which results from computing the rank of the prediction target relative to the last of the set of reachable pages. Since the sum of MAE and MAE_c is constant for a given k, we define the normalized measure $st\_MAE = MAE/(MAE + MAE\_c)$. The normalized measure is interesting, since it takes into account the difficulty associated with the prediction, where a prediction is harder if it has more possible outcomes.

Table 7 presents the results for models created with $\gamma_a = 0$. For all configurations of k, the prediction accuracy improves with the order of the model, but the gain when moving to the fourth-order model is relatively small. It is interesting to note that, for first-order models, both the MAE and st_MAE metrics increase from $k=2$ to $k=3$ but decrease when moving to $k=4$. We remind the reader that the higher k is, the larger the data set used to construct the model, resulting in models with a larger number of states (as can be verified in Table 7) on which predictions are more difficult. Another aspect that should be taken into account is that, when k increases, the sessions in the training set cover a larger period of time. For example, for $k=4$, the model uses data from four periods of time and tests trails from the last period. In cases where the user's behavior changes over time, a model trained with data covering a larger period may have problems in generalizing for future user behavior.

TABLE 8
The MAE and st_MAE Measures on the CS Data Set for the Non-Temporal-Based Fivefold Cross Validation When $\gamma_a = 0.0$

order	*states*	*MAE*	*st_MAE*
1st	512	4.787	0.081
2nd	4000	3.471	0.059
3rd	9533	3.090	0.052
4th	13173	2.961	0.050

Table 8 gives the results obtained with the standard fivefold cross validation; we note that in [11], we used the standard cross validation rather than the temporal-based version used herein. When comparing the results, we must take into account the fact that each value presented corresponds to the average result for the five splits. Moreover, the number of states is larger than that obtained from the temporal-based split, since we used $num\_visits \geq 15$, which leads to more states being cloned. However, overall, the results are very similar to those obtained with the temporal-based cross validation.

Table 9 gives the results obtained when $\gamma_a = 0.08$. As expected, when γ_a increases, the accuracy of the conditional probabilities decreases, but we obtain models having a significantly smaller number of states. For a gain in the size of the model, there is a cost in the loss of prediction accuracy. For example, for $k = 4$ folds, a second-order model built with $\gamma_a = 0$ has 3,631 states, and when $\gamma_a = 0.08$, it has 829 states; that is, only 22.8 percent of the states are utilized in the latter case. However, the normalized error measure increases from 0.06 to 0.081, which corresponds to an increase of 31 percent.

3.4 Focused Discussion

We now analyze the relationship between the footrule and the st_MAE measures. Whereas the footrule measures the model's summarization ability, the st_MAE metric measures the model's prediction ability. To highlight the difference, summarization represents the knowledge present in a collection of trails and prediction attempts to generalize this knowledge so that the outcome of future events can be predicted. In this set of experiments, we set $mtl_{=}$ to 4 and the list sizes to the top_250 trails.

Fig. 15 presents the results for a second-order model, with γ_a set to the values in {0.00, 0.02, 0.04, 0.06, 0.08, 0.10}. Each plotted line corresponds to a cross-validation partition, and each point corresponds to a γ_a value. When γ_a increases, the value of the footrule decreases and the value of st_MAE increases. When adjusting a regression line for $k = 2$, that is, for the twofold partition, we obtain 0.979 for the coefficient of determination, meaning that 98 percent of the footrule variation is explained by the variation of st_MAE. The other partitions give even higher values for the coefficient of determination.

TABLE 9
The MAE and st_MAE Measures for the Fivefold Cross Validation on the CS Data Set When $\gamma_a = 0.08$

	k=2			**k=3**			**k=4**		
order	*states*	*MAE*	*st_MAE*	*states*	*MAE*	*st_MAE*	*states*	*MAE*	*st_MAE*
1st	411	4.415	0.099	464	5.227	0.105	511	4.944	0.083
2nd	725	4.352	0.097	796	5.164	0.104	829	4.877	0.081
3rd	1152	4.173	0.093	1301	4.724	0.095	1226	4.579	0.076
4th	1823	3.808	0.085	2518	4.270	0.086	2356	4.181	0.070

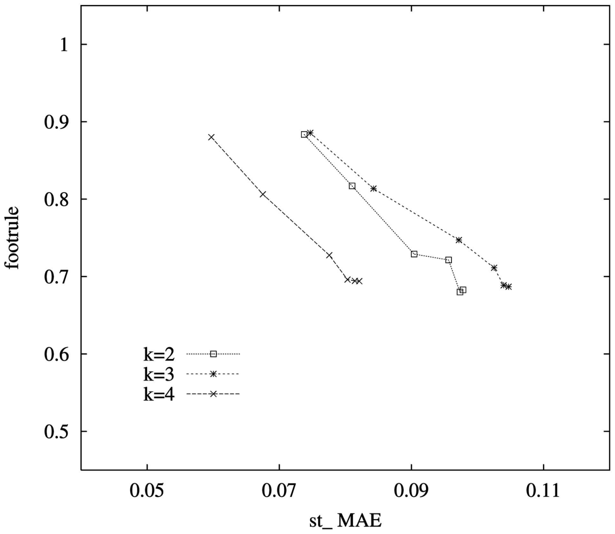


Fig. 15. The relationship between the footrule measure and the st_MAE measure for second-order models while varying both γ_a and the cross-validation partition on the CS data sets.

Fig. 16 presents the results for the fourfold partition and for $mtl_{=}$ set to 4 while varying both the model order and the value of γ_a. The first-order model corresponds to a single point, since the γ_a value has no effect in this case; moreover, it corresponds to the worst performance with respect to both measures. The regression line adjusted to the fourth-order results plot gives 0.989 for the coefficient of determination (which is the lowest among the three plotted lines). Therefore, the results suggest that prediction accuracy improves linearly with summarization ability.

Finally, we analyze the top_10 trails extracted from the MM data set with models of different orders. Whereas Table 10 presents the URLs occurring in the trails and

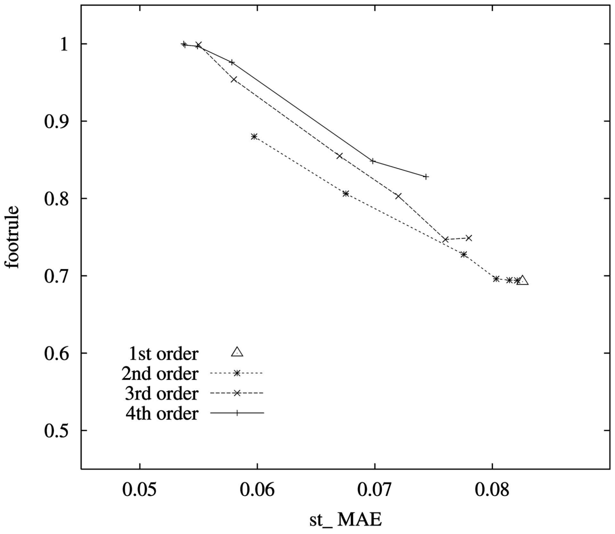


Fig. 16. The relationship between the footrule measure and the st_MAE measure for the $k = 4$ cross-validation partition while varying both γ_a and the model order on the CS data sets.

TABLE 10
The URLs Occurring in the Rules Given in Table 11

ID	*URL*
1	/
2	/music/machines/analogue-heaven
3	/manufacturers/roland
4	/music/machines/samples.html
5	/music/machines
6	/samples.html
7	/machines
8	/manufacturers
9	/music/machines/categories/software/windows
10	/categories/software/windows
11	/music/machines/categories/drum-machines/samples
12	/analogue-heaven
13	/search.cgi
14	/guide
15	/analogue-heaven
16	/music/machines/images/farley3.jpg
17	/guide/finding.html
18	/categories/software/windows/readme
19	/categories/drum-machines/samples
20	/analogue-heaven/email.html

TABLE 11
The top_10 Rules for the MM Data Set Obtained with Models of First, Second, and Third Order for $\gamma = 0$

rank	*1st Order*	*2nd Order*	*3rd Order*
1	2,15,1,F	**4,6,11,19**	5,1,4,6
2	2,16,5,1	**5,1,4,6**	4,6,11,19
3	**1,4,6,F**	**1,4,6,F**	5,1,9,10
4	2,16,5,F	**5,1,9,10**	5,1,12,3
5	**5,1,4,6**	**7,1,9,10**	1,4,6,F
6	5,1,13,F	**4,6,4,6**	4,6,4,6
7	2,15,1,8	1,9,10,F	1,14,17,8
8	5,1,8,F	**5,1,8,3**	5,1,8,3
9	2,15,20,F	**5,1,12,3**	7,1,9,10
10	**1,14,17,8**	4,6,2,15	1,9,10,18
footrule	0.236	0.792	1.000
overlap	0.300	0.800	1.000

provide an ID for each URL, Table 11 shows the 10 top-ranked trails having length 4 that are inferred from first-, second-, and third-order models, with $\gamma_a = 0.0$. The footrule and the overlap metrics are also indicated. As expected, the third-order model ranks accurately the trails of length 4, and we therefore use it as the reference ranking for the other models. The first-order model is able to identify three of the top_10 trails, although with a different ranking. The second-order model is able to identify eight of the top_10 trails. For lower order models, the trails in boldface were identified from the reference ranking.

4 Concluding Remarks

In previous work, we have presented a model for implementing a VLMC [9], [10] and a method for measuring the predictive power of such models [11]. Herein, we propose a new method to evaluate the summarization ability of the model, making use of the Spearman footrule metric to assess the accuracy with which a model represents the information content of a collection of user Web navigation sessions. In addition, we study the relationship of our newly proposed metric to evaluate the summarization ability of the model and the previously proposed metric to evaluate the predictive power of the model.

A model that accurately summarizes the information content of a collection of navigation sessions can provide a platform for techniques focused on identifying user navigation patterns. Moreover, a model showing strong predictive power provides the means to predict the next link choice of unseen user navigation sessions and, thus, can be used for prefetching links or in adaptive Web site applications.

We presented the results of an extensive experimental evaluation conducted on three real-world data sets, which provide strong evidence that there is a linear relationship between predictive power and summarization ability.

In this work, we have limited our study to the prediction of the next link choice; however, the method can be generalized to the prediction of longer sequences of navigation options.

The methods given can be used by a Web site owner to understand and adapt the site to its visitors' behavior. We note that the length of the navigation history a user takes into account when deciding which page to visit next varies from site to site. For example, when reading an online newspaper, a user probably chooses which article to read in the sports section independently of the contents of the pages read before reaching the sports section. On the other hand, when reading an online tutorial on how to perform a given task, a user is expected to follow trails in sequence, as suggested by the author.

Being able to measure the accuracy with which distinct higher order models represent the user behavior helps us to understand the history length users take into account while navigating within a site. Having established that, the site owner can use the corresponding n-order model to identify frequent navigation patterns within the site. The identified patterns can then be compared to the site's business objectives and guide the optimization of the site's underlying topology. In addition, a VLMC model can be used to provide adaptive pages by, for example, providing an ordered set of link suggestions focused on assisting the user to satisfy his navigation goals.

As future work, we are planning to incorporate our model within an adaptive Web site platform in order to assess it as an online tool to provide link suggestions for users navigating within a site.

Acknowledgments

J. Borges is supported by the POSC/EIA/58367/2004/Site-o-Matic Project (Fundação Ciencia e Tecnologia) cofinanced by FEDER.

References

[1] B. Mobasher, "Web Usage Mining and Personalization," *Practical Handbook of Internet Computing,* M.P. Singh, ed. Chapman Hall & CRC Press, 2004.

[2] M. Spiliopoulou, B. Mobasher, B. Berendt, and M. Nakagawa, "A Framework for the Evaluation of Session Reconstruction Heuristics in Web Usage Analysis," *INFORMS J. Computing,* no. 15, pp. 171-190, 2003.

[3] S. Schechter, M. Krishnan, and M. Smith, "Using Path Profiles to Predict HTTP Requests," *Computer Networks and ISDN Systems,* vol. 30, pp. 457-467, 1998.

[4] X. Dongshan and S. Juni, "A New Markov Model for Web Access Prediction," *IEEE Computing in Science and Eng.,* vol. 4, pp. 34-39, Nov. 2002.

[5] X. Chen and X. Zhang, "A Popularity-Based Prediction Model for Web Prefetching," *Computer,* pp. 63-70, 2003.

[6] M. Deshpande and G. Karypis, "Selective Markov Models for Predicting Web Page Accesses," *ACM Trans. Internet Technology,* vol. 4, pp. 163-184, May 2004.

[7] M. Eirinaki, M. Vazirgiannis, and D. Kapogiannis, "Web Path Recommendations Based on Page Ranking and Markov Models," *Proc. Seventh Ann. ACM Int'l Workshop Web Information and Data Management (WIDM '05),* pp. 2-9, 2005.

[8] J. Borges and M. Levene, "Data Mining of User Navigation Patterns," *Web Usage Analysis and User Profiling,* B. Masand and M. Spiliopoulou, eds., LNAI 1836, pp. 92-111, Springer, 2000.

[9] J. Borges and M. Levene, "A Clustering-Based Approach for Modelling User Navigation with Increased Accuracy," *Proc. Second Int'l Workshop Knowledge Discovery from Data Streams,* pp. 77-86, Oct. 2005.

[10] J. Borges and M. Levene, "Generating Dynamic Higher-Order Markov Models in Web Usage Mining," *Proc. Ninth European Conf. Principles and Practice of Knowledge Discovery in Databases (PKDD),* A. Jorge, L. Torgo, P. Brazdil, R. Camacho, and J. Gama, eds., pp. 34-45, Oct. 2005.

[11] J. Borges and M. Levene, "Testing the Predictive Power of Variable History Web Usage," *J. Soft Computing,* special issue on Web intelligence, 2006.

[12] R. Fagin, R. Kumar, and D. Sivakumar, "Comparing Top k Lists," *SIAM J. Discrete Math.*, vol. 17, no. 1, pp. 134-160, Nov. 2003.

[13] J. Borges and M. Levene, "An Average Linear Time Algorithm for Web Usage Mining," *Int'l J. Information Technology and Decision Making,* vol. 3, no. 2, pp. 307-319, June 2004.

[14] S. Jespersen, T. Pedersen, and J. Thorhauge, "Evaluating the Markov Assumption for Web Usage Mining," *Proc. Fifth ACM Int'l Workshop Web Information and Data Management,* pp. 82-89, 2003.

[15] G. Bejerano, "Algorithms for Variable Length Markov Chain Modelling," *Bioinformatics,* vol. 20, pp. 788-789, Mar. 2004.

[16] M. Perkowitz and O. Etzioni, "Towards Adaptive Web Sites: Conceptual Framework and Case Study," *Artificial Intelligence,* vol. 118, no. 2000, pp. 245-275, 2000.

José Borges received the MSc degree in electrical engineering and computer science in 1994 from the School of Engineering, University of Porto, and the PhD degree in computer science in 2000 from the University College of London. He is currently an auxiliary professor in the School of Engineering at the University of Porto. His main research interests are Web data mining and data analysis, and he has published several papers on the field of Web mining.

Mark Levene received the BSc degree in computer science from Auckland University, New Zealand, in 1982 and the PhD degree in computer science in 1990 from Birkbeck College University of London. He is currently a professor of computer science at Birkbeck College, where he is a member of the Information Management and Web Technologies Research Group. His main research interests are Web search and navigation, Web data mining, and stochastic models for the evolution of the Web. He has published extensively in the areas of database theory and Web technologies and has recently published a book called *An Introduction to Search and Engines and Web Navigation*.