Marcov Models for Web Access Prediction

Raluca Popa, Tihamér Levendovszky

praluca@aut.bme.hu, tihamer@aut.bme.hu

Abstract: The problem of predicting user's behavior on a Web site has fundamental significance due to the rapid growth of the World Wide Web. Although traditional Markov models have been found to be suited for addressing this problem, they have serious limitations. Thus, good predictions require new Markov models. Hybrid-order tree-like Markov models predict Web access precisely while providing high coverage and scalability. Selective Markov models intelligently select parts of different order Markov models such that the resulting model has a reduced state of complexity, while maintaining a high predictive accuracy. The goal of our work is to present and compare traditional Markov models and new Markov models, regarding the prediction of users behavior.

Keywords: Selective Markov models, web access prediction, All-kth-Order Markov model, Hybrid-order tree-like Markov model

1 Introduction

The navigation process of a user on a Web site can be modeled as a Markov chain. From this point of view, web pages represent the states of a Markov chain model, and hyperlinks between Web pages the transitions between the states.

The pages already requested by the user determine the pages that a user is likely to request in the future. Figure 1 presents the navigation possibilities between the pages of a Web site with five pages (p1, p2, p3, p4, p5). Information about Web usage contained in a Web link structure can be used to infer the transition probabilities between these states. In this way, the Markov chain model can be used to predict the Web pages that a user is likely to visit given a sequence of Web pages already visited by the user.

The organization of this paper is as follows. Section 2 covers related work. Section 3 describes the Markov models, namely, Section 3.1 presents the main performance measures for Markov models, Section 3.2 describes the limitations of traditional Markov models, and Section 3.3 summarizes some new Markov models, such as hybrid-order tree-like Markov model and selective Markov models.

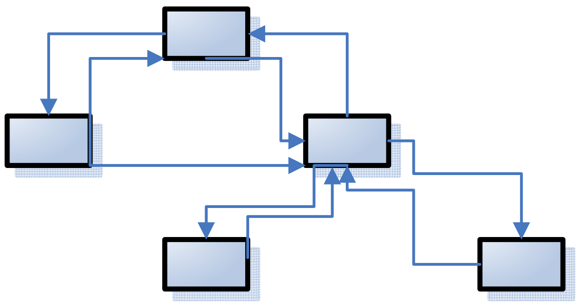

Figure 1
Navigation between pages of a Web site

2 Related Work

The problem of modeling and predicting a user's browsing behavior on a Web site has attracted significant research interest. Regarding the solutions in [1] is presented how to improve the Web cache performance, in [2] the Web prefetching by using N-hop Markov models; in [3] how to enhance search engines and understand and influence buying patterns [4], and personalize the browsing experience [5].

To predict the behavior of a user, we need a method for modeling and analyzing Web access sequences. By having this information, it can be deduced the future requests of the user. Other researchers have attempted to use traditional Markov models, which are employed to study stochastic processes [6], [7]. In general, they use the sequence of Web pages that a user has accessed as input, with the goal of building Markov models with which they can predict the page that the user will most likely access next. In [7], Markov models are used to predict the next page accessed by the user; and in [8] to categorize user sessions. However, in [6] is tested the performance of different order Markov models for Web access prediction and it was found that traditional Markov models seem to be inadequate for predicting the user's behavior. Therefore, new Markov models are needed for Web access prediction. In this paper, we present two technologies: hybrid-order tree-like Markov chain [10], [11] and selective Markov chain [12] for predicting a user's behavior.

3 Markov Models

3.1 Performance Measures for Markov Models

The act of a user browsing a Web site is commonly modeled by observing the set of the visited pages. A *Web session* (*W*) is a set of pages that a user visits, and it is represented by the *sequence* of pages *W* (*P*1,*P*2, … , *Pl*) that were accessed. In this sequence, *Pi* represents the *i*th page in the user's Web session.

The prediction model is built on a set of Web sessions, which refer to as a *training set*. This model will then be evaluated for accuracy on a previously unseen set of Web sessions, called the *test set* [12].

There are several performance metrics that should be taken into account to compare different Markov model-based techniques for solving the next-page prediction problem.

The *accuracy* of the model is the first. Accuracy measures the predictive ability of the model and it is determined using a separate set of Web sessions that was not used during training. One method to do this is by hiding the last page in each of the test set's Web sessions, and using the model to make a prediction of the resulting *trimmed* Web session. The accuracy is defined as the ratio of the number of Web sessions for which the model is able to correctly predict the hidden page to the total number of Web sessions in the test set.

The second performance measure that should be taken into account is the *number of states* of the model which measures the space and time-complexity of learning, and applying the model. A model that requires a large number of states can significantly limit the applicability of Markov models for applications in which fast predictions are critical for real-time performance or for applications with tight memory constraints. The number of states of a Markov model is defined as the total number of states for which a Markov model has estimated the most probable page to be accessed next.

The third is the *coverage* of the model that measures the number of times a Markov model was able to compute a prediction without resorting to the default prediction. The coverage of a Markov model is evaluated on a test set. It is defined as the ratio of the number of Web sessions whose state required for making a prediction was found in the model to the total number of Web sessions in the test set.

The last metric is the *model accuracy*, defined as the accuracy on the portion of the test set for which the model was able to locate the state required for prediction. Thus, it did not perform any default predictions.

Figure 2 depicts the difference regarding accuracy, coverage, model accuracy and model size with the order of Markov model.

3.2 Limitations of Traditional Markov Models

Traditional Markov models were used to predict the next Web page a user will most likely access by matching the user's current access sequence with the user's historical Web access sequences. Using the models, researchers compare elements of each historical Web access sequence's maximal prefixes with elements from the same-length suffixes of the user's current access sequence and obtain the historical sequences with the highest probability of matching elements [13], [14], [15], [16].

The Markov properties:

- Let X = X1X2…XL be a sequence of pages.
- The 0-order Markov model is the unconditional base-rate probability. Thus, the probability of the next page prediction doesn't depend on any (zero) of the pages preceding it.
    - Pr(Xi|X1X2…Xi-1)=Pr(Xi)
- The first-order Markov model looks at page-to-page transition probabilities
    - Pr(Xi|X1X2…Xi-1)=Pr(Xi|Xi-1)
- A *K*th-order Markov sequence model considers the conditional probability that a user transitions to an *i*-th page given his or her previous $k = i - 1$ page visits.
    - Pr(Xi|X1X2…Xi-1)=Pr(Xi|Xi-1Xi-2…Xi-k)

For the example shown in Figure 1, the 0-order Markov model has 5 states (i.e., *(p1), (p2), (p3), (p4), (p5)*), while the first-order Markov model has 8 states (i.e., *(p1, p2), (p1, p3), (p2, p1), (p3, p1), (p2, p3),(p3, p4), (p3, p5), (p4, p3)*).

Unfortunately lower-order Markov models (e.g., first- and/or second-order) are not very accurate in predicting future Web page access; since these models do not look far enough into the past to correctly discriminate users' behavioral modes. Thus, good predictions require higher-order models (e.g., third, forth-order).

Higher-order model states are different combinations of the actions observed in the data set, so the number of states tends to rise exponentially as the model order increases. Thus, high-order models are also extremely complex due to their large number of states, which increases their space and run-time requirements [12]. Figure 2 depicts the difference regarding accuracy, coverage, model accuracy and model size with the order of Markov model.

We can see that as the order of the model increases, the accuracy of the model decreases, accompanied by a decrease in the coverage. However, the model accuracy of the model continues to increase. This indicates that though higher-order model can locate states for only a small set of Web sessions, they are able to predict these Web sessions with a greater accuracy than lower-order models do. It is worth noting that as the order of the model increases, the number of states used for the model also increases dramatically.

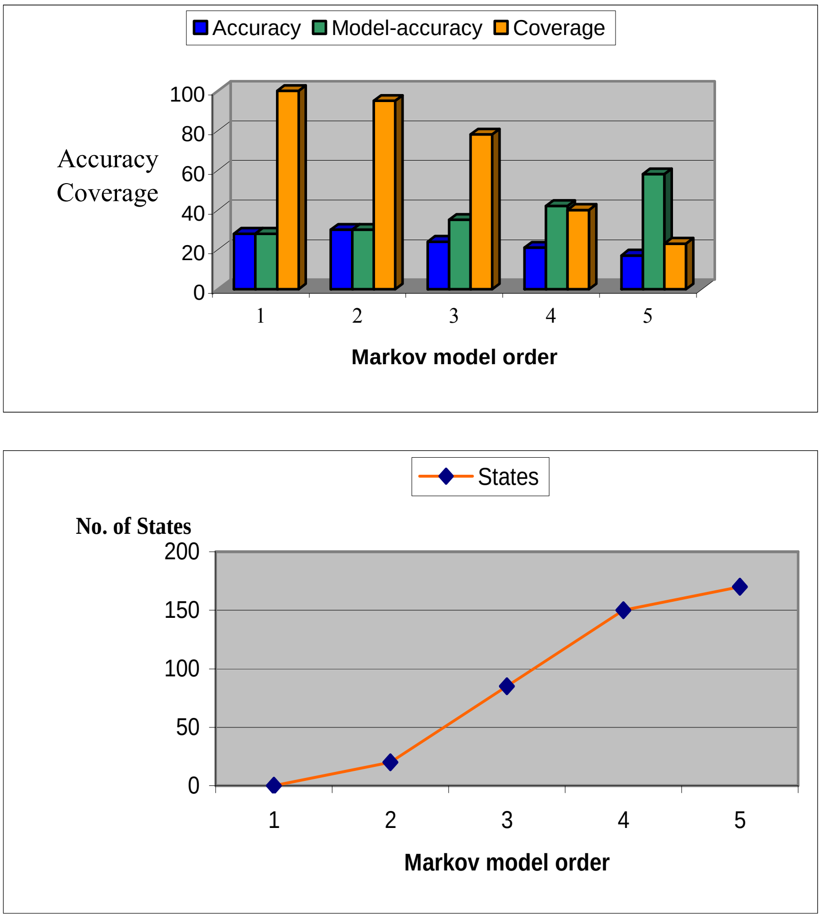

Figure 2
Comparing accuracy, coverage, model accuracy and model size with order of Markov model

One method to overcome the problem of low coverage on the test set is to train varying order Markov models and then combine them for prediction. In this scheme, for each test instance, the highest-order Markov model that covers the instance is used for prediction. This scheme is called the *All-Kth-Order Markov model* [5]. Even though the All-*K*th-Order Markov model solves the problem of low coverage, unfortunately it exacerbates the problem of model size as the states of all the different order Markov models are now part of the model.

3.3 New Markov Models

3.3.1 Hybrid Order Tree-like Markov Model

As mentioned previously, one disadvantage of higher-order Markov models for prediction is that their size grows rapidly with their order. This problem can be solved by using a tree-like Markov model (TMM).

A *K*-order TMM is a multiplayer tree formed from a historical Web access sequence database D_s^{k+1}, where each node represents a visited page, and each branch is the path to the node. A *K*-order TMM is depicted in Figure 3.

Each node records the *count*, which is the number of times the user has visited the node along the same route. The task of a *K*-order TMM is to register all Web access information. Thus, its height is (k + 2), and its width is no more than the number of sequences in the database. In this way, a *K*-order TMM cannot generate a large number of nodes.

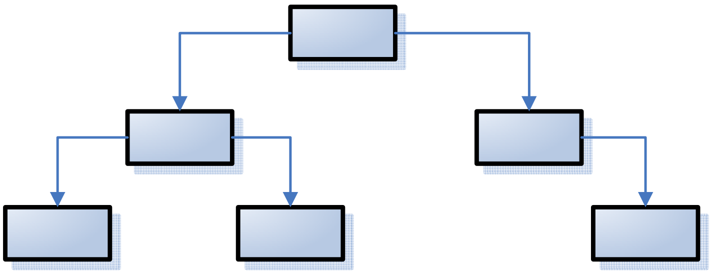

Figure 3
Structure of a tree-like Markov model

To overcome the limitations of higher-order Markov models, varying-order Markov models are trained, and then combined for prediction. Hybrid-order tree-like Markov models (HTMM) offer two methods for combining models for predicting the user's behavior. The first one is the *accuracy voting* [10] method and the second one describes *blending models* [17], [18] method used in data compression.

- **Accuracy Voting Method**

The accuracy voting procedure consists of three steps:

1 Set up 1~*N*-order TMMs from Web logs.

2 Extract 1~*N*-suffixes from the user's current access sequence (that is, 1~*N*-sequence) to the 1~*N*-order TMMs for prediction. Choose the prediction page with the maximum count as the prediction result, and then get the prediction

P1 U
Cou

set $PS = \{P1, P2, \ldots, PN\}$, where Pk $(1 \leq k \leq N)$ is the prediction result by K-order TMM.

3 Compute the prediction parameter of the pages in PS using (1.1). A prediction weight is calculated using (1.2) for each prediction result of different-order TMMs.

$$\text{Prediction}(PageX) = \sum_{i=1}^{N}(S_i \times W_i)\text{, where } PageX \in PS$$

$$S_i = \begin{cases} 1,(P_i = PageX) \\ 0,(P_i \neq PageX) \end{cases} (0 \leq i \leq N)\text{, } P_i(0 \leq i \leq N) \text{ is an element in } PS\text{,} \quad (1.1)$$

$W_i(0 \leq i \leq N)$ is the prediction width in P_i,

$$W_k = \frac{Accuracy_k}{\sum_i Accuracy_i}, \quad (1.2)$$

$W_k(0 \leq k \leq N)$ is the prediction weight of the k-order TMM result, $Accuracy_k(0 \leq k \leq N)$ is the k-order TMM's prediction accuracy.

Finally, pages are sorted by prediction parameters.

- **Blending Model Method**

Blending can be viewed as a bottom-up recursive procedure for computing a *mixture*, where a mixture is basically a weighted average of several probability estimates. Figure 4 presents the accuracy between HTMM and traditional Markov models.

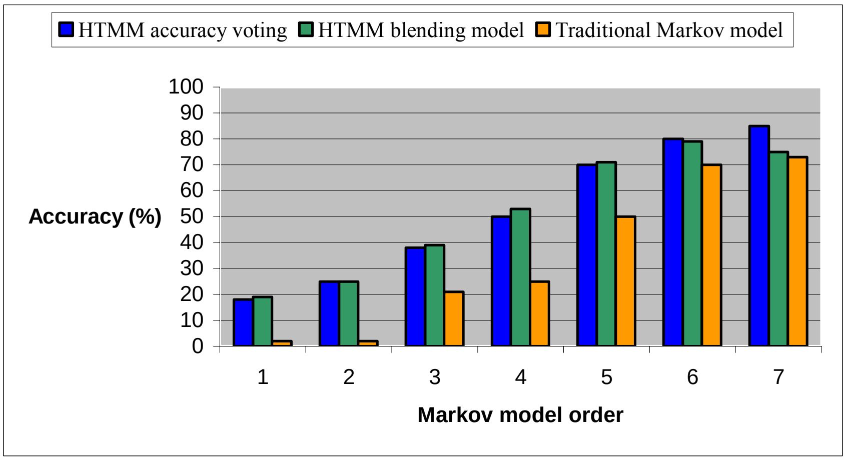

Figure 4
Accuracy comparison between HTMM and traditional Markov models

3.3.2 Selective Markov Models

In the case of selective Markov models (SMM) the starting point is also the All-*K*th-Order Markov model obtained by building a sequence of increasing order Markov models. Instead of using this model for prediction, as we sow this in section 3.2, in this case a number of techniques are used to eliminate certain states across the different order Markov models. The set of states that survive this step, then become the final model that is used for prediction [12]. The goals of this elimination of states is to reduce the state complexity and improve the prediction accuracy of the resulting model.

For an SMM the prediction algorithm is similar to that used by the All-*K*th-Order Markov model. For a given sequence of pages, first we have to identify all the states in the model that can be used to predict the next page, and then use the highest-order state among them to perform the prediction. The key step in the SMM prediction algorithm is the scheme used to determine the potential accuracy of a particular state.

Three different schemes were developed for having an increasing level of complexity. The first scheme, named *frequency-pruned Markov model* (FPMM) simply eliminates the states that have very low occurrence frequency. The second scheme, *confidence-pruned Markov model* (CPMM), uses a technique to identify states, where the conditional probabilities of the two most prominent pages are not significantly different. Thus such states are pruned. Finally, the third scheme, *error-pruned Markov model* (EPMM), uses an estimated-error-based approach to eliminate states with low prediction accuracy. In the rest of this section, we briefly describe these three models.

- **Frequently-Pruned Markov Model**

The main task in FPMM is to observe the states that occur with low frequency in the training set. Thus, those states tend to have also low prediction accuracies.

The amount of eliminating states in the FPMM scheme is verified by the parameter ϕ, referred to as the *frequency threshold*. Using this parameter, FPMM eliminates all the states of the different *k*th-order Markov models for $k > 1$, that occur in fewer than ϕ training set instances [12].

It is important to notice some observations regarding the FPMM.

1 The same frequency threshold is used for all the models regardless of their order.

2 The *pruning* policy is more likely to prune higher-order states since higher-order states occur with lower frequency.

3 The frequency threshold parameter ϕ, specifies the actual number of training set instances that must be supported by each state and not the fraction of

training set instances as is often done in the context of association rule discovery [20].

One of the limitations of the FPMM scheme is that it does not capture all the parameters that influence the accuracy of the state (i.e., the probability distribution of outgoing pages from a state is completely ignored). The next presented model considers not only the occurrence frequency of a state, but also weigh the probability distribution of the pages before making the pruning decisions.

- **Confidence-Pruned Markov Model**

CPMM determines for each state if the probability of the most frequently accessed page is significantly different from the probabilities of the other pages that can be accessed from this state. If the probability differences are not significant, then this state is unlikely to give high accuracy and it is pruned. In contrast, if the probability differences are significant, the state is retained.

If the probability difference between the most probable page and the second most probable page is above a certain threshold - *confidence threshold* ϕ_c **Error! Reference source not found.**- the state is retained. In contrast, if the probability difference between the two most probable pages is below the confidence threshold, the state is pruned.

$$\phi_c = \hat{p} - z_{\alpha/2}\sqrt{\frac{\hat{p}(1-\hat{p})}{n}}\ , \qquad (1.3)$$

where $\hat{p}$ is the probability of the most probable page, where $z_{\alpha/2}$ is the upper $\alpha/2$ percentage point of the standard normal distribution, and *n* is the frequency of the Markov state.

The degree of elimination of states in CPMM is controlled by $z_{\alpha/2}$ (confidence coefficient). As the value of $z_{\alpha/2}$ increases, the size of the confidence threshold ϕ_c decreases, resulting in increased pruning.

It is important to notice that even if the difference in the probabilities between the two most probable pages is relatively small, the state will most likely be retained. That means the in cases in which a state has a large number of outgoing pages, even small preferences toward one of these pages conveys significant information.

- **Error-Pruned Markov Model**

A *validation step* is a widely used approach to estimate the error associated with each state. During the validation step, the entire model is tested using part of the training set that was not used during the model-building phase.

In EPMM the final predictions are computed by using only the states of the model that have the smallest estimated error rate. Inspite of the fact that EPMM also uses the All-*K*th-Order Markov model as its starting point, like FPMM and CPMM, it differs from those two schemes in the way pruning is done. As seen earlier in both FPMM and CPMM schemes, a single parameter (ϕ or ϕ_c) is computed and the whole Markov model is pruned using this parameter. However, in the case of EPMM a higher-order state is pruned by comparing its error rate with the error rate of its lower-order states.

Two error-based pruning strategies were developed. Both of these methods follow a similar approach of pruning but differ in the way the error rate is computed for each state. In the first scheme, referred as *overall error pruning*, every lower-order Markov state has a single error rate value that is computed over the entire validation set. In the second scheme, known as *individual error pruning*, each of the lower-order states has many error rates associated with it, one for each one of its corresponding higher-order states.

Conclusions

In the first part of this paper we have presented traditional Markov models and their limitation in predicting a user's behavior. In the second part we have summarized new Markov models, both based on All-Kth-Order Markov model.

Hybrid-order tree-like Markov model is intelligently merging tree-like structure that aggregates the access sequences by pattern matching and a hybrid-order method that combines varying-order Markov model. This model can predict Web access precisely, providing high coverage and good scalability.

Selective Markov models are obtained by selectively eliminating a large fraction of the states of the All-*K*th-Order Markov model. These state-pruning approaches were motivated by the observation than as the order of the Markov model increases, the limited availability of large training datasets makes it impossible to accurately estimate the conditional probabilities of all the higher-order states.

In our future work we would like to build a cache engine, with the basis on the presented Markov algorithms. Caching in the World Wide Web improves response time.

References

[1] S. Schechter, M. Krishnan, M. D. Smith: Using Path Profiles to Predict HTTP Requests, in Computer Networks and ISDN Systems, Vol. 30, Nos. 1-7, Apr. 1998, pp. 457-467

[2] V. N. Padmanabhan, J. C. Mogul: Using Predictive Prefetching to Improve World Wide Web Latency, in Computer Comm. Rev., Vol. 26, No. 3, July 1996, pp. 22-36

[3] S. Brin, L. Page: The Anatomy of a Large-Scale Hypertextual Web Search Engine, in Proceedings of the 7th International World Wide Web Conference, Elsevier, New York, 1998, pp. 107-117

[4] E. Chi, J. Pitkow, J. Mackinalay, P. Pirolli, R. Gossweiler, S. Card: Visualizing the Evolution of Web Ecologies, in Proceedings of ACM Conference on Human Factors in Computing Systems (CHI 98), ACM Press, New York, 1998, pp. 400-407

[5] E. Chi, J. Pitkow, P. Pirolli: The Scent of a Site: A system for Analyzing and Predicting Information Scent, Usage, and Usability of a Web Site, in Proceedings of ACM Conference on Human Factors in Computing Systems (CHI 2000), 2000, pp. 161-168

[6] P. Pirolli, J. Pitkow: Distribution of Surfers' Paths through the World Wide Web: Empirical Characterization, in World Wide Web, Vol. 2, Nos. 1-2, Jan. 1999, pp. 29-45

[7] A. Papoulis: Probability, Random Variables, and Stochastic Processes, in McGraw Hill, New York, 1991

[8] R. R. Sarukkai: Link Prediction and Path Analysis Using Markov Chains, in Computer Networks, Vol. 33, No. 1, June 2000, pp. 377-386

[9] I. V. Cadez et al.: Visualization of Navigational Patterns on Web Site Using Model-based Clustering, in tech. report MSR-TR-0018, Microsoft Research, Redmond, Wash., 2000

[10] X. Dongshan, S. Junyi: A New Markov Model for Web Access Prediction, in Computing in Science & Engineering, 2002, pp. 34-39

[11] J. Snopek, I. Jelínek: Web Access Predictive Models, in International Conference on Computer Systems and Technologies, 2005

[12] M. Deshpande, G. Karypis, Selective Markov Models for Predicting Web-Page Access, in ACM Transactions on Internet Technology, Vol. 4, No. 2, May 2004, pp. 168-184

[13] S. Balsamo, A. Di Marco, P. Inverardi, M. Simeoni: Model-based Performance Prediction in Software Developement: A Survey, in IEEE Transactions on Software Engineering, May 2004, Vol. 30, No. 5, pp. 295-310

[14] D. Dhyani, W. NG Keong, S. Bhowmick: A Survey of Web Metrics, in ACM Computing Surveys, 2002, Vol. 34, No. 4, pp. 469-503

[15] Zhu, J., Hong, J., Hughes, J. G.: Using Markov Chains for Link Prediction in Adaptive Web Sites. Springer-Verlag Berlin, Heidelberg, 2002

[16] Brian D. Davison. Learning Web Request Patterns. In A. Poulovassilis and M. Levene (eds), Web Dynamics: Adapting to Change in Content, Size, Topology and Use, Springer 2004, pp. 435-460

[17] J. G. Cleary, I. H. Witten: Data Compression Using Adaptive Coding and Partial String Matching, in IEEE Trans. Comm., Vol. 32, No. 4, Apr. 1984, pp. 396-402

[18] J. G. Cleary, W. J. Teshan: Unbounded Length Contexts for PPM, in Computer J., Vol. 40, Nos. 2-3, Mar. 1997, pp. 67-75

[19] S. Bunton: A Generalization and Improvement to PPM's Blending, in tech. report UW-CSE-97-01-10, Dept. of Computer Science and Eng., Univ. of Wash., Seattle, Wash., 1997

[20] R. Agrawal, T. Imielinsky, A. Swami: Mining Association Rules between Sets of Items in Large Databases, in Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data. ACM Press, 207-216