An Approach to Web Page Prediction Using Markov Model and Web Page Ranking.

Article *in* Journal of Convergence Information Technology · December 2009

DOI: 10.4156/jcit.vol4.issue4.10 · Source: DBLP

4 authors:

Ruma Dutta
Netaji Subhash Engineering College
21 PUBLICATIONS **177** CITATIONS
SEE PROFILE

Anirban Kundu
Netaji Subhash Engineering College
144 PUBLICATIONS **693** CITATIONS
SEE PROFILE

Rana Dattagupta
Jadavpur University
55 PUBLICATIONS **219** CITATIONS
SEE PROFILE

Debajyoti Mukhopadhyay
Bennett University
275 PUBLICATIONS **2,231** CITATIONS
SEE PROFILE

Journal of Convergence Information Technology
Volume 4, Number 4, December 2009

An Approach to Web Page Prediction Using Markov Model and Web Page Ranking

Ruma Dutta[1,4], Anirban Kundu[1,4], Rana Dattagupta[2], Debajyoti Mukhopadhyay[3,4]
[1]*Netaji Subhash Engineering College,Garia, Kolkata 700152, India*
{rumadutta2006, anik76in}@gmail.com
[2]*Jadavpur University, Kolkata 700032, India*
rdattagupta@cse.jdvu.ac.in
[3]*Calcutta Business School, Diamond Harbour Road, Bishnupur 743503, India*
debajyoti.mukhopadhyay@gmail.com
[4] *WIDiCoReL, Green Tower, Block C, Flat 9/1, Golf Green, Kolkata 700095, India*
doi: 10.4156/jcit.vol4.issue4.10

Abstract

Markov Models have been widely used for predicting next Web-page from the users' navigational behavior recorded in the Web-log. This usage-based technique can be combined with the structural properties of the Web-pages to achieve better prediction accuracy. This paper proposes one of the pre-fetching techniques relying both on Markov Model and Ranking which considers the structural properties of the Web. In this paper, prediction accuracy is realized as a linear function of transition probability of first order Markov Model and ranking of the Web-page. The chance of the predicted Web-page being the next Web-page would be higher if the prediction accuracy of the Web-page is higher.

Keywords

Web-page Prediction, Markov Model, Rank, Regression

1. Introduction

The rapid expansion of the World Wide Web (WWW) has created an opportunity to disseminate and gather information online. There is an increasing need to study the behavior of Web-user to serve better by reducing the access latency using efficient Web-prediction technique. Many researchers use different techniques which usually employ Markov model of order-k for predicting next Web-page in real time. Navigational behavior of the user is being recorded in the Web-log as an input for Markov model. But the limitation of Markov model is, lower order Markov models are coupled with low accuracy, whereas higher order Markov models are associated with high state-space complexity and also the sequences are not available in Web-log. These motivate us to integrate some other feature to be considered with lower order Markov Model for better prediction accuracy. As the link structure also shows a good promise in the field of Web prediction, we have exploited one of the link structure feature, Web-page ranking [1] to integrate with 1st order Markov Model. Our Web-page Ranking is based on link structure of the Web-pages. This paper involves incorporating Web-page ranking with Markov Model using linear regression and we present a model for Web-page prediction where estimated Prediction accuracy found from linear regression plays an important factor for selecting the next Web-page. By linear regression, this paper intends to establish a relationship between the two datasets of Web-log. This relationship is exploited to decide next Web-page. Moreover, this prediction model is online prediction model.

This paper discusses Related Work in Section 2. Section 3 discusses Our Approach. Section 4 shows the Experimental Result and Section 5 concludes the paper.

2. Related Work

There are several architectures and related algorithms for developing Web predictor. Most of the researchers emphasized on the Markov model and N-grams. An order-k Markov predictor is a scheme which calculates the conditional probability 'p' of accessing Web-page 'P', such that previous accesses were in order of P1,P2....,Pk. More formally, the existing predictor models mostly follow the equation, the probability p=Probability (P|PkPk-1...P1) for computing purpose. Nth-order Markov models, when parameterized by a length of N, essentially represent

the same functional structure as N-grams. These systems analyze the past access history on the Web server, maps the sequential access information in N consecutive cell (known as N-grams) for building prediction models. N-gram methods include two sub-methods: 'point-based' and 'path-based'. 'Point-based' prediction makes the prediction using the last visited URL, which precisely 1st order Markov Model does. In contrast, 'path-based' prediction uses more than one Web-page as the observation in order to make prediction. 'Path-based' prediction basically resembles with higher order Markov model. As it is already discussed, 1st order Markov model or 'point-based' prediction does not make accurate prediction since it neglects the previously visited Web-page information to discriminate the different access patterns. As a result, 'path-based prediction' is more popular. In this area, there is a question on how to choose the best N for N-grams. [2]-[8] discussed several ways to build N-grams and empirically compared their performance on real Web-log data. An empirical study was performed [11] on the tradeoffs between precision and applicability of different N-gram models, showing that the longer N-gram models make predictions accurately with a sacrifice on coverage. There was another approach suggested a way to make predictions based on Kth-order Markov models[7]. Since they prefer longer paths more than shorter ones, their algorithm has the shortcoming that the longer path are more rare in the Web-log history, thus the noise in longer paths could be higher than in shorter path. This may result in reduction in prediction accuracy.

To ease the implementation of higher order Markov model, Markov tree has been used, where for storing Web-log, tree structure has been used. The example of Markov tree for Web-log data ABACD is shown in Figure 1. To accommodate the different orders of Markov model depending upon the applicability, Prediction by Partial Match model (PPM) has been proposed by [9]. The working principle of PPM is as follows. For Web-log data ABACD if the input pattern is AC, then next Web-page is D. If AC is not found in input pattern, only A is found, then the next Web-pages are B and C. So depending upon the input pattern found, the order of the Markov model is changed. This model improves the coverage of input pattern.

In the area of Ranking hyper-linked text, 'PageRank' algorithm is most popular [12]. 'PageRank' algorithm is basically link analysis algorithm which assigns a numerical weightage to each Web-page among a set of hyper-linked documents. The purpose of measuring Rank is to measure relative importance of a Web-page within the set. PageRank was developed by L. Page and S. Brin as part of a research project about a new kind of Search Engine [12].

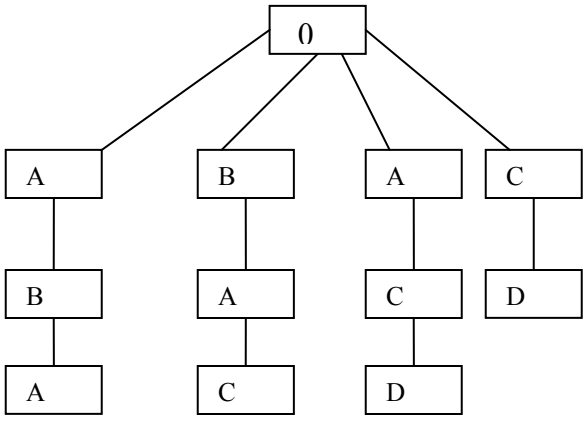

Figure 1: 2nd order Markov Tree for Web-log data ABACD

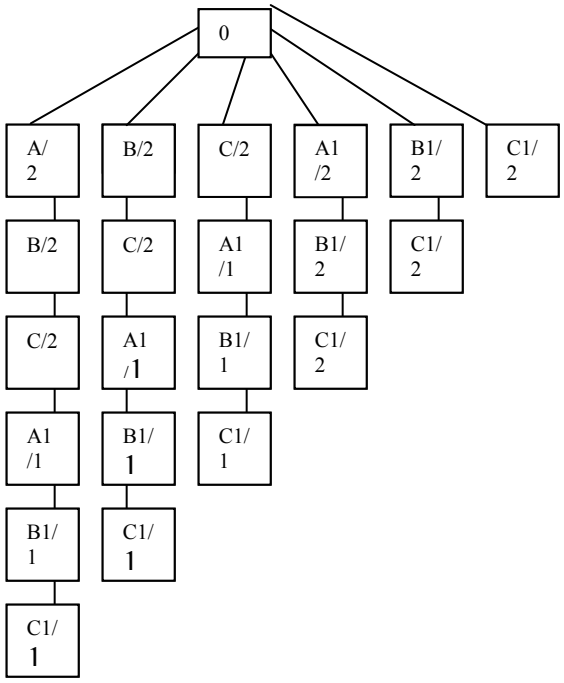

Figure 2: PPM for (ABCA1B1C1) (ABC) (A1B1C1)

3. Proposed Model

This paper proposes an approach for Web-page prediction through linear regression where prediction accuracy depends on the transition probability (refer Definition 2) and also on the ranking of the links of

current Web-page. We realize the model through three phases in offline and using the parameters found in these phases, next Web-page can be predicted.

In Phase I, 1st order Markov model is used to find out the Web-pages for which transition probability is non-zero. Then in Phase II, values of Ranking of these Web-pages are calculated and for those Web-pages which have Ranking beyond threshold value are considered for Phase III. In Phase III, the prediction accuracy is estimated through linear regression and is incorporated in the system as the deciding factor of prefetching the next Web-page.

The process mentioned above is done in both offline and online. The Phase I, Phase II and Phase III are learned to find out regression coefficients and Web-page ranking in offline. In online for Phase I transition probability of 1st order Markov Model is calculated and Ranking of the next Web-pages are found out stored after calculation in offline. Thereafter prediction accuracy is evaluated in the linear model, found in offline. There are two sets of data. First set data is used for Phase I and Phase II and second set data is used for Phase III.

Phase I:

Definition 1: Markov Model - A Markov Model (MM) is a directed graph (V, A) with vertices representing states $V=\{v_1, v_2,\ldots,v_n\}$ and arcs , $A = \{(i,j) \mid v_i,v_j \in V\}$, showing transitions between states. Each arc (i, j) is labeled with a probability p_{ij} of transitioning from v_i to v_j.

Definition 2: Transition Probability for 1st order Markov model - The transition probability is the probability of going to one of the next state (i.e. next Web-page) from current state (i.e., Web-page). For 1st order Markov model, current Web-page decides the next Web-page. The formula for transition probability for 'n' number of Web-pages is given in equation (i)

$$TP_i^j = \frac{AC_i^j}{\sum_{k=1}^{n} AC_k^i} \quad \text{(i)}$$

where , AC= no. of accesses
i = current Web-page
j = next Web-page in the Web log
TP_i^j= Transition Probability from ith Web-page to jth Web-page.

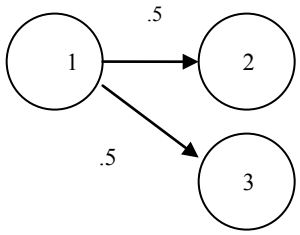

Figure 3: Markov Model

Figure 3 shows the Markov Model of three states. The states are 1, 2, 3. The transition probability from 1 to 2 is .5 and 1to 3 is .5.

The first step is to clean the raw data for a given Web-log. Documents that are not requested directly by user are filtered out. These are image requests in the Web-log that have been retrieved automatically after accessing requests to a document containing the links to the files. These Web-pages are actually noise in Web-log data. Moreover, to achieve better accuracy the Web-pages (current Web-pages for 1st order) which are accessed at least two times in the Web log will be considered for transition probability calculation.

This phase maps all distinct Web-pages in Web-log to states of Markov Model. The output of this phase is the probable next Web-pages with their transition probabilities.

Phase II:

In this phase, the output Web-pages from Phase I are considered. These Web-pages are referred as candidate Web-pages. The Rank is calculated for each candidate Web-page. Motivation for this phase is from the academic literatures in which the importance of the Web-page is evaluated by analysis of its link structure. This gives some approximation of importance or quality of a document.

Figure 4 shows the structures of Web-pages (WP) and Web-site (WS). For example, WP11 has an Out-link to WP12 and an In-link from WP12 in Web-site WS1.

The justification is that a Web-page can have a high Rank if there are many Web-pages pointing to it or if there are some Web-pages that point to it which have a high Web-page Rank. Web-page Rank handles both these cases, by recursively propagating weights through the link structure of the Web. Web-page Rank will be decided for each candidate Web-page, as per the Algorithm 1 (Kundu et al., 2006).

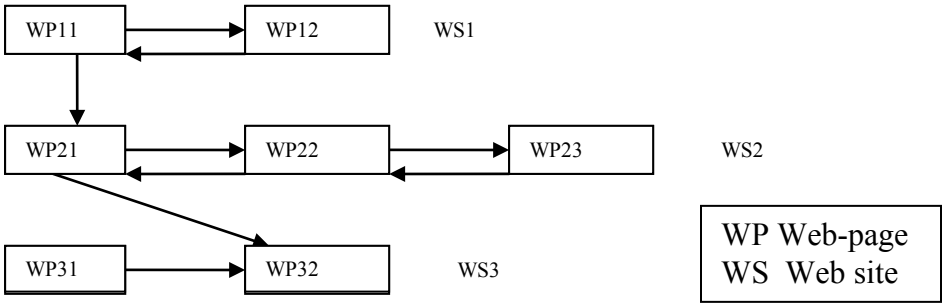

Figure 4. Pictorial View of In-link and Out-link connections of Web-pages

Definition 3: Connection Matrix: The link between Web-pages is stored in the form of a matrix. This matrix is referred in this paper as Connection Matrix (CMat). If a connection is present between two Web-pages, corresponding position in CMat should be filled up with numeric '1' else should be filled up with numeric '0'.

Definition 4: Out-link: The outward link from a Web-page to another Web-page is referred as out-link.

Algorithm 1: Algorithm for finding Web-page Rank.

Input: Number of Web-pages (no_wp)
Output: Web-page Rank of all Web-pages
Step 1 : Set Connection Matrix
Step 2: Calculate out-link of each Web-page
Step 3: Do-loop (start)
Step 4: For i= 1 to no_wp repeat Step 5-7
Step 5: For j= 1 to no_wp repeat Step 6-7
Step 6: sum_of_page_rank:=
sum_of_page_rank+page_rank[i]/outlink[j]

Step 7: page_rank [i] = (1 - 0.85) + 0.85* sum_of_page_rank (where .85 is damping factor found experimentally)
Step 8: Do loop (stop) after settle up the rank of each Web-page
Step 9: Stop.

The Web-pages having Ranking value beyond the threshold is considered for the next phase. The threshold value of Web-page Ranking is defined as follows:

$$ThPR = (MinPR+MaxPR)/2 \quad \text{(ii)}$$

MinPR = Minimum Web-page Ranking of the Web-page among the candidate Web-pages from current Web-page;

MaxPR = Maximum Web-page Ranking of the Web-page among the candidate Web-pages from current Web-page.

Phase III:

In this phase, linear regression is used to integrate Markov model and Web-page Rank. Here erHP is the prediction accuracy which is defined as the accuracy if the candidate Web-page under consideration would have been predicted as next Web-page. P is realized as a linear function of transition probability (TP) and Rank of Web-page (PR). For all of the Web-pages in training data PR and TP are calculated and P is calculated from test data. Thus a,b,c coefficients are found using least square method which is explained later. Once the values for a, b and c are evaluated by solving the three equations, the candidate Web-page with maximum value of P (estimated Prediction accuracy) will be selected for next Web-page. In this way, we integrate two features to get improved prediction accuracy. In on-line, transition probability and Web-page ranking of the candidate Web-pages are considered and using the value of a, b, c in equation (iii) P can be found out . The Web-page having highest value of P is the predicted Web-page.

$$P = a+ b *TP+ c* PR \quad \text{(iii)}$$

where, TP = Transition probability of the candidate Web-page;
PR = Web-page Rank of the candidate Web-page;
a, b, c are coefficients.
Prediction accuracy (P_{actual}) for current Web-page i and next Web-page j in the test data is

$$P_{actual} = \frac{\text{No.of Prediction s where j is the predicted Web - page for i}}{\text{Total no.of Prediction s where i appeared}} \quad \text{(iv)}$$

It has been observed experimentally that typical Prediction Accuracy(P_{acc}) which is shown in equation (v) is high if the model (equation (iii)) is followed.

$$P_{acc}=\frac{\text{No of Correct Prediction}}{\text{Total no of Predictions}} \quad (v)$$

Least Square Method:

For a given data set, (x_1, y_1, z_1), (x_2, y_2, z_2), ..., (x_n,y_n,z_n), where n>=3, the best fitting curve f(x) has the least square error, i.e.,

$$\prod = \sum_{i=1}^{n}[z_i - f(x_i - y_i)]^2 = \sum_{i=1}^{n}[z_i - (a + bx_i + cy_i)]^2 = \text{min} \quad (vi)$$

To obtain the least square error, the unknown coefficients(a, b, and c) must yield zero first derivatives.

$$\frac{\delta \prod}{\delta a} = 2\sum_{i=1}^{n}[z_i - (a + bx_i + cy_i)]^2 = 0 \quad (vii)$$

$$\frac{\delta \prod}{\delta b} = 2\sum x_i[z_i - (a + bx_i + cy_i)]^2 = 0 \quad (viii)$$

$$\frac{\delta \prod}{\delta c} = 2\sum y_i[z_i - (a + bx_i + cy_i)]^2 = 0 \quad (ix)$$

Expanding the above equations, we have

$$\begin{cases} \sum_{i=1}^{n} z_i = a\sum_{i=1}^{n} 1 + b\sum_{i=1}^{n} x_i + c\sum_{i=1}^{n} y_i \\ \sum_{i=1}^{n} x_i z_i = a\sum_{i=1}^{n} x_i + b\sum_{i=1}^{n} x_i^2 + c\sum_{i=1}^{n} x_i y_i \\ \sum_{i=1}^{n} y_i z_i = a\sum_{i=1}^{n} y_i + b\sum_{i=1}^{n} x_i y_i + c\sum_{i=1}^{n} y_i^2 \end{cases}$$

This can be realized by the following matrix

$$\begin{bmatrix} Z \\ XZ \\ YZ \end{bmatrix} = \begin{bmatrix} 1 & X & Y \\ X & X^2 & XY \\ Y & XY & Y^2 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix}$$

where $X=\sum_{i=1}^{n} x_i$ $XZ=\sum_{i=1}^{n} x_i z_i$

$Y=\sum_{i=1}^{n} y_i$ $YZ=\sum_{i=1}^{n} y_i z_i$

$Z=\sum_{i=1}^{n} z_i$ $XY=\sum_{i=1}^{n} x_i y_i$

Solving the matrix, a, b, c parameters are found out as below.

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} Z \\ XZ \\ YZ \end{bmatrix}^{-1} \begin{bmatrix} 1 & X & Y \\ X & X^2 & XY \\ Y & XY & Y^2 \end{bmatrix}$$

The algorithm for predicting the next Web-page is given in algorithm 2 and algorithm 3.

Algorithm 2: Find_regression_coefficient
Input: Web-log of two dataset, Rank of Web-pages of the Web-logs
Output: Regression coefficients

Step 1: loop until EOF of first Web-log
Step 2: Read the next record of Web-log
Step 3: Calculate the transition probability of all candidate Web-pages as in equation (i)
Step 4: Find the rank of all candidate Web-pages
Step 5:Find the prediction accuracy from second Dataset as in (iv)
Step5: end-loop
Step 6: Find out regression coefficients using least square method.
Step 7: Stop

Algorithm 3: Predicting next Web-page
Input: Coefficients of regressions, Candidate

Web-pages W, rank of W
Output: Predicted Web-pages
Step 1: Find out the transition probabilities of W
Step 2: Find out the prediction accuracy by equation (iii)
Step 3: Identify the Web-page with highest prediction accuracy
Step 4: Stop.

4. Experimental Result

Web-log of a server has been taken for the experimentation. There are two sets of data. For prediction accuracy calculation test data set has been referred. Following table (Table 1) shows the sample data of transition probability and prediction accuracy.

Table 1: Sample Data for Transition probability and Prediction accuracy

Current Web-page	Next Web-page	Transition probability	Prediction accuracy
2	3	0.5	0.6
2	5	0.5	0.4
4	10	0.66	0.33
4	5	0.33	0.66
5	7	0.5	0.75
5	6	0.5	0.25

Table 2 shows this relationship between Web-page rank and prediction accuracy P as defined in this paper..

Table 2: Sample Data For Web-page Rank and Prediction Accuracy

Web-page No.	Web-page Rank	Prediction Accuracy
3	0.4	0.5
5	0.4	0.5
10	0.3	0.33
5	0.4	0.66
7	0.4	0.75
6	0..3	0.25

The **correlation coefficient** between Web-page rank and prediction accuracy is **0.851714**. the correlation coefficient between transition probability and prediction accuracy is also 0.8518. So, Web-page rank plays an important role in prediction accuracy as defined in this paper.

Table 3 shows typical prediction accuracy with our data for 1st order Markov Model, 2nd order Markov Model, PPM and our method.

Table 3: Typical Prediction Accuracy

1st Order Markov Model	2nd Order Markov Model	Our Approach	PPM Model
66%	66%	81%	71%

From the data of Table 3, it can be concluded that only history is not sufficient to predict the next Web-page. With introduction of Web-page ranking in deciding the next Web-page we got better result..

5. Conclusion

This paper discussed the fact that Markov model definitely plays a key role in Web-page prediction, but to get better prediction accuracy, some more features have to be considered. Here we have considered the Web-page ranking dependent on link structure. As Web-page rank is involved, this Web-prediction technique is server based. Experimentally it is found that the Web-page rank plays decider role in Web-page prediction.

6. References

[1] Kundu, A., Dutta, R. and Mukhopadhyay, D. (2006)“ An Alternate Way to Rank Hyper-linked Web-pages,” *9th International Conference on Information Technology, ICIT 2006 Proceedings; Bhubaneswar, India; IEEE Computer Society Press, New York, USA,* December 18-21, 2006, pp.297-298.

[2] Bernardo, A., Huberman et al (1998) “Strong Regularities in World Wide Web Surfing,” *Science*, vol. 3, Apr 1998, pp. 95-97.

[3] Davison,, B. D. (2004) “Learning Web Request Patterns,” *Web Dynamics – Adapting to Change in Content, Size. Topology and Use, Springer*, 2004, pp.435-459.

[4] Duchamp, D. (1999) “Prefetching Hyperlinks,” *USENIX Symposium on Internet Technologies and Systems*,1999.

[5] Mukhopadhyay, D., Mishra, P. and Saha, D. (2007) “An Agent Based Method for Web Page Prediction,” *1st KES Symposium on Agent and Multi-Agent Systems – Technologies and Applications, AMSTA 2007 Proceedings, Wroclow, Poland, Lecture Notes in Computer Science, Springer-Verlag,* Germany, May 31-June 1, 2007, pp.219-228.

[6] Mukhopadhyay, D., Dutta, R. Kundu, A. and Kim, Y. (2006) “A Model for Web Page Prediction using Cellular Automata,” *The 6th International Workshop MSPT 2006 Proceedings, Youngil Publication, ISBN 89-8801-90-0,*

ISSN 1975-5635, Republic of Korea, November 20, 2006, pp.95-100.

[7] Pitkow, J.E. and Pirolli, P. (1999) "Mining Longest Repeating Subsequences to Predict World Wide Web Surfing," *USENIX Symposium on Internet Technologies and Systems,* 1999, pp.139-150.

[8] Kroeger, T. M., Long, D.D.E. and Jeffrey C. Mogul (1997) "Exploring the Bounds of Web Latency Reduction from Caching and Prefetching," *USENIX Symposium on Internet Technologies and Systems*,1997.

[9] Palpanas, T. (1966) "Web Prefetching using Partial Match Prediction," *Technical Report CSRG-376, Graduate department of Computer Science, University of Toronto*, 1966.

[10] Xin Chen, X. and Zhang, X. (2003) "A Popularity-Based Prediction Model for Web Prefetching," *IEEE Computer Society*, March, 2003, pp. 63-70.

[11]Z.Su, Z., Yang, Q., Lu, Y. and Zhang, H. (2000) "Whatnext: A Prediction System for Web Requests Using N-gram Sequence Models," *First International Conferences on Web Information Systems and Engineering Conferences, Hong Kong*, June 2000, pp. 200-207.

[12] Page, L. and Brin, S. (1998) "The Anatomy of a Large-Scale Hypertextual Web Search Engine," *7th International World Wide Web Conference*, 1998, Brisbane, ustralia, pp. 107-11.