Expert Systems with Applications 38 (2011) 3422–3430

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

A novel prediction model based on hierarchical characteristic of web site

Chu-Hui Lee *, Yu-lung Lo, Yu-Hsiang Fu

Department of Information Management, Chaoyang University of Technology, 168 Jifong E. Rd., Wufong, Taichung 41349, Taiwan, ROC

ARTICLE INFO

Keywords:
Web usage mining
Prediction
Data preprocessing
Markov model
Bayesian theorem

ABSTRACT

Internet has developed in a rapid way in the recent 10 years,and the information of web site has also been increasing fast. Predicting web user's behavior becomes a crucial issue following the purposes like increasing the user's browsing speed efficiently, decreasing the user's latency as well as possible and reducing the loading of web server. In this paper, we propose an efficient prediction model, two-level prediction model (TLPM), using a novel aspect of natural hierarchical property from web log data. TLPM can decrease the size of candidate set of web pages and increase the speed of predicting with adequate accuracy. The experiment results prove that TLPM can highly enhance the performance of prediction when the number of web pages is increasing.

© 2010 Elsevier Ltd. All rights reserved.

1. Introduction

Web mining technique has been developed and updated for satisfying the requirement of web users. The concept of web mining is applying the techniques of data mining on Internet, web sites and web services, for instance, association rule, clustering algorithm, sequential pattern analysis. Web mining is to extract the knowledge and pattern automatically (Etzioni, 1996). Browsing, consumption and transaction data of users are stored on the web site; therefore, this kind of data environment is quite suitable for applying data mining technique to mine the useful and valuable information from the gold mine source, web log file, improving the quality of service (QoS) of web site (Das & Turkoglu, 2009), decreasing user's latency and increasing customer satisfaction. The goal of web usage mining is to find out the useful information and increase the usability from the web data or web log file for the web applications (Arayaa, Silvab, & Weberc, 2004; Cooley, Mobasher, & Srivastava, 1997; Domènech, Gil, Sahuquillo, & Pont, 2006; Eirinaki, 2004; Kosala & Blockell, 2000; Liu, Chen, & Song, 2002; Mobasher, Cooley, & Srivastava, 2000), such as personalization (Cho & Kim, 2004; Cho, Kim, & Kim, 2002; Mobasher et al., 2000), prediction, pre-fetching and caching, which are efficient ways for increasing the user's browsing speed, decreasing the user's latency and reducing the loading of web server.

Markov model is assumed to be a suitable probability model and usually used for predicting web user's behavior (Chen & Zhang, 2003; Dhyani, Bhowmick, & Ng, 2003; Jespersen, Pedersen, & Thorhauge, 2003; Palpanas, 1998; Sarukkai, 2000). The behavior also can be represented in a sequential pattern where all the included traces are recorded by time order. Jespersens proposed HPG (hypertext probabilistic grammar) model (Jespersen et al., 2003) to predict next web page which will be browsed by user. HPG model consists of the sequence path of web page. Dhyani proposed two models to discuss the usage characterization (Dhyani et al., 2003). First, the random model, Morse's model, is used for modeling the web page accesses. Secondly, the Markov model is assumed to satisfy the Markov ergodic property. It means that the transition probability between any states is used to model the usage of web pages. The high-order-based Markov model is using n-order of Markov model to construct the tree structure, such as PPM (prediction by partial match) (Palpanas, 1998), LSR-PPM (LSR: longest repeating subsequences) (Pitkow & Pirolli, 1999) and PB-PPM (PB: popularity based) (Chen & Zhang, 2003), and to search the trees to find out the results. Unfortunately, the higher the order is, the more complex the tree structure and node number are. In the further research work, LSR-PPM and PB-PPM have improved the above problem. However, it faces the high-computational high-order-based Markov model.

In this paper, we use a novel aspect of natural hierarchical property from web log data. The web pages can be organized and belong to a certain category in the web site. We can view the web pages as a two-level data. First is directory and second is page itself. We propose an efficient prediction model, which is called two-level prediction model (TLPM) (Lee & Fu, 2008). In level one, we predict the following category using Markov model. In level two, the desired web page is predicted by Bayesian model. Using the concept of category in the prediction model, we can find whether it is useful to reduce the time complexity. The experiment results prove that TLPM improves the speed of the prediction with adequate accuracy.

* Corresponding author. Tel.: +886 4 23323000; fax: +886 4 23742337.
E-mail address: chlee@cyut.edu.tw (C.-H. Lee).

0957-4174/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2010.08.128

The rest of this paper is organized as follows: Section 2 describes the related work. The methodology will be presented in Section 3. Section 4 shows the experiment results. The conclusion and future work are discussed in Section 5.

2. Related work

All the user's activities are totally recorded in the web log file; therefore, user's browsing paths can be analyzed through the web usage mining procedure to find out how the web site is accessed by users. Web usage mining procedure (Fig. 1) (Cooley, Mobasher, & Srivastava, 1999; Srivastava, Cooley, Deshpande, & Tan, 2000; Wang & Liu, 2003) includes three main sub-tasks: (1) preprocessing, (2) pattern discovery and (3) pattern analysis. First, preprocessing is to create each user's session and recognize sequential pattern from the web log file. Next, the pattern discovery is to develop the mining algorithms like statistical analysis, association rules, clustering algorithm, classification, sequential pattern and dependency modeling that are used to extract the initial rules or patterns. Finally, the rules or patterns which we are interested in will be found in the pattern analysis.

2.1. Web log file

The log file consists of four kinds of records: access log, error log, referrer log and agent log. The forms of web log file are usually of two types, common log file (CLF) and extended log file (ELF). The common log file includes access log and error log. The common log can be extended to extended log file by appending referrer log and agent log.

2.2. Data preprocessing

Data preprocessing is a most important step in web usage mining and also a complex task after the web log file is obtained. The result of data preprocessing will affect the effects of pattern discovery and pattern analysis. The processes of data preprocessing include five sub-tasks: (1) data cleaning, (2) user identification, (3) user session identification, (4) path completion and (5) transaction identification. ELF is used in step 4 and step 5. In the related research work, Tanasa proposed the advanced data preprocessing processes (Tanasa & Trousse, 2004); Sen et al. discussed the point of view of future trend of web data analysis (Sen, Dacin, & Pattichis, 2006). Tao et al. proposed an integration of web log files and intentional browsing data (IBD), which is a kind of new data source of collection of web user's online data (or usage) and can be used to improve the effectiveness of web usage mining applications (Tao, Hong, & Su, 2008).

2.2.1. Data cleaning

The first-step of data preprocessing is to remove irrelevance content, for example the image files like *.gif and *.jpg by checking the string of record of log file. If the record is created by web spider or web clawer, it will be removed by comparing to the robots.txt. Finally, it should be confirmed that the format of web page depends on the purpose of mining, for instance *.html for the static web page or *.cgi, *.pl, *.asp, *.aspx and *.jsp for the dynamic web page. The format of web page can be checked on the field of http request in log file.

2.2.2. User identification

The step of user identification will be processed after data cleaning. User identification is a very complex sub-task, because the web log file is possibly recorded by a single web server or proxy servers, combined from multiple web servers. In this step, Address or DNS, Authuser, Referrer URL and Usage agent are relevant fields in log file as shown in Table 1. Address or DNS, that is IP address, is in common log file. Referrer URL and Usage agent are in extended log file. For field selection of log file, Inbarani proposed a point of view which is using a rough set to select the appropriate fields for feature selection (Inbarani, Thangavel, & Pethalakshmi, 2007). Users always get an IP address when they connect to Internet. Therefore, IP address or Authuser can be used to identify each user. Referrer URL can be used to distinguish the web user's browsing paths. Usage agent can be used to recognize and judge what the different operation environments are when users use the same IP address or the source of log file is multiple.

2.2.3. User session identification

User's session which is identified by the step of user identification can be built to understand how web users browse. However, setting a threshold, which is an appropriate time interval to

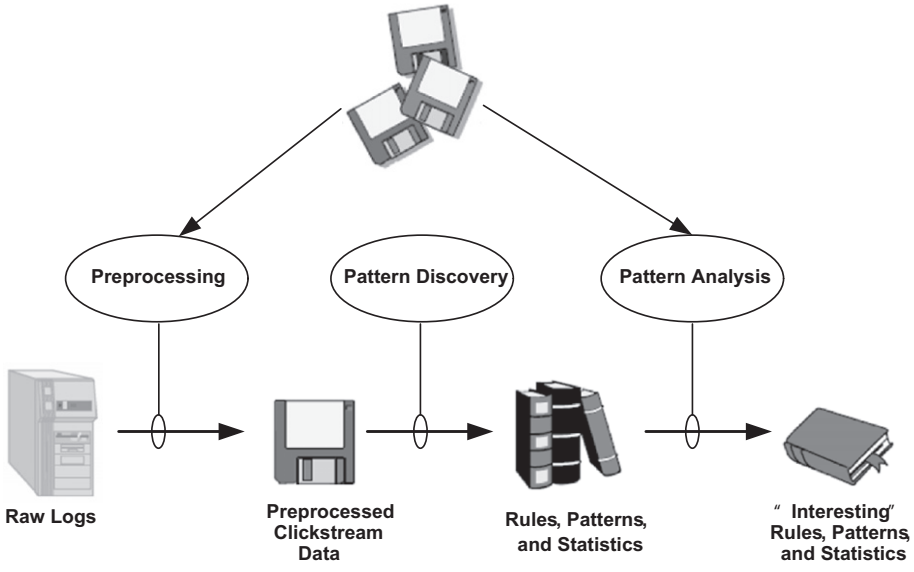

Fig. 1. Web usage mining procedure (Srivastava et al., 2000).

Table 1
Description of common log file.

Content	Description
163.17.9.84	Address or DNS
–	RFC931
–	Authuser
[08/Nov/2007:20:35:52 +0800]	Date
"GET/menu.htm HTTP/1.1"	Http request
200	Status code
1266	Transfer volume
–	*Referrer URL
–	*Usage agent

* This field only appears in the extended log file (ELF).

segment the original web user's session, can get more suitable user's browsing paths. The default threshold of timeout is 30 min (Arayaa et al., 2004; Srivastava et al., 2000; Suresh & Padmajavalli, 2006); most used timeout for research work is 25.5 min (Facca & Lanzi, 2005). For example, assume there is a user's path "ABCDGEF" and the timeout is set by 30 min. The paths "ABCDG" and "EF" can be recognized by the threshold, because the time interval between page A and page E is over 30 min.

2.2.4. Path completion

The appropriate web user's paths can be acquired through the step of user session identification. However, some users' actions of browser, such as "back button", are always hidden in the sessions without recording. This kind of action can be discovered by the field of Referrer URL in ELF with the topology structure of web site to make the user's paths more complete. Following the example in previous section, in the path ABCDG, the field of Referrer URL of page G is B in the EFL, hence the result of complete path will be ABCDCBG.

2.2.5. Transaction identification

All the paths that consist of pages are collected in the user session which is obtained by previous four processed steps. The difference between session and transaction is that session is to collect all the browsed pages in the whole surfing process and transaction can be each page or all the pages in the session, which means transaction can be divided into various small parts from a major one or combined by various small transaction into a big one. Suresh mentioned that the different results of changing session to transaction depends on the method of separation (Suresh & Padmajavalli, 2006), for instance maximal forward reference, reference length and time window. Maximal forward reference is that a transaction will be stopped when the backward action happens. Reference length judges whether the browsing time of a page is enough to be collected into a transaction or not. Time window is using a threshold W to separate a transaction into two when the reading time between previous page and latter page is greater than the threshold.

2.3. Markov model

Markov model is assumed to be a suitable probability model to predict users' browsing behaviors (Dhyani et al., 2003; Dhyani, Ng, & Bhowmick, 2002; Jespersen et al., 2003; Pallis, Angelis, & Vakali, 2007; Sarukkai, 2000). The state space $S = \{s_1, s_2, \ldots, s_k\}$ exists in the Markov model. The sequential pattern can be represented as $\{s_t; t = 1,2,\ldots,n\}$, and s_t denotes the sequence of state at time t. In the state sequence, the transition of any two states is based on the transition probability. The transition probability that is corresponding to the first-step transition probability of Markov model can be denoted by $p_{ij} = \Pr\{s_t = j | s_{t-1} = i\}$ and $p_{ij} = p_{ij}^1$, where p_{ij} denotes the probability of current state j at time t that depends on the previous state i at time $t-1$. The first-step transition matrix P can be created by calculating the transition probability between all states; any two elements in the transition matrix P are independent.

The n-step transition probability is denoted by $p_{ij}^n = \Pr\{s_t = j | s_{t-n} = i\}$, where p_{ij}^n is a probability of the state j at time t that depends on the state i at time $t-n$. The n-step transition matrix is denoted by P^n which can be calculated by Chapman–Kolmogorov equation:

$$p_{ij}^n = \sum_{l=1}^{k} p_{il}^{n-1} \cdot p_{lj} \tag{1}$$

$$p_{ij}^{(m+n)} = \sum_{l=1}^{k} p_{il}^m \cdot p_{lj}^n = P_{ij}^{(m+n)} \tag{2}$$

$$P^{(n)} = P^{(n-1)} \cdot P = P^n \tag{3}$$

If P is a finite transition matrix and P^n is a n-step transition matrix, then $P^{(n)} = P^n$. If we want to establish the second-step or third-step transition matrix which can be calculated by above equations, then $P^{(2)} = P \cdot P = P^2$ or $P^{(3)} = P^2 \cdot P = P^3$.

2.4. Bayesian theorem

Bayesian theorem (Walpole, Myers, Myers, & Ye, 2002, chap. 2) also can be used to predict the most possible users' next request, as well as Markov model. It is assumed that S is the sample space, A and B are two events of the space and $P(B) > 0$. The conditional probability $P(A|B)$ is the probability of event A when the event B occurred. The conditional probability is denoted as follows:

$$P(A|B) = \frac{P(A \cap B)}{P(B)} \tag{4}$$

In the Bayesian theorem, some information is used to revise the prior probability and obtain the posterior probability. The processes are called Bayesian theorem and the equation is as follows:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{5}$$

If the event A is a partition of the sample space S, $A = \{A_1, A_2, \ldots, A_n\}$, then Bayesian theorem can be inferred as follows:

$$P(A_j|B) = \frac{P(B|A_j)P(A_j)}{P(B)} = \frac{P(B|A_j)P(A_j)}{\sum_{i=1}^{n} P(B|A_i)P(A_i)} \tag{6}$$

3. Two-level prediction model

In this paper, we proposed two-level prediction model (TLPM) that is based on the novel aspect of natural hierarchical property of web site. In the TLPM (Fig. 2), in level one, Markov model is used to predict the next possible category at time t based on the user's current state at time $t-1$ and the previous state at time $t-2$. In level two, Bayesian theorem is used to predict the next possible page which belongs to the predicted category. Finally, the prediction result of TLPM can be applied in pre-fetching and caching on web site, personalization, target sales, improving web site design, etc.

In the prediction scope (Fig. 3), TLPM decreases the scope in *top*-r_1 relevance categories in level one, for example, the first 2 relevance categories mean $r_1 = 2$. Bayesian theorem is used to predict the pages which belong to the predicted categories of level one and acquire *top*-r_2 pages, for instance, the first 5 pages mean $r_2 = 5$. In level one, the most possible category will be filtered

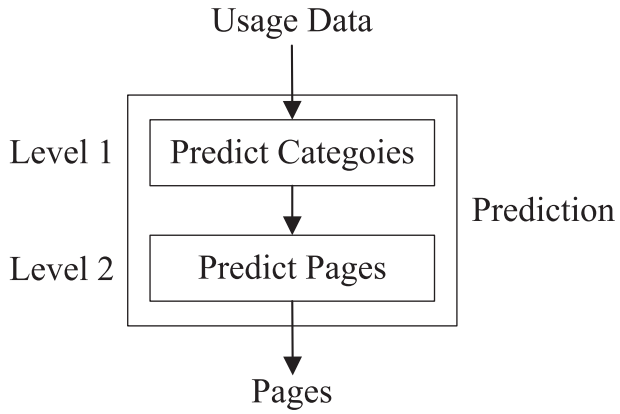

Fig. 2. Two-level prediction model.

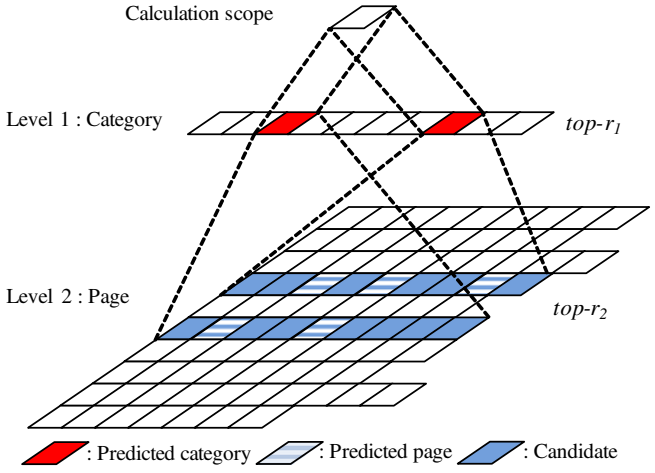

Fig. 3. Calculation scope.

out. In the level two, the predicted pages which belong to the predicted category of level one will be calculated and inferred by Bayesian theorem.

3.1. Pattern representation

In this paper, we notice the natural hierarchical characteristic of web pages from the field of http request of web log file. The hierarchical nature means that pages are stored in the hierarchy directory. The relevant documents are put into the same directory always. Using the directory to classify the web pages is seemed feasible. For example, *199.120.110.21 - - [01/Jul/1995:00:00:09 -0400] "GET /shuttle/missions/sts-73/mission-sts-73.html HTTP/1.0" 200 4085*, the main directory is "shuttle", sub-directories are "missions" and "sts-73" and the page is "mission-sts-73.html". The depth of the page "mission-sts-73.html" is three. The pages in the same directory are assumed in the same category. Therefore, the category for page "mission-sts-73.html" is "shuttle/missions/sts-73".

We can choose the layer of directory as we want to observe in our model. *Layer* is an integer which sets the level of directory to be observed. If the depth of original directory *DP* is smaller than *Layer*, then *Layer* equals *DP*, else *Layer* equals value of setting.

Definition 1

$$Layer = \begin{cases} DP & DP \leqslant Layer \\ Layer & else \end{cases}$$

For instance, because there are three levels of directory over page "mission-sts-73.html", $DP = 3$. If $Layer = 4$, it satisfies $DP \leqslant Layer$, then $Layer = 3$. If $Layer = 2$, then $Layer$ will not be changed.

It is assumed that D denotes the database which contains m user's session, hence the database is $D = \{Session_1, Session_2, \ldots, Session_m\}$. Each user's session can be presented as a sequential pattern of n categories and pages which are browsed by time order. The user's session i means the i th user's usage which is $Session_i = \{(category_1, page_1), (category_2, page_2), \ldots, (category_n, page_n)\}$ where i is the index of user.

3.2. Preprocessing framework

In the preprocessing framework (Fig. 4), step 1 is to create the similarity matrix S of categories by gathering statistics and analyzing the user's browsing behavior from web log file. Step 2 is to create first-step and second-step transition matrix P and $P^2$of Markov model. Step 3 is to create the relevance matrices (R andR^2) which is calculated by multiplying homologous position of first-step matrix P, second-step transition matrix P^2 and similarity matrix S. In this research work, the relevance is a very important impact factor in our prediction model.

3.2.1. Similarity matrix

The step 1 of preprocessing framework is to create the similarity matrix. The similarity matrix helps observe the correlations between categories. First, there are k categories when $Layer$ equals l, hence we can understand the browsing characteristic of category of each user in the $Layer$ l of web log file (Table 2). The $m \times k$ matrix consists of m users' browsing record and k categories. Each column of the matrix can be represented as a column vector, $V_i = \langle C_{1i}, \ldots, C_{hi}, \ldots, C_{mi} \rangle$, that represents how the category i is browsed by all users. In the matrix, $C_{hi} = 1$ means the category i is browsed by the user h, else $C_{hi} = 0$.

The similarity of any two categories can be calculated by set similarity and Euclidean distance as described in Eqs. (7) and (8). Further, the Euclidean distance can be normalized as Eq. (9). The total similarity equation can be acquired by giving weights to Eqs. (7) and (9).

Set similarity:

$$SetSim(V_i, V_j) = \frac{|V_i \cap V_j|}{|V_i \cup V_j|} \tag{7}$$

Euclidean distance:

$$D(V_i, V_j) = \sqrt{\sum_{k=1}^{m} (V_{ki} - V_{kj})^2} \tag{8}$$

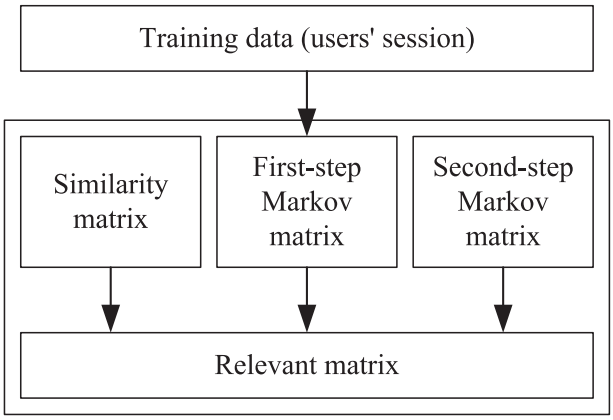

Fig. 4. Preprocessing framework.

Table 2
User session.

	C_1	C_2	$\cdots$	C_k
$Session_1$	1	0	$\vdots$	1
$Session_2$	0	1	$\vdots$	0
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
$Session_m$	1	1	$\cdots$	1

Normalization:

$$ND(V_i, V_j) = 1 - \sqrt{\frac{\sum_{k=1}^{m}(V_{ki} - V_{kj})^2}{m}} \tag{9}$$

Total similarity:

$$S_{ij} = W_{SS} \cdot SetSim(V_i, V_j) + W_D \cdot ND(V_i, V_j) \tag{10}$$

where $W_{SS} + W_D = 1$.

The matrix S denotes the $k \times k$ similarity matrix which can be established by calculating similarity between any two categories. The element in the similarity matrix, for example, S_{ij} is the similarity between category i and j. The matrix S is presented as follows:

$$S = \begin{array}{c} \\ C_1 \\ C_2 \\ \vdots \\ C_k \end{array} \begin{array}{c} \begin{array}{cccc} C_1 & C_2 & \cdots & C_k \end{array} \\ \begin{bmatrix} S_{11} & S_{12} & \cdots & S_{1k} \\ S_{21} & S_{22} & \cdots & S_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ S_{k1} & S_{k2} & \cdots & S_{kk} \end{bmatrix} \end{array} \tag{11}$$

3.2.2. Transition matrix of Markov model

The first-step transition matrix of Markov model P is created by the statistical method, that is to gather statistics and analyze the web log data as well as similarity matrix. The first-step transition matrix is presented as follows:

$$P = \begin{array}{c} \\ C_1 \\ C_2 \\ \vdots \\ C_k \end{array} \begin{array}{c} \begin{array}{cccc} C_1 & C_2 & \cdots & C_k \end{array} \\ \begin{bmatrix} P_{11} & P_{12} & \cdots & P_{1k} \\ P_{21} & P_{22} & \cdots & P_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ P_{k1} & P_{k2} & \cdots & P_{kk} \end{bmatrix} \end{array}$$

where

$$P_{ij} = \frac{Number(i,j)}{\sum_{j=1}^{k} Total\ Number(i,j)} \tag{12}$$

The element in the transition matrix P_{ij} is the transition probability from category i to j. The transition probability can be calculated by Eq. (12), where the numerator is the total number that category i transits to category j and the denominator is total number that category i transits to every category. The transition matrixes are $P^2, \ldots, P^n$, which can be established by the Eq. (3).

3.2.3. Relevance matrix

The relevance matrix is established by multiplying homologous position of similarity matrix and transition matrixes. In general, R^n is created by S and P^n. For example, R is created by S and P and R^2 is established by S and P^2. In this paper, we assume that there is high relevance between categories, which denotes high similarity and transition probability; and the relevance is a very important impact factor of users' behavior. The relevance matrix is denoted as follows:

$$R^n = \begin{array}{c} \\ C_1 \\ C_2 \\ \vdots \\ C_k \end{array} \begin{array}{c} \begin{array}{cccc} C_1 & C_2 & \cdots & C_k \end{array} \\ \begin{bmatrix} R^n_{11} & R^n_{12} & \cdots & R^n_{1k} \\ R^n_{21} & R^n_{22} & \cdots & R^n_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ R^n_{k1} & R^n_{k2} & \cdots & R^n_{kk} \end{bmatrix} \end{array}$$

where

$$R^n_{ij} = S_{ij} \cdot P^n_{ij} \tag{13}$$

The element in the relevance matrix is the relevance between any two categories, for example, R^1_{ij} is the relevance between category i and j by multiplying the homologous position of similarity matrix S and transition matrix P, which means $R^1_{ij} = S_{ij} \cdot P^{n=1}_{ij}$ by using Eq. (13).

3.3. Two-level prediction strategy

TLPM uses the relevance matrix to predict user's next behavior. In the prediction schema, the aim is to predict the unknown position C that depends on the current position B and the previous position A. In this research work, we proposed TLPM to reduce the prediction scope and the number of candidate pages: level one is to predict the category and level two is to predict the page.

3.3.1. Level one: predict category

The purpose of level one is to find out the most possible category set θ of current state C_t that depends on the previous two states C_{t-1} and C_{t-2} (Fig. 5). After the predicted category set θ is acquired, only the pages that belong to set θ will be considered to be the candidates for the prediction of level two. Hence, the prediction scope will be reduced by filtering of level one.

In level one, we use the relevance matrix to filter the categories. $R^n_{C_{t-n}}$ denotes a row vector which is $\langle C_{t-n,1}, C_{t-n,2}, \ldots, C_{t-n,k} \rangle$ of the row C_{t-n} of the relevance matrix R^n. The two row vectors $R^1_{C_{t-1}}$ and $R^2_{C_{t-2}}$ are chosen from the relevance matrixes when $n = 1$ and $n = 2$ to make the predicted category set θ the $top\text{-}r_1$ categories. The set θ is defined as follows.

Definition 2

$$\theta = \left\{ C_t | C_t \text{ is one of the} top - r_1 \text{ categories in the row vector } R^1_{C_{t-1}} \text{ or } R^2_{C_{t-2}} \right\}$$

For example, it is assumed that there are three categories in a web site. The two 3×3 relevance matrices which are R^1 and R^2 (Fig. 6) can be obtained when $n = 1$ and $n = 2$. If $C_{t-1} = C_2$ and $C_{t-2} = C_1$ are known, then the two row vectors $R^1_{C_2} = \langle 0.10, 0.25, 0.15 \rangle$ and $R^2_{C_1} = \langle 0.27, 0.23, 0.12 \rangle$ can be obtained. If we want to get the $top\text{-}r_1 = 2$ categories, then just two categories C_2 (0.25) in R^1 and C_2 (0.27) in R^2 are desired. Hence, the prediction result of level one is $\theta = \{C_1, C_2\}$.

3.3.2. Level two: predict web page

In level two, Bayesian theorem is used to calculate the probability of the successor $page_b$ (Fig. 7) which belongs to the set θ. The predicted page set τ contains the $top\text{-}r_2$ pages.

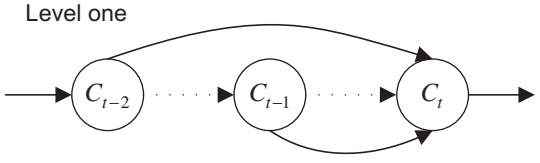

Fig. 5. Category prediction in level one.

$$R^1 = \begin{array}{c} \\ C_1 \\ C_2 \\ C_3 \end{array} \begin{array}{c} \begin{array}{ccc} C_1 & C_2 & C_3 \end{array} \\ \begin{bmatrix} 0.34 & 0.20 & 0.22 \\ 0.10 & 0.25 & 0.15 \\ 0.17 & 0.23 & 0.27 \end{bmatrix} \end{array}$$

$$R^2 = \begin{array}{c} \\ C_1 \\ C_2 \\ C_3 \end{array} \begin{array}{c} \begin{array}{ccc} C_1 & C_2 & C_3 \end{array} \\ \begin{bmatrix} 0.27 & 0.13 & 0.12 \\ 0.11 & 0.18 & 0.10 \\ 0.12 & 0.11 & 0.20 \end{bmatrix} \end{array}$$

Fig. 6. Relevance matrixes R^1 and R^2.

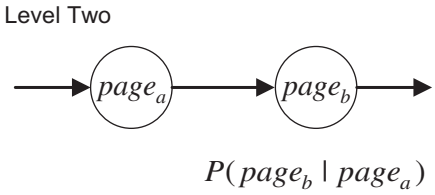

Fig. 7. Page prediction in level two.

In Fig. 7, the probability of $page_b$ is calculated by Bayesian theorem as in Eq. (14), which depends on the current page $page_a$, and the candidate of $page_b$ is limited to the page that just belongs to the set θ.

Bayesian theorem formula:

$$P(page_{b_i}|page_a) = \frac{P(page_a|page_{b_i})P(page_{b_i})}{\sum_{j=1}^{r} P(page_a|page_{b_j})P(page_{b_j})} \quad (14)$$

There are totally r candidate pages. P_θ is a set that includes all the candidate pages. The candidate pages set P_θ and the predicted page set τ are defined as follows.

Definition 3

$P_\theta = \{page_{b_j}|$The category of $page_{b_j}$ belongs to the set θ, where $1 \leqslant j \leqslant r\}$

Definition 4

$\tau = \{page_{b_i}|page_{b_i}$ is the $top - r_2$ pages in the $P_\theta\}$

4. Experiment

In this section, we first introduce the database, results of data preprocessing and setting of the experiment. Next, we discuss the accuracy, execution time, average number of predicated categories and pages. Hit-ratio is used for the accuracy measurement. The execution time is to measure the time cost of prediction. Finally, how many categories and pages are needed when predicted by TLPM are discussed. The experiment environment in this paper is HP Compaq CQ45–101TX laptop, the hardware is Intel Core 2 Duo P7350 2.0G CPU, 4 GB RAM with Microsoft Windows Vista Ultimate SP1 operation system and the software is Java 1.6.0 Update 11 with IDE Gel RC40.

4.1. Web log file, data processing and settings

The experimental databases are chosen from the Internet Traffic Archive web site (http://ita.ee.lbl.gov/), that is two-month web log file of NASA and ClarkNet web site in 1995. In the month selection, July of NASA (NASA_Jul) and September of ClarkNet (ClarkNet_Sep) of the web log files are chosen. The sizes of the chosen data are 195 MB and 164 MB. As the format of databases is CLF, the steps 4 and 5 of data preprocessing processes are skipped. For the rest of the experiment, the record which includes the status code is "OK", the http request is "GET" and the suffix of URL ends with ".html". The results of preprocessing are given in Table 3.

Table 3 shows the information of number of users, session and average length of path after data preprocessing. The NASA_Jul includes 135,920 user sessions and ClarkNet_Sep includes 117,051 user sessions. Table 4 presents the number of categories and pages in each layer. The numbers of pages have some variations in different layers because the same pages may belong to different sub-categories. In addition, the reason for choosing these two databases is to test how the performances of TLPM working on different data environments are. The first is NASA_Jul which includes about 720–730 web pages. The second is ClarkNet_Sep which includes about 5000–6000 pages.

The settings of experimental data are randomly choosing 10,000 sessions, and the length of the session is greater than 5. The chosen data is separated into 80% training data and 20% testing data. The settings of parameter of TLPM are $top\text{-}r_1 = 10$ for predicted category set θ (the first 10 categories) in level one, $top\text{-}r_2 = 10$ for predicted page set τ (the first 10 pages) in level two, and execution time is the average of 50 times of prediction by TLPM.

4.2. Execution time and improved ratio

The result of execution time is recorded in milliseconds (ms) and compared with Bayesian theorem, first-step transition probability of Markov model (Markov_1-step) and TLPM. The prediction result through TLPM gets quite well-improved ratio than that through Bayesian theorem and Markov model. The execution time and improved ratio of NASA_Jul are shown in Tables 5 and 6. The

Table 3
Information of log files.

	NASA_Jul	ClarkNet_Sep
User	68,225	77,340
Session	135,920	117,051
Avg. length	2.91	2.08

Table 4
Categories and pages of log file.

Layer	NASA_Jul		ClarkNet_Sep	
	Categories	Pages	Categories	Pages
1	19	724	37	5075
2	58	725	514	6071
3	208	729	686	5937
4	227	724	773	5992
5	227	723	812	6071

Table 5
Execution time in NASA_Jul case.

Layer	Bayesian	Markov_1-step	TLPM
1	2044.96	1896.44	1817.00
2	2045.32	1894.64	1316.28
3	2060.40	1917.58	891.68
4	2066.96	1913.52	931.78
5	2033.72	1885.30	916.88

Table 6
Improved ratio in NASA_Jul case.

Layer	Bayesian (%)	Markov_1-step (%)
1	11.15	4.19
2	35.64	30.53
3	56.72	53.50
4	54.92	51.31
5	54.92	51.37
Avg.	42.67	38.18

average improved ratio is 42.67% when compared to Bayesian and 38.18% when compared to Markov_1-step, respectively. In Table 6, the improvement in execution time is improved from about 30.53% to 56.72% from layer 2 to layer 5, but only slightly in the layer 1. The highest improved ratio is 56.72% in layer 3 when compared to that using Bayesian theorem.

The execution time and improved ratio of ClarkNet_Sep are shown in Tables 7 and 8. The average improved ratio is 66.66% and 64.40% with Bayesian and Markov_1-step, respectively. The improvement in execution is quite good from about 80.40% to 85.19% from layer 2 to layer 5, but slightly exceeds in Bayesian theorem and is inferior in Markov_1-step in layer 1. In these two cases, we observe that the improved ratio in layer 1 is rather low compared to other layers. It is concluded that just using one layer of directory would not classify pages properly.

4.3. Hit-ratio

The accuracy measurement of the experiment is Hit-ratio as in Eq. (15). Hit_{ratio} is the percentage of the $request_{all}$ which can be successfully predicted. The $cache_{access}$ is the total number of pages which can be found in the predicted page set τ. The $request_{all}$ is the total number of pages which are browsed by the user

$$Hit_{ratio} = \frac{cache_{access}}{request_{all}} \tag{15}$$

As we have mentioned, the purpose of this paper is to increase the prediction speed with adequate prediction accuracy. The experiment results of Hit-ratio compared to Bayesian theorem and first-step transition probability of Markov model are shown in Tables 9 and 10.

This experiment shows that the Hit-ratio of our method is similar to the ratio of other methods. In the result of NASA_Jul (Table 9), the Hit-ratio of TLPM is slightly inferior to Bayesian theorem and first-step of Markov model in layer 1 and layer 2. In layer 3 to layer 5, Hit-ratio is not much reduced when predicted by TLPM. In the result of ClarkNet_Sep (Table 10), the Hit-ratio of TLPM seems to show the same phenomenon.

Table 7
Execution time in ClarkNet_Sep case.

Layer	Bayesian	Markov_1-step	TLPM
1	13815.74	12916.34	13474.74
2	16941.80	15889.00	2508.64
3	16593.58	15598.64	3012.98
4	16718.42	15672.18	2976.10
5	17223.30	16168.08	3169.62

Table 8
Improved ratio in ClarkNet_Sep case.

Layer	Bayesian (%)	Markov_1-step (%)
1	2.47	−4.32
2	85.19	84.21
3	81.84	80.68
4	82.20	81.01
5	81.60	80.40
Avg.	66.66	64.40

Table 9
Hit-ratio in NASA_Jul case.

Layer	Bayesian (%)	Markov_1-step (%)	TLPM (%)
1	68.44	68.39	68.34
2	68.81	68.83	68.35
3	67.90	67.92	63.46
4	68.50	68.50	63.28
5	68.07	68.05	63.02
Avg.	68.35	68.34	65.29

Table 10
Hit-ratio in ClarkNet_Sep case.

Layer	Bayesian (%)	Markov_1-step (%)	TLPM (%)
1	50.80	50.79	50.80
2	50.89	50.91	50.84
3	51.65	51.66	51.52
4	51.85	51.85	51.64
5	51.76	51.79	51.31
Avg.	51.39	51.40	51.22

4.4. Prediction analysis of TLPM

From Sections 4.2 and 4.3, we know that TLPM is efficient in execution time with adequate accuracy for the prediction process. Next, we will discuss the key point that makes TLPM efficient in detail. Table 11 shows the average number of predicted pages P_θ after level one prediction of TLPM. All the categories and pages for each layer are the candidate page in Bayesian theorem and first-step of Markov model as shown in Table 4. In the result of NASA_Jul and ClarkNet_Sep, the average numbers of P_θ are 619.07 pages (85.51% of total pages) and 4844.43 pages (95.46% of total pages) in the set when *Layer* = 1. Hence, each category of TLPM still contains too many pages and is not suitable for prediction. Therefore, the advantage of TLPM is not presented when *Layer* = 1.

The size of candidate page set can be reduced dramatically after layer 2. It proves that the TLPM can eliminate the irrelevance candidate pages and achieve the purpose of reducing the set P_θ through the two-level framework using the natural hierarchical property of web log file; we can observe that the number of pages is decreasing while the layer is increasing. It is concluded that the execution time is reduced while the layer is setting higher.

Table 12 shows the Hit-ratio of level-one prediction. The Hit-ratio of level one is getting lower and lower from layer 1 to layer5 in the case of NASA_Jul. The result is given in Table 9 that shows the final Hit-ratio of our method for case NASA_Jul. On the other hand, in Table 10, it is almost the same accuracy in case

Table 11
Category set analysis of TLPM.

Layer	NASA_Jul $Avg.P_\theta$	ClarkNet_Sep $Avg.P_\theta$
1	619.07	4844.43
2	416.40	448.12
3	105.74	478.64
4	95.33	383.29
5	96.30	395.87

Table 12
Hit-ratio of level-one prediction.

Layer	Level 1 (TLPM)	
	NASA_Jul (%)	ClarkNet_Jul (%)
1	99.97	99.76
2	98.93	82.03
3	79.26	81.47
4	76.79	80.79
5	76.66	80.23

Table 13
Time analysis in NASA_Jul case.

Layer l	TLPM				Total
	Level 1		Level 2		
1	49.96	(2.7%)	1767.04	(97.3%)	1817.00
2	153.04	(11.6%)	1163.24	(88.4%)	1316.28
3	614.18	(68.9%)	277.50	(31.1%)	891.68
4	677.50	(72.7%)	254.28	(27.3%)	931.78
5	665.72	(72.6%)	251.16	(27.4%)	916.88

Table 14
Time analysis in ClarkNet_Sep case.

Layer l	TLPM				Total
	Level 1		Level 2		
1	77.54	(0.6%)	13397.20	(99.4%)	13474.74
2	1362.84	(54.3%)	1145.80	(45.7%)	2508.64
3	1815.72	(60.3%)	1197.26	(39.7%)	3012.98
4	2032.22	(68.3%)	943.88	(31.7%)	2976.10
5	2172.22	(68.5%)	997.40	(31.5%)	3169.62

ClarkNet_Sep, even it equals the compared method in the layer 1 and the reason is the Hit-ratios of level one of TLPM are quite high and all over 80%. It also shows that the accuracy of level one will affect the result of level two of TLPM.

Further, Tables 13 and 14 demonstrate the execution time in each level of TLPM. Hence, we can easily understand the percentage of time TLPM takes for each level. In layer 1 of NASA_Jul case, execution time in level 1 is just 2.7%. It means there are still too many pages in each category and also reveals that the set P_θ is too big; hence, most of the execution time will be consumed in level two of TLPM for calculating the huge amount of candidate pages. In layer 2, the number of categories is getting more and the execution time is also getting longer that is 11.6%. The performance gets better than layer 1 and also gets some improvement in prediction time. From layer 3 to layer 5, more execution time is taken for level one of TLPM which is about 68.9–72.7%, which means TLPM has enough categories to filter out the related categories for predicting. Hence, it just spends less time that is about 27.3–31.1% to calculate the probability of candidate pages in level two of TLPM. This phenomenon is also shown in Table 14.

5. Conclusion and future work

As the information of page of web site is huge and develops rapidly, increasing the user's browsing speed efficiently, decreasing the user's latency as well as possible and reducing the loading of web server become very important issues. In this paper, we use a novel point of view of the natural hierarchical characteristic of web log file and propose a two-level prediction model (TLPM). In level one, we filter the most possible categories which will be browsed by the user. In level two, we predict the pages which belong to the predicted categories of level one to archive the goal of reducing prediction scope more efficiently through the two-level framework. The experiment result proves that TLPM can archive the purpose and improve the efficiency of prediction about 38.18–66.66% by the way of finding out the important category in level one and decreasing the candidate page set in level two. In the future work, we can further improve the efficiency and accuracy of TLPM by modifying or reorganizing the architecture of model and attempt applying different algorithms of data mining, such as clustering algorithm, for the related issues to provide the application of personal prediction for every web user on the web site.

Acknowledgement

The authors would like to thank the reviewers who made many valuable suggestions and comments for us. This work was supported by National Science Council under Grant NSC 99-2221-E-324-042.

References

Arayaa, S., Silvab, M., & Weberc, R. (2004). A methodology for web usage mining and its application to target group identification. *Fuzzy Sets and Systems, 148*, 139–152.
Chen, X., & Zhang, X. (2003). A popularity-based prediction model for web prefetching. *IEEE Computer, 36*(3), 63–70.
Cho, Y. h., & Kim, J. K. (2004). Application of web usage mining and product taxonomy to collaborative recommendations in e-commerce. *Expert Systems with Applications, 26*, 233–246.
Cho, Y. H., Kim, H. K., & Kim, S. H. (2002). A personalized recommender system based on web usage mining and decision tree induction. *Expert Systems with Applications, 23*, 329–342.
Cooley, R., Mobasher, B., & Srivastava, J. (1997). Web mining: information and pattern discovery on the world wide web. In *Proceedings of the 9th IEEE international conference on tools with artificial intelligence* (pp. 558–567).
Cooley, R., Mobasher, B., & Srivastava, J. (1999). Data preparation for mining world wide web browsing patterns. *Journal of Knowledge and Information System, 1*(10), 1–27.
Das, R., & Turkoglu, I. (2009). Creating meaningful data from web logs for improving the impressiveness of a web site by using path analysis method. *Expert Systems with Applications, 36*, 6635–6644.
Dhyani, D., Bhowmick, S., & Ng, W. K. (2003). Modelling and predicting web page accesses using markov processes. In *Proceedings of the 14th IEEE international workshop on database and expert systems applications* (pp. 332–336).
Dhyani, D., Ng, W. K., & Bhowmick, S. (2002). A survey of web metrics. *ACM Computing Surveys, 34*(4), 469–503.
Domènech, J., Gil, J. A., Sahuquillo, J., & Pont, A. (2006). Web prefetching performance metrics: a survey. *Performance Evaluation, 63*, 988–1004.
Eirinaki, M. (2004). *Web mining: A roadmap*. Athens University of Economics and Business, Department of Informatics.
Etzioni, O. (1996). The world-wide web: Quagmire or gold mine? *Communications of the ACM, 39*(11), 65–68.
Facca, F. M., & Lanzi, P. L. (2005). Mining interesting knowledge from weblogs: A survey. *Data and Knowledge Engineering, 53*, 225–241.
Inbarani, H. H., Thangavel, K., & Pethalakshmi, A. (2007). Rough set based feature selection for web usage mining. In *IEEE international conference on intelligence and multimedia applications* (pp. 33–38).
Jespersen, S., Pedersen, T. B., & Thorhauge, J. (2003). Evaluating the Markov assumption for web usage mining. In *Proceedings of the 5th ACM international workshop on web information and data management* (pp. 82–89).
Kosala, R., & Blockell, H. (2000). Web mining research: A survey. *ACM Special Interest Group on Knowledge Discovery and Data Mining Explorations, 2*(1), 1–15.
Lee, C. H., & Fu, Y. H. (2008). Two levels of prediction model for user's browsing behavior. In *IAENG international conference on internet computing and web services* (pp. 751–756).
Liu, L., Chen, J., & Song, H. (2002). The research of web mining. In *Proceedings of the 4th IEEE world congress on intelligent control and automation* (pp. 2333–2337).
Mobasher, B., Cooley, R., & Srivastava, J. (2000). Automatic personalization based on web usage mining. *Communications of the ACM, 43*(8), 142–151.
Pallis, G., Angelis, L., & Vakali, A. (2007). Validation and interpretation of web users' sessions clusters. *Information Processing and Management, 43*, 1348–1367.
Palpanas, T. (1998). *Web prefetching using partial match prediction*. University of Toronto, Department of Computer Science.
Pitkow, J., & Pirolli, P. (1999). Mining longest repeating subsequences to predict world wide web surfing. In *Proceedings of the 2nd conference on USENIX symposium on internet technologies and systems* (pp. 139–150).
Sarukkai, R. R. (2000). Link prediction and path analysis using Markov chains. *Computer Networks, 33*, 377–386.
Sen, A., Dacin, P. A., & Pattichis, C. (2006). Current trends in web data analysis. *Communications of ACM, 49*(11), 85–91.

Srivastava, J., Cooley, R., Deshpande, M., & Tan, P. N. (2000). Web usage mining: Discovery and applications of usage patterns from web data. *ACM Special Interest Group on Knowledge Discovery and Data Mining Explorations, 1*(2), 12–23.

Suresh, R. M., & Padmajavalli, R. (2006). An overview of data preprocessing in data and web usage mining. In *The 1st IEEE international conference on digital information management* (pp. 193–198).

Tanasa, D., & Trousse, B. (2004). Advanced data preprocessing for intersites web usage mining. *IEEE Intelligent Systems, 19*(2), 59–65.

Tao, Y. H., Hong, T. P., & Su, Y. M. (2008). Web usage mining with intentional browsing data. *Expert Systems with Applications, 34*, 1893–1904.

Walpole, R., Myers, R., Myers, S., & Ye, K. (2002). *Probability and statistics for engineers and scientists* (7th ed.). Prentice Hall (pp. 82–87).

Wang, B., & Liu, Z. (2003). Web mining research. In *Proceedings of the 5th IEEE international conference on computational intelligence and multimedia applications* (pp. 84–89).