An Improved Algorithm of Mining Preferred Browsing Paths

Hongbo Li
Institute of Web Intelligence
Chongqing University of Posts and Telecommunications.
Chongqing, China

Ning Wang
Institute of Web Intelligence
Chongqing University of Posts and Telecommunications.
Chongqing, China

Yu Wu
Institute of Web Intelligence
Chongqing University of Posts and Telecommunications.
Chongqing, China

***Abstract*—Existing algorithms of mining preferred browsing paths just consider the influence of user visiting times, but ignore the accuracy influenced by other factors. In order to solve the problem, an improved algorithm which imports page similarity and support-preference concepts is proposed. Firstly a Web-log-based user access matrix is set up. Then by calculating the angel cosine similarity and support-preference, the 2-items preferred browsing sub-path set is obtained. Finally all the sub-paths are combined. Experiments show that the algorithm is more accurate and efficient.**

Keywords-Preferred Browsing Paths; Support-preference; Similarity Matrix

I. INTRODUCTION

With the rapid development of Internet and the increasing number of websites, more and more people get used to obtaining information by visiting the website. Web data mining has become a research focus. As web information is varied, how to dig out useful information for enterprise or individual has become an important issue [1]. Web is a loosely distributed information system, which has unstructured, dynamic and incomplete features [2], whereas Web log has a perfect structure. When a user browsed a website, his/her IP address, access time, the viewed pages and other accessing information are all recorded in the Web server logs [3]. Analyzing Web logs and discovering users' preferred browsing features [4] can help reconstruct the topology of the site and optimize the site structure to provide better personalized service, and can also attract more users to visit the website [5-7].

There are many mature Web log mining technologies and systems, such as Web Miner System, Speed Trace, Web Identity and Access Management (Web-IAM), Technology Acceptance Model (TAM) and Simple Web Log Miner System (SWLMS). Another commonly uses application is Web personalized service. Web personalized service is always provided by the site and is served for different users, and different users always get different view and structure of the site when they access the site. The tasks of a Web personalized service are divided into two aspects, one is to provide users the required information, on the other hand, to provide better access experience for users to make the Web-site more attractive.

The algorithms of Web log mining focus mainly on obtaining users' browsing models. All these algorithms consist of two major steps, including to obtain user's browsing sequences and to discover browsing paths of interest to users. The algorithms can be divided into two categories, which are user frequent accessing path algorithms and user preferred path algorithms. The character of frequent accessing path mining algorithm was converting the browsing history records to browsing sequence. In searching for user frequent accessing paths, maximal forward sequence algorithm (MFSA) [8] was one of the most commonly used algorithms. MFSA first used user turn-back form to browse sub sequence, and then used association rules to find frequent access paths by mining the sub sequence. MFSA obtained user browsing sequences on the basis of the accuracy of user session identification, and it required effective Web log preprocessing, which was a time-consuming task, so that MFSA was restrict. Reference length approach involved sequence pattern according to the time that users spent on each Web page. Apriori algorithm, which was based on association rules, adopted an approach to scan browsing sequences repeatedly and generated a large number of candidate sets to get frequent item sets [9]. FP-Growth algorithm didn't generate candidate sets to discover frequent patterns, but developed a FP-tree storage structure for mining the whole set of frequent patterns [10]. User preferred path algorithms used some rules to obtain 2-items preferred browsing sub-paths first, and then merged the sub-paths to generate preferred path. Xing et al proposed a preferred path mining algorithm called NPPMA [11]. In NPPMA, the concept of support-preference was first mentioned and all users were treated as a whole to discover user preferred paths. The concept of support-preference reflected the ratio of the intensity under the constraint of the specific Web structure.

The mining algorithms mentioned above were mostly based on user's accessing times, and were always not accurate. In order to solve this problem, we propose an improved algorithm of mining preferred browsing paths, which is called User Preferred Browsing Paths Algorithm (UPBPA). On the basis of the concept of support-preference, it uses the included angle cosine similarity formula to calculate the similarity between the pages to get sub-paths, and merges sub-paths to obtain user preferred browsing paths. Theoretical analysis and

This paper is supported by the Natural Science Foundation of Chongqing of China (CSTC, 2012jjA40027)

experimental results both show that UPBPA is more accurate and effective on mining user preferred browsing paths.

II. BASIC CONCEPT

Four concepts for the UPBPA algorithm are defined in this section. They are network topology map of website, page distance, choice-preference concept and support-preference concept.

A. Definition 1 Network Topology Map of Website

A site is organized by different pages that linked to each other in someway.. The topological structure of a site regards every page of the site as a node, link relationship between two pages as a directed edge in directed graph. Thus, define a directed graph $\boldsymbol{G}= (\boldsymbol{V}, \boldsymbol{E})$, where $\boldsymbol{V}$ represents the set of pages in the site, and $\boldsymbol{E}$ represents the set of link relationship between pages, which is shown in Figure 1.

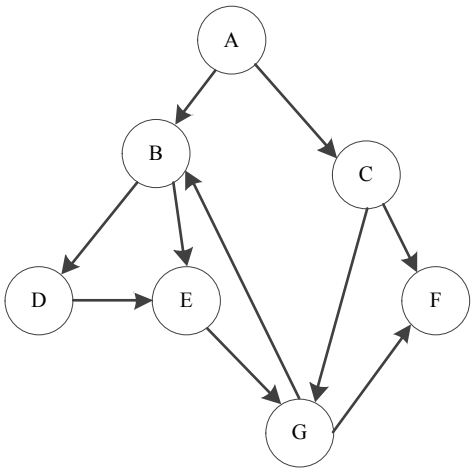

Figure 1. The Network Topology Map of Website

B. Definition 2 Page Distance

Given any two rows i and j from a matrix, the row vectors of which are recorded as $\boldsymbol{X}$, $\boldsymbol{Y}$, and the number of elements of both in row vector is n. X_i and Y_j represent the value of corresponding elements in the two rows respectively. The similarity distance between pages is defined as Page Distance, the computational formula of which is as follows:

$$d_{ij} = \frac{\sum_{i=1, j=i}^{n} (X_i \times Y_j)}{\sqrt{\sum_{i=1}^{n} (X_i)^2 \times \sum_{j=1}^{n} (Y_j)^2}} . \quad (1)$$

Wang et al calculated the similarity between pages by using Hamming distance in [12], the algorithm of which is denoted as Wang [12] below in our paper. In the calculation of Hamming distance, the impact of the specific elements values to the similarity distance didn't be considered. But the visit frequency of page is very important to the access of similar pages. Therefore, in this paper, we use the angle cosine formula between vectors to measure the similarity between pages, which can get similar pages more accurately than Wang [12].

C. Definition 3 Choice-Preference Concept

Suppose users have n different choices after leaving page A and then the average choice degree of each choice is:

$$\overline{S} = (\sum_{i=1}^{n} S_i) / n , \quad (2)$$

where S_i represents the choice-support of the ith choice. And the preference of the kth choice (k=1,2,…,n) can be defined as:

$$P = S_k / \overline{S} . \quad (3)$$

D. Definition 4 Support-Preference Concept

Suppose choice-support and choice-preference of a choice are S and P. Then its support-preference is defined as:

$$P_s = S \times P . \quad (4)$$

III. ALGORITHM OF MINING PREFERRED BROWSING PATHS

In order to optimize the structure of website, we need to discover the user preferred browsing mode. This section demonstrates how to establish user access matrix, calculate the similarity distance and support-preference respectively to obtain 2-items preferred browsing sub-paths, and combine the sub-paths to generate user preferred browsing paths.

A. Matrix Representation of Web Logs

1) Establish User Access Matrix: ECLM log format is commonly used in Web logs. The structure of ECLM log is shown in TABLE I. The attributes of IP, date referrer and agents are very important basic information in Web data mining research.

Source Web log data need to be preprocessed first and the preprocessing mainly consisits of data cleaning, user recognition and session recognition. Data cleaning deletes records which have nothing to do with the research, such as records with jpg, css, js as the suffixes. User recognition helps separate the different access users. Session recognition reconstruct the user access path on the basis of user recognition. In this paper we use heuristic method based on the combination of time and references to recognize the user sessions.

The data after preprocessing stored in the database is expressed as the collection of $\boldsymbol{L}$=<<URL_R, URL>, c>, where <URL_R, URL> represents the session from URL_R to URL, "c" represents the visit times of session from URL_R to URL. Each <URL_R, URL> is a browsing sub-path. Then we can

build User access matrix using the collection of ***L*** in the database to start preferred browsing path mining.

TABLE I. ECLM WEB LOG STRUCTURE

Name	Description
remotehost	IP address of host
authuser	authorized user
date	access time
request	type of request
status	status code
bytes	bytes transmitted
referrer	source page
agent	user agent

User access matrix is shown as formula (5):

$$\boldsymbol{M}_{(n+1)(n+1)}=\begin{array}{c} \\ NULL \\ URL_1 \\ URL_2 \\ \vdots \\ URL_n \end{array}\begin{array}{c} \begin{array}{ccccc} NULL & URL_1 & URL_2 & \cdots & URL_n \end{array} \\ \begin{bmatrix} A_{00} & A_{01} & A_{02} & \cdots & A_{0n} \\ A_{10} & A_{11} & A_{12} & \cdots & A_{1n} \\ A_{20} & A_{21} & A_{22} & \cdots & A_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ A_{n0} & A_{n1} & A_{n2} & \cdots & A_{nn} \end{bmatrix} \end{array} . \quad (5)$$

In the matrix above, the row vector represents URL_R, the column vector represents URL, and the element value represents corresponding value of *c*. Adding a row of NULL and a column of NULL, representing the original URL user stays, because users may visit by inputting URL directly or link from other websites. For there are no URLs visited, we take NULL instead. If there are *n* pages in a website, the corresponding user access matrix is a *n*+1 matrix.

2) Establish Similarity Matrix: The similarity distance is calculate by angel cosine formula which is shown in formula (1),the similar matrix ***M***' is also generated, as shown in formula (6):

$$\boldsymbol{M}'_{(n+1)(n+1)}=\begin{array}{c} \\ NULL \\ URL_1 \\ URL_2 \\ \vdots \\ URL_n \end{array}\begin{array}{c} \begin{array}{ccccc} NULL & URL_1 & URL_2 & \cdots & URL_n \end{array} \\ \begin{bmatrix} d_{00} & d_{01} & d_{02} & \cdots & d_{0n} \\ d_{10} & d_{11} & d_{12} & \cdots & d_{1n} \\ d_{20} & d_{21} & d_{22} & \cdots & d_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ d_{n0} & d_{n1} & d_{n2} & \cdots & d_{nn} \end{bmatrix} \end{array} , \quad (6)$$

where the element value d_{ij} represents the page distance between URL_i and URL_j calculated by formula (1).

B. Mining Preferred Browsing Path

1) Obtain 2-items Preferred Browsing Sub-path: Here we set the value of similarity threshold ∂ and visit the similar matrix ***M***'. If the value of any element in ***M***' is less than or equal to ∂, we'll add the corresponding sub-path <URL_i, URL_j> to 2-items preferred browsing candidate sub-path set.

The candidate sub-path set gotten above considers the similarity of the pages, but ignores the influence of visit times. So we should add new rules to confirm the candidate sub-path set. In this paper we import support-preference concept to confirm.

We use formula (4) to calculate the support-preference value of each element p_{ij} in user access matrix ***M***, and set the support-preference threshold value as ∂'. If any p_{ij} is greater than or equal to ∂', and the corresponding sub-path <URL_i, URL_j> is also in the candidate set, we add <URL_i, URL_j> to 2-items preferred browsing sub-path set.

To get the sub-path set, it is very important to set the value of threshold. In obtaining 2-items preferred browsing candidate sub-path set, if the threshold value we set is too small, it will lose some rules, if too big, it will generate redundancy, while obtaining 2-items preferred browsing sub-path is just the opposite.

2) Generate Preferred Browsing Path: To avoid redundancy and unnecessary omissions, we merge the 2-items preferred browsing sub-paths gradually. The merging process must obey two rules. The first rule is that only the same items of sub-paths can be merged, and the second rule is that the items automatically add 1 after merging complete each time until the sub-paths can't be merged. Finally we put the paths of same items in the same set, and the paths which can't be merged in each set are the preferred browsing paths.

3) Algorithm Description: User preferred browsing path algorithm UPBPA.

Input: user access matrix ***M***, similarity threshold T_a, support-preference threshold T_b

Output: user preferred browsing path set pre_path_set

Step 1. Visit matrix ***M***, calculate the page distance and obtain similarity matrix ***M***'

For *i*=0 **To** *N* **do begin**
  For *j*=0 **To** *N* **do begin**
    Use angle cosine formula to calculate page distance *d*;
    d_{ij}=*d*;
  end
end

Step 2. Visit similarity matrix ***M***', add <URL_i, URL_j> to 2-items preferred browsing sub-path candidate set pre_2_set where the value of <URL_i, URL_j> is less than T_a;

For *i*=0 **To** *N* **do begin**
  For *j*=0 **To** *N* **do begin**
    If d_{ij}<=T_a **Then begin**
      pre_2_set = <URL_i, URL_j>;
    end
  end
end

Step 3. Filter pre_2_set to obtain 2-items preferred browsing paths interest_2_set;

For *i*=0 **To** *N* **do begin**
  For *j*=0 **To** N **do begin**
    Set the support-preference value *t*;
    If *t*>=T_b and <URL_i, URL_j> in pre_2_set **Then begin**
      interest_2_set = <URL_i, URL_j>;
    end

```
    end
end
```
Step 4. Merge 2-items preferred browsing sub-path to obtain 2-items preferred browsing paths;
```
k=2; //k represent the length of sub-path merged presently;
While sub-path can be merged do begin
    For i=1 To the path number in itemset_{k-1} do begin
      l_1= the ith path of itemset_{k-1};
      For j=i To the path number in itemset_{k-1} do begin
        l_2= the jth path of itemset_{k-1};
        If first k-2 paths in l_2=last k-2 paths in l_1 then begin
          merge l_1 and l_2 to itemset_k;
        end
      end
      If l_1can not be merged then begin
        add l_1 to pre_path_set;
      end
      k++;
    end
end
```

C. An Example

In this section, an example is showed to express the mining process. Here we use *A*, *B*, *C*… to represent website URLs. We suppose formula (7) is the user access matrix generated by logs of a website after preprocessing:

$$\boldsymbol{M}_{7\times7}=\begin{array}{c|ccccccc} & NULL & A & B & C & D & E & F \\ NULL & 0 & 10 & 5 & 2 & 0 & 4 & 0 \\ A & 0 & 0 & 5 & 0 & 0 & 2 & 0 \\ B & 1 & 0 & 0 & 5 & 3 & 4 & 0 \\ C & 0 & 2 & 2 & 0 & 3 & 5 & 0 \\ D & 1 & 2 & 0 & 2 & 0 & 0 & 2 \\ E & 0 & 1 & 0 & 0 & 2 & 0 & 2 \\ F & 1 & 0 & 1 & 0 & 0 & 1 & 0 \end{array}. \quad (7)$$

Mining processes are as follows. In the first step, the similar distance and the similar matrix had been obtained, as shown in formula (8):

$$\boldsymbol{M}'_{7\times7}=\begin{array}{c|ccccccc} & NULL & A & B & C & D & E & F \\ NULL & 1 & 0.51 & 0.30 & 0.64 & 0.55 & 0.28 & 0.43 \\ A & 0.51 & 1 & 0.21 & 0.57 & 0 & 0 & 0.75 \\ B & 0.30 & 0.21 & 1 & 0.63 & 0.43 & 0.28 & 0.40 \\ C & 0.64 & 0.57 & 0.63 & 1 & 0.17 & 0.41 & 0.62 \\ D & 0.55 & 0 & 0.43 & 0.17 & 1 & 0.55 & 0.16 \\ E & 0.28 & 0 & 0.28 & 0.41 & 0.55 & 1 & 0 \\ F & 0.43 & 0.75 & 0.40 & 0.62 & 0.16 & 0 & 1 \end{array}. \quad (8)$$

In the second step, we set the value of similarity threshold ∂=0.3 and visit the similar matrix $\boldsymbol{M}'$. If the value of any element in $\boldsymbol{M}'$ is less than or equal to ∂, we add the corresponding sub-path <URL_i, URL_j> to 2-items preferred browsing candidate sub-path set. The 2-items preferred browsing candidate sub-paths set we obtained is: {<NULL, B>, <NULL, E>, <A, B>, <B, NULL>, <B, A>, <B, E>, <C, D>, <D, C>, <D, F>, <E, NULL>, <E, B>, <F,D>}.

In the third step, we set the value of support-preference threshold 3, visit 2-items preferred browsing candidate sub-path set, calculate the support-preference of each candidate sub-path, and choose candidate sub-path whose value of support-preference is more than 3, finally we obtain 2-items preferred browsing sub-paths: {<NULL, B>, <NULL, E>, <A, B>, <B, E>, <C, D>}.

In the last step, 2-items preferred browsing sub-paths is merged to get the final 2-items preferred browsing paths: {<NULL, E>, <NULL, B, E>, <A, B ,E>, <C, D>}.

IV. Experimental Results

A. Experimental Data

In this section, real-world data is used to validate the accuracy and efficiency of our approach. All the experiments are conducted on a machine with Windows7 operating system, Intel Core2 CPU and 2G memory. The development language is JAVA and the IDE environment is Eclipse. Experimental data summarized in TABLE II are gathered from Web logs on a certain website.

TABLE II. Website Log Information

URL Number	Log Date & Time	Original Records Number	Preprocessed Records Number
59	2013/9/6 0:00-2013/9/10 23:59	1477036	22889

B. Evaluation Criteria

All the time, when it comes to evaluate the performance and results of a Web data mining algorithms, experts and scholars usually predict the results by using reasonable mining algorithms, and then compare the mining results with the existing historical data, finally calculate the accuracy of the mining results. However, the known historical and experience data can't be represented and measured accurately and completely when in actual evaluation, that is to say, there isn't an absolute and accurate preferred path. To address this problem, we propose a new evaluation criterion, which can evaluate the mining results about preferred paths objectively and reasonably. We define the concept of precision, which refers to the evaluation of user satisfaction on preferred browsing path by algorithms. Typically, user preferred paths are a collection of the pages that users like to visit frequently, but in fact, each user may be only interested in one or several paths in those preferred browsing paths.

The specific concept of precision is as follows. Suppose that the user preferred sub-path obtained by preferred paths mining algorithms is represented as $\boldsymbol{T}_i$, where *i* represents the number of preferred sub-paths, for example, $\boldsymbol{T}_2$ represents 2-items preferred browsing sub-paths. Define the max length of

preferred sub-path is n, and $L_i = |\boldsymbol{T}_i|$, where L_i represents the number of elements in preferred sub-path $\boldsymbol{T}_i$, for example, if the set of 2-items preferred browsing sub-paths is $\boldsymbol{T}_2$={<A,B>, <A,C>, <C,D>, <D,E>}, L_2 is corresponding to 4. Suppose that the probability that user choosing each preferred sub-path is the same, define the probability of user choosing ith preferred sub-path as P_i=1/L_i. So referring to the above concepts, we define the concept of precision as shown in Equation (9):

$$Q = \frac{\sum_{2}^{n} P_i}{n-1} = \frac{\sum_{2}^{n} (1/L_i)}{n-1}. \quad (9)$$

C. Results and Analysis

In order to verify the precision and efficiency of mining result obtained by UPBPA, we design two groups of experiments to compare UPBPA with Wang [12]. Precision and CPU execution time of the two algorithms are compared in the two groups of experiments. Comparison results are as follows.

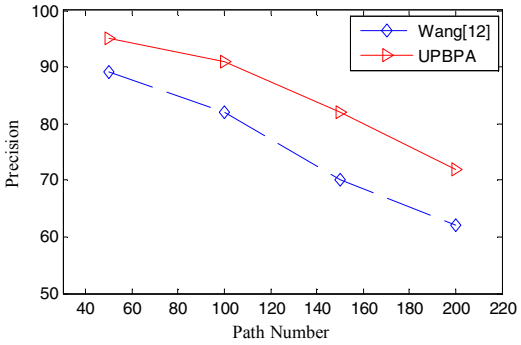

Figure 2. The Comparison of Precision

In the Wang [12], the similarity threshold is calculated by a formula defined, which means the number of 2-items preferred browsing sub-paths is certain, so we set a reasonable similarity threshold to discover the same number of 2-items preferred browsing sub-paths which Wang [12] obtained, and by setting reasonable support-preference threshold extract user preferred browsing paths to compare the two algorithms. As we can see in Figure 2, the precision of UPBPA is higher than that of Wang [12].

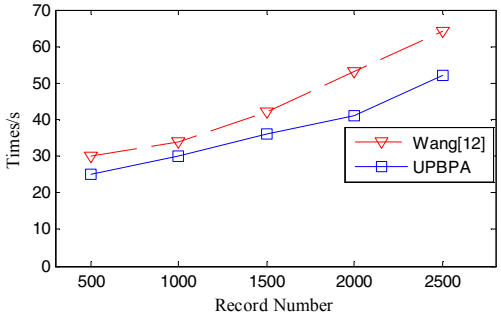

Figure 3. The Comparison of CPU Execution Time

By comparison of execution time, we divide the logs into 5 test cases which respectively include 500 records, 100 records, 1500 records, 2000 records and 2500 records. Then we get Figure 3. It is obvious that UPBPA excels Wang [12] in efficiency, and with the increase of records number, the execution time of the two algorithms all increases, which means records number has a great effect on both of the two algorithms.

V. Conclusions and Future Work

The concepts of angle cosine similarity and support-preference are imported to discover preferred browsing paths in this paper. The algorithm uses user access matrix to get 2-items preferred browsing sub-paths and combine the sub-paths to obtain user preferred browsing path. Experimental results show that the algorithm can behave better than the Wang [12] in both accuracy and efficiency. So the algorithm can be applied to discover the user preferred browsing path of E-commerce websites. In the further research, more attributes such as user browsing time and page length will be considered to extract user preferred browsing paths to make the results more comprehensive and accurate.

References

[1] LIU Zhengtao, WANG Jiandong. Research on Web data space. Computer Engineering and Applications, vol. 48, 2012, pp. 12-18.

[2] WANG Yuanzhuo, JIN Xiaolong, CHENG Xueqi. Network big data: present and future. Chinese Journal of Computers, vol. 36, 2013, pp. 1125-1138.

[3] FEI Hongxiao, QIN Siming, LI Wenxing, LI Qinxiu, DONG Xin. Web user clustering based on Internet. Computer Systems & Applications, vol. 19, 2010, pp. 62-65.

[4] CHEN Xiaoli. Research on personalized learning based on users' behavior. Computer Knowledge and Technology, vol. 5, 2009, pp. 2779-2781.

[5] M.A. Bayir, I.H. Toroslu, M. Demirbas, A. Cosar. Discovering better navigation sequences for the session construction problem. Data & Knowledge Engineering, vol. 73, 2012, pp. 58-72.

[6] LI Zhiyi, YI Meilian. Summary of website optimization based on the user experience. Information Science, vol. 31, 2013, pp. 150-154.

[7] R.J. Kuo, L.M. Lin. Application of a hybrid of genetic algorithm and particle swarm optimization algorithm for order clustering. Decision Support Systems, vol. 49, 2010, pp. 451-462.

[8] Chen M S, Park J S, Yu P S. Data mining for path traversal patterns in a Web environment. Proceedings of the 16th International Conference on Distributed Computing Systems, HongKong, 1996, pp. 385-392.

[9] Yang D L, Yang S H, Hong M C. An efficient web mining for session path patterns. Proceedings of International Computer Symposium 2000 on Software Engineering and Database Systems, Taiwan, 2000, pp. 107-112.

[10] Han J, Pei J. Mining frequent patterns without candidate generation, Proceedings 2000 ACM-SIGMOD International Conference on Management of Data, Dalas, 2000, pp. 1-12.

[11] XING Dongshan, SHEN Junyi, SONG Qinbao. Discovering preferred browsing paths from Web logs Chinese Journal of Computers, vol. 26, 2003, pp. 1518-1523.

[12] WANG Sibao, LI Shengyin. Mining user preferred browsing path based on Web logs. Computer Application and Software, vol. 29, 2012, pp. 164-167.