International Conference on Information Science and Technology
March 26-28, 2011 Nanjing, Jiangsu, China

Improve on Frequent Access Path Algorithm in Web Page Personalized Recommendation Model

Yuhua Chen, Xin Chen and Haoyi Chen

***Abstract*—Web logs record actions and behaviors of users. By mining and analyzing these logs we can find users browsing and access patterns, and this is very important and useful to the web site optimization and recommender. This paper first analyses the association-rules-based personalized recommender model which is very popular in web site recommender system, points out the limitation of the frequent access path algorithm in this model, and then improves it. At last, the paper shows by the test results that the improved algorithm can advance the recommending quality.**

I. Introduction

With the fast development and extensive application of Internet technology, the information on the website has been growing exponentially. The website users are satisfied because it meets their informational needs. However, they are also suffering from such huge amount of information, as well as those problems brought by the World Wide Web due to its distributive, dynamic, massive, heterogeneous, complex and open properties. Thus, a novel technique is badly in need, which could automatically dig out desirable information from the huge pool of resources as quick as a flash, withdrawing it and at the same time filtering out other information unwanted. Fortunately, the emergence of personalized recommendation technology relieves us of infinite data and the commercialized world by saving us plentiful time and energy on searching information. In addition, this new technology also successfully changes the service of website from webpage-centered mode to user-oriented one. It supplies users with personalized services and forges ahead toward the realization of supreme level of the whole Internet services. Many research works have been done in this area such as [1]-[6]. But the recommendation results are not satisfactory. In this case, it is of vital significance to find new ways or to improve existing technology in order to uplift the quality of personalized recommender system.

In this paper, we first outline the association-rule-based personalized recommender model which is very popular in web site recommender system. Then we focus on the frequent access path algorithm used in the recommender model, point out the limitation in this algorithm, and then improve it. At last, the paper shows the test results.

II. General model of association-rule-based personalized recommender system

Association-rule-based personalized recommender model is generally made up of the offline mining module and online recommendation module. System structure is shown in figure 1. Offline mining module is divided into web log data pre-processing stage and web log data mining stage, the main task is to use web log mining algorithm to identify the user's access patterns from the web log data after pre-processing, and update and maintain the user's behavioral patterns database. Online recommendation module provides the user smart and real-time recommendation which user may be interested in the contents of based on the user's current browser behavior, as well as behavioral patterns in database.

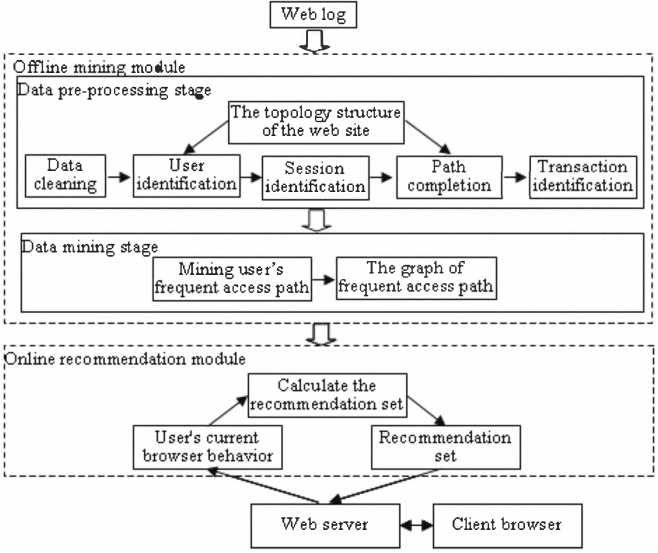
Web log
Offline mining module
Data pre-processing stage
The topology structure of the web site
Data cleaning
User identification
Session identification
Path completion
Transaction identification
Data mining stage
Mining user's frequent access path
The graph of frequent access path
Online recommendation module
Calculate the recommendation set
User's current browser behavior
Recommendation set
Web server
Client browser

Fig. 1. Framework of association-rule-based personalized recommender model

A. Data pre-processing stage

Data cleaning: Delete the irrelevant redundancy with mining in web log data, at the same time, convert useful web log information to an appropriate data format.

User identification: Identify each corresponding user from the log records.

Session identification: A user session is a series of browser requests of a user from entering the site to leaving the site. Different users' visit pages belong to different sessions. Web server logs record each user's visit pages that a user may visit the site many times in a certain period of time, and session identification is to identify the user's visit record into a single session which can respond a user's visit habits.

Path completion: According to server logs and network topology information, educe user's entire access path.

Manuscript received October 28, 2010.

Yuhua Chen is with Computer Science and Technology College, Dalian Maritime University, Dalian, China. phone: 8613614080538; e-mail: yuhuach@ sina.com).

Xin Chen, is with Computer Science and Technology College, Dalian Maritime University, Dalian, China. (e-mail: yuhuach@ sina.com).

Haoyi Chen, is with Computer Science and Technology College, Dalian Maritime University, Dalian, China. (e-mail: yuhuach@ sina.com).

978-1-4244-9442-2/11/$26.00 ©2011 IEEE

Transaction identification: Identify each transaction from the session because transaction is more accurate than session to respond a user's visit habits.

B. Data mining stage

Mining user's frequent access path: First convert the log data to the MFP (Maximal Forward Reference) set and store it in the database, then mine the MFP set by Apriori-based algorithm to get the user's frequent access path. Frequent access path can help us know the user's visit habits.

The graph of frequent access path: Create frequent access path graph from the user's frequent access paths, prepare for calculating recommendation set in the online stage.

III. THE SHORTAGE IN FREQUENT ACCESS PATH ALGORITHM AND THE IMPROVEMENT ON THIS ALGORITHM

Fig.2 as shown is a frequent access path graph after web mining based on the frequent access path algorithm in [2] for one Website.

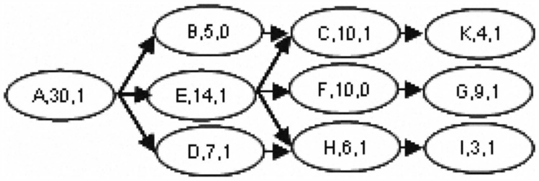

Fig. 2. Frequent access path graph pre-improving

Two limitations were found out in the frequent access path algorithm in [2], which were described as follows.

1) In the generation algorithm of frequent access path graph, the line 20 and 27, respectively, for the same page of the same MFP and the same page of the different MFP that if there were any page the user holding time t_i> = TX, and the properties of the original value of the page is 0, then update this value to 1. That is, a page A in the previous MFP_1 is a navigation page but when it first appears as the content page, we update page A to the content page. Or a page A in MFP_1 is a navigation page, but in MFP_2 is a content page, thus we update A to the content page.

Fig.2, in the item (C, 10, 1), 10 represents the appearance number of C, 1 represents that C is the content page. However, due to C in the MFP may be a navigation page, but as long as C appears in the back of MFP and C is the content page, we will update C to the contents page, that is, if the first 6 times appearance of C are navigation pages, and the 7th is the content page, then we will update C to the content page. Accumulating the total number of C's appearances (including the content pages and navigation pages) is 10.

Since we do not recommend navigation page to the user, the navigation page is not required by the user, so when page C is a navigation page, we do not need to accumulate the weights of vertices (the number of visiting the corresponding web page of vertex), we only need to accumulate the appearance number of that C are content pages. Thus, in the following recommendation process, we can accurately recommend the pages (content pages) which user like to the user.

For example: Fig.2, after the user visiting the page E, we may recommend the user three pages, C, F, H respectively, which are the top 3 pages in user's visit times. Page C's visit times are 10, page G's visit times are 9, page H's are 6. However, in C's 10 visits, the times of that C is a navigation page are 8, and the times of that C is content page are 2. Page K which not be recommended, may be is a content page for all 4 times visits, more than C's times of content pages. So that we may not truly understand the behavior patterns of users, the pages that we recommend may not the user need, so the quality of the recommendation may have a sharp decline. We should recommend page K replace of C to the user, so the number of the weight should not be accumulated on the number of navigation pages.

2) In Fig.2, the user visited the page C, the user may through the page B to page C, or through the page E to page C. But the weight of page C (that is, the number of visiting the corresponding web page of vertex) is the total number which a user visits the page C, it did not distinguish the number between how many times the page is visited from the B and how many times the page is visited from the E.

For example: page A is the home page of a web site, page E is the home page of sports, page B is the home page of blog, page C is a sports star's blog page. Fig.2 shows that a user like to visit page C from the page E and B, most likely the user visit the page C from B more frequent than from the page E, so when the user visits sports home E he will continue to visit the page C is not necessarily greater than the probability of H, that is, if the visit times of page E to page H are 5 , and page E to page C is 3, page F is a navigation page, so that we should first recommend the content of the page H, following the recommendation of the content page C.

The improved frequent access path algorithm is shown as follows.

Input: user's visit transaction set T= {t1,t2,⋯,tn }, minimum support ρ min

Output: the frequent access path graph G

```
1) MFPS←Φ //MFPS holds the maximum forwarding visit path set
2) for(S=l; S<=n; S++)//
3) {Y1 =X1,j=2; i=2; // {X1, X2....,Xm} is the visit web page list of ts, i is the seeking cursor in{ X1, X2....,Xm}, j is the extending cursor for MFP
4) flag= YES; //flag shows the moving direction of the cursor when finding the MFP in ts
5) while (i<=m) {
6) if(Xi==Yk) for some l<=k<=j{//
7) if (flag = =YES){
8) if(list{Yl,Y2, ⋯⋯, Yj-1} not found in MFPS)
9) {{Yl,Y2 , ⋯ ⋯ , Yj-1}→MFPS, set the appearance time as 1 for {Yl,Y2, ⋯⋯, Yj-1}}
10) else
11) {accumulate the visit time for { Yl,Y2, ⋯⋯, Yj-1}} //end if
12) }//end if
13) j=k+1; i=i+1;
14) flag = NO;
15) }//end if
16) else
```

```
17) {Yi =Xi ;j=j+1;i=i+l; flag=YES; }//
18) }//end while
19) if (flag = =YES)
20) {if the { Yl,Y2, ……, Yj-1} is not found in the MFPS, then { Yl,Y2, ……, Yj-1}→MFPS, else accumulate the visit time for { Yl,Y2, ……, Yj-1}}//find the last MFP
21) }//end for
22) for every MFPi∈MFPS{
23) if ((the visit time of MFPi / |T|)> $\rho_{min}$){ //the MFPi is frequent
24) count the visit time for Tm<TX, is the holding time of web page Xm in MFPi //count the time of Xm when Xm is visited as a content page but not a navigation page
25) if (there are vertexes in G for top j web pages in MFPi) {accumulate the visit time when it is a content page for every correspond vertex, set up correspond vertexes in G for bottom m-j web pages, build directed edges, earmark visit time of content pages on correspond vertex}
26) else {set up correspond vertexes for MFPi, build directed edges, earmark visit time of content pages on correspond vertex }
27) }//end if
28) }//end for
```

The modified graph of frequent access path according advanced algorithm is shown in Fig.3.

At this point, the frequent access path graph is a digraph G = (V, W (V), E), which V is the vertex set, that the page URL set; E is directed edges that the hyperlink relations between the two web pages; W (V) is the weights of vertices that the number of visiting the corresponding content web page of vertex.; Get rid of the property item of vertex because at this time in frequent access path graph each vertex's corresponding web page is content page.

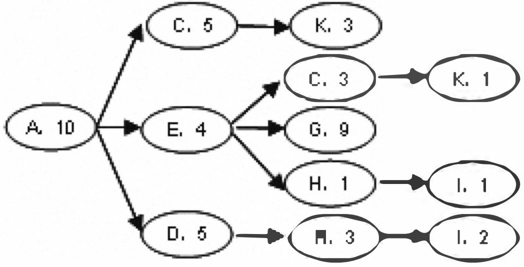

Fig.3. Improved frequent access path graph

IV. Recommendation Quality Analysis

A. Experimental data

In this paper, the experimental data are logs of user accesses to the music machines web site (currently at http://machines.hyperreal.org) from 02/12/97 through 4/30/99 provided by the University of Washington in the United States. During that time, site content did change somewhat, but the basic structure and most of the files remained untouched.

B. Experimental evaluation method

In this paper, the evaluation method is given by Bamshad Mobasher to analyze the quality of its recommendation, including the precision, coverage and F-measure [6].

For each transaction t in the evaluation set, we select the top n pageviews in t as the surrogate for a user's active session window. The active session window is the portion of the user's click stream used by the recommendation engine in order to produce a recommendation set. We call this portion of the transaction t the active session with respect to t, denoted by as_t. Both of the CF-based techniques take as_t and a recommendation threshold τ as inputs and produce a set of pageviews as recommendations. We denote this recommendation set by $R(as_t,\tau)$. Note that $R(as_t,\tau)$ contains all pageviews whose recommendation score is at least τ (in particular, if τ =0, then $R(as_t,\tau)$= P, where P is the set of all pageviews.

The set of pageviews $R(as_t,\tau)$ can now be compared with the remaining portion of t, i.e., with $t-as_t$, to measure the recommendation effectiveness using 3 different metrics, namely, precision, coverage, and the F1 measure.

The precision of $R(as_t,\tau)$ is defined as:

$$precision(R(as_t,\tau)) = \frac{|R(as_t,\tau)\cap(t-as_t)|}{|R(as_t,\tau)|} \qquad (1)$$

The coverage of $R(as_t,\tau)$ is defined as:

$$coverage(R(as_t,\tau)) = \frac{|R(as_t,\tau)\cap(t-as_t)|}{|t-as_t|} \qquad (2)$$

The precision measures the degree to which the recommendation engine produces accurate recommendations (i.e., recommends pageviews that will have be visited by the user in the remaining portion of the user's session). On the other hand, the coverage measures the ability of the recommendation engine to produce all of the pageviews that are likely to be visited by the user. Ideally, one would like high precision and high coverage. A single measure that captures this is the F1 measure, defined in terms of precision and coverage:

$$F1 = \frac{2\times precision\times coverage}{precision + coverage} \qquad (3)$$

The F1 measure attains its maximum value when both precision and coverage are maximized.

C. Experimental plan and condition

In this paper, the experimental data selected from January 9, 1999 to January 12, 1999, four days 134,785 logs for analysis. After Data Cleaning there are 36,011 logs, 7393 users to be identified, 8792 user sessions. Finally, we select 649 user session data which the length more than 10 and less than 40 as a data set, in which random data about 2 / 3 of the record as a training set, used to analyze user access patterns for recommended sequence set, the remaining 1 / 3 records as a test set for the quality of the recommended evaluation.

In this paper, experiments equipment is a PC with P4 2.8GHz CPU, 512MB memory, the program runs in Windows XP, the programming language is C # language.

In the testing, we remove the navigation pages in test data set because that we do not accumulate the number of navigation pages in the modified graph of frequent access path. If the user holding time of a page t <5s, we judge the page is a navigation page. In the experiment, we define minimum support threshold of the association rules is 0.2, The length of association rules front item (viz. sliding window) is 2, the length of association rules consequence is uncertain [7].

D. Experimental results and analysis

The experimental results are shown in Fig .4 to Fig.6.

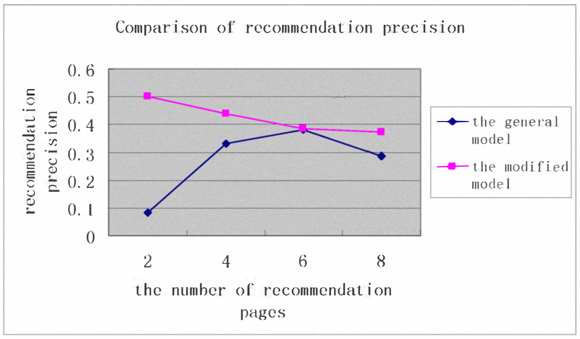

Fig. 4. Comparison of the recommendation precision before and after the improvement

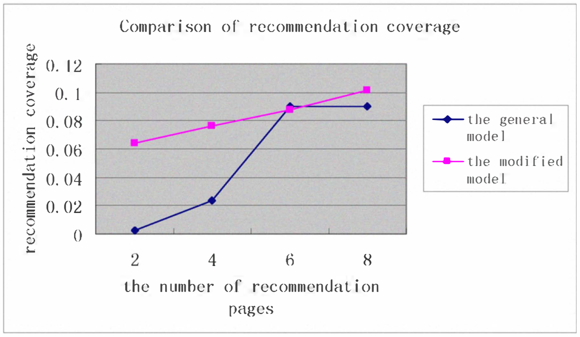

Fig. 5. Comparison of the recommendation coverage before and after the improvement

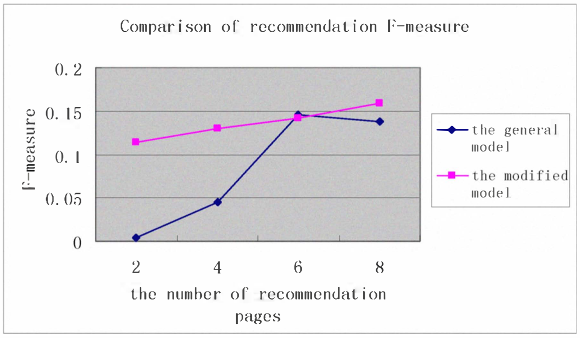

Fig. 6. Comparison of the F-measuring before and after the improvement

It shows in Fig.4 to Fig.6 that the recommendation precision, recommendation coverage and the F-measuring before improvement are obviously higher than after improvement.

V. Conclusion

In this paper we focus on the frequent access path algorithm in the association-rules-based personalized recommender model which is very popular in web site recommender system, point out the shortage in this algorithm, and then improve it. At last, the paper shows by the test results that the improved algorithm can advance the recommending quality.

References

[1] X. Fu, J. Budzik, K. Hammond, “Mining Navigation History for Recommendation,” in *Proceedings of the 5th International Conference on Intelligent User Interfaces*, New Orleans, LA, ACM, 2000, pp. 106-112.

[2] D. B. Dai, J. Yin, “Web Personalized Recommendation Service Based on the Combination of Web Usage Mining and Web Content Mining,” *Computer Engineering and Applications*, vol. 18, pp. 162-165, Jun. 2005.

[3] W. Gaul, L. Schmidt-Thieme, “Recommender systems based on user navigational behavior in the internet,” *Behaviormetrika*, vol.29, pp. 1-22, 2002.

[4] B. Mobasher, “WebPersonalizer: A Server-Side Recommender System Based on Web Usage Mining,” *Technical Report TR99-110*, Department of Computer Science, Depaul University, Chicago, IL, USA, 2001.

[5] B. Mobasher, Dai H, Luo T, et al., “Effective Personalization Based on Association Rule Discover form Web Usage Data,” in *Proceedings of the 3rd ACM Workshop on Web Information and Data Management*, Atlanta, USA, pp. 9-15, 2001.

[6] B. Mobasher, H. Dai, T. Luo, et al., “Improving the Effectiveness of Collaborative Filtering on Anonymous Web Usage Data,” in *Proceedings of ACM Workshop on Web Information and Data Management*, Seattl, 2001, pp. 53-60.

[7] Z. X. Ding, J. Y. Wang, D. L. Wang, et al., “A Web Personalized Recommendation Method on Uncertain Consequent Association Rules,” *Computer Science*, vol. 30, pp. 69-72, Dec.2003.