SRDFA: A Kind of Session Reconstruction DFA

Jinjing Huang, Lei Zhao*, and Jiwen Yang

School of Computer Science and Technology, Soochow University, China 215006
weijintu@163.com, zhaol@suda.edu.cn

Abstract. Session reconstruction is a crucial step in web usage mining. This paper proposes a kind of session reconstruction DFA called SRDFA, which can do session reconstruction for these webpages with or without frame. Moreover, SRDFA can be used to do session reconstruction on those websites which open URLs in new windows. This paper also takes an example to show that sessions reconstructed by SRDFA are more close to users' actual browsing path.

Keywords: web usage mining, session reconstruction, DFA.

1 Introduction

With the rapid development of Internet, more and more people pay attention to web usage mining [1,11], which can discover frequent access patterns based on user's access log. Web usage mining not only can provide personalized service to users, but also be beneficial to the designers to reconstruct the website.

In order to discover the frequent access patterns, the first thing we should do is to preprocess access log to obtain user's sessions which are used for mining association rules [2]. The precision of sessions will affect the acquisition of frequent itemsets [8,12] directly, so how to get the accurate sessions from users' log is the chief problem. Furthermore, the structure of website becomes more and more complex: on the one hand, the technology of dynamic website design is popular and the designers would like to make web applications based on B/S frame. At the same time, a large number of webpages with frame appear in web applications. On the other hand, in some websites whose URLs can be opened in new windows so that users can browse website in asynchronous parallel mode. As a result, the traditional methods of session reconstructions are no longer feasible to this new situation. Based on the theory of DFA, this paper proposes one kind of session reconstruction DFA called SRDFA, which can respectively be suitable for webpages with frame or not; in addition, it can be seen that SRDFA is also available to do session reconstruction for these websites which open URLs in new windows.

* Corresponding author.

J. Cao et al. (Eds.): NPC 2008, LNCS 5245, pp. 50–60, 2008.

2 Related Work

The object of web usage mining is user's access log with the format of CLF or ECLF [3]. In the stage of preprocessing, there are mainly two kinds of methods for session reconstruction: time-oriented heuristics and navigation-oriented heuristics. Paper [4,5,9] introduce the two heuristics.

In time-oriented heuristics, session data is reconstructed by analyzing the session duration time or the time between consecutive web page requests (page stay time) [4]. Session duration time represents the total time of one session is limited with a threshold of δ (usually δ=30mins) and page stay time means the time spent on any page is limited with a threshold of δ (usually δ=10mins)[4].

Supposed that there are a series of page requests $p_1, p_2, \ldots, p_k$, the access time of which are $t_1, t_2, \ldots, t_k$ respectively. According to the first time-oriented heuristics, if $t_k - t_1 \leq 30mins$, these pages can constitute one session. However, based on the second time-oriented heuristics, if pages $p_1, p_2, \ldots, p_k$ form a session, then time spent on each page is less than 10 minutes ($t_{i+1} - t_i \leq 10mins$).

In navigation-oriented heuristics [4,9], pages of one session can access each other through direct or indirect hyperlink. If $p_1, p_2, \ldots, p_k$ have already been constituted a session, p_{k+1} can be joined into this session if there exist a hyperlink from p_i to $p_{k+1}(i \in [1, k])$. If many pages contain such hyperlink, p_i is the nearest one to p_k.

After session reconstruction, sessions can not be used for mining association rules directly, for the reason that the log data sometimes isn't intact. In other words, such pages generated by clicking "back" button in the previous page are not recorded in service log, because they have already been stored in the local cache. For instance, page p_1 has hyperlinks toward page p_2 and p_3 , user A first clicks this hyperlink to reach page p_2 , then he comes back to p_1 by clicking "back" button, and then go to page p_3 from p_1. Obviously, the real access path of user A is $p_1 \rightarrow p_2 \rightarrow p_1 \rightarrow p_3$, however, the access path in log file is $p_1 \rightarrow p_2 \rightarrow p_3$. Some papers such as [4] and [10] study this problem and paper [10] proposes the method of path supplement.

Reference [6] and [7] have already done some research on session reconstruction and path supplement. In the two papers, DFA theory is applied to do session reconstruction, however, this DFA isn't suitable for webpage with frame. Thus this paper proposes a kind of session reconstruction DFA called SRDFA, which is different from that one in reference [6]. SRDFA is not only suitable for these webpages with frame, but also is applicable to these websites which open URLs in new windows.

3 SRDFA

Considering the flexibility of web design, if we only use time-oriental or navigation-oriental heuristics to reconstruct sessions, maybe a real session is separated into several ones. Moreover, today many websites open some URLs in

new windows, which results in that traditional algorithm of path supplement can not reveal the actual access path. Based on the fact this paper proposes a kind of DFA to do session reconstruction called SRDFA, which can automatically accomplish session reconstruction for a section of users' access log.

3.1 Webpages without Frame

For the webpages without frame, we can use the DFA to reconstruct sessions as figure 1 depicts. Paper [6] has introduced this DFA in details.

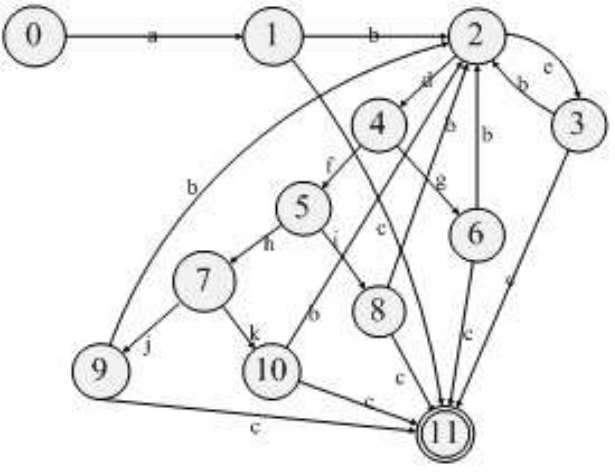

Fig. 1. The DFA used in webpages without frame

One DFA is composed of five elements defined as $M = (S, \sum, f, S_0, Z)$. In figure 1, $S = \{1, 2, \ldots, 11\}, \sum = \{a, b, \ldots, k\}, S_0 = 0, Z = 11$, and the meaning of each state and character has been explained in paper [6].

3.2 Webpages with Frame

Nowadays, due to the adoption of new technology, the contents of access log have become more complex. For example, Webpage with frame is a kind of special webpage, which divides the browser window into several regions as table 1 shows. And each region is filled with different page.

The webpage shown in table 1 is made up of three regions, which in fact are different pages. In this case, the contents of catalogue and banners framework usually display statically while the main framework changes its contents after users click different hyperlinks. Generally speaking, page B usually is welcome page, so it can be cleaned from the log. In catalogue framework, many hyperlinks about different subjects are provided for users.

Table 1. The structure of webpage with frame

Banners framework(B)	
Catalogue framework(C)	Main framework(M)

Before session reconstruction we should extract some fields such as HostID, Date, URL (the address of page request), Referrer (the refer page of the current page), Agent from the cleaned log. However, for webpages with frame (such as struts framework website based on MVC mode), URL should be understood in a broad sense instead of just as address.

Take a website based on MVC mode as an example. User submits a request to web application through a table or a URL. After receiving this request, the controller begins to search for the corresponding action. If no such action, the controller will send the response to JSP or static page (html/xml) directly; but if such action exists, then the field of URL is filled with a string of characters ended with .do, which can be viewed as a generalized URL. Table 2 shows the log data of a dynamic framework website based on MVC mode.

Table 2. Data of a dynamic framework website based on MVC mode

HostId	Date	URL	Referrer
A	[19/Mar/2008:15:40:23 +0800]	a.jsp	-
A	[19/Mar/2008:15:40:25 +0800]	b.do	a.jsp
A	[19/Mar/2008:15:40:35 +0800]	c.do	b.do
A	[19/Mar/2008:15:41:38 +0800]	d.do	a.jsp
A	[19/Mar/2008:15:42:56 +0800]	d.do	d.do
A	[19/Mar/2008:15:43:28 +0800]	d.do	d.do
A	[19/Mar/2008:15:45:12 +0800]	d.do	d.do

In some dynamic web applications, there is a fact that after clicking the different hyperlinks in one page, the address of URL displaying in the page is same although we get different contents. In order to find out which hyperlink user clicked, we have to resort to the access log. In some website different actions are generated from different hyperlinks, we just need to confirm the clicked action which is recorded in the field of URL. For example, supposed that page A has 3 hyperlinks B, C, D, and they correspond to 3 actions $b.do, c.do, d.do$ respectively. If user clicks hyperlink B, then the field of URL will be recorded as $b.do$. Therefore, we can easily identify which hyperlink was clicked. Let's review the table 2, we can find the user traversed on the same action that is $d.do$ from record 5 to 8. The reasons for this case are as follows: firstly, several hyperlinks in one page

may share the same action; secondly, some URL in dynamic website contains parameters which are removed in the step of log cleaning. For example, suppose URL is "d.do?BH=076001&BM=JZGGNJXXX" which includes two parameters BH and BM. After the step of log cleaning, the URL converts into "$d.do$". Regardless of which situation, $d.do$ should be considered as an important action. So the designer would pay more attention to page $a.jsp$ which generates action $d.do$, and it is better to provide a short cut for user to access $a.jsp$ conveniently.

3.3 Path Supplement Based on Multi Window

First of all, the data structure of record in session is defined as table 3 shows.

Table 3. The data structure of record

IP	user
date	the access time of page request
url	url of page request (generalized url)
refer	the refer page of current page
new_window	whether opened from new window

Suppose page p_1 has two hyperlinks towards page p_2 and p_3 respectively which are opened in new windows. It means that page p_1 is not close when page p_2 or p_3 are opened. Besides, there is a hyperlink toward page p_5 in page p_2.Then suppose the session is p_1, p_2, p_5, p_3 before doing path supplement. According to the traditional path supplement algorithm, p_2 and p_1 should be inserted after p_5 for the reason that $p_3.refer \neq p_5$.url. Therefore, the new session is p_1, p_2, p_5, p_2, p_1, p_3. However, because of the particularity of this website, while p_5 is open, p_1 isn't close. So user can click p_3 in the page p_1 directly after visiting p_5, rather than coming back to page p_1 by clicking "back" button in page p_5. Obviously, the real session is p_1, p_2, p_5, p_1, p_3.

As describing in section 3.2, when the main framework changes its content, the catalogue framework is not variable, so users can easily change their interests to watch pages about different subjects by clicking hyperlinks in the catalogue framework, rather than coming to the main page through clicking the "back" button.

The algorithm of path supplement based on multi window [7] as follows:

Suppose p and q are any two consecutive records in a session, and $q.refer = s.url \neq p.url$.

(1)If $s.url$ isn't in the current session, put the current session into database and regard record q as the first record of a new session, then go to (6).

(2)Suppose the next record of s is t and judge the section of records from s to p is consecutive or not. If it is, then go to (3); Otherwise to find the discontinuous record r, if $r = s$, then insert q after r, go to (6); if $r \neq s$, go to (4).

(3)From t to p, there is one record whose URL is opened in a new window or not. If there isn't such record, then go to (5). Otherwise, suppose this record is

x and the record in front of x is y. If there are several records are opened in new windows, x is the one closest to s. If $y = s$, insert s and q after p, go to (6); If $y \neq s$, insert the section of records from y to s after p, then insert q, go to (6).

(4) From record t to r, there is one record whose URL is opened in a new window or not. If there isn't such record, then go to (5). Otherwise, suppose the record is x and the record in front of x is y. If there are several records are opened in new windows, x is the one closest to s. If $y = s$, insert s and q after r, go to (6); If $y \neq s$, insert the section of records from y to s after r, then insert q, go to (6).

(5)Do path supplement with the traditional method ("back" mode).

(6)Path supplement finishes.

3.4 Session Reconstruction DFA

Because of the flexibility of web applications, such DFA introduced in section 3.1 can not be used in the webpages with frame. Thus we design a new DFA called SRDFA, which adds several states to the original DFA. On the first aspect, SRDFA contains some states for webpages with frame. On the second aspect, SRDFA does the path supplement based on two modes which are "back" mode and "multi window" mode.

Figure 2 is SRDFA which can automatically finish session reconstruction for a section of users' access log.

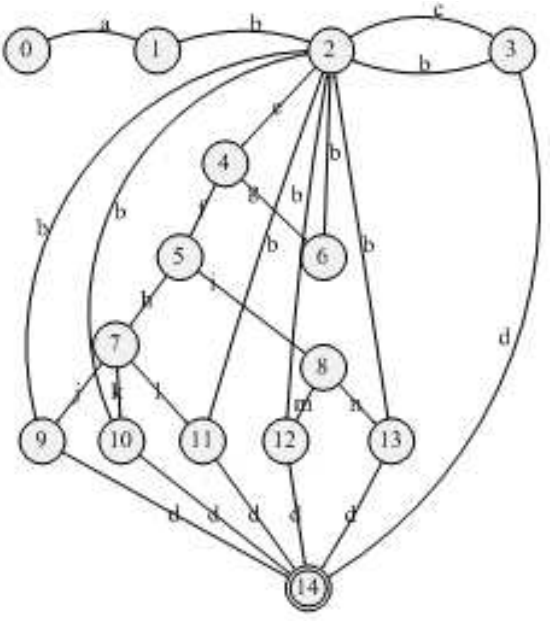

Fig. 2. SRDFA

Suppose p and q are two consecutive records of log, the meaning of each state in SRDFA as follows:

State 0: the beginning state, start to accept the first record of access log;

State 1: accept the next record;

State 2: judge the two consecutive records belong to the same user or not;

State 3: current session terminates;

State 4: judge the time interval of two consecutive is greater than10 minutes or not;

State 5: judge whether $q.refer = p.url$ or not;

State 6: the same as 3;

State 7: judge which method of path supplement should be used;

State 8: judge whether $q.refer = q.url$ or not;

State 9: do path supplement by the "back" mode;

State 10: do path supplement by "multi window" mode;

State 11: the same as 3;

State 12: accept the next record until its URL isn't equal to $p.url$, and terminate the current session;

State 13: insert the current record into the current session;

State 14: the terminal state, session reconstruction finishes.

The meaning of each character as follows:

a: the first record p;

b: the next record q;

c: $p.IP \neq q.IP$ (the consecutive pages belong to different users);

d: there is no next record in the log;

e: $p.IP = q.IP$ (the consecutive pages belong to the same user);

f: $q.date - p.date \leq 10mins$ (the time interval between p and q is less than 10 minutes);

g: $q.date - p.date > 10mins$ (the time interval between p and q is greater than 10 minutes);

h:$q.refer \neq p.url$ (p is not the refer page of q);

i: $q.refer = p.url$ (p is the refer page of q);

j: records from $q.refer$ to p are consecutive and their URLs are not opened in new windows;

k: records from $q.refer$ to p aren't consecutive or they are consecutive but there exist one record whose URL is opened in a new window;

l: $q.refer$ isn't in current session;

m: $q.url = q.refer$;

n: $q.url \neq q.refer$.

4 Experimental Results

Table 4 is a section of access log from a dynamic struts framework website based on MVC mode. In table 4, URLs are replaced by letters and Record-Id represents the sequence of these records.

In this website, $a.jsp$ plays the role of catalogue framework. URLs opened from catalogue page are all opened in new windows according to the section 3.3. Thus, $b.jsp$ and $f.jsp$ are view as opened in new windows from $a.jsp$ in this log.

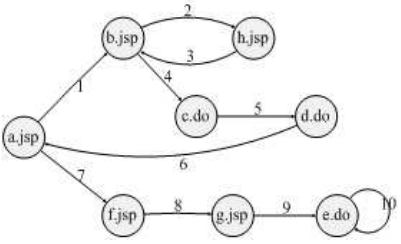

Fig. 3. The actual path of 192.168.151.79

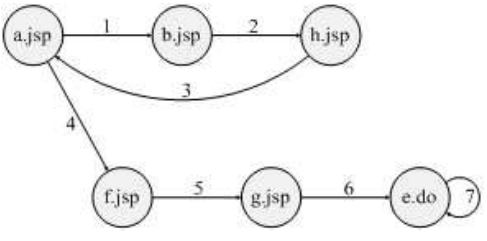

Fig. 4. The actual path of 192.168.151.65

In table 4, there are two different users 192.168.151.79 and 192.168.151.65. Figure 3 and figure 4 are the actual access paths of these two users respectively and numbers above the arrow represent the sequence of browsing the website.

For this section of log, if we just use the first time-oriented heuristics which has been described in section 2, we can obtain sessions like this :$\{a.jsp, b.jsp, a.jsp, b.jsp, h.jsp, c.do, d.do, f.jsp, g.jsp, e.do, e.do, e.do\}$ and$\{a.jsp, b.jsp, h.jsp, f.jsp, g.jsp, e.do, e.do\}$. In addition, if we adopt the second time-oriented heuristics, sessions $\{a.jsp, b.jsp\}$, $\{a.jsp, b.jsp, h.jsp, c.do, d.do, f.jsp, g.jsp, e.do, e.do, e.do\}$, $\{a.jsp, b.jsp, h.jsp, f.jsp, g.jsp, e.do, e.do\}$ can be obtained.

After that, path supplement should be done. Take $\{a.jsp, b.jsp, h.jsp, f.jsp, g.jsp, e.do, e.do\}$ as an example, if we use the traditional method just considering "back" mode, then $\{a.jsp, b.jsp, h.jsp, b.jsp, a.jsp, f.jsp, g.jsp, e.do, e.do\}$ can be obtained.

Obviously, the results can not really reflect the actual access path, which means previous methods aren't suitable for this framework website. Thus, we

Table 4. Data of a dynamic framework website based on MVC mode

Record-Id	HostId	Data	URL	Referrer
1	192.168.151.79	[19/Mar/2008:17:01:23 +0800]	a.jsp	main.jsp
2	192.168.151.79	[19/Mar/2008:17:01:36 +0800]	b.jsp	a.jsp
3	192.168.151.79	[19/Mar/2008:17:12:56 +0800]	a.jsp	main.jsp
4	192.168.151.79	[19/Mar/2008:17:13:26 +0800]	b.jsp	a.jsp
5	192.168.151.79	[19/Mar/2008:17:15:18 +0800]	h.jsp	b.jsp
6	192.168.151.79	[19/Mar/2008:17:15:37 +0800]	c.do	b.jsp
7	192.168.151.79	[19/Mar/2008:17:16:18 +0800]	d.do	c.do
8	192.168.151.79	[19/Mar/2008:17:20:02 +0800]	f.jsp	a.jsp
9	192.168.151.79	[19/Mar/2008:17:20:16 +0800]	g.jsp	f.jsp
10	192.168.151.79	[19/Mar/2008:17:22:16 +0800]	e.do	g.jsp
11	192.168.151.79	[19/Mar/2008:17:22:38 +0800]	e.do	e.do
12	192.168.151.79	[19/Mar/2008:17:23:53 +0800]	e.do	e.do
13	192.168.151.65	[19/Mar/2008:18:15:12 +0800]	a.jsp	main.jsp
14	192.168.151.65	[19/Mar/2008:18:15:26 +0800]	b.jsp	a.jsp
15	192.168.151.65	[19/Mar/2008:18:16:03 +0800]	h.jsp	b.jsp
16	192.168.151.65	[19/Mar/2008:18:17:23 +0800]	f.jsp	a.jsp
17	192.168.151.65	[19/Mar/2008:18:17:42 +0800]	g.jsp	f.jsp
18	192.168.151.65	[19/Mar/2008:18:18:16 +0800]	e.do	g.jsp
19	192.168.151.65	[19/Mar/2008:18:18:23 +0800]	e.do	e.do

adopt SRDFA to do session reconstruction. Suppose p and q are any two consecutive records of the log and the simple process of session reconstruction by SRDFA is as follows:

(1)Accept the first record denoted as p on the state 0, then come to state 1 to accept the next record denoted as q; On the state 2, it can be found that p and q belong to the same user (192.168.151.79), thus come to state 4; The time interval between p and q is less than 10mins, so come to state 5; After that, come to state 8 because $p.url = q.refer(a.jsp)$; On the state 8, we can judge that $q.refer(a.jsp) \neq q.url(b.jsp)$, so come to state 13; Insert record q into the current session on this state, then continue to accept the next record. At this moment, p and q are the second and third record respectively.

(2)Accept the third record by the aforementioned method. On the state 4, we can find the time interval of p and q is greater than 10mins, so come to state 6; On this state, the current session $\{a.jsp, b.jsp\}$ is terminated and is put into the database, then the next record is accepted. Reset p and q, thus p is the third record while q is the fourth one.

(3)Accept $b.jsp, h.jsp$ by the same method, the current session is $\{a.jsp, b.jsp, h.jsp\}$, then continue to accept the next URL ($c.do$). Based on the theory of section 3.2, $c.do$ is viewed as a generalized URL. On the state 5, it can be found that $q.refer \neq q.url(b.jsp \neq h.jsp)$, so come to state 7; On the state, the SRDFA jumps to state 9 and does path supplement by "back" mode, so the current session is $\{a.jsp, b.jsp, h.jsp, b.jsp, c.do\}$.

(4)Then put the next record into current session based on the method described above, the current session is $\{a.jsp, b.jsp, h.jsp, b.jsp, c.do, d.do\}$. Then accept the next record whose URL is $f.jsp$, and come to state 7 from state 5 as $q.refer \neq q.url(a.jsp \neq e.do)$; b.jsp is opened in new window, so do path supplement by "multi window" mode on state 10. The current session is $\{a.jsp, b.jsp, h.jsp, b.jsp, c.do, d.do, a.jsp, f.jsp\}$.

(5)Continue to accept next two records $g.jsp$ and $e.do$, and the current session is $\{a.jsp, b.jsp, h.jsp, b.jsp, c.do, d.do, a.jsp, f.jsp, g.jsp, e.do\}$. Accept next URL $(e.do)$, then $q.url = q.refer(e.do = e.do)$ is found on the state 8, so come to state 12, and record $e.do$ doesn't need repeat join in the session. Continue to accept next record until its URL is not e.do or is null. Then put current session $\{a.jsp, b.jsp, h.jsp, b.jsp, c.do, d.do, a.jsp, f.jsp, g.jsp, e.do\}$ into the database.

(6)The rest records accepted by the aforementioned method and the paper dosen't explain in details. The SRDFA accepts records until the next URL is null, then comes to the terminal state (state 14). There are three sessions in the database $\{a.jsp, b.jsp\}$, $\{a.jsp, b.jsp, h.jsp, b.jsp, c.do, d.do, a.jsp, f.jsp, g.jsp, e.do\}$ and $\{a.jsp, b.jsp, h.jsp, a.jsp, f.jsp, g.jsp, e.do\}$.

According to figure 3 and figure 4, it is obviously found that sessions reconstructed by SRDFA more close to the actual paths of users, which provide more accurate data for mining association rules. Besides, SRDFA can be used in webpages with or without frame and applied to the website which open URLs in new windows, so SRDFA has more advantages than the traditional methods of session reconstruction.

5 Conclusion

The user's web log becomes more and more complex, which brings new challenge and opportunity to web usage mining. This paper proposes a kind of session reconstruction DFA called SRDFA. On the one hand, it can be applied to webpages with frame or not respectively; On the other hand, it is also suitable for these websites which open URLs in new windows. The experiment results show sessions reconstructed by SRDFA more close to users' actual access paths. The future work is that we can optimize the SRDFA with addition of states for transaction reconstruction, which means we can obtain transactions used for mining association rules directly by the DFA.

6 The References Section

References

1. Srivastava, J., Cooley, R., Desphande, M., Tan, P.: Web Usage Mining, Discovery and Applications of usage patterns from web data. SIGKDD Explorations 1(2), 12–23 (2000)
2. Ye, Y., Chiang, C.-C.: A Parallel Apriori Algorithm for Frequent Itemsets Mining. In: Proceedings of the Fourth international Conference on Software Engineering Research, Management and Applications. IEEE, Los Alamitos (2006)

3. Luotnen A.: The Common Log File Format, http://www.w3.org/Daermon/User/Config/Logging.html
4. Bayir, M.A., Toroslu, I.H., Cosar, A.: A New Approach for Reactive Web Usage Data Processing. In: Proceedings of the 22nd International Conference on Data Engineering Workshops. IEEE, Los Alamitos (2006)
5. Berendt, B., Mobasher, B., Spiliopoulou, M., Nakagawa, M.: A Framework for the Evaluation of Session Reconstruction Heuristics in Web Usage Analysis. INFORMS Journal of Computing, Special Issue on Mining Web-Based Data for E-Business Applications 15(2) (2003)
6. Jinjing, H., Lei, Z., Jiwen, Y.: Web Sessions Reconstruction based on DFA. Computer Engineering and Applications (accepted) (chinese)
7. Jinjing, H., Lei, Z., Jiwen, Y.: Path Supplement in Session Reconstruction based on multi window. Computer Applications and Software (accepted)(chinese)
8. Ye, Y., Chiang, C.-C.: A Parallel Apriori Algorithm for Frequent Itemsets Mining. In: Proceedings of the Fourth international Conference on Software Engineering Research. IEEE, Los Alamitos (2006)
9. Cooley, R., Mobasher, B., Srivastava, J.: Data Preparation for Mining World Wide Web Browsing Patterns. Knowledge and Information Systems 1(1) (1999)
10. Liehu, L., Haipeng, Z., Yafeng, Z.: Data preprocessing Method Research for Web Log Mining. Computer Technology and Development 17(7), 45–48 (2007) (chinese)
11. Cooley, R., Mobasher, B., Srivastava, J.: Web mining: Information and pattern discovery on the world wide web. In: Proceedings of the 9th IEEE Internatioal Conference on Tools with Artificial Intelligence (ICTAI 1997), Newposrt Beach, CA (1997)
12. El-Sayed, M., Ruiz, C., Rundensteiner, E.A.: FS-Miner: Efficient and Incremental Mining of Frequent Sequence Patterns in Web logs. In: WIDM 2004, Washington,USA (2004)