

Behaviour Recovery and Complicated Pattern Definition in Web Usage Mining

Long Wang and Christoph Meinel

Computer Department, Trier University,
54286 Trier, Germany
{wang, meinel}@ti.uni-trier.de

Abstract. Data mining includes four steps: data preparation, pattern mining, and pattern analysis and pattern application. But in web environment, the user activities become much more complex because of the complex web structure. So user behaviours recovery and pattern definition play more important roles in web mining than other applications. In this paper, we gave a new view on behaviour recovery and complicated pattern definition. We used several methods to recover different user behaviours, such as simple behaviour, sequence visiting, tree structure behaviour, acyclic routing behaviour and cyclic routing behaviour. Based on various recovered behaviours, we raised how to define complicated usage patterns. These usage patterns include constraint association rules, constraint sequential patterns, deepest access paths, shortest access paths, tree structure accessing patterns, parallel visiting patterns, circle visiting patterns and so on. We also gave some experiment results about these complicated access patterns which reveal some interesting usage behaviours.

1 Introduction

Web usage mining is an application of data mining methods in web field. Traditional data mining methods include association rules, sequential patterns, clustering, classifications [1] and other methods combining with machine learning methods such as Naïve Byes, Neural net and Markov model. The steps for data mining include data preparation, pattern mining, and pattern analysis and pattern application [2]. Web usage mining aims to find the hidden usage patterns from web usage data, and in web environment, these usage data are mostly recorded as web server logs.

However, it is acknowledged that the web application is totally different from other traditional data mining application, such as “Goods Basket” model. We can interpret this problem from two aspects:

1) *Weak Relations between user and site:*

Visitors could access the web site at any time from any place and even without any clear idea about what they want from the web. On the other hand, it is not easy for the site to discriminate different users. WWW brings great freedom and convenience for users and sites, and great varieties among them as well. So the relation between supply and demand becomes weak and vague.

2) *Complicated behaviours:*

Hyperlink and back tracking are the two important characters in web environment, which make user's activities more complicated. For the same visited contents, different users can access them with different patterns. Also, the user's behaviours are recorded as visiting sequence in web logs, which can not exactly reflect the user's real behaviours and web site structures.

In web environment, user behaviours are hidden in usage data, so access patterns are various and different from other applications.

In this paper, we give a new view on web usage mining. We prefer on recovering user's activities from usage data and defining new usage patterns from recovered behaviours. The task of behaviour recovery aims to recover user's activities from these URL sequences. The user's complicated activities through the site, such as back tracking and circle visiting, are recorded as URL sequences in web logs. We give several methods to recover different behaviours. They are simple behaviour (single item and sequence access), tree structure behaviour, acyclic routing and cyclic routing behaviour. The choice of proper recovery methods depends on what kind of usage patterns we want to mine in the succeeding step. Based on these recovered behaviours, we extend the definition for web usage mining. Besides the traditional usage patterns, we also raised Constraint Association Rules, Constraint Sequential Patterns, Tree Structure Accessing Patterns, Parallel Visiting Patterns and Circle Visiting Patterns.

The rest of this paper is organized as follows. In section 2, we review the work related with the problem we consider here. In section 3 we describe the formal definitions of behaviours recovery and complicated pattern definition. Section 4 concerns on the five recovery methods, we give a real example of how to recover user's navigational behaviours. Section 5 is a detailed explanation of how to define complicated usage patterns from user's recovered behaviours. In section 6, we give some experiment results that describe the recovered behaviours and complicated usage patterns. Section 7 is a short conclusion.

2 Related Works

The four general steps for data mining don't give clear tasks for behaviour recovery and usage pattern definitions. Behaviour recovery should be included in data preparation, and pattern definitions are made before pattern mining. So these two steps should exist between data preparation and pattern mining.

The tasks of data preparation in web usage mining include:

- Collection of usage data for web visitors: Most usage data are often recorded as kinds of web server logs. In E-Commerce and other services that need user registration, usage data are recorded as other file format.
- User identification: it is easy to identify different users in E-Commerce and other registration situation, though it can not be

avoided that some private personal registration information is misused by hackers. But for common web sites, it seems not easy to identify different users. In this situation, user can freely visit the web site. User's IP, Cookies and other limited client information, such as agent and version of OS and browsers, can be used for user identification. In this step, the usage data for different users are separately collected.

- Session construction: after user identification, different sessions for the same user should be reconstructed from this user's usage data collected in the second step. A session is a visit performed by a user from the time she enters the web site till she leaves [3]. Two time constraints are needed for this reconstruction, one is that the duration for any session can not exceed a defined threshold; the other is that the time gap between any two continuously accessed pages can not exceed another defined threshold.
- Behaviour recovery: reconstructed sessions are not enough to depict the varieties of user navigation behaviours in web usage mining. In most cases, any kinds of usage behaviours are only recorded as a URL sequence in sessions. The revisiting and back tracking result in the complexity of user navigation, so the task of recovery aims to rebuild the real user behaviour from the linear URLs in sessions. We will give a detailed explanation in section 4.

Pattern mining aims to mine the defined patterns from the usage data collected in the step of data preparation. The already existing patterns include association rules (or frequent item set), sequential patterns, the most forward access patterns [4], tree structure access patterns [5], clustering and classification. These patterns are originally mined from the sessions, while not from the recovered behaviours; also the desired patterns are defined even before the data preparation, while not after the behaviour recovery, so it is inevitable that these patterns can not exactly depict the usage behaviours. In [6] a query mechanism is raised that can query the user behaviours. This mechanism allows the input of freely defined access patterns, and then it queries these patterns from reorganized usage data. But this query mechanism can not depict complicated patterns such as tree structure access patterns and cyclic routing patterns. And also the storage reorganization for usage data requires much memory space.

3 Problem Statements

In web usage mining, time set, users set and web pages set are the three key entities, which are defined as a T , U and P . A session is a visit performed by a user from the time when she enters the web site to the time she leaves. A session is a page sequence ordered by timestamp in usage data record and is defined as $s = \langle p_1, p_2 \dots p_m \rangle$ ($p_i \in P$, $1 \leq i \leq m$), and these pages can form another page set $s' = \{p'_1, p'_2 \dots p'_k\}$, ($p'_i \in s$, $p'_i \supset p'_j$, $1 \leq i, j \leq k$). User behaviour is recovered from the session for this user and de-

defined as $\mathbf{b} = (s', \mathbf{R})$, where \mathbf{R} is the relations among s' and all user behaviours \mathbf{b} s form a behaviour set named \mathbf{B} .

An access pattern is some pages organized in an explainable way that some users accessed at some time, and it is mined from recovered user behaviours. “Explainable way” means that access pattern can give a reasonable and meaningful interpretation, either from web site semantics or web structure, or even from time factor and geographic factor. Access pattern is defined as $\mathbf{A} = (\mathbf{B}', \mathbf{M})$, where \mathbf{B}' is mined from \mathbf{B} and have the same format as \mathbf{B} and \mathbf{M} is the meaningful interpretation for \mathbf{B}' .

The goal of behaviour recovery is to recover the user behaviour \mathbf{b} from the right session s , because user behaviours are hidden in sessions. The goal of access pattern definition is to define the required pattern we want to mine from the recovered behaviours.

4 Behaviours Recovery

As explained in the former sections, sessions reconstructed from usage data record only the accessed pages sequence ordered by timestamp, but not reveal the real activities for these pages. It is user behaviours that record user’s interest and real activities through the web site. From the point of site structure, a web site is a directed graph, and in this graph, the vertex is a page and the directed edge is a hyperlink between two pages. From the point of site semantics, a web site is a knowledge web characterized with concept hierarchy. So user behaviours reveal not only the required content during his visiting, but also some of site structure and concept hierarchies, and also reveal the activities on these structures and hierarchies characterized with revisiting, back tracking and book marking. So user behaviour recovery is necessary for usage understanding.

On the other hand, all the user behaviours are latent in sessions, so different recovering methods can construct different behaviours. The strategy for proper recovering method is correlated with what kind of access patterns we want to mine in the next step. From simple to complex, we show here some strategies of behaviour recovery. We illustrate this problem from a real session collected from our server logs (www.telematik-institut.org). We reconstruct the sessions based on the strategy explained in section 2. This session is listed as following:

$$s = \langle 0, 292, 300, 304, 350, 326, 512, 510, 513, 512, 515, 513, 292, 319, 350, 517, 286 \rangle$$

In this session, the pages are labelled with IDs. For simplicity, in the later sections, we replace all the pages with their corresponding IDs if there is not extra explanation. In this session, this user raised 17 page/times access requirements. 0 and 286 were accessed separately as entrance and exit pages. Besides the entrance and exit pages, there are two groups of pages, one group is (300, 304, 326, 510, 513, 319, 517, 515) which were accessed only once, and the other group is (292, 350, 512) which were accessed more than once. Now we will explain several strategies for recovering different user behaviours.

4.1 Simple Behaviours Recovery

This strategy is the simplest one and overlooks all the repeated pages in a session. It includes two kinds of behaviours.

The first is that user behaviours are represented with only those unique accessed pages, which is the simplest recovery strategy. So simple user behaviours can be recovered from this session as:

$$s' = \{0, 292, 300, 304, 350, 326, 512, 510, 513, 515, 319, 517, 286\}.$$

The second method is that user behaviours are represented with those unique accessed pages and also the access sequence among these pages. For those pages accessed more than once, we concern only the first happening. Based on this thinking, user behaviours for this session can be recovered as:

$$\langle 0 - 292 - 300 - 304 - 350 - 326 - 512 - 510 - 513 - 515 - 319 - 517 - 286 \rangle$$

From the user behaviours recovered by the first method, association rules and frequent item sets can be mined in further step. Sequential patterns can be mined from the user behaviours recovered by the second method.

4.2 Tree Structure Behaviours Recovery

The simple recovery strategies listed above play great importance in data mining. While in web usage mining, revisiting and back tracking are the two important characters in user behaviours, which take place as the form that some pages were accessed more than once during a session. Those pages accessed more than once led to different access directions, which formed behaviours like tree structure. Tree structure behaviours not only depicted the visiting patterns, but also revealed some conceptual hierarchy on site semantics.

In the tree structure behaviour, each different page happens only once. To recover access tree t from session s , we used a page set named P to store the unique pages that already exist in t , and we also used a pointer pr pointing to the last recovered node during recovering in t . The recovery strategy is:

- 1) Set $t = \mathbf{NULL}$;
- 2) Read the first entrance page in s as the tree root r , let pr pointing to r and insert this page to P ;
- 3) Read new page from s and judge if the same page exist in P ;
 - i) Exist in P :
 - 4) Find this already existing node n in t and set pr point to this node,
 - 5) Go to step 3.
 - ii) Not exist in P :
 - 4) Insert this new page to P ,
 - 5) Create a new node and insert this new node as a new child for pr ,
 - 6) Let pr point to this new node,
 - 7) Go to step 3.

The tree structure behaviours for the above session can be recovered with our strategy as the following figure:

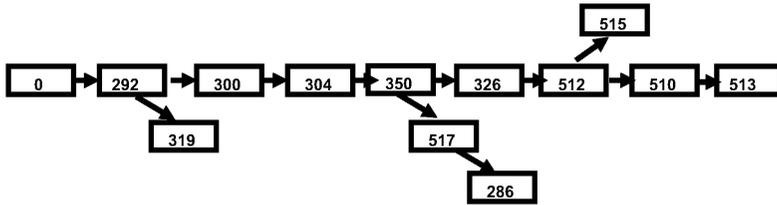


Fig. 1. Tree Structure behaviour

Tree structure behaviours can help to mine those access patterns with tree structure and also help to mine most forward sequential patterns or deepest access path.

4.3 A Cyclic Routing Behaviours Recovery

A web site is a complex directed graph, and further we can take one visiting for a user as a routing process to find interesting contents. Even for the same page contents, different users could access them with different routing methods, which revealed personal difference in visiting custom and understanding.

“Acyclic routing behaviour” means that during one visiting (session), there existed at least two different pages between which there were at least two different access paths. This behaviour indicates that user can access the same destined content from the same start content but via different paths. With acyclic routing behaviours, we can further query the shortest path and most popular path between two pages.

The final recovered behaviour is like a lattice structure defined as l , and we also used P to store unique pages in l , and pr pointing to the last recovered node during recovering in l . We used the following strategy to rebuild the acyclic routing in a session.

- 1) Set $l = \text{NULL}$;
- 2) Read the first entrance page in s as the top node t , let pr pointing to t and insert this page to P ;
- 3) Read new page from s and judge if the same page exist in P ;
 - i) Exist in P :
 - 4) Find this same existing node n in l and judge the relation between n and pr ,
 - a) n can be backward tracked from pr
 - 5) Set pr point to n ,
 - 6) Go to step 3.
 - b) n can be forward tracked from pr
 - 5) Build new edge directed from pr to n , if there is not directed edge from pr to n .
 - 6) Set pr point to n ,
 - 7) Go to step 3.

- c) n can not be tracked from pr in a single direction
 - 5) Build new edge directed from pr to n ,
 - 6) Set pr point to n ,
 - 7) Go to step 3.
- ii) Not exist in P :
 - 4) Insert this new page to P ,
 - 5) Create a new node and insert this new node as a new child for pr ,
 - 6) Let pr point to this new node,
 - 7) Go to step 3.

Based on the above strategy, we can recover acyclic routing behaviour from the same example. The following figure displays the final result.

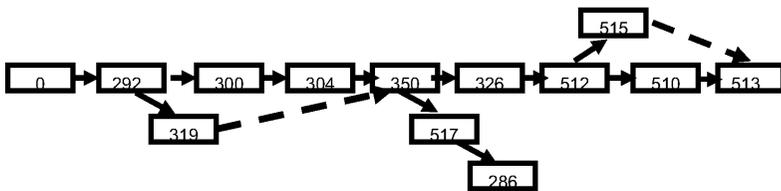


Fig. 2. Acyclic routing behaviour

4.4 Cyclic Routing Behaviours Recovery

Besides recovering acyclic routing behaviours, we also tried to recover cyclic routing behaviours. The premise for recovering tree structure, acyclic routing and cyclic routing behaviours is the happenings of revisiting and back tracking during one session. Revisiting and back tracking are two special characters in web usage mining, which can be seldom recorded in super markets. The revisiting happened sometimes due to the site structure. Currently most sites have complex structure and even frame-based structure, so this phenomenon happens more popularly. The repeated pages usually play great importance in usage navigation, web structure and web semantics.

We have made a statistic for revisiting and back tracking on two web sites, one is www.telematik-institut.org and the other is www.tele-task.de. The former is frame-based, while the latter is free designed. Even after filtering those adjacent repeat pages, we found that in 12% and 4% of sessions there exist revisiting or back tracking phenomenon.

The strategy for recovering cyclic routing behaviour is similar with but more complicated than recovering acyclic routing, because this strategy builds backward links for repeat pages, and tracking among pages needs complex structures. The following figure shows cyclic routing behaviours recovered from the same example.

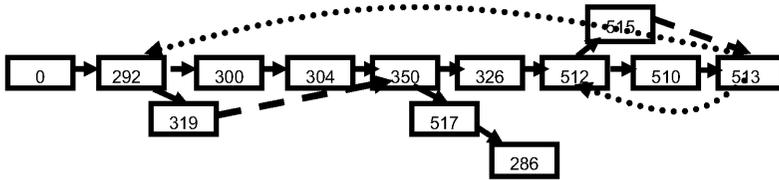


Fig. 3. Cyclic routing behavior

Different recovery strategies can yield different user behaviors, so the choice of proper strategy depends on what kind of patterns we want to mine in the next step.

5 Complicated Pattern Definitions

The tasks of pattern definition aim to define access patterns wanted to mine from recovered user behaviours. Within one visit for a user, there may exist more than one different access patterns. Even for the same access patterns, there may be more than one explanation. Due to the complicated structure of web site and varieties in user behaviours, it is hardly to numerate all the kinds of access patterns. Access patterns are the reflection of the site content and structures and must be interpreted by them. For the same page set but with different user behaviours, we can get different access patterns. Concrete pattern definition is necessary to understand user visiting tendency and interest. Besides to define the traditional patterns, we pay more attentions on how to define some new complicated patterns, which depict the user real visiting interest more exactly.

5.1 Traditional Pattern Definition

Similar to the other data mining applications, there are some already known usage patterns, such as association rules, frequent item sets, sequential patterns, clustering and classifiers [1]. We here concern on the patterns that related with association rules and sequential patterns.

Association rules and frequent item sets aims to find those co-occurred pages in sessions by a number of users. In this case, the user access patterns can be represented by these co-visited pages. Further investigating these co-visited pages, there may be content or structural relations among them.

Sequential pattern is an extension of association rules and frequent item sets. It cares not only the co-occurred pages in sessions, but also the visiting sequence among them.

It is very clear that repeated pages could not exist in these two kinds of patterns, and all the repeat pages are removed in behaviour recovery.

5.2 New Pattern Definition

Access pattern in web environment can be much more complicated than other pattern mining. The traditional patterns listed above are not enough to depict the user behav-

aviours in web environment, but they are the basic patterns to define more complicated patterns. Before defining access patterns, we must make an investigation about single user behaviours through the site, which is the task in behaviour recovery. As explained in former sections, time, user and pages are the three key factors in user behaviours and in access pattern definition as well, and also, the complicated site structures give us heuristics to define some more complicated patterns. From simple to complicated, we defined the new patterns as followings.

5.2.1 Constraint Association Rules

The difference between association rules and constraint association rules is that the latter aims to mine association rules with specified constraints defined before mining process. The constraints can be defined from the structure of user behaviour. Here we give some constraint association rules we used.

Frequent entrance pages and frequent exit pages: these two kinds of patterns are the simplest to understand user behaviour. We can find how the user was induced to begin his visit and why he left the site.

Frequent entrance pages with specified exit pages and frequent exit pages with specified entrance pages: these two kinds of patterns are to find the binary relations between entrance pages and exit page. The most favourite pairs can interpret the relations between entrance and exit.

Frequent pages that led to different access pages: not only from the point of site structure, but also from the factors that affect user behaviours, the pages that led to different access paths are very important to understand the accessing varieties from the same pages. Such pages are repeated pages, and it is due to the happening of the revisiting and back tracking during one session. From the semantics level, the contents of such pages should have multi-concept attributes; from the site structure, many different pages may be linked to such pages.

5.2.2 Constraint Sequential Patterns

Just like constraint association rules, constraint sequential patterns can help to understand those sequential patterns with constraints defined from site structure or user behaviour. We used the following patterns:

Frequent sequence with specified start page or end page: sequence patterns defined in section 5.1 have not any constraints before mining. With these constraint patterns here, we can find the most popular paths after or before defined pages, and these paths can show the different importance in deciding different access patterns.

Frequent sequence with or without hyperlinks: not all the continuously accessed pages in a session are connected with hyperlinks, and there are surely also some continuous pages without any hyperlinks. The use of bookmark by local browser, scripts, search query and some interactive forms can produce some unlinked pages in a session. The site holder should strengthen those favourite sequences with hyperlinks, but those unlinked sequences are the most desired contents that the corresponding users wanted.

Deepest access path after one page: in this case, all the user behaviours are recovered as a tree structure and every branch in this tree can be looked as an access path for this user. This kind of patterns are the longest branches after one page in this tree, and they are not only one of the most favourite paths for some visitors, but also can give visitors as much information as possible.

5.2.3 Frequent Tree Structure Access Patterns

The deepest access path list above has already related with tree structure patterns, because its definition is based on the recovered tree structure behaviours, but it takes a form of page sequence and reveals some linear relations among pages.

Tree structure pattern displays the detailed repeated pages and their diverse access paths. The most favourite tree access patterns are the most popular parts in web structure, or maybe display more reasonable organization for desired contents.

5.2.4 More Complicated Access Patterns

More complicated access patterns are the patterns with directed and undirected graph structure, and they are complicated routing within the site to find proper contents or just for navigations. Such access patterns are more exactly close to the users' real activities through the web sites.

Tree structure pattern ignores all the repeated pages in session, though the construction for this tree depends on those repeated pages as listed in section 4.2. This means that from the final constructed tree, we do not know the previous pages for the repeated pages.

More complicated accessing pattern concerns on such revisit and back tracking information. In our work, we focus on two kinds of patterns, one is *parallel visiting patterns* and the other is *circle visiting patterns*.

Parallel visiting patterns show the different access paths between one couple of start page and end page. We can further find the shortest paths between any two pages. This kind of pattern can be mined from recovered acyclic routing behaviours.

Circle visiting patterns aim to explain the frequent back tracking paths in sessions. Back tracking can happen because of site structure as frame-based site or because of some reminding information about previous pages when user accessed succeeding pages. They can be mined from those recovered cyclic routing behaviours.

6 Experimental Results

Two kinds of server logs are used in our experiments. One is collected from www.telematik-institut.org and the other is from www.tele-task.de. The time durations are over 5 moth and 14 months respectively. The general differences between these two sites and their logs are in the following table:

Table 1. Difference between TI and TT

	telematik-institut(TI)	tele-task(TT)
Site Scale	Middle	Small
Design Style	Frame based	Free designed
Pages	11988	1785
Users	27626	14854
Sessions	62570	29013

Firstly we calculate the session length for these two sites, and figure 4 shows the distribution of sessions by length. We can find a similar distribution by length for these two sites. An average session has 3~4 pages.

We think that different pages could play different roles for the user behaviour reconstruction: some pages are always visited as entrance pages for sessions, some pages are much easier to be exit pages, and other pages are as middle or navigation pages. The distribution by different roles can give us more about web structure and usage information

Let S be the set of all sessions and $|S|$ be the number of sessions in S . For a single page p , we set $Sum(p)$ be the accessed number in S , and $Sum(p)$ can be larger than $|S|$ because of revisiting in sessions. We denote by $E(p)$ the number of sessions that accessed p as entrance page, $M(p)$ the number of sessions that accessed p as middle page and $X(p)$ the number as exit page. To evaluate the performance of these roles for a page p , we defined four properties:

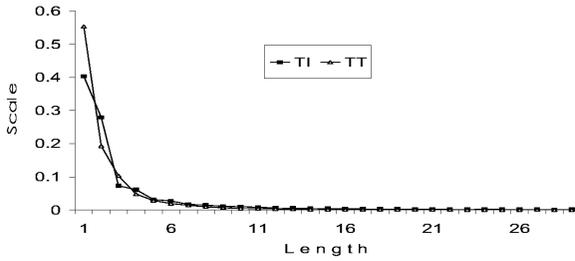


Fig. 4. Distribution of sessions by length

1) outer entrance support:

$$oe-support(p) = \frac{E(p)}{|S|}$$

2) inner entrance support:

$$ie-support(p) = \frac{E(p)}{E(p) + M(p) + X(p)}$$

3) outer exit support:

$$ox-support(p) = \frac{X(p)}{|S|}$$

4) inner exit support:

$$ix-support(p) = \frac{X(p)}{E(p) + M(p) + X(p)}$$

The general statistics about entrance and exit pages for two sites are shown in the next table:

Table 2. Entrance and exit page for TI and TT

	Telematik-institut(TI)	tele-task(TT)
Total pages	11988	1785
Entrance pages	6072	215
Exit pages	5295	225

The distribution for pages by these properties is shown in the next four figures:

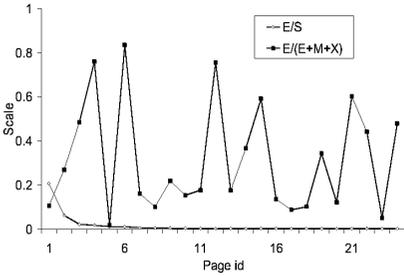


Fig. 5. Measures for entrance page (TI)

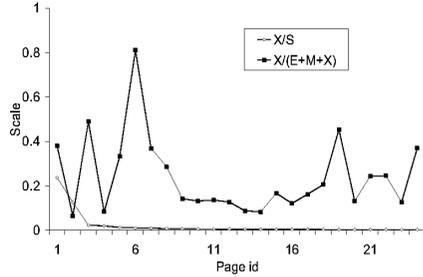


Fig. 6. Measures for exit page (TI)

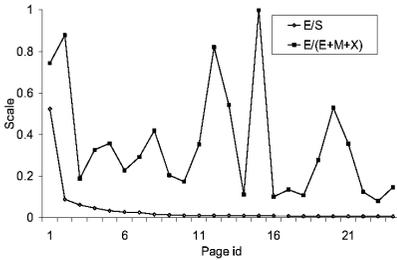


Fig. 7. Measures for entrance page (TT)

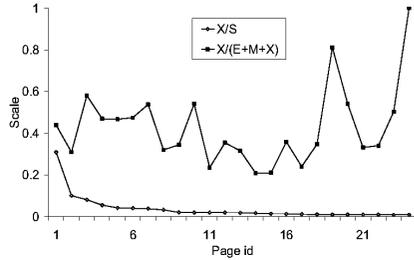


Fig. 8. Measures for exit page (TT)

From these four figures, we found that some pages are always accessed with entrance pages with their *ie-support* near to 1, and some pages are always exit pages with their *ix-support* near to 1. There are also some entrance pages with their *oe-support* near to or even larger than *ie-support* (in Fig 5 and 6), so do some exit pages, which happened due to the repeat visiting in sessions.

We also found that the distribution of entrance and exit pages revealed the web structure. For TT site, the homepage is the most important entrance page, because this site is dedicated to a project and homepage has the major content. But TI site is greatly different from TT site, because it owns two language versions and diverse information, and many users entered this site by search engine. But we find some similar usage tendency on these two sites if the homepages are ignored: exit page is better to discriminate different group of users than entrance page. The next two figures are made based on most frequent 24 entrance and exit pages. Exit page has a higher average scale than entrance page, as shown 2.1% vs. 1.6% for TI and 2.6% vs. 1.8% for TT.

7 Conclusions

In this paper, we tried a new view on web usage mining. We focused on user behaviour recovery and access pattern definition, because they are the basis for further pat-

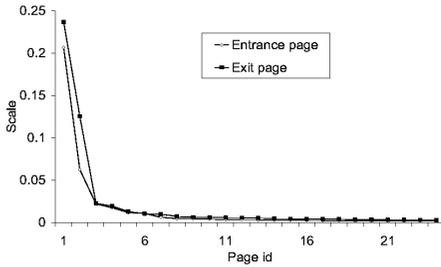


Fig. 9. Entrance vs. Exit (TI)

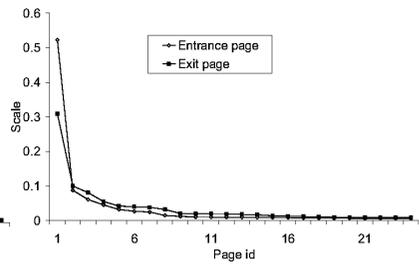


Fig. 10. Entrance vs. Exit (TT)

tern mining and require more concerning. Behaviour recovery enhances the tasks in data preparation, and new defined access patterns help to understand the diversities among users and also the complicated web structures. The new defined access patterns are good extension and combination of traditional patterns.

But for usage mining, pattern mining is the core. In the future, we will try to use different mining methods to mine our defined patterns. We have already developed some pattern mining methods, which can give us 11 basic patterns. These basic patterns can be used to further get incremental usage information.

References

1. J. Srivastava, R. Cooley, M. Deshpande and P. Tan: Web Usage Mining: Discovery and Application of Usage Patterns from Web Data, ACM SIGKDD, (2000)
2. R. Cooley, B. Mobasher and J. Srivastava: Data Preparation for Mining World Wide Web browsing Patterns, Knowledge and Information System, Vol.1, No.1, pp.5-32, (1999)
3. M. Spiliopoulou, B. Mobasher, B. Berendt and M. Nakagawa: A Framework for the Evaluation of Session Reconstruction Heuristics in Web Usage Analysis, INFORMS Journal on Computing, 15, 171-190
4. Ming-Syan Chen, Jong Soo Park and etc.: Data Mining for Path Traversal Patterns in a Web Environment. Proceedings of the 16th International Conference on Distributed Computing Systems (1996)
5. Mohammed J. Zaki: Efficiently Mining Frequent Trees in a Forest. In SIGKDD'02 (2002)
6. Bettina Berendt, Myra Spiliopoulou: Analysis of navigation behaviour in web sites integrating multiple information systems. The VLDB Journal, (2000)