UBB Mining: Finding Unexpected Browsing Behaviour in Clickstream Data to Improve a Web Site's Design

I-Hsien Ting
Department of Computer Science
The University of York
Heslington, York YO105DD, United Kingdom
derrick@cs.york.ac.uk

Chris Kimble
Department of Computer Science
The University of York
Heslington, York YO105DD, United Kingdom
kimble@cs.york.ac.uk

Daniel Kudenko
Department of Computer Science
The University of York
Heslington, York YO105DD, United Kingdom
kudenko@cs.york.ac.uk

Abstract

*This paper describes a novel web usage mining approach to discover patterns in the navigation of websites known as **U**nexpected **B**rowsing **B**ehaviours (UBBs). By reviewing these UBBs, a website designer can choose to modify the design of their website or redesign the site completely. UBB mining is based on the **C**ontinuous **C**ommon **S**ubsequence (CCS), a special instance of **C**ommon **S**ubsequence (CS), which is used to define a set of expected routes. The predefined expected routes are then treated as rules and stored in a rule base. By using the predefined route and the UBB mining algorithm, interesting browsing behaviours can be discovered. This paper will introduce the format of the expected route and describe the UBB algorithms. The paper also describes a series of experiments designed to evaluate how well UBB mining algorithms work.*

1. Introduction

Web usage mining is now the most popular technique for analysing a user's browsing behaviour in Clickstream data. The results can not only be used to understand how a user navigates their way through a site, but also to provide a better service [9], an adaptive website [10] and website personalisation [11]. Among all of these applications, web site design is the most important success factor for a website, especially in E-commerce. It is therefore of value to be able to analyse browsing behaviour and apply the results to improve the website design [6].

Most of the research about using web usage mining techniques to discover user's browsing behaviour is based on the direct method. This method processes the Clickstream data directly to find 'an interesting pattern'. However, different patterns of user's browsing behaviour can have very different meanings in different websites. What is interesting in one context may not be interesting in another. In this paper, we propose a novel web usage mining method called Unexpected Browsing Behaviour mining, or UBB mining, to get around this problem.

UBB mining is useful for website designers to understand how a user browses their website, especially for those website designers who want to redesign their website. The concept behind this method of web usage mining is that the designer of the site ought to be able to define patterns of 'expected browsing behaviour', and then by using this as a template, we should be able to discover any unexpected deviations from these routes in the Clickstream data.

The rationale behind this is that the designer of the site is the person who best understands the overall design concept of the site and so is best placed to define an expected route through it. Using this predefined expected route and the proposed UBB mining algorithms, browsing routes that do not match the expected route are then identified as instances of unexpected browsing behaviour. The website designer can then use these to find the reason why this behaviour occurs and act accordingly.

We consider UBB mining to be a form of data mining, since it discovers regularities (or actually "irregularities") in the Clickstream data. The search space for these regularities is restricted by additional knowledge, in our case the expected browsing route.

The structure of this paper is as follows: a brief overview of the background and motivation is introduced in section 1. In section 2, some related literature concerning user's browsing behaviour discovery and the concept of common sequence is discussed. Then, some examples are used to show the need for UBB mining to discover UBBs in section 3. The detail of the UBB mining algorithm is presented in section 4 and the performance of UBB mining is evaluated in section 5.

Section 6 presents the conclusions of this paper and outlines some directions for future work.

2. Literature Review

2.1. Discovering User's Browsing Behaviours

In browsing behaviour discovery, there are two popular approaches to finding an interesting pattern. The first is to use a visualisation technique to present the user's browsing history in a visualised graph or map. For example, Canter et al. (1985) proposed a way to group different user's browsing behaviour into six indices that can characterise navigational behaviour [3]. The Footstep map is another popular visualization tool that uses the concept of a spanning tree to convert the user's browsing route into a footstep map [5]. The Footstep graph is a visualisation technique that improves the footstep map by using an x-y plot to present the user's browsing pattern based on browsing sequence and browsing time. The user's browsing trend can be very easily extracted using the footstep graph [15]. The advantage of these visualisation tools is that the results are very easy to be read and can be understood by the human eye. However, the weakness of this kind of technique is that it is not robust enough to deal with a large amount of complex Clickstream data.

The second technique is to use a data mining technique to analyse Clickstream data; this is also known as web usage mining [4]. Applying the traditional mining algorithms, many interesting user's browsing patterns can be discovered. For example, a clustering algorithm can group the users into suitable clusters according to their browsing behaviour for measuring the similarity [12]. An association rule algorithm can discover the relationship between different user's browsing routes and trend analysis or sequential mining algorithms can be used to discover a sequential pattern in user's browsing behaviour [7]. Similarly, the HPG algorithm can be used to model user's browsing behaviour and find a user's preferred trail [2]. The advantage of this kind of browsing behaviour discovery technique is that a large amount of data can be processed very efficiently [11]. However, the results produced by this kind of tool are sometimes difficult to interpret and explain.

2.2. Common Subsequence (CS) and Longest Common Subsequence (LCS)

In this paper, we propose a sequential mining algorithm called UBB mining. UBB mining is based on the concept of continuous common subsequence (CCS), which is special instance of the common subsequence (CS). CS is a well-designed sequential mining algorithm. Given that there are two sequences X and Y, then if Z is a subsequence of both X and Y, we say that Z is a common subsequence of both X and Y.

For example, if *X={a,b,c,d,e,f,g}* and *Y={b,d,e,h,i}* then the common subsequence of sequence *X* will be *CS={b},{d},{e},{b,d},{d,e},{b,e},{b,d,e}*. Generally, the intersections between *X* and *Y* are found by using the *dynamic programming algorithm* [1]. One of the most interesting applications of the CS is the identification of the subsequence with maximum-length. This is known as *longest common subsequence* (LCS) or *edit-distance*. In above example, the LCS of sequence *X* and *Y* is *LCS={b,d,e}*.

In web usage mining, the CS and LCS are usually applied to clustering and sequential mining. The CS and LCS are also very useful in sequential mining to discover the relevance, co-occurrence and difference between sequences [8]. Finally, in some Clickstream clustering algorithms, the LCS is use as the core technique for an algorithm to measure the distance between different sessions' sequences.

3. Different Viewpoints of Browsing Patterns

In web usage mining research, such as that discussed above, many browsing patterns can be identified by using various techniques. The researcher usually uses a web mining algorithm to find an interesting pattern. The interesting pattern will then be 'explained' by the researcher using some characteristic of the pattern. However, identical patterns can have different meanings in websites with different structure or attributes or even in the same website at a different time.

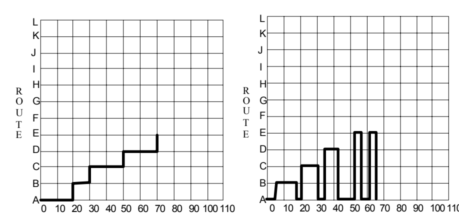

Figure1. (a) The user's browsing pattern that presented as *Upstairs* pattern in footstep graph (b) The user's browsing pattern that presented as *Fingers* pattern in footstep graph

For example, in our previous research, a visualisation technique called *footstep graph* was developed to model the user's browsing behaviour and was used to identify interesting patterns [14]. In figure 1(a), the footstep graph shows an *Upstairs* pattern. In some e-commerce websites, this kind of browsing pattern means the user is surfing the website smoothly. However, in another site the user may not be following the path the designer intended. Another example, shown in figure 1(b), is when the user's browsing behaviour presents a *Fingers* pattern. In some cases, this means the user's has fallen into a browsing loop. Normally, this kind of user's browsing pattern will indicate that there is some problem in the website, and a redesign of the website is necessary. However, this browsing pattern may sometimes reflect the website designer's design concept. Finally, a pattern such as a *Fingers* pattern can sometimes be a transitory pattern that simply indicates that the user is exploring a new site.

As with the visualisation techniques described above, traditional web usage mining techniques, which use a direct-method to process the Clickstream data and generate patterns, are also problematical. The results are not always easy to explain and sometimes a designer cannot identify a problem. Thus, the discovered pattern may not be very useful, especially for website redesign. The UBB mining approach proposed in this paper is therefore based on the website designer's viewpoint. Through this approach, the website designer can get the information they need about how their website is used and so it will be easier for them to identify instances of unexpected browsing behaviour. The results should then be of direct help to the website designer when reviewing or redesigning their site.

4. UBB Mining

Figure 2 shows the UBB mining process and its two prerequisites steps: *data pre-processing* and *pattern restoration*. UBB mining itself also consists of two steps: *route segmentation* and *unexpected browsing behaviour discovery*. A database of predefined expected routes is also an essential component of UBB mining. The detail of these four steps and the components in the process will be discussed in more detail below.

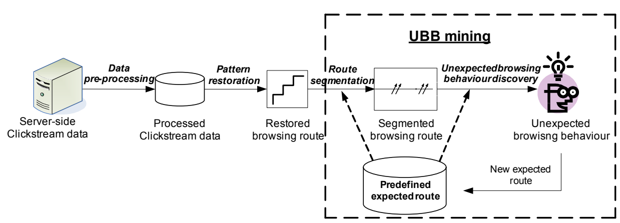

Figure 2. The process of the UBB mining and prerequisites steps

4.1. Data Pre-processing and Data Restoration

The raw server-side Clickstream data must be pre-processed to clean the noise, incomplete or irrelevant data before using it for web usage mining. For example, Clickstream data created by 'Bots' needs to be removed to ensure the data is really from a 'user' [14]. User and Session identification are also important steps of the data pre-processing. User identification is in order to distinguish the Clickstream data from different users. Generally, this is achieved by using the user's IP address, login / username or cookies to identify the user. Session identification then divides the user's browsing history into a number of distinct sessions. This can be a problem, when there is no login / logout data and is usually done by defining a time-out threshold (e.g. 30 minutes) to mark the end of a session [13]. In addition to data pre-processing, some data is lost due to caching in either the browser or a proxy server. This lost data must be restored to make sure the user's browsing pattern is as correct and complete as possible [16].

4.2. Predefined Expected Route Database

The predefined expected route-base is used to store the predefined expected route. For UBB mining, the expected route-base is the core of the entire technique; it is an essential component for both steps of UBB mining. The definition of the expected route is based on the concept of the continuous common subsequence (CCS). The details of the concept of CCS and the expected route definition method are discussed below.

4.2.1. Continuous Common Subsequence *(CCS)*. In some cases, common subsequence *(CS)* and longest common subsequence *(LCS)* are useful to measure the similarity between sequences. However, in sequential mining for web usage mining, the user's browsing behaviour is a continuous behaviour. The browsing behaviour will be very different if the browsing sequences between two are not identical. For example, consider the two browsing sequences $A=\{a,b,c,a,d\}$ and $B=\{a,b,c,d\}$. These two sequences all match the $LCS=\{a,b,c,d\}$ and both are very similar, but in terms of their browsing behaviour, they are quite different. In order to discover a UBB, a concept based on a CCS is proposed in this paper.

As discussed in section 2 of this paper, a CCS is a special instance of a CS. Assuming there are two sequences X and Y then, for $X=\{a,b,c,d,e,f,g\}$ and $Y=\{b,d,e,h,i\}$, the CS of the sequence X and Y will be $\{b\},\{d\},\{e\},\{b,d\},\{d,e\}\{b,e\}\{b,d,e\}$.

The CCS is the concurrence nodes in two different sequences are the same and continuous. For example, the CCS in above two sequences X and Y will be $\{d \rightarrow e\}$. Sometimes the CCS can be divided into many sub-CCS when necessary. For instance, there is a CCS $A=\{a \rightarrow b \rightarrow c \rightarrow d \rightarrow e\}$, then it can be divided into $A_1=\{a \rightarrow b\} \rightarrow \{c \rightarrow d \rightarrow e\}$ or $A_2=\{a \rightarrow b \rightarrow c\} \rightarrow \{d \rightarrow e\}$ or $A_3=\{a\} \rightarrow \{b \rightarrow c\} \rightarrow \{d \rightarrow e\}$…etc.

4.2.2. The Expected Route. In this paper, the concept of a CCS is used to define the expected route. The expected route is predefined, usually by the website designer or the person who has overall responsibility for site content (e.g. marketing manager, website owner, or website manager).

The expected route must follow the concept of CCS. For example, consider a predefined expected route $ER = R_1\{x_1 \rightarrow x_2\} \rightarrow F_1\{p;a\} \rightarrow R_2\{x_3 \rightarrow x_4\}$. $R_1\{x_1 \rightarrow x_2\}$ is a restricted subsequence, which the subsequence of a user's browsing must in order not to be identified as a UBB. $F_1\{p;a\}$ on the other hand, is a flexible subsequence, it also can be treated as a threshold CCS. The p in F_1 represents number of pages and a represents the attributes of the pages. Time, as discussed in section 3, can also be represented in the flexible subsequence as t. However, this will not be discussed in this paper.

To illustrate the whole process, consider the following expected route ER:

$$ER=R_1\{index \rightarrow product\_index\} \rightarrow F_1\{p_1;a_1\} \rightarrow R_2\{cart \rightarrow checkout\}$$

And the following user browsing route UR:

$$UR=index \rightarrow product\_index \rightarrow product1 \rightarrow product2 \rightarrow service \rightarrow cart \rightarrow checkout$$

Using the above notation, the user's browsing route can now be segmented into the following three sub-sequences:

$$UR=R_1\{index \rightarrow product\_index\} \rightarrow F_1\{product1 \rightarrow product2 \rightarrow service\} \rightarrow R_2\{cart \rightarrow checkout\}$$

The detail of how this segmentation is done will be discussed in the next section.

4.3. Browsing Route Segmentation

After the data pre-processing and data restoration, the next step is data segmentation. In this step, the user's browsing route is broken into segments based on the pre-defined expected route. In addition, the segmentation algorithm also carries out some preliminary UBB detection. The pseudo code of the segmentation algorithm is presented in figure 3.

At the beginning, every user's browsing route is put into a pool of UBBs, and the UBB mining algorithm only processes the browsing routes in this pool. Once the browsing route has been put into a pool of expected browsing routes, the route will not be processed any more.

First, the segmentation algorithm will extract the first restricted subsequence (R_1) from an ER. Then, the algorithm will search every continuous subsequence in a UR until every restricted subsequence (R) in the ER is matched. If not, the algorithm will put the UR into the UBB pool and the algorithm will not be performed again.

When there is a continuous subsequence that matches a restricted subsequence, the algorithm will give the name R_k to the continuous subsequence (S_pName) that means it is an appropriate subsequence to the restricted subsequence R_k. The algorithm then will extract next restricted subsequence of the ER, and the algorithm will be performed again. Once all of the restricted subsequences in the ER are matched and the UR is still alive, the UR will be put into the expected browsing behaviour pool. After performing the segmentation algorithm, all continuous subsequences of the ER that have not been recognized as a restricted subsequence will be treated as a flexible subsequence in next step.

```
Input ER_i ← Expected route i
Input UR_j ← User Browsing Route j
Output Segmented UR_j
Begin procedure Segmentation algorithm:
pointer=0
for k=0 to Number of R
    for m=0 to Number of Nodes in R_k
        for n=pointer to Number of Nodes in UR_j
            If all R_km=UR_jn then
                S_pName=R_k
                pointer=CurrentNode
            else
                UR_j → Unexpected route; end algorithm
            end if
        end for
    end for
end for
End procedure
```

Figure 3. The pseudo code of the segmentation algorithm

4.4. Unexpected Browsing Behaviour Discovery

The main work of the UBB discovery step is to test is a flexible subsequence in a UR matches the setting of an appropriate flexible subsequence in an ER. In this step, the UBB discovery algorithm will only process the UR in the expected browsing route pool. Assuming an ER:

$$ER=R_1\{index \rightarrow product\_index\} \rightarrow F_1\{p_1;a_1\} \rightarrow R_2\{cart \rightarrow checkout\}$$

And a segmented UR:

$$UR=R_1\{index \rightarrow product\_index\} \rightarrow F_1\{product1 \rightarrow product2 \rightarrow service\} \rightarrow R_2\{cart \rightarrow checkout\}$$

The UBB discovery algorithm will test is every restricted rule (R_1 and R_2) of the ER exists in the segmented UR. To make sure it is necessary to perform the UBB mining algorithm. Then, the algorithm will test is the F_1 in the segmented UR matches the threshold of the F_1 in the ER. For example, considering there are three different threshold settings of the F_1 in the ER.

(1) When *p=2* and *a=any* in the $F_1\{p;a\}$ then:

In the F_1 of the ER, *a=any* means any page in the F_1 of the UR is expected. Under this situation, the UR will be recognised as an UBB that is verified by the predefined ER. The reason why is even the F_1 of the UR passed the threshold of *a=any*, but it cannot pass the threshold of *p=2*.

(2) When *p=3* and *a=any* in the $F_1\{p;a\}$ then

Under this situation, the UR will be recognised as an expected browsing behaviour. The F_1 of the UR not only passed the threshold of *a=any*, but also passed the threshold of *p=3*.

(3) When *p=3* and *a=product* in the $F_1\{p;a\}$ then

In the F_1 of the ER, *a=product* means only the product related page in the F_1 of the UR is expected. Under this situation, the UR will be recognised as an unexpected browsing behaviour. The reason why is even the F_1 of the UR passed the threshold of *p=3*, but one of user browsed page is related to service. The UR therefore cannot pass the threshold *a=product*, and be recognized as an UBB.

```
Input ER_i ← Expected route I
Input UR_j ← Segmented User Browsing Route_j
Output matching result
Begin procedure UBB discovery algorithm:
for k=0 to number of SubIR_i and SubUR_j
        if SubIR_ik ≠ SubUR_jk then
                UR_j → unexpected route; end algorithm
        end if
end for
for m=0 to number of F in ER_i, n=0 to number of F in UR_j
        if F_m p ≥ F_n p and F_n a ∈ F_m a then
                UR_j → expected browsing behaviour
        else
                UR_j → unexpected browsing behaviour
        end if
end for
End procedure
```

Figure 4. The pseudo code of the UBB discovery algorithm

Figure 4 is the pseudo code of the UBB discovery algorithm. After performing the UBB mining algorithm, a UR that is still alive in the UBB pool will be recognised as a UBB, because there are no any ER support it to be an expected browsing behaviour.

Once the UBB has been discovered, a website designer can by this to review their website, and to check is there any potential problem in the website. If the website designer thinks that the discovered UBB is acceptable to be an expected route, it can then be defined as an expected route and stored in the expected route-base.

5. Experiment Results

In this section, we first present some expected browsing behaviours and UBBs that were discovered by using the UBB mining algorithm in an experiment based on the use of a website for a module of a university degree course. The teaching for this module was based on face-to-face lectures with supplementary material provided in web pages. The second part of this section is a performance evaluation of the UBB mining algorithm using different numbers of predefined expected routes and user sessions.

Experiment Results. To test the UBB mining algorithm, we used the server-side Clickstream data from the website described above. After data pre-processing, the total number of user sessions were 386. The designer of the site also produced a number of predefined expected routes based on his expectation of the student's use of the site. Some of the expected routes are listed below.

1. ER_1=R_1\{MIS\}
2. ER_2=R_1\{MIS→MIS_overview\}→F_1\{p=1, a=lecture\}
3. ER_3=R_1\{MIS→MIS_overview→$lecture_i$\}→F_1\{p=3, a=$lecure_i$_related\}→R_2\{MIS_overview→$lecture_{i+1}$\}

For example, route 3 contains two restricted rules and one flexible rule. The first restricted rule (R_1) requests a user to browse the *MIS* page first, then *MIS_overview* page and then i^{th} *lecture* page. Then, if the user browses more than three pages that are related to the i^{th} *lecture* page, they will pass the threshold of the flexible rule (F_1). Finally, the user must follow the second restricted rule (R_2) to browse the *MIS_overview* page and then the *next lecture* page ($i+1^{th}$).

After performing the UBB mining algorithm using the predefined expected routes to process the 386 user sessions, some expected browsing behaviours and UBBs were discovered. Table 1 shows five examples of the expected user's browsing behaviour, which matched the predefined expected routes. The frequencies in table 1 show how many times this specific browsing route occurred in the entire 386 user sessions. Table 2 shows five UBBs discovered in this experiment (an *External* page in table 2 means that a user leaves the website).

Table 1. Discovered expected browsing behaviours

No.	Route	Frequency
1	mis.html	58
2	mis.html→ MIS_Overview.html	7
3	Mis.html→ MIS_Overview.html→ Information_Revolution.html	4
4	mis.html→ MIS_Overview.html→ Strategic_Responses.html	3
5	mis.html→ MIS_Overview.html→ Basic_Assumptions.html→ Books.html→ MIS_Overview.html→ Information_Revolution.html→ Books.html	1

Once the expected browsing behaviours and UBBs are discovered, a website designer will often immediately understand how their website is being used. After reviewing the results, they can add the UBB as an expected browsing behaviour if the UBB is acceptable for them, or, on the other hand, they can change an expected browsing behaviour to an UBB if they think the expected browsing behaviour is not what they thought it to be.

Table 2. Discovered UBBs

No.	Route	Frequency
1	Mis.html→ External→ mis.html	6
2	mis.html→ External→ mis.html→ External→ mis.html	4
3	Technology_and_Change.html→ Bureaucratic_Organisations.html→ Networked_Organisations.html	4
4	Bureaucratic_Organisations.html	3
5	mis.html→ mis.html	3

However, the website designer must review their website's structure or redesign their website, if the discovered UBB is really not what website designer expected. For instance, the website designer should review their website's design to find the reason why no.1 and no.2 route in table 2 occur in their website. The website designer may also wants to highlight the link in *MIS_overview* page to guide the user to their website via the *MIS_overview* page in no.3 route of table 2.

Performance Evaluation. To evaluate the performance of the UBB mining algorithm, we tested the processing time of the UBB mining algorithm under different number of predefined expected routes (10, 20, 30, 40 and 50 routes) and different number of user sessions (100, 200… 900, and 1000 user sessions). The average number of browsed pages of a user session is six pages, and the average percentage of restricted rule and flexible rule in a predefined expected route are both 50%. The performance evaluation was performed under Windows XP operation system, 512Mb RAM, 1.3 GHz Intel Pentium 4 CPU. Figure 5 shows the performance evaluation result of the UBB mining algorithm.

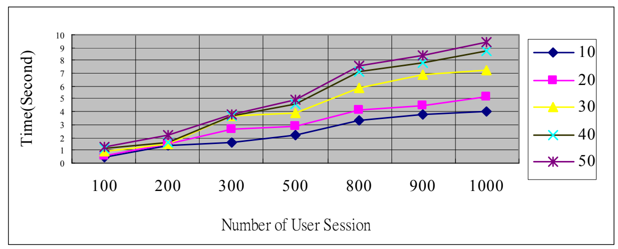

Figure 5. Performance evaluation of the UBB mining algorithm

The data of X-axis in figure 5 denotes the number of user sessions, and the data of Y-axis is the processing time of the UBB mining algorithm. The figure shows the performance of the UBB mining algorithm is very good. It not only processes the data and discovers the unexpected routes very fast (less than ten seconds), but the raise of the processing time is also not very intension with the increasing of the number of user sessions and expected routes.

6. Conclusions

In this paper, we proposed a novel web usage mining approach called UBB mining. UBB mining is based on detecting deviations from predefined patterns, which are specified by the designer or owner of the site. UBB mining is a sequential mining technique based on the concept of a continuous common subsequence. There are two algorithms included in the UBB mining: the segmentation algorithm and the UBB discovery algorithm. Through the UBB mining, a website designer can discover interesting user's browsing behaviours, which are unexpected. The results can not only be used by a designer to review, improve or redesign their website, but can also be used to model a user's browsing behaviour.

Our future research will move in two directions. The first concerns the way in which the expected routes are created. In the above example, the designer of the site was able to define an expected route in the form of a pattern of continuous common subsequences. With larger sites and different designers, this task could be more of a problem. We are currently working on a tool that will allow a designer to browse a site and "record" an expected route. Secondly, we now want to move from the *pattern discovery and analysis* step to the *recommendation and action* step. We will then be able to focus on how to apply the results that are discovered by UBB mining for improving or redesigning a website's design.

7. References

[1] Banerjee, A., and Ghosh, J., "Clickstream Clustering using Weighted Longest Common Subsequences", *In Proceedings of the 1st SIAM International Conference on Data Mining: Workshop on Web Mining*, 2001

[2] Borges, J. and Levene M., "Data Mining of User Navigation Patterns", *In Masand B. M. and Spiliopoulu, M. (Eds.): Web Usage Analysis and User Profiling, International WEBKDD'99 Workshop, San Diego, California, USA, August 15, 2000 LNCS 1836,* pp. 92-111, Springer-Verlag

[3] Canter, D., River, Rod., and Storrs, G., "Characterizing user navigation through complex data structure", *Behaviour and Information Technology*, Vol.4, No.2, 1985, pp. 93-102

[4] Cooley, R., Tan. P. N., and Srivastava, J., "Discovery of Interesting Usage Patterns from Web Data", *In Masand B. M. and Spiliopoulu, M. (Eds.): Web Usage Analysis and User Profiling, International WEBKDD'99 Workshop, San Diego, California, USA, August 15, 2000, LNCS 1836*, pp. 163-182, Springer-Verlag

[5] Dömel, P., "WebMap - A Graphical Hypertext Navigation Tool", *In Proceedings of the Second International WWW Conference*, Chicago, USA, 1994

[6] Fu, Y., Creado, M., and Ju, C., "Reorganizing Web Sites Based on User Access Patterns", *In Proceedings of the 10th International Conference on Information and Knowledge Management (CIKM'01)*, November 5-10, 2001, Atlanta Georgia, USA, pp. 583-585

[7] Hooker, G. and Finkelman, M., "Sequential Analysis for Learning Models of Browsing", *In Proceedings of WebKDD 2004 Workshop on Web Mining and Web Usage Analysis,*2004, Seattle, WA, USA

[8] Kothari, R., Mittal, P, Jain, V., and Mohania, M., On Using Page Cooccurrences for Computing Clickstream Similarity, *In Proceedings of SIAM International Conference on Data Mining*, May 1-3, 2003, San Francisco, CA, USA

[9] Lee, W. P., Liu, C. H., and Lu, C. C., "Intelligent Agent-based Systems for Personalized Recommendation in Internet Commerce", *Expert Systems with Applications*, Vol.22, 2002, pp.275-284

[10] Perkowitz, M., and Etzioni, O., "Adaptive Web Sites", *Communications of the ACM*, Vol. 143, Issue 8,2000, pp.152-158

[11] Mobasher, B., Dai, H., Luo, T., and Nakagawa, M.," Discovery and Evaluation of Aggregate Usage Profile for Web Personalization", *Data Mining and Knowledge Discovery*, Vol.6, 2002, pp. 61-82

[12] Nasraoui, O., Cardona, C., Rojas, C., "Mining Evolving Web Clickstreams with Explicit Retrieval Similarity Measures", *In Proceedings of International Web Dynamics Workshop, International World Wide Web Conference,* May 2004 New York, NY, USA

[13] Srivastava, J., Cooley, R., Deshpande, M., and Tan, P. N., "Web Usage Mining: Discovery and Applications of Usage Patterns from Web Data", *SIGKDD Explorations*, Vol. 1 , Issue 2, 2000, pp.12-23

[14] Tan, P. N., and Kumar, V., "Discovery of Web Robot Sessions Based on their Navigation Patterns", *Data Mining and Knowledge Discovery*, Vol. 6, 2000, pp.9-35

[15] Ting, I. H., Kimble, C., and Kudenko, D. "Visualizing and Classifying the Pattern of User's Browsing Behaviour for Website Design Recommendation", *In Proceedings of the First International Workshop on Knowledge Discovery in Data Stream* (ECML/ PKDD 2004) Pisa, Italy, 24 September, 2004, pp.101-102

[16] Ting, I. H., Kimble, C., and Kudenko, D., "A Pattern Restore Method for Restoring Missing Patterns in Server Side Clickstream Data", *In Zhang, Y. et al. (Eds.): APWeb 2005, LNCS 3399*, pp. 501-512, 2005, Springer-Verlag