APD-A Tool for Identifying Behavioural Patterns Automatically from Clickstream Data

I-Hsien Ting, Lillian Clark, Chris Kimble, Daniel Kudenko, and Peter Wright

Department of Computer Science, The University of York
Heslington, York YO105DD, United Kingdom
{derrick, Lillian.clark, kudenko, kimble, pcw}@cs.york.ac.uk

Abstract. Clickstream can be a rich source of data for analysing user behaviour, but the volume of these logs makes it difficult to identify and categorise behavioural patterns. In this paper, we introduce the Automatic Pattern Discovery (APD) method, a technique for automated processing of Clickstream data to identify a user's browsing patterns. The paper also includes case study that is used to illustrate the use of the APD and to evaluate its performance.

Keywords: Web Usage Mining, Clickstream Data, Browsing Behaviour, Traversal Pattern.

1 Introduction

A thorough understanding of how users navigate or browse through a site is important in the development of web-based applications, especially when including features such as recommendations, advertising or personalisation. This understanding is also critical for performing usability testing [2], performance evaluation and website redesign. Analysis of Clickstream data can aid our understanding of user browsing behaviour by providing detailed information on the patterns generated by users as they navigate through a website hosted on that server. However, the size and nature of Clickstream logs can make pattern detection and classification difficult and time-consuming.

This paper is a paper that looks at the application of KDD to website design. In our previous work [9], we have introduced the "Footstep" graph as a means of visualising user's browsing patterns from Clickstream data. In addition, we have also shown how certain types of patterns such as "Stairs", "Mountain", and "Fingers" could be indicative of navigational problems in a site, e.g. in [3]. Our next challenge was to automate pattern generation and classification in order to process large Clickstream data files efficiently. In this paper, we propose a methodology known as Automatic Pattern Discovery (APD) method. APD is based on the concept of sequential mining to analyse browsing behaviour by detecting basic browsing elements and defining patterns for each user.

This paper is organised into six sections. Section 1 introduces the Footstep graph; user browsing patterns are described in section 2 and, in section 3, the three step APD process is introduced. In section 4, the basic browsing elements and the algorithms of

B. Apolloni et al. (Eds.): KES 2007/ WIRN 2007, Part II, LNAI 4693, pp. 66–73, 2007.

the pattern detection steps are described and a case study to discover browsing pattern is presented in section 5. The experiment results and the evaluation of the APD algorithm are also shown in this section. The conclusion is presented in section 6.

2 Footstep Graph and Users' Browsing Patterns

In [4] we introduced the "Footstep" graph, a visualisation tool for identifying user's browsing patterns. The Footstep graph is based on a simple x-y plot where the x-axis represents time (in seconds) and the distance between points indicates the time between two nodes. The y-axis represents the nodes on the user's browsing route and the changes in the vertical axis indicate a transition from one node to another. The Footstep graph (figure 1) not only indicates the time-trends of a user's browsing history but also illustrates the relationship between each browsing node, transforming complex and unorganised Clickstream data.

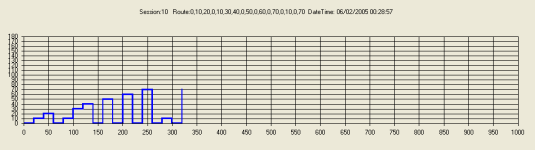
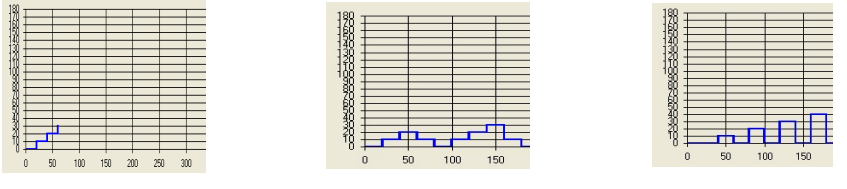


Fig. 1. A Sample Footstep Graph

2.1 Identifying User's Browsing Patterns

In our previous research, Footstep graphs were produced to demonstrate how this visualisation technique could help analyse user's browsing behaviour. From this research, we found certain frequently occurring patterns that were illustrative of user browsing behaviour. These patterns were named "Stairs" (including "Upstairs" and "Downstairs"), "Mountain" and "Fingers".

The Upstairs pattern is created when the user moves forward through previously unvisited pages in the website. An example of a Stairs pattern is shown in figure 2(a). These patterns are similar to Canter's "Path" pattern, which indicates that the user is exploring the website [1].



(a) (b) (c)

Fig. 2. (a) An example of a *Stairs* pattern (b) An example of a *Mountain* pattern **(c)** An example of a *Fingers* pattern

The Mountain pattern is a Downstairs pattern is immediately followed by an Upstairs one. An example of a Mountain pattern is shown in figure 2(b). This pattern is equivalent to Canter's "Loop" pattern, which indicates that the user is searching the site for a specific target. The Fingers pattern is found when the user moves directly from one page within the site to another and then directly returns to the original page see figure 2(c). This pattern is equivalent to Canter's "Spike" pattern and indicates that the user may have fallen into a browsing loop.

3 Automatic Pattern Discovery (APD)

The complete APD process contains three main steps: Clickstream Data Pre-processing, Browsing Route Transformation and Automatic Pattern Discovery. In turn, the automatic pattern discovery step contains three further sub-steps: browsing route segmentation, pattern definition and pattern detection. Figure 3 shows the whole APD process; the individual steps are discussed in detail below.

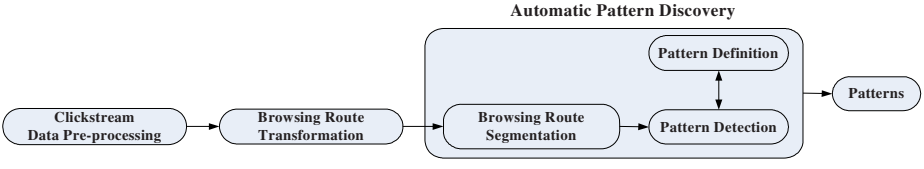


Fig. 3. The APD process

3.1 Clickstream Data Pre-processing

Clickstream data pre-processing is a necessary step not only for APD but also for any web usage mining technology. A standard data pre-processing process has been well-developed in current web usage mining research [4]. In general, the process should include the following steps: data filtering, data cleaning, user identification, session identification, 'bot' detection and data formatting and path restoration [9].

3.2 Users' Browsing Route Transformation

To discover a user's browsing pattern the browsing route must first be transformed to a number-based sequence. For example, a user's browsing route is shown in table 1. After transformation, a number-based sequence emerges as shown in table 2. In this case, the sequence starts from 0 and the increasing order of the sequence is 10. The transformation algorithm will search each node in the user's browsing route and assign each one a sequence number. In addition, the algorithm will also check whether each node has occurred before in the user's browsing route. If so, the algorithm will assign the same sequence number to the same nodes (e.g., the node No.3 /support/index.php has already occurred in the user's browsing route and has a sequence number '0', so the sequence number for the node No.3 is also '0').

Table 1. A sample user browsing route

No.	Date and Time	Accessed URL
1	15/06/2005,02:24:16	/support/index.php
2	15/06/2005,02:25:18	/support/contact.php
3	15/06/2005,02:26:20	/support/index.php

Table 2. A number-based sequence and time duration after the browsing route transformation

Number-Based Sequence	Time Duration	Accessed URL
0	0	/support/index.php
10	62	/support/contact.php
0	62	/support/index.php

3.3 Automatic Pattern Discovery

In sequential mining, the focus is always on the segmentation or transformation method of the sequence. A good segmentation method can help the sequential mining to generate better results. In APD, the segmentation method is easier than other sequential mining methods. As the APD methodology compares the sequence order of each node in entire route and the relationship between every two nodes, the segmentation method is to divide each node of a route into individual element. For example, a route A={0→10→20→30} (the arrow symbol in A means from one node to another node) will be divided to A'={0,10,20,30}, and a segmented route will never look like A2'={0,10,20→30}, because the continuous nodes are not allowed in the route after segmentation. The segmented route can then be used for the pattern detection step.

4 Basic Users' Browsing Elements, Pattern Detection and Pattern Definition

4.1 Basic Browsing Elements

(1) Level-1 Elements: Same, Up and Down

The segmented browsing route is transformed to basic level-1 elements by comparing the relation between every two nodes. These elements are known as "Same", "Up" and "Down". A Same element occurs when user continuously browses the same web pages by either refreshing a page, opening the same page or other activity. If there are two nodes in the route {ai, ai+1} and ai=ai+1 (e.g. {2, 2}) then the relationship between these two pages will be assigned the level-1 element Same (Figure 4). An Up element occurs when the user has moved through the website by using forward browsing behaviour, i.e. moving from one web page to another web page they have not yet visited. (Figure 5). A Down element occurs when the user has moved through a website by using backward browsing to one they have visited before, causing the order of the sequence to be lower. If there are two nodes {ai, ai+1} in a route and ai>ai+1, then the relationship between these two pages will be assigned the level-1 element Down (Figure 6).

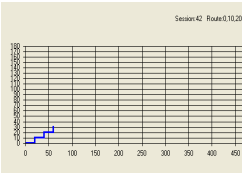


Fig. 4. The *Up* element

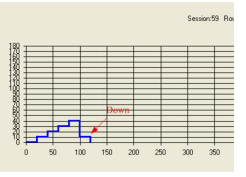


Fig. 5. The Down element

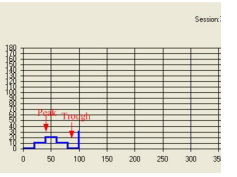


Fig. 6. *Peak* and *Trough* elements

(2) Level-2 Elements: Peak and Trough

The next level of browsing elements, Level-2 elements, are based on measuring the relationship between each contiguous element of the browsing route to discover changes in browsing directions or turning points. These Level-2 elements, Peak and Trough, are be used to define these turning points.

A Peak occurs when the browsing direction shifts from a forward to a backwards direction. For example if there are two nodes {bi, bi+1} in a route, and bi= Up and bi+1=Down (e.g., {Up, Down}) then the relationship between these two nodes will be assigned the level-2 element Peak. A Trough occurs when the browsing direction shifts from backwards to forwards. (Figure 7).

Table 3. Level-1 route transformation

```
For each Raw Users' Browsing Node Ai
  If Ai ≠ last node of route then
    If Ai<Ai+1 then
      Level-1 elements Bj='Up' (j=0…n)
    Else if Ai>Ai+1 then
      Level-1 elements Bj='Down'
    Else if Ai=Ai+1 then
      Level-1 elements Bj='Same'
```

Table 4. Level-2 route transformation

```
For each Level-1 Node Cp
  If Cp ≠ last node of route then
    If Cp='Up' and Cp+1='Down' then
      Level-2 elements Dq='Peak' (q=0…m)
    Else if Cp='Down' and Cp+1='Up' then
      Level-2 elements Dq='Trough'
```

4.2 Pattern Detection and Pattern Definition

(1) Level-1 and level-2 Element Detection

The pattern detection step of the APD sequentially processes the route based on the concept of level-1 and level-2 elements. The algorithms for level-1 and level-2 browsing route transformations are shown in table 3 and 4.

(2) Pattern Definition and Detection

Before pattern detection, it is essential for the website designer to define what patterns are of interest. However, the way in which the patterns are defined may vary. For example, some stakeholders may define a Mountain pattern as being {Up, Peak, Down, Trough, Up} rather than simply {Up, Peak, Down}, or maintain that a Fingers pattern should contain at least two fingers rather than one finger only {Trough, Peak, Trough}. The pattern definition must therefore be flexible as there is no standard definition for any one pattern. Once elements have been detected and desirable

patterns defined, the pattern detection step can detect these patterns automatically by matching the routes and the pattern rules according to the pattern detection algorithm.

5 Case Study and Experiment Result

In this section, we describe a case study in which the APD was used to discover browsing patterns automatically. Some results of the case study are shown in this section and the performance of the APD algorithm is evaluated [3].

5.1 Two Examples of Pattern Definition

For this case study, the researcher was studying student's use of a website related to one of the courses they were studying. The rules for Upstairs, Downstairs, Mountain, and Fingers patterns, based on the level-2 based routes, were defined as follows:

a. The *Mountain* and *Fingers* pattern
If there are continuous elements {Up, Peak, Trough}, {Up, Peak, Down}, {Trough, Peak, Down}, {Up, Peak}, and {Peak, Down} in the level-2 based route, then they should be recognised as a Mountain pattern. If there are continuous elements {Peak, Trough}, in the level-2 based route, then they should be recognised as a Fingers pattern. All remaining {Peak} elements in the level-2 based route should be recognised as a Fingers pattern.

c. The Downstairs and Upstairs pattern
All remaining {Down} elements in the level-2 based route should be recognised as a Downstairs pattern. All remaining {Up} elements in the level-2 based route should be recognised as an Upstairs pattern.

5.2 Experiment Result

After the pattern rules were defined, the APD method was used to analyse the Clickstream data. We used the standard data pre-processing approach; including a pattern restore method [9], to pre-process the raw Clickstream data. Then the pre-processed Clickstream data was transformed to number-based sequence and each route in the Clickstream data was segmented (table 5).

The segmented routes were then transformed to level-2 routes by using level-1 and level-2 based route transformation (Table 6). Finally, the patterns were automatically identified and categories (table 7).

Table 5. Segmented browsing routes after transformation

Session Number	Number-based sequence
1	0,1,2
2	0,1,2,1,2
10	0,1,2,0,1,3,4,0,5,0,6,0,7,0,1,0,7

Table 6. Level-1 and level-2 based user's browsing routes

Session Number	Level-1 based browsing routes	Level-2 based browsing routes
1	up, up	up, up
2	up, up, down, up	up, peak, trough, up
10	up, up, down, up, up, up, down, up, down, up, down, up, down, up, down, up	up, peak, trough, up, up, peak, trough, peak, trough, peak, trough, peak, trough, peak, trough, up

Table 7. The final users' browsing patterns

Session Number	Patterns
1	Upstairs
2	Mountain, Upstairs
10	Mountain, Mountain, Finger, Finger, Finger, Finger, Upstairs

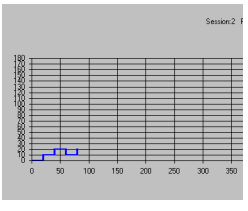


Fig. 7. Footstep graph of session 2

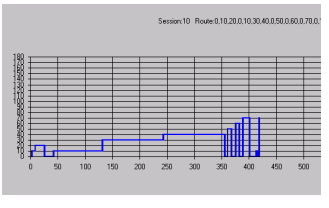


Fig. 8. Footstep graph of session 10

5.3 Performance Evaluation

A series of experiments was held to evaluate the performance of the APD algorithm. Tests were run using cleansed Clickstream ranging from 10 user sessions up to 1000 user sessions. The size of the cleansed Clickstream data files averaged 348 Kb for 500 sessions and 802Kb for 1000 sessions. The performance of the APD algorithm is shown in figure 9, where the x-axis represents the number of sessions per file and the y-axis represents the time in seconds.

According to figure 9, the execution time of the APD increases with number of sessions, and presents stable equimultiple increasing. Different average numbers of

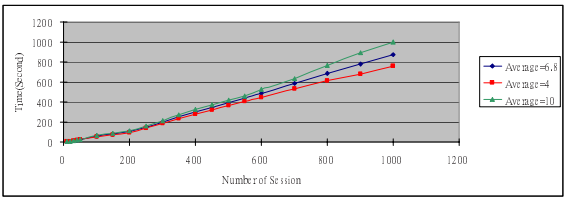


Fig. 9. Performance evaluation of the APD algorithm

browsing nodes do not affect performance. This shows the performance of the APD is under control and can be scaled-up to handle large volumes of data.

6 Conclusion

In this paper, described a novel sequence mining approach called APD, which was used to identify pre-identified patterns automatically. The paper also introduced the concepts of level-1 and level-2 elements of browsing behaviour, which constitute a vital part of the APD. This paper has shown that, by following the APD method, browsing patterns can be automatically identified efficiently, even when large amounts of Clickstream data are involved. We believe that such an approach can be an important tool for both web usage mining and for HCI research into browsing behaviour.

References

1. Canter, D., Rivers, R., Storrs, G.: Characterising users navigation through complex data structures. Behaviour and Information Technology 4(2), 93–102 (1985)
2. Chi, E.H.: Improving web usability through visualisation IEEE Internet Computing, March-April 2002, pp. 64–71 (2002)
3. Clark, L., Ting, I., Kimble, C., Wright, P., Kudenko, D.: Combining Ethnographic and Clickstream Data to Identify Browsing Strategies Information Research 11(2), paper 249 (2006), Available at http://InformationR.net/ir/11-2/paper249.html
4. Cooley, R., Mobasher, B., Srivastava, J.: Data Preparation for Mining World Wide Web Browsing Patterns. Journal of Knowledge and Information System 1(1), 5–32 (1999)
5. Cooley, R., Tan, P.N., Srivastava, J.: Discovery of Interesting Usage Patterns from Web Data, LNCS Vol. In: Masand, B., Spiliopoulou, M. (eds.) Web Usage Analysis and User Profiling. LNCS (LNAI), vol. 1836, pp. 163–182. Springer, Heidelberg (2000)
6. Eick, S.G.: Visual Analysis of Website Browsing Patterns. In: Borner, K., Chen, C. (eds.) Visual Interface to Digital Libraries, pp. 65–77 (2002)
7. Ezeife, C.I., Lu, Y.: Mining Web Log Sequential Patterns with Position Coded Pre-Order Linked WAP-Tree, Data Mining and Knowledge Discovery, 10, 5–38 (2005)
8. Ting, I.H., Kimble, C., Kudenko, D.: Visualising and Classifying the Pattern of User's Browsing Behaviour for Website Design Recommendation. In: Paper presented at the International Workshop on Knowledge Discovery in Data Stream, Pisa, Italy, vol. 24, pp. 101–102 (September 2004)
9. Ting, I.H., Kimble, C., Kudenko, D.: A Pattern Restore Method for Restoring Missing Patterns in Server Side Clickstream Data. In: Zhang, Y., Tanaka, K., Yu, J.X., Wang, S., Li, M. (eds.) APWeb 2005. LNCS, vol. 3399, pp. 501–512. Springer, Heidelberg (2005)
10. Xing, D., Shen, J.: Efficient Data Mining for Web Navigation Patterns. Information and Software Technology 46(1), 55–63 (2004)
11. Yen, S.J., Lee, Y.S.: An Efficient Data Mining Algorithm for Discovering Web Access Patterns, In Zhou, X. In: Zhou, X., Zhang, Y., Orlowska, M.E. (eds.) APWeb 2003. LNCS, vol. 2642, pp. 187–192. Springer, Heidelberg (2003)