Visualizing and Classifying the Pattern of User's Browsing Behavior for Website Design Recommendation

I-Hsien Ting, Chris Kimble, Daniel Kudenko

Department of Computer Science,
The University of York,
Heslington, York, Y010 5DD, UK
{derrick, kimble, kudenko}@cs.york.ac.uk

Abstract. E-commerce website design is a highly complex and subjective task which is often very difficult to get right from the start. Once a design has been implemented, customer behavior usually needs to be monitored in order to correct bugs and inefficiencies. While there are tools available that record and visualize customer click stream data, the information provided by these tools cannot be directly translated into changes in the website design. Rather, an expert needs to go through a lengthy process of manual data analysis in order to identify and overcome weaknesses of websites. In this paper, we describe a click stream visualization technique using footstep graphs. We describe the experiment that produced these graphs and show how specific patterns can be identified that both point to problems with the website design and contain information about how to remedy them. We contend that the results we present in this paper represent the first step towards a technique for producing automatic recommendations for website improvements.

1 Introduction

Click stream data from e-commerce customers is a principal resource for evaluation and analysis of the quality of a website design. Many tools have been proposed in the recent past for the visualization of click stream data [1, 2, 3, 4, 5, 8, 9, 10, 12, 17]. However, these tools only produce (often rather complex) visualizations of the patterns of use and do neither point directly to weaknesses in the website design nor translate directly into the required improvements. Usually, an expert needs to go through an often lengthy and complex data analysis process in order to identify and overcome the problems. This issue is illustrated in Figure 1, which shows the complete process, starting from the initial website design, through web data mining to the actions that modify and improve the website. To date, web-mining systems stop at the pattern discovery and analysis stage; there is very little work on the automation of recommendation and action.

In this paper, we take a first step toward an approach to tackle this issue by proposing a method to extract and visualize the click stream data. We show how specific categories of patterns that contain information about weaknesses in website design and ways to overcome them can be identified. In order to identify these general patterns we recorded user's actions and intentions while performing a set of predefined tasks on four specific websites. This data was used to construct what we have called footstep graphs. Within these graphs we have identified categories of patterns with specific properties that point directly to potential problems with the design of the website. This in turn has resulted in a specific set of recommendations for modifications.

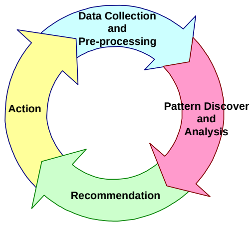

Fig. 1. The KDD process of web usage mining (adopted from [11])

The structure of this paper is as follows. Section 1 describes the background and motivation of this paper; some related literature is discussed in section 2 and section 3 deals with our methodology, the detail of the experiment and the concept of a footstep graph. Section 4 presents and discusses some interesting patterns that were discovered using the footstep graph and section 5 presents some recommendations for changes to the website design. Section 6 concludes the paper and provides a brief outline of future work.

2 A Review of Some Related Work

Visualization techniques are claimed to be the best way to analyze and understand click stream data [15]. Through visualization we can discover the interesting patterns more easily than we could by looking at raw log file data. In addition, there is also the possibility of generating recommendations from these patterns [6].

The most basic way to visualize the user's click stream is using the spanning tree technique to convert a log file into the user's browsing map using tools such as Webmap [5], Naviz [4] and History Graph [7]. Using such tools is easy, however this technology is not robust enough to construct a user's browsing map when the amount of click stream data is too large or complex.

WebQuilt [8] is a tool that uses a proxy server to log the user's click stream. It uses directed graphs to construct a visualization of the user's browsing path. The thickness and the color of the arrows indicate the user's browsing behavior. The thicker arrows denote a more heavily traversed path, and darker arrows mean that more time is spent. Vividence Clickstream [17] and ClickViz [2] use a similar approach to visualize the user's click stream data. Some visualization tools use 3-D or multidimensional graphics, which can incorporate more features in one graph. For example, Disk tree [3], VISIP [4], Parallel Coordinate [9] and Scalable Framework [12] all use this kind of technology.

However, all of these tools focus on showing the user's browsing path and its structure; they do not help to identify trends and identifying the user's browsing history is not very easy. One way to do this is to normalize the user's browsing path based on the browsing time and sequence. Stragrams method [1] is one technique for doing this based on a two-dimensional graph. In the next section, we will introduce another method, which we call a "footstep graph". The footstep graph is not only based on the user's browsing sequence but also the user's browsing time. Both the user's browsing pattern and general trend can easily be elicited using this graph.

3 Method, Experimental Design and Results

In this section, we will describe the concept underlying our methodology and details of our experiment, and discuss the footstep graph technique using some real browsing data from our experiment.

3.1 The concept behind our Methodology

Figure 2 shows the underlying concept of our methodology. The initial experiment is conducted on the client-side where a full and complete set of data can be collected. Our intention is that once an interesting pattern has been discovered on the client side, it will be stored as a general rule-base and will later be matched to patterns discovered in server-side data. Ultimately, the aim is to generate recommendations for a wider range of web sites based on these patterns and the rule-base.

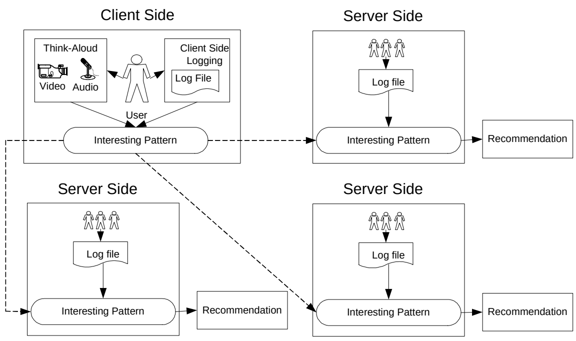

Fig. 2. A conceptual model of our methodology

The user click stream from the initial experiment is logged directly in a log file using a client-side logging agent. The user's thoughts, reactions and intentions are recorded on audio and video. The client-side logging agent records all requests directly without caching so that no data is missed and the user is asked to "think aloud" while they browse the site so that their experience of the site can be captured. Thus, the complete pattern of the user's requests in the client-side log file can be linked to a record of the user's thoughts (as expressed when thinking aloud) and behavior (as captured on video) while they browsed the site. Using this approach, we are able to identify patterns of browsing behavior that can be linked directly to a user's experiences at the time that the pattern occurs [13].

3.2 The Design of the Experiment

We designed an experiment based on the above methodology that presented users with a series of predefined tasks to be completed on a number of predefined web sites. Both the tasks and the sites were selected to be "typical". Four online bookshops were selected and four tasks designed to be repeated at each. A large number of online bookshops were surveyed to find sites with identifiable weaknesses. Then, a series of tasks were designed to highlight these weaknesses. These four bookshops were Blackwell[1], Aarons[2], John Smith & Son[3] and Amazon[4].

The experience of the person conducting the experiment is captured using the "think-aloud" method. The user's click stream is logged to a log file. After pre-processing, the log file is stored in the database with its sequence, and each *URL* that appears in the log file is associated with an index letter.

3.3 Footstep Graphs

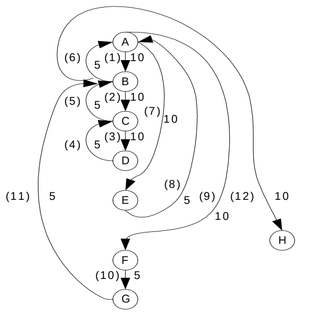

Fig. 3. Complex browsing path in footstep map

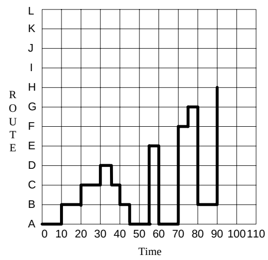

Fig. 4. Data from Fig 3 presented as a footstep graph

As mentioned in Section 2, a tree-like graph (also called "footstep map") is often used to visualize click stream data. It is a directed graph; A→B in this graph means from node A to B, and a letter in a circle means the node. Figure 3 shows an example. The number in parentheses denotes the sequence of the click stream, and the number without parentheses denotes the time spent between two click streams. However, when the user's browsing path becomes too complex, such as the figure 3, the footstep map is difficult to follow. In order to overcome this weakness of the footstep map, we propose another visualization tool, which we call a footstep graph.

The footstep graph is based on a simple 2D plot: the x-axis in this graph represents time and the y-axis represents the nodes on the user's browsing route. Horizontal dis-

---

1 Blackwell http://www.blackwell.com
2 Aarons http://www.aarons-books.co.uk
3 John Smith & Son http://www.johnsmith.co.uk
4 Amazon http://www.amazon.com

tance in this graph represents the time between two nodes, and changes in the vertical axis indicate a transition from one node to another node. Figure 4 is based on the same data as the footstep map shown in figure 3. The user's browsing pattern can now be more easily and quickly be discovered.

4 Some Interesting Patterns

Several interesting patterns were identified using experiment and footstep graph described in the previous section. In this section, we will describe those patterns and relate them to the tester's experience when the patterns occurred.

4.1 The *Upstairs* and *Downstairs* Pattern

The *stairs* pattern is the most general pattern that was found. Two subtypes of the stairs pattern were identified, the *Upstairs* and *Downstairs* patterns. Figure 5 shows the *Upstairs* pattern and figure 6 is the *Downstairs* pattern.

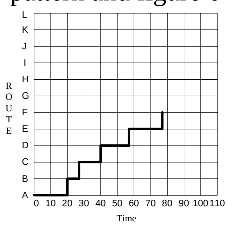

Fig. 5. The *Upstairs* pattern (tester 4 doing task 2 in the Blackwell website)

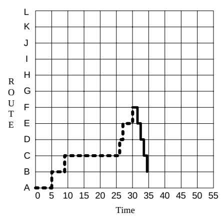

Fig. 6. The *Downstairs* pattern (tester 4 doing task 4 in the John Smith & Son website)

***Upstairs*.** The *Upstairs* pattern is found when the user only moves forward in the website to accomplish the task. For example, if the user's browsing path is A→B→C→D→E→F, then it will be represented as an *Upstairs* pattern in the footstep graph (Figure 5). When the task is completed using this pattern, it usually means that is was "successful", that is, from the tester's real experience, they feel that they have moved smoothly through the website.

However, not all of *Upstairs* patterns are successful. There are two exceptions that indicate some problem for the user when they browse the website. The first is when there are too many stairs in the *Upstairs* pattern. More stairs mean the user needs to complete more steps to finish the task. Therefore, the structure of the website is too complex for the task. The second problem is when the time between two nodes is too long. In our experiment, there were three reasons for long browsing times. (1) Slow searching or page downloading speed, (2) bad page or search result layout means that the user takes a long time to find what he wants and (3) ambiguous link or page content makes the user confused and a long time is needed to make a decision.

Downstairs. The Downstairs pattern is a similar pattern but the user moves backward through the website, that is they return to previously visited pages. For example, if the user's browsing path is F→E→D→C→B, then it will be represented as a Downstairs pattern (In figure 6, follows by the Upstairs pattern). The Downstairs pattern happens after a forward pattern, such as Upstairs, valley or fingers pattern. It is not possible for a Downstairs pattern to exist in isolation in the footstep graph.

In our experiment, we found that the *Downstairs* pattern happened in three situations. The first was when the user realized that there was some kind of problem when browsing the website. However, they did not know what caused the problem and did not know to which page they needed to return in order to solve it. In this situation, they simply used the back button to review a page until they found a suitable restart page. The second situation is similar to the situation above but this time the user knows which page to return to. Unfortunately, the website does not provide the function to return to that page and so they use the back button to get there. However, *Downstairs* patterns do not necessarily reflect a problem with the website, but can sometimes be caused simply by a habit that some users display. Some users by default use the back button to find the page they visited before, they do so even if the website provides a suitable link in the page.

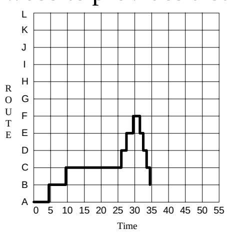

Fig. 7. Mountain pattern (tester 4 doing task 4 in John Smith & Son website)

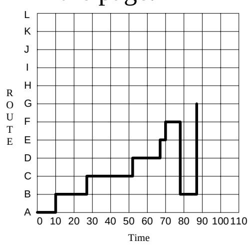

Fig. 8. *Valley* pattern (tester 5 doing task 4 in Aarons website)

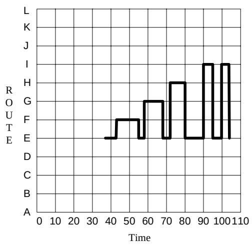

Fig. 9. *Fingers* pattern (Part of tester 1 doing task3 in Aarons website)

4.2 The *Mountain* Pattern

The *Mountain* pattern happens when a *Downstairs* pattern immediately follows an *Upstairs* pattern. Figure 7 is an example of a Mountain pattern in footstep graph. When this happens the conclusions we drew from an *Upstairs* pattern will no longer hold: we cannot say that the *Upstairs* part of the Mountain pattern represents a successful pattern.

In most situations, the *Mountain* pattern is an unsuccessful pattern. The reason is the user spends a lot of time browsing but they end up returning to the original web page without finishing the task. For example, in figure 7, the user spends 35 seconds browsing only to end up back at the second web page without finishing the task. If the user does not give up the task after a *Mountain* pattern has occurred, then other patterns such as *Upstairs*, *Valley* or *Fingers* pattern develop.

4.3 The *Valley* Pattern

The *Valley* pattern occurs after the user goes directly back to a page they visited earlier and then goes to another new page. The further the distance between these two pages the "deeper" the *Valley* pattern. Figure 8 shows an example of *Valley* pattern in a footstep graph.

In our experiments, we found some examples for this kind of pattern. The first one occurred when the current *Valley* was the last *Valley* in a single task and the user finished the task. In this situation, the *Valley* pattern denoted successful browsing. The second occurrence was when a *Downstairs* pattern, *Fingers* pattern or another *Valley* pattern followed the current *Valley* pattern. In this case, the *Valley* pattern indicated a user's mistake that was repeated later.

4.4 The *Fingers* Pattern

A *Fingers* pattern in a footstep graph indicates that a user has fallen into a browsing loop. There are two possible reasons for this. The first is that the user is confused about where he should go, possibly because of an ambiguous statement on the website. The second is that too many mistakes have occurred and the user needs to go back to a page they visited earlier to restart. Figure 9 shows an example of a *Fingers* pattern in a footstep graph.

When this kind of pattern happens, it usually represents unsuccessful browsing. The more fingers in the *Fingers* pattern, the more the pattern is unsuccessful. The slimmer the finger is, the faster the loop repeats itself. Even if a successful pattern (such as *Upstairs* pattern) occurs after the *Fingers* pattern, we still cannot say that this is a successful pattern because the user has already wasted so much time. In the user's experience in our experiment, the *Fingers* pattern was always negative.

5 From Interesting Pattern to Recommendation

In this section we will present recommendations for improving a website design, that are based on patterns that are discovered from a footstep graph.

5.1 Recommendations from *Upstairs* Pattern

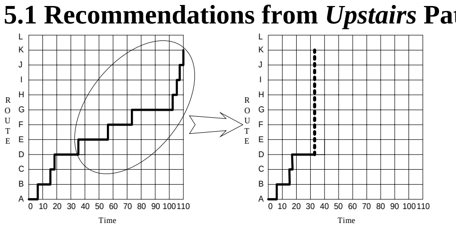

Fig. 10. Smoothing the *Upstairs* pattern 1 (tester 6 doing task 2 in Amazon website)

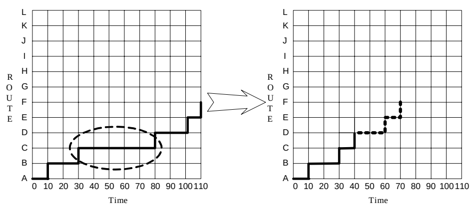

Fig. 11. Smoothing the *Upstairs* pattern 2 (tester 7 doing task 2 in Amazon website)

As discussed in section 4, there are two possible indications that the *Upstairs* pattern is an unsuccessful pattern. If there are too many stairs, the website should try to reduce the steps necessary to accomplish the task (Figure 10). Recommendations that can be provided to the website designer are along the lines of: "Add more related links (shortcuts) to help a user reach the final page and thus reduce the user's browsing steps".

If the user's browsing time in some stairs of the footstep graph is too long, the website should aim to reduce the user's browsing time (Figure 11). Here a suitable recommendation might be "Highlight the link leading to the next page", "Make the web content simpler and clearer" or "Improve the speed of the search function".

5.2 Recommendations from *Mountain* and *Valley* Patterns

Recommendations from the *Mountain* pattern. When a *Mountain* pattern occurs, there are three ways to smooth it (Figure 12). The first is to provide a direct link to the earlier visited page. After deploying this recommendation in the website, a user need not use the page-by-page method to return to an earlier visited page (*Downstairs* pattern 1 in Figure 12).

The second way is to make a recommendation directly to the user when a mistake happens. For example, the website can generate a recommendation such as "Did you mean: Harry Potter" when a user makes a spelling mistake. Then, the user can find the desired page with a few additional steps (modification 2 in Figure 12).

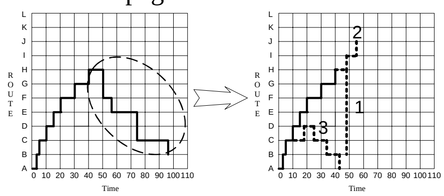

Fig. 12. Smoothing the *Mountain* pattern

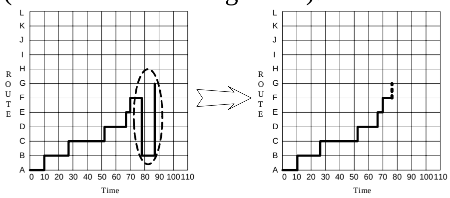

Fig. 13. Smoothing the *Valley* pattern (tester 5 doing task 4 in Aarons website)

The third way is to make the *Mountain* smaller (modification 3 in Figure 12). The website can offer some recommendation or warning such as, "You will go to the checkout following this path" If it is not the user's goal, they will realize the mistake at an earlier stage.

Recommendations from the *Valley* pattern. The reason why the *Valley* pattern happens was discussed in section 4.3. To correct this, the website designer might add some additional links so that the user does not need to go back to a previously visited page to find another page. For example, in figure 13, there is not a direct link from node F to G in the Aarons bookshop's website. Therefore, the tester had to go back to node B to get to node G. If the website designer could add a direct link to node G in node F, then the *Valley* pattern would be smoothened. In this way, the *Valley* pattern is transformed into an *Upstairs* pattern.

5.4 Recommendations from the *Fingers* Pattern

The recommendation based on *Fingers* patterns is similar to the one for *Valley* patterns, as one of the reasons why a *Fingers* pattern occurs is that the website does not provide related links in some pages. Therefore, the user needs to back up to an earlier visited page to find the link to the page they want to go. In this situation (i.e., when the *Fingers* pattern ends in a successful transaction), it is possible to turn the *Fingers* pattern into a successful *Upstairs* pattern (Figure 14). Providing the relevant links in the respective pages (as indicated by the *Fingers* pattern) can realize this.

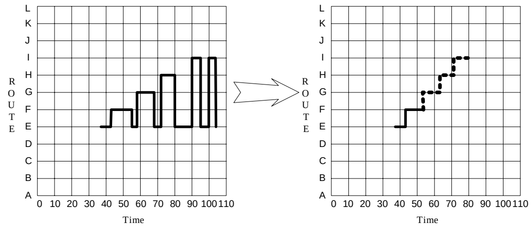

Fig. 14. Smoothing the *Fingers* pattern (Part of tester 1 doing task3 in Aarons website)

However, in most cases the *Fingers* pattern happens because the user falls into a loop, and cannot successfully complete his task. The user has lost his way and he cannot find the information he needs to recover. Therefore, when this kind of pattern happens, the website designer needs to review their website design thoroughly to try to find the source of the problem, and improve the navigation help provided by the website. Some missing link should be added to the web page that is the focus point of the loop. However, normally it is not known exactly which link is missing, since the user's real goal can not necessarily be extracted from the footstep graph.

6 Conclusions and Future Work

In this paper, we introduced the footstep graph for visualizing the user's Clickstream data. We also proposed a research methodology that is based on working from the user's perceptions towards a web usage mining technology for generating recommendation in order to improve the website design. Using a footstep graph, the user's click stream data can be visualized and any interesting pattern can be discovered more easily and quickly than with other visualization tools. Moreover, through the ability to link this to the user's experience, any bugs or inefficiencies in website also can also be identified. Finally we have also classified some of the patterns in our experiment and outlined some appropriate recommendation to remedy the weaknesses of the websites.

In the future, our work will move from client-side logs to server-side logs. This is likely to be a non-trivial move. For example, the *Downstairs* pattern seen in the client-side data might not always be present in server side files as the server and/or the browser may use cached files for backward browsing. Our next research goal is therefore to identify and understand the differences between client-side and server-side data and to find ways of dealing with them. As we indicated in the introduction, our ultimate goal is based around the idea of automatically generating recommendation for improving a websites design, however, there is still much work to be done to achieve this.

References

1. Berendt, B.: Detail and Context in Web Usage Mining: Coarsening and Visualizing, In: Kohavi, R., Masand, B., Spiliopoulou, M. and Srivastava, J. (eds.): WEBKDD 2001 - Mining Web Log Data Across All Customers Touch Points, Lecture Notes in Computer Science, Vol. 2356. Springer-Verlag, Berlin Heidelberg New York (2002) 1-24
2. Brainerd, J. and Becker, B.: Case Study: E-commerce Clickstream Visualization, In Proceedings of the IEEE Symposium on Information Visualization 2001, 22-23 October (2001) 153-156
3. Chi, E. H.: Improving Web Usability Through Visualization, Internet Computing, Vol.6, Issue.2, March-April (2002) 64-71
4. Cugini, J., and Scholtz, J.: VISIP: 3 D Visualization of Paths through Websites, In Proceedings of the International Workshop on Web-Based Information Visualization (WebVis'99), Florence, Italy (1999) 259-263
5. Dömel, P.: WebMap - A Graphical Hypertext Navigation Tool, In Proceedings of The Second International WWW Conference, Chicago, USA (1994)
6. Gaul, W. and Schmidt-Thieme, L.: Recommender Systems Based on Navigation Path Features, In Proceedings of Web Mining Workshop, The Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, USA (2001)
7. Hirsch, F. J., Meeks, S. and Brooks, C. L.: Creating Custom Graphical Web Views Based on User Browsing History, A Poster Presentation at the Sixth International World Wide Web Conference, Santa Clara, California, USA, May 7 (1997)
8. Hong, J. I. and Landay, J. A.: WebQuilt: A Framework for Capturing and Visualizing the Web Experience, In Proceedings of The Tenth International World Wide Web Conference (WWW10), Hong Kong, China, May (2001) 717-724
9. Inselberg, A. and Dimsdale, B.: Parallel coordinates: A tool for visualizing multi-dimensional geometry, In proceedings of Visualization 90, San Francisco, CA, USA (1999) 361-370
10. Keim, D. A.: Information Visualization and Visual Data Mining, IEEE Transactions on Visualization and Computer Graphics, Vol.7, No.1 (2002) 100-107
11. Lee, J., Podlaseck, M., Schonberg, E., Hoch, R.: Visualization and analysis of Click-stream data of online stores for understanding web merchandising, Journal of data mining and knowledge discovery, Vol. 5 (2001) 59-84
12. Lopez, N., Kreuseler and Schumann, H.: A scalable framework for information visualization, Transactions on Visualization and Computer Graphics (2001)
13. Nielsen, J., Farrell, S., Molich, R. and Snyder, C.: E-Commerce User Experience, Nielsen Norman Group (2001)
14. Prasetyo, B., Pramudiono, I., Takahashi, K., Toyoda, M. and Kitsuregawa, M.: Naviz: User Behavior Visualization of Dynamic Page, In Proceedings of 13th Data Engineering Workshop, Kurashiki, Japan, 04-06 March (2002) C5-3
15. Simonson, J., Fuller, G. and Tiwari, A.: A Survey of Web History Data Analysis and Visualization, In http://www.math.grinnell.edu/~lindseyd/ResearchState.html (Access date: 15 May 2004)
16. Usability Testing: Think-Aloud Protocol, User-Centered Information Design Workbook, In http://www.washington.edu/webguides/workbook/ (Access date: 23 April 2004)
17. Vivince. Vivince ClickStreams, In http://www.vividence.com/resources/public/solutions/demo/demo-print.htm (Access date: 16 May 2004)