Visual Web Mining

Amir H. Youssefi
Rensselaer Polytechnic Inst.
Troy, NY 12180, U.S.A.
youssefi@cs.rpi.edu

David J. Duke
University of Leeds
Leeds, LS2 9JT, U.K.
djd@comp.leeds.ac.uk

Mohammed J. Zaki
Rensselaer Polytechnic Inst.
Troy, NY 12180, U.S.A.
zaki@cs.rpi.edu

ABSTRACT

Analysis of web site usage data involves two significant challenges: firstly the volume of data, arising from the growth of the web, and secondly, the structural complexity of web sites. In this paper we apply Data Mining and Information Visualization techniques to the web domain in order to benefit from the power of both human visual perception and computing; we term this Visual Web Mining. In response to the two challenges, we propose a generic framework, where we apply Data Mining techniques to large web data sets and use Information Visualization methods on the results. The goal is to correlate the outcomes of mining Web Usage Logs and the extracted Web Structure by visually superimposing the results. We design several new information visualization diagrams.

1. INTRODUCTION

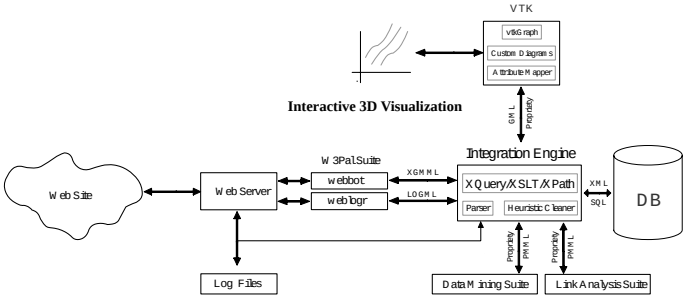

Figure 1: Sample implementation architecture of VWM

We define *Visual Web Mining* (VWM) as application of Information Visualization techniques on results of Web Mining in order to further amplify the perception of extracted patterns and visually explore new ones in web domain.

The *Visual Web Mining Framework*, initially described in [6], provides a prototype implementation for applying information visualization techniques on the results of Data Mining, in this case Spade Sequence Mining Algorithm [7]. The same is applicable to the other related data mining algorithms for mining the websites e.g. Tree Mining and Graph Mining.

2. VISUAL WEB MINING ARCHITECTURE

A robot (webbot [1]) is used to retrieve the pages of the website. In parallel, Web Server Log files are downloaded and processed through a sessionizer and a LOGML [5] file is generated.

Data Mining Suite, *Link Analysis Suite* and *User Profiling/Modeling Suite* need special data formats as input and produce output in propriety formats hence the Integration Engine is required to covert the data.

The *Integration Engine* is a suite of programs for data preparation i.e. extracting, cleaning, transforming, integrating data and finally loading into database and later generating graphs in XGML. The engine uses XQuery, XSLT and Regular Expressions on both standard and propriety data formats.

Much effort is put into enhancing performance of the transformation system in the Integration Engine and the database.

We extract user sessions from web logs, this yields results of roughly related to a specific user. User sessions are then converted into a special format for *Sequence Mining* using cSPADE (continues Spade).

Outputs are frequent *contiguous* sequences with a given minimum support. These are imported into a database, and non-maximal frequent sequences are removed, i.e. we consider only the maximal (based on subsequence relation) frequent contiguous sequences. Later different queries are executed against this data according to some criterion, e.g. *support* of each pattern, length of patterns, etc. We unify different URLs which correspond to the same webpage in the final results.

2.1 Structures

2.1.1 Graphs

We have extended VTK for graph visualization by developing a library called vtkGraph [2]. Two approaches to visualize the output of data mining have been implemented using this library. In the first approach [6] we extract a spanning tree from the site structure, and use this as the framework for presenting access-related results through glyphs and color mapping.

In the second approach, the link analysis is treated as a directed graph. This graph is then laid out, using one of the following two approaches: **A)** a form of Sugiyama layout, described by Auber [1] or **B)** using higher dimensional embedding [3]. Here the graph is first drawn within a higher dimensional space, and then projected into 2 or 3 dimensions via the *Principal Components* of this space.

2.1.2 Stream Tubes

Variable-width tubes showing mined access paths with different traffic are introduced on top of the web graph structure. Here, depending on the type of visualization diagram, particular weights, e.g. support of a single click-stream, total sum of support on all

[1] http://www.w3.org/Robot/

[2] see [2] and http://www.cs.bath.ac.uk/∼djd/graphs.html

WWW2004, May 17–22, 2004, New York, New York, USA.
ACM 1-58113-912-8/04/0005.

click-streams, or the support extracted from Mining algorithm is mapped onto the width (radius) of the stream tube. Color mapping can be used on the number of users leaving the website (or a cluster of these in a zoomed view), a property of the graph structure (such as the Strahler value), or simply the number of hits of some branch.

2.2 Design and Implementation of Diagrams

Figure 2 is a visualization of web usage based on Strahler numbering for assigning colors to the edges. Strahler numbers were originally an attempt to give quantitative information about the complexity or *shape* of a tree. Herman et al [4] were the first to propose using them for graph visualization. They described how these numbers could be used to provide visual cues about tree structure. The Strahler numbers were then generalized further by Auber [1] to provide a means of characterizing the structural complexity of a graph.

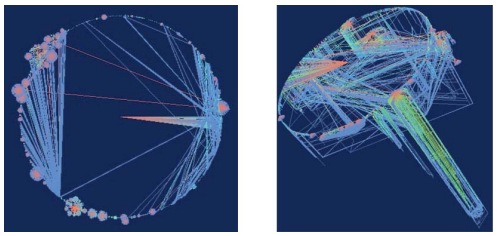

Figure 2: Left: 2D visualization with Strahler Coloring applied on web usage logs. Right: Adding third dimension enables visualization of more information and clarifies user behavior in and between clusters. Center node of circular basement is first page of web site from which users scatter to different clusters of web pages. Color spectrum from Red (entry point into clusters) to Blue (exit points) illustrates behavior of users.

The *cylinder-like* part of figure 2 (right) is visualization of web usage of surfers as they browse a long HTML document converted from LaTeX by LaTeX2HTML software.

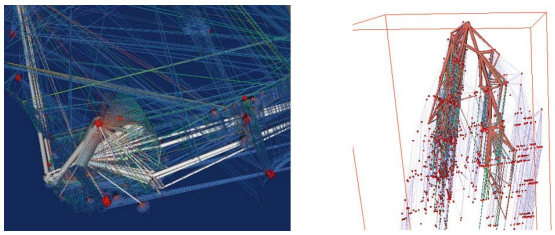

Figure 3: Left: Frequent access patterns extracted by web mining process with a special range of *confidence* values are visually superimposed as a graph of node-tubes (as opposed to node-links). Thickness of the tubes represents aggregated *support* of mined sequences which correnponds to frequency of found patterns. Size of clickable *Cube Glyphs* represents hit of webpages. Right: Superimposition of Web Usage on top of Web Structure with higher order layout. Top node is the first page of website. Hierarchical output of layouts make analysis easier.

In figure 3, combination of a few visualizations techniques facilitates finding new patterns and analyzing the ones already found by data mining algorithms. Our software framework design enables flexible selection of mappings between data attributes and visualization dimensions suitable in different diagrams e.g. confidence-transparency mapping.

Using figure 4, a web analyzer can easily identify which parts of the website are *cold* parts with few hits and which parts are *hot* ones with many hits. This also paves the way for making exploratory changes in website and to analyze the changes in user access. For instance, a webmaster can change link structure (e.g. by adding a hyper-link to a cold cluster from the first page of website or any page in hot clusters) and observe users' navigation paths in the real world.

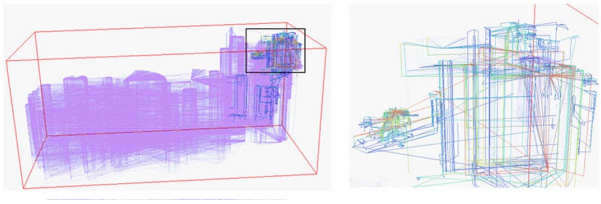

Figure 4: Left: Superimposition of website dynamics(colored) on top of its static structure(gray). Right: Zoom view of colored region with layout of Web Usage taken from Web Graph basement. The basement itself is removed for clarity.

3. CONCLUSIONS

We extend definition of *Visual Web Mining* and design different visualization diagrams, each of which gives a new viewpoint for exploring frequent patterns of user access on a website. Different *visual superimposition* diagrams are of particular interest. We classify webpages into two classes of *hot* and *cold* ones, attracting *high* and *low* number of visitors. A webmaster can make exploratory changes to website structure (e.g. by adding links between hot and cold parts) and analyze the change in user access patterns in the real world.

Given the interactive and graphical nature of our work, we encourage the reader to browse through the figures and download video clips of the system in action from the VWM project page [3].

4. REFERENCES

[1] D. Auber. *Outils de visualisation de larges structures de donnes*. PhD thesis, Universite Bordeaux I, 2002.

[2] D. Duke. Modular techniques in information visualization. In *Proceedings of the 1st Australian Symposium on Information Visualization*, volume 9, pages 11–18, 2001.

[3] D. Harel and Y. Koren. Graph drawing by high dimensional embedding. In *Proceedings Graph Drawing 2002*. Springer Verlag, 2002.

[4] I. Herman, M. Delest, and G. Melancon. Graph visualization and navigation in information visualization: A survey. *IEEE Transactions on Visualization and Computer Graphics*, 6(1), 2000.

[5] J. Punin, M. Krishnamoorthy, and M. J. Zaki. Logml: Log markup language for web usage mining. In *WebKDD Workshop, ACM SIGKDD*, pages 88–112, 2001.

[6] A. H. Youssefi, D. J. Duke, M. J. Zaki, and E. P. Glinert. Toward visual web mining. In *Visual Data Mining Workshop IEEE Int'l Conf. on Data Mining*, 2003.

[7] M. J. Zaki. Spade: An efficient algorithm for mining frequent sequences. *Machine Learning Journal*, 42:31–60, 2001.

[3] http://www.cs.rpi.edu/∼youssefi/research/