ISSN:2348 9510
International Journal Of Core Engineering & Management(IJCEM)
Volume 1, Issue 1, April 2014

Mining Web Logs to Improve Website Organization

Bhavna Thakre

Abstract

Web site design is critical to the success of electronic commerce and digital government. Effective design requires appropriate evaluation methods and measurement metrics. We define Web site navigability as the extent to which a visitor can use a Web site's hyperlink structure to locate target contents successfully in an easy and efficient manner. In this research, we propose an algorithm to automatically find pages in a website whose location is different from where visitors expect to find them. The key insight is that visitors will backtrack if they do not find the information where they expect it: the point from where they backtrack is the expected location for the page. We present an algorithm for discovering such expected locations that can handle page caching by the browser. Expected locations with a significant number of hits are then presented to the website administrator. We also present algorithms for selecting expected locations (for adding navigation links) to optimize the benefit to the website or the visitor. We also include feature that websites without a clear separation of content and navigation, it can be hard to differentiate Between visitors who backtrack because they are browsing a set of target pages, and visitors who Backtrack because they are searching for a single target page. While we have proposed using a time threshold to distinguish between the two activities.

Keywords-Web site design, Web log mining, Web site navigability, page caching.

INTRODUCTION

The structure of a typical Web graph consists of Web pages as nodes, and hyperlinks as edges connecting related pages. Web Structure Mining is the process of discovering structure Information from the Web. This can be further divided into two kinds based on the kind of structure information used.

Hyperlinks: A Hyperlink is a structural unit that connects a location in a Web page to different location, either within the same Web page or on a different web page. A hyperlink that connects to a different part of the same page is called an Intra-Document Hyperlink, and a hyperlink that connects two different pages is called an Inter-Document Hyperlink. Document Structure: In addition, the content within a Web page can also be organized in a tree-structured format, based on the various HTML and XML tags within the page. Mining efforts here have focused on automatically extracting document object model (DOM) structures out of documents.

In general, it is hard to organize a website such that pages are located where visitors expect to find them. This problem occurs across all kinds of websites, including B2C shops, B2B marketplaces, corporate websites and content websites. We propose an algorithm to solve this problem by discovering all pages in a website whose location is different from the location where visitors expect to find them. The key insight is that visitors will backtrack if they do not find the page where they expect it: the point from where they backtrack is the expected location for the page. Expected locations with a significant number of hits are presented to the website administrator for adding navigation links from the expected location to the target page.

We also present algorithms for selecting the set of navigation links to optimize the benefit to the website or the visitor, taking into account that users might try multiple expected locations for a target page.

It also has a module to differentiate between visitors who backtrack because they are browsing a set of target pages or visitors who backtrack because they are searching for a single target page. We have proposed using a time threshold to distinguish between the two activities

ISSN:2348 9510
International Journal Of Core Engineering & Management(IJCEM)
Volume 1, Issue 1, April 2014

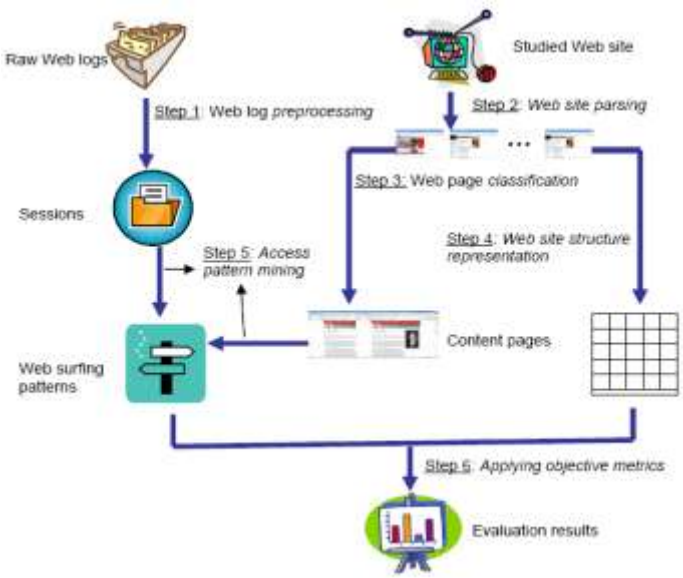


Fig: Web Mining–Based Method for Evaluating Web site Navigability

Literature survey

There has been considerable work on mining web logs however none of them include the idea of using backtracks to find expected locations of web pages.

Perkowitz et al. investigate the problem of index page synthesis, which is the automatic creation of pages that facilitate a visitor's navigation of a website. By analyzing the web log, their cluster mining algorithm finds collections of pages that tend to co-occur in visits and puts them under one topic. They then generate index pages consisting of links to pages pertaining to a particular topic.

Nakayama et al. also try to discover the gap between the website designer's expectations and visitor behavior. Their approach uses the inter-page conceptual relevance to estimate the former, and the inter-page access co-occurrence to estimate the latter. They focus on website design improvement by using multiple regressions to predict hyperlink traversal frequency from page layout features.

Chen et al. present an algorithm for converting the original sequence of log data into a set of maximal forward references and filtering out the effect of some backward references which are mainly made for ease of traveling

Pei et al. propose a novel data structure, called Web access pattern tree for efficient mining of access patterns from pieces of logs. Shahabi et al. [6] capture the client's selected links, page order, page viewing time, and cache references. The information is then utilized by a knowledge discovery technique to cluster visitors with similar interests.

A Web Mining–Based Web site Navigability Evaluation Method

We propose a method for evaluating Web site navigability that builds on the analysis of Web logs and takes into consideration both Web site structure and visitors access/surfing patterns. A Web log refers to a collection of records that objectively document visitors surfing behaviors on a Web site. We focus on prominent, frequent access patterns and extract them from Web logs rather than directly examining all records in logs. Focusing on visitors prominent access patterns is advantageous in several ways. First, infrequent visiting behaviors recorded in log records can distract or even mask analysis results and therefore should be filtered. By concentrating on frequent patterns, we uncover prominent surfing behaviors likely to be observed in future visits. Thus, our analysis of Web site visiting patterns not only reveals the navigability of the design but also points to areas in which the current structure design could be improved. In addition, the prominent surfing patterns extracted can be applied easily to predict the performance of a Web site and identify desirable enhancements to be implemented.

Web site Parsing

A Web site can be parsed by a web spider program that gathers and parses all its pages. Web spiders, or crawlers, are software programs that automatically collect pages from the Web by following hyperlinks exhaustively. In this study, we use Spiders RUs, developed by Chau et al. (2005), to gather Web pages by specifying the URL of the homepage of the studied Web site as the starting URL.

ISSN:2348 9510
International Journal Of Core Engineering & Management(IJCEM)
Volume 1, Issue 1, April 2014

When accessing a page, Spiders RUs downloads the page and extracts its hyperlinks, downloads the pages pointed to by the extracted hyperlinks, and then extracts their hyperlinks. The process continues until Spiders RUs has visited all pages on the Web site. Subsequently each page downloaded by Spiders RUs is parsed to generate useful features from the page, as we detail next.

Web page Classification

Each page downloaded and parsed in Web site parsing is classified as either a content page or an index page. An index page mostly consists of hyperlinks pointing to other Web pages and primarily serves navigational purposes. In contrast, a content page typically contains textual contents and other information potentially of interest to visitors; thus, content pages are generally the pages for which visitors search. We use an automatic classifier to differentiate index and content pages. Specifically, we use a classifier built upon Support Vector Machine (SVM), 2 a common computational method for searching for a hyper plane that best separates two classes. We choose SVM primarily because of its effectiveness in text-based classification tasks.

Our goal is to classify Web pages on the basis of their structure rather than their content; therefore, we take a feature-based classification approach (Chau 2004) instead of a conventional keyword-based approach. Based on our observations and a review of related research, we select six fundamental page structure features pertaining to size and number of links: (1) number of outgoing links, (2) number of internal out links (links pointing to other pages in the same domain), (3) number of external out links (links pointing to other pages not in the same domain), (4) length of the document, (5) number of words in the document, and (6) number of anchor text words in the document. about Web site design and administration, Each downloaded page possesses a specific value for each feature. We randomly selected a set of downloaded pages as the training sample and asked two domain experts, highly experienced in and knowledgeable to classify them manually. These "manually tagged" pages then enable us to construct a SVM model.

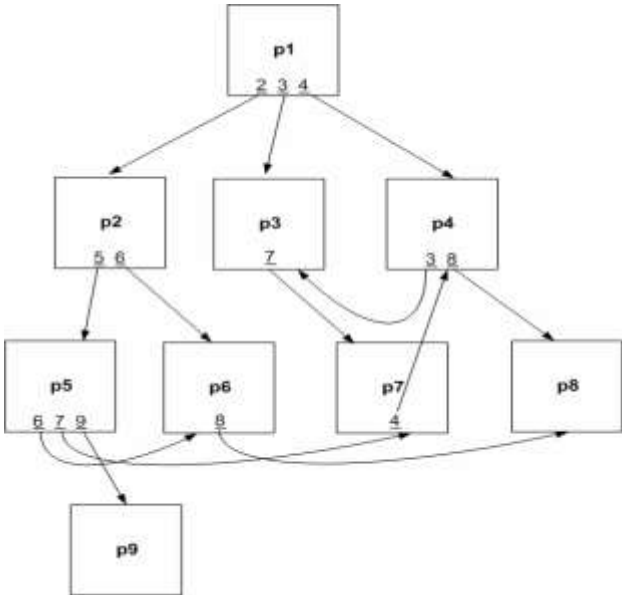


Fig: Representation and Analysis of Web site Structure:

Web site Structure Representation

We represent the navigational structure of a Web site using a distance matrix of the pages on the Web site. The indexes of the matrix are Web pages, wherein the elements of the matrix denote the distance between any two pages. We consider a Web site a directed graph and denote pages as vertices and hyperlinks as edges. We measure the distance between two pages with the distance definition in graph theory. Specifically, the distance from vertex *u* to vertex *v* in a graph is the length of the shortest path from *u* to *v*. Accordingly, the distance from page 1 *p* to page 2 *p* on a Web site can be measured by the length of the shortest path from 1 *p* to 2 *p*, or the number of hyperlinks surfed or number of clicks required. When no path connects page 1 *p* to page 2p, the distance from 1 *p* to 2 *p* is considered ∞. For a Web page *p*, we represent the set of Web pages pointed to by the hyperlinks on page *p* as *t*(*p*) . In this case, we can obtain *t*(*p*) by parsing *p*. We denote *D*(*p*,*l*) as a set of Web

ISSN:2348 9510
International Journal Of Core Engineering & Management(IJCEM)
Volume 1, Issue 1, April 2014

pages for which the distance from page p to another page in $D(p,l)$ is l clicks ($l = 1,2,L$).

An Algorithm for Finding $D(p,l)$

Input: a set of Web pages in a Web site.
Output: $D(p,l)$ for each Web page p .
for each Web page p ,
finding $t(p)$ by parsing Web page p
end for
for each Web page p
$i = 1$
$D(p,1) = t(p)$
while $D(p,i) \neq \varphi$
$D(p,i+1) = \varphi$
for each $k \in D(p,i)$
for each $h \in t(k)$
if $h \notin D(D(p,j))$ for all $j \leq i$
$D(p,i+1) = D(p,i+1) \cup \{h\}$
end if
end for
end for
$i = i + 1$
end while
end for

Optimizing the Set Of Navigation Links

We consider three approaches for recommending additional links to the web site administrator (or automatically adding links):

1. First Only: Recommend all the pages whose frequency of occurrence in the first expected location is above an administrator specified threshold.

2. Optimize Benefit: Recommend the set of pages that optimize benefit to the website, where benefit is estimated based on the fraction of people who might give up on not finding a page.

3. Optimize Time: Recommend the set of pages that minimize the number of times the visitor has to backtrack, i.e., the number of times the visitor does not find the page in an expected location.

Finding Expected Locations

Single Target: Consider the case where the visitor is looking for a single specific target page T. We expect the visitor to execute the following search strategy:

1. Start from the root.
2. While (current location C is not the target page T) do

(a) If any of the links from seem likely to lead to T, follow the link that appears most likely to lead to .
(b) Else, either backtrack and go to the parent of C with some (unknown) probability, or give up with some probability.

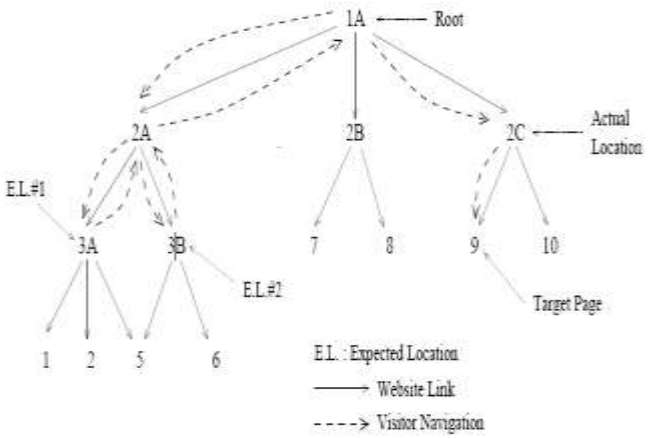
When the visitor reaches the same page again in step 2(a), she will follow a different link since the estimates of whether a link is likely to lead to will have been updated.

Set of Targets: Now consider the scenario where the visitor wants to find a set of target pages T1;2:;Tn. The search pattern is similar, except that after finding (or giving up on) Ti, the visitor then starts looking for Ti+:

1. For i:= 1 to n
(a) If =1, start from the root, else from the current location C.

(b) While (current location C is not the target page Ti) do If any of the links from seem likely to lead to Ti, follow the link that appears most likely to lead to. Else, either backtrack and go to the parent of C with some probability, or give up on Ti and start looking for Ti+1 at step 1(a) with some probability. In this scenario, it may be hard to distinguish the target pages from the other pages by simply looking at the web log. We discuss this issue after first giving an example.

ISSN:2348 9510
International Journal Of Core Engineering & Management(IJCEM)
Volume 1, Issue 1, April 2014



Website & Search Pattern

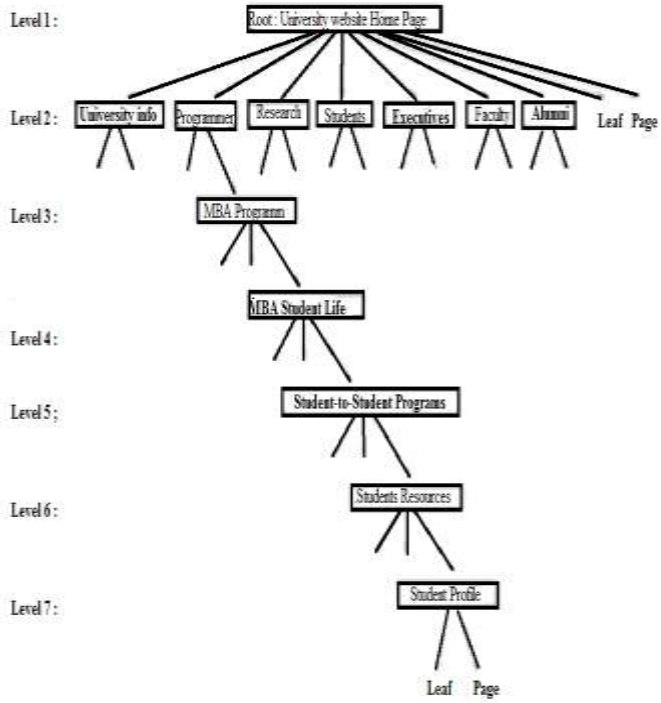
If there is no browser caching, it is conceptually trivial to find a backtrack point: it is simply the page where the previous and next pages in the web log (for this visitor) are the same. The HTTP standard states that the browser should not request the page again when using the browser's history mechanism. In practice, some browsers use the cached page when the visitor hits the "back" button, while use the cached page when the visitor hits the "back" button, while others incorrectly request the page again. It is possible to disable caching by setting an expiration date (in the Meta tag in the page), but this can significantly increase the load on website.

Rather than rely on disabling browser caching, we use the fact that if there is no link between pages P1and 2, the visitor must have hit the "back" button in the browser to go from P1to 2. Thus to find backtrack points, we need to check if there is a link between two successive pages in the web log. We build a hash table of three edges in the website to efficiently check if there is a link from one page to another

Experiment

We used the web server log from an university

to evaluate our algorithm. fig shows the structure of the website. There are 7 levels in this website hierarchy. Starting with the root as level 1, there are 7 directories (interior nodes in the hierarchy) in level 2, 20 directories in level 3, 21 directories in level 4, 13 directories in level 5, 2 directories in level 6 and 2 directories in level 7.



Conclusions

The current research examines Web site navigability with a particular focus on the structural aspect of Web site design. We proposed a novel algorithm to automatically discover pages in a website whose location is different from where visitors expect to find them. Our key insight is that visitors will backtrack if they do

Not find information where they expect it. The point from where they backtrack is the expected locations for the page.

We presented an algorithm for discovering such backtracks that also handles browser caching. We contribute to web metrics research. Most prior research on web metrics (e.g. Dhyani et al. 2002; Zhang et al. 2004) attempts to evaluate Web site structure design using a Web site's structural information and typically considers all the pages on a Web site equally important. In response, we integrate content, structure, and usage information to evaluate Web site

ISSN:2348 9510
International Journal Of Core Engineering & Management(IJCEM)
Volume 1, Issue 1, April 2014

navigability in an innovative and comprehensive fashion. cost effective (greatly reducing data collection costs), efficient (producing evaluation results in minutes), and flexible (offering baseline measurements that can be extended or refined).

References

[1] T. Nakayama,H. Kato, and Y. Yamane. Discovering the gap between web site designers' expectations and users' behavior.
In *Proc. of the Ninth Int'l World Wide Web Conference*, Amsterdam, May 2000.

[2] J. Pei, J. Han, B. Mortazavi-asl, and H. Zhu. Mining access patterns efficiently from web logs. In *Proc. of the 4th Pacific-Asia Conf. on Knowledge Discovery and Data Mining*, pages 396–407,April 2000.

[3] C. Shahabi, A. M. Zarkesh, J. Abidi, and V. Shah. Knowledge discovery from users web-page naviagtion. In *Proc. of the 7th*
IEEE Intl. Workshop on Research Issues in Data Engineering (RIDE), pages 20–29, 1997. [7] M. Spiliopoulou and L. C. Faulstich.Wum: A web utilization miner. In *Proc. of EDBT WorkshopWebDB98*,Valencia, Spain, March 1998
.
[4] Chau, M. "Applying Web Analysis in Web page Filtering," in *Proceedings of the ACM/IEEE-CS Joint Conference on Digital Libraries*, Tucson, Arizona, USA, June 7-11, 2004, p. 376.

[5] Chau, M., Qin, J., Zhou, Y., Tseng, C., and Chen, H. "SpidersRUs: Automated Development of Vertical Search Engines in Different Domains and Languages," in *Proceedings of the ACM/IEEE-CS Joint Conference on Digital Libraries*, Denver, Colorado, USA, June 7-11, 2005, pp. 110-111. Cooley, R., Mobasher, B., and Srivastava, J. "Data preparation for mining World Wide Web browsing patterns".

[6] Knowledge and Information Systems 1:1, 1999, pp. 1-27. Dhyani, D., Ng, W. K., and Bhowmick, S. S. "A survey of web metrics," *ACM Computing Survey* 34 (4), 2002, pp 469-503.

[7] Information Systems, 3, June 2000.
Hevner, A., March, S., Park, J., and Ram, S., "Design science in information systems research," *MIS Quarterly*, 28(1), 2004, pp. 75-105.

[8] West, D. B. *Introduction to Graph Theory*, 2nd edition. Prentice Hall, 2001. Yang, Y. and Liu, X. "A Re-examination of Text Categorization Methods," *Proceedings of the ACM Conference on Research and Development in Information Retrieval (SIGIR'99)*, 1999, pp. 42-49.Zhang, Y., Zhu, H., and Greenwood, S., "Web site complexity metrics for measuring navigability," *Proceedings of the Fourth International Conference on Quality Software*, 2004, pp. 172-179.

[8] web mining–based objective metrics for measuring web site navigability*"Design Science Track* Xiao Fang* College of Business Administration University of Toledo
Toledo, Ohio Michael Chau* School of Business
University of Hong Kong Pokfulam, Hong Kong

[9] Mining Web Logs to Improve Website Organization "Ramakrishnan Srikant
IBM Almaden Research Center Yinghui Yang
& Dept. of Operations & Information Management Wharton Business School University of Pennsylvania

Bhavna Thakre

Computer Science & Engineering

Laxmi Narayan College Of Technology

Indore, India

E mail-bhavnathakre.it@gmail.com