

Enhanced Algorithm for Mining the Frequently Visited Page Groups

Yiling Yang, Xudong Guan, Jinyuan You

Dept. of Computer Sci. & Eng., Shanghai Jiaotong University, China, 200030

{y.yl, xdguan}@263.net, you-jy@cs.sjtu.edu.cn

Abstract

Web page grouping based on user's navigation patterns can help web masters with site organization and content arrangement. Traditional data mining techniques like association rule and clustering can be applied with little changes to the user sessions extracted from web server's log files, however, the result often comes out poor. In this paper, an enhanced algorithm for mining the frequently visited groups of web pages is proposed with the consideration of both the site topology and the content of web pages. As for the calculation of the support value of a page group, those earlier algorithms take into account only the number of occurrences of a page group in the user sessions. In this proposed algorithm, such calculation of support includes another two arguments, the content-link ratio of the web page and the inter-linked degree of the page group. The enhanced algorithm pays more attention to the relationship between the content pages. Hence, the frequently visited page groups with more interestingness are rendered.

1. Introduction

With the great development of the WWW, many web-based organizations try to re-arrange their web sites in order to facilitate their customers. Web site designing is becoming harder because of the increase of the scale and complexity of the sites. Straightforward usage statistics are no longer adequate for designers to optimize the site structure. Web usage mining, the application of data mining techniques to the web server's log files, is a promising solution to help the administrators to understand more in-depth usage information of their web sites.

An important topic in the web usage mining is the grouping of the web pages, i.e. grouping the pages into classes based on the site visitors' browsing behaviors. By analyzing these groups, the webmaster may reorganize the site to facilitate its users.

In this paper, the grouping of web pages based on both the users' navigation and the site topology is studied. A

key point of the paper is that a frequently visited page group is less interesting if the pages in the group are inter-linked to a high degree. For example, page A, B, and C are not inter-linked, but they are often accessed together by a number of users. If they are adjusted to be inter-linked, the users will easily find what they need.

We can not apply data mining algorithms directly to the raw web server logs because the log files don't represent a user's browsing behavior reliably for a number of reasons [1, 2]. In our approach, the log files are first processed to identify the user sessions. A user session is considered a set of the page-requests that occurred during a single visit to a web site. The process of identifying the user sessions from the log files, including data cleaning, user identification, and session identification, is described in detail in [1]. After this pre-processing phase, the grouping algorithm can be applied to the identified user sessions to find the frequently visited page groups.

2. Related Work

There are several commercially web log analyzers [4, 5] that provide primitive mechanisms for reporting the user activities, such as the times of visits and the URLs of the users. However, these tools usually provide little information of the relationships among the requested files.

There are also some other analysis tools for the web log, for example, WEBMINER [3] and SpeedTracer [2], that provide more sophisticated analysis, such as association rules, sequential patterns, and clustering of pages. However, the mining results of these tools are considered less interesting in this paper. For instance, assume a web site has heavy requests on these five pages: default.html, news_index.html, news1.html, news2.html, and news3.html and the site topology is shown in Figure 1.

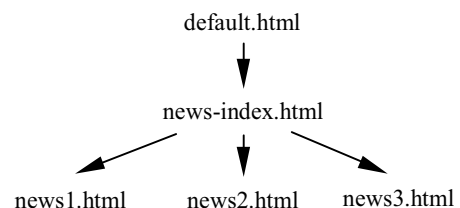


Figure 1. The topology of five web pages

Among these pages, the default.html is the home page of the site, and the news_index.html is a navigation page to guide the users to the news pages. The other three news pages are often called content pages. Assume that we apply the page grouping algorithm in SpeedTracer to the log file and the result may be {default.html, news_index.html, news1.html} and {news_index.html, news1.html, news2.html}, etc. In fact, the first group of the result is not "interesting" for the reason that the three pages in it have been inter-linked in the site topology. Moreover, the home page and the navigation page often have more requests than the other three content pages. So the true interesting group, {news1.html, new2.html, new3.html}, often hides deeply in the computed results or even gets lost when the request numbers differ a lot.

It's difficult to discover the interesting page groups only through the log files because the information that the log files can provide is limited. In this paper, two factors, the content of the web pages and the site topology are used to assist the mining algorithm. The premise is that the site topology and page content are obtained firstly. This can be done through URL crawling and page analysis.

3. The Enhanced Algorithm

Before going to the detail of the algorithm, we first introduce two important arguments, the normalized content-link ratio (NCLR) and the group inter-link degree (GILD).

3.1. Normalized Content-Link Ratio

One of the differences between the web pages and the database records is that the web pages have various formats and carry different types of contents. According to the purpose and the content of a page, we can classified the web pages into five main types:

1. **Home page** - a page that is usually visited by the users as the first page.
2. **Content page** - a page that contains various information provided by the web site.
3. **Navigation page** - a page that has many links to guide the users to the content pages.
4. **Search page** - a page that provides key words searching.
5. **Personal page** - a page that presents information of a biographical or personal nature.

Due to the nature of the personal pages, it is ignored in the approach. One of our goals is to find out the frequently visited page groups that contain as many as possible content pages. In order to emphasize the content pages during the process of page grouping, the other three types of web pages should have fewer weights in the mining algorithm. For example, the weight of the home

pages may be smaller than that of the content pages because we pay more attention to the relationships between the content pages. We define *Content-Link Ratio* (CLR) of each web page as the ratio of the web page size to the page links. For instance, if a web page has 3 links and it's size is 4k bytes, the CLR of this page is 4/3 (if the unit of page size is set to be 1k bytes).

The CLR value of any given web page is greater than zero and less than infinity. We map CLR to a floating-point NCLR between 0 and 1 so that the support values of page groups distribute in well proportion and hence the threshold can be set effectively. The definition of *Normalized Content-Link Ratio* (NCLR) of a web page is:

$$NCLR = (1 - e^{-CLR})$$

With this definition, a page with small content size and large number of link will have a small NCLR value, and vice versa. One exception in the definition is, if the page doesn't have any outgoing link at all, the CLR will be invalid, but we can define NCLR of that page to be 1. As the result, the range of NCLR is (0,1].

3.2. Group Inter-Link Degree

A good mining algorithm should find the page groups that the users are mostly interested in and those pages in the same group should not already be inter-linked to a high degree. To achieve this, we introduce the *Group Inter-Link Degree* (GILD) of a page group. By analyzing the site topology, we can easily know whether there are links from any given page A to page B. From this, a page group G can be taken as a directed graph — Graph(G), with pages as nodes and links as directed edges. Based on this, we define GILD of a group G as follows:

$$GILD(G) = \begin{cases} N(G) / (|G| * (|G| - 1)) & , |G| > 1 \\ 0 & , |G| = 1 \end{cases}$$

N(G) - number of edges in Graph(G)
|G| - number of pages in group G

The GILD value of a group is 0 when there is no link between any two pages in the group. The GILD value is 1 when the pages in the group are fully inter-linked, which means that there is no need to present such group in the final result no matter how frequently the pages in it have been requested together. The above definition is not a precise measure of the actual link degree, since the distribution of the edges in the group is not taken into account. But this definition is more practical in that GILD can be easily computed. The GILD measure used in finding the interested page groups do not need to be very precise since it will consumes much more time to compute than necessary if we use more complicated measures, like the number of sub-graphs in Graph(G), in the definition of GILD. In the implementation, only the unique ID of

source and destination URL of every link are recorded into the database through the page analyzer. This method can reduce the demand of memory space for keeping the directed graph.

3.3. The Enhanced Algorithm

The approach to mining the frequently visited page groups is similar to the discovery of the large itemsets in mining association rules. To mine the frequently visited page groups from the user sessions, only the distinct pages are needed in each session. Any duplication of pages caused by backward traversals was first eliminated in each session. We assume that FG_k is the set of frequently visited page groups, each consisting of k pages, and the *support* value of every group in FG_k is greater than a predefined threshold T . The support in the traditional algorithm denotes the number of the user sessions containing all the pages in a group [6]. We apply the algorithm to the user sessions by extending the meaning of the support. The support of a group G in our enhanced algorithm is defined as:

$$\text{Support}(G) = O(G) * H_{\text{NCLR}}(G) * (1 - \text{GILD}(G))$$

- $O(G)$ – The number of user sessions containing all the pages in G
- $H_{\text{NCLR}}(G)$ – The harmonic mean NCLR of all the pages in G
- $\text{GILD}(G)$ – The group inter-link degree of G

One point to be explained is why we choose the harmonic mean NCLR in the above definition. Although the other means could also be adopted, but the harmonic means can balance the distribution of NCLR in the group better than the other means.

The process of finding the frequently visited groups is an iterative method. First, we find out the FG_1 that is the top visited pages with support exceeding T . Next, the FG_2 is generated based on the FG_1 and FG_3 is based on the FG_2 , and so forth.

```

1:count the NCLR of all distinct pages appeared;
2:initialize  $FG_1$  as the top requested single
   page groups with Support  $\geq T$ ;
3:for (i=2;i<=k;i++) {
4: Sort the pages of groups in  $FG_{i-1}$  in lexicographical order;
5: for each group  $\{x_1, \dots, x_{i-1}\}$  in  $FG_{i-1}$  {
6: for each group  $\{y_1, \dots, y_{i-1}\}$  in  $FG_{i-1}$  {
7: if ( $x_2=y_1$  and ...and  $x_{i-1}=y_{i-2}$ ) {
8: construct a new group  $G=\{x_1, \dots, x_{i-1}, y_{i-1}\}$ ;
9: if ( $G$  not already in  $FG_i$ ) {
10: test all other combinations of subgroups of  $G$ 
    with size (i-1);
11: if (all such subgroups are in  $FG_{i-1}$ )
12: if (Support( $G$ )  $\geq T$ ) add  $G$  into  $FG_i$ ;
13: }
14: }

```

```

15: }
16: }
17:}

```

Figure 2. Enhanced algorithm for mining the frequently visited page groups

The enhanced algorithm need compute two extra parameters, NCLR and GILD, compared with the normal algorithm. Since the NCLR and the source and destination URL of every link of the web pages have been recorded into the database before starting the mining algorithm, the only additional computation during the mining algorithm is for the support value of each page group. Moreover, the new algorithm will prune the uninterested groups in advance that would keep iterating in the normal algorithm. This saves lots of memory space and CPU time. Hence, the introduction of NCLR and GILD will not reduce the performance of the enhanced algorithm heavily.

4 Experiment

The enhanced algorithm has been implemented and tested on a data set collected from the SJTU's web server log (<http://www.sjtu.edu.cn>). The experiments are carried out on a Pentium III 450 with 128M RAM running Windows 98.

We test the algorithm on a subset of the log file, which included 100,000 records with a total size of 9MB. The log contains 417 distinct web pages and the total number of the user sessions is 1902. We compared the normal algorithm (which define $\text{Support}(G)$ as $O(G)$) with the enhanced algorithm using different thresholds as below. The number of page groups in each FG_i is shown in Table 1.

Table 1. Experiment results

Method	T	$ FG_1 $	$ FG_2 $	$ FG_3 $	$ FG_4 $	$ FG_5 $	$ FG_6 $	$ FG_7 $
Normal	50	29	87	126	97	36	5*	0
Enhanced	30	29	16	1 ⁺	0	0	0	0
Enhanced	15	49	71	24	1 ⁺⁺	0	0	0
Enhanced	9	68	128	109	36	2 ⁺⁺⁺	0	0

Note: $|FG_i|$ - Number of groups in FG_i

*: The 5 groups here all contains: a). The same three uninteresting pages: home page ($/'$), Chinese version home page ($/'chinese/index.htm'$), and a navigation page ($/'chinese/Navigate.htm'$); and b). Other three a little more interesting pages.

⁺, ⁺⁺, ⁺⁺⁺: The three groups here should have contained what we are interested in, the reason why not is discussed below.

The above results show that the computation using the enhanced algorithm shrinks much more rapidly, while the original $|FG_1|$ is the same. In addition, by decreasing the threshold value, the enhanced algorithm can reveal some

hidden groups (like $+$ and $++$), which the normal algorithm can not.

Yet, there are still some flaws in the algorithm. One is due to the introduction of scripts in the web pages. In order to make the web site more adaptive, a page may simply be a link to another URL, by including a section of JavaScript in its body, like:

```
<script language="javascript">
  window.location="new URL"
</script>
```

If no special measures are taken, the algorithm will calculate the NCLR and GILD arguments by parsing the content of the original page. In many cases, the original page only contains pure HTML script, and the link numbers we collect is 0. So the NCLR will be 1 and GILD of the group containing this page will be low. As a result, this page is very likely to appear in the final result, providing that its occurrence in the user sessions is high. In the above table, group $++$ contains four pages that are all pure HTML script pages used to point to other navigation pages in another web server (www2.sjtu.edu.cn).

It seems that this problem can be solved by a script analyzer that can recognize these jumps and replace the original script page with the destination page in the following computations. Yet the scripts are not like static web pages, they may contain condition statements and decide their behavior through the inputs from the page viewers. For example, there are several candidate pages for a script to load. The script can decide which page to be loaded by checking the type of browsers the user is currently using. It's beyond the log analyzer's ability to solve this problem. However, fortunately, most of these pages are very simple; they only contain a single statement carrying the new URL. If these simple pages can be treated correctly, the mining algorithm will be excellent, if not perfect.

Another problem is how to treat the dynamic generated pages. If we totally ignore all URLs that contain the question marks '?' that indicate the URLs are dynamic pages, we will lose some interesting rules, like the grouping of the news pages, which are most likely to be fetched from the database. On the other hand, if we trace all dynamic generated pages, it will take too much time but contribute very little to the result.

All these problems need to be solved not only in the domain of web usage mining, but also in all other application domains that need to take the content of the

web page into account. How to eliminate or reduce the effects of these factors are for the future work.

5 Conclusions and Future Work

The grouping of web pages based on the user sessions has been studied. An enhanced grouping algorithm is proposed which considers both the site topology and the content of web page. The support value of a group changes with the content-link ratio of web page and the inter-linked degree of the group, which makes us pay more attention to the relationship between the content pages. In our experiment, the enhanced algorithm performed much well than the normal.

The future work is concentrated on grouping the users and providing the customized site. Furthermore, we are trying to combine the web usage mining with other web-based techniques, such as e-commerce, to make web sites more manageable and smart.

Acknowledgement

We thank Chang T. Zhang of University of New Hampshire for his good suggestions to this paper.

Reference

- [1] R. Cooley and J. Srivastava, "Data preparation for mining world wide web browsing patterns", *Journal of Knowledge and Information Systems*, 1999, 1(1).
- [2] K.L. Wu, P.S. Yu, and A. Ballman, "SpeedTracer: a web usage mining and analysis tool", *IBM System Journal*, 1998, 37(1), pp. 89~105.
- [3] B. Mobasher, N. Jain, E-H. Han, and J. Srivastava, "Web mining: pattern discovery and from world wide web transactions", Technical Report 96-050, University of Minnesota, Sep, 1996.
- [4] <http://www.webtrends.com>
- [5] <http://www.marketwave.com>
- [6] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules", *Proc. of the 20th VLDB conference*, pp. 487-499, Santiago, Chile, 1994.
- [7] Y.L. Yang, "Data preparation and browsing patterns recognition in web log mining", Master Degree Thesis, Xi'an Jiaotong Univ., Mar, 1999.
- [8] Y.L. Yang, X.D. Guan, L.N. Lu, J.Y. You, "SWLMS: A Web Log Mining System", *Journal of Shanghai Jiaotong Univ.(Chinese Edition)*, to be appeared.