Volume 5, Issue 5, May 2015 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Web Mining Methods to Improve Weblog Analysis

Deepti Juneja Thakral
Research Scholar
Jagannath University, Jaipur,
Rajasthan, India

Abstract - Most of the websites have a hierarchical organization of content. This organization may be quite different from the organization expected by visitors to the website and sometime it is unclear where a specific document is located. There are many algorithms that automatically search pages in a website whose location is different from where visitors expect to find them. In this paper an algorithm is present for discovering such expected locations. Expected locations with a significant number of hits are then presented to the website administrator. We also present an algorithm for selecting expected locations (for adding navigation links) to optimize the benefit to the website or the visitor. Beside the structure of the website, users' preference to target pages is another key factor for analyzing the location or node importance. Clearly a specific document which is visited frequently or where users stay for a long while indicates that it has a higher degree of preference. This paper introduces the duration's as a weight of the node to measure the preference.

Keyword: Backtracking, milestone coefficient, Expected Location, web log mining, node.

I. INTRODUCTION

The evolution of the Internet has lead to an enormous propagation of the available information and the personalization of this information space has become a necessity. The knowledge obtained by learning web user's preferences can be used to improve the effectiveness of their web sites by adapting the web information structure to the users' requirement. Automatic knowledge extraction from web log files can be useful for identifying such reading patterns.
However it is hard to find appropriate tools for analyzing raw web log data to retrieve significant and useful information. Recently, the advantages of data mining techniques for discovering usage patterns from web data (i.e. web log mining or web usage mining) made it possible to mine typical user profiles from the vast amount of access logs. Web usage mining can be viewed as the extraction of usage patterns from access log data containing the behavior characteristics of users.

II. OPTIMIZING THE SET OF NAVIGATION LINKS

The proposed algorithm is used for finding user's pattern. Before applying this process on the weblogs, preprocessing on weblogs is required for removing the redundant data logs and other non beneficial information for finding the required patterns. So the first preprocessing is done on weblogs.

A. Preprocessing web logs

Preprocessing is done by using the following steps:
1) Data Cleaning: First step in Preprocessing is data cleaning which is used to remove the trashy entries. The following process is used to remove trashy entries:

Algorithm: Data Cleaning

```
For each transaction,
        If transaction T contain any ("404 (not found)","*.css","*.gif"," other trashy entries") Then
                Remove transaction T
        End If
End For
```

2) Relocate the data: After removing the trashy transactions this step is used to rearrange the data according to user IP addresses to form each user's request cluster.
3) Identifying Target Pages: Next step is to identify target pages. If **t**here is a clear separation between content pages and index (or navigation) pages on website then web network topology is used to find out the target pages while if website not have a clear separation between content and index pages, we can use a time threshold to distinguish whether or not a page is a target page. Pages where the visitor spent more time than the threshold are considered target pages. For identifying target pages we can also combine these two methods for websites with hybrid (structured + unstructured) structure.

4) Find Expected Location: 1. For each visitor, partition web log such that each subsequence terminates in a target page.
2. For each visitor and target page, find any expected locations for that target page:

Let $\{P_1, P_2, ..., P_n\}$ be the set of visited pages,
where P_n is a target page.
Let $B := \phi$ denote the list of backtrack pages.
a) for $i := 2$ to n-2 begin
b) if (($P_{i-1} = P_{i+1}$) or (no link from P_i to P_{i+1}))
c) Add P_i to B. // P_i is a backtrack point.
end
if (B not empty)
Add <P_n, B, P_{n-1}> to <current URL, backtrack list, Actual Location> table

B. Algorithm: Optimizing the set of Navigation Links

The proposed algorithm is used for finding the optimized set of navigation links. Below is the procedure for it:

Algorithm: Optimizing the set of Navigation Links

```
Let L be the list of set of pages recommended by the Optimize Time (explained later)
  For each page p in L
     Calculate milestone coefficient M (p)
     (explained later)
  End for
Sort list L according to milestone coefficient M
```

The page top in the list is milestone node (page) and the one most recommended. This would be the geographical and indication node of entire website with features of high connectivity to other pages and higher level of user preference.

1) Optimize Time: This algorithm recommends 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. The following process is used for this:

Algorithm: Optimize Time

```
Repeat
  For each record begin
    Let m be the number of expected locations in this record.
    For j := 1 to m
      Increment support of value(CE_j) by m+1-j.
  end
  Sort pages by support.
  P := Page with highest support (break ties
       at random).
  If (support (P) >= S_t) begin
    Add <P, support (P)> to list of
    recommended pages.
    For each record begin
      For k: = 1 to n begins
        If value (CE_k) = P
          Set CE_k, CE_k+1, ..., CE_n to null;
       End
     End
```

2) Milestone coefficient: Milestone node is the geographical and indication node of entire website with features of high connectivity to other pages and high level of user preference.

Milestone coefficient, defines the importance of the nodes, expressed as

$$\mathbf{M=R_c * W_c + R_d * Wd + R_{t(k)} * Wt}$$

Where W_c stands for connectivity weight, W_d stands for depth weight and Wt stands for preference weight and $W_c + W_d + Wt = 1$

And R_c, R_d, $R_{t(k)}$ is Relative connectivity, Relative depth, and Node preference respectively. These parameters require information related to the website structure.

3) Create the website structure: This is an important step which create the hierarchal (tree) structure of the website so that connectivity of each node can be calculated.

4) Relative connectivity: The in-degree of a node is the number of nodes coming to node in question while the out-degree of a node is the number of nodes coming from node in question.

So the connectivity of a node will be represented by
C=I (in-degree) +O (out-degree) and
Relative connectivity is calculated as
Rc = C/Tc
Where Tc is the sum of connectivity of all nodes in a website.
Relative connectivity of the node is calculated for finding the relation among the nodes.

5) Relative depth: Once we have the tree structure of the website, depth level D can be easily calculated. Nodes at higher level are usually navigational pages with links to lower level nodes which usually consist of content or service information. So the relative depth, which can also be used to measure the importance of a node, is measured as:
Rd=1/D

6) Node preference: User preference is an important factor for analyzing of requirement. Degree of preference of a node, in terms of time duration, is measured as
T=Tj-Tj-1
Node preference is expressed as
Rt = T/Ta
Where T is visited duration of this node and Ta is the sum of all nodes visited duration.

III. CONCLUSION

The proposed algorithm works for both structured and unstructured website because timestamp is also taken into account to identify the content pages. The goal of optimize time algorithm is to minimize the number of backtracks the visitor has to make. While milestone coefficient defines the importance of a node according to website structure and used preferences in the above expression.
So the proposed algorithm is used to find out the node(s) with highest importance that should be located at a relatively prominent position, which can be used as reference coordinates by browsers.

REFERENCES

[1] Dai Junxiang, "*Self-Adaptive Websites Recommen-dation System Framework*". Doctoral Dissertation, Hunan University, Changsha, 2005. (in Chinese)
[2] Wang Shuzhou, "*Study of Adaptive Web Site Based on Web Mining*", Master Thesis, Harbin University of Science and Technology, Harbin, 2003. (in Chinese)
[3] W.Cohen, H.Hirsh, "*Joins that Generalize: Text Classification Using WHIRL*", Proc. of the 4th International Conference on Knowledge Discovery and Data Mining, New York, 1998(32), pp.169-173..
[4] Randall M. Rohrer, John L. Sibert, David S. Ebert, "*A Shape-based Visual Interface for Text Retrieval*", IEEE Computer Graphics and Applications, 2000, 19(5), pp. 40-46.
[5] M. Kobayashi and Takedak, "*Information Retrieval on the Web*", ACM Computing Surveys(CSUR), 2001, 32(2), pp.144-173.
[6] Du Huifeng, "*Customerized Recommendation Model Based on Web Mining*", CNAIS, 2006. (in Chinese)
[7] Deng Ying, Li Ming, "*Research on Web Mining and Tools*", Computer Engineering and Applications, 2002(20), pp.92-94. (in Chinese)
[8] Lu Lina, "*Sequential Patterns Recognition in Web Log Mining*", Mini-Micro Systems, 2000, 21(5), pp. 481-483. (in Chinese)
[9] F.Valdez, M.Chignell, "*Browsing Models for Hypermedia Databases*". Proc. of the Human Factors Society (32nd Annual Meeting), Santa Monica,1988,196.
[10] S.Mukherjea, Y.Hara, "*Focus Context Views of World Wide Web Nodes*", Proc. of the 8th ACM Conference on Hypertext, ACM Press, Southampton, 1997, pp.187-196.g
[11] Xing Dongshan, Shen Junyi, Song Qinbao, "*Discovering Preferred Browsing Paths from Web Logs*", Chinese Journal of Computers, 2003, 26(11), pp. 1518-1523. (in Chinese)

