Website Structure Improvement through Web Mining

Chhaya N. Shejul[1], Prof. B. Padmavathi[2]

[1] Computer Engineering Department, GHRCEM,

Pune University, Pune, India

chhauu.s@gmail.com

[2] Asst .Prof. in Computer Engineering Department, GHRCEM,

Pune University, Pune, India

bpadma_cse@yahoo.com

Abstract— facilitating effective user navigation through designing well-structured web site becomes a big challenge. This is because gap between users expectations and web developers understanding of how the website sh6uld be structured. The numbers of methods have been proposed to reorganize website to improve user navigation. As they completely reorganize website, new structure becomes unpredictable. We propose a mathematical programming method to reorganize Web structure with minimum changes in order to achieve better navigation efficiency. Heavily disoriented user should get more benefit from less disoriented user.

Keywords— Mathematical programming, Web mining, Website design, User navigation

INTRODUCTION

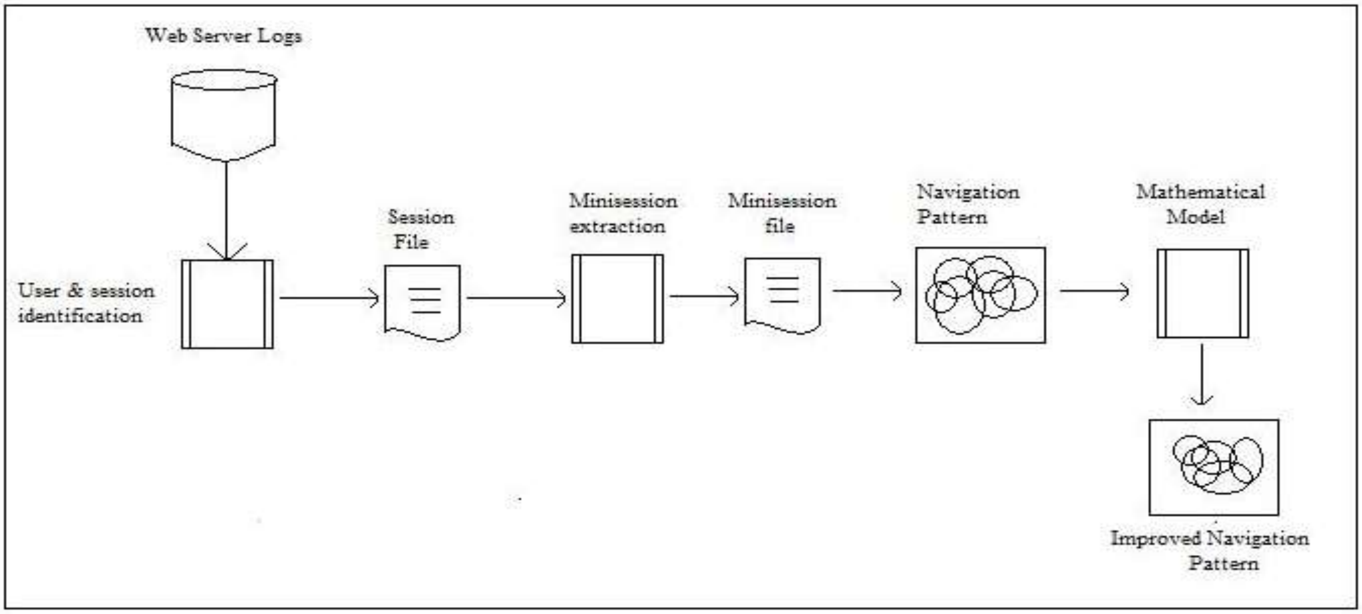
There are three most important activities on web i.e. search, navigation, and transaction. When users find Website which he wants using search engines then before making transactions, Navigation is very important. As the Internet growing rapidly, it is quite difficult to find the desired information on the web. So users having difficulty in locating the targets are very likely to leave a website even if its information is of high quality. The reason behind poor website design is that difference in user expectations and the web developers understanding of how website should be structured. Because of this, user cannot get their desired information. This problem is difficult as developers may not have a clear understanding of users' expectations.

Previous proposed method has focused on various issues, such as extracting template from web pages, finding relevant pages of given page. Our proposed system is closely related to improving user navigation using user navigation data. Generally, there are two classes for improving navigation efficiency i.e. web transformation and web personalization.

Web personalization is dynamically reconstituting web pages for particular users using their profile and navigation data. In our proposed system we are concerned primarily with web transformations which reorganize link structure of website. Web transformation approaches are more appropriate for websites that have a built-in structure and store relatively static and stable contents. Web transformation approaches create or modify the structure of a website used for all users.

LITERATURE SURVEY

Web personalization is the process of customizing a website according to needs of specific user. Perkowitz and Etzioni [8] propose a method that automatically synthesizes index pages. These pages contain links to pages pertaining to particular topics which based on the co-occurrence frequency of pages in user navigation. Mobasher et al. [18], [19], [20]



proposed a method which create clusters of user's profiles from weblogs. After that, dynamically generate links for users who are classified into different categories according to their access patterns. Nakagawa and Mobasher [21] propose a hybrid personalization system. This system can dynamically switch between the user's position in the site and recommendation models based on degree of connectivity.

Web transformation involves altering the structure of a website to facilitate the navigation for a large no. of users [22] instead of personalizing pages for specific users. Fu et al. [23] describe a method to reorganize webpages so as to provide users with their desired information in fewer clicks and have easy navigation. But, this method considers only local structures in a website instead of the site as a whole, so the new structure may not be always optimal. Gupta et al. [25] propose a heuristic method based on simulated annealing. In this method webpages are relink to improve navigability. Use of the aggregate user preference data can improve the link structure in websites for both wired and wireless devices. However, this approach takes relatively a long time (10 to 15 hours) to run even for a very small website. Lin [26] design an integer programming models to reorganize a website based on the cohesion between pages to reduce search depth for users and information overload. There are many differences between web transformation and personalization approaches. First, personalization approaches dynamically reconstitute pages for individual users while, transformation approaches create or modify the structure of a website used for all users. So, there is no predefined/built-in web structure for personalization approaches. Second, Personalization approaches need to collect information associated with individual users (known as user profiles) in order to understand the preference of individual users. This is time-consuming process is not required for transformation approaches.

Fig1. Proposed System Architecture

Third, transformation approaches make use of aggregate usage data from weblog files and do not require tracking the past usage for each user while dynamic pages are typically generated based on the users' traversal path. Thus, personalization approaches are more suitable for dynamic websites whose contents are more volatile and transformation have a built-in approaches are more appropriate for websites that structure and store relatively static and stable contents

III IMPLEMENTATION

Architecture

We use web server log files as input to our system, which consist of user navigation data. In order to examine the interaction between user and a website, the web log files must be broken up into user sessions file. A session is a group of activities performed by a user during his visit to a site. Previous studies propose timeout methods to demarcate sessions from raw weblog files. A session may include one or more target pages, as a user may visit several targets during a single session. The metric used in our propose system is the number of paths traversed to find one target; we use a different term mini session to refer to a group of pages visited by a user for only one target. So, a session may contain one or more mini sessions, each of which comprises a set of paths traversed to reach the target. The page-stay timeout heuristic algorithm is use to demarcate mini sessions. If the time spent on a web page is greater than a timeout threshold then that page is considered as target page. The intension is that a user generally spends more time reading on the page that they find relevant than those they do not. Though it is not possible to identify user sessions unerringly from weblog files, we find the page-stay heuristic algorithm an appropriate method for the context of our problem. After extracting mini session, we build navigation graph. On that various constraints and mathematical model is applied to get improved navigation pattern with minimal changes.

Mathematical MODEL

Our problem can be regarded as a special graph optimization problem. We consider a website as a directed graph, with nodes representing pages and arcs representing links. Let N be the set of all webpages and λ_{ij}, where i, j ε N, denote page connectivity in the current structure, with λ_{ij} = 1 indicating page i has a link to page j, and λ_{ij} =0 otherwise.

The current out-degree for page i is denoted by $W_{i} = \sum_{j \in N} \lambda ij$ From the log files, we obtain the set T of all mini sessions. For a mini session S ε T, we denote tgt(S) the target page of S. Let L_m (s) be the length of S, i.e., the number of paths in S, and L_p (k, s), for $1 \leq k \leq L_m$ (s), be the length of the kth path in S, i.e., the number of pages in the kth path of S. We further define docno(r; k; S), for $1 \leq k \leq Lm_{(s)}$ and $1 \leq r \leq Lp_{(k, s)}$, as the rth page visited in the kth path in S. Take the mini session S in Fig. 2 for example, it follows that L_m (s) = 3; L_p (1, s) =3, and docno (1; 1; S) = A, as this mini session has three paths and the first path has three pages (A, D, and H) in which page A is the first page.

We define E = {(i; j): i; j ε N and ∃S ε T such that i ε S and j = tgt (S)} and N_E = {i: (i, j) ε E}. In essence, E is the set of candidate links that can be selected to improve the site structure to help users reach their targets faster. Our problem is to determine whether to establish a link from i to j or (i, j) ε E. Let x_{ij} ε (0, 1) denote the decision variable such that $x_{ij =}$ 1 indicates establishing the link.

As explained earlier, Webmasters can set a goal for user navigation for each target page, which is denoted by bj and is termed the path threshold for page j. Given a mini session S with target page j and a path threshold bj, we can determine whether the user navigation goal is achieved in S by comparing the length of S, i.e., L_m (S), with path threshold (bj) for the target page of S. If the length of S is larger than bj, it I indicates the user navigation in S is “below” the goal. Then, we need to alter the site structure to improve the user navigation in S to meet the goal. Otherwise, no improvement is needed for S.

Intuitively, given path thresholds, we can determine which mini sessions need to be improved and hence are relevant to our decision (termed relevant mini sessions). Table 1 provides a summary of the notations used in this paper. The problem of improving the user navigation on a website while minimizing the changes to its current structure can then be formulated as the mathematical programming model below:

Minimize

TABLE 1
Summary of Notation

Notation	Definition
S	A mini session that contains the set of paths traversed by a user to locate one target page.
T	The set of all identified mini sessions.
T^R	The set of all relevant mini sessions.
N	The set of all web pages
Λij	1 if page i has a link to page j in the current structure; 0 otherwise
E	The set of candidate links which can be selected for improving user navigation.
E^R	The set of relevant candidate links
N_E	The set of source node of links in set E
W_i	The current out degree of page i
C_i	The out degree threshold for page i
P^i	The no. of links that exceed the out-degree threshold c_i in page i
M	Multiplier for penalty term in the objective function
b_j	The path threshold for mini session in which page j is the target page
$a^s{}_{ijkr}$	1 if I is the rth path and j is the target page in mini session s; 0 otherwise
x_{ij}	1 if the link from page i to j is selected , 0 otherwise
$C^s{}_{kr}$	1 if in mini session s, a link from rth page in the kth path to the target is selected; 0 otherwise
Tgt(s)	**The target page of mini session s**

$$\sum_{(i,j)\in E}^{'} x_{ij}\,[1-\lambda\mathrm{ij}(1-\in)]+m\sum_{i\in N_E}^{'} p_i$$

Subject to

$$c_{kr}^{s}=\sum_{(i,j)\in E}\begin{matrix}a_{ijkr}^{E}x_{ij};\,r=1,2,\ldots,L_p(k,s),\\ k=1,2,\ldots,L_m(s),\forall s\in T^R\end{matrix}\qquad \ldots(1)$$

$$\sum_{k=1}^{b_j} \sum_{r=1}^{L_p(k,s)} c_{kr}^{s} \geq 1; \forall s \in T^R, \quad j = tgt(s) \qquad \dots (2)$$

$$\sum_{j:(i,j)\in E} x_{ij}(1 - \lambda ij) + w_i - p_i \leq c_i; \forall i \in N_E \qquad \dots (3)$$

$$x_{ij} \in \{0,1\}, p_i \in \{0\} \cup Z^+, \forall (i,j) \in E, i \in N_E \qquad \dots (4)$$

The objective function minimizes the cost needed to improve the website structure, where the cost consists of two components: 1) the number of new links to be established (the first summation), and 2) the penalties on pages containing excessive links, i.e., more links than the out-degree threshold (Ci), in the improved structure (the second summation).

We have noted that some existing links may often be neglected by users due to poor design or ambiguous labels. Such links should be improved first before any new links are established. Therefore, we introduce [1-λij (1-ε)], where ε is a very small number, in the objective function to let the model select existing links whenever possible. Note that if (1 - ε) is not present, then there is no cost in choosing an existing link, and this could lead to a number of optimal. As an extreme example, if (1- ε) is removed and the penalty term is not included, the costs of establishing new links, i.e., $\sum_{j:(i,j)\in E} x_{ij}(1 - \lambda ij)$ when selecting all existing links are the same as the costs when none of them is selected. This occurs because there is no cost in selecting an existing link, i.e., (1-λij) = 0, when λij = 1. Thus, we add (1-ε) to impose a very small cost on improving an existing link such that the model will select the minimal number of existing links for improvement.

Choice of Parameter Values for the Model

a. Path Threshold

The path threshold represents the goal for user navigation that the improved structure should meet and can be obtained in several ways. First, it is possible to identify when visitors exit a website before reaching the targets from analysis of weblog files. Hence, examination of these sessions helps make a good estimation for the path thresholds. Second, surveying website visitors can help better understand users' expectations and make reasonable selections on the path threshold values. For example, if the majority of the surveyed visitors respond that they usually give up after traversing four paths, then the path threshold should be set to four or less. Third, firms like comScore and Nielsen have collected large amounts of client-side web usage data over a wide range of websites. Analyzing such data sets can also provide good insights into the selection of path threshold values for different types of websites.

Although using small path thresholds could result in more improvements in web user navigation in general, our experiments showed that the changes (costs) needed increase significantly as the path threshold decreases. Sometimes, additional improvements in user navigation from using a small threshold are too little to justify the increased costs. Thus, Webmasters need to cautiously consider the tradeoff between desired improvements to user navigation and the changes needed when selecting appropriate values for path threshold. A cost benefit analysis that compares "benefits" and "costs" of using different path thresholds can be useful for this purpose. In the context of our problem, we can view the number of new links needed as the cost and the improvement on user navigation (this, for instance, can be measured as the average number of paths shortened by the improved structure) as the benefit. The benefit-cost ratio (BCR) that is used for the analysis of the cost effectiveness of different options can be expressed as (improvement on user navigation)/ (number of new links).

b. Out- Degree Threshold

Webpages can be generally classified into two categories: index pages and content pages. An index page is designed to help users better navigate and could include many links, while a content page contains information users are interested in and should not have many links. Thus, the out-degree threshold for a page is highly dependent on the purpose of the page and the website. Typically, the out-degree threshold for index pages should be larger than that for content pages. In general, the out-degree threshold could be set at a small value when most webpages have relatively few links, and as new links are added, the threshold can be gradually increased. Note that since our model does not impose hard constraints on the out-degrees for pages in the improved structure, it is less affected by the choices of out-degree thresholds as compared to those in the literature.

c. Multiplier for the Penalty Term

The use of the penalty term can prevent the model from adding new links to pages that already have many links. This helps keep the information load low for user at the cost of inserting more new links into other pages with small out-degrees. Generally, if a website have both pages with small out-degrees and pages with very large out-degrees, then it is reasonable to use a large multiplier (m) to avoid clustering too many links in a page. If the out-degrees are relatively small for all pages, then it could be more appropriate to use a relatively small multiplier to minimize the total number of new links added. When our model is used for website maintenance, a small multiplier could be used in the beginning when out-degrees are generally small for most pages, and as new links are inserted, a larger multiplier is needed to prevent adding extra links to pages that already have many links.

IV RESULT SET

We experimented the model with two out-degree thresholds, i.e., C = 20 and C = 40, and two multipliers for the penalty term, i.e., m = 0 and m = 5, on each synthetic data set. Noticeably, the times for generating optimal solutions are low for all cases and parameter values tested, ranging from 0.05 to 24.727 seconds. This indicates that the MP model is very robust to a wide range of problem sizes and parameter values. Particularly, the average solution times for website with 1,000, 2,000, and 5,000 pages are 0.231, 1.352, and 3.148 seconds. While the solution times do go up with the number of webpages, they seem to increase within a reasonable range.

Besides these data sets, two large websites with 10,000 and 30,000 pages were generated and experimented with 300,000, 600,000, and 1.2 million mini sessions to emphasize the fact that the model presented here is scalable to an even larger extent. The solution times are also remarkably low even in this case, varying from 1.734 to 33.967 seconds. In particularly, the average solution times for websites with 10,000 and 30,000 pages are 3.727 and 6.086 seconds, respectively. While the solution times also increase with the size of the website, they seem to increase linearly or slower.

TABLE 2

Evaluation Result on Improved Website Using Number of Paths per Mini Session for T=5 Min

Multiplier for penalty	**Avg. no. of paths improved Website and no. of new link needed**	
	Out-degree threshold	Out-degree threshold c=40

term(m)		c=20				
	b=1	b=2	b=3	b=1	b=2	b=3
0	1.335 (5,794)	1.589 (1145)	1.785 (467)	1.335 (5794)	1.589 (1145)	1.785 (467)
1	1.436 (5794)	1.632 (1166)	1.825 (482)	1.439 (5813)	1.650 (1214)	1.827 (502)
5	1.346 (5794)	1.693 (1182)	1.855 (514)	1.351 (5839)	1.680 (1399)	1.840 (555)

V Conclusion

We proposed a mathematical programming model to improve the navigation effectiveness of a website while minimizing changes to its current structure, a critical issue that has not been examined in the literature. The MP model was observed to scale up very well, optimally solving large-sized problems in a few seconds in most cases on a desktop PC. The comparison showed that our model could achieve comparable or better improvements than the heuristic with considerably fewer new links. To validate the performance of our model, we have defined two metrics and used them to evaluate the improved website using simulations.

VI Future Scope

The paper can be extended in several directions in addition. For example, techniques that can accurately identify users' targets are critical to our model and future studies may focus on developing such techniques. As another example, our model has a constraint for out-degree threshold, which is motivated by cognitive reasons. The model could be further improved by incorporating additional constraints that can be identified using data mining methods. For instance, if data mining methods find that most users access the finance and sports pages together, then this information can be used to construct an additional constraint.

Acknowledgement

I express true sense of gratitude towards my project guide **Prof. Padmavathi B.**, of computer department for her invaluable co-operation and guidance that she gave me throughout my research. I specially thank our **P.G coordinator Prof. Vidhya Damdhere** for inspiring me and providing me all the lab facilities, which made this research work very convenient and easy. I would also like to express my appreciation and thanks to our principal **Dr. D. D. Shah** and all my friends who knowingly or unknowingly have assisted me throughout my hard work

REFERENCES:

[1] D. Dhyani, W.K. Ng, and S.S. Bhowmick, "A Survey of Web Metrics," ACM Computing Surveys, vol. 34, no. 4, pp. 469-503, 2002.

[2] X. Fang and C. Holsapple, "An Empirical Study of Web Site Navigation Structures' Impacts on Web Site Usability," Decision Support Systems, vol. 43, no. 2, pp. 476-491, 2007.

[3] J. Lazar, Web Usability: A User-Centered Design Approach. Addison Wesley, 2006.

[4] D.F. Galletta, R. Henry, S. McCoy, and P. Polak, "When the Wait Isn't So Bad: The Interacting Effects of Website Delay, Familiarity, and breadth," Information Systems Research, vol. 17, no. 1, pp. 20- 37, 2006.

[5] J. Palmer, "Web Site Usability, Design, and Performance Metrics," Information Systems Research, vol. 13, no. 2, pp. 151-167, 2002.

[6] V. McKinney, K. Yoon, and F. Zahedi, "The Measurement of Web-Customer Satisfaction: An Expectation and Disconfirmation Approach," Information Systems Research, vol. 13, no. 3, pp. 296-315, 2002.

[7] T. Nakayama, H. Kato, and Y. Yamane, "Discovering the Gap between Web Site Designers' Expectations and Users' Behavior," Computer Networks, vol. 33, pp. 811-822, 2000.

[8] M. Perkowitz and O. Etzioni, “Towards Adaptive Web Sites: Conceptual Framework and Case Study,” Artificial Intelligence, vol. 118, pp. 245-275, 2000.

[9] Y. Yang, Y. Cao, Z. Nie, J. Zhou, and J. Wen, “Closing the Loop in Webpage Understanding,” IEEE Trans. Knowledge and Data Eng., vol. 22, no. 5, pp. 639-650, May 2010.

[10] J. Hou and Y. Zhang, “Effectively Finding Relevant Web Pages from Linkage Information,” IEEE Trans. Knowledge and Data Eng., vol. 15, no. 4, pp. 940-951, July/Aug. 2003.

[11] H. Kao, J. Ho, and M. Chen, “WISDOM: Web Intrapage Informative Structure Mining Based on Document Object Model,” IEEE Trans. Knowledge and Data Eng., vol. 17, no. 5, pp. 614-627, May 2005.

[12] H. Kao, S. Lin, J. Ho, and M. Chen, “Mining Web Informative Structures and Contents Based on Entropy Analysis,” IEEE Trans. Knowledge and Data Eng., vol. 16, no. 1, pp. 41-55, Jan. 2004.

[13] C. Kim and K. Shim, “TEXT: Automatic Template Extraction from Heterogeneous Web Pages,” IEEE Trans. Knowledge and Data Eng., vol. 23, no. 4, pp. 612-626, Apr. 2011.

[14] M. Kilfoil et al., “Toward an Adaptive Web: The State of the Art and Science,” Proc. Comm. Network and Services Research Conf., pp. 119-130, 2003.

[15] R. Gupta, A. Bagchi, and S. Sarkar, “Improving Linkage of Web Pages,” INFORMS J. Computing, vol. 19, no. 1, pp. 127-136, 2007.

[16] C.C. Lin, “Optimal Web Site Reorganization Considering Information Overload and Search Depth,” European J. Operational Research, vol. 173, no. 3, pp. 839-848, 2006.

[17] M. Eirinaki and M. Vazirgiannis, “Web Mining for Web Personalization,” ACM Trans. Internet Technology, vol. 3, no. 1, pp. 1-27, 2003.

[18] B. Mobasher, H. Dai, T. Luo, and M. Nakagawa, “Discovery and Evaluation of Aggregate Usage Profiles for Web Personalization,” Data Mining and Knowledge Discovery, vol. 6, no. 1, pp. 61-82, 2002.

[19] B. Mobasher, R. Cooley, and J. Srivastava, “Automatic Personalization Based on Web Usage Mining,” Comm. ACM, vol. 43, no. 8, pp. 142-151, 2000.

[20] B. Mobasher, R. Cooley, and J. Srivastava, “Creating Adaptive Web Sites through Usage-Based Clustering of URLs,” Proc. Workshop Knowledge and Data Eng. Exchange, 1999.

[21] M. Nakagawa and B. Mobasher, “A Hybrid Web Personalization Model Based on Site Connectivity,” Proc. Web Knowledge Discovery Data Mining Workshop, pp. 59-70, 2003.

[22] B. Mobasher, “Data Mining for Personalization,” The Adaptive Web: Methods and Strategies of Web Personalization, A. Kobsa, W. Nejdl, P. Brusilovsky, eds., vol. 4321, pp. 90-135, Springer-Verlag, 2007.

[23] C.C. Lin and L. Tseng, “Website Reorganization Using an Ant Colony System,” Expert Systems with Applications, vol. 37, no. 12, pp. 7598-7605, 2010.

[24] Y. Fu, M.Y. Shih, M. Creado, and C. Ju, “Reorganizing Web Sites Based on User Access Patterns,” Intelligent Systems in Accounting, Finance and Management, vol. 11, no. 1, pp. 39-53, 2002.

[25] R. Gupta, A. Bagchi, and S. Sarkar, “Improving Linkage of WebPages,” INFORMS J. Computing, vol. 19, no. 1, pp. 127-136, 2007.

C.C. Lin, “Optimal Web Site Reorganization Considering Information Overload and Search Depth,” European J. Operational Research, vol. 173, no. 3, pp. 839-848, 2006