Review on Adaptive Web Interfaces

[1]Yadwinder singh, [1]Himanshu Aggarwal
[1]*University College of Engineering, Department of Computer Engineering, Punjabi University, Patiala, Punjab, India*

Abstract-Today web has grown tremendously, and in this era there are no. of websites with different types of contents. User interface is the main thing which is noticed in the website and it must be good to sustain in market. Adaptive web interface is the main thing that can be implemented on web site so that the interface must change according to user likes and priorities. The main goal of this paper is to impose the comprehensive survey of Adaptive web interfaces, various techniques proposed, various issues and challenges are presented.

Keywords – *Web usage mining, web server log, Preprocessing, Pattern Discovery, Pattern Analysis, Clustering, Data mining.*

I. INTRODUCTION

Actionable knowledge that is taken by mining web logs and results have been implemented automatically with which the interface of website changes automatically leads to Adaptive web interfaces. In the last few years, adaptive web sites have focused more and more attention from data mining researchers. Adaptive web sites dynamically improve their structure and presentation in order to satisfy needs of individual users [1], [3-5]. There can be various attributes which must be considered for constructing the procedure or algorithm to implement to have the adaptive web interface. Different type of users have different type of interest , and different types of web sites have various contents , all of these things must be considered to have the adaptive web interface. Many different visitors approach a popular web site – each with his or her own goals and concerns. Consider, for example, the web site for a typical computer science department [2]. The site contains an amalgam of research project descriptions, course information, lists of graduating students, pointers to industrial affiliates, and much more. Each nugget of information is of value to someone who would like to access it readily. One might think that a well organized hierarchy would solve this problem, but we've all had the experience of banging our heads against a web site and crying out ''it's got to be here somewhere [2]. Sites may be adaptive in two basic ways. Customization is adapting the site's presentation to the needs of individual visitors, based on information about those individuals. In order to specialize itself to individual users, the site must maintain multiple copies of itself and gather quite a bit of information from users. Providing such information to the site can be time consuming and may be an invasion of piracy [6]. Optimization is improving the site's structure based on interactions with all visitors. Instead of making changes for each individual, the site learns from numerous past visitors to make the site easier to use for all including those who have never used it before [6].

II. WEB MINING

Web Mining [7] is that area of Data Mining which deals with the extraction of interesting knowledge from the World Wide Web. Let's have an example suppose somebody has ordered for drawing book online than knowing that how many of those who have ordered for drawing book have also ordered for colors is the information to be extracted by the web mining. It is of 3 types:

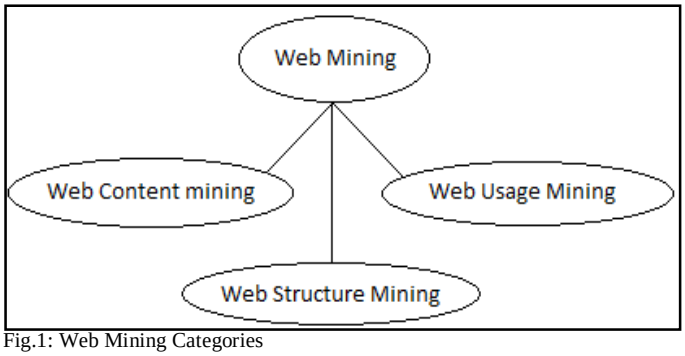

Fig.1: Web Mining Categories

A. Web Content Mining

It is that part of Web Mining which focuses on the raw information available in web pages; source data mainly consist of textual data in web pages [7].

B. Web Structure Mining

It is that part of Web Mining which focuses on the structure of web sites, it mainly consist of structural information in web pages [7].

C. Web Usage Mining

It deals with the extraction of knowledge from server log files; source data mainly consist of the logs that is collected when users access web servers [7] Web usage mining itself can be classified further depending on the kind of usage data considered:

1. Web Server Data

The user logs are collected by the Web server. Typical data includes IP address, page reference and access time.

2. Application Server Data

Commercial application servers have significant features to enable e-commerce applications to be built on top of them with little effort. A key feature is the ability to track various kinds of business events and log them in application server logs.

3. Application Level Data

New kinds of events can be defined in an application, and logging can be turned on for them thus generating histories of these specially defined events. It must be noted, however, that many end applications require a combination of one or more of the techniques applied in the categories above.

Web usage mining is the application of data mining techniques to discover usage pattern from Web data, in order to understand and better serve the needs of Web-based applications. Web usage mining consists of three phases, namely preprocessing, pattern discovery, and pattern analysis [13]. These are briefly explained as follows.

- *Preprocessing*

Commonly used as a preliminary data mining practice, data preprocessing transforms the data into a format that will be more easily and effectively processed for the purpose of the user. The different types of preprocessing in Web Usage Mining are— usage, content, and structure preprocessing.

- *Pattern Discovery*

WUM can be used to uncover patterns in server logs but is often carried out only on samples of data. The mining process will be ineffective if the samples are not a good representation of the larger body of data. The various pattern discovery methods are— Statistical Analysis, Association Rules, Clustering, Classification, Sequential Patterns, and Dependency Modeling.

- *Pattern Analysis*

The need behind pattern analysis is to filter out uninteresting rules or patterns from the set found in the pattern discovery phase. The most common form of pattern analysis consists of a knowledge query mechanism such as SQL. Content and structure information can be used to filter out patterns containing pages of a certain usage type, content type, or pages that match a certain hyperlink structure.

III. RELATED WORK

A. The Index Page Synthesis Problem

Index page synthesis problem [6]: given a web site and visitor access log, create new index pages containing collections of links to related but currently unlinked pages. A web site is restricted to a collection of HTML documents residing at a single server – dynamically generated pages or multiple servers are not yet handled. An access log is a document containing one entry for each request answered by the web server. Each request lists at least the origin of the request, the URL requested, and the time of the request. Related but unlinked pages are pages that share a common topic but are not currently linked at the site; two pages are considered linked if there exists a link from one to the other or if there exists a page that links to both of them. Rather than attempting to understand the content of every page at a site and to figure out which are related, approach is based on analysis of each visit [6]. Define a visit to be an ordered sequence of pages accessed by a single visitor in a single session. Visit coherence assumption is made: the pages a user visits during one interaction with the site tend to be conceptually related [6].

B. PageGather Algorithm

The PageGather algorithm [8] uses cluster mining to find collections of related pages at a web site. In essence, PageGather takes a web server access log as input and maps it into a form ready for clustering; it then applies cluster mining to the data and produces candidate index-page contents as output. The algorithm comprises of six steps:

1. Process access log into visits.
2. Compute the co-occurrence frequencies between pages and create a similarity matrix.
3. Create the graph corresponding to the matrix and find maximal cliques in graph.
4. Rank the cluster found.
5. For each cluster, create a web page consisting of links to the documents in the cluster.
6. Present the clusters to the Webmaster for evaluation.

Drawback of PageGather algorithm is that it relies on the human Web masters to determine the appropriateness of the generated index pages in final check. This will create a bottleneck for the workflow, especially for sites that have many web pages to be indexed [8].

C. Improvement to Page Gather System

Page Gather system is improved by automating the index-page creation process completely [8]. It is done by introducing a cost function, which is then used to judge the quality of the

index pages created. Let Overall Cost be the overall cost of the browsing web pages and index pages. Let Page Cost be the cost of flipping index pages and Transition Cost be the cost of switching from one web page to another. Then cost models are as follows:

Overall Cost = PageCost + TransitionCost (1)

When the number of index pages increases, the Transition Cost should decrease while, at the same time, the PageCost associated with the need to flip through the index pages increases [8]. In the algorithm by Perkowitz and Etzioni they first computes clusters from the web logs and then put all clusters in index pages, so that each cluster will correspond to one index page. In our experience, we have found that often each cluster will contain a large number of index pages [8]. When hundreds of hyperlinks are included in an index page, it is very difficult for a user to find the information he/she is looking for. In addition, we feel that there should be a limited number of index pages; if the user is required to read a huge number of index pages then it might defeat the purpose of including the index page in first place [8]. Therefore, in index page construction, we will include two parameters. Let the L be the number of hyperlinks we would like to include in each index page, and let M be the number of index pages we wish to build. Algorithm Construct Index Pages takes the parameters M and L and the clusters constructed by RDBC algorithm, and produces M index pages as output [8]:

Algorithm *ConstructIndexPages*(Clusters,M,L)

```
1) For j=1 to M
2) {
3) Sort Clusters by the frequency count of the top L web pages;
4) Extract the top L web pages from the first cluster and insert
   their hyperlinks into an index page;
5) If( No cluster is left or size of each cluster < L), stop;
6) }
```

More importantly, based on these cost functions, we can find a minimal value M for the OverallCost and its corresponding number of index pages to build. This optimal index-page construction process represents another major contribution of our work. What we do is to analyze the overall cost as a function of the number of index pages M to construct, based on a fixed value of L. We can then find empirically the best value for M so as to minimize the overall cost of user browsing effort [8].

D. Avanti Project

Several researchers have studied the problem of creating web interfaces that can adapt to user behaviors based on web logs. Examples include path prediction algorithms that guess where the user wants to go next in a browsing session so that the server can either pre-send documents or short-cut browsing paths. For instance, Web Watcher [11] learns to predict what links users will follow on a particular page as a function of their specified interests. A link that WebWatcher believes a particular user is likely to follow will be highlighted graphically and duplicated at the top of the page when it is presented. The Avanti Project [9] focuses on dynamic customization based on users' needs and tastes. As with the WebWatcher, Avanti relies partly on users providing information about them when they enter the site. Based on what it knows about the user, Avanti attempts to predict both the user's eventual goal and his/her likely next step. Avanti will prominently present links leading directly to pages it thinks a user will want to see. Additionally, Avanti will highlight links that accord with the user's interests.

E. Strudel Approach

It attempts to separate the information available at a web site from its graphical presentation. Instead of manipulating web sites at the level of pages and links, web sites may be specified using Strudel's view-definition language [10]. With all of the site's content so encoded, its presentation may be easily adapted.

F. Collaborative filtering

Rating is done by the users according to their interests and then customization is done and is known as collaborative filtering. In this type of customization, rating of objects is done by users then a pattern is found that which users have the same type of interest and then an object purchased by one user is recommended to the another user having the same type of interest. A simple form of collaborative filtering is used by, for example, Amazon.com; the Web page for a particular book may have links to other books commonly purchased by people who bought this one. Firefly uses a more individualized form of collaborative filtering in which members may rate hundreds of CDs or movies, building up a very detailed personal profile; Firefly then compares this profile with those of other members to make new recommendations.

G. Client side customization

It is a technique in which a user has his/her own associated agent who learns about his/her interests and customizes his/her web experience accordingly. The AiA project [11, 12] explores the customization of web page information by adding a ``presentation agent'' who can direct the user's attention to topics of interest.

IV. CONCLUSION

This paper has been presented the various techniques for creating the adaptive web interfaces. It presented the PageGather algorithm which is helpful in creating the index pages and its drawbacks and modifications are also presented. Adaptive web interface is the today's need as our web is growing day by day. We have to consider the various interests of the users and various web sites having various types of contents for having a Adaptive web interface. Future work is expected to focus on creation of Adaptive web interfaces by exploiting various techniques.

ACKNOWLEDGEMENT

The authors are grateful to Dr. Himanshu Aggarwal, Professor, Department of Computer Engineering, Punjabi University, Patiala for their support and guidance.

V. REFRENCES

[1] Tadeusz Morzy, Marek Wojciechowski, Maciej Zakrzewicz, Piotr Dachtera, and Piotr Jurga, Poznan University of Technology, ul.Piotrowo 3a, Poznan, Poland, Implementing Adaptive User Interface for Web Applications.

[2] Mike Perkowitz, Oren Etzioni , Department of computer science and Engineering,Box 352350, University of Washington , Seatle, WA 98195, USA, Towards adaptive web sites: conceptual framewoek and case study.

[3] Perkowitz, M., Etzioni, O.(1997) Adaptive web sites: an AI challenge. Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence, Nagoya, Japan,16-23.

[4] Perkowitz, M., Etzioni, O.(1999) towards adaptive web sites: conceptual Framework and case study. Proceedings of the Eighth International World Wide Web Conference, Toronto, Canada, Computer Networks 31, 1245-1258.

[5] Yan, T.W., Jacobsen, M.Garcia-Molina, H., Dayal, U.(1996) from User Access Patterns to Dynamic Hypertext Linking. Proceedings of the fifth International World Wide Web Conference, Paris, France, Computer Networks 28, 1007-1014.

[6] Mike Perkowitz, Oren Etzioni , Department of computer science and Engineering, Box 352350, University of Washington, Seatle, WA,98195{map,etzioni}@cs.washington.edu,(206)616-1845Fax(206)543-969, Adaptive web sites: Automatically Synthesizing Web pages

[7] Federico Michele Facca and Pier Luca Lanzi, Artificial Intelligence and Robotics Laboratory, Dipartimento di Elettronica e Informazione, Politecnico di Milano, Recent Development in Web Usage Mining Research .

[8] Qiang Yang, Charles X.Ling and Jianfeng Gao, Department of computer science Hong Kong University of science and technology Clearwater bay, Kowloon Hong Kong, Department of Computer Science, University of western Ontario, Ontario N6A 5B7, Canada, Microsoft Research Asia, China, Mining Web logs for Actionable Knowledge.

[9] A. Fox, S. Gribble, Y. Chawathe, and E. Brewer.Adapting to Network and Client Variation Using Infrastructural Proxies: Lessons and Perspectives.IEEE Personal Communications, 5(4):10-19, 1998

[10] M. Fernandez, D. Florescu, J. Kang, A. Levy and D. Suciu, System demonstration — Strudel: a Web-site management system, in: ACM SIGMOD Conf. on Management of Data,1997.

[11] E. Andre´, W. Graf, J. Mu¨ller, H.-J. Profitlich, T. Rist and W. Wahlster, AiA: adaptive communication assistant for effective infobahn access, Document, DFKI, Saarbru¨cken,1996.

[12] T. Rist, E. Andre´ and J. Mu¨ller, Adding animated presentation agents to the interface, in: Proc. 1997 Int. Conf. on Intelligent User Interfaces, Orlando, FL, 1997, pp. 79–86.

[13] Mobasher B, Cooley R, Srivastava J. "Creating Adaptive Web Sites Through Usage-Based Clustering of URLs". In: IEEE Knowledge and Data Engineering Workshop (KDEX'99).

Yadwinder Singh is a M.Tech student at University college of engineering at Punjabi University Patiala. His Research area is Web Log mining and Adaptive Web Interfaces.

Dr.Himanshu Aggarwal, Ph.D., is currently serving as Professor in Department of Computer Engineering at Punjabi University, Patiala. He has more than 20 years of teaching experience and served academic institutions such as Thapar Institute of Engineering & Technology, Patiala, Guru Nanak Dev Engineering College, Ludhiana and Technical Teacher's Training Institute, Chandigarh. He is an active researcher who has supervised more than 30 M.Tech. Dissertations and contributed 50 articles in various Research Journals. He is guiding PhD to 7 scholars and three has completed his PhD. He is on the Editorial and Review Boards of several Journals of repute. His areas of interest are Software Engineering, Computer Networks, Information Systems, ERP and Parallel Computing