IMPROVEMENT OF WEB DATA CLUSTERING USING WEB PAGE CONTENTS

Yue Xu and Li-Tung Weng
School of Software Engineering and Data Communications
Queensland University of Technology
GPO Box 2434
Brisbane, QLD 4001, Australia
yue.xu@qut.edu.au, l.weng@student.qut.edu.au

Abstract This paper presents an approach that discovers clusters of Web pages based on Web log data and Web page contents as well. Most existing Web log mining techniques are access-based approaches that statistically analyze the log data without paying much attention on the contents of the pages. The log data contains various kinds of noise which can significantly influence the performance of pure access-based web log mining. The method proposed in this paper not only considers the frequence of page co-occurrence in user access logs, but also takes into account the web page contents to cluster Web pages. We also present a method of using information entropy to prune away irrelevant papges which improves the performance of the web page clustering.

Keywords: Web log mining, clustering, Web contents, information entropy

1. Introduction

Navigating through large web sites for finding desired information or some particular product to purchase can be tedious and frustrating as the WWW has grown to a huge size. Web agents that can consider users' own information needs and guide the users to their desired information are in great demand. This inspired the research on Web personalization. According to Mobasher [7], Web personalization can be described as any action that makes the Web experience of a user customized to the user's taste or preferences. In order to achieve Web personalization, we need to generate user profiles that characterize users' needs and interests. There are two major approaches used to generate user profiles. One approach is to take explicit rating information from users [6]. The user profiles consist of previous users' ratings and the current user's ratings as well. These ratings characterize the users' preferences for the items in the Web site. This approach relies heavily on human input and brings extra work. Recently, an increasing number of researchers have focused their study on applying data mining techniques to web server log analysis for automatically generating user profiles. Some proposed approaches analyze previous users web logs to discover user navigation patterns such as popular navigation sessions, item association rules, page clusters, and user clusters [2, 9]. These patterns form the user profiles that capture and model the behavioral characteristics of users interacting with a Web site. Obviously building user profiles based on Web usage mining will not bring any extra work to users and the coverage of the user profiles can be large if a great amount of logs are available. Moreover, the user profiles are dynamically generated from user navigation logs and not subjective ratings provided by user themselves, and thus the system performance does not degrade over time.

A variety of knowledge discovery techniques have been applied to obtain usage patterns. Clustering is one of the popular used techniques for grouping user navigation sessions or Web pages. The authors of [8] have proposed methods to classify users by clutering user navigation sessions. An algorithm called PageGather has been used to discover groups of pages based on page co-occurrence [9]. Mobasher research group has proposed several techniques based on clustering to extract usage knowledge for the purpose of Web personalization [7]. Web usage mining techniques are access-based approaches that statistically analyze Web server logs and do not pay attention on the content of the pages visited by users. However, solely relying on Web usage data for obtaining user profiles can be problematic since the Web usage logs contain noise such as missing page requests and containing irrelevant page requests made by users. These missing or irrelevant page requests can make great impact on the quality of user profiles obtained from the logs. For generating more accurate user profiles and thus providing more reliable recommendations, we should consider both Web usage data and Web content data in the generation of user profile.

In this paper, we will present an approach that generates groups of Web pages by using both Web logs and Web page contents. The page co-occurrence information and page topic information are considered at the same time when generating the user profile. The rest of the paper is organized as follows. Section 2 introduces briefly the data preprocessing. In section 3, we define some similarity measures to characterize Web pages and use fuzzy clustering method CARD [8] to obtain page clusters. Section 4 presents the use of information entropy to refine user navigation sessions. Finally, Section 5 summarizes the paper.

2. Data Preparation

For any data mining application, the first step is to prepare a suitable data set to which data mining techniques are applied. The data set can be created by preprocessing some original data sources. The original data sources for generating the user profiles in our case include Web log data and Web contents. The essential tasks in log data preprocessing include data cleaning, user identification, and user session identification. The output of the preprocessing is a collection of user navigation sessions. A log file is an ordered set of requests made by users. A user's page request often results in several log entries in the log file since images and scripts are down-loaded in addition to the page HTML or JSP file itself. Actually only the page but the images or scripts is the real item interested by the user. Therefore, in order to capture the user's navigation intention, elimination of those irrelevant items from the Web log file becomes necessary. This is the task of data cleaning.

User requests are stored in the order that the server receives them. If multiple users are browsing the site concurrently, their requests are intermininglied in the log file. The goal of session identification is to identify the page requests made by each user and construct the page accesses into sessions. Since a user may visite a web site more than once during a period of time, a user navigation session is usually defined as a sequence of pages visited by the same user such that no two consecutive pages are separated by more than, for example, 30 minutes [2]. Table 2.1 shows an example of a cleaned log segment that is obtained from the log data of the Web site of the Centre for Information Technology Innovation, Faculty of Information Technology at Queensland University of Technoloy. Table 2.2 shows the set of user navigation sessions obtained from the log segment.

3. Web Page Clustering

The task we address in this section is to find clusters of related Web pages based on user access logs and Web contents. Data mining techniques have been widely applied to categorize Web objects such as Web pages. However, due to the ambiguity and imcompleteness in Web usage data, the categories in Web objects may not have crisp boundaries. This fuzzy nature in Web clustering makes the straightforward use of data mining techniques sometimes not very effective. A number of fuzzy clustering methods have been developed [5]. The most known method of fuzzy clustering is the Fuzzy c-Means (FCM) algorithm [1]. The FCM method is applicable only to object data that represents the objects by feature vectors. Hathaway's RFCM algorithm [4] extends the FCM to relational data that represents the objects by numerical values representing the degrees to which pairs of objects in the data set are related. However, both the FCM and the RFCM have a requirement that the number c of clusters

Item ID	Page Request
A1	127.0.0.1 - - [05/Sep/2003:10:27:55 10000] "GET /index.jsp HTTP/1.1" 200 8274
A2	127.0.0.1 - - [05/Sep/2003:10:28:33 10000] "GET /research/index.jsp HTTP/1.1" 302 -
A3	127.0.0.1 - - [05/Sep/2003:10:29:01 10000] "GET /research/sdl/index.jsp HTTP/1.1" 200 29217
A4	127.0.0.1 - - [05/Sep/2003:10:32:22 10000] "GET /research/sdl/projects/ HTTP/1.1" 404 -
A5	127.0.0.1 - - [05/Sep/2003:10:32:38 10000] "GET /research/sdl/robotsoccer.jsp HTTP/1" 200 40018
A2	127.0.0.1 - - [05/Sep/2003:11:29:33 10000] "GET /research/index.jsp HTTP/1.1" 302 -
A3	127.0.0.1 - - [05/Sep/2003:11:30:01 10000] "GET /research/sdl/index.jsp HTTP/1.1" 200 29217-
A4	127.0.0.1 - - [05/Sep/2003:11:32:42 10000] "GET /research/sdl/projects/ HTTP/1.1" 404 -
A6	127.0.0.1 - - [05/Sep/2003:11:32:48 10000] "GET /research/sdl/researchers.jsp HTTP/1.1" 200 21963
A7	127.0.0.1 - - [05/Sep/2003:12:09:06 10000] "GET /pubs/index.jsp HTTP/1.1" 200 19757
A8	127.0.0.1 - - [05/Sep/2003:12:09:19 10000] '"GET /pubs/2002.jsp HTTP/1.1" 200 43204
A9	127.0.0.1 - - [05/Sep/2003:12:10:46 10000] "GET /people/index.jsp HTTP/1.1" 200 18494
A10	127.0.0.1 - - [05/Sep/2003:12:10:49 10000] "GET /people/students.jsp HTTP/1.1" 200 47065

Table 2.2. An example set of user's sessions

Session Number	User Sessions
1	$A_1 \longrightarrow A_2 \longrightarrow A_3 \longrightarrow A_4 \longrightarrow A_5$
2	$A_2 \longrightarrow A_3 \longrightarrow A_4 \longrightarrow A_6$
3	$A_7 \longrightarrow A_8 \longrightarrow A_9 \longrightarrow A_{10}$

has to be specified prior to the clustering. Based on Frigui & Krishnapuram's Competitive Agglomeration (CA) [3], Nasraoui proposed the competitive Agglomeration algorithm for relational data (CARD) [8] that can cluster data into the optimal number of clusters without requiring a prior specified value for c. In this section, we first define the pairwise relational measurements to characterize the similarity beween Web pages in terms of user access frequency and page contents, we then present the use of the fuzzy clustering algorithm CARD to obtain Web pages clusters based on the page similarity.

3.1 Similarity Between Web Pages

3.1.1 Page Co-occurrence Measurement. If two pages frequently appear together in the same sessions, the two pages are considered related with each other in some aspect which may be interested by the user. Let N_s be the number of sessions, N_i be the number of sessions which contains page requests to page i, and N_i^j be the number of sessions in which page j is visited after page i has been already visited. The probability of a user requesting page i is defined as: $p(i) = \frac{N_i}{N_s}$. The probability of a user visiting page j after the user has already visited page i is defined as: $p(j/i) = \frac{N_i^j}{N_s}$. The joint probability of page i and page j given below is used to measure the co-occurrence frequency between page i and page j:

$$p(i,j) = \begin{cases} max\{p(i)p(j/i), p(j)p(i/j)\} & i \neq j \\ 1 & i = j \end{cases}$$

The $p(i,j)$ represens the probability that page i and page j are visited together in a session. We calculate the co-occurrence frequency of each pair of pages and create a matrix based on the frequency. Table 3.1 is the co-occurrence frequency matrix derived from the sessions given in Table 2.2.

Table 3.1. Co-occurrence frequency matrix obtained from the sessions in Table 2.2

	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10
A1	1	0.11	0.11	0.11	0.11	0	0	0	0	0
A2	0.11	1	0.44	0.44	0.22	0.22	0	0	0	0
A3	0.11	0.44	1	0.44	0.22	0.22	0	0	0	0
A4	0.11	0.44	0.44	1	0.22	0.22	0	0	0	0
A5	0.11	0.22	0.22	0.22	1	0	0	0	0	0
A6	0	0.22	0.22	0.22	0	1	0	0	0	0
A7	0	0	0	0	0	0	1	0.11	0.11	0.11
A8	0	0	0	0	0	0	0.11	1	0.11	0.11
A9	0	0	0	0	0	0	0.11	0.11	1	0.11
A10	0	0	0	0	0	0	0.11	0.11	0.11	1

3.1.2 Page Content Similarity Measurement.

Web usage mining relies only on user access log data to discover useful patterns. However, the web logs may contain some data that does not really reflect user's navigation intention. One reason is that some data may be missing due to caching by the browser. Another reason is that the user may visited some irrelevant pages on their way to find the desired information. These irrelevant pages in the user's navigation sessions may make significant impact on the quality of the mining results. With a navigation goal in his/her mind, a user will visit the pages which should be content relevant. This suggests that the page contents can be used to eliminate or alleviate the impact caused by the missing data or the irrelevant pages.

In order to use page contents, the Web pages should be well characterized. One of the commonly used techniques for textual documents in Information Retrieval is to repreent each document as a feature vector. In this paper, the features are extracted from the meta data embedded in HTML files. We assume that associated with each page there is a specific tag in its HTML file that provides a list of topics and a weight for each topic. The list of topics is specified by the Web site designer and used as the feature vector or topic vector to measure the similarity among pages.

Suppose that there are totally n topics involved in a web site, and that associated with each page in the site there is an n-dimensional vector which characterizes the relevancy of each topic to the page. The ith element in the vector represents the relevancy assigned to the ith topic. Let H is the set of all pages in the site, $\forall h_i \in H$, there is a n-dimensional vector denoted as $T_i =< t_{i1}, \dots, t_{in} >$, where t_{ij} represents the relevancy of the jth topic to the page h_i. The similarity between page h_i and page h_j can be measured by the *cosine* of the angle between their topic vectors which is calculated by: $cosine(T_i, T_j) = \frac{T_i \cdot T_j}{||T_i||_2 \times ||T_j||_2}$.

For the Web site of the Centre for Information Technology Innovation, there are 19 topics. For simplicity, we use 1 or 0 to weight the topics in the topic vectors involved. Table 3.2 gives the topic vectors of each page in the log segment shown in Table 2.1. A 10 × 10 similarity matrix, as given in Table 3.3, can be produced in terms of the pairwise *consine* value of each pair of pages. In this paper, we use a simple combination of the two measurements, as described below, to measure the similarity of two pages, where $0 \leq \alpha \leq 1$.

$$sim(i, j) = (1 - \alpha) * p(i, j) + \alpha * cosine(T_i, T_j) \qquad (3.1)$$

Table 3.2. Topic Vectors of the pages requested in the log segment in Table 2.1

Page	Topic Vector
A1	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
A2	0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0
A3	0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0
A4	0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0
A5	0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0
A6	0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0
A7	0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0
A8	0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0
A9	0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0
A10	0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0

3.2 Fuzzy Clustering

Let $X = \{x_1, \dots, x_n\}$ be a set of given data. For a given constant c, $2 \leq c \leq n$, the data set is to be partitioned into c clusters. Assume $U = [u_{ij}]$ is the membership degree with which the data point x_j belongs to the cluster i and

Table 3.3. Content similarity matrix obtained from the log in Table 2.1

	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10
A1	1	0.33	0.24	0.41	0.33	0.24	0.24	0.24	0.24	0.33
A2	0.33	1	0.71	0.41	0.5	0	0	0	0	0
A3	0.24	0.71	1	0	0.71	0	0	0	0	0
A4	0.41	0.41	0	1	0.41	0	0	0	0	0
A5	0.33	0.5	0.71	0.41	1	0	0	0	0	0
A6	0.24	0	0	0	0	1	0	0	1	0.71
A7	0.24	0	0	0	0	0	1	1	0	0
A8	0.24	0	0	0	0	0	1	1	0	0
A9	0.24	0	0	0	0	1	0	0	1	0.71
A10	0.33	0	0	0	0	0.71	0	0	0.71	1

$\mathbf{R} = [r_{ij}]$ is the relational data corresponding to pairwise distances between data points, where $r_{ij} = \|x_i - x_j\|^2$ and $\| \cdot \|$ is any inner product induced norm. Hathaway et al. [4] have proved that the squared Euclidean distance, $d_{ik}^2 = \|x_k - v_i\|^2$, from feature vector x_k to the center of the i^{th} cluster, c_i, can be written in terms of the relation matrix $\mathbf{R}$ as $d_{ik}^2 = (\mathbf{R}v_i)_k - v_i\mathbf{R}v_i/2$, where v_i is the i^{th} cluster center defined by $v_i = \frac{(u_{i1}^m,\ldots,u_{in}^m)^t}{\Sigma_{j=1}^n u_{ij}^m}$. where $m > 1$.
For a pre-defined relatively large value $c = c_{max}$, the CARD algorithm [8] starts by partitioning the data set into c clusters. As the algorithm progresses, in each iteration after the membership values u_{ik} is updated, the clusters whose cardinality N_i as defined below is less than a threshold whill be discarded, and thus the number of clusters is reduced.
The CARD algorithm is described as follows (see [8] for details).

1 Given relational data $\mathbf{R} = [r_{ij}]$. Choose a value for η_0, τ, ρ, ϵ, and c_{max}, generate randomly the membership matrix U^0 which determines a fuzzy c-partition, and set iteration number $t = 0$.

2 Calculate the cluster centers $v_i^{(t)}$ $(i = 1, \ldots, c)$ using $v_i = \frac{(u_{i1}^m,\ldots,u_{in}^m)^t}{\Sigma_{j=1}^n u_{ij}^m}$.

3 Update the membership values $u_{ik}^{(t+1)}$ by $u_{ik}^{(t+1)} = u_{ik}^{FCM} + u_{ik}^{Bias}$, where, $u_{ik}^{FCM} = [\Sigma_{j=1}^c (\frac{d_{ik}}{d_{jk}})^2]^{-1}$, $u_{ik}^{Bias} = \frac{\eta(t)(N_i - \overline{N}_k)}{d_{ik}^2}$, $N_i = \Sigma_{j=1}^n u_{ij}$, $\overline{N}_k = \frac{\Sigma_{j=1}^c (1/d_{jk}^2) N_j}{\Sigma_{j=1}^c (1/d_{jk}^2)}$, $\eta(t) = \eta_0 e^{-t/\tau}$, and d_{ik} is calculated by $d_{ik}^2 = (\mathbf{R}v_i)_k - v_i\mathbf{R}v_i/2$.

4 Calculate N_i, $i = 1, \ldots, c$. If $N_i < \rho$, remove the i^{th} row from both $U^{(t+1)}$ and $U^{(t)}$, update $c = c - 1$.

5 If $\|U^{(t+1)} - U^{(t)}\| \leq \epsilon$ or a predefined number of iterations is reached, stop, otherwise set t=t+1 and return to step 2.

4. Session Pruning

In information theory, Shannon's measure of entropy is used as a measure of the information contained in a piece of data. For a random variable X with a set of possible values $< x_1, \ldots, x_n >$, having probabilities $p(x_i)$, $i = 1, \ldots, n$, if we had no information at all about the value X would be, the possibility for each value should be the same, i.e. $1/n$. In this case, X is in its most uncertain situation. According to information theory, the entropy of X reaches its maximum in this situation. On the other hand, if the entropy of X is close to zero, the value of X has few uncertainties. In this case, there should be a small set of values with high probabilities and others with very low probabilities. Based on this theory, we propose to use the entropies of topics to prune the sessions.

$\forall h_i \in H$, its topic vector is $T_i =< t_{i1}, \ldots, t_{in} >$. Each topic can be treated as a random variable with two possible values: involved or not involved. The information entropy of topic t_j to page h_i can be estimated by $H(t_{ij}) = -(p(t_{ij})logp(t_{ij}) + (1 - p(t_{ij}))log(1 - p(t_{ij})))$, where $p(t_{ij})$ is the probability of t_j being involved in h_i. Let $s_i =< h_{i1}, \ldots, h_{ir} >$ be a session, $T_{ij} =< t_{ij_1}, \ldots, t_{ij_n} >$ be the topic vector of page h_{ij} with $1 \leq j \leq r$, $T_{s_i} =< t_{s_i1}, \ldots, t_{s_in} >$ be the topic vector of s_i, and $p(t_{ij_k})$ is the probability of the k^{th} topic being involved in h_{ij}. The probability of the k^{th} topic being involved in s_i denoted as $p(t_{s_ik})$ $(1 \leq k \leq n)$ can be calculated by $\frac{\Sigma_{j=1}^{j=r} p(t_{ij_k})}{r}$, and the information entropy of topic t_j to s_i can be estimated by $H(t_{s_ij}) = -(p(t_{s_ij})logp(t_{s_ij}) + (1 - p(t_{s_ij}))log(1 - p(t_{s_ij})))$.
The average entropy of all topics to the session can be calculated by the following equation:

$$H(s_i) = H(t_{s_i1}, \ldots, t_{s_in}) = (\Sigma_{k=1}^{k=n} H(t_{s_ik}))/n \qquad (4.1)$$

The entropy of a session estimates the certainty of the topics involved in the session. If the entropy is small, then there must be some topics with high probabilities and the others with very low probabilities. It means that the page contents in this session focus on a few topics which clearly exhibit the user's information needs. On the other hand, if the entropy is large, then the probabilities of the topics must be very close and low as well. In this case, it is hard to identify the user's information needs since the pages in this session involve many topics. In this paper, we use the average entropy calculated by Equation (4.1) to prune away sessions that have high entropy. Table 4.1 shows the entropies of the sessions in Table 2.2. The entropy of the first session in the table is higher than the other two sessions. The high entropy of the first session is caused by page A_1 which is the home page of the Web site. Because

A_1 involves many topics (see Table 3.2), it actually provides little information about the visitor's particular information needs. The sessions with high entropy value will be discarded, e.g. the first session in Table 4.1. Table 4.2 gives the clutering results to the original sessions in Table 2.2 and the set of pruned sessions as well. The results show that the entropy of the clusters of the pruned sessions is less than that of the original sessions. It also shows that the entropy is reduced when the page contents are used to evaluate the similarity among pages.

Table 4.1. Sessions and their entropies

User Sessions	Sesion Entropies
$A_1 \rightarrow A_2 \rightarrow A_3 \rightarrow A_4 \rightarrow A_5$	0.53
$A_2 \rightarrow A_3 \rightarrow A_4 \rightarrow A_6$	0.17
$A_7 \rightarrow A_8 \rightarrow A_9 \rightarrow A_{10}$	0.11

Table 4.2. Clustering results using ($\alpha = 0.5$) and without using ($\alpha = 0$) page contents

	α= 0			α=0.5		
	Clusters		Entropy	Clusters		Entropy
Original sessions	1	A_1: /index.jsp A_6: /research/sdl/researchers.jsp A_4: /research/sdl/projects/index.jsp A_2: /research/index.jsp A_3: /research/sdl/index.jsp A_5: /research/sdl/robotsoccer.jsp	0.24	1	A_1: /index.jsp A_4: /research/sdl/projects/index.jsp A_2: /research/index.jsp A_3: /research/sdl/index.jsp A_5: /research/sdl/robotsoccer.jsp	0.19
	2	A_9: /people/index.jsp A_{10}: /people/students.jsp A_7: /pubs/index.jsp A_8: /pubs/2002.jsp		2	A_9:/people/index.jsp A_6: /research/sdl/researchers.jsp A_{10}: /people/students.jsp	
				3	A_7: /pubs/index.jsp A_8: /pubs/2002.jsp	
Pruned sessions	1	A_6:/research/sdl/researchers.jsp A_4: /research/sdl/projects/index.jsp A_2: /research/index.jsp A_3: /research/sdl/index.jsp	0.14	1	A_4: /research/sdl/projects/index.jsp A_2: /research/index.jsp A_3:/research/sdl/index.jsp	0.06
	2	A_9: /people/index.jsp A_{10}: /people/students.jsp A_7:/pubs/index.jsp A_8: /pubs/2002.jsp		2	A_9: /people/index.jsp A_6: /research/sdl/researchers.jsp A_{10}: /people/students.jsp	
				3	A_7: /pubs/index.jsp A_8: /pubs/2002.jsp	

Some experiments have been conducted to test the improvement of Web page clustering by using page contents. In the experiments, a cluster was discarded if its cardinality (N_i) was less than $\rho = 3$. The experimental detail was omitted due to the space limit.

5. Conclusion

Most existing web mining techniques for finding user navigation patterns are access-based approaches that statistically analyze the log data and do not pay much attention on the content of the pages. In this paper we have proposed an approach that takes the page contents into account to find out page clusters. The proposed method prunes away the sessions that involve irrelevant topics by using session entropy which is based on well-established Information Theory. After the pruning, the quality of the clustering is improved since the impact of irrelevant papges has been weakened. There are potentialities of using the user clusters in at least two contexts. On the one hand, the user clusters can be used to assist web users with navigating large web sites by recommending relevant pages that are classified in the category which the user belongs to. On the other hand, the user clusters provide information for the web site desinger to better understand the user needs and how the users visit the site. As a result, the contents or the organization of the web site can be improved to meet the user needs.

References

[1] J. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum Press, New York, 1981.

[2] J. Borges and M. Levene. Data mining of user navigation patterms. In *Proceedings of the Web Usage Analysis and User Profiling*, volume 1, pages 31–36, 1999.

[3] G. Frigui and R. Krishnapuram. Clustering by competivive agglomeration. *Pattern Recognition*, 30(7):1109–1119, 1997.

[4] R. J. Hathaway, J. W. Devenport, and J. C. Bezdek. Relational dual of the c-means clustering algorithms. *Pattern Recognition*, 22(2):205–212, 1989.

[5] R. Kruse, F. Hoppner, F. Klawonn, and T. Runkler. *Fuzzy Cluster Anallysis*. John Wiley and Sons, 1999.

[6] G. Linden, B. Smith, and J. York. Amazon.com recommendations: item-to-item collaborative filtering. *Internet Computing*, 7(1):76–80, 2003.

[7] B. Mobasher. *Web Usage Mining and Personalization (Chapter one in book Practical Handbook ofInternet Computing)*. CRC Press LLC, to appear in 2004.

[8] Nasraoui O., H. Frigui, A. Joshi, and R. Krishnapuram. Mining web access logs using relational competitive fuzzy clustering. In *Proceedings of the Eight International Fuzzy Systems Association World Congress*, volume 1, pages 195–204, 1999.

[9] M. Perkowitz and O. Etzioni. Towards adaptive web sites: Conceptual framework and case study. *Artificial Intelligence*, 118:245 – 275, 2000.