CERIAS Tech Report 2001-21

User Content Mining Supporting Usage Content for Web Personalization

Malika Mahoui[1], Bharat Bhargava[1], Mukesh Mohania[2]
Center for Education and Research in
Information Assurance and Security
[1]Department of Computer Science, Purdue University
West Lafayette, IN 47907
[2]Department of Computer Science, Western Michigan University

User Content Mining Supporting Usage Content Mining for Web Personalization

Malika Mahoui[1], Bharat Bhargava[1], Mukesh Mohania[2]

[1]CERIAS and Department of Computer Science, Purdue University,
West Lafayette, IN, USA 47907
{mmahoui, bb}@cs.purdue.edu
[2]Department of Computer Science, Western Michigan University,
Kalamazoo, MI, USA 49008
mohania@cs.wmich.edu

CERIAS TR 2001-20

Abstract: In Web personalization usage mining has been used in combination of standard methods to help predict user needs based on their transaction histories. Although information in usage logs of a Web server reflects the interests of users to the site, the users are potentially not aware of all information needs that can be addressed through the Web server. User data (content) is a valuable source for discovering new users' needs that can be satisfied by the server. This paper proposes a new framework that combines both usage content mining and user content mining for Web personalization. We use automatically extracted keyphrases as the main features for representing server usage data, server content data and publicly available user data. This uniform representation is used to support a finer granularity for the mining techniques that we use such as clustering and similarity computation.*

1. Introduction

Web personalization has become an important tool for Web applications ranging from one-to-one marketing to Web browsing [1, 3, 7, 13, 15]. Collaborative filtering techniques have been largely used in Recommendation systems to help predict new user needs [15, 18]. Basically, these techniques rely on the user rating of items already used in the server to locate other server users called neighbors who have rated similarly the same items. In a second stage, items that have been highly rated by the neighbors but yet not accessed by the user will be recommended. Although automatic collaborative techniques are popular they suffer from some limitations such as difficulty to scale in presence of increasing number of items while maintaining good prediction performance [13, 14, 15].

Recently new techniques proposed to use Web usage mining to overcome some of these limitations [2, 14, 21, 22]. Traditional mining techniques such as clustering,

* This research is supported by the Center for Education and Research in Information Assurance and Security and NSF grants EIA-9805693, CCR-001788, and CCR-9901712.

association rules and sequential patterns have been applied to mine user transactions histories. While some approaches focused on mechanisms that help better structure Web sites [2, 19], other approaches have been used for discovering aggregate profiles and their application to Web personalization [14]. Compared to collaborative filtering techniques, usage mining has several advantages. Usage logs are automatically recorded by the server and does not interfere with user activities. Information present in the log is an accurate and an objective record of users activities, including navigational user paths through the server, server objects that have been considered by the user (i.e. purchased items). Therefore, subjectivity that is often associated with systems relying on users cooperation is minimized. Data gathered from the log includes a richer information compared to a set of predefined questions that the user is requested to comment on when using collaborative systems.

One main drawback of usage mining based techniques is that it is difficult to precisely specify user intentions. Terminal actions such as downloading a document or purchasing can only support on inferring weekly rules on user satisfaction. More recent approaches exploited usage mining as a support for collaborative filtering techniques [13]. Another drawback associated with usage logs is the relatively limited usage data available for mining. A new approach that integrates Web usage mining with Web content mining in Web personalization is proposed in [14].

A new alternative that we propose which has great potentials on increasing the amount of information available about the user is to include the user content in the mining process. The existing techniques using usage data assume that the user is fully aware of the capabilities of the server as far as his/her needs are. As a consequence, only information recording his/her usage behavior with the system is used; perhaps combined with information of similar user's behaviors [14]. As the size of Web servers continues to increase it is difficult to assume that a user has identified all his/her needs that can be satisfied by a server. To address this issue, we propose a new framework for Web personalization that integrates both usage content and user content to help predict user needs. The main contributions of the paper are as follows:

- Exploiting publicly available information about the user in Web personalization.
- Proposing a new framework for Web personalization that seamlessly integrates usage mining with content mining (server content and user content).
- Using automatically extracted keyphrases as the main feature for representing server usage data, server content data, and user data.
- Using a uniform and features based representation across all types of mined information to support a finer granularity for the mining techniques that we use such as clustering and similarity computation.

The paper is organized as follows. Section 2 presents the motivation of the new approach as well as related work. Section 3 describes the overall architecture of the framework. The different components of the framework are presented in sections 4 and 5. Section 6 includes a summary of the paper and suggestions for future work.

2. User content mining in Web personalization

Motivating example: Consider a large online bookstore (i.e. amazon.com) having a fraction of its users (i.e. 3%) affiliated to education. Apparently these users are making personal purchases of books and are not necessarily related to their work.

The Recommendation system that we propose will use information discovered from usage data, server content and user content to recommend for the user new items (i.e. books). As a member of a university, a user can potentially initiate more purchasing transactions to the server. For example, the analysis of the user's Web page will enable to discover that he/she is teaching three courses, and no book already purchased by the user relates to theses subjects. If this information is used in Web personalization, the system may add recommendation of books that could be adopted as textbooks for the courses; and potentially the user will be convinced with some of them, or at least order personal copies. Similar deduction could be inferred by extracting information about the projects the user is working on.

In this example, information extracted from usage data available at the server side enabled to locate user related documents (i.e. home page) that are publicly available in the Web. Once the documents are retrieved, they are matched against server documents (server content). Matching results are used to find items/services provided by the server not yet explored by the user. Other information extracted from usage data will allow to identify current user needs and match these needs to other users needs to identify an aggregate view of the user behavior. As an example, if the user has previously bought a book on C++, the usage mining process will result on recommending new books on C++ bought by other users or newly arrived ones, and also recommend related books such as data structure books.

Our work is related to the approach proposed by Bhamshad et all. [14]. They integrate usage content and server content mining into a framework that will be used by the recommendation engine. Two main components characterize their system: offline processing when data is preprocessed and aggregate usage profiles and aggregate content profiles are discovered. The online component consists mainly of the recommendation engine: given a user session, and a new request by the user to open a server page (pageview), the system derive pageviews that are related to user session including the requested pageview. The retrieved pages will be added to the requested pageview before its display by the browser. The only information that represents the user is the current session that he/she is running. Information about his/her previous accesses to the server are not explicitly exploited; it is rather diluted within the usage aggregate profiles.

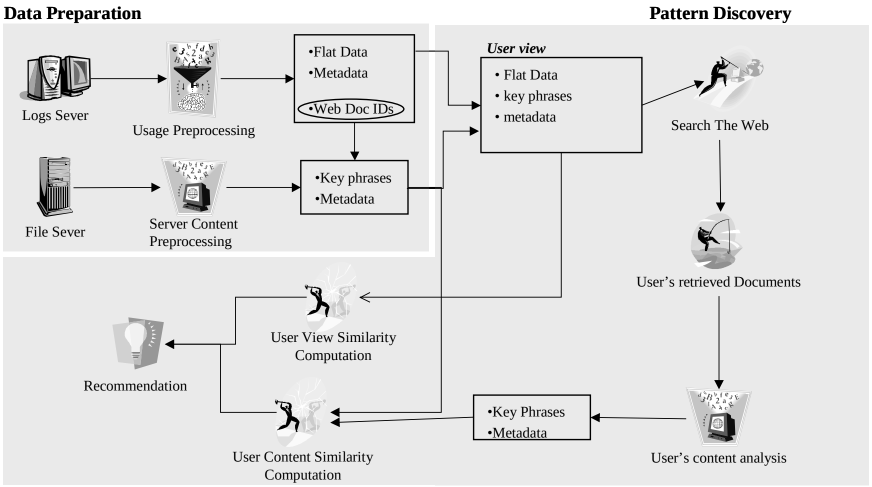
The new approach we propose brings more focus on the user. The usage data is centered around each individual user. Information about the interests of the user is not only limited to the server usage log. Additionally, we gather from the Web all information publicly available about the user (user content). This information is then matched against the server content to identify his/her interests from the server point of view. Unlike in [14] we focus on information extracted from pageviews rather than on the pageviews themselves. Providing such granularity will potentially improve the recommendations of the system. In order to achieve good performance while providing such granularity we use keyphrases automatically extracted from documents (i.e. pageviews) as descriptors for these documents. We use Kea, an automatic keyphrase extraction tool [6], that has shown to provide good quality results while reducing dramatically the vector space representation of documents. Actually, recommendations are drawn offline and they are used to customize the user interface to the server.

The next section presents the overall architecture of the framework. It will be described in more details in following sections.

3. Framework architecture

The overall architecture of the framework is depicted in figure 1. Two main components are emphasized: data preparation and pattern discovery. Data preparation consists of the first two steps of a typical Web mining process namely data selection and data preprocessing [9].

Fig. 1. Framework architecture



Structured data output from the processing of server content data and usage data is fed to the pattern discovery component. In this part of the mining process, we distinguish two main components. The first component utilizes only the user views (see section 4): given a user view, it finds user views that are similar. The second component includes few processing steps: retrieval of user documents from the Web, preprocessing of these documents and computation of their similarity to the server documents. The result of the two components is output as a set of recommendations that will be used to customize the user interface to the server.

4. Data Preparation

4.1. Data selection

The approach we propose is user data centric. The objective is to extract from the server all data that will enable (1) to identify his/her needs already addressed to the server and (2) to help in the process of locating documents in the Web that relate to the user, in order to extract potential needs not yet addressed by the server.

For objective 2, information such as IP address and user domain is not generally sufficient to properly identify the user in the Web. We need to add to this information data that the user has explored in the server to refine the search. For example if the only information that we have about the user will allow to identify a department in a university (i.e. physics department), information extracted from the usage logs showing that the user has purchased books in crystallography will enable to reduce the space of targeted users to those working on the crystallography project. Depending on the quality of data extracted from usage content, at the best we will be able to locate only those documents directly referencing the user; at the worst we will identify the smallest group in the Web to which the user belongs and interacts with.

We define some terminology that will be used throughout the paper:

Server content: similar to the definition of Web content [14], server content refers to the broad range of data extracted from the server. This includes the text extracted from HTML documents, local documents (i.e. pdf, ps files) resulting from search queries or directly accessed from Web pages; and also data residing in local databases accessible from search queries.

User content: user content is similar to server content, except that it refers to information sources extracted from the Web in which the user is involved (i.e. user home page, ps documents authored by the user or citing the user). Note that user content does not refer to information available at the server side.

Usage content: refers to all information that is automatically logged by the server to record users interaction with the server. It also includes information extracted from user profile and other files such as data registry.

User view: refers to all information extracted from usage content that will identify previous needs addressed to the server as well as his/her identity. A view is associated to each user and is the result of the usage content preprocessing (fig. 1).

Data available from the server content and usage content is classified in these categories:

- **Flat data:** which includes raw data related to search queries (search terms and result terms). Search results that are document references (hyperlinks) are not included in this category. Flat data also includes data gathered by the system which describes user preferences and helps in the identification of the user. This information is available from files such as user profile/preferences, registration data, etc.
- **Meta-data**: includes metadata that features user access points in Web pages such as hyperlinks, Web page options (i.e. buttons) etc. This data is extracted from XML and HTML tags. It includes metadata provided by the server to describe non-textual information such as images.

- **Document data**: includes the content of all types of accessible documents from the server such as HTML/XML, ps and pdf documents. In the case of Web pages, document data includes all the information accessible from the page.

We assume that the content of the server is mainly composed of documents from which we can extract text. For non textual documents we assume that the server provides metadata describing the content. We use metadata to describe the content of databases (extracted from the schema of the database)

To summarize, server content is a combination of document data and metadata. Usage content on the other hand includes the three types of data. Note that from the user side, the document data refers to the portion of the server that the user has accessed and for each accessed document, metadata refers to the document access points that the user has effectively clicked on.

The task of processing flat data is not difficult. The challenging task is to determine what part of the server document data is relevant to the user. Using document metadata accessed by the user is a partial solution to the problem. We also must include information that describes the content of the Web document. We use keyphrases automatically extracted from the document as part of the descriptors of the content of the document. Keyphrases are also used as the documents features to compute similarity between server documents and user documents (see section 5).

4.2. Server content preprocessing

Recall that the output of the server content preprocessing will be used to retrieve user related documents and to compute documents similarity (see fig. 1). The later involves the selection of a set of features for the vector representation of documents [17]. We choose to use automatically extracted keyphrases from the documents combined with the documents metadata as the features of the document. We use Kea system [6] to automatically extract keyphrases from each document. Kea uses machine learning techniques to build a model of keyphrases in a collection and then applies this model to identify likely keyphrases within document text. The most frequent keyphrases are included in the document features set. The number of keyphrases extracted from each document is a variable parameter; setting it to the value *five* has provided good results [8, 16].

The advantage of considering most frequent keyphrases instead of a subset of document keywords has many advantages. Some of them are listed in the following:

- Keyphrases include one or more words and therefore they are more likely to capture the semantic of a document compared to using single words. This feature is referred to as lexical affinity in Information Retrieval (IR) [10]. As an example, consider a cars' sale Web server and its clients. It is more interesting to quickly find out that one of the server's users is working on cars' sale field rather than on the broad class of sales. In this case, using keyphrases will speedup the discovery and analysis steps by targeting only those users working on cars' sale.
- KEA uses two attributes as indicators of whether a candidate keyphrase is a good keyphrase. These attributes are the distance from the start of the

document that the keyphrase first occurs and its tf.idf score. The tf.idf (term frequency x inverse document frequency) is a standard retrieval metric that discriminates between rare and commonly occurring keyphrases. These two attributes have been shown to be good measures in determining documents' keyphrases as it is demonstrated in [8, 16].

- Metadata is an important feature to integrate especially for representing non textual information such as images. As metadata include Web page access points (hyperlinks), they are considered as good descriptors of the content of documents.

In order to use these features for the similarity computation we select appropriate feature weights. We choose to use the standard retrieval metric tf.idf as keyphrase weight as it is commonly used in IR.

Regarding metadata features, we assume as in [14] that appropriate weights are given to the metadata by the server maintainer. Tf.idf scores combined to metadata weights are then properly normalized.

Given a server document *sd* its vector representation is given by:

$$\text{Matrix}(sd) = <w(f_1,sd),w(f_2,sd), \ldots, w(f_i,sd), \ldots, w(f_k,sd)>$$

Where $w(f_i,sd)$ corresponds to the weight of the feature f_i in document *sd*; k being the total number of features extracted from the server. Document vectors are stored in an inverted file containing a dictionary of all extracted features and posting files for each feature specifying the documents in which it appears and its corresponding weights. We also output a file for which each entry contains a document id along with the list of features that have been extracted from the document.

4.3. Usage preprocessing

This part includes the standard steps in usage preprocessing, namely data cleaning, and identification of users, user sessions, user transactions, and path completion [4]. In this step, we adapt our work in transaction log analysis [5, 11] to integrate new solutions that have been proposed in usage preprocessing. Indeed, with the recent advent in Web browsing technologies, such as local caches and proxy servers, the identification of users and the set of actions they perform during a session has become difficult. Even the identification of a user session is a difficult problem [4].

As part of data cleaning we identify the data that is relevant to our objectives namely flat data, metadata. Document data is not explicitly stored in the usage content. Regarding metadata extracted from Web pages, we consider only the hyperlinks. Hyperlink labels will be added to the list of flat data. Hyperlink references are used to retrieve the document ids.

In user identification one needs to achieve two objectives: being able to distinguish the actions performed by each user and being able to explicitly identify a user in the Web community.

Regarding the first objective we integrate the heuristics proposed in [4] to enhance the capabilities of the user identification system that we used in [5, 11]. We need to distinguish normal users from robots. We use the approach presented in [20] to detect robot actions and filter out this class of users.

Achieving the second objective relies on the availability of data gathered from the user and stored in files such as user profiles and data registry files. Data gathered about the identity of user varies from simple information about the domain of the machine used to connect to the server to a more detailed description of the user identity (name, address, etc.).

Once the users have been identified, the sequence of actions performed by each user, also called click-stream, needs to be broken down into sessions. As in [5, 11, 14], we assume that a session is a set of actions performed by the same user with no more than thirty minutes lapse between two actions. Finally, we consider a user transaction as the set of all actions performed within a session; that is a user transaction corresponds to a user session.

In a user transaction we distinguish two main groups of data. The first group contains only flat data, including hyperlink labels that have been accessed during user sessions. The second group contains identifiers of server documents that have been accessed during the sessions. Document identifiers need to be combined with the results of server prepossessing step in order to get the keyphrases and metadata of the documents referenced by the user.

5. Pattern discovery

Data output from server content and usage content preprocessing are used in the discovery process. Recall that the objectives are twofold. For each active user:

- Retrieve other users that present similar usage behavior. In other words given a user view, retrieve other user views that are similar.
- Retrieve user documents from the Web and search their similarity to the server documents.

The two objectives are referred below as usage content mining and user content mining respectively.

Common to the two objectives, is the step that consists of replacing all documents ids in user transactions by the set of features extracted from the documents. These features are a combination of metadata and keyphrases. The result added to the other flat data will compose the view of the user. In the rest of the paper user view refers only to server data (keyphrases, metadata, etc.). The information that enables to identify the user will be used only to retrieve user documents in the Web.

5.1. Usage content mining

The problem of retrieving for each user view the set of similar user views that are similar is comparable to the problem of document retrieval in IR which can be formulated as follows: given a user query, what are the documents that match best the terms of the query. Similarity between the query and the documents is computed. The results are ranked and the most significant ones are returned to the user. Using vector space representation model, both the query and documents have the same vector dimensionality computed over the set of features (i.e. keywords). Our problem can be formulated as follows: given a user view *uv*, what are the other user views that are most similar. Two approaches can be adopted for retrieving the set of similar user views.

- Standard approach: compute the similarity between *uv* and every user view.
- Clustering approach: the set of user views are first aggregated into clusters. Then we compute how close is *uv* to each user views cluster.

The standard approach is more precise to identifying individual users that have common interests with the user. The approach may potentially generate a large number of user views that are similar from which a large amount of data is extracted. The solution is to set a high threshold value to extract only the most important information. The clustering approach allows to capture a more global view about the general tendency of user interests and aggregate them into groups. We choose to work with the clustering approach.

The first step consists of applying a clustering algorithm such as k-means to the set of user views. Then a representative is derived from each cluster. The objective is to find a vector representation of the cluster over the features (keyphrases, metadata) space, that can be used in the similarity computation. As in [14], we derive a representative for each cluster by computing its vector mean. Given a cluster C, its vector representation is given by:

$$\text{Matrix(C)} = <m(f_1,C),m(f_2,C), \ldots, m(f_i,C), \ldots, m(f_k,C)>$$

Where $m(f_i,C)$ corresponds to the mean value (weight) for the feature f_i in cluster *C*. This value corresponds to:

$$m(f_i,C)=\frac{1}{|C|}\times\sum_{uv\in C} w(f_i,uv)$$

Once a representative of each cluster is derived we proceed to the similarity computation. For each user view we compute its similarity to each cluster representative. Using the cosine measure, the similarity between a user view ***uv*** and a cluster representative ***uvc*** is given by:

$$\text{Sim(uv, uvc)} = \sum_{i=1}^{k} w(fi,uv)\times w(fi,uvc) \Bigg/ \sqrt{\sum_{i=1}^{k} w^2(fi,uv)\times \sum_{i=1}^{k} w^2(fi,uvc)}$$

Score values exceeding some threshold value are selected. User views clusters associated to these scores are output. The analysis of these clusters basically consists of extracting from their vector representation the features that have not yet been explored by the user and include them as part of the recommendation (fig. 1). Another filtering might occur at this level if the size of the features extracted set is large.

5.2. User content mining

Three main steps are performed in this phase. First, locating user related documents, then preprocessing them to extract documents’ features; and finally perform similarity computation between users' documents and server's documents. Interesting patterns are output as a part of the recommendation set to the user.

Locating user documents

Results output from the usage preprocessing step includes two main types of data: user identification data and server data accessed by the user (user view)

Recall that the objective is to retrieve as much information about the user in order to determine his/her needs that have not been addressed to the server and make appropriate suggestions. Thus, the main information that is used to locate user documents from the publicly indexable Web is the user identification data. User view data is used to refine the search when user identification data is not sufficient.

For this process, we use multiple search engines such as AltaVista and Google; and we derive multiple queries to tailor the searching features of each Web crawler. The results returned from each search engine are combined and duplicates are eliminated. Documents associated to the query results are stored temporarily in the server for further processing. Documents that are Web pages are processed recursively to get all accessible pages using the hyperlink references. The resulting documents are referred to as user's retrieved documents in figure 1.

User's retrieved documents analysis

User's retrieved documents are processed in a similar way as we did for server content documents. For each document we extract the first *five* keyphrases (when available). Metadata is reduced to the hyperlink references as it is difficult to obtain the metadata describing the non-textual information. Features that are not part of the server vocabulary are discarded as the purpose is to find only user documents that are relevant to the server content. As a result, we associate to each user's retrieved document a set of features as document descriptors. The next step consists of attributing weights to document features. There are two main alternatives:

- Alternative 1: Consider user's retrieved documents as a collection; and attribute weights to features within the user collection.
- Alternative 2: Add user's retrieved documents to the server collection. Features weights are computed using server documents and potentially user documents.

Selecting one of the above alternatives mainly depends on the size of the user collection. In the rest of this section we assume that alternative 2 is used.

We use tf.idf scores as feature weights. And we associate to each user document *ud* its vector representation given by:

$$\text{Matrix(ud)} = <w(f_1,ud),w(f_2,ud), \ldots, w(f_i,ud), \ldots, w(f_k,ud)>$$

Where $w(f_i,ud)$ corresponds to the weight of the feature f_i in document *ud*. Note that k is the total number of features extracted from the server; this will simplify the computation of similarity measures between user documents and server documents.

Similarity computation

The final step in the mining process consists of measuring the similarity between each user's retrieved document and each of the server documents. We use the cosine measure to compute the similarity between a user retrieved document ***ud*** and every server document ***sd*** as given by the following:

$$\text{Sim(sd, ud)} = \sum_{i=1}^{k} w(fi, ud) \times w(fi, sd) \Bigg/ \sqrt{\sum_{i=1}^{k} w^2(fi, ud) \times \sum_{i=1}^{k} w^2(fi, sd)}$$

Score values exceeding some threshold value are selected. User retrieved documents associated to these scores are selected.

The next step in the analysis consists of extracting a set of most important features from the selected user documents such that none of theses features has been referenced in previous access made by the user. The importance of a feature is determined by computing its mean over all its occurrences across the set of selected user documents. The features mean values are then used to order the features and the *n* most important ones are included as part of the system's recommendation (figure 1).

6. Conclusion

We have presented a framework for Web personalization that integrates both usage data available from the Web server and user data gathered from the Web to better predicting user needs. Usage data helps in improving the recommendations that target the user's current needs; while user data helps the user discover new services/information that the server provides to addressing his/her needs that have not yet been explicitly expressed to the server.

Several Web related issues can be addressed using the framework. In Web security for example, combining usage content mining with user content mining can help in detecting misusage of the server information by the user [12].

In this framework we exploit previous techniques that we developed in Web usage mining and Web content mining. The integration of the two components does not seem to present any major problem as it will be confirmed in the implementation phase. We plan to conduct a performance evaluation study to assess the effectiveness of the documents features. The framework suggests several directions for future work. We plan to investigate new heuristics for user views' clustering and similarity computation. The objective is to reduce the average response time of the mining process while maintaining a good quality of the information that is recommended to the user. We also plan to integrate the notion of time in the mining process to better filter out the information extracted from the usage data and also from the user data gathered from the Web. Currently the framework is devised for an offline recommendation process. The next step would be to integrate an online component to provide a recommendation service for ad-hoc user requests for Web services.

7. References

1. J. Ben Schafer, J. Konstan, J. Riedl. Recommender Systems in E-Commerce. In Proc. of the ACM Conference on E-Commerce (EC-99), pp 158-166, 1999.

2. B. Berendt. Web Usage Mining, Site Semantics, and the Support of Navigation. In Workshop on Web Mining for E-Commerce (WEBKDD00), Boston, 2000.
3. A. Buchner, M. D. Mulvenna. Discovering Internet Marketing Intelligence through Online Analytical Web Usage Mining. SIGMOD Record, (4) 27, 1999.
4. R. Cooley, B. Mobasher, J. Srivastava. Data Preparation for Mining World Wide Web Browsing Patterns. Journal of Knowledge and Information Systems, (1) 1, 1999.
5. S.J. Cunningham, M. Mahoui. Search Behavior in Two Digital Libraries: A Comparative Transaction Log Analysis. Proc. of the 3rd International Conference on Asian Digital Libraries ICADL2000, Seoul, Korea, pp 193-200, 2000.
6. E. Frank, G. Paynter, I. Witten, C. Gutwin, C. Nevill-Manning. Domain-Specific Keyphrase Extraction. Proc. of the 16th Int. Joint Conference on Artificial Intelligence, Morgan-Kaufmann, pp 668-673, 1999.
7. S. Gomory, R. Hoch, J. Lee, M. Podlaseck, E. Schonberg. E-Commerce Intelligence: Measuring, Analyzing, and Reporting on Merchandising Effectiveness of Online Stores. Technical Report, IBM T. J. Watson Research Center, 1999.
8. S. Jones, M. Mahoui. Hierarchical Document Clustering Using Automatically Extracted Keyphrases. Proc. of the 3rd International Conference on Asian Digital Libraries ICADL2000, Seoul, Korea, pp 113-120, 2000.
9. R. Kosala, H. Blockee. Web Mining Research: A Survey. SIGKDD Explorations, 2000.
10. Y. Maarek, I.Z. Ben Shaul. Automatically Organizing Bookmarks per Contents. Journal of Computer Networks and ISDN Systems, Volume 28, issues 7–11, p. 1321, 1997.
11. M. Mahoui, S.J. Cunningham. A Comparative Transaction Log Analysis of Two Computing Collections. Proc. of the 4th European Conference on Research and Advanced Technology for Digital Libraries ECDL2000, Lisbonne, 2000.
12. M. Mahoui, B. Bhargava, M. Mohania. Data Mining For Web Security: UserWatcher. CERIAS Technical Report, Purdue University, 2001.
13. B. Mobasher, H. Dai, T. Luo, M. Nakagawa. Improving the Effectiveness of Collaborative Filtering on Anonymous Web Usage Data. To appear in Proc. of the IJCAI 2001 Workshop on Intelligent Techniques for Web Personalization (ITWP01), Seattle, 2001.
14. B. Mobasher, et al. Integrating Web Usage and Content Mining for More Effective Personalization. EC-Web 2000. Proc. of the International Conference on E-Commerce and Web Technologies (ECWeb2000), Greenwich, UK, 2000.
15. M. O'Conner, J. Herlocker. Clustering items for collaborative filtering. Proc. of the ACM SIGIR Workshop on Recommender Systems, Berkeley, CA, 1999.
16. G. W. Paynter, I. H. Witten, S. J. Cunningham. Evaluating Extracted Phrases and Extending Thesauri. Proc. of the 3rd International Conference on Asian Digital Libraries. Seoul, Korea, 2000.
17. G. Salton. Automatic Text Processing: The Transformation, Analysis and Retrieval of Information by Computer. Addison-Wesley, 1989.
18. B. Sarwar, G. Karypis, J. Konstan, J. Reidl. Analysis of Recommendation Algorithms for E-commerce. Proc. of the International Conference on E-Commerce, Minneapolis, 2000.
19. M. Spiliopoulou, C. Pohle, L. C. Faulstich. Improving the Effectiveness of a Web site with Web Usage Mining. In workshop on Web Usage Analysis and User Profiling (WebKDD99), San Diego, 1999.
20. P. Tan, V. Kumar. Discovery of Web Robot Sessions based on their Navigational Patterns. Technical Report, University of Minnesota, 2001.
21. P. S. Yu. Data Mining and Personalization Technologies. Proc. of the International Conference on Database Systems for Advanced Applications (DSFAA1999), Hsinchu, Taiwan, 1999.
22. O. R. Zaiane, M. Xin, and J. Han. Discovering Web access Patterns and Trends by Applying OLAP and Data Mining Technology on Web Logs. In Advances in Digital Libraries, pp. 19-29, Santa Barbara, 1998.