User Modeling and User-Adapted Interaction **13:** 311–372, 2003.

Web Usage Mining as a Tool for Personalization: A Survey

DIMITRIOS PIERRAKOS[1], GEORGIOS PALIOURAS[1],
CHRISTOS PAPATHEODOROU[2] and CONSTANTINE D. SPYROPOULOS[1]
[1]*Institute of Informatics and Telecommunications, NCSR 'Demokritos', Ag. Paraskevi, GR 15310 Greece. e-mail: paliourg@iit.demokritos.gr*
[2]*Department of Archive and Library Sciences, Ionian University, Corfu, GR 49100 Greece*

(Received 21 January 2001; accepted in revised form 18 April 2003)

Abstract. This paper is a survey of recent work in the field of web usage mining for the benefit of research on the personalization of Web-based information services. The essence of personalization is the adaptability of information systems to the needs of their users. This issue is becoming increasingly important on the Web, as non-expert users are overwhelmed by the quantity of information available online, while commercial Web sites strive to add value to their services in order to create loyal relationships with their visitors-customers. This article views Web personalization through the prism of personalization policies adopted by Web sites and implementing a variety of functions. In this context, the area of Web usage mining is a valuable source of ideas and methods for the implementation of personalization functionality. We therefore present a survey of the most recent work in the field of Web usage mining, focusing on the problems that have been identified and the solutions that have been proposed.

Key words. data mining, machine learning, personalization, user modeling, web usage mining

1. Introduction

Interest in the analysis of user behavior on the Web has been increasing rapidly. This increase stems from the realization that added value for Web site visitors is not gained merely through larger quantities of data on a site, but through easier access to the required information at the right time and in the most suitable form.

Estimates of Web usage expect the number of users to climb up to 945 million by 2004 (Computer Industry Almanac, May 2003). The majority of these users are non-expert and find it difficult to keep up with the rapid development of computer technologies, while at the same time they recognize that the Web is an invaluable source of information for their everyday life. The increasing usage of the Web also accelerates the pace at which information becomes available online. In various surveys of the Web, e.g. (Chakrabarti, 2000), it is estimated that roughly one million new pages are added every day and over 600 GB of pages change per month. A new Web server providing Web pages is emerging every two hours. Nowadays, more than three billion Web pages are available online; almost one page for every two people on the earth (UsaToday, April 2003). In the above, one notices the emergence of a spiral effect, i.e., increasing number of users causing

an increase in the quantity of online information, attracting even more users, and so on. This pattern is responsible for the 'explosion' of the Web, which causes the frustrating phenomenon known as '*information overload*' to Web users.

Moreover, the emergence of e-services in the new Web era, such as e-commerce, e-learning and e-banking, has changed radically the manner in which the Internet is being used, turning Web sites into businesses and increasing the competition between them. With competitors being 'one-click away', the requirement for adding value to e-services on the Web has become a necessity towards the creation of loyal visitors-customers for a Web site. This added value can be realized by focusing on specific individual needs and providing tailored products and services.

The personalization of services offered by a Web site is an important step in the direction of alleviating information overload, making the Web a friendlier environment for its individual user and hence creating trustworthy relationships between the Web site and the visitor-customer. In (Mobasher et al., 1999a) *Web Personalization* is simply defined as the task of making Web-based information systems adaptive to the needs and interests of individual users. Typically, a personalized Web site recognizes its users, collects information about their preferences and adapts its services, in order to match the users' needs. Web personalization improves the Web experience of a visitor by presenting the information that the visitor wants to see in the appropriate manner and at the appropriate time. In e-business, Web personalization additionally provides mechanisms to learn more about customer needs, identify future trends and eventually increase customer loyalty to the provided service.

Hitherto, Web personalization has been related mainly with recommender systems and customized Web sites for newspapers, electronic shops, etc. (Schafer et al., 2001), (Pretschner and Gauch, 1999). However, we claim that Web personalization comprises a variety of functions ranging from simple user recognition to more advanced functionality, such as performing certain tasks on behalf of the user. This functionality is offered by the Web personalization systems according to a personalization policy that specifies the manner in which personalization will be delivered to the final user.

In the majority of the existing commercial personalization systems, the personalization process involves substantial manual work and most of the time, significant effort on the part of the user. Despite these problems, the number of personalized Web pages is increasing. A survey by Datamonitor (2001) predicts that global investment in personalization technologies will reach $2.1 billion in 2006, where half of the investment will be made by firms in the financial services and retail sectors. Personalization technologies are also popular with telecommunications and entertainment firms. Other surveys reflect the adoption of this new technology by users. According to a poll conducted by Cyber Dialogue (2001) 56% of the participants said that 'they are more likely to shop at a site that offers personalization' and 63% were 'more likely to register for a site that offers personalization.'

One way to expand the personalization of the Web is to automate the adaptation of Web-based services to their users. Machine learning methods have a successful record of applications to similar tasks, i.e., automating the construction and adaptation of

information systems (Langley, 1999; Pohl, 1996 and Webb et al., 2001). Furthermore, the incorporation of machine learning in larger process models, such as that of Knowledge Discovery in Data (KDD or Data Mining), can provide a complete solution to the adaptation task. Knowledge Discovery in Data has been used to analyze data collected on the Web and extract useful knowledge. This effort was named *Web Mining* (Etzioni, 1996) and one branch of it is concerned with the analysis of usage data, i.e., records of how a Web service is used. This process is called *Web Usage Mining* and it is the focus of this paper.

Early work in Web usage mining (Srivastava et al., 2000) did not consider extensively its use for personalization. Its primary focus was on the discovery of decision-support knowledge, expressed in terms of descriptive data models to be evaluated and exploited by human experts. However, Web usage mining can also be a useful tool for Web personalization. All that is required for the application of Web usage mining to Web personalization is a shift of focus from the traditional, decision-support knowledge discovery, i.e., the static modeling of usage data, to the discovery of operational knowledge for personalization, i.e., the dynamic modeling of users. This type of knowledge can be directly delivered back to the users in order to improve their experience in the site, without the intervention of any human expert. Thus, it is now widely recognized that usage mining is a valuable source of ideas and solutions for Web personalization.

Based on this view of the Web usage mining process, we present here a survey of recent work for the benefit of research in Web personalization. Starting with an analysis of the Web personalization concept and its relation to Web usage mining, the emphasis subsequently, is on the methodology adopted in Web usage mining, the various solutions that have been presented in the literature and the way in which these methods can be applied to Web personalization systems. In reading the survey, the reader should bear in mind that Web usage mining is not a mature research area. As a result, the survey addresses also many open issues, both at a technical and at a methodological level.

The structure of the survey is as follows. Section 2 presents a roadmap of Web personalization comprising the functionality that can be offered by a Web personalization system, the design requirements of such a system, as well as the main approaches presented so far in the Web personalization literature. Section 3 presents the basic ideas behind the process of Web usage mining and its use for Web personalization. The next three sections (4 to 6) present in detail the three stages of the Web usage mining process, i.e., data collection, data preprocessing and pattern discovery, examining the majority of the existing methods employed in each stage. An important aspect of this survey is that it is not restricted to a mere examination of machine learning methods for pattern discovery, but it examines the issues that arise in all three stages of the mining process, which are of direct relevance to Web personalization. Section 7 presents the main parameters to be considered when designing the personalization policy of a site, introduces a number of Web personalization systems that adopt the approach of Web usage mining, and concludes with suggestions for the expansion of this type of personalization system. Finally, Section 8 summarizes the most interesting conclusions of this survey and presents promising paths for future research.

2. Web Personalization Roadmap

The term Web personalization encompasses methods and techniques that are used to deliver a value-added browsing experience to the visitors of a Web site. This value is attained by a variety of functions that can be offered by a Web personalization system, which make the interaction with the Web site easier, saving users' time, and hence satisfying one of the main goals of Web sites: the creation of loyal visitors. In the following subsections, we examine the personalization functions that can be offered, together with a set of requirements for the design and implementation of a Web personalization system.

2.1. PERSONALIZATION FUNCTIONS

A Web personalization system can offer a variety of functions starting from simple user salutation, to more complicated functionality such as personalized content delivery. Kobsa et al. (2001) recommends a classification of the Web personalization functions, which is extended here to a generic classification scheme. The proposed scheme takes into account what is currently offered by commercial systems and research prototypes, as well as what is potentially feasible by such systems. We distinguish between four basic classes of personalization functions: *memorization, guidance, customization* and *task performance support*. Each of these is examined in more detail below.

2.1.1. *Memorization*

This is the simplest form of personalization function, where the system records and stores in its 'memory' information about the user, such as name and browsing history. When the user returns to the site, this information is used as a reminder of the user's past behavior, without further processing. Memorization, is usually not offered as a stand-alone function, but as part of a more complete personalization solution. Examples of this class of functions are the following:

User Salutation: The Web personalization system recognizes the returning user and displays the user's name together with a welcome message. Various commercial sites employ salutation for their customers or registered users. Though this is a simple function, it is the first step towards increased visitor loyalty, since users feel more comfortable with Web sites that recognize them as individuals, rather than regular visitors.

Bookmarking: The system stores the Web pages that a user has visited in the past and presents them to the user by means of a personalized bookmarking schema for that site.

Personalized access rights: A Web site can use personalized access rights, in order to separate authorized users from common users. Different access rights may be required for different types of information, such as reports or product prices, or even for the execution of particular Web applications, such as ftp, or e-mail.

2.1.2. *Guidance*

Guidance as a personalization function refers to the endeavor of the personalization system to assist the user in getting quickly to the information that the user is seeking in a site, as well as to provide the user with alternative browsing options. This personalization function not only increases the users' loyalty but also alleviates in a great extent the information overload problem that the users of a large Web site may face. Examples of guidance functionality are the following:

Recommendation of hyperlinks: This function refers to the recommendation of a set of hyperlinks that are related to the interests and preferences of the user. The presentation of the recommended links is done either in a separate frame of the Web page or in a pop-up window. In (Kobsa et al., 2001), this function is described as adaptive recommendation and can take the form of recommendation of links to certain products, topics of information, or navigation paths that a user might follow. Recommendation of hyperlinks is one of the most commonly offered Web personalization functions, and is supported by a number of systems such as the *WebPersonalizer* (Mobasher et al., 2000b).

User tutoring: This functionality borrows the basic notion of Adaptive Educational Systems, and applies it to Web sites. A personalized site can offer guidance to an individual at each step of the user's interaction with the site, according to the user's knowledge and interests. This is achieved by either recommending other Web pages, or by adding explanatory content to the Web pages. An application of this function can be found in *Webinars* (Web seminars), which are live or replayed multimedia presentations conducted from a Web site.

2.1.3. *Customization*

Customization as a personalization function refers to the modification of the Web page in terms of content, structure and layout, in order to take into account the user's knowledge, preferences and interests. The main goal is the management of the information load, through the facilitation of the user's interaction with the site. Examples of this class are:

Personalized layout: This is a functionality inherited from Adaptive User Interfaces, where a particular Web page changes its layout, color, or the locale information, based on the profile of the user. This function is usually exploited by Web portals, such as Yahoo and Altavista, which are offering customized features in order to create personalized 'MyPortal' sites.

Content Customization. The content of the Web page presented to a user may be modified in order to adjust to the user's knowledge, interests, and preferences. For example, the same page can be presented to different users, in a summarized, or an extended form depending on the type of the user. An example of such a customized Web site is the UM2001 site (Schwarzkopf, 2001).

Customization of hyperlinks. Customization can also apply to the hyperlinks within a page. In this case, the site is modified by adding or removing hyperlinks within a particular page. This can lead to the optimization of the whole Web site structure by removing links that are unusable and modifying the site's topology to make it more usable.

Personalized pricing scheme. The Web site can provide different prices and payment methods to different users, such as discounts or installments to users that have been recognized by the site as loyal customers. An attempt of providing functionality similar to that was performed by amazon.com, which charged different customers with different prices for the same product. However, the attempt was legally challenged, due to the failure of communicating and justifying the reasons behind the price differences. Together with hyperlink recommendation, this functionality can also be employed by e-commerce sites to attract visitors that are not currently buyers.

Personalized product differentiation. In marketing terms, personalization can be a powerful method of transforming a standard product into a specialized solution for an individual.

2.1.4. *Task Performance Support*

Task performance support is a functionality that involves the execution of a particular action on behalf of a user. This is the most advanced personalization function, inherited from a category of Adaptive Systems known as personal assistants (Mitchell et al., 1994), which can be considered as client-side personalization systems. The same functionality can be envisaged for the personalization system employed by a Web server. Examples of this class of functions are:

Personalized Errands. The Web personalization system can perform a number of actions and assist the work of the user, such as sending an e-mail, downloading various items, etc. Depending on the sophistication of the personalization system, these errands can vary from simple routine actions, to more complex ones that take into account the personal circumstances of the user.

Personalized Query Completion. The system can either complete or even enhance, by adding terms, the queries of a user submitted either to a search engine, or to a Web database system. In this way, personalization can help in improving the performance of an information retrieval system.

Personalized Negotiations. The Web personalization system can act as a negotiator on behalf of a user and participate, for example, in Web auctions, (Bouganis et al., 1999). This is one of the most advanced task performance functions, requiring a high degree of sophistication by the personalization system, in order to earn the trust of the users.

2.2. REQUIREMENTS FOR THE DESIGN OF A WEB PERSONALIZATION SYSTEM

As described in the previous section, a variety of functions can be employed by a Web personalization system. These functions impose a number of requirements on the design of a personalization system, which aim at the development of a robust and flexible system. The following is a list of such generic requirements:

2.2.1. *Domain Specification*

The functionality offered by a Web personalization system is domain-sensitive. The same system operating in different domains, e.g. e-commerce, digital library, portal, etc., might offer different personalization functions. Thus, the domain under which the personalization system will operate should be specified and described thoroughly.

2.2.2. *User Identification*

The identification of the user who is accessing a Web site is important, in order to distinguish returning visitors from first-time visitors. However, due to privacy concerns and anonymity of the Web this is not always possible. What may be possible though is the identification of the goal, the objectives or the motivation of the user who is accessing the site. This is a critical issue, since accurate estimates of the user's objectives will trigger the proper personalization function, leading to an improved browsing experience.

2.2.3. *Efficient Acquisition of User Data*

A personalization system should be able to collect all the relevant user data that will be needed for personalization. The type and quantity of data needed depends on the personalization functions that are chosen to be performed. The collection of user data is a continuous process, while due to the nature of the Web the system should be able to handle both large volumes of data and also increased data volatility.

2.2.4. *Flexible Data Elaboration*

The collected data should be processed, in order to separate noise from relevant data, correlate and evaluate them and finally format them so as to be ready for personalization. Data elaboration is a domain-dependent process and thus high flexibility is required for the personalization system to be able to adjust to different domains and corresponding personalization functions.

2.2.5. *Efficient Construction of User Models*

The Web personalization system should be able to create and maintain efficiently and accurately the user models, i.e., the information that the system holds about the interests,

the knowledge, the objectives and the preferences of the users. The construction of the user models may either be done manually, or via a machine learning method. The manual construction process involves the self-specification of a model by a user and/or the analysis of user data by an expert in order to construct rules for classifying users into different types with specific characteristics. The automated construction of user models exploits machine learning techniques in order to create and describe the user models. The choice of manual construction or the use of a learning method depends, to a large extent, on the application domain and the functions that are offered.

2.2.6. *Practical and Legal Considerations*

Personalization functionality should be pertinent with the user's goals and objectives, and adhere to various important practical constraints, e.g. response time. Furthermore, the user's personal information should be protected at all times, while the user should be aware of the way in which such information is being collected and used. These are important issues in the design of a Web personalization system.

2.3. APPROACHES TO WEB PERSONALIZATION

During the evolution of the Web, personalization has been recognized as a remedy to the information overload problem and as a means of increasing visitor loyalty to a Web site. Due to the importance of personalization for Web-based services, several Web personalization techniques have been proposed in the past few years. Although it is not in the scope of the survey to present these techniques in detail, a brief overview of the most influential approaches is presented below.

Mobasher et al. (2000a) classify Web personalization techniques into three generic approaches:

(a) *Manual decision rule systems*. According to this approach, a Web-based service is personalized via manual intervention of its designer and usually with the cooperation of the user. Typically, static user models are obtained through a user registration procedure and a number of rules are specified manually concerning the Web content that is provided to users with different models. Two examples from a wide range of products that adopt this approach are Yahoo!'s personalization engine (Manber et al., 2000) and *Websphere Personalization* (IBM).
(b) *Content-based filtering systems*. This group of techniques applies machine learning methods to Web content, primarily text, in order to discover the personal preferences of a user. A tool that adopts this approach is NewsWeeder (Lang, 1995), which is able to adaptively construct user models from a user's browsing behavior, based on the similarity between Web documents containing news items. These models can be used to filter news items according to each user's requirements.
(c) *Social or collaborative filtering systems*. The aim of this approach is to personalize a service, without requiring the analysis of Web content. Personalization is achieved by searching for common features in the preferences of different users, which are

usually expressed explicitly by them, in the form of item ratings, and are recorded by the system. The *Recommendation Engine* (Net Perceptions) and *Websphere Personalization* (IBM) are examples of products that use also this method, while its most renowned application is in the amazon.com electronic shop.

Manual decision rule systems suffer from the same problems as other manually constructed complex systems, i.e., they require considerable effort in their construction and maintenance. Furthermore they usually require the user's involvement, which is a considerable disincentive for using the system.

The two automatic filtering approaches attempt to alleviate these problems through the use of machine learning techniques, which help in analyzing Web data and constructing the required user models. Their difference is one of emphasis. Content-based filtering applies learning techniques to the content of Web pages, i.e., the focus is on *what* the user is interested in. Collaborative filtering on the other hand is based on similarities between users, i.e., it focuses on *who* else is interested in the same things as the user.

The main problem with content-based filtering is the difficulty of analyzing the content of Web pages and arriving at semantic similarities. Even if one ignores multimedia content, natural language itself is a rich and unstructured source of data. Despite the significant process achieved in the research fields that deal with the analysis of textual data, we are still far from getting a machine to understand natural language the way humans do. Content-based filtering adopts a variety of statistical methods for the extraction of useful information from textual data, e.g. the TF-IDF vector representation (Salton, 1989) and Latent Semantic Indexing (Deerwester et al., 1990). Nevertheless, the problem of the analysis of Web content still remains and becomes even more critical when there is limited textual content. By reducing the emphasis on Web content, collaborative filtering addresses this important problem. Furthermore, collaborative filtering methods facilitate the exploitation of usage patterns that are not confined within strict semantic boundaries.

However, collaborative filtering methods are not free of problems either. Users that first rate new items cannot be given recommendations at all. In addition, the quality of the recommendation depends on the number of ratings that a particular user has made, leading to low quality recommendations for users that have rated a small number of items. Furthermore, collaborative filtering methods that use solely memory-based learning approaches, suffer from two additional problems: they do not scale well to large numbers of users and they do not provide any insight as to the usage patterns that existed in the data (Pennock et al., 2000). Recently, these problems have started to be addressed, by the development of model-based collaborative filtering methods (Breese et al., 1998) and hybrids of model and memory-based methods (Pennock et al., 2000).

3. The Role of Web Usage Mining in Personalization

During the last years, researchers have proposed a new unifying area for all methods that apply data mining to Web data, named *Web Mining*. Web mining tools aim to extract

knowledge from the Web, rather than retrieving information. Commonly, Web mining work is classified into the following three categories (Cooley et al., 1997b, Kosala and Blockeel, 2000): *Web Content Mining*, *Web Usage Mining* and *Web Structure Mining*. Web content mining is concerned with the extraction of useful knowledge from the content of Web pages, by using data mining. Web usage mining, aims at discovering interesting patterns of use, by analyzing Web usage data. Finally, Web structure mining is a new area, concerned with the application of data mining to the structure of the Web graph.

Web mining is a complete process rather than an algorithm. In the case of Web usage mining this process results in the discovery of knowledge that concerns the behavior of users. Originally, the aim of Web usage mining has been to support the human decision making process and, thus, the outcome of the process is typically a set of data models that reveal implicit knowledge about data items, like Web pages, or products available at a particular Web site. These models are evaluated and exploited by human experts, such as the market analyst who seeks business intelligence, or the site administrator who wants to optimize the structure of the site and enhance the browsing experience of visitors.

Despite the fact that the bulk of the work in Web usage mining is not concerned with personalization, its relation to automated personalization tools is straightforward. The work on Web usage mining can be a source of ideas and solutions towards realizing Web personalization. Usage data, such as those that can be collected when a user browses a specific Web site, represent the interaction between the user and that particular Web site. Web usage mining provides an approach to the collection and preprocessing of those data, and constructs models representing the behavior and the interests of users. These models can be used by a personalization system automatically, i.e., without the intervention of any human expert, for realizing the required personalization functions. This type of knowledge, i.e., the user models, constitutes *operational knowledge* for Web personalization. Hence, a Web personalization system can employ Web usage mining methods in order to achieve the required robustness and flexibility, as discussed in Section 2.2.

The close relation between Web usage mining and Web personalization is the main motivation for this survey. Considering its use for Web personalization, and being essentially a data mining process, Web usage mining consists of the basic data mining stages:

- *Data Collection*. During this stage, usage data from various sources are gathered and their content and structure is identified. For Web usage mining, data are collected from Web servers, from clients that connect to a server, or from intermediary sources such as proxy servers and packet sniffers. A number of techniques that have been employed at this stage, can be used to attain efficient collection of user data for personalization.
- *Data Preprocessing*. This is the stage where data are cleaned from noise, their inconsistencies are resolved, and they are integrated and consolidated, in order to be used as input to the next stage of Pattern Discovery. In Web usage mining,

this involves primarily data filtering, user identification and user session identification. The techniques that are used here can provide efficient data elaboration.

- *Pattern Discovery*. In this stage, knowledge is discovered by applying machine learning and statistical techniques, such as clustering, classification, association discovery, and sequential pattern discovery to the data. The patterns required for Web personalization, correspond to the behavior and interests of users. This is the stage where the learning methods are applied in order to automate the construction of user models.
- *Knowledge Post-Processing*. In this last stage, the extracted knowledge is evaluated and usually presented in a form that is understandable to humans, e.g. using reports, or visualization techniques. For Web personalization the extracted knowledge is incorporated in a Personalization module in order to facilitate the personalization functions.

Figure 1 summarizes graphically the above-described stages. Each stage presents various difficulties that are particular to Web usage mining. These problems have been addressed in recent work, which is presented in the following sections. It should be noted again here that Web usage mining is less mature than other application areas of data mining. As a result, some of the issues that have been studied in data mining research and are considered as separate stages of the data mining process, such as the problem definition and the evaluation of the extracted knowledge (Chapman et al., 2000), are still unexplored territory in Web usage mining.

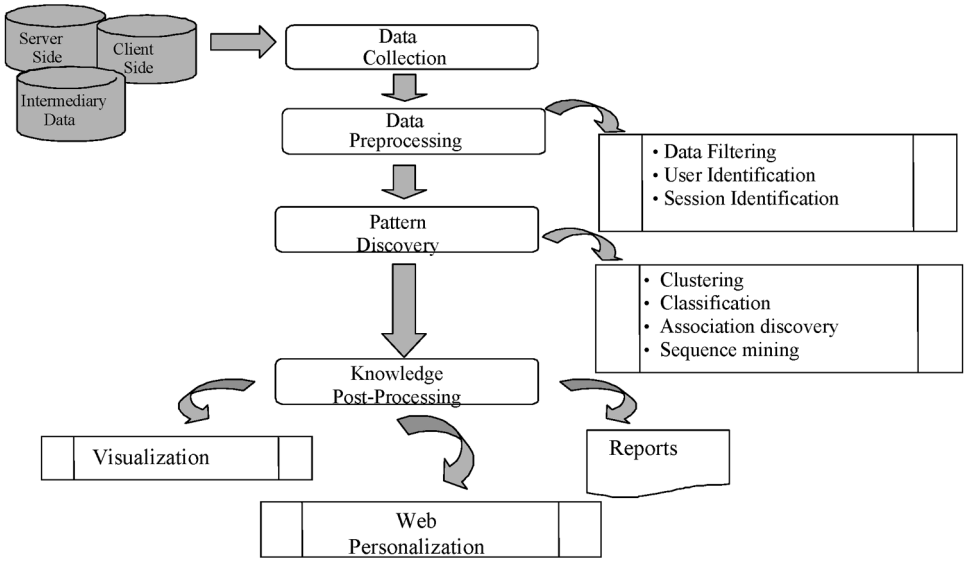

Figure 1 The web usage mining process

4. Data Collection

The first step in the Web usage mining process consists of gathering the relevant Web data, which will be analyzed to provide useful information about the users' behavior. There are two main sources of data for Web usage mining, corresponding to the two software systems interacting during a Web session: data on the Web server side and data on the client side. Additionally, when intermediaries are introduced in the client-server communication, they can also become sources for usage data, e.g. proxy servers and packet sniffers. Each of these sources is examined in more detail in the following subsections. At the end of the section, we are trying to associate the data collection methods with the requirements imposed by different classes of personalization functions.

4.1. SERVER SIDE DATA

4.1.1. *Server Log Files*

Server side data are collected at the Web server(s) of a site. They consist primarily of various types of logs generated by the Web server. These logs record the Web pages accessed by the visitors of the site. Most of the Web servers support as a default option the Common Log File Format, which includes information about the IP address of the client making the request, the hostname and user name, if available, the time stamp of the request, the file name that is requested, and the file size. The Extended Log Format (W3C), which is supported by Web servers such as Apache and Netscape, and the similar W3SVC format, supported by Microsoft Internet Information Server, include additional information such as the address of the referring URL to this page, i.e., the Web page that brought the visitor to the site, the name and version of the browser used by the visitor and the operating system of the host machine.

Web usage mining tools use Web server log files as the main data source for discovering usage patterns. However, log files cannot always be considered a reliable source of information about the usage of a site. The problem of data reliability becomes particularly serious for Web personalization, where it is important to identify individual users, in order to discover their interests. There are primarily two sources of data unreliability: *Web caching* and *IP address misinterpretation*.

The Web cache is a mechanism for reducing latency and traffic on the Web. A Web cache keeps track of Web pages that are requested and saves a copy of these pages for a certain period of time. Thus, if there is a request for the same Web page, the cached copy is used instead of making a new request to the Web server. Web caches can be configured either at the users' local browsers, or at intermediate proxy servers. The obstacle that is introduced is the same for both types of cache. If the requested Web page is cached, the client's request does not reach the corresponding Web server holding the page. As a result, the server is not aware of the action and the page access is not recorded into the log files. One solution that has been proposed is *cache-busting*, i.e., the use of special HTTP headers defined either in Web servers or Web pages, in

order to control the way that those pages are handled by caches. These headers are known as Cache-Control response headers and include directives to define which objects should be cached, how long they should be cached for, etc. However this approach works against the main motivation for using caches, i.e., the reduction of Web latency.

The second problem, IP misinterpretation in the log files, occurs for two main reasons. The first reason, also known as the 'AOL effect,' is the use of intermediate proxy servers, which assign the same IP to all users. As a result, all requests from various host machines that pass through the proxy server are recorded in the Web server log as requests from a single IP address. This can cause misinterpretation of the usage data. The same problem occurs when the same host is used by many users (e.g. members of a family). The opposite problem occurs when one user is assigned many different IP addresses, e.g. due to the dynamic IP allocation that is used for dial-up users by ISPs. A variety of heuristics have been employed in order to alleviate the problem of IP misinterpretation, which are discussed in Section 5.

Finally, information recorded at the Web servers' log files may pose a privacy threat to Internet users (Broder, 2000). Especially the referer field is considered private information and according to the Request for Comments (RFC) on *Hypertext Transfer Protocol — HTTP/1.1* 2616 (Fielding et al., 1999), the user should have the option to transmit or not the referer field through the Web browser. Unfortunately, most browsers do not comply currently with this recommendation.

4.1.2. *Cookies*

In addition to the use of log files, another technique that is often used in the collection of data is the dispensation and tracking of *cookies*. Cookies are short strings dispensed by the Web server and held by the client's browser for future use. They are mainly used to track browser visits and pages viewed. Through the use of cookies, the Web server can store its own information about the visitor in a cookie log at the client's machine. Usually this information is a unique ID that is created by the Web server, so the next time the user visits the site, this information can be sent back to the Web server, which in turn can use it to identify the user. Cookies can also store other kind of information such as pages visited, products purchased, etc., although the maximum size of a cookie cannot be larger than 4Kbytes, and thus it can hold only a small amount of such information.

The use of cookies is also not without difficulties. One problem is that many different cookies may be assigned to a single user, if the user connects from different machines, or multiple users may be using the same machine and hence the same cookies. In addition, the users may choose to disable the browser option for accepting cookies, due to privacy and security concerns. This is specified in HTTP State Management Mechanism which is an attempt of the Internet Engineering Task Force to set some cookie standards. This attempt is formalized in the Request for Comments (RFC) 2965 (Kristol and Montulli, 2000). Even when they accept cookies, the users can selectively delete some of them. Cookies are also limited in number. Only 20 cookies are allowed per

domain, and no more than 300 cookies are allowed in the client machine. If the number of cookies exceeds these values, the least recently used will be discarded.

4.1.3. *Explicit User Input*

Various user data supplied directly by the user, when accessing the site, can also be useful for personalization. User data can be collected through registration forms and can provide important personal and demographic information, as well as explicit user preferences. This is the case for some applications (e.g. online banking) where fairly detailed user data are available. However, this method increases the load on the user and it is generally considered a serious disincentive for visiting the site. Furthermore, we cannot always rely on user-supplied information, since it is often incomplete and inaccurate. Users tend to provide as little as possible personal information for reasons concerning their privacy.

Another type of user data collected at the server side is the query data to a Web server that are generated on e-commerce sites, portals and digital libraries (Büchner and Mulvenna, 1999). These data are provided to a local search engine, when the user is trying to identify specific information within the site. However, free-text data are particularly rich and there is no formal standard for describing them. As a result, it is difficult to make use of these data.

4.1.4. *External Data*

Finally, user data providing demographic information (Mulvenna and Büchner, 1997) can be acquired from third party suppliers who maintain user databases for various reasons. However, privacy concerns have led to the introduction of legal obstacles in the distribution of these data. For instance, the European Union's Directive (EU Directive) on Personal Data, which took effect on October 23, 1998, restricts the transfer of personal data to a non-EU country.

4.2. CLIENT SIDE DATA

Client side data are collected from the host that is accessing the Web site. One of the most common techniques for acquiring client side data is to dispatch a remote agent, implemented in Java or JavaScript (Shahabi et al., 1997, 2001). These agents are embedded in Web pages, for example as Java applets, and are used to collect information directly from the client, such as the time that the user is accessing and leaving the Web site, a list of sites visited before and after the current site, i.e., the user's navigation history, etc.

Client side data are more reliable than server side data, since they overcome caching and IP misinterpretation problems. However, the use of client side data acquisition methods is also problematic. One problem is that the various agents collecting information affect the client's system performance, introducing additional overhead when a user tries to access a Web site. Furthermore, these methods require the cooperation of users,

who may not allow an agent running on their side. Web users often activate security mechanisms for restricting the operation of Java and JavaScript programs from their browsers, in order to avoid running hazardous software. The result is the ineffective operation of agents collecting client side data.

An older technique used to collect data at the client side was through a modified version of the Mosaic browser (Cunha et al., 1995; Tauscher and Greenberg, 1997). This modification allowed the browser to record the Web pages that were visited by a user and send that, together with other information, such as time of access, and time of response, back to the server. Browser modification is not a trivial task for modern browsers, even when their source code is available, like Netscape Navigator. Furthermore, modified browsers that record the behavior of users are considered a threat to the user's privacy and thus it is difficult for users to accept their use.

4.3. INTERMEDIARY DATA

4.3.1. *Proxy Servers*

A proxy server is a software system that is usually employed by an enterprise connected to the Internet and acts as an intermediary between an internal host and the Internet so that the enterprise can ensure security, administrative control and caching services. Despite the problems that they cause, which were mentioned above, proxy servers can also be a valuable source of usage data.

Proxy servers also use access logs, with similar format to the logs of Web servers, in order to record Web page requests and responses from the server. The advantage of using these logs is that they allow the collection of information about users operating behind the proxy server, since they record requests from multiple hosts to multiple Web servers (Srivastava et al., 2000). However, the problems of caching and IP address misinterpretation that were mentioned in Section 4.1, are also applicable to proxy server data.

4.3.2. *Packet Sniffers*

A packet sniffer is a piece of software, or sometimes even a hardware device, that monitors network traffic, i.e., TCP/IP packets directed to a Web server, and extracts data from them. One advantage of packet sniffing over analyzing raw log files is that the data can be collected and analyzed in real time. Another important advantage is the collection of network level information that is not present in the log files. This information includes detailed timestamps of the request that has taken place, like the issue time of the request, and the response time, as well as whether or not the user has cancelled the request, by pressing the Web browser's stop button. The complete Web page that has been requested can also been included in the sniffed data (Feldmann, 1998).

On the other hand, the use of packet sniffers also has important disadvantages compared to log files. Since the data are collected in real time and are not logged, they may be lost forever if something goes wrong either with the packet sniffer or with the data transmission. For example, the connection may be lost, the packet sniffer

may lose packets, or the packets may arrive in a disordered state, and hence it will not be possible to process them correctly in real time. In addition, especially for e-commerce sites, TCP/IP packets are increasingly transmitted in encrypted format, reducing the ability of packet sniffers to extract useful information. Finally, since packet sniffing tools operate directly on the data transmitted over the Internet, they are considered a serious threat to the users' privacy, especially when used by intermediaries like Internet Service Providers.

4.4. DATA COLLECTION FOR PERSONALIZATION

Web usage data collection is the first step towards realizing the Web personalization functions. Almost all of the methods discussed above can be used in various personalization functions. For example, registration data can help in realizing all functions, since these data, if they are accurate enough, provide a precise view of the user profile in terms of preferences, interests and knowledge. Here, we attempt to identify interesting associations between the data collection methods and the corresponding requirements of the personalization functions. Following this approach we present each class of personalization functions, in relation to the corresponding data collection methods.

4.4.1. *Memorization*

Different data collection methods are needed for different memorization functions. User salutation requires explicit user input by means of registration data in order to obtain the user's name. The user name can be stored either in a local database or in a cookie file. Additional registration data are required for the implementation of personalized access policies. The validation of these data is vital, in order to avoid granting unauthorized access to various items. On the other hand, a bookmarking scheme can be simply realized by collecting the Web pages that a user has visited. Thus, log files and client agents can support this type of personalization function.

4.4.2. *Guidance*

Guidance functions typically require information about what the user is seeking in a Web site, as well as information about the user's knowledge level. This information can be assembled using server-side and client-side data, as well as intermediary sources like packet sniffers. However, the preciseness of client-side data makes them particularly suitable for this class of personalization functions.

4.4.3. *Customization*

This class of functions requires mainly information about the users' interests and preferences. This information is acquired by recording the users' browsing history. Both server-side and client-side data are appropriate for this purpose, although server log files are the main source of information that is typically being used. Simple log files

are augmented by purchase data from corresponding databases of the company, for the implementation of a personalized pricing scheme.

4.4.4. *Task Performance Support*

Performing a certain action on behalf of a user requires the collection of data that reveal the intention of the user to perform a certain task. This can be achieved by a combination of data collection methods. Hence, data collected by server logs and client agents can be used to describe the browsing behavior of a user, and they could be employed to deduce the user's intentions. However, a more accurate view of those intentions is often required which can only be obtained either by registration or query data.

5. Data Preprocessing

Web data collected in the first stage of data mining are usually diverse and voluminous. These data must be assembled into a consistent, integrated and comprehensive view, in order to be used for pattern discovery. As in most applications of data mining, data preprocessing involves removing and filtering redundant and irrelevant data, predicting and filling in missing values, removing noise, transforming and encoding data, as well as resolving any inconsistencies. The task of data transformation and encoding is particularly important for the success of data mining. In Web usage mining, this stage includes the identification of users and user sessions, which are to be used as the basic building blocks for pattern discovery. The accurate identification of users and user sessions is particularly important for Web personalization, because the models of individual users are based on user behavior encoded in user sessions and associated correctly with the corresponding users.

The data preprocessing step is to some extent domain-dependent, e.g. the content and the structure of a Web site affect the decision about which data are relevant. It is also strongly dependent on the type and the quality of the data and thus, it constitutes a hard engineering task in the Web usage mining process. In addition, there is a trade off between insufficient preprocessing, which will make the pattern discovery task more difficult and affect its results, and excessive preprocessing that may result in removing data with useful, but implicit, knowledge (Mulvenna et al., 1997).

5.1. DATA FILTERING

The first step in data preprocessing is to clean the raw Web data. During this step the available data are examined and irrelevant or redundant items are removed from the dataset. This problem mainly concerns log data collected by Web servers and proxies, which can be particularly noisy, as they record all user interactions. Due to these reasons, we concentrate here on the treatment of Web log data. Data generated by client-side agents are clean as they are explicitly collected by the system, without the intervention of the user. On the other hand, user supplied data (e.g. registration data)

need to be verified, corrected and normalized, in order to assist in the discovery of useful patterns.

Significant redundancy in log files is caused by the specification of the HTTP protocol, which requires a separate access request to the server for each file, image, video, etc. that is embedded in a Web page. Usually entries in log files that refer to image, sound, video files, CGI scripts and image map files are considered redundant. These files are downloaded without a user explicitly requesting them and thus, they are not part of the user's actual browsing activity. Therefore, these entries are usually removed from the log files (Cooley et al., 1999a). However, as mentioned above, the data preprocessing task is domain-dependent and removing those files may, sometimes, cause the loss of valuable information. This is the case, for example, when a site consists mainly of multimedia content.

Moreover, log entries corresponding to requests that have not been satisfied, e.g. requests that received HTTP error responses, are filtered out of the log file, though they may correspond to information that is needed by users. In addition, records created by spiders or crawlers, i.e., programs that download complete Web sites in order to update the index of a search engine are also removed, since they are not considered usage data. Crawlers and spiders can often be recognized through the User Agent field of the server logs, since most crawlers will identify themselves using this field. Another technique is to look at the pattern of traffic of a particular visitor. If the visitor is retrieving pages in an exhaustive pattern of following every hyperlink on every page within a site, then it is a crawler. Tan and Kumar (2002) propose a method for identifying spider sessions based on a variety of features that are extracted from access logs such as percentage of media files requested, percentage of requests made by HTTP methods, as well as features that are showing the breadth-first searching behavior of the spider.

Data referring to other Web sites, over which the site administrator has no control, are also considered irrelevant and are usually removed.

5.2. USER IDENTIFICATION

The identification of individual users who access a Web site is one of the most important issues for the success of a personalized Web site. Many of the existing commercial personalization systems require the users to identify themselves, by logging in before using the system. However, this process introduces a burden to the user that is not acceptable for many Web sites, e.g. the average e-commerce site. For this reason, a number of studies have proposed various approaches to automate the process of user identification. The most important of these are presented here.

The simplest approach is to assign a different user to each different IP identified in the log file. Several of the Web usage mining tools adopt this approach, despite its high degree of inaccuracy, due to the use of proxy servers. Cookies are also useful for identifying the visitors of a Web site (Kamdar and Joshi, 2000) by storing an ID, which is generated by the Web server for each user visiting the Web site, but since they are considered

a security and privacy threat, users may disable or delete them. Furthermore, if a user connects to the Internet using different machines, then that user cannot be identified correctly.

Due to these problems, other heuristic techniques have been proposed. One of these techniques is the use of special Internet services, such as the *inetd* and *fingerd* services (Pitkow, 1997), which provide the user name and other information about the client accessing the Web server. One problem with this method is that these services may also be disabled for security reasons. In addition, if the users operate behind proxy servers it is impossible to identify them with the use of the *inetd* and *fingerd* services, since the proxy server corresponds to one host IP address for many users.

Two more heuristic methods to overcome user identification problems are presented in (Cooley et al., 1999a). The first method performs an analysis of the Web server log file in Extended Log Format, looking for different Web browsers, or different operating systems, in terms of type and version, even when the IP remains the same. This information suggests the existence of different users. The second method presented in the same work combines the topology of a site together with access log referrer entries. If a request for a page originates from the same IP address as other already visited pages, and no direct hyperlink exists between these pages, then it is suspected that a new user is accessing the Web site. These two methods are also not without problems. Apart from their cost in terms of computational effort, there are also many cases where the heuristics fail, e.g. when a user opens different browsers or more than one browser window visiting different pages from the same site that are not directly hyperlinked. A further problem with the second heuristic is that it does not help in identifying the user fully, i.e., relating different visits of the same user to the site, at different times.

Schwarzkopf (2001) employs a different technique for user identification. A unique ID, generated by the Web server for each user, is included in the URL of the Web pages delivered to the users. Instead of storing this ID in a cookie file, the user is asked to create a bookmark for one of these pages, which includes this ID as part of its URL. The ID is used to identify the user, whenever the user returns to the site, via this bookmarked page, and is stored in the log file replacing the IP address of the user. This is a very simple technique, avoiding several problems that cookies have, such as the blocking of cookies by the user. However, this technique is also problematic. The main problem is that the identification process is only semi-automatic as the user must bookmark a page and use that page to access the site, otherwise the user's ID will not be used. Furthermore, the situation of a user accessing the Web site from different machines, also affects this technique.

5.3. USER SESSION IDENTIFICATION

The identification of user sessions has also received significant attention in Web usage mining projects, as sessions encode the navigational behavior of the users and they are thus important for pattern discovery. A user session is a delimited set of pages visited by the same user within the duration of one particular visit to a Web site.

Various heuristic methods have been used to identify user sessions. Spiliopoulou (1999) divides these methods into time-based and context-based. Examples of time-based heuristics are the use of an upper limit to the time spent on a page, or an upper limit to the total duration of a session. Access to specific types of page, or the completion of a conceptual unit of work can be considered context-based methods.

Time-based methods have been used in most of the literature, (e.g. Catledge and Pitkow, 1995; Borges and Levene, 1999; Nasraoui et al., 1999; Wu et al., 1998; Paliouras et al., 2000a and Pei et al., 2000). According to these approaches, a set of pages visited by a specific user is considered as a single user session if the pages are requested at a time interval not larger than a specified time period. This period, also known as *page viewing time* (Shahabi et al., 1997), varies from 25.5 min, which has been proposed initially for log analysis by Catledge and Pitkow (1995), to 24 h (Yan et al., 1996), whilst 30 min is used as the default (Cooley et al., 1999a). However, this heuristic is not very reliable since the exact actions of a user are not known and may vary considerably, e.g. a user may be reading the same page for a long time, or may have left the office for a while and return to reading the page, etc. In addition, the value of this timeout depends largely on the content of the site being examined.

An important problem with the common time-based methods is the use of the cache, which may lead the system to the conclusion that a session has ended, while the user is still retrieving pages from the cache. This problem can be alleviated by the introduction of special HTTP headers, as discussed in Section 4.1. Shahabi et al. (2001) propose an alternative method of measuring the viewing time for a page, using Java agents, that send back to the server the client's system time each time a new Web page is loaded or unloaded at the client's browser. However, external factors, such as network traffic and the type of browser, used by the visitor, introduce important obstacles to this method. Furthermore, the use of Java agents or JavaScript may be disabled by the user.

Yan et al. (1996) identify user sessions by modifying the NCSA httpd server, in order to include session identifiers in the Web pages. The first time a Web page is requested from a specific host-IP address, an identifier is embedded in this page corresponding to the start of a user session. This identifier is kept in subsequent requests coming from the same host, and a timeout mechanism is used to separate different session identifiers. However, caching mechanisms, affect the accuracy of this approach, since pages that are retrieved from caches are not included in the sessions.

User sessions are sometimes processed further to provide more useful entities. One such example is the concept of a *transaction* (Mobasher et al., 1996), a term derived from data mining and used mainly in market basket analysis. A transaction is a subset of related pages that occur within a user session. In order to identify transactions, Cooley et al. (1997a) made the assumption that transactions have a strong relation to the browsing behavior of a specific user, and can thus be identified using contextual information. Based on this assumption, the pages of a site are divided into three types: *navigational or auxiliary pages*, containing primarily hyperlinks to other Web pages and used only for browsing, *content pages*, containing the actual information of interest to the user and *hybrid pages*, combining both types of information. Although, there are pages

on the Web, which belong clearly in one of the three categories, such as index or home pages, this context-based classification is not very strict and depends on the users' perspective. A navigational page for one user may be a content page for another.

In (Cooley et al., 1999a) transactions are classified to *content-only* transactions, corresponding to the content pages that a user has chosen to view within a Web site and *auxiliary-content* transactions, corresponding to the paths to a content page. Two methods for identifying transactions of these two types are described: the *reference length*, and the *maximal forward reference*. The reference length method examines the time period that a user spends visiting a Web page. If this period is greater than a certain threshold, the page is assumed to contain useful information, i.e. it is a content page, and is added to a content-only transaction. Otherwise the page is considered navigational, and is added to an auxiliary-content transaction. Auxiliary-content transactions are 'closed' with the questionable assumption that the last page a user visits is always a content page. Any interruption to the browsing sequence caused by external factors may erroneously lead to the identification of a content page. Another problem with this method is that the definition of the appropriate threshold is strongly dependent on the content of the site.

The maximal forward reference method has been proposed by Chen et al. (1996). According to this approach, a transaction is defined as the set of pages visited by a user from the first page, as recorded in the log file, up to the page where the first backward reference has occurred. A backward reference is a page that is already included in the recorded transaction. The next transaction starts with the next forward reference, i.e. a new page that is not included in the recorded transaction. Using the classification of pages into content and navigational ones, maximal forward references are content pages, while the pages leading to the maximal forward references are navigational (auxiliary) pages. This method has an advantage over the reference length method since it is independent of the site content. However, it also suffers from an important problem, which is the caching of Web pages that prevents backward references to be recorded in log files. Cache-busting can solve this problem, with the cost explained above.

Finally, Ardissono and Torasso (2000), also employ contextual information to identify a sequence of actions, in a user's browsing sequence, that are related to a certain item, defined as the *Current Focus of Attention*. The actions that are performed without changing the 'browsing' focus, constitute the *local history*, which is used to analyze the user's behavior.

5.4. DATA PREPROCESSING FOR PERSONALIZATION

Removing noise and irrelevant data is the first step towards better Web personalization since information that is not related with the browsing behavior of the user can misguide the pattern discovery process. Furthermore, user identification is one of the most critical parameters of a Web personalization system. As a result, both data filtering and user identification are required in some form by most of the personalization functions.

5.4.1. *Memorization*

User identification is the only data preprocessing step required for memorization functions. In particular user salutation and the personalization of access rights require an accurate method of user identification such as user registration. The heuristic methods discussed in the previous subsections are not sufficient for this class of functions.

5.4.2. *Guidance*

The employment of user session identification methods is the first step towards better user guidance. User tutoring requires context-based methods, that reveal in a more detailed manner the knowledge of the user, whilst recommendation of hyperlinks can employ either context-based or time-based session identification methods.

5.4.3. *Customization*

The identification of user sessions is essential also for the realization of this class of personalization functions. Some functions, such as content customization and product differentiation, require information about the relation between Web pages that a user visits. For these functions, context-based methods seem more appropriate.

5.4.4. *Task Performance Support*

User session identification is also important in the identification of the user's intention to perform a certain task. In particular, context-based methods are suitable for negotiation functionality, since they can be employed to derive the user's interests in a certain topic or item that will be negotiated.

6. Pattern Discovery

In this stage, machine learning and statistical methods are used to extract patterns of usage from the preprocessed Web data. A variety of machine learning methods have been used for pattern discovery in Web usage mining. These methods represent the four approaches that most often appear in the data mining literature: clustering, classification, association discovery and sequential pattern discovery. Similar to most of the work in data mining, classification methods were the first to be applied to Web usage mining tasks. However, the difficulty of labeling large quantities of data for supervised learning has led to the adoption of unsupervised methods, especially clustering. Thus, the majority of the methods presented here are clustering methods.

Unlike the data preprocessing tools, the methods used for pattern discovery are domain-independent, meaning that they can be applied to many different domains, e.g. any Web site, without concern about the content of the Web site. Furthermore, most of the pattern discovery methods that have been used are

general-purpose, i.e., they are the same that have been applied to other data mining tasks. However, the particularities of Web data have also led to the development of some new methods.

This section provides an overview of the pattern discovery methods that have been used in Web usage mining so far. The focus is on the application task, i.e., the types of pattern that are sought, omitting the details of the learning algorithms, which can be found in the original articles. The presentation of clustering approaches occupies a large part of this section, proportional to the work found in the literature.

6.1. CLUSTERING

The large majority of methods that have been used for pattern discovery from Web data are clustering methods. Clustering aims to divide a data set into groups that are very different from each other and whose members are very similar to each other. Han and Kamber (2001), propose a general taxonomy of clustering methods that comprises the following categories:

- *Partioning methods*, that create k groups of a given data set, where each group represents a cluster.
- *Hierarchical methods*, that decompose a given data set creating a hierarchical structure of clusters.
- *Model-based methods*, that find the best fit between a given data set and a mathematical model.

Clustering has been used for grouping users with common browsing behavior, as well as grouping Web pages with similar content (Srivastava et al., 2000). One important constraint imposed by Web usage mining to the choice of clustering method is the fact that clusters should be allowed to overlap (Paliouras et al., 2000b). This is important for Web personalization since a user or a Web page may belong to more than one group.

A partioning method, was one of the earliest clustering methods to be used in Web usage mining by Yan et al. (1996). In this work, the Leader algorithm (Hartigan, 1975), is used to cluster user sessions. Leader is an incremental algorithm that produces high quality clusters. Each user session is represented by an n-dimensional feature vector, where n is the number of Web pages in the session. The value of each feature is a weight, measuring the *degree of interest* of the user in the particular Web page. The calculation of this figure is based on a number of parameters, such as the number of times the page has been accessed and the amount of time the user spent on the page. Based on these vectors, clusters of similar sessions are produced and characterized by the Web pages with the highest associated weights. The characterized sessions are the patterns discovered by the algorithm. One problem with this approach is the calculation of the feature weights. The choice of the right parameter mix for the calculation of these weights is not straightforward and depends on the modeling abilities of a human expert. Furthermore, the use of the Leader algorithm

is problematic, as the construction of the clusters depends on the presentation order of the input vectors. For instance, if three training vectors (a, b, c), are submitted in that order to the algorithm, a different set of clusters may result than if the vectors are submitted in a different order, e.g. (b, a, c).

A partitioning graph theoretic approach is presented by Perkowitz and Etzioni (1998, 2000), who have developed a system that helps in making Web sites adaptive, i.e., automatically improving their organization and presentation by mining usage logs. The core element of this system is a new clustering method, called *cluster mining*, which is implemented in the PageGather algorithm. PageGather receives user sessions as input, represented as sets of pages that have been visited. Using these data, the algorithm creates a graph, assigning pages to nodes. An edge is added between two nodes if the corresponding pages co-occur in more than a certain number of sessions. Clusters are defined either in terms of cliques, or connected components. Clusters defined as cliques prove to be more coherent, while connected component clusters are larger, but faster to compute and easier to find. A new index page is created from each cluster with hyperlinks to all the pages in the cluster. The main advantage of PageGather is that it creates overlapping clusters. Furthermore, in contrast to the other clustering methods, the clusters generated by this method group together characteristic features of the users directly. Thus, each cluster is a behavioral pattern, associating pages in a Web site. However, being a graph based algorithm, it is rather computationally expensive, especially in the case where cliques are computed.

Another partitioning clustering method is employed by Cadez et al. (2000) in the WebCANVAS tool, which visualizes user navigation paths in each cluster. In this system, user sessions are represented using categories of general topics for Web pages. A number of predefined categories are used as a bias, and URLs from the Web server log files are assigned to them, constructing the user sessions. The Expectation-Maximization (EM) algorithm, (Dempster et al., 1977) based on mixtures of Markov chains is used for clustering user sessions. Each Markov chain represents the behavior of a particular subgroup. EM is a memory efficient and easy to implement algorithm, with a profound probabilistic background. However, there are cases where it has a very slow linear convergence and may therefore become computationally expensive, although in the results in Cadez et al. (2000), it is shown empirically that the algorithm scales linearly in all aspects of the problem.

The EM algorithm is also employed by Anderson et al. (2001a) in two clustering scenarios, for the construction of predictive Web usage models. In the first scenario, user navigation paths are considered members of one or more clusters, and the EM algorithm is used to calculate the model parameters for each cluster. The probability of visiting a certain page is estimated by calculating its conditional probability for each cluster. The resulting mixture model is named Naïve Bayes mixture model since it is based on the assumption that pages in a navigation path are independent given the cluster. The second scenario uses a similar approach to (Cadez et al., 2000). Markov chains that represent the navigation paths of users are clustered using the EM algorithm, in order to predict subsequent pages.

An extension of partitioning clustering methods is fuzzy clustering that allows ambiguity in the data, by 'distributing' each object from the data set over the various clusters. Such a fuzzy clustering method is proposed in (Joshi and Joshi, 2000) for grouping user sessions, where each session includes URLs that represent a certain traversal path. The Web site topology is used as a bias in computing the similarity between sessions. The site is modeled using a tree, where each node corresponds to a URL in the site, while each edge represents a hierarchical relation between URLs. The calculation of the similarity between sessions is based on the relative position in the site tree of the URLs included in the sessions. Clustering is implemented using two newly devised algorithms: Fuzzy c-medoids and Fuzzy c-trimmed-medoids, which are variants of the Fuzzy *c* clustering method (Bezdek, 1981). Fuzzy clustering is also employed by Nasraoui et al. (1999), who use the Relational Fuzzy Clustering–Maximal Density Estimator (RFC-MDE) algorithm to cluster user sessions identified in the Web server logs. The employment of fuzzy clustering methods allows the creation of overlapping clusters, since they introduce a degree of item-membership in each cluster. However, this item-membership is specified by a membership function, the design of which is a non-trivial issue.

A hierarchical clustering approach is employed by Fu et al. (1999) who use the BIRCH algorithm (Zhang et al., 1996) for clustering user sessions. Data from the Web server log are converted into sessions represented by vectors, where each vector contains the ID of each Web page accessed, together with the time spent on that page. In order to improve the efficiency of the clustering algorithm and to discover more general patterns, sessions are generalized using the page hierarchy of the Web site as a bias. Each Web page in a session is replaced by a more general Web page according to the page hierarchy, using the attribute-oriented induction method (Han et al., 1992). The resulting *generalized sessions* are used as input to the BIRCH algorithm. BIRCH is a very efficient and incremental algorithm for processing large volumes of data. It can produce qualitative clusters by scanning the data only once, and improve them with additional scans. However, similar to the Leader algorithm, BIRCH also depends on the presentation order of the input data. Furthermore, due to the specification of the algorithm, it does not always create 'natural' clusters since each cluster, is allowed a maximum size of members (Halkidi et al., 2001).

A variety of model-based clustering methods have been used in (Paliouras et al., 2000b). A probabilistic method, Autoclass[⋆] (Hanson et al., 1991), a neural network, Self-Organizing Maps (Kohonen, 1997), a conceptual clustering method, COBWEB (Fisher, 1987), and its non-hierarchical variant, ITERATE (Biswas et al., 1998), are exploited in order to construct user community models, i.e., models for groups of users with similar usage patterns. Community models are derived as characterizations of the clusters and correspond to the interests of user communities.

[⋆] In the case of Autoclass, a mixture of single multinomial likelihood models was used, assuming conditional independence of the attributes given the class.

Autoclass has the advantage of a sound mathematical foundation that provides a disciplined way of eliminating some of the complexity associated with data. However, it is computationally expensive and requires prior assumptions about the data. Self-Organizing Maps is an artificial neural network model that is well suited for mapping high-dimensional data into a 2-dimensional representation space, where clusters can be identified. However, it requires preprocessing and normalization of the data, and the prior specification of the number of clusters. COBWEB is an incremental algorithm that provides a characteristic description of each cluster. The algorithm, though, suffers from scalability problems and is dependent on the order of the observed objects. ITERATE solves the order-dependence problem of COBWEB, but is not incremental and scalable.

Despite the variety of clustering methods that have been used for Web usage mining, hardly any work has been done on the comparison of their performance. The reason for this is the inherent difficulty in comparing clustering results, due to the lack of objective criteria independent of the specific application. In other words, there is no information about the correct clusters to be identified and any solution is valid, until it gets rejected subjectively by an expert in the field.

A domain-independent approach to the evaluation task is the use of statistical criteria for the quality of the resulting clusters (e.g. Theodoridis and Koutroubas, 1999; Halkidi et al., 2001). These criteria are independent of the specific application and they provide an indication of the performance of the clustering methods. However, they resemble the statistical criteria used by the clustering methods themselves and in most cases they could even be used in this manner, i.e., they could be incorporated into a new clustering method. This fact raises concerns about the impartiality of the criteria.

A simpler benchmarking method for evaluating cluster validity is proposed by Estivill-Castro (2002). The method involves the employment of data sets with known structure against which, the result of a clustering process can be evaluated. It is also suggested that both the data and the evaluation methodology used are distributed, in order to allow other researchers to apply their algorithms on the same data sets and evaluate them using the same methodology.

Paliouras et al. (2000b), also propose a cluster evaluation approach, based on criteria for the quality of the community models. The models are evaluated on the basis of two measures: *distinctiveness*, i.e., how different the models are from each other, and *coverage* of the domain, e.g. how many of the Web site pages are used in the models. These criteria are independent of the clustering method and make no assumption of non-overlapping clusters. They are thus appropriate for Web usage mining, although one could argue that they also have a bias for this application domain. From all methods examined, the Autoclass method seems to outperform the other methods in terms of these two criteria. The good performance of Autoclass can be justified by its sound probabilistic foundation, which leads the method to the approximation of the Bayesian optimal choice of clusters. However, one practical drawback of this method is its high computational cost, which justifies its limited use in Web usage mining applications.

Yet another approach to the evaluation of clustering methods is to apply them to a task where the desired outcome is predetermined. One such effort in the context of collaborative filtering is reported in (Breese et al., 1998), where the task was to predict the next choice of a user, e.g. the next page to visit in a Web site. Clearly, this approach is problem-specific and difficult to apply to different tasks. However, it provides useful insight into the performance of the clustering methods. Interestingly, the methods that are reported to be doing best in this work are again based on probabilistic models, like the Autoclass method mentioned above. A similar approach is considered by Cadez et al. (2000), where the predictive power of the algorithms is evaluated as the number of clusters change.

Table 1 presents a summarized overview of the clustering algorithms discussed above. The table presents the Web Usage Mining applications that employed the particular algorithms, the clustering approach that they pursue, as well as an indication of the strengths and weaknesses of each approach in the context of Web personalization.

6.2. CLASSIFICATION

In contrast to clustering, the goal of classification is to identify the distinguishing characteristics of predefined classes, based on a set of instances, e.g. users, of each class. This information can be used both for understanding the existing data and for predicting how new instances will behave. Classification is a supervised learning process, because learning is driven by the assignment of instances to the classes in the training data. In the context of Web usage mining, classification has been used for modeling the behavior of users and the characteristics of Web pages, based on preclassified sets of users and/or Web pages. Decision tree induction, neural networks and Bayesian classifiers are the methods that have mainly been used so far.

Decision rule induction is one of the classification approaches that were first applied to Web usage mining. Methods of this type examine each class in turn and try to construct a set of rules, covering the instances of the class, while excluding others that do not belong to it. A first attempt that followed this approach was that of Ngu and Wu (1997) in the SiteHelper system. The HCV (Wu, 1993) induction algorithm was used, receiving as input either the Web pages extracted from the Web server log files, or a set of keywords provided by the users, which were considered positive examples by the HCV algorithm. The result of the inductive process was a set of rules representing the user's interests. The HCV algorithm is good at learning very compact rules for noise free problem domains without continuous attributes. Unfortunately, its performance deteriorates when it is confronted with noise and/or continuous attribute domains.

A similar approach was followed in (Jörding, 1999), who employed the CDL4 algorithm (Shen, 1996), to create a decision rule list, in order to determine whether the visitor is interested in a certain subject. CDL4 is an incremental algorithm for learning the description of a finite set of classes from a sequence of training instances. A set

Table 1 Summary of clustering algorithms for Web usage mining

Algorithm	Application	Clustering method	Pros	Cons
Leader (Hartigan, 1975)	Clustering user sessions, (Yan et al., 1996)	Partitioning	• Incremental • Qualitative clusters	• Order dependent
PageGather (Perkowitz and Etzioni, 1998)	Index Page Synthesis, (Perkowitz and Etzioni, 2000)	Partitioning	• Overlapping clusters • Each cluster is a direct behavioral pattern	• Computationally expensive
EM (Dempster et al., 1977)	Clustering user sessions represented by Markov chains, (Cadez et al., 2000), (Anderson et al., 2001a)	Partitioning	• Memory efficient • Sound mathematical background	• Slow linear convergence • Computationally expensive
Fuzzy *c* clustering (Bezdek, 1981) Nasraoui et al. (1999)	Clustering user sessions, (Joshi and Joshi, 2000), (Nasraoui et al., 1999)	Partitioning Fuzzy Clustering	• Overlapping clusters	• Design of membership function
BIRCH (Zhang et al., 1996)	Clustering user sessions, (Fu et al., 1999)	Hierarchical	• Incremental • Efficient for high-dimensionality data	• Order dependent • Creation of 'unnatural' clusters
Autoclass (Hanson et al., 1991)	Clustering user sessions, (Paliouras et al., 2000*b*)	Model-based	• Strong Mathematical foundation • High-quality clusters	• Computationally expensive • Requires prior assumptions
Self-Organizing Maps (Kohonen, 1997)	Clustering user sessions, (Paliouras et al., 2000*b*)	Model-based	• Visualization of high-dimensional data	• Preprocessing and normalization of the data • Prior specification of the number of clusters
COBWEB (Fisher, 1987)	Clustering user sessions, (Paliouras et al., 2000*b*)	Model-based	• Incremental • Cluster characterization	• Order dependent
ITERATE (Biswas et al., 1998)	Clustering user sessions, (Paliouras et al., 2000*b*)	Model-based	• Order independent	• Not incremental • Not scalable

of heuristic rules is used to construct initially the training data for the algorithm. However, the rule induction process of CDL4 may result in very complex decision rules which are hard to interpret.

Decision tree induction and naive Bayesian classification have also been employed at this stage of Web usage mining. Decision tree induction usually involves the recursive partitioning of the training set, resulting in a tree, each branch of which from the root to a leaf is a conjunction of membership tests for a particular class. On the other hand, naive Bayesian classification (Duda and Hart, 1973) is based on the calculation of the conditional probabilities of each descriptive attribute, given each one of the classes. These probabilities are used within a Bayesian decision theoretic framework to calculate the probability of each new instance to belong in a particular class. Chan (1999) examines a variety of classification algorithms that follow these approaches. In this work, pages that a user has visited, called *interest pages*, are considered positive examples for the induction of Page Interest Estimators (PIE). PIEs can be decision trees constructed by algorithms, such as C4.5 (Quinlan, 1993), and CART (Breiman et al., 1984), decision rules constructed by the RIPPER (Cohen, 1995) algorithm, or even naive Bayesian classifiers. PIEs are used to predict if a page is of interest to a particular user. The various algorithms have been evaluated using words and phrases extracted from the Web pages as descriptive attributes. C4.5 achieved the best overall performance, using only single-word features, while the classification performance of CART, RIPPER and the naive Bayesian classifier improved with the use of phrase features.

Decision tree learning algorithms are fast and produce intuitive results, but suffer from scalability problems, especially on high-dimensional data. On the other hand, decision rule algorithms have stronger explanatory capabilities and often better performance than decision trees but are even more computationally expensive. Bayesian classifiers have exhibited scalability and speed but they make prior assumptions about the probability distributions.

A different classification approach is based on the use of rough set theory. Rough set theory aims at the detection of equivalence classes within the data, i.e., sets of identical instances given the attributes describing the data. Since this is rare in real-world data, rough sets are used to approximate such equivalence classes. Thus, a rough set for a class A is approximated by a lower approximation set of instances that certainly belong to A and an upper approximation set of instances that cannot be described as not belonging to class A. Although rough sets can be rather effective for dealing with imprecise and noisy data, they are very computationally expensive.

Rough set theory is exploited by Maheswari et al. (2001), to describe the user's navigational behavior. In this work, Web sessions identified in server log files are classified into positive and negative examples of a certain concept, such as the purchase of a certain product. A 'positive' weighted directed graph is created whose nodes correspond to Web pages, and the edges correspond to transitions between them. The weights represent the percentage of positive sessions containing that particular transition in the whole set of positive sessions. In a similar way, a 'negative' weighted directed

graph is created. New unclassified user sessions are also represented as weighted directed graphs, where the weight of an edge is a function of the weight of this edge in the positive and negative graphs of known examples. A set of attributes are then used to determine the degree of 'positiveness' or 'negativeness' of a particular link in a session based on the values of the corresponding weights of the link in the two graphs. These attributes define the equivalence classes, and each user session is assigned to one of the classes. Hence, the classes are used to define the positive and negative 'regions' in the instance space, based on the number of positive or negative sessions they contain. Thus, the link of a new unclassified session is assigned to an equivalence class by examining the values of the attributes in the session's graph, and subsequently classified as positive or negative if the respective class belongs to the positive or negative region.

The use of classification in Web usage mining has been far more limited than that of clustering. This is due to the requirement for preclassified data, which is not always possible with Web data, where items, i.e., pages or sessions, cannot always be assigned to categories apriori. Hence, unsupervised techniques, such as clustering, seem more applicable to the majority of problems in the area of Web usage mining, especially those related to Web personalization, where the restriction to predefined user types, required for classification, introduces a serious bias to the discovery process. On the other hand, classification methods can be very useful for the construction of descriptive models for individual users.

Table 2 gives an overview of the classification algorithms that have been employed by Web Usage Mining applications. The table, presents also the classification method followed by each approach, together with the main advantages and disadvantages of each method in the context of Web personalization.

6.3. DISCOVERY OF ASSOCIATIONS

Associations represent relations between items in a database that co-occur in an event, such as a purchase transaction, or a user session. An association is often represented by an association rule $A \Rightarrow B$, which implies a dependence relation between two sets of items, A and B. The union of A and B is called an *itemset*. Association rules are used to estimate the probability of B occurring, given A. The selection of an association rule is based on two pieces of information: *support*, i.e., the frequency with which the corresponding itemset ($A \cup B$) appears in a database, and *confidence*, i.e., the conditional predictability of B, given A, calculated as the ratio of (frequency of $A \cup B$)/(frequency of A). The most popular algorithm for finding association rules is Apriori (Agrawal and Srikant 1994), and its variants. Hipp et al. (2000) present a survey and comparison of Association Rule Mining algorithms.

In the context of Web usage mining, the discovery of association rules usually aims at the discovery of associations between Web pages based on their co-occurrence in user sessions (Mobasher et al., 1999b; Cooley et al., 1999b). The work of Mobasher et al. (1999b) is particularly interesting for personalization, as the ultimate objective is to use itemsets to dynamically recommend Web pages to the users. One interesting

Table 2 Summary of classification algorithms for Web usage mining

Algorithm	Application	Classification method	Pros	Cons
HCV (Wu, 1993)	Extraction of rules representing user interests, (Ngu and Wu, 1997)	Decision Rules	• Compact rules	• Cannot handle noise • Problems withcontinuous attributes
CDL4 (Shen, 1996)	Extraction of rules representing user interests, (Jörding, 1999)	Decision Rules	• Fast • Incremental	• Complex rules
RIPPER (Cohen, 1995)	Prediction of an interest page, (Chan, 1999)	Decision Rules	• Explanatory capabilities	• Computationally expensive
C4.5 (Quinlan, 1993); CART (Breiman et al., 1984)	Prediction of an interest page, (Chan, 1999)	Decision Trees	• Flexible, publicly available implementation	• Scalability with high-dimensional data
Naive Bayesian classification (Duda and Hart, 1973)	Prediction of an interest page, (Chan, 1999)	Bayesian Classification	• Fast • Scalable	• Attribute independence assumption
Rough Set Theory	Classification of sessions according to a concept, (Maheswari et al., 2001)	Rough Set Theory	• Noise and imprecision handling	• Computationally Expensive

conclusion of this work is that, although itemsets can be used directly as input to the recommendation engine in order to provide dynamic Web pages, this is not a very accurate method. For this reason, the itemsets are clustered, using the k-means algorithm, to produce *transaction clusters*. A transaction cluster represents a group of users with similar browsing behavior. However, transaction clusters were found to be inappropriate for managing data with a large number of dimensions, i.e., Web pages recorded in Web log files. For this reason, Web pages were also grouped into *usage clusters*. Usage clusters consist of Web pages clustered together, according to their frequency of co-occurrence in user transactions. The construction of usage clusters was done with the *Association Rule Hypergraph Partition* (ARHP) technique (Han et al., 1997), which does not require distance computations (Mobasher et al., 1999b). Furthermore ARHP is useful for efficiently clustering high-dimensional datasets without requiring preprocessing for dimensionality reduction.

The application of association rule mining methods to Web usage mining is limited, focusing primarily on the prediction of the most interesting next Web page for the user. However, this type of prediction is best modeled as a sequential prediction task for which

Table 3 Summary of association discovery algorithms for Web usage mining

Algorithm	Application	Pros	Cons
ARHP (Han et al., 1997)	Clustering sessions (Mobasher et al., 1999b)	• Effective for high-dimensional data	• Frequent item problem
Bayesian Networks	• Personal user models, according to taxonomic relations of topics (Schwarzkopf, 2001) • Personal models, based on stereotypes (Ardissono and Torasso, 2000)	• Sound probabilistic modeling	• Scalability

simple association rule mining is not appropriate. This is one of the main reasons for the limited amount of work on the use of simple association rules in Web usage mining. In the following section we shall present sequential variants of Apriori, which take into account the sequence of requests recorded in a Web log file, leading to more interesting results.

The limited use of association rules in Web usage mining and the fact that most approaches to this task are variants of the same algorithm, i.e., a priori, result also in the limited scope for comparative evaluation of different methods. One problem that is common to all methods is the *frequent item problem:* items occurring together with a high frequency will also appear together in many of the resulting rules, leading to variants of the same relationship, and thus introducing computational redundancy.

A different approach to discovering associations is proposed by Schwarzkopf (2001), who employs Bayesian networks for defining taxonomic relations between the topics covered by a particular Web site. The nodes in the network constructed for each user, correspond to a stochastic variable associated with a certain topic, while the arcs represent the probabilistic dependence between topics. The probability of each variable represents the *level of interest* of a particular user in that topic. Thus, the association networks provide a graphical representation of the users' interest profiles. The networks are updated whenever new navigation data about the user are obtained. This approach provides an interesting way to model the behavior of a user, under a sound probabilistic framework, but suffers from scalability problems, due to the initial construction of the networks which is performed manually. A similar approach, using Bayesian networks is also adopted by Ardissono and Torasso (2000), in order to revise an initial user model, that has been created using stereotypes.

The discussed association discovery algorithms are summarized in Table 3, together with the respective Web Usage Mining Application that has employed them, and the pros and cons for each algorithm.

6.4. SEQUENTIAL PATTERN DISCOVERY

Sequential pattern discovery introduces the element of time in the discovery process. The aim is to identify frequently occurring temporal patterns (event sequences) in the data.

This approach is particularly useful for the identification of navigational patterns in Web usage data. Two types of method have been used for the discovery of sequential patterns: deterministic techniques, recording the navigational behavior of the user, and stochastic methods that use the sequence of Web pages that have been visited in order to predict subsequent visits.

An example of a deterministic method is the one by Spiliopoulou et al. (1999a) who have used their Web Utilization Miner (WUM) tool for sequential pattern discovery. The MINT Processor module of WUM extracts sequence rules from the pre-processed Web log data. The MINT processor provides interactive mining, using constraints that are specified by a human expert. The MINT mining language is used for the interaction, which facilitates pattern specification in an SQL-like format. This language supports predicates that can be used to specify the content, the structure and the statistics of navigation patterns. The semi-automated discovery process that is supported by WUM may be a disadvantage for its use in a fully automated personalization system. However, one could easily devise a standard set of queries, which will be executed automatically, removing thus the requirement for the human expert.

An alternative method for discovering navigational patterns using clustering is proposed by Paliouras et al. (2000b). According to this method, user sessions are represented by the transitions between pages, which were performed during the session. User sessions represented in this manner are then clustered to produce community models, which correspond to the navigational behavior of users. This simple method provides only limited, first-order modeling of the sequential patterns. However, its empirical evaluation has indicated that it can produce interesting navigational patterns.

Finally, a deterministic approach is also employed by the Clementine tool of SPSS, which uses a sequential pattern discovery algorithm, known as CAPRI. CAPRI (Clementine A-Priori Intervals) is an association rule discovery algorithm that apart from discovering relationships between items, such as Web pages, also finds the order in which these items have been traversed using temporal information. CAPRI supports various types of domain knowledge, such as start and end items, concept hierarchies, navigation templates, as well as network topologies, for finding associations across time. The CAPRI discovery process includes three phases: the *Apriori Phase*, where frequent itemsets are found, the *Discovery Phase*, where sequence trees are built, one for each possible starting item, and the *Pruning Phase*, where sequences that overlap are eliminated. CAPRI produces results that could reveal common sequences within user-specified constraints. For example it can produce a rule that states: 'If events A, B and C occur in that order, events D, E and F always follow'. CAPRI is highly scalable, but it requires the pre-specification of many input parameters.

Borges and Levene (1999) present a stochastic approach to sequential pattern discovery from user sessions, which are modeled using a hypertext probabilistic grammar. Using the terminology of this grammar, Web pages are represented by 'non-terminal symbols', links between Web pages are represented by 'production rules' and sequences of Web pages are represented by 'strings.' A directed-graph breadth-first search

algorithm, using pre-specified threshold values for support and confidence, is employed in order to identify strings, i.e., user navigation sessions that will be included in the hypertext grammar and will describe the users' browsing behavior. The algorithm is very efficient but the output, i.e. the number of rules, is strongly dependent on the selected values of the input parameters, such as the confidence.

The typical example of stochastic methods is the Markov Model, which is also the most widely adopted method to sequential pattern discovery for link prediction. Markov Models are the natural candidate for this type of pattern discovery due to their suitability to modeling sequential processes. One of the earliest applications of Markov modeling to Web data was presented in (Bestavros, 1995), employing a first-order hidden Markov model in order to predict the subsequent link that a user might follow within a certain period of time. On the same task, Sarukkai (2000) employs Markov chains to model sequences of Web pages. A probability distribution over 'preceding' links in the user's browsing history is utilized in order to assign weights to 'preceding' links, and thus create the 'best' predictors of the subsequent links. A similar approach is followed by Zhu (2001), who additionally exploits the referrer information from the log file, i.e., the page that the user has followed in order to arrive at the requested page. Finally, the method presented by Cadez et al. (2000), which was mentioned in Section 6.1, also uses a mixture of a first-order Markov models. Different Markov models are used for different clusters in order to characterize and visualize the navigational behavior of various types of user. The same method was used by Anderson et al. (2001a) to describe the browsing behavior of Web users and predict subsequent requests.

A different technique is presented by Albrecht et al. (1999), who exploit four different Markov models for predicting Web pages within a hybrid structure named *maxHybrid*. The four models are the *Time Markov Model*, that predicts the next link to be followed, based only on the last page that has been requested, the *Second-order Markov Model*, that predicts subsequent links based on the last two pages that have been requested, the *Space Markov Model*, that uses the referring page, and the *Linked Space-Time Markov Model*, that associates the referring page and the last requested page in order to predict the next user's request. Once a page has been requested, the four Markov Models calculate the probability of the next page that will be requested. The maxHybrid model uses the model that gives the highest probability for prediction. Zukerman et al. (1999), performed comparisons of the four types of model, in terms of their performance in predicting the next request correctly. The results showed that the Linked Markov model has the best performance overall.

Pitkow and Pirolli (1999) follow a different method, by extracting the longest sequences with a frequency above a threshold value. These sequences are named *Longest Repeating Subsequences* (LRS) and are used as input to two types of Markov Models the *One-Hop LRS Model* and the *All-K*th-*Order LRS Model*, in order to predict subsequent requests. The One-Hop LRS model is similar to a first-order Markov model, while the All-*K*th-Order LRS Model is similar to *K*th-Order Markov Models. The use of the LRS requests, ignoring infrequent sequences, leads to a

reduction in computational complexity, which is a serious problem for the *K*th-Order models, without apparent loss in predictive accuracy.

The main advantage of Markov models is that they can generate navigation paths that could be used automatically for prediction, without any extra processing and thus they are very useful for Web personalization. In addition they are supported by a sound mathematical background. However, their main shortcoming is that they do not produce readable user models, that could provide insight about the usage of the system. On this issue Cadez et al. (2000) propose a method for the visualization of such models.

The order of Markov models that is appropriate for sequential pattern extraction from usage data remains an open question. Higher-order Markov models seem to be needed in order to achieve better predictions since longer paths contain more information, as shown in the results of Zukerman et al. (1999). This approach though, leads to a serious increase in computational complexity. At the same time, the work of Anderson et al. (2001a,b) that employed Markov models to predict user's requests, has shown that first-order Markov models can also produce accurate results.

An overview of the aforementioned sequential pattern discovery methods is presented in the Table 4. The Web usage mining application that employs each algorithm is also presented, as well as the sequential discovery method that is pursued and the advantages and disadvantages of each approach.

6.5. PATTERN DISCOVERY FOR PERSONALIZATION

The extraction of usage patterns from Web data is essential for the efficient construction of user models. This feature is useful for all classes of personalization functions except the simplest class of memorization functions, as explained below.

6.5.1. *Memorization*

User salutation and bookmarking are the types of function that do not require pattern discovery, since they only use explicitly provided data. Hence, the pattern discovery stage can be totally ignored for this class of functions. However, personalized access rights require an apriori classification of users into categories specified by the access policy and thus, classification methods might be useful for that task.

6.5.2. *Guidance*

Guidance functionality requires mainly the employment of association discovery or sequential pattern discovery methods to facilitate the identification of related pages, or navigational patterns respectively, which can be used subsequently, for either recommending new Web pages to the visitors of a Web site or for user tutoring.

Table 4 Summary of sequential pattern discovery algorithms for Web usage mining

Algorithm	Application	Sequential pattern discovery approach	Pros	Cons
Spiliopoulou et al. (1999a)	Extraction of sequence rules, (Spiliopoulou et al., 1999a)	Deterministic	• Scalable • Meaningful patterns	• Semi-automated procedure
Paliouras et al. (2000b)	Clustering of navigationalpatterns, (Paliouras et al., 2000b)	Deterministic	• Simple • Meaningful patterns	• Limited first-order modeling patterns
CAPRI	Discovery of temporaly ordered navigational patterns	Deterministic	• Scalable • Meaningful patterns	• Requires complex input
Borges and Levene (1999)	Extraction of navigation patterns from user sessions, (Borges and Levene, 1999)	Stochastic	Automatic generation of navigation paths	• Requires heuristics for output refinement
Markov Models	Link Prediction (Bestavros, 1995), (Sarukkai, 2000), (Zhu, 2001), (Anderson et al., 2001a), (Albrecht et al., 1999), (Zukerman et al., 1999), (Pitkow and Pirolli, 1999)	Stochastic	• Sound mathematical background • Automatic generation of navigation paths	• Hard-to-interpret models • Computationally expensive

6.5.3. *Customization*

Customization functionality requires the categorization of Web site pages and/or Web site visitors based on their knowledge, interests and preferences. Hence, classification techniques can be employed in the case where the categorization classes are predefined, while clustering techniques can be exploited in the case where those classes are derived from usage data. Especially overlapping clustering methods seem appropriate since they allow the assignment of users to more than one categories and hence they offer more flexible customization.

6.5.4. *Task Performance Support*

Performing an action on behalf of the user requires mainly the discovery of typical navigation paths of the user in order to decide when this action should be performed.

Sequential pattern discovery is required for this task. Moreover, association rule discovery can be employed to facilitate the analysis of user behavior within a Web site.

7. Web Personalization based on Web Usage Mining

7.1. DISCOVERING OPERATIONAL KNOWLEDGE FOR PERSONALIZATION

Most of the work in Web usage mining employs a post-processing stage where the discovered patterns are filtered and analyzed aiming to support the decision-making process of human experts, e.g. Web site administrators, who are responsible to act accordingly. In some cases, the decisions made by the humans who receive the extracted knowledge may lead to the personalization of Web services, but even in those cases Web usage mining is not an integral part of the personalization process. Furthermore, the requirement for manual processing of the generated knowledge introduces delays and information loss. The approaches that have been introduced to accomplish the task of knowledge post-processing, are:

- *Generation of reports*, containing results of statistical analysis (e.g. Wu et al., 1998).
- *Visualization*, which is a more effective method for presenting comprehensive information to humans. This technique has been adopted, among others, by the WebViz visualization tool (Pitkow and Bharat, 1994) and the Footprints system (Wexelblat and Maes, 1997) that present navigational patterns as graphs.
- *SQL-like query mechanisms*, which are used for the extraction of rules from navigation patterns. This technique has been adopted in the WebMiner (Cooley et al., 1999c) and the WUM (Spiliopoulou and Faulstich, 1998) systems. The WebSIFT system (Cooley et al., 1999b) also provides access to the discovered patterns, i.e., the frequent itemsets, through SQL queries, but it can also feed them to a visualization module.

A more interesting approach for personalization is the integration of Web usage mining in the personalization process, by feeding the extracted knowledge directly into a personalization module that adapts the behavior of the Web-based system accordingly. This operational knowledge is delivered subsequently to the users by the means of one or more of the personalization functions already discussed in this paper. The functionality offered by a Web personalization system depends primarily on the personalization policy followed by a Web site, i.e., the ways in which information is delivered to the final user. Figure 2 illustrates how Web usage mining can be integrated in a personalized Web-based information system, following this approach. Typically, the system is separated into two parts: the online and the offline part. The online modules are those that collect the usage data and do the actual personalization, using the generated knowledge. All other steps of the data mining process are usually done offline, in order to avoid degradation of the system's performance. This approach to Web personalization follows

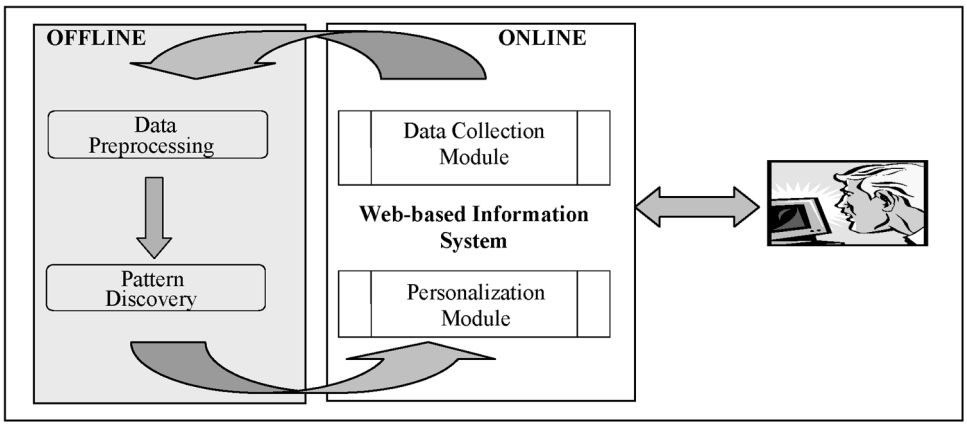

Figure 2 Web usage mining for personalization

the same basic principle that was proposed by Mobasher et al. (2000b), i.e., the separation of the system into an online and an offline part.

Despite the large variety of commercial systems and research prototypes that personalize Web-based services, the majority of personalization systems does not employ usage mining as explained above. In the remaining of this section, we examine a small number of systems that have adopted the proposed approach. The rest of the section is organized as follows. Subsection 7.2 presents the parameters that determine the personalization policy followed by a Web site. Subsection 7.3 presents several Web personalization systems, based on the personalization solution they offer and the Web usage mining methodology they adopt. We are focusing on research prototypes and commercial systems for which sufficient technical information is available in the literature. Other systems, especially commercial ones, may be following the same approach. However, the examination of these systems was unfeasible due to the lack of ample technical documentation, especially in terms of their pattern discovery methods. Subsection 7.4 summarizes the main features of the systems, paying particular attention to the use of ideas from Web usage mining. Finally, Subsection 7.5 presents ways to improve Web personalization systems further.

7.2. PERSONALIZATION POLICY

In Section 2 we discussed a variety of functions that can be supported by a Web personalization system. However, the manner in which these functions will be combined to provide a complete personalization solution depends on the personalization policy that the owner of the site wishes to follow. The personalization policy is determined by factors such as the domain of the site, the human and financial resources available, the type and the complexity of the content offered, and the constraints of the required response time. We examine here a number of technical parameters to be taken into account when designing the *personalization policy* of a site.

7.2.1. *Single-user/Multi-user*

A Web personalization system follows a single-user policy if the personalization functionality is based on personal user models, i.e., the interests, preferences and knowledge of each individual user. This is the case for instance when an e-commerce site is customizable to the buying behavior of a single user. On the other hand, a multi-user personalization policy is based on the use of aggregate models, such as user communities and stereotypes. For example a product can have the same discounted price for all users who have purchased a certain number of products.

7.2.2. *Static/Dynamic*

Personalization is considered static when the personalization functions are applied once in a user session. For example, an e-commerce site may be customized at the beginning of the session of a returning user, and no other changes are made during the rest of the session. On the other hand, dynamic personalization assumes the use of personalization functions at each step of the user's interaction with the site. For example, at each new request different recommendations may be provided to the user, depending on the user's recent browsing history. It should be noted that the choice between static and dynamic personalization is independent of the method of obtaining and maintaining information about the preferences of the user, which can be itself either static or dynamic. This policy parameter refers only to the approach of delivering personalized information to the user.

7.2.3. *Context-Sensitive/Context-Insensitive*

Personalization is considered context-sensitive when the personalization functions are adjusted to the browsing context of the user, during the session. For example, when a user is browsing books, an e-commerce site like amazon.com may not want to display recommendations about music. Alternatively, personalization can be insensitive to the browsing context of the user.

7.2.4. *Explanatory/Non-Explanatory*

Personalization is considered explanatory if an explanation is available for each personalization action performed. For example why the site recommends certain pages, or why the content of a particular page is summarized.

7.2.5. *Proactive/Conservative*

Personalization is considered proactive when certain personalization functions are performed without the user's intervention. This is the case for instance when the personalization system is allowed to perform actions on behalf of the user, such as media downloading or link redirection. On the other hand, conservative personalization leaves action control completely to the user.

7.2.6. *Converging/Diverging*

Personalization is considered converging when the personalization functions focus on a certain topic. For example, the links that are recommended or the content that is customized may direct the user to a certain topic or a certain type of product. If on the other hand more general information is provided, or other products that the user might be interested in, personalization is considered diverging.

7.3. EXISTING SYSTEMS

The personalization systems examined here have no fundamental difference from other Web usage mining systems. They employ the same methods for collecting and preprocessing the data and also for discovering interesting patterns. They only differ in the way that they post-process the discovered patterns, aiming to produce operational knowledge for personalization. Personalization is achieved by the means of various functions, implementing a certain personalization policy.

The examined systems can be divided into two generic categories: *multi-function* and *single-function* systems. Multi-function systems, can be customized to provide a variety of personalization functions, whilst single-function systems are configured to provide a single personalization functionality. The analysis of the systems presented here is based on the personalization solution they offer in terms of policy and functionality and the Web usage mining methodology they adopt in order to support this personalization solution.

7.3.1. *Multi-function Systems*

7.3.1.1. SETA (*Ardissono and Torasso, 2000*).

Personalization Solution. SETA (Ardissono et al., 1999; Ardissono and Torasso, 2000) is a prototype software platform that can be used to build customized e-commerce sites based on the users' needs. The system constructs user models for registered users and salutes them at the beginning of each session (simple memorization functionality). Furthermore, advanced personalization functionality (guidance and customization) is supported by a *Personalization Agent*, which uses the information in the user models and applies a set of *Personalization Rules* to deliver the following types of personalization:

- Recommendation of products related to the ones that the user has chosen to view.
- Content customization by varying the technical depth of each product.
- Product differentiation by presenting different features of each product.

The personalization policy of the system is context-sensitive and converging, since the products recommended to the user are directly related to the class of products that the user has chosen to view, focusing on the selected product. Furthermore, the system adopts a dynamic, single-user policy by customizing the appearance of the Web site at each individual user's step.

Web Usage Mining Methodology. SETA exploits a multi-agent architecture. One agent, the *Dialog Manager*, handles the interaction with clients and the consolidation of user data. Another agent, named *User Modeling Component*, is responsible for the initialization of user models, by matching personal information supplied by the users to stereotypical models. The revision of user models is performed dynamically using Bayesian Networks that deduce the users' behavior from features contained in the stereotypes. The revised user models are subsequently delivered to the Personalization Agent already mentioned above. The automated adaptation of user models is one of the advantages of the system, allowing more dynamic modeling than with the use of static stereotypes. However, this adaptability is limited, since it is based on a predefined set of features included in the stereotypes. Moreover, the system requires personal information to be collected at the initial step of the interaction with the user, which is undesirable, as mentioned in Section 4.1, due to the additional effort required by the users and the fact that the acquired information is not always complete and accurate.

7.3.1.2. Tellim (Jörding, 1999).

Personalization Solution. TELLIM (inTELLIgent Multimedia) is another prototype system. TELLIM personalizes the layout of Web pages at each step of the user's navigation, enhancing the presentation with multimedia content and virtual reality. A Web page, which contains detailed information for each product is created dynamically and presented to users, based on their individual preferences. Thus, content customization functionality is offered by the system. Memorization functionality is not offered in any form since the system does not store personal information. In addition to being multi-user, the personalization policy of TELLIM is context-sensitive and converging since no products of a different type are recommended, while at the same time the customized hyperlinks direct the user to more information about the same product.

Web Usage Mining Methodology. TELLIM collects information about the user's browsing behavior using a Java applet. This information is subsequently used as training data for the CDL4 algorithm (Section 6.2), and a set of rules is built, that reflect the interests of all users. Whenever a user is navigating the Web site, these rules are employed to create a temporary personalized user model, based on the behavior of the current user. The navigation of the user is used to update the rules. The construction and the update of the rules is performed offline, whilst the creation of the user model is performed in real time. One potential shortcoming of this system is the use of short-term user models, since no data about the users are kept in the system for use in other sessions of the same user. However, this is a pragmatic constraint in many e-commerce applications that do not require user registration and may not even expect to have frequent visits from the same user. A more practical problem is that the training data for the algorithm are constructed with the use of simple heuristic assumptions about the user's preferences, based on the user's actions, e.g. the selection of a link, or the interruption of downloading an image. The mapping of these actions to preferences, with the use of the heuristics may not correspond to the reality.

7.3.1.3. Schwarzkopf (2001).

Personalization solution. This system has been used for the customization of the UM2001 conference site, according to the visitor's interests. A tailor-made Web page is built for each individual user, offering information about specific parts of the site that are most likely to be of interest to the user by means of *announcements and reminders* (hyperlink customization). A bookmarking scheme is also implemented by means of *shortcuts* to pages that the user has visited (memorization functionality). Moreover, links to pages that have not been visited yet are recommended (guidance functionality), following a context-sensitive, diverging policy of presenting more generic information. Guidance and customization functionality are implemented following a static personalization policy since they are offered only at the beginning of a user session and not during the user's navigation.

Web Usage Mining Methodology. The system collects user information from Web server log files and performs user and session identification by assigning an ID, which is generated by the Web server. This ID is included in every Web page requested by the user and at the same time it is recorded in the log file replacing the IP address. The personalization solution is implemented by building offline the model of each visitor directly from the Web server log files, with the use of simple Bayesian networks, as described in Section 6.3. The model is updated automatically between user sessions, where the end of each session is determined by a 30 minutes pause in browsing. The model can also be updated by the user, who can interfere and provide personal information. However, as mentioned in Section 6.3 this approach is not scalable to larger Web sites.

7.3.1.4. Oracle9iAS Personalization (www.oracle.com).

Personalization Solution. The Oracle9iAS Personalization system, an optional part of the Oracle9i Application server, is a commercial product that offers Web personalization functionality. The system offers two types of personalization function: memorization and guidance. Memorization is offered by means of salutation to registered users, while guidance is implemented by multi-user, dynamic hyperlink recommendation. A Recommendation API enables applications employing Oracle9iAS Personalization, to deploy a variety of recommendation strategies, such as recommendation of Top items, recommendation of Cross-Sell items, or selection from 'Hot Picks', i.e., higher-margin products, perishable items etc. The recommendations are generated in real-time by predictive models that are built periodically. This functionality is offered continuously during the user session. At the same time, the system uses a content taxonomy in order to support contextual filtering, when making recommendations, i.e., suggest topics with similar content. Additionally, the system does not focus only on certain products but recommends more generic product classes, following a diverging policy.

Web Usage Mining Methodology. Oracle9iAS Personalization operates in combination with the ORACLE 9i database and uses data from both registered and

anonymous users. In the latter case, a set of Java API calls are used to capture the navigational behavior of users, e.g. pages visited, products purchased etc. Current session data are combined with other valuable information about the user such as demographics, ratings, transaction and purchase data and stored in an ORACLE database. Two mining methods are employed to create predictive models which are used to generate personalized recommendations: the *Predictive Association Rules* algorithm and the *Transactional Naïve Bayes* algorithm. The Predictive Association Rule algorithm is similar to the association discovery algorithms described in Section 6.3, using only a single item in the consequence of the rule, i.e., if A, B, and C, then D. The transactional Naïve Bayes algorithm is the same as the original Naïve Bayes algorithm although the input has a transactional format, i.e., it looks like a 'shopping basket' rather than a checklist and is better in cases where the customers buy only subsets of products. This format has the advantage of representing the data exactly in the way that they are stored in the database. The constructed models are represented by database tables and make predictions about new incoming data. This database representation of the models allows the calculation of item scores using PL/SQL procedures.

7.3.1.5. Netmind (www.mindlab.de).

Personalization Solution. NETMIND is a commercial system from Mindlab that produces multi-user recommendations. A specialized module named *Page Server*, operating as 'Reverse Proxy', receives the requested Web pages from the Web server of the site and transmits them back to users. Two different implementations of the Page Server can be used to offer the required personalization functionality. The first implementation, the *Advanced Page Server* modifies the received Web pages so as to produce context-sensitive hyperlink recommendations in a personalized manner, using a specialized module (the *Recommendation module*). The second implementation, the *Database Page Server*, is used to create dynamically Web pages on the basis of a content management system offering tailored content. Personalized layout is supported by both implementations and a dynamic personalization policy is followed, since Web pages are modified at each step of the user's navigation. Furthermore, the personalized information follows a context-insensitive, diverging policy with recommendations leading to more generic information.

Web Usage Mining Methodology. NETMIND is implemented using a modular architecture. Each user accessing the Web site is assigned a session ID, which is recorded in a log file instead of the IP addresses, and a timestamp by the *Session Manager* module. The user's navigation is recorded by another module, the *Tracker*. Neural network clustering algorithms are employed by the *Classfinder* module on the recorded data to group users and assign them to classes. New users are assigned to these classes by the *Classifier* module based on their current navigational behavior. The results of the user classification process are supplied to the *Page Server* module for implementing the personalization functionality.

7.3.1.6. Re:action (www.lumio.com)*.

Personalization Solution. Re:action is part of Lumio's Re:cognition commercial product suite, that delivers personalization functionality to the visitors of a Web site, based on their current context information, like pages visited, navigation paths and timing information. Personalized information such as recommendations and content are created by specialized system components named *Experience Advisors*, and assembled by another module, the *Experience Manager*. This personalized information is subsequently delivered to visitors using a variety of techniques. For applications that want to access directly the Experience Manager, the personalized information is deployed as a remote service using the Simple Object Access Protocol (SOAP), or the Java Remote Method Invocation (RMI). Another option is the integration of the system within a content management system. In this case XML based templating such as XSL and XSLT, is employed by the *Content Orchestration and Morphing Engine*, to deliver the content to visitors. An option for a Reverse Proxy architecture is also available. The personalized information is delivered by means of multi-user dynamic recommendations, together with content customization, using a context-sensitive, diverging personalization policy.

Web Usage Mining Methodology. Re:action implements a modular architecture and operates as an *Analytics-based Context Server*, which is a system that supports the collection of user data, and the extraction, management and deployment of user information. Users are identified and their information is collected through Javascript agents dispatched at the client side, or other code that is added to the requested Web page. The collected data reflect the visitor's experience, and are collected by the *Context Assembler* which extracts the required contextual attributes. Various types of knowledge is extracted using different components of the Re:cognition product suite. Hence, sequential knowledge such as browser behavior, and click streams is generated using Re:order, which is a tool that employs the CAPRI algorithm, segmentation knowledge such as visited pages, time spent on pages, order of pages and frequency of pages visited is generated by Re:search. Re:search generates also *Customer Profiles* which model the user's behavior. The generated knowledge is managed by a set of modules named *Experience Advisors*, which create the personal recommendations delivered subsequently to the Experience Manager for personalization.

7.3.2. *Single-function Systems*

7.3.2.1. Mobasher et al. (2000b) Yan et al. (1996) and Kamdar and Joshi (2000).

Personalization Solution. These three systems are research prototypes that are used to offer simple multi-user guidance functionality by means of static, i.e., once in a user session, recommendation of hyperlinks. The personalization policy converges to a certain topic following the user's current navigational behavior.

Web Usage Mining Methodology. WebPersonalizer (Mobasher et al., 2000b) is a system that is used for recommending Web pages to users. The online personalization

* Recently acquired by Exodus (www.exodus.gr)

module records the user's navigational behavior into a short-term model, the *active session*. Users are identified with methods described in (Cooley et al., 1999a). Group user models capturing common usage patterns are produced by a clustering method (Section 6.1). The recommendation engine matches the active user session to the clusters and recommends dynamically Web pages, in the form of hyperlinks that are embedded in the page that the user is currently visiting. The construction of the 'recommendation set' of links is based on a number of criteria, including the similarity between the current user session and each cluster, the presence of the recommended Web page in the active session, and the distance of the recommended Web page from the pages in the active session. The most recent navigation history of the user is given additional weight in the recommendation phase, through the use of an 'optimum window size' parameter. Similar approaches that employ clustering techniques to discover patterns in Web server log data for the adaptation of structure, are presented in Yan et al., 1996 and Kamdar and Joshi, 2000. The advantage of these systems is that they exploit only implicit usage information, without requiring the explicit provision of information by the user. On the other hand, the personal user models that are created by these systems represent only the current user browsing behavior, and may lead to different results for different sessions of the same user.

7.3.2.2. SiteHelper (Ngu and Wu, 1997).

Personalization Solution. SiteHelper is another research prototype that offers multi-user guidance functionality by means of static hyperlink recommendation focusing on a certain topic.

Web Usage Mining Methodology. SiteHelper, employs Web usage mining to adapt the structure of the site by dynamically personalizing the navigational information for different users. Sessions are identified by session IDs generated by the Web server and added to the Web pages requested. Information from Web access logs is augmented by an index of the Web pages in the site, which is supplied in the form of a dictionary. The SiteHelper tool employs classification techniques, as described in Section 6.2, in order to build a set of rules that represent the user's interests. Having discovered these rules the system can recommend Web pages to the users according to their interests. The construction of long-term personal user models, with the use of classification, is the main advantage of this approach. However, as mentioned above, this approach is not applicable to main e-commerce applications.

7.3.2.3. WUM (Spiliopoulou et al., 1999b).

Personalization Solution. WUM is a system that offers multi-user customization functionality by modifying the hyperlinks of a particular Web page to include links to pages that have been visited by customers and not by non-customers. The modified Web pages are presented once in a session, to non-customers, with the ultimate goal of turning them into customers. A context-sensitive, converging policy is pursued, focusing on certain topics within a particular context. The site remains unchanged if the visitor is already a customer.

Web Usage Mining Methodology. WUM divides the users accessing a specific site into 'short-time visitors,' 'non-customers' and 'customers,' according to the time spent browsing the site. Short-time visitors are filtered out of the log file. The navigation patterns of customers and non-customers are extracted using the WUM tool (Section 6.4) and compared offline, in order to identify interesting differences. A specialized proxy server, the Hyperview system (Faulstich, 1999), receives user requests, modifies the Web pages based on information provided by WUM, and sends back the modified Web pages to the users. The system uses only log file data, i.e., without requiring explicit input from the user. Furthermore, the navigation patterns are based on the current user browsing behavior, without constructing long-term user models for individual users.

7.4. ASSESSMENT OF EXISTING SYSTEMS

Web usage mining provides a new and promising approach to personalization. Some of the techniques that are used in Web usage mining have already been employed by the personalization systems. However, there are still a number of open issues regarding the personalization solution that these systems offer as well as the use of Web usage mining to realize this task. Tables 5, 6 and 7 provide a summary of the eleven systems presented in Section 7.3, according to the Web usage mining techniques they employ, the personalization functionality that they offer and the personalization policy that they follow. This section assesses the presented personalization solutions and the use of various Web usage mining techniques, aiming to identify trends and unexplored opportunities for Web personalization.

7.4.1. *Personalization Solutions*

In the majority of the systems examined above, hyperlink recommendation is the dominating functionality. This is due to the fact that initially research in Web personalization was defined as simply the recommendation of new hyperlinks in order to alleviate information overload. Hence, the Web usage mining methods that were exploited provided only that kind of functionality, using information directly from log files.

Various systems offer content or hyperlink customization. Although, almost all of these systems can be used in e-commerce applications, only one of the examined systems provides explicitly personalized product differentiation, and none of them offers personalized pricing schemes. One possible explanation is that this functionality requires a combination of accurate user data and connectivity to various databases, owned possibly by various departments within a large organization.

A number of functions have not been supported yet by any of the Web personalization systems that employ Web usage mining techniques:

- Task performance support. The initial bias of Web personalization towards enhancing the user's navigation within a site, rather than supporting other tasks, together with the fact that this particular functionality requires more sophisticated methods, are the reasons for the absence of task performance support by the examined systems.

Table 5 Summary of Web personalization systems, according to the Web usage mining techniques that they use

	Data preprocessing										
	Data collection					Session identification		Pattern discovery			
System	Access log	Cookies	Agents	User data	User identification	Time-Based	Context-Based	Clustering	Classification	Associations	Sequential patterns
SETA			√	√	√		√			√	
TELLIM			√				√		√		
(Schwarzkopf, 2001)	√				√	√				√	
Oracle9i*AS* Personalization			√	√	√		√		√	√	
NETMIND	√				√	√		√	√		
Re:action			√		√		√	√			√
WebPersonalizer	√	√			√	√	√	√		√	
(Yan et al., 1996)	√					√		√			
(Kamdar & Joshi, 2000)	√	√			√	√		√			
SiteHelper	√					√			√		
WUM	√					√					√

Table 6 Summary of Web personalization systems, according to the personalization functionality they offer

	Memorization			Guidance		Customization				
System	User salutation	Bookmarking	Personalized access rights	Recommendation of hyperlinks	User tutoring	Personalized layout	Content customization	Customization of hyperlinks	Personalized pricing scheme	Personalized product differentiation
SETA	✓			✓			✓			✓
TELLIM						✓	✓			
(Schwarzkopf, 2001)		✓		✓			✓	✓		
Oracle9i*AS* Personalization	✓			✓						
NETMIND				✓		✓	✓			
Re:action				✓			✓			
WebPersonalizer				✓						
(Yan et al., 1996)				✓						
(Kamdar & Joshi, 2000)				✓						
SiteHelper				✓						
WUM								✓		

Table 7 Summary of Web personalization systems, according to the personalization policy they pursue

System	Single/ Multi user	Static/Dynamic	Context sensitive/insensitive	Conver- ging/ Diverging
SETA	Single	Dynamic	Sensitive	Converging
TELLIM	Multi	Dynamic	Sensitive	Converging
(Schwarzkopf, 2001)	Single	Dynamic	Sensitive	Diverging
Oracle*9iAS* Personalization	Multi	Dynamic	Sensitive	Diverging
NETMIND	Multi	Dynamic	Insensitive	Diverging
Re:action	Multi	Dynamic	Sensitive	Diverging
WebPersonalizer	Multi	Static	Sensitive	Converging
(Yan et al., 1996)	Multi	Static	Sensitive	Converging
(Kamdar & Joshi, 2000)	Multi	Static	Sensitive	Converging
SiteHelper	Multi	Static	Sensitive	Converging
WUM	Multi	Static	Sensitive	Converging

- Tutoring functionality. The focus on alleviating the problem of information overload, rather than facilitating other user requirements, such as teaching, is the main reason for the lack of user tutoring functionality in the examined systems. There exists, though, a variety of Web-based personalized educational systems (Brusilovsky, 1998), which do not employ Web usage mining, but are based on traditional user modeling techniques.
- Personalized access rights, can only be offered through personal registration data, which the users are often unwilling to provide.
- Memorization functionality does not require Web usage mining and is for this reason absent from most of the examined systems.

In terms of policy, multi-user personalization is dominant, due to the difficulty of obtaining personal user models, especially for sites with many visitors. Dynamic personalization is offered mainly by more recent systems, whilst older systems preferred a static approach avoiding extensive processing, such as building a new Web page at each step of a user's session. Furthermore, a converging and context-sensitive personalization is usually pursued, due to the prime goal of Web site owners to keep visitors in their site, whilst at the same time focusing their attention on a particular type of information. Explanation is completely absent from the examined systems, as it may be considered to increase the processing load, without adding significantly to the attractiveness of a site. Finally, all systems follow a conservative policy since users are still rather cautious with the use of Web personalization, and are thus not ready to accept proactive personalization systems. The use of explanation could increase the acceptance of Web personalization and allow users to take advantage of more advanced functionality.

7.4.2. *Web Usage Mining Methodology*

7.4.2.1. *Data Collection*. At the stage of Data Collection, most of the systems mentioned above use information from access log files. In some cases, this information

is augmented with the use of cookies, in order to identify individual users, the use of agents for collecting client-side data, as well as with data provided explicitly by the users. Client-side data are richer and more reliable than access logs and are thus important for the construction of models for individual users. However, they require the implementation of separate agents to be attached to the online module, leading also to a potential degradation of the system's performance. Another data collection technique that has been ignored so far is the use of packet sniffers. This method can be of particular use to personalization, since it provides access to information that is not recorded in the access log, such as the content of the page that has been transmitted. One of the justifications for not using packet sniffers is that at their current state packet sniffers are consider a potential threat to system security. Furthermore, the use of packet sniffers requires additional software either in the online or in the offline module, which will convert and process the data included in the packets. However, packet sniffers are worth considering, as a method of augmenting the data collected from other sources.

Significant work is still needed on the protection of personal data, in order to alleviate privacy concerns during the collection of client side data, as well as during the use of cookies. In this direction, the World Wide Web Consortium (W3C) has announced the *Platform for Privacy Preferences Protocol* (*P3P*), aiming to become a general architecture facilitating privacy on the Web. The main goal of this attempt is to help users specify their personal privacy preferences, for example permitting the recording of personal data such as e-mail address and match them against privacy policies of Web sites. P3P specification employs XML and RDF to describe the syntax, structure and semantics of the exchanged information.

7.4.2.2. *Data Preprocessing*. Many of the systems examined here attempt to identify individual users, since user identification is important for the construction of individual user models. With the exception of the overly simplistic approach of mapping each IP to one user, cookies and user registration techniques are employed to perform this task. Though the technique of cookies provides a reasonable solution for implicit user identification, it is also not without problems and the user has the option of denying cookies, or even deleting them, due to privacy concerns. For this reason, explicit user identification, through a log-in procedure, as employed by SETA and Oracle9*i*AS Personalization, is still used widely.

Furthermore, user sessions are identified by all of the examined systems, using primarily time-based heuristics. However, there is a trade off between the simplicity and the accuracy of time-based heuristics, due to the problems with caching, proxy servers etc., that were mentioned in Section 5.3. The implementation of separate agents, like the one proposed by Shahabi et al. (1997), for estimating the timestamps of Web accesses have not been adopted yet by Web personalization systems. The use of context-based methods for session identification is also largely an open issue. Out of the systems examined here, five incorporate context-based methods. The use

of such methods may lead to better personalization, since it helps in identifying the interaction of individual users more accurately, as well as classifying their interests. However, these methods require additional computational effort in the session identification process that might cause performance problems.

7.4.2.3. *Pattern Discovery*. In the Pattern Discovery stage, Web personalization systems employ the standard machine learning and statistical methods that are used in all Web usage mining applications. These methods include clustering, classification, association discovery, and sequential pattern discovery. Out of these techniques, clustering has been recognized as the most appropriate method for discovering usage patterns, especially from access log data, where no preliminary knowledge, in the form of predefined classes, is usually available. However, almost all of the systems that implement clustering methods ignore an important aspect of navigational behavior, namely the sequence of requests in a user session. This is due to the limited bag-of-pages representation of the input data that is usually adopted. One step towards alleviating this problem, within the framework of usage clustering, is the use of a richer data representation, such as the use of page transitions as features in the input vectors (Paliouras et al., 2000b). An alternative is the use of Markov methods (Cadez et al, 2000, Anderson, 2001b) for modeling the navigational behavior of users.

As an unsupervised learning technique, association rule mining is also appropriate for simple log data and has been used for the discovery of associations among Web items, such as pages or products. The limited use of this method is mainly due to the relative immaturity of the approach, as compared to clustering, where a variety of different approaches exist. Other types of association, such as in the form of Bayesian networks have been used primarily on richer user data, collected with the use of agents. One justification for their less wide use on log data is the requirement for prior knowledge, in the form of model structure. Research efforts to develop model selection methods (Monti and Cooper, 1998; Heckerman et al., 2000) are still at an early stage, but they provide a promising direction for the future of Web usage mining and Web personalization.

Supervised learning methods, such as classification, seem less appropriate for an automated personalization system, due to the requirement for preclassification of the training data. The choice of categories, in a way that does not limit the process of discovery, is not a trivial task in Web personalization. Furthermore, the preclassification of a sufficient amount of data from the administrator of a Web site is difficult. However, there are personalization tasks, where the categories are easily defined and the data are naturally preclassified. One example of such a task is the discovery of the interests of an individual user, through the identification of the Web items that the user has chosen, as opposed to the remaining items in the site. Another potential use of classification, that has not been explored widely, is in the identification of stereotypical usage patterns, as proposed in (Paliouras et al., 1999).

The main surprise in the systems that we examined came from the limited use of sequential pattern discovery methods, which are particularly useful for modeling the

navigational behavior of the users. One potential reason for the hesitation in adopting these methods is the difficulty in interpreting the resulting models, e.g. the parameters of a Hidden Markov Model. Additional work in this research area is needed to address this problem. However, there are also many personalization applications that do not require the interpretability of the discovered usage patterns. In such situations, current methods for sequential discovery provide a very good solution. Furthermore, algorithms such as CAPRI, exploit sequential information while also producing reasonably interpretable models.

Finally, an important issue with all methods that seek for relations between Web pages by examining the navigation paths is the bias introduced by the existing structure of the site. Due to the fact that most people follow existing links, one could imagine that the probability of identifying relations between pages that are not connected is very small. Nevertheless, empirical results show that Web usage mining methods can still identify the need for a direct link between pages that are only indirectly connected.

7.5. OPEN ISSUES IN WEB USAGE MINING FOR PERSONALIZATION

In addition to the exploitation of Web usage mining techniques that have not been examined adequately in Web personalization, there are various other open issues that constitute promising directions for further research. Some of these are examined here.

A significant problem with most of the pattern discovery methods is their difficulty in handling very large scales of data, such as those typically collected on the Web. Despite the fact that most of the Web usage mining process can be done offline, the sizes of Web data, particularly access log data, are orders of magnitude larger than those met in common applications of machine learning. In a Web personalization system that is required to operate in real time, the scalability of the pattern discovery method can become critical. Scalability refers to the rate of increase in time and memory requirements with respect to a number of parameters, such as the number of users and pages. Even more critical, however, is the computational performance of the personalization module, incorporating the discovered patterns. Approaches that employ memory-based algorithms are particularly problematic, as they postpone generalization until run time. Model-based algorithms provide a more appropriate solution, by compressing the original data into generalized models. However, particular attention should also be paid to the scalability of these methods.

Another important requirement for the machine learning methods that are used in Web personalization is their ability to incrementally update the models that they construct. In a dynamic system, such as a Web site, it is unrealistic to assume that all of the training data will be available for reexamination every time new data are collected. Incremental machine learning algorithms have been studied since the early days of machine learning, e.g. (Utgoff, 1988), but most of the incremental methods that have been used so far in Web usage mining suffer from an important problem: they depend on the presentation order of the data (e.g. Yan et al., 1996; Fu et al., 1999).

A different but closely related problem is the incorporation of time in the discovered models. The behavior of users varies over time and it should affect the construction of models. For instance, the interest of a user in car sale advertisements is only expected to last until the user buys a car and then it should decrease suddenly. A Web personalization system should be able to adapt to the user's behavior, when this changes. This is an issue that has been examined in the areas of User Modeling (e.g. Koychev, 2000) and machine learning (e.g. Widmer and Kubat, 1996), but has not been given sufficient attention in Web personalization systems.

Additionally, there are a number of open issues concerning the use of the extracted knowledge for personalization, e.g. how often a new customized Web page or recommendation should be generated, what amount of data is considered sufficient in order to customize a site or generate a recommendation, and how can the subsequent browsing of the user can be used as feedback to the personalization process. The answers to these questions affect the design of the personalization module, as well as the choice of techniques for Web usage mining.

A related practical issue is the requirement for a common representation of the extracted knowledge, i.e., the user models generated by Web usage mining tools. In all of the examined systems, this knowledge is represented in a proprietary form that can be employed only by that particular system and cannot be shared among others. A more adaptable representation schema would facilitate the interoperability between systems. Work towards this direction is proposed by Cingil et al. (2000), who employ W3C standards, such as XML and RDF, to support collaborative filtering systems.

Moreover, there is a need for a shift in focus that is related to the functionality offered by Web personalization systems. As already discussed, almost all of the examined systems focus on the recommendation of links, or the customization of a Web site in terms of content, layout, and hyperlinks. However, there are other interesting functions, such as user tutoring and task performance support, that can exploit directly the usage data, and add more value to the browsing experience of the user. Furthermore, although the collection of personal data using registration forms is considered annoying for users, it can be motivated by the provision of advanced forms of personalization functionality, such as task performance support, or a personalized pricing scheme.

Privacy is also a major issue in Web usage mining systems employed for Web personalization. Kobsa (2001) suggests the following directives that could be pursued by Web sites that are offering personalization in order to adapt to privacy concerns:

- *Support of P3P*, to enable users and Web sites to 'exchange' their privacy preferences.
- *Intelligible Disclosure of Data*, to facilitate user comprehension of the 'system's assumptions about them.' This directive can be supported by natural language technology and visualization techniques.
- *Disclosure of Methods*, that are used to create the user model, although in the case of machine learning methods this is not always feasible.

- *Provision of organizational/technical means for users to modify their user model*, allowing users to be involved in the personalization process.
- *User model servers that support a number of anonymization methods*, to help users protect their anonymity.
- *Tailoring of user modeling methods to privacy preferences and legislation*, that provides a higher degree of flexibility to Web personalization systems.

The adoption of standards and the employment of an explanatory personalization policy, seem to be the common denominator of the above directives.

Finally, an important problem of Web personalization systems is the lack of studies comparing their performance. This is partly due to the difficulty in producing objective evaluation criteria. A reasonable solution to this problem may be to adopt a multi-level evaluation approach:

- *System evaluation*. At this level standard software engineering criteria, such as speed of response, memory management, scalability, portability and interoperability can be used. Such a system evaluation is performed in (Kobsa and Fink, 2003) who analyze a personalization server under realistic application environments.
- *Modeling performance*. At this level, the behavior of the implemented Web usage mining procedure is evaluated. For this purpose, objective criteria can be used from the area of machine learning, e.g. prediction accuracy, recall and precision.
- *Usability*. At this stage user studies are required, in order to evaluate the utility of personalization to individual users. Issues such as presentation style and clarity, system transparency and novelty of recommendation need to be assessed.

Clearly, carrying out a comparative evaluation of various systems at all three levels is a difficult task. However, the results of such an evaluation would be of great value to the design of effective Web personalization systems.

8. Conclusions

Web usage mining is an emerging technology that can help in producing personalized Web-based systems. This article provides a survey of the work in Web usage mining, focusing on its application to Web personalization. The survey aims to serve as a source of ideas for people working on the personalization of information systems, particularly those systems that are accessible over the Web.

Web personalization is seen as a fully automated process, powered by operational knowledge, in the form of user models that are generated by a Web usage mining process. A number of systems following this approach have been developed, using methods and techniques from Web usage mining, in order to realize a variety of Web personalization functions. In addition to the functions employed by existing systems, many other interesting ones have been neglected so far. The combination of recommen-

dation and customization functionality has been seen as the main solution to the information overload problem and the creation of loyal relations between the Web site and its visitors. However, other functions such as task performance support and user tutoring can certainly improve the experience of a Web site visitor.

It should be noted at this point, that Web usage mining is a very active research field and new approaches related to its application to Web personalization appear on a regular basis. However, Web usage mining is itself far from being a mature technology. As a result, there are a number of unsolved technical problems and open issues. Some of these have been presented in this survey. At the stage of data collection and preprocessing, new techniques and possibly new models for acquiring data are needed. One serious issue concerning data collection is the protection of the user's privacy. A poll by KDnuggets (15/3/2000–30/3/2000) revealed that about 70% of the users consider Web Mining as a compromise of their privacy. Thus, it is imperative that new Web usage mining tools are transparent to the user, by providing access to the data collected and clarifying the use of these data, as well as the potential benefits for the user. At the same time, one should be very careful not to burden the user with long-winded form-filling procedures, as these discourage users from accessing a Web site. Even the simple process of user registration is unacceptable for some Web-based services.

At the stage of pattern discovery, the main issue is the improvement of sequential pattern discovery methods and their incorporation into Web personalization systems. Sequential patterns are particularly important for modeling the dynamic aspects of the users' behavior, such as their navigation through a Web site. Also, the ability of pattern discovery methods to analyze efficiently very large data sets is essential, as the quantity of usage data collected in popular Web-based services exceeds that of most traditional applications of data mining. Result evaluation is another difficult issue, since most of the work in Web usage mining involves unsupervised learning, where the lack of 'ground truth' complicates the evaluation of the results. It is important to determine quantifiable performance goals for different Web usage mining tasks, in order to overcome this problem.

In addition to the various improvements to the Web usage mining process, there are a number of other issues, which need to be addressed in order to develop effective Web personalization systems. From the open issues that were mentioned in this survey, the treatment of time in the user models can be distinguished as being particularly difficult. The main source of difficulty is that the manner in which the behavior of users changes over time varies significantly with the application and possibly the type of the user. Therefore, any solution to this problem should be sufficiently parametric to cater for the requirements of different applications.

It is therefore evident that the integration of Web usage mining into the Web personalization process has introduced a number of methodological and technical issues, some of which are still open. At the same time the potential of this synergy between the two processes has barely been realized. As a result, a number of interesting directions remain unexplored. This survey has identified promising directions, providing at the same time a vehicle for exploration, in terms of Web usage mining tools and methods.

Acknowledgments

We would like to thank the four anonymous reviewers and the editor for their constructive comments.

References

Amazon, http://www.amazon.com/
CAPRI., http://www.mineit.com/
Computer Industry Almanac, http://www.c-i-a.com/
www.cyberdialogue.com
www.datamonitor.com
IBM WebSphere Personalization Server, http://www-3.ibm.com/ software/webservers/ personalization/
KDnuggets, http://www.kdnuggets.com/
MindLab, http://www.mindlab.de
Net Perceptions, http://www.netperceptions.com/
ORACLE, http://www.oracle.com
LUMIO, http://www.lumio.com
P3P, http://www.w3.org/P3P
SPSS Clementine, http://www.spss.com/clementine
USA Today, http://www.usatoday.com
W3C, World Wide Web Consortium, Extended Log Format, http://www.w3.org/TR/ WD-logfile
European Union Directive: http://www.europa.eu.int/comm/internalmarket/en/media/ dataprot/law/index.htm
Agrawal, R. and Srikant, R.: 1994, Fast algorithms for mining association rules, In: *Proceedings of the 20th VLDB Conference*, Santiago Chile, pp. 487–499.
Albrecht, D. W., Zukerman, I. and Nicholson, A. E.: 1999, Pre-sending Documents on the WWW: A Comparative Study, In: *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI99)*. (2), Stockholm, Sweden, pp. 1274–1279.
Anderson, C. R., Domingos, P. and Weld, D. S.: 2001a, Adaptive Web Navigation for Wireless Devices, In: *Proceedings of the 17th International Joint Conference on Artificial Intelligence*, (IJCAI-01), 879–884.
Anderson, C. R., Domingos, P. and Weld, D. S.: 2001b, Personalizing Web Sites for Mobile Users, In: *Proceedings of the 10th World Wide Web Conference*, (WWW10), 565–575.
Ardissono, L., Goy, A. Meo, R. Petrone, G. Console, L. Lesmo, L. Simone, C. and Torasso, P.: 1999, A configurable system for the construction of adaptive virtual stores, *World Wide Web (WWW)*, **2**(3), 143–159.
Ardissono, L. and Torasso, P.: 2000, Dynamic User Modeling in a Web Store Shell, In: *Proceedings of the 14th Conference ECAI*, Berlin, Germany, pp. 621–625.
Bestavros, A.: 1995, Using Speculation to Reduce Server Load and Service Time on the WWW, In: *Proceedings of CIKM'95: The 4th ACM International Conference on Information and Knowledge Management*, Baltimore, Maryland, 403–410.
Bezdek, J. C.: 1981, *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum Press: New York.
Biswas, G., Weinberg, J. B. and Fisher, D.: 1998, ITERATE: A conceptual clustering algorithm for data mining, *IEEE Transactions on Systems, Man and Cybernetics*, **28**, 100–111.
Borges, J. and Levene, M.: 1999, Data mining of user navigation patterns, In: *Proceedings of Workshop on Web Usage Analysis and User Profiling (WEBKDD), in conjunction with*

ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, CA., pp. 31–36.

Bouganis C., Koukopoulos, D. and Kalles, D.: 1999, A Real Time Auction System over the WWW, *Conference on Communication Networks and Distributed Systems Modeling and Simulation*, San Francisco, CA, USA, 1999.

Breese, J. S., D. Heckerman, D. and Kadie, K.: 1998, Empirical analysis of predictive algorithms for collaborative filtering, In: *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence*, San Francisco. Morgan Kaufmann Publishers, pp. 43–52.

Breiman, L., Friedman, J. H., Olshen, R. A. and Stone, C. J.: 1984, Classification and Regression Trees. *SIGMOD Record* **26**(4), Wadsworth: Belmont. CA, pp. 8–15.

Broder, A.: 2000, Data Mining, The Internet and Privacy. *WEBKDD'99, LNAI 1836*, pp. 56–73.

Brusilovsky, P.: 1998, Adaptive educational systems on the world-wide-web: A review of available technologies, In: Proceedings of workshop "www.Base & Tutoring", *Fourth International Conference in Intelligent Tutoring Systems*, San Antonio, TX, 1998.

Büchner, A. G. and Mulvenna, M. D.: 1999, Discovering Internet marketing intelligence through online analytical Web usage mining, *SIGMOD Record*, **27**(4), 54–61.

Cadez. I, Heckerman, D., Meek, C., Smyth, P. and White, S.: 2000, Visualization of Navigation Patterns on a Web Site Using Model Based Clustering. *Technical Report* MSR-TR-00-l8. Microsoft Research.

Catledge, L. D. and Pitkow, J. E.: 1995, Characterizing Browsing Strategies in the World Wide Web, *Computer Networks and ISDN Systems* **27**(6), Elsevier Science, 1065–1073.

Chakrabarti, S.: 2000, Data mining for hypertext: A tutorial survey, *ACM SIGKDD Explorations*, **1**(2), 1–11.

Chan, P. K.: 1999, A non-invasive learning approach to building Web user profiles, In: *Proceedings of 5th ACM SIGKDD International Conference, Workshop on Web Usage Analysis and User Profiling*, Springer, pp. 7–12.

Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C. and Wirth, R.: 2000, *CRISP-DM 1.0: Step-by-step data mining guide*. SPSS Inc.

Chen, M. S., Park, J. S. and Yu, P. S.: 1996, Data Mining for Path Traversal Patterns in a Web Environment, In: *Proceedings of the 16th International Conference on Distributed Computing Systems*, pp. 385–392.

Cingil, I., Dogac, A. and Azgin, A.: 2000, A Broader Approach to Personalization, *Communications of the ACM*, **43**(8), 136–141.

Cohen, W.: 1995, Fast effective rule induction, In: *Proceedings of the 12th International Conference on Machine Learning*, Morgan Kaufmann: San Mateo, CA, pp. 115–123.

Cooley, R., Mobasher, B. and Srivastava, J.: 1997a, Grouping Web page references into transactions for mining World Wide Web browsing patterns. *Technical Report* TR 97-021. Dept. of Computer Science, Univ. of Minnesota, Minneapolis, USA.

Cooley, R., Srivastava, J. and Mobasher, B.: 1997b, Web Mining: Information and Pattern Discovery on the World Wide Web, In: *Proceedings of the 9th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'97)*, pp. 558–567.

Cooley, R., Mobasher, B. and Srivastava, J.: 1999a, Data preparation for mining World Wide Web browsing patterns, *Journal of Knowledge and Information Systems*, **1**(1), 55–32.

Cooley, R., Tan, P. N. and Srivastava, J.: 1999b, WebSIFT: The Web Site Information Filter System, In: *Proceedings of the Web Usage Analysis and User Profiling Workshop (WEBKDD'99)*.

Cooley, R., Tan, P. N. and Srivastava, J.: 1999c, Discovering of interesting usage patterns from Web data. *TR 99–022*. University of Minnesota.

Cunha, C. A., Bestavros, A. and Crovella, M. E.: 1995, Characteristics of WWW Client-based Traces. *Technical Report* TR-95–010. Boston University, Department of Computer Science.

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K. and Harshman, R.: 1990, Indexing By Latent Semantic Analysis, *Journal of the American Society For Information Science*, **41**, 391–407.

Dempster, A. P., Laird, N. M. and Rubin, D. B.: 1977, Maximum likelihood from incomplete data via the EM algorithm, *Journal of the Royal Statistical Society B*, **39**, 1–38.

Duda, R. and Hart, P.: 1973, Pattern Classification and scene analysis, *Journal of Documentation*, New York: Wiley, **35**, 285–295.

Estivill-Castro, V.: 2002, Why so many clustering algorithms- A Position Paper, *SIGKDD Explorations*, **4**(1), 65–75.

Etzioni, O.: 1996, The world wide Web: Quagmire or gold mine, *Communications of the ACM*, **39**(11), 65–68.

Faulstich, L. C.: 1999, Building HyperView web sites. *Technical Report* B 99-09, Inst. of Computer Science, FU Berlin.

Feldmann, A.: 1998, Continuous online extraction of HTTP traces from packet traces, In: *Proceedings W3C Web Characterization Group Workshop.*

Fielding, R., Gettys, J., Mogul, J., Masinter, L., Leach, P. and Berners-Lee, T.: 1999, RFC 2616 – Hypertext Transfer Protocol – HTTP/1.1.

Fisher, D.: 1987, Knowledge acquisition via incremental conceptual clustering, *Machine Learning*, **2**, 139–172.

Fu, Y., Sandhu, K. and Shih, M. Y.: 1999, Clustering of Web Users Based on Access Patterns, In: *Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, Springer: San Diego.

Halkidi, M., Batistakis, Y. and Vazirgiannis, M.: 2001, On Clustering Validation Techniques, *Journal of Intelligent Information Systems*, **17**(2–3), 107–145.

Han, E. H., Karypis, G., Kumar, V. and Mobasher, B.: 1997, Clustering based on association rule hypergraphs, In: *Proceedings of SIGMOD'97 Workshop on Research issues in Data Mining and Knowledge Discovery*, 9–13.

Han, J., Cai, Y. and Cercone, N.: 1992, Knowledge discovery in databases: An attribute-oriented approach, In: *Proceedings of 18th International Conference on Very Large Databases*, Vancouver, Canada, pp. 547–559.

Han, J. and Kamber, M. 2001, *Data Mining: Concepts and Techniques*, Morgan Kaufmann Publishers.

Hanson, R., Stutz, J. and Cheeseman, P.: 1991, *Bayesian classification theory. TR-FIA-90-12-7-01.* AI Branch, *NASA Ames Research Center*, CA.

Hartigan, J.: 1975, *Clustering Algorithms.* John Wiley.

Heckerman, D., Chickering, D., Meek, C., Rounthwaite, R. and Kadie, C.: 2000, Dependency Networks for Density Estimation, Collaborative Filtering, and Data Visualization. *Technical Report* MSR-TR-00-16, Microsoft Research.

Hipp, J., Güntzer, U. and Nakhaeizadeh, G.: 2000, Algorithms for Association Rule Mining – A General Survey and Comparison, *SIGKDD Explorations*, **2**(1), 58–64.

Jörding, T.: 1999, A Temporary User Modeling Approach for Adaptive Shopping on the Web, In: *Proceedings of the 2nd Workshop on Adaptive Systems and User Modeling on the WWW*, UM'99, Banff, Canada, 75–79.

Joshi, A. and Joshi, K.: 2000, On Mining Web Access Logs, In: *ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, pp. 63–69.

Kamdar, T. and Joshi, A.: 2000, On Creating Adaptive Web Sites using WebLog Mining. *Technical Report* TR-CS-00-05. Department of Computer Science and Electrical Engineering University of Maryland, Baltimore County.

Kobsa, A., Koenemann, J. and Pohl, W.: 2001, Personalized Hypermedia Presentation Techniques for Improving Online Customer Relationships, *The Knowledge Engineering Review* **16**(2), 111–155.

Kobsa, A.: 2001, Tailoring Privacy to User's Needs. *Invited Keynote, 8th International Conference on User Modeling*, Sonthofen, Germany, 303–313.

Kobsa, A. and Fink, J.: 2003, Performance Evaluation of User Modeling Servers Under Real-World Workload Conditions, In: *Proceedings of the 9th International Conference on User Modeling*, Johnstown, PA, 143–153.

Kohonen, T.: 1997, *Self-organizing Maps (second edition)*. Springer Verlag: Berlin.

Kosala, R. and Blockeel, H.: 2000, Web Mining Research: A Survey, *SIGKDD Explorations*, **2**(1), 1–15.

Koychev, I.: 2000, Gradual Forgetting for Adaptation to Concept Drift, In: *Proceedings of ECAI 2000 Workshop 'Current Issues in Spatio-Temporal Reasoning'*, Berlin, Germany, pp. 101–106.

Kristol, D. and Montulli, L.: 2000, RFC 2965 – HTTP State Management Mechanism.

Lang, K.: 1995, NEWSWEEDER: Learning to filter news, In: *Proceedings of the 12th International Conference on Machine Learning*, Lake Tahoe, CA: Morgan Kaufmann, pp. 331–339.

Langley, P.: 1999, User modeling in adaptive interfaces, In: *Proceedings of the Seventh International Conference on User Modeling*, Banff, Canada, pp. 357–370.

Maheswari, Uma., Siromoney, V. A. and Mehata, K. M.: 2001, The Variable Precision Rough Set Model for Web Usage Mining, In: *Proceedings of the First Asia-Pacific Conference on Web Intelligence (WI'2001)*, Maebashi City, Japan, Oct 2001, Lecture Notes in Computer Science, **2198**, pp. 520–524, Springer Verlag.

Manber, U., Patel, A. and Robison, J.: 2000, Experience with Personalization on Yahoo, *Communications of the ACM*, **43**(8), 35–39.

Mitchell, T., Caruana, R., Freitag, D., McDermott, J. and Zabowski, D.: 1994, Experience with a learning personal assistant, Communications of the ACM, **37**(7), 81–91.

Mobasher, B., Jain, N., Han, E. and Srivastava, J.: 1996, Web Mining: Pattern Discovery, from World Wide Web Transactions. *TR-96050, Department of Computer Science*. DePaul University.

Mobasher, B., Cooley, R. and Srivastava, J.: 1999a, Automatic personalization based on Web usage mining. *TR-99010, Department of Computer Science*. DePaul University.

Mobasher, B., Cooley, R. and Srivastava, J.: 1999b, Creating adaptive web sites through usage-based clustering of URLs, In: *Proceedings of the 1999 IEEE Knowledge and Data Engineering Exchange Workshop (KDEX'99)*, 143–153.

Mobasher, B., Dai, H., Luo, T., Sung, Y. and Zhu, J.: 2000a, Integrating web usage and content mining for more effective personalization, In: *Proceedings of the International Conference on E-Commerce and Web Technologies (ECWeb2000)*, Greenwich, UK, pp. 165–176.

Mobasher, B., Cooley, R. and Srivastava, J.: 2000b, Automatic personalization based on Web usage mining, *Communications of the ACM*, **43**(8), 142–151.

Monti, S and Cooper, G. F.: 1998, Learning hybrid bayesian networks from data, In: Jordan M.I. (ed.), *Learning in Graphical Models*, Kluwer Academic Publishers, 521–540.

Mulvenna, M. D., Büchner, A. G., Norwood, M. T. and Grant, C.: 1997, The 'Soft-Push': mining internet data for marketing intelligence, In: *Working Conference: Electronic Commerce in the Framework of Mediterranean Countries Development*, Ioannina, Greece, pp. 333–349.

Mulvenna, M. D. and Büchner, A. G.: 1997, Data mining and electronic commerce, *Overcoming Barriers to Electronic Commerce, (OBEC '97)*, Malaga, Spain, 1–7.

Nasraoui, O., Krishnapuram, R. and Joshi, A.: 1999, Relational clustering based on a new robust estimator with applications to web mining, In: *Proceedings of the International Conf. North American Fuzzy Info. Proc. Society (NAFIPS 99)*, New York, 705–709.

Ngu, D. S. W. and Wu, X.: 1997, SiteHelper: A localized agent that helps incremental exploration of the world wide web. *Computer Networks and ISDN Systems: The International Journal of Computer and Telecommunications Networking*, **29**(8), 1249–1255.

Paliouras G., Karkaletsis, V., Papatheodorou, C. and Spyropoulos, C. D.: 1999, Exploiting learning techniques for the acquisition of user stereotypes and communities, In: *Proceedings of the International Conference on User Modeling, CISM Courses and Lectures*, **407**, pp. 169–178.

Paliouras, G., Papatheodorou, C., Karkaletsis, V., Tzitziras, P. and Spyropoulos, C. D.: 2000a, Large-Scale Mining of Usage Data on Web Sites. *AAAI Spring Symposium on Adaptive User Interfaces*. Stanford, California, 92–97.

Paliouras, G., Papatheodorou, C., Karkaletsis, V. and Spyropoulos, C. D.: 2000b, Clustering the Users of Large Web Sites into Communities, In: *Proceedings of International Conference on Machine Learning (ICML)*, Stanford, California, pp. 719–726.

Pei, J., Han, J. Mortazavi-Asl, B. and Zhu, H.: 2000, Mining access pattern efficiently from web logs, In: *Proceedings 2000 Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD'00)*, Kyoto, Japan, pp. 396–407.

Pennock, D., Horvitz, E., Lawrence, S. and Lee Giles, C.: 2000, Collaborative Filtering by Personality Diagnosis: A Hybrid Memory and Model-Based Approach. *UAI-2000: The 16th Conference on Uncertainty in Artificial Intelligence*. Stanford University, Stanford, CA, pp. 473–480.

Perkowitz, M. and Etzioni, O.: 1998, Adaptive sites: Automatically synthesizing Web pages, In: *Proceedings of the 15th National Conference on Artificial Intelligence*. Madison, Wisconsin, pp. 727–732.

Perkowitz, M. and Etzioni, O.: 2000, Adaptive Web Sites, *Communications of the ACM*, **43**(8), 152–158.

Pitkow, J., and Bharat, K.: 1994, WEBVIZ: A Tool for World-Wide Web Access Log Visualization, In: *Proceedings of the 1st International World-Wide Web Conference*. Geneva, Switzerland, 271–277.

Pitkow, J.: 1997, In search of reliable usage data on the WWW, In: *Proceedings of the 6th Int. World Wide Web Conference*, Santa Clara, CA, 451–463.

Pitkow, J. and Pirolli, P.: 1999, Mining longest repeating subsequences to predict WWW surfing, In: *Proceedings of the 1999 USENIX User Annual Technical Conference*, 139–150.

Pohl, W.: 1996, Learning about the User Modeling and Machine Learning, In: V. Moustakis, J. Herrmann (ed.), *International Conference on Machine Learning Workshop Machine Learning meets Human-Computer Interaction*, pp. 29–40.

Pretschner, A. and Gauch, S.: 1999, Personalization on the web. *Technical Report*, Information and Telecommunication Technology Center, Department of Electrical Engineering and Computer Science, The University of Kansas.

Quinlan, J. R.: 1993, *C4.5: Programs for Machine Learning*. San Mateo. CA.: Morgan Kaufmann.

Salton, G.: 1989, *Automatic Text Processing*. Addison-Wesley.

Sarukkai, R. R.: 2000, Link Prediction and Path Analysis Using Markov Chains, In: *Proceedings of the 9th World Wide Web Conference*. Amsterdam.

Schafer, B., Konstan, J. A. and Riedl, J.: 2001, E-commerce Recommendation Applications, *Data Mining and Knowledge Discovery*, **5**(1–2), 115–152, Kluwer Academic Publishers.

Schwarzkopf, E.: 2001, An adaptive web site for the UM2001 conference, In: *Proceedings of the UM2001 Workshop on Machine Learning for User Modeling*, pp. 77–86.

Shahabi, C., Zarkesh, A. M., Abidi, J. and Shah, V.: 1997, Knowledge discovery from user's Webpage navigation, In: *Proceedings of the 7th IEEE Intlernational Workshop on Research Issues in Data Engineering (RIDE)*, pp. 20–29.

Shahabi, C., Banaei-Kashani, F. and Faruque, J.: 2001, A Reliable, Efficient, and Scalable System for Web Usage Data Acquisition, In: *WebKDD'01 Workshop in conjunction with the ACM-SIGKDD 2001*, San Francisco, CA, August.

Shen, W. M.: 1996, An Efficient Algorithm for Incremental Learning of Decision Lists *Technical Report*, USC-ISI-96–012, Information Sciences Institute, University of Southern California.

Spiliopoulou, N. and Faulstich, L. C.: 1998, WUM: A Web Utilization Miner, In: *International Workshop on the Web and Databases*, Valencia, Spain, Springer LNCS 1590, 109–115.

Spiliopoulou, M.: 1999, Tutorial: Data Mining for the Web. *PKDD'99*. Prague, Czech Republic.

Spiliopoulou, M., Faulstich, L. C. and Wilkler, K.: 1999a, A data miner analyzing the navigational behavior of Web users, In: *Proceedings of the Workshop on Machine Learning in User Modeling of the ACAI99*, Chania, Greece, 54–64.

Spiliopoulou, M., Pohle, C. and Faulstich, L. C.: 1999b, Improving the effectiveness of a web site with Web usage mining, In: *Proceedings of the 1999 KDD Workshop on Web Mining, San Diego CA*, Springer-Verlag.

Srivastava, J., Cooley, R., Deshpande, M. and Tan, P. T.: 2000, Web Usage Mining: Discovery and Applications of Usage Patterns from Web Data. *SIGKDD Explorations*, **1**(2), 12–23.

Tan, P. N. and Kumar, V.: 2002, Discovery of Web Robot Sessions Based on their Navigational Patterns, *Data Mining and Knowledge Discovery*, **6**(1), 9–35.

Tauscher, L. and Greenberg, S.: 1997, How People Revisit Web Pages: Empirical Findings and Implications for the Design of History Systems, *International Journal of Human Computer Studies*, Special issue on World Wide Web Usability, **47**(1), 97–138.

Theodoridis, S. and Koutroubas, K.: 1999, *Pattern Recognition*. Academic Press.

Utgoff, P. E.: 1988, ID5: An incremental ID3, In *Proceedings of the 5th International Conference on Machine Learning*, pp. 107–120, San Mateo, CA, Morgan Kaufman.

Webb, G. I., Pazzani, M. J. and Billsus, D.: 2001, Machine Learning for User Modeling, *User Modeling and User-Adapted Interaction*, **11**, 19–29, Kluwer.

Wexelblat, A. and Maes, P.: 1997, Footprints: History-rich Web browsing, In: *Proceedings Conference Computer-Assisted Information Retrieval (RIAO)*, pp. 75–84.

Widmer, G. and Kubat, M.: 1996, Learning in the presence of concept drift and hidden contexts, *Machine Learning*, **23**(2), 69–101.

Wu, K., Yu, P. S. and Ballman, A.: 1998, Speedtracer: A Web usage mining and analysis tool, *IBM Systems Journal*, **37**(1), 89–105.

Wu, X.: 1993, The HCV Induction Algorithm, In: *Proceedings of the 21st ACM Computer Science Conference*, ACM Press, pp. 169–175.

Yan, T. W., Jacobsen, M., Garcia-Molina, H. and Dayal, U.: 1996, From User Access Patterns to Dynamic Hypertext Linking, WWW5/Computer Networks **28**(7–11), pp. 1007–1014.

Zhang, T., Ramakrishnan, R. and Livny, M.: 1996, BIRCH: an efficient data clustering method for very large databases, In: *Proceedings ACM-SIGMOD International Conference in Management of Data*, Montreal, Canada, pp. 103–114.

Zhu, T.: 2001, Using Markov Chains for Structural Link Prediction in Adaptive Web Sites, *UM 2001, LNAI 2109*, 298–300.

Zukerman, I., Albrecht, D. W. and Nicholson, A. E.: 1999, Predicting users' requests on the www, In: *Proceedings of the Seventh International Conference on User Modeling*, Banff, Canada, pp. 275–284.

Authors' vitae

Dimitrios Pierrakos is a Ph.D. candidate in Computer Science at University of Athens, Department of Informatics and Telecommunications. He also collaborated with the Institute of Informatics and Telecommunications in the National Center for Scientific Research 'Demokritos'. He received his B.Sc. in Physics from the University of Athens, and his M.Sc. in Information Technology from University College London. His research interests lie in the areas of user modeling, web mining and web personalization.

Georgios Paliouras is a researcher at the Institute of Informatics and Telecommunications, in the National Center for Scientific Research 'Demokritos'. He received his Ph.D. and M.Sc. in Computer Science from the University of Manchester (UK) and his B.Sc. (Hons) in Computing with Economics from the University of Central Lancashire (UK). His main research interest is in machine learning, but he also works in the areas of user modeling, text classification and information extraction. His current work centers around the facilitation of information access on the Web, with the aid of knowledge discovery from Web data. This includes the discovery of user communities, the use of machine learning for the construction of Web content filters and the discovery of information extraction patterns for Web pages.

Christos Papatheodorou holds a B.Sc. and a Ph.D. in Computer Science from the Department of Informatics, Athens University of Economics and Business. He is Assistant Professor at the Department of Archive and Library Sciences, Ionian University, Corfu, Greece, where he teaches Information Systems and Information Retrieval. Before joining Ionian University, he was in charge of the development of Digital Library and Information services at the Library of the National Centre for Scientific Research 'Demokritos', Athens, Greece. His research interests include User Modeling, Web Mining and Digital Library Usability. He has participated as project proposal evaluator in the European Union Information Society Technologies Research Programme and he has been involved in several national and international R&D projects.

Constantine D. Spyropoulos is an ECCAI Fellow and a research director at the Institute of Informatics and Telecommunications, in the National Center for Scientific Research 'Demokritos'. He received his Ph.D. and M.Sc. in Computer Science from Loughborough University of Technology (UK) and his B.Sc. in Applied Mathematics from the University of Ioannina (GR). He is head of the Software and Knowledge Engineering Laboratory and has been scientific responsible for many National and European projects. His current research and scientific activities focus on artificial intelligence, information extraction, multilingual generation, personalization and software internationalization and localization.