3

Data Mining for Web Personalization

Bamshad Mobasher

Center for Web Intelligence
School of Computer Science, Telecommunication, and Information Systems
DePaul University, Chicago, Illinois, USA
mobasher@cs.depaul.edu

Abstract. In this chapter we present an overview of Web personalization process viewed as an application of data mining requiring support for all the phases of a typical data mining cycle. These phases include data collection and preprocessing, pattern discovery and evaluation, and finally applying the discovered knowledge in real-time to mediate between the user and the Web. This view of the personalization process provides added flexibility in leveraging multiple data sources and in effectively using the discovered models in an automatic personalization system. The chapter provides a detailed discussion of a host of activities and techniques used at different stages of this cycle, including the preprocessing and integration of data from multiple sources, as well as pattern discovery techniques that are typically applied to this data. We consider a number of classes of data mining algorithms used particularly for Web personalization, including techniques based on clustering, association rule discovery, sequential pattern mining, Markov models, and probabilistic mixture and hidden (latent) variable models. Finally, we discuss hybrid data mining frameworks that leverage data from a variety of channels to provide more effective personalization solutions.

3.1 Introduction

The ultimate goal of any user-adaptive system is to provide users with what they need without them asking for it explicitly [89]. Automatic personalization, therefore, is a central technology used in such systems. In the context of the Web, personalization implies the delivery of dynamic content, such as textual elements, links, advertisement, product recommendations, etc., that are tailored to needs or interests of a particular user or a segment of users.

We distinguish between "automatic personalization" and what is sometimes referred to as "customization". Both customization and personalization refer to the delivery of content tailored to a particular user. What separates these two notions is who controls the creation of user profiles as well as the presentation of interface elements to the user. In customization, the users are in control of (often manually) specifying their preferences or requirements, based on which the interface elements are created. Examples of customization on the Web include customized Web sites, such as MyYahoo (www.yahoo.com), and a variety of e-commerce Web sites (such as www.dell.com)

P. Brusilovsky, A. Kobsa, and W. Nejdl (Eds.): The Adaptive Web, LNCS 4321, pp. 90–135, 2007.

that allow for manual configurations of systems or services before purchase. Automatic personalization, on the other hand, implies that the user profiles are created, and potentially updated, automatically by the system with minimal explicit control by the user. Examples of automatic personalization in commercial systems include Amazon.com's personalized recommendations, music or playlist recommenders such as Mystrand.com, and a variety of news filtering agents available today.

Traditional approaches to automatic personalization have included content-based, collaborative, and rule-based filtering systems. Each of these approaches is distinguished by the specific type of data collected to construct user profiles, and by the specific type of algorithmic approach used to provide personalized content. Generally, the process of personalization consists of a data collection phase in which the information pertaining to user interests is obtained and a learning phase in which user profiles are constructed from the data collected. Learning from data can be classified into memory based (also known as lazy) learning and model based (or eager) learning depending on whether the learning is done online while the system is performing the personalization tasks or offline using training data.

Standard user-based collaborative filtering and most content based filtering systems that use lazy learning algorithms are examples of the memory-based approach to personalization, while item-based and other collaborative filtering approaches that learn models prior to deployment are examples of model-based personalization systems.

Memory based systems simply memorize all the data and generalize from it at the time of generating recommendations. They are therefore more susceptible to scalability issues. Model-based approaches, that perform the computationally expensive learning phase offline, generally tend to scale better than memory based systems during the online deployment stage. On the other hand, as more data is collected, memory based systems are generally better at adapting to changes in user interests compared to model based techniques in which model must either be incremental or be rebuilt in order to account for the new data. These advantages and shortcomings have led to an extensive body of research and practice comprised of a variety of personalization or recommender systems that generally fall into the aforementioned categories.

Our goal in this chapter is not to provide an overview of automatic personalization, in general. Rather, we focus more specifically on *Web personalization* where the recommended objects come from a repository of Web objects (items or pages) browseable through navigation of links between the objects, usually in a particular Web site. Furthermore, we are particularly interested in a data mining approach to personalization where the goal is to leverage all available information about users of the Web site to deliver a personal experience.

Kohavi et al. [62] suggest five desiderata for success in data mining applications:

- data rich with descriptions to enable search for patterns beyond simple correlations;
- large volume of data to allow for building reliable models;
- controlled and reliable (automated) data collection;
- the ability to evaluate results; and
- ease of integration with existing processes (to build systems that can effectively take advantage of the mined knowledge).

Seldom are all these criteria satisfied in a typical data mining application. Personalization on the Web, and more specifically in e-commerce, has been considered the "killer app" for data mining, in part because many of these elements are indeed present. However, to be able to take full advantage of the flexibility provided by the data, and to effectively use the discovered models in an automatic personalization system, the process of personalization must be viewed as an application of data mining requiring support for all the phases of a typical data mining cycle [27], including data collection, pre-processing, pattern discovery and evaluation, in an off-line mode, and finally the deployment of the knowledge in real-time to mediate between the user and the Web.

The advantages and flexibilities afforded by the data mining approach to personalization come precisely from the fact that personalization is viewed as a holistic process rather than as individual algorithms or specific data types. Indeed, many of the traditional algorithms used for personalization can also be placed within the context of this process.

In this chapter we present a comprehensive view of the data mining approaches to personalization. We focus primarily on Web usage mining where the goal is to leverage data collected as a result of user interactions with the Web in order to learn user models and to use these models for personalization. We provide a detailed discussion of a host of data mining activities necessary for this process, including the preprocessing and integration of data from multiple sources, common pattern discovery techniques that are applied to this data in order to derive aggregate user models, and recommendation algorithms for combining the discovered knowledge with the current status of a user's activity in a Web site to provide personalized content to a user.

The remainder of this chapter is organized as follows. In Section 3.2 we provide a brief background on traditional approaches to automatic personalization and methods for profile generation based on different types of data. This discussion motivates our focus on the data mining approach. In Section 3.3, we discuss the essential data modeling and representation issues relevant to the personalization tasks, and in particular, provide a detailed discussion of the preprocessing and integration stage of the data mining cycle in the context of Web usage mining. Section 3.4, we consider a number of classes of data mining algorithms used particularly for Web personalization, and for each class, we present a number of specific approaches used in the literature. In this Section, we also discuss some of the shortcomings of the pure usage-based approaches and show how hybrid data mining frameworks, that leverage data from a variety of sources, can provide potential solutions to these shortcomings. Finally, in Section 3.5, we provide an overview dimensions along which personalization models can be evaluated an discuss some of commonly used evaluation metrics.

3.2 Automatic Personalization and Data Mining

The ability of a personalization system to tailor content and recommend items implies that it must be able to infer what a user requires based on previous or current interactions with that user, and possibly other users. The personalization task can therefore be viewed as a prediction problem: the system must attempt to predict the user's level of interest in, or the utility of, specific content categories, pages, or items, and rank these according to their predicted values. Furthermore, the task of delivering personalized content is often framed in terms of a recommendation task in which the system recommends items with the highest predicted interest values or utilities to an active user. In general, a personalization system can be viewed as a mapping of users and items to a set of "interest values". The view of personalization function as a prediction task comes from the fact that this mapping is not, in general, defined on the whole domain of user-item pairs, and thus requires the system to estimate the interest values for some elements of the domain.

Automatic personalization systems, generally, differ in the type of data and the method used to create user profiles, and in the type of algorithmic approaches used to make predictions. We will briefly describe each of these two dimensions below and provide an overview of the data mining approach to personalization which will guide our discussion in the remainder of this chapter.

3.2.1 Approaches to Personalization

From an architectural and algorithmic point of view personalization systems fall into three basic categories: Rule-based systems, content-filtering systems, and collaborative filtering systems. Our primary focus in this chapter is on model-based approaches to collaborative filtering in which models are learned through a variety of data mining techniques. However, we provide brief descriptions of each of these categories below. Additional details on traditional (e.g., memory-based) collaborative filtering techniques and content-based filtering algorithms can be found in Chapters 9 [117] and 10 [103] of this book, respectively. Furthermore, a great deal of work has focused on creating hybrid systems that combine various elements of these algorithms. A detailed characterization of hybrid recommender systems can be found in Chapter 12 [22].

Rule-Based Personalization Systems. Rule-based filtering systems rely on manually or automatically generated decision rules that are used to recommend items to users. Many existing e-commerce Web sites that employ personalization or recommendation technologies use manual rule-based systems. Such systems allow Web site administrators to specify rules, often based on demographic, psychographic, or other personal characteristics of users. In some cases, the rules may be highly domain dependent and reflect particular business objectives of the Web site. The rules are used to affect the content served to a user whose profile satisfies one or more rule conditions. Like most rule-based systems, this type of personalization relies heavily on knowledge engineering by system designers to construct a rule base in accordance to the specific characteristics of the domain or market research. The user profiles are generally obtained through explicit interactions with users. Some research has focused on machine learning techniques for classifying users into one of several categories based on their demographic attributes, and therefore, automatically derive decision rules that can be used for personalization [101].

The primary drawbacks of rule-based filtering techniques, in addition to the usual knowledge engineering bottleneck problem, emanate from the methods used for the generation of user profiles. The input is usually the subjective description of users or their interests by the users themselves, and thus is prone to bias. Furthermore, the profiles are often static, and thus the system performance degrades over time as the profiles age.

Content-Based Filtering Systems. In Content-based filtering systems, a user profile represent the content descriptions of items in which that user has previously expressed interest. The content descriptions of items are represented by a set of features or attributes that characterize that item. The recommendation generation task in such systems usually involves the comparison of extracted features from unseen or unrated items with content descriptions in the user profile. Items that are considered sufficiently similar to the user profile are recommended to the user.

In most content-based filtering systems, particularly those used on the Web and in e-commerce applications, the content descriptions are textual features extracted from Web pages or product descriptions. As such, these systems often rely on well-known document modeling techniques with roots in information retrieval [112] and information filtering [11] research. Both user profiles, as well as, items themselves, as represented as weighted term vectors (e.g., based on TF.IDF term-weighting model [112]). Predictions of user interest in a particular item can be derived based on the computation of vector similarities (e.g., using the Cosine similarity measure) or using probabilistic approaches such as Bayesian classification. Furthermore, in contrast with approaches based on collaborative filtering, the profiles are individual in nature, built only from features associated with items previously seen or rated by the active user. Chapter 5 of this book [76] provides a more detailed discussion of various approaches used in Web document modeling.

Examples of early personalized agents using this approach include Letizia [70], NewsWeeder [68], Personal WebWatcher [79], InfoFinder [66], Syskill and Webert [102], and the naïve Bayes nearest neighbor approach used by Schwab et al. [120]. A survey of the commonly used text-learning techniques in the context of content-based filtering can be found in [80].

The primary drawback of content-based filtering systems is their tendency to overspecialize the item selection since profiles are solely based on the user's previous rating of items. User studies have shown that users find online recommenders most useful when they recommend unexpected items [124], suggesting that using content similarity alone may result in missing important "pragmatic" relationships among Web objects such as their common or complementary utility in the context of a particular task. Furthermore, content-based filtering requires that items can be represented effectively using extracted textual features which is not alway practical given the heterogeneous nature of Web data.

A more detailed discussion of content-based filtering systems is provided in Chapter 10 [103].

Collaborative Filtering Systems. Collaborative filtering [64, 49] has tried to address some of the shortcomings of other approaches mentioned above. Particularly, in the context of e-commerce, recommender systems based on collaborative filtering have achieved notable successes [118]. These techniques generally involve matching the ratings of a current user for objects (e.g., movies or products) with those of similar users (nearest neighbors) in order to produce recommendations for objects not yet rated or seen by an active user. Traditionally, the primary technique used to accomplish this task is the standard memory-based k-Nearest-Neighbor (kNN) classification approach which compares a target user's profile with the historical profiles of other users in order to find the top k users who have similar tastes or interests.

However, collaborative filtering techniques have their own potentially serious limitations. The most important of these limitations is their lack of scalability. Essentially, kNN requires that the neighborhood formation phase be performed as an online process (i.e., the modeling phase is performed in real-time, in contrast to model-based approaches in which model learning is performed off-line from training data). As the numbers of users and items increase, this approach may lead to unacceptable latency for providing recommendations or dynamic content during user interaction.

Another limitation of kNN-based techniques emanates from the sparce nature of the dataset. As the number of items in the database increases, the density of each user record with respect to these items will decrease. This, in turn, will decrease the likelihood of a significant overlap of visited or rated items among pairs of users resulting in less reliable computed correlations. Furthermore, collaborative filtering usually performs best when explicit non-binary user ratings for similar objects are available. In many Web sites, however, it may be desirable to integrate the personalization actions throughout the site involving different types of objects, including navigational and content pages, as well as implicit product-oriented user events such as shopping cart changes, or product information requests.

A number of optimization strategies have been proposed and employed to remedy these shortcomings [2, 116, 140, 143]. These strategies include similarity indexing and dimensionality reduction to reduce real-time search costs and remedy the sparsity problems, as well as offline clustering of user records, allowing the online component of the system to search only within a matching cluster. A model-based variant of collaborative filtering is known as *item-based* collaborative filtering [114] in which, starting from the same user-rating profile databases, an item-item similarity matrix is built offline, and used in the prediction phase to generate recommendations. Rather than basing item similarity on content descriptions of the items, similarity between items is based on user ratings of these items. Each item is represented by a vector, and the similarities are computed using metrics such as cosine similarity and correlation-based similarity. The recommendation process predicts the rating for items not previously seen or rated by an active user using a weighted sum of the ratings, by that user, of items in the item neighborhood of the target item. Evaluation of the item-based collaborative filtering approach [35] has shown that item-based collaborative filtering can provide recommendations that are, in general, of similar quality when compared to memory-based collaborative approach.

Most data mining approaches to personalization can be viewed as extensions of collaborative filtering. In these approaches the pattern discovery algorithms take as input the historical rating or navigational profiles of past users and generate aggregate user models. The user models, in turn, can be used, in conjunction with the profile of an active user, to predict future user behavior or generate recommendations. This viewpoint will guide our presentation through the subsequent sections of this chapter.

A more detailed discussion of collaborative filtering systems is provided in Chapter 9 [117].

3.2.2 Approaches to User Profiling

All approaches to personalization, and to a greater degree, personalization based on data mining, require the collection of data that accurately reflect the interests of the users and their interactions with applications and items. Personalized systems differ, not only in the algorithms used to generate recommendations or make prediction, but also in the manner in which user profiles are built using this underlying data.

Rule-based and content-based personalization systems generally build an individual model of user interests and use this profile to tailor future interactions with only that user. As noted earlier, the content-based filtering systems require content features of items extracted from item descriptions, or relational attributes associates with items in the backend databases. In such systems the process of building a profile for a user requires two stages. First, the system must determine the level of user interest in a subset of items. This task may be accomplished implicitly by passively observing the user and using various heuristics to classify items as interesting or non-interesting [70, 79], or it can be based on explicit user judgment assigning ratings to items or manually identifying positive and negative examples [68, 102]. The transformation of each item (usually a Web page or document) into a bag or words (vector) representation, with each token being assigned a weight using methods such as *TF.IDF* [112] or minimum description length [109]. The profile is then used to recommend other similar items to the user. A major disadvantage of approaches based on an individual profiles is the lack of serendipity as recommendations are very focused on the user's previous interests. Also, the system depends on the availability of content descriptions of the items being recommended.

In the case of rule-based systems, particularly those based on demographic filtering, each user profile may be represented by a vector of personal and demographic attributes, sometimes called a *fingerprint*. In e-commerce and Web analytics applications, the visitor fingerprints may also include such computed attributes as total amount spent as well as the recency and frequency of purchase or visit. Few systems use demographic data within the recommendation process. This is due to the fact that such data is more difficult to collect on the Web and, when collected, tends to be of poor quality. Also, recommendations purely based on demographic data have been shown to be less accurate than those based on the item content and user behavior [101]. In Lifestyle Finder [65], externally procured demographic data (Claritas's PRIZM) was used to enhance demographic attributes obtained from the user, through an iterative process where the system only requests information pertinent to classifying the user into one of 62 demographic clusters defined within the PRIZM classification. Once classified, objects most relevant to that demographic cluster are recommended to the user.

In collaborative filtering, the system not only uses the profile for the active user but also maintains a database of other users' profiles. In contrast to content-based filtering in which item-to-item similarities form the basis for recommendation generation, collaborative systems rely on user-to-user similarities. Profiles are generally represented as a vector or set of ratings providing the user's preferences on a subset of items. An active user's profile is used to find other users with similar preferences, referred to as the active user's neighborhood. Note that as opposed to content-based filtering, the actual content descriptions of items are not part of the profile.

While traditional collaborative filtering only uses rating data, hybrid collaborative approaches that utilize both content and user rating data have also been proposed [6, 28, 75]. Furthermore, both in the case of collaborative and content-based filtering, various approaches have been explored to integrate ontological domain knowledge with user profiles [77, 45, 122, 145, 139]. In the presence of a domain ontology, the user profiles may actually reflect the structure of the domain, and thus may require a more complex representation than the flat representations used in standard approaches.

Regardless of the algorithmic approach to personalization, the data for user profiling can be collected implicitly or explicitly. Explicit collection usually requires the user's active participation. In systems that rely on demographic or personal information user interaction may take the form of participating in online surveys at the time of registration or providing personal and financial information during a purchase (which can then be combined with offline demographic data available through a variety of data aggregation services). Similarly, as noted above, content-based filtering systems can also use either implicit or explicit user feedback to determine the level of user interest in items. Traditional collaborative filtering systems used in e-commerce generally use explicit user feedback in the form of ratings on individual items. However, many collaborative systems, particularly Web personalization systems that use clickstream or other types of behavioral data, attempt to measure user interest in individual or groups of items based on heuristic indicators (such as time spent viewing the item, whether the item is purchased, etc.). Many e-commerce systems, such as Amazon.com, monitor each customer's purchase and activity history and use information as part of the user profiles.

The advantage of using implicit feedback for user profiling is that it removes the burden associated with providing personal information from the user. The system collects relevant data, based on users' observed behavior, and infers user-specific information. Implicit profiling implies that the system must be able to track and monitor user behavior in order to identify browsing or buying patterns. Implicit data could be collected on the client or on the server side. Approaches to personalization can be classified based on whether these approaches have been developed to run on the client side or on the server-side. The key distinction between these personalization approaches is the breadth of data that are available to the personalization system. On the client side, data is only available about the individual user and hence the only approach possible on the client side is *individual*. On the server side, the business has the ability to collect data on all its visitors and hence both individual and collaborative approaches can be applied. On the other hand, server side approaches generally only have access to interactions of users with content on their Web site while client side approaches can access data on the individual's interaction with multiple Web sites.

Most client side applications are content-based systems aimed at personalized search across the Web or multiple repositories [99, 122, 26, 138]. The lack of common domain ontologies across Web sites, the unstructured nature of the Web, and the sparseness of available behavioral data currently reduce the possibilities for personalization of navigational as opposed to search based interactions with the Web as whole.

Collaborative personalization systems based on Web usage mining, which are the primary focus of the remainder of this Chapter, rely on clickstream and navigational data automatically collected by Web and application servers and stored in server log file. Another source of customer data are transaction databases, pre-sale and after-sale support data, or demographic information. Such data could be dynamically collected by a Web site or purchased from third parties. In many cases data is stored in different formats in multiple, disparate databases.

We focus primarily on profiles built from implicit user feedback, collected automatically by monitoring users' activity histories, generally on the server-side. Our discussion is mainly centered around the application of data mining methodology and machine learning techniques that attempt to learn group profiles and generate user models that can be used to tailor a Web site's interactions with future users.

For a detailed discussion of various approaches to Web user profiling see Chapter 2 of this book [40].

3.2.3 Data Mining Approach to Personalization

The foregoing background motivates our focus on data mining (and more specifically, Web usage mining) as an approach to personalization. What makes the data mining approach to Web personalization different from the other approaches discussed above, is that Web usage mining is not a specific algorithm, but rather it follows the typical data mining cycle. As such, it provides a great deal of flexibility for leveraging different data channels in a comprehensive manner, and allows for the personalization tasks to be better integrated with other existing applications. Furthermore, because of the focus of data mining on efficient model-based pattern discovery algorithms, personalized systems based on data mining tend to be more scalable than those based on traditional approaches such as standard collaborative filtering.

Web usage mining [31, 130, 81] can be defined as the automatic discovery and analysis of patterns in clickstream and associated data collected or generated as a result of user interactions with Web resources on one or more Web sites. The goal of Web usage mining is to capture, model, and analyze the behavioral patterns and profiles of users interacting with a Web site. The discovered patterns are usually represented as collections of pages, objects, or resources that are frequently accessed by groups of users with common needs or interests.

Traditionally, the goal of Web usage mining has been to support the decision making processes by Web site operators in gaining better understanding of their visitors, create a more efficient or useful organization for the Web sites, and to do more effective marketing. However, these models can also be used by adaptive systems automatically in order to achieve various personalization functions.

The overall process of Web personalization based on Web usage mining consists of three phases: data preparation and transformation, pattern discovery, and recommendation. Of these, only the latter phase is performed in real-time.

The data preparation phase transforms raw Web log files into user profile or Web transaction data that can be processed by data mining tasks. This phase also includes data integration from multiple sources, such as backend databases, application servers, and site content. A variety of data mining techniques can be applied to this data in the pattern discovery phase, such as clustering, association rule mining, sequential pattern discovery, and probabilistic modeling. The results of the mining phase are transformed into aggregate user models, suitable for use in the recommendation phase. The recommendation engine considers the active user's profile in conjunction with the discovered patterns to provide personalized content.

In the following sections, we provide a detailed overview of the techniques and algorithms used in each of these phases.

3.3 Data Collection, Preprocessing, and Modeling

Viewing personalization as a data mining application, the aim is to create a set of user-centric data models (user profiles), representing the interests and activities of all users, that can be used as input to a variety machine learning algorithms for pattern discovery. The output from these algorithms, i.e., the patterns discovered, can then be used for predicting future interests of users. The exact representations of these user models differ based on the approach taken to achieve personalization and the granularity of the information available. The pattern discovery tasks would therefore differ in complexity based on the expressiveness of the user profile representation chosen and the data available.

3.3.1 Data Modeling and Representation

For the purposes of our discussion, we assume the existence of a set of m users, $U = \{u_1, u_2, \cdots, u_m\}$ and a set of n items, $I = \{i_1, i_2, \cdots, i_n\}$. We represent the profile for a user $u \in U$ as an n-dimensional vector of ordered pairs,

$$u^{(n)} = \langle (i_1, s_u(i_1)), (i_2, s_u(i_2)), \cdots, (i_n, s_u(i_n)) \rangle, \tag{3.1}$$

where i_j's $\in I$ and s_u is a function for user u assigning (possibly null) interest scores to items.

In a typical data mining approach, such profiles are collected over time and stored for all users interacting with the system. Conceptually, the database of all user profiles can be represented as the $m \times n$ matrix, $UP = [s_{u_k}(i_j)]_{m \times n}$, where $s_{u_k}(i_j)$ is the degree of interest in item i_j by a user u_k.

Formally, a *personalization system* can be viewed as a mapping $PS : \mathcal{P}(UP) \times U \times I \to R \cup \{\texttt{null}\}$, assigning interest values to pairs of users and items, according to a set of user profiles. Because the mapping PS is not, in general, defined on the whole domain of user-item pairs, the system must estimate or predict the interest scores of a given user for elements of the domain. Depending on the prediction algorithm used, the system may not be able to an interest score for a particular user-item pair, in which case the PS mapping produces a `null` value. In other words, the task of a personalization system can be viewed as one of predicting, for a given *target user* $u_k \in U$ and a *target item* $i_j \in I$, and the databases of user profiles UP, $PS(UP, u_k, i_j) = s_{u_k}(i_j)$.

Indeed, most of the approaches to personalization and user profiling, discussed in Sections 3.2.1 and 3.2.2 can be placed within this general framework. In content-based and some rule-based approaches, the user profile databases UP contains only a single profile, that of the target user, u_k, and the prediction of interest score, $s_{u_k}(i_j)$ for the target item i_j is based on its similarity to the user profile or based on the demographic or other personal attributes of the user. On the other hand, in the standard collaborative filtering context, the interest scores usually represent rating values from an ordered but discrete scale, and UP contains the past ratings of all users of the system. It that case, the prediction or estimation of the interest score for the target user is based, usually, on the similarity of that user's profile to other profiles in UP.

In the data mining approach to personalization, a variety of machine learning techniques are applied to UP in order to discover aggregate user models based on which a prediction is made for the target user. More specifically, in the context of personalization based on Web usage mining, our main focus in the remainder of this chapter, UP generally contains user transaction records representing their online activity (including clickthroughs or purchase transactions) in one or more sessions. The items are data abstractions representing pages, content categories, or products available on the Web site, and the interest scores are usually derived based on implicit observation of user activity on the Web site, such as time spent on a page, the purchase or selection of a product, etc.

Based on the above discussion, there are two important questions that must be answered before any type of pattern discovery or prediction can be performed: (a) what elements constitute the items in I, and (b) how is the function s_{u_k} defined for each user u_k? The answers to these questions, of course, depend on the type of approaches used for personalization and user profiling, the underlying application domain, and the types and sources of data available. In the knowledge discovery framework, the generation of the user-centric data representation is achieved through the application of several (often domain-specific) data collection, manipulation, and transformation operations. Collectively, we call the application of these operations the *data preprocessing* stage.

The goal of the preprocessing stage is to transform the raw data into a set of data abstractions that can be used in the above general framework. This includes the extraction and transformation of features or attributes that can be used to represent each item, as well as the extraction and transformation of explicit or implicit user attributes that are used to determine users' interest in various items (i.e., the functions $s_{u_k}(\cdot)$). As noted earlier, the extraction and transformation tasks vary depending on the application domain and context of personalization. Because our primary focus is on Web personalization, i.e., personalization of the Web users' navigational experience, in the following discussion we focus primarily on data preprocessing tasks for Web usage mining.

3.3.2 Data Sources for Web Usage Mining

The primary data sources used in Web usage mining are the server log files, which include Web server access logs and application server logs. Additional data sources that are also essential for both data preparation and pattern discovery include the site files and meta-data (including content features and structural elements of pages), operational databases, application templates, and domain knowledge [32, 130, 82]. In some cases and for some users, additional data may be available due to client-side or proxy-level (Internet Service Provider) data collection, as well as from external clickstream or demographic data sources (e.g., ComScore, NetRatings, MediaMetrix, and Acxiom).

The most important of these sources for Web usage mining is the clickstream data recorded automatically by the Web and application servers in log files. This data represents the fine-grained navigational behavior of visitors. Each hit against the server, corresponding to an HTTP request, generates a single entry in the server access logs. Each log entry (depending on the log format) may contain fields identifying the time and date of the request, the IP address of the client, the resource requested, possible parameters used in invoking a Web application, status of the request, HTTP method used, the user agent (browser and operating system type and version), the referring Web resource, and, if available, client-side cookies which uniquely identify a repeat visitor.

Depending on the goals of the analysis, this data needs to be transformed and aggregated at different levels of abstraction. In Web usage mining, the most basic level of data abstraction is that of a *pageview*. A pageview is an aggregate representation of a collection of Web objects contributing to the display on a user's browser resulting from a single user action (such as a click-through). Conceptually, each pageview can be viewed as a collection of Web objects or resources representing a specific "user event", e.g., reading an article, viewing a product page, or adding a product to the shopping cart. At the user level, the most basic level of behavioral abstraction is that of a *session*. A session is a sequence of pageviews by a single user during a single visit. The notion of a session can be further abstracted by selecting a subset of pageviews in the session that are significant or relevant for the analysis tasks at hand. A session can be used directly as the user profile (as described in the formal representation given in 3.1). However, if the goal of analysis is to capture the behavior of users over time (i.e., over multiple sessions), all sessions belonging to a user can be combined and aggregated to create the profile for that user.

The content data in a site is the collection of objects and relationships that are conveyed to the user. For the most part, this data is comprised of combinations of textual material and images. The data sources used to deliver or generate this data include static HTML/XML pages, multimedia files, dynamically generated page segments from scripts, and collections of records from the operational databases. The site content data also includes semantic or structural meta-data embedded within the site or individual pages, such as descriptive keywords, document attributes, semantic tags, or HTTP variables. The underlying domain ontology for the site is also considered part of the content data. Domain ontologies may include conceptual hierarchies over page contents, such as product categories, explicit representations of semantic content and relationships via an ontology language such as RDF, or a database schema over the data contained in the operational databases.

The structure data represents the designer's view of the content organization within the site. This organization is captured via the inter-page linkage structure among pages, as reflected through hyperlinks. The structure data also includes the intra-page structure of the content within a page. For example, both HTML and XML documents can be represented as tree structures over the space of tags in the page. The hyperlink structure for a site is normally captured by an automatically generated "site map", usually represented as a directed graph. A site mapping tool must have the capability to capture and represent the inter- and intra-pageview relationships. For dynamically generated pages, the site mapping tools must either incorporate intrinsic knowledge of the underlying applications and scripts, or must have the ability to generate content segments using a sampling of parameters passed to such applications or scripts.

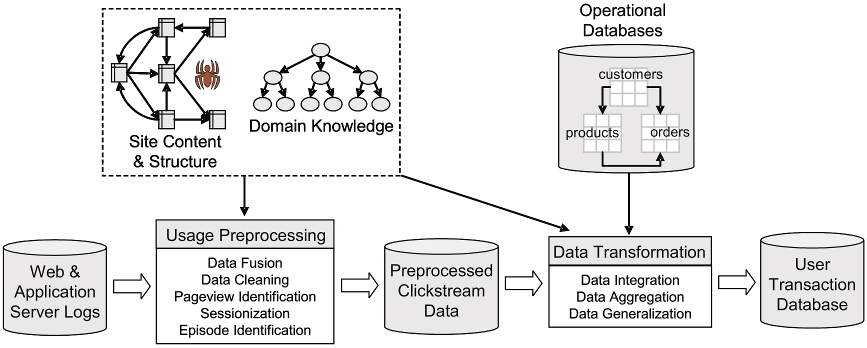

Fig. 3.1. Summary of the primary tasks and elements in usage data preprocessing.

Finally, the operational databases for the site may include additional information about user and items. Such data may include demographic information about registered users, user ratings on various objects such as products or movies, past purchase or visit histories of users, as well as other explicit or implicit representations of a user's interests. Product databases or content management systems may also include additional content descriptors and relational attributes that can be used as part of the representation of content information for items. Some of this data can be captured anonymously as long as there is the ability to distinguish among different users.

3.3.3 Data Preprocessing for Web Usage Mining

The goal of the preprocessing stage in Web usage mining is to transform the raw clickstream data into a set of user profiles (as described in the formal representation given in equation 3.1). From a navigational point of view each such profile captures a delimited sequence or a set of pageviews representing a user session. This sessionized data can be used as the input for a variety of data mining algorithms or further transformed and abstracted. Web usage data preprocessing presents a number of unique challenges which have led to a variety of algorithms and heuristic techniques for preprocessing tasks such as data fusion and cleaning, user and session identification, pageview identification [32]. The successful application of data mining techniques to Web usage data is highly dependent on the correct application of the preprocessing tasks.

Figure 3.1 provides a summary of the primary tasks and elements in usage data preprocessing. We provide a brief discussion of each of these elements below.

Data fusion refers to the merging of log files from several Web and application servers. This may require global synchronization across these servers. In the absence of shared embedded session ids, heuristic methods based on the "referrer" field in server log entries along with various sessionization and user identification methods (see below) can be used to perform the merging. This step is essential in "inter-site" Web usage mining where the analysis of user behavior is performed over the log files for multiple related Web sites [137].

Data cleaning involves tasks such as, removing extraneous references to embedded objects, style files, graphics, or sound files, and removing references due to spider navigations. The latter task can be performed by maintaining a list of known spiders, using heuristics, or using classification algorithms to build models of spider and Web robot navigations [135]. Also, not all client page requests are recorded in server access logs. Client-side or proxy-side caching can often result in missing references to those pages or objects that have been cached. Most of these missing references can be heuristically inferred through a process called *path completion* which relies on the knowledge of site structure and referrer information from server logs [32]. In the case of dynamically generated pages, form-based applications using the HTTP POST method result in all or part of the user input parameter not being appended to the URL accessed by the user, and thus not appear in server log entries (though, in the latter case, it is possible to re-capture the user input through packet sniffers on the server side).

Pageview identification is the process of aggregating a collection of objects or pages that should be considered an atomic unit for the purpose of analysis. This process is heavily dependent on the linkage structure of the site, as well as on the site contents. The level of abstraction captured in a pageview is also determined, in part, by the the underlying site domain knowledge and by the type of analysis required. In the simplest case, each HTML file has a one-to-one correlation with a pageview. In multi-framed sites, several files may make up a given pageview. In addition, it may be desirable to consider pageviews at a higher level of aggregation, where each pageview represents a collection of pages or objects, for examples pages related to the same concept category. In order to provide a flexible framework for a variety of data mining activities a number of attributes must be recorded with each pageview. These attributes include the pageview id (normally a URL uniquely representing the pageview), static pageview type (e.g., information page, product view, category view, or index page), and other metadata, such as content attributes (e.g., keywords or product attributes). In the context of our discussion, the pageviews represent the abstract items $i_j \in I$ (in equation 3.1) that are objects of personalization.

In Web usage mining it is necessary to distinguish between the activities of different users. In the absence of an authentication mechanism, a common approach to distinguishing among unique visitors is the use of client-side cookies. Not all sites, however, employ cookies, and due to privacy concerns, client-side cookies are sometimes disabled by users. IP addresses, alone, are not generally sufficient for mapping log entries onto the set of unique visitors. This is mainly due the proliferation of ISP proxy servers which assign rotating IP addresses to clients as they browse the Web. In such cases, it is possible to more accurately identify unique users through combinations of IP addresses and other information such as the user agents and referrers [32].

Assuming that unique user records can be identified, we refer to the sequence of logged activities belonging to the same user as the *user activity log*. *Sessionization* is the process of segmenting the user activity log of each user into sessions, each representing a single visit to the site. Web sites without the benefit of additional authentication information from users and without mechanisms such as embedded session ids must rely on heuristics methods for sessionization. The goal of a sessionization heuristic is the reconstruction, from the clickstream data, of the actual sequence of actions performed by one user during one visit to the site. Generally, sessionization heuristics fall into two basic categories: time-oriented or structure oriented. Time-oriented heuristics apply either global or local time-out estimates to distinguish between consecutive sessions, while structure-oriented heuristics use either the static site structure or the implicit linkage structure captured in the referrer fields of the server logs. Various heuristics for sessionization have been identified and studied [32]. A formal framework for measuring the effectiveness of such heuristics has been proposed [129], and the impact of different heuristics on various Web usage mining tasks has been analyzed in [12].

An *Episode* is a subset or subsequence of a session comprised of semantically or functionally related pageviews. Episode identification can be performed as a final step in preprocessing of the clickstream data in order to focus on the relevant subsets of pageviews in each user session. This task may require the automatic or semi-automatic classification of pageviews into different functional types or into concept classes according to a domain ontology or concept hierarchy. In highly dynamic sites, it may also be necessary to map pageviews within each session into "service-based" classes according to a concept hierarchy over the space of possible parameters passed to script or database queries [13]. For example, the episode may ignore the quantity and attributes of an item added to the shopping cart, and focus only on the action of adding the item to the cart.

The above preprocessing tasks ultimately result in a set of n pageviews, $P = \{p_1, p_2, \cdots, p_n\}$, and a set of v user transactions, $T = \{t_1, t_2, \cdots, t_v\}$, where each $t_i \in T$ is an l-length sequence of ordered pairs:

$$t = \left\langle (p_1^t, w(p_1^t)), (p_2^t, w(p_2^t)), \cdots, (p_l^t, w(p_l^t)) \right\rangle ,$$

where each $p_i^t = p_j$ for some $j \in \{1, \cdots, n\}$, and $w(p_i^t)$ is the weight associated with pageview p_i^t in the transaction t.

Each items $i_j \in I$ in the general framework of Section 3.3.1 (see Equation 3.1) can each represent a pageview. Note that a pageview in this context is not just a Web page, but as noted above, an abstraction which may represent a conceptual or functional entity in the application domain (e.g., a Web page, a product view or purchase, a task, or a content category). The notion of *user transaction* introduced above is meant to capture the activity of a user vis-a-vis these pageviews within the site during a particular session (thus, sometimes we refer to these transactions as sessions). The weights can be determined in a number of ways, in part based on the type of analysis or the intended personalization tasks. For example, in a standard collaborative filtering application, weights may be determined based on user ratings of items. In most Web usage mining tasks, the focus is generally on anonymous user navigational activity in which case the weights are either binary, representing the existence or non-existence of a pageview in the transaction; or a function of the duration of the pageview in the user's session.

Finally, one or more transactions or sessions associated with a given user u_k can be aggregated to form the final profile for that user resulting in user profile representation of Section 3.3.1, in which each item i_j is a pageview and the value of the interest function, $s_{u_k}(i_j)$, is determined as a function of the weight of the associated pageview, $w(p_j^t)$. If the profile is constructed from a single session, then it represents the short-term interests of that user during a single visit, while the aggregation of multiple sessions results in profiles that capture the user's long-term interests. The collection of these profiles will comprise the $m \times n$ matrix UP that can be used to perform various data mining tasks. For example, similarity computations can be performed among the profile vectors for clustering and kNN neighborhood formation tasks, or an association rule discovery algorithm can be applied (with pageviews as items) to find frequent itemsets of pageviews.

After the basic clickstream preprocessing steps, data from a variety of other sources must be integrated. The integration of content, structure and user data in various phases of the Web usage mining process may be essential in providing the ability to further analyze and reason about the discovered patterns. For example, the integration of semantic knowledge from the site content or semantic attributes of products can be used by personalization systems to provide more useful recommendations [33, 41, 59]. In e-commerce applications, the integration of both customer and product data (e.g., demographics, ratings, purchase histories) from operational databases with usage data can allow for the discovery of important business intelligence metrics such as customer conversion ratios and lifetime values [20, 61]. The use of structure data is necessary during preprocessing (for example in pageview identification, sessionization, and path completion). But, it can also be used to improve the results of model-based personalization techniques [90, 69].

One direct source of semantic knowledge that can be integrated into the mining process is the collection of content features associated with items or pageviews on a Web site. These features include keywords, phrases, category names, or specific attributes associated with items or products, such as price, brand, etc. Content preprocessing involves the extraction of relevant features from text and meta-data.

Extending the general framework of Section 3.3.1, each item i_j can be represented as a k- dimensional feature (or attribute) vector, where k is the total number of extracted features. Each dimension in a feature vector represents the corresponding feature weight associated with the item. Thus, the feature vector for an item i_j is given by:

$$i_j = \langle fw_j(f_1), fw_j(f_2), \cdots, fw_j(f_k) \rangle ,$$

where $fw_j(f_d)$, is the weight of the dth feature in $i_j \in I$, for $1 \leq d \leq k$. For features extracted from textual content of pages, the feature weight is usually the normalized tf.idf value for the term. In order to combine feature weights from meta-data (specified externally) and feature weights from the text content, proper normalization of those weights must be performed as part of preprocessing.

Further preprocessing on content features can be performed by applying text mining techniques. For example, classification of content features based on a concept hierarchy can be used to limit the discovered usage patterns to those containing pageviews about a certain subject or class of products. Performing clustering or association rule mining on the feature space can lead to composite features representing concept categories. In many Web sites it may be possible and beneficial to classify pageviews into functional categories representing identifiable "tasks" (such as completing an online loan application) [56]. The mapping of pageviews onto a set of concepts or tasks allows for the analysis of user sessions at different levels of abstraction according to a concept hierarchy or according to the types of activity performed by users [94, 37].

3.4 Pattern Discovery for Predictive Web User Modeling

As noted earlier, model-based collaborative techniques, including those used in the pattern discovery phase of Web usage mining, use a two stage process for recommendation generation. The first stage is carried out offline, where user behavioral data collected during previous interactions is mined and an explicit model generated for use in future online interactions. The second stage, is carried out in real-time as a new visitor begins an interaction with the Web site. Data from the current user session is scored using the models generated offline, and recommendations generated based on this scoring. The application of these models are generally computationally inexpensive compared to memory-based approaches such as traditional collaborative filtering, aiding scalability of the real-time component of the recommender system.

Model generation can be applied to explicitly and implicitly obtained user behavioral data. While the most commonly used implicit data is Web usage data, data pertaining to the structure and content are also often used. A number of data mining algorithms have been used for offline model building including Clustering, Classification, Association Rule Discovery, Sequential pattern Discovery, Markov models, and hidden (latent) variable models. In this section we briefly describe these approaches.

3.4.1 Personalization Approaches Based on Clustering

Clustering aims to divide a data set into groups or clusters where inter-cluster similarities are minimized while the similarities within each cluster are maximized. Generally speaking, clustering methods can be divided into three categories [47]:

- Partitioning methods, that create k partitions of a given data set, where each partition represents a cluster. The most widely used partitioning method is the k-means algorithm.
- Hierarchical methods either using a top-down approach (divisive) or a bottom-up approach (agglomerative) to create a hierarchy of clusters. Divisive methods start from the whole data set of items as a single cluster and recursively partition this data, while agglomerative methods start from individual items as clusters and iteratively combine smaller clusters.
- Model-based methods, that discover the best fit between data points given a mathematical model, usually specified as a probability distribution.

Various clustering algorithms have been used in standard collaborative filtering applications where interest scores for items are generally explicit ratings. Most of these approaches, however, generalize easily to the context of Web usage mining where the items are pageviews and interest scores are normally based on the characteristics of user behavior (such as pageview duration). In this context, clustering is usually used in one of two ways: to cluster users or to cluster items. In user-based clustering, users are grouped together based on the similarity of their user profiles in matrix UP (see Section 3.2.2). In item based clustering, items are clustered based on the similarity of the interest scores for these items across all users, or based on similarity of their content features or attributes. Some of the past methods used in this context include partitioning algorithms such as, K-means for item and user-based clustering [140], ROCK [95] for item-based clustering, agglomerative hierarchical clustering [95] for item-based clustering, divisive hierarchical clustering for user-based and item-based clustering [63], mixture resolving algorithms such as EM [34] to cluster users based on their item ratings [19] and Gibbs Sampling [19].

As noted earlier, the primary motivation behind the use of clustering (and more generally, model-based algorithms) in collaborative filtering and Web usage mining is to improve the efficiency and scalability of the real-time personalization tasks. For example, both user-based clustering and item-based clustering have been used as an integrated part of a Web personalization framework based on Web usage mining [88, 82]. Motivated by reducing the sparseness of the rating matrix, O'Connor and Herlocker proposed the use of item clustering as a means for reducing the dimensionality of the rating matrix [95]. Column vectors from the ratings matrix were clustered based on their similarity, measured using Pearson's correlation coefficient, in user ratings. The clustering resulted in the partitioning of the universe of items and each partition was treated as a separate, smaller ratings matrix. Predictions were then made by using traditional collaborative filtering algorithms independently on each of the ratings matrices. While some statistical methods such as sampling, as well as clustering, can mitigate the online computational complexity of collaborative filtering, these methods often result in reduced recommendation quality [72]. However, in the context of Web usage mining it has been shown that proper preprocessing of the usage data can help the clustering approach achieve prediction accuracy in par with standard k-nearest-neighbor approach [84].

A typical user-based clustering starts with the matrix UP of user profiles and partitions this multi-dimensional space into k groups of profiles (or Web transactions) that are close to each other based on a measure of distance or similarity among the vectors (such as the Pearson's correlation coefficient). Clusters obtained in this way can represent user or visitor segments based on their common navigational behavior or interest shown in various items. The discovered user segments are then employed in the user-based neighborhood formation task, rather than individual profiles [88].

In order to determine similarity between a target user and a user segment, the centroid vector corresponding to each cluster is computed and used as the aggregate representation of the user segment. For each cluster C_k, the centroid vector $\mathbf{v_k}$ is computed as: $\mathbf{v_k} = \frac{1}{|C_k|} \sum \mathbf{u_n}$, where $\mathbf{u_n}$ is the vector in UP for a user profile $u_n \in C_k$.

To make a recommendation for a target user u and target item i, a neighborhood of user segments that have a ratings or interest scores for i and whose aggregate profile v_k is most similar to u are selected. This neighborhood represents the set of user segments of which the target user is most likely to be a member. Given that the aggregate profile of a user segment contains the average interest scores for each item within the segment, a prediction can be made for item i in the same manner as in standard collaborative filtering using k-nearest-neighbor [49]. For example, the predicted score for a target item i and target user u can be computed as:

$$p_{u,i} = \bar{s}_u + \frac{\sum\limits_{v \in V} sim(u,v)(s_v(i) - \bar{s}_v)}{\sum\limits_{v \in V} |sim(u,v)|} \tag{3.2}$$

where V is the set of k most similar segments; $s_v(i)$ is the weight (average interest score) of i in the neighbor segment v; $\bar{s}_u$ and $\bar{s}_v$ are the average interest scores over all items for user u and segment v, respectively; and $sim(u,v)$ is the similarity between u and segment v.

As noted above, many other approaches based on user-based or item-based clustering have been used in the context of personalization based on Web usage mining. For example, an algorithm called *PageGather* has been used to discover significant groups of pages based on user access patterns [105, 106]. This algorithm uses, as its basis, clustering of pages using the graph-based Clique (complete link) clustering technique. The resulting clusters are used to automatically synthesize alternative static index pages for a site, each reflecting possible interests of one user segment. In PageGather an edge is added between two nodes (pages) if the corresponding pages co-occur in more than a certain number of sessions. Clusters are then generated by finding connected components or cliques within this graph. A new index page for the Web site is created from each cluster with hyperlinks to all the pages in that cluster. One advantage of this approach is that it creates overlapping clusters. However, the problem of finding (maximal cliques) in a graph is generally not computationally feasible for large graphs (i.e., for sites with many pages).

Because in Web usage mining it is often desirable to group users into multiple categories, a number of approaches based on fuzzy clustering have been explored. For example, a fuzzy clustering approach is proposed in [57] for clustering user sessions. The Web site hyperlink structure is used as a bias in computing the similarity between sessions by taking into account the relative position of pages within sessions in the site tree. The clustering algorithm used are variants of the Fuzzy C-means (FCM) clustering method [14] which allows one piece of data to belong to two or more clusters. Similarly, Nasraoui et al. [91] proposed an unsupervised relational clustering algorithm based on the competitive agglomeration algorithm to discover aggregate user models. This approach was later extended with fuzzy clustering algorithms such as Relational Fuzzy C-Maximal Density Estimator (RFC-MDE) and Fuzzy C Medoids algorithm (FCMdd), both of which are again based on FCM [92].

Most distance-based approaches, such as those described above, do not consider the sequential ordering inherent in Web transactions. Clustering can also be applied to Web transactions viewed as sequences rather than as vectors. For distance-based clustering algorithms to handle this type of data, a measure of distance (or similarity) which takes ordering among items into account is necessary. Some clustering approaches have integrated sequential representation of user session data and defined pairwise distance functions between user sessions [7, 132]. For example in [7] a graph-based algorithm was introduced to cluster Web transactions based on a function of longest common subsequences. The novel similarity metric used for clustering takes into account both the time spent on pages as well as a significance weight assigned to pages.

Model-based clustering algorithms also have the advantage of not requiring an explicit distance measure. Therefore, despite their potential high computational cost, they are often applicable in a more general context. For example, Cadez et al. [23] used the Expectation-Maximization (EM) algorithm [34] on a mixture of Markov models for clustering user sessions. Each Markov model in this framework captures the behavior of a particular subgroup of users according to their navigational activities. The algorithm was used as the basis of a tool called WebCANVAS, designed to visualize user navigation paths in each cluster. The EM algorithm was also used by Anderson et al. [5] for discovering predictive Web usage models. The user navigation sessions were assumed to belong to one or more clusters, and the EM algorithm was used to compute the model parameters for each cluster. The probability of visiting a certain page is estimated by calculating its conditional probability for each cluster. The standard Markovian assumption is made that occurrences of pages in a particular session are independent given the cluster, resulting in a Naive Bayesiam mixture model.

Another approach that has been used effectively in item-based clustering of Web usage data is *Association Rule Hypergraph partitioning* (ARHP) [46]. In this approach, first association rule mining (see Section 3.4.2) is used to discover a set E of frequent itemsets among the items (pageviews) in the set of all items I. These itemsets are used as hyperedges to form a hypergraph $H = \langle V, E \rangle$, where $V \subseteq I$. A hypergraph is an extension of a graph in the sense that each hyperedge can connect more than two vertices. The weights associated with each hyperedge can be computed based on a variety of criteria such as the average confidence of the association rules involving the items in the frequent itemset. The hypergraph H is recursively partitioned until a stopping criterion for each partition is reached resulting in a set of clusters. A connectivity measure for a vertex (a pageview appearing in the frequent itemset) with respect to a cluster is defined based on the weights of hyperedges connecting it to other vertices in the cluster. The vertices with connectivity measure greater than a given threshold value are considered to belong to the partition, and the remaining vertices are dropped from the partition. This approach has also been used in the context of Web personalization [88].

3.4.2 Personalization Using Association Discovery

Association rule discovery techniques, such as the Apriori algorithm [3], were initially developed as techniques for mining supermarket basket data but have since been used in various domains including Web mining [10]. Association rule discovery on usage data results in finding groups of items or pages that are commonly accessed or purchased together. This, in turn, enables Web sites to organize the site content more efficiently, or to provide effective cross-sale product recommendations. For example, a high-confidence rule such as

$$\{\text{special-offers/, /products/software/}\} \Rightarrow \{\text{shopping-cart/}\}$$

might provide some indication that a promotional campaign on software products is positively affecting online sales. Such rules can also be used to optimize the structure of the site. For example, if a site does not provide direct linkage between two pages A and B, the discovery of a rule $\{\text{A}\} \Rightarrow \{\text{B}\}$ would indicate that providing a direct hyperlink might aid users in finding the intended information.

The discovery of association rules from transaction data consists of two main parts: the discovery of frequent itemsets (i.e., itemsets which satisfy a minimum *support* threshold) and the discovery of association rules from these frequent itemsets which satisfy a minimum *confidence* threshold.

Given a set of transactions T and a set $I = \{I_1, I_2, \ldots, I_k\}$ of itemsets over T. The *support* of an itemset $I_i \in I$ is defined as

$$\sigma(I_i) = \frac{|\{t \in T : I_i \subseteq t\}|}{|T|}$$

An association rule, r, is an expression of the form $X \Rightarrow Y$ (σ_r, α_r), where X and Y are itemsets, $\sigma_r = \sigma(X \cup Y)$ is the support of $X \cup Y$ representing the probability that X and Y occur together in a transaction. The confidence for the rule r, α_r, is given by $\sigma(X \cup Y)/\sigma(X)$ and represents the conditional probability that Y occurs in a transaction given that X has occurred in that transaction. Additional metrics have been proposed in literature that aim to quantify the interestingness of a rule [97, 123, 136], however support and confidence as these are the most commonly used metrics when using association and sequence based approaches to personalization.

Although not as widely used as clustering for Web personalization, the results of association rule mining on the user profile and items space can result in models that, in conjunction with the activity or profile of a target user, can be used for recommendation generation [39, 71, 83, 115]. For example, in the collaborative filtering context, Sarwar, et al. [115], used association rules in the context of a *top-N* recommender systems for e-commerce. The preferences of the target user are matched against the items in the antecedent X of each rule, and the items on the right hand side of the matching rules are sorted according to the confidence values. Then the top N ranked items from this list are recommended to the target user.

One problem for association rule recommendation systems is that a system cannot give any recommendations when the dataset is sparse (which is often the case in Web usage mining and collaborative filtering applications), and hence larger itemsets often do not meet the minimum support constraint. Sarwar, et al. [115] rely on some standard dimensionality reduction techniques to alleviate this problem. Fu et al. [39] propose two potential solutions to this problem. The first solution is to rank all discovered rules calculated by the degree of intersection between the left-hand-side of rule and a user's active session and then to generate the top k recommendations. The second solution is to utilize collaborative filtering: the system finds "close neighbors" who have similar interest to a target user and makes recommendations based on the close neighbor's history.

In [71] a collaborative recommendation system was presented using association rules. The proposed mining algorithm finds an appropriate number of rules for each target user by automatically selecting the minimum support. The recommendation engine generates association rules for each user, among both users and items. If a user minimum support is greater than a threshold, the system generates recommendations based on user association, else it uses item associations.

Because it is difficult to find matching rule antecedents with a full user profile (e.g., a full session in Web usage mining context), association-based recommendation algorithms typically use a sliding window w over the target user's active profile or session. The size of this window is iteratively decreased until an exact match with the antecedent of a rule is found. A problem with the naive approach to this algorithm is that it requires repeated search through the rule-base. However, efficient trie-based data structures can be used to store the discovered itemsets and allow for efficient generation of recommendations without the need to generate all association rules from frequent itemsets. Such data structures are commonly used for string or sequence searching applications. In the context of association rule mining, the frequent itemsets are stored in a directed acyclic graph. The Frequent Itemset Graph is an extension of the lexicographic tree used in the "tree projection algorithm" [1]. The graph is organized into levels from 0 to k, where k is the maximum size among all frequent itemsets. Each node at depth d in the graph corresponds to an itemset, I, of size d and is linked to itemsets of size $d+1$ that contain I at level $d+1$. The single root node at level 0 corresponds to the empty itemset. To be able to match different orderings of an active session with frequent itemsets, all itemsets are sorted in lexicographic order before being inserted into the graph. The user's active session is also sorted in the same manner before matching with patterns.

Using this general framework, the recommendation engine matches the current user session window with the previously discovered frequent itemsets to find candidate items (pages) for recommendation. Given an active session window w and a group of frequent itemsets, the algorithm considers all the frequent itemsets of size $|w|+1$ containing the current session window by performing a depth-first search of the Frequent Itemset Graph is performed to level $|w|$. The recommendation value of each candidate is based on the confidence of the corresponding association rule whose consequent is the singleton containing the page to be recommended. If a match is found, then the children of the matching node n containing w are used to generate candidate recommendations. The details of this general recommendation algorithm [83] are given in Figure 3.2.

Association rules have also been used in conjunction with other data mining algorithms, such as clustering, in personalization based on Web usage mining (as well as other applications). In Section 3.4.1, we already described the item-based clustering approach used in [88] in which frequent itemsets of pageviews are organized in an *Association Rule Hypergraph*, and the resulting hypergraph is partitioned into pageview clusters.

Another approach that combines clustering and association rule mining is a two-level model-based collaborative filtering technique described in [133]. In the first level, a fuzzy C-Means clustering algorithm called Relational Fuzzy Subtractive Clustering (RFSC) is used to cluster the user sessions. Then the clusters are *defuzzified* by assigning the sessions to a cluster to with highest membership. This defuzzification process removes the noise and reveals the real structure in the data. In the second level, single-consequent association rules are discovered from within each cluster. For an active profile (a session) of a target user, the algorithm first finds the nearest cluster prototype, and then matches the profile with the antecedent of each rule in that cluster to find the matching score for each rule. The matching scores are weighted with the confidence of each rule to obtain the complete recommendation score of the item (page) appearing is the consequent of the rule.

```
Input:  an active session window $w = \{p_1, p_2, \ldots, p_n\}$ in lexicographic order
          Minimum confidence threshold $\alpha$
Output: Recommendation set $REC$

$REC = \emptyset$
$Node = root$; $depth = 0$;
repeat
    $depth$++;
    if $Node.children \neq \emptyset$
                and $\exists X \in Node.children$ with $X.itemset \subseteq \{p_1, \ldots, p_{depth}\}$
        then $Node = X$;
    else
        $Node$ = NULL; Break;
until $depth > |w|$
if $Node \neq$ NULL then
    for each node $N \in Node.children$ do
        let $c = N.support/Node.support$;
        if $c \geq \alpha$
            $p = (N.itemset - w)$
            $p.rec\_score = c$
            $REC = REC \cup \{p\}$
        end if
    end for
end if
```

Fig. 3.2. Recommendation Algorithm Based on Association Rules

3.4.3 Personalization Using Sequential Modeling

As with association rule discovery, Sequence rule discovery techniques [4] were also initially developed as techniques for mining supermarket basket data. The key difference between these algorithms is that while association rule discovery algorithms do not take into account the order in which items have been accessed, sequential pattern discovery algorithms do consider the order when discovering frequently occurring itemsets. Hence, given a user transaction $\{i_1, i_2, i_3\}$, the transaction supports the association rules $i_1 \Rightarrow i_2$ and $i_2 \Rightarrow i_1$ but not the sequential pattern $i_2 \Rightarrow i_1$.

When discovering sequential patterns from Web logs, two types of sequences are identified: Contiguous or Closed Sequences and Open Sequences [10]. Contiguous sequences require that items appearing in a sequence rule appear contiguously in transactions that support the sequence. Hence the contiguous sequence pattern $i_1, i_2 \Rightarrow i_3$ is satisfied by the transaction $\{i_1, i_2, i_3\}$ but not by the transaction $\{i_1, i_2, i_4, i_3\}$, as i_4 appears in the transaction between the items appearing in the sequence pattern. On the other hand, both transactions support the rule if it were an open sequence rule.

Given a transaction set T and a set $S = \{S_1, S_2, \ldots, S_n\}$ of frequent (contiguous) sequential patterns over T, the support of each S_i is defined as follows:

$$\sigma(S_i) = \frac{|\{t \in T : S_i \text{ is (contiguous) subsequence of } t\}|}{|T|}$$

The confidence of the rule $X \Rightarrow Y$, where X and Y are (contiguous) sequential patterns, is defined as

$$\alpha(X \Rightarrow Y) = \frac{\sigma(X \circ Y)}{\sigma(X)},$$

where $\circ$ denotes the concatenation operator. The Apriori algorithm used in association rule mining can also be adopted to discover open and contiguous sequential patterns. This is normally accomplished by changing the definition of support to be based on the frequency of occurrences of subsequences of items rather than subsets of items [4].

In the context of Web usage mining, contiguous sequential patterns can be used to capture *frequent navigational paths* among user trails [128, 119]. In contrast, items appearing in open sequential patterns, while preserving the underlying ordering, need not be adjacent, and thus they represent more general navigational patterns within the site. Frequent item sets, discovered as part of association rule mining, represent the least restrictive type of navigational patterns, since they focus on the presence of items rather than the order in which they occur within user session.

An approach for efficiently representing contiguous navigational sequences is to insert each sequence into a trie structure. A well-known example of this approach is the notion of *aggregate tree* introduced as part of the WUM (Web Utilization Miner) system [128]. The aggregation service of WUM extracts the transactions from a collection of Web logs, transforms them into sequences, and merges those sequences with the same prefix into the aggregate tree (a trie structure). Each node in the tree represents a navigational subsequence from the root (an empty node) to a page and is annotated by the frequency of occurrences of that subsequence in the transaction data (and possibly other information such as markers to distinguish among repeat occurences of the corresponding page in the subsequence). WUM uses a powerful SQL-like mining query language, called MINT, to discover generalized navigational patterns from this trie structure. MINT includes mechanism to specify sophisticated constraints on pattern templates, such as wildcards with user-specified boundaries, as well as other statistical thresholds such as support and confidence.

It is also possible to insert frequent sequences (after or during sequential pattern mining) into a trie structure [104, 86]. In the context of personalization, sequential patterns are typically stored in a single trie structure with each node representing an item and the root representing the empty sequence. Recommendation generation can be achieved in $O(s)$ by traversing the tree, where s is the length of the current user transaction deemed to be useful in recommending the next set of items. Mobasher et al. [86] use a fixed size sliding window, w, over the current transaction for recommendation generation. Hence the maximum depth of the tree required to be generated is $|w| + 1$. The size of the trees generated during the offline mining can be controlled by setting different minimum support and confidence thresholds. Thus, a similar general algorithm used in Section 3.4.2 (see Figure 3.2 for generating recommendations from frequent item-sets, can easily be adopted in the context of open and contiguous sequential patterns.

An empirical evaluation of association and sequential pattern based recommendation showed that site characteristics such as site topology and degree of connectivity can have a significant impact on the usefulness of sequential patterns over non-sequential (association) patterns [90]. Additionally, it has also been shown that contiguous sequential patterns are particularly restrictive and hence are more valuable in page prefetching applications (were the intent is to predict the immediate next page to be accessed) rather than in the more general context of recommendation generation [86].

Another type of approach for sequential modeling is based on stochastic methods that from the sequences of pageviews in user sessions learn probabilistic models that can used for predicting subsequent visits. One such approach is to model the navigational activity in the Web site as a Markov chain. A Markov model is represented by the 3-tuple $\langle A, S, T \rangle$ where A is a set of possible actions, S is the set of n states for which the model is built and T is the Transition Probability Matrix that stores the probability of performing an action $a \in A$ when the process is in a state $s \in S$. Specifically, $T = [p_{i,j}]_{n \times n}$, where $p_{i,j}$ represents the probability of a transition from state s_i to state s_j. The *order* of the Markov model corresponds to the number of prior events used in predicting a future event. So, a kth-order Markov model predicts the probability of the next event by looking at the past k events. Given a set of all paths R, the probability of reaching a state s_j from a state s_i via a (non-cyclic) path $r \in R$ is given by: $p(r) = \prod p_{k,k+1}$, where k ranges from i to $j-1$. The probability of reaching s_j from s_i is the sum over all paths: $p(j|i) = \sum\limits_{r \in R} p(r)$.

In the context of recommendation systems, A is the set of items and S is the visitor's navigation history, defined as a k-tuple of items visited, where k is the order of the Markov model. In Web usage analysis, they have been proposed as the underlying modeling machinery for Web prefetching applications or to minimize system latencies [36, 98, 107, 113]. Such systems are designed to predict the $next$ user action based on a user's previous surfing behavior. On the other hand, Markov models can also be used to discover high-probability user navigational paths in a Web site. For example, Borges and Levene [17] modeled user sessions as a hypertext probabilistic grammar (or alternatively, an absorbing Markov chain) whose higher probability paths correspond to the user's preferred trails. An algorithm is provided to efficiently mine such trails from the model.

As the order of the Markov model increases, so does the size of the state space, S. On the other hand the coverage of that space reduces, leading to an inaccurate transition probability matrix. To counter the reduction in coverage, various Markov models of differing order can be trained and used to make predictions. The resulting model is referred to as the *All-Kth-Order* Markov model [107]. The downside of using the All-Kth-Order Markov model is the large number of states. Selective Markov models that only store some of the states within the model have been proposed as a solution to this problem [36]. A post pruning approach is used to prune out states that cannot be expected to be accurate predictors. Three pruning approaches based on the support, confidence and estimated error were proposed.

Rather than pruning states as a post process, sequence rule discovery and association rule discovery algorithms actively prune the state space during the discovery process using support. A further post pruning, based on confidence of the discovered rules, is also carried out. Hence the Selective Markov model is analogous to sequence rule discovery algorithms. Note however that the actual pruning process based on confidence proposed by Deshpande and Karypis [36] is not the same as that carried out during sequence rule discovery. Evaluation of Selective Markov models has shown that up to 90% of states can be pruned without a reduction in accuracy.

Other types of stochastic methods include various mixture models [23, 100, 141]that have been used to model navigational patterns. We have already discussed some of these approaches and their use in clustering approach to personalization (see Section 3.4.1). Recent work in this area has shown that mixture models are able to capture more complex, dynamic user behavior. This is in part because the observation data (i.e., the user-item space) in some applications (such as large and very dynamic Web sites) may be too complicated to be modeled by basic probability distributions such as a normal or a multinomial distribution. In particular, each user may exhibit different "types" of behavior corresponding to different tasks, and common behaviors may each be reflected in a different distribution within the data.

The general idea behind mixture models (such as a mixture of Markov models) is as follow. We assume there exist k types of user behavior (or k user clusters) within the data, and each user session is assumed to be generated via a generative process which models the probability distributions of observed variables and hidden variables. First, a user cluster is chosen with some probability; then the user session is generated from a Markov model with parameters specific to that user cluster. Next, the probabilities associated with the user cluster are estimated (usually via the EM [34] algorithm), as well as the parameters of each mixture component. Mixture-based user models can provide a great deal of flexibility. For example, a mixture of first-order Markov models [23] can not only probabilistically cluster user sessions based on similarities in navigation behavior, but also characterize each type of user behavior using a first-order Markov model, thus capturing popular navigation paths or characteristics of each user cluster. New user sessions can be easily fit into the model, and dynamic predictions or recommendations can be generated based on the probablitiy of association of the target user profile to various clusters. However, mixture models tend to fall victim to overfitting problems, largely due to their naive data generation assumptions. A more detailed discussion of mixture models is provided in the next section.

3.4.4 Approaches Based on Latent Variable Models

Latent variable models (LVMs) [8, 38] have recently become popular modeling approaches in data mining related fields and, particularly, in Web usage mining. By introducing latent variables as hidden factors underlying observation data, LVMs use probabilistic approaches to effectively discover the structural and semantic relationships within the data.

Two commonly used latent variable models are *Factor Analysis* (FA) models and *Finite Mixture Models* (FMM). By learning a low dimensional latent space from a high dimensional observation space, FA models aim at summarizing and explaining the complex dependency relationship among the observation data. Factor analysis models have a long history of successful applications in many domains, including in patterns recognition. Only recently, however, they have been effectively used the context of collaborative filtering [25] and personalization based on Web usage mining [144]. However, in the context of Web user modeling, FA models, as in most clustering approaches, generally ignore the sequential information conveyed in user sessions.

FMMs, on the other hand, use a finite number of components to model the observation data. Theoretically, the component models can be any probability distribution. In FMMs, one generally assumes the existence of k components (each component is a probability distribution) that account for all the observation data. Each single observation (e.g., a user's rating for an item, or a pageview in a user session) is assumed to be generated by the following process: first, a component with a certain probability is chosen, and then the chosen component is used to generate the observations. As noted earlier, the EM algorithm is usually used to fit the model and estimate the parameters associated with each component.

For example, a mixture of multinomial models is proposed in [24] to analyze the e-commerce transaction data. Transactions generated by individual users are probabilistically clustered into k groups, where each cluster is modeled by a multinomial distribution. The experiments show that the mixture model distinctly outperforms non-mixture techniques (a single multinomial model) in predicting out-of-sample individual behavior. As we noted in Section 3.4.3, in [23], a mixture of first-order Markov models was proposed to cluster Web users in which each component was modeled as a first-order Markov model. It was formally shown that the mixture of first-order Markov models is not first-order Markov model, and that it can model much more complex user behavior. A mixture of hidden Markov models [141] is also proposed for modeling clickstreams of Web surfers. In addition to user-based clustering, this approach can also be used for automatically page categorization.

Mixture models tend to have their own shortcomings. From the data generation perspective, each individual observation (such as a user session) is generated from one and only one component model. The probability assignment to each component only measures the uncertainty about this assignment. This assumption limits this model's ability of capturing complex user behavior, and more seriously, may result in overfitting problems [110].

Probabilistic Latent Semantic Analysis (PLSA) [52] provides a reasonable solution to some of these problems. PLSA adopts a totally different data generation idea. In the context of Web user navigation, each observation (a user visiting a page) is assumed to be generated as follows. First a user is selected with a certain probability. Next, conditioned on the selected user, a hidden variable is selected. Finally, the page to visit is slected conditioned on the chosen hidden variable. Since each user usually visits multiple pages, this data generation process ensures that each user is explicitly associated with multiple hidden variables, thus eliminating the overfitting problem associated with above mixture models. The PLSA model also uses the EM algorithm to estimate the parameters which probabilistically characterize the hidden variables underlying the co-occurrence observation data, and measure the relationship among hidden variables and observed variables. Due to its great flexibility, the PLSA model has been successfully used in a variety of application domains, including information retrieval [51], text learning [18, 60], and co-citation analysis [29, 30].

Another type of hidden variable model is the *Latent Dirichlet Allocation* (LDA) model [16]. The LDA model uses two levels of hidden variables. Each observation is assumed to be a multinomial distribution of k hidden variables, and each multinomial distribution is further constrained by a global variable with dirichlet distribution. The two levels of hidden variables are used to ensure that training observations and non-training observations can be generated via the same process. A side effect of having two levels of hidden variables is that exact inference for the LDA model is not feasible. Methods such as Variational Bayes [16], Markov Chain Monte Carlo [44] and Expectation-Propagation (EP) [78] are proposed to learn the model. Recently, the LDA model has been used in text mining [16], author-topic analysis [131], and collaborative filtering [73]. Although PLSA and LDA seem to be quite different in terms of parameter learning, research has shown that they are essentially equivalent in terms of modeling method, and PLSA is just a *Maximum A Posterior* (MAP) estimation of LDA model [42]. LDA introduces an extra set of hidden variables and is able to naturally fit in new data. However this also makes the learning of the LDA model computationally more expensive than PLSA.

In order to see how a hidden variable modeling approach, such as PLSA, can be used in the context of personalization, we provide more detail on a general recommendation algorithm, based on PLSA, that can be adopted both in the context of standard collaborative filtering [53], as well as, in Web usage mining [55, 56].

As in the approaches based on clustering, the PLSA-based approach begins with the discovery of user segments with similar behavior: Given a set of n user profiles, $UP = \{u_1, u_2, \cdots, u_n\}$, and a set of m items, $I = \{i_1, i_2, \cdots, i_m\}$ the PLSA model associates an unobserved factor variable $Z = \{z_1, z_2, \cdots, z_l\}$ with observations in the data. Each observation corresponds to the interest score $s_{u_k}(i_j)$ for an item i_j in the user profile for a user u_k (i.e., a rating or a weight associated with a pageview).

For a target user u and a target item i, the following joint probability can be defined:

$$Pr(u,i) = \sum_{k=1}^{l} Pr(z_k) \bullet Pr(u|z_k) \bullet Pr(i|z_k)$$

In order to explain the observations in (UP, I), we need to estimate the parameters $Pr(z_k)$, $Pr(u|z_k)$, and $Pr(i|z_k)$, while maximizing the following likelihood $L(UP, I)$ of the observation data:

$$L(UP,I) = \sum_{u \in UP} \sum_{i \in I} s_u(i) \bullet \log Pr(u,i)$$

where $s_u(i)$ is the interest score (e.g., rating) of user u for item i.

The Expectation-Maximization (EM) algorithm [34] is used to perform maximum likelihood parameter estimation. Based on initial values of $Pr(z_k)$, $Pr(u|z_k)$, and $Pr(i|z_k)$, the algorithm alternates between an expectation step and maximization step. In the expectation step, posterior probabilities are computed for latent variables based on current estimates, and in the maximization step, Lagrange multipliers [52] are used to obtain the re-estimated parameters. Iterating the expectation and maximization steps monotonically increases the total likelihood of the observed data $L(UP, I)$, until a locally optimal solution is reached.

Next the segments of user profiles that have similar underlying interests are identified. For each latent variable z_k, a user segment C_k is created and all user profiles having probability $Pr(u|z_k)$ exceeding a certain threshold μ are selected. If a user profile's probability does not exceed the threshold for any latent variable, it is associated with the user segment of highest probability. Thus, every user profile will be associated with at least one user segment, but may be associated with multiple segments. This allows authoritative users to have broader influence over predictions, without adversely affecting coverage in sparse rating data.

For each user segment C_k, the associated user profiles are aggregated into a weighted profile vector $\mathbf{v_k}$, computed as the mean vector or centroid of all $u_i \in C_k$. This the aggregate profile for a user segment to be represented in the original n-dimentioal space of items. To make a recommendation for a target user u and target item i, a neighborhood of user segments is selected that have defined interest scores for i and whose aggregate profile v_k is most similar to u. This neighborhood represents the set of user segments of which the target user is most likely to be a member, based on a measure of similarity (such as the Pearson's correlation coefficient which is usually used with rating data). A prediction for item i can now be derived using equation 3.2, used earlier in Section 3.4.1 in the context of the clustering approach.

3.4.5 Hybrid Models for Web Personalization

Pure usage-based approaches to personalization have some important drawbacks. The recommendation process relies on the existing user transactions or rating data, thus items or pages added to a site recently cannot be recommended. This is commonly referred to as the "new item problem". Furthermore, because such systems do not take into account the semantic or structural knowledge inherent in the underlying domain, they generally lack the ability to recommend complex objects or concepts based in their semantic attributes or based on other information channels available in the particular application domain. This limitation also hampers the ability of these systems to explain or reason about the discovered user models or recommendations.

In traditional collaborative filtering a number of hybrid approaches have been proposed. The most common form of hybrid recommender combines content-based and collaborative filtering [28, 75]. Other approaches have also incorporated other information sources such as user demographics [101, 127]. A detailed examination of different approaches to create hybrid recommender systems is presented in [21] (see also Chapter 12 of this book [22]). In the following we focus primarily on the data mining approaches to personalization, and particularly those based on Web usage mining, in which various information channels have been integrated in the knowledge discovery and recommendation generation processes.

Integration of Content Features with Usage-Based Models. A common approach to resolving the "new item problem" is to integrate content characteristics of pages with the user-based data (i.e., navigational or rating data). Generally, in these approaches, keywords are extracted from the content on the Web site and are used to either index pages by content or classify pages into various content categories. In Web personalization, this approach would allow the system to recommend pages to a user, not only based on similar users, but also (or alternatively) based on the content similarity of these pages to the pages user has already visited. The semantic information extracted as keyword-based features can be leveraged at various steps in the knowledge discovery process, namely in the preprocessing phase, in the mining phase, or during the post-processing of the discovered patterns.

A direct approach for the integration of content and usage data for Web personalization is to transform each user profile in UP (see Section 3.3.1), into a "content enhanced" profile containing the semantic features of the underlying items. This process, performed as part of data preprocessing, involves mapping each item or page in a user profile to one or more content features extracted from items, or a set of concepts (for example, from an externally available concept hierarchy). The the range of this mapping can be the full feature space or the concept space obtained as described above. Conceptually, the transformation can be viewed as the multiplication of the user-item matrix UP by the item-feature or an item-concept matrix. The result is a new matrix $UF = \{t'_1, t'_2, \ldots, t'_m\}$, where each t'_i is a k-dimensional vector over the feature (or concept) space. Thus, a user profile can be represented as a concept vector, reflecting that user's interests in particular concepts or topics. A variety of data mining algorithms can then be applied to this transformed user data.

For example, in [94], usage mining is enhanced by mapping user navigational data to concepts in an ontology underlying a particular Web site. The semantic annotation of the Web content is assumed to have been performed a priori. In order to mine interesting patterns, first user transactions are semantically enriched with concept labels, and then the transformed transaction space is mined to extract patterns reflecting users' changing interest in terms of concepts.

Following a similar approach, in [37] Web usage logs are enriched with semantics derived from the content features extracted from of the Web site's pages. The extraction of the keywords that describe each Web page is performed using standard information retrieval based techniques. These keywords are then mapped to the categories of a predefined concept hierarchy. The enhanced Web logs are then used as input to the Web mining process. The output consists of patterns representing users' navigational behavior in the form of clusters or association rules. This set of patterns is then used as the recommendation basis for each user or group of users, resulting in a broader yet semantically focused set of recommendations.

Haase et al. create semantic user profiles from usage and content information to provide personalized access to bibliographic information on a Peer-to-Peer bibliographic network [45]. The semantic user profile consists of the expertise, recent queries, recent relevant instances and a set of weights for the similarity function.

The integration of content features with user models can also be performed during or after the mining phase. In this case, patterns are discovered independently from the user profile data and the content data, and then combined in the recommendation generation process. For example, the results of user-based clustering can be combined with "content profiles" derived from the clustering of content features in pages [87]. The feature clustering is accomplished by applying a clustering algorithm to the transpose of the item-feature matrix *UF*, described above. This approach treats each feature as a vector over the space of items. Thus the centroid of a feature cluster can be viewed as a set (or vector) of items with associated weights. This representation is similar to that of aggregate models described in Section 3.4.1, however, in this case the weight of an item in the aggregate model represents the prominence of the features of the item that are associated with the corresponding cluster. The combined set of aggregate content and usage models can then be used seamlessly to generate recommendations.

Such approaches have also been useful in the context of e-commerce recommender systems. For example, Niu et al. [93] build customer profiles based on a product hierarchy in order to learn customer preferences. Ghani and Fano [41] proposed a recommender system based on a custom-built knowledge base of product semantics. The focus is on generating "soft" attributes from the online marketing text, describing the products browsed, and using them to generate cross category recommendations.

One type of integration approach is that for each user, one builds a local prediction model using algorithms such as naïve Bayes or k-Nearest Neighbor based on content data. Then all the individual models are integrated to form a global model via approaches such as linear combinations or probabilistic combination. An example of such integration is shown in [142], where a combined recommendation model is proposed. For each user, a probabilistic SVM (Support Vector Machine) model is built only based on the content information of this user's interested items. These individual models enable the system to make predictions for unvisited/unrated items only based on the content information of these items. Then all the individual models were combined under a hierarchical Bayesian framework, and the final prediction is the result of combining predictions from all individual models.

Finally, a number of approaches have attempted to integrate content and usage data based on hidden variable and mixture models (see Section 3.4.4). For example, in [108], an extension of the PLSA model was used to handle three-way co-occurrence data including users, items, and content features. The proposed extended PLSA model is used to discover the hidden relationships among users, items and attributes. A limitation of this approach is that, since the three-way observation data does not exist, and is generated subjectively from other observation data, it may not be consistent with the original navigational or content data.

Jin et al. [54] proposed a more robust approach based on hidden variable models in which users' navigational data and the content features associated with items are seamlessly integrated using a maximum entropy approach [111]. The goal of a maximum entropy model is to find a probability distribution which satisfies all the constraints in the observed data while maintaining maximum entropy. One of the advantages of such a model is that it enables the unification of information from multiple knowledge sources in one framework. First, probabilistic user models are discovered from the usage data, based on the PLSA approach, and used one set of constraints for the maximum entropy framework. Secondly, for content information, Latent Dirichlet Allocation (LDA) [16] is used to discover the hidden semantic relationships among visited items and specify another set of constraints based on these item association patterns. These two set of constraints are used in a unifying maximum entropy framework to generate recommendations.

Integration of Structured Semantic Knowledge and Usage-Based Models. The integration of content features with usage-based personalization is desirable when we are dealing with sites where text descriptions are dominant and other structural relationships in the data are not easy to obtain, e.g., news sites or online help systems, etc. Keyword-based approaches, however, are incapable of capturing more complex relationships among objects at a deeper semantic level based on the inherent properties associated with these objects. For example, potentially valuable relational structures among objects such as relationships between movies, directors, and actors, or between students, courses, and instructors, may be missed if one can only rely on the description of these entities using sets of keywords.

To be able to recommend different types of complex objects using their underlying properties and attributes, the system must be able to rely on the characterization of user segments and objects, not just based on keywords, but at a deeper semantic level using the domain ontologies for the objects. For instance, in a traditional personalization system on a university Web site might recommend courses in Java to a student, simply because that student has previously taken or shown interest in Java courses. On the other hand, a system that has knowledge of the underlying domain ontology, might recognize that the student should first satisfy the prerequisite requirements for a recommended course, or be able to recommend the best instructors for Java course, and so on.

This observation has led to a number of efforts that attempt to use "ontological user profiles" for personalization. For example, Middleton et al. [77] use an ontological profile for a user within their research paper recommendation system, QuickStep. The profile is based on a topic hierarchy alone. They also attempt to use externally available ontologies based on personnel records and user publications to address the cold-start problem for their recommendations system. The existence of such additional knowledge, while applicable in their specific application domain, cannot however be assumed in a general e-tailer scenario.

Dai and Mobasher [33] provide a general framework for integrating domain knowledge with Web usage mining for user based personalization. The primary focus of the proposed approach is to transform aggregate user models, that are the results of pattern discovery (clustering) on Web transaction data, into ontology enhanced aggregate models. In the initial discovery phase, each "page-level" aggregate models, m, is represented as a vector of item-weight pairs. Specifically, given a session cluster c, the aggregate model m as a set of pageview-weight pairs obtained by computing the centroid of c. Thus, m can be viewed as vector over the n-dimensional space if items (pages): $m = \langle w_m(p_1), w_m(p_2), \cdots, w_m(p_n) \rangle$ where $w_m(p_i)$ is the average weight of p_i across all sessions in the cluster c. Using the domain ontology, objects instances of ontology classes are extracted from each page p_i, and m is transformed into an object-level model $om = \{\langle o_1, w_{o_1} \rangle, \langle o_2, w_{o_2} \rangle, \cdots, \langle o_k, w_{o_k} \rangle\}$ in which each o_i is an object instance in the underlying domain ontology and w_{o_i} represents o_i's significance.

Objects that belong to the same class are combined to form an aggregated pseudo object belonging to that class. In this aggregation process, attribute values for objects of the same class are combined using aggregation functions for different attributes, defined in the domain ontology. The transformed aggregate model represents a set of objects accessed together frequently by a group of users in the same cluster. This new object-space is then used as input to additional data mining algorithm and for generating recommendations for pages that are similar at the object level. An important benefit of aggregation is that the pattern volume is significantly reduced, thus relieving the computation burden for the recommendation engine.

Kearny et al. [59] also investigate how Web usage data may be combined with semantic domain knowledge to provide a deeper understanding of user behavior. In particular, an "impact" measure is introduced based on information theory that captures the influence of a given concept from the domain ontology on user behavior. The impact measures for each of the concepts within the ontology are then combined to create an ontological profile for each user. This approach also begins by mapping each page within user sessions onto the concepts in the ontology. Then the specific instances are generalized to an Ontological Profile (OP). Thus, each page can be represented as a vector over the set of concepts where each dimension measures the degree to which the page belongs to the corresponding concept. In a similar manner as in [33], a composite distance measure based specific domain characteristics of each concept is defined and used as part of the mining and recommendation generation process.

Using Linkage Structure for Model Learning and Selection. Aside from the content features associated with items or pages, there are other information channels and knowledge sources that can be leveraged in the data mining approach to personalization. These include structured semantic information such as that available from domain ontologies or relational databases, and, in the context of Web personalization, the hyperlink structure of the Web site. We discuss the integration of ontological information in the next section. Here, we focus our attention to approaches that have used linkage information as part of the mining and recommendations processes.

Based on their study on the impact of site characteristics on the usefulness of sequential patterns over non-sequential (association) patterns, Nakagawa and Mobasher [90] proposed a hybrid recommendation system that switched between different recommendation algorithms based on the degree of connectivity in the site and the current location of the user within the site. The study showed that the performance of each recommendation model depends, in part, on the structural characteristics of the Web site. For example, in a highly connected Web site with short navigational paths, non-sequential models perform well by achieving higher overall precision and recall than sequential pattern models. In this hybrid approach, a measure of localized connectivity (LCM) is defined with respect to the current page being visited by the user. A logistic regression function is then learned from a set of training user profiles based on the LCM values of pages within the profiles and the best recommendations achieved for each user. This function is then used as a switching criterion to select the best recommendation model for the target user. Evaluation of this approach revealed that the hybrid model outperformed the base recommendation models in both precision and recall.

In [92], the site's hierarchical linkage structure is treated as an implicit concept hierarchy that is exploited in computing the similarity between pages. This similarity function allows for a more robust comparison of sessions that contain pages that are different but structurally related.

Lin and Zaiane [69] proposed a hybrid Web recommender system that combines access history and the content of visited pages, as well as the connectivity between the pages on a Web site, in order to model users' concurrent information needs and generate navigational patterns. These simultaneous goals of users are called "missions". A mission is a sub-session with a consistent goal as determined based on the content similarity of the pages within the session. These missions are in turn clustered to generate navigational patterns, and augmented with their linked neighborhood and ranked according to their authority determined based on site connectivity. These new clusters (i.e., augmented navigational patterns) are provided to the recommendation engine. When a visitor starts a new session, the session is matched with these clusters to generate a recommendation list.

3.5 Evaluating Personalization Models

As in any data mining application, before the discovered models can be deployed as part of a personalization framework, it is essential to evaluate their accuracy and effectiveness. The evaluation of personalization models, however, is an inherently challenging task for several reasons. First, the various modeling approaches and recommendation algorithms, such as those described in the previous section, may require different evaluation metrics. Secondly, the required personalization actions may be quite different depending on the underlying domain, intended application, and the data gathered for personalization. Finally, there is a lack of consensus among researchers and practitioners as to what factors most affect quality of service in personalized systems. Ultimately, the goal of evaluation in this context is to judge the "quality" of recommendations (or personalized content) generated by the system. The factors mentioned above, however, affect how this notion of quality is defined in different settings and according to the personalization task.

Herlocker et al. [50] have identified several types of personalization tasks performed by typical systems. These tasks include providing annotations in context (i.e., annotating items or existing content with prediction scores), finding (some or all) "good" items, recommending a sequence of items, providing decision support for browsing or making purchases, and providing credible recommendations. While evaluating the performance of a personalization system vis-a-vis these tasks generally requires measuring the *accuracy* of recommendations, the aforementioned study indicates that some accuracy metrics are more appropriate for a given task than others. Here, we briefly discuss some of the most commonly used accuracy metrics, and then we consider some of the other factors that impact the quality of recommendations.

The most common approach to evaluation in collaborative filtering systems is to measure the effectiveness of the system's predictive accuracy. Such metrics measures how close the recommender system's predicted ratings are to the actual user ratings. Particularly when dealing with user ratings of items, a frequently used metric is the *Mean Absolute Error* (MAE) [121, 49], which measures the average absolute deviation between a predicted rating and the user's actual rating. Several related accuracy metrics have been proposed for the prediction task with numeric ratings, including root mean squared error and mean squared error, that implicitly assign a greater weight to predictions with larger errors, and the normalized mean squared error [43] that aims to normalize MAE across datasets with varying rating scales.

Massa and Avesani suggest another variant of MAE called the mean absolute user error that calculates the mean absolute error for each user and then averages over all users [74]. This was based on their observation that recommender systems tend to have lower errors when predicting ratings by prolific raters rather than less frequent ones.

This metric is particularly useful when the number of items in the test set varies for each user. For example, this metric may be appropriate if the number of items in the test set is based on a percentage of items rated by a user.

While the MAE and its variants are useful in measuring the accuracy of predictions, they may not provide a complete picture of how good the recommendations are. These metrics may be less appropriate for tasks such as finding “good” items [50], where a ranked result is returned to the user. In such systems the target users usually only view items at the top of the ranking, and thus the accuracy of predictions for items of no interest to the user is not a determinant factor.

Classification metrics, on the other hand, measure the frequency with which a recommender system makes correct or incorrect decisions about recommending an item. Two commonly used metrics in this context are *Precision* and *Recall* which are standard metrics used in evaluating information retrieval effectiveness, but have also been adopted to evaluate ranked ordering of recommended items in personalization [58, 116, 15]. While precision measures the probability that a recommended item is relevant, recall measures the probability that a relevant item is recommended. In order to compute precision and recall in recommender systems, it is necessary to distinguish between the item set that is returned to the user (i.e., selected or recommended), and the item set that is not. One approach in doing so is to determine the set of top N recommended items for a fixed N and consider the remaining items as not recommended.

One advantage of metrics such as precision and recall is that they can be used in the evaluation of personalization systems in which the underlying user preferences are not determined by numeric ratings. This, of course, is the case when dealing with navigational data in which an item is either visited or it is not. In the context of numeric ratings on a continuous scale, it would be necessary to first transform ratings into a binary scale. For example, a rating scale of 1–5 may be transformed into a binary scale by converting every rating of 4 or 5 to “relevant” and all ratings between 1 and 3 to “nonrelevant”. The determination of which items are relevant and which are not poses its own unique challenges when dealing with Web navigation data. A recorded visit to a particular page on a Web site, cannot necessarily be taken as an indication of relevance or interest. One approach that can address this problem is to record the amount of time spent on each page during a session (the pageview duration). To accurately convert this data into a binary scale, it is usually necessary to standardize the pageview durations with respect to the mean duration for that page. In this way, pageviews that last significantly less than the mean duration can be removed from the relevant list.

It should also be noted that there is often a trade-off between precision and recall, so it is important to consider both of these metrics for a given system. Some metrics attempt to combine precision and recall into a single number. One such metric is the The F1 measure which is computed as the harmonic mean of precision and recall [9, 85]. A more general form of the F1 measure can be devised which allows for weighting one of these metrics more than the other, depending on their relative importance in a particular application domain.

A measure that provides and alternative to precision and recall is the Receiver Operating Characteristic (ROC) which has roots in signal detection theory [48]. The ROC metric attempts to measure the extent to the system can successfully distinguish between signal (relevance) and noise. It assumes that the information system will assign a predicted level of relevance to every potential item. The ROC-curve is a plot of the systems sensitivity (the probability of signal, or, in the context of recommendation, the true positive rate) by the complement of its specificity (the probability of noise, or, in the recommendation context, the complement of the true negative rate). Generally, to compare the recommendation accuracy in two systems, the size of the area under the ROC-curve is measured with a larger value indicating better performance.

As noted earlier, the focus of the aforementioned metrics is generally the evaluation of recommendation accuracy. However, the research and practice in personalization technologies has led to the emerging consensus that measuring accuracy alone may not paint a complete picture of how users view the recommendations. The recommendations, or more generally the personalized content generated by the system, must also be "useful" to users. For example, a system that only recommend highly popular items (such as best seller books, or in the context of Web usage, highly visited pages in a site), may be quite accurate based on the above measures, but one can argue that such items are not particularly useful for the users of the system.

Recent user studies have found that a number of issues can affect the perceived usefulness of personalization systems including, trust in the system, transparency of the underlying recommendation algorithm, ability for a user to refine the system generated profile, and diversity of recommendations [134, 125, 145]. Therefore, the evaluation of personalization systems needs to be carried out along a number of dimensions, in addition to accuracy, some of which are better understood that others and have well established metrics available. The key dimensions along which personalization systems can be evaluated (aside from accuracy) include the coverage, utility, explainability, robustness, scalability, and user satisfaction.

Coverage measures the percentage of the universe of items that the recommendation system is capable of producing. For the prediction task it is calculated as the ratio of items for which the system can provide recommendations to all available items. Since it may not be practical to compute predictions for all user-item pairs in the system, this metric is usually estimated by selecting a random sample of user-item pairs, attempting to generate a prediction for each pair, and measuring the percentage for which a prediction was provided. An alternative is to calculate coverage as a percentage of items of interest to a user rather than considering the complete universe of items [50]. If the predictive accuracy is computed by withholding a selection of ratings and then predicting those ratings, the coverage can be measured as the percentage of withheld items for which a prediction is obtained.

The notion of "usefulness" suggests that measuring the utility of a recommendation for a user may be required. Breese et al. [19] suggested a metric based on the expected utility of the recommendation list. The utility of each item is calculated by the difference in vote for the item and a "neutral" weight. The metric is then calculated as the weighted sum of the utilities of all items in the list where the weight signifies the probability that an item in the ranked list will be viewed or selected by the user. This likelihood that a user will view or select each successive item is defined by an exponential decay function, where the decay factor is described by a half-life parameter. The basic, and rather strong, assumption behind this metric is that the true utility (in terms of cost/benefit analysis) rapidly (exponentially) drops as the search length increases.

The utility of recommendations or personalized content produced by the system can also be viewed in terms of their novelty. If the system only produces obvious recommendations, even if accurate, the recommendation may not be perceived as useful by the users. Clearly, the novelty of recommendations is not only user-specific, but also domain dependent and, therefore, measuring it would require domain specific metrics. For example, in the context of Web navigation, several metrics have been proposed that measure utility based on the distance of the recommended item from the current page (referred to as navigation distance) [5]. Although novelty may be an important consideration, it should be noted that several studies have found that there is, in fact value in providing user with some "obvious" recommendations [134]. Such recommendations tend to increase user confidence in the system leading to the a user perception that the system does generate credible recommendation; an important factor in the success of the personalization system.

A number of metrics have been proposed in literature for evaluating the robustness of a recommender system. Such metrics attempt to provide a quantitative measure of the extent to which an attack can affect a recommender system. Stability of prediction [96] measures the percentage of unrated (user,items) pairs that have a prediction shift less that a predefined constant. Power of an attack [96] on the other hand measures the average change in the gap between the predicted and target rating for the target item. The target item is the item that the attack is attempting to push or nuke. The power of attack metric assumes that the goal of the attack is to force item ratings to a target rating value. Noting that the effect of an attack on an items current rating is not necessarily going to affect its ability to be recommended, Lam and Herlocker [67] proposed an alternative metric called the Change in Expected change in top-N occupancy. It is calculated as the average expected occurrence of the target items in the top-N recommendation list of users.

The performance and scalability dimension aims to measure the response time of a given recommendation algorithm and how easily it can scale to handle a large number of concurrent requests for recommendations. Typically, these systems need to be able to handle large volumes of recommendation requests without significantly adding to the response time of the Web site that they have been deployed on.

Finally, attempts to measure user satisfaction range from using business metrics for customer loyalty such as RFM and life-time value through to more simplistic measures such as recommendation uptake. For example, the físchlár video recommendation system [126] implicitly obtains a measure of user satisfaction by checking is the recommended items were played or recorded.

3.6 Conclusions

In this chapter we have presented a comprehensive discussion the Web personalization process viewed as an application of data mining which must therefore be supported during the various phases of a typical data mining cycle. We have discussed a host of activities and techniques used at different stages of this cycle, including the preprocessing and integration of data from multiple sources, and pattern discovery techniques that are applied to this data. We have also presented a number of specific recommendation algorithms for combining the discovered knowledge with the current status of a user's activity in a Web site to provide personalized content to a user. The approaches we have detailed show how pattern discovery techniques such as clustering, association rule mining, and sequential pattern discovery, and probabilistic models performed on Web usage collaborative data, can be leveraged effectively as an integrated part of a Web personalization system.

While a research into personalization has led to a number of effective algorithms and commercial success stories, a number of challenges and open questions still remain.

A key part of the personalization process is the generation of user models. The most commonly used user models are still rather simplistic, representing the user as a vector of ratings or using a set of keywords. Even where more multi-dimensional or ontological information has been available, the data is generally mapped onto a single user-item table which is more amenable for most data mining and machine learning techniques. To provide the most useful and effective recommendations, personalization systems need to incorporate more expressive models. Some of the discussion on the integration of semantic knowledge and ontologies in the mining process suggests that some strides have been made in this direction. However, most of this work has not, as of yet, resulted in true and tested approaches that can become the basis of the next generation personalization systems.

Another important and difficult of challenge is the modeling of user context. In particular profiles commonly used today lack in their ability to model user context and dynamics. Users access different items for different reasons and under different contexts. The modeling of context and its use within recommendation generation needs to be explored further. Also, user interests and needs change with time. Identifying these changes and adapting to them is a key goal of personalization. However, very little research effort has been expended the evolution of user patterns over time and their impact on recommendations. This is in part due to the trade-offs between expressiveness of the profiles and scalability with respect to the number of active users.

Solutions to these important challenges are likely to lead to the creation of the next-generation of more effective and useful Web personalization and recommender systems that can be deployed in increasingly more complex Web-based environments.

References

1. Agarwal, R., Aggarwal, C., Prasad, V.: A tree projection algorithm for generation of frequent itemsets. Journal of Parallel and Distributed Computing **61**(3) (2001) 350–371
2. Aggarwal, C.C., Wolf, J.L., Yu, P.S.: A new method for similarity indexing for market data. In: Proceedings of the 1999 ACM SIGMOD Conference, Philadelphia, PA (June 1999) 407–418
3. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules. In: Proceedings of the 20th International Conference on Very Large Data Bases (VLDB'94), Santiago, Chile (September 1994) 487–499

4. Agrawal, R., Srikant, R.: Mining sequential patterns. In: Proceedings of the International Conference on Data Engineering (ICDE'95), Taipei, Taiwan (March 1995) 3–14
5. Anderson, C., Domingos, P., Weld, D.: Adaptive web navigation for wireless devices. In: Proceedings of the 17th International Joint Conference on Artificial Intelligence, Seattle, Washington (August 2001) 879–884
6. Balabanovic, M., Shohan, Y.: Fab: Content-based, collaborative recommendation. Communications of the ACM **40**(3) (1997) 66–72
7. Banerjee, A., Ghosh, J.: Clickstream clustering using weighted longest common subsequences. In: Proceedings of the Web Mining Workshop at the 1st SIAM Conference on Data Mining, Chicago, Illinois (April 2001)
8. Bartholomem, D., Knott, M.: Latent Variable Models and Factor Analysis. Oxford University Press, New York, USA (1999)
9. Basu, C., Hirsh, H., Cohen, W.: Recommendation as classification: Using social and content-based information in recommendation. In: Proceedings of the Recommender System Workshop at AAAI 98, Madison, Wisconsin (July 1998) 11–15
10. Baumgarten, M., Büchner, A.G., Anand, S.S., Mulvenna, M.D., Hughes, J.: User-driven navigation pattern discovery from internet data. web usage analysis and user profiling. In Masand, B., Spiliopoulou, M., eds.: Web Usage Analysis and User Profiling: Proceedings of the WEBKDD'99 Workshop. Lecture Notes in Computer Science 1836. Springer-Verlag (2000) 74–91
11. Belkin, N., Croft, B.: Information filtering and information retrieval. Communications of ACM **35**(12) (2001) 29–37
12. Berendt, B., Mobasher, B., Nakagawa, M., Spiliopoulou, M.: The impact of site structure and user environment on session reconstruction in web usage analysis. In Zaïane, O.R., Srivastava, J., Spiliopoulou, M., Masand, B., eds.: Proceedings of WEBKDD 2002 - Mining Web Data for Discovering Usage Patterns and Profiles. Volume 2703 of LNCS. Springer Berlin / Heidelberg (2003) 159–179
13. Berendt, B., Spiliopoulou, M.: Analysis of navigation behaviour in web sites integrating multiple information systems. VLDB Journal, Special Issue on Databases and the Web **9**(1) (2000) 56–75
14. Bezdek, J.C.: Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum Press, New York (1981)
15. Billsus, D., Pazzani, M.J.: Learning collaborative information filters. In: Proceedings of the 15th International Conference on Machine Learning (ICML'98), Madison, Wisconsin (July 1998) 46–53
16. Blei, D., Ng, A., Jordan, M.: Latent dirichlet allocation. Journal of Machine Learning Research **3** (2003) 993–1022
17. Borges, J., Levene, M.: Data mining of user navigation patterns. In Masand, B., Spiliopoulou, M., eds.: Web Usage Analysis and User Profiling: Proceedings of the WEBKDD'99 Workshop. LNAI 1836. Springer-Verlag (1999) 92–111
18. Brants, T., Chen, F., Tsochantaridis, I.: Topic-based document segmentation with probabilistic latent semantic analysis. In: Proceedings of the Eleventh International Conference on Information and Knowledge Management, Washington D.C. (November 2002) 211–218
19. Breese, J.S., Heckerman, D., Kadie, C.: Empirical analysis of predictive algorithms for collaborative filtering. In: Proceedings of the Fourteenth Annual Conference on Uncertainty in Artificial Intelligence, Madison, Wisconsin (July 1998) 43–52
20. Büchner, A., Mulvenna, M.D.: Discovering internet marketing intelligence through online analytical web usage mining. SIGMOD Record **4**(27) (1998) 54–61
21. Burke, R.: Hybrid systems for personalized recommendations. In Mobasher, B., Anand, S.S., eds.: Intelligent Techniques in Web Personalisation. LNAI 3169. Springer-Verlag (2005) 133–152

22. Burke, R.: Hybrid web recommender systems. In Brusilovsky, P., Kobsa, A., Nejdl, W., eds.: The Adaptive Web: Methods and Strategies of Web Personalization. Volume 4321 of Lecture Notes in Computer Science. Springer-Verlag, Berlin Heidelberg New York (2007) This Volume
23. Cadez, I., Heckerman, D., Meek, C., Smyth, P., White, S.: Model-based clustering and visualization of navigation patterns on a web site. Journal of Data Mining and Knowledge Discovery **7**(4) (2003) 399–424
24. Cadez, I., Smyth, P., Ip, E., Mannila, H.: Predictive profiles for transaction data using finite mixture models. Technical Report Technical Report No. 01–67, Information and Computer Science Department, University of California, Irvine, Irvine, CA (2001)
25. Canny, J.: Collaborative filtering with privacy via factor analysis. In: Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Tampere, Finland (August 2002) 238–245
26. Cassel, L., Wolz, U.: Client side personalization. In: Proceedings of the Second DELOS Network of Excellence Workshop on Personalization and Recommender Systems in Digital Libraries, Dublin, Ireland (June 2001)
27. Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., Wirth, R.: Crisp-dm 1.0: Step-by-step data mining guide. http://www.crisp-dm.org (2000)
28. Claypool, M., Gokhale, A., Miranda, T., Murnikov, P., Netes, D., Sartin, M.: Combining content-based and collaborative filters in an online newspaper. In: Proceedings of the ACM SIGIR '99 Workshop on Recommender Systems: Algorithms and Evaluation, Berkeley, California (August 1999)
29. Cohn, D., Chang, H.: Probabilistically identifying authoritative documents. In: Proceedings of the Seventeenth International Conference on Machine Learning, Stanford, CA (June 2000) 167–174
30. Cohn, D., Hofmann, T.: The missing link: A probabilistic model of document content and hypertext connectivity. In Todd K. Leen, T.G.D., Tresp, V., eds.: Advances in Neural Information Processing Systems 13. MIT Press, Vancouver, Canada (2001) 430–436
31. Cooley, R., Mobasher, B., Srivastava, J.: 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), Newport Beach, CA (November 1997) 558–567
32. Cooley, R., Mobasher, B., Srivastava, J.: Data preparation for mining world wide web browsing patterns. Journal of Knowledge and Information Systems **1**(1) (1999) 5–32
33. Dai, H., Mobasher, B.: A road map to more effective web personalization: Integrating domain knowledge with web usage mining. In: Proceedings of the International Conference on Internet Computing, IC03, Las Vegas (June 2003) 58–64
34. Dempster, A., Laird, N., Rubin, D.: Maximum likelihood from incomplete data via the em algorithm. Journal of Royal Statistical Society **B**(39) (1977) 1–38
35. Deshpande, M., Karypis, G.: Item-based top-n recommendation algorithms. ACM Transactions on Information Systems **22**(1) (2004) 1–34
36. Deshpande, M., Karypis, G.: Selective markov models for predicting web-page accesses. ACM Transactions on Internet Technology **4**(2) (2004) 163–184
37. Eirinaki, M., Vazirgiannis, M., Varlamis, I.: Sewep: Using site semantics and a taxonomy to enhance the web personalization process. In: Proceedings of the 9th SIGKDD International Conference on Data Mining and Knowledge Discovery (KDD'03), Washington, DC (August 2003) 99–108
38. Eveitt, B.: An Introduction to Latent Variable Models. Champman and Hall, New York, USA (1984)
39. Fu, X., Budzik, J., Hammond, K.J.: Mining navigation history for recommendation. In: Proceedings of the 2000 International Conference on Intelligent User Interfaces, New Orleans, LA, ACM Press (January 2000) 106 – 112

40. Gauch, S., Speretta, M., Chandramouli, A., Micarelli, A.: User profiles for personalized information access. In Brusilovsky, P., Kobsa, A., Nejdl, W., eds.: The Adaptive Web: Methods and Strategies of Web Personalization. Volume 4321 of Lecture Notes in Computer Science. Springer-Verlag, Berlin Heidelberg New York (2007) This Volume
41. Ghani, R., Fano, A.: Building recommender systems using a knowledge base of product semantics. In: Proceedings of the Workshop on Recommendation and Personalization in E-Commerce, at the 2nd Int'l Conf. on Adaptive Hypermedia and Adaptive Web Based Systems, Malaga, Spain (May 2002)
42. Girolami, M., Kaban, A.: On an equivalence between plsi and lda. In: Proceedings of the 26th Annual International ACM SIGIR Conference (SIGIR'03), Toronto, Canada (July 2003) 433–434
43. Goldberg, K., Roeder, T., Gupta, D., Perkins, C.: Eigentaste: A constant time collaborative filtering algorithm. Information Retrieval **4**(2) (2001) 133–151
44. Griffiths, T., Steyvers, M.: Finding scientific topics. Proceedings of the National Academy of Sciences, PNAS 2004 **101** (April 2004) 5228–5235
45. Haase, P., Ehrig, M., Hotho, A., Schnizler, B.: Personalized information access in a bibliographic peer-to-peer system. In: Proceedings of the AAAI Workshop on Semantic Web Personalization, AAAI Workshop Technical Report (2004) 1–12
46. Han, E., Karypis, G., Kumar, V., Mobasher, B.: Hypergraph based clustering in high-dimensional data sets: A summary of results. IEEE Data Engineering Bulletin **21**(1) (March 1998) 15–22
47. Han, J., Kamber, M.: Data Mining: Concepts and Techniques. Morgan Kaufmann, San Francisco, CA (2001)
48. Hanley, J.A., McNeil, B.J.: The meaning and use of the area under a receiver operating characteristic (roc) curve. Radiology (143) (1982) 29–36
49. Herlocker, J., Konstan, J., Borchers, A., Riedl, J.: An algorithmic framework for performing collaborative filtering. In: Proceedings of the 22nd ACM Conference on Research and Development in Information Retrieval (SIGIR'99), Berkeley, CA (August 1999) 230–237
50. Herlocker, J.L., Konstan, J.A., Terveen, L.G., Riedl, J.: Evaluating collaborative filtering recommender systems. ACM Transactions on Information Systems **22**(1) (2004) 5–53
51. Hofmann, T.: Probabilistic latent semantic indexing. In: Proceedings of the 22nd International Conference on Research and Development in Information Retrieval, Berkeley, CA (August 1999) 50–57
52. Hofmann, T.: Unsupervised learning by probabilistic latent semantic analysis. Machine Learning Journal **42**(1) (2001) 177–196
53. Hofmann, T.: Latent semantic models for collaborative filtering. ACM Transactions on Information Systems **22**(1) (2004) 89–115
54. Jin, X., Zhou, Y., Mobasher, B.: A unified approach to personalization based on probabilistic latent semantic models of web usage and content. In: Proceedings of the AAAI 2004 Workshop on Semantic Web Personalization (SWP'04), San Jose, CA (2004)
55. Jin, X., Zhou, Y., Mobasher, B.: Web usage mining based on probabilistic latent semantic analysis. In: Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD04), Seattle, WA (August 2004) 197–205
56. Jin, X., Zhou, Y., Mobasher, B.: Task-oriented web user modeling for recommendation. In: Proceedings of the 10th International Conference on User Modeling (UM'05), Edinburgh, UK (July 2005) 109–118
57. Joshi, A., Krishnapuram, R.: On mining web access logs. In: Proceedings of the ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery (DMKD 2000), Dallas, Texas (May 2000)

58. Karypis, G.: Evaluation of item-based top-n recommendation algorithms. In: Proceedings of the tenth International conference on Information and knowledge management (CIKM'01), Atlanta, Georgia (October 2001) 247–254
59. Kearney, P., Anand, S.S., Shapcott, M.: Employing a domain ontology to gain insights into user behaviour. In: Proceedings of the 3rd Workshop on Intelligent Techniques for Web Personalization, at IJCAI 2005, Edinburgh, Scotland (August 2005)
60. Kim, Y., Chang, J., Zhang, B.: a empirical study on dimensionality optimization in text mining for linguistic knowledge acquisition. In: Proceedings of the Seventh Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD-03), Seol, Korea (April 2003) 111–116
61. Kohavi, R., Mason, L., Parekh, R., Zheng, Z.: Lessons and challenges from mining retail e-commerce data. Machine Learning **57**(1–2) (2004) 83–113
62. Kohavi, R., Provost, F.: Applications of data mining to electronic commerce. Data Mining and Knowledge Discovery **5**(1–2) (2001) 5–10
63. Kohrs, A., Mérialdo, B.: Clustering for collaborative filtering applications. In: Proceedings of the International Conference on Computational Intelligence for Modelling, Control & Automation (CIMCA'99), Vienna, Austria (February 1999)
64. Konstan, J., Miller, B., Maltz, D., Herlocker, J., Gordon, L., Riedl, J.: Grouplens: Applying collaborative filtering to usenet news. Communications of the ACM **40**(3) (1997) 77–87
65. Krulwich, B.: Lifestyle finder: Intelligent user profiling using large-scale demographic data. AI Magazine **18**(2) (1997) 37–45
66. Krulwich, B., Burkey, C.: Learning user information interests through extraction of semantically significant phrases. In: Proceedings of the AAAI Spring Symposium on Machine Learning in Information Access, Stanford, California (March 1996)
67. Lam, S.K., Riedl, J.: Shilling recommender systems for fun and profit. In: Proceedings of the 13th international World Wide Web conference (WWW'04), New York, NY (May 2004) 393–402
68. Lang, K.: Newsweeder: Learning to filter netnews. In: Proceedings of the 12th International Conference on Machine Learning, Tahoe City, California (July 1995) 331–339
69. Li, J., Zaiane, O.: Using distinctive information channels for mission-based recommender systems. In: Proceedings of the sixth WEBKDD workshop: Webmining and Web Usage Analysis (WEBKDD04), in conjunction with the 10th ACM SIGKDD conference (KDD'04), Seattle, Washington (August 2004)
70. Lieberman, H.: Letizia: An agent that assists web browsing. In: Proceedings of the 14th International Joint Conference in Artificial Intelligence (IJCAI'95), Montreal, Quebec, Canada (August 1995) 924–929
71. Lin, W., Alvarez, S.A., Ruiz, C.: Efficient adaptive-support association rule mining for recommender systems. Data Mining and Knowledge Discovery **6** (2002) 83–105
72. Linden, G., Smith, B., York, J.: Amazon.com recommendations: Item-to-item collaborative filtering. IEEE Internet Computing **7**(1) (2003) 76–80
73. Marlin, B.: Modeling user rating profiles for collaborative filtering. In: Proceedings of the 17th Annual Conference on Neural Information Processing System (NIPS'03), Vancouver, B.C., Canada (December 2003)
74. Massa, P., Avesani, P.: Trust-aware collaborative filtering for recommender systems. In: Proceedings of International Conference on Cooperative Information Systems, Larnaca, Cyprus (October 2004) 492–508
75. Melville, P., Mooney, R., Nagarajan, R.: Content-boosted collaborative filtering. In: Proceedings of the SIGIR2001 Workshop on Recommender Systems, New Orleans, LA (September 2001)

76. Micarelli, A., Sciarrone, F., Marinilli, M.: Web document modeling. In Brusilovsky, P., Kobsa, A., Nejdl, W., eds.: The Adaptive Web: Methods and Strategies of Web Personalization. Volume 4321 of Lecture Notes in Computer Science. Springer-Verlag, Berlin Heidelberg New York (2007) This Volume
77. Middleton, S.E., Shadbolt, N.R., Roure, D.C.D.: Ontological user profiling in recommender systems. ACM Transactions on Information Systems **22**(1) (2004) 54–88
78. Minka, T., Lafferty, J.: Expectation-propagation for the generative aspect model. In: Proceedings of the 18th Conference on Uncertainty in Artificial Intelligence, Edmonton, Alberta, Canada (August 2002) 352–359
79. Mladenic, D.: Personal web watcher: Implementation and design. Technical Report IJS-DP-7472, Department of Intelligent Systems, J. Stefan Institute, Slovenia (1996)
80. Mladenic, D.: Text-learning and related intelligent agents: A survey. IEEE Intelligent Systems **14**(4) (July/August 1999) 44–54
81. Mobasher, B.: Web usage mining. In Wong, J., ed.: Encyclopedia of Data Warehousing and Data Mining. Idea Group Publishing (2005) 1216–1220
82. Mobasher, B.: Web usage mining and personalization. In Singh, M.P., ed.: Practical Handbook of Internet Computing. CRC Press (2005)
83. Mobasher, B., Dai, H., Luo, T., Nakagawa, M.: Effective personalization based on association rule discovery from web usage data. In: Proceedings of the 3rd ACM Workshop on Web Information and Data Management (WIDM01), Atlanta, Georgia (November 2001)
84. Mobasher, B., Dai, H., Luo, T., Nakagawa, M.: Improving the effectiveness of collaborative filtering on anonymous web usage data. In: Proceedings of the IJCAI 2001 Workshop on Intelligent Techniques for Web Personalization (ITWP01), Seattle, WA (August 2001)
85. Mobasher, B., Dai, H., Luo, T., Nakagawa, M.: Discovery and evaluation of aggregate usage profiles for web personalization. Data Mining and Knowledge Discovery **6**(1) (2002) 61–82
86. Mobasher, B., Dai, H., Luo, T., Nakagawa, M.: Using sequential and non-sequential patterns for predictive web usage mining tasks. In: Proceedings of the IEEE International Conference on Data Mining, Maebashi City, Japan (December 2002) 669–672
87. Mobasher, B., Dai, H., Luo, T., Sun, Y., Zhu, J.: Integrating web usage and content mining for more effective personalization. In: E-Commerce and Web Technologies: Proceedings of the EC-WEB 2000 Conference. Lecture Notes in Computer Science (LNCS) 1875, Springer (September 2000) 165–176
88. Mobasher, B., Dai, H., T. Luo, M.N.: Discovery and evaluation of aggregate usage profiles for web personalization. Data Mining and Knowledge Discovery **6** (2002) 61–82
89. Mulvenna, M., Anand, S.S., Büchner, A.G.: Personalization on the net using web mining. Communication of ACM **43**(8) (2000) 122–125
90. Nakagawa, M., Mobasher, B.: A hybrid web personalization model based on site connectivity. In: Proceedings of the WebKDD 2003 Workshop, at the ACM-SIGKDD Conference on Knowledge Discovery in Databases (KDD'2003), Washington, DC (August 2003)
91. Nasraoui, O., Frigui, H., Krishnapuram, R., Joshi, A.: Extracting web user profiles using relational competitive fuzzy clustering. International Journal on Artificial Intelligence Tools **9**(4) (2000) 509–526
92. Nasraoui, O., Krishnapuram, R., Joshi, A., Kamdar, T.: Automatic web user profiling and personalization using robust fuzzy relational clustering. In Segovia, J., Szczepaniak, P., Niedzwiedzinski, M., eds.: Studies in Fuzziness and Soft Computing. Volume 105. Springer-Verlag, Heidelberg (2002) 233–261
93. Niu, L., Yan, X., , Zhang, C., , Zhang, S.: Product hierarchy-based customer profiles for electronic commerce recommendation. In: Proceedings of the 1st International Conference on Machine Learning and Cybernetics. (2002) 1075–1080

94. Oberle, D., Berendt, B., Hotho, A., Gonzalez, J.: Conceptual user tracking. In: Proceedings of the Atlantic Web Intelligence Conference (AWIC'03), Madrid, Spain (May 2003) 155–164
95. O'Connor, M., Herlocker, J.: Clustering items for collaborative filtering. In: Proceedings of ACM SIGIR'99 Workshop on Recommender Systems: Algorithms and Evaluation, Berkeley, California (August 1999)
96. O'Mahony, M., Hurley, N., Kushmerick, N., Silverstre, G.: Collaborative recommendations: A robustness analysis. ACM Transactions on Internet Technologies **4**(4) (2004) 344–377
97. Padmanabhan, B., Tuzhilin, A.: Unexpectedness as a measure of interestingness in knowledge discovery. Decision Support Systems **27**(3) (1999) 303–318
98. Palpanas, T., Mendelzon, A.: Web prefetching using partial match prediction. In: Proceedings of the 4th International Web Caching Workshop (WCW99), San Diego, CA (March 1999)
99. Parent, S., Mobasher, B., Lytinen, S.: An adaptive agent for web exploration based on concept hierarchies. In: Proceedings of the 9th International Conference on Human Computer Interaction, New Orleans (August 2001) 903–907
100. Pavlov, D.: Sequence modeling with mixtures of conditional maximum entropy distributions. In: Proceedings of the Third IEEE International Conference on Data Mining (ICDM'03), Melbourne, Florida (November 2003) 251–258
101. Pazzani, M.: A framework for collaborative, content-based and demographic filtering. Artificial Intelligence Review **13**(5-6) (1999) 393–408
102. Pazzani, M., Billsus, D.: Learning and revising user profiles: The identification of interesting web sites. Machine Learning **27** (1997) 313–331
103. Pazzani, M.J., Billsus, D.: Content-based recommendation systems. In Brusilovsky, P., Kobsa, A., Nejdl, W., eds.: The Adaptive Web: Methods and Strategies of Web Personalization. Volume 4321 of Lecture Notes in Computer Science. Springer-Verlag, Berlin Heidelberg New York (2007) This Volume
104. Pei, J., Han, J., Mortazavi-Asl, B., Zhu, H.: Mining access patterns efficiently from web logs. In: Proceedings of the 4th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD'00), Kyoto, Japan (April 2000) 396–407
105. Perkowitz, M., Etzioni, O.: Adaptive web sites: Automatically synthesizing web pages. In: Proceedings of the 15th National Conference on Artificial Intelligence, Madison, WI (July 1998) 727–732
106. Perkowitz, M., Etzioni, O.: Adaptive web sites. Communications of ACM **43**(8) (2000) 152–158
107. Pitkow, J., Pirolli, P.: Mining longest repeating subsequences to predict www surfing. In: Proceedings of the 2nd USENIX Symposium on Internet Technologies and Systems, Boulder, Colorado (October 1999)
108. Popescul, A., Ungar, L., Pennock, D., Lawrence, S.: Probabilistic models for unified collaborative and content-based recommendation in sparse-data environments. In: Proceedings of 17th Conference in Uncertainty in Artificial Intelligence, Seattle, WA (August 2001) 437–444
109. Rissanen, J.: Modelling by shortest data description. Automatica **14** (1978) 465–471
110. Rivasseau, J.: Understanding and applying lda model to first-order markov chains. Univ. of british columbia, canada, technical report, Univ. of British Columbia, Canada, Canada (2003)
111. Rosenfeld, R.: Adaptive statistical language modeling: A maximum entropy approach. Phd dissertation, CMU (1994)
112. Salton, G., McGill, M.: Introduction to Modern Information Retrieval. McGraw-Hill, New York, NY (1983)

113. Sarukkai, R.: Link prediction and path analysis using markov chains. In: Proceedings of the 9th International World Wide Web Conference, Amsterdam (May 2000)
114. Sarwar, B., Karypis, G., Konstan, J., Riedl, J.: Item-based collaborative filtering recommendation algorithms. In: Proceedings of the 10th International WWW Conference, Hong Kong (May 2001) 285–295
115. Sarwar, B.M., Karypis, G., Konstan, J., Riedl, J.: Analysis of recommender algorithms for e-commerce. In: Proceedings of the 2nd ACM E-Commerce Conference (EC'00), Minneapolis, MN (October 2000) 158–167
116. Sarwar, B.M., Karypis, G., Konstan, J.A., Riedl, J.: Application of dimensionality reduction in recommender system - a case study. In: Proceedings of the WebKDD 2000 Web Mining for E-Commerce Workshop at ACM SIGKDD 2000, Boston (August 2000)
117. Schafer, J.B., Frankowski, D., Herlocker, J.L., Sen, S.: Collaborative filtering recommender systems. In Brusilovsky, P., Kobsa, A., Nejdl, W., eds.: The Adaptive Web: Methods and Strategies of Web Personalization. Volume 4321 of Lecture Notes in Computer Science. Springer-Verlag, Berlin Heidelberg New York (2006) This Volume
118. Schafer, J., Konstan, J., , Riedl, J.: Recommender systems in e-commerce. In: Proceedings of the ACM Conference on Electronic Commerce, Denver, Colorado (November 1999) 158–166
119. Schechter, S., Krishnan, M., Smith, M.D.: Using path profiles to predict http requests. In: Proceedings of the 7th International World Wide Web Conference, Brisbane, Australia (April 1998)
120. Schwab, I., Kobsa, A., Koychev, I.: Learning about users from observation. In: Adaptive User Interfaces: Papers from the 2000 AAAI Spring Symposium, Menlo Park, CA, AAAI Press (2000)
121. Shardanand, U., Maes, P.: Social information filtering: Algorithms for automating word of mouth. In: Proceedings of the 1995 ACM Conference on Human Factors in Computing Systems (CHI'95), Denver, Colorado (May 1995) 210–217
122. Sieg, A., Mobasher, B., Burke, R.: Inferring user's information context from user profiles and concept hierarchies. In: Proceedings of the 2004 Meeting of the International Federation of Classification Societies, IFCS 2004, Chicago, IL (July 2004) 563–574
123. Silberschatz, A., Tuzhilin, A.: What makes patterns interesting in knowledge discovery systems. IEEE Transactions on Knowledge and Data Engineering **8**(6) (1996) 970–974
124. Sinha, R., Swearingen, K.: Comparing recommendaions made by online systems and friends. In: Proceedings of Delos-NSF Workshop on Personalisation and Recommender Systems in Digital Libraries. (June 2001)
125. Sinha, R., Swearingen, K.: The role of transaprency in recommender systems. In: CHI '02 extended abstracts on Human factors in computing systems. (2002) 830–831
126. Smeaton, A., Murphy, N., O'Connor, N.E., Marlow, S., Lee, H., McDonald, K., Browne, P., Ye, J.: The físchlár digital video system: a digital library of broadcast tv programmes. In: Proceedings of the 1st ACM/IEEE-CS Joint Conference on Digital Libraries, Roanoke, Virginia (June 2001) 312–313
127. Smyth, P.: Probabilistic model-based clustering of multivariate and sequential data. In Heckerman, D., Whittaker, J., eds.: Proceedings of the Seventh International Workshop on AI and Statistics, Los Gatos, CA, Morgan Kaufmann (January 1999)
128. Spiliopoulou, M., Faulstich, L.: Wum: A tool for web utilization analysis. In: Proceedings of EDBT Workshop at WebDB'98. LNCS 1590, Springer Verlag (1999) 184–203
129. Spiliopoulou, M., Mobasher, B., Berendt, B., Nakagawa, M.: A framework for the evaluation of session reconstruction heuristics in web usage analysis. INFORMS Journal of Computing - Special Issue on Mining Web-Based Data for E-Business Applications **15**(2) (2003)

130. Srivastava, J., Cooley, R., Deshpande, M., Tan, P.: Web usage mining: Discovery and applications of usage patterns from web data. SIGKDD Explorations **1**(2) (2000) 12–23
131. Steyvers, M., Smyth, P., Rosen-Zvi, M., Griffiths, T.: Probabilistic author-topic models for information discovery. In: Proceedings of the International Conference on Knowledge Discovery and Data Mining (KDD'04), Seattle, Washington (August 2004) 306–315
132. Strehl, A., Ghosh, J.: Relationship-based clustering and visualization for high-dimensional data mining. INFORMS Journal Of Computing, Special Issue on Web Mining, (A. Tuzhilin and L. Rashid, guest Eds.) **15**(2) (2003) 208–230
133. Suryavanshi, B.S., Shiri, N., Mudur, S.P.: Improving the effectiveness of model based recommender systems for highly sparse and noisy web usage data. In: Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence (WI'05), Compiegne, France (September 2005) 618–621
134. Swearingen, K., Sinha, R.: Beyond algorithms: An hci perspective on recommender systems. In: Proceedings of the ACM SIGIR Workshop on Recommender Systems, New Orleans. LA (September 2001)
135. Tan, P., Kumar, V.: Discovery of web robot sessions based on their navigational patterns. Data Mining and Knowledge Discovery **6** (2002) 9–35
136. Tan, P., Kumar, V., Srivastava, J.: Selecting the right objective measure for association analysis. Information Systems **29**(4) (2004) 293–313
137. Tanasa, D., Trousse, B.: Advanced data preprocessing for intersite web usage mining. IEEE Intelligent Systems **19**(2) (2004) 59–65
138. Teevan, J., Dumais, S.T., Horvitz, E.: Personalizing search via automated analysis of interests and activities. In: Proceedings of 28th ACM SIGIR Conference on Research and Development in Information Retrieval, Salvador, Brazil (August 2005) 449–456
139. Trajkova, J., Gauch, S.: Improving ontology-based user profiles. In: Proceedings of the Recherche d'Information Assiste par Ordinateur, RIAO 2004, University of Avignon (Vaucluse), France (April 2004) 380–389
140. Ungar, L., Foster, D.P.: Clustering methods for collaborative filtering. In: Proceedings of the AAAI98 Workshop on Recommendation Systems, Madison Wisconsin (July 1998)
141. Ypma, A., Heskes, T.: Categorization of web pages and user clustering with mixtures of hidden markov models. In: Proceedings of the WEBKDD 2002 Workshop: Web Mining for Usage Patterns and User Profiles, at SIGKDD 2002), Edmonton, Alberta, Canada (July 2002)
142. Yu, K., Schwaighofer, A., Tresp, V., Ma, W., Zhang, H.: Collaborative ensembling learning: Combining collaborative and content-based information filtering. In: Proceedings of the 19th Conference on Uncertainty in Artificial Intelligence (UAI'03), Acapulco, Mexico (August 2003) 616–623
143. Yu, P.S.: Data mining and personalization technologies. In: Proceedings of the International Conference on Database Systems for Advanced Applications (DASFAA99), Hsinchu, Taiwan (April 1999) 6–13
144. Zhou, Y., Jin, X., Mobasher, B.: A recommendation model based on latent principle factors in web navigation data. In: Proceedings of the 3rd International Workshop on Web Dynamics at WWW 2004 Conference, New York (2004)
145. Ziegler, C., McNee, S.M., Konstan, J.A., Lausen, G.: Improving recommendation lists through topic diversification. In: Proceedings of the 14th international World Wide Web conference, Chiba, Japan (May 2005) 22–32