A SURVEY OF RECOMMENDATION SYSTEMS
IN ELECTRONIC COMMERCE

Chih-Ping Wei, Michael J. Shaw, and Robert F. Easley

Chih-Ping Wei currently is a visiting scholar of University of Illinois at Urbana-Champaign and Associate Professor at Department of Information Management, College of Management, National Sun Yat-Sen University, Taiwan, R.O.C. Micheal J. Shaw is Leonard C. and Mary Lou Hoeft Endowed Chair Professor in Information Technology Management and Director of the Center for Information System and Technology Management at the University of Illinois at Urbana-Champaign. Robert F. Easley is Assistant Professor of MIS, Department of Management, Mendoza College of Business, University of Notre Dame. The authors are grateful for the supports from National Science Council, Taiwan, R.O.C., Beckman Institute for Advanced Science and Technology, Center for International Education and Research in Accounting at the University of Illinois at Urbana-Champaign, and Mendoza College of Business at the University of Notre Dame.

Please address all future correspondence to: Chih-Ping Wei, Department of Information Management, College of Management, National Sun Yat-Sen University, Kaohsiung, Taiwan, R.O.C. Phone: +886-7-525-2000 ext 4729. Fax: +886-7-525-4799. Email: cwei@mis.nsysu.edu.tw.

November 2001
A SURVEY OF RECOMMENDATION SYSTEMS IN ELECTRONIC COMMERCE

Abstract

In an electronic commerce (E-commerce) environment, information users or online customers may experience information overload and make use of services that seek to help them in selecting from an overwhelming array of information or products. Merchandisers too may seek to better manage customer relationships that lead to higher customer satisfaction and loyalty. In response, recommendation systems have emerged as a class of e-service that not only addresses the challenge of information overload by suggesting information or products that are of most interest to users, but also facilitates the delivery of such services to customers. This chapter aims at providing a comprehensive review of recommendation approaches and their associated techniques. Broadly, recommendation systems can be classified into popularity-based, content-based, collaborative-filtering-based, association-based, demographics-based, and reputation-based recommendation approaches. Representative recommendation systems will be depicted in this chapter.

Keywords: Recommender System, Electronic Commerce, Content-based Recommendations, Collaborative Filtering Recommendations, Association-based Recommendations, Demographics-based Recommendations, Reputation-based Recommendations
1. INTRODUCTION

Advances in information and networking technology have not only facilitated the creation, distribution and access of online information, they have also fostered conducting business transactions on the Internet. The amount of data in this global information space is increasing far more rapidly than the ability of an individual information user or online customer to process it. As a result, information overload has become a critical challenge facing individuals, giving rise to an e-service opportunity: developing recommendations that are typically personalized for individuals. In providing this service, merchandisers seek not only to increase sales, but also to better manage customer relationships that lead to higher loyalty and greater competitive barriers. Recommendation systems (also known as recommender systems) have emerged as a new class of e-service product, to address the challenge of information overload by suggesting information or products of greatest interest to users (from the information user or online customer perspective) and to facilitate the management of customer relationships or even to provide personalized services to customers (from the merchandiser or information provider perspective).

Recommendation systems typically suggest items (information, products or services) that are of interest to users based on customer demographics, features of items, and/or user preferences (e.g., ratings or purchasing history). In the context of content-based sites, recommendation systems can be used to facilitate selective dissemination from a large stream of information to users or to support effective management of information on the part of individuals. For instance, GroupLens provides personalized recommendations of Usenet news items from a high-volume, high-turnover discussion list service on the Internet, based on the opinions of other users (Resnick et al., 1994; Konstan et al., 1997). Using the ratings of news articles evaluated by users, GroupLens identifies other users whose information needs or tastes are similar to those of a given user and recommends news articles that they have liked. Siteseer, developed by Rucker and Polanco (1997), is a Web-page recommendation system that uses an individual’s bookmarks and the organization of bookmarks within folders for predicting and recommending relevant pages. Essentially, Siteseer looks at each user’s folders and bookmarks, and measures the degree of overlap of each folder with other people’s folders. Accordingly, Siteseer forms dynamically defined virtual communities of interest, particular to each user and specific to each of the user’s categories of interest. Siteseer provides as recommendations those pages that have been bookmarked by the user’s virtual neighbors, giving preference to pages drawn from folders with the
highest overlap as well as those held within multiple folders in the neighborhood. The digital library of
ACM (www.acm.org/dl), on the other hand, suggests to a user new binders to be included in his/her
personal bookshelf based on binders of other users who share similar professional memberships or
subscriptions.

Recommendation systems have also seen considerable success on E-commerce sites to suggest
products to their customers. An early entrant into the e-service area, and one that capitalized on a new
form of customer data, the navigation and browsing habits of customers across many different sites, is
double-click.com, which provides the service of pushing personalized advertisements into ad slots on web
sites. The data are acquired from cookie files, which track the movements of customers through many
web sites, and can be used to deduce customers’ product preferences. Recently, recommendation systems
have been used, for example, by Bostondine to recommend restaurants in and around Boston, by Sepia
Video Guide to make customized video recommendations, by Movie Critic, MovieFinder.com, and
Morse to recommend movies, by CDNow.com to recommend albums related to the album located by a
user, and by amazon.com to recommend books (Schafer, et al., 1999; Ansari, et al., 2000). A well-known
example of this practice is Amazon.com’s service of offering a personalized recommendation based on a
customer’s past purchases and interests, and on the preferences of similar customers. Such
recommendations provide a customer service whose quality is commensurate with the customer’s
assessment of the value or appropriateness of the recommendation. If the quality of this service is high,
the recommendations may lead to increased customer satisfaction and loyalty, as well as increased sales.

The difficulty and importance of making a quality recommendation underlie the development of a
wide range of technical approaches to developing recommendations, a survey of which is the main focus
of this chapter. The remainder of the chapter is organized as follows. In Section 2, a framework for
recommendation systems will first be presented. Section 3 proposes a taxonomic classification for
recommendation systems whose differences will be illustrated using the proposed framework. Various
recommendation approaches will be detailed in Sections 4-7. Finally, a summary of the chapter and
important practical considerations and research issues will be provided in Section 8.

2. FRAMEWORK FOR RECOMMENDATION SYSTEMS

Recommendation systems are used to determine a set of items (information, products or services) that are
of greatest interest to users. As shown in Figure 1, in a typical recommendation scenario, there is a set of
n users \( U = \{u_1, u_2, \ldots, u_n\} \) and a set of m items \( I = \{i_1, i_2, \ldots, i_m\} \). Each user \( u_i \) may be associated with
his/her demographic data and has a list of items \( I_{u_i} \) (where \( I_{u_i} \subseteq I \) and \( I_{u_i} \) can be an empty set) on which the
user has expressed his/her preferences. The preference of a user \( u_i \) on an item \( i_j \) (denoted as \( p_{ij} \)) can be a subjective rating explicitly stated by the user or an implicit measure inferred from purchase, browsing and navigation data in user activity. On the other hand, a set of features may be used to describe each item \( i_j \in I \). Thus, a recommendation system is often based on user demographics, features of items, and/or user preferences for deriving its recommendation decisions. Although other types of information may be adopted by some recommendation systems, we limit our discussions on these three information types commonly used for reaching recommendation decisions. Regarding the recommendation decision itself, it can be made for a specific user \( u_a \) (where \( u_a \in U \)), called an active user, on those items that have not explicitly been rated or chosen by this user. Alternatively, it may suggest a new item \( i_{new} \) (where \( i_{new} \notin I \)) to those users who might be interested. In the following subsections, potential information for and types of recommendation decisions will be discussed in detail.

2.1 Information for Recommendation Decisions
User demographics, referring to the characteristics of users that potentially affect their likes and dislikes, represent one type of information on which recommendation decisions can be made. User demographics typically include such attributes as age, gender, occupation, income level, education level, hobbies, and so
on. For instance, based on the heuristics “teenage boys like basketball” and “medical physicians like golf,” items related to basketball and golf can be suggested to customers with those demographic characteristics, respectively.

A second type of information that can be used for making recommendation decisions is the set of attributes of the items themselves. Items can be described by their features, extrinsic or intrinsic. Extrinsic features (e.g., color, brand, manufacturer, and sub-category/category of a product) refer to those that are difficult or impossible to derive by automatically analyzing the content of items and, if required, need to be supplied by other sources (e.g., domain experts). Conversely, intrinsic features are derivable by the analysis of the content of items, such as representative keywords extracted from news articles or web pages. It is worth mentioning that information about category hierarchy of items, if needed, can also be captured in the feature set for describing the items.

The third type of information relevant to recommendation decisions is user preferences. The user preference score\(^1\) on an item can be a binary measure (of preference or choice) or on a discrete numerical scale (representing degree of preference). User preferences can be explicit ratings provided by users or implicit measures inferred from available user activity data (Herlocker et al., 1999; Claypool et al., 2001). Explicit ratings are directly obtained from users by instructing them to evaluate and rate each item typically on a numerical scale. Even though explicit ratings are often observed in real-world practices, requiring users to provide explicit ratings may alter users’ normal patterns of browsing in online settings. Unless users perceive potential benefits, they may not have any incentive to provide explicit ratings. As manifested by a study conducted by Sarwar et al. (1998), users are reading a lot more articles than they are rating. Thus, explicit ratings may not be as reliable as often presumed (Claypool et al., 2001). On the other hand, implicit ratings are derived from data sources such as purchase history, web logs, or cookie data for user browsing and navigation patterns (e.g., time spent reading or the amount of scrolling on a Web page), thereby leveraging data already collected for other purposes (Breese et al., 1998, Herlocker et al., 1999; Claypool et al., 2001). Implicit ratings may be less accurate than explicit ones for measuring users preferences. Nevertheless, there is evidence that the time spent on a page, the amount of scrolling on a page and their combination may strongly correlate with explicit ratings (Claypool et al., 2001).

Any given recommendation system may use all or some part of the three types of information shown

\(^1\) In many studies, preference scores are often referred to as ratings. However, in this chapter, we prefer preference scores to ratings since the former covers both explicit ratings and implicit ratings, while the latter typically implies subjective ratings (i.e., explicit ratings).
in Figure 1. In fact, the review of existing recommendation systems shows that each of them only uses a subset of information to arrive recommendations. No system uses all the three types of information for recommendations.

### 2.2 Types of Recommendation Decisions

A recommendation decision may suggest a set of items from among those that have not explicitly been rated or chosen by an active user \( u_a \). Alternatively, it may select a subset of users that may be interested in a newly available item \( i_{\text{new}} \). Accordingly, three types of recommendation decisions can be:

- **Prediction**: Prediction expresses the predicted preference for item \( i_j \notin I_{u_a} \) for an active user \( u_a \). This predicted value is within the same scale as for the user preferences (Sarwar et al., 2001).

- **Top-N recommendation**: It is a list of \( N \) items, \( I_r \subset I \), that the active user \( u_a \) will like the most. The recommended list must be on items not already rated or chosen by \( u_a \), i.e., \( I_r \cap I_{u_a} = \emptyset \) (Sarwar et al., 2001).

- **Top-M users**: For a newly available item \( i_{\text{new}} \), a recommendation system suggests a list of \( M \) users who will like \( i_{\text{new}} \) most. In some recommendation approaches, the recommendation on the top-M users for a new item can be viewed as a set of prediction decisions, each of which is made for a user. Subsequently, the top \( M \) users who are predicted to like \( i_{\text{target}} \) most are recommended.

The first two types of recommendations (prediction and top-N recommendation) can be further categorized along a degree-of-personalization dimension. A recommendation decision can be **personalized** to specific users or **non-personalized** if it is independent of users. For instance, the average customer ratings displayed by Amazon.com and MovieFinder.com are non-personalized top-N recommendations, while Amazon.com’s book recommendations that are based on a customer’s past purchases and interests are personalized top-N recommendations.

### 3. CLASSIFICATION OF RECOMMENDATION SYSTEMS

Many recommendation systems have been proposed in the literature. Based on the type of data and technique used to arrive at recommendation decisions, recommendation systems can broadly be classified into the following approaches:

1. **Popularity-based recommendation approach**: In real-world practices, customers often want to know about the most popular items within a given community as a means of finding out what they should pay attention to. This has precedents in best-seller lists in physical store settings. Using user preferences as input, the popularity-based recommendation approach computes within-community
popularity measures (e.g., percentage of customers who purchased an item) or summaries (e.g., number of customers who purchased an item, average ratings for an item, etc.) (Schafer et al., 2001). Thus, the most popular items are recommended to users. The popularity-based recommendation approach serves to clarify the sense in which personalization techniques only form a subset of recommendation systems, since this approach does provide recommendations without personalization. Although this approach can only deliver non-personalized recommendations, it is popular due to its simplicity and efficiency.

2. **Content-based recommendation approach:** The content-based recommendation approach rests on the notion that the features of items can be useful in recommending items. It conforms to content-based information filtering that assumes that the degree of relevance (to a particular user) of an item can be determined by its content (represented by its features) (Alaspector et al., 1998). The content-based recommendation approach tries to recommend items similar to those a given user has liked in the past (Balabanovic and Shoham, 1997; Herlocker et al., 1999). Thus, the features of items and a user’s own preferences are the only factors influencing recommendation decisions for the user with this approach.

3. **Collaborative filtering recommendation approach:** The collaborative filtering recommendation approach is also called social filtering or the user-to-user correlation recommendation approach. A collaborative filtering system identifies users whose tastes are similar to those of a given user and recommends items they have liked (Balabanovic and Shoham, 1997). Users of a collaborative filtering system share their opinions regarding items that they consume so that other users of the system can better decide which items to consume (Herlocker et al., 1999). With this method, user preferences are the sole input to recommendation decisions.

4. **Association-based recommendation approach:** The association-based recommendation approach relies on user preferences to identify items frequently found in association with items which a user has chosen, or for which a user has expressed interest in the past (Schafer et al., 2001). Item-associations can take the form, for example, of a set of items that have been rated as similar to a particular item, or of co-occurrence of items that users often preferred or purchased in common. Such item-associations, once identified, can then be employed to recommend items to users. For instance, the prediction of the preference score of an active user on an item can be based on the active user’s preference scores over similar items.

5. **Demographics-based recommendation approach:** This approach recommends items to a user based on the preferences of other users with similar demographics. Unlike other recommendation approaches in which recommendations are made at the item level, a demographics-based recommendation system typically generates recommendations at the more general category level. As such, this approach involves learning and reasoning with relationships between user demographics
and expressed category preferences, where the expressed category preferences of a user are derived from individual user preferences stated previously and the category hierarchies of items.

6. **Reputation-based recommendation approach**: This approach focuses on identifying users that a user respects (directly or transitively) and then using the opinions of these selected individuals as a basis for recommendations. The reputation-based recommendation approach has its roots in social practices involving reviewers or opinion leaders (Lynch 2001). It captures a totally different aspect of socially based information discovery, where one makes assessments about people in a more general way rather than directly comparing preferences.

Table 1 summarizes the characteristics of each recommendation approach. The popularity-based recommendation approach involves simple techniques for recommendations. On the other hand, the reputation-based recommendation approach is still in its conception stage. To the best of our knowledge, although few reputation mechanisms have been proposed for electronic marketplaces (Zacharia et al., 2000), no existing recommendation systems have incorporated a reputation mechanism for delivering reputation-based recommendations. Thus, in the following sections, we will only review representative techniques for the remaining recommendation approaches.

Each recommendation approach has its strengths and limitations. To complement one’s limitations by another’s strengths, several hybrid techniques have been proposed. For example, Fab, proposed by Balabanovic and Shoham (1997) is a hybrid content-based, collaborative filtering recommendation system for suggesting Web pages to users. Sarwar et al. (2000) propose a recommendation technique that integrates collaborative filtering and association-based approaches. Due to space limitation, hybrid techniques will not be discussed in further detail.
Table 1: Characteristics of Different Recommendation Approaches

<table>
<thead>
<tr>
<th>Recommendation Approach</th>
<th>Information Used</th>
<th>Types of Recommendations</th>
<th>Degree of Personalization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Popularity-based</td>
<td>User preferences</td>
<td>Top-N recommendation</td>
<td>Non-personalized</td>
</tr>
<tr>
<td>Content-based</td>
<td>Features of items and individual user preferences</td>
<td>Prediction, top-N recommendation, and top-M users</td>
<td>Personalized</td>
</tr>
<tr>
<td>Collaborative Filtering</td>
<td>User preferences</td>
<td>Prediction and top-N recommendation</td>
<td>Personalized</td>
</tr>
<tr>
<td>Association-based</td>
<td>User preferences</td>
<td>Prediction and top-N recommendation</td>
<td>Personalized</td>
</tr>
<tr>
<td>Demographics-based</td>
<td>User demographics, individual user preferences, and features of items (specifically, category hierarchy of items)</td>
<td>Prediction, top-N recommendation, and top-M users</td>
<td>Personalized</td>
</tr>
<tr>
<td>Reputation-based</td>
<td>User preferences and reputation matrix</td>
<td>Top-N recommendation and, possibly prediction</td>
<td>Personalized</td>
</tr>
</tbody>
</table>

4. CONTENT-BASED RECOMMENDATION APPROACH

For a given user, content-based recommendation systems recommend items similar to those the user has liked in the past (Balabanovic and Shoham, 1997; Herlocker et al., 1999). The content-based approach automatically learns and adaptively updates the profile of each user. Given a user profile, items are recommended for the user based on a comparison between item feature weights and those of the user profile. If a user rates an item differently than a recommendation system suggested, the user profile can be updated accordingly. This process is also known as relevance feedback. The content-based recommendation approach has its roots in content-based information filtering, and has proven to be effective in recommending textual documents. Examples of the content-based recommendation systems include Syskill & Webert for recommending Web pages (Pazzani et al., 1996), NewsWeeder for recommending news-group messages (Lang, 1995), and InformationFinder for recommending textual documents (Krulwich and Burkey, 1996).

Assume the set of items that a user has rated or chosen be the training set with respect to the given user. As shown in Figure 2, the phases involved in a content-based recommendation system generally include:

1. Feature Extraction and Selection: Extract and select relevant features for all items in the collection.
2. Representation: Represent each item with the feature set determined in the previous phase.
3. User Profile Learning: Automatically learn or adaptively update the user profile model for each user.
4. Recommendation Generation: Generate recommendations by performing reasoning on the corresponding user profile model.

![Diagram of Content-based Recommendation Approach]

**Figure 2: Process of Content-based Recommendation Approach**

### 4.1 Feature Extraction and Selection

The feature extraction and selection phase is undertaken to determine a set of features that will be used for representing individual items. If items involve extrinsic features, they need to be specified by domain experts. For example, Alspector et al. (1998) developed variants of content-based recommendation systems for movie selection based on such features as category (e.g., comedy, drama, etc.), MAPP rating, Maltin rating, Academy Award, length, origin, and director of movies. However, if intrinsic features are involved, extraction of features by analyzing the content of items is required. An automatic feature extraction mechanism is only available for limited domains. In the domain consisting of textual documents, the most effective domain of the content-based recommendation approach, the text portion of the documents is parsed to produce a list of features (typically consisting of nouns or noun phrases) none of which is a number, part of a proper name, or belongs to a pre-defined list of stop words.

After the feature specification (for extrinsic features) or extraction (for intrinsic features), feature selection is initiated to choose a small subset of features that ideally is necessary and sufficient to describe the target concept (Piramuthu, 1998). The feature selection process not only improves learning efficiency but also has the potential to increase learning effectiveness (Dumais et al., 1998). Various feature selection methods have been proposed, using such techniques as statistical analysis, genetic algorithms, rough sets theory, and so on. For example, in statistical analysis, forward and backward stepwise multiple regression are widely used to select features. In the forward stepwise multiple regression, the analysis proceeds by adding features to a subset until the addition of a new feature no longer results in an improvement in the explained variance ($R^2$ value). The backward stepwise multiple regression starts with the full set of features and seeks to eliminate features with the smallest contribution to $R^2$ value (Kittler, 1975). Siedlecki and Sklansky (1989) adopted genetic algorithms for feature selection by encoding the initial set of $f$ features as $f$-element bit string with 1 and 0 representing the presence and absence respectively of features in the set, with classification accuracy employed as the fitness function.
Modrzejewski (1993) proposed a rough set-based feature selection method to determine the degree of dependency of sets of attributes for selecting binary features. Features resulting in a minimal preset decision tree, with minimal length of all paths from root to leaves, are selected. For interested readers, a summary of and empirical comparisons on various feature selection methods can be found in (Piramuthu, 1998).

However, in the case of recommending textual documents, hundreds or thousands of features can be extracted, and the feature selection methods described above may become computationally infeasible. Thus, most feature selection methods developed for textual documents adopt an evaluation function that is applied to features independently. A feature selection metric score is then assigned to each feature under consideration. The top $k$ features with the highest feature selection metric score are selected as features for representing documents, where $k$ is a predefined number of features to select. Several evaluation functions for feature selection have been proposed, including TF (within-document term frequency), TF×IDF (within-document term frequency × inverse document frequency), correlation coefficient, mutual information, and a $\chi^2$ metric (Dumais et al., 1998; Lam and Ho, 1998; Lewis and Ringuette, 1994; Ng et al., 1997).

4.2 Representation
In the representation phase, each item is represented in terms of features selected in the previous phase. Each item in the training set is labeled to indicate its preference (dependent variable) by a particular user and assigned a value for each feature (independent variable) selected. The task of representing extrinsic features of an item is straightforward and is essentially achieved during the feature extraction and selection phase. Feature-values of an item originally supplied by domain experts are used. On the other hand, for representing a textual document by a set of intrinsic features extracted and selected in the previous phase, a binary value (e.g., indicating whether or not the feature appears in the document) or a numerical value (e.g., frequency of occurrence in the document being processed) is assigned to each feature. Different document representation schemes have been proposed, including binary, TF, IDF and TF×IDF (Yang and Chute, 1994).

4.3 User Profile Learning
For each user, the purpose of this phase is to construct a user profile model for establishing the relationship between preference scores (dependent variable) and feature-values (independent variables) from the training examples pertinent to the user. The learning implementation can draw on statistical, inductive learning, and Bayesian probability methods. For example, Alspector et al. (1998) adopted the
statistical method (specifically, a multiple linear regression model) and inductive learning algorithm (specifically, CART) for movie recommendations. Mooney and Roy (2000) used the Bayesian probability method for learning user profiles in order to obtain book recommendations.

A multiple linear regression model is based on the most natural assumption of a linear influence of each of the features involved on the preferences. Thus, it takes the form of:

\[ p_{im} = \sum_{j=1}^{k} w_j f_{mj} + b \]

where \( p_{im} \) denotes the preference score of the user \( i \) on the item \( m \),

\( w_j \) is the coefficient associated with the feature \( j \),

\( f_{mj} \) is the value of the \( j \)th feature for the item \( m \), and

\( b \) represents the bias.

Creation of such a user profile model for each user is essentially equivalent to a multiple linear regression on the set of features and its solution can be obtained using the least-squares technique (Alspector et al., 1998).

To address the potential nonlinear dependencies between individual features, inductive learning algorithms have been adopted for learning user profiles in the content-based recommendation approach. In this inductive learning framework, preference scores on items in the training set can be treated as a continuous decision or a discrete class membership, while the features of the item are attributes potentially affecting the decision. Consequently, a decision tree induction algorithm (e.g., ID3 (Quinlan, 1986) or its descendant C4.5 (Quinlan, 1993), CHAID (Kass, 1980), or CART (Breiman et al., 1984)), a decision rule induction algorithm (e.g., CN2 (Clark and Niblett, 1989)), or a backpropagation neural network (Rumelhart et al., 1986) can be employed to address the target learning task.

A decision tree induction algorithm is a supervised learning method that constructs a decision tree from a set of training examples. It typically adopts an iterative method to construct decision trees, preferring simple trees to complex ones, based on the theory that simple trees are more accurate classifiers for future instances. On the other hand, a decision rule induction algorithm induces from training examples a set of decision rules, each of which is represented by an “if <complex> then predict <class>” format, where <complex> is a conjunct attribute test and <class> represents a decision outcome. The rule construction proceeds in an iterative fashion and each iteration searches for a better and significant complex for predicting a decision class via a general-to-specific or specific-to-general search.
A backpropagation neural network is a multi-layered, fully-connected network (Rumelhart et al., 1986). The network has one input layer, one or more hidden layers, and one output layer. Typically, each layer contains multiple nodes, each of which is connected to every node in the next adjacent layer through a weighted link. In a backpropagation neural network, training is achieved by adjusting its weights each time it sees an input-output pair (i.e., a training example). Each iteration requires a forward pass and a backward pass. The forward pass involves presenting the training example to the network and letting activations flow until they reach the output layer. During the backward pass, the actual output of the network produced from the forward pass is compared with the target output of the training example and error estimates are computed for the output nodes. The error estimates of the output nodes are then employed to derive error estimates for the hidden nodes. Accordingly, the weights connected to the output nodes can be adjusted in order to reduce those errors in output layer. Finally, errors are propagated back to the weights stemming from the input nodes. Through repeatedly presenting the set of training examples to the network, the weights between nodes are expected to converge. As a result, the user profile model is encoded in the weights within the selected network topology.

### 4.4 Recommendation Generation

Once user profile models are induced, recommendations can be generated. Since the features of items and a user’s past preferences are the only factors influencing recommendation decisions, all of the three types of recommendations can be suggested. To estimate the predicted preference score on item $i_j \notin I_{ua}$ for an active user $u_a$, the item is first represented with the features selected previously. Subsequently, the reasoning on the user profile model (e.g., a regression model, a decision tree, a set of decision rules, or a trained backpropagation neural network) corresponding to the active user is performed to predict the preference score of $u_a$ on the item $i_j$. To produce the top-$N$ recommendation for the active user $u_a$, the predicted preference score on each item that has not explicitly been rated or chosen by $u_a$ is obtained as described previously. Afterward, the top $N$ items with the highest predicted preference score are included in the recommendation list. To recommend the top-$M$ users for a new item $i_{new}$, a set of prediction decisions are made, one for each user, followed by the selection of the top $M$ users who are predicted to like $i_{new}$ most.

### 4.5 Summary

The content-based recommendation approach recommends for a given user items similar to those the user has liked in the past. Since individualized user profiles are induced, personalized recommendations can be
achieved. Due to the relevance feedback process, a content-based recommendation system can adaptively update the profile of each user. As mentioned, items are recommended based on features of items rather than on the preferences of other users. This allows for the possibility of providing explanations that list content features that caused an item to be recommended, potentially giving readers confidence in the system’s recommendations and insight into their own preferences (Mooney and Roy, 2000).

However, the content-based recommendation approach has several shortcomings. In many domains, the items are not amenable to any useful feature extraction methods (e.g., movies, music albums, and videos). For such domains, the efforts of domain experts to specify for extrinsic features and to assign feature-values for each item are unavoidable, thus limiting the applicability of content-based recommendation approach. Furthermore, over-specialization is another problem associated with this approach. When the system can only recommend items scoring highly against a user’s profile, the user is restricted to seeing items similar to those the user has liked in the past (Balabanovic and Shoham, 1997).

5. COLLABORATIVE FILTERING RECOMMENDATION APPROACH

The collaborative filtering recommendation approach is very different from the content-based one. Rather than recommending items because they are similar to items a user has liked in the past, the collaborative filtering approach recommends items based on the opinions of other users. Typically, by computing the similarity of users, a set of “nearest neighbor” users whose known preferences correlate significantly with a given user are found. Preferences for unseen items are predicted for the user based on a combination of the preferences known from the nearest neighbors. Thus, in this approach, users share their preferences regarding each item that they consume so that other users of the system can better decide which items to consume (Herlocker et al., 1999). The collaborative filtering approach is the most successful and widely adopted recommendation technique to date. Examples of collaborative filtering systems include GroupLens (Resnick et al., 1994; Konstan et al., 1997), the Bellcore video recommender (Hill et al., 1995), and Ringo (Shardanand and Maes, 1995). Amazon.com also uses a form of collaborative filtering technology, though the specifics of their implementation are not published.

As mentioned, the collaborative filtering approach utilizes user preferences to generate recommendations. Several different techniques have been proposed for collaborative filtering recommendations, including neighborhood-based, Bayesian networks (Breese et al., 1998), singular value decomposition with neural net classification (Billsus and Pazzani, 1998), and induction rule learning (Basu et al., 1998). Due to space limitation, we will only review the neighborhood-based collaborative filtering techniques since they are the most prevalent algorithms used in collaborative filtering for
recommendation. As shown in Figure 3, the process of a typical neighborhood-based collaborative filtering system can be divided into three phases (Sarwar et al., 2000):

1. **Dimension Reduction**: Transform the original user preference matrix into a lower dimensional space to address the sparsity and scalability problems.

2. **Neighborhood Formation**: For an active user, compute the similarities between all other users and the active user and to form a proximity-based neighborhood with a number of like-minded users for the active user.

3. **Recommendation Generation**: Generate recommendations based on the preferences of the set of nearest neighbors of the active user.

![Figure 3: Process of Collaborative Filtering Recommendation Approach](image)

### 5.1 Dimension Reduction

The dimension reduction phase transforms the original user preference matrix into a lower dimensional space to address the sparsity and scalability problem often encountered in collaborative filtering recommendation scenarios. The original representation of the input data to a collaborative filtering system is an $n \times m$ user preference matrix, where $n$ is the number of users and $m$ is the number of items. This representation may potentially pose sparsity and scalability problems for collaborative filtering systems (Sarwar et al., 2000). In practice, when a large set of items are available, users may have rated or chosen a very low percentage of items, resulting in a very sparse user preference matrix. As a consequence, a collaborative filtering recommendation system may be unable to make any recommendations for a particular user. On the other hand, a collaborative filtering recommendation system requires the user similarity computation that grows with $n$ and $m$, and thus, suffers serious scalability problem.

To overcome the described problems associated with the original representation, the sparse matrix can be transformed into a lower dimensional representation using the Latent Semantic Indexing (LSI) method (Sarwar et al., 2000). Essentially, this approach uses a truncated singular value decomposition to obtain a rank-$d$ approximation of the original $n \times m$ user preference matrix. This reduced representation alleviates the sparsity problem as all the entries in the $n \times d$ matrix are nonzero, which means that all $n$ customers now have their preferences on the $d$ meta-items. Moreover, the performance on computing user similarities and its scalability are improved dramatically as $d \ll m$ (Sarwar et al., 2000).
5.2 Neighborhood Formation

The goal of neighborhood formation is to find, for an active user $u_a$, an ordered list of $l$ users $N = \{n_1, n_2, \ldots, n_l\}$ such that $u_a \not\in N$ and $\text{sim}(u_a, n_i) \geq \text{sim}(u_a, n_j)$ for $i < j$. This phase is in fact the model-building process for the collaborative filtering recommendation approach. Several different similarity measures have been proposed (Shardanand and Maes, 1995; Herlocker et al., 1999; Sarwar et al., 2000), including

- **Pearson correlation coefficient**: The Pearson correlation coefficient is the most commonly used similarity measure in collaborative filtering recommendation systems. It is derived from a linear regression model. The similarity between an active user $u_a$ and another user $u_b$ using the Pearson correlation coefficient is calculated as:

$$
\text{sim}(u_a, u_b) = \frac{\sum_i (p_{ai} - \bar{p}_a)(p_{bi} - \bar{p}_b)}{\sqrt{\sum_i (p_{ai} - \bar{p}_a)^2} \sqrt{\sum_i (p_{bi} - \bar{p}_b)^2}}
$$

where $p_{ai}$ represents the preference score of the user $u_a$ on item $i$,

$\bar{p}_a$ is the average preference score of the user $u_a$, and

$m$ is the number of items or meta-items in the reduced representation.

- **Constrained Pearson correlation coefficient**: The constrained Pearson correlation coefficient takes the positivity and negativity of preferences into account (Shardanand and Maes, 1995). A preference score below the midpoint of the rating scale (e.g., 4 in a 7-point rating scale) is considered as negative, while a preference score above the midpoint is positive. Accordingly, the constrained Pearson correlation coefficient is used so that only when both users have rated an item positively or both negatively, the correlation coefficient between them will increase. The similarity between an active user $u_a$ and another user $u_b$ using the constrained Pearson correlation coefficient is given as:

$$
\text{sim}(u_a, u_b) = \frac{\sum_i (p_{ai} - mp)(p_{bi} - mp)}{\sqrt{\sum_i (p_{ai} - mp)^2} \sqrt{\sum_i (p_{bi} - mp)^2}}
$$

where $mp$ is the midpoint of the rating scale.

- **Spearman rank correlation coefficient**: The Spearman rank correlation coefficient, a nonparametric method, computes a measure of correlation between ranks instead of actual preference scores:

$$
\text{sim}(u_a, u_b) = \frac{\sum_i (\text{rank}_{ai} - \text{rank}_a)(\text{rank}_{bi} - \text{rank}_b)}{\sqrt{\sum_i (\text{rank}_{ai} - \text{rank}_a)^2} \sqrt{\sum_i (\text{rank}_{bi} - \text{rank}_b)^2}}
$$

- **Cosine similarity**: Two users $u_a$ and $u_b$ are considered as two vectors in the $m$ dimensional item-space or in the $d$ dimensional meta-item-space in the reduced representation. The similarity between them is measured by computing the cosine of the angle between the two vectors, which is given by:
\[
\text{sim}(u_a, u_b) = \cos(\bar{\mathbf{a}}, \bar{\mathbf{b}}) = \frac{\bar{\mathbf{a}} \cdot \bar{\mathbf{b}}}{\|\bar{\mathbf{a}}\|_2 \|\bar{\mathbf{b}}\|_2} = \frac{\sum_{i=1}^{m} p_{ai} \cdot p_{bi}}{\sqrt{\sum_{i=1}^{m} p_{ai}^2} \sqrt{\sum_{i=1}^{m} p_{bi}^2}}
\]

- **Mean-squared difference:**

The mean-squared difference, introduced in Ringo (Shardanand and Maes, 1995), measures the dissimilarity between an active user \(u_a\) and another user \(u_b\) as:

\[
dissim(u_a, u_b) = \sum_{i=1}^{m} (p_{ai} - p_{bi})^2
\]

According to an empirical evaluation study conducted by Herlocker et al. (1999), the Pearson correlation coefficient, whose performance was similar to that of the Spearman correlation coefficient, outperformed the cosine similarity and the mean-squared difference. Shardanand and Maes (1995) empirically evaluated different similarity measures (including Pearson correlation coefficient, constrained Pearson correlation coefficient and mean-squared difference) and suggest that the constrained Pearson correlation coefficient achieved the best performance in terms of the tradeoff between the prediction accuracy and the number of target values that can be predicted. On the other hand, the mean-squared difference outperformed its counterparts in prediction accuracy, but it produced fewer predictions than others did.

After the \(n \times n\) similarity matrix is computed for \(n\) users using a desired similarity measure, the next task is to actually form the neighborhood for the active user. There are several schemes for neighborhood selection (Herlocker et al, 1999; Sarwar et al., 2000), including:

- **Weight thresholding:** This scheme, used by Shardanand and Maes (1995), is to set an absolute correlation threshold, where all neighbors of the active user with absolute correlations greater than the given threshold are selected.
- **Center-based best-k neighbors:** It forms a neighborhood of a pre-specified size \(k\), for the active user, by simply selecting the \(k\) nearest users.
- **Aggregate-based best-k neighbors:** The aggregate-based best-\(k\) neighbors scheme, proposed by Sarwar et al. (2000), forms a neighborhood of size \(k\) for the active user \(u_a\) by first selecting the closest neighbor to \(u_a\). The rest \(k-1\) neighbors are selected as follows. Let, at a certain point there are \(j\) neighbors in the neighborhood \(N\), where \(j < k\). The centroid of the current neighborhood is then determined as:

\[
\bar{C} = \frac{1}{j} \sum_{\bar{V} \in N} \bar{V}
\]

A user \(w\), such that \(w \notin N\) is selected as the \(j+1\)-st neighbor only if \(w\) is closest to the centroid \(\bar{C}\). Subsequently, the centroid is recomputed for \(j+1\) neighbors and the process
continues until $|N| = k$. Essentially, this scheme allows the nearest neighbors to affect the formation of the neighborhood and can be beneficial for very sparse data sets (Sarwar et al., 2000).

5.3 Recommendation Generation

After the nearest neighbors of the active user are identified, subsequent recommendations can be generated. Since the collaborative filtering process is initiated for a particular user, the collaborative filtering recommendation approach is typically for prediction and top-$N$ recommendation decisions. The collaborative filtering recommendation approach is not applicable to produce the top-$M$ users for a newly available item since no user preference is available for this item.

To estimate the predicted preference score on the item $i_j \not\in I_{ua}$ for an active user $u_a$, the following methods can be employed:

1. **Weighted average**: To combine all the neighbors’ preference scores on the item $i_j$ into a prediction, the weighted average method is to compute a weighted average of the preference scores, using the correlations as the weights. This basic weighted average method, as used in Ringo (Shardanand and Maes, 1995), makes an assumption that all users rate on approximately the same distribution.

2. **Deviation-from-mean**: The method, taken by GroupLens (Resnick et al., 1994; Konstan et al., 1997), is based on the assumption that users’ preference score distribution may center on different points. To account for the differences in means, the average deviation of a neighbor’s preference score from that neighbor’s mean preference score is first computed, where the mean preference score is taken over all items that the neighbor has rated. The average deviation from the mean computed across all neighbors is then converted into the active user’s preference score distribution by adding it to the active user’s mean preference score. Using the deviation-from-mean method, the predicted preference score of the active user $u_a$ on the item $i$ is calculated as:

$$ p_{ai} = \bar{p}_a + \frac{\sum_{i=1}^{k} (p_{ai} - \bar{p}_a) \cdot \text{sim}(u_a, u_i)}{\sum_{i=1}^{k} \text{sim}(u_a, u_i)} $$

3. **Z-score average**: To take into account the situation where the spread of users’ preference score distributions may be different, the z-score average method was proposed by Herlocker et al. (1999) by extending the deviation-from-mean method. In this method, neighbors’ preference scores on the item $i$ are converted to z-scores and a weighted average of the z-scores are derived as the predicted preference score of the active user $u_a$ on the item $i$:

$$ p_{ai} = \bar{p}_a + \frac{\sum_{i=1}^{k} \left( \frac{p_{ai} - \bar{p}_a}{\sigma_u} \right) \cdot \text{sim}(u_a, u_i)}{\sum_{i=1}^{k} \text{sim}(u_a, u_i)} $$
An empirical evaluation study conducted by Herlocker et al. (1999) showed that the deviation-from-mean method performed significantly better than the weighted average method. However, the z-score average method did not perform significantly better than the deviation-from-mean method, suggesting that differences in spread between users’ preference score distributions might have no effect on prediction accuracy.

To produce the top-N recommendation for the active user $u_a$, the predicted preference score on each item that has not explicitly been rated or chosen by $u_a$ is derived first. Afterward, the top $N$ items with the highest predicted preference score are included in the recommendation list.

5.4 Summary
The collaborative filtering approach delivers personalized recommendations based on the opinions of other users and provides several advantages that are not provided by the content-based recommendation approach (Balabanovic and Shoham, 1997; Herlocker et al., 1999). By using other users’ opinions rather than features of items, the collaborative filtering approach can be employed to recommend items whose content is not easily analyzed by automated feature extraction techniques. Furthermore, this approach is capable of recommending items on the basis of quality and taste. Finally, since other users’ opinions influence what is recommended, the approach is able to provide serendipitous recommendations to a user (i.e., recommend items that are dissimilar to those the user has liked before); thus avoiding the over-specialization problem associated to the content-based recommendation approach.

However, in addition to the sparsity and scalability problems, the collaborative filtering approach incurs other problems of its own. Items that have not been rated or chosen by a sufficient number of users cannot be effectively recommended. Thus, the collaborative filtering approach potentially tends to recommend popular items (Mooney and Roy, 2000). On the other hand, although newly available items are frequently of particular interest to users, it is impossible for the collaborative filtering approach to recommend those items that no one has yet rated or chosen (Balabanovic and Shoham, 1997; Condliff et al., 1999; Mooney and Roy, 2000). Furthermore, for a user whose tastes are unusual compared to the rest of the population, there will not be any other users who are particularly similar, leading to poor recommendations (Condliff et al., 1999). Finally, different items may be highly similar in their features. The collaborative filtering approach cannot find this latent association and treats these items differently (i.e., the synonym problem). Thus, the lack of access to the content of the items prevents similar users from being matched unless they have rated the exact same items (Sarwar et al., 2000).
6. ASSOCIATION-BASED RECOMMENDATION APPROACH

The association-based recommendation approach relies on user preferences to identify items frequently associated with those items which a user has expressed interest in or chosen in the past (Schafer et al., 2001). Depending on the technique used for such association discovery, item-associations can be classified into two types: item-correlations and association rules. For a target item, an item-correlation technique searches for items that have been rated as similar to the target item. Once the set of similar items is identified, the predicted preference score of an active user on the target item is then based on the active user’s preference scores for these similar items (Schafer et al., 2001). Apparently, item-correlation techniques are best applicable to the multi-point scaling scheme used for describing user preferences.

Alternatively, the association rule discovery technique (Agrawal et al., 1993; Agrawal and Srikant, 1994) can be adopted to find interesting co-occurrences of items in a set of transactions, where each transaction in this recommendation context corresponds to a distinct user and consists of a list of items that the user liked or purchased. Since the association rule discovery technique is concerned mainly with the co-occurrence of items, the user preferences need to be transformed into the described representation of transactions. An association rule is an implication of the form $X \Rightarrow Y$, which represents the notion that if the set of items $X$ occurs, another set of items $Y$ will often occur. To recommend items to an active user, if his/her transaction supports the left-hand side ($X$ in the previous example) but not the right-hand side ($Y$ in the previous example) of an association rule, the set of items $Y$ will be recommended. In the following subsections, the item-correlation and association rule techniques for recommendations are detailed.

6.1 Item-Correlation Techniques for Recommendations

Taking user preferences as input, an item-correlation technique searches for a set of items that have been rated as similar to a target item. Assume the set of $k$ most similar items to be $\{i_1, i_2, \ldots, i_k\}$ and their corresponding similarities to be $\{s_{i1}, s_{i2}, \ldots, s_{ik}\}$. Once the set of similar items are identified, the prediction of the preference score of an active user on the target item is then computed by taking a weighted average of the active user’s preference scores on these similar items (Schafer et al., 2001). Based on this process, an item-correlation technique for recommendations consists of two main phases: similarity computation and recommendation generation.

To determine the similarity between two items $i$ and $j$, the users who have rated both of these items (called co-rated users) are first selected and a similarity method is then applied to determine the similarity measure between items $i$ and $j$. Different similarity measures have been proposed, using such methods as
cosine similarity, Pearson correlation similarity and adjusted-cosine similarity (Sarwar et al., 2001). In the cosine similarity method, two items are thought of as two vectors in the $p$ dimensional user-space (where $p$ is the number of co-rated users). As with the cosine similarity measure discussed in Section 5, the similarity between two items is measured by computing the cosine of the angle between these two vectors. Similarly, the Pearson correlation coefficient measures the similarity between two items $i$ and $j$ based on the set of co-rated users $U$, as follows:

$$\text{sim}(i, j) = \frac{\sum_{u \in U} (p_{ui} - \bar{p}_i)(p_{uj} - \bar{p}_j)}{\sqrt{\sum_{u \in U} (p_{ui} - \bar{p}_i)^2} \sqrt{\sum_{u \in U} (p_{uj} - \bar{p}_j)^2}}$$

where $p_{ui}$ denotes the preference score of the user $u$ on the item $i$, and $\bar{p}_i$ is the average preference score of the $i$-th item over the set of co-rated users $U$.

The cosine similarity does not take into account the differences in rating scale between different users. Accordingly, the adjusted cosine similarity standardizes a user’s preference score by his/her average and measures the similarity between items $i$ and $j$ as:

$$\text{sim}(i, j) = \frac{\sum_{u \in U} (p_{ui} - \bar{p}_u)(p_{uj} - \bar{p}_u)}{\sqrt{\sum_{u \in U} (p_{ui} - \bar{p}_u)^2} \sqrt{\sum_{u \in U} (p_{uj} - \bar{p}_u)^2}}$$

where $\bar{p}_u$ is the average of the $u$-th user’s preference scores.

Once the set of similar items are identified for a target item using a similarity measure, the next phase is to combine preference scores of the active user on the set of similar items to arrive at a predicted preference score on the target item. The weighted average method is typically employed for deriving the prediction. In a manner similar to that discussed in Section 5, the weighted average method tries to capture how the active user rates similar items. It computes the prediction on the target item for the active user by taking the weighted average of the preference scores given by the active user on the items similar to the target item, using the item similarities as the weights (Sarwar et al., 2001).

To produce the top-$N$ recommendation for the active user by an item-correlation technique, the predicted preference score on each item for which a preference score has not been given by the active user is derived as discussed previously. Subsequently, the top $N$ items with the highest predicted preference score are included in the recommendation list. However, as with the collaborative filtering approach, item-correlation techniques are not able to recommend the top-$M$ users for a newly available item since no user preference is available for this item.
6.2 Association Rule Techniques for Recommendations

The association rule discovery technique represents another alternative to the association-based recommendation approach (Sarwar et al., 2000). It finds interesting co-occurrences of items in a set of transactions. Formally, the association-rule mining problem is defined as follows (Agrawal et al., 1993; Agrawal and Srikant, 1994). Let $I = \{i_1, i_2, \ldots, i_m\}$ be a set of items. Let $D$ be a set of transactions, where each transaction $T$ is a set of items such that $T \subseteq I$. In the recommendation context, each transaction corresponds to a user and contains a set of items that the user liked or purchased. An association rule is an implication of the form $X \Rightarrow Y$, where $X \subseteq I$, $Y \subseteq I$, and $X \cap Y = \emptyset$. The association rule $X \Rightarrow Y$ holds in $D$ with confidence $c$ if $c\%$ of transactions in $D$ that contain $X$ also contain $Y$. The rule $X \Rightarrow Y$ has a support $s$ in $D$ if $s\%$ of transactions in $D$ contains $X \cup Y$. Given a set of transactions $D$, the problem of mining association rules is to generate all association rules that have support and confidence greater than the user-specified minimum support and minimum confidence. To efficiently find all association rules satisfying the user-specified minimum support and minimum confidence, the Apriori algorithm proposed by Agrawal and Srikant (1994) is often employed.

As mentioned, the association rule discovery technique concerns mainly the co-occurrence of items in a set of transactions. Thus, the user preferences need to be transformed into the described representation of transactions. If the user preference on an item is a binary measure, the transformation can be straightforward. An item $i$ will be included in the transaction of a user $a$ only if $p_{ai}$ is 1. However, if the user preference is on a numerical scale, the decision of whether an item will be included in a user’s transaction can be based on a pre-specified threshold, a mean-based method, or other methods. For example, given a threshold $\alpha$, an item $i$ will be included in the transaction of a user $a$ if $p_{ai} \geq \alpha$; otherwise, it will not be shown in the transaction. Likewise, in a mean-based method, an item $i$ will be included in the transaction of a user $a$ if $p_{ai} \geq \tilde{p}_a$, where $\tilde{p}_a$ is the average preference score of the user $a$. Other transformation methods can be developed to reflect the nature of user preferences and the target recommendation problem.

To recommend the top-$N$ items to an active user based on the set of association rules discovered, we first find the association rules that are supported by the active user (i.e., association rules whose left-hand-side items appear entirely in the transaction of the active user). Let $I_p$ be the set of unique items that are suggested by the right-hand-side of the association rules selected and are not shown in the transaction of the active user. Afterward, those items in $I_p$ are sorted based on the confidence of the selected association rules. If a particular item is recommended by multiple association rules, the highest confidence is used.
Finally, the top $N$ items are chosen as the recommended set for the active user. Since association rules are discovered from a set of transactions, each of which contains what a user has liked or purchased previously, the association rule discovery technique is not suitable to recommend the top-$M$ users for a newly available item because it will not be shown in any association rules found.

6.3 Summary

The association-based recommendation approach recommends items to users based on the correlations or associations between items. Since it takes the user preferences as its source input information, personalized recommendations can be achieved. Similar to the collaborative filtering recommendation approach, the association-based recommendation approach is capable of recommending items on quality and taste. Finally, because the correlations or associations between items are relatively static, item similarity or association rules can be pre-computed to improve the online scalability of an association-based recommendation technique (Sarwar et al., 2001).

On the other hand, the association-based recommendation approach encounters problems similar to the collaborative filtering recommendation approach. When a large set of items are available, users may have rated or chosen a very low percentage of items, resulting in the sparsity problem. As a result, items rated or chosen by a limited number of users cannot be effectively recommended. Furthermore, as mentioned previously, the association-based recommendation approach is incapable of recommending the top-$M$ users for a new item or including the new item in the top-$N$ recommendation for a user. Finally, the synonymy problem (i.e., different items may be highly similar in their features) cannot be addressed in the association-based recommendation approach.

7. DEMOGRAPHICS-BASED RECOMMENDATION APPROACH

The demographics-based recommendation approach recommends items to a user based on the preferences of other users whose demographics are similar to those of the user. Unlike other approaches in which recommendations are made at the item level, a demographics-based recommendation system typically generates recommendations at the category level in order to deliver more generalized recommendations and to address sparsity and synonym problems. Hence, this approach involves learning and reasoning with relationships between user demographics and expressed category preferences, where the expressed category preferences of a user are derived from individual user preferences stated previously and the category hierarchies of items. The demographics-based recommendation approach has been applied to deliver personalized advertisements on Internet storefronts (Kim et al., 2001).
7.1 Process of Demographics-based Recommendation Approach

As shown in Figure 4, the process of a demographics-based recommendation system typically can be decomposed into the following phases:

1. Data transformation: Generate a set of training examples each of whose input attributes are the demographics of a user and decisions outcomes are category preferences of the user.

2. Category Preference Model Learning: Automatically induce the preference model for each category based on the training examples pertaining to the category.

3. Recommendation Generation: Given the demographic data of a user, generate recommendations by performing reasoning on the category preference models induced previously.

As mentioned, the data transformation phase generates a set of training examples for subsequent learning of the category preference model and generation of recommendations. Input attributes of a training example are the demographic descriptions of a user that potentially affect his/her category preferences. Given the demographic data of a user, the generation of input attribute values for a user is quite straightforward. However, if individual user preferences were expressed at the item level, the generation of a user’s category preferences requires a transformation based on the category hierarchies of items. Several transformation methods have been proposed for deriving category preferences of users (Kim et al., 2001). We first assume that the user preferences are binary measures (e.g., like/dislike, purchased or not) where favorable preferences (e.g., like and purchased) are denoted as 1 while unfavorable preferences are denoted as 0. The described transformation methods can easily be modified for numerically-scaled user preferences.

1. Counting-based (frequency threshold) method: This method uses the frequency of favorite preferences of a user on all items in a category to decide whether the user prefers the category or not. Let $p_{ai}$ be the binary preference score of the user $a$ on the item $i$, $C_j$ be the category $j$, $cp_{ai}$ be the derived binary preference score of the user $a$ on the category $j$, and $w$ be the pre-specified frequency threshold. The counting-based method is as follows:

$$
cp_{aj} = \begin{cases} 
1 & \text{if } \sum_{i \in C_j} p_{ai} \geq w \\
0 & \text{otherwise}
\end{cases}
$$

2. Expected-value-based method: This method takes into account the number of items in each category
and determines whether a user prefers a category based on the expected value, as follows:

\[ cp_{aj} = \begin{cases} 
1 & \text{if } \sum_{i \in C_j} p_{ai} \geq \alpha \left( \frac{\sum_{j} \sum_{i \in C_j} p_{ai} \times N_j}{\sum_j N_j} \right) \\
0 & \text{otherwise} 
\end{cases} \]

where \( \alpha \) is a multiplier for the expected value and \( N_j \) is the number of items in the category \( j \).

3. Statistics-based method: This method sets a threshold based on such statistical values as mean and median. For example,

\[ cp_{aj} = \begin{cases} 
1 & \text{if } \sum_{i \in C_j} p_{ai} \geq \alpha \left( \frac{\sum_{j} \sum_{i \in C_j} p_{ai}}{C} \right) \\
0 & \text{otherwise} 
\end{cases} \]

where \( C \) is the number of categories.

For a main category, the category preference of a user can be derived from his/her preferences on its subcategories. For example, a user is considered to prefer a main category \( j \) if he/she prefers any subcategory of \( j \) or a certain percentage of the subcategories of \( j \).

After the data transformation, each user corresponds to a training example with a binary preference decision on each category. Subsequently, the category preference model learning phase is initiated to induce a preference model for each category based on all the training examples pertaining to the category. As with the user profile learning phase in the content-based recommendation approach, a decision tree induction algorithm, a decision rule induction algorithm, or a backpropagation neural network can be employed for the target learning task. Accordingly, for each category in the category hierarchy, a classification model is constructed to capture the relationships between user demographics and preferences of the category. Once a set of category preference models is induced, recommendations can be generated for an active user. In this approach, all of the three types of recommendations are plausible since recommendations are generated using user demographics, the category to which a target item belongs, and the category preference models relevant to the target item. Given the demographic data of an active user and the category to which a target item belongs, the prediction on whether the active user will prefer the target item can be made by reasoning on the category preference models relevant to the target item. To produce the top-\( N \) recommendation for the active user, the preference prediction on each category is first obtained. Since inductive learning algorithms described above are capable of estimating prediction accuracy, the top-\( N \) items with the highest prediction accuracy are then included in the recommendation list. Finally, to recommend the top-\( M \) users for a new item, a set of prediction decisions
can be made, one for each user, based on the category preference models associated with the target item’s
category and its supercategories. Subsequently, the top $M$ users who are predicted to like the new item
most are selected.

7.2 Summary
The demographics-based recommendation approach recommends items to a user based on the preferences
of other users whose demographics are similar to that of the user. Since it relies on individual user
preferences and user demographics to arrive at recommendation decisions, personalized recommendations
can be achieved. The demographics-based approach typically produces recommendations at the category
level. Thus, the effect of the sparsity and synonym problems on recommendation accuracy can be reduced.
Finally, online scalability is improved with the demographics-based approach because the category
preference models can be constructed off-line and the resulting models are small in size and efficient in
reasoning.

The demographics-based approach may encounter the following limitations. Although coarser
granularity in recommendations has the advantage in addressing the sparsity and synonym problems, a
user may not like all the items within a category suggested by this approach. Conversely, a dis-
recommendation of a category does not necessarily mean that the user does not like all the items in the
category. In other words, though the demographics-based approach may be able to achieve high-quality
recommendations at the category level, its recommendation accuracy may suffer at the item level.
Moreover, potential applications of the demographics-based approach may represent another source of
limitation. User demographics cannot be assumed to be available, complete, and reliable. In many
application domains, the acquisition and update of user demographics raises serious privacy issues, and
can be extremely difficult, if not impossible.

8. CONCLUSIONS
In an electronic commerce environment, recommendation systems have emerged not only to address the
challenge of information overload for consumers of information or products, but also to facilitate the
delivery of e-services to customers. This chapter surveyed the major recommendation approaches and the
techniques associated with their implementation. However, the techniques covered in this chapter are by
no means exhaustive. For example, collaborative filtering recommendation systems using Bayesian
networks, neural networks and inductive learning algorithms were not covered. Various hybrid
recommendation techniques that seek to seamlessly integrate different recommendation approaches are
not reviewed in detail. As users demand higher-quality recommendations and as E-commerce expands its
coverage into a wireless environment (the so-called mobile commerce or M-commerce), recommendation approaches will continue to evolve and new techniques will be devised, incorporating an ever richer set of data sources, such as precise real-time geographic location.

As with decision support tools, recommendation systems should be integrated with existing Web-based electronic commerce systems in which information users or online customers consume information or search and purchase products. In addition, a recommendation system may have strong implications for re-designing and personalizing E-commerce websites. As recommendation systems continue to evolve, frequent modifications of existing electronic commerce systems can be anticipated. Thus, component-based software development may represent a promising approach in achieving increased interoperability and reusability of electronic commerce systems.

A closer examination of successful E-commerce businesses suggests the appropriateness and importance of adopting multiple recommendation systems to support recommendation services to their customers. No single recommendation system can satisfy customers differing in their recommendation requirements, quality- or coverage-wise. From a practical viewpoint, it is desirable to investigate the effect of customer characteristics on the effectiveness of recommendation approaches. Furthermore, recommendation approaches may demonstrate varying degrees of effectiveness in supporting diverse commercial activities, such as advertising, marketing, or price discovery (e.g., through auctions). Empirical evaluations along this line of investigation represent an essential and interesting research direction with great practical implications.
REFERENCES


