Chapter 12

# Web-based Recommendation Systems for Personalized e-Commerce Shopping

Chih-Ping Wei, Robert F. Easley[1], and Michael J. Shaw

Abstract: In an e-commerce environment, personalization has taken on an important role in improving service levels, and fostering customer loyalty. In addition, the recommendation systems techniques that support many personalization systems are capable of customizing the recommendation of products and the display of advertisements to the individual level. This chapter provides a review of the major recommendation approaches used in web-based personalization, and their associated techniques. Broadly, these recommendation systems can be classified into demographics-based, collaborative-filtering-based, association-based, and content-based recommendation approaches.

Key words: Personalization, Recommender System, Electronic Commerce, Content-based Recommendations, Collaborative Filtering Recommendations, Association-based Recommendations, Demographics-based Recommendations

## 1. INTRODUCTION

Personalization of the e-commerce shopping experience holds great promise for improving customer service, increasing both customer satisfaction and the efficiency of the customer interaction, and engendering customer loyalty to a particular e-commerce site. At the same time, the manner in which the information necessary for customization is obtained remains an important issue for many customers (Personalization Consortium, 2000). In this chapter we explore existing recommendation systems that support e-commerce personalization, with particular attention to their data requirements and the extent to which the data may be obtained unobtrusively from the customer.

Personalization of a web site in general involves any action that tailors the web experience to make it more responsive to a particular user or set of users (Cingil et al., 2000, Mobasher et al., 2000). For example, a portal may offer the ability for a user to define the content provided, the layout of the pages, and perhaps even the structure of the site itself, based on explicit choices made by the user (Mulvenna et al., 2000). At e-commerce sites in particular, personalization involves the provision of content and services to customers in an attempt to meet their specific wants or needs absent an explicit request (Mulvenna et al., 2000, Adomavicius and Tuzhilin, 2001). In the e-commerce setting, the personalization of content becomes much more involved, expanding not only to include personalized product recommendations and advertisement displays, but also the storage and retrieval of personal information necessary to support efficient order processing and customer communication. The personalization of communication processes may even extend to include user preferences for push, pull or passive delivery of information and recommendations (Shafer et al., 2001).

A critical distinction between personalization approaches involves the degree to which a customer is required to explicitly reveal preferences or personal information, as opposed to having them indirectly determined, typically by analysis of browsing and purchase patterns and other information revealed through observable behaviors. Mulvenna et al. (2000) characterize three types of personalization systems: checkbox, Collaborative Filtering, and observational. These illustrate the full range of levels of explicit customer participation in providing personalization data. The checkbox type, wherein a customer explicitly reveals preferences, is at one end of the spectrum, and observational systems are at the other end. Collaborative Filtering, along with a number of other techniques suitable for personalization, are between the two extremes, with their position depending, to a large degree, on the type of recommendation needed and the details of the implementation.

Amazon.com is widely regarded as a leader in the implementation of personalization for e-commerce. Amazon's founder and C.E.O. Jeff Bezos is fond of saying, as he did at the PC Expo in 2000, "If we want to have 20 million customers, then we want to have 20 million stores" (Ferranti, 2000). Amazon offers an extensive set of personalization features, ranging from product reviews to wish lists and trusted friend lists that a user may establish, to 'quick pick,' 'new for you,' and other sets of recommendations offered based on past purchase patterns. A number of these are voluntary –

anyone can write, read and rate product reviews or setting up a trusted friends list allows the user to view recommendations of those on the list. Other features are driven by a recommendation engine, which works primarily with purchase pattern data collected for all purchases, to deliver recommendations based on the purchases of 'similar' customers.

The purchase data used in personalization is regarded as a key corporate asset, and Amazon is one of the only e-commerce companies that has managed to leverage this customer data by tying it in with its supply chain (Eads, 2000). However, the collection and use of this data does raise some privacy issues. For example, Amazon has faced inquiry from the Federal Trade Commission over the use of the personal data it collects (Wolverton, 2000). Amazon also created a stir by appearing at least to experiment with personalized pricing schemes (Regan, 2000). Bezos takes pains though to point out that Amazon does not request demographic information, and that any paid recommendations placed on the site would be clearly identified as such (InfoWorld, 2000). Also, the data is, for the most part, collected in an observational manner, which wins praise from usability experts (e.g. Nielsen, 1998) for minimizing the work imposed on the user, a problem, for example, with the checkbox approach.

The terms personalization and recommendation systems are often used interchangeably, and indeed there is a large overlap between the two concepts. In the e-commerce arena, personalization encompasses all but a few recommendation techniques – excluding only those that make no use of personal information. For example, Amazon also offers best-seller lists, which are a way of offering recommendations that is not at all personalized (aside from recording the decision to view it). There are also significant personalization opportunities that are not specifically related to personalizing the recommendation of products or advertisements, for example, the automatic retention of shipping and billing information that underlies Amazon's patented one-click check-out system. However, the true complexity and expense comes in the area of overlap between the two, where methods such as Collaborative Filtering are implemented to provide personalized recommendations. It is in the selection and implementation of recommendation systems that the issue of data requirements and possible strategies for data acquisition come into play. In the remainder of this chapter, we classify the various recommendation systems in use in personalization systems, and present detailed descriptions of the models that underlie them.

## 2. RECOMMENDATION SYSTEMS FOR PERSONALIZATION

Many recommendation systems have been employed as methods for e-commerce personalization. Based on the type of data required, the extent to which that data can be acquired observationally (i.e., indirectly or unobtrusively), and the techniques used to arrive at recommendation decisions, we classify existing recommendation systems into the following types:

1. *Demographics-based:* 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.

2. *Collaborative filtering:* 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.

3. *Association-based:* 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.

4. *Content-based:* 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) (Alspector 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.

*Table 1.* Characteristics of Different Recommendation Approaches

| **Recommendation Approach** | **Information Used** | **Degree of Observational Information Acquisition** |
|---|---|---|
| Demographics-based | User demographics, individual user preferences, and features of items (specifically, category hierarchy of items) | Low, direct revelation of demographic information required |
| Collaborative Filtering | User preferences | Mixed, user preferences can be observational or explicit |
| Association-based | User preferences | Mixed, user preferences can be observational or explicit |
| Content-based | Features of items and individual user preferences | Mixed to High, most information required relates to products, purchase history required. |

Table 1 summarizes the characteristics of each recommendation approach, arranged in increasing order of degree to which observational techniques may be used in obtaining the data required. The type of recommendations may consist of a set of items from among those that have not explicitly been rated or chosen by an active user $u_a$. Accordingly, two 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).

# 3. DEMOGRAPHICS-BASED RECOMMENDATION APPROACH

The demographics-based recommendation approach recommends items to a user based on the preferences of others whose demographics are similar to those of the user. A demographics-based recommendation system typically generates recommendations at the category level, rather than the individual item 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 previously-stated individual user preferences and the category hierarchies of items. The demographics-based recommendation approach can be applied, for example, to deliver personalized advertisements on Internet storefronts (Kim et al., 2001).

## 3.1 Process of Demographics-based Recommendation Approach

As shown in Figure 1, 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.

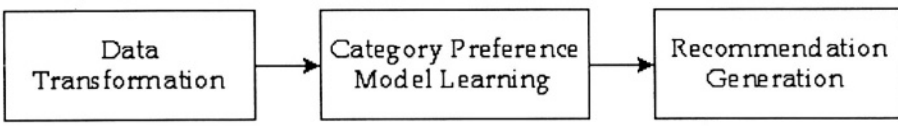


*Figure 1.* Process of Demographics-based Recommendation Approach

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:

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

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,

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

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 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 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, both types of recommendations are possible 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 of 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.

## 3.2 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 some limitations. 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 some e-commerce settings, the acquisition and update of user demographic data raises serious privacy issues, and can be quite difficult.

## 4. COLLABORATIVE FILTERING RECOMMENDATION APPROACH

The collaborative filtering recommendation approach is a commonly-used method, and differs from the demographic approach. Rather than recommending items based on user preferences and similar demographic profiles across users, the collaborative filtering approach recommends items based on the similarity of opinions across 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 2, 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.

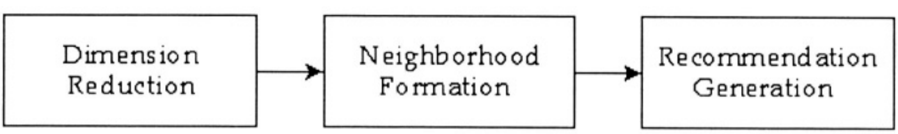


*Figure 2.* Process of Collaborative Filtering Recommendation Approach

## 4.1 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).

## 4.2 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 $u_a \notin N$ and $sim(u_a, n_i) \geq sim(u_a, n_j)$ 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:

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

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 scaling scheme (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:

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

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:

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

- *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:

$$sim(u_a,u_b)=cos(\vec{a},\vec{b})=\frac{\vec{a}\cdot\vec{b}}{\|\vec{a}\|_2\|\vec{b}\|_2}=\frac{\sum_i^m p_{ai}\cdot p_{bi}}{\sqrt{\sum_i^m p_{ai}^2}\sqrt{\sum_i^m p_{bi}^2}}$$

- *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:

$$dissim(u_a,u_b)=\overline{\sum_i^m (p_{ai}-p_{bi})^2}$$

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

  $\vec{C} = \frac{1}{j}\sum_{\vec{V} \in N} \vec{V}$ . A user *w,* such that $w \notin N$ is selected as the *j*+1-st

  neighbor only if *w* is closest to the centroid $\vec{C}$. 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).

### 4.3 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. To estimate the predicted preference score on the item $i_j \notin I_{u_a}$ 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:

$$p_{ai} = \bar{p}_a + \frac{\sum_{u=1}^{k}(p_{ui} - \bar{p}_u)\cdot sim(u_a, u_u)}{\sum_{u=l}^{k} sim(u_a, u_u)}$$

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 $i$:

$$p_{ai} = \overline{p}_a + \frac{\sum_{u=1}^{k} \frac{(p_{ui} - \overline{p}_u)}{\sigma_u} sim(u_a, u_u)}{\sum_{u=1}^{k} sim(u_a, u_u)}$$

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.

## 4.4 Summary

By using other users' opinions the collaborative filtering approach can be employed to recommend items whose content is not easily analyzed by automated feature extraction techniques. This approach is also capable of recommending items on the basis of quality and taste. Furthermore, 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 sparsity and scalability problems, the collaborative filtering approach incurs other problems. 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).

# 5. ASSOCIATION-BASED RECOMMENDATION APPROACH

The association-based recommendation approach relies on user preferences to identify items frequently associated with those items in which a user has expressed interest 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.

## 5.1 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 4, 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:

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

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:

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

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 4, 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.

## 5.2 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 $X \subset I$, $Y \subset I$, 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 $p_{ai} \geq \bar{p}_a$, where $\bar{p}_a$ 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.

### 5.3 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. Like collaborative filtering, the association-based approach is capable of recommending items on quality and taste. Also, 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 sparsity problems. As a result, items rated or chosen by a limited number of users cannot be effectively recommended. Finally, the synonymy problem (i.e., different items may be highly similar in their features) cannot be addressed in the association-based recommendation approach.

## 6. CONTENT-BASED RECOMMENDATION APPROACH

For a give 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 to be the training set with respect to the given user. As shown in Figure 3, the phases involved in a content-based 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 based on the training examples pertinent to the user.

4. Recommendation Generation: Generate recommendations by performing reasoning on the corresponding user profile model.

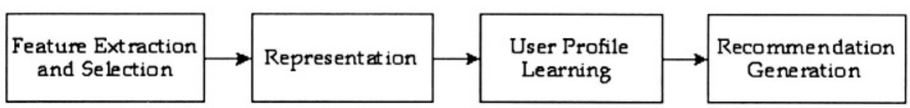


*Figure 3.* Process of Content-based Recommendation Approach

## 6.1 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 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 forward stepwise multiple regression, 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. 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).

## 6.2 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 an item's extrinsic features 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, to represent a textual document by a set of previously extract and selected intrinsic features, a binary value (e.g., indicating whether the feature appears in the document) or a numerical value (e.g., frequency 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).

## 6.3 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.

## 6.4 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 three types of recommendations can be made. To estimate the predicted preference score on item $i_j \notin I_{u_a}$ 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.

## 6.5 Summary

The content-based 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).

## 7. CONCLUSIONS

In an e-commerce environment, web-based personalization has proven to have great potential for improving transaction efficiency, providing suitable custom product recommendations, and engendering customer loyalty. This chapter classified the major approaches and described the techniques associated with the implementation of recommendation systems for web-based personalization. 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 into the wireless environment (the so-called mobile commerce or M-commerce), recommendation and personalization approaches will continue to evolve and new techniques will be devised, incorporating an ever richer set of data sources, such as real-time geographic location.

**Note**

1. Please address all future correspondence to: Robert F. Easley, Management Department, 311 Mendoza College of Business, University of Notre Dame, Notre Dame, IN 46556-5646. Phone: (219)631-6077. Fax: (219)631-5255. Email: reasley@nd.edu.

**Acknowledgement**

The authors are grateful for the supports from Mendoza College of Business at the University of Notre Dame, National Science Council, Taiwan, R.O.C., Beckman Institute for Advanced Science and Technology, and Center for International Education and Research in Accounting at the University of Illinois at Urbana-Champaign.

**References**

1. Adomavicius G., and Tuzhilin, A., "Using Data Mining Methods to Build Customer Profiles," *IEEE Computer,* 34(2), February 2001, pp.74-82.
2. Agrawal, R., Imielinski, T. and Swami, A., "Mining Association Rules Between Sets of Items in Large Databases," *Proceedings of the ACM SIGMOD International Conference on Management ofData,* Washington DC, 1993, pp. 207-216.
3. Agrawal, R. and Srikant, R., "Fast Algorithms for Mining Association Rules," *Proceedings of the 20th International Conference on Very Large Data Bases,* Santiago, Chile, 1994, pp. 487-499.
4. Alspector, J., Kolcz, A. and Karunanithi, N., "Comparing Feature-based and Clique-based User Models for Movie Selection," *Proceedings of the Third ACM Conference on Digital Libraries,* Pittsburgh, PA, 1998, pp. 11-18.
5. Balabanovic, M. and Shoham, Y., "Fab: Content-based, Collaborative Recommendation," *Communications ofthe ACM,* 40 (3), 1997, pp. 66-72.
6. Basu, C., Hirsh, H., and Cohen, W., "Recommendation as Classification: Using Social and Content-based Information in Recommendation," *Proceedings of the Workshop on Recommender Systems,* AAAI Press, 1998, pp. 11-15.
7. Billsus, D. and Pazzani, M. J., "Learning Collaborative Information Filters," *Proceedings ofthe Workshop on Recommender Systems,* AAAI Press, 1998.
8. Breese, J. S., Heckerman, D. and Kadie, C., "Empirical Analysis of Predictive Algorithms for Collaborative Filtering," *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence (UAI-98),* San Francisco, CA, 1998, pp. 43-52.
9. Breiman, L., Friedman, J., Olshen, R. and Stone, C., *Classification and Regression Trees,* Pacific Grove: Wadsworth, 1984.
10. Cingil, I., Dogac, A. and Azgin, A., "A Broader Approach to Personalization," *Communications ofthe ACM,* 43(8), August 2000, pp. 136-141.
11. Clark, P. and Niblett, T., "The CN2 Induction Algorithm," *Machine Learning,* 3(4), 1989, pp. 261-283.
12. Condliff, M. K., Lewis, D. D., Madigan, D., and Posse, C., "Bayesian Mixed-Effects Models for Recommender Systems," *Proceedings of Workshop on Recommender Systems: Algorithms and Evaluation,* Berkeley, CA, 1999.
13. Dumais, S., Platt, J., Heckerman, D., and Sahami, M., "Inductive Learning Algorithms and Representations for Text Categorization," *Proceedings of the ACM 7th International Conference on Information and Knowledge Management (CIKM '98),* Washington D.C., 1998, pp. 148-155.

14. Eads, S., "The Web's Still-Unfulfilled Personalization Promise," *Business Week Online,* August 4, 2000, available at: www.businessweek.com/bwdaily/dnflash/aug2000/nf2000084_506.htm.
15. Feranti, M., "Amazon's Bezos touts personalization," *InfoWorld,* June 28, 2000, available at: www.infoworld.com/articles/hn/xml/00/06/28/000628hnbezos.xml
16. Herlocker, J. L., Konstan, J. A., Borchers, A., and Riedl, J., "An Algorithmic Framework for Performing Collaborative Filtering," *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval,* Berkeley, CA, 1999, pp. 230-237.
17. Hill, W., Stead, L., Rosenstein, M., and Furnas, G., "Recommending and Evaluating Choices in a Virtual Community of Use," *Proceedings of the Conference on Human Factors in Computing Systems,* 1995, pp. 194-201.
18. Kass, G. V., "An Exploratory Technique for Investigating Large Quantities of Categorical Data," *Applied Statistics,* 29, 1980, pp. 119-127.
19. Kim, J. W., Lee, B. H., Shaw, M. J., Chang, H. L., and Nelson, M., "Application of Decision-Tree Induction Techniques to Personalized Advertisements on Internet Storefronts," *International Journal of Electronic Commerce,* 5(3), 2001, pp. 45-62.
20. Kittler, J., "Mathematical Methods of Feature Selection in Pattern Recognition," *International Journal of Man-Machine Studies,* 7, 1975, pp. 609-637.
21. Konstan, J. A., Miller, B. N., Maltz, D., Herlocker, J. L., Gordon, L. R. and Riedl, J., "GroupLens: Applying Collaborative Filtering to Usenet News," *Communications of the ACM,* 40 (3), 1997, pp. 77-87.
22. Krulwich, B. and Burkey, C., "Learning User Information Interests through Extraction of Semantically Significant Phrases," *Proceedings of the AAAI Spring Symposium on Machine Learning in Information Access,* Stanford, CA, 1996.
23. Lam, W. and Ho, C. Y., "Using A Generalized Instance set for Automatic Text Categorization," *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval,* Melbourne, Australia, 1998, pp. 81-89.
24. Lang, K., "NewsWeeder: Learning to Filter Netnews," *Proceedings of the 12th International Conference on Machine Learning,* San Francisco, CA, 1995, pp. 331-339.
25. Lewis, D. and Ringuette, M., "A Comparison of Two Learning Algorithms for Text Categorization," *Proceedings of Symposium on Document Analysis and Information Retrieval,* 1994.
26. Lynch, C., "Personalization and Recommender Systems in the Larger Context: New Directions and Research Questions," *Proceedings of the 2nd DELOS Network of Excellence Workshop on Personalization and Recommender Systems in Digital Libraries,* Dublin, Ireland, 2001.
27. Mobasher, B., Cooley, R., and Srivastava, J., "Automatic Personalization Based on Web Usage Mining," *Communications of the ACM,* 43(8), August 2000, pp. 142-151.
28. Modrzejewski, M., "Feature Selection Using Rough Sets Theory," *Proceedings of European Conference on Machine Learning,* 1993, pp. 213-226.

29. Mooncy, R. J. and Roy, L., "Content-Based Book Recommending Using Learning for Text Categorization," *Proceedings of the 5th ACM Conference on Digital Libraries,* San Antonio, TX, 2000, pp. 195-204.
30. Mulvenna, M. D., Anand, S. S., and Buchner, A. G., "Personalization on the Net Using Web Mining," *Communications of the ACM,* 43(8), August 2000, pp. 122-125.
31. Neilsen, J., "Personalization is Over-Rated," useit.com Alertbox, October 4, 1998, available at: http://www.useit.eom/alertbox/981004.html
32. Ng, H. T., Goh, W. B., and Low, K. L., "Feature Selection, Perceptron Learning, and A Usability Case Study for Text Categorization," *Proceedings of Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '97),* Philadelphia, PA, 1997, pp. 67-73.
33. Pazzani, M., Muramatsu, J., and Billsus, D., "Syskill & Webert: Identifying Interesting Web Sites," *Proceedings of the 13th National Conference on Artificial Intelligence,* Portland, OR, 1996, pp. 54-61.
34. Personalization Consortium. Press Release. "Personalization Information," April 5, 2000, available at: http://www.personalization.org/personalization.html (downloaded January 25, 2002).
35. Piramuthu, S., "Evaluating Feature Selection Methods for Learning in Data Mining Applications," *Proceedings of Thirty-First Annual Hawaii International Conference on System Sciences,* Kohala Coast, HI, 1998.
36. Quinlan, J. R., "Induction of Decision Trees," *Machine Learning,* 1(1), 1986, pp. 81-106.
37. Quinlan, J. R., *C4.5 Programs for Machine Learning,* San Mateo, CA: Morgan Kaufmann, 1993.
38. Regan, K., "Amazon's Friendly Deception," *E-Commerce Times,* September 18, 2000, available at: www.ecommercetimes.com/perl/slory/4310.html.
39. Resnick, P., lacovou, N , Suchak, M., Bergstrom, P., and Riedl, J., "GroupLens: An Open Architecture for Collaborative Filtering of Netnews," *Proceedings of the Conference on Computer Supported Cooperative Work (CSCW),* Chapel Hill, NC, 1994, pp. 175-186.
40. Rumelhart, D. E., Hinton, G. E., and Williams, R. J., "Learning Internal Representations by Error Propagation," In *Parallel Distributed Processing: Explorations in the Microstriictures of Cognition,* 1, D. E. Rumelhart and J. L. McClelland (Eds.), Cambridge, MA: MIT Press, 1986, pp. 318-362.
41. Sarwar, B. M., Karypis, G.. Konstan, J. A., and Riedl, J., "Analysis of Recommendation Algorithms for K-Commerce," *Proceedings of the 2nd ACM Conference on Electronic Commerce.* Minneapolis, MN. 2000, pp. 158-167.
42. Sarwar, B.. Karypis, G., Konstan, J.. and Riedl, J., "Item-Based Collaborative Filtering Recommendation Algorithms," *Proceedings of the 10th International Conference on World Wide Web.* Hong Kong, 2001, pp. 285-295.
43. Sarwar. B., Konstan, J.. Borchers. A., Herlocker, J., Miller, B., and Riedl, J., "Using Filtering Agents to Improve Prediction Quality in the GroupLens Research Collaborative Filtering System," *Proceedings of the ACM Conference on Computer Supported Cooperative Work (CSCW),* Seattle, Washington, 1998, pp. 345-355.

44. Schafer, J. B., Konstan, J. A., and Riedl, J., "E-Commerce Recommendation Applications," *Data Mining and Knowledge Discovery,* 5(1), 2001, pp. 115-153.
45. Siedlecki, W. and Sklansky, J., "A Note on Genetic Algorithms for Large-scale Feature Selection," *Pattern Recognition Letters,* 10(5), 1989, pp. 335-347.
46. Shardanand, U. and Maes, P., "Social Information Filtering: Algorithms for Automating 'Word of Mouth'," *Proceedings of Conference on Human Factors in Computing Systems,* 1995, pp. 210-217.
47. Wolverton, T., "Amazon, others add personal touches to home pages," June 13, 2000, available at: http://news.com.com/2102-1017-241869.html
48. Yang, Y. and Chute, C. G., "An Example-Based Mapping Method for Text Categorization and Retrieval," *ACM Transactions on Information Systems,* 12(3), 1994, pp. 252-277.