International Journal of Electronic Commerce

ISSN: 1086-4415 (Print) 1557-9301 (Online) Journal homepage: https://www.tandfonline.com/loi/mjec20

Utilizing Popularity Characteristics for Product Recommendation

Hyung Jun Ahn

To cite this article: Hyung Jun Ahn (2006) Utilizing Popularity Characteristics for Product Recommendation, International Journal of Electronic Commerce, 11:2, 59-80, DOI: 10.2753/JEC1086-4415110203

To link to this article: https://doi.org/10.2753/JEC1086-4415110203

Published online: 08 Dec 2014.

Utilizing Popularity Characteristics for Product Recommendation

Hyung Jun Ahn

ABSTRACT: This paper presents a novel approach to automated product recommendation based on the popularity characteristics of products. Popularity plays a significant role in the consumer purchasing process but has not been given much attention in recommendation research. A three-dimensional model of popularity is used to develop popularity classes of products. These are joined with the MovieLens dataset to create a hybrid movie recommendation system that combines genre and popularity information. As compared with collaborative filtering, the hybrid system shows positive results under the conditions of data sparsity and cold-starting. Many interesting issues for further research are suggested.

KEY WORDS AND PHRASES: Automated product recommendation, cold-starting, hybrid recommender system, naive Bayesian, popularity-based recommendation, popularity model, sparsity.

Automated product recommendation is widely used by many Internet shopping malls, where it plays a critical role in effective on-line marketing by promoting cross-selling and up-selling of products. As electronic commerce matures, the effectiveness of recommendation is winning recognition as a crucial factor for organizations under growing competitive pressure.

Researchers have produced successful recommendation methods that use data of many different kinds, including purchase history, product ratings by buyers, product characteristics, and demographic information of shoppers [1, 2, 5, 20]. Recommendation methods can be broadly categorized as either content-based methods or collaborative filtering methods [5, 6, 8, 12, 14, 15, 17, 18, 20, 30]. Successful results have been reported, and many of these methods have been implemented by real-world organizations. Nonetheless, they often entail problems, such as poor recommendation quality under data sparsity and limited ability to recommend new products or to new buyers [7, 19, 21, 29, 33]. In addition, many recommendation methods do not clearly explain to the user why they are recommending a specific product. This limits the potential uses of the recommendation results for further analysis [11].

The present paper proposes a novel approach to recommendation that uses the popularity characteristics of products. Popularity features play an important role in consumers' purchasing decisions because most consumers are influenced by how others feel about a product or how widely a product has been exposed in the market [3, 16]. Despite the significance of the popularity factor, it has not been given much attention in recommendation studies. The discussion that follows defines three dimensions of popularity and presents an algorithm for partitioning products into a reasonable number of popularity classes located in the three-dimensional space of popularity. The approach is applied to the MovieLens dataset, and a hybrid recommender system for movies is developed by combining popularity-class information with genre information. The performance of the combined system is experimentally

International Journal of Electronic Commerce / Winter 2006–7, Vol. 11, No. 2, pp. 59–80.

1086-4415/2007 $9.50 + 0.00.
DOI 10.2753/JEC1086-4415110203

compared with the collaborative filtering method under the conditions of data sparsity and cold-starting.

Overview of Literature

Categorization of Recommender Systems

Approaches to automated product recommendation can be broadly classified as either content-based filtering or collaborative filtering. Content-based approaches use content information, or features of products, to build profiles of products and buyers that are then used to calculate the match between a specific buyer and product [14, 18, 20]. Product categories and product descriptions have been used for content-based filtering in many studies [1, 2, 20, 22]. Collaborative filtering, in contrast, uses buyers' ratings of products rather than content information and finds people-to-people similarities between buyers. Using this information, it recommends products highly rated by similar buyers to a target buyer [6, 8, 12, 15, 17, 30].

A more fine-grained five-part classification has been devised by Burke [5]. Working from an extensive literature review, he classifies recommender systems as either collaborative, content-based, demographic, utility-based, or knowledge-based, depending on the types of data and processes they use. Demographic recommender systems resemble collaborative filtering in that they utilize people-to-people similarity for recommendation, but they use demographic information, and not buyers' rating data, for similarity calculation. Utility-based systems use buyers' utility functions to calculate the level of utility of a specific item for a specific buyer. Knowledge-based systems use inferences, often adopting techniques from artificial intelligence, to infer a match between buyer and product. Both utility-based systems and knowledge-based systems use features of products rather than buyers' ratings.

The various recommender systems exhibit different pros and cons, and this has led to the development of hybrid systems. Burke presents seven types of hybrid recommender systems: weighted, switching, mixed, feature combination, cascade, feature augmentation, and meta-level [5]. These methods either combine recommendation results from different methods (weighted, switching, and mixed), utilize different types of data together (feature combination), or pipeline two or more recommender systems (cascade, feature augmentation, and meta-level) in various ways.

The present paper presents a hybrid recommendation method that utilizes both buyers' ratings and genre information about movies. Thus, it can be located within the feature-combination category of the classification outlined above.

Collaborative Filtering

As reported by many researchers, collaborative filtering performs well and has won wide acceptance. Therefore, it is reasonable to compare its performance with that of the hybrid method proposed in this paper.

The literature offers many variations on collaborative filtering [8, 9, 10, 13, 18, 32]. Most collaborative filtering methods predict a buyer's rating of a specific product based on the way similar buyers have rated the same product. The Pearson correlation coefficient and cosine measure are frequently used to calculate similarities between buyers. Formally, the following formulae using Pearson's correlation coefficient have been widely adopted for the similarity calculation and the prediction, respectively [24, 26]:

$$sim(u_x, u_y) = \frac{\sum_{h=1}^{n'} \left(r_{u_x,i_h} - \bar{r}_{u_x}\right)\left(r_{u_y,i_h} - \bar{r}_{u_y}\right)}{\sqrt{\sum_{h=1}^{n'} \left(r_{u_x,i_h} - \bar{r}_{u_x}\right)^2} \sqrt{\sum_{h=1}^{n'} \left(r_{u_y,i_h} - \bar{r}_{u_y}\right)^2}}$$

for the similarity calculation between two users u_x and u_y where $r_{u,i}$ is the rating of product i by user u, $\bar{r}_u$ is the average rating of user u for all the products rated by the user, and n' is the number of items co-rated by both users.

$$p(u_a, i_a) = \bar{r}_{u_a} + \frac{\sum_{h=1}^{m'} sim(u_a, u_h)\left(r_{u_h,i_a} - \bar{r}_{u_h}\right)}{\sum_{h=1}^{m'} \left|sim(u_a, u_h)\right|}$$

for the prediction of a rating for item i_a by user u_a where m' is the number of other users who have also rated the item.

These formulae are also used in this paper for performance comparison.

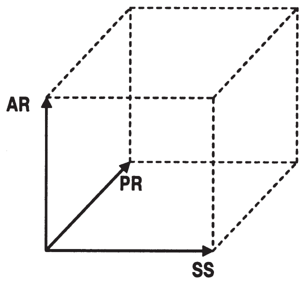
Collaborative filtering methods often attain successful results but are handicapped by not performing well under data sparsity, cold-starting, and lack of understandability. Sparsity of data refers to the problem of data deficiency, which occurs quite frequently because of the difficulty of collecting a sufficiently large set of rating data in most Internet shopping malls. The problem results in impaired similarity calculation and poor recommendation performance [29, 33]. Cold-starting refers to the difficulty of recommending products to a new buyer who has little or no purchasing history, or of recommending a new product when there are very few rating records [7, 19, 21]. Collaborative filtering methods have the further limitation of not adequately explaining their recommendations [11]. Most collaborative filtering methods use what may be called a "black-box" process in that it cannot explain what characteristics of buyer and product lead to the given recommendation, mainly because of the difficulty of intuitively describing specific results of similarity and prediction calculation. This limits the possibilities for further analysis and for more flexible use of the recommendation results.

Utilizing Popularity for Recommendation

Three Dimensions of Popularity and Popularity Measures

This paper proposes a new approach to automated recommendation, hereinafter termed popularity-based recommendation (PBR), that utilizes the popularity characteristics of products. Broadly speaking, there are two reasons for adopting popularity features for recommendation. First, popularity often represents important characteristics of a product. Many products are planned and designed from the outset to be strategically positioned in specific market segments with different numbers of potential consumers. By implication, the developers have intentionally provided these products with different popularity characteristics in order to appeal to consumers of different types. The extent to which a product is known to consumers via various channels usually indicates how broadly it has been advertised and promoted, and what types of consumers are being targeted. Second, the popularity of a product greatly influences consumer purchasing decisions. Different types of consumers show dissimilar ways of information seeking and problem solving in purchasing. For example, some consumers utilize extensive sources of information before making decisions, whereas others rely on easy, simple, and limited sources of information. Quite obviously the two groups will be affected in different ways by product popularity [16]. There are also differences in motivation between consumers choosing a product to purchase. According to Arnould, Price, and Zinkhan, some consumers are motivated by a "uniqueness or novelty need," and others by an "affiliation motive," which again implies that consumers are distinctively affected by certain popularity characteristics [3]. Despite their significance, as discussed above, recommendation systems have made only very limited use of popularity characteristics, such as recommending bestsellers to buyers [28].

The utilization of the concept of popularity in recommendation requires an understanding of its three key dimensions. The first dimension represents whether consumers perceive the product to be of high value. When used alone, this dimension represents a product's quality or level of satisfaction rather than its popularity, but when combined with the other two dimensions, it plays an important role in explaining the popularity features of a product. The second dimension represents the frequency of a product's being purchased regardless of its perceived value. It differs from the first dimension in that many products are mass-marketed and widely exposed but are not rated highly by consumers. The third dimension of product popularity is the size of the strong-support group for a product regardless of its average rating or frequency of purchase. Early adopters in the product life-cycle model [16] and the fans of "cult" movies are good examples of buyers who exhibit strong support for certain products. Figure 1 defines the three dimensions of popularity more formally. As shown in the figure, the first popularity dimension, *average rating* (AR), is calculated by dividing the sum of all buyer ratings of a product by the total number of buyers who have purchased it (or rated it). The second dimension of popularity, *percentage of being rated* (PR), is the total number of ratings of a product divided by the total number of potential buyers, which represents the product's level of exposure. The third popularity measure, *strong support* (SS), is the number of buyers who have shown strong support for the product divided by the total number of buyers who have purchased it (or rated it). The measure assumes that a buyer whose rating for a product exceeds a predefined threshold value has shown strong support for the product.



$$PR = \frac{\text{Number of buyers}}{\text{Number of total potential buyers}}$$

$$AR = \frac{\text{Sum of ratings by all buyers}}{\text{Number of total buyers}}$$

$$SS = \frac{\text{Number of buyers with strong support}}{\text{Number of total buyers}}$$

Figure 1. Three Dimensions of Popularity

In light of the foregoing, the *space of popularity* can be defined as a three-dimensional geometric space with SS, AR, and PR as its three rectangular axes.

Dataset for Experiment

The experiments described in this paper applied the popularity model to the MovieLens dataset, which is widely used in recommendation research [23]. Movies with fewer than five ratings were removed from the dataset. This left about 77,000 ratings for 838 movies by 943 raters. Each user had at least 20 ratings. Eighty percent of the raters, or buyers, were assigned to training, and 20 percent were assigned to test the recommendation performance (*see Figure 2*). The same divisions were used for the collaborative filtering method for comparison. The ratings of users range from 1 to 5 in the dataset. A rating of 4 or above indicated preference for a movie, and 2 or below dislike. A rating of 5, indicating strong preference for a movie, was used as the threshold value to calculate the SS dimension of popularity.

Overview of Recommendation Procedure

The recommendation procedure using the popularity-based rating method is summarized in Table 1, where each step of the procedure is described along with the data components in Figure 3. In essence, the PBR approach can be defined as finding out what popularity characteristics a buyer prefers and recommending products that exhibit them.

In the method's first step, the three-dimensional space of popularity is partitioned into discrete popularity classes where each class is shaped as either a cuboid (rectangular box) or a cube in the space. Each movie in the dataset located at a point of the popularity space is assigned a popularity class. Profiles of the movies are constructed by combining popularity-class information from the preceding step with genre information provided by the original dataset. Since some movies belong to more than one genre in the MovieLens dataset, this assigns one or more <*genre ID, popularity class*> pairs to each movie. In

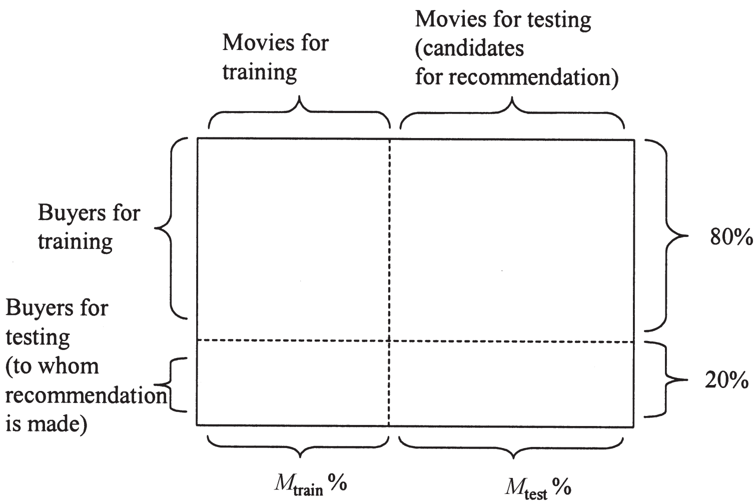


Figure 2. Dataset for Experiments

Step	Description	Output	Data used
1	Derivation of popularity classes	Discrete popularity classes. Each class is either a cuboid (rectangular box) or a cube in the three-dimensional popularity space.	All ratings for movies
2	Profiling of movies	All movies in dataset are assigned one or more <*genre ID*, *popularity class*> pairs.	Output from Step 1; genre information from dataset
3	Construction of virtual baskets from ratings of training buyers	Virtual shopping history of training buyers containing <*genre ID*, *popularity class*> pairs for movies each buyer has rated.	Output from Step 2; movies in (1) of Figure 3 for each buyer b_{train}
4	Calculation of sample probabilities	Sample probabilities that represent strength of association between <*genre ID*, *popularity class*> pairs from the baskets.	Output from Step 3
5	Creation of preference profiles for testing buyers	Preference profiles for each buyer that contain preference scores for all <*genre ID*, *popularity class*> pairs.	Output from Step 2; movies in (2) of Figure 3 for each buyer b_{test}
6	Recommendation	Movies are recommended according to preference profile constructed in Step 5.	Output from Steps 2 and 5; movies in (3) of Figure 3 for each buyer b_{test}

Table 1. Recommendation Procedure Using PBR Method.

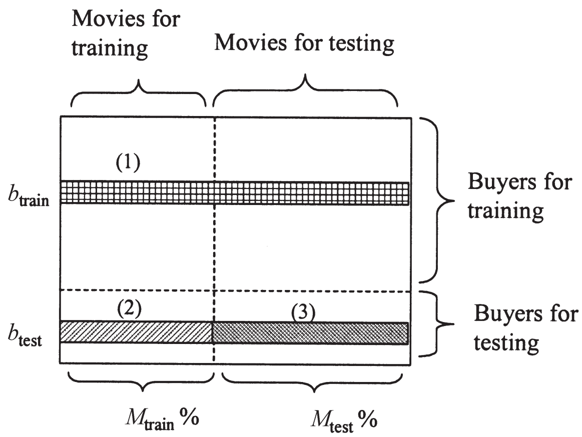


Figure 3. Breakdown of the Dataset for the Recommendation Procedure

the third step, virtual shopping baskets are constructed for the buyers in the training-buyer section by assuming that they have purchased or watched the movies they rated in the dataset. The basket represents the buyer's purchase history and, as in the Market Basket Analysis technique [4], is used to calculate sample probabilities for showing the strength of association between different <*genre ID, popularity class*> pairs in the fourth step. In the fifth step, preference profiles are constructed for the testing buyers in accordance with their virtual purchase histories (using each buyer's ratings of training movies as the purchase history). The generated profiles contain the buyer's preference scores for each <*genre ID, popularity class*> pair. The naive Bayesian method is partly used in this step to estimate preference scores for some <*genre ID, popularity class*> pairs for which a buyer has not shown any explicit preference. In the last step, movies are recommended to the buyer according to the buyer's preference profile—movies that belong to <*genre ID, popularity class*> pairs with higher preferences scores in the profile are recommended first.

These steps are explained in detail in the following sections.

Partitioning the Popularity Space and Categorization of Movies

As stated earlier, this paper presents a hybrid recommender system that utilizes popularity characteristics and movie genre information. The MovieLens dataset comprises 18 different genres, and a movie can be assigned to more than one genre. Thus, every movie is assigned one or more pairs of <*genre ID, a popularity class*>, which from here on will be denoted simply as $<g_i, p_j>$ pairs.

Popularity classes are derived by dividing each dimension of the popularity space into four discrete unit spaces, as shown in Figure 4. Each unit space's length for each dimension is:

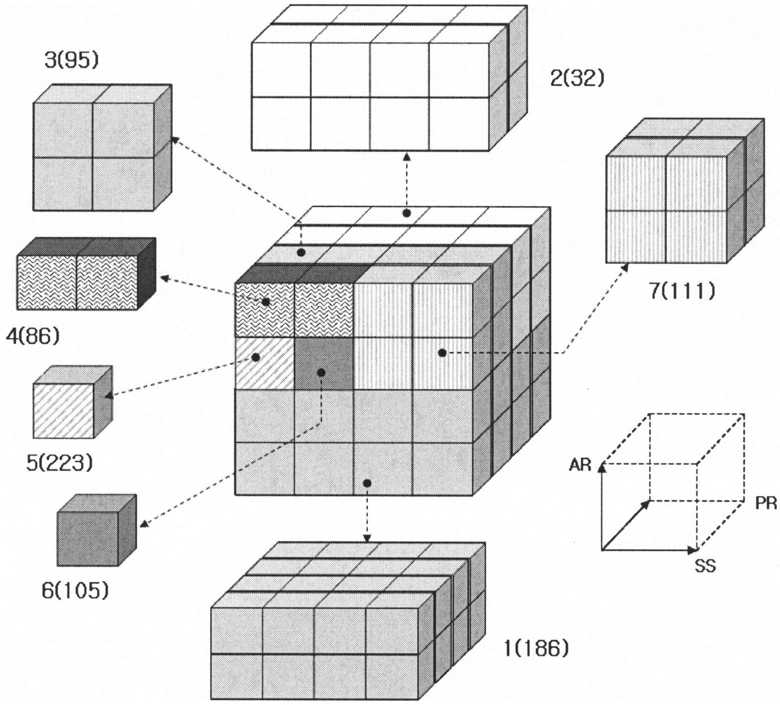


Figure 4. Partitioning of Movies Based on Popularity Measures

$$\frac{Maximum\_Value\_of\_Dimension - Minimum\_Value\_of\_Dimension}{4}.$$

This initially classifies movies into discrete unit spaces where there are, in all, 4 × 4 × 4 = 64 spaces. Since sample probabilities will be calculated in the later phase for each $<g_i, p_j>$ pair, and probabilistic associations among the pairs will be derived, the number of spaces, 64, is still too large, given that the MovieLens dataset comprises 18 genres. If there are, in all, 64 × 18 = 1152 $<g_i, p_j>$ pairs for the categorization of movies, the probabilities for each pair as well as for the associations between the pairs will be adversely affected by the insufficient number of data and thus unreliable. Moreover, having as many as 64 different categories of products in terms only of popularity is counter-intuitive. Considering these problems of complexity and sparsity, the paper fixes 10 as the maximum number of popularity classes and will not accept a class with fewer than 30 movies, but bear in mind that this is a design choice for the purposes of the paper rather than an optimum. The following algorithm is used to partition the three-dimensional popularity space into a small number of popularity classes (≤ 10) under the constraints given above. Note that the size of a class or a space in the algorithm denotes the number of movies in the training data that belong to the class or the space.

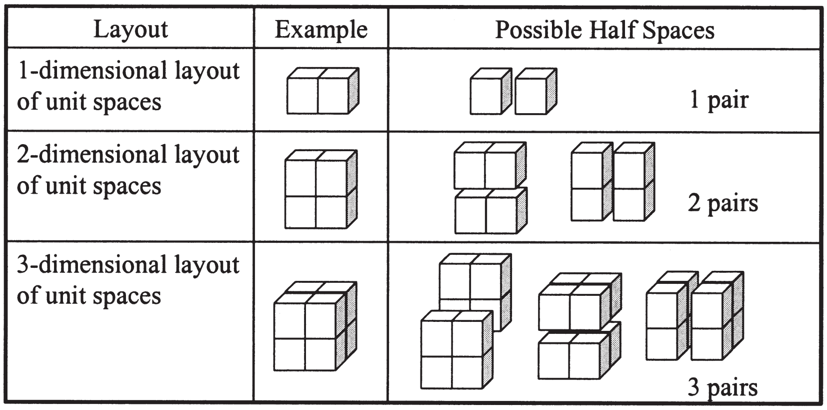


Figure 5. Half Spaces According to Different Types of Given Space

Step 1. NSC (miNimum Size of a Class) = 30.
Step 2. MSC (Maximum Size of a Class) = *k* ; *k* is a small number (e.g., 100).
Step 3. Begin with the entire space of popularity.
Step 4.1. If the given space is a unit space, register it as a final class; go to Step 5.
Step 4.2. If the size of the given space is smaller than MSC, register it as a final class; go to Step 5.
Step 4.3. If neither of the above is true, divide the given space into all possible pairs of halves (there can be one to three pairs of half spaces according to the shape of the given space, as in Figure 5). Choose the largest of the half spaces in all the pairs, and, if it is smaller than MSC, register it as a final class and go to Step 4.1 with its counterpart in the same pair. If it is not smaller, go to Step 4.1 twice with the half space and its counterpart in the same pair respectively.
Step 5. If any remaining subspace is unregistered, wait until it is divided and registered as classes (there can be multiple flows of partitioning going through Step 4.1 to Step 4.3).
Step 6. If the number of registered classes is smaller than or equal to 10, and the smallest block is larger than NSC, stop. If not, increase MSC by 1 and go to Step 3.

The advantages of this partitioning method can be summarized as follows:

1. Partitioning the space into a manageable number of classes enables the use of the popularity characteristics as discrete metrics rather than continuous ones. This makes it easier to use the measures.
2. Grouping movies into popularity classes facilitates identifying the relational characteristics between classes. For example, the probabil-

Class	Description
1	Overall poorly rated
2	High exposure and highly rated
3	Low-to-medium exposure, highly rated, low-to-medium level of strong support
4	Low-to-medium exposure, rated highly, low-to-medium level of strong support
5	Low exposure, rated OK, small strong support
6	Low exposure, rated OK, medium level of strong support
7	High strong support, low-to-medium level of exposure, rated good

Table 2. Brief Description of Popularity Classes.

ity that a preference for one class will imply a preference for another class can be easily estimated with sample statistics between the $<g_i, p_j>$ pairs.

Compared with clustering approaches such as the *k*-means method [31], partitioning makes the classes easier to understand because each partition is clearly sectioned with discrete boundaries in each dimension, rather than grouped around clustering centers with unintuitive and confusing boundaries. Furthermore, unlike clustering, partitioning does not require an assumption about the underlying parameters of clusters, such as initial cluster centers and number of clusters. This is a valuable property because easy understanding of product profiling makes for easier and wider utilization of the measures (as will be discussed further on).

Partitioning results in the seven popularity classes shown in Figure 4. The numbers near a class represent, respectively, the ID of the class and the number of movies that belong to it. For example, 7(111) represents the seventh class with 111 movies associated. Table 2 provides brief descriptions of the classes. As can be seen, the 32 movies in the second class are highly exposed movies with high average ratings. The 111 movies in the seventh class appear to be less exposed but have large, strong support groups and high average ratings. One can roughly infer that the movies in the second class are blockbusters, whereas those in the seventh class are cult movies favored by special groups. Figure 6 shows the distribution of movies in each class in the three-dimensional space of popularity and in all of its two-dimensional subplanes.

Construction of Shopping Baskets for Training Buyers

The shopping baskets for training buyers are constructed using the class and genre information of movies to calculate the probabilities for association among the pairs. Two types of baskets are constructed for each user: a preference basket and a dislike basket. A buyer's preference basket contains $<g_i, p_j>$ pairs for movies the buyer has rated 4 or above. That is, $basketPref_b = \{<g_i, p_j> \mid <g_i, p_j>$ is a profile of a movie that buyer b has rated 4 or above$\}$. Similarly, a buyer's dislike basket is defined as $basketDislike_b = \{<g_i, p_j> \mid <g_i, p_j>$ is a

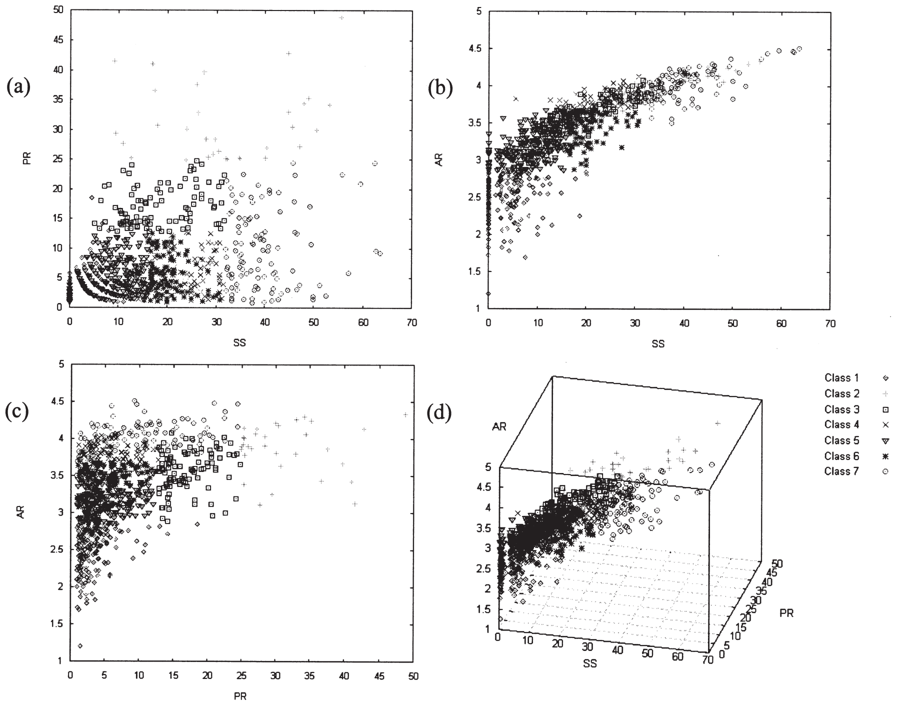


Figure 6. Scatter Plots of the Movies of Each Class

Notes: (a) two-dimensional plot with X = SS and Y = PR; (b) two-dimensional plot with X = SS and Y = AR; (c) two-dimensional plot with X = PR and Y = AR; (d) three-dimensional plot with X = S, Y = PR, and Z = AR.

profile of a movie that buyer b has rated 2 or below}. Note that the baskets only represent whether or not the pairs appear in the buyer's ratings, not the count of the pairs. In some instances, a buyer's preference and dislike baskets may both contain the same $<g_i, p_j>$ pair. Using the baskets constructed with the preceding method, three types of sample probabilities are calculated, as shown in Table 3.

Constructing Testing Buyer Profiles and Recommending Movies

A testing buyer profile is defined as a vector of 3-tuples ($<g_1, p_1, v_1>$, $<g_1, p_2, v_2>$, . . . , $<g_n, p_m, v_{nm}>$) of length $n \times m$ when there are n genres and m popularity classes, and v_k represents the preference value of the buyer for each $<g_i, p_j>$.

Constructing a testing buyer's profile begins with 0 preference values for all $<g_i, p_j>$ pairs. That is to say, initially a testing buyer profile vector is ($<g_1, p_1, 0>$, $<g_1, p_2, 0>$, . . . , $<g_n, p_m, 0>$). Then, for each buyer rating of a training movie of 4 or above, 1 is added to all the $<g_i, p_j>$ pairs of the profile that correspond to the movie. Conversely, 1 is subtracted from the values of the $<g_i, p_j>$ pairs that correspond to the movie if the rating is 2 or below. After going through

Probability	Description
$P_{like}(<g_i, p_j>)$	Probability that pair $<g_i, p_j>$ appears in a preference basket.
$P_{like}(<g_i, p_j> \mid <g_k, p_l>)$	Conditional probability that pair $<g_i, p_j>$ appears in preference baskets that contain $<g_k, p_l>$, where $i \neq k$ or $j \neq l$.
$P_{dislike}(<g_i, p_j> \mid <g_k, p_l>)$	Conditional probability that pair $<g_i, p_j>$ appears in dislike baskets of buyers who have $<g_k, p_l>$ in their preference baskets, where $i \neq k$ or $j \neq l$.

Table 3. Probabilities for Naive Bayesian.

all the buyer's ratings of the training movies, for example, what is left is a vector that may look like the following: $((<g_1, p_1, 3>, <g_1, p_2, -2>, \ldots, <g_i, p_j, 0>, <g_i, p_{j+1}, 0> \ldots, <g_n, p_m, 2>)$. As can be seen, some $<g_i, p_j>$ pairs still have 0 values because the buyer has not explicitly shown any preference or dislike for them, whereas other pairs are marked with explicit values for preferences or dislikes. For the pairs that have not been assigned an explicit value, the naive Bayesian method is used to calculate the preference values for them indirectly [31]. Suppose that a buyer has shown explicit preferences for $<g_{l1}, p_{l1}>, \ldots, <g_{ln}, p_{ln}>$ pairs and explicit dislikes for $<g_{d1}, p_{d1}>, \ldots, <g_{dm}, p_{dm}>$ pairs. If, for any x and y, $<g_x, g_y>^L$ denotes an event where the user likes the $<g_x, g_y>$ pair, and $<g_x, g_y>^D$ denotes an event where the user dislikes the pair, then:

$$P\left(<g_i,p_j>^L | <g_{l1},p_{l1}>^L \ldots <g_{ln},p_{ln}>^L <g_{d1},p_{d1}>^D \ldots <g_{dm},p_{dm}>^D\right)$$
$$= \frac{P\left(<g_{l1},p_{l1}>^L \ldots <g_{ln},p_{ln}>^L <g_{d1},p_{d1}>^D \ldots <g_{dm},p_{dm}>^D | <g_i,p_j>^L\right)}{P\left(<g_{l1},p_{l1}>^L \ldots <g_{ln},p_{ln}>^L <g_{d1},p_{d1}>^D \ldots <g_{dm},p_{dm}>^D\right)}$$
$$P\left(<g_i,p_j>^L\right). \tag{1}$$

(1) represents the probability that a buyer who has shown explicit preferences and dislikes for certain pairings of genre and popularity class will like $<g_i, p_j>$. Since only the ordering of $<g_i, p_j>$ pairs will be utilized for the recommendation, the naive Bayesian method discards the denominator from (1). Thus:

$$P\left(<g_{l1},p_{l1}>^L \ldots <g_{ln},p_{ln}>^L <g_{d1},p_{d1}>^D \ldots <g_{dm},p_{dm}>^D | <g_i,p_j>^L\right) P\left(<g_i,p_j>^L\right). \tag{2}$$

Assuming independence among the conditional probabilities and using the probabilities calculated in the preceding subsection, (2) transforms to:

$$P_{like}\left(<g_{l1},p_{l1}> | <g_i,p_j>\right) \ldots P_{like}\left(<g_{ln},p_{ln}> | <g_i,p_j>\right) P_{dislike}\left(<g_{d1},p_{d1}> | <g_i,p_j>\right) \ldots P_{dislike}\left(<g_{dm},p_{dm}> | <g_i,p_j>\right) P_{like}\left(<g_i,p_j>\right). \tag{3}$$

Profiling method	Genre/popularity	Preference value
Explicitly preferred	<g3, p2>	4
...	...	...
...	...	...
Explicitly preferred	<g11, p5>	1
Indirectly rated	<g7, p2>	0.78655
Indirectly rated	<g2, p3>	0.23526
...	...	...
Explicitly disliked	<g6, p7>	–1
Explicitly disliked	<g2, p4>	–2
...	...	...

Table 4. Example Profile Table for Test Buyer.

As can be seen, all the results calculated by (3) will have values between 0 and 1. Therefore, if the 3-tuples are sorted in a testing buyer profile in descending order of preference values, all indirectly rated $<g_i, p_j>$ pairs will be located, by the naive Bayesian method, between explicitly preferred pairs and explicitly disliked ones, as shown in Table 4. For a given buyer, all the candidate movies for recommendation are scored according to the preference values, and the highly ranked candidates are finally recommended. For movies with more than one genre, the average of all corresponding $<g_i, p_j>$ pair scores is used for the recommendation.

Experiments

Overview

As shown in Figure 2, 80 percent of the buyers were used to build movie profiles, and the remaining 20 percent were used to test the recommendation performance. The movies were also divided into a training set and a test set, with the training-set movies used to build test-buyer profiles, and the test-set movies used as candidates for recommendation to the buyers. Several experiments were performed using the movie and test-buyer profiles constructed through the procedure introduced earlier. First, a basic experiment was performed for different proportions of divisions between training movies and test movies. This experiment also tested whether utilizing popularity characteristics resulted in improved performance as compared with baseline methods. Second, the recommendation performance was tested under different levels of sparsity of rating data. Third, a cold-starting experiment was performed for imaginary new users by increasing the number of training movies for test buyers from one to ten. In all three experiments, the results were compared to those obtained by the collaborative filtering method, using the formulae introduced earlier. The average rating of top five recommended movies was used as the measure of performance. This measure was used because the widely used mean absolute error (MAE) requires prediction of individual ratings, which the PBR method cannot generate. The average rating of top N recommended items is a good measure of performance in this case because it directly measures buyer satisfaction with the N recommended items.

Performance Using All Available Ratings

In this experiment, the whole set of movies was divided into M_{train} percent training movies and M_{test} percent test movies (*see Figure 2*). Using all the available ratings, the recommendation performance of the PBR method was compared to (1) a random recommendation where five movies were recommended randomly, (2) the collaborative filtering method using the formulae introduced earlier, and (3) a genre-only method applying the same approach as the PBR method using only the genre information of movies—that is, using only $<g_i>$ for movie and buyer profiles.

The proportion of M_{train} was increased from 20 to 80 percent, by 20 per each step. The results were averaged for 20 iterations. On the whole, as shown in Figure 7, the PBR method performed less well than the collaborative filtering method, but was a significant improvement on the random-recommendation and genre-only methods, which demonstrates the effect of using the popularity characteristics. Subsequent experiments used $M_{\text{train}} = 40$ because at that point both collaborative filtering and PBR showed the best performance. This was also a reasonable choice for the cold-starting experiment presented below, since only a relatively small number of profiling movies is needed to simulate a cold-starting situation.

Sparsity

Next, the performance of the PBR method was compared with the collaborative filtering method under different sparsity levels. The following measure of sparsity, adopted in other articles, was used for the experiment [24, 27]:

$$\text{Sparsity} = \frac{(\text{Number of full ratings}) - (\text{Actual number of ratings})}{(\text{Number of full ratings})}.$$

Ratings were randomly chosen and deleted according to the number of required ratings for each sparsity level. The initial sparsity of the MovieLens data with all ratings was 0.877. Thus, the sparsity level was increased from 0.88 to 0.99 by 0.01 each step. The result shown in Figure 8 demonstrates an interesting and valuable property of the PBR method. Under low levels of sparsity, collaborative filtering method outperformed PBR, but when the sparsity level was around 0.96, PBR began to perform as well as or better than collaborative filtering. Under the more severe sparsity levels, such as 0.97, 0.98, and 0.99, the PBR method clearly outperformed the collaborative filtering method. The slopes of the two methods showed different patterns, and it is evident that the performance of the PBR method dropped more gently than

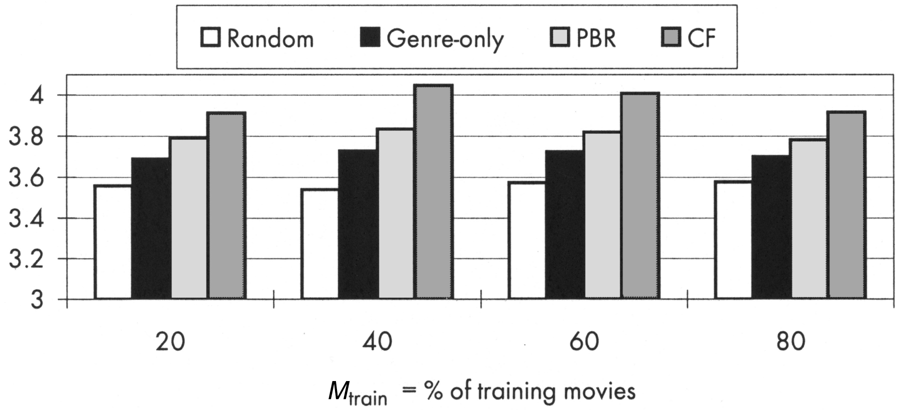


Figure 7. Performance Comparison for Different Proportions of Profiling Movies
y-axis represents the average rating of top five recommended movies.

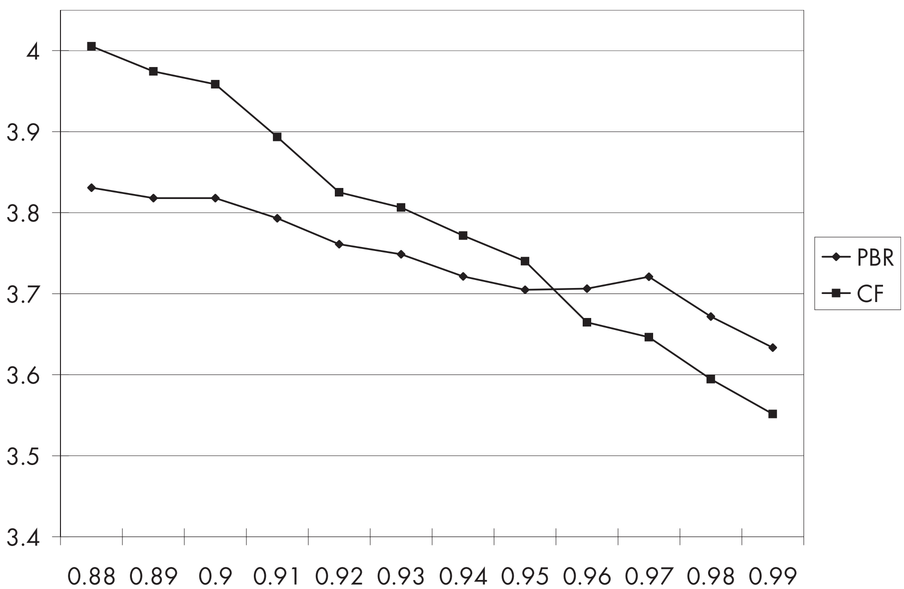


Figure 8. Performance Comparison Under Different Levels of Sparsity
y-axis represents the average rating of top five recommended movies and *x*-axis represents different levels of sparsity.

that of the collaborative filtering method. As an explanation, one may postulate that the PBR method is more robust to sparsity because, for profiling movies, it uses aggregate measures of popularity that may not change sensitively as the number of ratings decreases, whereas the similarity calculation in the collaborative filtering method uses all the ratings individually.

Cold-Starting for New Shoppers with Few Purchase Records

The next experiment tested the performance of PBR under a cold-starting situation for new buyers. The PBR method was applied to an artificial cold-starting situation where new shoppers were presented to the recommender with a very small number of purchases on record or the number of ratings in the case of the MovieLens dataset. For each testing buyer, the number of rated movies for profiling was increased from one to ten, and candidate movies were recommended using the profiles. The same procedure was also applied to the collaborative filtering method for comparison, where, again, one to ten movies were increasingly used to calculate similarity between test buyers and training buyers. In order to get better practical insights from the experiment, it was repeated for 12 sparsity levels ranging from 0.88 to 0.99. The result was very positive for the PBR method, as shown in Figure 9. At all the sparsity levels and most of the purchasing-number levels, the PBR method performed better than, or at least equal to, the collaborative filtering method.

Discussion and Further Research Issues

The experiments described in this paper clearly show the advantages of the PBR method. Although it did not perform as well as the collaborative filtering method when used with full ratings, it showed superior performance under data sparsity and cold-starting situations for new buyers. Since sparsity is very common, and there are always new buyers or buyers with only a few purchases on record in Internet shopping malls, these performance characteristics are very significant.

Apart from these advantages, PBR has several desirable properties. First, it is computationally much less complex than collaborative filtering in real-time recommendation. For a given buyer, collaborative filtering requires similarity calculation with N reference buyers using M_{train} movies where both N and M_{train} can be huge numbers in practice. For the calculation of similarity, $N \times M_{\text{train}}$ iterations are required when using the widely adopted similarity metrics, such as the Pearson correlation coefficient or the cosine measure. After the similarity values are ready, N iterations are again required for each candidate movie in order to generate predictions [25]. In contrast, with the PBR method, the profiling of a buyer requires only M_{train} iterations, and in actual recommendation, it takes constant time for each candidate movie. The complexities of the two methods are summarized in Table 5 using the Big O notation.

Another feature of the PBR method is that the recommendation process is much more understandable. While the collaborative filtering method has only limited ability to explain its recommendation results, the PBR method provides explicit understanding of the buyer's preferences for all the $<g_i, p_j>$ pairs and thus of the fact that each product has been recommended according to these preferences. This capability can enable interesting further use of the recommendation result. For example, if the recommendation result is unsatisfactory for buyers with explicit preferences for certain genres and

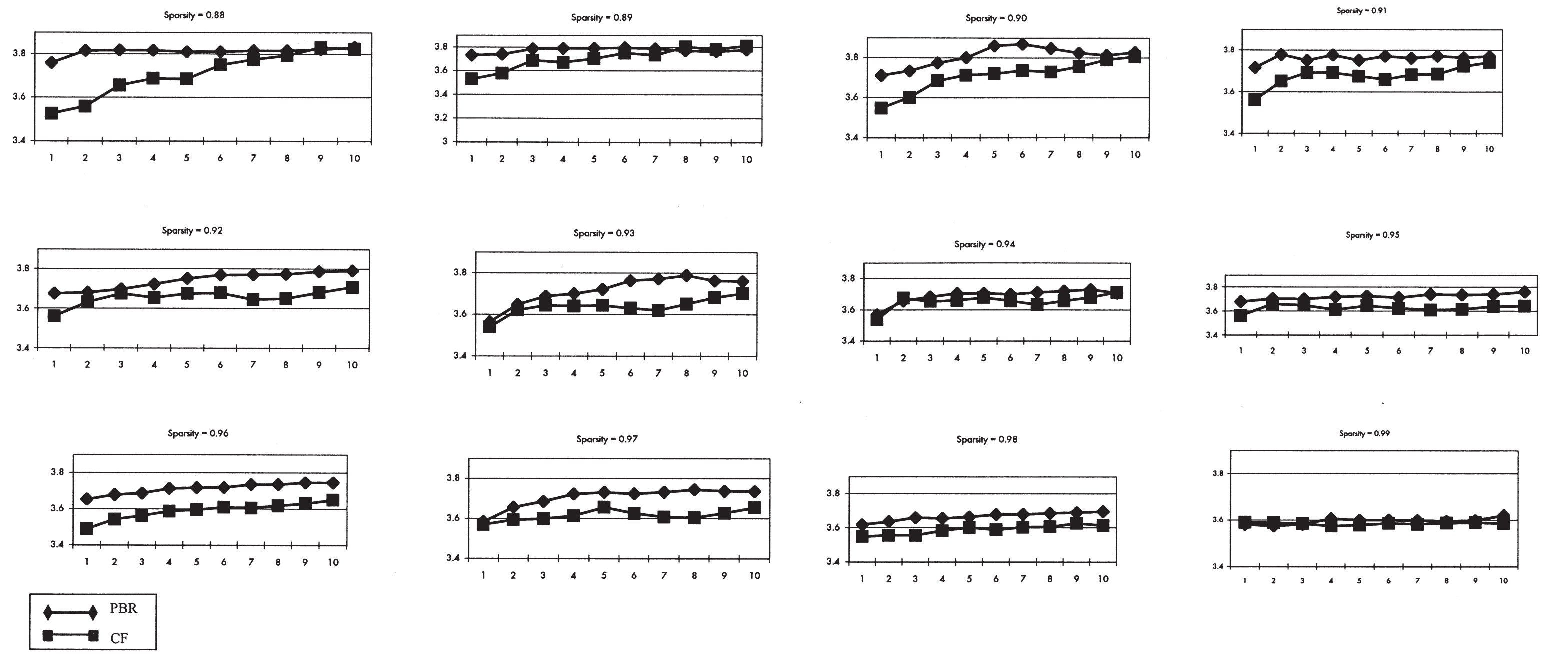


Figure 9. Experiments Under Cold-Starting Situation for New Buyers
X-axis: number of ratings used for profiling (BPR) and similarity calculation (CF); *y*-axis: average of top five recommendations.

			Required number of iterations proportional to problem size when N = 1,000 and M_{train} = 1,000	
	Similarity calculation or training	Recommendation (or prediction)	Similarity calculation or training	Recommendation (or prediction)
Collaborative filtering method	$O(N \times M_{train})$	$O(N)$	1,000,000	1,000
PBR method	$O(M_{train})$	constant time	1,000	1

Table 5. Comparison of Complexities.

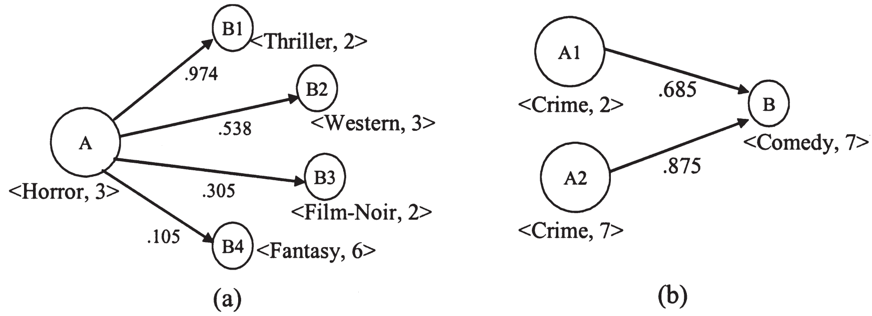


Figure 10. Examples of Probabilities Between <g_i, p_j> Pairs

popularity classes, this may mean that there are not enough candidate products that correspond to the pairs, and one can replenish them strategically. The probabilities calculated for each <g_i, p_j> pair can also open up possibilities for new ways of recommendation. For example, based on the probabilities in Figure 10(a), a simple recommendation strategy can be implemented for a buyer group that has shown explicit preference for horror movies in popularity class 3. Similarly, the probabilities in Figure 10(b) can be utilized to choose a target group for promotion of a specific comedy genre movie that belongs to popularity class 7.

If the three popularity measures for a new product can be estimated with reasonable accuracy, the explicit understandability of the PBR method may improve recommendation performance in another type of cold-starting situation: recommendation of new products with little or no sales history. The estimate is feasible in practice because companies often design or introduce products based on explicit strategy or positioning strongly related to their popularity characteristics. One can also attempt an estimate based on people's opinions, as when one consults a group of experts within or outside of an organization who have experience of marketing similar products. This would be an interesting issue for further research.

The PBR method has some limitations that require further investigation. This paper used 10 as the maximum number of popularity classes to limit the number of probabilities generated, maintain a certain level of reliability, and make the recommendation process understandable. However, changing the maximum number of classes may improve the recommendation performance. Moreover, PBR may not be applicable to domains where the values of the three dimensions of popularity cannot be estimated for each product because of insufficient data or for some other reason. Also, the effectiveness of the partitioning of the popularity space may be dependent upon shopping-mall characteristics and thus require different approaches in different domains. Experiments with different datasets are required for generalization of the PBR method. Finally, since the PBR method generates only the ordering of <g_i, p_j> pairs, it cannot compute individual prediction values for each candidate product for each buyer. This is a shortcoming of the PBR approach compared with collaborative filtering and other recommendation methods, and as such it presents another challenging issue for further research.

In addition to the research issues raised so far, there are several other interesting topics worthy of consideration. First, since the collaborative filtering method performs better when the number of ratings is large enough, a hybrid recommendation method combining collaborative filtering and PBR could be successful. Second, while this study only combined genre information with popularity features, there are many other types of data that could potentially bring synergy when used with popularity features. For example, certain demographic features might be effective in combination with the popularity features because people of different age groups or different cultures may be affected differently by popularity. Third, different recommendation methods could be applied for products in different popularity classes in order to ascertain which method performs best in each class, and this might lead to the development of another hybrid recommendation system. Fourth, different domains or products may demonstrate different effects of using popularity characteristics. Investigating what features of domains and products determine the effectiveness of certain popularity features would be another meaningful research subject.

Conclusion

This paper presents a novel approach to automated product recommendation using the popularity characteristics of products. The proposed model of popularity was combined with movie genre information to build a hybrid method called popularity-based recommendation. When compared with the widely used collaborative filtering method, the PBR system showed significant improvement under data sparsity and cold-starting conditions. This outcome demonstrates that PBR would be of great practical value for recommendation in many Internet shopping malls. The other benefits of PBR include a more understandable recommendation process and less computational complexity.

The most significant academic contribution of the research described in this paper is that it constitutes the first effort, to the author's best knowledge, to develop a model of popularity for recommender systems. Since the model, in essence, presents a new way of utilizing a set of meaningful product features that have been overlooked so far, the author believes that it may open up many interesting opportunities for future research.

REFERENCES

1. Adomavicius, G.; Sankaranarayanan, R.; Sen, S.; and Tuzhilin, A. Incorporating contextual information in recommender systems using a multidimensional approach. *ACM Transactions on Information Systems, 23*, 1 (2005), 103–145.

2. Ahn, A.; Kim, J.K.; Choi, I.Y.; and Cho, Y.H. A personalised recommendation procedure based on dimensionality reduction and Web mining. *International Journal of Internet & Enterprise Management, 2*, 3 (2004), 280–298.

3. Arnould, E.; Price, L.; and Zinkhan, G. *Consumers.* New York: McGraw-Hill, 2002.

4. Berry, M.J.A., and Linoff, G.S. *Data Mining Techniques: For Marketing, Sales, and Customer Relationship Management.* New York: Wiley, 2004.

5. Burke, R. Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction, 12*, 4 (2002), 331–370.

6. Cohen, W.W., and Fan, W. Web-collaborative filtering: Recommending music by crawling the Web. *Computer Networks, 33*, 1–6 (2000), 685–698.

7. Cylogy. *Personalization overview.* www.cylogy.com/library/personalization_overview-kb.pdf.

8. Greco, G.; Greco, S.; and Zumpano, E. Collaborative filtering supporting Web site navigation. *AI Communications, 17*, 3 (2004), 155–166.

9. Han, P.; Xie, B.; Yang, F.; and Shen, R. A scalable p2p recommender system based on distributed collaborative filtering. *Expert Systems with Applications, 27*, 2 (2004), 203–210.

10. Hayes, C., and Cunningham, P. Context boosting collaborative recommendations. *Knowledge-Based Systems, 17*, 2–4 (2004), 131–138.

11. Herlocker, J.L.; Konstan, J.A.; and Riedl, J. Explaining collaborative filtering recommendations. In E.F. Churchill, J. McCarthy, C. Neuwirth, and T. Rodden (eds.), *Proceedings of the 2000 ACM Conference on Computer Supported Cooperative Work.* New York: ACM Press, 2000, pp. 241–250.

12. Herlocker, J.L.; Konstan, J.A.; Terveen, L.G.; and Riedl, J.T. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems, 22*, 1 (2004), 5–53.

13. Hofmann, T. Latent semantic models for collaborative filtering. *ACM Transactions on Information Systems, 22*, 1 (2004), 89–115.

14. Kim, B.-D., and Kim, S.-O. A new recommender system to combine content-based and collaborative filtering systems. *Journal of Database Marketing, 8*, 3 (2001), 244–252.

15. 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), 77–87.

16. Kotler, P., and Armstrong, G. *Principles of Marketing.* Englewood Cliffs, NJ: Prentice Hall International, 1991.

17. Lee, D.-S.; Kim, G.-Y.; and Choi, H.-I. A Web-based collaborative filtering system. *Pattern Recognition, 36*, 2 (2003), 519–526.

18. Li, Y.; Lu, L.; and Li, X. A hybrid collaborative filtering method for multiple-interests and multiple-content recommendation in e-commerce. *Expert Systems with Applications, 28*, 1 (2005), 67–77.

19. Maltz, D., and Ehrlich, E. Pointing the way: Active collaborative filtering. In I.R. Katz, R. Irvin, R.L. Mack, L. Marks, M.B. Rosson, and J. Nielsen (eds.), *Proceedings of the ACM CHI 95 Human Factors in Computing Systems.* New York: ACM Press, 1995, pp. 202–209.

20. Melville, P.; Mooney, R.J.; and Nagarajan, R. Content-boosted collaborative filtering for improved recommendations. In R. Dechter, and R. Sutton (eds.), *Proceedings of the Eighteenth National Conference on Artificial Intelligence.* Menlo Park, CA: AAAI Press, 2002, pp. 187–192.

21. Middleton, S.E.; Alani, H.; Shadbolt, N.R.; and De Roure, D.C. Exploiting synergy between ontologies and recommender systems. Paper presented at the Eleventh International World Wide Web Conference, Honolulu, HI, May 7–11, 2002.

22. Mirzadeh, N.; Ricci, F.; and Bansal, M. Feature selection methods for conversational recommender systems. In X. Chu (ed.), *Proceedings of 2005 IEEE International Conference on e-Technology, e-Commerce and e-Service.* Los Alamitos, CA: IEEE Computer Society, 2005, pp. 772–777.
23. MovieLens. *Movielens dataset.* www.grouplens.org.
24. Papagelis, M.; Plexousakis, D.; Rousidis, I.; and Theoharopoulos, E. Qualitative analysis of user-based and item-based prediction algorithms for recommendation systems. Paper presented at the third Hellenic Data Management Symposium, Athens, June 2004.
25. Papagelis, M.; Rousidis, G.; Plexousakis, D.; and Theoharopoulos, E. Incremental collaborative filtering for highly-scalable recommendation algorithms. In *Proceedings of the 15th International Symposium on Methodologies for Intelligent Systems.* New York: ACM Press, 2005, pp. 553–561.
26. Resnick, P.; Iacovou, N.; Suchak, M.; Bergstrom, P.; and Riedl, J. Grouplens: An open architecture for collaborative filtering of netnews. In R. Furuta and C. Neuwirth (eds.), *Proceedings of ACM 1994 Conference on Computer Supported Cooperative Work.* New York: ACM Press, 1994, pp. 175–186.
27. Sarwar, B.; Karypis, G.; Konstan, J.A.; and Riedl, J. Item-based collaborative filtering recommendation algorithms. In V.Y. Shen, N. Saito, M.R. Lyu, and M.E. Zurko (eds.), *Proceedings of the Tenth International World Wide Web Conference.* New York: ACM Press, 2001, pp. 285–295.
28. Schein, A.I.; Popescul, A.; and Ungar, L.H. Croc: A new evaluation criterion for recommender systems. *Electronic Commerce Research, 5*, 1 (2005), 51–74.
29. Schein, A.I.; Popescul, A.; Ungar, L.H.; and Pennock, D.M. Methods and metrics for cold-start recommendations. In K. Järvelin, M. Beauliew, R. Baeza-Yates, and S.H. Myaeng (eds.), *Proceedings of 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval.* New York: ACM Press, 2002, pp. 253–260.
30. Vezina, R., and Militaru, D. Collaborative filtering: Theoretical positions and a research agenda in marketing. *International Journal of Technology Management, 28*, 1 (2004), 31–45.
31. Witten, I.H., and Frank, E. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementation.* San Francisco: Morgan Kaufmann, 2000.
32. Yu, K.; Schwaighofer, A.; Tresp, V.; Xu, X.; and Kriegel, H.-P. Probabilistic memory-based collaborative filtering. *IEEE Transactions on Knowledge & Data Engineering, 16*, 1 (2004), 56–69.
33. Zeng, C.; Xing, C.-X.; Zhou, L.-Z.; and Zheng, X.-H. Similarity measure and instance selection for collaborative filtering. *International Journal of Electronic Commerce, 8*, 4 (2004), 115–129.

HYUNG JUN AHN (hjahn@waikato.ac.nz) is a senior lecturer at the Waikato Management School, University of Waikato, New Zealand. He received his Ph.D. in 2004 from the Graduate School of Management, Korea Advanced Institute of Science and Technology (KAIST), and taught two courses there as an adjunct professor before joining the Waikato Management School. His research interests include multi-agent systems, intelligent information systems, and e-supply-chain management. He has published in *Decision Support Systems, International Journal of Cooperative Information Systems,* and *Expert Systems with Applications.*