A Hybrid Recommendation Method with Reduced Data for Large-Scale Application

Sang Hyun Choi, Young-Seon Jeong, and Myong K. Jeong

***Abstract*—Most recommendation algorithms attempt to alleviate information overload by identifying which items a user will find worthwhile. Content-based (CB) filtering uses the features of items, whereas collaborative filtering (CF) relies on the opinions of similar customers to recommend items. In addition to these techniques, hybrid methods have also been suggested to improve the performance of recommendation algorithms. However, even though recent hybrid methods have helped to avoid certain limitations of CB and CF, scalability and sparsity are still major problems in large-scale recommendation systems. In order to overcome these problems, this paper proposes a novel hybrid recommendation algorithm HYRED, which combines CF using the modified Pearson's binary correlation coefficients with CB filtering using the generalized distance-to-boundary-based rating. In the proposed recommendation system, the nearest and farthest neighbors of a target customer are utilized to yield a reduced dataset of useful information by avoiding scalability and sparsity problem when confronted by tremendous volumes of data. The use of reduced datasets enables us not only to lessen the computing effort, but also to improve the performance of recommendations. In addition, a generalized method to combine CF and CB system into a hybrid recommendation system is proposed by developing on the normalization metric. We have used this HYRED algorithm to experiment with all possible combination of CF and statistical-learning-based CB filtering. These experiments have shown that the use of reduced datasets saves computational time, and neighbor information improves performance.**

***Index Terms*—Data mining, electronic commerce, hybrid recommendation, information filtering.**

I. Introduction

As INTERNET users surf the Web to find information or products of interest, they are faced with terminologies, such as personalized search, retrieval, filtering, intelligent agent, etc. The prevalence of these terminologies reflects the efforts of commercial sites and researchers to suggest information or products that meet customers' needs. Such recommendations are an essential part of attracting customers. Online companies have the capability to acquire customers' preferences, and then, use them to recommend products on a one-to-one basis in real time and, more importantly, at a much lower cost to company. The software that makes such customized responses possible is commonly called recommendation systems. According to the type of information used to form their responses to customers, recommendation systems can be further categorized as content-based (CB) filtering, collaborative filtering (CF), or hybrid ones.

Personalized recommendation algorithms, such as CF and CB filtering, have been extensively researched. CB recommendation algorithms operate by matching customer interests with product features that describe the characteristics of items. However, this algorithmic approach suffers because of the difficulty of finding a few common features to represent items, and then, getting users' feedback on their own preferences concerning the features. CF algorithms, which are the most successful recommendation technique, have been suggested to overcome this problem. The CF technique relies on the rating of items or on the transactional data of each specific user [10]. CF identifies customers or neighbors, whose purchasing patterns are similar to those of the target user, and recommend items based on the information of similar customers. Despite its success in many settings, the CF approach nevertheless has been reported as having several major limitations, including scalability, sparsity, and new item problems [14]. To address these shortcomings, although researchers have proposed many variations of CB and CF, as well as many hybrid approaches that combine CF and CB, these problems are still significant limitations against improving recommendation systems.

This research proposes a hybrid algorithm combining a modified Pearson's correlation coefficient-based CF and distance-to-boundary (DTB) based CB. The study has focused on developing a hybrid approach that is to suggest a high-quality recommendation method for a tremendous volume of data. HYRED, the hybrid method proposed here, can be used effectively and efficiently for three reasons. First, HYRED proposes the concept of neighborhood in CF to efficiently analyze the transaction data. The use of the nearest and farthest neighbors of a target customer yields a reduced dataset of useful information for solving the scalability problem. The organization of the training dataset has been restricted to the items purchased by a target user and his or her farthest neighbors so that the number of training datasets can be reduced considerably. At the testing step, we have found only items purchased by nearest neighbors and predicted the score of each item. The use of fewer training and testing datasets enables us not only to lessen the computing

Manuscript received February 19, 2009; revised July 22, 2009 and November 17, 2009; accepted March 11, 2010. Date of publication April 29, 2010; date of current version August 18, 2010. This work was supported by the Ministry of Education, Science, Technology and Korea Institute for Advancement of Technology under the Human Resource Training Project for Regional Innovation. This paper was recommended by Associate Editor M. Funabashi.

S. H. Choi is with the Department of Industrial and Systems Engineering, Engineering Research Institute, Gyeongsang National University, Jinju 660-701, Korea (e-mail: chois@gsnu.ac.kr).

Y.-S. Jeong is with the Department of Industrial and Systems Engineering, Rutgers University, New Brunswick, NJ 08854 USA (e-mail: ysjeong@eden.rutgers.edu).

M. K. Jeong is with the Department of Industrial and Systems Engineering and RUTCOR, Rutgers University, New Brunswick, NJ 08854 USA and the Department of Industrial and Systems Engineering, KAIST, Daejon 305-701, Korea (e-mail: mjeong@rci.rutgers.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSMCC.2010.2046036

effort, but also to improve the performance of recommendations. The processes of filtering irrelevant data by using the neighborhood concept of CF make it possible to consider the items that are likely to be purchased by a target user.

Second, we propose the generalized rating system based on the distance of an item to the decision boundary of a classifier. In this concept, the item closer to the class of purchased items may have a higher probability of being sold. The experiment shows that the DTB-based rating improves the performance of recommendation than either pure CB or CF. Our algorithm has calculated the distance from alternative items to the ones purchased by a target user, whose items have been selected from statistical classifiers. This selection method used neighborhood information and delivered better performance than was gained with pure CF.

Finally, we propose a generalized hybrid recommendation algorithm by using a weighted coefficient in which the DTB and CF methods are special cases of our generalized algorithm. The weighting scheme makes this algorithm adequate for generalized applications, and HYRED is flexible enough for application with any available datasets. Moreover, HYRED, when weighting is properly valued, has yielded better results than pure DTB, pure CF, and simple combined hybrid method.

II. Background and Prior Research

Many large-scale applications such as mobile content Websites involve huge numbers of items and customers. A particularly popular form of mobile content is the wallpaper image for mobile phones. Although the market for this content is growing rapidly as related technologies evolve, customers who use mobile phones have been experiencing much frustration when searching for the images they want, owing to inefficient sequential search. The mobile contents are usually very cheap and frequently purchased by customers. Therefore, it is time consuming and inefficient to additionally collect and analyze customer behavioral information, and then, recommend contents. It is certainly desirable for content providers to recommend mobile contents to customer using the minimal information of transactional and item feature data.

In the case of transactional data in this research, a customer is represented by the Boolean feature vector of a set of items. The value of each element in the vector is determined by whether this customer has purchased the corresponding item in past transactions. The value of "1" simply indicates that a purchase has occurred, and "0" indicates that no purchase has occurred. The variable y_{ji} represents whether the jth item I_j has been purchased by the ith customer c_i. When multiple customers are involved, a matrix composed of vectors representing N customers' transactions can be made to capture past transactions [7], [15], [17]. As shown in Table I, this matrix is called the customer profile matrix. In such mobile content business, even when many transactions have been recorded, the customer profile matrix can still be extremely sparse, i.e., very few elements whose value is 1 are in the profile matrix. This problem, commonly referred to as the sparsity problem, has a major negative impact on the effectiveness of a CF approach [14].

TABLE I
Customer Profile Matrix

	Transactional data/CF			
Item	c_1	…	c_i	c_N
I_1	y_{11}	$y_{ji}\in\{0 \text{ or}1\}$		y_{1N}
I_2	y_{21}			y_{2N}
I_j…	…			…
I_M	y_{M1}			y_{MN}

TABLE II
Item Feature Matrix

	Item feature/CB			
Item	f_1	…	f_l	f_L
I_1	x_{11}	$x_{jl}\in[0,1]$		x_{1L}
I_2	x_{21}			x_{2L}
I_j…	…			…
I_M	x_{M1}			x_{ML}

In the case of the feature data of an item, the item can be described by a set of features. Each item has its own specific values. The variable x_{jl} represents the value of jth item with respect to lth feature. The values are classified numerically or by category. The categorical values can be assigned to numerical ones, and then, normalized to a value between 0 and 1. The matrix that includes a set of items being considered is referred to as the item feature matrix. Table II shows the item feature matrix.

CB approaches use feature descriptions of the rated items as a means to learn the relationship between the ratings of a single user and the descriptions of the items rated. The information required for CB approaches can be found in Table II. To reach a recommendation, CF approaches use the rating or transactional data of each of the customers on a set of items. Table I shows the information for a collaborative approach. The set of data used in our hybrid approach HYRED consists of the transactional data and the set of item features that describe the characteristics of items.

A. *Collaborative Filtering*

Much researches related to CF recommendations have been done [2], [16], [18], [19], [24], [29]. An excellent survey of different recommendation systems for various applications can be found in [1] and [22]. CF systems try to predict the utility of items for a particular user based on the items previously rated by other users [12]. The utility of an item for a user is estimated based on the utilities assigned to the item by those users "similar" to the user. CF systems recommend products to a target customer based on the opinions or transactions of other similar customers. These systems employ statistical techniques to find a set of customers, known as neighbors, who have a history of agreeing with the target customer. Basu *et al.* [4] proposed that the farthest-first traversal was a good heuristic for initialization in prototype-based partitional clustering algorithms. The goal in the farthest-first traversal is to find the points that are maximally separated from each other in terms of a given distance function. The farthest-first traversal is focusing on the distances between any two points within cluster, while our farthest neighbor method is to deal with the correlation between one target point (or customer) and remaining ones with binary data in which distance-based functions are not appropriate.

Once neighbors of the target customer are found, these systems use several algorithms to produce recommendation items. The CF process is generally divided into problem representation, neighborhood formation, and recommendation generation. The most important step in the CF algorithms involves computing the similarity, $\mathrm{sim}(c_T, c_i)$ between a target user c_T and a number of other users c_i, where $i = 1, \ldots, N$ [9], [21]. The main goal of neighborhood formation is to find an ordered list of K users $C = \{c_1, c_2, \ldots, c_K\}$, $K \leq N$, for each target user c_T, so that $c_T \in N$ and $\mathrm{sim}(c_T, c_1)$ are the maximum, $\mathrm{sim}(c_T, c_2)$ is the next maximum, etc. To compute the similarity, many researchers [21] have been using the following normal correlation coefficient formula:

$$\begin{aligned}\mathrm{sim}(c_T, c_i) &= \mathrm{Corr}(c_T, c_i)\\ &= \frac{\sum_j (y_{jT} - \overline{y_T})(y_{ji} - \overline{y_i})}{\sqrt{\sum_j (y_{jT} - \overline{y_T})^2 \sum_j (y_{ji} - \overline{y_i})^2}}. \qquad (1)\end{aligned}$$

B. CB Filtering

The utility of an item for a user is estimated based on the similarity of an item to other items purchased by the user. The CB approach to recommendation has its roots in information retrieval and information filtering [11]. In these two areas of research, items are characterized by a set of attributes or features. A recommendation is usually computed by extracting a set of features from each item and these features are then used to determine the appropriateness of the item for recommendation purposes. The best-known measures for classifying such items are the Naïve Bayes, vector space model, and K-nearest neighbor (KNN) approaches [23]. The KNN classifier is memory-based and requires no model to be fit [13]. A new item is classified according to the most common class among the KNNs most similar to it. The distance between a query item $\mathbf{x_j} = (x_{j1}, x_{j2}, \ldots, x_{jL})$ and R training items $\mathbf{x_{(r)}} = (x_{r1}, x_{r2}, \ldots, x_{rL})$, $r = 1, \ldots, R$ can be calculated by Euclidean distance as follows:

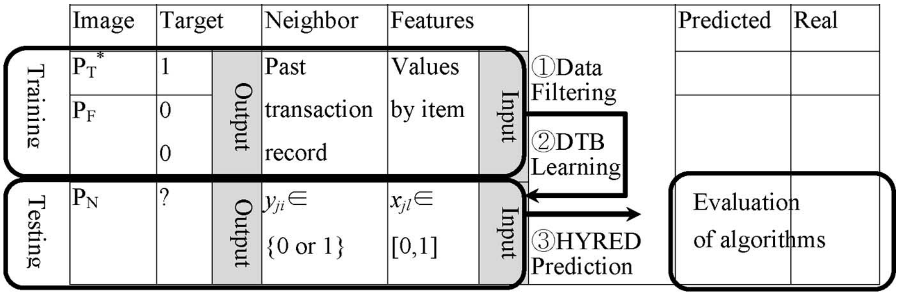
$$d(\mathbf{x_j}, \mathbf{x_{(r)}}) = \sqrt{\sum_{l=1}^{L} (x_{jl} - x_{(r)l})^2}. \qquad (2)$$

Sometimes there are a large number of product features; consequently, existing systems rely heavily on preprocessing steps that select or extract important features from products. CB analyzes the content of the information source or of the features of products that have been rated or purchased. This analysis is used to create a profile of the user's interests in terms of content regularities in the information that was highly rated or of the features of products that were purchased. Tong and Chang [27] suggested the use of a support vector machine (SVM) to conduct effective relevance feedback for image retrieval. The algorithm selected the most informative images so as to query a user and quickly trained a boundary that separated the rest of the dataset from those images that satisfied the user's query concept. Shin and Cho [25] applied SVM to response modeling in direct marketing. Cheung *et al.* [6] proposed the use of the SVM as one among several statistical classifiers. The performance of the SVM has been related to the margin with which it separates the data, not to input dimensionality. The SVM has proved superior in experiments on a number of high-dimensional datasets. However, the researchers in these experiments have not mentioned other classifiers or included them in their experiments. In this paper, we will present several popular classifiers for CB, including the SVM.

C. Hybrid Approach

Several recommendation systems have used their own hybrid approaches that combine the CF and CB methods, and thus, helped to avoid certain limitations of CF and of CB. Burke [5] made a survey of existing recommender techniques and systems, and also presented and evaluated a hybrid recommender system. The research analyzed all the possible combinations for producing hybrid systems and presented and evaluated a new hybrid: a knowledge-based/collaborative cascade hybrid as an evolution of an existing knowledge-based recommender system for restaurants. The results indicated that this hybrid model performed better than the pure recommender system. As a hybrid system, Smyth and Cotter [26] suggested a personalization system that used the learned user profiles to target appropriate content items for individual end users. By integrating both CB and CF strategies, a personalization engine provided a powerful personalization solution. The CB recommendation is performed based on program records that consisted of a set of features, including program name, genre, country of origin, cast, studio, director, writer, etc., and CF recommendation on the judgment of user's interest in programs in the form of grading icons.

Claypool *et al.* [8] presented a hybrid approach based on a prediction on a weighted average of the CB prediction and the CF prediction. The weights of the CB and CF predictions are determined on a per-user basis, thus allowing the system to determine the optimum mix of CB and CF recommendations for each user. Pazzani [20] proposed a hybrid method called the collaboration via content model. In collaboration via content model, the CB profile of each user was exploited to detect similarities among users. The user's CB profile contained weights for the terms that indicate that a user would like an object. As in CF, the prediction was determined by a weighted average of all users' predictions for that item or product by using the correlation between profiles as the weight. Another hybrid system, VISCORS, combined CF with CB image retrieval [15]. The CF module produced the initial list of recommended items. The CB image-retrieval module refined the query to learn the customer's current preferences. Most of the previous researches into using hybrid methods have required either additional information from



*: P_T : A set of images purchased by the target user

P_F: A set of images purchased by farthest neighbors

P_N: A set of images purchased by nearest neighbors

Fig. 1. Framework of HYRED.

users as well as transactional data and product feature data, or a great deal of feedback from a specific user. Thus, these methods have been difficult to apply to real-life problems because the additional information is hard to get. In contrast, the research in this paper has used only recorded transactional data and product feature information in focusing on the problem.

III. Hybrid Recommendation Approach With Reduced data

The purpose of this research is to propose a new hybrid recommendation approach based on only the information of transactional and item feature data that can be easily obtained from corporate transactional system. The simplest combined hybrid would be a linear combination of recommendation scores. Our research has focused on developing a hybrid approach to use them to make high-quality recommendations in the face of a tremendous volume of data. HYRED has used the concept of nearest and farthest neighbors that yields a reduced dataset of useful information for solving the scalability problem. The framework of HYRED combines both CF and DTB for CB filtering. In this framework, the CF technique has been used to overcome the sparsity problem caused by a few "1" values in transactional data, and DTB has been applied to classify the images, based on the feature values of each one, into one of two classes.

The DTB algorithms require the most informative images so as to quickly learn a boundary to separate the images that the target customer has purchased. The use of DTB algorithms to classify the images as purchased or nonpurchased confronts us with the problem of overfitting because the number of purchased images is much smaller than the nonpurchased images.

We get the reduced customer profile matrix in Fig. 1 by finding the nearest and farthest neighbors of a target customer. The matrix contains the dataset of useful information for solving the scalability problem. As shown in the numbered step of Fig. 1, HYRED has performed three steps. The data-filtering step, which occurs first, yields a reduced dataset, then the DTB learning step determines the characteristics of purchased images, and subsequently, moves on to the DTB prediction step to suggest a set of images among the images purchased by the neighbors of a target customer.

A. Binary Similarity Measure for Neighbor Formation

We have modified a binary similarity measure using the Pearson product-moment correlation coefficient because the customer profile consists of only transactional data represented by a binary value, either 0 or 1 [3]. For binary data, this similarity between two users c_T and c_i is computed by the following formula:

$$\begin{aligned}\mathrm{sim}(c_T, c_i) &= \mathrm{Corr}(c_T, c_i)\\ &= \frac{AD - BC}{\sqrt{(A+B)(A+C)(D+B)(D+C)}}\end{aligned} \tag{3}$$

where A, B, C, and D, respectively, represent the number of products purchased by the two users to be compared (A), the number of products purchased by the first, but not the second (B), those purchased by the second, but not the first (C), those not purchased by either user (D). The similarity between two users is based on the transactional data in the customer profile matrix and is used in this research as a tool for filtering the data at the training and testing stages.

B. DTB for CB Filtering

The purpose of DTB learning is to find the decision boundary separating items purchased by the target user and to recommend items based on the distance between an item and the decision boundary. New items closer to the class of purchased items may have a higher possibility of being sold soon. The distance of an item to the SVM decision boundary has been used for the rating of items in CB systems [6], [25]. However, DTB from other classifiers sometimes may give better performance. This paper generalizes the DTB concept for other classifiers for its applications in recommendation systems.

Suppose a target customer and the farthest neighbors have purchased m items, where $n(P_T)+n(P_F) = m$, and $n(\odot)$ is the number of elements in set $\odot$. Then,we can get a training dataset with m observations$\{(x_j, y_j)\}_{j=1}^{m}$, where $x_j \in \mathbf{X} \subseteq \Re^L$ is an image feature vector with its corresponding binary class label

$y_j \in \{0, 1\}$ and the value of "1" represents the purchased item, while the value of "0" the nonpurchased item. The distance between an item and a decision boundary can be calculated as follows [25]:

$$\mathrm{dist}(x_j, f_j) = \frac{|f(\mathbf{x}, \alpha, b)|}{\left|\sum_{j \in \mathrm{SV}s} \alpha_j y_j x_j\right|^2} \tag{4}$$

where

$$f(\mathbf{x}, \alpha, b) = w\mathbf{x} + b = \sum_{j=1}^{m} \alpha_j y_j \mathbf{x}_j^T x + b. \tag{5}$$

The distance of an item to a SVM decision boundary can be used in computing rankings for recommendations.

We can generalize this idea to other popular classifiers and present the formula of the DTB for the linear discriminant analysis (LDA) and the quadratic discriminant analysis (QDA). Suppose the vectors $\boldsymbol{x}_j^1$ and $\boldsymbol{x}_j^0$ be the vector of feature values of items purchased and nonpurchased, respectively. The distance of an item to a LDA decision boundary can be calculated as follows:

$$\mathrm{DB}_{\mathrm{LDA}}(\mathbf{x}) = (\boldsymbol{\mu}_j^1 - \boldsymbol{\mu}_j^0)^T \sum\nolimits^{-1} x + C \tag{6}$$

where

$$\boldsymbol{\mu}_j^1 = [\mu_1^1, \mu_2^1, \ldots, \mu_L^1], \quad \boldsymbol{\mu}_j^0 = [\mu_1^0, \mu_2^0, \ldots, \mu_L^0]$$

$$\sum = \begin{bmatrix} \sigma_{11} & \cdots & \sigma_{1L} \\ \vdots & \ddots & \vdots \\ \sigma_{L1} & \ldots & \sigma_{LL} \end{bmatrix}$$

$$C = -(1/2)(\boldsymbol{\mu}_j^1 + \boldsymbol{\mu}_j^0)^T \sum\nolimits^{-1} (\boldsymbol{\mu}_j^1 - \boldsymbol{\mu}_j^0) + \log(p(w_j^1)/p(w_j^0))$$

and $\boldsymbol{\mu}_j^1$ and $p(\omega_j^1)$ are the mean vector and prior probabilities of feature values of purchased items, respectively. $\boldsymbol{\mu}_j^0$ and $p(\omega_j^0)$ are the mean vector and prior probabilities of nonpurchased items. Σ is the common covariance matrix of two classes, purchased and nonpurchased items, with respect to each of features. In case of a QDA, the DTB can be obtained as a following quadratic equation:

$$\begin{aligned} \mathrm{DB}_{\mathrm{QDA}}(\mathbf{x}) = \frac{1}{2}\Bigg\{ &-(x - \boldsymbol{\mu}_j^1)^T \sum\nolimits_{\text{purchased}}^{-1} (x - \boldsymbol{\mu}_j^1) \\ &+ (x - \boldsymbol{\mu}_j^0)^T \sum\nolimits_{\text{nonpurchased}}^{-1} (x - \boldsymbol{\mu}_j^0)\Bigg\} + C \end{aligned} \tag{7}$$

where constant $C = (1/2)\log(|\sum_{\text{nonpurchased}}|/|\sum_{\text{purchased}}|) + \log(p(w_j^1)/p(w_j^0))$.

C. Procedure of HYRED

The framework of HYRED is implemented in practice via a detailed three-stage procedure. Based on the CF concept, HYRED focuses on the effective reduction of a huge dataset and through recommendation that use the DTB concept on efficient prediction. This effectiveness and efficiency were achieved by transforming a huge real-world dataset into a useful dataset through the following three steps as shown in Fig. 1.

Step 1: Data filtering. This step is to efficiently reduce the computational complexity caused by scalability and the difficulty of finding similarities between users that arises because of sparsity. In the case of sparsity, it is highly probable that the similarity between two given users is zero, which is useless. Even for pairs of users that are positively correlated, such correlation measures may be unreliable. In this step, the CF algorithm yields a reduced dataset. Based on the reduced dataset, the DTB learning algorithm is applied to the large-scale application. The reduced datasets are made up of the items purchased by each of the target users (P_T), his or her nearest neighbors (P_N), and farthest neighbors (P_F). The nearest and farthest neighbors are found by calculating Pearson's product moment coefficients between two users. The larger its value, the closer the neighbor is, and the smaller the value, the farther away. We assume that the nearest neighbor usually has common characteristics in buying an item, and in the opposite direction, assume that the farthest neighbor displays dissimilar characteristics. The set of images purchased by the target user and farthest neighbors will be used for training at the DTB learning step. Finally, the set of images purchased by the nearest and farthest neighbors will be used for the purpose of evaluating the algorithms at the step of HYRED prediction.

Step 1.1: Make a customer profile. An initial input matrix that consists of the customer profile matrix and the item feature matrix is created. The customer profile matrix of transactional data is applied to the CF algorithm. DTB algorithms use the item feature matrix.

Step 1.2: Compute the similarity between a target user and others. For all customer, find a customer similarity matrix based on transactional data consisting of binary values whose element is in the symmetric matrix $S_{N \times N}$. The matrix represents the degree of similarity between a customer and other customers. The larger the value, the more similarity becomes. Similarity is calculated using the Pearson product-moment correlation coefficient

Step 1.3: Obtain a reduced dataset. We obtain the farthest neighbors and nearest ones having the smallest and the largest coefficients, respectively. The target dataset is reduced to those sets of images purchased by the target user and farthest and nearest neighbors. Fig. 1 shows the reduced matrix.

Step 2: DTB learning. In this step, we want to train the DTB to find the decision boundary of items purchased by the target user. The training data is made up of the items purchased by the target user and the farthest neighbors. The items purchased by the target user are assigned a 1, and items bought by the farthest neighbors are assigned a 0. Each item has its own specific feature values. The purpose of the DTB learning is to find the decision boundary separating two kinds of classes. We shall consider DTB in the binary classification setting. Given training data $\{x_1, \ldots, x_M\}$ that are vectors in some space $\mathbf{X} \subseteq \Re^L$ and their labels $\{y_1, \ldots, y_M\}$ where $y_j \in \{0, 1\}$, DTB is the function that separates the training data. All vectors lying on one side of the hyperplane are labeled as 0, and all vectors lying on the other side are labeled as 1.

Step 2.1: Setup for training. The data manipulation tasks for training DTB algorithms are performed in this step. The feature values of items consisting of P_T and P_F are extracted and used as input. The output values are set to 0 and 1 for P_F and P_T, respectively.

Step 2.2: Do DTB learning. Learning is performed with input $\{x_{jl}\}$ and output $\{y_{ji}\}$. The purpose of the DTB learning is to find the hyperplane separating the sets of P_F and P_T. Among the products not purchased until now by c_T, but purchased earlier by c_T's neighbors, find the products that are likely to be purchased by c_T.

Step 3: HYRED prediction. This step is to predict the rating scores of the tested images by consolidating both the CF and DTB scores. The CF score means the correlation coefficients between two customers, whereas the DTB score indicates the distance between the decision boundary and an item. To integrate the CF and DTB scores, the calculation of normalization should require harmonization of the scale unit between the two scores. To obtain the normalized CF score and DTB score of the jth image, we, respectively, propose new metrics as follows:

$$S^{\mathrm{CF}}(I_j) = \frac{\sum_{i \in P_N} S_{Ti} y_{ji}}{\sum_j \sum_{i \in P_N} S_{Ti} y_{ji}} \tag{8}$$

$$S^{\mathrm{DTB}}(I_j) = \frac{f(I_j)}{\sqrt{\sum_j f(I_j)^2}} \tag{9}$$

where i is in the set of nearest neighbors to the target user, $S_{Ti} = \mathrm{sim}(c_T, c_i)$ is the similarity between a target user and the ith user, as shown in (3), and $f(I_j)$ is the distance between the decision boundary and the jth image.

The testing data for score prediction is made up of the items that have been purchased by KNNs and have not been purchased by a target customer. To predict the rating values for the items, we suggest a new hybrid measure of CF and a CB classifier as follows:

$$C(I_j) = S^{\mathrm{CF}}(I_j) + \lambda^* S^{\mathrm{DTB}}(I_j) \qquad \forall_j \in P_N \tag{10}$$

where $\lambda \in [0, \infty]$ is constant and the weight coefficient between CF score and DTB score.

Step 3.1: Data setup for prediction. The items consisting of P_N are extracted and used for prediction.

Step 3.2: Compute CF score. To obtain the CF score for each of the items in testing dataset P_N, we calculate the aggregated sum of the similarity values between users and the purchasing record for each item. Then, we obtain the normalized values by (8).

Step 3.3: Compute DTB score. We calculate the DTB score for each item by testing the dataset by using the function that describes the hyperplane and is found at step 2. Then, the normalized values of (9) are obtained.

Step 3.4: Compute HYRED score. Finally, we calculate the hybrid score using (10) after computing the CF and DTB scores.

Step 3.5: Recommend items. HYRED suggests the most valued products, in the order of magnitude of the aggregated values computed at step 3.4 and also evaluates the performance measures of accuracy and computational time.

Remark 1: Conventional CF and CB measures are special cases in our generalized methodology. For example, when $\lambda = 0$, the $C(I_j)$ will be the CF measure, but with $\lambda = \infty$, $C(I_j)$ is equal to DTB measure. By choosing the appropriate weight, our proposed method can achieve improved performance.

In the case of the CF score, we use the usual CF algorithm, which requires a similarity measure for binary transaction data; in this case, similarity measure is Pearson's product-moment correlation coefficient. The similarity between two users is based on the transaction data in the customer profile matrix. In collaborative recommendation systems, KNNs have been selected in the order of similarity values. On the other hand, in order to obtain the DTB score estimation based on classifier training, we use the distance from the decision boundary to each point of the items. After determining the DTB score of an item that has a feature vector, the distance from the boundary to each image is used as an estimate of the customer's preference. The larger its value, the more preferable the product is. We compute the distance of each image by using the decision boundary that the DTB training step determined as applicable for the set of images that KNNs purchased.

In the formula of the HYRED measure, the weight coefficient λ is used to combine the values of the CF and DTB scores. The weight coefficient is not associated with a particular method, but associated instead with a linear aggregation. Its generality lies in the fact that by selecting weights, we can implement different recommendation methods. In the application domain that is well represented by the features of items, we take a larger value of λ on the DTB score. In the application domain in which neighbors well represent the transaction pattern of users, we take a smaller value in which λ is less than 1 on the CF score. In practice, λ can be chosen based on the performance of HYRED on the validation data.

IV. Evaluation

A. Experimental Data

Multimedia such as wallpaper, music, character images, and similar material has been the most popular content among the huge inventory of material available from the fast-growing mobile Web. We performed an experimental study of a comprehensive recommendation approach using real-world transactional data of mobile phone wallpaper images that were characterized by ten features. Data from 8776 wallpaper images purchased from a Korean code division multiple access (CDMA) carrier were obtained for the study. The experimental data consisted of 1) actual purchasing records of 1921 customers for 8776 images and 2) image characteristics. Each customer had bought from 3 to 89 images over six months that began June 1, 2004. The characteristics of these images are represented by nine color features based on the HSV (hue, saturation, and value of color) that Kim *et al.* [15] used. We added a characteristic, such as movie star, cartoon character, landscape, or other key word to the subject category of the images. The customer was then represented by a Boolean feature vector of 8776 images. The value for each

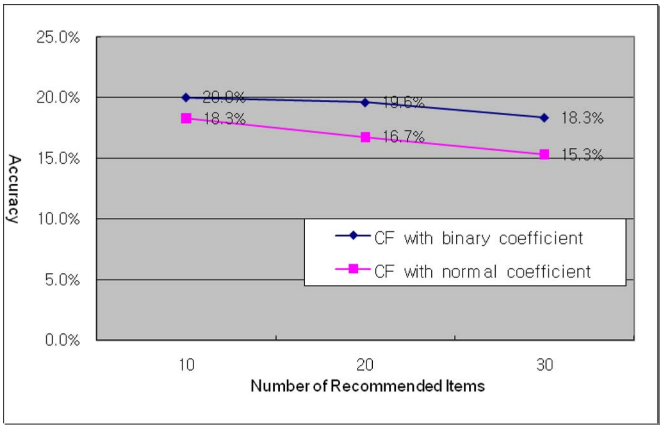


Fig. 2. Average accuracy of CF using normal and binary coefficient.

element was determined by whether this consumer had purchased the corresponding image in past transactions. Overall, our sample data were very sparse, with values of 55 321 (0.33%) among 1921 × 8776 possible purchases.

B. Evaluation Measures

The main focus in evaluating the effectiveness of recommendation algorithms is to determine the extent to which the recommended products match the actual customer purchase list. Accuracy has been widely used in recommendation systems research to evaluate the quality of the recommendations [27]. Accuracy is defined as the ratio of the number of products in both the test set and the top-N set to the number of products in the top-N set. We calculated the computational time for each method with a goal of evaluating the efficiency of each one.

We used a two-tiered experimental design to show that the consolidation procedure used for HYRED enables us to reduce the real-world large-scale data of both customers and items and also improve performance. First, we compared the performance of CF and DTB for the full dataset and for the reduced dataset in which there are four methods for DTB. Second, we compared the performance of HYRED according to the value of a HYRED weight within the setting of an efficient dataset. Because we have the transaction history during the T period, we used the history during the first *T–t* period as training data and the transactions during the latest rest period as validation data.

C. Experimental Results

At the first experiment, we have evaluated two similarity measures using the transactional data of wallpaper images. We split the real transaction data into training and testing portion, and use testing data to compute accuracy values to determine a suitable similarity measure. We have performed these experiments and the results are presented in Fig. 2. It shows that CF recommendation method based on a binary similarity measure has achieved the better results than the normal measure-based method. The modified similarity measure will be used as a mean to get neighbors because it is more suitable to the binary transaction data.

We conducted experimental studies to compare the results of CF, DTB, a simple combined hybrid method, and HYRED. Most of existing hybrid approaches required the information on ratings from users of each item in the set of items or the demographic information of users. To our best knowledge, there are little available recommendation systems that use both binary transactional and item feature data. We implemented the simple combined hybrid method that was able to recommend items using only the binary transaction and item feature data.

To evaluate the effect of reduced data, we performed our experiments by comparing the results of CF and CB using full and reduced data, respectively. The next experiment was to compare the results of HYRED with respect to the value range of the weight coefficient and the number of items recommended. We compared the results of reduced CF, reduced CB, HYRED, and a simple combined hybrid method that used a linear combination of recommendation scores. Finally, we compared the computational times of different algorithms. All our experiments were implemented using MATLAB 7.04. We ran all our experiments on a Windows XP-based PC with Intel Pentium IV processor having 3 GB of RAM.

First, we conducted experimental studies to compare CF, DTB, and HYRED for both full and reduced datasets in which DTB included KNN, LDA, QDA, and SVM. The experimental data consisted of transactional data from 1921 customers and all item profile matrixes. In order to measure the accuracy and computational time of each algorithm, the experiment involved the top 30 customers who had previously purchased more than 65 images from the company. We used 60% of the transaction data for training and the rest for testing. The organization of the training dataset was restricted to the items purchased by a target user and his (or her) farthest neighbors. In this research, we used one training dataset. If we had large numbers of both items purchased by target user and neighbors, then it was possible to split the training dataset into several dataset for the cross validation. The K users with the most similarity values were selected as nearest neighbors. The M users with the least values were selected as farthest neighbors. The used threshold values in this research are $K = 5$ and $M = 20$. With the use of these numbers, we obtained the improved performance in most of experiments. After we conducted the experiments, we evaluated the measures customer by customer and calculated average customer values. First of all, Figs. 3 and 4 show the average accuracies for CF and DTB using full datasets and reduced datasets. Overall, the recommendation system using a reduced dataset shows better performance than the one using a full dataset. Fig. 3 shows the average accuracy for the results of CFs using full and reduced datasets with respect to the number of items recommended. The accuracies of CF in using reduced datasets were much higher than those of CF using full datasets. This difference is because use of the reduced dataset got rid of the useless data of irrelevant customers and gave higher weight to relevant customers.

Fig. 4 shows the average accuracy for the results of DTBs using full and reduced datasets. The accuracies of DTBs using reduced datasets were much higher than those of DTBs using full datasets. In particular, the results of SVM and KNN were better than those of LDA and QDA. Although most of recent

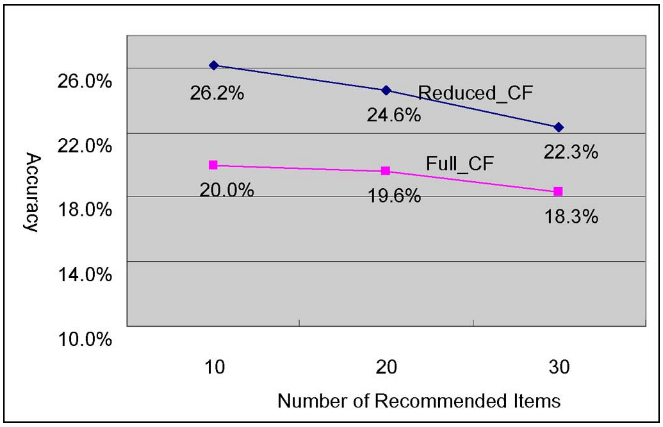


Fig. 3. Average accuracy of CF using full and reduced datasets.

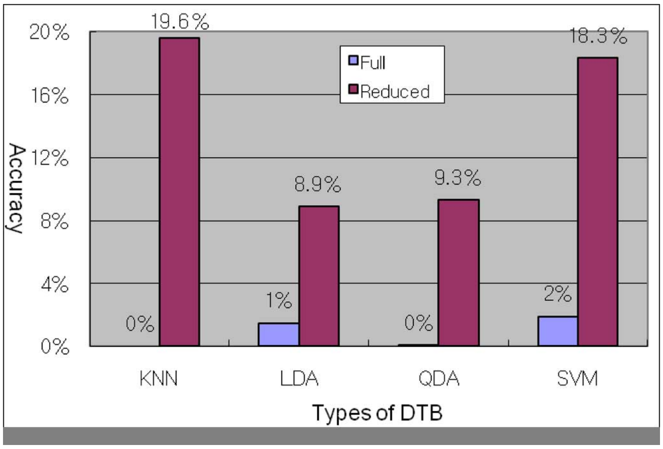
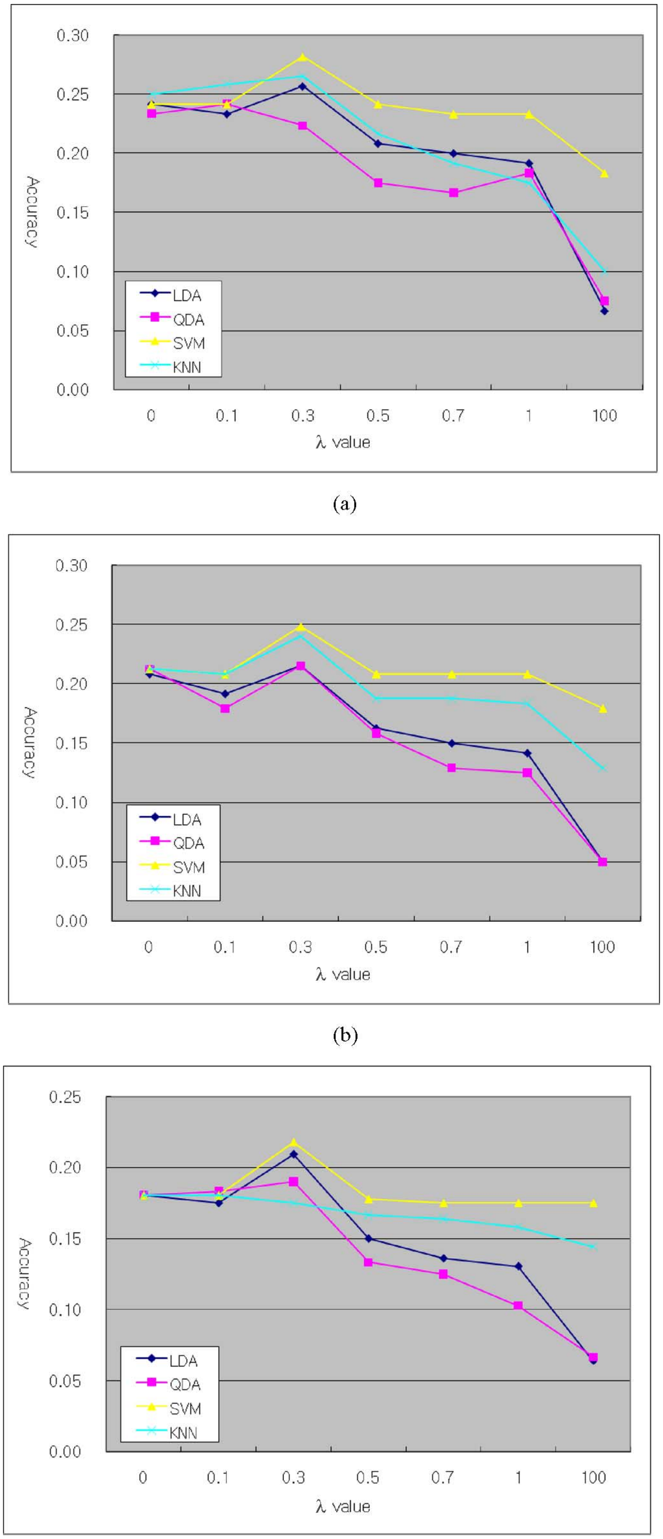


Fig. 4. Average accuracy of DTB using full and reduced datasets.



(a)

(b)

(c)

Fig. 5. Average accuracies of the HYRED for different numbers of items recommended. (a) Number of items recommended is 10. (b) Number of items recommended is 20. (c) Number of items recommended is 30.

recommendation papers [6], [25], [27], [30] proposed SVM as a promising alternative among CB algorithms, in this experiment both SVM and KNN showed the best performance among DTBs.

Fig. 5(a)–(c) shows the average accuracy for the HYRED with respect to the range of the weight coefficient and the number of items recommended. We have used reduced dataset and the hybrid method that combines CF and each of CB methods at the specific λ value. The accuracies of hybrid method were best when the number of items recommended is 10. The value of the weight coefficient λ has significant impact on the prediction quality. The result shows that the accuracies of hybrid method were best when the weight coefficient is 0.3. In particular, the results of SVM and KNN were better than those of LDA and QDA. In the viewpoint of the number of items recommended, the average accuracy was worsened with increases in the number of items to be analyzed. From the experiment, we set the value of λ to 0.3 and the number of items recommended to 10 as optimum values for our subsequent experiments.

We compared the results of CF and DTBs with reduced data, HYRED, and a simple combined hybrid method that use a linear combination of recommendation scores. In the simple combined hybrid method, CF score was obtained by applying the general correlation similarity measure, and CB score was the best one among scores obtained by applying DTBs with full dataset. To determine the weights of CF and CB scores, we performed an experiment where we varied the value of weights to be used and

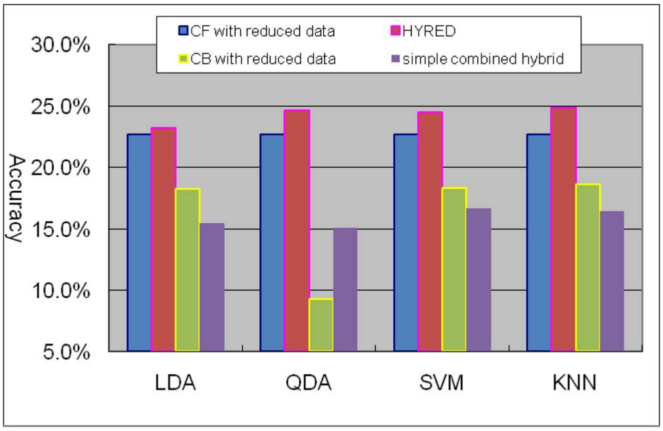


Fig. 6. Performance comparisons of the average accuracy.

TABLE III
COMPARISON OF AVERAGE COMPUTATIONAL TIME (UNIT IN SECONDS)

Data type	CF	DTB			
		K-NN	LDA	QDA	SVM
Full Dataset	38.95	0.25	0.234	0.047	24707.00
Reduced /Hybrid	1.766	0.031	0.016	0.031	303.797

computed average accuracy. We have presented the results of CF and DTBs with reduced data, HYRED, and a simple combined hybrid method in Fig. 6. It shows that the accuracies of both CF and HYRED were better than those of DTBs and simple hybrid method. The results of HYRED were slightly better than those of CF. We draw two conclusions from these results. First, CF with reduced data provides better performance than DTBs at these experiments. Second, HYRED achieve the better result than simple combined hybrid.

Remark 2: The results of DTB were not good in comparison with those of CF. This poor results were because the features describing items were restricted to the availability of real world commercial data. Because of customer privacy and company confidentiality policies, the company provided only transactional data without customer demographics and released only color and tone as features of each image. If we had used better feature data in describing the items, we would have obtained better results for both DTB and HYRED.

We have also compared the computational time of different algorithms. Using the reduced datasets decreased the number of items for analysis from 8776 to about 1000, which was the average number of items purchased by a target customer, 10–30 nearest neighbors, and 5–10 farthest neighbors. Table III shows the computational time of the full and reduced datasets. In cases in which CF and SVM used a reduced dataset, only 4.5% and 1.2%, respectively, of the full dataset was required.

These results show that the use of a reduced dataset based on the CF concept gave us better results from the viewpoints of both effectiveness and efficiency. The hybrid methods, HYRED and CF, achieved better results than CB. We conclude that HYRED is a viable solution to any problem that contains binary transactional data and item feature data.

V. CONCLUSION

A proper recommendation algorithm can be devised and applied to fit the type of information available in a specific domain. Despite the adaptation of a suitable algorithm to fit a recommendation problem, huge amounts of data can be a barrier in designing the efficient recommendation systems. Our research proposes a hybrid recommendation algorithm with reduced data as a way to deal with large-scale recommendation problems.

The experimental results showed that the accuracy of our proposed algorithms with reduced datasets was improved compared to existing algorithms with full datasets. In addition, our hybrid method using CF and DTB performed better than the pure DTB and simple combined hybrid methods. This study proposed the generalized rating system based on the DTB of a classifier. The recommendation system using the DTB of a KNN classifier performed better than that of a SVM classifier that has been widely used in CB settings. Finally, the best results in terms of accuracy were gained with HYRED's use of weighted coefficients between the CF and DTB scores. The HYRED is to apply to the general product recommendation problem that includes the information of transactional and item feature data. The results obtained are specific to the case analyzed because of lack of available real-world commercial data. Future work will apply the framework to other recommendation domains.

ACKNOWLEDGMENT

The authors would like to thank the reviewers for their constructive comments and helpful suggestions.

REFERENCES

[1] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 6, pp. 734–749, Jun. 2005.

[2] M. Balabanovic and M. Shoham, "FAB: Content-based collaborative recommendation," *Commun. ACM*, vol. 40, no. 3, pp. 66–72, 1997.

[3] C. Baroni-Urbani and M. W. Buser, "Similarity of binary data," *Syst. Zool.*, vol. 25, no. 3, pp. 251–259, 1976.

[4] S. Basu, M. Bilenko, and R. J. Mooney, "A probabilistic framework for semisupervised clustering," in *Proc. 10th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2004, pp. 59–68.

[5] R. Burke, "Hybrid recommender systems: Survey and experiments," *User Model. User-Adapted Interact.*, vol. 12, no. 4, pp. 331–370, 2002.

[6] W. Cheung, J. Kwok, M. Law, and K. Tsui, "Mining customer preference ratings for product recommendation using the support of vector machine and the latent class model," in *Proc. Int. Conf. Data Mining Methods Eng.*, 2000, pp. 601–610.

[7] Y. B. Cho, Y. H. Cho, and S. H. Kim, "Mining changes in customer buying behavior for collaborative recommendations," *Expert Syst. Appl.*, vol. 28, no. 2, pp. 359–369, 2005.

[8] M. Claypool, A. Gokhale, T. Miranda, P. Murnikov, D. Netes, and M. Sartin, "Combining content-based and collaborative filters in an online newspaper," in *Proc. ACM SIGIR Workshop Recommender Syst.*, 1999, pp. 1–11.

[9] A. Demiriz, "Enhancing product recommender systems on sparse binary data," *Data Mining Knowl. Discov.*, vol. 9, no. 2, pp. 147–170, 2004.

[10] D. Goldberg, D. Nichols, B. Oki, and D. Terry, "Using collaborative filtering to weave an information tapestry," *Commun. ACM*, vol. 35, no. 12, pp. 61–70, 1992.

[11] N. Good, J. Schafer, J. Konstan, A. Borchers, B. Sarwar, J. Herlocker, and J. Riedl, "Combining collaborative filtering with personal agents for better

recommendations," in *Proc. Conf. Amer. Assoc. Artif. Intell. (AAAI-1999)*, pp. 439–446.

[12] E. Han and G. Karypis, "Feature-based recommendation system," presented at the 14th ACM Int. Conf. Inf. Knowl. Manag., Bremen, Germany, Oct. 31–Nov. 5, 2005.

[13] T. Hastile, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. New York: Springer-Verlag, 2001.

[14] Z. Huang, H. Chen, and D. Zeng, "Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering," *ACM Trans. Inf. Syst.*, vol. 22, no. 1, pp. 116–142, 2004.

[15] C. Kim, J. Lee, Y. Cho, and D. Kim, "VISCORS: A visual-content recommender for the mobile web," *IEEE Intell. Syst.*, vol. 19, no. 6, pp. 32–39, Nov./Dec. 2004.

[16] D. Kim and B. Yum, "Collaborative filtering based on iterative principal component analysis," *Expert Syst. Appl.*, vol. 28, no. 4, pp. 823–830, 2005.

[17] Y. Kim, B. Yum, J. Song, and S. M. Kim, "Development of a recommender system based on navigational and behavioral patterns of customers in e-commerce sites," *Expert Syst. Appl.*, vol. 28, no. 2, pp. 381–393, 2005.

[18] J. Konstan, B. Miller, D. Maltz, J. Herlocker, L. Gordon, and J. Riedl, "GroupLens: Applying collaborative filtering to Usenet news," *Commun. ACM*, vol. 40, no. 3, pp. 77–87, 1997.

[19] G. Linden, B. Smith, and J. York, "Amazon.com recommendations: Item-to-item collaborative filtering," *IEEE Internet Comput.*, vol. 7, no. 1, p. 76-80, Jan./Feb. 2003.

[20] M. Pazzani, "A framework for collaborative, content-based, and demographic filtering," *Artif. Intell. Rev.*, vol. 13, p. 393-408, 1999.

[21] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Analysis of recommendation algorithms for e-commerce," in *Proc. ACM E-Commerce 2000 Conf*, pp. 158–167.

[22] J. Schafer, J. Konstan, and J. Riedl, "Recommender systems in e-commerce," in *Proc. ACM E-Commerce*, 1999, pp. 158–166.

[23] M. Schultz and T. Joachims, "Learning a distance metric from relative comparison," presented at the Neural Inf. Process. Syst., Whistler, B.C., 2003.

[24] U. Shardanand and P. Maes, "Social information filtering: Algorithms for automating 'word of mouth'," in *Proc. ACM CHI Conf. Human Factors Comput. Syst.*, 1995, pp. 210–217.

[25] H. Shin and S. Cho, "Response modeling with support vector machines," *Expert Syst. Appl.*, vol. 30, pp. 746–760, 2006.

[26] B. Smyth and P. Cotter, "A personalized television listings service," *Commun. ACM*, vol. 43, no. 8, pp. 107–111, 2000.

[27] S. Tong and E. Chang, "Support vector machine active learning for image retrieval," presented at the 9th ACM Int. Conf. Multimedia, Ottawa, Canada, 2001.

[28] V. Vapnik, *The Nature of Statistical Learning Theory*, 2nd ed. Berlin, Germany: Springer-Verlag, 1999.

[29] B. Xiao, E. Aimeur, and J. Fernandez, "PCFinder: An intelligent product recommendation agent for e-commerce," in *Proc. IEEE Int. Conf. E-Commerce*, 2003, pp. 181–188.

[30] J. Xu and K. Araki, "A SVM-based personal recommendation system for TV programs," presented at the 12th Int. Conf. Multimedia Model., Beijing, China, 2006.

Sang Hyun Choi received the B.S. degree in industrial engineering from Hanyang University, Seoul, Korea, in 1991, the M.S. degree in industrial engineering and the Ph.D. degree in management information systems from Korea Advanced Institute of Science and Technology, Daejeon, Korea, in 1993 and 1998, respectively.

From 1998 to 2002, he was a Senior Consultant at the Entrue Consulting, LG CNS. He is currently an Associate Professor in the Department of Industrial and Systems Engineering, Gyeongsang National University, Jinju, Korea. He has authored or coauthored more than 20 papers in international conference proceedings and journals. His research interests include recommendation systems, data mining, electronic commerce, and information systems planning.

Young-Seon Jeong received the B.S. degree in industrial engineering from Chonnam National University, Gwangju, Korea, in 1997, and the M.S. degree in industrial engineering from Korea University, Seoul, Korea, in 2001, respectively. He is currently working toward the Ph.D. degree in the Department of Industrial and Systems Engineering, Rutgers University, New Brunswick, NJ.

From 2001 to 2005, he was a System Engineer at SK C&C, Seoul, Korea. His current research interests include spatial modeling of wafer map data, wavelet application for functional data analysis, and statistical modeling for intelligent transportation system

Myong K. Jeong received the B.S. degree in industrial engineering from Hanyang University, Seoul, Korea, in 1991, the M.S. degree in industrial engineering from Korea Advanced Institute of Science and Technology, Daejeon, Korea, in 1993, the M.S. degree in statistics and the Ph.D. degree in industrial and systems engineering from the Georgia Institute of Technology, Atlanta, GA, in 2002 and 2004, respectively.

Before joining Rutgers University, he was an Assistant Professor at the University of Tennessee, Knoxville. From 1993 to 1999, he was a Senior Researcher at the Electronics and Telecommunications Research Institute, Daejeon. He is currently an Assistant Professor in the Department of Industrial and Systems Engineering and the Center for Operation Research, Rutgers University, New Brunswick, NJ. His research interests include statistical data mining, recommendation systems, machine health monitoring, and sensor data analysis.

Dr. Jeong is currently an Associate Editor of IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING. He was the recipient of the prestigious Freund International Scholarship and the National Science Foundation CAREER Award in 2002 and 2007, respectively.