Discovery-oriented Collaborative Filtering for Improving User Satisfaction

Yoshinori Hijikata
hijikata@sys.es.osaka-u.ac.jp

Takuya Shimizu
tshimizu@nishilab.sys.es.osaka-u.ac.jp

Shogo Nishida
nishida@sys.es.osaka-u.ac.jp

Graduate School of Engineering Science, Osaka University
1-3 Machikaneyama, Toyonaka
Osaka 560-8531, JAPAN

ABSTRACT
Many recommender systems employed in commercial web sites use collaborative filtering. The main goal of traditional collaborative filtering techniques is improvement of the accuracy of recommendation. Nevertheless, such techniques present the problem that they include many items that the user already knows. These recommendations appear to be good when we consider accuracy alone. On the other hand, when we consider users' satisfaction, they are not necessarily good because of the lack of discovery. In our work, we infer items that a user does not know by calculating the similarity of users or items based on information about what items users already know. We seek to recommend items that the user would probably like and does not know by combining the above method and the most popular method of collaborative filtering.

Author Keywords
collaborative filtering, discovery ratio, novelty, profile of acquaintance

ACM Classification Keywords
H.3.3 Information Search and Retrieval: Information filtering

INTRODUCTION
Although the internet has allowed us to publish information easily, people are faced with a problem called "information overload" by which they become unable to find suitable contents or products (after here "items"). A recommender system is one of the solutions for this problem. It finds suitable items for users based on their preference, experience, or demographic information [22]. Two approaches are useful for building recommender systems: collaborative filtering and content-based filtering [23]. Collaborative filtering requires no analyses of the contents of the items. For that reason, it has been used to build recommender systems in various domains [8, 12, 15, 21, 26].

Traditionally, researchers of collaborative filtering (CF) have insisted on improving the accuracy of recommendation using metrics such as precision/recall or mean absolute error (MAE). However, CF has a problem in which most of the items in the generated recommendation are items that the user already knows. Little diversity arises with respect to the contents (e.g., topic, genre, author) among the items in the generated recommendation. Therefore, users easily get tired of the recommendations; the possibility becomes high that they give up using the recommendation service. In other words, the problem occurs that users are not satisfied with the recommendations because of the lack of discovery or the lack of diversity [28].

This work is intended to improve the ability to discover items that are unknown to the user while retaining the ability to make an accurate recommendation for recommender systems. Particularly, we propose a discovery-oriented CF algorithm. The biggest difference between our algorithm and a pure CF algorithm is that our algorithm uses not only a profile of preference conservatively used by the pure CF algorithm but also a profile of which items users already know (profile of acquaintance). We collect profiles of acquaintance by letting users give ratings of acquaintance for items. Ratings of acquaintance are given on a two-point rating scale: "known item" (mapped to value 1) and "unknown item" (mapped to value 0). Although imposing additional ratings on users increases their burdens, we expect that they accept it if they are satisfied with the recommendations.

The contributions of our research are as follows:

- **Prediction of unknown items** We propose a method for predicting items unknown to a user. Using the ratings of acquaintance, it calculates the similarity between users or items, and calculates the probability that a user knows an unrated item. Hereinafter, we call this probability the "predicted value of acquaintance". It generates a list of items that seem to be unknown to the user. We measure the accuracy of the prediction of unknown items.
- **Recommendation of items from the user's preference**

and acquaintance We propose several algorithms, collectively designated as discovery-oriented CF algorithms, for recommending items that a user prefers and does not know. The first algorithm generates the integrated ratings by combining the ratings of preference and the ratings of acquaintance, then it applies the CF algorithm to the matrix of the integrated ratings. The second algorithm combines a predicted value of preference and that of acquaintance. The third algorithm identifies a set of items that seem to be unknown to the user and recommends items with a high predicted value of preference from the items in the set. We examine the effectiveness of these algorithms with respect to the novelty [11] when we introduce them to user-based CF [9, 21] and item-based CF [6, 25].

- **Examination of user satisfaction** We compare our discovery-oriented CF algorithm with a pure CF algorithm and Ziegler's topic-diversification algorithm [28] with respect to the users' satisfaction. We asked users to receive recommendations from these algorithms and rate the recommended items and item lists on satisfaction to the following usage objectives of a recommender system: (1) Purchase of items, (2) On-demand listening of items (Actually, items are music data), and (3) Discovery of new items.

The remainder of the paper is organized as follows. First, we explain the two prominent CF algorithms. We also explain popular non-accuracy metrics for evaluation that have been proposed, along with our proposed metric with respect to discovering items unknown to a user. In addition, we present our method for predicting unknown items and our discovery-oriented CF algorithms. Then, we present an empirical evaluation of the proposed algorithms using novelty. We also discuss the situation in which the user's ratings have a bias in the number of ratings of preference and that of acquaintance. Finally, we compare our algorithm and other algorithms from the view point of user satisfaction. After that, we describe some related works and offer some conclusions and some future directions of research.

COLLABORATIVE FILTERING

A basic idea of CF algorithm is recommending to the user items those items which a user group with similar preference likes. This section explains two basic algorithms; user-based CF algorithm and an item-based CF algorithm.

User-based CF Algorithm

We define a set of users as $A = \{a_1, a_2, \dots, a_n\}$, a set of items as $B = \{b_1, b_2, \dots, b_m\}$ and a rating of user a_i for item b_k as $r_i(b_k)$. The process of a user-based CF algorithm consists of two steps.

- **Neighborhood formation.** Assuming a_i as a target user, the similarity $s(a_i, a_o)$ for all $a_o \in A \setminus \{a_i\}$ is calculated based on the similarity between r_i and r_o. In general, Pearson correlation or cosine similarity is used for this calculation. The top-M most-similar users are selected as members of a_i's neighborhood $neighbor(a_i) \subseteq A$.

- **Rating prediction.** For all items b_k which are rated using a member of a_i's neighborhood $a_o \in neighbor(a_i)$ and which are not rated by a_i, a predicted value of preference $p_i(b_k)$ is calculated as follows.

$$p_i(b_k) = \bar{r_i} + \frac{\sum_{a_o \in A'_i} s(a_i, a_o) * (r_o(b_k) - \bar{r_o})}{\sum_{a_o \in A'_i} |s(a_i, a_o)|} \quad (1)$$

Note that $A'_i := \{a_o | a_o \in neighbor(a_i)\}$ and $\bar{r_i}$ is the average of the all users' ratings to item b_i.

Finally, a top-N recommendation list $L_{p_i} : \{1, 2, \dots, N\} \rightarrow B$ is generated based on predicted values p_i. A function L_{p_i} reflects the ranking in descending order by assigning a rank to an item with the highest predicted value at first.

Item-based CF Algorithm

The item-based CF algorithm calculates the similarity between items. These items can be considered as similar items when users give a similar rating for two items (b_k, b_e). In this case, we should set the similarity $s(b_k, b_e)$ high. Cosine similarity is commonly used for calculating $s(b_k, b_e)$. For each b_k, the top-M most similar items are selected as b_k's neighborhood $neighbor(b_k) \subseteq B$. The predicted value $p_i(b_k)$ is calculated as follows.

$$p_i(b_k) = \frac{\sum_{b_e \in B'_k} (s(b_k, b_e) \cdot r_i(b_e))}{\sum_{b_e \in B'_k} |s(b_k, b_e)|} \quad (2)$$

Note that $B'_k := \{b_e | b_e \in neighbor(b_k)\}$. The process for generating a top-N recommendation list L_{p_i} is the same as the user-based CF process.

EVALUATION METRICS

Many researchers have used accuracy metics for evaluating recommender systems. Pupular accuracy metics are mean absolute error (MAE), precision and recall. MAE measures how small the difference is between the predicted value and the real user rating on preference [5, 9]. Precision and recall judge how much the recommendation list includes the user's favorite items [24]. In detail, precision shows the ratio of the user's favorite items to all the recommended items. Recall shows how much of the user's favorite items in the test set are recommended. Recently, non-accuracy metrics are beginning to be proposed. In this section, we explain the four popular non-accuracy metrics (coverage, novelty, serendipity and intra-list similarity) (See [11] for the details) and our new non-accuracy metric (discovery ratio).

Coverage

Coverage measures the percentage of a dataset for which the recommender system can provide a prediction [9, 18]. Systems with higher coverage become more valuable to users because the users can find many good items if the systems can predict many of the items in the dataset.

Novelty and Serendipity

Novelty and serendipity measure the "non-obviousness" of the recommendations [11]. We can say that this recommendation is novel if a recommended item is unknown and favorite for a user. Assuming C_i as a set of items that are unknown and favored by user a_i in a test set, the precision of novelty and recall of novelty of the recommendation list L_i are represented as follows (A symbol $\Im L_i$ is the image of map L_i and presents all items in a recommendation list).

$$Precision(Novelty) = \frac{|C_i \cap \Im L_i^x|}{|\Im L_i|} \quad (3)$$

$$Recall(Novelty) = \frac{|C_i \cap \Im L_i|}{|C_i|} \quad (4)$$

A serendipitous recommendation helps a user find a surprisingly interesting item which he might not have discovered independently. For measuring serendipity, it is necessary to measure how the recommended items attract and surprise the user [11]. However, no concrete calculation method has been proposed so far because it is difficult to measure.

Intra-List Similarity

Intra-list similarity captures the diversity of the recommendation list [28]. It is calculated by summing up the topical similarities among items in the list. Topical similarity is calculated according to various features (e.g., genre, author and other discerning characteristics). Higher intra-list similarity denotes lower diversity.

Discovery Ratio

Discovery ratio measures how many unknown items are in the recommendation list. It differs from novelty in that it is independent of the user's preference. A higher discovery ratio denotes that a user does not know many items in the recommendation list. Assuming D_i as a set of user's unknown items in a test set, the discovery ratio is represented as follows.

$$discovery\ ratio = \frac{|D_i \cap \Im L_i|}{|\Im L_i|} \quad (5)$$

DISCOVERY-ORIENTED RECOMMENDATION

Our discovery-oriented CF algorithms use not only a profile of preference that is conservatively used by the pure CF algorithm but also a profile of acquaintance described in the Introduction. Using the two kinds of profile, they try to recommend a user's unknown and favorite items. In this section, we explain a method for predicting a user's unknown items and our five kinds of discovery-oriented CF algorithm.

Predicting Unknown Items

We use a profile of acquaintance to predict a user's unknown items. We hypothesize that a target user knows items that a user group with similar acquaintance knows. This hypothesis is the same as that of the CF algorithm for preference. Therefore, we try to predict a user's known items by applying the CF algorithm to profiles of acquaintance. According to the process of the CF algorithm, a user group with profiles of acquaintance similar to the target users' is identified. Then, items unknown to the group are considered as items that seem to be unknown to the target user.

We define user a_i's rating of acquaintance for item b_k as $h_i(b_k)$. By applying user-based CF or item-based CF algorithm to the ratings of acquaintance, a predicted value of acquaintance $p_i^{know}(b_k)$ is calculated as follows respectively.

$$p_i^{know}(b_k) = \bar{h_i} + \frac{\sum_{a_o \in A_i'} s(a_i, a_o) * \left(h_o(b_k) - \bar{h_o}\right)}{\sum_{a_o \in A_i'} |s(a_i, a_o)|} \quad (6)$$

$$p_i^{know}(b_k) = \frac{\sum_{b_e \in B_k'} \left(s(b_k, b_e) \cdot h_i(b_e)\right)}{\sum_{b_e \in B_k'} |s(b_k, b_e)|} \quad (7)$$

The predicted value $p_i^{know}(b_k)$ can be regarded as the probability that user a_i knows item b_k. An ordered list $L_{p_i^{know}}$ is generated based on $p_i^{know}(b_k)$. An item with higher $p_i^{know}(b_k)$ is ranked higher in $L_{p_i^{know}}$. By sorting items in $L_{p_i^{know}}$ in reverse order, we obtain a list $L_{p_i^{unknow}}$ in which an item that is not known with higher probability is ranked higher.

Discovery-oriented CF Algorithms

The followings are the detail explanation of our five kinds of discovery-oriented CF algorithm.

Rating-Integrating Algorithm: RIA

In Rating-Integrating Algorithm (RIA), integrated ratings are first generated by combining ratings of preference and ratings of acquaintance. The integrated rating becomes higher when the user rates the item in high grade with respect to the preference and also rates it as an "unknown item". A recommendation list is generated by applying the CF algorithm to the integrated ratings.

We add the ratings of preference and the ratings of acquaintance with a weight to generate integrated ratings. Weight $\alpha \in [0, 1]$ is defined as the impact that ratings of acquaintance exert on integrated ratings, so $(1 - \alpha)$ is the impact that ratings of preference exert on integrated ratings. Note that the ratings of preference should be translated into 0-1 scale. $r_i'(b_k)$ denotes the translated rating of preference. Integrated rating $r_i^{uni}(b_k)$ is calculated as follows.

$$r_i^{uni}(b_k) = (1 - \alpha) \times r_i'(b_k) + \alpha \times (1 - h_i(b_k)) \quad (8)$$

A recommendation list is generated by applying the CF algorithm to the matrix of the integrated ratings $r_i^{uni}(b_k)$.

Prediction-Combining Algorithm: PCA

A predicted value of preference and that of acquaintance can be calculated separately by applying the CF algorithm to each kind of profile. We propose some algorithms which combine these two kinds of predicted value.

Prediction-Combining Algorithm (Rank Addition): PCA (RA)

Prediction-Combining Algorithm (Rank Addition) (PCA(RA)) combines list L_{p_i} generated by applying the CF algorithm to the profile of preference and list $L_{p_i^{unknow}}$ generated by applying the CF algorithm to the profile of acquaintance. For each item, we add its rank on list L_{p_i} and that on list $L_{p_i^{unknow}}$. A recommendation list $L_{p_i^*}$ is generated based on this value.

procedure PCA(RA) $(L_{p_i},\ \alpha)$ {
$B_i \leftarrow \Im L_{p_i}$;
$b \in B_i$:compute $p_i^{know}(b)$;
compute $L_{p_i^{know}} : \{1, 2, \ldots, |B_i|\} \rightarrow B_i$ using p_i^{know};
for all $b \in B_i$ **do**
$L^{-1}_{p_i^{unknow}}(b) \leftarrow |B_i| - L^{-1}_{p_i^{know}}(b)$;
$p_i^*(b) \leftarrow L^{-1}_{p_i}(b) \cdot (1-\alpha) + L^{-1}_{p_i^{unknow}}(b) \cdot \alpha$;
end do
compute $L_{p_i^*} : \{1, 2, \ldots, |B_i|\} \rightarrow B_i$ using p_i^*;
return $L_{p_{i*}}$;
}

Figure 1. Prediction-Combining Algorithm (Rank Addition)

procedure PCA: $(L_{p_i},\ \alpha)$ {
$B_i \leftarrow \Im L_{p_i}$;
$b \in B_i$:compute $p_i^{know}(b)$;
for all $b \in B_i$ **do**
$p_i^*(b) \leftarrow (1-\alpha) * p_i\,(b) + \alpha * \left(1 - p_i^{know}\,(b)\right); //for\ (VA)$
$\left(p_i^*(b) \leftarrow p_i\,(b) * \left(1 - p_i^{know}\,(b)\right);\right) //for\ (VM)$
end do
compute $L_{p_i^*} : \{1, 2, \ldots, |B_i|\} \rightarrow B_i$ using p_i^*;
return $L_{p_{i*}}$;
}

Figure 2. Prediction-Combining Algorithm (Value Addition, Value Multiplication)

The details of PCA(RA) are presented in Fig. 1. First, for each item $b \in B_i$ in the list L_{p_i}, the predicted value of acquaintance $p_i^{know}\,(b)$ is calculated by applying the CF algorithm to the profiles of acquaintance. List $L_{p_i^{know}}$ is generated from the predicted values $p_i^{know}\,(b)$. By sorting items in $L_{p_i^{know}}$ in reverse order, we obtain a list $L_{p_i^{unknow}}$ with items, which the user does not know with higher probability, are ranked higher. Next, for each item $b \in B_i$, $p_i^*(b)$ is calculated by adding the rank on list $L_{p_i^i}$ and that on list $L_{p_{unknow}}$ with weight α. Weight $\alpha \in [0, 1]$ shows the impact that each kind of list exerts on the combined list $L_{p_i^*}$. The new recommendation list $L_{p_i^*}$ is generated by sorting items by $p_i^*(b)$ in ascending order.

Prediction-Combining Algorithm (Value Addition): PCA(VA)

Prediction-Combining Algorithm (Value Addition) (PCA (VA)) adds the predicted value of preference $p_i\,(b)$ and that of acquaintance $p_i^{know}\,(b)$. Note that the ratings of preference should be translated into 0-1 scale. For each item $b \in B_i$, score $p_i^*(b)$ is calculated as follows (see also Fig. 2).

$$p_i^*(b) = (1-\alpha) * p_i\,(b) + \alpha * \left(1 - p_i^{know}\,(b)\right) \tag{9}$$

List $L_{p_i^*}$, which is shown to the user, is generated by sorting items by $p_i^*(b)$ in descending order.

Prediction-Combining Algorithm (Value Multiplication): PCA (VM)

We can regard the predicted value of acquaintance $p_i^{know}\,(b)$ as a probability which user a_i knows item b, then we can regard $1 - p_i^{know}\,(b)$ as a probability which user a_i does not know item b. To recommend favorite and unknown items with higher probability, Prediction-Combining Algorithm (Value Multiplication) (PCA (VM)) calculates score $p_i^*(b)$ as follows (see also Fig. 2):

$$p_i^*(b) = p_i\,(b) * \left(1 - p_i^{know}\,(b)\right) \tag{10}$$

List $L_{p_i^*}$ shown to the user is generated by sorting items by $p_i^*(b)$ in descending order. It is necessary to set an appropriate α for PCA(RA) and PCA (VA), whereas PCA (VM) presents the advantage that there is no need to do so.

procedure IEA $(B_i,\ \alpha)$ {
$b \in B_i$:compute $p_i^{know}(b)$;
set $B_i' \leftarrow \left\{B_i \mid p_i^{know}(b) \leq \alpha\right\}$;
for all $b \in B_i'$ **do**
compute $p_i(b)$;
end do
compute $L_{p_i^*} : \left\{1, 2, \ldots, |B_i'|\right\} \rightarrow B_i'$ using p_i;
return $L_{p_i^*}$;
}

Figure 3. Independently Evaluating Algorithm

Independently Evaluating Algorithm: IEA

Generally, it is important to know how much a user likes a specific item. Different degrees of preference are applicable, even among favorite items. The system should recommend items that the user likes more at the higher rank. However, the degree of acquaintance is probably less important for the user than that of preference because, once the user sees an item and favors it, she would purchase or remember it. We think that calculating the strict degree of acquaintance is not so important to users.

Independently Evaluating Algorithm (IEA) considers the above difference of the property among preference and qcquaintance. It identifies the set of items that seem to be unknown to a user and ranks the items in the set based on the predicted values of preference. Details of this algorithm are shown in Fig. 3. First, for each item $b \in B_i$, the predicted values of acquaintance $p_i^{know}\,(b)$ are calculated by applying the CF algorithm to the profiles of acquaintance. Items with $p_i^{know}\,(b)$ lower than threshold $\alpha \in [0, 1]$ are selected as a set of items (denoted as B_i') that seem to be unknown to the user. Next, for each item $b \in B_i'$, the predicted value of preference $p_i\,(b)$ is calculated. List $L_{p_i^*}$, which is shown to the user, is generated by sorting items by $p_i\,(b)$ in descending order.

EXPERIMENT ON NOVELTY

We conducted an evaluation to verify that our algorithms are effective for acquiring favorite and unknown items. Basically, our algorithms use CF algorithm for predicting unknown items. Firstly, we evaluated the ability to predict unknown items. Next, we see whether our algorithm can make the novelty of recommendation higher than the original CF algorithm. We also see the relations between novelty, precision of preference, and the discovery ratio by changing the parameter α. Finally, we compare the performances of all algorithms.

Table 1. Rating distribution

	known	unknown	sum
favorite	72.7	21.9	94.6
unfavorite	36.3	69.1	105.4
sum	109.0	91.0	200

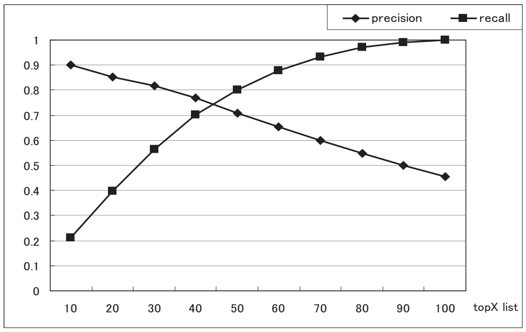

Figure 4. Prediction of unknown items

Data Set

We collected 20000 rating data for the experiment. Specifically, we had 100 users rate 200 music data selected randomly from a music database composed of 1000 music data. In this subsection, we explain our methods for building the music database and collecting the rating data.

We built our original music database for the experiment. The music database includes 1000 music data. The music database comprises the music title, singer's name, release year, music category, and URL for previewing the music data. We constructed this database by getting a license for commercial music from JASRAC [13] and getting a license for commercial music data from a major commercial music site to use in the experiment. Categories of 1000 music data are Japanese pop music (J-Pop; 700 music data), Japanese traditional-style music (enka; 75 music data), Japanese animation music (anime; 75 music data), and foreign music (150 music data). A broad range of music data from the 1960s until now was selected for J-Pop and foreign music.

We collected rating data from 100 people (younger than teenagers, 13; twenties, 51; thirties, 5; forties, 18; fifty and older, 13 people). Then 200 music data selected randomly from the music database are presented to each user; the profile of preference and that of acquaintance are collected by asking users to rate the music data. Ratings of preference are given in five scales (1–5) and ratings of acquaintance are recorded as binary (1 or 0). For music which is unknown to the user, the rating of preference is assigned by the user after previewing it. The total time taken to complete all the ratings, including the previewing time, is 3 h on average per user. To calculate metrics, the ratings of preference were binarized like ratings ranged from four to five into "favorite" and ratings ranged from one to three to "unfavorite". The average distribution of the ratings of preference and those of acquaintance given by 100 users is shown in Table 1. We divided the 200 music data rated by each user into a training set composed of 100 music data and a test set composed of 100 music data.

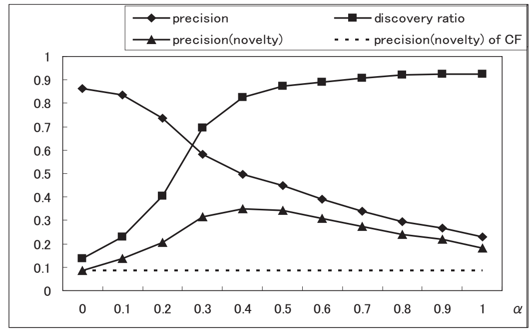

Figure 5. Results for precision, the discovery ratio, and precision of novelty for the Rating-Integrating Algorithm

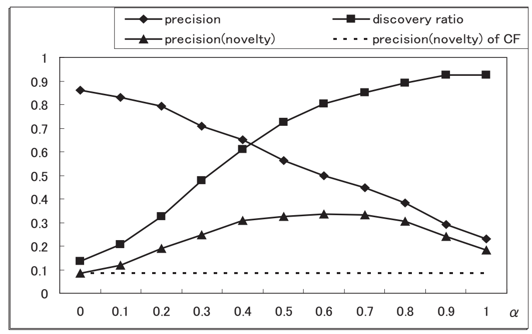

Figure 6. Results for precision, the discovery ratio, and precision of novelty for PCA (RA)

Prediction of Unknown Items

We examined the performance of our method in predicting items that are unknown to a user. We calculated the predicted values of acquaintance for 100 music data in the test set and generated a list in which an item with a lower predicted value of acquaintance is ranked higher. User-based CF was applied to the ratings of acquaintance to calculate the predicted values of acquaintance. We analyzed the precision and recall of the predictions of acquaintance for the generated top-X $(X \in [10, 20, \ldots, 90, 100])$ list. The result is shown in Fig. 4. The precision for the top-10 list is 0.9. It is apparent that the method can predict items that are unknown to the user with high precision. In addition, a curve of the recall is convex upward, thereby showing that the method can predict items that are unknown to the user. In fact, 80% of the unknown items in the test set are included in the top-50 list.

We also used item-based CF in the experiment. However, the result was worse than that of the case of using user-based CF. In our collection of rating data, the number of users who rated the same pair of items, which is necessary to calculate the similarity between these items, is smaller than the number of items which are commonly rated by two users, which is necessary for calculation of the similarity between these users. Consequently, the reliability for the calculated similarity between the items is lower than that between the users. We used user-based CF in the following experiments described in subsequent sections.

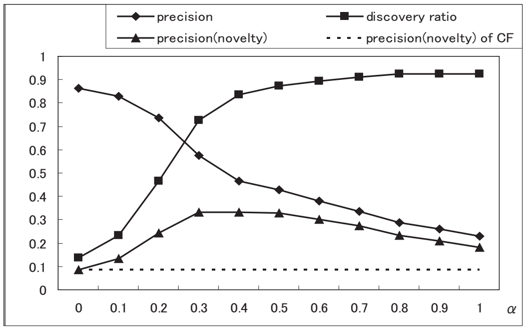

Figure 7. Results for precision, the discovery ratio, and precision of novelty for PCA (VA)

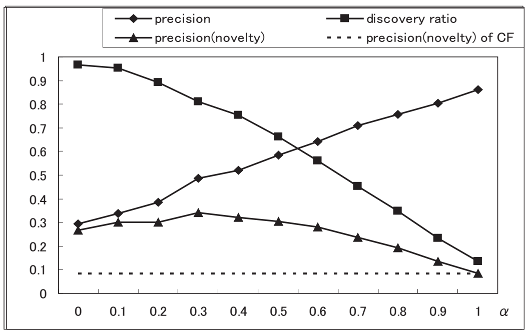

Figure 8. Results for precision, the discovery ratio, and precision of novelty for the Independently Evaluating Algorithm

Precision of Preference, Discovery Ratio, and Novelty

A top-5 recommendation list is generated for every discovery-oriented CF algorithm. We calculated the precision of preference, the discovery ratio, and the precision of novelty for each list. We changed $\alpha \in [0, 0.1, \ldots, 0.9, 1.0]$ to determine its optimal value. We compared the performance among our algorithms and the original user-based CF (CF) algorithm.

Fig. 5, 6, 7 and 8 shows the precision of preference, the discovery ratio, and the precision of novelty of RIA, PCA(RA), PCA(VA), IEA by changing α respectively. In RIA, PCA(RA) and PCA(VA), as α becomes larger, the precision decreases and the discovery ratio increases. The precision of novelty becomes the highest at α=0.4 for RIA, α=0.6 for PCA(RA) and α=0.3 for PCA(VA). In IEA, as α becomes larger, the precision inreases and the discovery ratio decreases. The precision of novelty becomes the highest at α =0.3.

The results of CF and PCA(VM) including those of RIA, PCA(RA), PCA(VA) and IEA (at α for realizing the best novelty) is shown in Table 2. Apparently, the precisions of novelty of our discovery-oriented CF algorithms are higher than that of CF. However, no apparent difference exists among our proposed algorithms. For that reason, we can not judge which algorithm is the best for improving the novelty.

BIAS IN THE NUMBER OF RATINGS

In the experiment described in the previous section, we assumed that a user always gives an item both a rating of preference and that of acquaintance. However, in real usage in commercial recommendation sites, it is conceivable that a user does not give a rating of preference to an item that is unknown to him. He must preview it when a user tries to give a rating of preference to an unknown item. Music data do not always have sample data for preview. Furthermore, a task of previewing music data consumes her time and energy. Consequently, music data rated as "unknown item" are not always given a rating of preference. Therefore, our algorithms must be useful in situations where a user's ratings have a bias in the number of ratings of preference and that of acquaintance.

Table 2. A comparison of precision, the discovery ratio, and precision of novelty among algorithms

	precision	discovery ratio	precision(novelty)
RIA	0.50	0.83	0.35
PCA(RA)	0.50	0.80	0.34
PCA(VA)	0.58	0.73	0.33
PCA(VM)	0.42	0.87	0.33
IEA	0.49	0.81	0.34
CF	0.86	0.13	0.085

Experimental methodology

We modified our proposed algorithms as follows to deal with the bias in the number of ratings. RIA requires that an item has both a rating of preference and that of acquaintance to generate an integrated rating. We ignore a rating of acquaintance for an item which is rated as "unknown item" and is not given a rating of preference. PCA must calculate a predicted value of preference and that of acquaintance. The predicted value of preference is calculated in the same manner as for an item in which both kinds of rating are missing. The predicted value of acquaintance is $p_i^{know}(b) = 0$ and the rank on the list $L_{p_i^{unknow}}$ is 1. IEA must identify a set of items which seem to be unknown to a user. An item which is rated as "unknown item" and which is not given a rating of preference is to be included in the set described above.

We conducted an experiment to see the performances of our discovery-oriented CF algorithms in a situation where a user's ratings have a bias in the number of ratings of preference and that of acquaintance. In this experiment, we examined the difference of novelty among our algorithms. We also examined the percentage of music data with a rating of "unknown item" in the recommendation list. We think music data which are rated as "unknown item" and which are not given a rating of preference should be added to a nomination list for recommendations (set of music data which might be worth recommending to the user) because the user does not know its content. Therefore, the nomination list includes music data for which the user does not give both a rating of preference and that of acquaintance and music data which he gives a rating of "unknown item" and does not give a rating of preference. We used the same dataset in the previous section. We generated a bias by partly deleting ratings of preference for music data with a rating of "unknown item". Specifically, we generated a new training set in which $X\%$ ($X \in [25, 50, 75]$) of music data with a rating of "unknown item" have the ratings of preference.

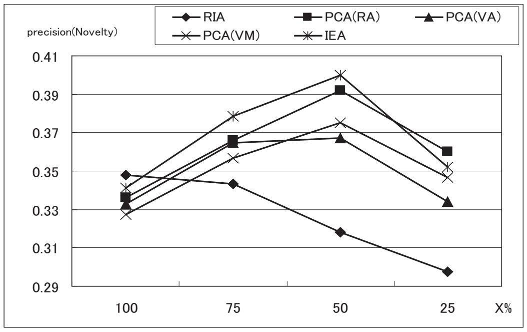

Figure 9. Results for precision of novelty

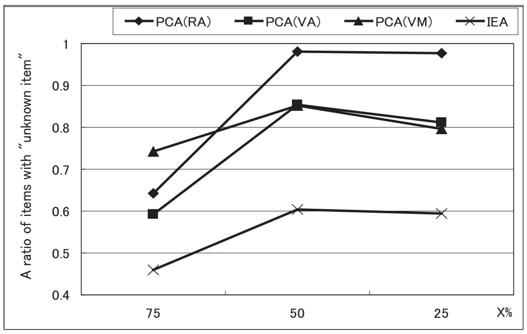

Figure 10. Ratio of items which have a rating of "unknown item" in the recommendation list

Result

A top-5 recommendation list was generated for every algorithm from among 100 music data in the test set and music data which are rated as "unknown items" and which are not given a rating of preference in the training set. We calculated the precision of novelty for each list. Fig. 9 shows the result for each of our algorithms in the case of changing X. The results of RIA, PCA(RA), PCA(VA), and IEA are the highest values among the results with different α. Comparing our algorithms, the precision of novelty for RIA is lower than that for the other algorithms. This is attributed to the fact that only RIA uses no ratings of acquaintance for music data which are rated as "unknown items" and which are not given a rating of preference. Therefore, we conclude that RIA is not appropriate for situations with a bias in the number of ratings of preference and that of acquaintance.

The other algorithms show little difference in the precision of novelty. Therefore, we examined the percentage of music data with a rating of "unknown item" in the recommendation list. Fig. 10 shows the results obtained when changing X. When X=25, 50, for PCA(VA) and PCA(VM), the percentage of music data with a rating of "unknown item" in the recommendation list is about 80%. For PCA(RA), the percentage is nearly 100%. On the other hand, for IEA, the percentage is about 60%, which shows that the recommendation lists generated by PCA(RA), PCA(VA), and PCA(VM) include many music data which have already been rated as "unknown". Actually, PCA treats the predicted value of acquaintance for these music data as 0 and the rank on the list of unknown items for them as 1. Therefore, PCA tends to rank these music data high on the recommendation list.

Discussion

In a commercial web site, a user may not try to preview the music data even though he is allowed to preview it. When a user conducts initial ratings before the first recommendation, he sees basic information about the presented music data (e.g., artist name, release date, content description, and customer review). It seems more likely that he does not get interested in the music data if he makes the decision not to preview the music data after seeing the information. Consequently, if he is recommended the above music data, there is less probability that he will like it. When sample data for preview are not provided, the user does not give rating of preference even if he becomes interested in the music data after seeing the basic information. When it is recommende by the system, the probability he will like it is the same as the probability he will like music data without both kinds of ratings.

Even if we consider the second case, the probability of liking music data with only a rating of "unknown item" is lower than the probability of liking music data without both kinds of ratings. Furthermore, a user would easily discover the tendency of the recommendation when a recommendation list includes many music data with a rating of "unknown item". Then, the user can guess which music data are recommended next. Accordingly, the unpredictability of the recommendation becomes low and the possibility exists that the user's satisfaction for the recommendation becomes low even though the novelty of the recommendation is high. In this respect, recommending many music data with a rating of "unknown item" is a problem. For these reasons, in the situation in which the user's ratings have a bias in the number of ratings of preference and that of acquaintance, IEA is better than PCA.

EXPERIMENT ON USER SATISFACTION

Our off-line experiment has shown that our algorithm improves the precision of novelty. However, it is not apparent that our algorithm achieves high user satisfaction because it puts an extra burden, which is a rating operation with respect to the acquaintance, on users. In this section, we compared our algorithm on user satisfaction with pure CF algorithm and Ziegler's Topic Diversification Algorithm (TDA), which obtains higher user satisfaction than that of CF algorithm (The outline of this algorithm is presented in "Related Work" Section).

Experimental Methodology

We selected IEA (α=0.3, 0.6) among five kinds of our poposed algorithm. IEA achieves the highest precision of novelty when α=0.3. It achieves good balance among the precision of preference and that of novelty when α=0.6. Fig. 11 shows the categories used for TDA. We created these categories for our music database. Table 3 shows the result of off-line experiment using the same dataset in "EXPERIMENT ON NOVELTY" Section). The novelty of TDA is worse than IEA although it is better than that of CF.

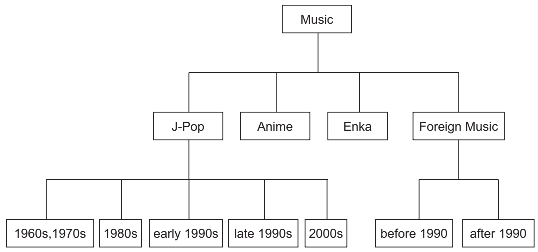

Figure 11. Categories used for TDA

Table 3. Performance of four algorithms in the offline analysis

	precision (preference)	discovery ratio	precision (novelty)
IEA($\alpha = 0.3$)	0.49	0.81	0.34
IEA($\alpha = 0.6$)	0.64	0.56	0.28
CF	0.86	0.14	0.085
TDA	0.84	0.17	0.094

40 graduate and undergraduate students participated in this experiment. They gave a rating of preference and that of acquaintance to 100 music data randomly selected from our music database. They received a recommendation list generated from the rest of the 900 music data by one of the four algorithms (IEA(α=0.3, 0.6), CF and TDA). We used the same rating data of other users in "EXPERIMENT ON NOVELTY" Section.

We think that it is difficult for users to give the degree of their satisfaction if there is no objective to receive a recommendation. We defined the three types of objective to give the degree of satisfaction.

- **Satisfaction for purchase** Purchase of items is one of the objectives in commercial web sites. This type of satisfaction is important for commercial web sites. Users are also happy to meet an item which they want to buy.

- **Satisfaction for on-demand listening** Users want to listen to their favorite music when they use a service like internet radio which is a free music program with advertisements. This type of satisfaction is important for both of sponsors and users.

- **Satisfaction for discovery** Many users in commercial web sites consult a recommendation to find new items or to broaden their knowledge of unfamiliar area. If they get interested in some items or areas, they search information about them. They might want to buy those items while searching the information. This type of satisfaction is important for both of commercial web sites and users.

A top-5 recommendation list is shown to the user. The users gave a rating of preference and that of acquaintance to a recommended item. We asked the users to suppose each of the above objectives to receive recommendations. They gave the degree of satisfaction in 5 scales (1-5) to the recommendation list and each of the recommended items for every objective. We offered the recommendation list five times. We evaluated the user satisfaction from the following two kinds of metrics: (i) the degree of satisfaction to the recommendation list (After here "DSL") and (ii) the ratio of satisfactory items in the recommendation list (After here "RSI"). To calculate the second metric, the degrees of satisfaction were binarized like degrees ranged from four to five into "satisfactory" and degrees ranged from one to three to "unsatisfactory".

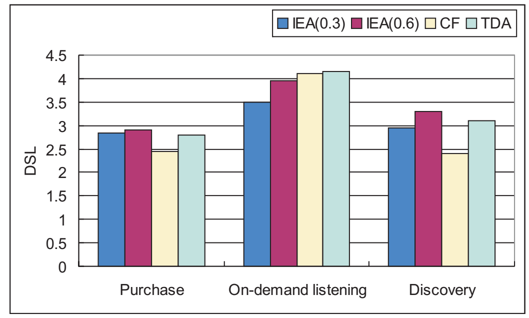

Figure 12. Degree of satisfaction to the recommendation list (DSL)

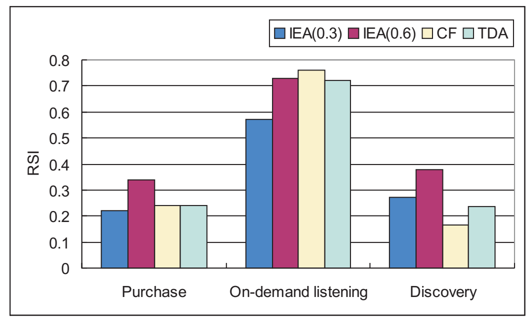

Figure 13. The ratio of satisfactory items in the recommendation list (RSI)

Result

Fig. 12 shows the result on DSL. For the satisfaction for purchase, DSL of IEA(α=0.3, 0.6) is better than that of CF. But, it is almost the same as that of TDA. For the satisfaction for on-demand listening, DSL of IEA(α=0.3) is worse than those of other algorithms. We think this is due to its low precision of preference. For listening to music on the fly, the precision of preference affects the user's satisfaction. For the satisfaction of discovery, DSL of IEA(α=0.3, 0.6) is better than that of CF. However we cannot find a apparent difference among IEA(α=0.3, 0.6) and TDA.

Fig. 13 shows the result on RSI. For the satisfaction for on-demand listening, RSI of IEA(α=0.3) is worse than those of other algorithms. This is as same as the result on DSL. For the satisfaction for purchase and that for discovery, DSL of IEA(α=0.6) achieves the best result. It is apparent that the improvement from the results of other algorithms for RSI is bigger than that for DSL.

From the above results, for the objective of purchasing items and finding new items, IEA(α=0.6) recommends more items

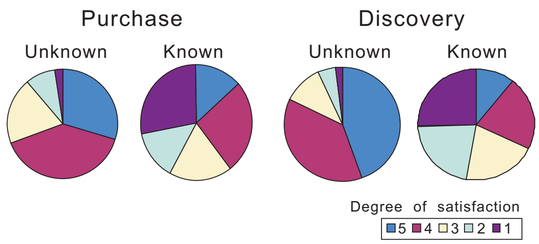

Figure 14. The ratio of the degrees of satisfaction for the music data whose ratings of preference are 5

that satisfies the user than other algorithms. Unfortunately we cannot show that our algorithm provides a recommendation list that satisfies a user. However, we think that users will be happy when they meet many satisfactory items. It is also good for commercial web sites that the users meet items which may lead to the future purchase. There is a possibility that our discovery-oriented CF algorithm can keep users to use a recommender system without their boredom and improve the sales in commercial web sites.

Relationship between Satisfaction and Ratings

We found that our algorithm recommends more satisfactory items than other algorithms in the previous subsection. Here a question occurs that which kind of ratings among ratings of preference and those of acquaintance affect more a user's satisfaction. We calculated the correlation coefficient among degrees of satisfaction and each kind of ratings. The correlation coefficients are 0.42, 0.72 and 0.38 for purchase, on-demand listening and discovery respectively in the case of ratings of preference. Those in the case of ratings of acquaintance are -0.09, 0.15, -0.21. It seems that ratings of acquaintance have no influence (or bad influence) on the user's satisfaction.

This result contradicts the results in the previous subsection epsecially for the purpose of purchase and discovery. We focused on only the music data with high ratings of preference and examined whether the degrees of satisfaction differ among ratings of acquaintance. Fig. 14 shows the distribution of the degrees of satisfaction for the music data whose ratings of preference are 5 for the purpose of purchase and discovery. The ratios of satisfactory items are about 70% and 80% in the items that are unknown to the users for the purpose of purchase and discovery respectively. However, they are about 40% and 30% in the items that are known to the users. From the above results, preference is the most fundamental factor for the influence to the users' satisfactions. However, acquaintance also affects the users' satisfactions especially for the items which will be preferred by them.

RELATED WORK

Research for improving users' satisfaction with recommendations are related to our work. Three approaches are used for improving the users' satisfaction: an approach particularly addressing recommendation algorithms, an approach particularly addressing an explanation in recommender systems and an approach particularly addressing interactions with users.

Some studies have emphasized a recommendation algorithm for improving user satisfaction [1, 2, 14, 28]. Ziegler et al. [28] addressed the problem of insufficient diversity with respect to the contents among items in a recommendation list. They increased the diversity of a recommendation list by combining a list generated using a CF algorithm and a list generated based on topic similarity among items. Kato et al. [14] proposed a content-based method keeping the balance between accurate recommendations and unexpected recommendations. The method identifies the genre the user might prefer. Then, it mixes promising items selected from the favorite genre and those selected from another genre. Ardissono et al. [2] developed a system that recommends sightseeing destinations for heterogeneous tourist groups. It reduces the conflict related to the requirements among the members. Adomavicius et al. [1, 19] incorporates contextual information into a recommendation process. Recommendations fitting the user's short-term preference which is affected by the situation might improve the user's satisfaction.

Good explanations in recommender systems were able to increase users' satisfaction [3, 4, 10, 17, 27]. Herlocker et al. [10] showed that providing explanations can improve the users' acceptance of CF systems. Bilgic et al. [3] examined several explanations by asking users to rate items when only seeing the explanations and when reviewing the real items. Bonard et al. [4] proposed that the usefulness of recommender systems can be improved by including more information about users who give recommendations. McNee et al. [17] displayed the degree of confidence about the prediction to a recommended item. Users become able to select items based on their tolerance for risk.

Some studies have specifically addressed user interaction in recommender systems [7, 16, 20]. Otsubo [20] developed a recommender system which shows many possible items to users rather than making a deep inference about the users' preferences. He reduces the cost of information acquisition by simplifying the operation. Fujimori et al. [7] proposed a system for combining a function for a user's active search with a system's passive recommendation. It enables users to obtain a recommendation that differs slightly from their usual preferences by selecting a user group. McCarthy et al. [16] proposed an interface for feedback in which the user can easily give a modification requirement to the recommendation. Otsubo, Fujimori and McCarthy show that an interaction with the recommender system is an important factor for users' satisfaction.

CONCLUSION

We proposed five recommendation algorithms to improve the novelty of a recommendation. Our algorithms predict items that are unknown to a user using information about items that users already know (a rating of acquaintance). By combining this method and the traditional CF algorithm, our algorithms recommend a user's unknown and favorite items. We conducted an experiment using 20000 rating data collected from 100 users. The result showed that our algorithms are effective for improving the novelty of the recommendations. Considering the usage in real commercial recommendation sites, we also conducted an experiment in a situation in which some items have no rating of preference and with a rating of "unknown item". We examined the novelty of the recommendation and the percentage of the items which are already rated as "unknown items" in the recommendation list. The result showed that the Independently Evaluating Algorithm (IEA) is the best of the five algorithms because it does not include many items which already have a rating of "unknown item". Finally, we conducted an experiment on user satisfaction. For the objective of purchasing items and finding new items, IEA achieves a better result than pure CF and Topic Diversification Algorithm in including many satisfactory items in the recommendation list. In future studies, we will examine explanation functions that not only consider the user's preference and the user's acquaintance.

REFERENCES

1. Adomavicius, G., et al.: 2005, 'Incorporating contextual information in recommender systems using a multidimensional approach'. *ACM Transactions on Information Systems* 23(1), 103-145.

2. Ardissono, L., et al.: 2001, 'Tailoring the recommendation of tourist information to heterogeneous user groups'. *Proc. of AH2001*, LNCS 2266, pp. 280–295.

3. Bilgic, M. and R.J. Mooney: 2005, 'Explaining recommendations: satisfaction vs. promotion'. *In Beyond Personalization Workshop, IUI'05*.

4. Bonhard, P., et al.: 2006, 'Accounting for taste: using profile similarity to improve recommender systems'. *Proc. of CHI'06*, pp. 1057-1066.

5. Breese, J., D. Heckerman, and C. Kadie: 1998, 'Empirical analysis of predictive algorithms for collaborative filtering'. *Proc. of UAI'98*, pp. 43–52.

6. Deshpande, M. and G. Karypis: 2004, 'Item-based top-n recommendation algorithms'. *ACM Transactions on Information Systems* 22(1), 143–177.

7. Fujimori, H., et al.: 2004, 'Visualizing neighborhood for collaborative filtering'. *Proc. of WISS 2004*, pp. 25–30.

8. Goldberg, D., et al.: 1992, 'Using collaborative filtering to weave an information tapestry'. *Comm. of the ACM* 35(12), 61–70.

9. Herlocker, J., et. al: 1999, 'An algorithmic framework for performing collaborative filtering'. *Proc. of SIGIR'99*, pp. 230–237.

10. Herlocker, J.L., Konstan, J.A. and Riedl, J.: 2000, 'Explaining collaborative filtering recommendations'. *Proc. of CSCW2000*, pp. 241–250.

11. Herlocker, J., et al.: 2004, 'Evaluating collaborative filtering recommender systems'. *ACM Transactions on Information Systems* 22(1), 5–53.

12. Hill, W., et al.: 1995, 'Recommending and evaluating choices in a virtual community of use'. *Proc. of CHI'95*, pp. 194-201.

13. Japanese Society for Rights of Authors, Composers and Publishers (JASRAC). http://www.jasrac.or.jp/ .

14. Kato, Y., Kawaguchi, K. and Hakozaki, K.: 2005, 'A recommender system indicating accurate and subconscious items for on-line shopping'. *Journal of IPSJ* 46 (No. SIG13(TOD 27)), 53–64.

15. Linden, G., Smith, B., and York, J.: 2003, 'Amazon.com recommendations: item-to-item collaborative filtering'. *IEEE Internet Computing* 4(1), 76–80.

16. McCarthy, K., et al.: 2005, 'Experiments in dynamic critiquing'. *Proc. of IUI'05*, pp. 175–182.

17. McNee, S.M., et al.: 2003, 'Confidence displays and training in recommender systems'. *Proc. of INTERACT '03*, pp. 176–183.

18. Middleton, S., Shadbolt, N. and De Roure, D.: 2004, 'Ontological user profiling in recommender systems'. *ACM Transactions on Infomation Systems* 22(1), 54–88.

19. Ono, C., et. al: 2007, 'A Context-Aware Movie Preference Model Using a Bayesian Network for Recommendation and Promotion'. *Proc. of UM 2007*, LNCS Vol. 4511, pp. 257–266.

20. Otsubo, G.: 2005, 'Gards - Information recommendation system which assumes ever changing human preference'. *Proc. of WISS 2005*, pp. 25–30.

21. Resnick, P., et al.: 1994, 'GroupLens: An open architecture for collaborative filtering of netnews'. *Proc. of CSCW'94*, pp. 175–186.

22. Resnick, P. and Varian, H.: 1997, 'Recommender systems'. *Comm. of the ACM* 40(3), 56–58.

23. Riecken, D.: 2000, 'Personalized views of personalization'. *Comm. of the ACM* 43(8), 26–158.

24. Sarwar, B., et al.: 2000, 'Application of dimensionality reduction in recommender systems'. *Proc. of ACM WebKDD Workshop*.

25. Sarwar, B., et al.: 2001, 'Item-based collaborative filtering recommendation algorithms'. *Proc. of WWW2001*, pp. 285–295.

26. Shardanand, U. and Maes, P.: 1995, 'Social information filtering: Algorithms for automating 'word of mouth". *Proc. of CHI'95*, pp. 210–217.

27. Tintarev, N. and Masthoff, J. 'A survey of explanations in recommender systems'. *In IEEE 3rd International Workshop on Web Personalisation, Recommender Systems and Intelligent User Interfaces*: 2007.

28. Ziegler, C., et al.: 2005, 'Improving recommendation lists through topic diversification'. *Proc. of WWW2005*, pp. 22–32.