Hybrid Collaborative and Content-based Music Recommendation Using Probabilistic Model with Latent User Preferences

Kazuyoshi Yoshii†* **Masataka Goto**‡ **Kazunori Komatani**†
Tetsuya Ogata† **Hiroshi G. Okuno**†

†Graduate School of Informatics, Kyoto University *JSPS Research Fellow (DC1)
‡National Institute of Advanced Industrial Science and Technology (AIST)
{yoshii,komatani,ogata,okuno}@kuis.kyoto-u.ac.jp m.goto@aist.go.jp

Abstract

This paper presents a hybrid music recommendation method that solves problems of two prominent conventional methods: collaborative filtering and content-based recommendation. The former cannot recommend musical pieces that have no ratings because recommendations are based on actual user ratings. In addition, artist variety in recommended pieces tends to be poor. The latter, which recommends musical pieces that are similar to users' favorites in terms of music content, has not been fully investigated. This induces unreliability in modeling of user preferences; the content similarity does not completely reflect the preferences. Our method integrates both rating and content data by using a Bayesian network called an aspect model. Unobservable user preferences are directly represented by introducing latent variables, which are statistically estimated. To verify our method, we conducted experiments by using actual audio signals of Japanese songs and the corresponding rating data collected from Amazon. The results showed that our method outperforms the two conventional methods in terms of recommendation accuracy and artist variety and can reasonably recommend pieces even if they have no ratings.

Keywords: hybrid method, probabilistic model, collaborative filtering, content-based recommendation.

1. Introduction

Needs for music recommendation emerge today [1] because we can access various music databases through the Internet. To find favorite musical pieces by using retrieval systems, we have to execute queries repeatedly by ourselves. Therefore, we are often at a loss as to what queries are appropriate. To solve this problem, it is desirable that recommender systems select probably-preferred pieces from the database by estimating our preferences. So far, two major recommendation techniques have been proposed: collaborative filtering and content-based recommendation, which have complementary advantages as described below.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page.
© 2006 University of Victoria

Collaborative methods [2–4] recommend pieces to a user by considering someone else's ratings of those pieces. For example, suppose that there is a target user who likes pieces A and B. If there are many other users who like A, B, and C, C will probably be recommended to the target user. This technique is widely utilized in practical web-shopping services (e.g., Amazon and iTunes music store) and has been demonstrated to be rather effective. However, there are two problems. The first problem is that pieces that have not been rated (e.g., newly-released CDs and minor songs) cannot be recommended. Therefore, there are not many chances of encountering unexpected favorite pieces. The second problem is that artists of the recommended pieces tend to be the same and are often well-known to a target user. Such recommendations may be unsatisfactory or meaningless.

Content-based methods [5–7] recommend pieces that are similar to users' favorites in terms of music content such as moods and rhythms. This leads to a rich artist variety; various pieces including unrated ones can be recommended. To achieve this, it is necessary to associate user preferences with the music content by using a practical database where most users tend to rate few pieces as favorites. However, reliable methods for doing this have not been established. This is because a lot of attention has been paid to developing music retrieval systems [8] in which queries that represent user preferences are prepared by users. Although Hoashi *et al.* [5] tried to model user preferences, their method was only verified in an impractical database with 12 subjects where the balance in ratios of positive and negative ratings was kept to some extent. Logan [6] did not address actual user ratings by assuming a set of pieces in an album-CD as a set of favorite pieces of a particular user. To use Celma's WEB-based recommender system [7], which was not evaluated, users should list their favorite artists used as queries.

To solve the problems in the two methods, we propose a hybrid method that integrates both rating and content data. This enables more accurate recommendations with a rich variety. Many ratings by other users aid the reliable modeling of preferences by compensating for the insufficiency of pieces rated by a target user. A possible way of implementing a hybrid method is to use collaborative and content-based methods in parallel or in cascade [9, 10]. However, substantial user preferences cannot be still captured because observable data (ratings and contents) do not completely reflect the preferences. To solve this, our method is based on an extended version of a three-way aspect model proposed by Popescul *el al.* [11] that directly represents unobservable user preferences as a set of latent variables. In our method, the distribution of mel-frequency cepstral coefficients (MFCCs) is modeled as music content. To our knowledge, our method is the first to apply the aspect model to content extracted not from additional documents (e.g., review comments) but from actual media files.

The rest of this paper is organized as follows. Section 2 specifies a recommendation task, and Section 3 introduces the conventional recommendation methods. Section 4 explains our hybrid recommendation method. Section 5 reports on our experiments using practical rating data from Amazon, and Section 6 summarizes the key points.

2. Specification of Music Recommendation

We firstly describe three requirements for designing recommender systems and then define a recommendation task.

2.1. Our Goal

We aim to satisfy the following requirements:

1. **High recommendation accuracy** A better system will recommend more favorite pieces and fewer disliked ones from a practical database in which the number of ratings given by each user is not sufficient.
2. **Rich artist variety** If the recommended pieces were by various artists unfamiliar to a target user, the chances of finding new artists who play music matching the likes of the user would increase.
3. **Solving new-item problem** This enables users to find appropriate selections that have unfortunately been given few/no ratings. In addition, the variety in recommended pieces is enhanced.

Collaborative systems and content-based systems cannot satisfy all the three requirements, as discussed in Section 1. We believe that their merits will be combined by reflecting both collaborative data (ratings of other users) and content-based data (acoustic features) in the recommendation.

2.2. Recommendation Task

An objective of music recommendation is to rank musical pieces that have not been rated by a target user. Let indexes of users and those of pieces be $U = \{u|1, \cdots, N_U\}$ and $M = \{m|1, \cdots, N_M\}$, respectively. Here, N_U and N_M are the number of users and of pieces. We assume that U and M are registered in a system in advance. Additional metadata (e.g., titles, artist names, and genres) are not necessary. Rating data should be also reserved in the system. In this paper, we focus on scores on a 0-to-4 scale as rating data; let $r_{u,m}$ be a rating score given to piece m by user u, where $r_{u,m}$ is an integer between 0 and 4 (4 being the best). By collecting all the ratings, rating matrix R is obtained by

$$R = \{r_{u,m}|1 \leq u \leq N_U, 1 \leq m \leq N_M\}. \quad (1)$$

When user u has not rated piece m, ϕ is substituted for $r_{u.m}$ as a symbol of representing an "empty" score for convenience. Note that most scores in R are empty in practical data because each user has rated a few pieces in M. Collaborative methods use only R for the recommendation.

To use content-based methods, content data is required. We assume that audio signals of the pieces represented by M are available. The content of each piece is represented as a single vector of several acoustic features extracted from the corresponding audio signal. Let the indexes of those features be $T = \{t|1, \cdots, N_T\}$, where N_T is the number of features (a dimension of the feature vector). Here, $c_{m,t}$ is defined as the t-th feature value of piece m. By collecting all the feature vectors, content matrix C is obtained by

$$C = \{c_{m,t}|1 \leq m \leq N_M, 1 \leq t \leq N_T\}. \quad (2)$$

Given a target user $u \in U$, content-based methods use C and not R but $\{r_{u,m}|1 \leq m \leq N_M\}$ for the recommendation. That is, they do not use scores given by other users in R.

3. Conventional Recommendation Methods

We review typical recommendation methods. which were used for comparison experiments in Section 5.

3.1. Collaborative Filtering

Collaborative methods try to predict unknown rating scores of a target user for musical pieces that have not been rated by the user, considering someone else's scores of those pieces. Given a target user u, let $\widetilde{r}_{u,m}$ be a predicted rating score of user u for piece m, which is given by

$$\widetilde{r}_{u,m} = \bar{r}_u + k \sum_{\{u'|u' \neq u, u' \in U\}} w_{u,u'}(r_{u',m} - \bar{r}_{u'}), \quad (3)$$

where $\bar{r}_u$ and $\bar{r}_{u'}$ are the average rating score of user u and that of user u', respectively. The value $w_{u,u'}$ is a weight that reflects the *preference similarity* between users u and u', and k is a normalizing factor such that the absolute values of the weights sum to unity. That is, $\sum_{u'} |w_{u,u'}| = 1$. After the score prediction, pieces are ranked according to $\widetilde{r}_{u,m}$.

There are several measures for calculating the similarity. The most popular one may be the Pearson correlation coefficient, which shows stable performance in many tasks [3]. By using this measure, the similarity is defined as

$$w_{u,u'} = \frac{\sum_m (r_{u,m} - \bar{r}_u) \sum_m (r_{u',m} - \bar{r}_{u'})}{\sqrt{\sum_m (r_{u,m} - \bar{r}_u)^2 \sum_m (r_{u',m} - \bar{r}_{u'})^2}}, \quad (4)$$

where the summations over m are for the pieces rated by both users u and u'. However, there are usually very few of those pieces when rating matrix R is sparse. Therefore, this basic calculation method often fails.

To solve this problem, empty scores in rating matrix R are replaced with a default score r_D. We empirically set r_D to 2.5, which is a biased score (c.f., a neutral score is 2 on a 0-to-4 scale), because most users tend to provide high scores (3 and 4) more often than low ones (0 and 1).

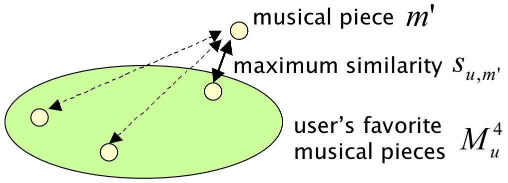

Figure 1. Similarity calculation between user preference and musical pieces in content-based recommendation method.

3.2. Content-based Recommendation

Content-based methods try to rank musical pieces on the basis of music-content similarity by representing user preferences in the music-content space. Let a content vector of piece m be $\boldsymbol{c}_m = (c_{m,1}, \cdots, c_{m,N_T})$. Let a set of pieces which were given r-scores ($0 \leq r \leq 4$) by user u be $M_u^r = \{m | r_{u,m} = r\}$. Given a target user u, Logan's method [6], which may be the most basic one, is applied as follows:

1. If M_u^4 is not empty (i.e., user u has provided 4-scores), we focus on M_u^4. Otherwise, we then focus on M_u^3 if M_u^3 is not empty. After this, we explain the algorithm in the case of using M_u^4. A set of content vectors $\{\boldsymbol{c}_m | m \in M_u^4\}$ represents a preference of user u. We call those vectors *preference vectors*.
2. By using a similarity measure, the similarity between each preference vector $\boldsymbol{c}_m$ $(m \in M_u^4)$ and content vector $\boldsymbol{c}_{m'}$ $(m' \in M, r_{u,m'} = \phi)$ is calculated, as shown in Figure 1. Let the maximum similarity be $s_{u,m'}$, which indicates how likely user u will like piece m'. Then, $s_{u,m'}$ is calculated for each piece m'.
3. Pieces $\{m' | r_{u,m'} = \phi\}$ that have not been rated by user u are ranked according to the similarity $s_{u,m'}$.

The cosine measure is often used to calculate the vector similarity [5], a method also used in this paper.

In our recommendation task, we should deal with the case where user u has provided only scores of 0 and/or 1 (i.e., the user has only rated disliked pieces). In this case, we modify the method so that the similarity between the content of the disliked pieces and that of the recommended pieces is minimized. Note that if user u provides only neutral scores (2-scores), random pieces are recommended.

4. Hybrid Recommendation Method

To meet the three requirements described in Section 2.1, we propose a hybrid method that integrates rating and content data. First, we discuss a problem for modeling user preferences. Next, we explain an unified probabilistic model.

4.1. Problem

To achieve the hybrid recommendation, it is necessary to reflect both rating and content data in modeling of user preferences. However, a problem is that representations of user preferences are different between collaborative methods and content-based methods. The former represents a preference of user u as a N_M-dimensional vector that contains rating scores of all pieces $(r_{u,1}, \cdots, r_{u,N_M})$ (see Section 3.1). The latter represents the preference as a set of N_T-dimensional feature vectors of favorite pieces (see Section 3.2).

In those representations, user preferences are only indirectly represented; observable data such as ratings and features do not completely reflect the preferences. In addition, to build a hybrid recommender system, ad-hoc rules may be used to forcibly merge the two different representations.

4.2. Solution

To solve this problem, we associate rating and content data with newly-introduced variables that represents user preferences. In this paper, we use a Bayesian network called a three-way aspect model proposed by Popescul *et al.* [11]. This model has a set of *latent* variables that directly describe *substantial* preferences that cannot be observed. Those preferences are statistically estimated with theoretical proof. This will contribute to reliable recommendation.

Unfortunately, Popescul's method cannot be directly applied to our task because it was designed for document recommendation. The document content is represented on the basis of a "bag-of-words" model originally proposed in the field of language processing — the contents of a document are represented as a set of frequencies of words.

To apply the three-way aspect model to music recommendation, the contents of a musical piece should be represented as a single vector in which all dimensions are semantically equivalent. For example, each dimension always represents a word frequency in Popescul's method. In addition, all dimensions of each vector should sum to unity.

After this, we firstly discuss how to apply the three-way aspect model to music recommendation. Then, we explain an implementation of our aspect model.

4.2.1. Three-way Aspect Model with Bag of Timbres

To meet the above-mentioned requirements, we propose a *"bag-of-timbres"* model that represents the contents of a musical piece as a set of weights of *polyphonic timbres*. An idea of polyphonic timbres was proposed by Aucouturier *et al.* [12]. The polyphonic timbres represent the perceptual "sounds" not of individual instruments but of their combinations. Those timbres are important features that characterize textures of musical pieces. In addition, there is a merit that modeling of polyphonic timbres can be applied to various signals because separation of instrument parts, which is very difficult, is not required.

In the three-way aspect model, observation data is associated with one of the latent variables $Z = \{z | z_1, \cdots, z_{N_z}\}$, where N_z is the number of latent variables, as shown in Figure 2. The latent variables represent user preferences; each latent variable conceptually corresponds to a *genre*, and a set of proportions of the genres reflects a musical taste of each user. A possible explanation of this model is that a user stochastically chooses a genre according to his or her preference, and then the genre stochastically "generates" pieces and polyphonic timbres. In this model, the conditional independence of users, pieces, and polyphonic timbres through the latent genres is assumed. Note that the aspect model allows multiple genres per user, unlike most clustering methods that assign each user to a single genre class.

4.2.2. Modeling of Polyphonic Timbres

To model timbres of audio signals, mel-frequency cepstral coefficients (MFCCs) have often been used in many studies of genre classification [13]. Aucouturier *et al.* [12] proposed a method of modeling polyphonic timbres for similarity-based audio clustering. Their method applies a Gaussian mixture model (GMM) to MFCCs of each musical piece. The similarity between two pieces is measured as the reciprocal of the distance between the corresponding GMMs, which is obtained by using a sampling method.

To obtain a "bag-of-timbres" of each piece, we also build a GMM by using MFCCs of that piece. We assume that each Gaussian represents MFCC-distribution of a particular polyphonic timbre; mixture weights of Gaussians correspond to weights of timbres. However, Aucouturier's method cannot be applied because Gaussians in a GMM are different from those in another GMM. In other words, each GMM represents a different combination of polyphonic timbres.

Our unique idea to solve this problem is that "bags-of-timbres" of all pieces share the same combination of Gaussians. Means and covariances of the Gaussians are estimated by using numerous MFCCs extracted *not from each piece but from all the pieces*, and mixture weights of the Gaussians are discarded in this estimation. Weights of polyphonic timbres in each piece are obtained as mixture weights of the *fixed* Gaussians in that piece; only the mixture weights are re-estimated by using MFCCs of the single piece.

First, 13-dimensional MFCCs are extracted from audio signals sampled at 16.0 kHz by applying short-time Fourier transformation (STFT) with a Hanning window of 200 [ms]. The shifting interval is 100 [ms]. Then, 28-dimensional feature vectors are obtained (MFCCs, energy, and their delta components). Let feature vectors extracted from piece m be $\{\boldsymbol{f}_{m,i} | 1 \leq i \leq I_m\}$, where I_m is the number of feature vectors. Next, the parameters of the Gaussians are estimated for all the pieces by using the Expectation-Maximization (EM) algorithm [14], where the number of Gaussians is set to 10. That is, $N_T = 10$.

Let the estimated mean vector and covariance matrix of the t-th Gaussian be $\boldsymbol{\mu}_t$ and Σ_t, respectively. The value $c_{m,t}$ is a weight of timbre t in piece m, which is obtained by

$$c_{m,t} = k_m \sum_{i=1}^{I_m} \frac{1}{(2\pi)^{\frac{28}{2}} |\Sigma_t|^{\frac{1}{2}}} \exp\left(-\frac{1}{2} D^2(\boldsymbol{f}_{m,i}, \boldsymbol{\mu}_t)\right), \quad (5)$$

where D^2 is the squared Mahalanobis distance given by $(\boldsymbol{f}_{m,i} - \boldsymbol{\mu}_t)^T \Sigma_t^{-1} (\boldsymbol{f}_{m,i} - \boldsymbol{\mu}_t)$ and k_m is a normalizing factor such that $\sum_t c_{m,t} = 1$ ($m \in M$).

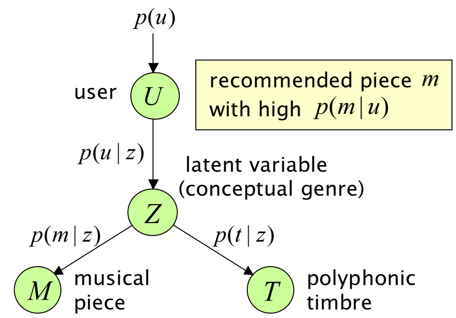

Figure 2. Asymmetric representation of our proposed aspect model using polyphonic timbre weights as music content.

4.2.3. Estimation of Model Parameters

An asymmetric specification of the joint probability distribution $p(u, m, t, z)$ over U, M, T, and Z is given by

$$p(u, m, t, z) = p(u)p(z|u)p(m|z)p(t|z). \quad (6)$$

An equivalent symmetric specification is obtained by

$$p(u, m, t, z) = p(z)p(u|z)p(m|z)p(t|z). \quad (7)$$

Marginalizing out z, we obtain the joint probability distribution $p(u, m, t)$ over U, M, and T as follows:

$$p(u, m, t) = \sum_z p(z)p(u|z)p(m|z)p(t|z). \quad (8)$$

Model parameters $p(z)$, $p(u|z)$, $p(m|z)$, and $p(t|z)$ are determined using the EM algorithm [14] to find a local maximum of the log-likelihood of the training observation data. Let $n(u, m, t)$ be an indicator of how much user u likes polyphonic timbre t in piece m, which is given by

$$n(u, m, t) = r_{u,m} \times c_{m,t}, \quad (9)$$

where $r_{u,m}$ and $c_{m,t}$ are defined as follows:

- $r_{u,m}$ is a rating score of user u for piece m. In our method, a default rating score (2.5) is substituted for empty scores, as described in Section 3.1.
- $c_{m,t}$ is a weight of polyphonic timbre t in piece m.

Given training data in this form, the log likelihood L is

$$L = \sum_{u,m,t} n(u, m, t) \log p(u, m, t). \quad (10)$$

Therefore, the corresponding EM algorithm is given by

E step

$$p(z|u, m, t) = \frac{p(z)p(u|z)p(m|z)p(t|z)}{\sum_{z'} p(z')p(u|z')p(m|z')p(t|z')}, \quad (11)$$

M step

$$p(u|z) \propto \sum_{m,t} n(u, m, t)p(z|u, m, t), \quad (12)$$

$$p(m|z) \propto \sum_{u,t} n(u, m, t)p(z|u, m, t), \quad (13)$$

$$p(t|z) \propto \sum_{u,m} n(u, m, t)p(z|u, m, t), \quad (14)$$

$$p(z) \propto \sum_{u,m,t} n(u, m, t)p(z|u, m, t). \quad (15)$$

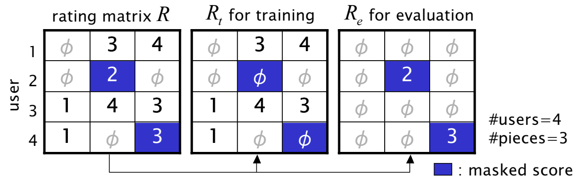

Figure 3. Example of dividing rating matrix R into training matrix R_t and evaluation matrix R_e.

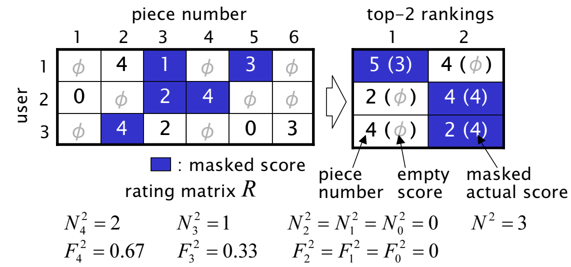

Figure 4. Example of evaluating results.

The E and M steps are iterated alternately until the log-likelihood L converges to a local maximum. For practical use, it is better to adopt an extended version of the EM algorithm (e.g., the deterministic annealing EM algorithm [16]) to cope with the data sparseness. In this paper, N_Z is set to 10. After the training, musical pieces are ranked for target user u according to $p(m|u) \propto \sum_t p(u, m, t)$.

5. Evaluation

To compare our hybrid method with the conventional methods described in Section 3, we performed experiments.

5.1. Experimental Conditions

To perform reliable experiments, it is ideal to use large-scale rating data in which the number of ratings given by each user is sufficient to a certain extent. However, the construction of that data via subjective experiments is extremely time-consuming. One possible way is to collect rating scores from web sites [15]. Amazon provides application programming interfaces (APIs) [17] that allow us to download almost all information in web sites.

The musical pieces we used are Japanese songs of single-CDs that were ranked in the weekly top-20 sales rankings from Apr. 2000 to Dec. 2005. The corresponding scores with user IDs were collected from Amazon. If a user has rated multiple pieces, we can identify the scores given by the same user. However, there were many unreliable users and pieces that had few/no scores. To solve this, we extracted users and pieces so that the number of scores given by a user and that of scores given to a piece were always more than 4. As a result, N_U was 316 and N_M was 358. The density of actual scores in rating matrix R was 2.19%, which is still sparse. This means R is practical data.

By using the above mentioned rating data, we compared our hybrid method based on the aspect model (called *AM*) with the two conventional methods: collaborative filtering (called *CF*) and content-based recommendation (called *CB*). The value for N_T (the number of polyphonic timbres) in *CB* and *AM* was 10, while N_Z (the number of latent variables) in *AM* was set to 10.

5.2. Evaluation Metrics

The experiments were conducted by using 10-fold cross validation; rating matrix R was randomly divided into training matrix R_t and evaluation matrix R_e by masking 10% of actual scores in R, as shown in Figure 3. The three methods ranked musical pieces for each user by using R_t without R_e. The rankings were evaluated from the viewpoints of recommendation accuracy and artist variety.

The recommendation accuracy was evaluated by examining the total top-x rankings of all users ($x = 1, 3, 10$). However, we can only use the actual scores in R_e for the evaluation because R_e includes many empty scores. Here, we focus on $x \times N_U$ pieces in top-x rankings of N_U users. Let N_r^x be the number of scores that were masked in R_t but were actually r ($0 \leq r \leq 4$) in R_e. Figure 4 shows an example in the case of $x = 2$. The term N^x is defined as $N^x = \sum_r N_r^x$, which is much less than $x \times N_U$. A ratio F_r^x is then obtained by $F_r^x = N_r^x / N^x$. Therefore, the higher value of F_4^x indicates the better performance. Note that the chance rate of F_r^x is not 20% but is the same as the ratio of pieces that were given r-scores in R. The chance rates of $F_4^x, \cdots, F_0^x$ are 57.9%, 19.1%, 8.6%, 4.9%, and 9.5%.

The artist variety was evaluated by calculating the recommendation variety for each user. Given a target user u, v_A^x is defined as the number of artists in top-x rankings. Here, v_M^x is defined as the number of pieces by new artists whose pieces have not been rated by user u. Let V_A^x and V_M^x be the average of v_A^x and v_M^x over all users ($V_A^x, V_M^x \leq x$). The higher values of V_A^x and V_M^x indicate the richer variety.

5.3. Experimental Results

Table 1 presents the recommendation accuracy. The values $F_4^x (x = 1, 10)$ by *AM* are much higher than those by *CF* and *CB*. F_4^3 by *AM* is almost equal to that by *CB*. Although F_4^{10} by *CB* was greatly degraded from F_4^3 by *CB*, F_4^{10} by *AM* is almost equal to F_4^3 by *AM*. The values F_0^x and F_1^x ($x = 1, 3, 10$) by *AM* are lower than those by *CF* and *CB*. These results indicate that *AM* outperforms *CF* and *CB* in terms of the capability of recommending the favorites.

Note that N_4^x and N^x by *AM* are much lower than those by *CF*, as indicated in Table 2. To explain this, we set up a hypothesis that the same artists tend to be recommended by *CF* because most users tend to rate pieces by the same artists. If this is right, N^x (the number of pieces that have been actually rated) will surely become large in *CF*.

The proof of our hypothesis is demonstrated by examining the artist variety shown in Table 3. The values for V_A^x and V_M^x by *CF* are much lower than those by *AM*. Although

Table 1. Evaluation of recommendation accuracy.

	F_4^x			F_3^x			F_2^x			F_1^x			F_0^x		
x	CF	CB	AM	CF	CB	AM	CF	CB	AM	CF	CB	AM	CF	CB	AM
1	77.6%	85.0%	**92.0%**	13.8%	5.00%	4.00%	3.45%	5.00%	4.00%	0.86%	5.00%	0.00%	4.31%	0.00%	0.00%
3	77.5%	**82.5%**	80.3%	15.4%	12.5%	11.5%	3.08%	2.50%	6.56%	0.04%	2.50%	1.64%	3.52%	0.00%	0.00%
10	70.8%	69.4%	**79.5%**	18.1%	17.9%	10.6%	6.51%	4.48%	6.21%	0.10%	5.22%	1.24%	3.61%	2.99%	2.48%

Table 2. The number of evaluated pieces.

	N_4^x			N^x		
x	CF	CB	AM	CF	CB	AM
1	**90**	17	23	**116**	20	25
3	**176**	33	49	**227**	40	61
10	**294**	93	128	**415**	134	161

Table 3. Evaluation of artist variety.

	V_A^x			V_M^x		
x	CF	CB	AM	CF	CB	AM
1	1.00	1.00	1.00	0.627	0.933	**0.938**
3	2.49	**2.93**	2.80	2.09	2.76	**2.78**
10	7.58	**9.30**	8.68	8.01	9.17	**9.33**

Table 4. Verification of capability for recommending no-rated musical pieces.

	F_4^x		F_3^x		F_2^x		F_1^x		F_0^x		N^x	
x	CB	AM	CB	AM	CB	AM	CB	AM	CB	AM	CB	AM
1	89.5%	100%	5.26%	0.00%	0.00%	0.00%	5.26%	0.00%	0.00%	0.00%	19	1
3	76.9%	80.0%	12.8%	0.00%	5.12%	0.00%	5.12%	0.00%	0.00%	20.0%	39	5
10	64.1%	72.7%	19.8%	9.10%	5.34%	0.00%	6.87%	0.00%	3.81%	18.2%	131	11

V_A^x by *AM* is lower than that by *CB*, V_M^x by *AM* is higher that by *CB*. These results indicate that *AM* can produce the most suitable recommendation with a rich variety.

5.4. Capability for Solving New-item Problem

We verified the capability of *AM* for recommending pieces that have not been rated. To do this, 10-fold cross validation was performed by masking actual scores given to 10% of all the pieces. Table 4 lists the results. *CF* cannot recommend unrated pieces. Because *AM* considers not only music content but also user ratings, the number of recommended unrated pieces that can be evaluated is small. However, it seems that *AM* as well as *CB* can reasonably recommend the most favorite pieces even if they have no ratings.

6. Conclusion

This paper has presented a music recommendation method that simultaneously considers user ratings and content similarity. Our hybrid method is based on a three-way aspect model, which can directly represent substantial (unobservable) user preferences as a set of latent variables introduced in a Bayesian network. Probabilistic relations over users, ratings, and contents are statistically estimated. Experimental results showed that our method outperforms conventional collaborative or content-based methods in recommendation accuracy. We can conclude that the high recommendation accuracy was achieved by the reliable modeling of user preferences and the integration of rating and content data.

In the future, we will try to use automatically-described various features such as tempi, rhythms and genres for representing the music content. It is also necessary to deal with entry of unregistered users and musical pieces by incrementally training the aspect model. In addition, we plan to apply our method to community assistance by associating a user with other users who have similar preferences.

Acknowledgments

This research has been partially supported by a Grant-in-Aid for Scientific Research from JSPS Research Fellowship.

References

[1] A. Uitdenbogerd and R. van Schyndel, "A review of factors affecting music recommender success," *ISMIR*, 2002.

[2] U. Shardanand and P. Maes, "Social information filtering: Algorithms for automating "Word of Mouth"," *ACM CHI'95 Conference on Human Factors in Computing Systems*, 1995, pp. 210–217.

[3] J. Breese, D. Heckerman, and C. Kadie, "Empirical analysis of predictive algorithms for collaborative filtering," *UAI*, 1998, pp. 43–52.

[4] W. Cohen and W. Fan, "Web-collaborative filtering: Recommending music by crawling the Web," *Computer Networks*, vol. 33, no. 1–6, pp.685–698, 2000.

[5] K. Hoashi, K. Matsumoto, and N. Inoue, "Personalization of user profiles for content-based music retrieval based on relevance feedback," *ACM Multimedia*, 2003, pp.110–119.

[6] B. Logan, "Music recommendation from song sets," *ISMIR*, 2004, pp. 425–428.

[7] O. Celma, M. Ramirez, and P. Herrera, "Foafing the music: A music recommendation system based on RSS feeds and user preferences," in *ISMIR*, 2005, pp.464–457.

[8] R. Typke, F. Wiering, and R. Veltkamp, "A survey of music information retrieval systems," *ISMIR*, 2005, pp. 153–160.

[9] P. Melville, R. Mooney, and R. Nagarajan, "Content-boosted collaborative filtering," *SIGIR*, 2001.

[10] C. Hayes, "Smart Radio: Building Community-Based Internet Music Radio." *Doctoral Thesis*, Trinity College Dublin, 2003.

[11] A. Popescul, L. Ungar, D. Pennock, and S. Lawrence, "Probabilistic models for unified collaborative and content-based recommendation in sparse-data environments," *UAI*, 2001, pp. 437–444.

[12] J.-J. Aucouturier, F. Pachet, and M. Sandler, "The way it sounds": Timbre models for analysis and retrieval of music signals," *IEEE Trans. Multimedia*, vol. 7, no. 6, pp. 1028–1035, 2005.

[13] J.-J. Aucouturier and F. Pachet, "Musical genre: A survey," *New Music Research*, vol. 32, no. 1, pp. 83–93, 2003.

[14] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Royal Statistical Society*, B, vol. 39, pp. 1–38, 1977.

[15] M. Zadel and I. Fujinaga, "Web services for music information retrieval," *ISMIR*, 2004, pp.478–483.

[16] N. Ueda and R. Nakano, "Deterministic annealing EM algorithm," Neural Networks, Vol.11, No. 2, pp.271-282, 1998.

[17] Amazon Web Services: www.amazon.com/gp/aws/landing.html.