IMPROVING EFFICIENCY AND SCALABILITY OF MODEL-BASED MUSIC RECOMMENDER SYSTEM BASED ON INCREMENTAL TRAINING

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

We aimed at improving the efficiency and scalability of a hybrid music recommender system based on a probabilistic generative model that integrates both collaborative data (rating scores provided by users) and content-based data (acoustic features of musical pieces). Although the hybrid system was proved to make accurate recommendations, it lacks efficiency and scalability. In other words, the entire model needs to be re-trained from scratch whenever a new score, user, or piece is added. Furthermore, the system cannot deal with practical numbers of users and pieces on an enterprise scale. To improve efficiency, we propose an incremental method that partially updates the model at low computational cost. To enhance scalability, we propose a method that first constructs a small "core" model over fewer virtual representatives created from real users and pieces, and then adds the real users and pieces to the core model by using the incremental method. The experimental results revealed that the proposed system was not only efficient and scalable but also outperformed the original system in terms of accuracy.

1 INTRODUCTION

Music recommender systems play important roles in current e-commerce to help users discover their favorites in huge databases [1]. For example, many e-commerce sites (e.g., Last.fm and Amazon.com [2]) use collaborative filtering techniques to recommend musical pieces to the user by examining how someone else has rated them. Although these techniques have been considered to be effective, they suffer from the famous "new item" problem. That is, non-rated pieces cannot be recommended. In addition, the variety of recommendations tends to be poor because most users mainly rate musical pieces by a small number of popular artists. This indicates that there still remains much room for enhancing the Long-Tail effect [3].

To overcome these limitations, content-based filtering techniques, which recommend musical pieces similar to user's favorites in terms of musical content, are attracting attention of many researchers. However, a major approach based on *automatic content analysis* for musical audio signals [4, 5] has not fully gained the popularity of end users. In contrast, Pandora, which depends on *manual content annotation* for commercial titles, is a well-known successful radio station on the Internet. This indicates that we should take into account important factors, such as cultural backgrounds and popularity on the market, that contribute to making reasonable recommendations but cannot be obtained from audio signals. However, this annotation-based approach lacks portability because it is not practical to manually annotate all compositions by amateurs in social networking services (e.g., MySpace.com).

To make reasonable recommendations under any conditions, it is necessary to take a flexible hybrid approach that can integrate various types of available data. This improves robustness against inappropriate data in real world (e.g., malicious rating scores, erroneous automatic annotations, and inconsistent manual annotations).

We therefore developed a hybrid recommender system using a probabilistic model that integrates both collaborative and content-based data in a theoretical way [6]. Although our system overcame the shortcomings of conventional techniques, critical problems in efficiency and scalability emerged when we tried to apply our system to e-commerce where several millions of users and pieces are managed. The system can neither promptly adapt recommendations to each user according to changes in his or her rating scores nor incrementally register new users and pieces. This is because the model should always be trained from scratch, where the time for training is proportional to both numbers of users and pieces.

To improve efficiency, we propose an online method that updates only partial parameters of the model related to changes in the observed data. This enables the system to incrementally incorporate these changes into the model. To improve scalability, we propose a method that builds a small "core" model over fixed numbers of virtual representatives created from large numbers of real users and pieces. The core model is then updated while incrementally registering real users and pieces.

The rest of this paper is organized as follows. Section 2 reviews our recommender system. Section 3 explains the proposed methods. Section 4 reports on the experiments. Section 5 summarizes the key findings of this study.

2 HYBRID MUSIC RECOMMENDER SYSTEM

We will first define a recommendation task and then explain the original version of our recommender system [6].

2.1 Task Statement

The objective of music recommendation is to rank musical pieces that have not been rated by a target user. We let $U = \{u|1, \cdots, N_U\}$ be the indices of users and $M = \{m|1, \cdots, N_M\}$ be those of pieces, where N_U is the number of users and N_M is that of pieces. We assumed that U and M were *registered* in the system in advance.

Collaborative data are rating scores, which are also registered in the system. In this paper, we focus on scores on a 0-to-4 scale as rating data. We 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 rating scores, rating matrix R is obtained by

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

When user u has not rated piece m, ϕ is substituted for $r_{u.m}$ as a symbol, representing an "empty" score for convenience. Note that most scores in R are empty in actual data because all users have rated a few pieces in M.

Content-based data are acoustic features automatically extracted from the polyphonic audio signals of all musical pieces, M. We assumed that each piece would be represented as a single vector of musical features. Let $T = \{t|1, \cdots, N_T\}$ be the indices of these features, where N_T is the total number (a dimension of the vector). Here, $c_{m,t}$ is defined as the t-th element value of piece m. By collecting all the feature vectors, content matrix C is obtained by

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

The method of extracting features we use is based on the *bag-of-timbres* model [6]. Note that we can incorporate mannual annotations into calculating maxtix C.

2.2 Recommendation Method

To integrate the collaborative and content-based data, we used a probabilistic generative model, called a three-way aspect model [7]. It explains the generative process for the observed data by introducing a set of latent variables. These variables correspond to *conceptual genres*, which are not given in advance. As part of the generative process, the model directly represents user preferences (how much each genre is preferred by a target user), which are statistically estimated with a theoretical proof.

The observed data are associated with latent variables, $Z = \{z|1, \cdots, N_z\}$, where N_z is the total number of these, as outlined in Fig. 1. Each latent variable corresponds to a conceptual genre. Given user u, the set of conditional probabilities $\{p(z|u)|z \in Z\}$ reflects the musical taste of user u. One possible interpretation is that user u stochastically selects genre z according to his or her preference $p(z|u)$, and genre z then stochastically generates piece m and acoustic feature t according to their probabilities, $p(m|z)$ and $p(t|z)$. We assumed the conditional independence of users, pieces, and features through the latent genres. This is the key point of our model.

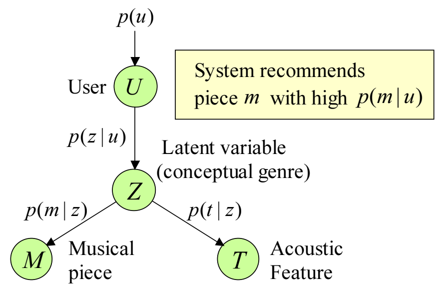

Figure 1. Asymmetric representation of aspect model.

2.2.1 *Formulation of Three-way Aspect Model*

We will now explain the mathematical formulation for the three-way aspect model. The assumption of conditional independence over U, M, and T through Z leads to an asymmetric specification for the joint probability distribution $p(u, m, t, z)$, which is given by

$$p(u, m, t, z) = p(u)p(z|u)p(m|z)p(t|z), \quad (3)$$

where $p(u)$ is the prior probability of user u. $p(u, m, t, z)$ is the probability that user u will select genre z and simultaneously listen to timbre t in piece m.

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

$$p(u, m, t) = \sum_z p(u)p(z|u)p(m|z)p(t|z), \quad (4)$$

where the unknown model parameters are $\{p(z|u)|z \in Z, u \in U\}$, $\{p(m|z)|m \in M, z \in Z\}$, and $\{p(t|z)|t \in T, z \in Z\}$, which are estimated by using rating matrix R and content matrix C. After these are estimated, musical pieces are ranked for given user u' according to $p(m|u') \propto \sum_t p(u', m, t) \propto \sum_{t,z} p(z|u')p(m|z)p(t|z)$.

2.2.2 *Estimation of Model Parameters*

We will next explain how the model parameters are estimated. Let a tuple (u, m, t) be an event where user u listens to timbre t in piece m. Here, we assumed that each event would occur independently. The likelihood of the parameters for the observed data is given by

$$L' = \prod_{u,m,t} p(u, m, t)^{n(u,m,t)}, \quad (5)$$

where $n(u, m, t)$ is the number of events (u, m, t). In this study, we assumed that $n(u, m, t)$ was proportional to the product of $r_{u,m}$ and $c_{m,t}$. That is, $n(u, m, t) \propto r_{u,m} \times c_{m,t}$. This is based on the general observation that event (u, m, t) occurs more frequently if user u prefers piece m more or the weight of timbre t in piece m is higher.

Given the observed data (rating matrix R and content matrix C), the log-likelihood, L, is obtained by

$$L = \sum_{u,m,t} n(u, m, t) \log p(u, m, t). \quad (6)$$

To estimate the parameters that maximize Eq. (6), we use the deterministic annealing EM (DAEM) algorithm [8], which can avoid the local maximum problem.

3 IMPROVED EFFICIENCY AND SCALABILITY

Recommender systems can be categorized into memory-based and model-based groups in terms of methodology. The former always uses the entire data, R and C, to make recommendations. The latter, on the other hand, uses these data to train the assumed models of estimating user preferences. These models are then used to make recommendations. The latter can generally achieve better responses in ranking musical pieces once the models are constructed. However, the computational cost involved in training these models tends to be high. Our system, which belongs to the latter group, also suffers from this disadvantage.

3.1 Problems and Approach

The computational complexity of training the aspect model via the EM algorithm is $O(N_U N_M N_T N_Z) \approx O(N_U N_M)$, taking into account that N_T and N_Z, which are set to 64 and 10, remain constant. This causes two serious issues. One concerns efficiency; this costly training is required whenever the observed data changes. The other concerns scalability; both the computational time and memory load rapidly increase according to $O(N_U N_M)$. Although the efficiency and scalability are important factors for practically managing recommender systems on a commercial scale, they have scarcely been taken into account.

To improve the efficiency, we propose an incremental training method for the three-way aspect model. Ours is an extended version of a method given by Zhang *et al.* [9] that efficiently updates the basic (two-way) aspect model used for collaborative filtering. To improve the scalability, we integrate the incremental training method with a clustering method while improving the recommendation accuracy. Note that the use of clustering methods degrades the accuracy in general due to some approximations [10].

3.2 Incremental Training Method

We will now explain the incremental training method that updates the partial parameters of the aspect model. After this, we will call the model that is initially obtained using the EM-based training method a *base model*. On the other hand, we will call the model that is obtained by incrementally training the base model an *updated model*.

Our method individually addresses the following three cases to obtain the updated model:

1. Updating the base model for a registered user ($\in U$) who provides new rating scores.
2. Extending the base model for a non-registered user ($\notin U$) who provides some rating scores.
3. Extending the base model for a non-registered piece ($\notin M$) that has no rating scores.

While the size of the model (the number of parameters) remains unchanged in the first case, it increases in the others because non-registered users or pieces are added.

3.2.1 *Updating Profiles of Registered Users*

Given specific user u, conditional probabilistic distribution $\{p(z|u)|z \in Z\}$, which is called a *user profile*, captures the user's musical preference. Recall that $p(z|u)$ represents how likely user u is to select conceptual genre z according to the musical preference. The model assumes that the profiles of all users are independent. Therefore, when a user newly provides (changes) some rating scores, we only need to update his or her profile without affecting the profiles of the others to keep the log-likelihood maximized. This contributes to improved efficiency.

We aimed at updating the profile of registered user u': $\{p(z|u')|z \in Z, u' \in U\}$, where user u' has provided some new rating scores. We assumed that model parameters other than the profile of user u' would be constant. Therefore, the maximization of log-likelihood L is equivalent to that of the sum of terms concerning user u' in L. We let $L_{u'}$ be the log-likelihood for the observed data concerning user u', which is given by

$$L_{u'} = \sum_{m,t} n(u',m,t) \log p(m,t|u') \quad (7)$$

$$= \sum_{<m,t|u'>} \log \sum_z p(m|z)p(t|z)p(z|u'), \quad (8)$$

where we introduced a new operator, $\sum_{<m,t|u'>}$, for X (X is arbitrary), which represents $\sum_{m,t} n(u',m,t)X$. Using Jensen's inequality, we can rewrite Eq. (8) as

$$L_{u'} = \sum_{<m,t|u'>} \log \sum_z \frac{p(m|z)p(t|z)}{\delta_{m,t}} p(z|u')\delta_{m,t} \quad (9)$$

$$\geq \sum_{<m,t|u'>} \sum_z \frac{p(m|z)p(t|z)}{\delta_{m,t}} \log p(z|u') + \sum_{<m,t|u'>} \log \delta_{m,t}, \quad (10)$$

where $\delta_{m,t}$ is given by $\delta_{m,t} = \sum_z p(m|z)p(t|z)$.

Because $p(m|z)$ and $p(t|z)$ are constant, the maximization of $L_{u'}$ is equivalent to that of the first term of Eq. (10). This introduces the following maximization problem:

$$\text{Maximize} \quad \sum_{<m,t|u'>} \sum_z \frac{p(m|z)p(t|z)}{\delta_{m,t}} \log p(z|u'), \quad (11)$$

$$s.t. \sum_z p(z|u') = 1, \quad (12)$$

where Eq. (11) is an objective function and Eq. (12) is a constraint function. $\{p(z|u')|z \in Z\}$ are the variables to be optimized. Using the Lagrangian multiplier method, we can finally obtain the user-profile updating formula:

$$p(z|u') = \frac{\sum_{m,t} n(u',m,t) \frac{p(m|z)p(t|z)}{\sum_{z'} p(m|z')p(t|z')}}{\sum_{m,t} n(u',m,t)}. \quad (13)$$

3.2.2 *Creating Profiles of Non-registered Users*

We aimed at creating the profile of non-registered user u': $\{p(z|u')|z \in Z, u' \notin U\}$, where user u' has some rating scores for registered pieces, $\{r_{u',m}|m \in M\}$. Note that these scores were not used for training the base model. In this case, we can apply the updating formula (13) to create the profile by using $p(m|z)$ and $p(t|z)$ that were estimated by using the rating scores of registered users, U.

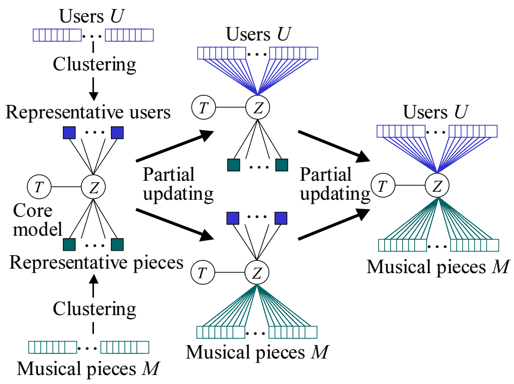

Figure 2. Overview of scalability enhancement method.

The computational complexity of creating (updating) the profile of user u' is $O(\Delta N_M)$, where ΔN_M is the number of pieces that were rated by user u'. We only need to recalculate the ΔN_M terms concerning these pieces in each summation of the updating formula (13).

3.2.3 Registering Non-registered Musical Pieces

We aimed at estimating the probabilities that piece m' will be generated from conceptual genres: $\{p(m'|z)|z \in Z\}$, where m' is a non-registered piece. Here, we can obtain the probabilistic distribution $\{p(z|m')|z \in Z\}$ in the same way as that described above. After the distribution is obtained, $p(m'|z)$ is given by $p(m'|z) \propto p(z|m')/p(z)$. Note that no piece m' has been rated by users. That is, only content-based data $\{c_{m',t}|t \in T\}$ are available. Therefore, the constant-time formula is obtained by

$$p(z|m') = \frac{\sum_t c_{m',t} \frac{p(t|z)}{\sum_{z'} p(t|z')}}{\sum_t c_{m',t}}. \qquad (14)$$

3.3 Scalability Enhancement Method

Let us now explain the scalability enhancement method, which enables the system to efficiently deal with large numbers of users and musical pieces. Figure 2 shows the overview of the method. The method is first used to construct a "core" model for fewer virtual *representative users and pieces* by normally using EM-based offline training. Then, all users, U, and all pieces, M, which are virtually regarded as non-registered users and pieces, are added to the core model by using incremental training. Note that there are two orders in adding U and M. We determined the best one in Section 4.4. Here, the problem is how to create representative users and pieces from U and M.

To solve this problem, we introduced clustering such as the K-means method. All users, U, were classified into a small number of user groups. We used the Pearson correlation coefficient to calculate the similarity two users had in preferences. This is a typical measure in collaborative filtering (see [6]). On the other hand, all pieces, M, were classified into a small number of music groups according to the Euclidean distance between the feature vectors of two pieces. Finally, the representative users and pieces were determined as the centroids of these groups.

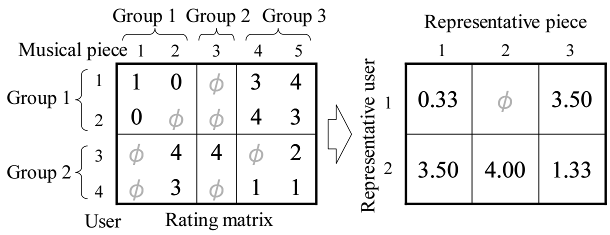

Figure 3. Calculation of new rating matrix for representative users and musical pieces.

One problem remaining is how to create a rating matrix and a content matrix for these representatives. These matrices are used to train the core model. The former is obtained as outlined in Fig. 3. A score provided to a representative piece by a representative user is the average of *actual* scores that were provided to musical pieces in the corresponding music group by users in the corresponding user group. The latter is obtained by calculating the average of feature vectors in each music group.

4 EXPERIMENTAL EVALUATION

Here, we report on several experiments that were conducted to evaluate our methods.

4.1 Experimental Conditions

The collaborative and content-based data (R and C) were prepared in the same way as in our previous study [6]. The musical pieces we used were Japanese songs on single CDs that were ranked in the weekly top-20 sales rankings from 2000 to 2005. The corresponding rating scores were collected from Amazon.co.jp. After unreliable users and pieces had been removed that had less than four scores, N_U was 316 and N_M was 358. The percentages for scores $0, \cdots, 4$ in rating matrix R correspond to 57.9%, 19.1%, 8.57%, 4.85%, and 9.54%. The density of R was 2.19%.

4.2 Evaluation Measure

The experiments were conducted with 10-fold cross validation, i.e., training matrix R_t and evaluation matrix R_e were created from rating matrix R by randomly masking 10% of the actual scores in R, as outlined in Fig. 4.

We used an evaluation measure we had previously proposed [6] to calculate the accuracy of recommendations. This measure calculates the ratio of favorites to the number of recommended pieces whose scores are masked over all users. We examined the entire top-x rankings of all users ($x = 1, 3, 10$). Figure 5 shows an example in the case of $x = 3$. Note that we could not evaluate all the recommended pieces (the total was xN_U) because most of them had not actually been rated by users (the corresponding scores were ϕ in R_e). Here, we let N_r be the total number of recommended pieces whose scores were masked but were actually r ($0 \le r \le 4$), and let N be $N = \sum_r N_r$. Obviously, N was much less than xN_U. We let A_r be the ratio of N_r to N, That is, $A_r = N_r/N$. A higher value for A_4 indicates better performance. If random pieces are recommended, A_4 will become 57.9%, which is the same as the percentage for score 4 in R.

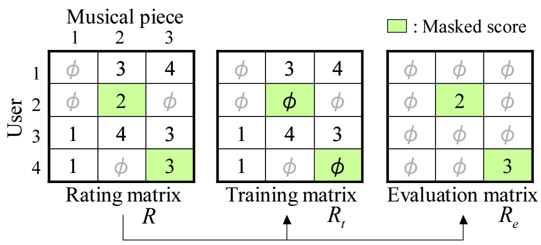

Figure 4. Data preparation for 10-fold cross validation.

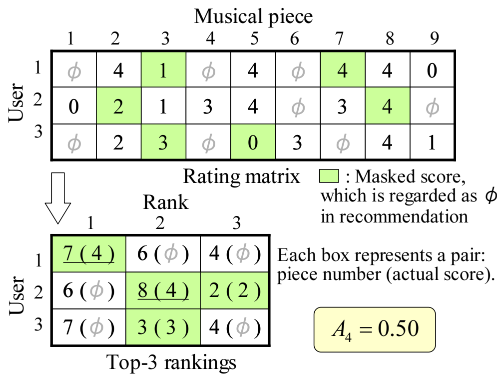

Figure 5. Calculation of recommendation accuracy.

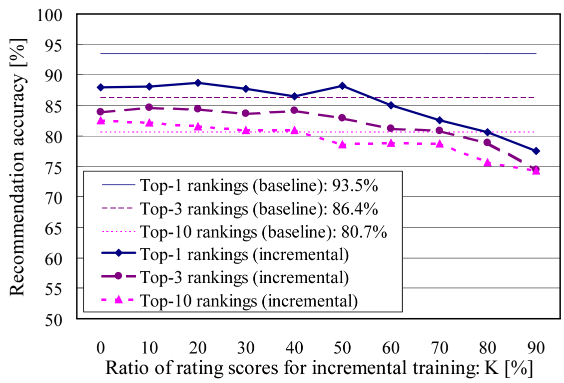

Figure 6. Decrease in recommendation accuracy A_4 according to increase in scores for incremental training.

4.3 Evaluation of Incremental Training Method

We evaluated our incremental training method for the three cases described in Section 3.

4.3.1 Recommendations to Registered Users

An objective was to observe the decrease in the accuracy of recommendations according to the decrease in the percentage of rating scores that were used to construct the base models. In addition, we examined the differences in accuracy between base and updated models.

Let us first explain the experimental procedures. Using rating matrix R_t, we prepared a base model and a total of ten updated models. The former was constructed by using R_t as training data. The latter was obtained as follows:

1. A temporary rating matrix, R'_t, was prepared by randomly masking the K% ($K = 0, 10, 20, \cdots, 90$) of actual scores in training matrix R_t.
2. A temporary base model was built by using R'_t as training data.
3. An updated model was obtained by incrementally adding the masked scores, i.e., by using R_t.

Each model was used to rank the musical pieces. To calculate accuracies, we used evaluation matrix R_t in all the settings. These procedures were iterated ten times while switching the ten rating matrices that were prepared for 10-fold cross validation described in Section 4.2.

Figure 6 plots the results, which shows that our method can appropriately adapt recommendations according to the increase in rating scores. We found that the accuracy hardly deteriorated even when the amount of rating scores used to update the base model was increased to that for building it ($K = 50$). Although the largest difference was about 5% in examining the top-1 rankings ($x = 1$), we can say that a sufficiently high accuracy was maintained.

4.3.2 Recommendations to Non-registered Users

An objective was to compare recommendations to registered users with those to users who had not been registered in terms of recommendation accuracy, A_4. Smaller differences in accuracies indicate better performance.

We will now explain the experimental procedures:

1. 10% of users, U_{new}, were randomly selected from U. They were regarded as non-registered users. We let U_{reg} be the remaining users (registered users).
2. A reduced training matrix, R'_t, was obtained by removing U_{new} from training matrix R_t. That is, the size of matrix R'_t was reduced to 90% of that of R_t.
3. A temporary base model was constructed by using R'_t as training data.
4. To calculate the recommendation accuracy for U_{reg}, we first did the following procedures:
   (a) Profiles of U_{reg} in the base model were updated by using R'_t again.
   (b) Recommendations based on the updated profiles were evaluated by using the rating scores of U_{reg} in evaluation matrix R_e.

   To calculate the recommendation accuracy for U_{new}, we then did the following procedures:
   (a) Profiles of U_{new} were created by using the rating scores of U_{new} that were removed in step (2).
   (b) Recommendations based on the created profiles were evaluated by using the rating scores of U_{new} in evaluation matrix R_e.

These procedures were iterated ten times while switching the ten rating matrices that were prepared for 10-fold cross validation. To evaluate the average and variance in accuracy, we repeated this experiment ten times.

Figure 7 plots the results, which demonstrates that our incremental method can make accurate recommendations to non-registered users as well as to registered users. We found that the variances in accuracy tended to differ at a significant level of 5% through F-tests. However, t-tests revealed that there were no differences in the average accuracies in the three types of rankings.

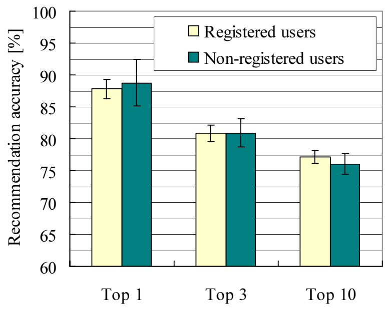

Figure 7. Recommendation accuracy A_4 for registered and non-registered users.

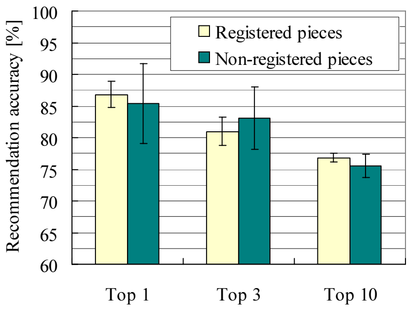

Figure 8. Recommendation accuracy A_4 for registered and non-registered musical pieces.

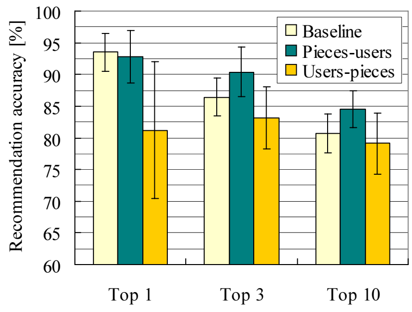

Figure 9. Recommendation accuracy A_4 by baseline (original) system and two scalable systems.

4.3.3 Recommendations of Non-registered Musical Pieces

An objective was to compare recommendations of registered musical pieces with those of non-registered pieces. in terms of recommendation accuracy, A_4. Smaller differences in accuracies indicate better performance.

The experimental procedures were similar to those described in Section 4.3.2 except that "U_{new}" and "U_{reg}" were replaced with "M_{new}" and "M_{reg}," where M_{new} were 10% randomly selected from M, and M_{reg} were the remaining pieces (registered pieces). A base model was trained by using partial data concerning M_{reg}. An updated model was obtained by extending the base model to the entire data including M_{new}.

Figure 8 plots the results, which shows that our incremental method can accurately recommend non-registered pieces as well as registered pieces. We found no differences in the average accuracies in the three types of rankings through F-tests and t-tests.

4.4 Evaluation of Scalability Enhancement Method

An objective is to compare the baseline system described in Section 2 with two scalable systems called a *pieces-users* system and a *users-pieces* system in terms of recommendation accuracy, A_4. The two scalable systems shared the same core model for virtual representative users and pieces, and updated it in different ways corresponding to the lower and upper paths in Fig. 2, where the numbers of these representatives were set to 50 each.

Figure 9 plots the results, which shows that the pieces-users system outperformed the others. This system gained significant advantages over the baseline system in the average accuracies in the top-3 and top-10 rankings although there were no advantages in the top-1 rankings. This indicates that the DAEM algorithm used for training the core model worked better due to the reduction in the sparseness of data by grouping U and M. The accuracies obtained with the users-pieces system, on the other hand, deteriorated. To create profiles of U, we should not use the content-based data of *virtual* representative pieces but of *real* pieces, M, because the sums in Eq. (13) operate over M. Equation (14), in contrast, does not need to operate over U. That is, all real pieces should be recovered in advance of adding all real users.

5 CONCLUSION

We presented an incremental training method and its application to scalability enhancement for our model-based hybrid music recommender system that uses collaborative and content-based data. The incremental training method efficiently updates the partial parameters of the probabilistic model with theoretical proofs according to the growth of the observed data. The scalability enhancement method, which can speed up model training a hundred fold, has the potential to improve the accuracy of recommendations. That is, we found a breakthrough to overcome the trade-off, i.e., accuracy v.s. efficiency and scalability, which has been considered to be unavoidable. In the future, we plan to test our system with a larger database.

6 REFERENCES

[1] A. Uitdenbogerd and R. van Schyndel, "A Review of Factors Affecting Music Recommender Success," *ISMIR*, 2002, pp. 204–208.

[2] G. Linden and B. Smith, "Amazon.com Recommendations: Item-to-Item Collaborative Filtering," *IEEE Internet Computing*, pp. 76-80, Vol. 7, No. 1, pp. 76–80, 2003.

[3] C. Anderson, "The Long Tail: Why the Future of Business Is Selling Less of More," Hyperion, 2006.

[4] 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.

[5] B. Logan, "Music Recommendation from Song Sets," *ISMIR*, 2004, pp. 425–428.

[6] K. Yoshii, M. Goto, K. Komatani, T. Ogata, and H. G. Okuno., "Hybrid Collaborative and Content-based Music Recommendation Using Probabilistic Model with Latent User Preferences," *ISMIR*, 2006, pp. 296–301.

[7] 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.

[8] N. Ueda and R. Nakano, "Deterministic Annealing EM Algorithm," *Neural Net.*, Vol. 11, No. 2, pp. 271–282, 1998.

[9] L. Zhang, C. Li, Y. Xu, and B. Shi, "An Efficient Solution to Factor Drifting Problem in the pLSA Model," *CIT*, 2005, pp. 175–181.

[10] L. Ungar and D. Foster, "Clustering Methods for Collaborative Filtering," *Workshop on Recommendation Systems*, 1998.