Content-based Music Filtering System
with Editable User Profile

Yoshinori Hijikata   Kazuhiro Iwahama   Kazuki Takegawa   Shogo Nishida
Graduate School of Engineering Science
Osaka University, Osaka 560-8531, JAPAN

{hijikata, iwahama, tagegawa, nishida}@nishilab.sys.es.osaka-u.ac.jp

ABSTRACT

Information filtering systems, which recommend appropriate information to users from enormous amount of information, are becoming popular. One method of information filtering is content-based filtering that compares a user profile with a content model. Many systems using content-based filtering deal with text data, and few systems deal with music data. We propose a content-based filtering system for music data by using a decision tree. Compared with other filtering methods, a decision tree can eliminate noise features, which are not related to the user's preference, and can allow the user to edit the learned user profile. We conduct an experiment by using real music data and users to validate the effectiveness of our system compared with other filtering methods.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

General Terms

Documentation, Experimentation, Human Factors

Keywords

content-based filtering, music recommendation, user profile, decision tree, customization

1. INTRODUCTION

Recently not only text or picture data but also music data are increasing on the Internet. Major portal sites such as amazon.com and Yahoo! are beginning to distribute commercial music data. People are facing a problem of information overload also for music data. Information filtering systems (or recommender systems) have been studied for solving the problem of information overload [1][2]. Information filtering systems provide users with appropriate information that meets their preferences or interests from enormous amount of information. There are two methods for realizing information filtering systems. One is content-based filtering, and the other is collaborative filtering [3][4]. Because collaborative filtering does not see the content of information, many collaborative filtering systems deal with various kinds of item such as music and movie. On the contrary, because content-based filtering has difficulty in dealing with multimedia data, most researchers still target text information.

In this research, we develop a content-based filtering system of music. Before explaining our method, we explain a general idea of filtering music data. Firstly, the system extracts feature parameters from music data and creates a model of music called *content model* using the extracted parameters. The *user profile*, which represents the user's preference, is also created by using these extracted parameters. The system decides which item to recommend to the user by comparing the content model with the user profile. Here several methods can be available in how to build a content model and a user profile or in how to compare them. To our knowledge, there are two methods [5, 6] in the field of music filtering. One method represents a content model and a user profile as vectors of feature parameters and compares them by vector distance [5]. The other method represents a user profile as weights of feature parameters to the user's preference and uses them as coefficients of a linear evaluation function with a threshold for deciding whether or not the system should recommend the music data [6]. The former method deals with all types of feature parameter equally. However, important feature parameters to the user's preference may differ in individuals. The cluster, which is the music data set the user likes or that the user dislikes, may spread along the axis whose parameter is not related to the user's preference in multidimensional space. On the other hand, the latter method can avoid the noise effect caused by unimportant features because the weight of each feature is set by the correlation coefficient with the user's preference. However this method can divide music data only by one hyperplane and cannot represent the user's preference which has several clusters in multidimensional space.

Sometimes the availability of customizing the user profile is considered important in the field of information filtering [7]. This is because we cannot avoid errors in the user profile if it is generated by machine learning, and we have to modify them. We propose a content-based music filtering system using a decision tree[9]. Constructing a decision tree for each user as his user profile realizes the classification

SAC'06 April 23-27, 2006, Dijon, France
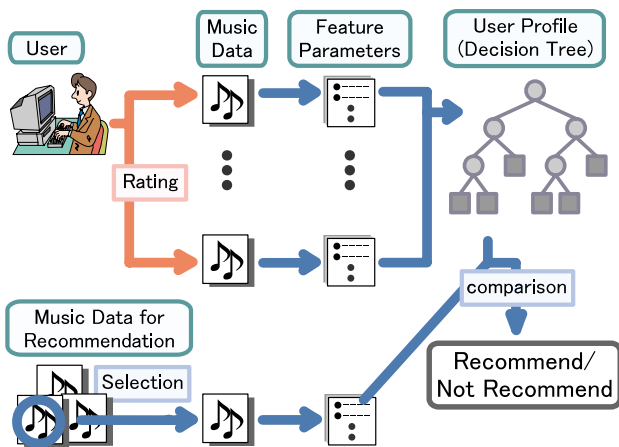
(a) Overview of our filtering system

Figure 1: Overview of our filtering system.

which uses only important feature parameters to his preference. Moreover, because classification rules in a decision tree are readable by human, the user can edit the error part of his user profile. Our system targets noted polyphonic music data. The format of music data in our system is MIDI. We also select popular music (pop music) as the target music genre. The reason why we limit music genre is that the instrumental part and construction of music are largely different in genres. The overview of our filtering mechanism is depicted in Figure 1. The user rates music data from the viewpoint whether or not it fits the user's preference. The system extracts the feature parameters (the pair of attribute and attribute value) from the rated data and constructs a user profile (actually a decision tree). Then it extracts the feature parameters from unrated music data and compares them with the user profile. It decides whether to recommend or not recommend from the comparison result. We conducted an experiment by using real music data and users and compared our method with several conventional methods which are based on vector distance and with SVM (support vector machine) which is the latest machine learning algorithm.

The organization of the remainder of this paper is as follows. Firstly we describe the related works. Then we explain feature parameters and a filtering method in our system. After that we conduct an experiment to see the effectiveness of our method. Finally we offer some conclusions.

2. RELATED WORKS

As related works to our research, we raise content-based filtering for text data, music information retrieval, automatic playlist generation, content-based music filtering and collaborative filtering.

Content-based filtering has been studied for text information [1][2]. The most popular method of content-based filtering is relevance feedback based on vector space model [10]. Modeling the user profile in this method relies on the occurrence probability of words. Words, which the user is originally interested in, leave as high-frequency elements in the word vector as the user searches and browses documents for long time. However there is no feature parameter in mu-

sic which occurs in some songs but does not occur in other songs like words. This needs to develop a filtering method suited for music data.

In the area of music and computer, many researchers work on music information retrieval[11, 12, 13] (not music filtering or music recommendation). The core of music information retrieval is to find occurrences of a small fragment of music in a larger body of music [14]. The basic techniques for pattern matching are edit distance [15, 16, 17], n-gram [18, 19, 20], dynamic programming (DP) based matching algorithm [21, 22, 23, 24, 25, 26], hidden Markov model [27, 28, 29] and inverted file [30]. Although many music information retrieval systems have been developed, they just search similar music data to the small fragment of music and do not model the user's daily preference to music.

Some researchers work on the generation of playlists from the user's music collection. Pauws and Eggen's method [31] does a clustering in the user's music collection and provides the clusters as playlists to the user. The user judges each music data in the playlist and can remove the music data that do not meet his/her expectations. By using these judgments as teacher signals, this method uses an inductive learning algorithm based on decision trees to find important features that might explain the removal of music data. Weight to the feature parameter is adjusted accordingly, and the method does the clustering again. Logan's method [32] extracts the audio spectrum and the type of instruments playing as feature parameters. A playlist is generated by selecting N closest music data in multi-dimensional space to a seed music data provided by a user. Although the latter method can be seemed as a recommendation, it does not model a user's daily preference to music.

As we explained in Section 1, Chen[5] and Kurose[6] implemented content-based filtering systems for music data. However the recommendation by Chen's system is influenced by feature parameters which are not related to the user's preference. Kurose's system cannot achieve complex classification. Our method builds a decision tree as a user profile for each user and decides which music data to recommend by only using important features to the user's preference.

Collaborative filtering [33, 1] is usually used for music recommendation. It achieves high precision in recommending music items. Because of the difficulties in extracting features from music data, it is difficult to achieve the same performance for music data by content-based filtering. However there are serious problems called *sparsity* and *first-rater* in collaborative filtering [8]. It cannot achieve good performance when there are not so many users who have rated a required number of items or it cannot recommend new items which no user has rated yet. We think that both types of filtering are indispensable for music recommendation.

3. FEATURE PARAMETER

We use feature parameters of music which are widely used in existing music information filtering/retrieval systems or which have been seen as common in music theory. They also should have variety among songs in the genre of pop music. We surveyed feature parameters of music to find such parameters. Firstly we listed popular feature parameters from previous researches. Then we implemented programs to extract those feature parameters and saw their distribution in real pop music data. Finally, we selected feature parameters for our filtering system based on the distribution.

Table 1: Survey result about feature parameter.

	1. meter	2. tonality	3. average tempo	4. rhythm	5. key
whole music	diff.	diff.	0.301	0.591	diff.

	6. ratio of major chord	7. ratio of minor chord	8. ratio of sus4 chord
whole music	0.226	0.319	1.611

	9. tone	10. average pitch	11. average diff. of pitch	12. average duration	13. average diff. of duration
melody CH	diff.	0.092	0.195	0.315	0.723
chord CH	no diff.	0.068	0.759	0.613	0.915
bass CH	no diff.	0.134	0.528	0.432	0.663
drum CH	no diff.	N/A	N/A	0.831	2.164

☐ Parameters which are not used in our system

3.1 Selection of Feature Parameter

Feature parameters in music can be represented in time-series patterns or in pairs of attribute and attribute value in one music data. We adopted the latter format because information filtering systems should estimate what type of music that the user generally prefers from the set of music data rated by the user. We calculate feature parameters in whole of the music, melody channel (after here "CH"), chord CH, bass CH and drum CH. This is because our research targets pop music, which mainly consists of the above four parts.

From feature parameters used in the existing systems, we selected tempo, tonality, rhythm, tone, meter, pitch, the difference of pith and the difference of duration. They are used by many researchers such as Chen[5], Kurose[6], Ikezoe[35], Satou[36], Kumamoto[37], Clausen[30] and Doraisamy[20]. Note that the average difference of pitch is calculated from the pitches between a note and its next note and that the average difference of duration is calculated from the durations between a note and its next note. From feature parameters which are common in music theory, we use the percentage of chord type and key. Table 1 shows the feature parameters we will survey.

3.2 Feature Extraction and CH Estimation

Out of the selected feature parameters in Table 1, Parameter 1 through Parameter 3 and Parameter 9 through Parameter 11 are described explicitly in MIDI data. Parameter 12 and Parameter 13 can be obtained by calculating delta time between "note on" event and "note off" event in MIDI data. Parameter 4 can be extracted by Ikezoe's method [35]. It is necessary to estimate chord for extracting Parameter 5 to Parameter 8. In our research, we estimate chord every half bar. This is because the change of chord can happen in half bar at the shortest. Our method collects all notes in half bar and calculates the root chord from the collected notes. It estimates chord from the relationship between the root chord and the other notes.

We also need to estimate CH. The estimating methods of drum CH, bass Ch and chord CH are simple and their precision is nearly 100%. However it needs some devices for estimating melody CH with high precision. We surveyed the characteristics of melody CH and found that the number of notes which are played simultaneously, the average difference of pitch and CH number are remarkable in melody CH.

Our estimating method gives a score to each CH from the viewpoint of the above three characteristics and selects the CH which have the highest score as melody CH. We conducted an experiment with 50 MIDI data to see whether or not this method estimates melody CH correctly, and found that it can estimate melody CH with 94%.

3.3 Result of Survey

We surveyed the distributions of the above feature parameters of 30 MIDI data. Table 1 depicts the results. The actual calculation of distribution is done by dividing the standard deviation of each feature parameter by its average because all parameters are not less than 0. Note that "N/A" in Table 1 represents that the feature parameter does not exist in the CH. We eliminated feature parameters which have hairbreadth dispersion. Such parameters are tone (in chord CH, bass CH and drum CH) and average pitch (they are shown in gray elements in Table 1). From this result, we use the left feature parameters in our filtering system.

4. FILTERING METHOD

In our method, one user's profile is presented as a decision tree. We use C4.5 [9] as an algorithm of decision tree because it can deal with both continuous value and categorical value. The user profile is constructed as follows. The user rates some music data presented by the system on a scale of one to three: "like", "neutral" and "dislike". Although rating itself can rely on other discrete scales like 5-scale or 7-scale, the decision tree cannot deal with grades. Therefore, we represent users' preferences with no grade. The system conducts learning from the user's ratings and makes a decision tree. A node of the decision tree has a condition for its feature parameter. A leaf node of the decision tree has a class that expresses "like", "neutral" and "dislike". The class with highest frequency among examples in the leaf node is assigned to the leaf node. The content model is made for each music data. It consists of feature parameters extracted from music data by the system. When the system conducts filtering, it searches the decision tree with the content model of the target music data. When the system reaches a leaf node and its class is "like", the system recommends the music data to the user. The user listens to the recommended music data and rates it. The system learns the decision tree again based on the user's updated ratings.
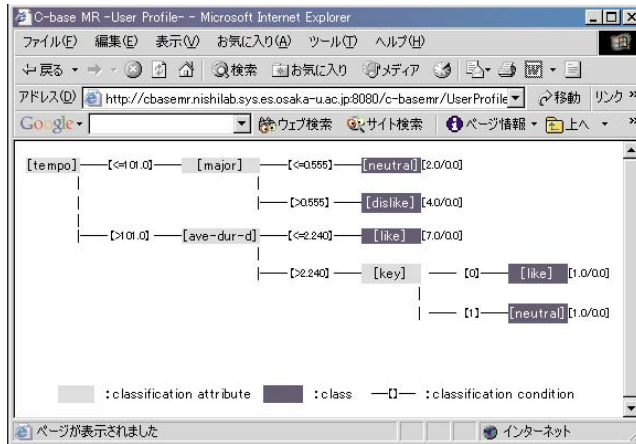
5. PROTOTYPE SYSTEM

We built a prototype system in Java and Java servlet whose screenshots are shown in Figure 2. After the user rates music data (Figure 2-(a)), the user profile is constructed (Figure 2-(c)). The system recommends new music data suitable to the user by using the user profile (Figure 2-(b)). The decision tree in Figure 2-(c) is displayed by clicking the "user profile" button in the menu bar in Figure 2-(a) and 2-(b). An anchor to the page for editing the user profile (after here "user profile editing page") (Figure 2-(d)) is embedded in a node and a link in the decision tree. In the user profile editing page, when an internal node is selected, the user can change the type of feature parameter and its value and also can change the internal node to a leaf node. When a leaf node is selected, the user can change the class and also can change the leaf node to the internal node. When a link is selected, the user can change the value to the feature pa-
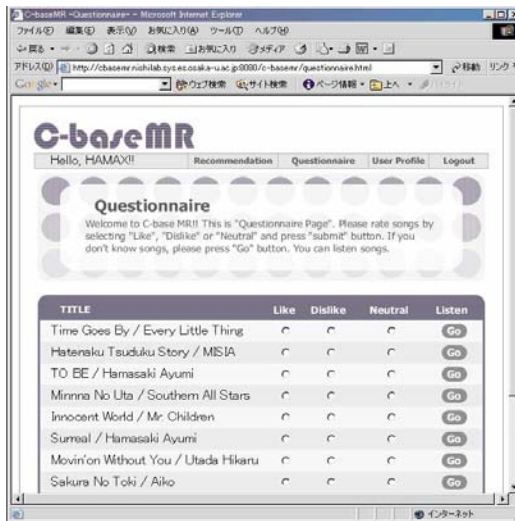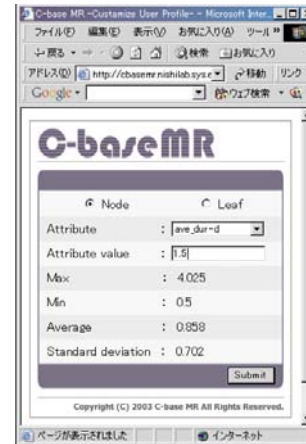
(a) Questionnaire page
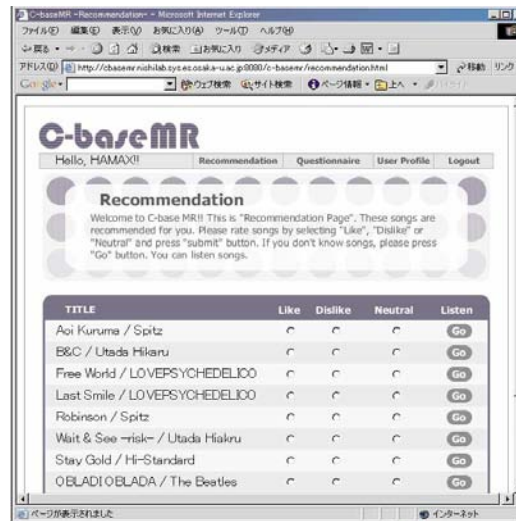
(b) Recommendation page

(c) User profile page

(d) User profile editing page

Figure 2: Screenshot of prototype system.

rameter. This page displays the maximum value, minimum value, average and standard deviation of the selected feature parameter so that the user can determine the threshold of the feature parameter.

6. EVALUATION

We conducted an experiment with real music data and users to compare our method with other filtering methods from the viewpoint of precision, recall and their improvement rate (the ratio of the evaluation parameter of the proposed method in relation to that of the other method). Firstly we compared with random recommendation. This is to see how much our method improves the recommendation quality compared to a method with no validity in recommendation and saw how many examples are required to learn a user profile at the very least. After that, we compared with conventional machine learning algorithms using vector distance: K-means method (after here "K-means"), tree clustering and k-nearest neighbor method (after here "k-NN") and also with the latest machine learning algorithm: sup-

port vector machine 'after here "SVM"). Finally, we saw the effectiveness of editing the user profile.

6.1 Experimental Method

We used 200 MIDI data which are (1) music data in RWC music database[39], (2) original music data on the Internet which we gained permission from the authors to use in the experiment and (3) music data whose copyrights are in JAS-RAC[40] and which we gained permission from the author of the MIDI data and also from JASRAC to use in the experiment. All the music data are the ones which the users in this experiment have never listened to. This is because not only the content of music but also other features such as the artist's name or gender may affect the user's rating. We asked 10 users to listen to 200 music data and rate all data. The distribution of each user's rating value is shown in Table 2. We divided 200 data into 100 learning data and 100 test data. After that, we also asked the users to describe their preference in text. The user should write down his preference by using music features if he can. If he cannot

1053

| User A	I like music with slow tempo and lively melody. I dislike music with fast tempo, rock music (with many low-pitch parts) and punk music. I gave "neutral" value to the music data other than the aboves.
User B	I like music with regularly beaten drums and with fast tempo. I like music with regularly repeated bass. I dislike normal music with medium tempo. I gave "neutral" value to the music data other than the aboves.
User C	I like bright and cute music. I dislike fast or dark music and music with low sound. I rated other music "neutral".

(a) Answers to the questionnaire
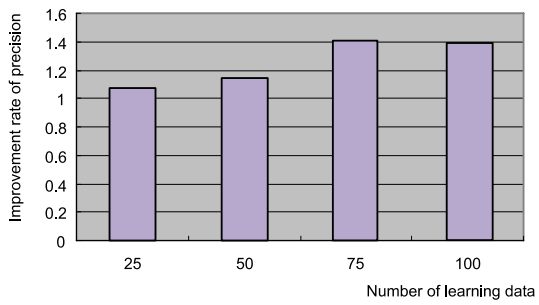
(b) Comparison with random recommendation
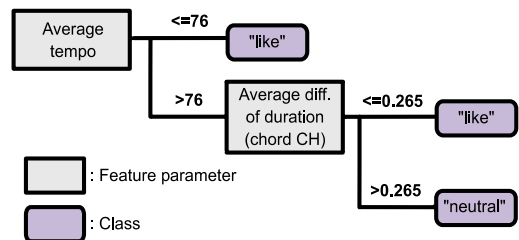
(c) Decision tree of User A (50 data for learning)
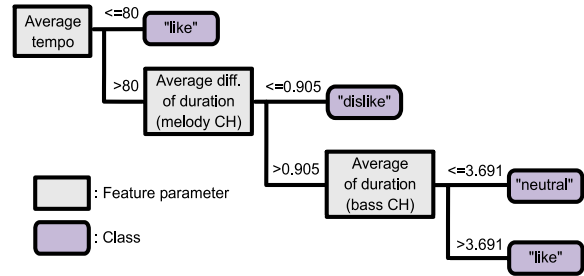
(d) Decision tree of User A (75 data for learning)
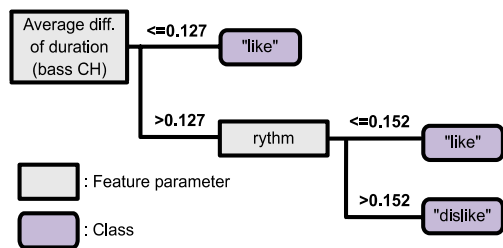
(e) Decision tree of User B (75 data for learning)

Figure 3: Result of experiment.

Table 2: Distribution of users' rating values

User	"like"	"dislike"	"neutral"
User A	87	53	60
User B	54	51	95
User C	72	74	54
User D	69	91	40
User E	91	29	80
User F	109	45	46
User G	71	43	86
User H	79	12	109
User I	109	43	48
User J	76	15	109

express in music features, he can write his preference however he likes. Figure 3-(a) shows examples of their answers [1]. This is for examining the result of recommendation in detail. Finally we asked the same 10 users to see their user profile and edit it from the user profile editing page. If the user feels that he cannot edit his user profile appropriately, he does not have to edit it.

6.2 Comparison with Random Recommendation

We built four decision trees by changing the number of learning data as 25, 50, 75 and 100 and compared them with random recommendation. Figure 3-(b) shows the average improvement rate of precision of 10 users (the improvement rate of recall almost equals to this). When the number of learning data is small, there is little difference between the proposed method and random recommendation (the improvement rate is about 1.1). This is because the system cannot learn the decision tree well enough to achieve good performance. Figure 3-(c) shows a user's (User A in Figure 3-(a)) decision tree in the case that the number of learning data is 50. From his preference written by himself in Figure 3-(a), we can see that he prefers music with slow tempo and lively melody. In Figure 3-(c), although the first branching parameter (feature parameter in the root node) is tempo, the second branching parameter is the average difference of duration in the chord CH. This means that this decision tree represents the user's evaluation standard which the tempo is slow, but does not represent the user's evaluation standard which the melody is lively.

When the number of learning data is 75 and over, the proposed method becomes better than random recommendation (the improvement rate is about 1.4). Figure 3-(d) shows User A's decision tree in the case that the number
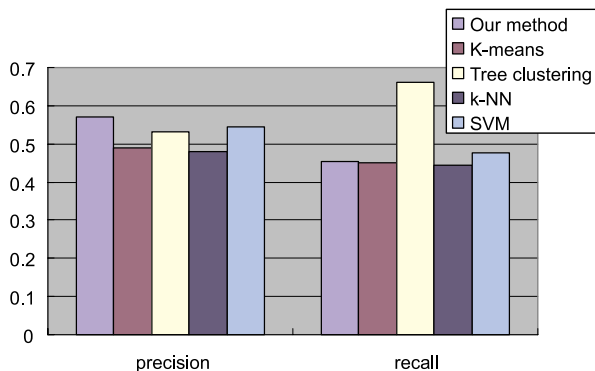
Figure 4: Result of comparing with other machine learning algorithms.

Table 3: Number of recommended items by four machine learning algorithms.

| | User | | | | | | | | | |
	A	B	C	D	E	F	G	H	I	J
Our method	27	7	61	23	43	81	7	5	87	11
K-means	35	3	47	52	46	85	3	11	100	13
Tree clustering	96	0	3	0	95	85	5	0	99	0
k-NN	28	6	22	28	42	62	24	33	72	46
SVM	30	11	36	31	46	53	39	30	58	28

of learning data is 75. The first branching parameter is tempo and the second branching parameter is the average difference of duration in melody CH not in chord CH. This means that this decision tree represents also the user's evaluation standard which the melody is lively. We conducted one-sided t-test in the case of 75 and 100 learning data and found a significant difference in precision between the proposed method and random recommendation at the 5% level of significance. From this, we can say that at least 75 data are required for learning a user profile.

6.3 Comparison with Other Machine Learning Algorithms

We compared our method with K-means, tree clustering, k-NN and SVM using 75 learning data. The number of clusters in K-means is the number of leaf nodes in the decision tree. The number of neighbors in k-NN is ten. Figure 4 shows the results. We can see that the precision of our method is better than that of K-means and k-NN under the condition of the almost same recall. From this result, we can say that our method narrows down the music data that the user will like better than K-means and k-NN.

To understand the reason that the precision of our method becomes good, we checked the user's preference written in text in Figure 3-(a). We can see that some users tend to focus on a few features to rate music data like "I like music with fast tempo." or "I dislike music with many low pitch parts." Figure 3-(e) depicts User B's decision tree. From Figure 3-(d) and 3-(e), we see that the decision tree uses only the important features which are related to the user's preference. From Figure 3-(a), we can see that feature parameters which the user thinks important to his preference differ by users. For example, User A focuses on tempo and melody, User B focuses on drum and bass. From Figure 3-

Table 4: Diversity of users' preference and quality of reccommendation.

| | Decision tree | | K-means | |
	Group with limited preference	Group with wide preference	Group with limited preference	Group with wide preference
Precision	0.578	0.554	0.496	0.491
Recall	0.552	0.306	0.563	0.285

(d) and 3-(e), feature parameters for branching in a decision tree are different by users. If important feature parameters are fixed in advance for all users, K-means and k-NN can achieve good performance by using only those feature parameters. However, we think that such assumption is not realistic.

The result of tree clustering is that four users out of ten users could not receive any music data as recommendation (see Table 3). This is due to the clustering mechanism of tree clustering. Tree clustering conducts a bottom-up grouping, which means it expands a cluster from two examples. Upper-level nodes tend to branch a cluster which includes a few examples and exists far from other clusters. Because the system cuts off the lower branches, it may produce $n - 1$ clusters with a few data and 1 cluster with many data in an extreme case. The recall for the users who receive recommendation from tree clustering is high (Figure 4). This is because the most frequent class in the biggest cluster becomes "like". As seen in Table 3, when the user receives recommendation, he receives many music data. We can say that it is difficult to use tree clustering for music filtering systems because its recommendation depends on the class distribution in the user's ratings.

The precision of SVM is slightly less than that of our method. The recall of SVM is slightly larger than that of our method. These two methods' F-values, which is the average of precision and recall, are almost same. Although SVM also uses the feature values which are not related to the user's preference, its powerful classification ability surpasses the above shortcoming. But the classification mechanism of SVM is not understandable for human. Under the same recommendation performance, we think decision tree is better because it allows users to edit the user profile for getting a better recommendation.

6.4 Diversity of user's preference

There is a possibility that the precision of recommendation may change between the case that the user likes only one type of music and the case that the user like various kinds of music. We divided ten users into a group whose preference is limited to one type of music (after here "group with limited preference") and a group whose preference is on many types of music (after here "group with wide preference") from the result of questionnaire. We examined the precision and recall in each group. Actually, we found that User A, C, D, E, I and J is in the group with limited preference and User B, F, G and H is in the group with wide preference. Table 4 shows the precision and recall of the decision tree and K-means in each group.

In both group, the precision of the decision tree is better than that of K-means under the condition of almost the

Table 5: Change of precision when editing user profile.

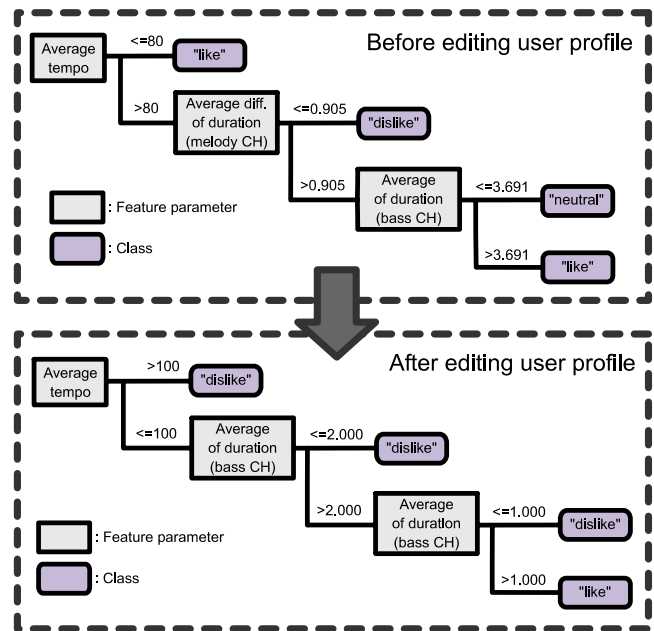| User	Editing user profile?	Precision (before)	Precision (after)
A	yes	0.851	0.956
B	no	---	---
C	yes	0.426	0.444
D	yes	0.522	0.500
E	no	---	---
F	yes	0.617	0.700
G	no	---	---
H	no	---	---
I	yes	0.426	0.596
J	yes	0.545	0.632

same recall. However we do not see a big improvement especially in the group with wide preference. This is because not only the decision tree but also K-means creates several clusters and furthermore we decided on the number of seeds in K-means as the number of leaf nodes in the decision tree. Although we could not find an especial improvement when the user's precision is on many kinds of music, we still found that decision tree provides a steady recommendation in both the cases that the user's preference is limited and the user's preference is diversified.

6.5 Edit of User Profile

We compared the precision before the edit of user profile and that after the edit of user profile. Table 5 shows the result. Out of 10 users, six users edited their user profiles. Out of the six users, the precisions of five users improved. Figure 5 depicts User A's editing. From the User A's preference written by himself in Figure 3-(a), User A dislikes music with fast tempo, rock music and punk music. In his editing, he took a strategy to exclude the type of music which he dislikes. He edited his user profile by using tempo as a first branching attribute and set the threshold for excluding the type of music which he dislikes. Then he used duration in bass CH which is affected by rock music and punk music as a second branching attribute. The result is that the user profile has become to represent the user's preference better and the precision has improved. However one user's precision before the editing is higher than that after the editing; further more, four users did not edit the user profile. One reason we think is that some users do not understand deeply the type of music they like. The other reason is that if users do not have the knowledge about music to some extent, they cannot decide which feature parameter to focus to change the user profile. From the above results and discussions, although it is not necessarily easy for a user to edit his user profile and the precision may decline when he edits it poorly, the editing function of user profile provides a chance to get better recommendation.

7. CONCLUSIONS

In this paper, we proposed a content-based music filtering method using a decision tree and implemented a prototype system in Java and Java servlet. Compared with conventional methods based on vector distance, we found that the

Figure 5: An example of editing user profile.

decision tree can represent well the feature of music data which the user likes and that our method narrows down the music data for better recommendation. Compared with the latest machine learning algorithm (SVM), the decision tree performs almost the same quality of recommendation. Furthermore a decision tree is readable for users. We found that when the user knows well what type of music he likes and has the knowledge about music to some extent, he can edit his user profile and he receives better recommendation with high precision. As a future work, we will work on helping users who does not have so much knowledge about music to change his user profile.

8. REFERENCES

[1] Resnick, P. and Varian, H.R.: Recommender Systems, *Comm. of the ACM*, Vol. 40, No. 3, pp. 56–89 (1997).

[2] Loeb, S. and Terry, D.: Information Filtering, *Comm. of the ACM*, Vol. 35, No. 12, pp. 26–81 (1992).

[3] Ramakrishnan, N.: PIPE: Web Personalization by Partial Evaluation, *IEEE Internet Computing*, Vol. 4, No. 6, pp. 21–31 (2000).

[4] Riecken, D.: Personalized Views of Personalization, *Comm. of the ACM*, Vol. 43, No. 8, pp. 26–158 (2000).

[5] Chen, H. and Chen, A.L.P.: A music recommendation system based on music data grouping and user interests, In *Proc. of the tenth International Conference on Information and Knowledge Management (CIKM'2001)*, pp.231-238, (2001).

[6] Kurose, T., Kajikawa, Y. and Nomura, Y.: Music Recommendation System Using KANSEI Informations, In *Proc. of the DEWS2003*, http://www.ieice.org/iss/de/DEWS/proc/2003/papers/8-P/8-P-06.pdf (2003).

[7] Vassileva, J.: A Practical Architecture for User Modeling in a Hypermedia-Based Information Systems, In *Proc. of 4th International Conference on*

User Modeling (UM'1994), pp. 115–120 (1994).

[8] Umeki, H.: Network Community Formation and Support Technologies, *Journal of the JSAI*, Vol. 14, No. 6, pp. 943–950 (1999).

[9] Quinlan, J.R.: C4.5 Programs for Machine Learning, Morgan Kaufmann (1993).

[10] Hijikata, Y.: User Profiling Technique for Information Recommendation and Information Filtering, *Journal of the JSAI*, Vol. 19, No. 3, pp. 365–372 (2004).

[11] ISMIR: The International Conferences on Music Information Retrieval, http://www.ismir.net/

[12] Byrd, D. and Crawford, T.: Problems of Music Information Retrieval in the Real World, *Information Processing and Management*, Vol.38, pp.249-272 (2002).

[13] Downie, J.S.: Music Information Retirval, *Annual Review of Information Science and Technology*, Vol.37, pp.295-340 (2003).

[14] Lubiw, A. and Tanur, L.: Pattern Matching in Polyphonic Music as a Weighted Geometric Translation Problem, In *Fifth International Conference on Music Information Retrieval (ISMIR2004)* (2004).

[15] Mongeau, M. and Sankoff, D.: Comparison of Musical Sequences, *Computers and the Humanities*, Vol.24, pp.161-175 (1990).

[16] McNab, R.J., et al.: Towards the Digital Music Library: Tune Retrieval from Acoustic Input, In *Proc. of ACM Digital Libraries* (1996).

[17] Lemstrom, K.: String Matching Techniques for Music Retrieval, PhD thesis, University of Helsinki, Department of Computer Science, Report A-2000-4 (2000).

[18] Downie, J.S.: Evaluating a Simple Approach to Musical Information Retrieval: Conceiving Melodic N-grams as Text, PhD thesis, University of Western Ontario (1999).

[19] Doraisamy, S. and Ruger, S.: Robust Polyphonic Music Retrieval with N-grams, *Journal of Intelligent Information Systems*, Vol.21, No.1, pp.53-70 (2003).

[20] Doraisamy, S. and Ruger, S.: A Polyphonic Music Retrieval System using N-grams, In *Fifth International Conference on Music Information Retrieval (ISMIR2004)* (2004).

[21] Hewlett, W. and Selfridge-Field, E.: Melodic Similarit: Concepts, Procedures, and Applications, *Computing in Musicology*, Vol.11, Cambridge: MIT Press (1998).

[22] Dannenberg, R.B.: An On-line Algorithm for Real-Time Accompaniment, In *Proc. of the 1984 International Computer Music Conference*, pp.193-198 (1985).

[23] Ghias, A., et al.: Query by Humming - Musical Information Retrieval in an Audio Database, In *Proc. of ACM Multimedia 95*, pp.231-236 (1995).

[24] Sonoda, T. and Muraoka, Y.: A WWW-based Melody Retrieval System -An Indexing Method for a Large Database-, In *Proc. of International Computer Music Conference (ICMC 2000)*, pp. 170-173 (2000).

[25] Roger Jang, J.-S., Chen, J.-C. and Kao, M.-Y.: MIRACLE: A Music Information Retrieval System with Clustered Computing Engines, In *Proc. of*

Second International Conference on Music Information Retrieval (ISMIR2001) (2001).

[26] Ito, A., et al.: Comparison of Features for DP-Matching based Query-by-Humming System, In *Proc. of Fifth International Conference on Music Information Retrieval (ISMIR2004)* (2004).

[27] Meek, C. and Birmingham, W.P.: Johnny Can't Sing: A Comprehensive Error Model for Sung Music Queries, In *Proc. of Third International Conference on Music Information Retrieval (ISMIR2002)*, pp.124-132 (2002).

[28] Birmingham, W., et al.: Managing a Personal Music Library, *Dr. Dobbs Journal*, Vol.28, No.9 (2003).

[29] Shifrin, J. and Birmingham, W.: Effectiveness of HMM-based Retrieval on Large Databases, In *Proc. of Fifth International Conference on Music Information Retrieval (ISMIR2004)* (2004).

[30] Clausen, M., et al.: PROMS: A Web-based Tool for Searching in Polyphinic Music, In *Proc. of First International Conference on Music Information Retrieval (ISMIR2000)* (2000).

[31] Pauws, S. and Eggen, B.: PATS: Realization and User Evaluation of an Automatic Playlist Generator, In *Proc. of The 3rd International Conference on Music Information Retrieval (ISMIR 2002)*, pp. 222-230 (2002).

[32] Logan, B.: Content-based Playlist Generation: Exploratory Exmeriments, In *Proc. of The 3rd International Conference on Music Information Retrieval (ISMIR 2002)*, pp. 295-296 (2002).

[33] Resnick, P., et al.: GroupLens : An Open Architecture for Collaborative Filtering of Netnews, In *Proc. of the Conference on Computer Supported Cooperative Work (CSCW'94)*, pp.175-186 (1994).

[34] Typke, R., Wiering, F., Veltkamp, R.C.: A Search Method for Notated Polyphonic Music with Pitch and Tempo Fluctuations, In *Proc. of the Fifth International Conference on Music Information Retrieval (ISMIR 2004)* (2004).

[35] Ikezoe, T., Kajikawa, Y. and Nomura, Y.: Music Database Retrieval System with Sensitivity Words Using Music Sensitivity Space, *IPSJ Journal*, Vol. 42, No. 12, pp. 3201–3212 (2001).

[36] Satou, A., Ogawa, J., Horino, J. and Kitagami, H.: A Discussion about the Realization of Impression-based Retrieval System for Music Collection, *IPSJ SIGNotes MUSic and computer*, No. 39, pp. 51–56 (2001).

[37] Kumamoto, T. and Ohta, K.: Automatically and Numerically Expressing Impressions of a Music Piece for Music-Retrieval based on User's Impressions, *IPSJ SIGNotes DataBase System*, No. 127, pp. 89–96 (2002).

[38] Berry, M.J.A. and Linoff, G.: Data Mining Techniques, For Marketing, Sales, and Customer Support, Wiley Computing Publishing (1997).

[39] RWC Music Database, http://staff.aist.go.jp/m.goto/RWC-MDB/index-j.html

[40] Japanese Society for Rights of Authors, Composers and Publishers (JASRAC), http://www.jasrac.or.jp/ejhp/index.htm