Journal of Intelligent Information Systems, 24:2/3, 113–132, 2005

A Music Recommendation System Based on Music and User Grouping*

HUNG-CHEN CHEN
Department of Computer Science, National Tsing Hua University, Hsinchu, Taiwan, Republic of China

ARBEE L.P. CHEN alpchen@cs.nccu.edu.tw
Department of Computer Science, National Chengchi University, Taipei, Taiwan, Republic of China

Received May 14, 2004; Revised October 15, 2004; Accepted November 8, 2004

Abstract. In this paper, we present a music recommendation system, which provides a personalized service of music recommendation. The polyphonic music objects of MIDI format are first analyzed for deriving information for music grouping. For this purpose, the representative track of each polyphonic music object is first determined, and then six features are extracted from this track for proper music grouping. Moreover, the user access histories are analyzed to derive the profiles of user interests and behaviors for user grouping. The content-based, collaborative, and statistics-based recommendation methods are proposed based on the favorite degrees of the users to the music groups, and the user groups they belong to. A series of experiments are carried out to show that our approach performs well.

Keywords: access histories, music recommendation, perceptual properties, recommendation methods, user profiles

1. Introduction

Concerning a large number of items available on the Internet, the systems which provide the services for users to look for their favorite items are urgently needed. One of the most important services for the users to escape from this *information overloading* problem is the recommendation service. The recommendation service is to recommend items that users may be interested in based on users' predefined preferences or users' access histories. Various items have been considered in these recommendation systems, such as music objects (Sharadanand and Maes, 1995; Kuo and Shan, 2002), books (Mooney and Roy, 2000), movies (Movielen, 2003), news (Billsus and Pazzani, 1999), and webpages (Balabanovic and Shoham, 1997; Joachims et al., 1997; Lieberman, 1995; Rucker and Polanco, 1997; Wu et al., 2001). In this section, we first introduce two major approaches for general recommendation systems, i.e., the *content-based filtering* approach and the *collaborative filtering* approach. Then, we introduce the recommendation systems which are

*Part of the contents of this paper has been published in ACM proceedings of International Conference on Information and Knowledge Management, 2001 designed for recommending music objects and point out their limitations compared with our system.

In the content-based filtering approach, the *user profiles* are first formed by extracting features of the data items which have been accessed in the past. Based on the user profiles, the system recommends only the data items that are highly relevant to the user profiles by computing the similarities between the features of the data items and the user profiles. Examples of such systems are NewsWeeder (Lang, 1995), Infofinder (Krulwich and Burkey, 1996), and News Dude (Billsus and Pazzani, 1999). In this approach, the representation of data items and the description of user preferences in profiles are key issues, which dominate the effectiveness of the recommendation (Cheung and Tian, 2004). However, recommendation systems adopting the content-based filtering approach can only recommend the data items in which the user has indicated his/her interest. Other potential interesting data items of the user cannot be explored in such recommendation systems if he/she has never accessed before.

Instead of computing the similarities between the features of the data items and the user profiles, the collaborative approach computes the similarities between the user profiles. Users of similar profiles are grouped together to share the information in their profiles. The main goal of the collaborative approach is to make the recommendation among the users in the same group. Examples of such systems are Ringo (Sharadanand and Maes, 1995) and Siteseer (Rucker and Polanco, 1997). Adopting the collaborative filtering approach, the system has a high possibility to recommend surprising data items by the nature of information sharing, which cannot be achieved by the content-based filtering approach.

Some systems use both contend-based and collaborative filtering approaches. For example, Tapestry (Goldberg et al., 1992) and GroupLens (Konstan et al., 1997) allow users to comment on Netnews and group users by computing the similarities of their ratings of newsgroups. In addition, for the process of recommendation, users have to describe the features of data items which they are interested in to specify their profiles. For the video data, the recommendation system is developed in Basu et al. (1998). The user interests are derived from the types, the actors, and the scenarios of videos that the users accessed in the past. The users are also required to specify the satisfactory degrees of the accessed videos. With respect to the videos, users who specify similar satisfactory degrees are grouped together for collaborative recommendation. Similarly, the Personalized Television system (Smith and Cotter, 2000) provides a personalized list of recommended programs. The FAB system (Balabanovic and Shoham, 1997) analyzes the accessed webpages to derive the user profiles and compares the user profiles to group users for collaborative recommendation. The OTS (Wu et al., 2001) employs the techniques of association rule mining (Agrawal and Srikant, 1994) to derive user interests and behaviors to be used as the user profiles. After classifying the user profiles into clusters, three kinds of recommendation methods are then provided using these clusters.

In this paper, we focus on the recommendation service for music objects. Due to the complex semantics contained in the music objects, the representation of music objects becomes critical for the recommendation service. Moreover, the method to automatically derive the user interests and behaviors for music objects is another concerned problem.

Our goal is to construct a complete music recommendation system, which overcomes the limitations of the previous systems.

The simplest service of music recommendation is accomplished by using the *keyword-based filtering* approach to notify users when appropriate music objects arrive. The mechanism for this notification service is described as follows. For an incoming music object, the corresponding description is manually attached to the music object, such as the music genre, title, and composer. The users are required to specify their preferences in music terms. The users' preferences will be compared with the descriptions of the music objects. Once matched, the system will send a notification of the matched music objects to the users. However, the task of manual description takes tremendous efforts when available music objects grow explosively. Therefore, this kind of music recommendation is impractical in the real world.

Ringo in Sharadanand and Maes (1995) is a pioneer collaborative music recommendation system. In Ringo, each user is requested to make ratings for some music objects. These ratings constitute the personal profile. Several algorithms are proposed to measure the similarity between two users' profiles. For collaborative recommendation, only the ratings of the users whose profiles are similar to the target user are considered. Whether a music object will be recommended is then based on the weighted average of the ratings considered.

On the contrary, a content-based personalized music filtering system is introduced in Kuo and Shan (2002). The system learns the user's preferences by mining the melody patterns from the music objects in the user's access history. Using these melody patterns, a melody preference classifier is then constructed for each user. An incoming music object will be recommended to the user if it is classified into the preferred class. In this system, only the pitch information is considered for feature extraction. Ignoring other information, e.g., duration and loudness, provided in the music objects limits the system to deal with other kinds of user preferences. For example, the user may prefer the music objects with slower tempo. However, the system in Kuo and Shan (2002) cannot actually reflect this preference for the users during recommendation. Moreover, due to the inherence of the content-based filtering approach, this system cannot provide any surprising recommendation results as in the collaborative approach.

In this paper, we propose an alternative way of music recommendation, which overcomes the limitations of the previous works. Instead of textual descriptions, we fully consider the perceptual properties of music objects, such as pitch, duration, and loudness, which can be directly extracted from the music objects. Based on the various features derived from these perceptual properties, music objects are then grouped automatically. For users, their interests and behaviors are derived from the access histories and recorded in user profiles for further user grouping. Based on the favorite degrees of the users to the music groups, and the user groups they belong to, three recommendation methods are proposed to satisfy different needs of the users. Both the perceptual properties of the music objects and the derivation of user interests and behaviors are considered in the proposed system. As a result, our recommendation system can be more satisfactory to users than the previous systems.

The rest of this paper is organized as follows. In Section 2, the music recommendation system is introduced with detailed explanations for track selector, feature extractor and classifier. In Section 3, three recommendation methods implemented in our system are presented. In Section 4, a series of experiments are performed with the experiment results to prove that our approach is feasible. Finally, a conclusion is given in Section 5.

2. Music recommendation system

The Music Recommendation System (MRS) is a website which provides the service of music recommendation based on music grouping and user grouping. The music objects in the database of MRS, as well as the incoming music objects, are candidates for music recommendation. As shown in figure 1, the system consists of seven function blocks, namely, the track selector, the feature extractor, the classifier, the profile manager, the recommendation module, the interface, and the database. When a new music object is inserted in the database of the MRS, it goes through the track selector and feature extractor. Using the features extracted from a music object, a dynamic classifier is designed for music grouping. Each incoming music object is properly assigned to certain music group by the classifier. The profile manager is implemented for the purpose of updating the access histories of the users. When the user accesses a music object from the list of music objects or the recommendation results, the profile manager records the information of the music object in the user's access history. Using the information recorded by the profile manager, three recommendation mechanisms, i.e., the content-based (CB) method, the collaborative (COL) method, and the statistics-based (STA) method, are designed for different users' needs.

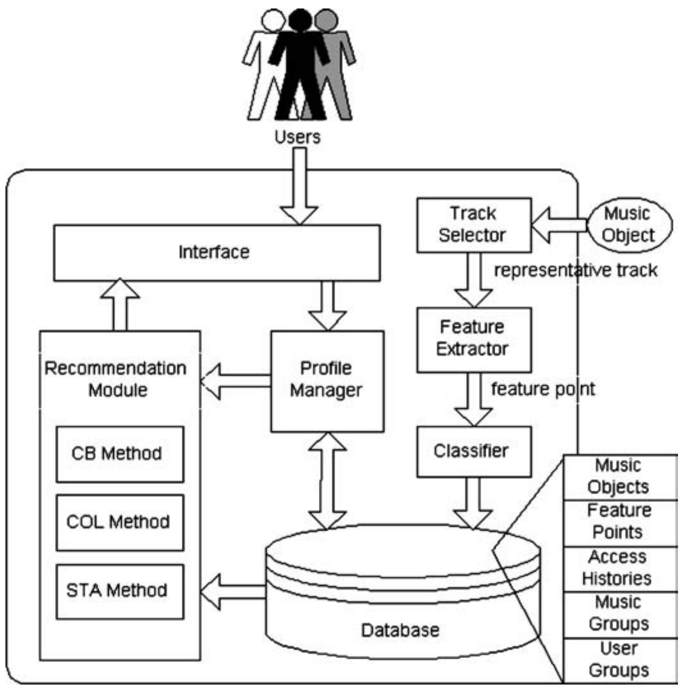

Figure 1. The system architecture of the MRS.

The three function blocks, i.e., the track selector, the feature extractor, and the classifier, will be described in the following subsections. The profile manager and the three recommendation methods will be presented in Section 3. In addition, the interface will be shown in Section 4.

2.1. *Track selector*

In the MRS, the music objects are of MIDI format. There are two kinds of music objects, i.e., monophonic music objects and polyphonic music objects. Usually, a polyphonic music object consists of several tracks, one for melody and the others for accompaniments. We observe that the track for melody contains much more distinct notes with different pitches than the tracks for accompaniment. In Uitdenbogerd and Zobel (1998), the method used to extract a melody from a MIDI file has been developed, which considers all tracks of the MIDI file and chooses the notes with the highest pitch for the melody. Unfortunately, this method may result in an extracted melody containing the notes which belong to the tracks of accompaniment when the melody has rests. Moreover, this method cannot handle the case that the melody does not at the highest voice part.

Different from the method used in Uitdenbogerd and Zobel (1998), we use a measure of *pitch density* to select a *representative track*. The representative track is regarded as the track which contains the melody of the corresponding music object. The pitch density of a track is defined as follows:

$$pitch\ density = \frac{NP}{AP} \tag{1}$$

where *NP* is the number of distinct pitches in the track and *AP* is the number of all distinct pitches in MIDI standard, i.e., 128.

The pitch densities of all tracks of the target music object are computed by Eq. (1). The track with the highest density is then selected as the representative track of a polyphonic music object.

2.2. *Feature extractor*

The purpose of the feature extractor is to extract features from the perceptual properties of the representative track. These extracted features are used to represent the music objects for later music grouping. The six features are described as follows. Other kinds of musical features extracted from perceptual properties are introduced in Mckay and Fujinaga (2004).

a. *Mean (MP) and standard deviation (SP) of the pitch values*
   From the representative track, we compute the mean and standard deviation of the pitches.
b. *Pitch density (PD)*
   The definition of pitch density has been given in Eq. (1).
c. *Pitch entropy (PE)*

The *pitch entropy PE*, derived from Sayood (2000), is defined as follows:

$$PE = -\sum_{j=1}^{NP} P_j \log P_j \tag{2}$$

where P_j is defined as follows:

$$P_j = \frac{N_j}{T} \tag{3}$$

where N_j is the total number of notes with the corresponding pitch in the representative track, T is the total number of notes in the representative track.
In Eq. (2), *PE* has a maximum value when each P_j is the same.

d. *Tempo degree (TD)*
The *tempo degree* is defined as a ratio of the number of fast measures to the number of measures in the representative track. A measure is a fast measure if the average note duration in the measure is shorter than one.

e. *Loudness (LD)*
The feature of *loudness* is defined as the average value of the *note velocities* which can be derived from MIDI files.

2.3. *Classifier*

After extracting features, each music object in the MRS is represented as a 6-tuple, (MP, SP, PD, PE, TD, LD). These features will be used to classify music objects into music groups. A music group is represented by a centroid and six thresholds, each for a feature. These thresholds are used to restrict the number of music groups, which can be decided by a method to be illustrated in Section 4.2.2. For easier illustration, we show an example of the classification using two features, PD and PE. For each incoming music object, there are two situations to consider when performing the classification. In the first situation, no music group exists in the database. Therefore, the new feature point of the incoming music object will be the centroid of a new group. In the second situation, some music groups exist. The distances between the new feature point and each group centroid are computed. The group with the minimum distance is selected for consideration. There are two cases to consider. In case 1, if the new feature point falls into the rectangular area of the selected group formed by the centroid and the two thresholds δ_{PD} and δ_{PE} (for features PD and PE, respectively), the new feature point will be assigned to the selected group, as shown in figure 2. The centroid of the group is recomputed accordingly. In case 2, if the new feature point does not fall into the area of any group, it will become the centroid of a new group, as shown in figure 3. Note that the classifier using multiple features may produce too many groups. We use a maximum number of groups K to limit the total number of music groups. If more than K groups are created, we enlarge the thresholds and re-classify the music objects such that the number of groups become less than or

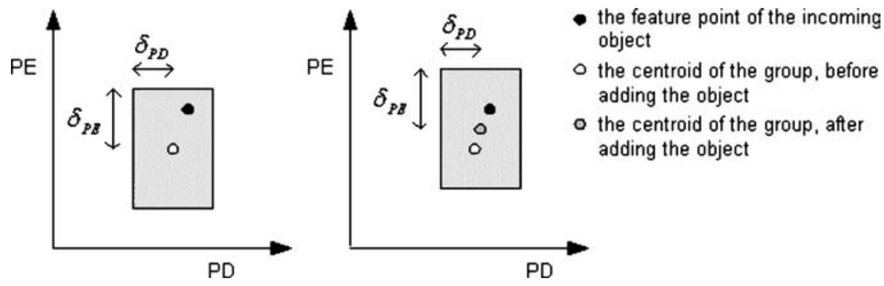

Figure 2. The new feature point falls into the area of a group (case 1).

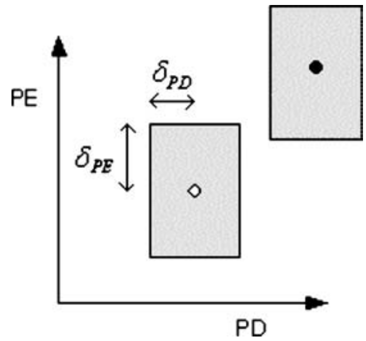

Figure 3. The new feature point does not fall into the area of any group (case 2).

equal to K.

3. Recommendation mechanisms

In this section, the access history of the user in the MRS is first introduced. We also present the three recommendation methods.

3.1. The profile manager

In the MRS, the profile manager is implemented for the purpose of updating the access histories of the users. When the user accesses a music object from the list of music objects or the recommendation results, the profile manager will record the object information into the access history. An example of the access history is shown in Table 1.

As shown in Table 1, the information of each accessed music object, i.e., the access time, the object id, the corresponding music group which the object belongs to, and the corresponding transaction, is recorded in the access history. The transaction is defined as the set of music objects accessed at the same time. Note that the transaction id is monotonically

Table 1. A sample of the access history.

Access time	Object ID	Music group	Transaction
2004/3/06 AM 11:47:03	1	B	T1
2004/3/06 AM 11:47:03	23	C	T1
2004/3/12 AM 10:11:25	7	D	T2
2004/3/12 AM 10:11:25	5	C	T2
2004/3/12 AM 10:11:25	32	B	T2
2004/3/16 AM 09:51:33	16	A	T3
2004/3/16 AM 09:51:33	19	B	T3
2004/3/16 AM 09:51:33	42	A	T3
2004/3/20 AM 08:31:12	31	D	T4
2004/3/20 AM 08:31:12	63	C	T4
2004/3/20 AM 08:31:12	26	A	T4
2004/3/22 AM 10:24:49	53	B	T5
2004/3/22 AM 10:24:49	12	A	T5

increasing.

3.2. The CB method

Based on the content-based filtering approach, the purpose of the CB method is to recommend the music objects that belong to the music groups the user is recently interested in. To capture the recent interests of the user, we analyze the latest transactions in the access history as follows. In the following example, we only use the latest five transactions for simplicity.

Each transaction is assigned a different weight, where the latest transaction has the highest weight. Moreover, the music group containing more accessed music objects in a transaction has a higher weight than other groups in the same transaction. The weight GW_i of music group G_i is computed as follows:

$$GW_i = \sum_{j=1}^{n} TW_j \times MO_{j,i} \tag{4}$$

where TW_j is the weight of transaction T_j; n is the number of latest transactions used for analysis, and $MO_{j,i}$ is the number of music objects which belong to music group G_i in transaction T_j.

These weights will be recorded in a *preference table* for the user. After calculating the weight for each music group, the MRS ranks all the music groups. The music group with a greater weight takes a higher priority of recommendation. To avoid recommending a large number of music objects to users, the MRS limits the number of music objects for recommendation. According to the GW_i, different numbers of music objects from the music groups will be recommended. The number of music objects R_i from each music group is decided as follows:

$$R_i = \left\lceil N \times \frac{GW_i}{\sum_{k=1}^{M} GW_k} \right\rceil \tag{5}$$

where N is the number of music objects in the recommendation list; GW_i is the weight of the target group, and M is the total number of music groups in *MRS*.

For music group G_i, we select the latest R_i music objects which have not been accessed by the user. In the recommendation list, the music objects will be sorted by the corresponding group weight in the decreasing order. In the same music group, the latest music object will be first recommended.

Example 1. Take the user's access history shown in Table 1 as an example. Assign the weights 0.4096, 0.512, 0.64, 0.8, and 1 to T1, T2, T3, T4, and T5, respectively. Using Eq. (4), the weight for each music group is calculated, as shown in Table 2.

According to Table 2, the total weight of all music groups is 8.6752. Suppose the number of music objects to be recommended is 20. By applying Eq. (5), the result is shown in Table 3.

Take music group A for example. The latest eight music objects in music group A will be selected for recommendation. Note that $\sum_{i=1}^{M} R_i$ may be more than N (i.e. $N = 20$ in our example), and we select the first N music objects for recommendation.

Table 2. The preference table for the user.

Music group	Weight
A	3.08
B	2.5616
C	1.7216
D	1.312

Table 3. Number of music objects to be recommended in each group.

Music group	Number of recommended music objects
A	8
B	6
C	4
D	4

3.3. The COL method

As described above, the recommendation of the CB method depends on the users' interests and the interests are derived from the users' access histories. Therefore, the users will never get a recommendation of the music objects belonging to the music groups they never accessed before. That is, the CB method tends to provide expected and interesting music objects for users. Based on the collaborative approach, the purpose of the COL method is to provide surprising findings due to the information sharing between *relevant users*.

To refer to the information from other users, we group the users first. In the COL method, we apply the technique proposed in Wu et al. (2001) for user grouping. The idea of the technique is to derive the profiles of user interests and behaviors from transactions in the access histories. Users with similar profiles of interests and behaviors will be identified as relevant users. In Wu et al. (2001), the *large-1 itemsets* derived from transactions in the access history are used for user interests and the *large-2 itemsets* are used for user behaviors. Two examples are shown as follows. Example 2 shows the process of capturing user interests and Example 3 shows the process of capturing user behaviors.

Example 2. Suppose there are five transactions in the access history as shown in Table 4.

We construct the *interest table* from the access history of the corresponding user as shown in Table 5.

Table 4. The access history of a user.

Transaction	Music groups in transaction
T1	A, C, E
T2	B, C, E, F
T3	C, D, E, F
T4	B, C, D, F
T5	A, G

Table 5. The interest table.

Music group	Count	First transaction (FT)	Last transaction (LT)
A	2	T1	T5
B	2	T2	T4
C	4	T1	T4
D	2	T3	T4
E	3	T1	T3
F	3	T2	T4
G	1	T5	T5

Table 6. The supports of the music groups.

Music group	Support
A	0.4
B	0.5
C	0.8
D	0.67
E	0.6
F	0.75
G	1

In this method, the *support* of a music group is calculated by Eq. (6). This equation indicates that only the transactions after the first transaction of a music group are considered in the support measure. Compared with the traditional data mining approaches, the effect of the transactions before the first transaction of a music group is ignored.

$$Support = \frac{Count}{T_c - FT + 1} \tag{6}$$

where T_c denotes the current transaction number.

Therefore, we compute the support for each group by Eq. (6). Suppose the T_c is 5. The support for each music group is shown in Table 6:

If the minimum support is 75%, there will be three large-1 itemsets, i.e., music groups C, F and G, which form the *interest profile* for the user.

Example 3. Take the access history shown in Table 4 for example. We construct the behavior table and compute the support of each music group pair as shown in Table 7.

If the minimum support is 0.65, there will be four large-2 itemsets, i.e., pairs AG, CD, CF and DF. Therefore, the *behavior profile* {AG, CD, CF, DF} is derived for the user. After deriving the interest profile and the behavior profile of the user, we construct an I-B matrix and transform it into an I-B vector. For example, if a user has interests {C, F, G} and behaviors {AG, CD, CF, DF}, the I-B matrix of the user is shown in Table 8.

Then, we transform the I-B matrix to the I-B vector (0000001 000000 11010 0010 000 10 1). Therefore, each user has a corresponding I-B vector. According to the I-B vector, we compute the Euclidean distance between two users. Then, we apply the clustering algorithm to group users.

In the COL method, we capture user interests and behaviors from transactions in the user's access history by applying the technique proposed in Wu et al. (2001). The users are then grouped based on their interests and behaviors. To make a recommendation for a user, the weights of each music group associated with the relevant users in the same group will be averaged. These average weights will be recorded in a *reference table* for the user. When the user chooses the COL method for recommendation, the recommendation module will compute the difference of the weights for each music group in the associated

Table 7. The behavior table with the corresponding support.

Music group pair	Count	FT	LT	Support
AC	1	T1	T1	0.2
AE	1	T1	T1	0.2
AG	1	T5	T5	1
BC	2	T2	T4	0.5
BD	1	T4	T4	0.5
BE	1	T2	T2	0.25
BF	2	T2	T4	0.5
CD	2	T3	T4	0.67
CE	3	T1	T3	0.6
CF	3	T2	T4	0.75
DE	1	T3	T3	0.33
DF	2	T3	T4	0.67
EF	2	T2	T3	0.5

Table 8. The I-B matrix.

	A	B	C	D	E	F	G
A	0	0	0	0	0	0	1
B		0	0	0	0	0	0
C			1	1	0	1	0
D				0	0	1	0
E					0	0	0
F						1	0
G							1

preference table and the reference table. According to the weight differences, the COL method recommends music objects to the user in a way similar to the CB method. Example 4 shows the process to construct a reference table and to make recommendation using the COL method. As in the CB method, the latest five transactions in each access history are considered.

Example 4. Suppose there are three persons UA, UB, and UC in user group U. Table 9 shows the partial access histories of UA, UB and UC. We omit access time and object id for clearer illustration.

Assign the weights 1, 0.8, 0.64, 0.512, and 0.4096 to the latest five transactions, respectively. We apply the Eq. (4) in the CB method to each access history. The result is shown in Table 10.

Table 9. The latest five transactions in the access histories of users UA, UB, and UC.

Partial access history of UA		Partial access history of UB		Partial access history of UC	
Music group	Transaction	Music group	Transaction	Music group	Transaction
B	T8	E	T13	A	T11
C	T8	F	T13	C	T11
D	T9	A	T14	B	T12
C	T9	A	T14	B	T12
B	T9	B	T15	A	T13
A	T10	C	T15	A	T13
B	T10	A	T16	D	T14
A	T10	D	T16	C	T14
D	T11	B	T17	F	T14
C	T11	A	T17	A	T15
A	T11	B	T17	C	T15
B	T12	E	T17	B	T15
A	T12			D	T15
				C	T15

Table 10. The preference tables for users UA, UB and UC.

Preference table of UA		Preference table of UB		Preference table of UC	
Music group	Weight	Music group	Weight	Music group	Weight
A	3.08	A	2.824	A	2.6896
B	2.5616	B	2.64	B	2.024
C	1.7216	C	0.64	C	3.2096
D	1.312	D	0.8	D	1.8
		E	1.4096	F	0.8
		F	0.4096		

To make a recommendation for UA, the reference table for UA is constructed as shown in Table 11.

The weight for each music group in the reference table is subtracted from that in the preference table, and the result is shown in Table 12.

In the COL method, the music group with zero or negative weight difference will not be recommended to the user. Therefore, we recommend music groups C, E, and F to UA. The numbers of music objects from music groups C, E, and F for recommendation are decided by Eq. (5), as shown in Table 13. Note that the M in Eq. (5) is set to 3 in this case.

The order of the music objects to be recommended is decided by the same way as the CB method.

Table 11. The reference table for user UA.

Music group	Weight
A	2.7568
B	2.332
C	1.9248
D	1.3
E	0.7048
F	0.6048

Table 12. The table of weight differences.

Music group	Weight difference
A	−0.3232
B	−0.2296
C	0.2032
D	−0.012
E	0.7048
F	0.6048

Table 13. Number of music objects to be recommended in each group for user UA.

Music group	Number of recommended music objects
C	3
E	10
F	8

3.4. The STA method

The third recommendation method is based on the statistics. We define the *long-term hot music group* as the music group containing the most music objects in the access histories of all users. Similarity, we define the *short-term hot music group* as the music group containing the most music objects in the latest five transactions in the access histories of all users. When the user chooses this recommendation method, the MRS recommends the latest N music objects (which have not been accessed by the user), half from the long-term hot music group and the other half from the short-term hot music group to the user.

4. Experiments

The implementation of the MRS is shown in Section 4.1. Moreover, the results of a series of experiments are shown and explained in Section 4.2.

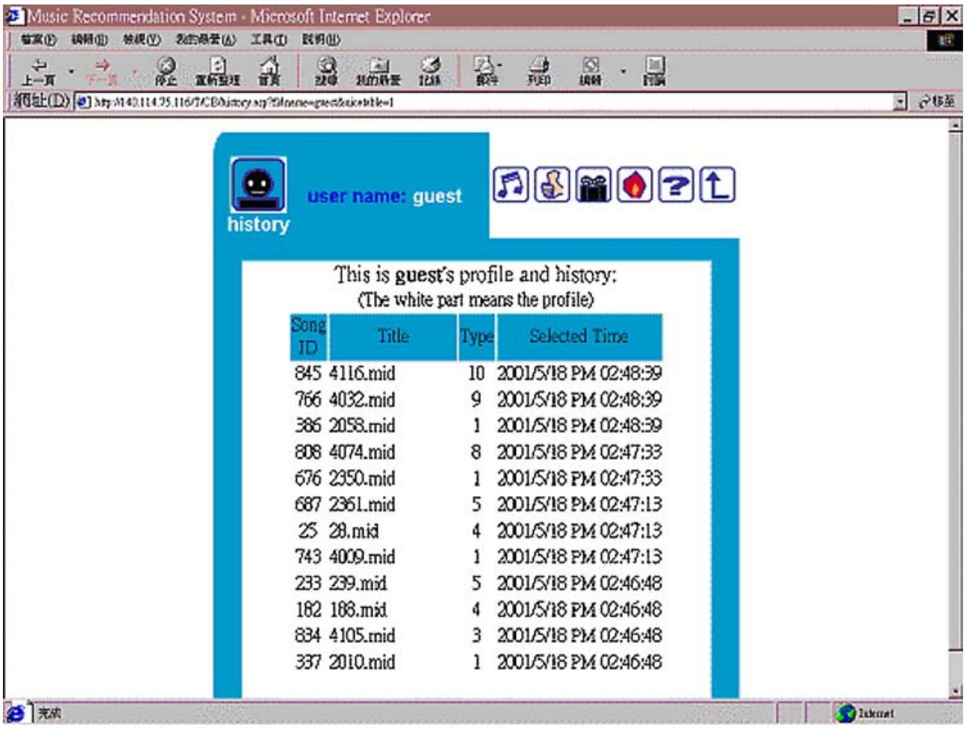

Figure 4. The access history.

4.1. *Implementation*

Figure 4 shows the user's access history. Figure 5 shows the recommendation result by applying the CB method. Figure 6 shows all operators in the MRS.

4.2. *Experiment results*

In this subsection, we show the experiment results, including the effectiveness of the track selector, the effectiveness of the feature selection, and the quality of recommendations.

4.2.1. Effectiveness of the track selector. To evaluate the effectiveness of the track selection method, we ask an expert to select a representative track from each MIDI file. Then, we apply our method on the same testing data set of 100 MIDI files. An 83% correctness rate is achieved by our method.

4.2.2. Effectiveness of the feature selection. The data set of 100 MIDI files is first classified by the expert into five groups, i.e., lyric music, jazz music, rock music, country music, and classical music. Then, we apply the K-means algorithm (Jain et al., 1988) to classify the

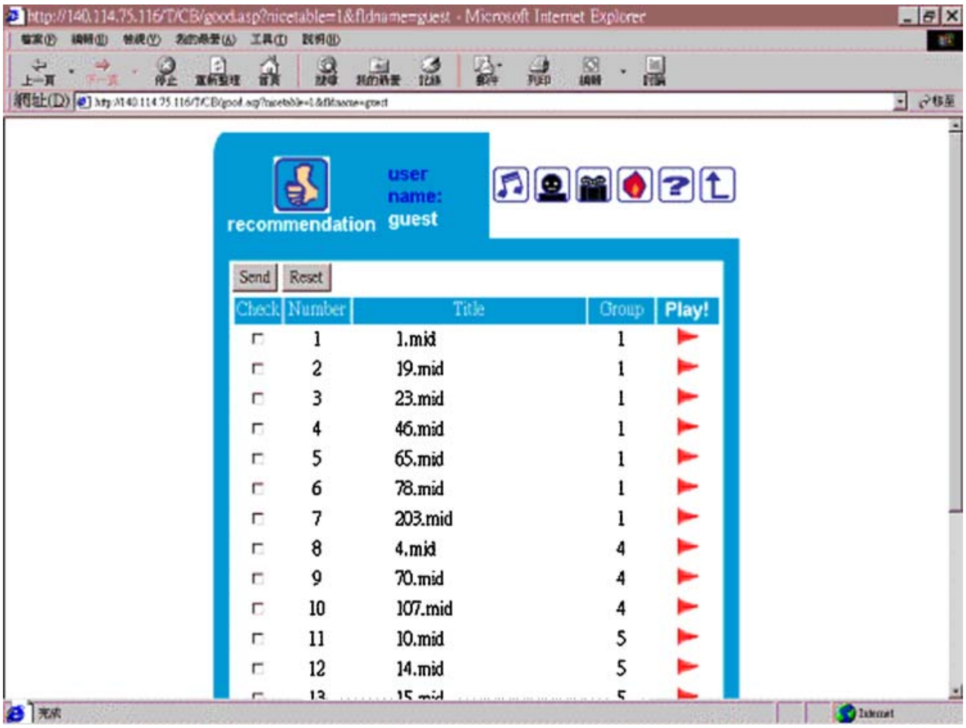

Figure 5. The recommendation by using the CB method.

same data set on the six features, respectively. By comparing the results, the *error rate* is computed as follows:

$$error\ rate = \frac{\sum_{i=1}^{5} E_i}{100} \tag{7}$$

where E_i is the number of music objects mistakenly classified into group i.

The error rate with respect to each feature is shown in Table 14.

According to the error rates shown in Table 14, we select the feature PE, which has the lowest error rate to be the representative feature from the perceptual property of pitch. To consider the influence of other perceptual properties on classification, we choose the features TD and LD. To compare with the error rates by single features, we apply the

Table 14. The error rates by features.

Feature	MP	SP	PD	PE	TD	LD
Error rate	65%	65%	60%	56%	66%	62%

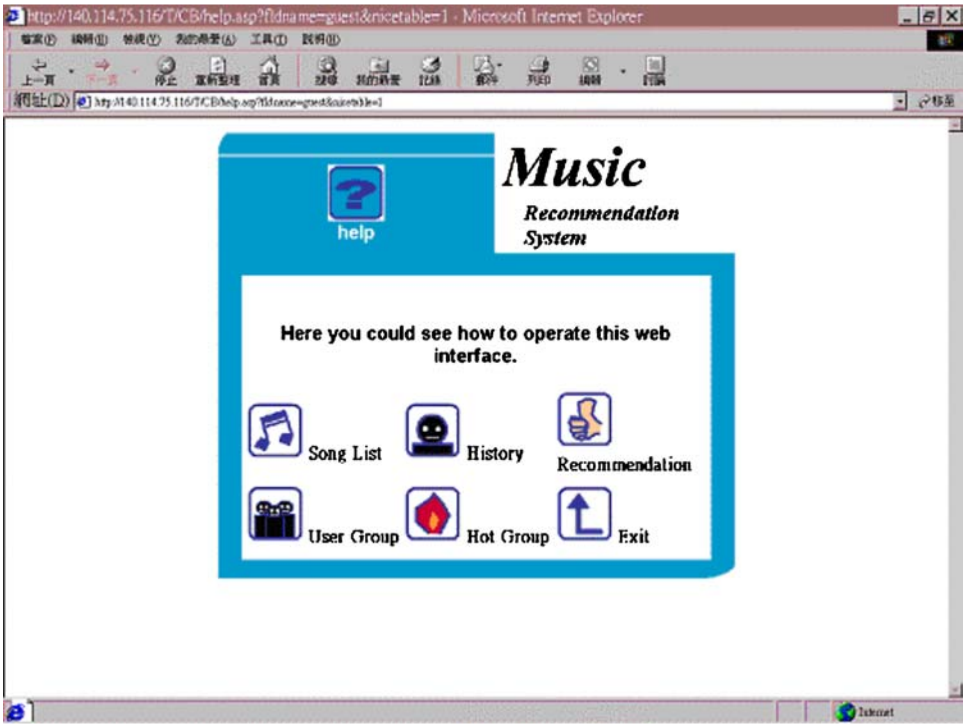

Figure 6. The operators in the MRS.

K-means algorithm to classify the same data set based on the three feature sets (PE, TD), (PE, LD), (TD, LD), and (PE, TD, LD). By using Eq. (7), the result of error rates is shown in Table 15.

The result shown in Table 15 indicates that it is better for the system to use multiple perceptual properties to represent a music object. According to the classification by using the K-means algorithm based on each feature, the mean of the distances between the feature points and the centroids of the associated groups is used as the threshold of the corresponding feature. This threshold is then used in the classifier of the MRS.

4.2.3. Quality of recommendations. We invite 10 students to perform the experiments. In each recommendation method, the MRS lists 20 music objects for users. The quality of the

Table 15. The error rates by feature sets.

Feature sets	(PE, TD)	(PE, LD)	(TD, LD)	(PE, TD, LD)
Error rate	43%	47%	39%	44%

Table 16. The precision of recommendation based on the classification by a single feature.

Single feature	PE (%)	TD (%)	LD (%)
The CB method	37	39	35
The COL method	19	23	18
The STA method	24	27	21

Table 17. The precision of recommendation based on the classification by feature sets.

Feature sets	(PE, TD) (%)	(PE, LD) (%)	(TD, LD) (%)
The CB method	59	51	62
The COL method	17	23	22
The STA method	29	31	26

recommendation is measured by *precision* defined as follows:

$$precision = \frac{N_A}{N} \tag{8}$$

where N is the number of music objects in the recommendation list and N_A is the number of music objects which the user accesses in the recommendation list.

The results are shown in Tables 16 and 17. Note that we select 20 as N in the experiments. For the CB method, the recommendation based on feature sets is better than the recommendation based on a single feature. The result indicates that it is better to use multiple features to represent a music object, which coincides with the result in Section 4.2.2. For the three recommendation methods, the precision of the CB method is better than the COL and the STA methods. The reason is that the CB method considers only private information of the user. On the contrary, the COL method tends to provide surprising music objects for users, which may be interesting. In addition, the STA method provides hot music groups derived from all access histories. Therefore, the recommendation result of the STA method is better than that of the COL method.

5. Conclusion

In this paper, we present a music recommendation system to provide a personalized service of music recommendation. Our contribution is to integrate both the content of music objects and the opinions of the relevant users for better music recommendation. For each user, we determine the user's favorite degrees to the music groups in the proposed CB method. For providing surprising music objects, we further take into account the opinions from the users of the same user group. The COL method is then proposed for this purpose. In the MRS, we design a classifier to automatically group the music objects based on the extracted features from their melodies. This classifier avoids the overhead from the task of manual classification. Moreover, we derive both the interest and behavior profiles from the users' access histories for user grouping. The proposed technique to derive user profiles has the adaptability to large number of accessed music objects. We also perform a series of experiments to show that our recommendation system is practical.

The MRS can be regarded as a basic framework. The function blocks, such as the track selector, the feature extractor, the classifier, and the recommendation module, can be replaced by the alternatives. For example, more features can be extracted for music grouping by modifying the feature extractor. Similarly, we can adopt different techniques to construct the classifier, such as machine learning or data mining, based on the properties of the system and the recommendation services. Therefore, our recommendation system is also flexible.

Our recommendation system can be further enhanced in some ways. For example, to reduce the time to browse all recommended music objects, the summarization of music objects may be necessary. Due to the complex semantics contained in the music objects, new features can be investigated for more effective music grouping. Moreover, other recommendation methods to satisfy various user requirements can be developed.

Acknowledgment

This work was partially supported by National Science Council, Republic of China under Contract Numbers NSC 93-2213-E-004-013 and NSC 93-2752-E-007-004-PAE.

References

A movies recommender system, http://www.movielens.umn.edu, 2003.

Agrawal, R. and Srikant, R. (1994). Fast Algorithm for Mining Association Rules. In *Proceedings of International Conference on Very Large Data Bases*, pp. 487–499.

Balabanovic, M. and Shoham, Y. (1997). Fab: Content-based Collaborative Filtering Recommendation. *Communications of the ACM*, 40, 66–72.

Basu, C., Haym, H., and Cohen, W.W. (1998). Recommendation as Classification: Using Social and Content-Based Information in Recommendation. In *Proceedings of National Conference on Artificial Intelligence*, pp. 714–720.

Billsus, D. and Pazzani, M. (1999). A Hybrid User Model for News Story Classification. In *Proceedings of International Conference on User Modeling*.

Cheung, K.W. and Tian, L.F. (2004). Learning User Similarity and Rating Style for Collaborative Recommendation. In *Proceedings of European Conference on IR Research*, 7, pp. 395–410.

Goldberg, D., Nichols, D., Oki, B.M., and Terry, D. (1992). Using Collaborative Filtering to Weave an Information Tapestry. *Communications of the ACM*, 35, 61–70.

Jain, A.K. and Dubes, R.C. (1988). *Algorithms for Clustering Data*. 1st edn., Prentice Hall.

Joachims, T., Freitag, D., and Mitchell, T. (1997). WebWatcher: A Tour Guide for the World Wide Web. In *Proceedings of International Joint Conference on Artificial Intelligence*, pp. 770–775.

Krulwich, B. and Burkey, C. (1996). Learning User Information Interests through Extraction of Semantically Significant Phrases. In *Proceedings of AAAI Spring Symposium on Machine Learning in Information Access*.

Lang K. (1995). Newsweeder: Learning to Filter Netnews. In *Proceedings of International Conference on Machine Learning*, pp. 331–339.

Lieberman, H. (1995). An Agent that Assists Web Browsing. In *Proceedings of International Joint Conference on Artificial Intelligence*, pp. 924–929.

Mckay, C. and Fujinaga, I. (2004). Automatic Genre Classification Using Large High-level Musical Feature Sets. In *Proceedings of International Symposium on Music Information Retrieval*.

Mooney, R.J. and Roy, L. (2000). Content-Based Book Recommending Using Learning for Text Categorization. In *Proceedings of ACM International Conference on Digital Libraries*.

Konstan, J.A., Miller, B.N., Maltz, D., Herlocker, J.L., Gordon, L.R., and Riedl, J. (1997). Grouplens: Applying Collaborative Filtering to Usenet News. *Communications of the ACM*, 195–240.

Kuo, F.F. and Shan, M.K. (2002). A Personalized Music Filtering System Based on Melody Style Classification. In *Proceedings of IEEE international Conference on Data Mining*, pp. 649–652.

Rucker J. and Polanco, M.J. (1997). Personalized Navigation for the Web. *Communications of the ACM*, 40, 73–89.

Sayood, K. (2000). *Introduction to Data Compression*. 2nd edn., Morgan Kaufmann.

Sharadanand, U. and Maes, P. (1995). Social Information Filtering: Algorithms for Automating 'Word of Mouth'. In *Proceedings of CHI'95 Conference on Human Factors in Computing Systems*, pp. 210–217.

Smith, B. and Cotter, P. (2000). A Personalized Television Listings Service. *Communications of the ACM*, 43, 107–111.

Uitdenbogerd, A.L. and Zobel, J. (1998). Manipulation of Music For Melody Matching. In *Proceedings of ACM International Conference on Multimedia*, pp. 235–240.

Wu, Y.H., Chen, Y.C., and Chen, A.L.P. (2001). Enabling Personalized Recommendation on the Web Based on User Interests and Behaviors. In *Proceedings of IEEE International Workshop on Research Issues in Data Engineering (RIDE)*, pp. 17–24.