# A Multimedia Contents Recommendation for Mobile Web Users

**Mee Yeon Kang [a], Yoon Ho Cho [b], Jae Kyeong Kim [a]**

[a] *School of business Administration, KyungHee University, Seoul, 130-701, Korea*
[b] *School of e-Business, Kookmin University, Seoul, 136-702, Korea*
*kkangmee@lycos.co.kr, www4u@kookmin.ac.kr*

**Abstract**

As mobile market grows more and more fast, the mobile contents market, especially music contents for mobile phones have recorded remarkable growth. In spite of this rapid growth, mobile web users experience high levels of frustration to search the desired music. New musics are very profitable to the content providers, but the existing collaborative filtering (CF) system can't recommend them. To solve these problems, we propose an extended CF system to reflect the user's real preference by representing the characteristics of users and musics in the feature space. We represent the musics using the music contents based acoustic features in multi-dimensional feature space, and then select a neighborhood with the distance based function. Furthermore, this paper suggests a recommendation for procedure for new music by matching new music with other users' preference.
The suggested procedure is explained step by step with an illustration example.

***Keywords:***

Collaborative filtering, recommender system, mobile commerce, multimedia contents

## 1. Introduction

Recently as the mobile devices are distributed extensively, the mobile web service which can connect to the World Wide Web in anytime and anywhere is spreading more rapidly. The mobile web service providers are also growing with the tremendous speed year by year. The mobile web contents are over the 2,000 types, among them, the most popular contents are the multimedia contents like music, character images and etc. Music contents, especially, have grown with the fast growth and their market share takes large portion [7].

In spite of the popularity and fast growth of the music contents, many users experience high level of frustration in searching for the specific musics which they really want due to the existing inefficient sequential search. When a user logs on to the music service site with his mobile phone, he is presented with the weekly or monthly best-seller or the most recent introduction music list. The user pages through the list and selects an entry to pre-listen the music for checking out its sound contents hoping that the music will be what he wants. If the user liked the music, he might save the music in his mobile phone, or if not, he should repeat the same steps until he meets the desired music or decides to give up purchasing.

These difficulties result from a mobile phone's limitations such as the small LCD, tiny keypad, and sophisticated browsers compared to PCs [13]. For this reason, many users tend to eschew to search musics by their mobile phones; therefore the purchase conversion rate drops in mobile web environment. Moreover it is hard to explain the features of music like the tempo, pitch, beat and other spectral features and to know one's preference exactly. Consequently the repetitive searching process and music contents' unique characters make both contents providers and mobile users need a recommender system.

A recommender system assists customers in finding the musics they would like to purchase. Collaborative filtering (CF) is known as the most successful system in recommender systems. But the CF system results in many problems because of its input data representation. And in mobile web environment, new musics are very frequently supplied, and their purchasing ratio is considerably high. However existing CF systems can't recommend new musics.

In this paper we propose an extended CF-based recommender system for mobile web users, which can solve these problems. The system extracts the music's various content-based acoustic features such as MFCCs, tempo, beat, and pitch, and then represents the extracted music in a multi-dimensional feature space. Therefore the proposed recommender system can identify the real neighbors and recommend new musics. This paper describes the procedure of systems, and explained step by step with an illustration example.

## 2. Related Work

### 2.1 Collaborative Filtering

Collaborative filtering (CF) is the most familiar, most widely implemented and most mature of the technologies in recommender system [1]. It is defined as one which makes recommendations by finding correlations among users of a recommender system [8]. The goal of CF system is to suggest items for a target user based on his previous preference and the opinions of other like-mined users. But the CF system has many problems such as the sparsity, and scalability problem.

The sparsity problem occurs due to the sparse user profile which is organized with user-item matrix. In this profile, the number of items is very large; on the other hand the number of users is relatively small. As a result the CF system can't compute the similarity accurately and results in poor quality of recommendation.
The scalability problem occurs due to rapidly growing the number of users and items. When the CF system finds the neighbors, it spends too much computation time, and scales poorly in practice.
Many researches are proposed to solve these problems by various techniques such as product taxonomy, SVD and web usage mining [2,6,11,12].

Beyond these problems the CF system has a fundamental problem related its input data representation. The general CF system uses a user profile which is composed with $m$ users' ratings about $n$ items, and is represented by the $m \times n$ user-item binary matrix. When a user has purchased a music, the CF systems set its rating 1.0, or zero. With this input data representation, the CF system is hard to identify user's real neighbors. The CF system uses only common musics for find the target user's neighbors. Therefore if someone whose preference is similar to the target user's didn't purchase common musics, then he didn't be selected the target user's neighbor.

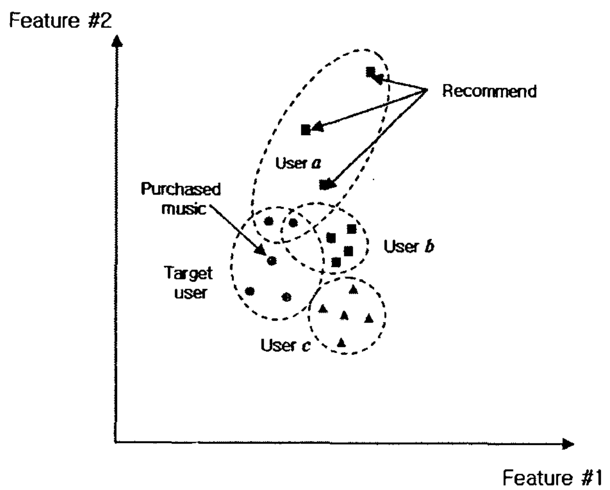


Figure 1 - CF system in feature space

Figure 1 shows the CF system's problems related the input data representation. Generally the CF system selects user $a$ as the target user's neighbor because he purchased the common musics of the target user. But user $a$'s preference is quite different from that of target user when the system represent their purchased musics in feature space. On the other hand, user $b$ and $c$ are more similar to the target user rather than $a$. Moreover the CF system can't find hidden neighbors. The hidden neighbor is defined as the person whose preference is similar to that of the target user, but he didn't purchase the common music. In Figure 1, user $c$ is the hidden neighbor.

And the CF system has the new-item ramp up problem [1]. The CF system follows from a comparison between the target user and other users based on the users' ratings. Therefore new musics that have few ratings can't be easily recommended. In the mobile web environment, new musics are very frequently provided, and their purchasing ratio is very high. But the CF system may not solve this problem.

We describe how to solve these problems related the input data representation and new item recommendation in Section 3.

### 2.2 Music Information Retrieval

Main thread in music information retrieval (MIR) is to automate the retrieval process, and there are various methods for automating the retrieving process. To automate the process, they used the various techniques such as music genre classification, indexing, and clustering. Especially researches for the genre classification have been growing amount of attention since the digital musics emerged on the Internet. They assume that the members of particular music genre share certain characteristics typically related to the instrumentation, rhythmic structure, and pitch content of the music. Therefore if we have extracted the features of music, then we could classify the music automatically.

Before extracting the features, we choose which features to use. There are a large number of the different features set, mainly originating from the area of speech recognition, have been proposed to represent audio signal. Recently many nonspeech signals analysis techniques are proposed [14, 8]. There have been a number of related feature set, especially Tzanetakis et al. [14] presents the feature set which include both speech and nonspeech audio signal analysis. They proposed the content based acoustic features, which are classified into timbral features, rhythmic content features, and pitch content features.

Timbral features are mostly originated from traditional speech recognition techniques. Typical timbral features include spectral centroid, spectral rolloff, spectral flux, energy, zero crossing, mel-frequency cepstral coefficients (MFCCs) [10]. The spectral centroid measures the balance of the spectrum, the spectral rolloff measures skewness of the spectrum, the spectral flux measures amount of local spectral change, the energy measures the amplitude

distribution of the signal, and zerocrossing measure the noise of the signal.
Rhythmic content features characterize the movement of music signals over time and constrain information about the regularity of the rhythm, the beat and tempo information [10].
Pitch content features describe the melody and harmony information about music signals and are extracted based on various pitch detection techniques.

For the system to extract the features of music, we decide to use the MARSYAS system [13], which is known as one of the best feature extraction system in MIR. MARSYAS could extract a number of the music features, but we decide to use the above described features.

## 3. Overall Recommendation Procedure

### 3.1 Overall Process

We design the system to deal with the input data representation and a new music ramp-up problem on the mobile web. The system is organized with four modules, and the overall processes are shown in Figure 2. The process follows the general CF, and is added a new music recommendation module.

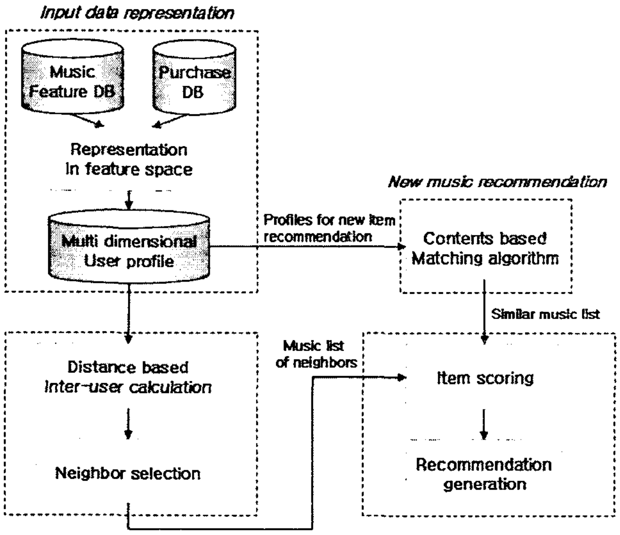


Figure 2 - Overall Process

*The input data representation module* creates a user profile with the music features database and purchase database. The music feature database is constructed of the features, which described in Section 2.2. The purchase database is constructed of the individual user's purchasing or pre-listening histories. In this module, the system makes each user's preferred music set, which reflects the individual user's preference.

*The neighborhood formation module* selects the similar users to the target user. The existing CF systems calculate the inter-user correlation by the Cosine formulation or Pearson Coefficient; however these are hard to identify the neighbors who are really close to the target user as described in Section 2.1. This module calculates the distance between the target user and other users, the close users select his neighbors. Finally it gathers neighbors' purchasing or pre-listening musics, and gives the list of them to the recommendation generation module.

*The recommendation generation module* creates a recommendation list for the target user. This module scores the musics which neighbors purchased or pre-listened. Basically this score increases when more neighbors purchased or pre-listened.

*The new music recommendation module* solves the new music ramp-up problem. This module finds the most similar music by the content-based matching algorithm, and gives these musics and their ratings to the recommendation generation module. This new music recommendation module is explained particularly in Section 4.

### 3.2 User Profile Creation

A user profile, which is the key component for CF system, includes the information of user's preferences about music. This module is organized with two phases. The first phase is the feature extraction phase, and the second phase is the user profile construction phase which collects the each user's preferred music set and organizes a user profile.

*Phase 1: The features extraction*

To extract features of the musics, we use the MARSYAS. The timbral features consist of average spectral centroid, average spectral rolloff, average spectral flux, energy, and zerocrossing. In original MARSYAS system, they also use the variances each feature, but we except these values due to the system performance. MFCC features consist of the mean of the first five MFCC coefficients over the frames. The rhythmic features consist of six features from the rhythm histogram. Pitch features are five features from the pitch histograms.

The system extracts features of the individual music, which consists of 21 features. Finally the system represents the extracted music in the multi-dimensional feature space as a point.

*Phase 2: The user profile construction*

In the mobile environment users could take action such as purchasing, pre-listening or giving up saving some music. Therefore we set a rating about user's action assuming that these actions reflect the user's preference about the corresponding music.

If user $a$ has taken action to the music $x_i^a$, it could be given a rating $r_i$. We define the $r_i$ as the following:

$$r_i = \begin{cases} 1.0 & \text{If user } a \text{ has purchased the music } x_i \\ 0.5 & \text{If user } a \text{ has pre-listened the music } x_i \\ 0 & \text{Otherwise} \end{cases} \quad (1)$$

The general CF system uses $m \times n$ user-item binary matrix as a user profile, but the amount of musics are very large, therefore we use the whole users' preferred music set as the user profile. The preferred music set is the collection of musics which a user purchased or pre-listened, and we define the PMS as the following:

$$\begin{aligned} \text{User } a\text{'s } PMS &= \{(x_i^a,\ r_i) \mid 1 \le i \le n\} \\ &= \{(x_1^a,\ r_1), (x_2^a,\ r_2), \cdots, (x_n^a,\ r_n)\} \end{aligned} \quad (2)$$

$x_i^a$ is the music which the user $a$ purchased or pre-listened, $n$ is the number of the musics in user $a$ 's PMS, and $r_i$ is the music $x_i^a$ 's rating. The illustration of the PMS is shown in Figure 3.

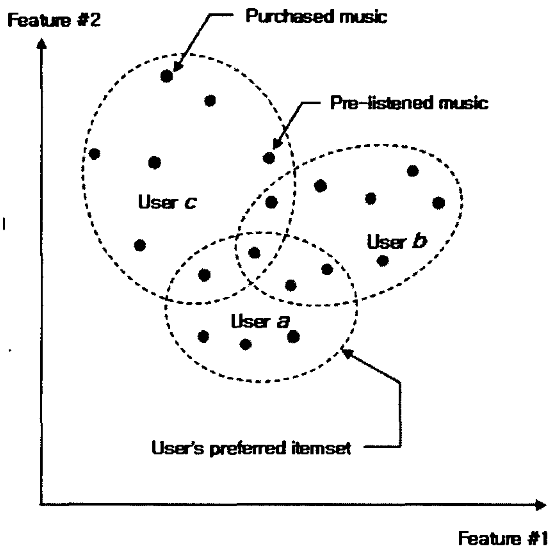


Figure 3 - User's preferred music set

In Figure 3, user $a$, $b$, and $c$ are represented in the 2-dimensional feature space. We represent all of the musics, and bind the purchased and pre-listened musics as the user's PMS by a dotted line for easy understanding. The user $a$'s PMS includes 7 musics, and consists of 2 purchased musics and 5 pre-listened musics, which reflect his preference. User $b$ and $c$'s PMS also are composed by this way. The user profile is composed by the collection of the individual user's PMS.

### 3.3 Neighborhood Formation

Because all users and musics are represented in multi-dimensional feature space, the system can find the target user's neighbors with the distance based function. If a user were close to the target user in this space, his preference would be similar to the target user. As shown in Figure 1, the existing CF system is hard to identify the real neighbors like user $b$ and can't find the hidden neighbors like user $c$. With the distance among users, the system can solve these problems.

To calculate the distance among users, we regard a user's PMS as a cluster. There are many inter-cluster distance functions; especially the Centroid Euclidean distance function is the most popular method. This function is simple and easy to compute. But it doesn't work well except the condition which each cluster is fairly distributed and the cluster's form is a circle. The individual user's preference isn't same and their distributions are difference from each other as shown in Figure 3. Therefore we use the Average Inter-Cluster distance function [3] which considers the variance of the clusters.

This function calculates the inter-cluster distances. For example, cluster A has the 3 members, and cluster B has 4 members. This function computes the all distance between 3 members in cluster A and 4 members in cluster B, total 12 This can reflect the cluster's variance, but can't reflect the user's preference. Therefore we modify the function into the weighted inter-cluster average distance function. This function uses the rating $r_i$ as the weight, so it can consider users' preferences. The detail is shown as the follow:

$$d(a,b) = \frac{\sum_{j=1}^{n_b} \sum_{i=1}^{n_a} \sqrt{\sum_{k=1}^{l} (r_i \times x_{ik}^a - r_j \times x_{jk}^b)^2}}{n_1 \cdot n_2} \quad (3)$$

Where $d(a, b)$ is the distance function between user $a$ and $b$, and. $n_1$ and $n_2$ is the number of musics in user $a$'s $PMS$ and user $b$'s $PMS$, respectively. $r_i$ and $r_j$ is the rating of the music $x_i^a$ in user $a$'s $PMS$ and music $x_j^b$ in user $b$'s $PMS$, respectively. $x_{ik}^a$ and $x_{jk}^b$ is the value of $k$th feature of the music $x_i^a$ and music $x_j^b$, respectively, and $l$ is the number of features.

We sort the users according to the distance value by ascending order, and then select the $L$ neighbors. Finally we determine the neighborhood set $H$ for target user $c$, and it defined as $H = \{h_1, h_2, \ldots, h_L\}$ such that $c \notin H$.

### 3.4 Recommendation Generation

The proposed system generates a list of $N$ musics, $R = \{M_1, M_2, \ldots, M_N\}$ such that $M_i \notin$ {the music that the target user $c$ has already purchased}. For this recommendation list, we use $PLS(c,i)$, which denotes the Purchase Likeliness Score of the target user $c$ on music $x_i^a$ $PLS(c, x_i^a)$ is computed as the following:

$$PLS(c, x_i^a) = \frac{\sum_{a \in H} (r_i - \bar{r}) \times sim(c,a)}{\sum_{a \in H} sim(c,a)} \quad (4)$$

Where $\bar{r}$ is the user $a$'s average rating, and $r_i$ is the rating on the music $x_i^a$. $u$, $w$ are the other users included the neighborhood set $H$.

Because we use the distance function, we have to covert the distance values into similarity values, and normalize them. $sim(c,a)$ denotes the similarity function between target user $c$ and user $a$, which computed as the following:

$$sim(c,a) = \frac{\underset{u,w \in H}{Max}[\,d(u,w)\,] - d(c,a)}{\underset{u,w \in H}{Max}[\,d(u,w)\,] - \underset{u,w \in H}{Min}[\,d(u,w)\,]} \quad (5)$$

### 3.5 New Music Recommendation

In mobile web environment, new musics are frequently provided and their purchasing ratio is very high, but the existing CF system can't recommend new musics. The CF systems use the common items to find neighbors. Therefore they can't recommend musics that anyone didn't have been purchased. If new musics had ratings, they could be recommended.

The new music recommendation module is organized with two phase. The first phase is the determination of new musics recommendation which decides whether to recommend a new music or not. The second phase is the finding the most similar music which searches the music similar to a new music.

*Phase 1: Determination of new musics recommendation*

The recommendation module doesn't include the music purchased by the target user in the recommendation list. But the new music doesn't be purchased by the target user; therefore new musics can be included in the target user's *PMS*. For convenience we define the considered user set as $C = \{\,c\,\} \cup H$.
And a new music also has the above mentioned features, which can be extracted by MARSYAS. The system extracts the new music's features by the same way described in Section 3.2, and represents the music in the feature space.
With the considered user set C and new music's features, we suppose that a new music comes to any members' *PMSs* in C, it would be preferred by them. Therefore we define a condition which decides whether to recommend new music.

If the new music $y$ is included in the preferred boundaries of any members in $C$, we determine to recommend $y$ for the target user. The preferred boundary is defined as the follow:

$$y_k \in M_k \pm 3\sigma$$
$$M_k = \frac{\sum_{i=1}^{n} r_i \times x_{ik}}{n} \quad (6)$$

Where $M_k$ is the weighted mean and $\sigma$ is the standard deviation of the $k$th features of all musics in user $a$'s PMS. This function assumes that $n$ is over 30 and user $a$'s PMS follows the normal distribution.

But the number of musics in users' PMSs can be smaller than 30. In this case, their PMSs follow the t-distribution; therefore we define another preferred boundary as the follow:

$$y_k \in M_k \pm t(0.025, n-1) \times \frac{\sigma}{\sqrt{n}} \quad (7)$$

If $y_k$ belongs to the 95% confidence interval of user $a$'s PMS, it would be preferred by user $a$. The illustration of this is shown in Figure 4.

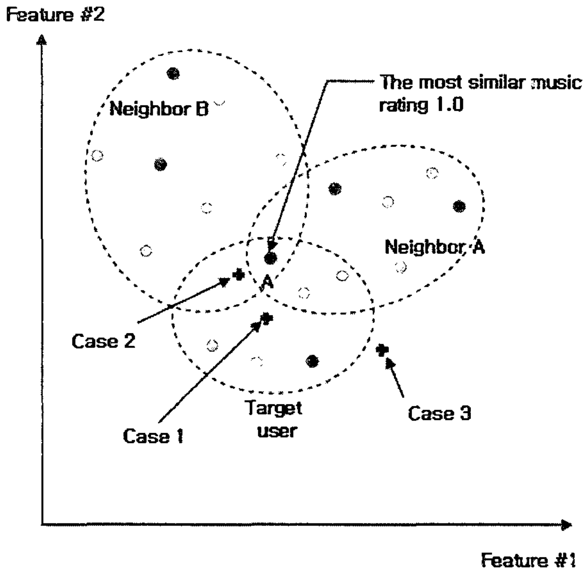


Figure 4 - New music recommendation

As shown Figure 4, there can be the 3 cases. The case 1 is that $y$ is included in one user's preferred boundary in C. the case 2 is that $y$ is included in some users' preferred boundaries in C. the case 3 is that $y$ is included in nobody's preferred boundary in C.
Because the case 2 is included in many users' PMS, its purchasing possibility is higher than the case 1. On other hand the case 3 isn't included anyone's PMS, it can't be recommended.

*Phase 2: Finding the most similar music*

The system determined whether to recommend the new music or not, but the new music still doesn't have a rating. To solve this problem, the system searches for the most similar music in the PMS where it belongs to.
All musics are represented in the feature space. The music which is the most similar to the new music has the highest possibility to prefer by the user who it belongs to. So the most similar music is that has the shortest distance from the new music. Therefore we find the most similar music as the follow:

$$MSM = \min\{d(x_i^a, y)\} \qquad (8)$$

$$d(x_i^a, y) = \sqrt{\sum_{k=1}^{l} (x_{ik}^a - y_k)^2} \qquad (9)$$

Where $MSM$ denotes the most similar music and $d(x_i^a, y)$ is the function that calculates distance between $y$ and $x_i^a$. $x_i^a$ is the music purchased by the user $a$, $x_{ik}^a$ is the its $k$th feature value, and $y_k$ is $k$th feature value of the new music, $l$ is the number of features.

The illustration is also showed in Figure 4. As shown in phase 1, the case 1 and 2 can be recommended, but we focus on the case 2. The case 2's MSM is the music A, and its rating value is 1.0. Therefore the case 2 is given the rating 1.0.

Through this way, new music can have a rating. With this rating, the system calculates the new music's PLS value. Therefore we generate a recommendation list R with combination of old musics and new musics.

# 4. Example

For helping to understand our system, in this section we present examples. The examples are organized with 2 examples: the first is the CF recommendation described method in Section 3.2 ~ 3.4. And the second is the new music recommendation described in Section 3.5.

### 4.1 The CF recommendation based music's features

There are five users' PMSs, which are the target user Kim, user Ahn., Kang, Lee, and Chae, and Table 1 shows the music feature database. For convenience, we represent the musics and users in 2-dimensional feature space.

Table 1- The music feature database

| Feature / Music | 1 | 2 |
|---|---|---|
| $m_1$ | 0.2 | 0.5 |
| $m_2$ | 0.5 | 0.3 |
| $m_3$ | 0.6 | 0.4 |
| $m_4$ | 0.4 | 0.2 |
| $m_5$ | 0.5 | 0.6 |
| $m_6$ | 0.8 | 0.8 |
| $m_7$ | 0.9 | 0.9 |
| $m_8$ | 0.4 | 0.4 |
| $m_9$ | 0.6 | 0.5 |
| $m_{10}$ | 0.7 | 0.4 |
| $m_{11}$ | 0.6 | 0.3 |
| $m_{12}$ | 0.4 | 0.4 |
| $m_{13}$ | 0.3 | 0.4 |

- Kim's $PMS = \{(m_3, 0.5), (m_5, 1.0), (m_8, 1.0), (m_9, 0.5)\}$
- Ahn's $PMS = \{(m_5, 0.5), (m_6, 1.0), (m_7, 1.0), (m_9, 1.0\}$
- Lee's $PMS = \{(m_1, 0.5), (m_8, 1.0), (m_{12}, 1.0), (m_{13}, 1.0\}$
- Kang's $PMS = \{(m_2, 1.0), (m_3, 1.0), (m_{10}, 1.0), (m_{11}, 0.5)\}$
- Chae's $PMS = \{(m_2, 0.5), (m_4, 1.0), (m_{11}, 1.0), (m_{12}, 0.5)\}$

With these PMSs and music feature database, the system calculates the distance between Kim and other users, and The result are as the follows:
d(Kim,Ahn) is 0.43, d(Kim,Lee) is 0.19, d(Kim,Kang) is 0.29, and d(Kim,Chae) is 0.25. If the number of the neighbors, L, were 3, the system would select Lee, Kang and Chae. Then the system forms Kim's neighborhood H as the follow:

H = {Lee, Kang, Chae}

The illustration of this is shown in Figure 5.

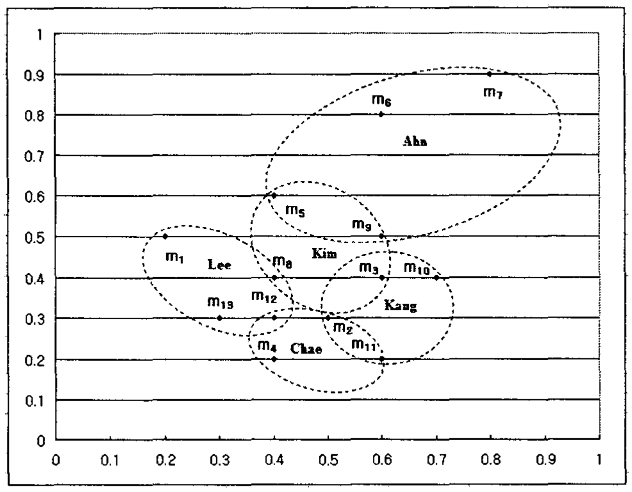


Figure 5 – Music and User in feature space

In Figure 5, interestingly, Chae didn't prefer the common music, but she could be Kim's neighbors. Chae didn't any common musics, but she can be Kim's neighbor. That is, Chae would be the hidden neighbors described in Section 2.1. On the other hand, Ahn purchased the common musics, but he is included in the H.
If the system followed the general CF system's method, the result is quite different. In general CF, when the system calculates the similarity between Kim and other users with Cosine formulation, the result is the follow:
Cos(Kim, Ahn) is 0.5, Cos(Kim, Lee) and Cos(Kim, Kang) is 0.25, and Cos(Kim, Chae) is zero. Therefore the system can't find the hidden neighbor - Chae, and select the unsimilar user - Ahn as Kim's neighbor.

Before calculation the PLS, the system should convert the distance to the similarity. In Section 3.4, we define the similarity function, which is normalized with the maximum and minimum distance. But in this example the number of neighbors is too small, therefore the similarity isn't accurate.

To solve this difficulty, we set the maximum 0.5, and the minimum 0.01. With this maximum and minimum value, we calculate the similarity of neighbors-Lee, Kang, Chae, and the result is the follows:
Sim(Kim, Lee) is 0.6, sim(Kim, Kang) is 0.4, and sim(Kim, Chae) is 0.5.

To generate the recommendation list R, we select the music having the high lank among musics in the neighbors' PMSs. Basically CF system excepts the music which the target user already purchased, so the considered musics are music $m_1$, $m_2$, $m_4$, $m_{10}$, $m_{11}$, $m_{12}$, and $m_{13}$. PLS(Kim, $m_1$) is 0.23, PLS(Kim, $m_2$) is 0.48, PLS(Kim, $m_4$) is 0.77, PLS(Kim, $m_{10}$) is 0.73, PLS(Kim, $m_{11}$) is 0.52, PLS(Kim, $m_{12}$) is 0.52, and PLS(Kim, $m_{13}$) is 0.73. If the number of the recommended music, N, were 5, finally the recommendation list R is composed of the musics as the follow:

R={$m_4$, $m_{10}$, $m_{11}$, $m_{12}$, and $m_{13}$}

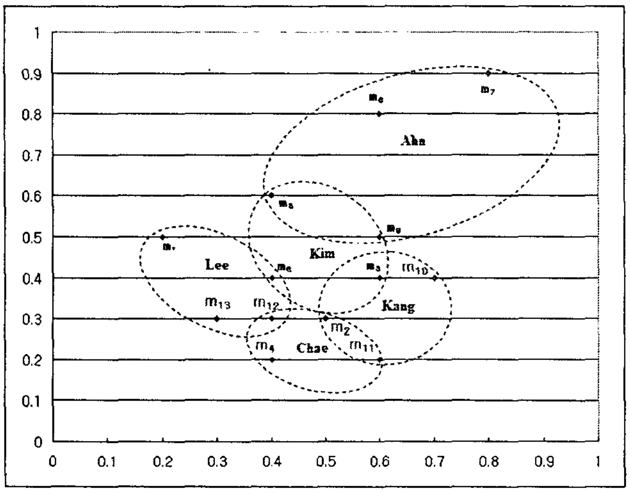


Figure 6 - The recommended musics

Figure 6 shows the recommended musics in feature space. The recommended musics are close to Kim's PMS. Look at the $m_2$, which is closest among musics in neighbors' PMS, but it is recommended. Because few members prefer the $m_2$, and its rating is relatively low, therefore its score drops.

### 4.2 The new music recommendation

Before the new music recommendation procedure, let's check the neighbors are really similar to the target user Kim. Table 2 shows the weighted mean of each feature. Lee is similar in the feature 1 and 2, Kang is in the feature 2, and Chae is in the feature 1 and 2. Actually it seems different the Figure 6, but it is the result reflected the each user's preference. Therefore if you considered its music's rating information, then you can understand better.

Table 2 – The mean of each feature

| Feature / User | 1 | 2 |
|---|---|---|
| Kim | 0.35 | 0.36 |
| Lee | 0.3 | 0.31 |
| Kang | 0.525 | 0.3 |
| Chae | 0.39 | 0.3 |

If the new music $m_{14}$, $m_{15}$ and $m_{16}$ came to the market, we should decide whether to recommend them or not. The table 3 shows their feature profile.

Table 3 - New music's profile

| Feature / Music | 1 | 2 |
|---|---|---|
| $m_{14}$ | 0.55 | 0.3 |
| $m_{15}$ | 0.35 | 0.35 |
| $m_{16}$ | 0.4 | 0.8 |

In this example, all of the users didn't prefer over 30. Therefore with the equation (7) we determine the preferred boundary. The $M_k$ is already shown in Table 2 and the standard deviation $\sigma$ is show in the Table 3.

Table 4 – The standard deviation of each feature

| Feature / User | 1 | 2 |
|---|---|---|
| Kim | 0.06 | 0.18 |
| Lee | 0.14 | 0.06 |
| Kang | 0.17 | 0.14 |
| Chae | 0.15 | 0.09 |

With the t-value, we can determine each user's preferred boundary, is shown in Table 5. In 95% confident interval, all of users' degree of freedom is 3, then t-value is 3.182.

Table 4 - Preferred boundary

| Feature / User | 1 | 2 |
|---|---|---|
| Kim | (0.26,0.44) | (0.26~0.65) |
| Lee | (0.07~0.53) | (0.21~0.41) |
| Kang | (0.25~0.8) | (0.07~0.53) |
| Chae | (0.14~0.63) | (0.16~0.44) |

Let's check that the new music $m_{14}$, $m_{15}$, and $m_{16}$ are included in any members' preferred boundary in C. The important thing is that the new music has to be in all of the features' preferred boundaries. $m_{14}$ is included in Kang and Chae, and $m_{15}$ is in Kim, Lee, Kang, and Chae but $m_{16}$ isn't in anyone's boundary. Therefore we decide to recommend $m_{14}$, $m_{15}$.

Next, we search the MSM in the PMS where the new music is included in. So we search the $m_{14}$'s MSM in Kang and Chae's PMS.
First we search the $m_{14}$'s MSM in where it is included. Based on the distance, $m_2$ and $m_{11}$ is $m_{14}$'s MSM for Kim

and Chae, respectively. The rating about $m_2$ and $m_{11}$ is both 1.0, so $m_{14}$ is given this. As the same way, $m_{15}$'s MSM can be calculated.
Second, we calculate the $m_{14}$ and $m_{15}$'s PLS, and the result is 0.75. With the same way, we compute the $m_{15}$'s PLS, which is 0.753.
Finally, we add these PLS value to the recommendation list described in Section 4.1. And we already set L to be 5, so the final recommendation list for Kim is as the follow:

$$R = \{m_4, m_{15}, m_{14}, m_{10}, m_{13}\}$$

## 5. Conclusion

In this paper we propose the extended CF recommender system, which can recommend music more accurately, and new music. The proposed system represents the all musics in the feature space, and form the individual user's PMS with his purchasing history. The system regard a user's PMS as a cluster, compute the distance between the target user and other users, and find his neighbors. This can identify the real neighbors and find the hidden neighbors.
Moreover, the system represents the new musics in the space, and searches the MSM for the new music. With the MSM's rating, the system reorganizes the recommendation list.
And we made an example, which can show step by step how the system do find target user's neighbors, and how can recommend new musics. In this example, we verify our system is more accurate than the general CF, and generate the recommendation list including both new musics and other existing musics.
We hope that this system can find the real neighbors, recommend more accurately in mobile web environment, and the performance of proposed method for new music recommendation works better than the general CF system.

## References


[1] Burke R., (2002), *Hybrid recommender systems: Survey and experiments*, User Modeling and User-Adapted Interaction.

[2] Cho Y.H., J.K. Kim, (2004), *Application of web usage mining and product taxonomy to collaborative recommendation in e-commerce*, Expert systems with application.

[3] Everitt B.S., S. Landau, M. Leese, (2001), Cluster Analysis", Oxford University Press.

[4] Goto M., Y. Muraoka, (1994), *A beat tracking system for acoustic signals of music*, ACM mulmimedia, pages 365-372.

[5] Kim, C.Y., J.K. Lee, Y.H. Cho, D.H. Kim, (2004), *VISCORS: a Visual Contents Recommender System on the Mobile Web*, IEEE Intelligent Systems, Special issue on Mining the Web for Actionable Knowledge, Accepted for publication.

[6] Kim J.K, Y.H. Cho, (2003), *Using Web Usage Mining and SVD to Improve E-commerce Recommendation Quality*, LNCS (Lecture Notes Computer Science): Intelligent Agents and Multi-Agent Systems, pp 86-97.

[7] Korea IT industry Promotion Agency, (2003), *The Digital Contents Market Report*.

[8] Lambrou, P. Kudmakis, R. Speller, M. Sandler, A. Linney, (1998), *Classification of audio signals using statistical features on time and wavelet transorm domains*, In Proc. Int. Conf. Acoustic, Speech and Signal processing, Vol.6, pages 3621-3621

[9] Pazzani M.J., (1999), *A framework for collaborative, Content-Based and Demographic Filtering*, Artificial Intelligence Review.

[10] Rabiner L., B.H. Juang, (1993), *Fundamentals of Speech recognition*", Prentice-Hall, NJ.

[11] Sarwar B.M., G. Karypis, J.A. Konstan, J.T. Riedl, (2001), *Application of dimensionality reduction in recommender system - a case study*, ACM, 2001.

[12] Sarwar B.M., G. Karypis, J.A. Konstan, J.T. Riedl, (2000), *Analysis of Recommendation Algorithms for E-Commerce*, In Proceedings of ACM Conference on Electronic Commerce.

[13] Tzanetakis G., (2000), *MARSYAS: A framework for audio analysis*, Organized sound, Vol.4(3).

[14] Tzanetakis G., (2002), *Musical genre classification of audio signals*, IEEE Transactions on Speechand Audio Processing,.

[15] Uitdenbogerd A. L., Justin Zobel, (1998), *Manipulation of Music for Melody Matching*, Proceeding of the sixth ACM International Conference on Multimedia.

[16] Lambrou, P. Kudmakis, R. Speller, M. Sandler, A. Linney, (1998), *Classification of audio signals using statistical features on time and wavelet transorm domains*, In Proc. Int. Conf. Acoustic, Speech and Signal processing, Vol.6, pages 3621-3621