

Music Recommendation Based on Acoustic Features and User Access Patterns

Bo Shao, Dingding Wang, Tao Li, and Mitsunori Ogihara

Abstract—Music recommendation is receiving increasing attention as the music industry develops venues to deliver music over the Internet. The goal of music recommendation is to present users lists of songs that they are likely to enjoy. Collaborative-filtering and content-based recommendations are two widely used approaches that have been proposed for music recommendation. However, both approaches have their own disadvantages: collaborative-filtering methods need a large collection of user history data and content-based methods lack the ability of understanding the interests and preferences of users. To overcome these limitations, this paper presents a novel dynamic music similarity measurement strategy that utilizes both content features and user access patterns. The seamless integration of them significantly improves the music similarity measurement accuracy and performance. Based on this strategy, recommended songs are obtained by a means of label propagation over a graph representing music similarity. Experimental results on a real data set collected from <http://www.newwisdom.net> demonstrate the effectiveness of the proposed approach.

Index Terms—Dynamic audio similarity, music recommendation, user access patterns.

I. INTRODUCTION

A. Music Recommendation

WITH the advancements of the web technologies, there is a dramatic increase in online music stores and services. Music is now more pervasive than ever, and listeners nowadays have easier access to the tremendous online music data. This significantly increases the difficulty in the effective and accurate selection of music pieces, which raises the requirements of better music recommendation approaches. Music recommendation is the procedure of providing a music listener a list of music pieces that he/she is likely to enjoy listening to. Music recommendation should base on a good understanding of the user preference and the music pieces in the collection. Therefore, the key to a success music recommendation is to develop a good measurement strategy of the music similarity and an effective recommendation method based on the similarity measurement. Our

Manuscript received July 16, 2008; revised February 06, 2009. Current version published September 04, 2009. The work of T. Li was supported in part by the National Sciences Foundation under Grant IIS-0546280 and in part by IBM Faculty Research Awards. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Sylvain Marchand.

B. Shao, D. Wang, and T. Li are with the School of Computer Science, Florida International University, Miami, FL 33199 USA (e-mail: bshao001@cs.fiu.edu; dwang003@cs.fiu.edu; taoli@cs.fiu.edu).

M. Ogihara is with the Department of Computer Science, University of Miami, Coral Gables, FL 33146, USA (e-mail: ogihara@cs.miami.edu).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TASL.2009.2020893

goal for the music recommendation is to satisfy the following two requirements.

- *High Recommendation Accuracy*: A good recommendation system should output a relatively short list of songs in which many pieces are favored by the user and few pieces are not.
- *High Recommendation Novelty*: Good novelty is defined as rich artist variety and well-balanced music content variety. Music content represents the information of genre, timbre, pitch, rhythm, and so on [37]. Well-balance means that the music content is diverse and informative while not diverging much from the user's preferences.

Various music recommendation approaches have been developed, and user demographic information, music contents, user listening history, and the discography (e.g., Last.fm, Goombah, and Pandora) have been used for music recommendations [3], [4], [22], [26]–[29], [38]. These approaches can be generally divided into two groups: collaborative-filtering methods and content-based methods.

Collaborative-filtering methods recommend songs by identifying similar users or items based on ratings of items given by users [1], [5], [14]. If the rating of an item by a user is unavailable, collaborative-filtering methods estimate it by computing a weighted average of known ratings of the items from similar users. Thus, for collaborative-filtering methods to be effective, large amount of user-rating data are required. This is a major limitation [33], [34]. *Content-based methods* provide recommendations based on the meta-data such as genre, styles, artists, and lyrics [28], [31], [41], and/or the acoustic features extracted from audio samples [15], [17], [19], [20]. Since acoustic contents are susceptible to feature extraction, music recommendation is considered different from movie recommendation, in which meta-data is generally the only available information [24]. In music recommendation, the reflective and consistent acoustic features can represent song-specific characteristics such as genre, timbre, pitch, and rhythm. Comparing with the acoustic features, a large portion of meta-data are the descriptions of contents given by musicians. Music meta-data are thus very time-consuming to obtain and not capable of providing adequate information for describing listeners' preferences [19].

Recently probabilistic models and hybrid algorithms [16], [30], [41] have been proposed to overcome the aforementioned limitations by combining contents and user ratings. Yoshii *et al.* [41] attempted to integrate both rating and content data. They utilized Bayesian network to statistically estimate the probabilistic relations over users, ratings and contents. Popescu *et al.* [30] proposed a probabilistic model similar to the one suggested by Yoshii *et al.* to take advantage of both collaborative-filtering and content-based recommendations. Jung *et al.*

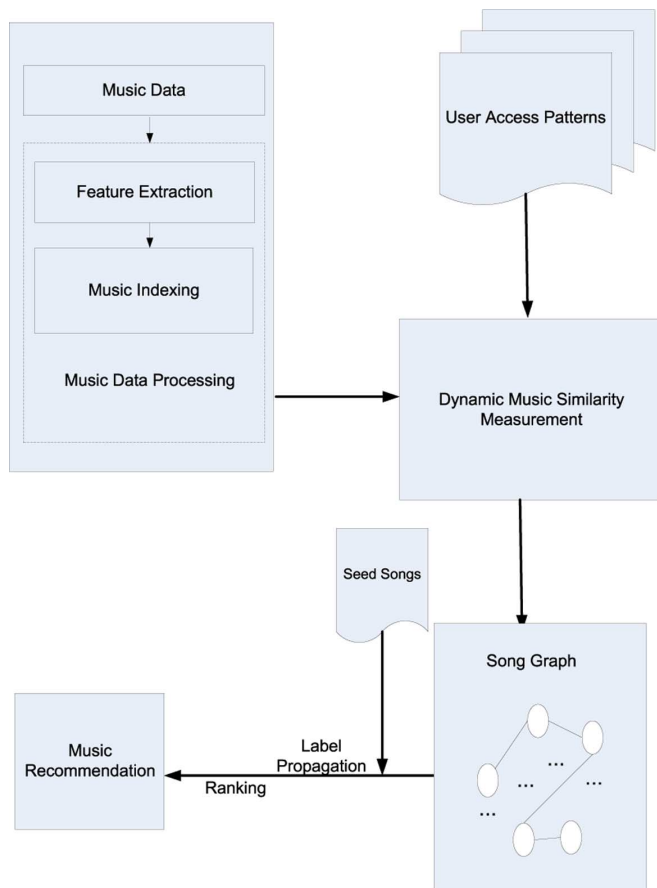


Fig. 1. Framework of the proposed music recommendation approach.

[16] designed a hybrid method that combines collaborative-filtering and content-based methods to improve recommendation performance. However, these models and methods significantly degraded when they were short of corresponding user access data as illustrated in our experiments later in this paper.

B. Contributions of the Paper

This paper proposes a music recommendation approach by incorporating collaborative-filtering and acoustic contents of music. This approach employs a novel dynamic music similarity measurement strategy, which significantly improves the similarity measurement accuracy and efficiency. This measurement strategy utilizes the user access patterns from large numbers of users and represents music similarity with an undirected graph. Recommendation is calculated using the graph Laplacian and label propagation defined over the graph.

Fig. 1 shows the framework of our proposed music recommendation system. First music data and user access patterns are collected and preprocessed. Then dynamic music similarity measurement is then used to compute the similarities between pairs of songs and construct the song graph. Finally, when seed songs are given, label propagation and ranking are performed for music recommendation. In the rest of the paper, we call our recommendation approach as DWA since it utilizes dynamic weighting scheme based on user access patterns.

The proposed DWA approach is tested through experiments on a real data set constructed by anonymous users at <http://www.newwisdom.net>

and has been adopted for music recommendation at that website.

C. Organization of the Paper

The rest of the paper is organized as follows. Section II presents the dynamic similarity measurement in detail. Section III describes the feature extraction and music indexing methods that are used in this approach. Section IV explains the method for obtaining recommended list of music based on the similarity measurement. Section V depicts the experiments that were conducted. Finally, Section VI draws the conclusion and discusses possible future work.

II. DYNAMIC MUSIC SIMILARITY MEASUREMENT

A. Audio Similarity

Extraction of audio features for music similarity search has been well studied in the literature [10], [21], [23]. The use of acoustic features is justified by the fact that similar music pieces use similar instruments and possess similar sound textures [8].

The music features are vectors in a multidimensional space, and the distance between the representation vectors characterizes and quantifies the closeness between two pieces of music. Traditionally, there are two popular distance functions for measuring similarity in multimedia retrieval [9], [23], [32]: (weighted) Minkowski distance and cosine similarity. The assumption of using Minkowski distance function is that the similar objects should be close in all dimensions as all the dimensions are treated equally. For weighted Minkowski distance, weights are introduced to identify important features. Thus, the weighted Minkowski distance function is based on the static weighting scheme that assumes similar songs should be close in the same way (w.r.t to the same set of weights). The cosine similarity is computed as the cosine of the angle between two input vectors. Although both distance functions have been previously used in music retrieval, they have the following two drawbacks.

- *Uniform Weights for Acoustic Features:* In the Minkowski distance measurement, every audio feature is assigned with the equal weight when determining the similarity of music. This could be inappropriate given that people might be more sensitive to certain acoustic features than the others. This problem is further complicated when feature weights vary from one type of music to another. For example, for Rock, the audio intensity is an important feature in determining music similarity while it becomes a much less important feature for classic music. Thus, it is essential to assign dynamic weights to different acoustic features.
- *Subjective Perception of Music:* It is well known that the perception of music is subjective to individual users. Different users can have totally different opinions for the same pieces of music. Using a fixed set of weights for acoustic features is likely to fail in accounting for the taste of individual users. It is thus important to assign different weights to audio features based on the taste of individual users.

To address the above two issues, we propose a novel dynamic similarity measurement scheme. This scheme utilizes the access patterns of music from a considerable number of users. It is

TABLE I
EXAMPLE OF USER ACCESS PATTERNS

	m_1	m_2	m_3	m_4
u_1	1	1	0	0
u_2	1	1	0	0
u_3	0	0	1	1
u_4	0	0	1	1

based on the assumption that two pieces of music are similar in human perception when they share similar access patterns across multiple users. Table I illustrates the assumption. This table shows a toy example of user access patterns on four pieces of music by four different users. In this Table I represents that the music piece is accessed by the corresponding user while 0 indicates not. It is clear that m_1 and m_2 are similar from the user's point of view because they are accessed by users u_1 and u_2 , but not by users u_3 and u_4 . Also, m_3 and m_4 are similar to each other in that they are accessed by users u_3 and u_4 , but not by u_1 and u_2 . Similar ideas have been successfully applied to image retrieval to improve the accuracy of similarity measurement [12], [13], [25].

B. Dynamic Weighting Schemes

1) *Introduction:* A simple approach capable of combining acoustic features and user access patterns for similarity measurement is to compute the similarity based on each representation and then combine the two similarity measurements linearly. By incorporating the user access patterns of music, the combined similarity measurement can more accurately reflect human perception of music than the one based only on acoustic features. A major drawback with such an approach is that user access patterns are usually sparse. Only for a relatively small number of pieces of music, their user access data are adequate to provide robust estimation of similarity with other pieces of music. This drawback will substantially limit the impact of the use of user access patterns. Also, since the approach uses the Minkowski distance for the audio-based similarity calculation, it does not provide a means for estimating the weights on acoustic features, the essential components in making similarity measurement that is both genre-dependent and user-dependent.

2) *Problem Formulation:* Thus, the calculation of appropriate similarity measures can be cast as a learning problem aimed to assign approximate weights to each feature [39]. To automatically determine the weights for audio features, the metric learning approach [13], [40], which learns appropriate similarity metrics based on the correlation between acoustic features and user access patterns of music, needs to be explored. Given that human perception of music is well approximated by its user access patterns, a good weighting scheme for acoustic features should lead to a similarity measurement that is consistent with the one based on user access patterns. Let $m_i = (\mathbf{a}_i, \mathbf{u}_i)$ denote the i th piece of music in the data set, where \mathbf{a}_i and \mathbf{u}_i represent its acoustic features and user access patterns, respectively. Let $S_a(\mathbf{a}_i, \mathbf{a}_j; \mathbf{w}) = \sum_l a_{i,l} a_{j,l} w_l$ be the sound-based similarity measurement between the i th and the j th pieces of music when the parameterized weights are given by \mathbf{w} . Let $S_u(\mathbf{u}_i, \mathbf{u}_j) = \sum_k u_{i,k} u_{j,k}$ be the similarity measurement be-

tween the i th and j th pieces of music based on their user access patterns. Here for each k , $u_{i,k}$ denotes whether the k th user accesses the i th piece of music. To learn appropriate weights \mathbf{w} for audio features, we can enforce the consistency between similarity measurements $S_a(\mathbf{a}_i, \mathbf{a}_j; \mathbf{w})$ and $S_u(\mathbf{u}_i, \mathbf{u}_j)$. The above idea leads to the following optimization problem:

$$\begin{aligned} \mathbf{w}^* &= \arg \min \sum_{i \neq j} (S_a(\mathbf{a}_i, \mathbf{a}_j; \mathbf{w}) - S_u(\mathbf{u}_i, \mathbf{u}_j))^2 \\ \text{s.t. } \mathbf{w} &\geq 0. \end{aligned} \quad (1)$$

Let p be the number of content features. The summation in (1) is rewritten as follows:

$$\begin{aligned} &\sum_{i \neq j} (S_a(\mathbf{a}_i, \mathbf{a}_j; \mathbf{w}) - S_u(\mathbf{u}_i, \mathbf{u}_j))^2 \\ &= \sum_{i \neq j} \left(a_{i,1} a_{j,1} w_1 + \cdots + a_{i,p} a_{j,p} w_p - \sum_k u_{i,k} u_{j,k} \right)^2 \\ &= \sum_{i \neq j} \left((a_{i,1} a_{j,1} w_1 + \cdots + a_{i,p} a_{j,p} w_p)^2 \right. \\ &\quad \left. - 2(a_{i,1} a_{j,1} w_1 + \cdots + a_{i,p} a_{j,p} w_p) \right. \\ &\quad \left. \times \left(\sum_k u_{i,k} u_{j,k} \right) + \left(\sum_k u_{i,k} u_{j,k} \right)^2 \right) \end{aligned}$$

where $a_{i,l}$ is l th feature in the acoustic feature set a_i and $a_{j,l}$ is l th feature in the acoustic feature set a_j . Let n be the number of pieces of music, and let

$$A = \begin{bmatrix} a_{1,1} a_{2,1} & a_{1,2} a_{2,2} & \cdots & a_{1,f} a_{2,f} \\ & \cdots & \cdots & \\ a_{n-1,1} a_{n,1} & a_{n-1,2} a_{n,2} & \cdots & a_{n-1,f} a_{n,f} \end{bmatrix}$$

and

$$U = \begin{bmatrix} \sum_{i \neq j} a_{i,1} a_{j,1} \left(\sum_k u_{i,k} u_{j,k} \right) \\ \vdots \\ \sum_{i \neq j} a_{i,f} a_{j,f} \left(\sum_k u_{i,k} u_{j,k} \right) \end{bmatrix}$$

where A is a $(C_n^2 \times p)$ matrix and U an $(p \times 1)$ matrix. Thus, (1) is equivalent to

$$\begin{aligned} \mathbf{w}^* &= \arg \min \left[\frac{1}{2} \times 2(A\mathbf{w})^T(A\mathbf{w}) - U^T \mathbf{w} \right] \\ &= \arg \min \left[\frac{1}{2} (\mathbf{w}^T (2A^T A) \mathbf{w} + (-2U^T) \mathbf{w}) \right] \\ \text{s.t. } \mathbf{w} &\geq 0. \end{aligned} \quad (2)$$

This optimization problem can be addressed using quadratic programming techniques [11].

3) *Discussions:* A similar strategy can be applied to make the similarity measurement dependent on the preferences of individual users. This is accomplished by selecting a subset of users whose access patterns are similar to those of the active users and then use only those selected in the estimation of music similarity. In other words, the quantity $S_u(\mathbf{u}_i, \mathbf{u}_j)$ in (1) is estimated only based on those users that are deemed similar. An important issue in employing such an approach is the method and the cost of selecting similar users. One possibility is to use the min-wise

hash indexing scheme (to be discussed in Section III-B), in which a set of t independent hash functions are applied to each component of the user access pattern vector, which is of dimension n and the minimum of the t values is chosen as the hash value of each component. Then two representations are compared for similarity by simply counting how many components have the same hash value. By applying a simple threshold to the count, similar users can be selected. The time that it takes to compute similarity is $O(n)$ for each pair of users, assuming that the hash values have been already computed. Therefore, the selection of similar users to the active user requires time $O(nm)$, where m is the number of users. This possibility is not explored here in this paper since the number m of the dataset is small.

III. MUSIC FEATURE EXTRACTION AND INDEXING

A. Feature Extraction

There has been a considerable amount of research in extracting descriptive features from music signals for music genre classification and artist identification [10], [21], [23], [37].

In this paper, we employ timbral features and wavelet coefficient histograms for feature extraction. The extracted feature set consists of the following three components and total 80 features.

1) *Mel-Frequency Cepstral Coefficients*: Mel-Frequency Cepstral Coefficients (MFCCs) is a feature set that is highly popular in speech processing. It is designed to capture short-term spectral-based features. The features are computed as follows: First, for each frame, the logarithm of the amplitude spectrum based on short-term Fourier transform is calculated, where the frequencies are divided into thirteen bins using the Mel-frequency scaling. Next, this vector is then decorrelated using discrete cosine transform. This is the MFCC vector. In this study, the first five bins are selected, and the mean and variance of each over the frames are then computed.

2) *Short-Term Fourier Transform Features (STFT)*: This is a set of features related to timbral textures and is not captured using MFCC. It consists of the following five types of features: spectral centroid, spectral rolloff, spectral flux, zero crossings, and low energy. More detailed descriptions of STFT can be found in [37].

Spectral Centroid is the centroid of the magnitude spectrum of short-term Fourier transform and is a measure of spectral brightness. *Spectral Rolloff* is the frequency below which 85% of the magnitude distribution is concentrated. It measures the spectral shape. *Spectral Flux* is the squared difference between the normalized magnitudes of successive spectral distributions. It measures the amount of local spectral change. *Zero Crossings* is the number of time domain zero crossings of the signal. It measures noisiness of the signal. *Low Energy* is the percentage of frames that have energy less than the average energy over the whole signal. It measures amplitude distribution of the signal. We compute the mean for all five types and the variance for all but zero crossings.

3) *Daubechies Wavelet Coefficient Histograms (DWCH)*: Daubechies wavelet filters are a set of filters that are widely used in image retrieval (see [6]). Daubechies Wavelet Coefficient Histograms, proposed in [21], are features extracted in the following manners: first, the Daubechies-8 (db_8) filter with

seven levels of decomposition (or seven subbands) is applied to 30 s of monaural audio signals; then, the histogram of the wavelet coefficients is computed at each subband; after that, the first three moments of a histogram, i.e., the average, the variance, and the skewness, are calculated from each subband; in addition, the subband energy, defined as the mean of the absolute value of the coefficients, is computed from each subband. More details of DWCH can be found in [21].

B. Music Indexing

Once the features/signatures for each song are obtained, efficient data structures can be built for similarity search. In this study, min-wise hashing [2] is used to speed up similarity computation for large data sets, especially in online calculation. The key idea is that we can create a small signature for each song and the resemblance of any pair of songs s_i and s_j can be accurately estimated based on their min-wise hashing signatures.

The min-wise hashing signature is computed as follows. Given a signature of size r , r independent random hash functions f_1, \dots, f_r are firstly generated. For a song s_i (s_i is the feature set of song i), the t th component of its signature is given by

$$\min\{f_t(d) \mid d \in s_i\}.$$

where d represents any feature in the feature set.

In doing so, the minimal hash value in s_i for the t th hash function f_t is reserved. Note that the same hash function f_t is used for every song to generate its t th signature component. Let S^i and S^j be the signatures of s_i and of s_j thus obtained, respectively. Let S_t^i and S_t^j be the t th components of S^i and S^j . We say that they match at t if $S_t^i = S_t^j$. The resemblance between s_i and s_j can be then measured by the proportion of the number of matches between S^i and S^j to r , the number of components.

The min-wise hashing estimator is unbiased. An error bound was given in [2] and the accuracy increases with the resemblance value. Note that the number of matches between two signatures can be computed in $O(r)$ time and that r is independent of the size of database.

IV. MUSIC RECOMMENDATION OVER SONG GRAPH

In the previous section, we described the acoustic feature extraction and presented an efficient method to compute the similarities between pairs of songs. We are now ready to construct the song graph.

A. Song Graph

Definition 1 (Song Graph): A song graph is an undirected weighted graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where:

- 1) $\mathcal{V} = \mathcal{I}$ is the node set (\mathcal{I} is the song set, which means that each song is represented as a node on the graph \mathcal{G});
- 2) \mathcal{E} is the edge set. Associated with each edge $e_{pq} \in \mathcal{E}$ is the similarity w_{pq} , which is nonnegative and satisfies $w_{pq} = w_{qp}$.

Once the song graph is constructed, music recommendation can be treated as a label propagation from labeled data (i.e., items with ratings) to unlabeled data. In its simplest form, the

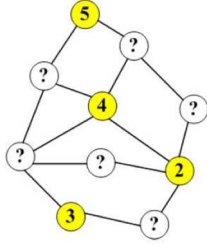


Fig. 2. Illustration of a recommendation task. The colored (shaded) nodes represent the rated items with their corresponding ratings. The others are the unrated items, whose ratings are unknown.

label propagation is like a random walk on a song graph \mathcal{G} [36]. Using diffusion kernel [18], [35], the label propagation is like a diffusive process of the labeled information [42], [43]. Zhu *et al.* [43] utilizes the harmonic nature of the diffusive function, Zhou *et al.* [42] emphasize the spread of label information in a consistent and iterative way. Motivated from the previous research, we emphasize the global and coherent nature of label propagation and use the Green's function of the Laplace operator for music recommendation [7].

B. Label Propagation on Graph

Given a graph with edge weights T , the *combinatorial Laplacian* is defined to be $L = D - T$, where D is the diagonal matrix consisting of the row sums of W ; i.e., $D = \text{diag}(Te)$, $e = (1 \cdots 1)^T$.

Green's function is defined on the generalized eigenvectors of the Laplacian matrix

$$L\mathbf{v}_k = \zeta_k D\mathbf{v}_k, \quad \mathbf{v}_p^T D\mathbf{v}_q = \mathbf{z}_p^T \mathbf{z}_q = \delta_{pq} \quad (3)$$

where $0 = \zeta_1 \leq \zeta_2 \leq \cdots \leq \zeta_n$ are the eigenvalues and the zero-mode is the first eigenvector $\mathbf{v}_1 = \mathbf{e}/\sqrt{n}$. Then we have

$$G = \frac{1}{(D - T)_+} = \sum_{k=2}^n \frac{\mathbf{v}_k \mathbf{v}_k^T}{\zeta_k}. \quad (4)$$

In practice, the expansion after some K terms is truncated and the K vectors are stored. Green's function is computed on the fly. Therefore, the storage requirement is $O(Kn)$.

The recommendation on the song graph is illustrated in Fig. 2. Let $\mathbf{y}^T = (y_1, \dots, y_n)$ be the rating for a user. Given an incomplete rating $\mathbf{y}_0^T = (5, ?, ?, 4, 2, ?, ?, ?, 3)$, the question is to predict those missing values. Using Green's function, we initialize $\mathbf{y}_0^T = (5, 0, 0, 4, 2, 0, 0, 0, 3)$, and then compute the complete rating as the linear influence propagation

$$\mathbf{y} = G\mathbf{y}_0 \quad (5)$$

where G is the Green function built from the song graph.

C. Music Ranking

After label propagation, the ratings for unrated songs are obtained and many of them might have the same rating. In practice, a ranked list of the items to be recommended is required.

The music ranking over a song graph \mathcal{G} can be treated as the problem of finding the shortest path from the seed song node to the rest of the nodes in the song graph. The edges with low similarity have already been eliminated, so only the remaining edges can be used to construct shortest paths. For any $M \geq 1$, to recommend M songs after a seed song s , we simply select the M songs that are the closest to s . The standard single-source shortest-path algorithm produces the shortest path to any node in time $O(|\mathcal{V}|^2 + |\mathcal{E}|\log|\mathcal{V}|)$ where $|\mathcal{V}|$ is the number of nodes and $|\mathcal{E}|$ is the number of edges in the graph. The time that it takes for identifying M closest nodes after the shortest path length is obtained can be $O(M|\mathcal{V}|)$.

V. EXPERIMENTS AND EVALUATION

In this section, we present the performance evaluation of our music recommendation system, including effectiveness and novelty analysis. Various case studies and the user study show the promising recommendation quality of our system.

A. Data Collection

The music data were collected from <http://www.newwisdom.net>. It is a website in Chinese language with major functions of education and entertainment. This website has approximately 6000 registered users visiting its forums regularly. These users also listen to music and meanwhile create their own favorite playlists (called CDs on this website). Currently the website has a collection of more than 10 000 songs and hundreds of playlists. More than 80% of songs were from famous Chinese artists, others were from famous American, European, Japanese, and Korean artists. The songs covered many different genres including pop, classic, jazz, rock, country and hip-hop.

In the experiments described next, we sampled 2829 songs from the playlists created by "serious" users in the same group on the website. The criterion for a "serious" user is the number of songs in his/her playlists. We eliminated those whose playlists containing either less than 10 or more than 20 songs from the data collection. Those users are assumed to be either "too uninterested" or "too eager." and then defined not "serious." This culling process leaves us 274 playlists.

B. Data Processing

We process the collected songs and user playlists to get the content features and user access patterns. Then our dynamic weighting scheme and music ranking algorithm are applied to generate the recommendation identifications of music pieces.

1) *Acoustic Feature Representation*: For each song, a music sample using the third 30-s block (i.e., between time 1'00" and 1'30") is generated, given the songs in our test domain tend to have introductory nonvocal part in the first 60 s. Then the content features of the 30 second block are extracted using the approach described in Section III-A. After feature extraction, each music track is represented as a 80-dimensional feature vector: $F_i = (F_{i,1}, \dots, F_{i,80})$. As described in Section III-A, the first 12 features are based on the magnitude of the STFT (e.g., means and variances of spectral centroid, rolloff, flux, zero crossings, and low energy), the next 52 features represents the means and variances of MFCCs, and the last 16 features are DWCH features.

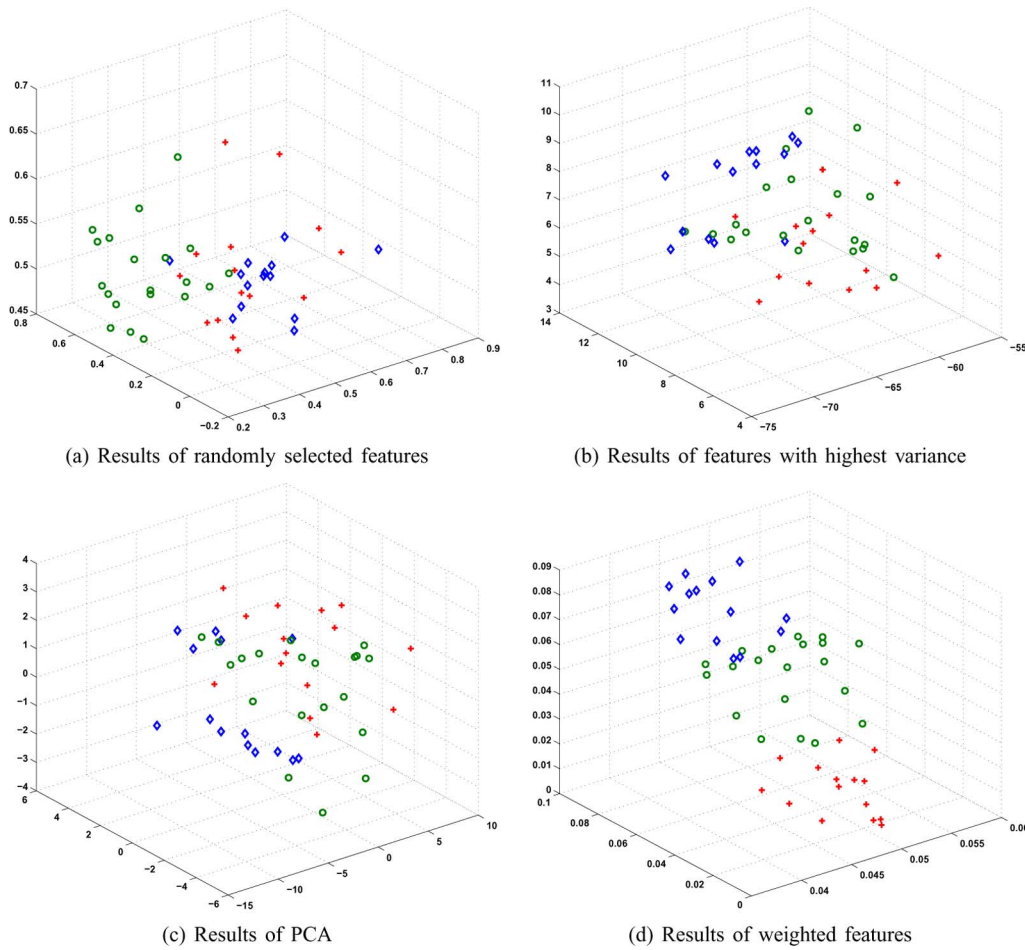


Fig. 3. Evaluation on weighting schemes.

2) *User Access Pattern Representation*: The access pattern of a user is represented as a 0/1-vector. Its dimension is equal to the number of songs available. For each i , the i th entry of the vector is 1 if the user added the song in his/her playlist and 0 otherwise.

3) *Recommendation List Generation*: By combining the user access pattern data with the content features of the songs, the weight is generated for each feature using the dynamic weighting scheme described above. Then the music ranking algorithm aforementioned is employed to output the desired number of music pieces as our recommendations. In the experiments, the values of the ratings for the seed songs are set to be the same.

C. Evaluation on Dynamic Weighting Schemes

First of all, the performance of the dynamic weighting schemes is evaluated. In order to do so, we take a sample dataset consisting of 50 songs from three different classes. Note that the classes are determined by a group of users. Now we use the following methods to scatter positions of the 50 songs, and compare them in Fig. 3. Note that each subfigure visualize the grouping results of different methods where each shape (there are three shapes: diamond, circle, and star) represents a class of songs.

- 1) Randomly select three original content features and scattering the position of each song based on these features.
- 2) Choose three content features with highest variances and scattering positions of the 50 songs.
- 3) Use principal components analysis (PCA) to select three principal components associated with the largest eigenvalues of the covariance matrix.
- 4) Choose three features with the highest weights by the dynamic weighting scheme (DWA).

From Fig. 3, we observe that the dynamic weighting approach (DWA) outperforms the other feature selection methods in separating three groups of songs: the features selected by DWA are highly relevant to the grouping. It shows the fact that the features associated with the learned weighted from the user access patterns have the description power to distinguish the music pieces, while features with large variances or covariances does not help much in this case.

D. Comparison on Different Recommendation Approaches

To demonstrate the performance of DWA, we compare the performance of the following five approaches:

- **Content-Based Approach (CBA)** This is solely based on acoustic content features extracted from the pieces of songs.

TABLE II
RESULTS FOR ARTIST VARIETY COMPARISON. THE NUMBERS ARE ROUNDED
TO INTEGERS TO BE PRACTICALLY MEANINGFUL

Approach	CBA	APA	HA	DWA
Average Number of Artists	8	5	7	8

- **Artist-Based Approach (ABA)** This is solely based on artist, namely, it recommends songs only from the same artist.
- **Access-Pattern-Based Approach (APA)** This is based on user access patterns. It selects the top songs with the highest co-occurrence frequency in the same playlists with the input song. This can also be thought as the item-based collaborative filtering method.
- **Hybrid Approach (HA)** This is the approach explained in Section I. It tries to integrate the collaborative filtering method and content-based method based on the algorithms described in [16].
- **DWA** This is based on our approach, which first utilizes user access patterns to dynamically learn weights for each content features and then perform label propagation and ranking for music recommendation.

We conduct several sets of experiments to compare the performance of the listed approaches. The first two comparisons are designed to test the recommendation novelty and the playlist generation experiment is to examine the recommendation prediction ability, while the user study conducted is to assess the overall recommendation performance from the viewpoints of the end users.

1) *Artist Variety Comparison:* In this experiment, we evaluate how artist variety is achieved in different approaches. Since artist-based approach consider songs from the same artists, we only have to compare approach CBA, APA, HA, and DWA. For each of the 2829 songs, ten songs are chosen for the recommendation output. We count the number of distinct artists that the ten songs come from. From the statistical results listed in Table II, we can see that content-based approach and our dynamic-weighting approach recommend songs with the richest artist variety, which is better than the hybrid approach and the access-pattern-based approach.

2) *Content Variety Comparison:* In this experiment, we evaluate if content variety as described in I are well balanced in different approaches.

First of all, we cluster the 2829 songs using K-means algorithm according to their content features, and then, we study how many clusters the ten songs recommended by each approach belong to. Also, we calculate the average distance among the ten recommended songs of each of the 2829 seed songs using their content features. The more the clusters and/or the larger the distances, the more diverse the ten songs, i.e., the more opportunity to get novel recommendation results.

From the experimental results listed in Table III, we can clearly observe that content-based approach recommends songs with the highest content similarity, and the variety is very low. On the contrary, the access-pattern-based approach and the artist-based approach are diverse enough but lack of content

TABLE III
RESULTS FOR CONTENT VARIETY COMPARISON

Approach	Mean of Average Distance	Mean of Average Number of Clusters
CBA	2.55	2
ABA	8.74	5
APA	10.01	6
HA	5.28	4
DWA	5.88	4

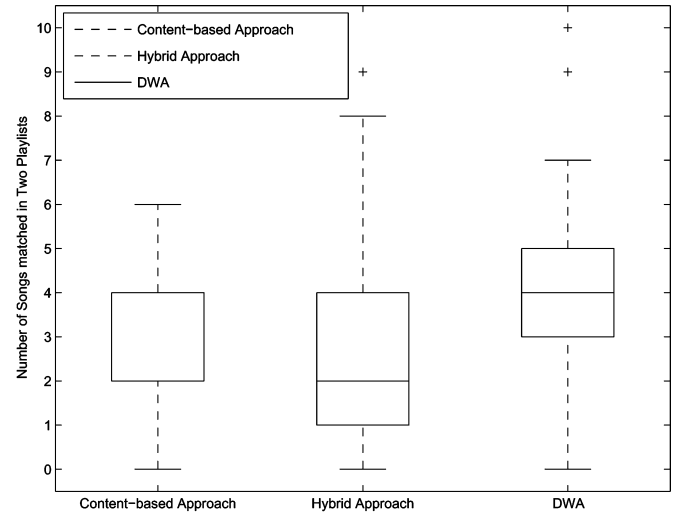


Fig. 4. Number of songs matched in user playlists and the playlists generated by different approaches.

similarity. Hybrid approach and our dynamic-weighting approach have comparable performance in well-balancing the content variety.

3) *Playlist Generation Comparison:* Since playlists are generally a good means to reflect the interests of users, by comparing how accurate we can generate the whole original playlists from part of songs in them using different methods, we can analyze the ability of the approaches to predict the interests and preferences of the users.

In this set of experiments, we randomly select 200 playlists from the dataset of 274 playlists, and run hybrid approach and our dynamic-weighting approach on the data for the two approaches to learn. Then we randomly select five songs from each of the rest 74 playlists, and generate 74 new playlists, each of which contains 50 distinct songs based on the ordered recommendation lists of the these five songs. Then we check how many of the songs in the rest of each original playlists (the number of songs available for checking varies from 5 to 15) match the songs in the new larger playlists. Fig. 4 lists the box-plot results of the comparison among content-based approach, hybrid approach, and our dynamic-weighting approach.

From Fig. 4 and Table IV, we clearly see that our DWA approach outperforms content-based approach and the hybrid approach. If we check the data in detail, we can find that for predicting some playlists, when there is enough song co-occurrence information, the hybrid approach works very well and have the comparable performance with our dynamic-approach. However, when dealing with new song sets and there are very little song co-occurrence data, the hybrid approach is almost

TABLE IV
TIMES OF ONE APPROACH OUTPERFORMS THE OTHER TWO
BY COMPARING THE MATCHES IN TWO PLAYLISTS

Approach	CBA	HA	DWA
Winning Rounds	8	20	37

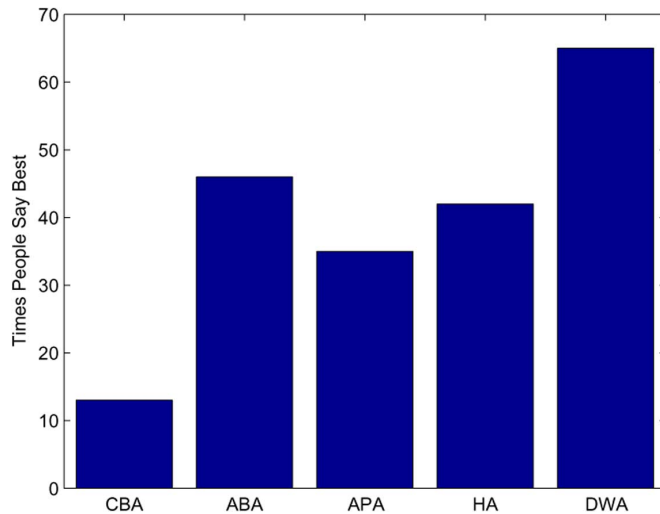


Fig. 5. Times people say one approach is the best among all approaches.

degraded to content-based approach. On the contrary, our dynamic-weighting approach is trying to predict the recommended songs based on the weights already learned and the content features extracted, it can keep the similar performance when dealing with new song sets.

4) *User Study*: We develop a web interface and invite the users from the website to assess the recommendation results of different approaches. The interface can be found at <http://www.newwisdom.net/music/songUserStudy.jsp>.

For each song, we list the recommended songs (song titles and singers) using the five approaches described above. For each seed song that interests the user, he/she is invited to choose those that also interest him/her in the recommended list, and also select the best approach based on their perception. Note that the songs presented to the visitors are randomized and there is no fixed song appearance order. We asked the visitors to rate the recommended songs as well as the overall impression of all the five approaches for a given seed song.

To submit a feedback, the user must choose one and only one best approach from the five, but he/she can select any number of songs from the recommendation list as he/she likes. To make different songs have nearly equal chances to be exposed to the users for judgment, the selection of songs from the repository is also randomized. By collecting the IP addresses of the users, we know that more than 50 users (59 IP addresses) participated in the user study, and the recommendation results of 166 distinct songs are assessed by one or some of them. Altogether there are 201 submission of feedbacks. Table V lists the statistical results of the user study and Fig. 5 compares the number of times people claim that an approach is the best among the five approaches.

By checking the statistical results of the user study listed in Table V, we can clearly see that our approach outperforms all the rest. For example, in row “r1,” there are 69 times that the

TABLE V
RESULTS OF USER STUDY. FOR EACH i , $1 \leq i \leq 10$, THE ROW “ r_i ” SHOWS THE TOTAL NUMBER OF TIMES THAT SONGS AT THE i TH POSITION IN THE RECOMMENDATION LIST IS SELECTED BY USERS FOR EACH APPROACH. THE ROW “SUM” LISTS THE CORRESPONDING SUMMATION OF ALL THE VALUES FOR EACH OF THE FIVE APPROACHES

	Approach				
	CBA	ABA	APA	HA	DWA
r1	25	47	38	48	69
r2	31	54	44	52	60
r3	19	34	41	49	52
r4	17	37	33	51	58
r5	22	38	45	47	44
r6	19	49	31	44	43
r7	13	22	27	52	47
r8	22	14	25	19	42
r9	7	17	24	32	39
r10	16	19	28	16	38
sum	191	331	336	410	492

recommended songs in position 1 by our dynamic-weighting approach are considered to be valuable recommendations while for hybrid method, there are only 48 times. In Fig. 5, we also know that our dynamic-weighting approach is regarded as the best one among the five choices for most users at most times. Users sometimes also think the recommended songs from the same artists are what they prefer, but as we all know, that recommendation does not give users enough novel information.

VI. CONCLUSION

Both collaborative-filtering and content-based recommending schemes have their own advantages and limitations. In this paper, we propose a novel dynamic music similarity measurement scheme that integrates the acoustic content features and user access patterns. This scheme is based on the assumption that two pieces of music are similar in human perception when they share similar access patterns across multiple users. To calculate the new similarity measure, we use the metric learning approach, which learns appropriate similarity metrics based on the correlation between acoustic features and user access patterns of music, to automatically determine the weights for audio features. After obtaining the music similarity, music recommendation can be treated as a label propagation from labeled data (i.e., items with ratings) to unlabeled data. Comparing with other probabilistic models and hybrid approaches, our method incorporates the content similarity data and collaborative filtering information seamlessly. Experimental results and user study on a real data set demonstrate the recommendation quality of our proposed approach outperforms the others.

Although our proposed recommendation scheme has been tested to be effective, there are several venues for further research. One natural direction is to extent our current framework for personalized music recommendation. Furthermore, we can investigate more comprehensive music content features for similarity measurements.

REFERENCES

[1] J. S. Breese, D. Heckerman, and C. Kadie, “Empirical analysis of predictive algorithms for collaborative filtering,” in *Proc. 14th Annu. Conf. Uncertainty Artif. Intell.*, 1998, pp. 43–52.

- [2] A. Z. Broder, M. Charikar, A. M. Frieze, and M. Mitzenmacher, "Min-wise independent permutations," *J. Comput. Syst. Sci.*, vol. 60, no. 3, pp. 630–659, 2000.
- [3] R. Cai, C. Zhang, L. Zhang, and W.-Y. Ma, "Scalable music recommendation by search," in *Proc. MULTIMEDIA'07: Proc. 15th Int. Conf. Multimedia*, 2007, pp. 1065–1074.
- [4] H.-C. Chen and A. L. P. Chen, "A music recommendation system based on music data grouping and user interests," in *Proc. CIKM '01: Proc. 10th Int. Conf. Inf. Knowledge Manag.*, New York, 2001, pp. 231–238.
- [5] W. W. Cohen and W. Fan, "Web-collaborative filtering: Recommending music by crawling the web," *Comput. Netw.*, vol. 33, no. 1–6, pp. 685–698, 2000.
- [6] I. Daubechies, *Ten Lectures on Wavelets*. Philadelphia, PA: SIAM, 1992.
- [7] C. Ding, R. Jin, T. Li, and H. D. Simon, "A learning framework using green's function and kernel regularization with application to recommender system," in *Proc. KDD '07: 13th ACM SIGKDD Int. Conf. Knowledge Discovery Data Mining*, New York, 2007, pp. 260–269.
- [8] W. J. Dowling and D. L. Harwood, *Music Cognition*. San Diego, CA: Academic, 1986.
- [9] J. Foote, M. Cooper, and U. Nam, "Audio retrieval by rhythmic similarity," in *Proc. ISMIR '02*, 2002, pp. 265–266.
- [10] J. Foote and S. Uchihashi, "The beat spectrum: A new approach to rhythm analysis," in *Proc. IEEE Int. Conf. Multimedia Expo*, 2001, pp. 881–884.
- [11] P. E. Gill, W. Murray, and M. H. Wright, *Practical Optimization*. New York: Academic, 1981.
- [12] J. He, M. Li, H.-J. Zhang, H. Tong, and C. Zhang, "Manifold ranking based image retrieval," in *Proc. ACM Multimedia*, 2004.
- [13] X. He, W.-Y. Ma, and H.-J. Zhang, "Learning an image manifold for retrieval," in *Proc. ACM MM*, 2004.
- [14] J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl, "An algorithmic framework for performing collaborative filtering," in *SIGIR '99: Proc. 22nd Annual Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 1999, pp. 230–237.
- [15] Y.-C. Huang and S.-K. Jenor, "An audio recommendation system based on audio signature description scheme in mpeg-7 audio," in *2004 IEEE Int. Conf. Multimedia Expo*, 2004, vol. 1, pp. 639–642.
- [16] K.-Y. Jung, D.-H. Park, and J.-H. Lee, "Hybrid collaborative filtering and content-based filtering for improved recommender system," in *Proc. Comput. Sci.—ICCS 2004*, Berlin/Heidelberg, Germany, 2004, pp. 295–302.
- [17] P. Knees, T. Pohle, M. Schedl, and G. Widmer, "Combining audio-based similarity with web-based data to accelerate automatic music playlist generation," in *Proc. MIR'06: 8th ACM Int. Workshop Multimedia Inf. Retrieval*, New York, 2006, pp. 147–154.
- [18] R. Kondor and J. Lafferty, "Diffusion kernels on graphs and other discrete input spaces," in *Proc. 2002 Int. Conf. Mach. Learn. (ICML)*, 2002.
- [19] Q. Li, B.-M. Kim, D.-H. Guan, and D.-W. Oh, "A music recommender based on audio features," in *SIGIR '04: Proc. 27th Annual Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, New York, 2004, pp. 532–533.
- [20] T. Li and M. Ogihara, "Content-based music similarity search and emotion detection," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2004, vol. 5, pp. 705–708.
- [21] T. Li, M. Ogihara, and Q. Li, "A comparative study on content-based music genre classification," in *Proc. SIGIR*, 2003, pp. 282–289.
- [22] B. Logan, "Music recommendation from song sets," in *Proc. ISMIR*, Oct. 2004, pp. 425–428.
- [23] B. Logan and A. Salomon, "A content-based music similarity function," Cambridge Res. Lab., 2001, Tech. Rep. CRL 2001/02.
- [24] P. Melville, R. Mooney, and R. Nagarajan, "Content-boosted collaborative filtering for improved recommendations," in *Proc. 18th National Conf. Artif. Intell. (AAAI-02)*, 2002.
- [25] H. Muller, T. Pun, and D. Squire, "Learning from user behavior in image retrieval: Application of market basket analysis," *Int. J. Comput. Vis.*, vol. 56, no. 1–2, pp. 65–77, 2004.
- [26] N. Oliver and L. Kreger-Stickles, "Papa: Physiology and purpose-aware automatic playlist generation," in *Proc. 7th Int. Conf. Music Inf. Retrieval*, Oct. 2006, pp. 250–253.
- [27] F. Pachet, P. Roy, and D. Cazaly, "A combinatorial approach to content-based music selection," *IEEE Multimedia*, vol. 7, no. 1, pp. 457–462, Jul. 2000.
- [28] S. Pauws, W. Verhaegh, and M. Vossen, "Fast generation of optimal music playlists using local search," in *Proc. 7th Int. Conf. Music Inf. Retrieval*, Oct. 2006, pp. 138–143.
- [29] J. C. Platt, C. J. C. Burges, S. Swenson, C. Weare, and A. Zheng, "Learning a Gaussian process prior for automatically generating music playlists," in *Advances in Neural Information Processing Systems 14*, 2002, pp. 1425–1432.
- [30] A. Popescul, L. Ungar, D. Pennock, and S. Lawrence, "Probabilistic models for unified collaborative and content-based recommendation in sparse-data environments," in *17th Conf. Uncertainty Artif. Intell.*, Seattle, WA, Aug. 2–5, 2001, pp. 437–444.
- [31] R. Ragno, C. J. C. Burges, and C. Herley, "Inferring similarity between music objects with application to playlist generation," in *Proc. 7th ACM SIGMM Int. Workshop Multimedia Inf. Retrieval*, 2005, pp. 73–80.
- [32] Y. Rui and T. S. Huang, "Optimizing learning in image retrieval," in *Proc. IEEE Comput. Vis. Pattern Recognition*, 2000, pp. 236–243.
- [33] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Application of dimensionality reduction in recommender systems—a case study," in *Proc. ACM WebKDD Workshop*, 2000.
- [34] J. B. Schafer, J. Konstan, and J. Riedl, "Recommender systems in e-commerce," in *Proc. EC '99: Proc. 1st ACM Conf. Electronic Commerce*, 1999, pp. 158–166.
- [35] A. J. Smola and R. Kondor, "Kernels and regularization on graphs," in *Proc. 16th Annu. Conf. Learning Theory 7th Kernel Workshop*, 2003, pp. 144–158.
- [36] M. Szummer and T. Jaakkola, "Partially labeled classification with Markov random walks," in *Advances in Neural Information Process. Syst.*, 2001, vol. 14.
- [37] G. Tzanetakis and P. Cook, "Music genre classification of audio signals," *IEEE Trans. Speech Audio Process.*, vol. 10, pp. 293–302, 2002.
- [38] A. Uitendbogerd and R. van Schyndel, "A review of factors affecting music recommender success," in *Proc. ISMIR*, 2002.
- [39] D. Wettschereck and D. W. Aha, "Weighting features," in *Proc. Case-Based Reasoning, Research and Development, First Int. Conf.*, 1995, pp. 347–358.
- [40] E. P. King, A. Y. Ng, M. I. Jordan, and S. Russell, "Distance metric learning, with application to clustering with side-information," in *Advances in Neural Information Processing Systems 15*, 2003, pp. 505–512.
- [41] K. Yoshii, M. Goto, K. Komatani, T. Ogata, and H. G. Okuno, "Hybrid collaborative and content-based music recommendation using probabilistic model with latent user preferences," in *Proc. ISMIR*, 2006.
- [42] D. Zhou, O. Bousquet, T. Lal, J. Weston, and B. Schölkopf, "Learning with local and global consistency," in *Proc. 18th Annu. Conf. Neural Inf. Process. Syst.*, 2003.
- [43] X. Zhu, Z. Ghahramani, and J. Lafferty, "Semi-supervised learning using Gaussian fields and harmonic functions," in *Proc. ICML*, 2003.



Bo Shao received the B.S. degree in mining engineering from Northeastern University, Shenyang, China, in 1992, and the M.S. degree in computer sciences and applications from Southeast University, Nanjing, China, in 1995. He is currently pursuing the Ph.D. degree in the School of Computing and Information Sciences, Florida International University, Miami.

His primary research interests are music information retrieval and data mining.



Dingding Wang received the B.S. degree from the Department of Computer Science, University of Science and Technology of China, Hefei, in 2003, and the M.S. degree in telecommunications and networking from Florida International University (FIU), Miami, in 2006. She is currently pursuing the Ph.D. degree in the School of Computing and Information Sciences, FIU.

Her research interests are data mining and information retrieval.



Tao Li received the Ph.D. degree in computer science from University of Rochester, Rochester, NY, in 2004.

He is currently an Assistant Professor in School of Computing and Information Sciences, Florida International University, Miami. His primary research interests are data mining, machine learning, information retrieval, and bioinformatics.

Prof. Li is a recipient of an NSF CAREER Award in 2006 and multiple IBM Faculty Research Awards.



Mitsunori Ogihara received the Ph.D. degree in information sciences from the Tokyo Institute of Technology, Tokyo, Japan, in 1993.

He is currently Professor of computer science at the University of Miami, Coral Gables, and Director of Data Mining in the Center for Computational Science at the university. He is on the editorial board for the journals *Theory of Computing Systems* and *International Journal of Foundations of Computer Science*.

Prof. Ogihara is a Distinguished Scientist member of Association for Computing Machinery. He is a recipient of an NSF CAREER Award in 1997.