Neurocomputing 254 (2017) 22–32

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Efficient karaoke song recommendation via multiple kernel learning approximation

Chu Guan[a], Yanjie Fu[b,*], Xinjiang Lu[c], Enhong Chen[a,*], Xiaolin Li[d], Hui Xiong[e]

[a] Department of Computer Science, University of Science and Technology of China, China
[b] Department of Computer Science, Missouri University of Science and Technology, United States
[c] School of Computer Science, Northwestern Polytechnical University, China
[d] School of Business, Nanjing University, China
[e] Management Science and Information Systems Department, Rutgers University, United States

ARTICLE INFO

Article history:
Received 14 February 2016
Revised 21 October 2016
Accepted 27 October 2016
Available online 6 March 2017

Keywords:
Kaorake recommendation
Multiple kernel learning
Singing competence

ABSTRACT

Online karaoke allows users to practice singing and distribute recordings. Different from traditional music recommendation, online karaoke need to consider users' vocal competence besides their tastes. In this paper, we develop a karaoke recommender system by taking into account vocal competence. Alone this line, we propose a joint modeling method named MKLA by adopting bregman divergence as the regularizer in the formulation of multiple kernel learning. Specially, we first extract users' vocal ratings from their singing recordings. Due to an ever-increasing number of recordings, the evaluations in large-scale kernel matrix may cost lots of time and internal storage. Therefore, we propose a sample compression method to eliminate users' vocal ratings, exploit an MKL method, and learn the latent features of the vocal ratings. These latent features are simultaneously fed into a bregman divergence and then we use the trained classifier to predict the overall rating of a user with respect to a song. Enhanced by this new formulation, we develop the SMO method for optimizing the MKLA dual and present a theoretical analysis to show the lower bound of our method. With the estimated model, we compute the matching degree of users and songs in terms of pitch, volume and rhythm and recommend songs to users. Finally, we conduct extensive experiments with online karaoke data. The results demonstrate the effectiveness of our method.

1. Introduction

Online karaoke has become a hybrid media form designed to integrate popular music, video, images and the live musical performance of users. Nowadays, users can access online karaoke service with only a microphone and a computer connected to the internet. Thus, karaoke recommendation is importance because it can help users identify appropriate karaoke songs, receive high ratings, and improve karaoke experience.

Unlike classic music recommendation, online karaoke has unique characteristics. For example, if one receives a high rating on a karaoke song, he/she may not just favors this song but also has vocal competence to sing the song well. Therefore, users usually care about whether their vocal competence meets the vocal requirements of songs when they choose karaoke songs. These unique characteristics of online karaoke provide us an opportunity to enhance karaoke recommendation. However, this is a non-trivial task. There are three major challenges. First, since historical song recordings encode the information about users' vocal competence, careful methods need to be designed to learn the representations of users vocal competence. Second, as the representations of users' vocal competence will be utilized to predict the overall ratings of karaokes, it is very difficult to find the optimal representations which can help enhance the prediction of overall ratings. Finally, the modeling method needs to be robust and enable to produce accurate recommendations in practice.

Indeed, in the decision process of choosing karaoke songs, users not only take the style and content of songs, but also consider the degree of matching requirements of songs to their vocal competence. In this way, they can sing the chosen songs well and receive high scores. With the development of computational acoustic analysis, we can extract multiple-aspect vocal ratings (e.g. ratings of pitch,volume and rhythm) by analyzing their karaoke singing recordings. Specially, after preprocessing the karaoke records, we obtain audio records encoding users' vocal performance and then

* Corresponding authors.
E-mail addresses: guanchu@mail.ustc.edu.cn (C. Guan), fuyan@mst.edu (Y. Fu), cheneh@ustc.edu.cn (E. Chen).

http://dx.doi.org/10.1016/j.neucom.2016.10.083
0925-2312/

extract ratings of pitch, volume and rhythm. Later, we exploit a multiple kernel learning to model the generative process of vocal ratings. Therefor, we can learn the latent features of users' vocal competence.

To tackle the first challenge, a multiple kernel learning is employed to examine the multi-aspect vocal ratings and estimate the latent features of vocal competence of users. In recent years, multiple kernel learning (MKL) based methods have been proposed to consider multiple kernels or the combination of kernels rather than a single fixed kernel [16][19]. Kernel methods can easily handle the data with non-linear structure by mapping data into a high-dimensional space (known as *feature space*). Since MKL considers multiple kernels, it can characterize the geometrical structure of different aspect for the vocal rating data. However, evaluations in MKL may cost lots of time and internal storage and there are many variables to be learned in optimization. For example, given n karaoke recordings and m overall ratings, the kernel matrix is $n \times m$ which suggests that a computational complexity is at least $O(n \times m)$. To reduce the computational complexity, we present a joint modeling method which incorporates bregman divergence as a regularization in the formulation of multiple kernel learning. Furthermore, to efficiently optimize the proposed algorithm, we develop SMO method to optimize the MKLA dual rather than the intermediate saddle point problem on which all state-of-the-art MKL solvers are based [18,42].

To this end, in this paper, we develop a song recommender system for online karaoke by mining the correlations among overall ratings given by a karaoke machine and multi-aspect vocal ratings given by acoustic analysis. Alone this line, we propose a joint modeling method to provide accurate and personalized recommendations. Specially, we first define and extract the multi-aspect vocal (i.e., pitch, volume, and rhythm) ratings of a user for a song based on their karaoke recordings using acoustic analysis. We then exploit an MKL method to model the generative process of vocal ratings. Moreover, we employ breman divergence as a regularization term to reduce the computational complexity, to enhance the optimization efficiency thus can recommend songs for each user. In addition, we solve this objective via a developed SMO method for parameter estimation. Finally, we conduct extensive experiments with real world online karaoke data. The results demonstrate the effectiveness of the proposed method.

2. Preliminaries

In this section, we first formalize the problem of karaoke songs recommendation, then introduce the definitions and collections of multi-aspect vocal ratings given by acoustic analysis and overall ratings given by a karaoke machine, and finally illustrate the overview of the Multiple Kernel Learning Approximation, named MKLA.

2.1. Problem statement

In this paper, we aim at developing a karaoke recommender system by modeling the impact of users vocal competence on choosing a karaoke song. Formally, given a user, the developed recommender system should return a ranked list of karaoke songs for him/her, such that the ranked song list can help to maximize the expectation or probability of receiving highest overall ratings of the karaoke performance. Essentially, the central tasks are (1) to learn and extract the vocal competence of users and vocal requirements of songs, and (2) to incorporate the degree of matching users vocal competence to songs vocal requirements for karaoke recommendation.

2.2. Multi-aspect vocal ratings and overall rating

We first introduce multi-aspect vocal ratings. The karaoke singing record of a user for a song is associated with users and audio signal. Therefore, we can model such user-audio relations using a kernel matrix, with each element representing a single-aspect vocal rating of a user for a song. In particular, we denote the matrix as $\mathbf{X}$. Then, x_{ij} in $\mathbf{X}$ denotes the rating of the vocal feature j sung by the user i. For example, the rhythm rating of user #1 for song #2 is 88.

Assume that an online karaoke service uses a binary rating system and rates a karaoke record as good (+1) or bad (-1). Let $\{\mathbf{X}, \mathbf{y}\}$ denote the observed data, where $\mathbf{X}$ is the matrix of multi-aspect vocal ratings of users' recordings, and $y_i = \{+1, -1\}$ denotes the binary overall ratings of karaoke performance.

Mathematically, we use an MKL framework, which involves a linear combination of m pre-define base kernels $\{\kappa_p(\cdot,\cdot)\}_{p=1}^m$, which induce the feature mapping to take the form $\Phi(\cdot) = [\Phi_1^\top, \Phi_2^\top, \ldots, \Phi_m^\top]^\top$. In MKL, we look for a decision function $f(\mathbf{x}) \in \mathcal{H}_p$ where $\mathcal{H}$ is the RKHS associated with combination of kernels . In order to learn such function $f(\mathbf{x})$, a solution is to solve the following MKL problem [32]:

$$\min_{\{\omega_p\}_{p=1}^m, b, \xi} \frac{1}{2}\left(\sum_{p=1}^{m} \|\omega_p\|_{\mathcal{H}_p}\right)^2 + C\sum_{i=1}^{n}\xi_i$$
$$s.t.\ y_i\left(\sum_{p=1}^{m}\omega_p^\top \Phi_p(\mathbf{x_i}) + b\right) \geq 1 - \xi_i, \xi_i \geq 0, \forall i \quad (1)$$

where $\{\mathbf{x_i}, y_i\}$ is the set of karaoke recordings, $\mathbf{w}_p$ and $\mathcal{H}_p$ denotes the weight and corresponding feature space for pth kernel, ξ and b present the slack variables and bias term and C is the regularization parameter.

2.3. The Overview of our model

Fig. 1 shows that our proposed method consists of three major steps as follows:

Extracting Multi-Aspect Vocal Ratings: Given a group of users, we first collect their historical karaoke recordings. Then, in order to characterize users singing competence, we extract the features of pitch, volume, and rhythm as multi-aspect ratings while removing the background music. Furthermore, the recommended songs should be a binary vector, thus, we adopt a pre-defined threshold to determine such songs.

Learning Users Vocal Competence: We propose a sample compression method to select most representative vocal ratings and treat these ratings of pitch, volume and rhythm of users for songs as a matrix. Then, we map the ratings into a high-dimensional space and learn the latent representations of users competence simultaneously.

Exploiting the Matching Degree between Users and Songs for Karaoke Recommendation: While mapping the singing ratings into a high-dimensional space, the selected kernels can characterize the geometrical structure of different aspect for users' vocal competence. Therefore, we jointly combine the multiple kernel learning and bregman divergence as an unified objective. In addition, an efficient optimization approach is developed to solve this formulation. Finally, we can predict the overall ratings with respect to the learned factor matrices. The top-n songs with highest ratings are recommended.

3. Real-time karaoke song recommendation

In this section, we introduce the Real-time Karaoke Song recommendation via multiple kernel learning approximation.

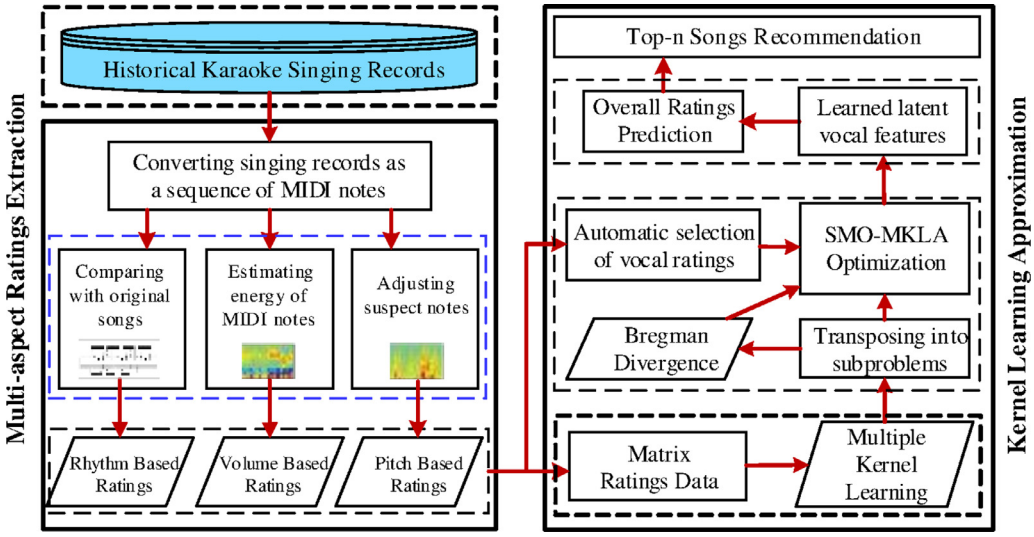


Fig. 1. The framework of multiple kernel learning approximation.

3.1. Multi-aspect vocal ratings acquisition

To obtain a user's multi-aspect vocal ratings, we first convert the waveform of a singing record to a sequence of MIDI notes. A typical MIDI file contains both the singing melody and its accompaniment. Most melodies are not on the same tune with the ground-truth music scores [38]. For example, the largest value of a MIDI note may not associated with the singer, but the instruments. In practice, a MIDI note τ is converted from Hertz, i.e., $\tau = \lfloor 12 \times \log_2\left(\frac{Hz}{440}\right) + 69.5 \rfloor$. Then we perform a cleaning procedure to remove the background music and obtain users' singing characteristics. Here, we adopt the strategy in [38] which uses the original acoustic sound to measure the correctness of a singing performance of pitch, volume and rhythm. Formally, given a cover version c and the original version c', we let $Seq(c) = \{\tau_1, \tau_2, \ldots, \tau_K\}$ and $Seq(c') = \{\tau'_1, \tau'_2, \ldots, \tau'_K\}$ be the MIDI note sequences of c and c', respectively.

Pitch-based ratings. In analysis of singing performance, the pitch is related to the degree of highness or lowness of a tone. In other words, to achieve a high score, users should sing a sequence of correct notes with appropriate duration. The notes of background accompaniment are often above or below the singing record so that the mixture of the background accompaniment and the vocal sound is harmonic. Based on this observation, a sequence of MIDI notes can be adjusted by shifting the suspect notes several octaves up or down, so that the range of adjusted notes conforms to the normal range. For a MIDI note τ_t in $Seq(c)$, if τ_t is abnormal, then we adjust it as $\tau'_t = \tau_t - \lfloor \tau_t - \bar{\tau} + 6|\tau| \rfloor$, where $\bar{\tau}$ is the average value of MIDI notes in $Seq(c)$ and $|\tau|$ is the normal range of the sung notes in a sequence and $|\tau| = 24$ in practice. The adjusted sequence is denoted as $\tilde{Seq}(c)$ which is used for pitch-based ratings, i.e.,

$$R^{pitch} = \tilde{Seq}(c). \tag{2}$$

Volume-based ratings. Volume refers to the intensity of sound in a piece of music. A simple strategy for extracting volume-based ratings is to compare a cover version with the original version. After adjusting abnormal elements of $Seq(c)$ and $Seq(c')$ by using Eq. (2), we have two adjusted sequences of MIDI notes $\tilde{Seq}(c)$ and $\tilde{Seq}(c')$. Then a volume-based rating of c is computed by:

$$R^{volume} = \mathcal{I} \times \exp\left[\mathrm{sim}\left(\tilde{Seq}(c), \tilde{Seq}(c')\right)\right], \tag{3}$$

where $\mathrm{sim}(\cdot)$ is used to measure the similarity between $\tilde{Seq}(c)$ and $\tilde{Seq}(c')$. $\mathcal{I}$ is associated with the range of a rating. For example, if a pitch-based rating is between 0 and 100, then $\mathcal{I} = 100$.

Rhythm-based ratings. Rhythm represents the onset and duration of successive notes and rests performed by a user. Professional singers sometimes elicit emotional response from the audience during the liberty of the time. However, in the scenario of karaoke, users have to follow the flow of the accompaniment because of the prerecorded accompaniment. Thus, the strategy of extracting rhythm-based ratings is based on the comparison of the onsets of notes sung in cover versions and original versions. In this work, we adopt Dynamic Time Warping (DTW) [4] which can calculate the similarity between two time series based on finding an optimal match between them even if they are not identical in size. For two sequence $\tilde{Seq}(c)$ and $\tilde{Seq}(c')$, we have the DTW distance between them, i.e., $\mathrm{Sim}_{DTW}(\tilde{Seq}(c), \tilde{Seq}(c'))$. Then

$$R^{rhythm} = \mathcal{I} \times \exp\left[\mathrm{Sim}_{DTW}\left(\tilde{Seq}(c), \tilde{Seq}(c')\right)\right], \tag{4}$$

where $\mathcal{I}$ is configured with the same setting adopted in Eq. (3), i.e. $\mathcal{I} = 100$.

3.2. Automatic selection of representative vocal ratings

To eliminate redundant vocal ratings, we first select a set of representative vocal ratings for each user. Our selection strategy is mainly based on Bernstein's inequality, which have been proven to be effective in many machine learning algorithms. By taking the information of sample variance into account [6,14], Bernstein's inequality is as follows:

Bernstein's inequality. Let $\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x}_m \in [0, 1]$ be independent random variables where $m > 5$, then for $\delta \in (0, 1)$ with probability at least $1 - \delta$, we have

$$\left| \frac{1}{m}\sum_{i=1}^{m} E[\mathbf{x_i}] - \frac{1}{m}\sum_{i=1}^{m} \mathbf{x_i} \right| \le \sqrt{\frac{2\hat{V}_m \ln(2/\delta)}{m}} + \frac{7 \ln 2/\delta}{3m} \tag{5}$$

where $\hat{V}_m$ is the sample variance, i.e., $\hat{V}_m = \sum_{i \neq j} (\mathbf{x_i} - \mathbf{x_j})^2 / 2m(m-1)$. Our sampling strategy aims to draw vocal ratings with low variance and the number of drawn samples should be proportional to the size of the sample space.

Stopping criterion. To determinate when to stop sampling, we need to construct a sequence d_t which is used to set the confidence parameter δ. In this paper, we make $d_t = \frac{1}{p^t}\frac{\delta(p-1)}{p}$, where

$p > 1$, because it can merely ensure that $\sum_{t=1}^{\infty} d_t \le \delta$. Given t samples, we define c_t as the empirical bound with a half width of $1 - d_t$ confidence interval

$$c_t = \sqrt{\frac{2\hat{V}_m \ln(2/\delta)}{m}} + \frac{7 \ln 2/\delta}{3m}, \tag{6}$$

and define the event ε as $\varepsilon = \bigcap_{t\ge 1}\{|\bar{\mathbf{x}}_t - \mu| \le c_t\}$, where $\bar{\mathbf{x}}_t$ is the mean of the t samples available. We construct the stopping criterion that the event ε holds with probability $1 - \delta$ at least. In the tth iteration, we draw $\mathbf{x}_t$ and set $B^t_{left} \leftarrow \max(B^{t-1}_{left}, |\bar{\mathbf{x}}| - c_t)$ and $B^t_{right} \leftarrow \min(B^{t-1}_{right}, |\bar{x}| + c_t)$.

The confidence interval for $|\bar{\mathbf{x}}_t|$ is not wider than the confidence interval for $|\mu|$. It implies that $||\bar{\mathbf{x}}_t| - |\mu|| \le c_t$, and it is easy to see that

$$||\bar{\mathbf{x}}_t| - |\mu|| \le c_t \le \epsilon(|\bar{\mathbf{x}}_t| - c_t) \le \epsilon|\mu| \tag{7}$$

Hence, we stop sampling when $|\bar{x}_t|$ is close to $|\mu|$ within relative error ϵ. The sampling process is contained in Algorithm 1.

Algorithm 1 The learning process of MKLA.

Input: $\{x_i, y_i\}_{i=1}^N$; m candidate kernels $K_k(x_i, x_j)_{k=1}^m$
Output: the weight vectors ω_l, $l = 1, \ldots, m$

```
Initialize ξ > 0, {ω_k}_{k=1}^m; B_left = 0, B_right = ∞; iteration index t = 1;
while (1 + ε)B_left < (1 − ε)B_right do
    Obtain overlineX_t;
    B^t_left ← max(B^{t−1}_left, |X̄_t| − c_t);
    B^t_right ← max(B^{t−1}_right, |X̄_t| + c_t);
end while
for t = 1, ..., T do
    Compute ∇_d L_B, g_k(α, γ) and θ(α, γ)
    for k = 1, ..., m do
        Compute the gradient (Hessian) of MKLA dual
        Update ∇_α D and ∇²_α D
    end for
end for
```

To this end, for user i, we extract the three aspect ratings and aggregate them into a vector, i.e., $\mathbf{x}_i = \{R_i^{pitch}, R_i^{volume}, R_i^{rhythm}\}$. After extracting users' vocal ratings, we aggregate them as a matrix $\mathbf{x}$.

3.3. Multiple kernel learning approximation

We introduce the proposed joint model which combines the modelings of multiple kernel learning and bregman divergence together. By solving this joint optimization problem, we can learn the optimized latent representations of pitch, volume and rhythm which preserve the structural information of the multi-aspect rating matrix while effectively discriminate the karaoke overall ratings.

The Modeling of MKL. Given a matrix of multi-aspect ratings $\mathbf{X}$, the formulation of multiple kernel learning in Eq. (1) is proven to be equivalent to

$$\min_{\{\omega_p\}_{p=1}^m, b, \xi, \gamma\in\Delta} \frac{1}{2}\frac{(\sum_{p=1}^m \|\omega_p\|_{\mathcal{H}_p})^2}{\gamma_p} + C\sum_{i=1}^n \xi_i$$
$$s.t. \quad y_i\left(\sum_{p=1}^m \omega_p^\top \Phi_p(\mathbf{x_i}) + b\right) \ge 1 - \xi_i, \xi_i \ge 0, \forall i \tag{8}$$

where $\Delta = \{\gamma : \sum_{p=1}^m \gamma_p = 1, \gamma_p \ge 0, \forall p\}$ and the choice of p usually enforces the kernel combination to be sparse ($p = 1$) or non-sparse ($p > 1$). The rest variables $\{\gamma_p\}_{p=1}^m$, b and $\boldsymbol{\xi}$ can be obtained by solving the corresponding dual problem where the kernel matrix is a fixed combination of kernels.

Then we aim to reduce the computational complexity of MKL which can be measured by a variety of formal approaches, e.g., Vapnik–Chervonenkis dimension, covering number and Rademarcher complexity. In this paper, we choose Rademarcher complexity, because it is particularly amenable in experiment [12].

According to the theorem present in [3], the Rademacher complexity of kernel classes can be described in terms of the eigenvalues of the gram matrix. If $\lambda \ge 1$,then for every $r \ge \frac{1}{n}$, there is a certain constant c such that

$$\frac{2}{\sqrt{n}}\sum_{i=0}^{\infty}\min(r, \lambda_i) \le \mathbb{E}[\mathcal{R}(\mathcal{H}; r)]$$

Here, we can clearly find out that the Rademacher complexity for kernel classes can be bounded by the tail sum of the eigenvalues. To learn a low-rank combination of kernels, a straight idea is that to constrain the kernel by a tail sum of the eigenvalues. However, it may lost much information of kernel classes and lead to an inaccurate classification performance. Instead of optimizing kernel directly over a cut-off point of the tail sum of the eigenvalues, we propose an approximate algorithm of MKL which preserves both representative and structural information of the kernel matrix.

The Modeling of MKLA. To achieve greater generalized discriminating power, our optimization problem also intends to capture the structural information of vocal ratings in the high-dimensional feature space. Let D_Φ be a Bregman divergence which defines the distance between two kernel matrix K and K_0:

$$D_\Phi(K, K_0) = tr(KK_0^{-1}) - \log\det(KK_0^{-1}) - N. \tag{9}$$

The Bregman divergence is not symmetric and does not obey the triangle inequality [20]. Here, we impose Bregman divergence as a regulariser in MKLA.

Then we can incorporate Bregman Divergence into the objective function. Here, we first present the dual formulation of Eq. (8):

$$F = \min_{\gamma\in\Delta}\max_{\alpha} -\frac{1}{2}(\alpha\circ\mathbf{y})^\top\left(\sum_{p=1}^m \gamma_p K_p\right)(\alpha\circ\mathbf{y}) + \alpha\mathbf{e}, \tag{10}$$

where $\boldsymbol{\alpha}$ is the vector of Lagrange multipliers, $\boldsymbol{\alpha}\circ\mathbf{y}$ presents the component-wise multiplication and $\mathbf{e}$ is a one-dimensional vector with 1 elements. Consider the gradient of F is $f = \nabla F$, the Bregman divergence generated by F is given by $r_F(\mathbf{d}) = F(\mathbf{d}) - F(\mathbf{d}_0) - (\mathbf{d} - \mathbf{d}_0)^t f(\mathbf{d}_0)$ and its gradient is $\nabla = f(\mathbf{d}) = f(\mathbf{d}_0)$. Then we generalize the Bregman divergence as the squared p-norm and have the objective function of MKLA:

$$I_B = \min_{\mathbf{d}\ge\mathbf{0}}\max_{\alpha\in\mathcal{A}} \mathbf{1^t}\alpha - \frac{1}{2}\sum_k d_k\alpha^{\mathbf{t}}H_k\alpha + \lambda r_F(\mathbf{d}) \tag{11}$$

Our primal objective leads to the intermediate saddle point problem and Lagrangian.

$$L_B = \mathbf{1^t}\alpha - \sum_k d_k(\gamma_k + \frac{1}{2}\alpha^t H_k\alpha) + \lambda r_F(\mathbf{d}) \tag{12}$$

$$\nabla_{\mathbf{d}} L_B = 0 \Rightarrow f(\mathbf{d}) - f(\mathbf{d_0}) = g(\alpha, \gamma)/\lambda$$
$$\Rightarrow \mathbf{d} = f^{-1}(f(\mathbf{d_0}) + g(\alpha, \gamma)/\lambda) = f^{-1}(\theta(\alpha, \gamma)) \tag{13}$$

where $H_k = YK_kY$, g is a vector whose entries $g_k(\alpha, \gamma) = \gamma_k + \frac{1}{2}\alpha^t H_k\alpha$ and $\theta(\alpha, \gamma) = f(\mathbf{d_0}) + g(\alpha, \gamma)/\lambda$. By using an optimization algorithm, the optimal value of γ turns out to be zero and then the optimisation can be carried out.

SMO-MKLA Optimization. We develop the SMO method for optimizing the MKLA dual which is mainly built around the LibSVM code [9]. We repeatedly choose two variables and optimizing them while holding all other variables constant. If $\alpha_1 \leftarrow \alpha_1 + \triangle$

and $\alpha_2 \leftarrow \alpha_2 + s\triangle$, the dual simplifies to

$$\triangle^* = \arg\max_{\mathbf{L}\le\triangle\le\mathbf{U}} (1-y_1y_2)\triangle - \frac{1}{8\lambda}\left(\sum_k (a_k\triangle^2 + 2b_k\triangle + c_k)^q\right)^{\frac{2}{q}}. \tag{14}$$

Here, $a_k = H_{11k} + H_{22k} + 2sH_{12k}$, $b_k = \alpha^t(H_{:1k} + sH_{:2k})$ and $c_k = \alpha^t H_k \alpha$. If $y_1y_2 = 1$, then $\mathbf{L} = \max(-\alpha_1, \alpha_2 - C)$ and $\mathbf{U} = \min(C - \alpha_1, \alpha_2)$, otherwise, $\mathbf{L} = \max(-\alpha_1, -\alpha_2)$ and $\mathbf{U} = \min(C - \alpha_1, C - \alpha_2)$. Note that $\boldsymbol{\Delta^*}$ can not be learnt for arbitrary, but we can still find the global optimum using a variety of methods. Here, we adopt Brent's algorithm but the Newton–Raphson method to compute the gradient (Hessian) of the dual.

$$\nabla_\alpha D = 1 - \sum_k d_k H_k \alpha = 1 - H\alpha \tag{15}$$

$$\nabla^2_\alpha D = -H - \frac{1}{\lambda}\sum_k \nabla_{\theta_k} f^{-1}(\theta)(H_k\alpha)(H_k\alpha)^t \tag{16}$$

where $H_k = YK_kY$, $\nabla_{\theta_\mathbf{k}} f^{-1}(\theta) = (2-q)\theta_q^{2-2q}\theta_k^{2q-q} + (q-1)\theta_q^{2-q}\theta_k^{q-2}$ and $\theta_k = \frac{1}{2\lambda}\alpha^t H_k \alpha$ Instead of computing the gradient $\boldsymbol{\nabla} D$ repeatedly, we speed up variable selection and change $\boldsymbol{\alpha}$ for each kernel $H_k\boldsymbol{\alpha}$ separately. This involves $O(M)$ work in all where M is the number of kernels. The learning process of MKLA is shown in Algorithm 1.

3.4. Top-n song recommendation

In the recommendation stage, we use the learnt latent factor matrices to predict overall ratings of songs and recommend top-n songs the learnt kernel matrix K^*. Note that the distance between two recordings $\Phi(\mathbf{x}_i)$ and $\Phi(\mathbf{x}_j)$ in training set can be directly computed as $K(i,i) + K(j,j) - 2K(i,j)$. We now consider the problem of computing the distance between two new recordings $\Phi(\mathbf{z}_i)$ and $\Phi(\mathbf{z}_j)$. The optimal solution to the kernel learning problem is $K^* = X^\top W^X$, where $W^* = I + XMX^\top$. Then the Mahalanobis distance between $\Phi(\mathbf{z}_i)$ and $\Phi(\mathbf{z}_j)$ can be computed via inner products entirely:

$$\begin{aligned}\Phi(\mathbf{z}_\mathrm{i})^\top \mathrm{W}\Phi(\mathbf{z}_\mathrm{j}) &= \Phi(\mathbf{z}_i)^\top(I + XMX^\top)\Phi(\mathbf{z}_j)\\ &= \Phi(\mathbf{z}_i)\Phi(\mathbf{z}_j) + \Phi(\mathbf{z}_i)^\top XMX^\top\Phi(\mathbf{z}_j)\\ &= \kappa(\mathbf{z}_i,\mathbf{z}_j) + \mathbf{k}_1^\top M\mathbf{k}_2, \text{ where } \mathbf{k_i}\\ &= [\kappa(\mathbf{z_i},\mathbf{x_1}),\ldots,\kappa(\mathbf{z_i},\mathbf{x_n})]^\top.\end{aligned} \tag{17}$$

Thus, Eq. (17) can be used to compute the kernelized distances between recordings with respect to the learnt kernel function. Then decision score of a recordings according to the optimal parameters is

$$f(\mathbf{z}) = \sum_{i=1}^n \alpha_i y_i \left(\sum_{p=1}^m \gamma_p K_p(\mathbf{x}_i,\mathbf{z})\right) + b \tag{18}$$

Therefore, we can recommend top-n songs with the highest overall ratings and labeled as $+1$.

3.5. Bound analysis

In this section, our analysis mainly focus on the automatic selection of representative vocal ratings and our regularized multiple kernel learning. To optimizing the MKLA dual, we use sampling strategy to select the most representative vocal ratings and adopt approximate projections to speed up convergence. Therefore, we present the bound analysis of both sampling approximation and low-rank kernel learning.

Analysis of vocal ratings selection. Kernel methods typically suffer from at least quadratic running-time complexity in the number of observations n, as this is the complexity of computing the kernel matrix. In this work, we propose a selection strategy based on Bernsteins inequality to eliminate redundant vocal ratings.

While selecting a subset is computational efficient, it may not lead to the best performance[11]. Given the matrix $\Psi \in \mathbb{R}^{n\times r}$ and a subset of $\{x_i\}_{i=1}^n$ with p elements chosen without replacement. Let $\Psi_1,\ldots,\Psi_n \in \mathbb{R}^r$ be the n row of Ψ, then we have the submatrix of $\boldsymbol{\Psi}$. Specially, we first define the matrix $\triangle \in \mathbb{R}^{r\times r}$:

$$\triangle = \frac{1}{n}\Psi^\top\Psi - \frac{1}{p}\Psi_I^\top\Psi_I = \frac{1}{n}\sum_{i=1}^n \Psi_i\Psi_i^\top - \frac{1}{p}\sum_{i\in I}\Psi_i\Psi_i^\top.$$

As shown in [12] and [32], we have $\mathbb{E}\Delta = 0$. Then we extends the matrix case to the classical result of Hoeffding [29] and have:

$$\sum_{j=1}^p M_j = \frac{1}{p}\sum_{j=1}^p\left(\frac{1}{n}\sum_{i=1}^n \Psi_i\Psi_i^\top - \sum_{i=1}^n z_i^j\Psi_i\Psi_i^\top\right)$$

where $z^j \in \mathbb{R}^n$ is a random element such that $\mathbb{P}(z_i^j = 1) = \frac{1}{n}$ for all $i \in 1,\ldots,n$ and $j \in 1,\ldots,p$. Since $\mathbb{E}M_j = 0$ and $\lambda_{max}(\frac{1}{n}\Psi^\top\Psi)/p$, we have

$$\begin{aligned}\lambda_{max}\left(\sum_{j=1}^p \mathbb{E}M_j^2\right) &\le \frac{1}{p}\lambda_{max}\left(\frac{1}{n}\sum_{i=1}^n \Psi_i\Psi_i^\top\Psi_i\Psi_i^\top\right)\\ &\le \frac{R^2}{p}\lambda_{max}\left(\frac{1}{n}\Psi^\top\Psi\right)\end{aligned} \tag{19}$$

Applying the matrix Bernstein inquality [1], we obtain the probability bound:

$$r\exp\left(-\frac{t^2/2}{\frac{R^2}{p}\lambda_{max}\left(\frac{1}{n}\Psi^\top\Psi\right) + \frac{1}{p}\lambda_{max}\left(\frac{1}{n}\Psi^\top\Psi\right)\frac{t}{3}}\right) \tag{20}$$

Along this line, the bound of sampling approximation shows the relation of the number of the selected vocal ratings and the accuracy of song recommendation.

Bound of Regularized Low-rank Approximation. In the training stage, we incorporate bregman divergence as a regularization for low-rank kernel learning. Therefore, the bound of low-rank kernel learning shows that the maximal marginal degrees of freedom provides a quantity which, up to logarithmic terms, is sufficient to scale with, in order to incur no loss of prediction performance.

Let $\Phi \in \mathbb{R}^{n\times n}$ such that $K = \Phi\Phi^\top$. If K had rank r, then we can instead choose $\Phi \in \mathbb{R}^{n\times r}$ [11]. We now consider the regularized low-rank approximation $L_\gamma = \Phi N_\gamma \Phi^\top$, where

$$\begin{aligned}N_\gamma &= \Phi_I^\top(\Phi_I\Phi_I^\top + p\gamma I)^{-1}\Phi_I = \Phi_I^\top\Phi_I(\Phi_I^\top\Phi_I + p\gamma I)^{-1}\\ &= I - \gamma(\Phi_I^\top\Phi_I/p + \gamma I^{-1})\end{aligned} \tag{21}$$

Note that $\Psi = \Phi(\frac{1}{n}\Phi^\top\Phi + \gamma I)^{-1/2} \in \mathbb{R}^{n\times n}$, we can rewrite N_γ of Eq. (21) as

$$N_\gamma = I - \left(\gamma\frac{1}{p}\Phi^\top\Phi + \gamma I\right)^{-1} \tag{22}$$

In order to obtain a lower-bound on N_γ, it suffices to have an upper-bound of the form:

$$\lambda_{max}\left(\frac{1}{n}\Phi^\top\Phi - \frac{1}{p}\Phi_I^\top\Phi_I\right) \le t \tag{23}$$

Assume $\frac{\gamma/\lambda}{1-t} \le 1$, and define $\pi_t = \mathbb{P}_I\left(\lambda_{max}\left[\frac{1}{n}\Phi^\top\Phi - \frac{1}{p}\Phi_I^\top\Phi_I\right] > t\right)$, then the lower bound is:

$$B = \pi_t(1 + R^2/\lambda) + (1-\pi_t)\left(1 - \frac{\gamma/\lambda}{1-t}\right)^{-2} \tag{24}$$

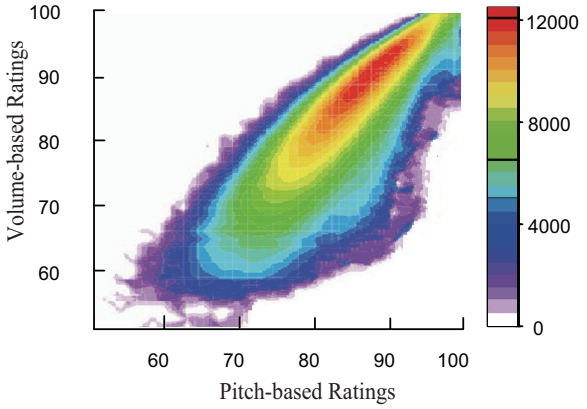


Fig. 2. The distribution of singing recordings w.r.t average of pitch-based ratings and volume-based ratings.

Table 1
Statistics of the data set.

# Users	# Positive songs	# Negative songs	# Vocal features
28,472	669,890	96,761	213

The bound of sampling approximation shows the relation of the number of the selected vocal ratings and the accuracy of song recommendation. In the training stage, we incorporate bregman divergence as a regularization for low-rank kernel learning. Therefore, the bound of low-rank kernel learning shows that the maximal marginal degrees of freedom provides a quantity which, up to logarithmic terms, is sufficient to scale with, in order to incur no loss of prediction performance.

4. Experiments

In this section, we present the experimental results to illustrate the performance of our method on real-world karaoke data. All computations were carried out on a 16-core Intel Xeon machine with 16 GB of memory. All the codes were written in Matlab.

4.1. Experiment setup

Data Description. We evaluate our method on the real world karaoke data from August 2011 to June 2012. To alleviate the sparsity problem, we only consider the songs which have been sung more than 3 different users and users who have sung more than 10 songs. In the Fig. 2, we can observe that more than 80% users can perform an overall rating more than 70. By applying a rating threshold $\sigma = 70$, the song recommendation is reduced into a binary classification problem. A song is a positive sample, if the overall rating is more than σ, otherwise a negative sample. Table 1 presents the statistics of the data used in the experiments.

Comparison Algorithms. we evaluate the following algorithms.

- AveKernel [21]: AveKernel is the average combination of the base kernels. Specifically, the combination coefficients is given by $\mu = \frac{1}{M}$ and the maximum margin classifier is learnt by SVM.
- $\boldsymbol{\ell_p}$-MKL [19]: The classifier and kernel combination coefficients are optimized under the constraint $\|\mu\|_2 \leq 1$.
- SimpleMKL [31]: The classifier and the kernel combination coefficients are optimized by solving the $\boldsymbol{\ell_1}$-MKL problem.
- WEMAREC [10]:WEMAREC is a weighted and ensemble matrix approximation method for accurate and scalable recommendation.
- S-LMKL [17]: S-LMKL is sample-wise alternating optimization algorithm for training localized multiple kernel learning.

The regularization parameter C in AveKernel, L_p-MKL ($p = 2$) and WEMAREC is chosen from a sufficiently large range $\{2^{-5}, 2^{-3}, \ldots, 2^{15}\}$ according to [16] by four-fold cross-validation on each training set.

The base kernels are the same as those in [35], which include ten Gaussian kernels with the widths of [0.5, 1, 2, 5, 7, 10, 12, 15, 17, 10] and ten polynomial kernels with degrees of one to ten. For our proposed MKLA, we take advantage of variance information of different kernels. Here, we present the definition of kernel variance:

Definition 1 (Kernel Variance)**.** Consider $\mathbf{K}_{\boldsymbol{\mu}}$ is the non-negative linear combination of μ base kernels and a training data set $(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N) \in \mathbf{R}^d \times \pm 1$,where the first N_1 data belongs to $+1$ and the following data belongs to -1. The kernel variance of $\mathbf{K}_{\boldsymbol{\mu}}$ is:

$$V_{\mathbf{K}_\mu} = \begin{pmatrix} \frac{\mathbf{K_{11}}}{N_1} & 0 \\ 0 & \frac{\mathbf{K_{22}}}{N_2} \end{pmatrix} - \begin{pmatrix} \frac{\mathbf{K_{11}}}{N} & \frac{\mathbf{K_{12}}}{N} \\ \frac{\mathbf{K_{21}}}{N} & \frac{\mathbf{K_{22}}}{N} \end{pmatrix}, \quad \textit{where}$$
$$\mathbf{K}_\mu = \begin{pmatrix} \mathbf{K_{11}} & \mathbf{K_{12}} \\ \mathbf{K_{21}} & \mathbf{K_{22}} \end{pmatrix} \tag{25}$$

Among 50 different base kernels, we select five kernels with the highest variance for each data set. To be consistent with previous works, the experiments for different MKL algorithms are all based on the C-SVC formulation.

4.2. Performance comparison

The results of classification accuracy and corresponding time cost are listed in Table 3, where the highest accuracy and those whose difference from the highest accuracy are not statistically significant are shown in bold for each data set. From these experimental results, we have the following observations.

(1) *Classification accuracy*. In Table 2, proposed MKLA achieves the overall best classification performance in two data sets. In addition, as shown in Fig. 4, with the number of new rating features increasing, the proposed MKLA method can achieve better performance. In this way, we can prove the effectiveness of the new rating features. Then, the sample sizes of the two data sets are relatively large, so our proposed algorithm takes advantage of the variance information of samples and is the least affected by the sample compression. The main difference of these algorithms lies at that our MKLA is not susceptible to the rank of kernel matrix, overfitting, and the size of noise. Our method can further improve the recommendation accuracy than WEMAREC and S-LMKL, since the appeared ratings of songs are treated more importantly in the the new low-rank matrix approximation method. Another significant difference between MKLA and WEMAREC is the construction of submatrices. In WEMAREC, each submatrix is constructed by random sampling, while in MKLA the submatrix is made of selected ratings based on Bernsteins inequality. In addition, the eigenvalues of the low-rank kernels in MKLA preserve the structural information from the full-rank kernel matrix, while tr($\mathbf{K}$) is used in AveKernel, SimpleMKL and ℓ_p-MKL. The aforementioned experimental results indicate the efficiency of employing Bregman divergence in our MKLA to incorporate the structural information.

(2) *Robustness*. Although our algorithm shows degraded classification accuracy when C is 2^3 and 2^{-1}, the difference from

Table 2
Classification accuracy comparison (mean ± *Standard Derivation*) on different regularization parameters. Result in boldface means the best one.

	Classification accuracy					
	AveKernel	ℓ_2-MKL	SimpleMKL	WEMAREC	S-LMKL	MKLA
$C = 2^{-5}$	**83.6 ± 3.1**	68.8 ± 3.7	81.5 ± 7.9	35.6 ± 6.7	51.2 ± 3.1	79.6 ± 4.1
$C = 2^{-1}$	73.6 ± 1.8	75.7 ± 1.3	74.4 ± 1.0	72.6 ± 2.2	56.6 ± 1.0	**75.7 ± 1.1**
$C = 2^{3}$	**70.1 ± 1.8**	66.5 ± 2.4	63.1 ± 2.6	67.5 ± 2.3	64.1 ± 2.5	63.2 ± 3.5
$C = 2^{7}$	79.4 ± 3.6	**83.9 ± 1.1**	81.5 ± 3.1	82.6 ± 2.1	73.5 ± 3.1	80.2 ± 2.7
$C = 2^{11}$	96.3 ± 0.7	76.9 ± 1.3	97.5 ± 0.6	76.3 ± 1.2	93.5 ± 0.6	**97.5 ± 2.4**
$C = 2^{15}$	87.3 ± 1.6	87.9 ± 1.7	**90.1 ± 1.1**	76.5 ± 0.4	60.1 ± 2.1	88.2 ± 1.9

Table 3
Comparison of the running time of the different kernel learning algorithms using the weighted degree kernel.

	Cross-validation/Training time					
	AveKernel	ℓ_2-MKL	SimpleMKL	WEMAREC	S-LMKL	MKLA
$C = 2^{-5}$	12.7/0.1	7.8/2.6	3.8/0.1	4.9/0.5	5.8/0.3	**1.6/0.1**
$C = 2^{-1}$	397.1/0.6	477.5/42.8	240.8/14.1	281.7/14.2	307.2/17.9	**137/20.6**
$C = 2^{3}$	58.2	57.2/2.1	11.9/0.5	13.2/0.8	12.5/0.7	**9.4/0.3**
$C = 2^{7}$	32.8/0.1	30.2/0.9	9.8/0.6	11.4/0.7	14.6/0.9	**5.7/0.6**
$C = 2^{11}$	83.8/0.1	75.7/1.9	42.7/2.3	48.6/2.8	51.4/3.8	**32.3/1.8**
$C = 2^{15}$	105.3/0.2	104.7/5.1	24.5/2.0	26.3/5.1	28.2/6.0	**21.1/4.7**

the highest accuracy is still not statistically significant. By assigning automatic selection of vocal ratings, the advantage of variance information is the largest with relatively high regularization parameter. This is expected because the samples and outliers may come from different distributions. Note that our approximation error bound applies to most kernel functions (Section 3.5), and preliminary experimental results with these kernels have shown the superiority of our sampling scheme compared with other low-rank approximation methods in Fig. 6. Therefore the variance information can be exploited in a principled manner.

(3) *Computational efficiency.* Table 3 includes cross-validation time and training time, which demonstrates the computational efficiency of our MKLA. The average time costs by AveKernel, ℓ_2-MKL and WEMAREC are about 114.9, 120.4 and 55.6 times longer than those used by MKLA. Comparing with AveKernel, S-LMKL, ℓ_2-MKL and SimpleMKL, MKLA is less affected by the outliers and better focused on the structure of distribution in the data sets. Moreover, comparing with ℓ_2-MKL and SimpleMKL, our MKLA utilize SMO method instead of sample-wise optimization and also avoids compute sparsity kernels in each iteration. These two factors make our proposed MKLA more computationally efficient than the other algorithms.

In sum, the aforementioned experimental results indicate that our MKLA can handle large size data sets and achieve good performance among the algorithms in comparison. In term of classification performance to time cost, the proposed MKLA is clearly the best one.

Performance of low-rank approximations. In this series of experiments, we compute the rank p which is necessary to achieve a predictive performance at most 1% than with $p = n$. The Bregman divergence is added to all the algorithms as a regulariser, so we can compute the classical generalization performance. Fig. 5 plots classification results as the rank of kernel is varied on the data sets. The trend is the same when we decrease the rank of kernel. We can find out that the MKLA appears to be relatively stable across the operating range of p. In most of the cases as expected, the algorithm does not appear to be significantly worse for other values of p. Therefore, it is hoped that MKLA can be used to learn sparse kernel combinations as well as non-sparse ones. In addition to its low runtime complexity, MKLA is also capable of benefiting from sparse matrix structure. For such matrices, adaptive methods requires the computation of $n \times n$ residuals, which may be dense even in the case that rating matrix is extremely sparse. In contrast, MKLA requires only the storage of much smaller $\ell \times n$ matrices. This benefit of MKLA is highly relevant for extremely large datasets where sparse approximations to similarity matrices are formed using rating selection algorithms that only store the most significant entries in each rating matrix.

4.3. Effectiveness of bregman divergence

In this subsection, we show the interpretable nature of our proposed method for discriminative analysis in karaoke song recommendation. A spectrogram is a visual representation of the spectrum of frequencies in a sound Fig. 3 shows several examples of acoustic analysis by spectrogram. Since spectrogram is a visual representation of the spectrum of frequencies, we can extract spectrogram by applying a 512-point STFT [8]. We observe that our method can preserve the singing spectra structures and the bregman divergence enhance the classification performance. It mainly owes that the best performance of MKLA is not limited to the best kernel in the candidate set, and can characterize the geometrical structure of different aspects for the singing records. Therefore, the proposed method captures the invariable spectro-temporal structures in audio signals.

4.4. Effectiveness of latent vocal presentation

In this subsection, we show the interpretable nature of our proposed method for discriminative analysis in karaoke song recommendation. Let $\mathbf{K}_1, \mathbf{K}_2, \ldots, \mathbf{K}_p$ denote p kernel matrices of latent vocal features learnt from the optimization problem. Here, we select the kernel matrix with highest weight, i.e., $\mathbf{K}_1$ and present some interpretation of $\mathbf{K}_1$.

Visualization of latent vocal features. Each row in the rating matrix $\mathbf{K}$ can be treated as vocal features for a user. For example, $\mathbf{K}_{(j,:)}$ can be viewed as a vector of latent features of jth user. To illustrate $\mathbf{K}^{(1)}$ in a 2-D figure, we adopt t-SNE[1], which is

[1] http://homepage.tudelft.nl/19j49/t-SNE.html

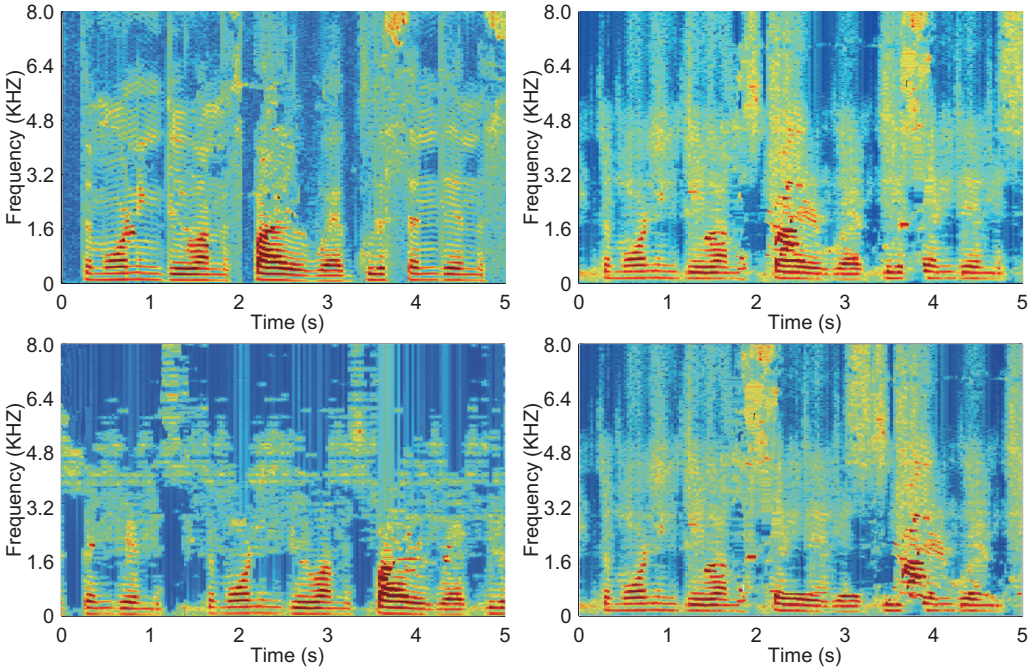


Fig. 3. The spectrogram of the singing recordings (in log scale).

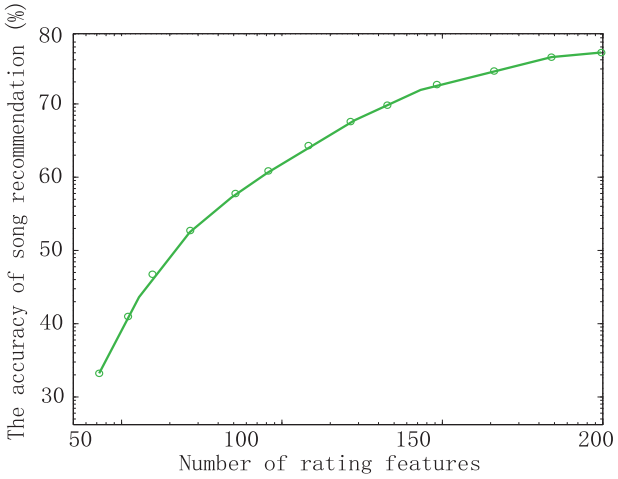


Fig. 4. Average accuracy of additional rating features on the karaoke song recordings.

commonly used for the visualization of high-dimensional data [39], to assist our data analysis. Then $\mathbf{K}_1$ is illustrated in Fig. 7(a), where each circle corresponds to a song and its size is proportional to the singing frequency in the song dataset. Then, we randomly select four types of songs and highlight them by orange circles, as shown in Fig. 7(b). We observe that the songs of a type are much more easily to be concentrated. Therefore, given a set of karaoke songs, our method can identify vocal features and simultaneously classify the songs of different types (e.g., Rock, Pop and Classic).

5. Related work

In the literature, a lot of methods have been proposed to address song recommendations, such as [27,41]. Traditional song recommendation systems are proposed for discovering songs which satisfy users' listening interest. [5,40] proposes a content-based model which uses low level features, such as moods and rhythms, to represent user's preference of the songs. In recent years, recommender systems are mainly dominated by content-based and collaborative filtering approaches. Content-based (CB) recommender systems learn the user's preference for specific types of songs by analyzing the songs' descriptions. The prediction of the unrated songs is based on ratings for similar songs rated by the same user. In Collaborative Filtering (CF) strategies, the prediction of the unrated songs is based on the opinion of users with similar tastes. Most of the work in recommender systems has focused on recommending the most relevant items to individual users [13], but the circumstance of the user typically is not considered when the recommendations take place.

On the other hand, kernel methods are also applied to perform these prediction [15,19]. The space and computational complexity of MKL methods mainly depend on the number of training samples and the computational complexity of the base learner [24]. Existing MKL algorithms learn the weights of base kernels based on a training set [8]. Many of the sampling techniques use m active data points selected from the training set of size $n(>m)$. But it is impossible to find the optimal subset due to combinatorics. The active data points could be selected randomly, but in general, a better performance can be expected if the points are selected according some criterion [33][29]. Instead of choosing all points in one single draw before the experiment, adaptive sampling strategies [26] adjust the sampling grid iteratively to the complexity of sample space. While these sampling algorithms have proven effective in large-scale data sets, they can be improved by incorporating variance information. Alternatively, [25,36] uses sampling techniques to generate low-rank kernel matrix. Research in this field is mainly based on classical theoretical results [26] , which shows that the error of a subset of k columns approximation can be bounded by the optimal rank-k matrix. To speed up kernel algorithms, a variety of sampling schemes have been used theoretically [8,21,30]. For example, Incomplete Cholesky Decomposition associated with the Nyström, can be viewed as a specific sampling method. As noted in [23], the Nyström approximation attempts to complete a low-rank matrix with its random entries. In this paper, we have access to the underlying approximation function and

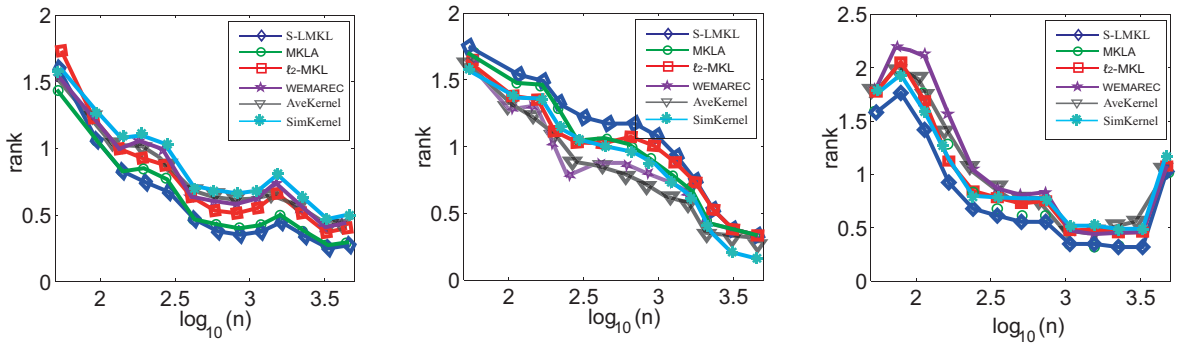


Fig. 5. Ratio of the sufficient rank to obtain 1% worse predictive performance. From left to right: regularization parameter $C = 2^{-1}, C = 2^{3}$ and $C = 2^{8}$.

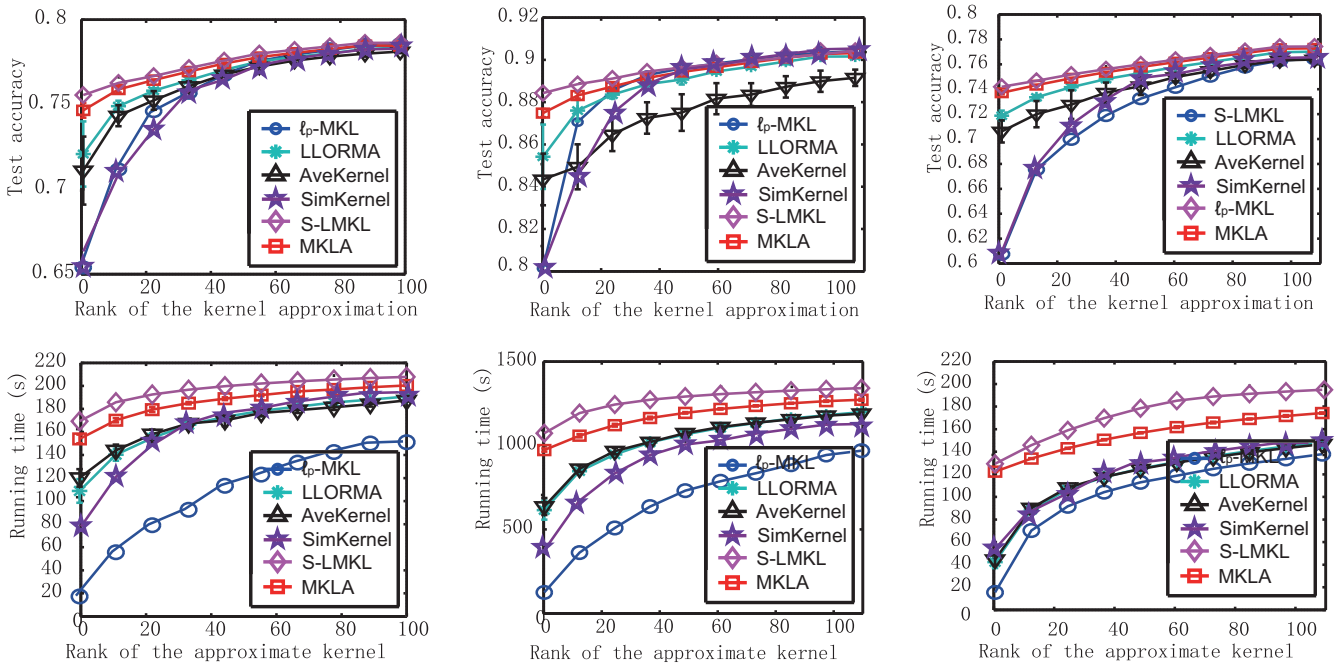
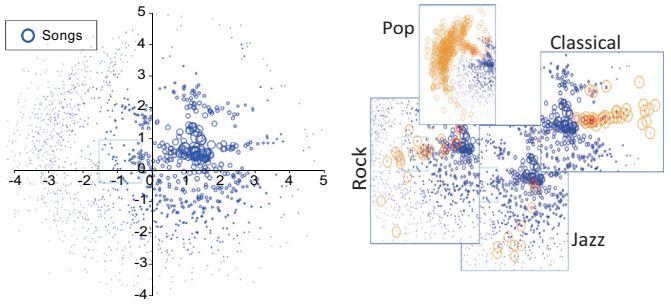


Fig. 6. Ratio of the sufficient rank to obtain 1% worse predictive performance. From left to right: regularization parameter $C = 2^{-1}, C = 2^{3}$ and $C = 2^{8}$.



(a) Latent Factor $\mathbf{K}_1$ in $\mathbb{R}^2$ (b) Four Genres of Songs

Fig. 7. Each song is presented as a blue circle. With four different genres of songs highlighted one by one, we can observe that songs of the same genre are easily concentrated.

then we can generate desired matrix entries on-the fly. Moreover, we show how to employ the empirical bounds which is recently introduced by [1].

Low rank techniques are often employed to ease the burden of memory in a lot of machine learning problems [7]. In addition, incomplete Cholesky decomposition has been employed in low-rank kernel representation [2,22]. For example, in [35][23], incomplete Cholesky decomposition is used for classification and embedding problems. Other examples use low-rank decompositions to speed up kernel learning algorithms such as [21] and [2]. In our paper, we focus on learning a low-rank kernel matrix by using similarity and constrains on distance. In recent work, kernel learning methods use semi-definite programming. [21] learns kernel matrices with the selected given kernels when the labels are given. Previous work attempts to recommend songs that is perceptually similar to what users have previously listened to, by measure the similarity between the audio signals. The similarity metrics are usually defined ad hoc, by incorporating prior knowledge about music audio [28]. To assess the required precision in approximating kernel matrix, a key is to understand the typical predictive performance of kernel methods. For example, the performance of the square loss can be obtained from its bias-variance decomposition. Moreover, the degree of freedom also plays an important role of an implicit number of parameters [37]. It is applicable to many non-parametric estimation methods consisting in smoothing the response vector by a linear operator.

Karaoke singing recommendation is relatively a new area, because users' singing skills, such as pitch, volume and rhythm should be taken into account in the karaoke song

recommendations. However, karaoke songs typically contain background accompaniments and it does not make sense to directly compare users' singing performance with the original song recordings. To tackle this problem, [34] proposed a learning-to-rank scheme for recommending songs based on an analysis of singer's vocal competence. They require a professional recording process to extract users' singing characteristics, namely singer profiles, and build a learning-to-rank model recommending songs matching users' vocal competence. There are two major drawbacks in this system: one is that it requires a complex vocal competence extraction process; the other is that it does not consider users' potential ability. For example, users' singing skill will improve even their performance scores are not good in the singing history.

6. Conclusion

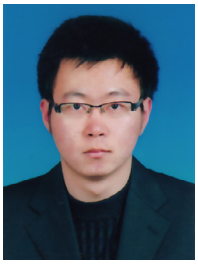
In this paper, we proposed a joint modeling method for karaoke recommendation by mining history karaoke singing records. Specially, we first defined and extracted multi-aspect vocal (i.e., pitch, volume, and rhythm) ratings of users for songs based on their records. Since we need to learn the representations of the vocal competence of users, we exploited a multiple kernel learning method to map vocal ratings in a high-dimensional feature space. Besides, to enhance the efficiency of our method, we developed a sample compression method based on the empirical Bernstein bound and incorporated the Bregman divergence as a regularizer in the MKLA formulation. Furthermore, we devised an effective method to solve the joint objective function, to place the emphasis on optimizing the MKL dual and moreover, to effectively recommend karaoke songs. Finally, extensive experiments with real-world online karaoke data demonstrated the effectiveness of the proposed method comparing to the state-of-the-art benchmark algorithms.

Acknowledgments

This research was partially supported by grants from the National Key Research and Development Program of China (Grant No. 2016YFB1000904), the National Science Foundation for Distinguished Young Scholars of China (Grant No. 61325010) and the Fundamental Research Funds for the Central Universities of China (Grant No. WK2350000001).Professor Hui Xiong is partially supported by National Natural Science Foundation of China (71531001).

References

[1] F.R. Bach, Sharp analysis of low-rank kernel matrix approximations, in: COLT, 30, 2013, pp. 185–209.
[2] F.R. Bach, M.I. Jordan, Predictive low-rank decomposition for kernel methods, in: Proceedings of the 22nd International Conference on Machine Learning, 2005.
[3] P.L. Bartlett, O. Bousquet, S. Mendelson, Local rademacher complexities, Ann. Stat. 33 (4) (2005) 1497–1537.
[4] D.J. Berndt, J. Clifford, Using dynamic time warping to find patterns in time series, in: Proceedings of the KDD Workshop, 10, 1994, pp. 359–370.
[5] D. Bogdanov, M. Haro, F. Fuhrmann, A. Xambó, E. Gómez, P. Herrera, Semantic audio content-based music recommendation and visualization based on user preference examples, Inf. Process. Manag. 49 (1) (2013) 13–33.
[6] J.K. Bradley, R.E. Schapire, Filterboost: regression and classification on large datasets, in: Proceedings of the Advances in Neural Information Processing Systems, 2007.
[7] J. Cai, Y. Tang, J. Wang, Kernel canonical correlation analysis via gradient descent, Neurocomputing 182 (2015) 322–331.
[8] P. Castro, E. Petit, A. Farjallah, W. Jalby, Adaptive sampling for performance characterization of application kernels, in: Proceedings of the Concurrency and Computation: Practice and Experience, 2013.
[9] C.-C. Chang, C.-J. Lin, Libsvm: a library for support vector machines, in: Proceedings of the ACM Transactions on Intelligent Systems and Technology (TIST), 2011.
[10] C. Chen, D. Li, Y. Zhao, Q. Lv, L. Shang, Wemarec: accurate and scalable recommendation through weighted and ensemble matrix approximation, in: Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, 2015, pp. 303–312.
[11] R. Chitta, R. Jin, T.C. Havens, A.K. Jain, Approximate kernel k-means: solution to large scale kernel clustering, in: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2011.
[12] C. Cortes, M. Kloft, M. Mohri, Learning kernels using local rademacher complexity, in: Proceedings of the Advances in Neural Information Processing Systems, 2013.
[13] Y. Fu, H. Xiong, Y. Ge, Z. Yao, Y. Zheng, Z.-H. Zhou, Exploiting geographic dependencies for real estate appraisal: a mutual perspective of ranking and clustering, in: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2014, pp. 1047–1056.
[14] W. Gao, Z.-H. Zhou, On the doubt about margin explanation of boosting, Artif. Intell. 203 (2013) 1–18.
[15] M. Gönen, E. Alpaydin, Localized multiple kernel learning, in: Proceedings of the 25th International Conference on Machine Learning, ACM, 2008.
[16] M. Gönen, E. Alpaydın, Multiple kernel learning algorithms, J. Mach. Learn. Res. 12 (2011) 2211–2268.
[17] Y. Han, K. Yang, Y. Ma, G. Liu, Localized multiple kernel learning via sample–wise alternating optimization, IEEE trans. Cybern. 44 (1) (2014) 137–148.
[18] P. Jain, B. Kulis, I.S. Dhillon, Inductive regularized learning of kernel functions, in: Proceedings of the Advances in Neural Information Processing Systems, 2010, pp. 946–954.
[19] M. Kloft, U. Brefeld, S. Sonnenburg, A. Zien, Lp-norm multiple kernel learning, J. Mach. Learn. Res. (2011) 953–997.
[20] B. Kulis, M.A. Sustik, I.S. Dhillon, Low-rank kernel learning with Bregman matrix divergences, J. Mach. Learn. Res. 10 (2009) 341–376.
[21] A. Kumar, A. Niculescu-Mizil K. Kavukcuoglu, H. Daumé, A binary classification framework for two-stage multiple kernel learning, 2012.
[22] A. Kumar, P. Rai, H. Daumé III, Co-regularized spectral clustering with multiple kernels, in: Proceedings of the NIPS 2010 Workshop: New Directions in Multiple Kernel Learning, 2010.
[23] S. Kumar, M. Mohri, A. Talwalkar, Sampling methods for the Nyström method, J. Mach. Learn. Res. (2012) 981–1006.
[24] B. Liu, S.-X. Xia, Y. Zhou, Unsupervised non-parametric kernel learning algorithm, Knowl.-Based Syst. 44 (2013) 1–9.
[25] W. Liu, B. Qian, J. Cui, J. Liu, Spectral kernel learning for semi-supervised classification, in: Proceedings of the 21st International Joint Conference on Artifical Intelligence IJCAI, 2009, pp. 1150–1155.
[26] X. Liu, L. Wang, J. Zhang, J. Yin, Sample-adaptive multiple kernel learning, Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, AAAI Press, 2014, pp. 1975–1981.
[27] K. Mao, L. Shou, J. Fan, G. Chen, M. Kankanhalli, Competence-based song recommendation: matching songs to one's singing skill, IEEE Transactions on Multimedia 17 (3) (2013) 396–408.
[28] J. Mekyska, E. Janousova, P. Gomez-Vilda, Z. Smekal, I. Rektorova, I. Eliasova, M. Kostalova, M. Mrackova, J.B. Alonso-Hernandez, M. Faundez-Zanuy, et al., Robust and complex approach of pathological speech signal analysis, Neurocomputing 167 (2015) 94–111.
[29] V. Mnih, C. Szepesvári, J.-Y. Audibert, Empirical Bernstein stopping, in: Proceedings of the 25th International Conference on Machine Learning, ACM, 2008.
[30] D. Paulin, L. Mackey, J.A. Tropp, Deriving matrix concentration inequalities from kernel couplings, 2013, ArXiv preprint arXiv: 1305.0612.
[31] A. Rakotomamonjy, F. Bach, S. Canu, Y. Grandvalet, Simplemkl, J. Mach. Learn. Res. 9 (2008) 2491–2521.
[32] A. Rakotomamonjy, S. Chanda, Lp-norm multiple kernel learning with low-rank kernels, Neurocomputing 143 (2014) 68–79.
[33] P.K. Shivaswamy, T. Jebara, Variance penalizing adaboost, in: Proceedings of the advances in Neural Information Processing Systems, 2011, pp. 1908–1916.
[34] L. Shou, K. Mao, X. Luo, K. Chen, G. Chen, T. Hu, Competence-based song recommendation, in: Proceedings of the 36th International ACM SIGIR Conference, 2013, pp. 423–432. ACM
[35] S. Si, C.-J. Hsieh, I. Dhillon, Memory efficient kernel approximation, in: Proceedings of the 31st International Conference on Machine Learning, 2014.
[36] V. Sindhwani, A.C. Lozano, Non-parametric group orthogonal matching pursuit for sparse learning with multiple kernels, in: Proceedings of the Advances in Neural Information Processing Systems, 2011, pp. 2519–2527.
[37] Z. Sun, N. Ampornpunt, M. Varma, S. Vishwanathan, Multiple kernel learning and the SMO algorithm, in: Proceedings of the Advances in Neural Information Processing Systems, 2010.
[38] W.-H. Tsai, H.-C. Lee, Automatic evaluation of karaoke singing based on pitch, volume, and rhythm features. audio, speech, and language processing, IEEE Trans. 20 (4) (2012) 1233–1243.
[39] L. Van der Maaten, G. Hinton, Visualizing data using t-SNE, J. Mach. Learn. Res. 9 (2579–2605) (2008) 85.
[40] L. Wu, Q. Liu, E. Chen, N.J. Yuan, G. Guo, X. Xie, Relevance meets coverage: a unified framework to generate diversified recommendations, ACM Trans. Intell. Syst. Technol.(TIST) 7 (3) (2016) 39.
[41] X. Wu, Q. Liu, E. Chen, L. He, J. Lv, C. Cao, G. Hu, Personalized next-song recommendation in online karaokes, in: Proceedings of the 7th ACM Conference on Recommender Systems, ACM, 2013, pp. 137–140.
[42] J. Zhuang, I.W. Tsang, S.C. Hoi, A family of simple non-parametric kernel learning algorithms, J. Mach. Learn. Res. 12 (2011) 1313–1347.



Chu Guan received his B.E. degree in Computer Science from Northeastern University. In 2011, he was recommended to University of Science and Technology of China with an exemption from examination as a graduate student and received his M.E. degree in School of Computer Science and Technology, University of Science and Technology of China in 2013. Now, he is a Ph.D. candidate under advisory of Prof. Enhong Chen in the School of Computer Science and Technology and Laboratory of Semantic Computing.

Yanjie Fu received the B.E. degree from the University of Science and Technology of China (USTC), Hefei, China, 2008, the M.E. degree from the Chinese Academy of Sciences (CAS), Beijing, China, 2011, and the Ph.D. degree from Rutgers University. He is currently an assistant professor at Missouri University of Science and Technology. His research interests include data mining, urban computing, mobile intelligence, and personalization techniques. He has published in refereed journals and conference proceedings, such as TKDE, TKDD, TMC, KDD, ICDM, SDM.

Xinjiang Lu is currently a Ph.D. candidate in Computer Science at Northwestern Polytechnical University, Xi'an, China. He received the M.S. degree in Software Engineering from Northwestern Polytechnical University in 2011, and the B.E. degree in Computing Mathematics from Xinjiang University, Urumqi, China, 2007. His research interests include data mining and mobile intelligence.

Enhong Chen received the BS degree form Anhui University, master's degree from the Hefei University of Technology and the Ph.D. degree in computer science from USTC. He is currently a professor and the vice dean of the School of Computer Science, the vice director of the National Engineering Laboratory for Speech and Language Information Processing of University of Science and Technology of China (USTC), winner of the National Science Fund for Distinguished Young Scholars of China. His research interests include data mining and machine learning, social network analysis and recommender systems. He has published lots of papers on refereed journals and conferences, including IEEE Transactions on Knowledge and Data Engineering, IEEE Transactions on Mobile Computing, KDD, ICDM, NIPS, and CIKM. He was on program committees of numerous conferences including KDD, ICDM, SDM. He received the Best Application Paper Award on KDD-2008 and Best Research Paper Award on ICDM-2011. He is a senior member of the IEEE.

Xiaolin Li received her Ph.D. degree in computer science from School of Computer Science and Technology, Jilin University, China in 2005. She joined Department of Computer Science and Technology of Nanjing University from 2005 to 2007 as a postdoctor. Currently she is an associate Professor at Department of Marketing and E-Business, School of Management, Nanjing University , China. Her current research interests include data mining, business intelligence , decision making.

Hui Xiong (SM'07) is currently a Full Professor and Vice Chair of the Management Science and Information Systems Department, and the Director of Rutgers Center for Information Assurance at the Rutgers, the State University of New Jersey, where he received a two-year early promotion/tenure (2009), the Rutgers University Board of Trustees Research Fellowship for Scholarly Excellence (2009), and the ICDM-2011 Best Research Paper Award (2011). He received the B.E. degree from the University of Science and Technology of China (USTC), China, the M.S. degree from the National University of Singapore (NUS), Singapore, and the Ph.D. degree from the University of Minnesota (UMN), USA. His general area of research is data and knowledge engineering, with a focus on developing effective and efficient data analysis techniques for emerging data intensive applications. He has published prolifically in refereed journals and conference proceedings (3 books, 60+ journal papers, and 90+ conference papers). He is a co-Editor-in-Chief of Encyclopedia of GIS, an Associate Editor of IEEE Transactions on Data and Knowledge Engineering (TKDE), IEEE Transactions on Big Data (TBD), ACM Transactions on Knowledge Discovery from Data (TKDD), and ACM Transactions on Management Information Systems (TMIS). He has served regularly on the organization and program committees of numerous conferences, including as a Program Co-Chair of the Industrial and Government Track for the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), a Program Co-Chair for the IEEE 2013 International Conference on Data Mining (ICDM), and a General Co-Chair for the IEEE 2015 International Conference on Data Mining (ICDM). He is an ACM Distinguished Scientist and a senior member of the IEEE.