Vocal Competence Based Karaoke Recommendation: A Maximum-Margin Joint Model

Chu Guan* Yanjie Fu† Xinjiang Lu‡ Hui Xiong† Enhong Chen* Yingling Liu*

Abstract

In online karaoke, the decision process in choosing a song is different from that in music radio, because users usually prefer songs that meet their vocal competence besides their tastes. Traditional music recommendation methods typically model users' personalized preference for songs in terms of content and style. However, this can be improved by considering the degree of matching the vocal competence (e.g. pitch, volume, and rhythm) of users to the vocal requirements of songs. To this end, in this paper, we develop a karaoke recommender system by incorporating vocal competence. Along this line, we propose a joint modeling method named CBNTF by exploiting the mutual enhancement between non-negative tensor factorization (NTF) and support vector machine (SVM). Specifically, we first extract vocal (i.e., pitch, volume, and rhythm) ratings of a user for a song from his/her singing records. Since these vocal ratings encode users' vocal competence from three aspects, we treat these vocal ratings as a tensor, exploit an NTF method, and learn the latent features of users' vocal metrics. These factorized features are simultaneously fed into an SVM classifier and then we use the trained classifier to predict the overall rating of a user with respect to a song. In addition, we propose an enhanced objective function to exploit the mutual enhancement between NTF and SVM, and devise an effective method to solve this objective as a coupled least-squares optimization problem via a maximum margin framework. With the estimated model, we compute the similarity between users and songs in terms of pitch, volume and rhythm and recommend songs to users. Finally, we conduct extensive experiments with real-world online karaoke data. The results demonstrate the effectiveness of our method.

Keywords: Karaoke-song recommendation, Singing competence, Non-negative tensor factorization

1 Introduction

Online karaoke is a music service that allows users to sing karaoke, practice singing, distribute recordings, and challenge friends. Users can access online karaoke service with only a microphone and a computer connected to the internet. Nowadays, online karaoke is increasingly popular as a social entertainment platform, and a large amount of karaoke songs are available. Thus, karaoke recommendation is important because it can help these users identify appropriate karaoke songs, receive high ratings, and moreover, improve karaoke experience.

Unlike classic music recommendation, online karaoke has unique characteristics [1]. For example, if one receives a high overall rating on a karaoke song, he/she probably not just favors the song but also has vocal competence to sing the song well. Thus, when choosing karaoke songs, users often care about whether their vocal competence meets the vocal requirements of songs. The unique characteristics of online karaoke provide us an opportunity to exploit vocal competence for enhancing karaoke recommendation. However, this is a non-trivial task. There are three major challenges. First, as historical karaoke singing records encode the information about users' vocal competentence [2], careful methods need to be designed to learn the representations of users' vocal competence from these singing records. Second, since the representations of users' vocal competence will be utilized to predict overall ratings of karaokes, it is rather difficult to find the optimal representations that can help enhance the prediction of overall ratings. Finally, due to the evaluation bias of karaoke machine and the sparse singing records of a user for a song, a user's vocal competence learned from historical overall ratings might be over-fitted and cannot fully capture his/her inborn vocal competence in real world. The modeling method thereby needs to be robust enough to overcome the data bias.

Indeed, in the decision process of choosing karaoke songs, users not only take the content and style of songs into account, 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 multi-aspect vocal ratings (e.g., ratings of pitch, volume, and rhythm) by analyzing users' karaoke singing recordings. Specifically, after preprocessing the karaoke records, we obtain the audio records which encode users' vocal performance. Then, given such records, we extract ratings of pitch, volume and rhythm. Later, we exploit a non-negative tensor factorization method to model the generative process of vocal ratings as *user* $\times$ *song* $\times$ *audio*. Therefore, we can factorize the extracted multi-aspect vocal ratings and learn the latent features of users' vocal competence.

To tackle the first challenge, a tensor factorization is employed to examine the multi-aspect vocal ratings and estimate the latent features of vocal competence of users. A straight idea is to feed the factorized latent features to a classifier and use such classifier to predict the overall rat-

*University of Science and Technology of China, China, guanchu@mail.ustc.edu.cn, {cheneh,ylliu22}@ustc.edu.cn. (Corresponding Author: Enhong Chen)

†Rutgers University, USA, {yanjie.fu,hxiong}@rutgers.edu

‡Northwestern Polytechnical University, Xi'an, China, xjlu@mail.nwpu.edu.cn

Copyright © by SIAM
Unauthorized reproduction of this article is prohibited.

ings. However, to help tensor factorization to learn the most discriminative latent factors, it is not effective to model the decomposition and classification independently [3]. Consequently, we present a joint modeling method to exploit the mutual enhancement between NTF and SVM. We attempt to find a non-negative decomposition for the multi-aspect vocal ratings as well as learn a classifier in the factorized space. Therefore, the decomposition in our procedure has potential to enhance the classification performance. Furthermore, to overcome the bias and sparsity of karaoke overall ratings, we plug-in an additive term into the representation of users' vocal competence, so that we allow the competence representation to vary during the learning process.

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. Along this line, we propose a joint modeling method to incorporate vocal competence by exploiting the mutual enhancement between NTF and SVM. Specifically, 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 NTF method to model the generative process of vocal ratings as *user* $\times$ *song* $\times$ *audio*. Moreover, we feed the factorized latent features of user competence, song requirements, and vocal measurements into an SVM classifier and use this classifier to classify the overall ratings. In addition, we propose an enhanced objective function by jointly modeling both multi-aspect rating factorization and overall rating prediction and solve this objective as a coupled least-squares optimization problem via a maximum margin 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 Competence Based Nonnegative Tensor Factorization, named CBNTF.

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) then to incorporate the degree of matching users' vocal competence to songs' vocal requirements for karaoke recommendation.

Table 1: Mathematical Notations.

Symbol	Description
$\mathcal{X}$	tensorial ratings data ($\in \mathbb{R}^{I_1 \times I_2 \times I_3}$)
x_{ijk}	rating of k-th vocal feature in j-th song sung by i-th user
y	label of a singing record
N	dimension of $\mathcal{X}$
I_n	length of the n-th mode of $\mathcal{X}$
R	rank of $\mathcal{X}$
$\mathbf{U}^{(n)}$	n-th latent factor matrix ($\in \mathbb{R}^{I_n \times R}$)
$\mathbf{u}_i^{(s)}$	i-th principal component of $\mathbf{U}^{(s)}$
$\Delta\mathbf{u}_i$	potential singing competence of i-th user
δ	the constraint of $\Delta\mathbf{u}$
λ	regularization parameter
$\|\cdot\|_F$	Frobenius norm
$Seq(\cdot)$	sequence of MIDI notes

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 three-dimensional information: (1) user, (2) song, and (3) audio signal. Therefore, we propose to model such user-song-audio relations using a three-dimensional tensor, with each element representing a single-aspect vocal rating of a user for a song. In particular, we denote the tensor as $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$, where I_1 is the number of users, I_2 is the number of songs, and I_3 is the number of audio features. Then, x_{ijk} in $\mathcal{X}$ denotes the rating of the vocal feature k in the song j sung by the user i, for example, the rhythm rating of user #1 for song #2 is 88.

Mathematically, we use the CP decomposition of $\mathcal{X}$ [4], which is formulated as:

(2.1)
$$\mathcal{X} \approx \sum_{r=1}^{R} \mathbf{u}_r^{(1)} \circ \mathbf{u}_r^{(2)} \circ, \ldots, \circ \mathbf{u}_r^{(N)} = [\mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \ldots, \mathbf{U}^{(N)}].$$

Here, we denote I_n as the length of n-mode and R as the rank of tensor $\mathcal{X}$, such that $\mathbf{U}^{(n)} = [\mathbf{u}_1^{(n)}, \ldots, \mathbf{u}_R^{(n)}] \in \mathbb{R}^{I_n \times R}$. Table 1 lists the notations used in this paper.

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

2.3 The Overview of Our Model

Figure 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

Copyright © by SIAM
Unauthorized reproduction of this article is prohibited.

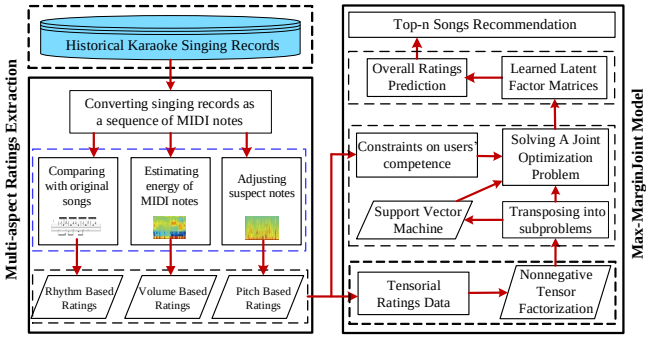


Figure 1: The Framework of Competence Based Nonnegative Tensor Factorization.

vector, thus, we adopt a pre-defined threshold to determine such songs.

Learning Users' Vocal Competence: We treat the ratings of pitch, volume and rhythm of users for songs as a rating tensor. Then, we propose a maximum-Margin joint model to factorize multi-aspect vocal ratings, and learn the latent representations (i.e., latent factors) of users' competence simultaneously.

Exploiting the Matching Degree between Users and Songs for Karaoke Recommendation: We assume that all the users has their own potential singing competence, i.e., $\mathbf{u_i'} = \mathbf{u_i} + \triangle\mathbf{u_i}$, where $\mathbf{u_i}$ is i-th user's singing competence extracted from the recordings and $\triangle\mathbf{u_i}$ is the potential singing competence. In our formulation, $\triangle\mathbf{u}$ is imposed with a constraint, i.e., $\triangle\mathbf{u} \leq \delta$, where δ is a predefined threshold to bound users' potential vocal competence. Then, we jointly combine the tasks of tensor decomposition and SVM classification as a 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 three learned latent factor matrices. The top-n songs with highest ratings are recommended.

3 Vocal Competence Based Karaoke Recommendation

In this section, we introduce the vocal competence based karaoke recommendation method.

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 and most melodies are not on the same tune with the ground-truth music scores. 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 [2] 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.,

$$R^{pitch} = \tilde{Seq}(c). \tag{3.2}$$

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.(3.2), 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:

$$R^{volume} = \mathcal{I} \times \exp\left[\mathrm{sim}\left(\tilde{Seq}(c), \tilde{Seq}(c')\right)\right], \tag{3.3}$$

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) [5] 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

$$R^{rhythm} = \mathcal{I} \times \exp\left[\mathrm{Sim}_{DTW}\left(\tilde{Seq}(c), \tilde{Seq}(c')\right)\right], \tag{3.4}$$

where $\mathcal{I}$ is configured with the same setting adopted in Eq.(3.3), i.e. $\mathcal{I} = 100$.

To this end, for song j sung by user i, we extract the three aspect ratings and aggregate them into a vector, i.e., $\mathbf{x}_{ij:} = \{R_{ij}^{pitch}, R_{ij}^{volume}, R_{ij}^{rhythm}\}$. After extracting users' vocal ratings, we aggregate them as a three-dimensional tensor $\mathcal{X}$.

Copyright © by SIAM
Unauthorized reproduction of this article is prohibited.

3.2 The Maximum-Margin Joint Model

We introduce the proposed maximum-margin joint model which combines the modelings of multi-aspect vocal ratings and overall ratings 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 tensor while effectively discriminate the karaoke overall ratings via the max-margin learning process.

The Modeling of Tensor. Given a tensor of multi-aspect ratings $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$, the nonnegative factorization of $\mathcal{X}$ in terms of the CP decomposition is as follows:

$$\mathcal{X} \approx \sum_{r=1}^{R} \mathbf{u}_r^{(1)} \circ \mathbf{u}_r^{(2)} \circ \mathbf{u}_r^{(3)} \tag{3.5}$$

Note that the audio-level features contain the information of users' singing competence which can be extracted from their recordings. The audio-level features learned in Eq.(3.5) are denoted as the s-mode $\mathbf{U}^{(s)}$. For the i-th user, the corresponding ratings of $\mathbf{U}^{(s)}$ is $\mathbf{u}_\mathbf{i}^{(\mathbf{s})}$. Then we can transfer the classification from $\{\mathcal{X}, \mathbf{y}\}$ into $\{\mathbf{u}_\mathbf{i}^{(\mathbf{s})}, y_i\}$ $(1 \le i \le I_s)$. The optimal factorization of non-negative tensorial data is reformulated as a coupling least-squares optimization problem and then we can update one column vector at a time. In particular, the minimization problem in the s-th mode is:

$$\begin{aligned} &\min_{\mathbf{u}_\mathbf{i}^{(\mathbf{s})}} : \|\mathbf{x}_\mathbf{i}^{(\mathbf{s})} - \mathbf{u}_\mathbf{i}^{(\mathbf{s})}\|^2 + \Omega(\mathbf{x}_\mathbf{i}^{(\mathbf{s})}) \\ &s.t. \quad \mathbf{u}_\mathbf{i}^{(\mathbf{s})} \ge 0, 1 \le i \le I_s, \end{aligned} \tag{3.6}$$

where $\Omega(\cdot)$ is a regularized penalty.

Then, we use tensor decomposition to capture the underlying patterns in the user-song-audio tensor. Combining the vectors of the rank-one components, we get three latent factor matrices representing pair relations $\{\text{user}, \text{song}\}$, $\{\text{user}, \text{audio}\}$ and $\{\text{song}, \text{audio}\}$.

The Modeling of SVM. To achieve greater generalized discriminating power, our optimization problem also intends to reduce the misclassification error and maximize the margin of the classifier in the feature space. With the discriminative labels of each recording $\{y_i\}_{i=1}^{I_s}$ and the hyperplane parameter $\mathbf{w}$, the acquired latent factor matrices can be fed into an SVM loss function, such as:

$$L(\mathbf{u}_i^{(s)}, y_i, \mathbf{w}) = \max\{0, 1 - y_i \mathbf{w}^\top \mathbf{u}_\mathbf{i}^{(\mathbf{s})}\}. \tag{3.7}$$

Furthermore, in karaoke song recommendation applications, although some features rely on the content information of songs, however, if a user's singing recordings are sparse, we can not fully predict a user's inherent singing competence, because his/her singing ratings may suffer from overfitting problem. Moreover, it is still possible for a user to choose difficult songs if he/she likes challenges. Hence, besides the observed ratings, our proposed method takes users' potential singing competence into consideration. Formally, the potential singing competence of the i-th user is denoted as $\Delta\mathbf{u}_\mathbf{i}^{(\mathbf{s})}$ and is formulated as an additive parameter subjected to $\mathbf{u}_\mathbf{i}^{(\mathbf{s})}$, i.e.,

$$\mathbf{u}_\mathbf{i}^{(\mathbf{s})} \leftarrow \mathbf{u}_\mathbf{i}^{(\mathbf{s})} + \Delta\mathbf{u}_\mathbf{i}^{(\mathbf{s})}$$

To provide prior information of $\Delta\mathbf{u}_\mathbf{i}^{(\mathbf{s})}$, there is a constraint imposed on users' potential competence in our objective function, such that $\|\Delta\mathbf{u}_\mathbf{i}^{(\mathbf{s})}\| \le \delta_i$. The bound δ_i has a similar effect of the standard deviation in the Gaussian noise model [6]. Another reason to use this constraint to bound $\Delta\mathbf{u}_\mathbf{i}^{(\mathbf{s})}$ is that there is an intuitive geometric interpretation in the resulting formulation, as shown in Section 6.

The Maximum Margin Joint Model. As stated before, the goal of our proposed approach is to find a nonnegative decomposition for a tensorial data as well as learn a classifier in the factorized space. With users' potential competence considered, we have the following problem:

$$\begin{aligned} \min_{\mathbf{u}_i} :& \gamma\|\mathbf{x}_i^{(s)} - \mathbf{u}_i^{(s)}\|^2 \\ &+ \mathbf{w}^\top\mathbf{w} + \rho\sum_{i=1}^{I_s} L(y_i, \mathbf{w}\cdot(\mathbf{u}_i^{(s)} + \Delta\mathbf{u}_\mathbf{i}^{(\mathbf{s})}) + b) \\ s.t. \quad & \mathbf{u}_i \ge 0,\ \Delta\mathbf{u}_\mathbf{i}^{(\mathbf{s})} > \delta,\ 1 \le i \le I_s, \end{aligned} \tag{3.8}$$

where γ and ρ are parameters to control the approximate error and classification loss respectively, and b is the bias term. Eq.(3.8) results in a set of latent factors that simultaneously reduce the reconstruction error while ensuring a low misclassification error.

Note that the traditional solution for SVM classifiers is generally obtained in the dual domain [7]. However, since the weight vector $\mathbf{w}$ and the components $\mathbf{u}_\mathbf{i}^{(\mathbf{s})}$ are inherently coupled in Eq.(3.8), it is complicated to obtain the dual formulation. Inspired by the idea of primal optimizations of non-linear SVMs [8], we adopt the well-known kernel trick here to capture the non-linear structures implicitly. Therefore, the weight vector $\mathbf{w}$ can be replaced with a functional form

$$f(\mathbf{u}) = \sum_{i=1}^{I_s} \alpha_i \mathbf{k}(\mathbf{u_i}, \mathbf{u}), \tag{3.9}$$

where $\mathbf{k}(\cdot,\cdot)$ is a kernel as given by Mercer's theorem [9]. After replacing $\mathbf{w}$ by $f(\mathbf{u})$, Eq.(3.8) is revised as follows:

$$\begin{aligned} \min_{\mathbf{u}_i^{(s)}} : \quad & \gamma\|\mathbf{x}_i^{(s)} - \mathbf{u}_i^{(s)}\|^2 + \sum_{i=1}^{I_s} L(y_i, \sum_{j=1}^{I_s} \mathbf{k}(\mathbf{u}_\mathbf{i}^{(\mathbf{s})}, \mathbf{u}_\mathbf{i}^{(\mathbf{s})})\alpha_i) \\ &+ \lambda \sum_{i,j=1}^{I_s} \alpha_i\alpha_j \mathbf{k}(\mathbf{u}_\mathbf{i}^{(\mathbf{s})} + \mathbf{\Delta u}_\mathbf{i}^{(\mathbf{s})}, \mathbf{u}_\mathbf{j}^{(\mathbf{s})} + \mathbf{\Delta u}_\mathbf{j}^{(\mathbf{s})}) \\ s.t. \quad & \mathbf{u}_i^{(s)} \ge 0,\ \Delta\mathbf{u}_\mathbf{i}^{(\mathbf{s})} > \delta,\ 1 \le i \le I_s, \end{aligned} \tag{3.10}$$

which is the objective function of our proposed jointly maximum margin model. Here, $\lambda = 1/\rho$ and γ is the relative weight between the loss function and the margin.

Copyright © by SIAM
Unauthorized reproduction of this article is prohibited.

Algorithm 1 The Learning Process of CBNTF

Input: The tensorial training data and their corresponding class labels, i.e., $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \cdots \times I_N}$, where $y_i \in \{-1, +1\}$, $i = 1, 2, \cdots, I_n$, given a kernel k

Output: The principle decomposition components of each mode $\{\mathbf{U}_1, \cdots, \mathbf{U}_N\}$ and classifier coefficients vector α

```
repeat
    for n = 1 to N do
        compute ∇_α on Eq.(4.13)
        update H_α on Eq.(4.14)
        update α on Eq.(4.15)
        for i = 1 to I_n do
            update u_i^(s) on Eq.(4.17)
            update H_{u_i}^(s) on Eq.(4.18)
        end for
    end for
until max iteration or convergence
```

4 Solving the Joint Optimization Problem

Many well-known methods can be adopted to solve the optimization problem formulated in Eq.(3.10), such as Newton's method and Gradient Descent. Although Newton's method enjoys a faster convergence rate than gradient descent, calculating the Hessian inverse may be expensive if the size of Hessian is large. Furthermore, since Hessian is not invertible for any kernel, we adopt conjugate gradient. Without computing the invert of the Hessian, we can achieve a reasonable solution with only a couple of steps.

Update α. The first-order gradient of Eq.(3.10) with respect to α is

$$\nabla_\alpha = 2\lambda \mathbf{K}\alpha + \sum_{i=1}^{I_r} \mathbf{k}_i \frac{\partial L}{\partial t}|_{t=\mathbf{k_i^T}\alpha}, \tag{4.11}$$

where $\frac{\partial L}{\partial t}$ is the partial derivative of $L(y,t)$ with respect to its second argument. Moreover, it implies that the optimal function can be formulated as a linear combination of kernel functions evaluated at training samples. Notice that L can be any loss function, such as Hinge loss and ϵ-insensitive loss. Here, we consider a quadratic loss, i.e., the L_2 penalization of the training errors

$$L(y_i, f(\mathbf{u_i})) = \max(0, 1 - y_i f(\mathbf{u_i}))^2. \tag{4.12}$$

For a given value of vector α, a point $\mathbf{u_i}$ is a support vector if the loss on this point is non-zero, i.e., $y_i f(\mathbf{u}_i)$. We can reorder the training points such that the first n entries are support vectors. Then, let $\mathbf{I^o}$ be the $n \times n$ diagonal matrix with first n entries being 1 and others 0. The gradient with respect to α is

$$\nabla_\alpha = 2(\lambda \mathbf{K}\alpha + \mathbf{K}\mathbf{I^o}(\mathbf{K}\alpha - \mathbf{Y})), \tag{4.13}$$

and the Hessian is,

$$H_\alpha = 2(\lambda \mathbf{K} + \mathbf{K}\mathbf{I^o}\mathbf{K}). \tag{4.14}$$

Each step consists of the following update:

$$\alpha \leftarrow \alpha - \gamma H_\alpha^{-1} \nabla_\alpha, \tag{4.15}$$

where γ is the step size for line search or backtracking. In our experiment, we use the default value of $\gamma = 1$.

Update $\mathbf{u}_i^{(s)}$**.** Let $u_{i_n}^{(s)}$ be the n-th element of $\mathbf{u}_i^{(s)}$. While all the other factor matrices are fixed, the problem of Eq.(3.10) becomes a quadratic problem and analytically solvable. Therefore, we can learn $\mathbf{u}_i^{(s)}$ by updating $u_{i_n}^{(s)}$ one by one. For clarity, the update rule of $u_{i_n}^{(s)}$ is as follows

$$\begin{aligned} \mathbf{u}_{i_n}^{(s)} &= \sum (x_{i_1,\ldots,i_N} \prod_{l \neq n} u_{i_l k}^{(l)}), \forall k \\ &= (\mathbf{B}_{i_n}^{(s)} + \lambda \mathbf{I}_K)^{-1} c_{i_n}^{(s)}. \end{aligned} \tag{4.16}$$

where the (k_1, k_2) entry of $B_{i_n}^{(s)} \in \mathbb{R}^{K \times K}$ is $\sum_{u_{i_n}^{(s)}} (\prod_{l \neq n} \mathbf{u}_{i_l k_1}^{(l)} \prod_{l \neq n} \mathbf{u}_{i_l k_2}^{(l)})$, and the k-th entry of $c_{i_n}^{(n)} \in \mathbb{R}^K$ is $\sum_{u_{i_n}^{(s)}} (x_{i_1,\ldots,i_N} \prod_{l \neq n} \mathbf{u}_{i_l k}^{(l)})$.

Then, the gradient of Eq.(3.10) with respect to $\mathbf{u_i}^{(r)}$ is:

$$\begin{aligned} \nabla_{u_i} = &-2\gamma \mathbf{x}_i^{(s)} + 2\gamma \mathbf{u}_i^{(s)} + 2\lambda \alpha_i \sum_{j=1}^{I_s} \alpha_j \mathbf{u}_j^{(s)} + \\ &2(\sum_{j=1}^{n_\nu} l_j \alpha_j \mathbf{u}_j^{(s)})[i \in n_\nu] + \alpha_i \sum_{j=1}^{n_\nu} l_j \mathbf{u}_j^{(s)}). \end{aligned} \tag{4.17}$$

The Hessian with respect to $\mathbf{u_i}^{(r)}$ is:

$$\mathbf{H}_{\mathbf{u_i}^{(r)}} = 2\gamma + (2\lambda \alpha_i^2 + 4 l_i \alpha_i [i \in n_\nu]) \mathbf{I}_{ns}, \tag{4.18}$$

where $[i \in n_\nu]$ is an indicator function and $\mathbf{I}_{ns}$ is an identity matrix of sized I_s. To avoid calculating the inverse, we adopt Cholesky decomposition which also takes $O(K^3)$. Thus, in each iteration, updating each row of all the factor matrices only takes $O(|\Omega| NK(N+K) + K^3 \sum_{n=1}^{N} I_n)$.

Solution of $\mathbf{\Delta u_i^{(s)}}$. Our goal is to construct separating hyperplanes in the feature space using individuals' singing ratings and the mapped potential competence. However, the mapped competence region may correspond to an irregular shape in the feature space, which brings difficulties to our optimization problem. Thus, we propose an approximation strategy for updating $\Delta \mathbf{u}_i$ based on the first-order Taylor expansion of k, which is $k(\mathbf{u}_i + \Delta \mathbf{u}_i, \cdot) = k(\mathbf{u}_i, \cdot) + \Delta \mathbf{u}_i^T k'(\mathbf{u}_i, \cdot)$ where $k'(\mathbf{u}_i, \cdot)$ denotes the gradient of k with respect to $\mathbf{u}_i$. By fixing $\mathbf{\Delta u_i^{(s)}}$ to $\mathbf{\Delta \bar{u}_i^{(s)}}$, the problem of Eq.(3.10) can be converted to a simple second-order cone program (SOCP) which yields a solution $w = \sum_i y_i \bar{\alpha}_i$. An optimal solution of $\mathbf{\Delta u_i^{(s)}}$ is thus acquired, i.e.,

$$\mathbf{\Delta u_i^{(s)}} = y_i \delta_i \frac{\mathbf{v_i}}{\|\mathbf{v_i}\|} \tag{4.19}$$

where $\mathbf{v_i} = \sum y_j \bar{\alpha}_j k'(\mathbf{u_i}, \mathbf{u_j} + \mathbf{\Delta \bar{u}_j})$. The details of the proposed maximum-margin joint model are summarized in Algorithm 1.

5 Top-n Song Recommendation

In the recommendation stage, we use the learned latent factor matrices to predict overall ratings of songs and recommend top-n songs. Formally, given a set of I test

Copyright © by SIAM
Unauthorized reproduction of this article is prohibited.

users denoted as a 3-order tensor $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$, we first solve the maximum-margin joint model which yields three latent feature matrices $\mathbf{U}^{(1)}$, $\mathbf{U}^{(2)}$ and $\mathbf{U}^{(3)}$. Each row $\mathbf{u}_i^{(1)}$, $\mathbf{u}_j^{(2)}$ and $\mathbf{u}_k^{(3)}$ of these factor matrices correspond to the latent factors associated with each particular user, song, and audio. Then the features in $\mathbf{U}^{(1)}$ are taken as input for the SVM classifier and predict the labels $\mathbf{y}_i^{(1)}$. Here, $\mathbf{y}_i^{(1)}$ is a binary vector where $\mathbf{y}_{ij}^{(1)} = 1$ means that the i-th user has singing competence to handle the j-th song. Thus, for each user, we can pick out songs labeled $+1$ for song recommendation. Further more, a tensor $\hat{\mathcal{X}}$ can be computed by multiplying three latent factor matrices $\mathbf{U}^{(1)}$, $\mathbf{U}^{(2)}$ and $\mathbf{U}^{(3)}$ together via tensor product

$$\hat{\mathcal{X}} = \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} \times_3 \mathbf{U}^{(3)} \tag{5.20}$$

where $\times_n$ is the tensor product to multiply a matrix on the n-th dimension with a tensor. The element $\hat{x}_{ijk}$ denotes the predicted rating of the vocal feature k in song j sung by the user i. The predicted overall rating R_{ij} of song j for user i is $R_{ij} = \sum_{k=1}^{K} \hat{x}_{ijk}$, Therefore, we can recommend top-n songs with the highest overall ratings and labeled as $+1$.

6 Experimental Results

We provide an empirical evaluation of the performances of the proposed method on real-world karaoke data.

6.1 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 Figure 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 2 presents the statistics of the data used in the experiments.

Table 2: Statistics of the data set.

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

Evaluation Metrics. We use the following metrics to evaluate the performances of our karaoke song recommendation algorithm.

- **MAE.** Mean absolute error takes the mean of the absolute difference between each prediction and ratings for users in the test set.

$$\text{MAE} = \frac{1}{|\text{T}|} \sum_{i,j,k} |\mathbf{y}_{i,j,k} - \hat{\mathbf{y}}_{i,j,k}|,$$

where $\mathbf{y}_{i,j,k}$ denotes actual singing rating of vocal-feature value k in karaoke song j sung by user i, $\hat{\mathbf{y}}_{i,j,k}$ represents the predicted vocal-feature value and $|\text{T}|$ is the number of predicted values. The MAE is the average absolute deviation of predictions to the ground truth data. The smaller MAE indicates the better prediction accuracy.

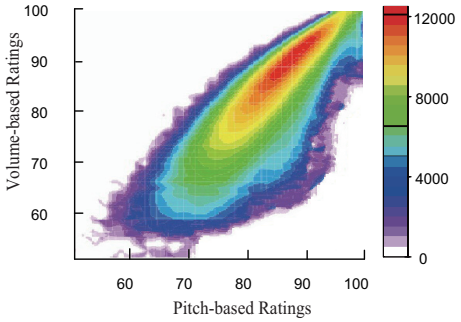


Figure 2: The distribution of singing recordings w.r.t average of pitch-based ratings and volume-based ratings.

- **MacroF1.** The macroF1 metric calculates the average of F_1 scores of all the labels.

$$\text{MacroF1}(h, \mathcal{D}) = \frac{1}{n} \sum_{i=1}^{n} \frac{2 \times \|h(\mathbf{x}_i) \cap \mathbf{Y}_i\|_1}{\|h(\mathbf{x}_i)\|_1 + \|\mathbf{Y}_i\|_1}.$$

- **MicroF1.** The microF1 is the calculation of F_1 regardless of classes. The F_1-score can take both the precision and the recall into account, thus can be viewed as a harmonic mean of precision and recall.

$$\text{MicroF1}(h, \mathcal{D}) = \frac{2 \times \sum_{i=1}^{n} \|h(\mathbf{x}_i) \cap \mathbf{Y}_i\|_1}{\sum_{i=1}^{n} \|h(\mathbf{x}_i)\|_1 + \sum_{i=1}^{n} \|\mathbf{Y}_i\|_1}.$$

Here, the function $h(\mathbf{x}_i)$ counts the "hits" of the prediction model for a song $\mathbf{x}_i$ where a rating from the ground truth is among the ratings predicted.

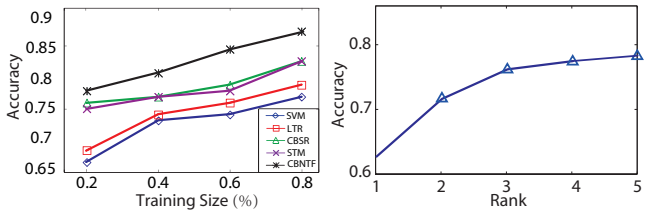
- **AUC.** The area under the ROC curve, is also used to capture the overall recommendation performance [10]. The range of AUC is the interval $[0, 1]$ and the AUC of a random classifier is 0.5.

Baseline algorithms. We compared our proposed algorithm with the following baseline methods, in which **CBSR** is a state-of-the-art algorithm for song recommendation:

- **Support Vector Machine (SVM)**: Support Vector Machine, a typical maximum-margin classifier, is applied after vectorizing the input tensor on each class using the one vs. all scheme. We use the open source software LIBSVM [11].
- **Support Tucker Machine (STM)**: Support Tucker Machine [12] is a tensor-based model of SVM.
- **Logistic Tensor Regression (LTR)**: Logistic Tensor Regression is a modified tensor-based logistic regression method for classification [13].
- **Competence Based Song Recommendation (CBSR)**: This is a learning-to-rank scheme for recommending songs, which takes singers' vocal competence into consideration [14].

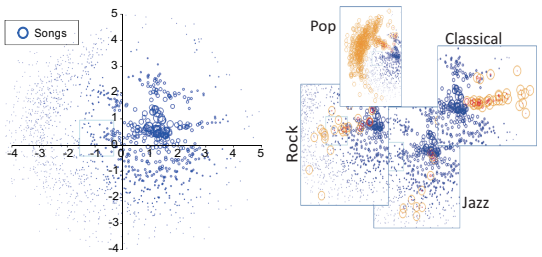
In the recommendation stage, we set a threshold rating σ and recommended songs whose predicted rating are more than σ. Note that the karaoke site has scored the users' singing performance. We randomly split the processed data into training data (70%) and test data (30%) and collected the results. We repeated this procedure for 10 times and reported the average performances.

Copyright © by SIAM
Unauthorized reproduction of this article is prohibited.



(a) The Size of Training data (b) The Rank of $\mathcal{X}$

Figure 3: Effectiveness of Model Parameters



(a) Latent Factor $\mathbf{U}^{(3)}$ (b) Four Genres of Songs

Figure 4: Each song is present 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.

6.2 Performance Comparison

First of all, to evaluate the influence of training dataset size over the classification accuracy, we conduct experiment using different configurations of training dataset size, as shown in Figure 3(a). We observe that the performance by all competing algorithms improves as the number of training samples increases. Our method achieves the best classification performance in most cases, because our method is able to learn more discriminative features when more training data are provided.

In addition, we investigated the influence of the tensor rank R on the classification performance, as shown in Figure 3(b). With different configurations of the tensor rank, we note that the classification performance become stable when the tensor rank is 5.

Furthermore, we evaluate the performance in terms of three kinds of metrics. From Table 3 we can make the following observations:

- In principle, our method is a discriminative-generative model that is formulated as a joint optimization framework of tensor factorization and support vector machine. Thus, the features learned by our method not only preserve the intrinsic multi-view structural information on tensorial audio signal data, but also include the discriminative information derived from the max-margin learning process.
- The classification performance of tensor-based approaches, such as STM and LTR, are superior to those vectorized-based approaches, such as SVM. STM makes full use of the data structural information and reduces the number of decision parameters of classification significantly. This is mainly because tensor-based feature representations can effectively preserve the structural information on the original data.

6.3 Effectiveness of Latent Factor Matrices

In this subsection, we show the interpretable nature of our proposed method for discriminative analysis in karaoke song recommendation. Let $\mathbf{U}^{(1)}$, $\mathbf{U}^{(2)}$, $\mathbf{U}^{(3)}$ denote three latent factors learned from the optimization problem, i.e., $\mathbf{U}^{(1)}$ captures user-song relations, $\mathbf{U}^{(2)}$ captures user-audio relations, and $\mathbf{U}^{(3)}$ captures audio-song relations, and we present some interpretation of $\mathbf{U}^{(3)}$. Note that the other two latent factors $\mathbf{U}^{(1)}$ and $\mathbf{U}^{(2)}$ are rather sparse because most users only sing a small set of songs.

Visualization of latent factor $\mathbf{U}^{(3)}$. Each row in the nonnegative matrix $\mathbf{U}^{(3)}$ can be treated as a song. For example, $\mathbf{U}^{(3)}(j,:)$ can be viewed as a latent feature of j-th song. To illustrate $\mathbf{U}^{(3)}$ in a 2-D figure, we adopt t-SNE[1], which is commonly used for the visualization of high-dimensional data [15], to assist our data analysis. Then $\mathbf{U}^{(3)}$ is illustrated in Figure 4(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 Figure 4(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).

6.4 Effectiveness of Users' Singing Competence

Users' potential singing competence $\Delta\mathbf{u}$ is bounded by δ and imposed as a constraint in our formulation. In this subsection, we visualize the classification performance over two different δ to get a more straightforward observation. Specially, we randomly pick a song and collect the corresponding karaoke records. As shown in Figure 5, purple dots and blue dots represent users performing good and bad respectively. The users' potential competence are represented by circles with their sizes proportional to individuals' potential competence of the corresponding user. We can find that the resulting formulation has an intuitive by adopting the bounded potential competence. Furthermore, by modeling $\Delta\mathbf{u}$ on singing competence, our method becomes more sensitive to users' multi-aspect ratings rather than the overall ratings. This is because we can find a choice of $\Delta\mathbf{u}$, such that $\mathbf{u}+\Delta\mathbf{u}$ is far from the decision boundary and will not be a support vector. Therefore, comparing to overall ratings, the multi-aspect ratings can reveal the more trust-worthy singing competence.

6.5 Convergence Issues

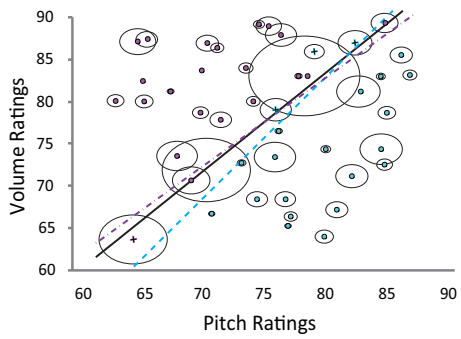
In this subsection, we discuss the convergence of the proposed CBNTF by evaluating the variation of root-mean-square error (RMSE) from the point of tensor reconstruction during the iterations. Here, the objective function proposed in Eq.(3.10) is a weighted combination of the NTF cost and the classifier (SVM) cost. We optimized this objective function using conjugate gradient and solved a set of convex

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

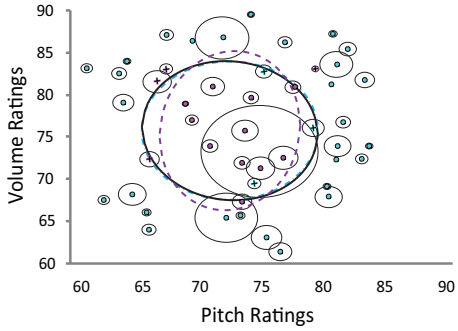
Copyright © by SIAM
Unauthorized reproduction of this article is prohibited.

Table 3: Recommendation Performance Comparison.

Method	MAE	AUC	MacroF1	MicroF1
SVM	0.2427 ± 0.0263	0.7852 ± 0.0226	0.7091 ± 0.0209	0.7198 ± 0.0121
CBSR	0.2091 ± 0.0398	0.7016 ± 0.0190	0.6481 ± 0.0255	0.6308 ± 0.0318
LTR	0.2083 ± 0.0213	0.7193 ± 0.0189	0.6681 ± 0.0263	0.6890 ± 0.0117
STM	0.2658 ± 0.0289	0.7562 ± 0.0234	0.7031 ± 0.0415	0.7151 ± 0.0325
CBNTF	$\mathbf{0.1831 \pm 0.0013}$	$\mathbf{0.7864 \pm 0.0272}$	$\mathbf{0.7173 \pm 0.0562}$	$\mathbf{0.7834 \pm 0.0423}$



(a) $\delta = 10$



(b) $\delta = 30$

Figure 5: Classification performance over constraint δ of users' potential singing competence.

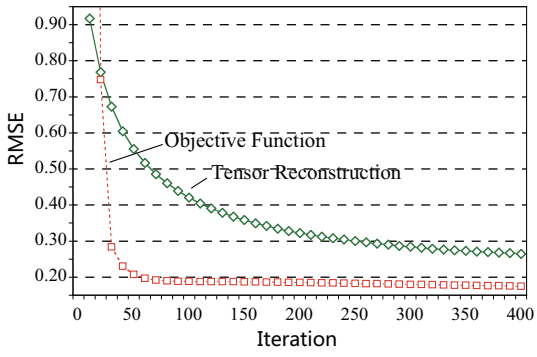


Figure 6: Convergence comparison of the objective function and tensor reconstruction. The x-axis shows the number of iterations, and the y-axis shows the RMSE on validation data (lower is better).

sub-problems in each step. The tensor reconstruction error, i.e., $\|\mathcal{X} - \sum_{r=1}^{R} \mathbf{u}_r^{(1)} \circ \mathbf{u}_r^{(2)} \circ \mathbf{u}_r^{(3)}\|_2$, is the the NTF cost which means the discrepancy between the approximation obtained by the proposed algorithm and the original data.

In Figure 6, we can see that tensor reconstruction could converge after 400 iteration and achieve a RMSE of 0.2683 on the validation set. In contrast, the objective function converges after 100 iterations and achieves a better RMSE. This implies that there is mutual enhancement between decomposition and classification in our procedure.

7 Related Work

In the literature, a lot of methods have been proposed to address song recommendations, such as [1, 16]. Traditional song recommendation systems are proposed for discovering songs which satisfy users' listening interest. [17] 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 [18], but the circumstance of the user typically is not considered when the recommendations take place.

On the other hand, matrix factorization methods are also applied to perform these prediction, such as [19]. Matrix factorization has become a popular CF technique. However, the similarity compared to other users will be poor for the users whose tastes are unusual to the population. As a generalization of matrix factorization, tensor factorization has been studied from an algebraic perspective and witnessed a renewed interest. Recently, tensor factorization methods have been used in various applications such as social network analysis and recommendation [20]. A supervised tensor factorization method via max margin has appeared recently [21]. However, a major problem with tensor factorization is that the prediction accuracy is typically influenced by the sparse observations in real datasets. Generalized coupled tensor factorization [22] and a few other studies [23] try to factorize observed tensors while incorporating side information simultaneously. 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

Copyright © by SIAM
Unauthorized reproduction of this article is prohibited.

hoc, by incorporating prior knowledge about music audio.

Karaoke singing recommendation is a relatively a new area, because users' singing skills 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, [14] 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: the 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.

8 Conclusion

In this paper, we proposed a joint modeling method for karaoke recommendation by mining historical karaoke singing records. Specifically, 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 nonnegative tensor factorization method to factorize vocal ratings as users×songs×audio. Besides, we used an SVM classifier to classify overall ratings and regularized the tensor factorization of vocal ratings by feeding the factorized latent features into the SVM classifier. Furthermore, we devised an effective method to solve the joint objective function, to simultaneously optimize both tensor factorization and SVM, 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.

Acknowledgements

This research was partially supported by grants from the National Science Foundation for Distinguished Young Scholars of China (Grant No. 61325010), the Science and Technology Program for Public Wellbeing (Grant No. 2013GS340302) and the Fundamental Research Funds for the Central Universities of China (Grant No. WK2350000001). Also, it was supported in part by Natural Science Foundation of China (71329201).

References

[1] K. Mao, L. Shou, J. Fan, G. Chen, and M. Kankanhalli, "Competence-based song recommendation: Matching songs to ones singing skill." IEEE, 2013.

[2] W.-H. Tsai and H.-C. Lee, "Automatic evaluation of karaoke singing based on pitch, volume, and rhythm features," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 20, no. 4, pp. 1233–1243, 2012.

[3] B. V. Kumar, I. Kotsia, and I. Patras, "Max-margin non-negative matrix factorization," *Image and Vision Computing*, vol. 30, no. 4, pp. 279–291, 2012.

[4] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM review*, vol. 51, no. 3, 2009.

[5] D. J. Berndt and J. Clifford, "Using dynamic time warping to find patterns in time series." in *KDD workshop*, vol. 10, no. 16. Seattle, WA, 1994, pp. 359–370.

[6] H.-X. Li, J.-L. Yang, G. Zhang, and B. Fan, "Probabilistic support vector machines for classification of noise affected data," *Information Sciences*, vol. 221, 2013.

[7] D. Meyer and F. T. Wien, "Support vector machines," *The Interface to libsvm in package e1071*, 2014.

[8] H. Yu, J. Kim, Y. Kim, S. Hwang, and Y. H. Lee, "An efficient method for learning nonlinear ranking svm functions," *Information Sciences*, vol. 209, 2012.

[9] G.-B. Huang, X. Ding, and H. Zhou, "Optimization method based extreme learning machine for classification," *Neurocomputing*, vol. 74, no. 1, pp. 155–163, 2010.

[10] T. Fawcett, "An introduction to roc analysis," *Pattern recognition letters*, vol. 27, no. 8, pp. 861–874, 2006.

[11] C.-C. Chang and C.-J. Lin, "Libsvm: a library for support vector machines," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, no. 3, 2011.

[12] I. Kotsia and I. Patras, "Support tucker machines," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 633–640.

[13] W. Guo, I. Kotsia, and I. Patras, "Tensor learning for regression," *Image Processing, IEEE Transactions on*, vol. 21, no. 2, pp. 816–827, 2012.

[14] L. Shou, K. Mao, X. Luo, K. Chen, G. Chen, and T. Hu, "Competence-based song recommendation," in *Proceedings of the 36th international ACM SIGIR conference*. ACM, 2013, pp. 423–432.

[15] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, no. 2579-2605, p. 85, 2008.

[16] X. Wu, Q. Liu, E. Chen, L. He, J. Lv, C. Cao, and G. Hu, "Personalized next-song recommendation in online karaokes," in *Proceedings of the 7th ACM Conference on Recommender Systems*. ACM, 2013.

[17] D. Bogdanov, M. Haro, F. Fuhrmann, A. Xambó, E. Gómez, and P. Herrera, "Semantic audio content-based music recommendation and visualization based on user preference examples," *Information Processing & Management*, vol. 49, no. 1, pp. 13–33, 2013.

[18] Y. Fu, H. Xiong, Y. Ge, Z. Yao, Y. Zheng, and 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.

[19] V. Kuleshov, A. T. Chaganty, and P. Liang, "Tensor factorization via matrix factorization," *arXiv preprint arXiv:1501.07320*, 2015.

[20] B. Hidasi and D. Tikk, "Fast als-based tensor factorization for context-aware recommendation from implicit feedback," in *Machine Learning and Knowledge Discovery in Databases*. Springer, 2012, pp. 67–82.

[21] F. Wu, X. Tan, Y. Yang, D. Tao, S. Tang, and Y. Zhuang, "Supervised nonnegative tensor factorization with maximum-margin constraint," 2013.

[22] B. Ermiş, E. Acar, and A. T. Cemgil, "Link prediction in heterogeneous data via generalized coupled tensor factorization," *Data Mining and Knowledge Discovery*, vol. 29, no. 1, pp. 203–236, 2015.

[23] E. Acar, T. G. Kolda, and D. M. Dunlavy, "All-at-once optimization for coupled matrix and tensor factorizations," *arXiv preprint arXiv:1105.3422*, 2011.

Copyright © by SIAM
Unauthorized reproduction of this article is prohibited.