Front. Comput. Sci., 2020, 14(2): 273–290
https://doi.org/10.1007/s11704-018-7072-6

RESEARCH ARTICLE

Leveraging proficiency and preference for online Karaoke recommendation

Ming HE[1], Hao GUO[1], Guangyi LV[1], Le WU[2], Yong GE[3], Enhong CHEN (✉)[1], Haiping MA[4]

1 Anhui Province Key Laboratory of Big Data Analysis and Application, University of Science and Technology of China, Hefei 230027, China

2 School of Computer and Information, Hefei University of Technology, Hefei 230026, China

3 Eller College of Management, The University of Arizona, Arizona 85721-0108, USA

4 iFlyTek Research, Hefei 230026, China

Abstract Recently, many online Karaoke (KTV) platforms have been released, where music lovers sing songs on these platforms. In the meantime, the system automatically evaluates user proficiency according to their singing behavior. Recommending approximate songs to users can initialize singers' participation and improve users' loyalty to these platforms. However, this is not an easy task due to the unique characteristics of these platforms. First, since users may be not achieving high scores evaluated by the system on their favorite songs, how to balance user preferences with user proficiency on singing for song recommendation is still open. Second, the sparsity of the user-song interaction behavior may greatly impact the recommendation task. To solve the above two challenges, in this paper, we propose an information-fused song recommendation model by considering the unique characteristics of the singing data. Specifically, we first devise a pseudo-rating matrix by combing users' singing behavior and the system evaluations, thus users' preferences and proficiency are leveraged. Then we mitigate the data sparsity problem by fusing users' and songs' rich information in the matrix factorization process of the pseudo-rating matrix. Finally, extensive experimental results on a real-world dataset show the effectiveness of our proposed model.

Keywords KTV, matrix factorization, recommendation system

Received February 26, 2017; accepted November 3, 2017

E-mail: cheneh@ustc.edu.cn

1 Introduction

Recent years have witnessed the rapid development of online KTV platforms. Examples of online KTV platforms include Kugou and 51mike. These platforms have attracted a large number of users to sing songs online. For instance, the online KTV platform Changba was set up in 2012 and now has more than 300 million users.

On online KTV platforms, users can organize their KTV parties easily, and select favorite songs to sing. After the singing is finished, the platform will evaluate the synthesis score (e.g., 0 to 100) according to user proficiency and provide details of the evaluation, including evaluations of every sentence's lyrics and other aspects. The details of the evaluation can help users correct their singing errors and improve their proficiency. Thus, the evaluations on these KTV platforms are different from traditional online entertainments (e.g., music listening), where users just give ratings to the items. Particularly, the system's evaluations of user proficiency can affect users' choices on songs for singing. Besides, when users intend to sing songs, they must consider the surrounding environment and other factors. For example, they can not sing songs at work, but they can listen to musics or view news at work. It leads to that the interactions between users and songs are much sparser than traditional online entertainments.

The unique characteristics of these online KTV platforms

bring new challenges to accurately recommend songs to users compared with traditional online entertainments recommendation. Firstly, *as users may not be achieving high scores on their favorite songs*, the first challenge is how to balance user preferences with user proficiency on users' selections on songs. The "user preferences" means the intensity of which a user likes the song and "user proficiency" means the score evaluated by the system. Particularly, we have quantified the inconsistency between user preferences and user proficiency in Section 2. Secondly, due to the fact that the number of songs sung by each user is much less than the number of items consumed by each user on the traditional entertainment websites (e.g., music listening), the rating data constructed from the interactions between users and songs is much sparser than traditional entertainments, which leads to that recommending unrated songs to users just using prior sparse ratings is critically challenging. In a word, the second challenge we need to address is what information can be used to mitigate the data sparsity problem.

To address the challenges mentioned above, in this paper, we focus our study on song recommendation for online KTV users. Along with this line, we propose an information-fused recommendation model based on matrix factorization denoted as InfoFuMF, which can balance user preferences with user proficiency and mitigate the sparseness challenge by fusing the user similarity, the song similarity and a pseudo-rating matrix into a unified framework. Specifically, to balance user preferences with user proficiency, we define two factors to measure user preferences and proficiency, and then construct the pseudo-rating matrix by leveraging the two weighted factors. Furthermore, to address the sparseness challenge in the constructed pseudo-rating matrix, we first employ users' information and songs' information to calculate user similarity and song similarity respectively. Particularly, to precisely depict the similarity between users, we construct an interest network according to users' singing behavior. Based on the interest network, we utilize the supervised random walks (SRW) [1] to learn the user similarity, which can fuse the interest network structure, the characteristics of nodes and edges of the network in a principled way. Then, we fuse the learnt song similarity, the learnt user similarity and the constructed pseudo-rating matrix into the unified matrix factorization model (InfoFuMF) to accurately recommend right songs to right online KTV users. Finally, we evaluate our approach by conducting extensive experiments with a behavioral log of online KTV users, and the experimental results have demonstrated the effectiveness of the proposed method.

Overview The remainder of the paper is organized as follows. In Section 2, we briefly discuss some related works. Section 3 introduces the problem definition and the rating function. Section 4 introduces our proposed InfoFuMF framework, including user similarity calculation, item similarity calculation, and the learning procedure of the InfoFuMF. Next, we report and analyze the experimental results in Section 5. Finally, we conclude this paper in Section 6.

2 Related work

In this section, we summarize the related work into the following three categories:

Content-based methods Content-based methods recommend songs which have similar audio content to the users' preferred songs. Most existing content-based methods first extract traditional features and then recommend songs based on the similarities between the feature vectors of songs. While the similarity metrics used to compute song/user's distance are often ad-hoc, there are two recent works trying to employ machine learning techniques to automatically learn a similarity metric [2, 3], which still depended on traditional features. To solve this shortage, Bogdanov et al. [4] presented recommendation approaches which employ a proposed semantic music similarity measure, and Guan et al. [5] proposed a music recommendation model by exploiting vocal competence which can compute user/song's similarity in terms of pitch, volume and rhythm. Lately, Soleymani et al. [6] proposed a content-based music recommendation system by exploiting five attributes extracted from psychological studies of user preference, which can increase the probability of unpopular but interesting songs to be recommended. Guo [7] proposed an efficient feature generation and selection method by adapting the feature selection for ranking method for music recommendation. Celma [8] presented a dynamic ensemble learning system, which could combine musicological data and machine learning models to provide a truly personalized music recommendation. Song et al. [9] studied how to fully exploit the aspect factors extracted from text to improve cross domain recommendation's performance, which can better capture user preferences. Nevertheless, content information is not easily extracted, especially acoustic information [10].

Collaborative filtering methods Collaborative filtering methods recommend songs by considering the preferences of similar users. There are two main subgroups of collaborative filtering for rating prediction. The first one is memory-based methods, which recommend songs for the target user based

on the consumed songs of similar users [11–13]. The other one is model-based, which makes recommendations using a trained compact model from the user-song rating matrix. Matrix factorization (MF) [14–16] is one of the representative methods, which is first proposed by Koren et al. [14] in dealing with data sparsity. As its effectiveness and efficiency in dealing with large rating matrices, some social-based recommendation approaches adopted the social information based on MF [17–22], which can better exploit the information of users' trust relationship and preferences. For example, Ma et al. [17] proposed the SoRec method which extends the basic MF model by integrating the social network. Recently, Li et al. [21] proposed two alternative models which incorporate the overlapping community into MF to solve the sparsity of user's social connections. Yan et al. [23] proposed a method to fuse item-based and user-based collaborative filtering algorithms into a hybrid model to improve the accuracy of music recommendation. Besides, there are some other investigated training models, such as the aspect models [24,25] and the ranking model [26, 27]. While these traditional approaches only recommend songs to users which meet user preferences, none of these approaches can balance user preferences with user proficiency when recommends songs to online KTV users.

Hybrid methods Hybrid methods combine two or more of the above methods. There are some hybrid methods explored in recommendation systems for other products [28, 29]. We borrow the idea of combining content and user behavior data of [30, 31], and we advance these works by considering the unique characteristics of the KTV data. In the purpose of music recommendation, Yoshii et al. [32] presented a hybrid method that ranks musical pieces while efficiently maintaining CF and content-based data for music recommendation. In addition, Li et al. [33] proposed a probabilistic hybrid music recommender to combine CF and traditional features, which helps alleviate the problem associated with data sparseness by utilizing audio features. Cheng and Tang [34] proposed a hybrid approach to provide personalized music recommendation by extracting audio features of songs and fusing them with user personalities using SVM. Benzi et al. [35] formulated a novel song recommendation model as a matrix completion problem by combining non-negative matrix factorization and total variation on graphs, which is very versatile. However, as the serious sparsity of the rating matrix on online KTV platforms, these methods can not be applied to recommend songs to users directly.

In a summary, although there are numerous music recommendation approaches, most of these works only consider user preferences when recommending songs to users. However, due to the unique characteristics of the online KTV platforms, we should consider not only user preferences but also user proficiency as both of them play an important role on users' selections on songs. What's more, as the severe sparsity of user/song's rating matrix, these methods can not be applied to recommend songs to online KTV users. In a word, as the application scenario is not the same as traditional music recommendations, we propose a new method for online KTV music recommendation in the following sections.

3 Problem definition

We claim that the factors considered by users for choosing a song on a KTV platform mainly contain their preferences and their proficiency on singing. Formally, *user preferences*, measuring the intensity of which a user likes the song, can be reflected by the number of songs' sung just like the traditional online entertainments. *user proficiency*, measuring the capability of which a user sings the song, is evaluated by the system and measured by a score from 0 to 100 and is different from traditional online entertainments where users give ratings to items. Let $U = \{u_1, \ldots, u_M\}$ and $S = \{s_1, \ldots, s_N\}$ be the sets of M users and N songs, respectively. We introduce a new rating function exploiting the two weighted factors:

$$R_{i,j} = \alpha \times \frac{c_{i,j}}{max\_c_i} + (1-\alpha) \times \frac{g_{i,j}}{max\_g_i}, \tag{1}$$

where $i = 1, \ldots, M$, $j = 1, \ldots, N$, $c_{i,j}$ indicates the number of times the song s_j sung by the user u_i, $max\_c_i = \max\{c_{i,1}, \ldots, c_{i,N}\}$, $g_{i,j}$ is the system's evaluation on user proficiency when u_i sings s_j, $max\_g_i = \max\{g_{i,1}, \ldots, g_{i,N}\}$, α can be regarded as trade-off between proficiency and frequency of songs' sung. From the function, we can see that the rating values are determined by the preferences ration and proficiency ratio, which can balance user preferences with user proficiency. With the conjecture that the rating values are governed by user preferences and user proficiency, the prediction on unrated songs can be modeled by combining user preferences and characterization of songs' features. Specifically, when $\alpha = 1$, the rating matrix only utilizes the information of user preferences, which is denoted as $R^{(prefer)}$. Conversely, when $\alpha = 0$, the rating matrix only utilizes the information of proficiency, which is denoted as $R^{(profic)}$. For ease of explanation, we list main notations in Table 1.

Formally, given the pseudo-rating matrix R, user behavior and song features, we wish to learn the user latent vectors X and song latent vectors Y, and then predict the rating matrix

$\widehat{R}$ on unrated songs. Finally, for each user u, we can obtain the predicted rating vector $\widehat{\mathbf{R}}_u$ and select the top-N songs as recommendations for singing.

Table 1 Mathematical notations

Notations	Meaning
U	Collection of users with element u_i
S	Collection of songs with element s_j
$c_{i,j}$	The number of times the song s_j sung by the user u_i
$g_{i,j}$	The evaluation of user proficiency when u_i sings s_j
R	The pseudo-rating matrix with element R_{ij}
$\hat{R}$	The predicted rating matrix with element $\hat{R}_{ij}$
$R^{(prefer)}$	The preferences' rating matrix ($R_{ij}^{(prefer)}$)
$R^{(profic)}$	The proficiency' rating matrix ($R_{ij}^{(profic)}$)
X	The user latent feature vectors
Y	The song latent feature vectors
B	User similarity matrix with element $B(i,m)$
C	Song similarity matrix with element $C(j,n)$
$\mathbf{p}_i$	Random walk score vector of user i with element p_{im}
$\mathbf{q}_i$	The starting vector of user i with element q_{im}
$\mathbf{w}$	The vector of edge features' weights with element w_k
a_{im}	The edge strength of each user pair (u_i, u_m)
D_i	Destination nodes to which u_i creates edges
L_i	No link nodes to which u_i does not create edges
I	Indicator variable with element I_{ij}
$\mathbf{\Psi_{im}}$	The edge features' vector between u_i and u_m
Φ_{jf}	The fth feature's value of the song s_j
$\mathbf{\Phi}$	The matrix of song features with element Φ_{jf}
Q	SRW's stochastic transition probability matrix (Q_{im})

4 Information-fused recommendation model

Based on the constructed pseudo-rating matrix, a natural idea is to adopt matrix factorization (MF) based models to predict ratings on unrated songs of users. We choose matrix factorization as it is simple and shows relatively high accuracy in collaborative-based recommendation [14,21]. Specifically, under the MF framework, we seek to make two specific latent feature vectors: one is the low-dimensional representations of two users as close as possible if they have similar node and interaction attributes, and the other one is the low-dimensional representations of two songs as close as possible w.r.t. their similarity in the space of user behavior. However, due to the data sparsity and auxiliary of the pseudo-rating matrix, it is hard to accurately learn the two latent feature vectors. Thus, we should utilize as much information as possible to mitigate the challenge of data sparsity. Hence, by integrating rich information of users and songs in a principled way, we design two new regularizations to constrain these two latent feature vectors. These two learnt latent feature vectors, which have successfully fused user information, song information and the pseudo-ratings into a uniform framework, provide a basis to devise an effective personalized song recommendation system for online KTV users. The result is our information-fused recommendation model based on Matrix Factorization abbreviated as *InfoFuMF* (Fig. 1).

In the rest of this section, we first introduce how to characterize the similarities between users based on their content and interactive information, and the similarities between songs based on rich features (e.g., the total times sung by users). We then propose our *InfoFuMF* model based on the computed similarities.

4.1 User similarity calculation

As we know, due to the homophily effect among users, users that have similar node information (e.g., age, gender), similar edge information (e.g., interaction information) and network structure (e.g., the common neighborhood) may possibly have similar item consumptions [36,37]. That means, all of this information is useful for measuring user similarity in our problem. In recent years, there are some algorithms for assessing node similarity, e.g., random walks with restarts (RWR). RWR has been proven to be a powerful tool for computing node proximities on graphs [1] from a technical perspective. With the extension of RWR, supervised random

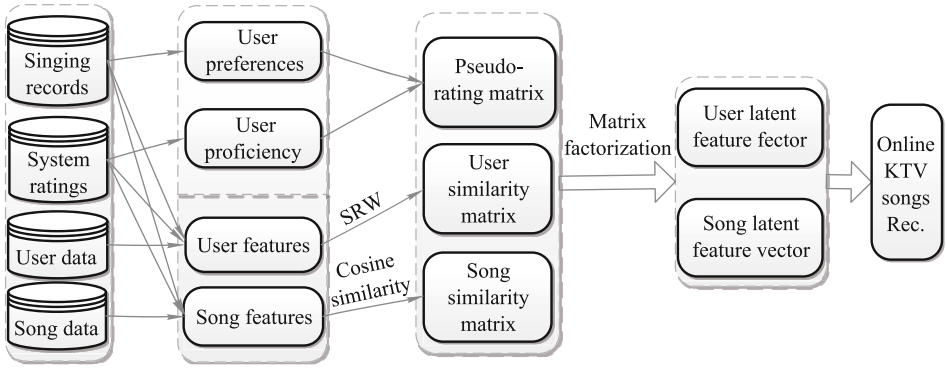


Fig. 1 The framework of InfoFuMF model

walks (SRW) [1] shows its better performance through its characteristic in learning how to bias a random walk with restart on the network in a supervised way, which can combine the network structure with the characteristics of nodes and edges of the network into a unified framework comparing with RWR. Therefore, here we exploit SRW for computing user proximities.

However, on online KTV platforms, the interactions among users are much sparser than traditional entertainments, especially the social links. Due to the fact that the user similarity is used to measure the similarity among users' interest in singing songs, instead of directly utilizing social relationships among users, we construct a network according to users' common singing behavior, which is named as *interest network*. Specifically, we construct an edge for two users if they sang at least *ncs* same songs. The value of *ncs* is determined experimentally and variant with platforms. More details about how to choose the value of *ncs* will be introduced in Section 2. If there is no other special instruction, the network of online KTV users is the *interest network* not *social network*. In the following, we will introduce how to use SRW to compute the similarity between users based on the interest network.

Given the user network graph $G(U, E)$, where U is all the user nodes and E is all the edges, if user node u_i and u_m have sung at least *ncs* same songs, an edge (u_i, u_m) is constructed. For each edge (u_i, u_m), we define a corresponding feature vector $\mathbf{\Psi}_{im}$ that describes the user node properties and the interaction attributes. Here the random walk transition probability for each edge (u_i, u_m) is assigned by the strength $a_{im} = f_{\mathbf{w}}(\mathbf{\Psi}_{im})$ according to the problem formulation in supervised random walks. Function $f_{\mathbf{w}}(\mathbf{\Psi}_{im})$ parameterized by the weight vector $\mathbf{w}$ takes the edge feature vector $\mathbf{\Psi}_{im}$ as input. $\mathbf{p_i}$ is the stationary distribution vector of the random walk with restarts from the seed user node u_i, which reflects the relevance scores between other user nodes and u_i. And $\mathbf{p_i}$ is computed as shown in Eq. (2) according to the edge strength function $f_{\mathbf{w}}(\mathbf{\Psi}_{im})$ with the parameter $\mathbf{w}$:

$$\mathbf{p_i} = (1-\mu)Q^{\mathrm{T}}\mathbf{p_i} + \mu\mathbf{q_i}, \tag{2}$$

where the summation of all the entries in $\mathbf{p_i}$ is equal to 1. Specially, Q is the random walk stochastic transition probability matrix and is computed as follows:

$$Q_{im} = \begin{cases} \frac{a_{im}}{\sum_{m_1} a_{im_1}}, & \text{if } (u_i, u_m) \in E, \\ 0, & \text{otherwise,} \end{cases} \tag{3}$$

where m_1 represents the m_1th user in the matrix. $\mathbf{q_i}$ is an $M\times 1$ starting vector, where the ith element is $\sum_{m=1}^{M} p_{im}$ and 0 for others, which incorporates the restart probability μ, i.e., with probability μ the random walk jumps back to seed node u_i and thus "restarts".

From Eq. (2), we can see that before getting the vector $\mathbf{p_i}$, our task is to learn the parameter $\mathbf{w}$ of function $f_{\mathbf{w}}(\mathbf{\Psi}_{im})$. Here we use the same method as in [1] in order to achieve the optimized parameter $\mathbf{w}$. Specifically, for each user u_i, she has two node sets D_i and L_i, where D_i is denoted as destination nodes to which u_i creates edges in the network graph $G(U, E)$, L_i is no link nodes to which u_i does not create edges, and $D_i \cup L_i \cup u_i = U$. Intuitively, we learn the parameter $\mathbf{w}$ in such a way that the random walk will be more likely to visit nodes in D_i than L_i, i.e., $p_{il} < p_{id}$, for each $d \in D_i$ and $l \in L_i$. Please note that p_{il} is the random walk score from user u_i to user u_l and p_{id} is the random walk score from user u_i to user u_d. In practice, it is hard to achieve a solution satisfying all the previous constraints ($p_{il} < p_{id}$). To make the constraints "soft", we introduce a loss function h that penalizes violated constraints. Thus, we obtain the optimization problem to learn the optimal set of $\mathbf{w}$ as follows:

$$\min_{\mathbf{w}} F(\mathbf{w}) = \|\mathbf{w}\|^2 + \varepsilon \sum_{u_i \in U} \sum_{u_d \in D_i, u_l \in L_i} h(p_{il} - p_{id}), \tag{4}$$

where ε is the regularization parameter that trades-off between the complex (i.e., norm of $\mathbf{w}$) and the fit of the model (i.e., how much the constraints can be violated). $h(\cdot)$ is the loss function that assigns a non-negative penalty according to the difference of the scores $p_{il} - p_{id}$. If $p_{il} - p_{id} < 0$ then $h(\cdot) = 0$, while for $p_{il} - p_{id} > 0$, also $h(\cdot) > 0$. In order to find the optimal set of parameters $\mathbf{w}$, we first derive the gradient of $F(\mathbf{w})$ with respect to $\mathbf{w}$, and use a gradient-based optimization method to find $\mathbf{w}$ that minimizes $F(\mathbf{w})$. The derivative is computed as follows:

$$\frac{\partial F(\mathbf{w})}{\partial w_k} = 2w_k + \varepsilon \sum_{u_i \in U} \sum_{u_d \in D_i, u_l \in L_i} \frac{\partial h(p_{il} - p_{id})}{\partial (p_{il} - p_{id})} \left(\frac{\partial p_{il}}{\partial w_k} - \frac{\partial p_{id}}{\partial w_k}\right). \tag{5}$$

It is simple to compute the derivative $\frac{\partial h(p_{il}-p_{id})}{\partial (p_{il}-p_{id})}$ [1]. However, it is not clear how to compute the derivative of the score $\mathbf{p_i}$ with respect to the vector $\mathbf{w}$. In this paper, referring to the method in [1], we apply a power-method based algorithm to compute this derivative by recursively applying the chain rule to the following derivative function in Eq. (6). More specifically, we first iteratively compute $\mathbf{p_i}$ and then repeatedly compute the derivative $\frac{\partial \mathbf{p_i}}{\partial w_k}$ based on the estimate obtained in the previous iteration.

$$\frac{\partial \mathbf{p_i}}{\partial w_k} = (1-\mu)Q^{\mathrm{T}} \frac{\partial \mathbf{p_i}}{\partial w_k} + (1-\mu)\frac{\partial Q}{\partial w_k}^{\mathrm{T}} \mathbf{p_i} + \mu \frac{\partial \mathbf{q_i}}{\partial w_k}, \tag{6}$$

which is computed by taking the derivative over Eq. (2). Here $\frac{\partial Q_{im}}{\partial w_k}$ is computed as follows based on Eq. (3):

$$\frac{\partial Q_{im}}{\partial w_k} = \begin{cases} \frac{\frac{\partial a_{im}}{\partial w_k}(\sum_{m_1} a_{im_1}) - a_{im}(\sum_{m_1} \frac{\partial a_{im_1}}{\partial w_k})}{(\sum_{m_1} a_{im_1})^2}, & \text{if } (u_i, u_m) \in E, \\ 0, & \text{otherwise.} \end{cases} \quad (7)$$

More details on how to compute the derivative of $F(\mathbf{w})$ and learn $\mathbf{w}$ are shown in Algorithm 1 and Algorithm 2.

Algorithm 1 Computation of the derivative of $F(\mathbf{w})$

Require: $G(U, E)$, interest network graph
$\mathbf{w}$, edge feature weight vector
ε, regularization parameter
$a_{im} = f_{\mathbf{w}}(\mathbf{\Psi}_{im})$, edge strength for each user pair (u_i, u_m)
$\mathbf{q_i}$ $M \times 1$ starting vector, the i-th element is $\sum_{m=1}^{M} p_{im}$ and others are 0
Ensure: $\frac{\partial F(\mathbf{w})}{\partial w_k}$, the derivative of $F(\mathbf{w})$
Build the random walk stochastic transition matrix Q and $\frac{\partial Q}{\partial w_k}$
for all $u_i \in U$ **do**
  Get D_i, L_i
  Initialize $\mathbf{p_i}^{(0)}$
  Initialize $\frac{\partial \mathbf{p_i}}{\partial w_k}^{(0)}, k = 1, \ldots, |\mathbf{w}|$
  $t = 1$
  while *not converged* **do**
    $\mathbf{p_i}^{(t)} = (1-\mu)Q^{\mathrm{T}}\mathbf{p_i}^{(t-1)} + \mu\mathbf{q_i}^{(t-1)}$
    $t = t + 1$
  end while
  $t = 1$
  while *not converged* **do**
    for $k = 1, \ldots, |\mathbf{w}|$ **do**
      $\frac{\partial \mathbf{p_i}}{\partial w_k}^{(t)} = (1-\mu)(Q^{\mathrm{T}}\frac{\partial \mathbf{p_i}}{\partial w_k}^{(t-1)} + \frac{\partial Q}{\partial w_k}^{\mathrm{T}}\mathbf{p_i}) + \mu\frac{\partial \mathbf{q_i}}{\partial w_k}^{(t-1)}$
    end for
    $t = t + 1$
  end while
end for
return
$\frac{\partial F(\mathbf{w})}{\partial w_k} = 2w_k + \varepsilon \sum_{u_i \in U} \sum_{u_d \in D_i, u_l \in L_i} \frac{\partial h(p_{il}^{(t-1)} - p_{id}^{(t-1)})}{\partial (p_{il}^{(t-1)} - p_{id}^{(t-1)})} (\frac{\partial p_{il}^{(t-1)}}{\partial w_k} - \frac{\partial p_{id}^{(t-1)}}{\partial w_k})$

Algorithm 2 Computation of $\mathbf{w}$ by applying alternating gradient descent procedure

Require: λ, the step size for gradient descent
Ensure: $\mathbf{w}$, edge feature weight vector
Initialize $w^{(0)}$
$t = 1$
while *not converged* **do**
  for $k = 1, \ldots, |\mathbf{w}|$ **do**
    $w_k^{(t)} = w_k^{(t-1)} - \lambda\frac{\partial F(\mathbf{w})}{\partial w_k}$
  end for
  $t = t + 1$
end while
return $\mathbf{w}$

Finally, we can obtain the user similarity matrix B based on users' learnt stationary distribution vectors. Specifically, in order to get the similarity B_{im} between user u_i and user u_m, we first compute their dissimilarity by exploiting the symmetrized Kullback Liebler (KL) distance:

$$KL(u_i, u_m) = \frac{1}{2}\sum_{m_1=1}^{M} p_{im_1}\log_2\frac{p_{im_1}}{p_{mm_1}} + \frac{1}{2}\sum_{m_1=1}^{M} p_{mm_1}\log_2\frac{p_{mm_1}}{p_{im_1}}, \quad (8)$$

and then get their similarity B_{im} by the function: $B_{im} = e^{-KL(u_i,u_m)}$, which fits the requirement of similarity definition that $B_{im} \in [0, 1]$ and $B_{im} = B_{mi}$.

4.2 Song similarity calculation

Unlike the user similarity calculation, there is no direct network structure among songs. Thus SRW is not appropriate for song similarity calculation. To calculate the similarity between songs, we first extract some important features from songs' information (e.g., the type of song, the total times sung by users) for describing each song, then we construct a feature matrix Φ for all songs, where Φ_{jf} means value of the fth feature for the song s_j. Then the similarity between two songs can be measured by comparing their corresponding feature vectors. Let the number of features be F, the cosine similarity between s_j and s_n is computed as follows:

$$C_{j,n}^{cos} = \sum_{f=1}^{F}\Phi_{jf}\Phi_{nf} \Big/ \sqrt{\sum_{f=1}^{F}\Phi_{jf}^2 \sum_{f=1}^{F}\Phi_{nf}^2}. \quad (9)$$

More detailed information about song features and song similarity calculation will be introduced in Section 5.2.

4.3 Model formulation of InfoFuMF

After acquiring the user similarity matrix B and song similarity matrix C, we then introduce the framework of our InfoFuMF as shown in Fig. 2. Specifically, according to the definition of pseudo-rating matrix R in Section 3, we can see that each entry R_{ij} is non-empty if and only if the user u_i has sung the song s_j. However, as most of users have only sung a few songs, it leads to that the matrix R is very sparse, which makes our task "how to effectively fill the matrix" very challenging. In order to solve this task, we formally solve our problem with the framework of matrix factorization, which is good at generating the user latent feature vectors X and song latent feature vectors Y on sparse matrices. What's more, by forcing the low-dimensional representations of two users as close as possible w.r.t their similarity in B, and the low-dimensional representations of two songs as close as possible w.r.t. their similarity in C, we effectively incorporate more information about users and songs into the framework

to improve the prediction ability and mitigate the problem of sparsity. Based on generated X and Y, the predicted rating of user u_i on song s_j can be defined as $\widehat{R}_{ij} =< X_{*i}, Y_{*j} >$, where $<>$ indicates the inner product, X_{*i} is ith user's latent feature vector and Y_{*j} is jth song's latent feature vector. Hence the problem of generating the optimal X and Y can be defined by minimizing the following criterion function as follows:

$$L(X,Y) = \frac{1}{2}\sum_{i=1}^{M}\sum_{j=1}^{N} I_{ij}(R_{ij} - X_{*i}^{\mathrm{T}}Y_{*j})^2 + \frac{\beta}{2}[tr(X^{\mathrm{T}}X) + tr(Y^{\mathrm{T}}Y)] + \frac{\gamma}{2}[tr(XL_BX^{\mathrm{T}})] + \frac{\delta}{2}[tr(YL_CY^{\mathrm{T}})], \tag{10}$$

where I_{ij} is the indicator variable that is equal to 1 if u_i has sung the song s_j and equal to 0 otherwise. L_B is the Laplacian matrix of B, defined as $L_B = Q - B$ with Q being a diagonal matrix whose diagonal entries $Q_{ii} = \sum_j B_{ij}$. L_C, the Laplacian matrix of C, can be calculated by the same way, and $tr(.)$ denotes the trace of a matrix. β, γ, and δ are model regularization parameters for controlling the complexity of models and the contribution from the user similarity and the contribution from the song similarity.

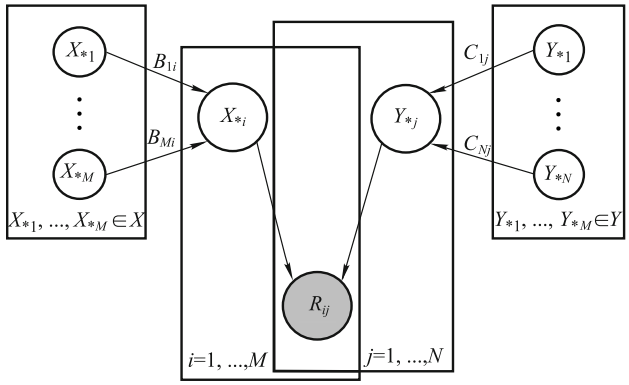


Fig. 2 The graphical model of InfoFuMF

In fact, as the objective function is not convex with regard to U and V, here we turn to use alternating gradient descent procedure to optimize the objective function, which is widely used in these kinds of models and could ensure the local minimum. Before giving the derivatives over the objective function, we first rewrite the objective function in Eq. (10) as follows:

$$\begin{aligned} L &= \frac{1}{2}\sum_{i=1}^{M}\sum_{j=1}^{N} I_{ij}(R_{ij} - X_{*i}^{\mathrm{T}}Y_{*j})^2 + \frac{1}{2}tr[X(\beta E_M + \gamma L_B)X^{\mathrm{T}}] \\ &\quad + \frac{1}{2}tr[Y(\beta E_N + \delta L_C)Y^{\mathrm{T}}] \\ &= \frac{1}{2}\sum_{i=1}^{M}\sum_{j=1}^{N} I_{ij}(R_{ij} - X_{*i}^{\mathrm{T}}Y_{*j})^2 + \frac{1}{2}\sum_{d=1}^{D} X_{d*}(\beta E_M + \gamma L_B)X_{d*}^{\mathrm{T}} \\ &\quad + \frac{1}{2}\sum_{d=1}^{D} Y_{d*}(\beta E_N + \delta L_C)Y_{d*}^{\mathrm{T}}, \end{aligned} \tag{11}$$

where E_M is the identity matrix with M dimensions, E_N is the identity matrix with N dimensions, D is the number of latent features in X and Y. Then we find

$$\begin{aligned} \frac{\partial L}{\partial X_{di}} &= (\sum_{j=1}^{N} I_{ij}Y_{dj}^2)X_{di} - \sum_{j=1}^{N} I_{ij}Y_{dj}(R_{ij} - X_{*i}^{\mathrm{T}}Y_{*j} \\ &\quad + X_{di}Y_{dj}) + (\beta X_{di} + \gamma L_{Bi*}X_{d*}^{\mathrm{T}}) \\ &= ((\sum_{j=1}^{N} I_{ij}Y_{dj}^2)X_{di} + \beta X_{di} + \gamma L_{Bi*}X_{d*}^{\mathrm{T}}) \\ &\quad - \sum_{j=1}^{N} I_{ij}Y_{dj}(R_{ij} - X_{*i}^{\mathrm{T}}Y_{*j} + X_{di}Y_{dj}). \end{aligned} \tag{12}$$

It is obvious that the rows of X are decoupled, thus we can apply the gradient descent to optimize one row of X at a time with the other rows fixed. Let E be a $M \times M$ matrix with $E_{ii} = \sum_{j=1}^{N} I_{ij}Y_{dj}^2$, $\mathbf{f}$ be a $M \times 1$ vector with $f_i = \sum_{j=1}^{N} I_{ij}Y_{dj}(R_{ij} - X_{*i}^{\mathrm{T}}Y_{*j} + X_{di}Y_{dj})$, then the gradient of each row of X can be represented as follows:

$$\frac{\partial L}{\partial X_{d*}} = (E + \beta E_M + \gamma L_B)X_{d*}^{\mathrm{T}} - \mathbf{f}. \tag{13}$$

However, from Eq. (11), we can see that both the rows and the columns of Y are not decoupled, which leads us to compute the gradient of each entry of Y by the function:

$$\frac{\partial L}{\partial Y_{dj}} = \beta Y_{dj} + \delta L_{C\,j*}Y_{d*}^{\mathrm{T}} - \sum_{i=1}^{M} I_{ij}X_{di}(R_{ij} - X_{*i}^{\mathrm{T}}Y_{*j}). \tag{14}$$

The following Algorithm 3 shows the overall learning procedure of InfoFuMF.

5 Experiments

In this section, we conduct experiments for evaluating the effectiveness of our approach in terms of both the rating prediction and song recommendation. Through the experiments, we try to answer the following questions:

1) How does InfoFuMF perform in a real online KTV dataset when compared with state-of-the-art methods?
2) How does the restart probability μ affect the performance of the recommendation?
3) How do the model regularization parameters γ and δ affect the accuracy of the recommendation?
4) How does InfoFuMF perform for different scales of the dataset when compared with baselines?

Algorithm 3 Learning procedure of InfoFuMF

Require: **R** rating matrix
Ψ_{im} edge feature vector for user pair (u_i, u_m), $1 \leqslant i, m \leqslant M$
Φ feature matrix of all songs
D number of latent features
λ step size for gradient descent
Ensure: X the user latent vectors and Y the song latent vectors
Compute the user similarity matrix B based on Ψ
Compute the song similarity matrix C based on Φ
Initialize X^0, Y^0
$t = 1$
while *not converged* **do**
  for $d = 1$ to D **do**
    $X^t_{d*} \longleftarrow X^{t-1}_{d*} - \lambda(\frac{\partial L}{\partial X_{d*}})^{\mathrm{T}}$
    for $j = 1$ to N **do**
      $Y^{(t)}_{dj} \longleftarrow Y^{(t-1)}_{dj} - \lambda \frac{\partial L}{\partial Y_{dj}}$
    end for
  end for
  $t = t + 1$
end while
return X^t, Y^t

5.1 Data set

The data set is provided by ihou which is an online KTV platform launched by iFLYTEK, which contains 196,940 user logs from January 20, 2012 to May 20, 2012. For each user u_i, the user logs include user profiles (i.e., age, gender), user singing records (each record consists of user, song, the evaluation score given by the system), the users whom u_i follows, the users who follow u_i and the messages sent by u_i to other users. The total number of songs in the system is 6992. We crawled the genre of each song from the baidu website. First, we prune the dataset for our analysis. Specifically, for users, we only keep those users who sang at least 16 distinct songs and were active in the system at least 7 days. For songs, we only keep those songs which are sung by at least 10 distinct users. After that, there are 10,008 users and their rich log information and 818 songs as described in Table 2. For each user, we split her records by 3:2 over time.

Table 2 The description of filtered data set

Data	#Users	#Songs	#Singing	Avg.#UserSinging
TrainSet	10,008	818	255,554	25.5
TestSet	10,008	818	170,368	17.0

5.2 Preparation

Before comparing the effectiveness of different models, here we conduct a preliminary experiment on the *ihou* data set to show the assumption "*users may not be achieving high scores on their favorite songs*", which has been claimed in Section 1. Considering the simplicity and effectiveness of evaluation metrics, we adopt Spearman's rho (sr) [38] to calculate the correlation of two ranking lists. For each user u_i, we first rank her singing songs ss_i by her preferences ($Rank_{1i}$) and her proficiency ($Rank_{2i}$) respectively. Then the correlation efficient sr_i of user u_i can be expressed as

$$sr_i = 1 - \frac{6\sum_{j=1}^{|ss_i|}(rank_{1i,j} - rank_{2i,j})^2}{|ss_i|^2 - |ss_i|}, \tag{15}$$

where $|ss_i|$ is the number of ss_i, j is the jth song in ss_i, $rank_{1i,j}$ is the ranking position of jth song in $Rank_{1i}$ and $rank_{2i,j}$ is the ranking position of jth song in $Rank_{2i}$. And the range of sr_i is from 0 to 1, where lower value of sr_i means that the $Rank_{1i}$ and $Rank_{2i}$ are more dissimilar. Distributions of all users' sr are shown in Fig. 3. We can observe that more than 92% users' Spearman's rho are not more than 0.50 and the average of Spearman's rho is 0.22, which confirms our assumption that *users may not be achieving high scores on their favorite songs*. It also implies the necessity of our proposed song recommendation model, which can balance user preferences with user proficiency on recommending songs to online KTV users.

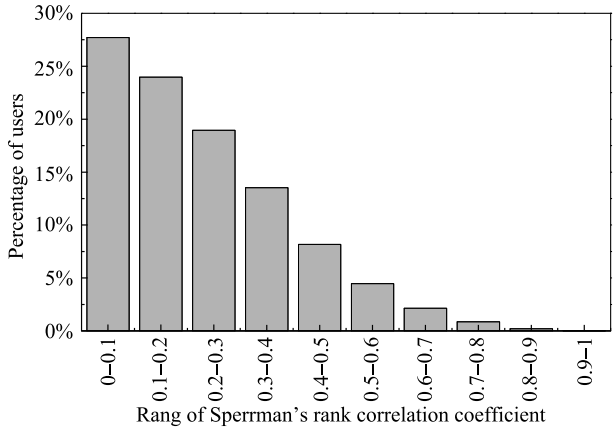


Fig. 3 Distributions of users over range of Spearman's rho

Next, we construct the pseudo-rating matrix's element R_{ij} according to the method proposed in Section 3, where the number of singing c_{ij} can be acquired from the singing records of u_i and the system's evaluation g_{ij} is the average value among the c_{ij} covers. Consequently, there are 232,842 non-empty elements in the rating matrix, which is far less than 7,953,702 empty elements in this matrix. To better understand user behavior on songs sung, we also illustrate the distributions of the number of distinct songs sung by users and the number of distinct users who sing each song in

Fig. 4. From the two figures, the average number of distinct songs sung by each user (25.12) is far less than 818 and the average number of users singing each song (284.87) is also far less than 10,008, which also reflects that the rating matrix is extremely sparse. This observation verifies the concern mentioned in Section 1.

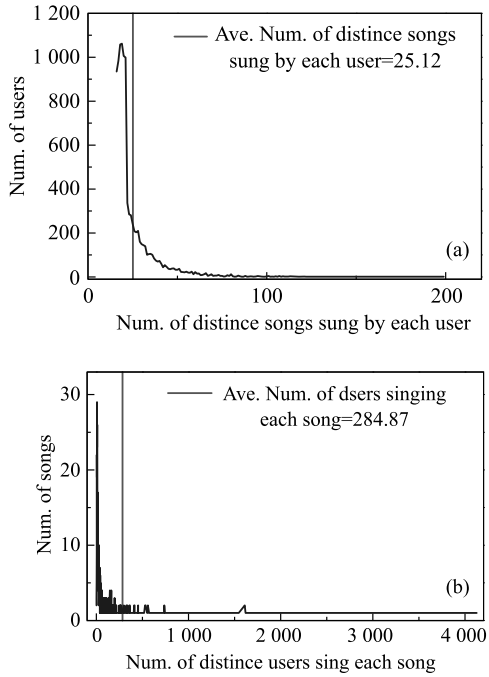


Fig. 4 The distributions of the number of distinct songs sung by users and the number of distinct users who sing each song, respectively. (a) Distribution of distince songs; (b) distribution of distince users

Interest network In order to compute the user similarity matrix B based on SRW, we construct the user interest network $G(U, E)$ at the training dataset. The edge feature vector Ψ_{im} for each edge (u_i, u_m) is constructed based on the following features: (1) Distance between u_i and u_m's ages, ad_{im}; (2) Distance of singing ability between u_i and u_m by measuring the difference of the two users' average singing scores, cad_{im}; (3) Number of common friends, ncf_{im}; (4) Number of common songs sung by u_i and u_m, ncs_{im}. Please note that features (1), (3) and (4) are used to consider users' preferences, and feature (2) is used to consider users' proficiency. Before utilizing these features, to make highly skewed features' distributions less skewed, we adopt the log transformation function [39]:

$$f' = \frac{\log(1+f)}{\max\{\log(1+f)\}}, \tag{16}$$

normalizing these features into the ranging [0,1], where f is the value of one of the mentioned edge features. Since the similar users are also the like-minded users in CF for song choice, we construct an edge for two user nodes u_i and u_m if they sang at least 4 of the same songs, instead of direct social links. In consideration of user singing behavior in the training dataset, the average number of distinct songs sung by each user is 15.5. If two users have sung at least 4 of same songs, we can assume that the two users have the same interest in songs with a higher probability. The density of the interest network is 10.93% when the number of common singing songs is not less than 4, which is enough to learn user similarity. We also analyze the number of the direct social links among KTV users. We find that the number is 5058 and the density of the social network is 0.00005, which is too sparse to learn user similarity. The extreme sparsity of direct social network also demonstrates the necessity of constructing the interest network. The edge strength function $f_{\mathbf{w}}(\Psi_{im})$ mentioned in Section 1 is defined by $a_{im} = f_{\mathbf{w}}(\Psi_{im}) = exp(\mathbf{w} \cdot \Psi_{im})$. By default, in the following experiments the regularization parameter ε is set to 1 as the same as in [1], the step size λ of gradient descent for computation of $\mathbf{w}$ in Algorithm 1, user latent vectors X and song latent vectors Y are all set to 0.01.

Song features The song similarity matrix C is calculated based on the following song features: (1) Genre of each song; (2) Average score of each song according to user singing records; (3) Singer ID of each song; (4) Average number of singing by each user who has sung the song; (5) Total number of users who have sung the song. The first three features indicate the basic attributes while the last two reflect the popularity of each song in the online KTV platform. The value of feature (2) ranges from 0 to 100. Feature (4) and feature (5) are integer value. Obviously, these three features are ordinal attributes, which means that we can utilize the cosine similarity method to compute the distance directly. Before utilizing these three features, we adopt the log transformation function as used in user features: $f' = \frac{\log(1+f)}{\max\{\log(1+f)\}}$. However, as feature (1) and (3) are non-ordinal attributes, we can not put them into the cosine similarity method together with features (2), (4) and (5) directly. To solve the problem, we introduce an index function hi combined with the cosine similarity method. Thus Eq. (9) in Section 4.2 has been changed to Eq. (17):

$$C^{cos}_{j,n} = \frac{hi(\Phi_{j1}, \Phi_{n1})}{3} + \frac{hi(\Phi_{j3}, \Phi_{n3})}{3} + \frac{\sum\limits_{f=2,4,5} \Phi_{jf}\Phi_{nf} \Big/ \sqrt{\sum\limits_{f=2,4,5} \Phi^2_{jf} \sum\limits_{f=2,4,5} \Phi^2_{nf}}}{3}, \tag{17}$$

where hi is the index function that assigns 0 or 1 according to Φ_{jf} and Φ_{nf} ($f = 1$ or $f = 3$). If $\Phi_{jf} = \Phi_{nf}$ then $hi = 1$, which implies that songs j, f belong to a same category. For

example, if $\Phi_{j3} = \Phi_{n3}$, it means that these two songs are sung by a same singer. While for $\Phi_{jf} \neq \Phi_{nf}$, $hi = 0$. According to Eq. (17), we have successfully fuse ordinal attributes and non-ordinal attributes into a same space and can compute the similarity between two songs directly.

Age segmentation The user's age is an important feature which can reflect user's interest, as users maybe have the same interests and habits as their peers. Although two users' ages are different, they still may have similar interests. Based on this hypothesis, we split users' age in different classes. To make more reasonable age splitting, we use a method based on information gain adopted in [40, 41] to get the age splits. The information entropy of the age A^Y is $Ent(A^Y) = -\sum_{y=1}^{|A^Y|} p_y \log(p_y)$, where $|A^Y|$ is the number of different years in A^Y and p_y is the proportion of users in age year A_y in this age class. Initially, the entire age range is viewed as a big class and then we partition it into several classes recursively. In each iteration, we use the weighted average entropy (WAE) to find the best split:

$$WAE(y_s; A^Y) = \frac{|A_1^Y(y_s)|}{|A^Y|} Ent(A_1^Y(y_s)) + \frac{|A_2^Y(y_s)|}{|A^Y|} Ent(A_2^Y(y_s)),$$

where A_1^Y and A_2^Y are two subclasses of class A_Y when being split at the y_sth year. The best split year induces a maximum information gain given by $\Delta E(y_s)$ which is equal to $Ent(A_Y) - WAE(y_s; A^Y)$.

5.3 Evaluation baselines

Since the problem of recommending the right songs to the right users on online KTV is under-explored, it is difficult to directly find existing works as baselines. As our model is inspired by PMF, we select PMF as a baseline. As we mentioned that the user similarity based on SRW can better depict users' interest and correlations than traditional methods (e.g., cosine similarity), to validate this hypothesis, we devise a baseline Normal information-fused MF model, denoted as NormalMF. Moreover, in order to answer the following two questions: (1) How does user interest network data improve collaborative filtering? (2) How does song information improve collaborative filtering? We design two baselines (UserMF, SongMF) which only takes advantage of information of one entity for improving the ability of song recommendation. Besides, since interest correlation we constructed in Section 4 may have a significant influence on the effect of recommendation, we also consider the recent works (SoRec, MFCModel, MF-TDP) about recommendation system in order to make comparisons between our proposed model and traditional social recommendation systems. Thus the baselines are used in our experiments as follows:

1) **NormalMF**: The only difference of NormalMF and InfoFuMF is that NormalMF utilizes the cosine similarity to calculate user similarity not SRW.
2) **UserMF**: This model employs the user similarity based on the supervised random walks algorithm described in Section 1 to regularize the MF procedure. It is a special case of our proposed model when $\delta = 0$.
3) **SongMF**: This baseline introduces the song similarity based on the song information described in Section 2. It is also a special case of our model when $\gamma = 0$.
4) **PMF**: It is the basic matrix factorization model that only models user-song matrix.
5) **SoRec** [17]: The SoRec model factorizes the rating and trust information simultaneously to solve data sparsity and poor rating prediction.
6) **MFCModel** [21]: The MFCModel exploits the community structure of user's social networks to improve the recommendation's accuracy.
7) **MF-TDP** [42]: The Mini-SGD-Based Matrix Factorization with Trust and Distrust Propagation (MF-TDP) exploits the trust and distrust information which are extracted from user's social information to improve the recommending accuracy by using the mini-batch stochastic gradient descent optimization.

Note that since the social links among online KTV users are much sparser than traditional entertaining platforms, the social recommendation systems will perform worse by utilizing the direct social information among users. Therefore, the social recommendation methods (SoRec, MFCModel and MF-TDP) in our experiments also exploit the interest network instead of the direct social network.

5.4 Evaluation metrics

To evaluate the effectiveness of our approach, we use a common metric mean squared error (MSE) [43] to measure the performance of rating prediction. Specifically, for better understanding our model's performance on rating prediction, we devise three different MSEs:
(1) MSE on preference:

$$\text{MSE}_{prefer} = \frac{\sum_{ij}(R_{ij}^{(prefer)} - \widehat{R}_{ij})^2}{N}, \tag{18}$$

which is used to measure the performance of prediction on

user preferences;
(2) MSE on proficiency:

$$\mathrm{MSE}_{profic} = \frac{\sum_{ij}(R_{ij}^{(profic)} - \widehat{R}_{ij})^2}{N}, \tag{19}$$

which is used to measure the performance of prediction on singing proficiency;
(3) MSE on average:

$$\mathrm{MSE}_{ave} = \frac{\mathrm{MSE}_{prefer} + \mathrm{MSE}_{profic}}{2}, \tag{20}$$

which is used to comprehensively measure the rating prediction performance.

On the other hand, we adopt the widely used indices: precision P, recall R and F1 score $F_1 = 2 * \frac{P*R}{P+R}$, which are used to evaluate the effectiveness of song recommendations.

5.5 Overall results

In our experiments, we split the records into two parts according to time. One part containing 60% of observations in the rating matrix is used as the training set and the other part with 40% of observations is used as the test set. We report both the rating prediction results (Fig. 5) and the ranking results (Table 3). Except for those parameters which have been set in Section 2, the remaining parameters are set by $\mu = 0.3$, $\beta = 0.1$, $\gamma = 0.1$, $\delta = 0.25$ and latent feature dimension $D = 20$ according to experimental results.

Rating prediction Figure 5 exhibits the MSE_{prefer}, MSE_{profic} and MSE_{ave} of all models. From the figure, we can see that PMF has the worst performance compared with InfoFuMF, NormalMF, UserMF and SongMF, which implies that employing the user similarity and song similarity can improve the accuracy of rating prediction. On MSE_{prefer}, we find that SoRec achieves the best performance (MSE_{prefer} =0.038), which shows that SoRec can accurately capture user's preferences on songs for singing compared with other models. However, it can not accurately predict user's proficiency on songs (MSE_{profic} = 0.195) compared with other models, such as InfoFuMF (MSE_{profic} = 0.052). And it also performs (MSE_{ave} = 0.116) worse than InfoFuMF (MSE_{ave} = 0.064) on predicting user rating on songs. In consideration of these observations, we can draw a conclusion that our proposed model InfoFuMF can accurately predict user preferences and user proficiency on songs.

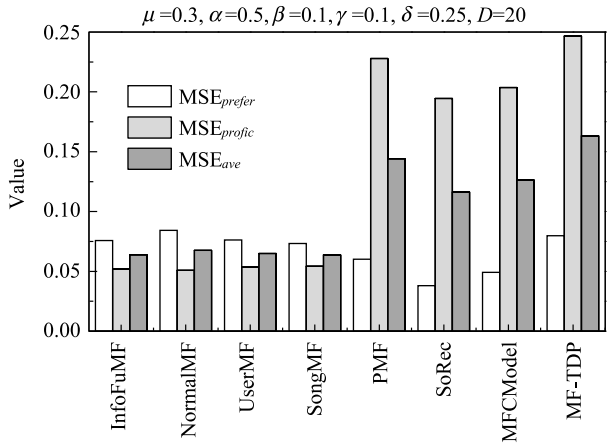


Fig. 5 Rating prediction evaluation on MSE_{prefer}, MSE_{profic} and MSE_{ave} with different methods

Song recommendations In order to evaluate our model's recommendation quality thoroughly, we compare its performance with baselines in terms of precision, recall and F1-score under different settings of recommendation songs (topN) including 10, 20, 30, 40 and 50. As shown in Table 3, the proposed InfoFuMF model consistently outperforms the other six models. Compared with InfoFuMF, NormalMF, UserMF and SongMF, the performance of PMF is generally the worst with the majority of numbers of recommendation songs, which is as expected because it does not utilize user similarity and song similarity. Particularly, the comparison between InfoFuMF and NormalMF demonstrates that user similarity based on SRW can better depict users' interest and correlations than cosine similarity. And the results of SongMF are close to InfoFuMf. It is probably resulted from that the information used to construct the song similarity matrix is richer than the information for the user similarity matrix, which contributes to better predicting user selections on songs for singing. We also observe that the three recommendation methods SoRec, MFCModel and MF-TDP are all consistently worse than InfoFuMF. The possible reason is that our data set can not be well applied by the three methods as the constructed online KTV user network has some unique characteristics, such as users tend to challenge other users who have achieved higher scores on singing some songs. What's more, when we increase the number of recommendation songs, the performance of the proposed model InfoFuMF first increases, and achieves the better F1-score 14.2% around 40 and then slightly decreases. In parameters' sensitive analysis's sections, we will choose "the number of top recommendation songs" *topN* as 40 on account of its performance.

Compared with baselines, our model InfoFuMF can comprehensively predict user rating on songs by balancing user preferences with user proficiency. Except for precisely predicting user rating, InfoFuMF can also accurately recommend songs to users for singing. For example, when *topN* =

Table 3 The comparison performance on evaluating recommendation quality in terms of precision, recall and F1-score with different methods

Index	topN	InfoFuMF	NormalMF	UserMF	SongMF	PMF	SoRec	MFCModel	MF-TDP
Precision	10	**11.9%**	11.7%	9.0%	11.8%	9.8%	7.6%	10.9%	8.4%
	20	**10.7%**	10.3%	8.4%	10.4%	7.5%	5.8%	10.1%	6.5%
	30	**9.9%**	**9.9%**	6.6%	**9.9%**	6.3%	5.0%	8.2%	5.6%
	40	**9.5%**	9.1%	6.1%	9.2%	5.6%	4.5%	7.0%	5.0%
	50	**8.4%**	8.2%	6.1%	8.3%	5.0%	4.0%	6.2%	4.5%
Recall	10	**11.1%**	11.0%	8.4%	11.0%	9.8%	7.8%	10.3%	8.5%
	20	**20.0%**	19.4%	15.6%	19.5%	14.5%	11.6%	18.9%	12.8%
	30	**27.9%**	27.8%	18.4%	**27.9%**	18.0%	14.9%	22.8%	16.1%
	40	**35.6%**	34.0%	22.7%	34.71%	21.0%	17.5%	25.7%	19.0%
	50	**39.6%**	38.3%	28.3%	38.6%	23.6%	19.7%	28.6%	21.4%
F1-score	10	**10.7%**	10.6%	8.0%	10.6%	9.1%	7.3%	9.9%	7.9%
	20	**13.0%**	12.6%	10.2%	12.7%	9.2%	7.3%	12.4%	8.0%
	30	**13.8%**	13.7%	9.1%	**13.8%**	8.8%	7.1%	11.4%	7.8%
	40	**14.2%**	13.6%	9.1%	13.8%	8.3%	6.8%	10.4%	7.5%
	50	**13.3%**	12.9%	9.5%	13.0%	7.9%	6.4%	9.7%	7.1%

40, the F1-score of InfoFuMF is 14.2%, while SoRec is only 6.4%. In a word, the performance of InfoFuMF on rating prediction and song recommendations demonstrate the effectiveness of our proposed model. However, as the main purpose of this paper is generating recommendation songs to KTV users, in the following section, we will focus on evaluating performance difference of variant parameters of InfoFuMF in terms of precision, recall and F1-score.

5.6 Impact of parameter α

In Section 3, we have claimed that the songs which users select to sing are mainly determined by user preferences and user proficiency. We show the experimental results with different α in Fig. 6. Specifically, α is a trade-off parameter between user preferences and user proficiency on singing. If $\alpha = 0$, users' selections on songs are simply determined by user proficiency. In contrast, if $\alpha = 1$, only user preferences determine users' selections on songs. Specifically, Fig. 6(a) exhibits the performance on rating prediction of different α from 0 to 1, given $\mu = 0.3, \beta = 0.1, \gamma = 0.1, \delta = 0.25$ and $D = 20$. We can observe that the MSE_{prefer} decreases when α increases, which conforms to the fact that InfoFuMF utilizes more information from user preferences as α increases. However, when α increases, the MSE_{profic} also increases as InfoFuMF weakens the contribution from user proficiency. As we known, the MSE_{ave} is devised to measure the rating prediction of InfoFuMF not only on user preferences but also on user proficiency. We find that InfoFuMF achieves the best performance $\text{MSE}_{ave} = 0.064$ when $\alpha = 0.5$. And InfoFuMF achieves worse performance $\text{MSE}_{ave} = 0.108$ when $\alpha = 0$ and $\text{MSE}_{ave} = 0.102$ when $\alpha = 1$. These observations demonstrate that the proposed model can not achieve the best performance on rating prediction when we only utilize the information from user preferences or user proficiency. We also exhibit the proposed model's performance on song recommendations with variant α in Fig. 6(b), given $\mu = 0.3$, $\beta = 0.1$, $\gamma = 0.1$, $\delta = 0.25$, $D = 20$ and $topN = 40$. We also find that InfoFuMF achieves the best performance $Precision = 9.5\%$, $Recall = 35.6\%$ and $F1 - score = 14.2\%$ when $\alpha = 0.5$, which shows that user proficiency plays an important role in users' selections on songs as well as user preferences. Similar

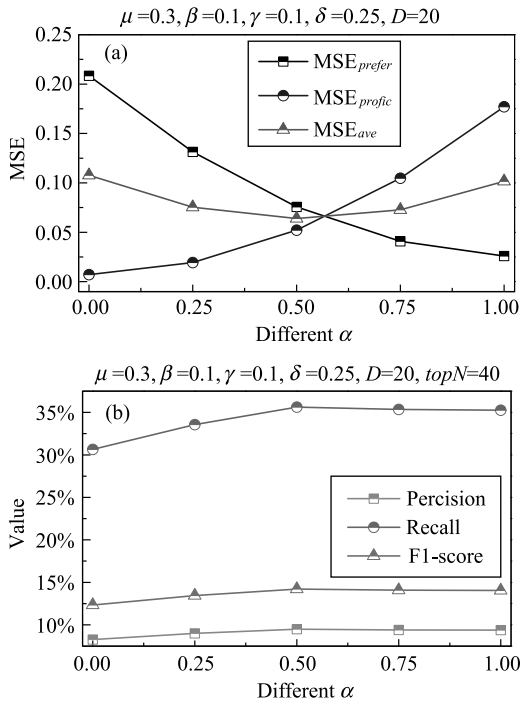


Fig. 6 InfoFuMF's performance with different α. (a) Performance on rating prediction; (b) performance on song recommendation

to the performance on rating prediction, if we simply employ user preferences or user proficiency, the model can not achieve a better result than $\alpha = 0.5$. In a word, both observations on rating prediction and observations on song recommendations confirm our assumption that not only user preferences but also user proficiency determine online KTV users' selections on songs.

5.7 Impact of parameter μ

During the procedure of computing user similarity by the SRW model, μ is an important parameter since it defines the probability the random walk jumps back to seed node and thus "restarts". In Fig. 7, we study how the parameter μ affects the performance of InfoFuMF. Figure 7 exhibits the impact of different μ from 0.1 to 0.9, given $\beta = 0.1$, $\gamma = 0.1$, $\delta = 0.25$, $topN = 40$ and $D = 20$. We can observe that the model achieves the best performance $Precision = 9.5\%$, $Recall = 35.6\%$ and $F1\text{-}score = 14.2\%$ when $\mu = 0.3$. A larger or smaller μ value will lead to hurt the performance of the proposed InfoFuMF model. For example, the model achieves the worst performance $Precision = 9.1\%$, $Recall = 34.0\%$ and $F1\text{-}score = 13.6\%$ when $\mu = 0.1$. It implies that the SRW can more effectively combine the network structure with the characteristics of nodes and edges, and learn user similarity better when $\mu = 0.3$ compared with other μ values.

5.8 Impact of parameters β, γ and δ

In our proposed InfoFuMF model, parameters β, γ and δ control the complexity of the model, and balance the contribution of user similarity with the contribution of song similarity. If $\gamma = 1$ and $\delta = 0$, we simply employ the information of user similarity and ignore the contribution of song similarity. If we decrease the value of γ and increase δ, we will utilize more information from song similarity to learn user interest on singing. For other situations, we fuse information from the user similarity and song similarity for matrix factoring and, moreover, to make song recommendations for online KTV users.

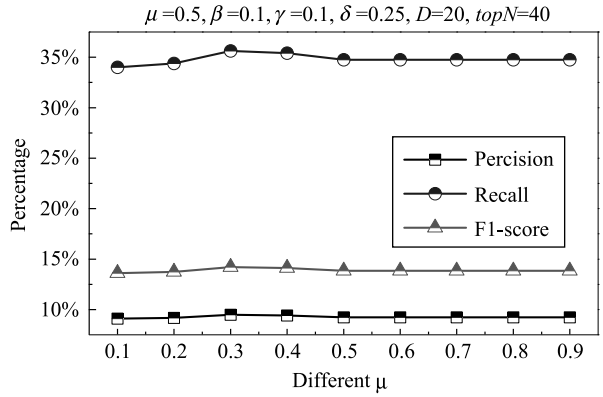


Fig. 7 InfoFuMF's performance variance with different μ

The impacts of β, γ and δ on precision, recall and F1-score are shown in Fig. 8. First, we analyze the performance variance with different β in Fig. 8(a). As β increases, the F1-score increases at first from 8.86% ($\beta = 0.01$) to 9.49% ($\beta = 0.075$). Afterwards, the F1-score achieves a little growth from 9.49% ($\beta = 0.075$) to 9.57% ($\beta = 0.25$). It implies that the InfoFuMF model can achieve a competitive result when β is around 0.075.

Parameters γ and δ control the contribution of user similarity and song similarity respectively. As shown in Fig. 8(b), at first, the F1-score increases from 13.8% ($\gamma = 0$) to 14.2% ($\gamma = 0.1$), but when γ surpasses a certain threshold, the F1-score does not increase with further increase of γ. It implies that the InfoFuMF model achieves the best performance when γ is around 0.1. The impact of parameter δ is exhibited in Fig. 8(c), and it shows that larger δ will increase the accuracy of song recommendations. However, we must pay attention to the fact that the increment of δ brings larger increment of performance compared with γ, which implies that the InfoFuMF model is more sensitive on parameter δ compared with parameter γ. Based on these observations from Fig. 8, we can draw a conclusion that richer information (users and

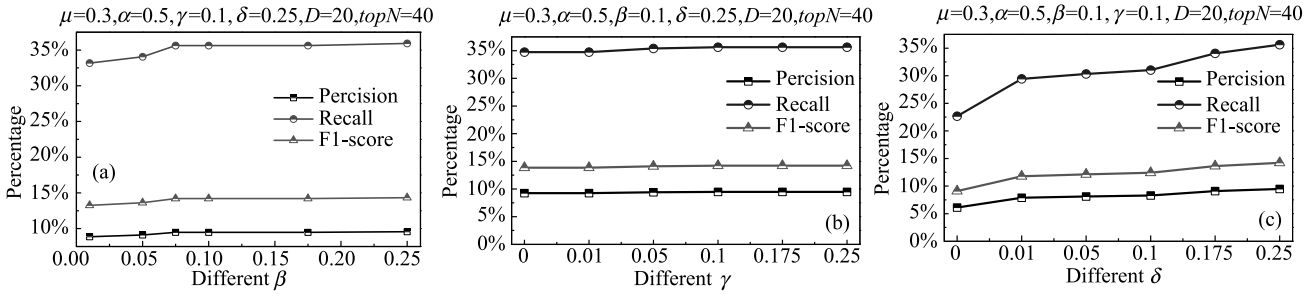


Fig. 8 The performance of precision, recall and F1-score with different assignments of β, γ and δ, respectively. (a) Performance on parameter β; (b) performance on parameter γ; (c) performance on parameter δ

songs) can generate better performance than purely using user information or song information for recommendations.

5.9 Importance of features

Above experiments have demonstrated the effectiveness of the InfoFuMF model and fully explored the impacts of different parameters, but the effect of each user feature or song feature on recommendation is unexplored. Here we evaluate all features to understand what information form the best song recommendation system. Specifically, we remove song features one by one from InfoFuMF model respectively to evaluate the importance of each song feature. Similarly, we also remove user features one by one from InfoFuMF respectively to evaluate the importance of each user feature. Based on above procedures, we can obtain all features' importance on recommending songs to online KTV users as shown in Fig. 9. According to Fig. 9, we can acquire the importance of each feature and draw several implications: (1) In consideration of both user features and song features, most of removed features take a little decrease on F1-score, which shows that the InfoFuMF model has a good robustness on recommending songs to online Karaoke users; (2) Figure 9(a) shows that removing feature "Number of common friends" or "Singing ability" from InfoFuMF achieves a larger performance loss (0.7%) compared with other user features, which implies that the feature "Number of common friends" or "Singing ability" can effectively improve the accuracy of user similarity calculation; (3) Based on Fig. 9(b), we can find that the feature "Average singing number" achieves the largest performance loss (0.6%) compared with other song features, which implies that the feature "Average singing number" plays the most important role in calculating song similarity. Considering the meaning of each feature, we can draw a rough conclusion that these features ("Singing ability" and "Average singing number"), which relate to user singing activity, can effectively improve the recommendation quality of the proposed InfoFuMF model. It also demonstrates that song recommendation system for online KTV users has unique characteristics compared with general music recommendation systems.

5.10 Scalability analysis

In this section, we conduct some experiments on measuring the scalability of our proposed InfoFuMF method compared with PMF, SoRec, MFCModel and MF-TDP. Note that in this section, as NormalMF, UserMF and SongMF are similar to InfoFuMF, we only keep the result of InfoFuMF for simplicity. Specifically, we obtain five sub-datasets with different user scales from the ihou dataset, including 100 users, 500 users, 1,000 users, 5,000 users and 10,008 users. On each of the 5 data sets, we run each model using the same parameter settings for fairness as shown in Fig. 10. For scalability analysis, we evaluate the performance on recommendation quality by F1-score with varying scales of users. As shown in Fig. 10, we can see that the accuracy of recommendations consistently decreases with users increasing, which implies

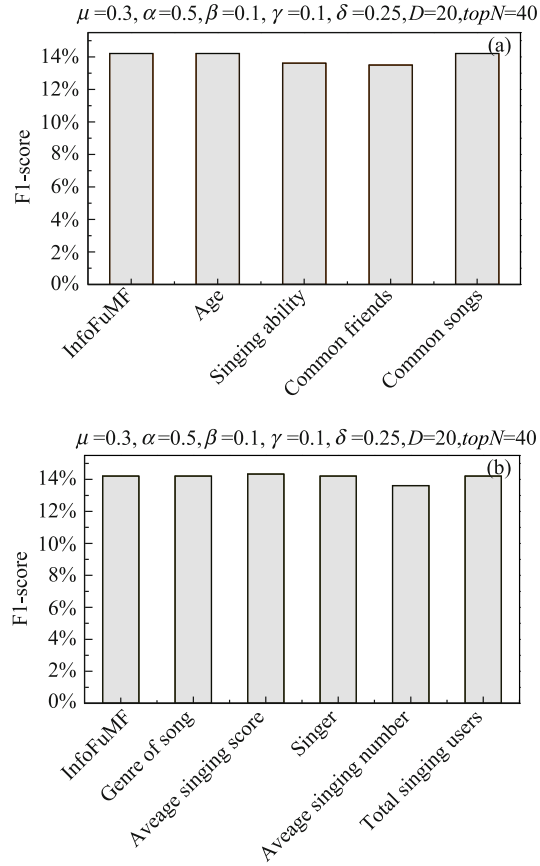


Fig. 9 Evaluation of user features and song features, respectively. (a) Evaluation of user features; (b) evaluation of song features

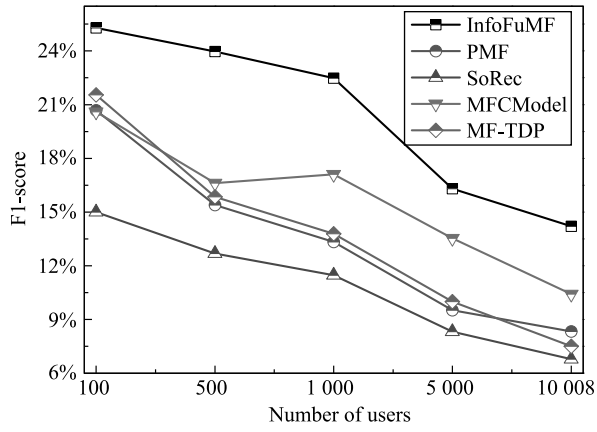


Fig. 10 The performance on recommendation quality of different models with variant scales of users

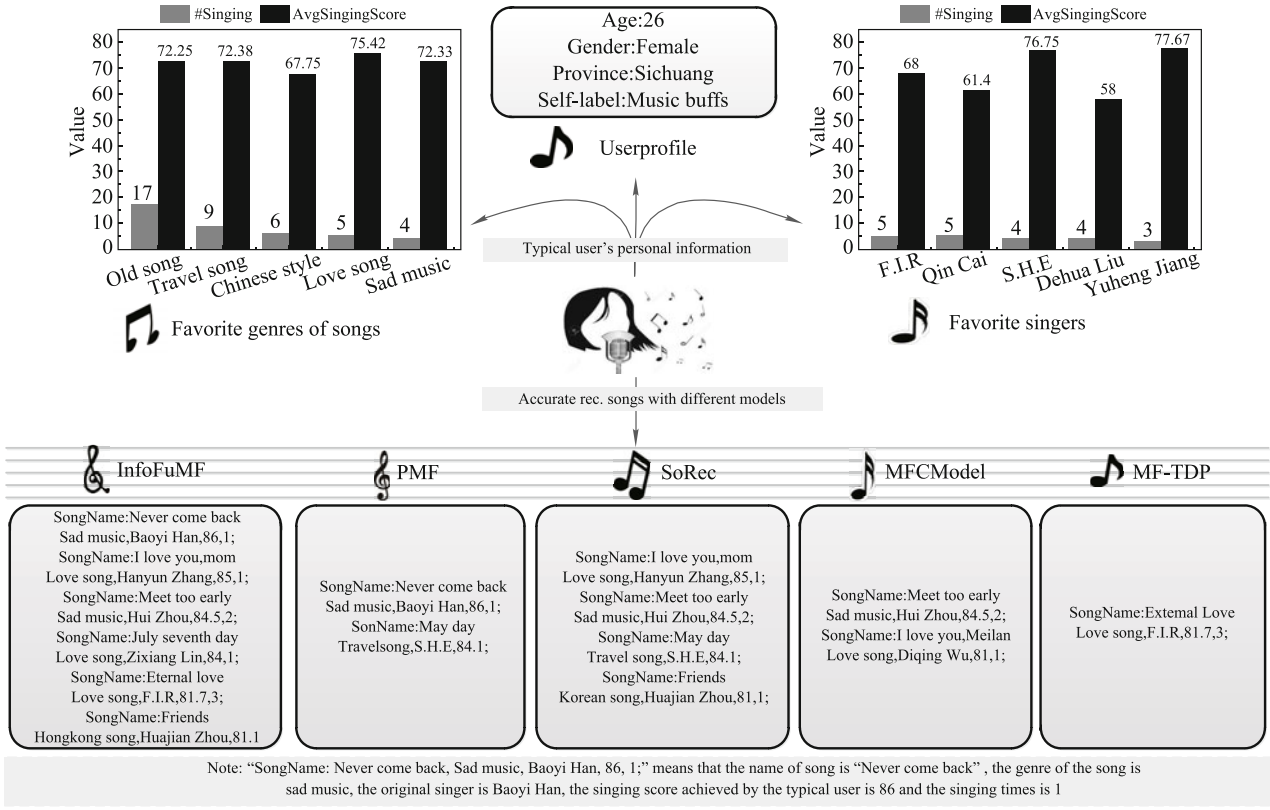


Fig. 11 A typical user's personal information and the accurate recommendation songs for this typical user on topN = 15 with different methods

that all models perform worse in larger scale of dataset for recommendations. The possible reason is that the sparsity of rating matrix becomes more severe with the increase of users. However, although the performance of InfoFuMF decreases with users increasing, it still performs the best compared with all baselines on all scales of users.

5.11 Case study

In order to better understand the performance of our model in terms of song recommendation for online KTV users, we conduct a case study on the *ihou* dataset.

As shown in Fig. 11, we first present the personal information of a typical user, including "User profile", "Favorite genres of songs" and "Favorite singers", which is used to depict the preferences and behavior of this typical user. According to the personal information, we can clearly see that, the user does not always sing the songs which the user performs the best. For example, the genre of songs, which is always sung by the user, is "Old song" (the number of singing on this genre is 17), while the genre of songs that the user achieved the highest average singing score (75.42) is "Love song". Except for user's personal information, we also exhibit the accurate recommendation songs on topN = 15 with different methods to better explore InfoFuMF's performance. As the NormalMF, UserMF and SongMF are similar to InfoFuMF, we omit the results of these three models in Fig. 11 for simplicity. From these listed accurate recommendation songs in Fig. 11, we can have an intuitive observation. The accurate recommendation songs of InfoFuMF is 6, while the accurate recommendations of PMF, SoRec, MFCModel and MF-TDP are 2, 4, 2 and 1 respectively, which demonstrates that the InfoFuMF model achieves the best accuracy on recommendation compared with other models. It is because employing the information of user similarity and song similarity improves recommendation quality. In other words, our proposed InfoFuMF model can accurately recommend songs to online-KTV users.

6 Conclusion

In this paper, we studied the problem of personalized Karaoke song recommendation task by leveraging the proficiency and preferences of singers. Specifically, to balance user preferences with user proficiency, we proposed a new rating function, which constructs a pseudo-rating matrix combining the number of singing and system evaluation scores. Further-

more, to mitigate the sparseness challenge in the pseudo-rating matrix, we first calculated user similarity and song similarity by utilizing richer user and song information respectively. Then, we fused the user similarity, the song similarity and the pseudo-rating matrix into the unified matrix factorization model (InfoFuMF), which can mitigate the challenge of data sparsity for accurately recommending songs to online KTV users. Finally, we conducted extensive experiments on a real-world data set, and the results clearly demonstrated the effectiveness of our proposed model compared with baselines. We believe this work could improve user experience of online KTV platforms.

In our future work, we will fuse more data from other platforms to better mitigate the data sparsity problem. What's more, as the cold start problem severely affects the experience of new users who first use the online Karaoke system, we will improve the proposed InfoFuMF model to better recommend songs to new users.

Acknowledgements The authors thank Qi Liu for valuable suggestions, and thank Liying Zhang for her help to polish English writing of this paper. This research was partially supported by grants from the National Key Research and Development Program of China (2016YFB1000904), the National Natural Science Foundation of China (Grant Nos. 61325010 and U1605251), and the Fundamental Research Funds for the Central Universities of China (WK2350000001). Le Wu gratefully acknowledges the support of the Open Project Program of the National Laboratory of Pattern Recognition (201700017), and the Fundamental Research Funds for the Central Universities (JZ2016HGBZ0749). Yong Ge acknowledges the support of the National Natural Science Foundation of China (NSFC, Grant Nos. 61602234 and 61572032).

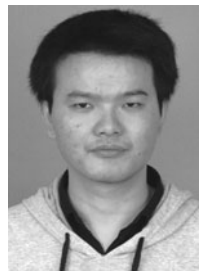
References

1. Backstrom L, Leskovec J. Supervised random walks: predicting and recommending links in social networks. In: Proceedings of the 4th ACM International Conference on Web Search and Data Mining. 2011, 635–644
2. McFee B, Barrington L, Lanckriet G. Learning content similarity for music recommendation. IEEE Transactions on Audio, Speech, and Language Processing, 2012, 20(8): 2207–2218
3. Liu N H. Comparison of content-based music recommendation using different distance estimation methods. Applied Intelligence, 2013, 38(2): 160–174
4. Bogdanov D, Haro M, Fuhrmann F, Gómez E, Herrera P. Content-based music recommendation based on user preference examples. Copyright Information, 2010, 33: 1–6
5. Guan C, Fu Y J, Lu X J, Xiong H, Chen E H, Liu Y L. Vocal competence based karaoke recommendation: a maximum-margin joint model. In: Proceedings of 2016 SIAM International Conference on Data Mining. 2016, 135–143
6. Soleymani M, Aljanaki A, Wiering F, Veltkamp R C. Content-based music recommendation using underlying music preference structure. In: Proceedings of 2015 IEEE International Conference on Multimedia and Expo. 2015, 1–6
7. Guo C. Feature generation and selection on the heterogeneous graph for music recommendation. In: Proceedings of the 9th ACM International Conference on Web Search and Data Mining. 2016, 715
8. Celma O. The exploit-explore dilemma in music recommendation. In: Proceedings of the 10th ACM Conference on Recommender Systems. 2016, 377
9. Song T H, Peng Z H, Wang S Z, Fu W J, Hong X G, Yu P S. Based cross-domain recommendation through joint tensor factorization. In: Proceedings of International Conference on Database Systems for Advanced Applications. 2017, 525–540
10. Wu X, Liu Q, Chen E H, He L, Lv J S, Cao C, Hu G P. Personalized next-song recommendation in online karaokes. In: Proceedings of the 7th ACM Conference on Recommender Systems. 2013, 137–140
11. Jin R, Chai Y J, Si L. An automatic weighting scheme for collaborative filtering. In: Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. 2004, 337–344
12. Herlocker J L, Konstan J A, Terveen L G, Riedl J T. Evaluating collaborative filtering recommender systems. ACM Transactions on Information Systems, 2004, 22(1): 5–53
13. Sarwar B, Karypis G, Konstan J, Riedl J. Item-based collaborative filtering recommendation algorithms. In: Proceedings of the 10th International Conference on World Wide Web. 2001, 285–295
14. Koren Y, Bell R, Volinsky C. Matrix factorization techniques for recommender systems. Computer, 2009, 42(8): 30–37
15. Sindhwani V, Bucak S, Hu J, Mojsilovic A. A family of non-negative matrix factorizations for one-class collaborative filtering problems. In: Proceedings of the ACM Conference on Recommender Systems. 2009
16. Liu J T, Wu C H, Liu W Y. Bayesian probabilistic matrix factorization with social relations and item contents for recommendation. Decision Support Systems, 2013, 55(3): 838–850
17. Ma H, Yang H X, Lyu R M, King I. Sorec:social recommendation using probabilistic matrix factorization. In: Proceedings of the 17th ACM Conference on Information and Knowledge Management. 2008, 931–940
18. Ma H, King I, Lyu R M. Learning to recommend with social trust ensemble. In: Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval. 2009, 203–210
19. Jamali M, Ester M. A matrix factorization technique with trust propagation for recommendation in social networks. In: Proceedings of the 4th ACM Conference on Recommender Systems. 2010, 135–142
20. Yang X W, Steck H, Liu Y. Circle-based recommendation in online social networks. In: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2012, 1267–1275
21. Li H, Wu D M, Tang W B, Mamoulis N. Overlapping community regularization for rating prediction in social recommender systems. In: Proceedings of the 9th ACM Conference on Recommender Systems. 2015, 27–34
22. Eck D, Lamere P, Bertin-Mahieux T, Green S. Automatic generation of social tags for music recommendation. In: Proceedings of the 20th International Conference on Neural Information Processing Systems.

2008, 385–392

23. Yan Y, Liu T L, Wang Z Y. A music recommendation algorithm based on hybrid collaborative filtering technique. In: Proceedings of Chinese National Conference on Social Media Processing. 2015, 233–240
24. Hofmann T. Collaborative filtering via gaussian probabilistic latent semantic analysis. In: Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval. 2003, 259–266
25. Si L, Jin R. Flexible mixture model for collaborative filtering. In: Proceedings of the 20th International Conference on Machine Learning. 2003, 704–711
26. Liu N N, Yang Q. Eigenrank: a ranking-oriented approach to collaborative filtering. In: Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. 2008, 83–90
27. Liu Q, Chen E H, Xiong H, Ding C H Q, Chen J. Enhancing collaborative filtering by user interest expansion via personalized ranking. IEEE Transactions on Systems, Man, and Cybernetics, Part B, 2012, 42(1): 218–233
28. Porteous I, Asuncion A U, Welling M. Bayesian matrix factorization with side information and dirichlet process mixtures. In: Proceedings of the 24th AAAI Conference on Artificial Intelligence. 2010, 563–568
29. De Campos L M, Fernández-Luna J M, Huete J F, Rueda-Morales M A. Combining content-based and collaborative recommendations: a hybrid approach based on Bayesian networks. International Journal of Approximate Reasoning, 2010, 51(7): 785–799
30. Park S, Kim Y D, Choi S. Hierarchical Bayesian matrix factorization with side information. In: Proceedings of the 23rd International Joint Conference on Artificial Intelligence. 2013, 1593–1599
31. Agarwal D, Chen B C. Regression-based latent factor models. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2009, 19–28
32. Yoshii K, Goto M, Komatani K, Ogata T, Okuno G H. An efficient hybrid music recommender system using an incrementally trainable probabilistic generative model. IEEE Transactions on Audio, Speech, and Language Processing, 2008, 16(2): 435–447
33. Li Q, Myaeng H S, Kim M B. A probabilistic music recommender considering user opinions and audio features. Information Processing and Management, 2007, 43(2): 473–487
34. Cheng R, Tang B Y. A music recommendation system based on acoustic features and user personalities. In: Proceedings of Pacific-Asia Conference on Knowledge Discovery and Data Mining. 2016, 203–213
35. Benzi K, Kalofolias V, Bresson X, Vandergheynst P. Song recommendation with non-negative matrix factorization and graph total variation. In: Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing. 2016, 2439–2443
36. Krivitsky N P, Handcock S M, Raftery E A, Hoff D P. Representing degree distributions, clustering, and homophily in social networks with latent cluster random effects models. Social Networks, 2009, 31(3): 204–213
37. Kooti F, Lerman K, Aiello M L, Grbovic M, Djuric N, Radosavljevic V. Portrait of an online shopper: understanding and predicting consumer behavior. In: Proceedings of the 9th ACM International Conference on Web Search and Data Mining. 2016, 205–214
38. Spearman C. The proof and measurement of association between two things. The American Journal of Psychology, 1904, 15(1): 72–101
39. Zumel N, Mount J, Porzak J. Practical Data Science with R. Manning Publications Co., 2014
40. Liu Q, Ge Y, Li Z M, Chen E H, Xiong H. Personalized travel package recommendation. In: Proceedings of the 11th IEEE International Conference on Data Mining. 2011, 407–416
41. Liu Q, Chen E H, Xiong H, Ge Y, Li Z M, Wu X. A cocktail approach for travel package recommendation. IEEE Transactions on Knowledge and Data Engineering, 2014, 26(2): 278–293
42. Forsati R, Mahdavi M, Shamsfard M, Sarwat M. Matrix factorization with explicit trust and distrust side information for improved social recommendation. ACM Transactions on Information Systems, 2014, 32(4): 17
43. Wackerly D, Mendenhall W, Scheaffer L R. Mathematical Statistics with Applications. Nelson Education, 2007

Ming He is currently a PhD student in the School of Computer Science and Technology at University of Science and Technology of China (USTC), China. His major research interests include recommendation system, user behavioral analysis, and machine learning. He has published several papers in refereed conference proceedings and journals, such as CIT'15, DASFAA'16, KSEM'17 and TWEB.



Hao Guo is currently a research engineer of Living Analytics Research Centre (LARC), School of Information System, Singapore Management University, Singapore. He received the MS degree of Computer Science from the University of Science and Technology of China, China in 2017 and the BE degree of Software Engineering from Northeastern University, China in 2014. His research interests lie in recommendation system and natural language processing, with an emphasis on deep recommendation system, question answering and reinforcement learning.

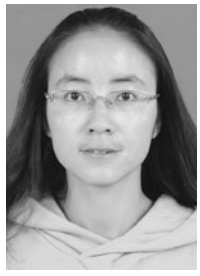
Guangyi Lv received the BE degree in Computer Science and Technology in 2013 from Sichuan University, China. He is currently a PhD student in the School of Computer Science and Technology at University of Science and Technology of China (USTC), China. His major research interests include deep learning, natural language

processing, and recommendation system. He has published several papers in refereed conference proceedings, such as AAAI'16, AAAI'17 and PAKDD'15.

Le Wu is currently a Faculty Member with the Hefei University of Technology (HFUT), China. She received the PhD degree in Computer Science from University of Science and Technology of China (USTC), China. Her general area of research is data mining, recommender system, and social network analysis. She has published several papers in referred journals and conferences, such as TKDE, TIST, AAAI, IJCAI, KDD, SDM and ICDM. Dr. Le Wu is the recipient of the Best of SDM 2015 Award.

Yong Ge is an assistant professor of Management Information Systems in University of Arizona, USA. He received the PhD degree in information technology from Rutgers, The State University of New Jersey, USA in 2013. His research interests include data mining and business analytics. He received the ICDM-2011 Best Research Paper Award. He has published prolifically in refereed journals and conference proceedings, such as TKDE, TOIS, TKDD, ACM SIGKDD, SIAM SDM, IEEE ICDM and ACM RecSys.

Enhong Chen is a professor and vice dean of the School of Computer Science at USTC, China. He received the PhD degree from USTC, China. His general area of research includes data mining and machine learning, social network analysis and recommender systems. He has published more than 100 papers in refereed conferences and journals, including IEEE Trans. KDE, IEEE Trans. MC, KDD, ICDM, NIPS and CIKM. He was on program committees of numerous conferences including KDD, ICDM, SDM. His research is supported by the National Science Foundation for Distinguished Young Scholars of China. He is a senior member of the IEEE.



Haiping Ma is a head of Big Data Research in IFLYTEK CO., LTD.. She received the PhD degree in information technology from University of Science and Technology of China, China. Her research interests include user behavior modeling and computational advertising. She has published in refereed journals and conference proceedings, such as WWW, CIKM, ICDM, KSEM, Neurocomputing and IJITDM.