Exploiting Music Play Sequence for Music Recommendation

Zhiyong Cheng[1], **Jialie Shen**[2*], **Lei Zhu**[3], **Mohan Kankanhalli**[1], **Liqiang Nie**[4]
[1]School of Computing, National University of Singapore
[2]Department of Computer and Information Sciences, Northumbria University, UK
[3]School of Information Technology & Electrical Engineering, The University of Queensland
[4]School of Computer Science and Technology, Shandong University, China
{jason.zy.cheng, leizhu0608, jialie, nieliqiang}@gmail.com, mohan@comp.nus.edu.sg

Abstract

Users leave digital footprints when interacting with various music streaming services. Music play sequence, which contains rich information about personal music preference and song similarity, has been largely ignored in previous music recommender systems. In this paper, we explore the effects of music play sequence on developing effective personalized music recommender systems. Towards the goal, we propose to use word embedding techniques in music play sequences to estimate the similarity between songs. The learned similarity is then embedded into matrix factorization to boost the latent feature learning and discovery. Furthermore, the proposed method only considers the k-nearest songs (e.g., $k = 5$) in the learning process and thus avoids the increase of time complexity. Experimental results on two public datasets demonstrate that our methods could significantly improve the performance on both *rating prediction* and *top-n recommendation* tasks.

1 Introduction

With the popularity of mobile devices and online music streaming service (e.g., Last.fm[1] and Spotify[2]), people are able to easily access tens of millions of songs. Such ubiquitous music consumption paradigm also poses many new challenges in develpoing smart music retrieval system, which can assist users to search their favorite music from such huge music collections [Schedl *et al.*, 2014; Shen *et al.*, 2012; 2010]. In real life, interactions between users and those online music service platforms generate various digital footprints, including users' demographical data, user-generated content (i.e., tags, comments and feedback), and logs of listening history. Such information is very valuable for music recommendation and has been extensively studied in previous literature [Knees and Schedl, 2013; Schedl *et al.*, 2015; Cheng *et al.*, 2016]. For example, (user, song, rating) or (user, song, playcount) triples, which are obtained from user's listening logs, have been widely used to infer user's music preference for effective song recommendation [Cheng and Shen, 2014]. However, music play sequence (MPS), which contains rich information about user listening behaviors and music content similarity, has not been well exploited for music recommendation in previous studies.

Users often continuously enjoy a set of songs during a certain time period without interruption, which is called a *listening session*. Recent studies reveal strong correlations among the songs played within one session [Hariri *et al.*, 2012]. For example, users usually listen to songs of the same *album*, *artist*, or *genre* in a listening session [Ji *et al.*, 2015]. Besides, users often organize songs into playlists and play songs in a playlist together under a certain context [Cheng and Shen, 2016; Cheng *et al.*, 2016], which indicates that the songs played together share some common characteristics [Hariri *et al.*, 2012; Wang *et al.*, 2016a]. In addition, to demonstrate the correlations among songs in a session, [Ji *et al.*, 2015] mapped songs into the Euclidean space and showed that the songs listened in a session are close to each other in this space. Therefore, music play sequences in listening sessions encode certain music similarities or correlations, which can be leveraged to boost the performance of music recommendation. In fact, MPS has been used in supporting music retrieval applications, such as playlist generation [Bonnin and Jannach, 2015]. Very surprisingly, it has not been well exploited in developing music recommender systems.

As one of the most effective recommendation techniques, matrix factorization (MF) [Koren *et al.*, 2009; He *et al.*, 2016] aims to learn latent vectors of users and items by decomposing the user-item rating/interaction matrix, which records user-item ratings or interaction times. While MF based approaches have achieved very promising performance, they general ignore the importance of interaction sequence. In this study, the main objective is to investigate the potential of exploiting MPS in music recommendation. Specifically, we attempt to leverage the information encoded in MPS into MF methods to improve the recommendation performance. Towards the goal, a *song2vec* method is developed to estimate the music similarities between songs, which are hidden in the MPS data. This method is inspired by the word2vec techniques (i.e., skip-gram) [Mikolov *et al.*, 2013] in NLP. Word2vec is a highly scalable prediction model for learning word embeddings from texts. It learns low-dimensional vec-

*Corrsponding Author
[1]http://www.last.fm/
[2]https//www.spotify.com

tor representations for words by analyzing their usages over a large text corpus. This technique is based on the distribution hypothesis [Sahlgren, 2008]: the words appearing in the same contexts represent close or similar meaning if their meanings are not identical [Vasile *et al.*, 2016]. A similar hypothesis can be applied to the MPS data. In song2vec, each song is treated as a "word", and song play sequence in a listening session can be regarded as a "sentence". The final experimental performance demonstrates that this method could effectively estimate the similarities between songs based on the play sequence data.

Further, a novel k-nearest song regularized matrix factorization method is developed to integrate the estimated music similarity into standard matrix factorization (MF) methods, such as biased MF (BMF) [Koren *et al.*, 2009] and weighted regression MF (WRMF) [Hu *et al.*, 2008], to facilitate the latent factor learning. The proposed method can effectively integrate the play sequence information via the co-factorization of the song-song similarity matrix and user-item interaction matrix. An advantage of the proposed method is that it benefits from both user-song interaction matrix and song-song similarity matrix, while avoiding the complexity of analyzing music content for computing song-song similarity, which itself is not a trivial task [Shen *et al.*, 2012; 2010]. To cope with the increase of time complexity introduced by the song-song similarity matrix, for each song, only the nearest k songs (e.g., $k = 5$) are considered in the factorization process.

We conduct experiments over two public music datasets Last.fm-1K [Celma Herrada, 2009] and 30Music datasets [Turrin *et al.*, 2015] on both *rating prediction* and *top-n recommendation* tasks. Results demonstrate that the proposed method outperforms matrix factorization methods [Koren *et al.*, 2009; Hu *et al.*, 2008] on both tasks, due to the ability of our model exploiting the information of music play sequence. In summary, the main contributions of this paper include:

- To best of our knowledge, this is the first study of embedding music play sequence information into matrix factorization for enhancing performance of music recommendation;
- We propose (1) a song2vec method, which could effectively leverage MPS to analyze the similarity between songs; and (2) an efficient k-nearest song regularized MF model, which can effectively leverage the song-similarity matrix to improve the recommendation performance without the increase of time complexity;
- We conduct comprehensive experimental study over two large public datasets to assess the proposed methods on rating prediction and top-n recommendation tasks. Results demonstrate that our method outperform all baselines in terms of different evaluation metrics.

The remainder of this paper is organized as follows: Section 2 gives a brief overview of related work; Section 3 details the song2vec method and k-nearest song regularized MF method; Section 4 and Section 5 introduce the experimental setup and reports the results, respectively. Finally, Section 6 concludes the paper.

2 Related Work

Our study is mainly related to three research directions: matrix factorization, music recommendation, and word embedding.

Matrix factorization Driven by its great success on Netflix competition [Koren *et al.*, 2009] and Yahoo!Music recommendations [Koenigstein *et al.*, 2011], matrix factorization (MF) is a well-established technique in recommender system [Koren *et al.*, 2009; He *et al.*, 2017; Zhang *et al.*, 2016]. The basic idea of MF is to decompose the user-item interaction matrix to learn the latent vectors of users and items. The dot product between the resulted user and item latent vectors is then applied to support effective recommendation. MF is originally proposed for explicit rating datasets. Later on, [Hu *et al.*, 2008] proposed a weighted matrix factorization method to deal with implicit feedback datasets. Recently, many techniques have been proposed to leverage content information in matrix factorization to enhance the recommendation performance, such as Factorization Machines [Rendle *et al.*, 2011], co-factorization [Fang and Si, 2011], and graph regularization [Benzi *et al.*, 2016; Wang *et al.*, 2016b].

Music recommendation aims to suggest users suitable music by inferring their music preferences. Different kinds of side information have been applied in matrix factorization to boost recommendation accuracy. A typical example is to add temporal dynamics and music taxonomy bias (i.e., artist, album, and genre) [Koenigstein *et al.*, 2011]. In [Cheng and Tang, 2016], both acoustic feature and user personalities are considered in the development of an intelligent recommendation algorithm. [Schedl *et al.*, 2015] explored the usage of user's demographic information in collaborative filtering. [Bu *et al.*, 2010] used multiple kinds of social media information (e.g., user friendship, tags, artists, genre, and album) and acoustic content. In recent years, more research efforts have been devoted to explore user-related contexts in music recommendation. For example, in [Cheng and Shen, 2014], music popularity trends and user's current location context are taken into consideration to facilitate personalized music recommendation. [Cheng and Shen, 2016] consider the influence of venue types on user's music preference and develop a venue-aware music recommender system. However, to the best of our knowledge, how to apply the information of user's music play sequence in matrix factorization to improve the recommendation performance has not been explored.

Word2vec [Mikolov *et al.*, 2013] is an effective method to learn embedding representations for words. It models the words' contextual correlations in word sequences, and has shown very promising results in capturing both syntactic and semantic relationships between words. More recently, the model has been extended to paragraph2vec [Le and Mikolov, 2014] and other variants for specific purposes. Since word2vec is effective in capturing the correlation between items, it has also been applied for item modeling and recommendation such as product [Vasile *et al.*, 2016], venue [Ozsoy, 2016], and music recommendation [Wang *et al.*, 2016a]. In the previous works, the learn feature vectors via word2vec are used to find the nearest items for

recommendation [Wang *et al.*, 2016a; Vasile *et al.*, 2016; Ozsoy, 2016]. Different from those methods, our approach applies the technique in music play sequence to mine song-to-song similarities, which are then used to regularize matrix factorization for latent factor learning.

3 Method Description

In this section, we introduce our proposed model, which is a joint matrix factorization method by integrating music similarity encoded in music play sequence (MPS) into the conventional user-item rating matrix factorization.

3.1 Background

Matrix factorization (MF) [Koren *et al.*, 2009] is one of the most effective recommendation algorithms in recommender systems. Given a matrix R of dimensions $|U| \times |I|$ to represent the rating data of all users on all items in a dataset. Each element $r_{u,i}$ denotes the rating of user u gave to item i. MF maps users and items into a latent factor space of dimensionality d and predicts ratings $\hat{r}_{u,i} \in R$ for a user u on item i according to

$$\hat{r}_{u,i} = p_u^T q_i \tag{1}$$

where $p_u \in \mathbb{R}^d$ and $q_i \in \mathbb{R}^d$ are the learned latent factor vector of user u and item i, respectively. In the context of music recommendation, p_u captures music preferences of user u, and q_i models the music characteristics of song i. $p_u^T q_i$ characterizes user u's overall interests over item i. With the consideration of biases caused by user and item effects, this basic model is extended to the biased MF [Koren *et al.*, 2009]:

$$\hat{r}_{u,i} = \mu + b_u + b_i + p_u^T q_i \tag{2}$$

where μ is the average rating, and b_u and b_i are the user and item biases. The parameters $\Theta = \{b_u, b_i, p_u, q_i\}$ are learned by minimizing the objective function:

$$\min_{q*,p*} \sum_{u,i} (r_{u,i} - \hat{r}_{u,i})^2 + \lambda \Omega(\Theta) \tag{3}$$

where $\Omega(\Theta)$ is the regularization term:

$$\Omega(\Theta) = (||p_u||_2^2 + ||q_i||_2^2 + b_u^2 + b_i^2) \tag{4}$$

where $||\cdot||_2$ is ℓ_2-norm [Gao *et al.*, 2012; Wang *et al.*, 2015; Nie *et al.*, 2015; Zhang *et al.*, 2013]. The above biased MF is developed to deal with the cases where explicit ratings are available, whereas in many cases, we can only observe user's interaction times on items, such as how many times a user played a song (i.e., playcounts). To cope with the problem in implicit feedback datasets, [Hu *et al.*, 2008] introduced a weighted regression MF (WRMF) method, in which binary variables $r_{u,i}$ and a new set of variables $\lambda_{u,i}$ are introduced. $r_{u,i} = 1$ if user u has interacted with item i; otherwise, $r_{u,i} = 0$. $\lambda_{u,i}$ is the *confidence* value of observing $r_{u,i}$. The objective function of WRMF is:

$$\min_{q*,p*} \sum_{u,i} \lambda_{u,i} (r_{u,i} - \hat{r}_{u,i})^2 + \lambda \Omega(\Theta) \tag{5}$$

where $\hat{r}_{u,i} = p_u^T q_i$ and $\Omega(\Theta) = (||q_i||_2^2 + ||p_u||_2^2)$.

3.2 The Proposed Method

The MF methods aim to discover the latent feature vectors of users and items based on the user-item interaction matrix, which only records the preference/interaction times of users on items. However, the schemes have not taken user access patterns on items into account. The order users interacted with items contains rich information about the correlation/similarity between items. For example, because of user's local preferences, items interacted at the same time or under similar contexts usually are more similar. When listening to music, users intend to enjoy the songs which fit their local preferences [Hariri *et al.*, 2012; Wang *et al.*, 2016a] together. Moreover, the songs currently played could invoke user's emotion, which affects the selection of subsequent songs. Thus, if the correlations/similarities of songs, which are encoded in the MPS, can be simultaneously modeled in MF, more effective latent feature vectors p_u and q_i can be learned. Based on the key observation, our method is built upon the standard MF model by integrating the information in MPS to facilitate more effective feature learning.

Song2vec Music Similarity Our method is motivated by the skip-gram negative sampling (SGNS) model [Mikolov *et al.*, 2013] in NLP. This method has been proven to be highly effective on capturing the relation between a word and its surrounding words in a sentence. Moreover, it can be trained very efficiently on large-scale datasets. Given the listening logs of users, a "corpus" with lots of play sequences could be constructed. Let $\mathcal{M}$ denote the entire sequence set and $m \in \mathcal{M}$ be a specific sequence. SGNS learns song representations by maximizing the objective function over the entire set $\mathcal{M}$ of music play sequences. The loss function is defined as follows,

$$L = \sum_{m \in \mathcal{M}} \sum_{s_i \in m} \sum_{-c \le j \le c, j \neq 0} logP(s_{i+j}|s_i) \tag{6}$$

where $P(s_{i+j}|s_i)$ denotes the probability of observing a neighboring song s_{i+j} given the current song s_i. Using the soft-max function, it can be defined as

$$P(s_{i+j}|s_i) = \frac{exp(\boldsymbol{v}_{\boldsymbol{s_i}}^T \boldsymbol{v}'_{\boldsymbol{s_{i+j}}})}{\sum_{s' \in \mathcal{V}} exp(\boldsymbol{v}_{\boldsymbol{s_i}} \boldsymbol{v}'_{\boldsymbol{s}})} \tag{7}$$

where $\boldsymbol{v}_{\boldsymbol{s}}$ and $\boldsymbol{v}'_{\boldsymbol{s}}$ are the input and output vector of song s in SGNS. c is the length of the context considered in the song sequences, and $\mathcal{V}$ denotes the vocabulary of songs. From the equations, we can see that SNGS could model the context of play sequence. Based on the learned vectors of songs using SGNS, the similarity between songs can be estimated using cosine similarity.

Objective Function Our method integrates the song-to-song similarity estimated based on song2vec into matrix factorization. The objective function of our method is:

$$\min_{q*,p*} \frac{1}{2} \sum_{u,i} (r_{u,i} - \hat{r}_{u,i})^2 + \frac{\alpha}{2} \sum_{i,j \neq i} (s_{i,j} - q_i^T q_j)^2 + \frac{\lambda}{2} \Omega(\Theta) \tag{8}$$

where $s_{i,j} \in \mathcal{S}$ is the similarity between song i and song j. $\hat{r}_{u,i}$ can be computed as Eq. 2. Since the similarity matrix $\mathcal{S}$

is a real-value dense matrix, time complexity of solving the objective function is significantly increased, comparing to the standard matrix factorization method. To alleviate the problem, for each song, only the k nearest songs are considered in our model. The objective function becomes:

$$\min_{q*,p*} \frac{1}{2}\sum_{u,i}(r_{u,i}-\hat{r}_{u,i})^2+\frac{\alpha}{2}\sum_{i,j\in n(i,k)}(s_{i,j}-q_i^T q_j)^2+\frac{\lambda}{2}\Omega(\Theta) \tag{9}$$

where $n(i,k)$ denotes the k nearest neighbors of song i.[3] From the equation, we can see that the added regularization term is to force the similar songs to be mapped into close positions in the latent space. Thus, the consideration of the top-k nearest songs should have a good effect on achieving the goal. This point has been validated in our experiments. It turns out that $k \in [3, 7]$ could achieve good results. Given such a small value of k, the song similarity matrix used in the learning process is highly sparse, comparing to the user-song rating/interaction matrix. Therefore, the increased time complexity due to the added regularization term becomes negligible (see the time complexity analysis below).

Notice that the similarities between songs (i.e., $s_{i,j}$) are pre-computed and then integrated into the objective function. Thus, they can be learned from external datasets, as long as the similarities of songs in the targeted datasets can be estimated. Therefore, our method can be generalized to other matrix factorization methods. In the above, we show how to mine the music similarity from music play sequence and integrate it into the biased MF method. Analogously, the same methodology could be applied in the WRMF method (Eq. 5) to integrate the play sequence information. To avoid redundant elaboration, we skip the description of the method applied in WRMF. Notice that in the top-n recommendation experiments (Sect. 4), we add the regularization term in WRMF to validate the effectiveness of the k-nearest song regularization method.

Optimization The objective function minimization (Eq. 3) can be solved by gradient descent. This process results in a local minimum solution. Let L denote the loss, the gradients of p_u, q_i, b_u, b_i are computed as:

$$\frac{\partial L}{\partial p_u} = \sum_{i=1}^{N}(\hat{r}_{u,i}-r_{u,i})q_i+\lambda p_u \tag{10}$$

$$\frac{\partial L}{\partial q_i} = \sum_{u=1}^{M}(\hat{r}_{u,i}-r_{u,i})p_u-\alpha\sum_{j\in n(i,k)}(s_{i,j}-q_i^T q_j)q_j+\lambda q_j \tag{11}$$

$$\frac{\partial L}{\partial b_u} = \sum_{u=1}^{M}\sum_{i=1}^{N}(\hat{r}_{u,i}-r_{u,i})+\lambda b_u \tag{12}$$

$$\frac{\partial L}{\partial b_i} = \sum_{u=1}^{M}\sum_{i=1}^{N}(\hat{r}_{u,i}-r_{u,i})+\lambda b_i \tag{13}$$

where M and N are the total number of users and songs in the datasets. Eq. 11 exploits the similarity of songs obtained from music play sequences, and other equations are the same as they are in the biased MF.

Algorithm Complexity The time complexity of computing the objective function in Eq. 9 is $O(d|R| + d|S_{i,j}| + d(M + N))$, where d is the dimension of the learned feature vectors p_u and q_i, and $|R|$ is the observed ratings in the user-song rating matrix. $|S_{i,j}|$ is the number of used similarity pairs. The time complexities of the gradients in Eq. 10, Eq. 11, Eq. 12, and Eq. 13 are $O(d|R| + dM)$ (all users are considered), $O(d|R| + d|S_{i,j}| + dN)$ (all songs are considered), $O(d|R| + dM)$, and $O(d|R| + dN)$, respectively. Therefore, the time complexity of one iteration is thus $O(d|R| + d|S_{i,j}| + d(M + N))$. The increasing time complexity of our method over bias MF is $O(d|S_{i,j}|)$. Reminding that for each song, only the k nearest songs are considered, namely, $|S_{i,j}| = k \cdot N$. Notice that k is very small (e.g., $k = 5$ in our experiments) and $|R| \gg M, N$, the increasing time complexity over biased MF is negligible. In practice, the time complexity of our algorithm could be equivalent to $O(d|R|)$, which is linear with the number of observed ratings in the rating matrix.

Table 1: Statistics of the evaluation datasets

Dataset	# Users	# Songs	# Play sessions	Sparsity
Lastfm-1k	979	71,097	519,043	96.76%
30Music	37,973	170,823	745,187	99.90%

4 Experimental Setup

To validate the effects of exploiting music play sequence in recommendation, we evaluate the proposed algorithm and its competitors over two tasks:

- **Rating prediction** aims to predict the rating $\hat{r}_{u,i}$ of user u given to item i. It is a standard task to evaluate the performance of recommendation on explicit rating datasets.
- **Top-n recommendation** is to evaluate the accuracy of the top results returned by recommendation methods. It is often used to evaluate the performance of recommendation methods on implicit datasets [He *et al.*, 2016].

Dataset Two publicly accessible datasets are used for evaluation: Last.fm-1k music[4] [Celma Herrada, 2009] and 30Music dataset[5] [Turrin *et al.*, 2015].

Last.fm-1k dataset has been widely used in music recommendation experiments. This dataset includes the listening history of 992 users and 961,417 songs, recorded by $(user, timestamp, artist, song)$ quadruples. Based on the quadruples records, we can extract listening sessions and also get the user-song playcount matrix. In implementation, listening sessions are extracted by concatenating the songs played by a user continuously without more than 800s interruption. The threshold (800s) is used to keep consistent with definition in the 30music dataset [Turrin *et al.*, 2015].

[3] Notice c in Eq. 6 denotes the number of neighbors defined based on the sequence of playlist, while k denotes the number of nearest neighbors defined by similarity.

[4] http://www.dtic.upf.edu/ ocelma/MusicRecommendationDataset/lastfm-1K.html

[5] http://recsys.deib.polimi.it/?page_id=54

30music dataset is a newly released large-scale dataset. It contains 45K users, 5.6 million tracks, 31 million play events, and 2.7 million user play sessions.

For both datasets, we (1) only keep the listening sessions with no less than 10 songs, (2) exclude the users only listened to less than 10 songs, and (3) exclude the songs which have been played by less than 10 users. The final number of users, songs, listening sessions, and the sparsity of the two datasets are reported in Table 1. As there is no explicit user-song ratings available, we converted the playcounts to ratings using the method described in [Choi *et al.*, 2012] for the rating prediction task. The user-song playcount matrix was converted to user-song rating matrix with rating in $[1, 5]$.

Evaluation metrics For the rating prediction task, we evaluate accuracy by mean absolute error (**MAE**) and root-mean-square error (**RMSE**). For the top-n recommendation, we used three ranking-based metrics: precision (**Pre@n**), mean average precision (**MAP@n**), and truncated normalized discounted cumulative gain (**NDCG@n**). The ranked lists returned are assessed with the ground-truth songs that user actually played. Pre@n measures the recommendation accuracy - ratio of the top n results are presented in the ground truth. MAP and NDCG evaluate the quality of the ranking list. We truncate the ranking list at 10 for all the metrics.

Baselines We compare our proposed algorithm to the following baselines.

- **Most Popular (MP)**: This baseline recommends items according to their popularity.
- **ItemKNN**: Item-based k-nearest neighbor prediction using Pearson correlation.
- **BMF** [Koren *et al.*, 2009]: this method is a standard in recommender systems. The objective function of this method is described in Eq. 3.
- **WRMF** [Hu *et al.*, 2008]: this is a MF method and a strong competitor for recommendation in implicit feedback datasets [He *et al.*, 2016; He and McAuley, 2016].

Notice that BMF and WRMF are designed for rating prediction and top-n recommendation, respectively. As our main purpose is to examine the potential of using play sequence information in music recommendation, we mainly compared our algorithm with BMF and WRMF for rating prediction and top-n recommendation, respectively. It is worth mentioning that we also compared to the user2vec-based recommendation method [Wang *et al.*, 2016a; Ozsoy, 2016] in comparisons. This method exploits the play sequences and doc2vec method [Le and Mikolov, 2014] to learn the latent vectors of users and items. Then the nearest items in the feature space are recommended to users. Because the results of this method are very poor, we did not present its results in this paper.

Parameter setting In the song similarity computation, we set the size of feature vector to 100 in skip-gram negative sampling, and set the window size to 5. We tested the influence of window size on the final performance in $\{3, 5, 7\}$, and found that set the size to 5 could obtain the best performance. The influence of the number of factors d in matrix factorization methods, the trade-off parameter α, and k in the proposed method are studied and analyzed in experiments.

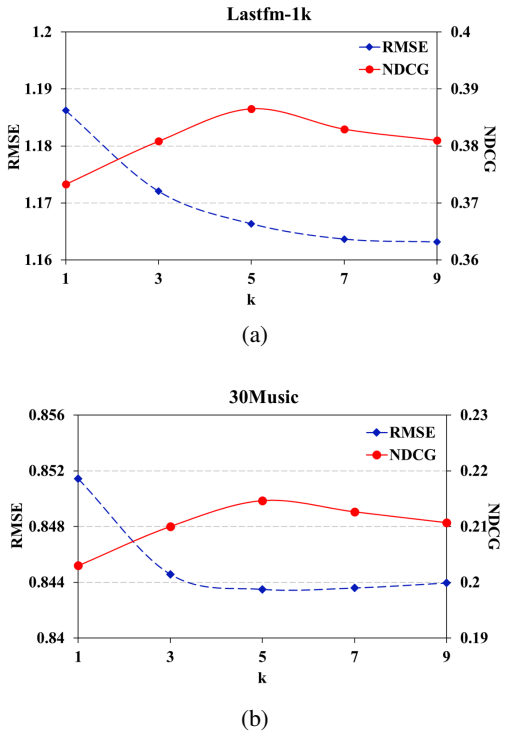


Figure 1: Performance across k

5 Experimental Results

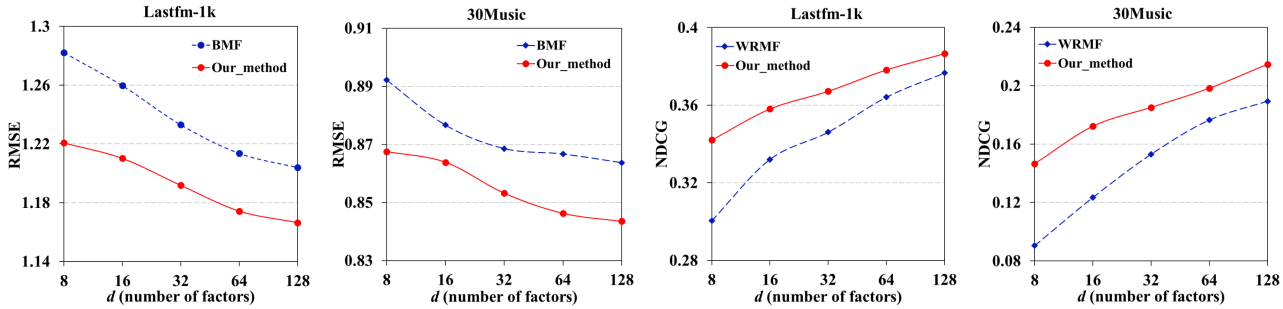
In this section, we report and analyze the experiment results. We first examine the effects of parameters in our algorithm, and then compare the algorithm with other competitors. The reported results are the average values gained over 5-fold cross validation.

5.1 Effects of parameters

In this section, we analyze the influence of two parameters in our algorithm: the number of nearest songs (i.e., k) and the trade-off parameter (i.e., α).

Influence of k Fig. 1 illustrates recommendation accuracy when k is set to different values. We only show the results of RMSE and NDCG, since the same trends can be observed over other metrics. The best performance is achieved when k is 5. When $k = 1$, the regularization effects are not significant. When the similarities of more song-to-song pairs are considered, the regularization takes more effects in the latent feature learning. However, it seems that the regularization becomes too strong when $k > 7$, and thus continuously increasing k will start to cause performance degradation.

Influence of α The trade-off parameter α in the proposed algorithm regularizes the influence of song's similarity in latent feature learning. We evaluate the effects of α on the final performance, and tune it in the range of $[0.1, 1.0]$ with interval 0.1. The results show that the change of α has very small effect on the final performance. For example, the variations of MAE and RMSE are less than 0.001 when the value of α varies in $[0.5, 1.0]$, indicating that our method is fairly robust with respect to α.



(a) RMSE vs. d (b) RMSE vs. d (c) NDCG vs. d (d) NDCG vs. d

Figure 2: Recommendation performance with different number of factors (d) in matrix factorization.

Table 2: Comparisons on rating predication task.

Dataset	Lastfm-1k		30Music	
Metric	MAE	RMSE	MAE	RMSE
MP	1.042	1.322	0.966	1.242
ItemKNN	0.882	1.240	0.724	1.054
BMF	0.887	1.204	0.579	0.864
Our method	**0.876***	**1.166***	**0.573**	**0.843***

Table 3: Comparisons on top-n recommendation task.

Dataset	Method	Pre@10	MAP	NDCG
Lastfm-1k	MP	0.208	0.143	0.138
	ItemKNN	0.281	0.191	0.300
	WRMF	0.344	0.238	0.364
	Our method	**0.377***	**0.264***	**0.386***
30Music	MP	0.002	0.001	0.002
	ItemKNN	0.133	0.114	0.143
	WRMF	0.174	0.159	0.189
	Our method	**0.203***	**0.173***	**0.215***

5.2 Performance comparisons with baselines

In this section, we report the results of our algorithm compared to the baselines. Unless otherwise specified, the results are obtained based on the parameter settings: $d = 128$, $\alpha = 0.5$, and $k = 5$, where d is the number of factors in matrix factorization.

Overall performance Table 2 and Table 3 show results on rating prediction and top-n recommendation, respectively. The symbol (*) after a numeric value denotes significant differences ($p < 0.05$, a two-tailed paired t-test) with the corresponding second best measurement. The proposed method outperforms baselines on both tasks across different datasets. It is evident that the improvement of performance over BMF and WRMF are consistently significant. It demonstrates that (1) song2vec can effectively capture the song's similarity based on large-scale listening sequence data; and (2) the integration of MPS in MF can substantially improve the music recommendation performance in terms of various metrics.

Performance vs. Number of Factors Fig. 2 illustrates recommendation accuracy with varying number of factors d. We would like to highlight the contribution of considering the song2vec in MF. Thus in this figure, we only compare our method to MF and WRMF in the two tasks, respectively. Note that MP and ItemKNN do not have the parameter of latent factors. Their performance will be horizontal lines in the figure. Thus we have not shown their performance. We show the evaluation by RMSE in rating prediction and NDCG in top-n recommendation, since the same or similar trends can be observed with other metrics. The experiments suggest that our method consistently outperforms BMF and WRMF on the two tasks, respectively. It demonstrates the effectiveness of our method and the potential of music play sequence information. More number of latent factors leads to better representation capability and more accurate prediction. Thus, with the increasing of d, the performance of all the methods is improved. In the meantime, it also increases the time complexity and the risk of over-fitting. It is clear that our method could achieve much better performance over its counterpart when d is relatively small. It demonstrates that with the consideration of MPS in MF, better performance can be achieved with a small number of latent factors, which is a desirable property, especially for large datasets with millions of users and billions of interactions.

6 Conclusion

Users leave a lot of listening behavior data online when interacting with social music streaming websites. Large scale of play sequences not only record user's music preferences but also encode song's characteristics. In this paper, we proposed to exploit the music play sequence information in matrix factorization to improve the performance of music recommendation. Inspired by the word2vec techniques, a song2vec method is developed to capture the similarity between songs based on their played sequences. Based on the estimated similarity, a k-nearest song regularized matrix factorization is proposed. Comprehensive experiments on two public datasets demonstrate the effectiveness of the proposed method, indicating the potential of exploiting play sequence information in music recommendation.

Acknowledgments

This research is supported by the National Research Foundation, Prime Ministers Office, Singapore under its International Research Centre in Singapore Funding Initiative.

References

[Benzi *et al.*, 2016] Kirell Benzi, Vassiiis Kalofolias, Xavier Bresson, and Pierre Vandergheynst. Song recommendation with non-negative matrix factorization and graph total variation. In *ICASSP*, 2016.

[Bonnin and Jannach, 2015] Geoffray Bonnin and Dietmar Jannach. Automated generation of music playlists: Survey and experiments. *ACM Comput. Surv.*, 47(2):26, 2015.

[Bu *et al.*, 2010] Jiajun Bu, Shulong Tan, Chun Chen, Can Wang, Hao Wu, Lijun Zhang, and Xiaofei He. Music recommendation by unified hypergraph: combining social media information and music content. In *ACM MM*, 2010.

[Celma Herrada, 2009] Òscar Celma Herrada. Music recommendation and discovery in the long tail. *PhD Thesis*, 2009.

[Cheng and Shen, 2014] Zhiyong Cheng and Jialie Shen. Just-for-me: An adaptive personalization system for location-aware social music recommendation. In *ACM ICMR*, 2014.

[Cheng and Shen, 2016] Zhiyong Cheng and Jialie Shen. On effective location-aware music recommendation. *ACM Trans. Inf. Syst.*, 34(2):13, 2016.

[Cheng and Tang, 2016] Rui Cheng and Boyang Tang. A music recommendation system based on acoustic features and user personalities. In *PAKDD*, 2016.

[Cheng *et al.*, 2016] Zhiyong Cheng, Jialie Shen, and Steven CH Hoi. On effective personalized music retrieval by exploring online user behaviors. In *ACM SIGIR*, 2016.

[Choi *et al.*, 2012] Keunho Choi, Donghee Yoo, Gunwoo Kim, and Yongmoo Suh. A hybrid online-product recommendation system: Combining implicit rating-based collaborative filtering and sequential pattern analysis. *Electron. Commer. Res. Appl.*, 11(4):309–317, 2012.

[Fang and Si, 2011] Yi Fang and Luo Si. Matrix co-factorization for recommendation with rich side information and implicit feedback. In *HetRec*, 2011.

[Gao *et al.*, 2012] Zan Gao, Anan Liu, Hua Zhang, Guangping Xu, and Yanbing Xue. Human action recognition based on sparse representation induced by l1/l2 regulations. In *ICPR*, 2012.

[Hariri *et al.*, 2012] Negar Hariri, Bamshad Mobasher, and Robin Burke. Context-aware music recommendation based on latent topic sequential patterns. In *ACM RecSys*, 2012.

[He and McAuley, 2016] Ruining He and Julian McAuley. Vbpr: visual bayesian personalized ranking from implicit feedback. In *AAAI*, 2016.

[He *et al.*, 2016] Xiangnan He, Hanwang Zhang, Min-Yen Kan, and Tat-Seng Chua. Fast matrix factorization for online recommendation with implicit feedback. In *ACM SIGIR*, 2016.

[He *et al.*, 2017] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. Neural collaborative filtering. In *ACM WWW*, 2017.

[Hu *et al.*, 2008] Yifan Hu, Yehuda Koren, and Chris Volinsky. Collaborative filtering for implicit feedback datasets. In *IEEE ICDM*, 2008.

[Ji *et al.*, 2015] Ke Ji, Runyuan Sun, Wenhao Shu, and Xiang Li. Next-song recommendation with temporal dynamics. *Knowledge-Based Systems*, 88:134–143, 2015.

[Knees and Schedl, 2013] Peter Knees and Markus Schedl. A survey of music similarity and recommendation from music context data. *ACM Trans. Multimed Comput. Commun. Appl.*, 10(1), 2013.

[Koenigstein *et al.*, 2011] Noam Koenigstein, Gideon Dror, and Yehuda Koren. Yahoo! music recommendations: modeling music ratings with temporal dynamics and item taxonomy. In *ACM RecSys*, 2011.

[Koren *et al.*, 2009] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.

[Le and Mikolov, 2014] Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *ICML*, 2014.

[Mikolov *et al.*, 2013] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, 2013.

[Nie *et al.*, 2015] Weizhi Nie, Anan Liu, Zan Gao, and Yu-Ting Su. Clique-graph matching by preserving global & local structure. In *IEEE CVPR*, 2015.

[Ozsoy, 2016] Makbule Gulcin Ozsoy. From word embeddings to item recommendation. *arXiv:1601.01356*, 2016.

[Rendle *et al.*, 2011] Steffen Rendle, Zeno Gantner, Christoph Freudenthaler, and Lars Schmidt-Thieme. Fast context-aware recommendations with factorization machines. In *ACM SIGIR*, 2011.

[Sahlgren, 2008] Magnus Sahlgren. The distributional hypothesis. *Italian Journal of Linguistics*, 20(1):33–54, 2008.

[Schedl *et al.*, 2014] Markus Schedl, Emilia Gómez, and Julián Urbano. Music information retrieval: Recent developments and applications. *Foundations and Trends in Information Retrieval*, 8(2–3):127–261, 2014.

[Schedl *et al.*, 2015] Markus Schedl, David Hauger, Katayoun Farrahi, and Marko Tkalčič. On the influence of user characteristics on music recommendation algorithms. In *ECIR*, 2015.

[Shen *et al.*, 2010] Jialie Shen, Meng Wang, Shuicheng Yan, HweeHwa Pang, and Xiansheng Hua. Effective music tagging through advanced statistical modeling. In *ACM SIGIR*, 2010.

[Shen *et al.*, 2012] Jialie Shen, HweeHwa Pang, Meng Wang, and Shuicheng Yan. Modeling concept dynamics for large scale music search. In *ACM SIGIR*, 2012.

[Turrin *et al.*, 2015] Roberto Turrin, Massimo Quadrana, Andrea Condorelli, Roberto Pagano, and Paolo Cremonesi. 30music listening and playlists dataset. In *ACM RecSys*, 2015.

[Vasile *et al.*, 2016] Flavian Vasile, Elena Smirnova, and Alexis Conneau. Meta-prod2vec: Product embeddings using side-information for recommendation. *RecSys*, 2016.

[Wang *et al.*, 2015] Meng Wang, Xueliang Liu, and Xindong Wu. Visual classification by ell_1-hypergraph modeling. *IEEE Trans. Knowledge Data Eng.*, 27(9):2564–2574, 2015.

[Wang *et al.*, 2016a] Dongjing Wang, Shuiguang Deng, Xin Zhang, and Guandong Xu. Learning music embedding with metadata for context aware recommendation. In *ACM ICMR*, 2016.

[Wang *et al.*, 2016b] Meng Wang, Weijie Fu, Shijie Hao, Dacheng Tao, and Xindong Wu. Scalable semi-supervised learning by efficient anchor graph regularization. *IEEE Trans. Knowledge Data Eng.*, 28(7):1864–1877, 2016.

[Zhang *et al.*, 2013] Hanwang Zhang, Zhengjun Zha, Yang Yang, Shuicheng Yan, Yue Gao, and Tat-Seng Chua. Attribute-augmented semantic hierarchy: towards bridging semantic gap and intention gap in image retrieval. In *ACM MM*, 2013.

[Zhang *et al.*, 2016] Hanwang Zhang, Fuming Shen, Wei Liu, Xiangnan He, Huanbo Luan, and Tat-Seng Chua. Discrete collaborative filtering. In *ACM SIGIR*, 2016.