Music Playlist Recommendation via Preference Embedding

Chih-Ming Chen[1,2], Chun-Yao Yang[1], Chih-Chun Hsia[1], Yian Chen[3], and Ming-Feng Tsai[1]

[1]Department of Computer Science, National Chengchi University, Taipei 116, Taiwan
[2]Research Center for Information Technology Innovation, Academia Sinica, Taipei 115, Taiwan
[3]Research Center, KKBOX Inc., Taipei 115, Taiwan

ABSTRACT

Music playlists usually contain some particular musical styles or atmospheres in which users would like to be involved. Music streaming services, such as Spotify, Apple Music, and KKBOX, even allow users to edit and listen to playlists online. While there have been some well-known methods that can nicely model the preference between users and songs, little has been done in the literature to recommend music playlists, each of which can be considered as a set of many individual songs, to users. In the light of this, this paper proposes a preference embedding based on a user-song-playlist graph to learn the preference representations of these three entities. After the embedding process, we then use the learned representations to perform the task of playlist recommendation. Experiments conducted on a real-world dataset show that the proposed embedding method outperforms the baseline of popularity; in addition, we also make a comparison with DeepWalk and LINE for the recommendation task, and the results show that the proposed method can stand comparison with the two state-of-the-art graph embedding techniques.

Keywords

Graph Embedding; Music Playlist Recommendations

1. INTRODUCTION

Music streaming services usually provide various ways for users to explore the music they may like, such as creating and sharing user playlists. Recommendation usually plays an important role in the exploration via predicting songs toward users according to their listening logs. In the literature, much has been studied about how to model the preference of users and items for an effective recommender system, such as [2, 4, 5]. However, little has been worked on how to recommend a combined set of items (i.e., playlists) based on user preference logs on individual items (i.e., songs). In light of this, we propose to use the Heterogeneous Preference Embedding (HPE) approach [1] based on a user-song-playlist graph toward the task.

RecSys 2016 Poster Proceedings, September 15-19, 2016, USA, Boston.

The idea behind embedding techniques is to compress the contextual/surrounding information of an object into its vector representation. In the field of natural language processing, the techniques are usually referred to as word embedding for language modeling and feature learning to map words or phrases into a low-dimensional vector space. In social network analysis, the similar idea has also been applied to learn the representations of vertices of a social graph that can keep the graph structure for further tasks such as community detection. Inspired by the idea, the HPE method [1] develops the techniques for music recommendation by learning the representations of user preference over items. Based on the HPE method, this paper proposes a preference graph over three entities (i.e., users, songs, and playlists) for embedding. Figure 1 plots the graph, in which the user preference over songs and playlists can be nicely embedded into the subsequent representations. In the experiments, a real-world music streaming dataset containing 50,000 users, 400,000 songs, and 130,000 playlists is employed to verify the effectiveness of the proposed method. As shown in the results, the network-embedding based methods all outperform the baseline of popularity; in addition, the proposed HPE method can bear comparison with the two state-of-the-art graph embedding techniques, DeepWalk and LINE.

2. METHODOLOGY

Given the users of $U = \{u_1, ..., u_{|U|}\}$, the songs of $S = \{s_1, ..., s_{|S|}\}$, and the playlists of $P = \{p_1, ..., p_{|P|}\}$, the task of playlist recommendation is to predict a set of the playlists matching the user preference. The proposed approach models the preference relationship as follows: each object is treated as an individual vertex v of the graph; then, the model iteratively updates each vertex representation Φ according to its proximity to the sampled vertices in the graph. The update procedure can be summarized as the following process of minimizing the set of sampled target-to-proximity pairs (v_i, v_j) [1]:

$$O = -\sum_{v_j \in \text{proximity}(v_i)} \log p(v_j|\Phi(v_i)) \qquad (1)$$

To conduct the task of playlist recommendation, we construct a preference graph, shown in Figure 1, with the following connections:

1. The *user-song* connection: a user is connected to a song if the user listens to the song more than t times. This connection is mainly to record the user preference over songs.

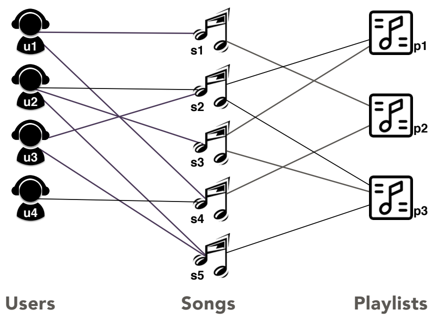

Figure 1: The *User-Song-Playlist* Graph for Preference Embedding.

	Popularity	(DeepWalk, w=2)	(DeepWalk, w=6)	(LINE, 2nd)	(HPE, w=2)	(HPE, w=6)
Precision@5	1.88%	12.85%	***13.19%**	7.14%	10.39%	12.22%
Precision@10	3.67%	12.16%	12.37%	6.66%	10.29%	**12.52%**
Precision@15	5.39%	11.69%	11.88%	6.30%	10.12%	***12.36%**
Precision@20	7.08%	11.32%	11.56%	6.05%	9.99%	***12.35%**

Table 1: Performance of Playlist Recommendation.

2. The *playlist-song* connection: a playlist is connected to a song if the playlist contains the song, which records the relations between songs and playlists.

Figure 1 shows the proposed User-Song-Playlist graph for preference embedding. In the graph, if only the pairs with direct connection are sampled, the objective function of embedding process for a user or a playlist can be expressed as follows:

$$O_v = \begin{cases} -\sum_{s \in Pref(u)} \log p(s|\Phi(u)) & \text{if } v \in U \\ -\sum_{s \in Plist(p)} \log p(s|\Phi(p)) & \text{if } v \in P \end{cases}. \quad (2)$$

In the process, songs can be considered as the connection between users and playlists. Therefore, the learned representation of users and playlists can be matchable via the connection by means of some simple similarity metrics, such as cosine distance and euclidean distance. Moreover, the proposed method can be extended to consider the indirect connections among vertexes, which is similar to the idea of traditional collaborative filtering.

3. EXPERIMENTAL RESULTS

In our experiments, the dataset is collected from the KKBOX music streaming service, consisting of 50,000 users, 400,000 songs, 130,000 playlists with a total of 16,000,000 user-to-song listening logs. We split the dataset as training and testing with the 50-50 ruling. For evaluation, we assume that, if a playlist contains over 70 percent songs that are listened by a user, the playlist is considered as the desired playlist for the user. The evaluation metrics we used in the experiments is the Precision@n, which indicates the hit ratio of the top n recommended playlists.

We compare different network embedding techniques, including HPE [1], DeepWalk [3], LINE [6], and one baseline approach by the song popularity. From Table 1, we can observe that the network-embedding based methods all outperform the baseline. The w refers to window size which is the model parameter. The larger w means adopting wider context information. In the embedding based methods, the learned representations preserve each user previous listening preference. So, the embedding based methods achieve better recommendation quality in terms of precision. In addition, the proposed HPE method obtains the most effective performance of 12.52% in terms of Precision@10.

4. CONCLUSION AND FUTURE WORK

In this paper, we propose a user-song-playlist graph for playlist recommendation by applying the HPE method. In the graph, songs can be considered as the connection between users with playlists, which is useful for the playlist recommendation. With the preference graph, the proposed method can achieve effective recommendation performance. Furthermore, the proposed method can also be extended to include other heterogeneous objects into the graph. So, in our future work, we will attempt to study how to construct other preference graphs with more heterogeneous information to carry out more advanced context-aware recommendations.

5. REFERENCES

[1] C.-M. Chen, M.-F. Tsai, Y.-C. Lin, and Y.-H. Yang. Query-based music recommendations via preference embedding. In *Proc. of ACM RecSys*, 2016.

[2] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.*, 22(1):5–53, 2004.

[3] B. Perozzi, R. Al-Rfou, and S. Skiena. Deepwalk: Online learning of social representations. In *Proc. of ACM SIGKDD*, 2014.

[4] S. Rendle. Factorization machines with libFM. *ACM Trans. Intell. Syst. Technol.*, 3(3):57:1–57:22, May 2012.

[5] J. D. M. Rennie and N. Srebro. Fast maximum margin matrix factorization for collaborative prediction. In *Proc. of ICML*, 2005.

[6] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei. Line: Large-scale information network embedding. In *Proc. of ACM WWW*, 2015.