2019 International Computer Science and Applications Conference (ICSAC 2019)

Context-aware Music Recommendation Based on Word2Vec

Yijie zhou
School of Information Engineering
Communication University of China
Beijing, China
510270279@qq.com

Pei Tian
School of Information Engineering
Communication University of China
Beijing, China
tianpei@263.net

***Abstract*—Musical contextual factors can affect the users' preferences for music greatly, so it is necessary to take the users' current contextual factors into account when making music recommendations to the user. However, we face two critical challenges: how to get the users' contextual information and how to integrate the contextual information into the recommender systems. In this paper, a method is proposed for extracting contextual factors using the word2vec algorithm which is concerned to context-aware information. The neural network model is used to obtain distributed representation of the music pieces. According to the learned distributed representation, the users' long-term and the short-term music preferences can be predicted. Then, a word embedding model is presented, which can incorporate the contextual information into the recommender system with the cosine similarity between the representations of tracks and the users. Finally, the most similar tracks are recommended to the target users.**

Keywords—Word2vec, music recommender system, word embedding, context-aware

I. INTRODUCTION

With the rapid development of the Internet and Web2.0 technology, more and more people have to spend a lot of time to obtain what they are interested in. In order to solve the problem of information overload, the most popular solutions are classification directory and search engines. With the growing scale of the Internet, however, the categories are becoming more and more detailed, which make users impossible to get interested information quickly and accurately. Therefore, the solution of classified directory to overload information is not adapt to the development. For search engines, to some extent, users can still find the information if they have clear targets and keywords. Unfortunately, this method requires users to know key words, and sometimes it is difficult to continually modify keywords to find the desired content. Therefore, in the case that the classification catalogue and the search engine cannot meet the needs of the users, the recommender system of the personalized service comes into being. The core of the system is to establish the connection between the users and the items or information, then infer the users' needs through their historical behaviors. Thus, the emergence of the recommender system greatly enhances the users' Internet experience [3].

The research of the recommender system has been rapidly developed in the past ten years, and has been widely applied in practical systems as well as used in various fields. Such systems are applied for books, movies, hashtags or even friendship recommendations. For example, Amazon and Taobao, in e-commerce, have achieved a great success in recommender systems. In terms of the field of music, Youtube, Pandora and NetEase Cloud Music also make a great success. The utilization of the recommender systems not only improves the efficiency of the users' work, but also brings economic benefits to companies.

In this paper, we focus on the music recommender systems. Music listeners, faced with tens of thousands of music pieces, can easily be overwhelmed. Therefore, it is difficult for listeners to find their interested music. The traditional recommender algorithms, such as collaborative filtering, recommend items by analyzing users' historical behavior, which have been proven useful. Most of the previous music recommendations rarely considered music contextual information. Listeners' music preferences, however, are often influenced by various factors, such as the environment, the emotional state and the activity, etc. For example, when users are running, they tend to listen to the music with full of strength. When they are resting, they prefer to listen to soothing tracks. But there obviously exists a challenge on how to gather contextual information of the systems. According to Adomavicius et al.[1], there are three ways to obtain the contextual information: fully observable context, partially observable context, and unobservable context. The context of music recommender system usually belongs to partially observable or unobservable context. We don't know listeners' activities and current mood directly, which lead to great difficulty to extract the contextual information. In [5], the author proposed a method to get contextual information from the user's short-term playlists. Because the tag in the playlist, such as 'summer', 'hot' and so on, can be clustered as contextual information.

As we all know, users usually listen to different music pieces at different surroundings, different mood and activities. That is to say, users have different music preferences under different contexts. Thus, we can get a conclusion that the current playlist sequences of users can carry the contextual information, such as the users' current emotions and the activities. As a result, the users' current contexts can be extracted from the music sequences the users recently listen to. For example, in NetEase Cloud Music, users can choose the contextual music from different playlists which created with related labels tagged by other listeners. Thus, considering the users' historical and current music records, we can infer users' music preferences under current surroundings and then recommend proper music pieces to their real-time requirements.

In this paper, we propose a context-aware music recommender system which uses the method of Word2Vec to extract the contextual information [7][8][9][10] and infer the users' contextual preferences from their listening records to improve the accuracy of the recommendation. Specifically speaking, the method can be divided into two steps.

1) We propose a song embedding model---the word2vec. We can learn the music vector of each song. Then according to the songs' vector, the average vectors which respectively represent the users' general and current.

2) A context-aware music recommendation method is proposed. Due to the approach, we can compute the most similar songs using the vectors we get in step1, and then recommend the proper songs to meet listeners' requirements.

The remainder of this paper is structured as follows. Sections 2 describes the details of context- aware music recommender systems. In section 3, we introduce the algorithm of word2vec and describe how to incorporate contextual information into music recommender systems. Finally, we generate the recommended songs to the users in section 4.

II. Contextual Music Recommendation

According to Markus Schedl et al., the music contextual information can be divided into two classes: environment-related context, which consists of features that can easily get, such as time, location, weather, etc., and user-related context, which contains users' current mood, activities, etc. In general, user-related contextual information can be derived from environment-related context [1][2].

A. Environment-related Context

As we all know, the user's surroundings, such as weather, temperature, time and other information can affect the user's current emotional state, and then influence the user's music preferences. For example, users tend to listen to different types of music in sunny days and in cold days. Therefore, when recommending songs to users, it is necessary to take the contextual information into account. Environment-related contextual information can be divided into the following categories: location, time, weather, and other factors. The location factors can be zip code, geographic coordinates, etc. Ricci once labelled each song with related place of interest. And then when users toured to the place, the related songs could be recommended to them. Time information can refer to the time of day, and day of week, etc. In different periods, the user's listening habits can be different. Weather factors, such as temperature and humidity, can also affect a user's music preferences. Other factors contain traffic conditions, the levels of noise, etc. All factors can affect user's mood and music preferences. Marius Kaminskas[11] puts forward several dimensions in the contextual information, such as traffic conditions (free road, traffic jam), road type (highway, city, serpentine), landscape (coast line, country side, urban), weather (cloudy, snowing), etc. In different situations, the user marks a score for a given song to obtain the user's preference for music in the certain context.

B. User-related Context Information

Compared with the indirect influence of environmental factors on user music preferences, user-related factors such as mood can directly affect musical preferences. User-related contextual information includes the following categories: users' activities, emotional state, social behavior and cultural context. Users' activities such as running, walking, speed and heart rate can affect the user's music preferences. Netease Cloud Music introduces a running FM that recommends songs by calculating the user's pace. For the emotional state, users' mood can directly affect their musical preferences. For example, the users tend to listen different types music when in a happy mood or a sad mood. Han et al.[12] proposed an emotion transition model to bridge between users' emotional state and the music features. As to social behavior, if there are companions around, it is also necessary to consider the preferences of the companions. For example, the music recommendation of the cafe should consider the preferences of a group. Finally, the cultural background can also affect one's musical habits. Compared with the environment-related context, the user-related context is more difficult to obtain. It cannot be directly accessed by an external device such as mobile, but it can be inferred from the environment-related context by some methods, such as machine learning.

III. Word2vec algorithm

In today's explosive growth of network, text, voice, music and video contains a large amount of information, which can be mined by natural language processing. This technology is now also used in the recommender field. This method can be described as a neural network which is used to analyze the input corpus. The neural model can extract every word in certain text and generate a vector that represents the word. That is what we need. Fortunately, the approach of word2vec can consider the contextual information of the certain word and encode the required word by taking the context words into account. Additionally, listening to music is a typical context-aware behavior. Therefore, the word2vec method can be applied to music recommender systems[4]. According to the foregoing, the users' historical musical playlists contain contextual information. Similar songs may be listened to in a similar environment, and the currently listened songs are able to reflect the user's current emotional state. That is to say, the users' conditions can be inferred by the playlists which they play at the moment. And then we can recommend music pieces to them under their current condition[5][6]. Next, we will further explain the word2vec algorithm.

A. Word2Vec Algorithm

Word2vec is a neural network model, which introduced by Google in 2013 to learn word embedding of natural language processing topics. The method mainly has two models: CBOW model and Skip-gram model. This paper focuses on CBOW model, which is a shallow neural network with a hidden layer only. We can predict the target word by entering it's contextual information. The word vectors can capture many language rules based on the context in the text.

The following step briefly introduces the word2vec algorithm.

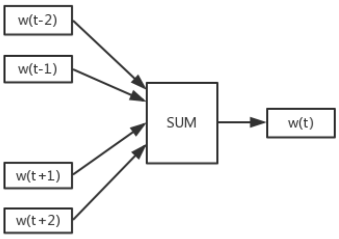


Fig. 1. The model of CBOW

The first layer on the left is the input layer, and we can enter some word vectors encoded with one-hot, as shown in Fig. 1. Suppose we set the number of sliding windows as 5, and the context word vectors in question are respectively $\mathrm{v}(w_{t-2}), \mathrm{v}(w_{t-1}), \mathrm{v}(w_{t+1}), \mathrm{v}(w_{t+2})$. The middle is a hidden layer, which is sum of the contextual word vectors and the output of the hidden layer is a vector, which can be expressed as $X_w = \sum_{i=1, i<|n|}^{5} v(w_{t-i})$. Finally, the third layer is the output layer, which is a Huffman tree. Each leaf node represents a word vector, as well as all leaf nodes of the Huffman tree represent all words in the corpus. The Hoffman tree is constructed based on the word frequency of corpus words. The learning of the word2vec algorithm is to find the probability ofp(w|context(w)), so we use the maximum likelihood function to construct the objective function, which is represented as Equation (1).

$$\mathrm{L} = \sum_{w \in C} \sum_{j=2}^{l} \left\{ \left(1 - d_j^w\right) \cdot \log\left(\sigma\left(X_w^T \theta_{j-1}^w\right)\right) + d_j^w \cdot \log\left(1 - \sigma\left(X_w^T \theta_{j-1}^w\right)\right) \right\} \tag{1}$$

where d_j^w is the jth node in the tree path, and θ_{j-1}^w is a vector of the (j-1)th non-leaf node. For the sake of convenience, we can rewrite Equation (1) as $\mathrm{L(w, j)}$.

$$\mathrm{L(w,j)} = \left(1 - d_j^w\right) \cdot \log\left(\sigma\left(X_w^T \theta_{j-1}^w\right)\right) + d_j^w \cdot \log\left(1 - \sigma\left(X_w^T \theta_{j-1}^w\right)\right) \tag{2}$$

Then we can optimize Equation (2) using the method of SGD. First of all, we take the partial derivative with respect to θ_{j-1}^w.

$$\frac{\partial \mathrm{L(w,j)}}{\partial \theta_{j-1}^w} = \left[1 - d_j^w - \sigma\left(X_w^T \theta_{j-1}^w\right)\right] X_w \tag{3}$$

Next the update formula can be described as Equation (4).

$$\theta_{j-1}^w = \theta_{j-1}^w + \beta\left[1 - d_j^w - \sigma\left(X_w^T \theta_{j-1}^w\right)\right] X_w \tag{4}$$

where β is the learning rate. Similarly, we can get the updated formula of vector X_w as Equation (5).

$$\mathrm{v(w)} = \mathrm{v(w)} + \upbeta \sum_{j=2}^{l} \frac{\partial \mathrm{L(w,j)}}{\partial X_w} \tag{5}$$

According to Equation (5), we can set an iteration number to get the optimal vector v(w).In the next section, we will introduce how to combine word representations with recommender systems.

B. Incorporating Context in Music Recommender Systems

Based on the algorithm of word2vec, we can learn the vectors of user's long-term and current music preferences. Especially, the users' current vectors contain contextual information. In this section, we introduce how to integrate the contextual information into the recommender systems. The method is divided into two parts: obtain the contextual music vectors, according to the users' historical listening records and then recommend the most similar music pieces to users.

Firstly, we need to create music vectors by using CBOW model in word2vec. To acquire the music vector, we build an objective function based on the sequences of playlists the user has listened to and then optimize this function to reach its maximum value. According to the introduction of the word2vec algorithm in the previous section, the objective function can be expressed as Equation (6).

$$\mathrm{L} = \sum_{u \in U} \sum_{m \in M} \sum_{j \le |c|, j \ne 0} \log p(m_i^u |\, m_{i+j}^u) \tag{6}$$

where c is the length of the context window, that is the contextual music sequences $\{m_{i-c}^u : m_{i+c}^u\}$. U is the user set which represent as $\mathrm{U} = \{u_1, u_2, \ldots, u_{|U-1|}, u_{|U|}\}$, and music set is defined as $\mathrm{M} = \{m_1, m_2, \ldots, m_{|M-1|}, m_{|M|}\}$, where $|U|$ and $|M|$ describe respectively as the number of users and music pieces. For each user U, his/her historical music sequences can be described as $H^u = \{m_1^u, m_2^u, \ldots, m_{|H-1|}^u, m_{|H|}^u\}$, which is sorted by the playing time. And $p(m_i^u | m_{i+j}^u)$ is the probability of the target songs m_i^u, which is normalized by softmax function shown in Equation (7).

$$p\left(m_i^u \middle| m_{i+j}^u\right) = \frac{\exp(v_{m_i^u}^T \cdot v'_{m_i^u})}{\sum_{m \in M} \exp(v_{m_i^u}^T \cdot v'_{m_i^u})} \tag{7}$$

where $v_{m_i^u}$ represent all the music pieces of the user and $v'_{m_i^u}$ is the vector of current music in question. Through the softmax function, we can obtain the music objective function by rewriting Equation (6). And then maximizing the function, we can obtain the music representations in space vector.

The vector v^u of a user's long-term music preferences can be inferred from the historical playlist records, which is defined as Equation (8).

$$v^u = \frac{1}{|H^u|} \sum_{m_i^u \in H^u} v_{m_i^u} \tag{8}$$

For the vector v^c of user's current preference, it can be obtained from the average of the music vectors under the context windows. It can be described as Equation (9).

$$v^c = \frac{1}{|c|} \sum_{m_i^u \in R} v_{m_i^u} \tag{9}$$

where c is the length of the context window.

Finally, a context-aware music recommender method is proposed by computing the similarity of the recommended music pieces and the user's total and current music preferences. In this paper, cosine similarity is used to compute the interests, whose formula can be defined as

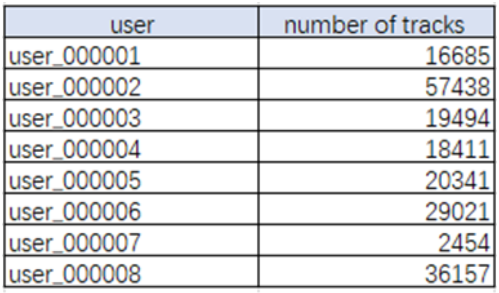
Equation (10).

$$\mathrm{sim} = \frac{\cos\left(v^{u}, v_{m_i^u}\right) + \cos\left(v^{c}, v_{m_i^u}\right)}{2} \tag{10}$$

Lastly, according to the similarity, we exactly recommend the most similar songs to the target user.

IV. Experient

In this paper, the dataset of last.fm is used to achieve the music recommendation. The dataset is composed of user, timestamp, artist, and track, and it contains 1000 users and more than 19 million songs. For the experiment, we selected part of the data for illustration. In the course of the experiment, we used 8 users and 200,000 songs as samples. Fig. 2 shows the numbers of tracks each of the 8 users listened to.



user	number of tracks
user_000001	16685
user_000002	57438
user_000003	19494
user_000004	18411
user_000005	20341
user_000006	29021
user_000007	2454
user_000008	36157

Fig. 2. The statistics of last.fm data set

In the experiment, we used python and genism, an NLP framework, to model the vectors of users and music pieces. For the entire dataset, firstly, we grouped it by the users, and then sorted the tracks of each user due to the timestamps. Then we built a word2vec model and applied it to the entire dataset, optimizing it with the SGD method to extract the word vector for each song. Fig. 3 shows the steps based on the word2vec recommendation. We extracted the word vectors of users and tracks, then compared the similarity, and finally generated the recommendation by ranking in reverse chronological order according to the similarity.

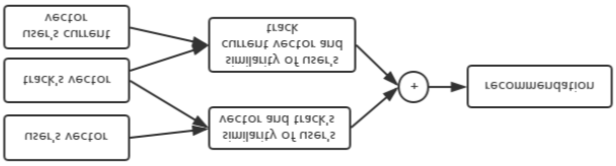
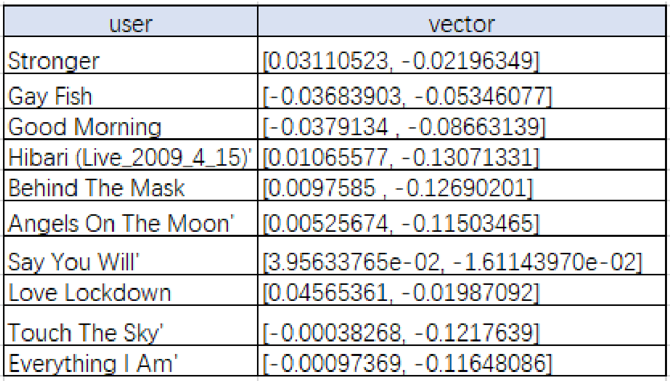

user's current vector
track's vector
user's vector
similarity of user's current vector and track
similarity of user's vector and track's
+
recommendation

Fig. 3. The experimental steps

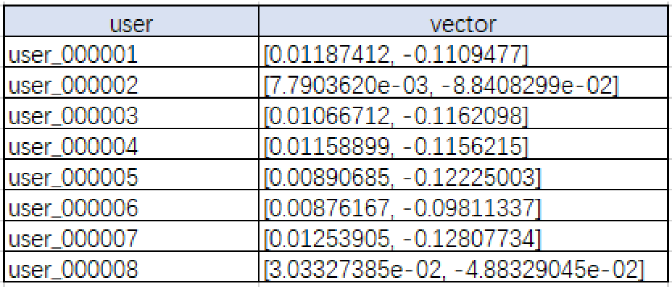
To obtain the tracks' vectors, we built a model of 150-dimensional music vector, selecting 5 adjacent tracks as the context, and ignoring the songs that had been played less than 3 times. Therefore, we set the model parameters as the num_features =150, min_word_count = 3, and context_size= 5, and then learned the vector of each track with these parameters. Fig.4 showed that we randomly selected 10 tracks and calculated the vector of each song (the first 2 dimensions were selected in the 150-dimensional features).



user	vector
Stronger	[0.03110523, -0.02196349]
Gay Fish	[-0.03683903, -0.05346077]
Good Morning	[-0.0379134 , -0.08663139]
Hibari (Live_2009_4_15)'	[0.01065577, -0.13071331]
Behind The Mask	[0.0097585 , -0.12690201]
Angels On The Moon'	[0.00525674, -0.11503465]
Say You Will'	[3.95633765e-02, -1.61143970e-02]
Love Lockdown	[0.04565361, -0.01987092]
Touch The Sky'	[-0.00038268, -0.1217639]
Everything I Am'	[-0.00097369, -0.11648086]

Fig. 4. The vector of 10 tracks

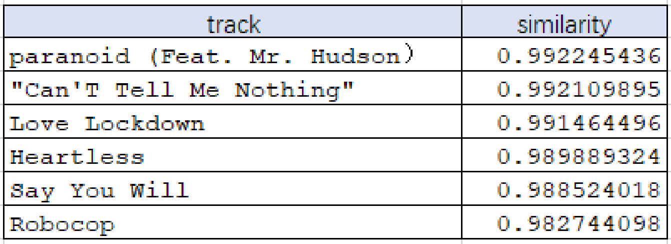
Then, according to the playlists of each user, we acquired the vector of each user's total music preference from his/her historical records. Fig. 5 shows the user's representation.



user	vector
user_000001	[0.01187412, -0.1109477]
user_000002	[7.7903620e-03, -8.8408299e-02]
user_000003	[0.01066712, -0.1162098]
user_000004	[0.01158899, -0.1156215]
user_000005	[0.00890685, -0.12225003]
user_000006	[0.00876167, -0.09811337]
user_000007	[0.01253905, -0.12807734]
user_000008	[3.03327385e-02, -4.88329045e-02]

Fig. 5. The representation of users

What's more, to obtain the vector of user's current music preference, we should take 5-10 tracks in order that the user currently listened to and computed his/her average representation as the user's current vector. Finally, according to Equation (11), we recommended the most similar tracks to the target user.



track	similarity
paranoid (Feat. Mr. Hudson)	0.992245436
"Can'T Tell Me Nothing"	0.992109895
Love Lockdown	0.991464496
Heartless	0.989889324
Say You Will	0.988524018
Robocop	0.982744098

Fig. 6. Tracks recommended to the user_000001 and their similarity

As shown in Fig. 6, six tracks were recommended to the user_000001 and the similarity of each song to the user were obtained according to the above algorithm.

V. Conclusion

This paper proposed a method of music recommendation, which can learn the music vector representations from users' historical music listening records which contain contextual information. And then according to track vectors, we can infer the total and current music preferences which are represented by the vectors as well, then compare the similarity between the tracks and users. Finally, we recommend the most similar tracks to target users. In this paper, we only recommend the tracks to a single user. The following research will combine word2vec with a user-based collaborative filtering model to enhance the accuracy and diversity of the recommendation.

REFERENCES

[1] Adomavicius, G., & Tuzhilin, A. (2011) Context-aware recommender systems. In Recommender systems handbook (pp. 217–253). Springer.

[2] Celma, O. (2010). Music recommendation. In Music recommendation and discovery (pp. 43–85). Berlin, Heidelberg: Springer.

[3] Smith, B., & Linden, G. (2017). Two Decades of Recommender Systems at Amazon.com. IEEE Internet Computing, 21(3), 12–18.

[4] Wang, D., Deng, S., Liu, S., & Xu, G. (2016). Improving Music Recommendation Using Distributed Representation. Proceedings of the 25th International Conference Companion on World Wide Web - WWW '16 Companion.

[5] Pichl, M., Zangerle, E., & Specht, G. (2015). Towards a Context-Aware Music Recommendation Approach: What is Hidden in the Playlist Name? 2015 IEEE International Conference on Data Mining Workshop (ICDMW).

[6] Hariri, N., Mobasher, B., & Burke, R. (2012). Context-aware music recommendation based on latenttopic sequential patterns. In:Proceedings of the Sixth ACM Conference on Recommender Systems - RecSys '12.

[7] Bengio, Y., Courville, A., & Vincent, P. (2013). Representation learning: A review and new perspectives. IEEE Transactions on Pattern Analysis and Machine Intelligence, 35(8), 1798–1828.

[8] Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.

[9] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In Advances in neural information processing systems (pp. 3111-3119).

[10] Bengio, Y., Courville, A., & Vincent, P. (2013). Representation learning: A review and new perspectives. IEEE Transactions on Pattern Analysis and Machine Intelligence, 35(8), 1798–1828.

[11] Baltrunas L, Kaminskas M, Ludwig B, et al. Incarmusic: Context-aware music recommendations in a car[C]//International Conference on Electronic Commerce and Web Technologies. Springer, Berlin, Heidelberg, 2011: 89-100.

[12] Han B J, Rho S, Jun S, et al. Music emotion classification and context-based music recommendation[J]. Multimedia Tools and Applications, 2010, 47(3): 433-460.