DEIM Forum 2019 F8-5

Personalized Music Recommendation with Triplet Network

Haoting Liang*†, Donghuo Zeng*‡, Yi Yu‡, Keizo Oyama‡
National Institute of Informatics,SOKENDAI, Tokyo,Japan
†S8halian@stud.uni-saarland.de, ‡{zengdonghuo, yiyu, oyama}@nii.ac.jp

***Abstract*—With many online music services emerged in recent years, effective music recommendation systems are desirable. Some common problems in recommendation system like feature representations, distance measure and cold start problems are also challenges for music recommendation. In this paper, we propose a triplet neural network, exploiting both positive and negative samples to learn the representation and distance measure between users and items, to solve the recommendation task.**

***Index Terms*—Personalized Recommendation system, triplet neural network**

I. Introduction

As summarized by [1, 16–18], current popular recommendation algorithms, like collaborative filtering-based recommendation and content-base recommendation, have achieved great success in the past few years but still have their own drawbacks. Collaborative filtering[7] relies on users history and their rating on items, which require too much human efforts and usually lead to popularity bias, cold start and sparsity matrix problem. Content-based methods[6] relies on the measurement of similarity between items, which solves some of the problems in collaborative filter. However, the performance of content-based methods depends on the item features and distance measure, which needs very carefully design. Both of these methods consider only users information or items information. But intuitively there should be some relationship between users and items.

Based on the above pros and cons, we propose a cross-modal music recommendation method. Instead of learning the correlation between two modalities directly [11–15] , we exploit both users preference and item features at the same time to learn their effective representation for the recommendation task and their relationship directly. What's more, in most recommendation systems, negative feedback, which means users dislike that items, are ignored. In our work, we use not only positive feedback but also negative feedback together to embrace more information of the users preference.

Inspired by [2] , we study a three branches network called triplet network for the music recommendation task. One sub-network is for user preference and the other two sub-networks with shared parameters are for positive items and negative items respectively. These three sub-networks will map user

*The first two authors have the same contribution to this work. Haoting was involved in this project during her internship in NationalInstitute of Informatics (NII), Tokyo. preference and items to the same latent semantic space. Therefore we can measure the similarity between users and items in this common latent space. The final objective is that the distance between user preference and positive items should be closer than that between user preference and negative items. By optimizing the network according to this objective, we want to learn the mapping sub-network for both users and items, as well as the distance measure function between user preference and items.

II. Related Work

A. Siamese Network

Siamese network [5] is a neural network with two branches. It takes 2 input from the same modal and output the distance or similarity between this two objects. In [10], a siamese CNN network is used to predict the hit songs before they become popular in the market. This siamese CNN network use pairwise ranking loss to learn the audio ranking loss among the songs.

B. Triplet Network

Triplet network are first proposed in [4] inspired by Siamese network [5]. In that work, triplet is used in a image classification task. Triplet network consists of 3 feed forward network with shared parameters. This kind of network is fed with 3 inputs and output 2 values. We can denote these 3 inputs as x, x_+ and x_- where x and x_+ are from the same class while x and x_- are from the different class. So the 2 outputs of the triplet is the pair of distances between x_+ and x_- against the reference x. Finally the two distance will be fed into a comparator to see which one is smaller and thus it is a binary classification problem.

III. Problem Formulation

A music recommendation system consists of three main components: users modelling, items modelling and user-item matching algorithm. These are the three main concerns of our system design. Users modelling aims at modelling the preference of users, which can be related to many different areas, such as users' gender, lifestyle, and listening history. In our system, we use tags labelled by users to represent users' personalized information. Because users can tag whatever they want, tags can cover many areas and reflect their preference. Items modelling is used to represent the music. In [3], features

* The first two authors have the same contribution to this work. Haoting was involved in this project during her internship in NationalInstitute of Informatics (NII), Tokyo.

used for modelling the items are classified into three types: editorial data, cultural data and acoustic data. In our work, we choose acoustic data as our music features, which is the most direct representation. User-item matching algorithm is the most important component in recommendation system. Our method will be explained in detail below.

Since we have to match features from two modalities: user tags and acoustic features, what we want to solve is a cross-modal retrieval problem. In cross-modal learning, one typical method is that separate networks are used for capturing the high-level embedding of information from different modalities and then maximize the correlation between paired examples. In our work, we would like to employ a similar idea to find the relationship between users and the items they like or dislike.

The main idea of our system is to map both music features and user tags to a common feature space, and calculate their distance in the common space. The input is a triplet tuple: (*user preference, positive item and negative items*). Assuming we have n user samples, then the input data should be

$$(U_t, I_t^+, I_t^-), t = 1, ...n$$

We define $\pi()$ as the mapping function for users and denote the embedding of users in the common space as $\mathcal{E}_u$

$$\pi : U \to \mathcal{E}_u$$

Similarly, we define $\phi()$ as the mapping function for items and denote the embedding of items in the common space as $\mathcal{E}_i$

$$\phi : I \to \mathcal{E}_i$$

The network exploits both positive and negative items with the objective that in the common space

$$\mathcal{D}(\mathcal{E}_u, \mathcal{E}_i^+) < \mathcal{D}(\mathcal{E}_u, \mathcal{E}_i^-)$$

where $\mathcal{D}()$ is a distance function used to measure the distance between users and items in the latent common space. $\mathcal{E}_i^+$ and $\mathcal{E}_i^-$ denote the embedding for positive items and negative items respectively.

To fulfill this learning objective, we can perceive this objective as a binary classification problem - the first distance is more or less than the second distance. So we need to define a label for this binary classification problem now. Let

$$o_{ij}^{U_t} = \mathcal{D}(\pi(U_t), \phi(i)) - \mathcal{D}(\pi(U_t), \phi(j))$$

where i and j stand for two items, and then apply sigmoid function on $o_{ij}^{U_t}$

$$P_{ij}^{U_t} = sigmoid(o_{ij}^{U_t})$$

which means if item i is closer to the user than item j, where $o_{ij}^{U_t}$ is negative, $P_{ij}^{U_t}$ will drop to 0, otherwise it will grow to 1.

Therefore we can consider $P_{ij}^{U_t}$ as labels of a binary classification problem. If item i is the positive item and j is the negative item, the label is 0, otherwise the label is 1. In formal definition, the true label should be

$$\overline{P}_{ij}^{U_t} = \begin{cases} 0 & i = I_t^+, j = I_t^- \\ 1 & i = I_t^-, j = I_t^+ \end{cases}$$

Finally, the problem become a binary classification problem: The input is a paired items(positive and negative) and user preference information. The goal is to learn the mapping function $\pi()$ for items and $\phi()$ for user tags as well as the distance function $\mathcal{D}()$ such that can classify whether the paired data is a pos-neg or a neg-pos pair.

Correspond to this formulation, the binary cross-entropy loss function can be used in our learning problem

$$min_{\pi,\phi,\mathcal{D}}\mathcal{L}(\mathcal{N}) = \sum_t -\overline{P}_{ij}^{U_t} log(P_{ij}^{U_t}) - (1-\overline{P}_{ij}^{U_t}) log(1-P_{ij}^{U_t})$$

IV. Network Architecture

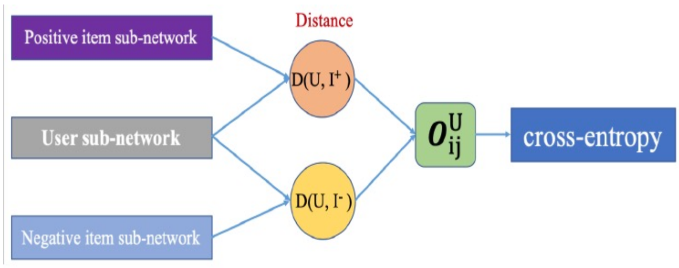

Fig. 1. Simplified network architecture

With the similar idea described in related work, our triplet network take 3 inputs and produce 2 distance output and compare them. Fig. 1 shows a simplified version of the network. One branch is for user tags and two branches are for music features. The upper one is for positive music and the bottom one for negative music. Unlike the triplet described above, in our network, only these two branches share the same parameters, which represent $\phi()$ stated in our formulation while the branch for user information in the middle, which represent $\pi()$ has its own parameters. This is because they are from two different modalities. When training, we use the same network for positive item sub-network and negative item sub-network. Firstly we take an item i and find out the embedding vector of this item. Then, we take the same network without performing any updates on weights or biases and input another item j to find out its embedding again. After obtaining the embedding of the three inputs, we need to find out the distance function $\mathcal{D}$. We use a Euclidean-like distance here. As shown in Fig. 2 The element-wise difference of user and music embedding vectors in the latent common space are calculated first and then followed by element-wise square operation. Then this squared difference vector is fed into one hidden layer to get the final weighted distance. What we want to learn from the training data, is the weights for the three sub-networks and for the fully connected layer for calculating distance.

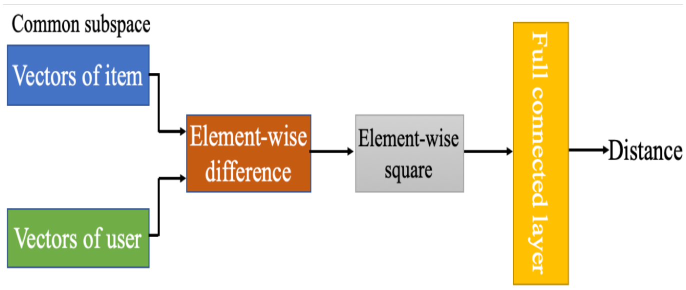

Fig. 2. Calculating distance in common space

A. Recommendation

When making decision for recommendation for a new user, we use the sub-network for items and the sub-network for user preference to get the embedding vector. In the scenario of recommend songs for a new user, after mapping the new user to the latent common space, the distance between the new user to existing items will be calculated, then the nearest items will be returned.

V. Dataset

A. Users Representation

The dataset consists of more than 26000 songs with their user tags crawled from last.fm. Each song has tens of tags labelled by users. In preprocessing, we consider tags of each song as a document and apply LDA topic model[8] over them. Based on the results, we decide to take the top seven topics to represent diverse semantics of music contents. Since these top seven topics representing categories of users' preferences, we take them as users' preferences. One user can be interested in one or more than one topic among the top seven topics and we represent them as a k-hot vector

$$\underbrace{[0,..1,..1,..]}_{7}$$

If dimension i is set to 1 that means the user is interested in topic i. There are 127 possible user groups in total and we train the network with all these possibilities of user preferences. So when recommending songs to a real user, the user will only need to choose some topics he likes and he must be one of the groups.

B. Music Representation

As for the songs, we use 30 second clip for each song in our experiment. The songs are processed by Mel-Frequency Cepstral Coefficients(MFCC) method [9]. The MFCCs of a signal are a small set of features (usually about 10-20) which concisely describe the overall shape of a spectral envelope. We usually used to extract the features that can represent the acoustic information. After processing, finally we get 20 frames for one song and 378-dims for one frame.

VI. Network Setting

The two sub-networks for music with shared parameters are fully connected networks with 4 hidden layers, and the 20*378 item features are flatten to 7560-dim vector before being fed into the network. Because the user vectors is 7-dims, in our design the dimensionality of the common space is 7. The subnetwork for user tags is also a fully connected network, with 4 hidden layers and the output is also 7-dims. We use SGD for optimization. Batch normalization and dropout are applied to the output of every layer. probabilities of dropout are set to 0.2. The network are trained 200 epochs with batch size 256.

VII. Experiment

During training stage, pairs of data, users and paired items features are fed into the network with labels 0 or 1 to learn the mapping and distance function. If the items is positive-negative pair, then the label will be 1 while it would be 0 if the items is negative-positive pair. We have done three experiments in order to evaluate the performance of our network.

The first experiment is that given a vector of user tags, the system should retrieve the nearest song. We evaluate the network's performance by calculating accuracy, which means among all the test samples, whether the returned song has the same topics with that of the user's interest.

$$accuracy = \frac{\text{Number of returned song with same topics}}{\text{number of all test samples}}$$

We compare the precision of triplet network and two branches neural network. The two branches neural network use positive-user pair or negative-user pair to train the network to learn item-user relation and their distance measure. It also return the item closest to the user in the common space when doing recommendation. The sub-networks of the two branches neural network is the same structure as the triplets network. Table I shows the result of these two methods. The performance of triplet is that 57.53% of the test samples are what users interested in while only 48.24% of the returned result of two branches network hit users interest.

TABLE I
COMPARISON BETWEEN DIFFERENT METHODS

Methods	Accuracy
Triplet	57.53%
Twonet	48.24%

Comparing these two methods, training with positive and negative items together actually helps to improve the performance of the network. This is because the objective can consider both situation at the same time.

In the second experiment, we want to investigate the influence of different dataset distributions. We compare the performance on balanced and unbalanced dataset. As mentioned before, we have 127 groups in total, but not all the groups have same number of training data. What's more, there are more positive samples than the negative samples. The ”Unbalanced” dataset denotes the original dataset we use where all groups have the same number of data pairs. In contrast, the ”balanced” dataset denotes the dataset where all 127 groups have the same number of data pairs with under-sampling. The ”1-ton” dataset is obtained from the orginal dataset with oversampling. In previous dataset, each positive item is only use to match one negative item and in this dataset we choose 10 negative items for each positive item. Table II shows the result on different datasets. It shows 1-to-n dataset have the best performance and unbalanced dataset is the worst.

TABLE II
DIFFERENT DATASETS

dataset	Accuracy
Unbalanced	57.53%
Balanced	60.04%
1-to-n balanced	62.89%

The third experiment is that given a song, the mapping sub-network for song will map audios to the common space and then retrieve nearest audios. Unlike the first experiment, where one item sub-network and one user sub-network is used, in this experiment the two sub-network we use are both the item sub-network. Table III shows the result of triplet network and two branches network. Consider both the first experiment and this one, we can find out that both in given user preference to retrieve songs task and given songs to retrieve audios task, the triplet network shows better performance than the two branches network. Another interesting observation is that this experiment in some extent prove that the distance measure function the network learn works well.

TABLE III
DIFFERENT METHODS ON RETRIEVING NEAREST AUDIO

Methods	Accuracy
Triplet	87.42%
Twonet	71.89%

VIII. CONCLUSION

In this work, we propose a triplet neural network taking both music and user information into consideration to do a personalized music recommendation. We use user tags instead of score or ranking as the feedback to the songs, which is more popular among users and can reflect more information from different aspects. From the result, we can come to conclusion that by exploiting both positive and negative items to train the network, a good mapping and distance measure functions can be learned. In the future, more structures of the sub-network and more types of distance measure can be investigated to gain better performance.

REFERENCES

[1] Campbell, M. Elroy, I. James, E. Clune, H. Wendell T, H. Frode, “Music recommendation system and method”

[2] C. Lei, D. Liu, W. Li, Z. Zha, H. Li, “Comparative Deep Learning of Hybrid Representations for Image Recommendations”, CoRR, 2016.

[3] Francois. P, “Knowledge Management and Musical Metadata”, 2005

[4] Elad. H, Nir. A, “Deep Metric Learning Using Triplet Network”, ICLR, 2015

[5] Jane. B, James. W, Leon. B, Isabelle. G, Yann. L “Signature verification using a siamese time delay neural ¨ network”, International Journal of Pattern Recognition and Artificial Intelligence, 1993

[6] Aaron. V, Sander.D, Benjamin. S, “Deep content-based music recommendation”, NIPS, 2013

[7] B. Schafer, D. Frankowski, J. Herlocker, S. Sen, “Collaborative filtering recommender systems”, The Adaptive Web

[8] David. B, Andrew. N, Micheal. J, “Latent Dirichlet Allocation”, NIPS, 2003

[9] Beth. L, “Mel Frequency Cepstral Coefficients for Music Modeling”, ISMIR, 2000

[10] Lang-Chi. Y, Yi-Hsuan. Y, Yun-Ning. Hung, Yi-An. Chen, “Hit Song Prediction for Pop Music by Siamese CNN with Ranking Loss”, ICASSP, 2018

[11] Yu, Yi and Tang, Suhua and Aizawa, Kiyoharu and Aizawa, Akiko, “Category-based deep CCA for fine-grained venue discovery from multimodal data”, IEEE transactions on neural networks and learning systems (2018),1-9.

[12] Yu, Yi and Tang, Suhua and Raposo, Francisco and Chen, Lei, “Deep cross-modal correlation learning for audio and lyrics in music retrieval”, ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM), 2019, 20-21.

[13] Zeng, Donghuo, Yi Yu, and Keizo Oyama, “Audio-Visual Embedding for Cross-Modal Music Video Retrieval through Supervised Deep CCA”, In 2018 IEEE International Symposium on Multimedia (ISM) (pp. 143-150). IEEE.

[14] Yu Y, Tang S, Aizawa K, et al. “Venuenet: Fine-grained venue discovery by deep correlation learning”, 2017 IEEE International Symposium on Multimedia (ISM). IEEE, 2017: 288-291.

[15] Yu, Yi, et al. ”Deep Learning of Human Perception in Audio Event Classification.” 2018 IEEE International Symposium on Multimedia (ISM). IEEE, 2018.

[16] Ricci, Francesco, Lior Rokach, and Bracha Shapira. ”Introduction to recommender systems handbook.” Recommender systems handbook. Springer, Boston, MA, 2011. 1-35.

[17] Alcalde, Vicenç Gaitan, et al. ”Method and system for music recommendation.” U.S. Patent No. 7,081,579. 25 Jul. 2006.

[18] Celma, Oscar. ”Music recommendation.” Music recommendation and discovery. Springer, Berlin, Heidelberg, 2010. 43-85.