

From Bipartite Network to Function: Online-Nearline-Offline Hybrid Music Recommendation

Pei Zhang¹

*Beijing University of Posts and Telecommunications
Beijing, 100876, China
E-mail: zp_163workmail@163.com*

Jie Yan^a; Yiyun Wang^b; Tong Lin^c

*Beijing University of Posts and Telecommunications
Beijing, 100876, China
E-mail: ^ayanjie@bupt.edu.cn; ^bwangyiyun@bupt.edu.cn; ^clintong@bupt.edu.cn*

Yunze Leng

*Beijing University of Posts and Telecommunications
Beijing, 100876, China
E-mail: lengyunze@bupt.edu.cn*

Jinpeng Chen

*Beijing University of Posts and Telecommunications
Beijing, 100876, China
E-mail: jpchen@bupt.edu.cn*

In order to solve problems of stubborn cold start, data sparseness and shortage of each single recommendation algorithm in the music recommendation, Online-Nearline-Offline hybrid MRS is used in this paper. By mixing collaborative filtering and PersonalRank algorithm according to their different characteristics, ONO hybrid model can rationally make up for the shortage reciprocally. Moreover, specific rules for cold start and K-Means for data sparseness are also put forward. For illustration, an entirely new user is utilized to go through the recommendation process both in single algorithm model and ONO hybrid model. The results show that the measured MAE reflecting the satisfaction is the most smooth and the lowest under the condition of hybrid MRS. The ONO hybrid MRS is superior in solving cold star problem, taking the capacity of data in each period into accounts and making up for the single algorithm shortages.

*ISCC2017
16-17 December 2017
Guangzhou, China*

¹This work is supported by Research Innovation Fund for College Students of Beijing University of Posts and Telecommunications, the National Natural Science Foundation of China under Grant No. 61702043 and the Fundamental Research Funds for the Central Universities under Grant No. 500417062.

© Copyright owned by the author(s) under the terms of the Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License (CC BY-NC-ND 4.0). <http://pos.sissa.it/>

1.Introduction

With the development of technology, more and more information of music has been discovered and analyzed in the form of image and text. In order to realize accurate recommendation, the Music Recommendation Systems is put forward as a mechanism for predicting what songs people may like by analyzing different kinds of data from users and songs. The fashionable MRSs like Last.fm (<https://www.last.fm>) based on statistical algorithm and Pandora (<https://www.pandora.com>) concerning about the DNA of music have won considerable profits and users for their outstanding hybrid algorithms [1]. In this paper, an Online-Nearline-Offline hybrid MRS comes up in combination of collaborative filtering and an algorithm named PersonalRank, which does not need to take the nature of music and complex algorithm into account. A dataset named Million Song Dataset (MSD) is used for testing the new system. The paper contains five parts. Related work introduces some current situations of MRS. The modeling part emphasizes the method used in Online-Nearline-Offline hybrid MRS. The Experiment part shows some analysis and results of the experiment. The last part puts forward conclusion and future plans.

2.Related Work

In the previous researches, several works proposed a variety of mature recommendation suitable for music [7, 8]. The collaborative filtering, a method of making automatic predictions about the interests of a user by collecting preferences or taste information from many users, is one of the earliest, most extensive and successful ways to apply to the current recommendation system, was first proposed by Goldberg in the literature [2]. A user-centric evaluation framework for recommendation systems argued a model as developed to evaluate the quality of perceived referrals for users who are more effective and satisfying systems since they've realized that the music promoted by collaborative filtering is often rated higher and popular, and some newly added songs are difficult to be recommended because there is not enough time to pay attention [3]. In order to pursue accurate recommendation, various hybrid recommendations are introduced. The authors mention a strategy that constructing and evaluating weighted hybrid recommendation prototype which contains different technical recommendations [4]. Most of researches have ignored some classical problems such as cold start and data sparseness, which are not the core of recommendation system but essential to affect the performance. In this paper, we suggest some feasible instruments to cope with the crucial problems and introduce a hybrid MRS.

3.Modelling

In this section, we present our MRS based on Online-Nearline-Offline (ONO) hybrid recommendations supplemented by collaborative filtering, content-based recommendation and PersonalRank algorithm.

3.1 Collaborative Filtering and Content-based Recommendation

The collaborative filtering can be user-based (User CF) and item-based (Item CF) mainly to construct parallel similarity in the sample so as to bridge the sample information dimension of being missing, focus on the joint interest between users and items. Both of them on the basis of graph-learning make good use of a set of rules in advance to quickly find a temporary area of nodes. The content-based recommendation, which is famous for highly robust recommended

result for certain music types that are not rich in user behavior, is composed of item representation, profile learning and recommendation generation and concentrates on the independency of each user and excellent transparency.

3.2 PersonalRank

Before talking about its algorithm, we introduce one of its dependencies, called the Weighted Bipartite Network.

3.2.1 Weighted Bipartite Network

There are two kinds of relationships on the social network, one is the user's interest degree to the item and the other is the user's social network relationship. A bipartite graph is a special network that contains two types of nodes, including the user nodes and the item nodes like a tag sticking on songs. Only different classes of nodes can be connected [5].

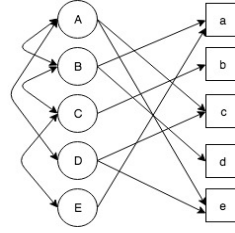


Figure 1: The Bipartite Graph Model in Combination of User's Social Network and Tags.

3.2.2 PersonalRank Algorithm

After establishing the Weighted Bipartite Network (as show in Fig. 1), Personal Rank algorithm is put forward for recommendation.

The Personal Rank algorithm is used to calculate the degree of interest of *user A* to all song tags, that is, to calculate the importance of node *B*, *C*, *a*, *b*, *c* and *d* for *user-A*. The importance is expressed in *PR*.

$$PR(A)=1, \quad PR(B)=0, \dots, PR(d)=0$$

Start from the node that *PR* value is not 0 each time, the probability of walk along the edge is α , the probability of stopping at the current point is $1 - \alpha$, firstly start from *A*, the probability of from *A* to *a* and *c* are both 0.5, then the *PR* value of *a* and *c* are:

$$PR(a)=PR(c)=\alpha \times PR(A) \times 0.5$$

Now points that the *PR* value is not 0 are *A*, *a* and *c*, so start from these points and repeat the process above until the convergence. The following *PR* calculation can be obtained:

$$PR(v) = \begin{cases} \alpha \times \frac{\sum_{v' \in in(v)} PR(v')}{out(v')} & \text{if } v \neq u \\ (1 - \alpha) + \alpha \times \frac{\sum_{v' \in in(v)} PR(v')}{out(v')} & \text{if } v = u \end{cases}$$

U is a node waiting for recommendation and are those nodes that may walk to *v*, which is one of the nodes that would arrive at after next walk. After walking randomly many times, the importance degree of each visited node to *U* will converge to a specific probability, called *PR(v)* as mentioned in the last equation.

3.3 Online-Nearline-Offline Hybrid Recommendation

Netflix, which has hosted the prestigious international recommendation competition, has adopted the hybrid system in the disclosed back-office system architecture.

The Online layer is the user-oriented recommendation service with high-performance and high-availability where a cache system is often designed to handle the problem of popular queries. The online layer will run a simple operation and reliable algorithm to generate a timely result. User CF will be placed in this layer for its quick calculation and dependence on continually changed user's social network. Generally speaking, the song list as recommended by User CF has more credibility since people believe a shrewd music recommendation system will do like that.

The Nearline layer deployed in the server will receive the request from the online layer, and use more complex algorithms to recalculate the results Online layer passes and updates the cache. PersonalRank will be in place for assisting and dealing with the problem of extensive recommendation. Another problem that cannot be ignored is the timely collection of user feedback, and timely is used to adjust the recommended results. Digging the user's feedback to adjust the recommended results gives great assistances. This feedback function is often sustained by the Nearline layer for more convenient collection of the front-end feedback and also ensuring the appropriate processing time.

On the other hand, the Nearline is a bridge between Online and Offline layers. As the Offline layer tends to manage long-term and massive user behavior logs, consume resources and long cycles, the outcome of Offline layer delivered to the line through Nearline layer can often reach the maximum accuracy. Taking into the characteristics of Item CF, it is decided to equip Offline to give more patience and time for better and more precise calculation.

However, before maturity of the system, there are too few users' data and tracks in the system to help accurate analysis, which results in data sparseness occurring apparently. The content-based recommendation is suitable for coping with the initial problem when data are far away enough.

3.4 Cold Start

The most commonly used recommendation methods offer recommendation based on a mass of historical data. Therefore, accurate cold start is indispensable for a remarkable personalized recommendation system. In order to provide more accurate recommendations, users are asked to complete the personal information, such as gender, age, favorite song tag, etc.

According to *Table 1*, vector u_i used to represent the i -th user and vector t_j used to represent the j -th song. By building a regression model, we can obtain a cluster of users called C_n^u and a cluster of song called C_m^t , where $n=1, \dots, N$, $m=1 \dots M$. According to the conditional probability $P(C_n^u | newuser)$, and find the maximum value n^* of P . After n^* is found, the song recommended to the new user will be

$$S = R_{n^*}^T X_{newuser, j}$$

where R_n^T connects the parameters of the model. $R_n^T = [R^0, R_i^T, R_j^T]$, $X_{newuser, j} = [1, u_{newuser}, t_n]$, where $u_{newuser} \in C_n^u$; and $t_n \in C_m^t$.

4. Experiment

In this section, we describe steps of experiment, evaluation standard and results of the experimental evaluation. In order to better evaluate the recommendation, ADMIN, an entirely

new user will go through the following steps to obtain eventual results. ADMIN has gone through the cold start and his initial marks for certain songs are collected.

4.1 Dataset

A MRS will be intelligent based on a mass of data necessarily, we evaluate our algorithm and approach on the Million Song Dataset (MSD) which is a freely available collection of audio features and data for a million of popular music tracks [6].

Moreover, we employ Last.fm's developer API (<https://www.last.fm/api>) to obtain basic information of relatively active users including their favorite tags, loving tracks or albums, and their friends as well. We set up the user's social networks by means of the relationship between each user, which will benefit the analysis and calculation of the similarity of each user later on.

In order to run an accurate recommendation smoothly, we define a set of 1,500 users with strong relation, 15,000 songs and 80 various tags. The whole existing data mentioned above are supposed as the initial and setup data in our MRS.

4.2 Experiment Steps

4.2.1 Quantization

After taking the data from dataset, we need to give each user an initial mark connected with certain role for each song which he/she has heard according to his/her historical song track, which is called quantization. Then we establish a user-song rating matrix with obtained data.

User/Song	Song A	Song B	...	Song N
User A	5	3	...	5
User B	2	0	...	2
User C	3	4	...	4
...
User N	1	2	...	5
ADMIN	5	2	...	0

Table 1: User-Song Rating Matrix

4.2.2 Reduced Data Sparseness

The amount of songs a user enjoys frequently in his/her life is much less than the total amount of songs MSD collects, and it is not possible for an user to rate near half of total songs; so data sparseness comes up which directly breaks the correctness of similarity down. Therefore, we have to reduce data sparseness before calculate the similarity of each user.

We mainly resort to a clustering algorithm called K-Means++ which is the mostly common and fastest clustering algorithm based on K-Means.

4.2.3 Similarity Computation

The similarity is an unavoidable factor when it comes to recommendation. There are two common methods to work out the similarity between users and items. For the purpose of convenient discussion, the items are taken as songs.

Compared with Cosine Similarity, the Adjusted Cosine Similarity fixes up the defect of not taking various standards to rate into account by subtracting the user's average rating for a song.

$$Sim = \frac{\sum_{c \in S_{ij}} (R_{i,c} - R_i)(R_{j,c} - R_j)}{\sqrt{\sum_{c \in S_i} (R_{i,c} - R_i)^2} \sqrt{\sum_{c \in S_j} (R_{j,c} - R_j)^2}}$$

S_{ij} stands for the song set of being commonly rated by *user-i* and *user-j*, S_i or S_j stands for the song set rated by *user-i* or *user-j* respectively. $R_{i,c}$ is the rating of *song-c* given by *user-i*.

4.3 Evaluation Standard

After recommendation, we need to know the feedback from ADMIN in order to know how well the system can be. Moreover, the first recommendation for an entirely new user has less accuracy due to stranger. That is a nice way that offers additional services which have a function that allows user to score song, and the *User-Song rating matrix* (see *Table 1*) can be updated simultaneously.

The predictive accuracy of the scoring forecast is generally calculated by the mean absolute error (MAE). For a *user-u* and *song-i*, $R_{u,i}$ is the actual rating *user-u* gives to *song-i*, and R is the forecast rating recommended algorithm gives.

The MAE makes use of absolute value to calculate the prediction error, which is defined as:

$$MAE = \frac{\sqrt{\sum_{u,i \in T} (R_{u,i} - R)^2}}{T}$$

Studies have shown that if the scoring system is based on an integer (i.e. the user's rating is an integer), then the prediction result will reduce MAE.

4.4 Results

Experimental results are shown in this section, we deliver our results by means of comparison.

User	C	A	E	F	B
Similarity	0.311962	0.315668	0.315709	0.319137	0.320312

Table 2 The New Neighbors of ADMIN

User	C	A	F	E	B
Similarity	0.289312	0.291202	0.301029	0.304361	0.324822

Table 3 The Old Neighbors of ADMIN

Table 2 shows the neighbors of ADMIN as obtained after reducing data sparseness with closer similarity and greater correctness. *Table 3* shows that the neighbors obtained directly without reducing data sparseness. It is not ambiguous to deduce that data sparseness can direct the divergent similarity which is harmful for exact evaluation.

In order to seek the recommendation accuracy and the user's satisfaction, MAE value is considered to be one of crucial judgement. The table listing below shows the comparison of several models applied to ADMIN for nine times (*A*: Only User CF. *B*: Only Item CF. *C*: Only PersonalRank. *D*: Only content-based recommendation. *E*: Online-Nearline-Offline hybrid recommendation. *F*: Online-Nearline-Offline hybrid recommendation without cold start).

Way/Time	1	2	3	4	5	6	7	8
A	0.8293 1	0.9231 2	0.6129 2	0.5711 6	0.8123 3	0.9132 2	0.4091 8	0.7630 1
B	0.6294 7	0.6588 1	0.5791 1	0.6121 9	0.7129 3	0.8211 8	0.5162 3	0.4178 9
C	0.9138 2	0.8388 1	0.9883 4	0.8312 3	0.9822 1	0.7298 3	0.6921 3	0.9023 3
D	0.7291 1	0.8123 1	0.9882 3	0.8591 2	0.8212 4	0.8375 5	0.7291 3	0.7624 1

E	0.3213 3	0.2312 4	0.2344 4	0.3012 3	0.1332 4	0.4923 1	0.4023 8	0.2920 1
F	1.2311 2	1.2902 3	0.9210 2	1.0231 2	0.8213 3	0.9183 1	1.0291 1	0.7821 2

Table 4 Measuring MAE for Six Situations

Without cold start, MRS does not know what songs or tags a strange user would like for lack of basic data, unwarranted guess will bring unexpected low satisfaction and bad feedback. Single recommendation algorithm is far from enough to carry out complex mission and hard to meet the user's various demands because it cannot deal with several defects at the same time, the cooperation is strongly required.

5. Conclusion and Future Work

The research work as presented analyzed the effect of integrating several basic recommendation algorithms with dealing with cold-start obstruction and data sparseness. In the end, we evaluate the MAE of single algorithm in comparison with Online-Nearline-Offline hybrid model, the results show that ONO hybrid model is superior in taking the capacity of data in each period into accounts. They also prove that the cold start will be a vital weakness to a music recommendation system because exact guess depends on a certain number of data to support the sponsor, otherwise, it will be easier to misunderstand the taste of user.

This research effort just begins, so the possible future developments are non-predictable. For the next steps, we will try to analyze the shift of user's habits and his social network, and then rely on analysis to deduce what the user likes during a certain period. The recommendation will be more dynamic and flexible.

References

- [1]Tan S, Bu J, Chen C, Xu B, Wang C, He X. *Using rich social media information for music recommendation via hypergraph model*. ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM). 2011;7S(1):1-22.
- [2]Goldberg, D. and Nichols, D, et al. *Using collaborative filtering to weave an information tapestry*[J]. Comm.of the ACM, 1992, 35(12): 61-70.
- [3]Pearl Pu, Li Chen, and Rong Hu. *A user-centric evaluation framework for recommender systems*. In RecSys '11, pages 157–164, New York, NY, USA, 2011. ACM.
- [4]Simon Franz, Thomas Hornung, Cai-Nicolas Ziegler, Martin Przyjaciel-Zablocki, Alexander Schatzle, and Georg Lausen. *On weighted hybrid track recommendations*. In ICWE, pages 486–489, 2013.
- [5]XinMeng Zhang, ShengYi Jiang. *A personalized recommendation algorithm based on weighted binary graph*. In ISSN 1001-9081, pages 654-655.
- [6]Aa'ron van den Oord, Sander Dieleman, Benjamin Schrauwen. *Deep content-based music recommendation*. Electronics and Information Systems department (ELIS), Ghent University, page 3.
- [7]Schedl M, Knees P, Gouyon F. *New Paths in Music Recommender Systems Research*[C]// the Eleventh ACM Conference. ACM, 2017:392-393.
- [8]Cheng Z, Shen J, Zhu L, et al. *Exploiting Music Play Sequence for Music Recommendation*[C]// Twenty-Sixth International Joint Conference on Artificial Intelligence. 2017:3654-3660.