ISSN 2616-7433 Vol. 3, Issue 1: 125-129, DOI: 10.25236/FSST.2021.030119

Combining User-Genre Preference Information with Neural Collaborative Filtering

Zehao Jiang[1], Yi Yi[2], Xingyu Zhu[3,*]

[1]*School of Computer Science, Harbin Institute of Technology, Harbin, 150001, China*
[2]*School of Mathematical Science, Shanghai Jiao Tong University, Shanghai, 200240, China*
[3]*Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen, 518055, China*
**Corresponding author. Email: 11711724@mail.sustech.edu.cn*

Abstract: *In this paper, the user-genre preference information is introduced into collaborative filtering (CF) to deal with the sparsity which CF suffers from. This work classifies items by genre information, it applies neural collaborative filtering model in each category and combines all models together to get a final prediction. These experiments in the work are conducted on well-known dataset in which hit ratio and Normalized Discounted Cumulative Gain (NDCG) are introduced into the evaluation. It indicates that our model has fast performance growth and good results.*

Keywords: *Genre Information, Collaborative Filtering, Recommendation System, Neural Network*

1. Introduction

Popular video platforms have reached hundreds of millions of video plays and millions of video uploads in each day. The recommendation system filters the result and presents items that are relevant to the users' previous preference.

As the most popular and successful algorithm in recommender system, collaborative filtering (CF) has shown its performance [1]. CF collects users' preference. Then the system finds some users with similar taste, also known as his neighbors, to recommend the user based on what his neighbors like. Sarwar et al. analyzed item-based CF algorithm with several different techniques to computing item-item similarities and found the item-based CF method performed better than user-based algorithm [2]. However, CF does not perform well when deal with sparse data or new-comer.

One way to reduce the sparsity problem of CF is using clustering methods. Gong proposed a method combining CF algorithm with both user clustering and item clustering [3]. In Gong's paper, users are clustered based on users' ratings on items, then the nearest neighbors of a target user can be found. After that, the item clustering collaborative filtering is used to produce the recommendations. This method is proved to be more scalable and accurate than the traditional CF model. Besides, Dhillon et al. introduced the clustering method into a hybrid system, where not only rating but also content-based information was used as features [4].

Another way to improve CF algorithm is matrix factorization, which considers latent factors [5]. Gu et al. designed a model for CF based on graph regularized weighted non-negative matrix factorization [6]. Luo et al. conducted matrix factorization method on large industrial datasets and showed its high accuracy [7]. He et al. designed a model called Neural network-based Collaborative Filtering (NCF), which combines matrix factorization with neural network and the model could express and generalize matrix factorization [8].

In this paper, we revisit a recently-brought collaborative filtering method using neural networks by He et al. and include our idea of utilizing the link of movies' genre information with users [8]. We also show the performance of our model on the famous MovieLens dataset.

2. Data

We use MovieLens 1M dataset, which contains more than 1 million ratings from 6 thousand users on 4 thousand movies released on February, 2003. Each movie is tagged with several genres which can be

ISSN 2616-7433 Vol. 3, Issue 1: 125-129, DOI: 10.25236/FSST.2021.030119

used as content-based information and is rated by several users with integers 1 to 5. We mainly focused on finding the relationship in different movie genres and the relation between users and movie genres with the following features:

1) Rating - represents user's attitude toward a movie. (Table 1)

Table 1: Rating List

UserId	MovieId	Rating
1	1	5
7	6	4
8	14	4
27	2	1

2) Genres - totally 18 movie genres. (Table 2)

Table 2: Movie Genre

MovieId	Movie Name	Genres
1	Lion King (1994)	Animation|Children's|Musical
2	Gone with the Wind (1939)	Drama|Romance|War
3	Roman Holiday (1953)	Comedy|Romance
4	The Shawshank Redemption (1994)	Drama

Data processing

We convert the rating list into a rating matrix. Each row of the matrix represents the user ID, and each column represents the item ID, which means the i^{th} row and the j^{th} column element is the rating of j^{th} item given by the i^{th} user.

3. Methodology

We will first introduce the NCF model and then present our modified genre-user preference information based NCF model.

3.1 NCF model

We summarize the whole process of NCF base on the method suggested by He et al. [8]. NCF (Figure 1) applies neural network into recommender system and it supercharges NCF modeling with non-linearities by combining the linearity of Matrix factorization (MF) and nonlinearity of multi-layer perception (MLP).

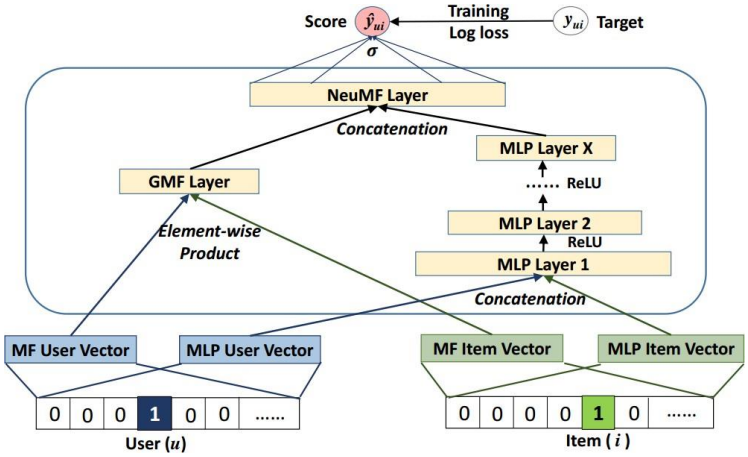


Figure 1: The framework of Neural collaborative filtering [8]

3.1.1 Matrix factorization

Matrix factorization (MF) makes NCF stand out from others. Basically MF finds the relationship between users hidden in a set of latent features. Even if two users have no ratings on the same movie, MF still possibly finds the similarity between them. It mostly resolves the sparsity problem that traditional CF suffers.

ISSN 2616-7433 Vol. 3, Issue 1: 125-129, DOI: 10.25236/FSST.2021.030119

However, without enough information on the new users is still a big problem and the system cannot work properly. Baltrunas et al. proposed a matrix factorization technique for context aware recommendation which model the interaction of contextual factors [9]. However, nothing about movie recommendation was mentioned. We came out with our own approach stated below.

3.2 NCF combined with user-genre preference information

In table 3, Correlation matrix CM(X, Y) represents the correlation coefficients between genre X and Y. User preference matrix P(u, g) stands for the preference coefficients of the user u with the genre g. Rating matrix R(u, m) represents the user u's opinion on movie m.

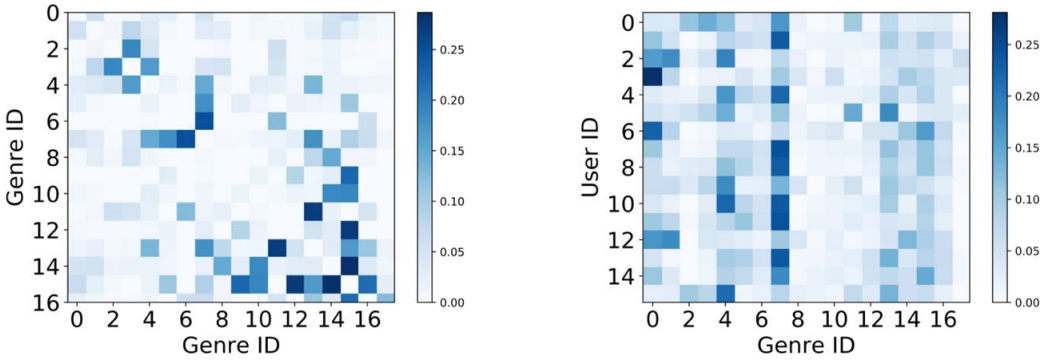
Table 3: Symbols and Meanings

Symbol	Meaning
U	User set
s	Size of set *U*
M	Movie set
q	Size of movie set *M*
G	Genre set
g	Size of genre set *G*
L(m)	Set of all the genres of movie *m*
T(m)	Total number of genres of movie *m*
CM(X,Y)	Correlation matrix
P(u,g)	User preference matrix
R(u,m)	Rating matrix
N(u)	Set of movies user u has rated
MG(m)	Genre list of movie *m*

Our method contains 4 steps:

1) Find the correlation between genres. If 2 genres appear in the same movie, there must be some correlation between them. However, if the movie has too many genre-types, their correlation should be less strong. Our correlation matrix is adapted from the correlation matrix proposed by Choi et al. [10]. Figure 2(a) shows the genre correlation matrix.

2) Calculate users' preference P_i over g kinds of genres. Assume user u_j likes m_i. Then we increase user's preference to all k genres this movie has as well as other genres which has correlation with genres this movie included. After all the movies are calculated, normalize P_1 to P_g and make their sum to 1. The preference matrix is in figure 2(b).



(a) genre correlation matrix (b) user-genre preference matrix

Figure 2: Genre correlation matrix and User-genre preference matrix

3) When using NCF framework to find nearest neighbors, we find nearest neighbors in a specific

ISSN 2616-7433 Vol. 3, Issue 1: 125-129, DOI: 10.25236/FSST.2021.030119

genre. The data of one model is only based on movies that belong to this genre. In each model, one rating data (test data) is sampled from one user's data for testing the model's performance, and all the other rating data (train data) of this user is used for model training. Meanwhile, a list of movies, which lacks rating from this user, is sampled to evaluate the model using the hit ratio (negative data). If test data is in the top 10 list among negative data, it is called a hit.

4) The preference matrix (figure 2(b)) is used to calculate weighted sum of the predictions from different models. Suppose pre_j is the prediction of movie m and user u by the model of j^{th} genre, then the prediction array of eighteen models is Pre = [pre_1, pre_2, ... , pre_{18}]. The formula to calculate the final score of movie m by user u is in Equation 1:

$$user\ u = \sum_{j=1}^{18} P(u,j) \cdot pre_j \tag{1}$$

4. Results

In this section, we evaluate our user-genre preference information based on NCF method for predicting user-item rating and use two protocols to show the performance.

4.1 Evaluation protocol

The definition of Hit Ratio (HR) is that among the m items in the test list, if n items in the top-K recommendation list belong to the test list, then the hit ratio is n. If there are *l* users with *l* different test lists, hit Ratio is calculated in Equation 2:

$$\text{HR} = \frac{\sum_{i=1}^{l} n_i}{\sum_{i=1}^{l} m_i} \tag{2}$$

Normalized Discounted Cumulative Gain (NDCG) is used here for evaluation. When results with high correlation appear in higher rank, the NDCG will be better.

We evaluate our method using HR and NCDG. We check whether the test item shows up in top-K rank for hit or miss and assign NDCG with the location-related score (the lower the rank, the lower the score). In the end, we take the average for both HR and NDCG as the final score of our method.

4.2 Overall performance

In Figure 3, the HR comes to 0.8675 after 20 epochs, which means the accuracy of our model is very high. Both figures show that the model learns faster in first several epochs and grows slowly and stably after that.

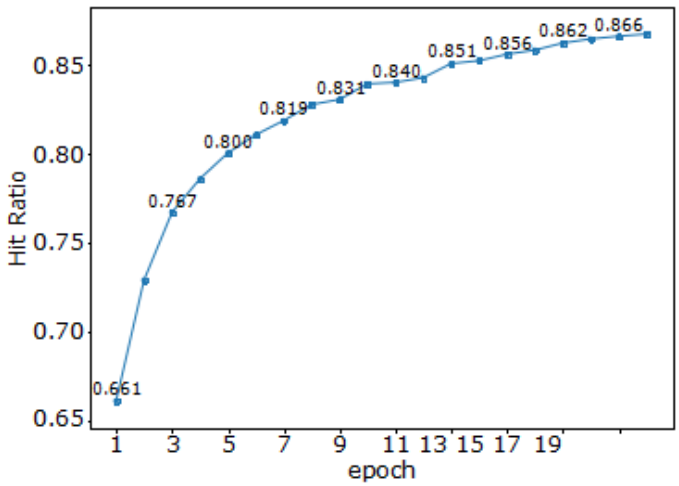


Figure 3: Hit Ratio of our model when K=10

ISSN 2616-7433 Vol. 3, Issue 1: 125-129, DOI: 10.25236/FSST.2021.030119

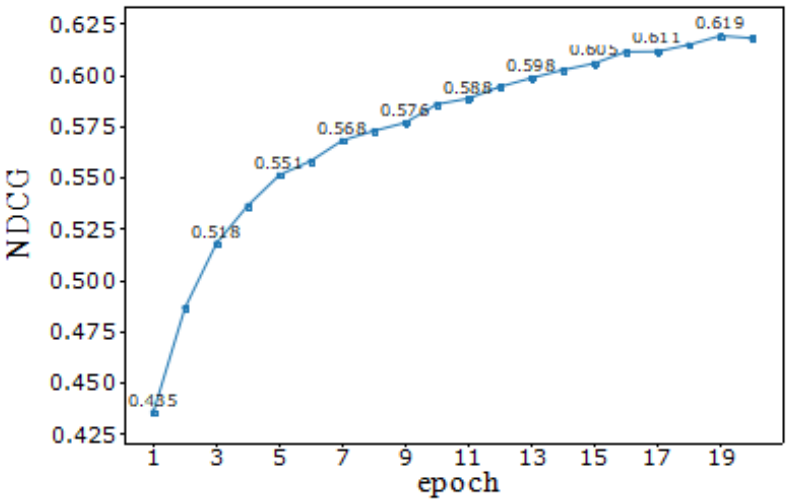


Figure 4: NDCG score of our model when K=10

5. Conclusion

Traditional collaborative filtering method faces major problems like sparsity and cold-start. In this work, we combined user-genre preference information with neural collaborative filtering to solve sparsity and constructed a user-genre preference matrix based on genre correlation and applied this to the GroupLens movie database. Besides, genre correlation based on a new criterion that genre number of each movie is considered.

Future researches can be conducted based on this work. Adding new features including producer or the year-of-release and generating an open API for actual use of this model into movie recommendation are fields that can be dig deeper.

References

[1] Bobadilla, J., Serradilla, F., and Bernal, J. (2010). A new collaborative filtering metric that improves the behavior of recommender systems. Knowledge-Based Systems, 23(6):520 –528.
[2] Sarwar, B., Karypis, G., Konstan, J., and Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. In Proceedings of the 10th international conference on World Wide Web, pages 285–295.
[3] Gong, S. (2010). A collaborative filtering recommendation algorithm based on user clustering and item clustering. JSW, 5(7):745–752.
[4] Dhillon, I. S., Mallela, S., and Modha, D. S. (2003). Information-theoretic co-clustering. In Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining, pages 89–98.
[5] Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. Computer, 42(8), 30-37.
[6] Gu, Q., Zhou, J., and Ding, C. (2010). Collaborative filtering: Weighted nonnegative matrix factorization incorporating user and item graphs. In Proceedings of the 2010 SIAM international conference on data mining, pages 199–210.SIAM
[7] Luo, X., Zhou, M., Xia, Y., and Zhu, Q. (2014). An efficient non-negative matrix-factorization-based approach to collaborative filtering for recommender systems. IEEE Transactions on Industrial Informatics, 10(2):1273–1284.
[8] He, X., Liao, L., Zhang, H., Nie, L., Hu, X., and Chua, T.-S. (2017). Neural collaborative filtering. In Proceedings of the 26th international conference on world wide web, pages 173–182.
[9] Baltrunas, L., Ludwig, B., & Ricci, F. (2011, October). Matrix factorization techniques for context aware recommendation. In Proceedings of the fifth ACM conference on Recommender systems (pp. 301-304).
[10] Choi, S.-M., Ko, S.-K., and Han, Y.-S. (2012). A movie recommendation algorithm based on genre correlations. Expert Systems with Applications, 39(9):8079–8085.