Proceedings of the 3rd International Conference on Signal Processing and Machine Learning
DOI: 10.54254/2755-2721/6/20230710

Collaborative filtering method based on graph neural network

Chaoyi Wang

School of Mechanical, Electrical & Information Engineering, Shandong University, Weihai, 264209, China

201900800126@mail.sdu.edu.cn

Abstract. An essential component of contemporary computer application technology is the recommender system. The collaborative filtering is one of RS's most crucial elements. acquiring knowledge about vector representations or, the model benefits from the combination of the graph neural network and model-based collaborative filtering since it can calculate the high-order connectivity in the item-user graph and perform better overall. This connectivity successfully and explicitly introduces the collaboration signal into the embedding process. Therefore, better embeddings also imply greater performance compared to more established collaborative filtering techniques, such as matrix factorization. The neural graph collaborative filtering (NGCF) algorithm will be primarily introduced in this article. In this paper, the performance of the NGCF algorithm is verified on several data sets, and the experimental results show that there is still room for improvement in the process of practical application. For instance, the NGCF algorithm is not appropriate for processing complicated data, and user cold start is an issue. This study offers a remedy for the difficulties the NGCF algorithm ran into in real-world use. Research on how to enhance the NGCF algorithm considering the issues will continue.

Keywords: collaborative filtering, recommendation, graph neural network.

1. Introduction

An essential component of contemporary computer application technology is the recommender system. The collaborative filtering is one of RS's most crucial elements. acquiring knowledge about vector representations or, the model benefits from the combination of the graph neural network and model-based collaborative filtering since it can calculate the high-order connectivity in the item-user graph and perform better overall. This connectivity successfully and explicitly introduces the collaboration signal into the embedding process. Therefore, better embeddings also imply greater performance compared to more established collaborative filtering techniques, such as matrix factorization. The neural graph collaborative filtering (NGCF) algorithm will be primarily introduced in this article [1]. Additional research and experimentation show certain issues with this strategy.

In general, the learnable models of the collaborative filtering method consist of two similar elements. 1) User and item embeddings, which show how users and items are related [2]. 2) interaction modeling, which may recreate past interactions using user and item embeddings [3]. For instance, in the case of user-item interactions with the inner product, matrix factorization (MF) directly relates the vector and matrix [4]. In addition to the interaction matrix, which needs more training, the matrix factorization, as shown in Figure 1, also comprises embeddings for users and things.

DOI: 10.54254/2755-2721/6/20230710

Despite the high success of their proposal, the authors claim that their methods are insufficient to provide appropriate embeddings for collaborative filtering (CF). The collaborative signal, which is latent in user-item interactions and exposes the behavioural similarity between users or objects, is the principal culprit because the embedding function does not explicitly encode it. Most current methods create the vectors using solely the features of the users or objects, ignoring user-item interactions. Therefore, the approaches must develop extra functions to make up for the fact that the embeddings are insufficient for collecting CF [5].

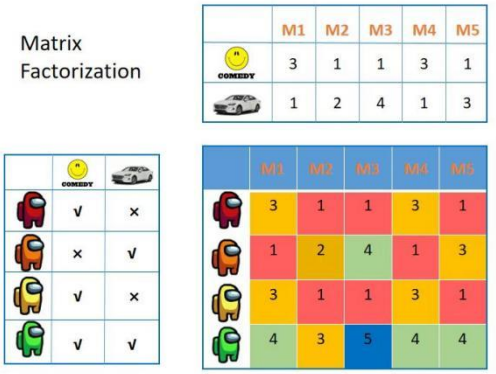

Figure 1. Architecture matrix factorization.

1.1. Matrix factorization

This method contains the core players of the learnable CF models, as seen in Figure 1. The users prefer what sort of goods and the qualities of objects used for describing the difference between items are highlighted in their own embeddings, which are specific to the users and the items. The interaction matrix is in the centre of it, and once we have this matrix, we must train the user and item embeddings to mimic the training set. After training the required embeddings, it will be evident what kinds of items this user prefers to see. However, the input data for this method is quite constrained and that it can only describe the relationship between users and objects in a direct manner; as a result, it is unable to acquire the actual human characteristics. The interaction matrix is always a type of sparse matrix today when we use the CF in real situations, thus the performance of this MF algorithm is not satisfactory, or the algorithm needs to be enhanced to be used in everyday life.

Therefore, the designers of the NGCF method overcome this difficulty by utilizing the high-order connection from several layers of user-item interactions. One of the fundamental ideas of neural graph collaborative filtering is high-order connectivity, a natural approach to convey collaborative signal in the interaction graph structure.

1.2. High-order connectivity

The left side of the link network in Figure 2 illustrates how consumers selects various objects. However, many CF algorithms adopt this graph layout and solely take into account direct connections. The goal is to make some recommendations to User 1, which is marked in this graph by a double circle. The graph is then converted to a tree data pattern. The user1 is represented by the right-hand side of this graph and has several layers of neighbours, the first of which is direct relationships with the objects that user1 used to view through or choose. Users' interactions with these things make up the second layer, and the same concept is at the third level.

The path that leads to u_1 from any cluster with a path length l greater than 1 is known as the high-order connectivity. The semantics of such high-order connectivity transmit signals for collaboration. For instance, the behaviour similarity between u_1 and u_2 is indicated by the path $u_1 \leftarrow i_2 \leftarrow u_2$when both users have specific interactions with i_2. Given that i_4 was previously consumed by user u_2 who are

DOI: 10.54254/2755-2721/6/20230710

comparable to u_1, the path $u_1 \leftarrow i_2 \leftarrow u_2 \leftarrow i_4$ provides more information and suggests that u_1 is more likely to adopt u_1.Additionally, according to the overall perspective of l=3, item i_4 is more likely to be attracted to user u_1 than item i_5 since there are two pathways that connect the $<i_4,u_1>$ while there is only one that does so for the $<i_5,u_1>$.

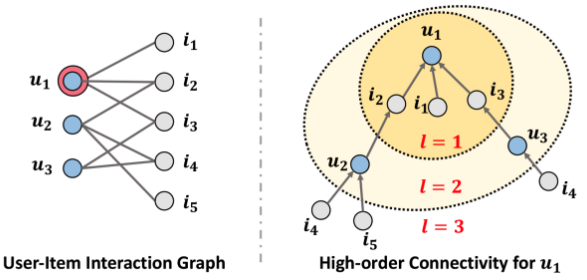

Figure 2. An excellent demonstration of the high-order connectivity and the user-item interaction graph. The user for whom suggestions should be made is node u_1 [1].

The neural map cooperative filtering (NGCF) technique is the major topic of this study. The effectiveness of the NGCF algorithm is tested on several datasets in this research. The findings of the experiment demonstrate that there is still potential for advancement in the NGCF algorithm's practical use. For instance, the NGCF algorithm has the issue of user cold start and is unsuitable for processing complicated data. The author of this work disputes the NGCF method and offers a resolution. This paper proposes to the newly joined user of the system construction or use of user profiles, a specific algorithm is designed to integrate most of the features of information type, use of distributed learning, and encryption technology to realize the collaborative modelling and enhance the validity of AI model, improve the security of the model.

2. Model architecture

2.1. Structure of embedding layers

The author uses an embedding vector $e_u \in \mathbb{R}^d (e_i \in \mathbb{R}^d)$ to reflect a user (an item), where d is size of the embedding. Apparently, the parameter matrix could well be constructed as follows:

$$E = [\underbrace{e_{u1}, \cdots, e_{uN}}_{user's\ embeddings}, \underbrace{e_{i1}, \cdots, e_{iN}}_{item's\ embeddings}] \quad (1)$$

It is important to note that the user and item embeddings start off in the embedding matrix. Additional information is necessary in the different layers of the NGCF architecture, which involves the insertion of more collaborative information to embeddings. As a result, the propagation should not be neglected.

2.2. Structure of embedding propagation layers

The message-passing architecture of GNNs is introduced by the following author in order to collect the complex graph information [6]. The one-layer propagation occurs first, and subsequently several layers are added.

2.2.1. Propagation in first order. It is quite essential that the features of the items can be viewed as a component of the user embedding when a user consumes or selects an item, in our opinion. The two main operations in this section should therefore be: message aggregation and construction [7].

Message Structure Construction: According to a connected user-item interaction, the structure of the message data from i to u is:

$$m_{u \to i} = f(e_i, e_u, p_{ui}) \quad (2)$$

DOI: 10.54254/2755-2721/6/20230710

Apparently, $m_{u\to i}$ is the message information, while $f(\cdot)$ is the message information encoding function. Since e_i and e_u are used as the input, to regulate the decay factor on each edge, the coefficient p_{ui} is attached. The $f(\cdot)$ is constructed as:

$$m_{u\to i} = \frac{1}{\sqrt{|N_u||N_i|}}(W_{1e_i} + W_2(e_i \odot)) \tag{3}$$

Author shows that $W_{1,}W_2 \in \mathbb{R}^{d'\times d}$ are weight matrices which can be effectively trained, while d' is the transformation size. This model considers not only the contribution of e_i, but also the interaction between the e_i and e_u, where $\odot$ where represents the element-wise specific product.

Message Aggregation: In this session, this framework defines the aggregation function as:

$$e_u^{(1)} = LeakyReLU(m_{u\leftarrow u} + \sum_{i\in N_u} m_{u\leftarrow i}) \tag{4}$$

After the first propagation layer, the $e_u^{(1)}$represents the representation of user u. Meanwhile, the $LeakyReLU$ have the ability to assist messages to encode much more varieties of signal information, which can be positive or tiny negative signals. This specific framework also can consider the self-connection of u into the structure: $m_{u\leftarrow u} = W_{1e_u}$, which keeps the original features.

2.2.2. Propagation in high-order. After acquiring each layer connection modeling, we may gain more embedding propagation layers to generate high-order information [8]. When it comes to the l-th step, the algorithm can obtain:

$$e_u^{(l)} = LeakReLU(m_{u\leftarrow u}^{(l)} + \sum_{i\in N_u} m_{u\leftarrow i}^{(l)}) \tag{5}$$

The messages that are being transmitted are defined as follows:

$$\begin{cases} m_{u\leftarrow i}^{(l)} = p_{ui}(W_1^{(l)} e_i^{(l-1)} + W_2^{(l)}(e_i^{(l-1)} \odot e_u^{(l-1)})) \\ m_{u\leftarrow i}^{(l)} = W_1^{(l)} e_u^{(l-1)} \end{cases} \tag{6}$$

where $W_1^{(l)}, W_2^{(l)} \in \mathbb{R}^{d_l\times d_{l-1}}$ are the matrices that are trainable, and d_l means the transformation size. $e_i(l-1)$ is what the item representation generated from at the message-passing part that is calculated before. Meanwhile it can also memories the message from $(l-1)$-hop neighbors.

In this method, the collaboration signal $u_1 \leftarrow i_2 \leftarrow u_2 \leftarrow i_4$ may be recorded. As a result, the various embedding propagation levels contribute to the learning process.

2.2.3. Propagation in matrix form. To acquire a global perspective of embedding propagation, the author gives the layer-wise propagation rule in matrix form [9]:

$$E^{(1)} = LeakyReLU((L+I)E^{(l-1)}W_1^{(l)} + LE^{(l-1)} \odot E^{(l-1)}W_2^{(l)}) \tag{7}$$

It is important that $E_{(l)} \in \mathbb{R}^{(N+M)\times d_{(l)}}$are the information or representation about users and items which can be obtained after l steps of embedding propagation. L represents the Laplacian matrix, which is for the user-item graph. The algorithm is constructed as:

$$L = D^{-\frac{1}{2}}AD^{-\frac{1}{2}} and A = \begin{bmatrix} 0 & R \\ R^T & 0 \end{bmatrix} \tag{8}$$

The user-item interaction matrix is illustrated as the $R \in \mathbb{R}^{N\times M}$. In the calculation session 0 represents all zero matrix. Meanwhile A means the adjacency matrix and D represents the diagonal degree matrix. So, the algorithm represents the t-th diagonal element $D_{tt} = |N_t|$ and the off-diagonal nonzero insertion $L_{ui} = \frac{1}{\sqrt{|N_u||N_i|}}$.

Through these sessions, algorithm can effectively update the representations for all users and things at once by employing the matrix-form propagation rule.

DOI: 10.54254/2755-2721/6/20230710

2.3. Model prediction

Algorithm would experience L layers propagation and this framework gets multiple representations for user u, namely $e_u^{(l)}, \cdots, e_u^{(L)}$. And we do the same operation on items, then we can get the final embeddings.

$$e_u^* = e_u^{(0)} \parallel \cdots \parallel e_u^{(L)}, e_i^* = e_i^{(0)} \parallel \cdots \parallel e_i^{(L)} \tag{9}$$

where $\parallel$ is the concatenation operation. Concatenation has the benefit of being simple, as there are no additional factors to understand.

The framework then uses the internal product to determine the user's choice for the desired item:

$$\hat{y}NGCF(u,i) = e_u^{*T} e_i^* \tag{10}$$

In this section, the algorithm merely makes use of the basic inner product interaction function.

2.4. Optimization

This approach maximizes the pairwise BPR loss to learn the parameters [10]. BPR assumes that witnessed interactions should be given greater prediction values than unobserved ones since they are more indicative of a user's preferences. The following describes the goal function:

$$Loss = \sum_{(u,i,j)\in O} -ln\sigma(\hat{y}_{ui} - \hat{y}_{uj}) + \lambda \parallel \Theta \parallel_2^2 \tag{11}$$

3. Experiments

This section primarily discusses what this paper have learned or have learned through the NGCF model. The first step in the proof is to show how to convert the high-order form, also known as the victor form, of embedding propagation into a matrix form that the code can easily use to train the data. The following are some issues This paper discovered throughout the course of this investigation.

3.1. The prove of the victor form propagation to matrix form

The first step is to insert each layer's parameters such that this paper may obtains the equation as:

$$\begin{aligned} e_u^{(l)} &= LeakyReLU(e_u^{(l-1)} w_1^{(l)} + \frac{1}{\sqrt{|N_u||N_i|}} \sum_{i\in N_u} \left[e_i^{(l-1)} W_1^{(l)} + \left(e_i^{(l-1)} \odot e_u^{(l-1)} \right) W_2^{(l)} \right]) \\ &= LeakyReLU(e_u^{(l-1)} w_1^{(l)} + \frac{1}{\sqrt{|N_u||N_i|}} [\sum_{i\in N_u} e_i^{(l-1)} W_1^{(l)} + \sum_{i\in N_u} \left(e_i^{(l-1)} \odot e_u^{(l-1)} \right) W_2^{(l)}]) \end{aligned} \tag{12}$$

Then this paper changes each layer's representations to the matrices form. Because the p_{ui} is the graph Laplacian norm, author can change the first part of these norm into Laplacian matrix.

$$\begin{aligned} E_u^{(l)} &= LeakyReLU(E^{(l-1)} W_1^{(l)} + \frac{1}{\sqrt{|N_u||N_i|}} \left(AE^{(l-1)} W_1^{(l)} + AE^{(l-1)} \odot E^{(l-1)} W_2^{(l)} \right)) \\ &= LeakyReLU\left(E^{(l-1)} W_1^{(l)} + LAE^{(l-1)} W_1^{(l)} + LE^{(l-1)} \odot E^{(l-1)} W_2^{(l)} \right) \\ &= LeakyReLU((L+I) AE^{(l-1)} W_1^{(l)} + LE^{(l-1)} \odot E^{(l-1)} W_2^{(l)}) \end{aligned} \tag{13}$$

Table 1. The Datasets.

Datasets	Users	Items	Interactions	Density
Gowalla	29,858	40,981	1,027,370	0.00084
Yelp2018	31,668	38,048	1,561,406	0.00130
Amazon-Book	52,643	91,599	2,984,108	0.00062
My Own Dataset	100,858	40,981	4,127,371	0.00099

DOI: 10.54254/2755-2721/6/20230710

Table 2. NGCF test on the amazon-book.

Indicators	The Best Four Numbers			
recall	0.15743	0.26833	0.30589	0.33690
precision	0.04835	0.03439	0.02788	0.02125
hit	0.054320	0.64800	0.70718	0.77587

Table 3. Different performance.

datasets	framework	recall	ndcg
Gowalla	MF	0.1291	0.1109
	GC-MC	0.1395	0.1204
	NGCF	0.1569	0.1327

The Table1 displays all of the datasets I utilized. Additionally, Table 2 shows the NGCF's 2018 Amazon-Book performance. In all the methods I examined, the hit rate indication is usually always the best. The performance of the three distinct frameworks for the Gowalla dataset is displayed in Table 3.

3.2. Some phenomena and solution

3.2.1. Larger or more complex datasets. As seen in Table 1, the GPU server informs me that my data is out of memory when I try my datasets on the NGCF. The datasets do, however, truly exist in our day-to-day lives. The matrix is not sparse enough, hence it might not be applicable in this circumstance. Perhaps the MF's performance at the time was superior to the NGCF's.

In practice or in daily life, NGCF cannot handle problems that are too complicated or overwhelming. In the case of data of a particular size, NGCF ought to be more effective. The program may need to use more and more algorithms to handle as many circumstances as feasible if the developers attempt to use the NGCF in our everyday lives. For instance, it is best to first compute before building a rudimentary model of the data to be processed. We can use the method to determine the sparsity and complexity based on the data model before selecting the best recommendation technique. This type of action can reduce the time cost and improve the performance of the recommendation system. Therefore, it is best to utilize the advised algorithm when it is possible to do so.

3.2.2. Cold start problems. There are a wide variety of cold stat difficulties. 1) Cold Start User refers to a new user with an empty embedding. 2) The term "item cold start" refers to the initial, inactive interaction between new things and users. System Cold Start refers to the difficulty with installing a new system in a new network. As a result, when the author tried to test the algorithm's performance on the user cold start problem using many users, I performed poorly. The precision is about 0.04286 and the recall is about 0.13813, which is significantly worse than the original datasets.

These issues today are fundamentally and inherently characterized by the absence of critical information. These issues can be resolved due to stronger models if users or goods models can obtain additional data or interactions from other areas of the apps or other businesses. For instance, the recommendation system can construct or employ user profiles, which are a type of label used to specifically describe user preferences, for the newly joined users to the system. The system might initially build an interaction matrix or function based on user profiles using this kind of data.

However, not all businesses are willing to give their data, thus recommendation systems require a certain sort of algorithm that integrates most information types in features that do not immediately reveal the facts. With the aim of accomplishing collaborative modeling and enhancing the efficacy of AI models, Federated Learning utilizes Distributed Learning and Encryption Technology. Additionally, this algorithm might obtain user profiles in a manner that ensures data privacy, security, and compliance with the law.

DOI: 10.54254/2755-2721/6/20230710

The primary goal of the recommendation system application should be to use these methods to obtain a better model because they can significantly lessen the cold start issues.

4. Conclusion

In this paper, the neural graph collaborative filtering (NGCF) algorithm is introduced in detail, and the performance of the NGCF algorithm is verified on several data sets. Although the NGCF algorithm solves the problem that the transmission signal of the interactive information between users and goods can't be expressed in the embedded layer, this paper finds that there are still other challenges in the practical application of the NGCF algorithm through a series of experiments. The NGCF algorithm is prone to the problem of insufficient GPU memory when dealing with complex data, so the NGCF algorithm is not suitable for complex data. In addition, by building many users to test the performance of the algorithm on the user's cold start problem, the performance of the algorithm is poor, the precision is about 0.04286 and the recall is about 0.13813. This result is far lower than the performance on the original data set. This paper makes several recommendations based on the identification of the issues, including the creation or use of user profiles for new users, the design of a particular algorithm to incorporate the features of the majority of information types, and the use of distributed learning and encryption technology to realize collaborative modeling, improve the efficacy of AI models, and increase the security of models. Research on how to enhance the NGCF algorithm considering the aforementioned issues will continue.

References

[1] Xiang Wang & Xiangnan He & Meng Wang &Fuli Feng & Tatseng Chua (2019) Neural Graph Collaborative Filtering, SIGIR 2019.

[2] Kula, M. (2015). Metadata embeddings for user and item cold-start recommendations. arXiv preprint arXiv:1507.08439.

[3] Xue, F., He, X., Wang, X., Xu, J., Liu, K., & Hong, R. (2019). Deep item-based collaborative filtering for top-n recommendation. ACM Transactions on Information Systems (TOIS), 37(3), 1-25.

[4] Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. Computer, 42(8), 30-37.

[5] Xiangnan He & Lizi Liao & Hanwang Zhang & Liqiang Nie & Xia Hu & Tat-Seng Chua (2017) Neural Collaborative Filtering in WWW.173-182

[6] Thomas N. Kipf & Max Welling (2017) Semi-Supervised Classification with Graph Convolutional Networks. 15In ICLR

[7] Ding, M., Cheng, X., & Xue, G. (2003, October). Aggregation tree construction in sensor networks. In 2003 IEEE 58th Vehicular Technology Conference. VTC 2003-Fall (IEEE Cat. No. 03CH37484) (Vol. 4, pp. 2168-2172). IEEE.

[8] Wang, G. A., Yang, S., Liu, H., Wang, Z., Yang, Y., Wang, S., ... & Sun, J. (2020). High-order information matters: Learning relation and topology for occluded person re-identification. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 6449-6458).

[9] Montavon, G., Binder, A., Lapuschkin, S., Samek, W., & Müller, K. R. (2019). Layer-wise relevance propagation: an overview. Explainable AI: interpreting, explaining and visualizing deep learning, 193-209.

[10] Rendle, S., Freudenthaler, C., Gantner, Z., & Schmidt-Thieme, L. (2012). BPR: Bayesian personalized ranking from implicit feedback. arXiv preprint arXiv:1205.2618.