Received June 6, 2020, accepted June 19, 2020, date of publication July 1, 2020, date of current version July 21, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3006141

Recommended System: Attentive Neural Collaborative Filtering

YANLI GUO AND ZHONGMIN YAN
Shandong University, Jinan 250100, China

Corresponding author: Yanli Guo (1789213513@qq.com)

This work was supported in part by the National Key Research and Development Program under Grant 2017YFB1400100, in part by NFSC under Grant 61772316, in part by SDNFSC under Grant ZR2018MF014, and in part by the SPKR&DP under Grant 2017CXGC0702.

ABSTRACT In recent years, neural networks have yielded immense success on speech recognition, computer vision and natural language processing. However, the exploration of neural networks on recommender systems has received relatively less scrutiny. In this work, we strive to develop techniques based on neural networks to tackle the key problem in recommendation — collaborative filtering — on the basis of implicit feedback. Although some recent work has employed deep learning for recommendation, they primarily used it to model auxiliary information, such as textual descriptions of items and acoustic features of music. When it comes to model the key factor in collaborative filtering — the interaction between users and item features, they still resorted to matrix factorization and applied an inner product on the latent features of users and items. And the collaboration signal hidden in the user-item interaction is not encoded during the embedding process. Therefore, the resulting embedding may not be sufficient to capture the collaborative filtering effect. By replacing the inner product with a neural architecture that can learn an arbitrary function from data, we present a general method named *ANCF(Attention Neural network Collaborative Filtering)*. ANCF captures collaborative filtering signals and refines the embedding of users and items according to the structure of the graph. By introducing an attention mechanism, the user vector and the item vector are learned on the user-item interaction graph, neighbor interaction information is aggregated to encode, and the embedding is propagated on the user-item interaction graph. This makes it possible to explicitly inject user-item collaboration signals into the embedding process. Extensive experiments conducted on two real world datasets show that ANCF's recall and ndcg have increased by 30% and 35%, so our proposed ANCF method has been significantly improved over the state-of-the-art method. Empirical evidence shows that using deeper layers of neural networks offers better recommendation performance.

INDEX TERMS Recommender systems, information filtering, neural networks, attention mechanism.

I. INTRODUCTION

In order to solve the personalized recommendation problem, artificial intelligence is gradually applied to recommendation systems. The key to personalized recommendation systems is to estimate the likelihood that a user will adopt a product based on historical interactions between the user and clicks. The idea of collaborative filtering is to use past behaviors or perspectives of existing user groups to predict what current users are most likely to like or interested in, and analyze similar preferences of users with similar behaviors to come up with effective suggestions. Common methods are parametric reconstruction of historical interactive users and items, and predicting user preferences based on parameters [3], [9].

In order to predict user preferences from key (and widely available) user behavior data, a large amount of research work has been invested in collaborative filtering (CF) [1], [5], and [9]. Generally speaking, a learnable CF model has two key components: 1) embedding, which transforms users and items into vectorized representations, and 2) interaction modeling, which reconstructs historical interactions based on the embeddings. Many research efforts have been devoted to enhancing these two parts. For example, matrix factorization (MF) directly embeds user / item ID as vectors and

The associate editor coordinating the review of this manuscript and approving it for publication was Gang Li.

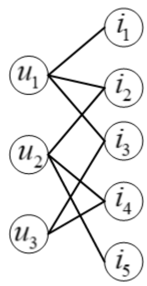


FIGURE 1. User-item interaction graph.

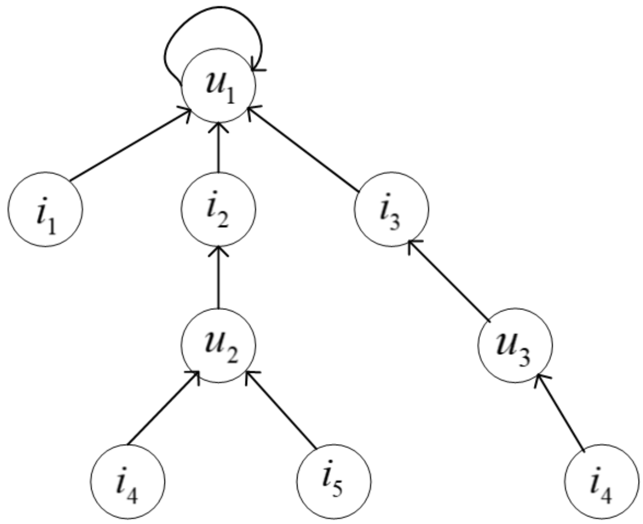


FIGURE 2. Message passing for user1(u_1).

uses inner products to model user-item interactions [12]; translation-based CF model uses Euclidean distance metric as interaction function [14]; collaborative deep learning expands the embedding function by integrating information obtained from the item edges [18].

Although the CF method is effective and versatile, due to the lack of coding of key collaborative signals, this is likely to fail to accurately reveal the similarity of users or items in user-item interactions. More specifically, since existing methods build the embedding function with the descriptive features (e.g., ID and attributes), without considering the user-item interactions in Fig1. It is particularly important to capture the user-item interactions and improve the quality of recommendations. In this work, we introduce the attention mechanism to aggregate the relationship weights between different items of the same user to learn the user and item vector representations, and improve the quality of recommendations.

To summarize, this work makes the following main contributions:

- We highlight the importance of leveraging collaborative signals in the embedding function.
- We propose the *Attention Neural Collaborative Filtering (ANCF)*, which is a new recommendation method based on neural network. This method uses an attention mechanism to assign different weights to different items of the same user, and encodes potential interactive signals through embedded propagation.
- We conduct extensive experiments on two real-world datasets to prove the effectiveness of our *ANCF* method and the prospect of collaborative filtering.

II. RELATED WORK

A. COLLABORATIVE FILTERING

In current recommendation systems, collaborative filtering methods are widely used. Traditional collaborative filtering methods are gradually combined with machine learning. For example, neural networks are used to solve the problem of implicit feedback [9]; neural networks are used for embedded propagation [5]; recommendations based on storage networks [4] for recommendations; conference-based recommendations [15]; topic-tag-based recommendations [16], and one-class recommendation [1].

B. IMPLICIT FEEDBACK

Although early literature on recommendations focused on displaying feedback [13], recently the focus has shifted to implicit data [10]. The user does not interact with the item, which does not mean that the user does not like the item; similarly, the user interacts with the item does not mean that the user likes the item. Collaborative filtering of implicit data is usually expressed as a recommendation problem. Explicitly expressing the implicit interaction data between users and items is more effective and more challenging for recommendation problems [10]. For this reason, the recently literatures [8], [9] focus on using deep learning techniques to enhance interaction capabilities to capture the non-linear interactions between users and items.

C. ATTENTION MECHANISM

The main idea of the attention mechanism is intuitively similar to human visual attention, ‘‘observing important parts‘‘. At present, attention mechanisms have been shown effective in various machine learning tasks such as image / video captioning [6], machine translation [2], and natural language processing [21]. Recently, neural network attention mechanisms have been used in recommendation systems [11]. For example, [2] proposed an attention-based approach to align codec frameworks for machine translation. [7] proposed an attention CNN network. The authors designed two attention mechanisms, namely local attention and global attention, to learn user and item representations better.

III. ATTENTIVE NEURAL COLLABORATIVE FILTERING

We now propose the ANCF model, the architecture of which is illustrated in Fig3. The framework consists of three parts: (1) the embedding layer, which is responsible for the initialization of user embedding and item embedding; (2) the attention propagation layer, which assigns different weights

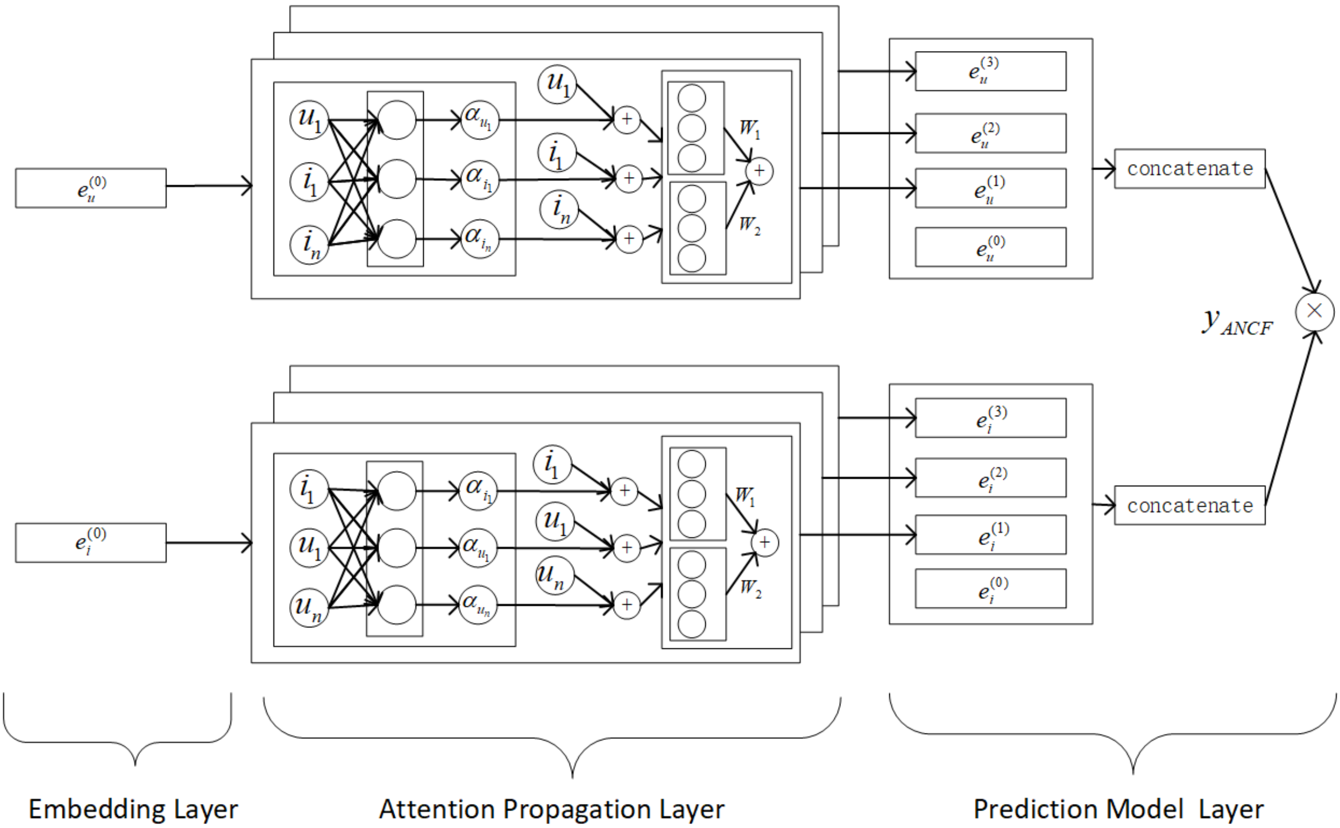


FIGURE 3. ANCF model structure (enhancing user and item representations through attention propagation layers, linking their outputs for final prediction).

to the information passed by different items of the same user by introducing an attention mechanism, and performs message embedding propagation; (3) the prediction model layer, which aggregates the embeddings from different propagation layers, and outputs the degree of user preference for the items.

A. EMBEDDING LAYER

The embedding layer is used to initialize user embedding and item embedding. According to the mainstream recommendation system model [3], [9], an embedding lookup table is described.

Enter the data that the user interacts with the item to establish a parameter matrix, which can be used as an embedded lookup table. We describe a user u (an item i) with an embedding vector $e_u \in R^d (e_i \in R^d)$, where d denotes the embedding size. Initialize users and items to the following format:

$$E = [e_{u_1}, \cdots, e_{u_N}, e_{i_1}, \cdots, e_{i_M}] \tag{1}$$

This embedding table is used as the initial state for user embedding and item embedding.It is optimized in an end-to-end manner. In traditional recommendation models (such as MF), these ID are embedded and fed into the interaction layer (or operator) to obtain prediction scores. Instead, in our ANCF method, we propagate the embedding on user-item interaction graph.

B. ATTENTION PROPAGATION LAYER

Next, a GNN-based messaging architecture [19] is built to obtain collaborative filtering signals along the graph structure and optimize user-item embeddings, as shown in Fig3.

1) MESSAGE CONSTRUCTION

The interaction between a user and an item directly reflects the user's preferences [20]. Similarly, an item (user) interacting with a user (item) can be regarded as a feature of the user (item), and the similarity of two users (item) can be measured. Based on the similarity between users, the attention mechanism can be used to calculate the weight of message passed between the same user and different items during the embedded propagation process. The message during the propagation is divided into two parts: (1) user-item interaction message (2) self-connection message of users, as shown in Fig1 and Figs2.

For a connected user-item pair (u, i) message, we define the message from i to u as:

$$m_{u \leftarrow i} = f(e_u, e_i). \tag{2}$$

where $m_{u \leftarrow i}$ is the message embedding(i.e., the information to be propagated). $f(\cdot)$ is a message encoding function, which takes embeddings e_u and e_i as input.

In this work, we implement $f(\cdot)$ as:

$$m_{u \leftarrow i} = W_1 e_i + W_2(e_i \otimes e_u). \quad (3)$$

where $W_1, W_2 \in R^{d_l \times d_{l-1}}$, are the trainable weight matrices to distill useful information for propagation, we use xavier to randomly initialize the weight matrix W_1 and W_2. We additionally encode the interaction between e_u and e_i into the message being passed via $e_i \otimes e_u$, where $\otimes$ denotes the element-wise product.

2) ATTENTION MECHANISM

In collaborative filtering, it is not accurate enough to predict user preferences using only the attribute characteristics of users and items. It is more reasonable to use user-item interaction signals. For example, in Fig2, items i_4 and i_5 with the same path length of 3 Obviously, users' interest in i_4 is higher than i_5, because there are two paths connected by $< i_4, u_1 >$, $i_4 \rightarrow u_2 \rightarrow i_2 \rightarrow u_1$, $i_4 \rightarrow u_3 \rightarrow i_3 \rightarrow u_1$, However, there is only one $< i_5, u_1 >$, $i_5 \rightarrow u_2 \rightarrow i_2 \rightarrow u_1$. By calculating the user's attention weight for i_4 and i_5, we can conclude that the user's interest in i_4 is higher. Therefore, we can use the neural network to obtain the embedding of users and items. By further using the attention mechanism, the user's attention to different items is calculated to update the embedding, and then the self-connection message embedding and the interactive message embedding are superimposed, and finally the prediction is made.

Formally, the attention network is defined as:

$$h_{1j} = \phi(W_1 e_{ij} + b_1) \quad (4)$$

$$\alpha_j = \frac{exp(e_u^T h_{1j})}{\sum_{p \in M} exp(e_u^T h_{1p})} \quad (5)$$

where $W_1 \in R^{d \times d}$ and $b_1 \in R^{d \times 1}$ are model parameters, where d is the dimensionality of the latent embedding spaces. We assume that the items are consecutively labeled from 1 to M, and e_{ij} represents the dense embedding vector of item j. We first feed the dense low-dimensional embedding of each item $j \in M$ through a multi-layer perceptron (MLP) to get the hidden representation h_{1j}. Function $\phi(\cdot)$ is the activation function and we utilize RELU to enhance nonlinear capability. Unlike traditional attention models that use the same context vectors for each input, we put the embedding e_u of user u as the context vector and measure the attention score α_j as the normalized similarity between h1j and e_u with the softmax function, which characterizes the importance of item j for user u. Different from the method [23], we compute the user representation as a sum of the item embeddings weighted by the attention scores as follows:

$$e_u = e_{u_N} + \sum_{j \in M} \alpha_j e_{ij} \quad (6)$$

where e_{u_N} represents the embedded representation of the Nth user. At this time, e_u contains both its own characteristics and the interactive characteristics of the user and the item.

3) MESSAGE AGGREGATION

During the propagation embedding process, messages aggregated from user's neighbors can be defined as:

$$e_u^l = LeakyReLU(m_{u \leftarrow u}^l + \sum_{i \in N_u} m_{u \leftarrow i}^l) \quad (7)$$

Through the activation layer, we can simply add and aggregate information, and update the user's own messages. The LeakyReLU activation function allows encoding of positive and negative signals. Note that in addition to messages propagating from neighbors, we also consider self-connected closed loops, which retain information about the original characteristics. Similarly, we can obtain the representation of e_i by propagating the information of item's connected users. All in all, the advantage of this section is to get the relationship between the user and the item as much as possible. The messages that are being propagated are defined as follows:

$$m_{u \leftarrow i}^{(l)} = W_1^{(l)} e_i^{(l-1)} + W_2^{(l)}(e_i^{(l-1)} \otimes e_u^{(l-1)}) \quad (8)$$

$$m_{u \leftarrow u}^{(l)} = W_1^{(l)} e_u^{(l-1)} \quad (9)$$

where $W_1, W_2 \in R^{d_l \times d_{l-1}}$, represents a trainable transformation matrix. This step stores messages from neighboring nodes. By increasing the attention mechanism to calculate the message passing weight between different user-item in the embedded propagation process, and paying attention to import adjacent user-item side messages, the user-item message weight can be enhanced.

C. PREDICTION MODEL

After propagating the L layer, we get multiple representations of user and item, namely $\{e_u^{(1)}, e_u^{(2)}, \cdots, e_u^{(l)}\}$ $\{e_i^{(1)}, e_i^{(2)}, \cdots, e_i^{(l)}\}$. Since messages obtained at different layers can represent messages delivered by different connections, they can reflect user preferences well. Therefore, we connect them to form the user's final embedding. Similarly, by performing the same operation on the item, we can obtain the final representation of the corresponding item:

$$e_u^* = e_u^{(0)} \parallel \cdots \parallel e_u^{(l)} \quad (10)$$

$$e_i^* = e_i^{(0)} \parallel \cdots \parallel e_i^{(l)} \quad (11)$$

where $\parallel$ is the connection operation [19], because the connection operation does not involve learning other parameters, it is very convenient. In this way, not only can the initial embedding be enriched by propagation embedding, but also the propagation range can be controlled by adjusting L. Finally, we use the inner product of users and items to evaluate user preferences for items:

$$y_{ANCF}(u, i) = e_u^{*^T} e_i^* \quad (12)$$

TABLE 1. Statistics of the datasets.

Dataset	Users	Items	Interaction	Density
Gowalla	29,858	40,981	1,027,370	0.00089
Amazon	52,643	91,599	2,984,108	0.00062

In this work, we emphasize the importance of attention mechanism, enhance user embedded representation by introducing attention mechanism, and evaluate user's preference for items by inner product of user-item representation.

IV. EXPERIMENTS

We perform experiments on two real datasets to evaluate our proposed method, especially the attention propagation layer. We aim to answer the following questions:

- RQ1:How does ANCF perform as compared with state-of-the-art CF method?
- RQ2:How do different hyper-parameter settings (e.g., number of layers, epoch) affect ANCF?

A. DATASET DESCRIPTION

To evaluate the performance of ANCF, we performed experiments on two benchmark datasets: Gowalla and Amazon-book, which are publicly available datasets. The main statistical information are as follows:

Gowalla:This is a check-in data set obtained from Gowalla [5], where users share their locations via check-ins. To ensure data quality, we use a 10-core setting, which keeps at least ten interactive users and items.

Amazon-Book:Amazon Reviews is a dataset that is widely used in product recommendations [5]. We also set up 10 cores to ensure that each user and item have at least ten interactions. Dataset composition: For each data set, we randomly select 80 of the user's historical interactions as the training set and the remaining 20 as the test set. In the training set, we randomly select 10 of the interactions as the validation set to adjust the hyper-parameters.

B. EXPERIMENTAL SETTINGS

1) EVALUATION INDICATORS

We consider all items that the user does not interact with as negative. Each method then outputs the user's preference score for the item, except for the active items in the training set. We evaluate the performance of ANCF by evaluating the recall and ndcg of different top-k [9].

2) BASELINES

To prove the effectiveness of ANCF, we compared it with the following methods:

- MF [12]: This is a matrix factorization optimized by Bayesian Personalized Ranking (BPR) loss optimization, which uses the user's direct interaction term as the target value of the interaction function.
- GC-MC [23]: This model uses a GCN [24] encoder to generate user and item representations, where only first-order neighbors are considered.

TABLE 2. Performance comparison.

Method	Gowalla		Amazon	
	recall	ndcg	recall	ndcg
MF	0.1113	0.1595	0.0323	0.0603
GC-MC	0.1203	0.1694	0.0312	0.0587
ANCF	0.1584	0.2279	0.0467	0.0731

3) PARAMETER SETTINGS

In our experiments, we fixed the embedding size of all models to 64. In terms of hyper-parameters, we applied a grid search to the hyper-parameters: adjusting the learning rate between $0.0001, 0.0005, 0.001, 0.005$. In addition, we use node discard technology for GC-MC and ANCF, and the ratio is adjusted to 0.0, 0.1, …0.8. We use Xavier to initialize the model parameters and execute the early stopping strategy, that is, if the recall does not increase on 50 consecutive epochs on the verification data, it will stop prematurely. We set the drop rate of the node to 0.1 and the message loss rate to 0.1.

We use discard to prevent overfitting of the neural network. Specifically, we use two techniques: node discard and message discard. Message discard discards outgoing messages randomly. Specifically, we discard the messages propagated in formulas (8) and (9) with probability $p_1 = 0.1$. In this way, in the Lth propagation layer, only a part of the message is helpful for representation. We also perform node dropping, randomly blocking a specific node with a $p_2 = 0.1$ probability and discarding all outgoing messages from this node.

C. PERFORMANCE Comparison(RQ1)

The performance comparison of all methods is as follows:

1) OVERALL COMPARISON

Table 2 shows the performance comparison of different methods, recall is to improve the recall rate on the basis of ensuring accuracy, ndcg is used to measure the quality of recommended results. We have the following results:

MF performed poorly on both datasets. This indicates that the inner product is not sufficient to capture the complex relationship between the user and the item, further limiting performance. Compared with the performance of MF, the performance of GC-MC verifies that merging first-order neighbors can improve representation learning. In all situations, the performance of ANCF performance is always better than other methods, which shows the effectiveness of the non-linear feature interaction between the user and the item embedding, and also shows that different propagation layers encode different information in the representation. It also shows that the introduction of the embedding propagation with attention mechanism can enhance the user and item representation, thus proving the importance of the weight of different neighbors to the user-item interaction. Fig4 and Fig5 and Fig6 and Fig7 show the recall and ndcg of different top-k. It can be seen from the figure that the recall and ndcg of ANCF are significantly higher than other methods, which illustrates the effectiveness of ANCF.

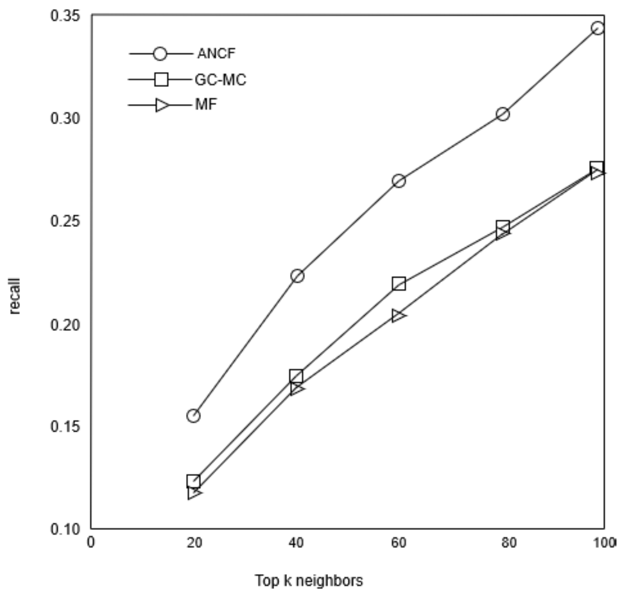


FIGURE 4. recall on Gowalla.

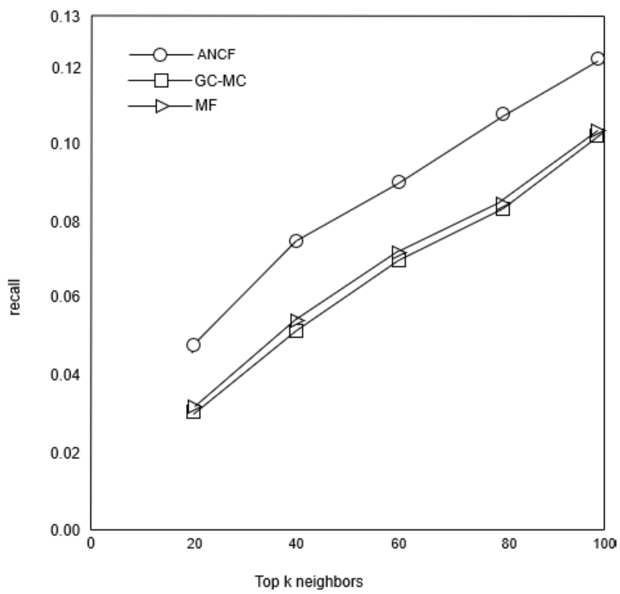


FIGURE 5. recall on Amazon.

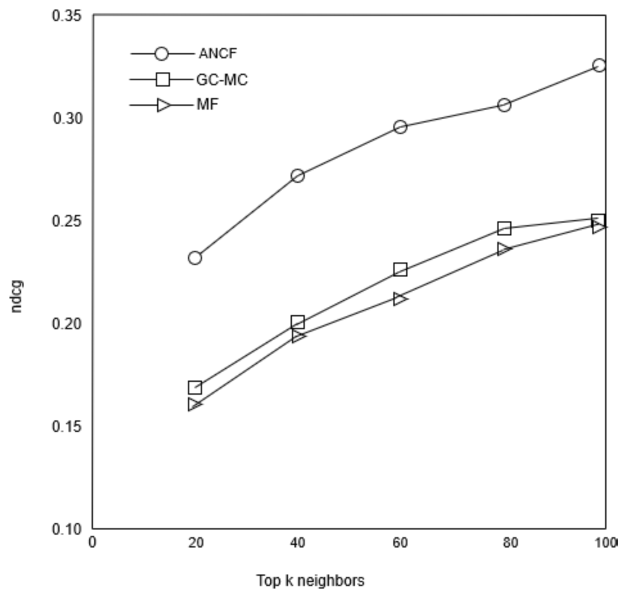


FIGURE 6. ndcg on Gowalla.

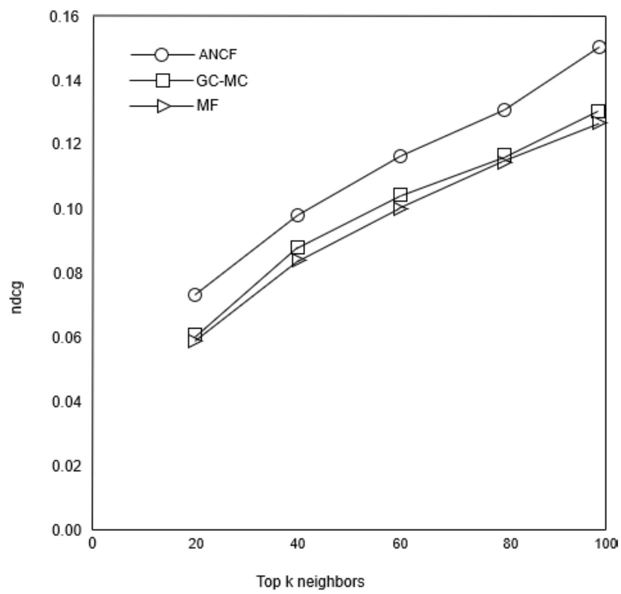


FIGURE 7. ndcg on Amazon.

D. STUDY OF ANCF(RQ2)

Since the embedding representation plays an important role in the attention propagation layer, we first study the impact of different layers on performance, then explore the impact of node loss rate on performance, and finally test performance with epoch.

1) EFFECT OF LAYER NUMBERS

In order to investigate whether the attention propagation layer can improve ANCF performance, we changed the depth of the model. We search for the layer number in the range of 1,2,3. Table3 summarizes the experimental results. We can draw the following conclusions:

Increasing the depth of ANCF can enhance the recommendation effect. Changing the number of propagation layers can obtain corresponding additional interactive signals, which also shows that the attention propagation layer can control the performance of the model.

To investigate how convolutional layers affect performance, we consider using different layers of ANCF-1 variants. In particular, we remove the representation interaction between the node and its neighbors from the message passing function, and we set it to the representation interaction

TABLE 3. Effect of attention propagation layer numbers(L).

Layer Numbers	Gowalla		Amazon	
	recall	ndcg	recall	ndcg
ANCF-1	0.1534	0.2247	0.0336	0.0649
ANCF-2	0.1557	0.2276	0.0339	0.0660
ANCF-3	0.1584	0.2279	0.0467	0.0731

TABLE 4. Effect of convolution layers.

Method	Gowalla		Amazon	
	recall	ndcg	recall	ndcg
ANCF-1	0.1534	0.2247	0.0336	0.0649
ANCF-GCMC-1	0.1474	0.2194	0.0299	0.0573
ANCF-MF-1	0.1470	0.2186	0.0294	0.0569

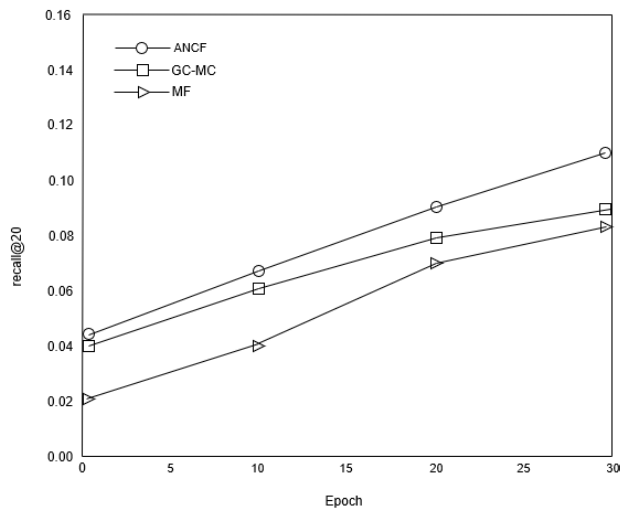


FIGURE 8. **recall@20 on Gowalla.**

between GC-MC and MF, termed ANCF-GC-MC- 1 and ANCF-MF-1. Table4 shows that ANCF-1 is consistently superior to all variants, and we attribute the improvement to the effect of attention mechanisms. Therefore, we verified the rationality and effectiveness of the attention propagation layers.

2) TESTING PERFORMANCE W.r.t. EPOCH

Fig8 and Fig9 shows the recall test performance for each epoch of ANCF, MF, and MC-GC. We can see that on two datasets, ANCF has faster convergence than GC-MC and MF. This result is reasonable, because when mini-batch optimizes interaction pairs, it will indirectly involve connected users and items. This proves the effectiveness of ANCF in propagating embeddings.

V. CONCLUSIONS AND FUTURE WORK

In collaborative filtering, learning the vector representation of users and items is the core of the recommendation system. From the early matrix factorization to the current deep learning, embedding representations are already made using existing features. The coding information hidden between

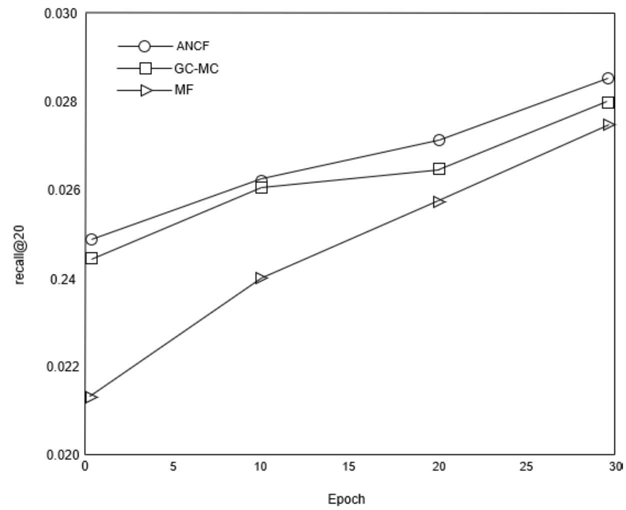


FIGURE 9. **recall@20 on Amazon.**

the user and the item cannot be represented in the embedding coding process, therefore, the result of embedding is insufficient to obtain the collaborative filtering effect. In this work, we incorporate attention mechanism into the propagation process. We have devised a new method ANCF, which uses the attention mechanism to capture different weights for user-item interactions. This makes it possible to explicitly inject user-item collaboration signals into the embedding process. A large number of experiments on two real world datasets prove the rationality and effectiveness of embedding the user graph structure with attention mechanism into the learning process. In the future, we will further improve ANCF by introducing a knowledge graph to learn more interactions between users and items. We hope that ANCF in the future can help users reason about online behaviors in order to achieve more efficient recommendations.

REFERENCES

[1] X. He, Z. He, J. Song, Z. Liu, Y.-G. Jiang, and T.-S. Chua, "NAIS: Neural attentive item similarity model for recommendation," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 12, pp. 2354–2366, Dec. 2018, doi: 10.1109/TKDE.2018.2831682.

[2] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proc. ICLR*, 2014, pp. 5013–5014. [Online]. Available: https://dblp.org/rec/journals/corr/BahdanauCB14

[3] Y. Cao, X. Wang, X. He, Z. Hu, and T.-S. Chua, "Unifying knowledge graph learning and recommendation: Towards a better understanding of user preferences," in *Proc. World Wide Web Conf. (WWW)*, 2019, pp. 151–161. [Online]. Available: https://dblp.org/rec/conf/www/0003W0HC19

[4] T. Ebesu, B. Shen, and Y. Fang, "Collaborative memory network for recommendation systems," in *Proc. 41st Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jun. 2018, pp. 515–524. [Online]. Available: https://dblp.org/rec/conf/sigir/EbesuSF18

[5] X. Wang, X. He, M. Wang, F. Feng, and T.-S. Chua, "Neural graph collaborative filtering," in *Proc. 42nd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2019, pp. 165–174. [Online]. Available: https://dblp.org/rec/conf/sigir/Wang0WFC19

[6] L. Chen, H. Zhang, J. Xiao, L. Nie, J. Shao, W. Liu, and T.-S. Chua, "SCA-CNN: Spatial and channel-wise attention in convolutional networks for image captioning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6298–6306. [Online]. Available: https://dblp.org/rec/conf/cvpr/ChenZXNSLC17

[7] S. Seo, J. Huang, H. Yang, and Y. Liu, ‘‘Interpretable convolutional neural networks with dual local and global attention for review rating prediction,’’ in *Proc. Elev. ACM Conf. Recomm. Syst. RecSys*, vol. 17, Aug. 2017, pp. 297–305.

[8] X. He and T.-S. Chua, ‘‘Neural factorization machines for sparse predictive analytics,’’ in *Proc. 40th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Aug. 2017, pp. 355–364. [Online]. Available: https://dblp.org/rec/conf/sigir/0001C17

[9] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, ‘‘Neural collaborative filtering,’’ in *Proc. WWW*, 2017, pp. 173–182. [Online]. Available: https://dblp.org/rec/conf/www/HeLZNHC17

[10] I. Bayer, X. He, B. Kanagal, and S. Rendle, ‘‘A generic coordinate descent framework for learning from implicit feedback,’’ in *Proc. 26th Int. Conf. World Wide Web*, Apr. 2017, pp. 1341–1350. [Online]. Available: https://dblp.org/rec/conf/www/BayerHKR17

[11] Y. Lu, R. Dong, and B. Smyth, ‘‘Coevolutionary recommendation model: Mutual learning between ratings and reviews,’’ in *Proc. World Wide Web Conf. World Wide Web (WWW)*, Apr. 2018, pp. 773–782. [Online]. Available: https://dblp.org/rec/conf/www/LuDS18

[12] Y. Koren, R. Bell, and C. Volinsky, ‘‘Matrix factorization techniques for recommender systems,’’ *Computer*, vol. 42, no. 8, pp. 30–37, Aug. 2009.

[13] R. Salakhutdinov, A. Mnih, and G. E. Hinton, ‘‘Restricted Boltzmann machines for collaborative fifiltering,’’ in *Proc. ICDM*, 2007, pp. 791–798. [Online]. Available: https://dblp.org/rec/conf/icml/SalakhutdinovMH07

[14] Y. Tay, L. Anh Tuan, and S. C. Hui, ‘‘Latent relational metric learning via memory-based attention for collaborative ranking,’’ in *Proc. World Wide Web Conf. World Wide Web (WWW)*, 2018, pp. 729–739. [Online]. Available: https://dblp.org/rec/conf/www/TayTH18

[15] J. Li, P. Ren, Z. Chen, Z. Ren, T. Lian, and J. Ma, ‘‘Neural attentive session-based recommendation,’’ in *Proc. ACM Conf. Inf. Knowl. Manage.* New York, NY, USA: ACM, Nov. 2017, pp. 1419–1428. [Online]. Available: https://dblp.org/rec/conf/cikm/LiRCRLM17

[16] Y. Gong and Q. Zhang, ‘‘Hashtag recommendation using attention-based convolutional neural network,’’ in *Proc. 25th Int. Joint Conf. Artif. Intell. (IJCAI)*. Palo Alto, CA, USA: AAAI Press, 2016, pp. 2782–2788. [Online]. Available: https://dblp.org/rec/conf/ijcai/GongZ16

[17] H. Wang, N. Wang, and D.-Y. Yeung, ‘‘Collaborative deep learning for recommender systems,’’ in *Proc. KDD*, 2015, pp. 1235–1244. [Online]. Available: https://dblp.org/rec/conf/kdd/WangWY15

[18] K. Xu, C. Li, Y. Tian, T. Sonobe, K.-I. Kawarabayashi, and S. Jegelka, ‘‘Representation learning on graphs with jumping knowledge networks,’’ in *Proc. ICML*, vol. 80, 2018, pp. 5449–5458. [Online]. Available: https://dblp.org/rec/conf/icml/XuLTSKJ18

[19] F. Xue, X. He, X. Wang, J. Xu, K. Liu, and R. Hong, ‘‘Deep item-based collaborative filtering for Top-N recommendation,’’ *ACM Trans. Inf. Syst.*, vol. 37, no. 3, pp. 1–25, Jul. 2019.

[20] Z. Y. Gao and C.-P. Chen, ‘‘DeepSA2018 at SemEval-2018 task 1: Multi-task learning of different label for affect in tweets,’’ in *Proc. 12th Int. Workshop Semantic Eval.*, 2018, pp. 226–230. [Online]. Available: https://dblp.org/rec/conf/semeval/GaoC18

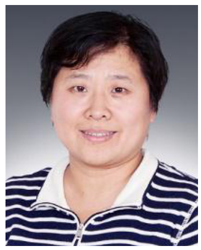
[21] R. van den Berg, T. N. Kipf, and M. Welling, ‘‘Graph convolutional matrix completion,’’ in *Proc. KDD*, 2017, pp. 1–9. [Online]. Available: https://dblp.org/rec/journals/corr/BergKW17

[22] T. N. Kipf and M. Welling, ‘‘Semi-supervised classification with graph convolutional networks,’’ in *Proc. ICLR*, 2017, pp. 1–14. [Online]. Available: https://dblp.org/rec/conf/iclr/KipfW17

[23] H. Ying, F. Zhuang, F. Zhang, Y. Liu, G. Xu, X. Xie, H. Xiong, and J. Wu, ‘‘Sequential recommender system based on hierarchical attention networks,’’ in *Proc. 27th Int. Joint Conf. Artif. Intell.*, Jul. 2018, pp. 3926–3932. https://dblp.org/rec/conf/ijcai/YingZZLXXX018

YANLI GUO was born in 1995. She received the B.S. degree in software engineering from Shandong Jianzhu University, in 2018. She is currently pursuing the M.S. degree in software engineering with Shandong University.

Since 2018, she has been conducting research with the Software and Data Engineering Research Center, School of Software. Her research interests include big data processing and recommendation systems.



ZHONGMIN YAN was born in September 1977. She graduated from the Department of Computer Science, Shandong University, in 1998. She received the M.S. and Ph.D. degrees in computer software from Shandong University, in 2001 and 2010, respectively.

She is currently an Associate Professor and the Master Tutor with Shandong University. She has been teaching with the School of Computer Science and Technology, Shandong University, from July 2001. She is also the Director of the Software Engineering Department, School of Software of Shandong University. In recent years, she has been engaged in the research of data integration, presided over and participated in a number of national and provincial projects and horizontal projects. She has hosted and participated in national and provincial level projects, such as the National Natural Science Foundation of China, the National Science and Technology Support Program, the Shandong Science and Technology Development Plan, and the Shandong Province Young and Middle-Aged Scientists Scientific Research Award Fund. She is a member of the Special Committee of Electronic Government Affairs and Office Automation of the Chinese Computer Society. Her research interests include large-scale data management and Web data integration and intelligent data analysis.