International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Traitement du Signal
Vol. 40, No. 2, April, 2023, pp. 775-781
Journal homepage: http://iieta.org/journals/ts

GCNE: Graph Convolution Networks with Explicitly Influence for Recommendation

Xili Cai[1], Jingchang Wang[2], Dewen Seng[2], Xuefeng Zhang[2,3*]

[1] School of IoT Technology, Hangzhou Polytechnic, Hangzhou 311402, China
[2] Key Laboratory of Complex Systems Modeling and Simulation, Hangzhou Dianzi University, Hangzhou 310018, China
[3] School of Information Engineering, College of Science and Technology, Ningbo University, Ningbo 315300, China

Corresponding Author Email: zhangxf@hdu.edu.cn

https://doi.org/10.18280/ts.400238

Received: 19 January 2023
Accepted: 1 April 2023

Keywords:
collaborative filtering, recommender system, embedding propagation, graph neural network

ABSTRACT

Graph Convolution Network (GCN) has become increasingly important for collaborative filtering with the modeling of user-item interaction graphs through embedding propagation. Existing work that can adapts GCN to well capture accurate user preference, which highly rely on learned representations with sufficient and high-quality training data. However, the neighborhood aggregation scheme in GCN enlarges the impact of interactions on representation learning, making the learning more vulnerable to interaction noises, since the user-item interaction graph is also modeled by same neural operations that may be unnecessary. In this paper, we propose to integrate the explicitly feedback (i.e., user-item ratings) representation of user-item interactions into the embedding process to enhance recommendation performance. We develop a novel Graph Convolution Network framework with Explicitly feedback (GCNE), which augments user-item representations by explicitly exploiting the user-item ratings feedback among entities in the predictive model, which better alleviates the interaction noises problem and data sparsity. Specifically, we introduce a adjacency matrix by regarding user behaviors and item ratings feedback as two bipartite graphs, such module could explicitly explore the propagation process of user interest and feedback influence, so as to enhance the robustness of recommendation systems. Extensive experiments demonstrate that GCNE can significantly improve the performance over various state-of-the-art baselines. Further analysis verifies the superior representation ability of our GCNE recommendation framework in alleviating the data sparsity and noise issues.

1. INTRODUCTION

Recommender systems have become increasingly important to alleviate the information overload for users in a variety of web applications, such as e-commerce systems, streaming video sites and location-based lifestyle apps [1]. Traditional recommender systems usually lean the low dimensional latent representations sorely from the user-item interaction data; for a recommender system, when the user-term interaction data is insufficient, the system's representation ability would be significantly affected. This is referred to as the data sparse problem [2]. To accurately infer the user preference, encoding user and item informative representations is the core part of effective collaborative filtering (CF) paradigms based on the observed user-item interactions [3]. CF addresses it by assuming that behaviorally similar users would exhibit similar preference on items, a common paradigm is to parameterize users and items for reconstructing historical interactions, and predict user preference based on the parameters [4]. However, efficient CF-based methods largely depend on the richness degree of the user-item interactions. Therefore, they have poor performance in some scenarios where item information or interactive records are sparse and insufficient, which often leads to the cold start problems in practical marketing [5]. To obtain high-quality embedding representations, Recent years have witnessed the development of GCN for embedding user-item interactions, to perform the information propagation along the user-item interactions to refine user embeddings based on the recursive aggregation schema. For example, Ying et al. [6] proposed a data efficient GCN algorithm PinSage, which combines efficient random walks and graph convolutions to generate embeddings of nodes (i.e., items) that incorporate both graph structure as well as node feature information. Wang et al. [4] developed a new recommendation framework neural graph collaborative filtering (NGCF), which exploits the user-item graph structure by propagating embeddings on it. This leads to the expressive modeling of high-order connectivity in user-item graph, effectively injecting the collaborative signal into the embedding process in an explicit manner. He et al. [7] proposed a new model named LightGCN, including only the most essential component in GCN neighborhood aggregation for collaborative filtering. Specifically, LightGCN learns user and item embeddings by linearly propagating them on the user-item interaction graph, and uses the weighted sum of the embeddings learned at all layers as the final embedding. Zhang et al. [8] proposed a knowledge-aware representation graph convolutional network for Recommendation (KCRec). The purpose is to enhance the representation ability of recommender system, by aggregating and propagating the user features and item attributes through the different importance of various relationships in KG, besides exploring the user high-order interest. Yuan et al. [9] proposed a novel dynamic heterogeneous graph convolutional network (DyHGCN) to jointly learn the structural characteristics of the social graph and dynamic diffusion graph, and encoded the temporal information into the heterogeneous graph to learn the users' dynamic preferences. Wang et al. [10] proposed KGCN, which combined knowledge graph and recommendation with GNNs to dig the relationship of goods on knowledge graph, and effectively capture the internal relationship of goods and alleviate data sparsity. Yang et al. [11] propose a spatio temporal aggregation method STAM to efficiently incorporate temporal information into neighbor embedding learning. Wu et al. [12] proposed a model-agnostic framework self-supervised graph learning (SGL) to supplement the supervised recommendation task with self-supervised learning on user-item graph. From the perspective of graph structure, they devised three types of data augmentation from different aspects to construct the auxiliary contrastive task. Xia et al. [1] proposed a novel self-supervised hypergraph Transformer framework (SHT) which augments user representations by exploring the global collaborative relationships in an explicit way.

Despite the effectiveness of the above GCN-based recommender models by providing state-of-the-art recommendation performance, the two challenges have not been well addressed in existing methods.

Firstly, data noise is ubiquitous in many recommendation scenarios due to a variety of factors. Data sparsity and skewed distribution issue still stand in the way of effective user-item interaction modeling, leading to most existing graph-based CF models being biased towards popular items [13, 14]. For example, most feedback that a user provides is implicit (e.g., clicks, views), instead of explicit (e.g., ratings, likes/dislikes). As such, observed interactions usually contain noises, e.g., a user is misled to click an item and finds it uninteresting after consuming it [15]. Moreover, the neighborhood aggregation scheme in GCNs enlarges the impact of interactions on representation learning, making the learning more vulnerable to interaction noises, since the user-item interaction graph is also modeled by same neural operations that may be unnecessary. Therefore, exploiting the feedback between the users is particularly important to model a new user representation, whereas considering the importance of different user-item interactions in user-item interaction graph is also necessary for the establishment of representation of items.

Secondly, the performance of graph-based recommendation largely depends on the construction of the bipartite graph. The majority of graph-based approaches aim to model the existence of user-item interactions [4, 7, 16], which cannot express the users' specific purchase intention, making it difficult to capture fine-grained user preferences. For example, Figure 1 shows a relation graph of 5 users and 5 items, which are connected by 8 edges. Each edge is associated with an explicitly feedback (i.e., user-item ratings) of 1 to 5 stars, representing the level of user preference of connected item. u_1 and u_2 both have the same rating on item i_2, which illustrate u_1 and u_2 have the similarity preference. According to their browsing history, u_2 is more likely to focus on item i_1, while u_4 is also more likely to focus on item i_1. Therefore, modeling the fine-grained reason of interaction is more conducive to providing personalized recommendation services for u_2 and u_4.

In order to handle the aforementioned shortcomings, we propose a novel Graph Convolution Network framework with Explicitly feedback (GCNE), which augments user-item representations by explicitly exploiting the feedback among entities in the predictive model. Different from the traditional GCN only considering user-item interaction graph for recommendation, our proposed graph learner further incorporates the explicitly exploiting the user-item ratings feedback, which directly could explicitly explore the propagation process of user interest and feedback influence, so as to enhance the robustness of recommendation systems. We summarize the contributions of this work as below:

●We propose a novel Graph Convolution Network framework with Explicitly feedback (GCNE), which consists of three layers: embedding layer, multiple embedding propagation and model prediction layer. GCNE augments user-item representations by explicitly exploiting the user-item ratings feedback among entities in the predictive model, which better alleviates the interaction noises problem and data sparsity.

●We introduce an adjacency matrix by regarding user behaviors and item ratings feedback as two bipartite graphs, such module could explicitly explore the propagation process of user interest and feedback influence, so as to enhance the robustness of recommendation systems.

●Extensive experiments demonstrate that GCNE can significantly improve the performance over various state-of-the-art baselines. Further analysis verifies the superior representation ability of our GCNE recommendation framework in alleviating the data sparsity and noise issues.

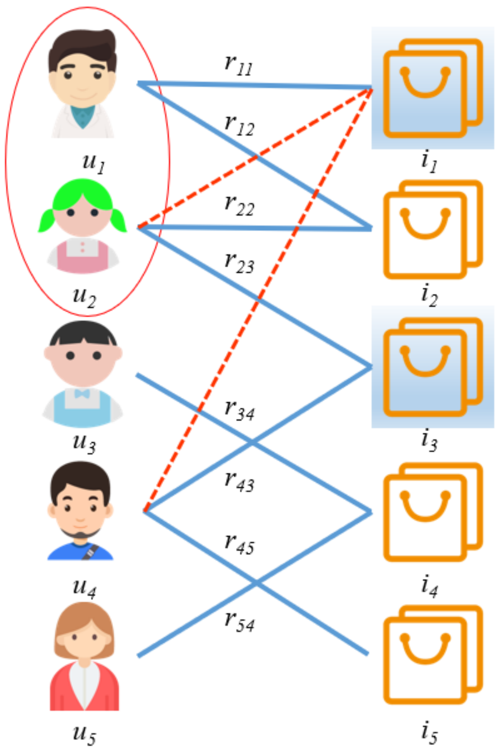


Figure 1. An illustration of explicitly feedback (user-item ratings) influence for recommendation through a toy example

2. METHOD

In this section, we first present our designed GCNE model, as illustrated in Figure 2. GCNE involves three key components: (1) an embedding layer offers and initialization of user embeddings and item embeddings; (2) multiple embedding propagation layers are stacked to propagate embeddings by injecting high-order connectivity; and (3) model prediction that aggregates refined embeddings from different propagation layers and output prediction results of recommendation. Lastly, we describe how to do model training for recommendation.

(1) Embedding layer

An embedding layer that offers and initialization of user embeddings and item embeddings. After being initialized in the embedding layer, the user-item potential collaborative relations will be captured in the subsequential training process of the recommendation model. Moreover, the node feature in graph will be mapped into low-dimensional space, the embeddings will be iteratively optimized.

(2) Embedding Propagation layer

We devise an embedding propagation layer, which refines a user's (or an item's) embedding by aggregating the embeddings of the interacted items (or users). By stacking user-item interaction graph and adjacency matrix, we can enforce the embeddings to capture the collaborative signal in user-item high-order representation.

(3) Prediction layer

The prediction layer that aggregates the refined embeddings from different propagation layers and outputs the affinity score of a user-item pair, and output prediction results of recommendation.

2.1 Embedding layer

Following mainstream recommendation models [4, 7], in the initial step, we describe each user u and item i with an embedding vector $e_u \in \mathbb{R}^d$ and $e_i \in \mathbb{R}^d$ respectively, where d denotes the embedding size. The embedding of N users and M items could be expressed by matrix E as follows:

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

After being initialized in the embedding initialization stage, the user-item potential collaborative signals then will be captured in the subsequential training process of the recommendation model. Moreover, the node feature in graph will be mapped into low-dimensional space, the embeddings will be iteratively optimized. The user and item embedding matrices then will be updated by aggregation and propagation.

2.2 Embedding propagation layers

In the general recommendation scenarios, the accurate modeling of the user-item high-order representation, based on the existing explicitly user-item interaction and implicitly feedback, is a core problem. In this section, borrowing the idea of the recent LightGCN [7], we explore the user-item representation based on the alignment of a user-item interaction graph and implicitly feedback by applying graph convolutional networks, which can significantly enhances the accuracy of the recommender system. We adopt the simple weighted sum aggregator and abandon the use of feature transformation and nonlinear activation. The graph convolution operation in GCNE is defined as:

$$\begin{aligned} e_u^{k+1} &= \sum_{i \in N_u} \frac{1}{\sqrt{|N_u|}\sqrt{|N_i|}} e_i^{(k)}, \\ e_i^{k+1} &= \sum_{u \in N_i} \frac{1}{\sqrt{|N_u|}\sqrt{|N_i|}} e_u^{(k)} \end{aligned} \tag{2}$$

The symmetric normalization term $\frac{1}{\sqrt{|N_u|}\sqrt{|N_i|}}$ follows the design of standard GCN [17], which can avoid the scale of embeddings increasing with graph convolution operations. After K layers propagation, we finally combine the embeddings obtained at each layer to form the final representation of a user (an item):

$$e_u = \sum_{k=0}^{K} \alpha_k e_u^{(k)}, \quad e_i = \sum_{k=0}^{K} \alpha_k e_i^{(k)} \tag{3}$$

where, α_k denotes the weight value of the k-th layer embedding in constituting the final embedding. In order to avoid unnecessary complexity of GCNE, we set α_k uniformly as $1/k+1$, which will lead to good performance in general.

In order to capture a holistic view of embedding propagation and facilitate implementation, we provide the matrix form of our proposed GCNE, the layer-wise propagation rule as follows:

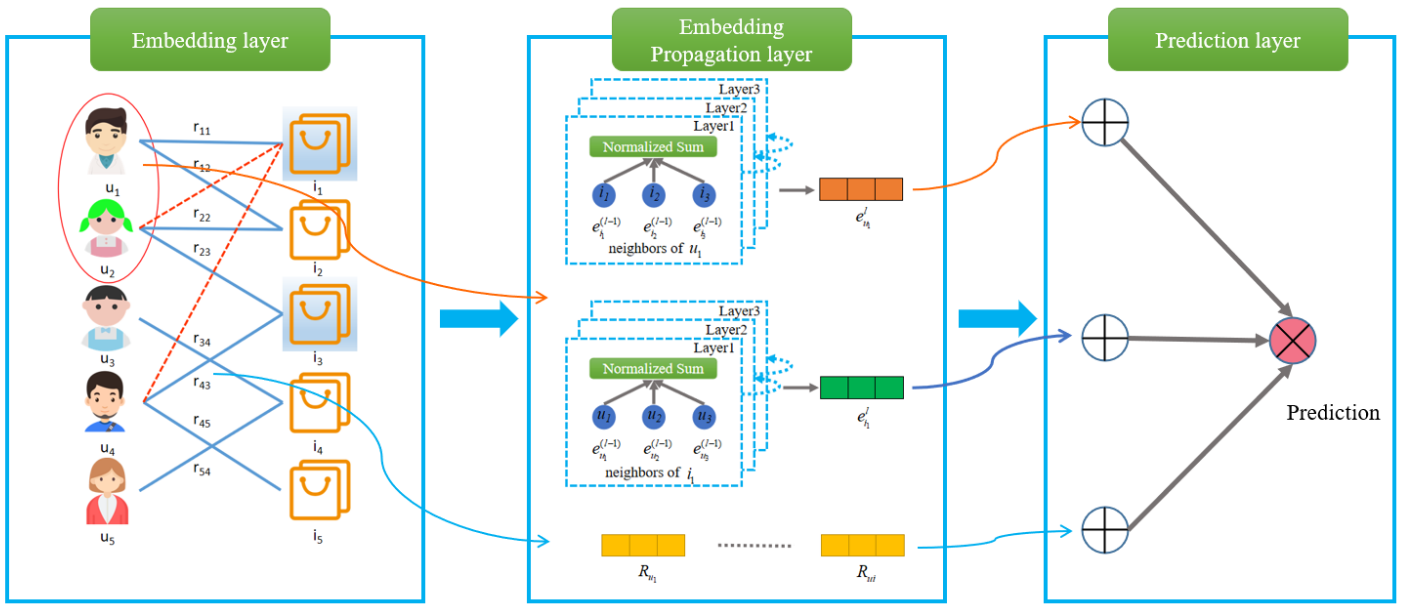


Figure 2. An illustration of our GCNE model architecture

$$E^{(k)}=\text{LeakyReLU}\left((\mathcal{L}+\text{I})E^{(k-1)}W_1^l+\mathcal{L}E^{(k-1)}\odot E^{(k-1)}W_2^k\right) \quad (4)$$

where, $E^{(k)} \in \mathbb{R}^{(N+M)*d_k}$ is the representation sets for users and items obtained after k layers of embedding propagation. $E^{(0)}$ is set as E at the initial message-passing iteration, L represents the Laplacian matrix for the user-item graph, which is formulated as:

$$\mathcal{L}=D^{-\frac{1}{2}}AD^{-\frac{1}{2}},\ A=\begin{pmatrix} 0 & R \\ R^T & 0 \end{pmatrix} \quad (5)$$

where, $R \in \mathbb{R}^{N+M}$ is the user-item interaction matrix, and M and N denote the number of users and items, respectively, A is the adjacency matrix and D is the diagonal degree matrix, each entry R_{ui} is rating if u has interacted with item i otherwise 0, which augments user-item representations by explicitly exploiting the feedback.

2.3 Prediction layer

When the aggregation layer and propagation layer are aggregated and propagated for l layers, we will find that multiple operations can not only learn the dependence of explicitly feedback relationships, but also explore the higher-order connectivity. We observe how the learning results are influenced in experiment, since the representations obtained in different layers emphasize the messages passed over different connections, they have different contributions in reflecting user preference. So, we connect and get final vector of user embedding e_u and vector of item embedding e_i. The range of l can not only effectively obtain the embedding vectors of users and items, but also avoid insufficient information extraction. A larger value indicates that the space and time complexity are higher, while a smaller value indicate the opposite. The model prediction is defined as the inner product of user and item final representations:

$$\hat{y}_{ui}=e_u^T e_i \quad (6)$$

which is used as the ranking score for recommendation generation.

3. EXPERIMENTS

In this section, we present the details of experiment setups in Section 3.1, 3.2, and 3.3, then the corresponding experimental results of our proposed GCNE in comparison with the state-of-the-art recommendation models in Section 3.4, especially the embedding propagation layer. To justify the designs in GCNE and reveal the reasons of its effectiveness, we perform embedding dimension analyses in NDCG in Section 3.5.

3.1 Datasets

To verify the unbiasedness of the model, we conduct experiments on three real-world datasets: Movielens, Amazon Games, and Book-crossing. These datasets are all publicly accessible, independent of each other, and vary in application domain, data size, and data sparsity.

Table 1. Statistics of the experimented datasets

Datasets	Users	Items	Interaction	Sparsity
MovieLens	610	9724	100836	0.0169
Amazon Games	3627	27540	132515	0.0013
Book-crossing	8001	14472	138575	0.0011

The attributes of the three datasets are shown in Table 1, Book-crossing datasets obtains higher data sparsity. The obvious observation is that the recommendation performance of the model under different sparsity levels. The details of the datasets are as follows:

(1) Book-crossing [8]: is widely applied datasets in book recommendations, which includes 8001 users of the Book-Crossing book community. There are 138575 ratings for 14472 books, with each rating ranging from 1 to 10.

(2) MovieLens [11]: MovieLens is a widely adopted benchmark dataset for recommendation models. In this work, we use MovieLens Latest Datasets (ml-latest-small), which is a small subset of MovieLens data, and which is consistent with MovieLens in form (including user and movie information and reviews). Specifically, there are 9,742 movies for 610, generating a data set of 100836 lines.

(3) Amazon Games: Amazon-review is a widely used datasets for product recommendation [18]. We select Amazon-Games from the collection. Similarly, we use the 10-core setting to ensure that each user and item have at least ten interactions.

3.2 Baselines

In order to test the performance improvement of the GCNE model, we conduct comparative experiments with some current advanced recommendation models. The models are introduced as follows:

(1) NGCF [4]: The NGCF model explicitly models the interaction information between users and items, constructs a user-item interaction graph, and on this basis, effectively embeds the collaboration signal into the user-item interaction graph propagation process to achieve high-order connectivity, which improves the embedded representation of users and items.

(2) NGCF-ITS [5]: The NGCF-ITS is a hybrid recommendation model that fuses with user-item interactions information and item temporal sequence relationships. It divides the item temporal sequences into several groups of subsequences through the sliding window strategy, then constructs the item temporal sequence relationships graph and aggregates the characteristics of item temporal sequences information.

(3) LightGCN [7]: LightGCN model simplifies NGCF model. It eliminates feature transformation and nonlinear activation in NGCF, because these designs have no positive impact on collaborative filtering, and reduces part of the noise of NGCF embedding. Compared with NGCF, LightGCN not only improves the operation efficiency, but also significantly improves the recommendation performance.

3.3 Evaluation protocols

In this section, we take all-ranking protocol and adopt four evaluation metrics: Recall@N and NDCG@N. The detailed introduction is shown as follow:

$$\mathrm{Re}call@N = \frac{1}{|\overline{U}|}\sum_{u\in\overline{U}}\frac{|R_u \cap I_u|}{|I_u|} \quad (7)$$

$$NDCG@N = \frac{1}{|\overline{U}|}\sum_{u\in\overline{U}}\frac{DCG@N}{IDCG@N} \quad (8)$$

In our experiments, the length of the recommendation N is in 10. $\overline{U}$ represents all users of test set. R_u denotes all item sets which were recommended by user u. I_u means item sets which have interacted with user u. The precision, recall rate and NDCG under N recommended items, respectively denoted as Recall@N and NDCG@N.

3.4 Performance comparison with state-of-the-arts

To evaluate our proposed model, the experiments are carried out on three real-world datasets, and to the accuracy performance of the propose model, we compare with other recommendation models, such as the NGCF based on GNN, NGCF-ITS based on NGCF and LightGCN model simplifies NGCF. The above model is described in Baselines.

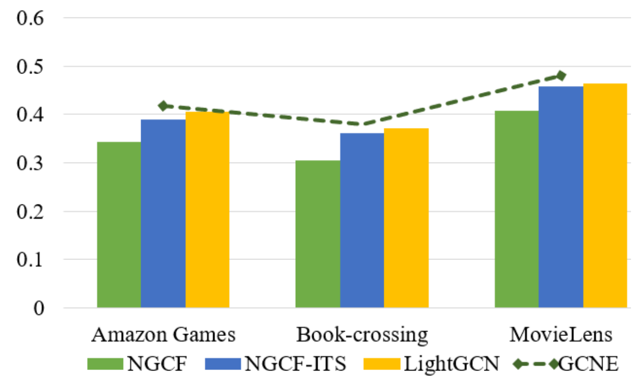
(1) Table 2 shows the experimental results on Movielens, Amazon Games, and Book-crossing with N=10. we have the following observations: our proposed model GCNE typically outperformed approaches, which reflects the accuracy and effectiveness. In addition, based GCN model use graph structures for message propagation to achieve performance improvements.

(2) GCNE aggregates explicitly feedback (i.e., user-item ratings) representation, this not only increases the model representation ability, but also boosts the performance for recommendation, and GCNE generally achieves better performance than other methods in most cases.

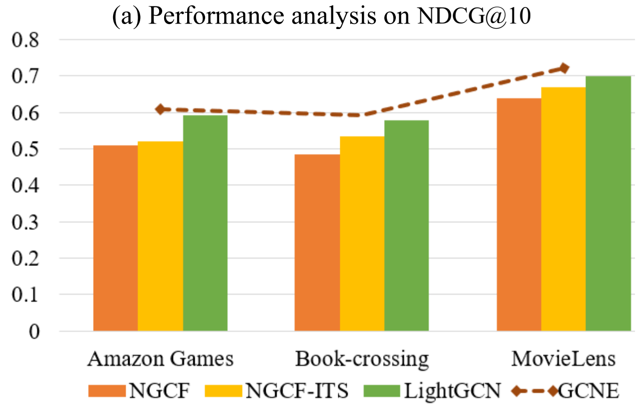
(3) GCNE consistently yields the best performance on three datasets. In particular, we analyze the impact of Recall@10 and NDCG@10 on three datasets in Table 2 and Figure 3, GCNE improves over LightGCN is 3.0% (Recall@10) on Amazon Games, 2.6% (Recall@10) on Book-crossing and 3.4% (Recall@10) on MovieLens. Meanwhile, GCNE improves over LightGCN is 3.2% (NDCG@10) on Amazon Games, 2.5% (NDCG@10) on Book-crossing and 3.2% (NDCG@10) on MovieLens. This phenomenon indicates that aggregation of item temporal sequence relationships into GNN-based recommendations has sufficient development prospects.

Table 2. Performance analysis on Recall@N and NDCG@N

Method	Amazon Games		Book-crossing		MovieLens	
	Recall@N	NDCG@N	Recall@N	NDCG@N	Recall@N	NDCG@N
NGCF	0.5105	0.3437	0.4847	0.3055	0.6402	0.4072
NGCF-ITS	0.5201	0.3896	0.5346	0.3618	0.6693	0.4579
LightGCN	0.5924	0.4049	0.5783	0.3709	0.6987	0.4649
GCNE	0.6099	0.4179	0.5934	0.3803	0.7228	0.4796
Improv.	3.0%	3.2%	2.6%	2.5%	3.4%	3.2%



(a) Performance analysis on NDCG@10



(b) Performance analysis on Recall@10

Figure 3. Performance analysis on NDCG@N and Recall@N

3.5 Impact of the embedding dimension d in NDCG

To analyze the importance of embedding dimension d on three datasets in NDCG, the dimension d is in the range of {32,50,64,128}. The experiment analysis is shown in Table 3 and Figure 4, the experiment performance is worst on three datasets with d=32. During the increase of d, the experiment data on three datasets tend to better, the effect of rising is especially obvious in d=32 to d=50 on Amazon Games, then the changing is slowly in d=64 to d=128. The experiment results reach the best performance in d=128 of 0.4699, 0.4087 and 3736 on all three datasets respectively.

Table 3. Analysis of different d on datasets in NDCG

Datasets	32	50	64	128
MovieLens	0.4601	0.4633	0.4649	0.4699
Amazon Games	0.3997	0.4021	0.4049	0.4087
Book-crossing	0.3633	0.3687	0.3709	0.3736

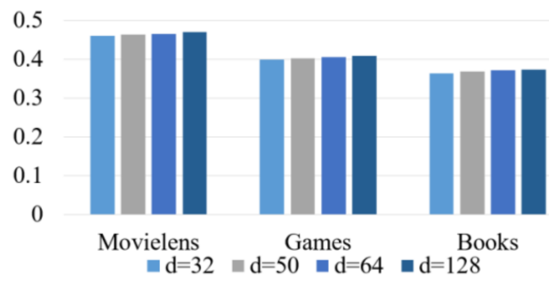


Figure 4. Analysis of different d on datasets in NDCG

The obvious observation is that the performance tends to converge with the increase of dimension. Intuitively, a larger dimension will result in better performance but a larger dimension leads to a longer training time. Therefore, we need to find a proper dimension to balance the trade-off between performance and time consumption in the future work.

4. CONCLUSION

In this paper, we propose to integrate the explicitly feedback (i.e., user-item ratings) representation of user-item interactions into the embedding process to enhance recommendation performance. We develop a novel Graph Convolution Network framework with Explicitly feedback (GCNE), which augments user-item representations by explicitly exploiting the user-item ratings feedback among entities in the predictive model, which better alleviates the interaction noises problem and data sparsity. Specifically, we introduce an adjacency matrix by regarding user behaviors and item ratings feedback as two bipartite graphs, such module could explicitly explore the propagation process of user interest and feedback influence, so as to enhance the robustness of recommendation systems. Extensive experiments demonstrate that GCNE can significantly improve the performance over various state-of-the-art baselines. Further analysis verifies the superior representation ability of our GCNE recommendation framework in alleviating the data sparsity and noise issues.

FUNDING

This work was supported in part by the Fundamental Public Welfare Research Program of Zhejiang Province (Grant No.: LGF22F020036) and in part by the National Experimental Teaching Demonstration Center of Computer Science in China.

REFERENCES

[1] Xia, L., Huang, C., Zhang, C. (2022). Self-supervised hypergraph transformer for recommender systems. In Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Washington, USA pp. 2100-2109. https://doi.org/10.1145/3534678.3539473

[2] Liao, J., Zhou, W., Luo, F., Wen, J., Gao, M., Li, X., Zeng, J. (2022). SocialLGN: Light graph convolution network for social recommendation. Information Sciences, 589: 595-607. https://doi.org/10.1016/j.ins.2022.01.001

[3] Rendle, S., Krichene, W., Zhang, L., Anderson, J. (2020). Neural collaborative filtering vs. matrix factorization revisited. In Proceedings of the 14th ACM Conference on Recommender Systems, Brazil, pp. 240-248. https://doi.org/10.1145/3383313.3412488

[4] Wang, X., He, X., Wang, M., Feng, F., Chua, T.S. (2019). Neural graph collaborative filtering. In Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, Paris, France, pp. 165-174. https://doi.org/10.1145/3331184.3331267

[5] Seng, D., Li, M., Zhang, X., Wang, J. (2022). Research on neural graph collaborative filtering recommendation model fused with item temporal sequence relationships. IEEE Access, 10: 116972-116981. https://doi.org/10.1109/ACCESS.2022.3215161

[6] Ying, R., He, R., Chen, K., Eksombatchai, P., Hamilton, W.L., Leskovec, J. (2018). Graph convolutional neural networks for web-scale recommender systems. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, London, United Kingdom, pp. 974-983. https://doi.org/10.1145/3219819.3219890

[7] He, X., Deng, K., Wang, X., Li, Y., Zhang, Y., Wang, M. (2020). LightGCN: Simplifying and powering graph convolution network for recommendation. In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, China, pp. 639-648. https://doi.org/10.1145/3397271.3401063

[8] Zhang, L., Kang, Z., Sun, X., Sun, H., Zhang, B., Pu, D. (2021). KCRec: Knowledge-aware representation graph convolutional network for recommendation. Knowledge-Based Systems, 230: 107399. https://doi.org/10.1016/j.knosys.2021.107399

[9] Yuan, C., Li, J., Zhou, W., Lu, Y., Zhang, X., Hu, S. (2021). DyHGCN: A dynamic heterogeneous graph convolutional network to learn users' dynamic preferences for information diffusion prediction. In Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2020, Ghent, Belgium, pp. 347-363. https://doi.org/10.1007/978-3-030-67664-3_21

[10] Wang, H., Zhao, M., Xie, X., Li, W., Guo, M. (2019). Knowledge graph convolutional networks for recommender systems. In The World Wide Web Conference, San Francisco, USA, pp. 3307-3313. https://doi.org/10.1145/3308558.3313417

[11] Yang, Z., Ding, M., Xu, B., Yang, H., Tang, J. (2022). Stam: A spatiotemporal aggregation method for graph neural network-based recommendation. In Proceedings of the ACM Web Conference 2022, Lyon, France, pp. 3217-3228. https://doi.org/10.1145/3485447.3512041

[12] Wu, J., Wang, X., Feng, F., He, X., Chen, L., Lian, J., Xie, X. (2021). Self-supervised graph learning for recommendation. In Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, New York, United States, pp. 726-735. https://doi.org/10.1145/3404835.3462862

[13] Krishnan, A., Sharma, A., Sankar, A., Sundaram, H. (201). An adversarial approach to improve long-tail performance in neural collaborative filtering. In Proceedings of the 27th ACM International Conference on Information and Knowledge Management, Torino, Italy, pp. 1491-1494. https://doi.org/10.1145/3269206.3269264

[14] Zhang, Y., Cheng, D.Z., Yao, T., Yi, X., Hong, L., Chi, E.H. (2021). A model of two tales: Dual transfer learning framework for improved long-tail item recommendation. In Proceedings of the Web Conference 2021, Ljubljana, Slovenia, pp. 2220-2231. https://doi.org/10.1145/3442381.3450086

[15] Wang, W., Feng, F., He, X., Nie, L., Chua, T.S. (2021). Denoising implicit feedback for recommendation. In Proceedings of the 14th ACM International Conference on Web Search and Data Mining, Israel, pp. 373-381.

https://doi.org/10.1145/3437963.3441800

[16] Wu, B., Zhong, L., Li, H., Ye, Y. (2022). Efficient complementary graph convolutional network without negative sampling for item recommendation. Knowledge-Based Systems, 256: 109758. https://doi.org/10.1016/j.knosys.2022.109758

[17] Kipf, T.N., Welling, M. (2016). Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907.

[18] He, R., McAuley, J. (2016). Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In proceedings of the 25th International Conference on World Wide Web, Montréal Québec, Canada, pp. 507-517. https://doi.org/10.1145/2872427.2883037