ConsisRec: Enhancing GNN for Social Recommendation via Consistent Neighbor Aggregation

Liangwei Yang, Zhiwei Liu, Yingtong Dou
Department of Computer Science, University of Illinois at Chicago
{lyang84,zliu213,ydou5}@uic.edu

Jing Ma
Business School, Sichuan University
jingma@stu.scu.edu.cn

Philip S. Yu
Department of Computer Science, University of Illinois at Chicago
psyu@uic.edu

ABSTRACT

Social recommendation aims to fuse social links with user-item interactions to alleviate the cold-start problem for rating prediction. Recent developments of Graph Neural Networks (GNNs) motivate endeavors to design GNN-based social recommendation frameworks to aggregate both social and user-item interaction information simultaneously. However, most existing methods neglect the social inconsistency problem, which intuitively suggests that social links are not necessarily consistent with the rating prediction process. Social inconsistency can be observed from both context-level and relation-level. Therefore, we intend to empower the GNN model with the ability to tackle the social inconsistency problem. We propose to sample consistent neighbors by relating sampling probability with consistency scores between neighbors. Besides, we employ the relation attention mechanism to assign consistent relations with high importance factors for aggregation. Experiments on two real-world datasets verify the model effectiveness.

CCS CONCEPTS

• **Computing methodologies → Neural networks**; • **Information systems → Social recommendation**.

KEYWORDS

Recommender System; Social Recommendation; Graph Neural Network

ACM Reference Format:
Liangwei Yang, Zhiwei Liu, Yingtong Dou, Jing Ma, and Philip S. Yu. 2021. ConsisRec: Enhancing GNN for Social Recommendation via Consistent Neighbor Aggregation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '21), July 11–15, 2021, Virtual Event, Canada.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3404835.3463028

1 INTRODUCTION

A recommender system predicts how likely a user is interested in an item [19–21, 26, 27]. However, due to the high cost of data collection, most existing recommender systems suffer from the cold-start problem [18]. To alleviate it, we can incorporate the social information among users [15, 32, 35], which functions as a side information of the user-item interaction. Previous endeavour [2] shows that users' online behaviors are greatly influenced by their social networks, such as the friendship on Wechat [36], following links on Twitter [16] and trusting links on shopping website [7]. Therefore, fusing the social links with user-item interactions is advantageous to improve the recommendation performance, which is defined as the social recommendation problem.

The recent developments of Graph Neural Networks (GNNs) [17, 19] help handle social recommendation tasks by simultaneously aggregating the information from both social graph and user-item graph [9, 24]. Based on the assumption that neighbors share similar contexts, GNN learns node embeddings by aggregating neighbor information recursively on graphs [5]. SocialGCN [33, 34] proposes to enhance user embedding by simulating how users are influenced by the recursive social diffusion process. GraphRec [6] and GraphRec+ [7] jointly model three types of aggregation upon social graph, user-item graph and item-item graph, to learn user&item embeddings comprehensively. DSCF [8] includes high-order social links through a sequential learning on random walks. MGNN [37] builds mutual social embedding layers to aggregate information from user-item rating graph and social graph.

However, most existing GNN-based social recommendation models ignore the **social inconsistency** problem [17]. Specifically, the social inconsistency suggests that social links are not necessarily consistent with the rating prediction process. Aggregating the information from inconsistent social neighbors spoils the ability of a GNN to characterize beneficial information for recommendation. The social inconsistency can be categorized into two levels:

- *Context-level:* It indicates that users connected in a social graph may have discrepant item contexts. We demonstrate the context-level social inconsistency in Figure 1(a). We use dash lines and the solid lines to represent user-item ratings and social connections, respectively. As seen, u_3 would be u_2's inconsistent neighbor because the items of u_3 are all *books*, while u_2's rated items all belongs to *sports*. They have rather discrepant item contexts.
- *Relation-level:* There are multiple relations when simultaneously modeling social graph and user-item graph. For example, besides social relations, we also distinguish user-item relations by their rating values. In Figure 1(a), we observe the u_1 and u_2 are social neighbors and both connected with t_1. However, u_1 highly likes t_1 (5 score) while u_2 dislikes it (1 score). It leads to the relation-level inconsistency because though socially connected, they are of inconsistent item preference.

SIGIR '21, July 11–15, 2021, Virtual Event, Canada

ACM ISBN 978-1-4503-8037-9/21/07…$15.00
https://doi.org/10.1145/3404835.3463028

To this end, we intend to empower the GNN model to solve the social inconsistency problem, which is non-trivial. On the one hand, the contexts for both users and items are rather complex and difficult to express explicitly. On the other hand, we should model multiple relations simultaneously and distinguish the consistent neighbors. Therefore, we propose a novel framework to tackle the social inconsistency problem when conducting social recommendation, which is named as ConsisRec. It is built upon a GNN model [13], aggregating neighbors to learn node embeddings. To alleviate the social inconsistency problem, ConsisRec first generates a query embedding for selecting consistent neighbors. Then, it employs a neighbor sampling strategy for the selection process, where the sampling probability is based on our proposed consistency scores between the query embedding and neighbor embeddings. After sampling, it adopts relation attention to tackle the relation-level inconsistency. As such, neighbors with consistent relations are assigned with high importance factors for aggregation. Therefore, the learned node embeddings for rating prediction are aggregated from consistent contexts and relations. The code is available online at https://github.com/YangLiangwei/ConsisRec. The contributions of this paper are listed as follows:

- To the best of our knowledge, we are the first work empowering the GNN model to tackle the social inconsistency problem when conducting social recommendation.
- We propose a novel framework, ConsisRec, to learn consistent node embeddings for rating prediction.
- Experiments on two real-world datasets show the effectiveness of ConsisRec. Detailed analyses of ConsisRec justify its efficacy.

2 PRELIMINARIES

Social recommendation problem consists of two set of entities, a user set $\mathcal{U} = \{u_1, u_2, ..., u_m\}$ and an item set $\mathcal{T} = \{t_1, t_2, ..., t_n\}$, where m and n are the total number of users and items, respectively. It includes two types of information, an incomplete rating matrix $\mathbf{R} \in \mathbb{R}^{m \times n}$ and the user-user social graph $\mathcal{G}_s = \{\mathcal{U}, \mathcal{E}_s\}$. The rating $\mathbf{R}_{u,t}$ denotes user u's preference to item t, where higher score is interpreted as more preferring. An social edge $(u_i, u_j) \in \mathcal{E}_s$ indicates that user i has an social connection with j, *e.g.*, trust.

The objective of social recommendation is to complete the rating matrix by fusing the rating matrix and the social graph. Therefore, we solve social recommendation problem by constructing a heterogeneous graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}_r|_{r=1}^{R}\}$, where $\mathcal{V}$ denotes both user and item nodes and $\mathcal{E}_r$ denotes edges upon relation r. Besides user-user and item-item links, we also distinguish the user-item links by their rating scores. The score set varies on different datasets. *E.g.*, Ciao dataset [30] has 6 rating values, *i.e.*, $\{0, 1, 2, 3, 4, 5\}$. Hence, the edges on Ciao have 8 types, *i.e.*, $R = 8$, one being social relation, one being item-item relation and the others being different rating values.

3 PROPOSED MODEL

The framework of ConsisRec is shown in Figure 1(b)[1]. It has *embedding layer*, *query layer*, *neighbor sampling* and *relation attention.*

[1]Although we only illustrate the one-layer version of ConsisRec in the figure, our model is flexible to have L layers as introduced in the following section.

3.1 Embedding Layer

Following existing works [10, 33], we maintain an embedding layer $\mathbf{E} \in \mathbb{R}^{d \times (m+n)}$, each column of which represents the trainable embedding for each node. We can index it to retrieve the embedding of a node $v \in \mathcal{U} \cup \mathcal{T}$ as $\mathbf{e}_v \in \mathbb{R}^d$. In the following sections, without specific statements, we use an index v to denote a node, which can either be a user or an item, while u and t specifically denote a user and an item node. Apart from node embeddings, we also train a relation embedding vector for each relation r to characterize *relation-level* social inconsistency, denoted as $\mathbf{e}_r$.

3.2 Query Layer

To overcome the social inconsistency problem, we should aggregate consistent neighbors to learn node embeddings. Since social inconsistency are both in context-level and relation-level, we should distinguish consistent neighbors for each pair (u, t). Therefore, ConsisRec employs a query layer to exclusively select consistent neighbors for the query pair (u, t). It generates a query embedding by mapping the concatenation of user and item embeddings:

$$\mathbf{q}_{u,t} = \sigma\left(\mathbf{W}_q^\top (\mathbf{e}_u \oplus \mathbf{e}_t)\right), \tag{1}$$

where $\mathbf{q}_{u,t}$ is the query embedding , $\mathbf{e}_u, \mathbf{e}_t \in \mathbb{R}^d$ are the embedding for node u and t, respectively, $\oplus$ denotes concatenation, $\mathbf{W}_q \in \mathbb{R}^{2d \times d}$ is the mapping matrix, and σ is a ReLU activation function. We design a query layer to dynamically sample neighbors based on different items. It is because when users buy different items, they would inquire different friends. Thus, u's rating score of t is related to friends who are familiar with this query item t.

3.3 Neighbor Sampling

Neighbor sampling has been applied to GNN to boost training [3, 4, 38] and improve ranking performance [25]. Compared with previous work, ConsisRec aims to deal with the inconsistency problem in social recommendation, and dynamic samples different social neighbors based on different items. Next, we present how to sample neighbors for learning the embedding of u and t. The framework of ConsisRec to aggregate node embeddings can be formalized as:

$$\mathbf{h}_v^{(l)} = \sigma\left(\mathbf{W}^{(l)\top}\left(\mathbf{h}_v^{(l-1)} \oplus \mathrm{AGG}^{(l)}\{\mathbf{h}_i^{(l-1)} | i \in \mathcal{N}_v\}\right)\right), \tag{2}$$

where σ is a ReLU activation function, $\mathbf{h}_v^{(l)} \in \mathbb{R}^d$ is the hidden embedding of node v at l-th layer, $\mathcal{N}_v$ is the sampled neighbors of node v, AGG is an aggregation function, and $\mathbf{W}^{(l)} \in \mathbb{R}^{2d \times d}$ is the mapping function. $\mathbf{h}_v^{(0)}$ is the initial node embedding of v, *i.e.*, $\mathbf{e}_v$.

Instead of equally aggregating all neighbors, we should emphasize more on consistent neighbors while ignoring those inconsistent neighbors. Therefore, we propose to use neighbor sampling method to select those consistent neighbors. The sampling probability for neighbor node i at l-th layer is defined by the consistency score between query $\mathbf{q}$ and all the neighbors as:

$$p^{(l)}(i; \mathbf{q}) = s^{(l)}(i; \mathbf{q}) / \sum_{j \in \mathcal{N}_v} s^{(l)}(j; \mathbf{q}). \tag{3}$$

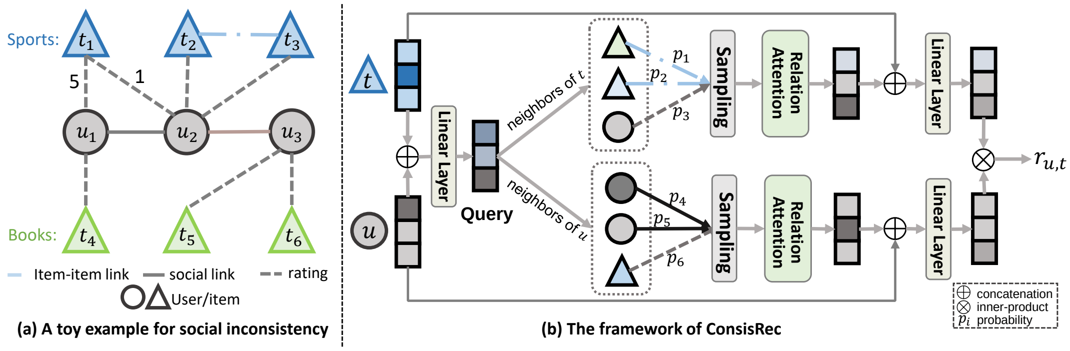

Figure 1: A toy example reflecting social inconsistency and the framework of ConsisRec.

where $s^{(l)}(i;\mathbf{q})$ denotes the consistency score between the neighbor i and the query $\mathbf{q}$ in l-th GNN layer. It is defined as:

$$s^{(l)}(i;\mathbf{q}) = \exp(-\|\mathbf{q} - \mathbf{h}_i^{(l)}\|_2^2), \tag{4}$$

where $\mathbf{h}_i^{(l)}$ denotes the node embedding of node i at l-th layer. For both nodes u and t, during the inference of rating score, we use the same query embedding. Thus, we ignore the subscript and write it as $\mathbf{q}$ for simplicity. We present this process as the sampling blocks in Figure 1(b), where the probabilities for neighbors are denoted as p_i. The number of sampled neighbors is proportional to the total number of neighbors, where the ratio is $0 \le \gamma \le 1$. As such, we sample more neighbors if a node is connected to more nodes.

3.4 Relation Attention

After sampling the neighbors, we should aggregate their embeddings as illustrated in Eq. (2). However, the *relation-level* social inconsistency suggests that we should distinguish different relations. To this end, we apply a relation attention module in ConsisRec for those sampled neighbors. It learns the importance of those sampled nodes by considering the associated relations.

The relation attention assigns an importance factor α_i for each sampled node i. We can rewrite the AGG function in Eq. (2) as:

$$\mathrm{AGG}^{(l)} = \sum_{i=1}^{Q} \alpha_i^{(l)} \cdot \mathbf{h}_i^{(l-1)}, \tag{5}$$

where $\alpha_i^{(l)}$ is the importance of the i-th neighbor sampled from Eq. (3) and Q denotes the total number of sampled neighbors. Assuming the relation for the edge (v, i) is r_i, we calculate α_i by adopting the self-attention mechanism as:

$$\alpha_i^{(l)} = \frac{\exp(\mathbf{w}_a^\top (\mathbf{h}_i^{(l-1)} \oplus \mathbf{e}_{r_i}))}{\sum_{j=1}^{Q} \exp(\mathbf{w}_a^\top (\mathbf{h}_j^{(l-1)} \oplus \mathbf{e}_{r_j}))} \tag{6}$$

where $\mathbf{e}_{r_i} \in \mathbb{R}^d$ represents the relation embedding of relation r_i and $\mathbf{w}_a \in \mathbb{R}^{2d}$ is trainable parameter for the self-attention layer and α_i is the attention weights. We illustrate the relation attention as the green block in Figure 1(b).

3.5 Rating Prediction and Optimization

After L layer propagation, we obtain the embedding of u and t, which are denoted as $\mathbf{h}_u^{(L)}$ and $\mathbf{h}_t^{(L)}$. We calculate the rating score of the user-item pair (u, t) by the inner-product of embeddings as:

$$\hat{R}_{u,t} = \mathbf{h}_u^{(L)} \cdot \mathbf{h}_t^{(L)}. \tag{7}$$

Then the loss function is defined as the Root Mean Squared Error (RMSE) between $\hat{R}_{u,t}$ and ground truth rating score $R_{u,t}$ among all (u, t) pairs in $\mathcal{E}_{\text{rating}}$, which is calculated as

$$\mathcal{L} = \sqrt{\frac{\sum_{(u,t)\in\mathcal{E}_{\text{rating}}} (R_{u,t} - \hat{R}_{u,t})^2}{|\mathcal{E}_{\text{rating}}|}}, \tag{8}$$

where $\mathcal{E}_{\text{rating}}$ is the set of all rating edges. We use Adam [12] as the optimizer with a weight decay rate of 0.0001 to avoid over-fitting.

4 EXPERIMENTS

4.1 Experimental Setup

4.1.1 Datasets. Ciao and Epinions [2] are two representative datasets [28–31] for studying social recommendation problem. We remove users without social links because they are out of social recommendation scope. Ciao has 7,317 users, 104,975 items with 111,781 social links. Epinions has 18,069 users, 261,246 items with 355,530 social links. We also linked items that share more than 50% of their neighbors.

4.1.2 Baselines. To justify the effectiveness of ConsisRec, we compare ConsisRec with 6 baseline methods, including matrix factorization methods, non-GNN graph embedding methods, and GNN-based methods. SoRec [22], SocialMF [11] and SoReg [23] incorporate social links with matrix factorization methods. CUNE [39] adopts collaborative graph embedding methods. GCMC+SN [1] and GraphRec [6] employ GNNs for learning node embeddings.

4.1.3 Evaluation Metrics. To evaluate the quality of the social recommendation, two common metrics, Mean Absolute Error (MAE) and Root Mean Square Error (RMSE), are adopted for the rating prediction task [6]. Note that lower values of both indicate better performance. And a small improvement in both may have a significant impact on the quality of top-N recommendation [14].

[2]https://www.cse.msu.edu/~tangjili/datasetcode/truststudy.htm

4.1.4 *Experimental Settings.* Each dataset is randomly split to 60%, 20%, and 20% for the training, validation, and testing, respectively. The grid search is applied for hyper-parameters tuning. We searched neighbor percent in $\{0.2, 0.4, 0.6, 0.8, 1.0\}$. For embedding size, we search in $\{8, 16, 32, 64, 128, 256\}$. The learning rate is searched in $\{0.0005, 0.001, 0.005, 0.01, 0.05, 0.1\}$. The batch size is searched in $\{32, 64, 128, 256, 512\}$. Only one GNN layer is used for both Ciao and Epinions datasets. To cope with the over-fitting problem, early stopping was utilized in all experiments, *i.e.,* stop training if the RMSE on the validation set is not improved for five epochs.

4.2 Performance Evaluation

Table 1: Overall comparison. The best and the second-best results are in bold and underlined, respectively.

Method	Ciao		Epinions	
	RMSE	MAE	RMSE	MAE
SoRec	1.2024	0.8693	1.3389	1.0618
SoReg	1.0066	0.7595	1.0751	0.8309
SocialMF	1.0013	0.7535	1.0706	0.8264
GCMC+SN	1.0301	0.7970	1.1070	0.8480
GraphRec	1.0040	0.7591	1.0799	0.8219
CUNE	1.0002	0.7591	1.0681	0.8284
ConsisRec	**0.9722**	**0.7394**	**1.0495**	**0.8046**
Improvement	2.79%	1.87%	1.74%	2.1%

The experiment results of all the methods are shown in Table 1. GCMC, GraphRec, CUNE and ConsisRec perform better than SoRec, SoReg and SocialMF, which shows GNN and graph embedding based methods have a better capability to aggregate neighbor information. ConsisRec achieves the best results on both Ciao and Epinions datasets. It has an 1.7% relative improvement on two datasets compared with the second-best one on average, which can be interpreted as a significant improvement [6]. The results show the benefits brought by tackling the social inconsistency problems.

4.3 Ablation Study

An ablation study is further made to evaluate different components in ConsisRec. We create three variants of ConsisRec, which are A, B, and C. A is built by removing the query layer, which directly uses user embedding instead of query embedding to select the corresponding neighbors. B is built by removing neighbor sampling, which aggregates all neighbors. C is built by removing relation attention, which assigns equal weights to edges with different relations. The experimental results are illustrated in Figure 2.

We can observe that ConsisRec consistently achieves the best performance against other variants, demonstrating that all components are necessary to yield the best results. Additionally, we observe that the variant *B* (removing neighbor sampling module) dramatically spoils the performance, which justifies the importance of selecting consistent neighbors. The worse performance of variant A and C compared with ConsisRec also proves the importance of query layer and relation attention, respectively.

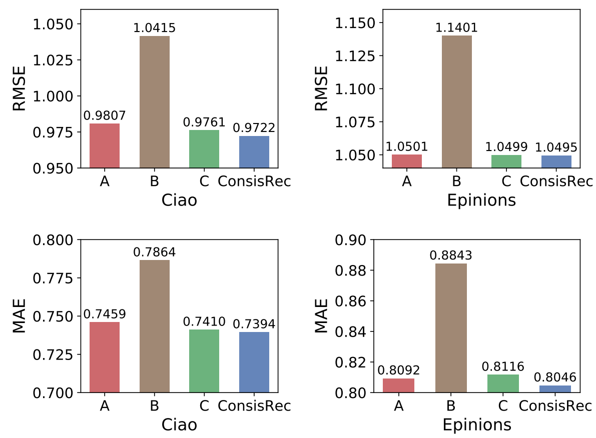

Figure 2: Ablation study of ConsisRec.

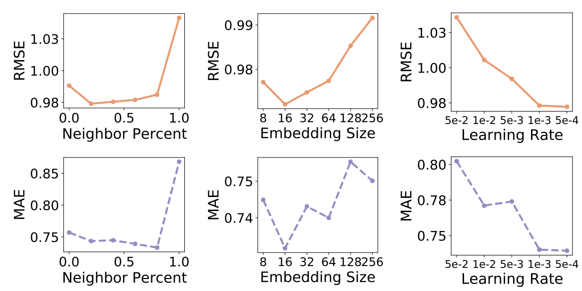

Figure 3: Parameter sensitivity on Ciao dataset.

4.4 Parameter Sensitivity

Influential hyper-parameters in ConsisRec includes neighbor percent, embedding size and the learning rate. Due to space limitation, we only represent results on Ciao dataset in Figure 3. For neighbor percent, we observe an obvious error increment when neighbor percent rising from 0.8 to 1.0, which results from aggregating those inconsistent neighbors. The best embedding size on Ciao is 16. Smaller embedding size is insufficient for representing node information, while large embedding size would lead to the over-fitting problem. Learning rate has a critical impact on model performance, which needs to be tuned carefully.

5 CONCLUSION AND FUTURE WORK

In this paper, we identify the social inconsistency problem related to social recommendation. The proposed ConsisRec contains three modifications on GNN to tackle the social inconsistency problem. Experiment results on two real-world datasets show the effectiveness of ConsisRec. Future work includes better ways to filter informative neighbors and identify the inconsistency problems inherited in other graph related research directions.

6 ACKNOWLEDGEMENTS

This work is supported in part by NSF under grants III-1763325, III-1909323, and SaTC-1930941.

REFERENCES

[1] Rianne van den Berg, Thomas N Kipf, and Max Welling. 2017. Graph convolutional matrix completion. *arXiv preprint arXiv:1706.02263* (2017).

[2] Lorenzo Bizzi and Alice Labban. 2019. The double-edged impact of social media on online trading: Opportunities, threats, and recommendations for organizations. *Business Horizons* 62, 4 (2019), 509–519.

[3] Wei-Lin Chiang, Xuanqing Liu, Si Si, Yang Li, Samy Bengio, and Cho-Jui Hsieh. 2019. Cluster-gcn: An efficient algorithm for training deep and large graph convolutional networks. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 257–266.

[4] Weilin Cong, Rana Forsati, Mahmut Kandemir, and Mehrdad Mahdavi. 2020. Minimal variance sampling with provable guarantees for fast training of graph neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1393–1403.

[5] Yingtong Dou, Zhiwei Liu, Li Sun, Yutong Deng, Hao Peng, and Philip S Yu. 2020. Enhancing graph neural network-based fraud detectors against camouflaged fraudsters. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 315–324.

[6] Wenqi Fan, Yao Ma, Qing Li, Yuan He, Eric Zhao, Jiliang Tang, and Dawei Yin. 2019. Graph neural networks for social recommendation. In *The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019*. ACM, 417–426.

[7] Wenqi Fan, Yao Ma, Qing Li, Jianping Wang, Guoyong Cai, Jiliang Tang, and Dawei Yin. 2020. A Graph Neural Network Framework for Social Recommendations. *IEEE Transactions on Knowledge and Data Engineering* (2020).

[8] Wenqi Fan, Yao Ma, Dawei Yin, Jianping Wang, Jiliang Tang, and Qing Li. 2019. Deep social collaborative filtering. In *Proceedings of the 13th ACM Conference on Recommender Systems*. 305–313.

[9] Zhiwei Guo and Heng Wang. 2021. A Deep Graph Neural Network-based Mechanism for Social Recommendations. *IEEE Transactions on Industrial Informatics* 17, 4 (2021), 2776–2783.

[10] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, YongDong Zhang, and Meng Wang. 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* (Virtual Event, China) *(SIGIR '20)*. Association for Computing Machinery, New York, NY, USA, 639–648. https://doi.org/10.1145/3397271.3401063

[11] Mohsen Jamali and Martin Ester. 2010. A matrix factorization technique with trust propagation for recommendation in social networks. In *Proceedings of the 2010 ACM Conference on Recommender Systems, RecSys 2010, Barcelona, Spain, September 26-30, 2010*. ACM, 135–142.

[12] Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

[13] Thomas N Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

[14] Yehuda Koren. 2008. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Las Vegas, Nevada, USA, August 24-27, 2008*. ACM, 426–434.

[15] Munan Li, Kenji Tei, and Yoshiaki Fukazawa. 2020. An Efficient Adaptive Attention Neural Network for Social Recommendation. *IEEE Access* 8 (2020), 63595–63606.

[16] Jovian Lin, Kazunari Sugiyama, Min-Yen Kan, and Tat-Seng Chua. 2013. Addressing cold-start in app recommendation: latent user models constructed from twitter followers. In *The 36th International ACM SIGIR conference on research and development in Information Retrieval, SIGIR '13, Dublin, Ireland - July 28 - August 01, 2013*. ACM, 283–292.

[17] Zhiwei Liu, Yingtong Dou, Philip S Yu, Yutong Deng, and Hao Peng. 2020. Alleviating the Inconsistency Problem of Applying Graph Neural Network to Fraud Detection. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*. ACM, 1569–1572.

[18] Zhiwei Liu, Ziwei Fan, Yu Wang, and Philip S. Yu. 2021. Augmenting Sequential Recommendation with Pseudo-PriorItems via Reversely Pre-training Transformer. *Proceedings of the 44th international ACM SIGIR conference on Research and development in information retrieval*.

[19] Zhiwei Liu, Xiaohan Li, Ziwei Fan, Stephen Guo, Kannan Achan, and Philip S Yu. 2020. Basket Recommendation with Multi-Intent Translation Graph Neural Network. *arXiv preprint arXiv:2010.11419* (2020).

[20] Zhiwei Liu, Mengting Wan, Stephen Guo, Kannan Achan, and Philip S Yu. 2020. Basconv: aggregating heterogeneous interactions for basket recommendation with graph convolutional neural network. In *Proceedings of the 2020 SIAM International Conference on Data Mining*. SIAM, 64–72.

[21] Zhiwei Liu, Lei Zheng, Jiawei Zhang, Jiayu Han, and S Yu Philip. 2019. JSCN: Joint spectral convolutional network for cross domain recommendation. In *2019 IEEE International Conference on Big Data (Big Data)*. IEEE, 850–859.

[22] Hao Ma, Haixuan Yang, Michael R Lyu, and Irwin King. 2008. Sorec: social recommendation using probabilistic matrix factorization. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management, CIKM 2008, Napa Valley, California, USA, October 26-30, 2008*. ACM, 931–940.

[23] Hao Ma, Dengyong Zhou, Chao Liu, Michael R Lyu, and Irwin King. 2011. Recommender systems with social regularization. In *Proceedings of the Forth International Conference on Web Search and Web Data Mining, WSDM 2011, Hong Kong, China, February 9-12, 2011*. ACM, 287–296.

[24] Nan Mu, Daren Zha, Yuanye He, and Zhihao Tang. 2019. Graph Attention Networks for Neural Social Recommendation. In *31st IEEE International Conference on Tools with Artificial Intelligence, ICTAI 2019, Portland, OR, USA, November 4-6, 2019*. IEEE, 1320–1327.

[25] Marc Najork, Sreenivas Gollapudi, and Rina Panigrahy. 2009. Less is more: sampling the neighborhood graph makes salsa better and faster. In *Proceedings of the Second ACM International Conference on Web Search and Data Mining*. 242–251.

[26] Guang Fu Sun, Le Wu, Qi Liu, Chen Zhu, and En Hong Chen. 2013. Recommendations based on collaborative filtering by exploiting sequential behaviors. *Journal of Software* 24, 11 (2013), 2721–2733.

[27] Peijie Sun, Le Wu, and Meng Wang. 2018. Attentive recurrent social recommendation. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018, Ann Arbor, MI, USA, July 08-12, 2018*. ACM, 185–194.

[28] Jiliang Tang, Huiji Gao, Xia Hu, and Huan Liu. 2013. Exploiting homophily effect for trust prediction. In *Sixth ACM International Conference on Web Search and Data Mining, WSDM 2013, Rome, Italy, February 4-8, 2013*. ACM, 53–62.

[29] Jiliang Tang, Huiji Gao, and Huan Liu. 2012. mTrust: Discerning multi-faceted trust in a connected world. In *Proceedings of the Fifth International Conference on Web Search and Web Data Mining, WSDM 2012, Seattle, WA, USA, February 8-12, 2012*. ACM, 93–102.

[30] Jiliang Tang, Huiji Gao, Huan Liu, and Atish Das Sarma. 2012. eTrust: Understanding trust evolution in an online world. In *The 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '12, Beijing, China, August 12-16, 2012*. ACM, 253–261.

[31] Jiliang Tang, Xia Hu, Huiji Gao, and Huan Liu. 2013. Exploiting local and global social context for recommendation. In *IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, August 3-9, 2013*. IJCAI/AAAI, 2712–2718.

[32] Le Wu, Junwei Li, Peijie Sun, Richang Hong, Yong Ge, and Meng Wang. 2020. DiffNet++: A Neural Influence and Interest Diffusion Network for Social Recommendation. *IEEE Transactions on Knowledge and Data Engineering* (2020).

[33] Le Wu, Peijie Sun, Yanjie Fu, Richang Hong, Xiting Wang, and Meng Wang. 2019. A neural influence diffusion model for social recommendation. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019, Paris, France, July 21-25, 2019*. ACM, 235–244.

[34] Le Wu, Peijie Sun, Richang Hong, Yanjie Fu, Xiting Wang, and Meng Wang. 2018. SocialGCN: An efficient graph convolutional network based model for social recommendation. *arXiv preprint arXiv:1811.02815* (2018).

[35] Le Wu, Peijie Sun, Richang Hong, Yong Ge, and Meng Wang. 2021. Collaborative neural social recommendation. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 51, 1 (2021), 464–476.

[36] Qitian Wu, Hengrui Zhang, Xiaofeng Gao, Peng He, Paul Weng, Han Gao, and Guihai Chen. 2019. Dual graph attention networks for deep latent representation of multifaceted social effects in recommender systems. In *The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019*. ACM, 2091–2102.

[37] Yang Xiao, Lina Yao, Qingqi Pei, Xianzhi Wang, Jian Yang, and Quan Z Sheng. 2020. MGNN: Mutualistic Graph Neural Network for Joint Friend and Item Recommendation. *IEEE Intelligent Systems* 35, 5 (2020), 7–17.

[38] Hanqing Zeng, Hongkuan Zhou, Ajitesh Srivastava, Rajgopal Kannan, and Viktor Prasanna. 2019. Graphsaint: Graph sampling based inductive learning method. *arXiv preprint arXiv:1907.04931* (2019).

[39] Chuxu Zhang, Lu Yu, Yan Wang, Chirag Shah, and Xiangliang Zhang. 2017. Collaborative User Network Embedding for Social Recommender Systems. In *Proceedings of the 2017 SIAM International Conference on Data Mining, Houston, Texas, USA, April 27-29, 2017*. SIAM, 381–389.