# A Heterogeneous Graph Neural Model for Cold-start Recommendation

Siwei Liu
s.liu.4@research.gla.ac.uk
University of Glasgow
Glasgow, Scotland

Iadh Ounis, Craig Macdonald and Zaiqiao Meng
first.lastname@glasgow.ac.uk
University of Glasgow
Glasgow, Scotland

## ABSTRACT

The users' historical interactions usually contain their interests and purchase habits based on which personalised recommendations can be made. However, such user interactions are often sparse, leading to the well-known *cold-start* problem when a user has no or very few interactions. In this paper, we propose a new recommendation model, named *Heterogeneous Graph Neural Recommender* (HGNR), to tackle the *cold-start* problem while ensuring effective recommendations for all users. Our HGNR model learns users and items' embeddings by using the Graph Convolutional Network based on a heterogeneous graph, which is constructed from user-item interactions, social links and semantic links predicted from the social network and textual reviews. Our extensive empirical experiments on three public datasets demonstrate that HGNR significantly outperforms competitive baselines in terms of the Normalised Discounted Cumulative Gain and Hit Ratio measures.

**ACM Reference Format:**
Siwei Liu, Iadh Ounis, Craig Macdonald and Zaiqiao Meng. 2020. A Heterogeneous Graph Neural Model for Cold-start Recommendation . In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '20), July 25–30, 2020, Virtual Event, China.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3397271.3401252

## 1 INTRODUCTION

Recommender systems lie at the heart of many online services such as e-commerce, social media platforms and advertising. The key functionality of a recommender system is to predict the items the users will find interesting among the large corpus of existing items. Among all recommendation models, latent factor-based models are the indispensable building blocks of effective recommendations [5, 11]. However, their effectiveness are commonly limited by the sparsity of user interactions, which leads to the well-known *cold-start* problem. To alleviate this problem, many existing approaches [6, 8, 9, 16] have used side information, especially the users' social relationships and reviews to enhance recommendation effectiveness. For example, Ma et al. proposed SoReg [6] to regularise the matrix factorisation model by leveraging social information. Similarly, Manotumruksa et al. [8] proposed to model the reviews left by users within a word2vec embedding space and regularised the model according to the similarities between the items reviews. Although such approaches are in general effective, it is worthwhile to note that the data sparsity issue exists not only in the user-item interactions but also in the side information. For example, most of the users are not included in the existing social networks. Consequently, incorporating the sparse side information does not adequately resolve the *cold-start* problem.

To enrich the side information available to the recommender system, we propose to predict social links between users based on existing social relationships. Furthermore, we also use a BERT-based language model [10] to explore the potential links between items based on reviews. Using these predicted links, as well as any existing interaction links and social links, we can build a heterogeneous graph. To effectively learn users and items' embeddings from this heterogeneous graph, we use the recently proposed Graph Neural Networks (GNNs). GNNs can effectively encode the local structure and propagate information from the nodes' local neighbours. Hence, by employing the multi-layer GNNs on our built heterogeneous graph, we encode all the interaction information, social information and users' reviews information into the users and items' embeddings.

To summarise, our contributions are threefold: 1) We propose to predict links between users based on the existing social information and links between items based on the users' reviews to alleviate the data sparsity issues; 2) We use the actual interaction links and the predicted links to build a heterogeneous graph. We then propose the HGNR model, which uses the GNNs technique to learn from this heterogeneous graph so that, the users and items' embeddings are generated for a pair-wise ranking-based recommendation framework; 3) We conduct extensive experiments on three public datasets, and show that our HGNR model can outperform strong baselines while especially alleviating the *cold-start* problem. The remainder of this paper is organised as follows. In Section 2, we position our proposed model in the literature. Section 3 details the architecture of our model and how to construct the heterogeneous graph. The experimental setup and the results of our empirical experiments are presented in Section 4, followed by concluding remarks in Section 5.

## 2 RELATED WORK

Collaborative Filtering (CF) models parameterise users/items as embeddings learned from the historical interaction data. Predictions are then made using the dot product of these embeddings. While Matrix Factorisation (MF) [5] attempts to predict explicit user ratings, BPR [11] and iMF [2] both use the more abundant implicit feedback and optimise their models with a pair-wise optimisation and an alternating-least-square optimisation, respectively. To enhance the recommendation accuracy, Ma et al. proposed SoReg [6] to incorporate the social information. Similarly, Manotumruksa et al. [8] proposed to regularise the model with the reviews left by users. However, limited by the linearity of the inner product, the traditional CF models mentioned above cannot capture the


*SIGIR '20, July 25–30, 2020, Virtual Event, China*

ACM ISBN 978-1-4503-8016-4/20/07…$15.00
https://doi.org/10.1145/3397271.3401252

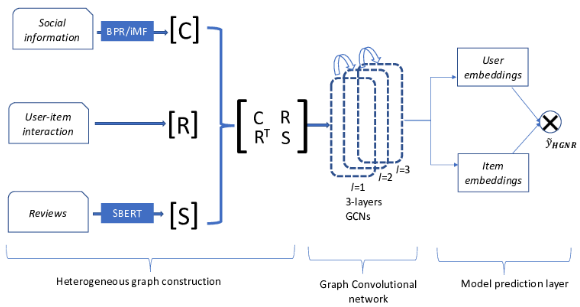


**Figure 1: An illustration of HGNR model architecture where [*R*], [*S*], [*C*] are three sub-graphs, containing the user-item interaction and predicted social and semantic links.**

non-linearity in the features. Therefore, recent deep learning based recommenders focus on modelling the complex relationship between users and items. For example, NeuMF [1] replaces the inner product with a non-linear neural network.

Distinct from the aforementioned models, recent graph-based recommender systems apply GNN techniques to generate embeddings learned from the user-item bipartite graph. For instance, GC-MC [12] employs the GCN technique to learn from the user-item interaction bipartite graph. However, only one layer of GCN is used in GC-MC, which stops it from capturing the high-order connectivities. Furthermore, Wang et al. [13] proposed the NGCF model to enhance performance by stacking three GCN layers. Wu et al. proposed Diffnet [14], which simulates the social influence by applying high-order GCNs to learn from the social network graph. However, Diffnet solely used any existing social links to modify the users' embeddings, which means that the items' embeddings cannot benefit from the GCNs. Different from these GNN approaches, our HGNR model employs three-layer GCNs to learn from a heterogeneous graph containing the classical interaction links together with the predicted links between users and items based on the existing social network and users' reviews. Therefore, in this paper, we advance the existing GCN-based models to extract both users and items' embeddings from not only the existing user-item bipartite graph structure but also from the inferred links between users and items.

## 3 MODEL ARCHITECTURE

In this section, we describe our proposed model, the architecture of which is illustrated in Figure 1. We firstly define all components and the input/output of our model followed by the construction of the heterogeneous graph. Next, we mathematically show how to generate embeddings and train the model.

### 3.1 Preliminaries

We consider a recommender system with a user set $U$ ($|U| = M$) and an item set $I$ ($|I| = N$). Let $\boldsymbol{R} \in \mathbb{R}^{M \times N}$ be the user-item interaction matrix, where the content of the matrix $\boldsymbol{R}^{M \times N}$ corresponds to explicit user ratings [5] or implicit feedback [2, 11]. We consider implicit feedback here because it is more abundant, therefore, $R_{ui} = 1$ if the user $u$ has interacted with the item $i$, otherwise $R_{ui} = 0$. We use $\boldsymbol{R}$ as one sub-graph of our heterogeneous user-item graph $\boldsymbol{H}$. Furthermore, we define two other sub-graphs of $\boldsymbol{H}$: the social sub-graph $\boldsymbol{S}$ and the review sub-graph $\boldsymbol{C}$. These three sub-graphs are defined through their types of links: an interaction link exists in $\boldsymbol{R}$ between a user $u$ and item $i$ if $R_{ui} = 1$; a social link exists in $\boldsymbol{S}$ between two users if they are socially connected in a social network; a semantic link exists in $\boldsymbol{C}$ between two items if they have obatined similar reviews from users. Our HGNR model takes the heterogeneous graph $\boldsymbol{H}$ as input, then obtains users and items' embeddings $\boldsymbol{E^l} \in R^{M \times D}$ through $l$ GCN layers, before finally making predictions.

### 3.2 Heterogeneous Graph Construction

As mentioned in Section 1, our heterogeneous graph considers three types of links between the users and items' nodes. The first type of links contains the user-item interaction information. The second type of links corresponds to the social links between users. The third type of links relates to links between items based on the semantic similarities of the reviews left by users for each item. The user-item interactions are commonly used to generate users and items' embeddings for recommendation [1, 2, 11, 13]. Here, we use the social links and semantic links to incorporate the social network information and we use the review information to facilitate the embedding generation process. Since the sub-graph **R** is available, we focus on how to generate the social sub-graph **S** and the review sub-graph **C**.

The social sub-graph **S** is a user-by-user square matrix initially filled by any existing social links between users. We use the existing social links to enhance the user embedding generation because one user's preferences can be affected by other users through social influence [6, 14]. However, in the most widely used item recommendation datasets, the social graph is usually sparse, since most of the users are not included in the social graph [16]. Hence, to incorporate more users, we use the existing social network to further predict additional links between the users. In this paper, we choose BPR [11] and iMF [2], two classical matrix factorisation methods, to predict social links, since they have been shown to be effective in social recommendations [15].

Reviews left by users can contain a rich semantic meaning, which can be used to exploit potential links between items and generate the review sub-graph **C**. For example, users might mention what type of item it is and how much they appreciate this item in their reviews. Such reviews could be used to extract useful information for item embedding generation purposes. For each item, we firstly concatenate its reviews if it has more than one review. Secondly, to extract the semantic information, we use the Sentence-BERT (SBERT) [10] model to generate fixed-size vector representations for each item. Finally, based on these vector representations, we measure the cosine similarities between all items and generate links between those items with the top-K cosine values.

With all sub-graphs now formally defined, we introduce our heterogeneous graph $\boldsymbol{H}$ as shown below:

$$\boldsymbol{H} = \begin{bmatrix} \alpha S & \boldsymbol{R} \\ \boldsymbol{R}^T & \beta C \end{bmatrix} \tag{1}$$

where $\alpha$ and $\beta$ are social and item parameters that control the contributions of the social and semantic links, respectively.

### 3.3 Embedding Generation

In this section, we explain how our proposed HGNR model uses the GCN technique to generate users and items' embeddings containing social and review information learnt from the heterogeneous graph $\boldsymbol{H}$ defined in Equation (1).

The graph Laplacian method is an effective technique to learn the nodes' embeddings from a graph structure [4, 9, 13]. Hence, to generate users and items' embeddings from $\boldsymbol{H}$, we also adopt this technique. As a consequence, we firstly need to compute the graph Laplacian matrix $\boldsymbol{\mathcal{L}}^{c}$ of the heterogeneous graph $\boldsymbol{H}$:

$$\mathcal{L}^{c} = D^{-\frac{1}{2}} \boldsymbol{H} D^{\frac{1}{2}} \tag{2}$$

where D is the diagonal degree matrix of $\boldsymbol{H}$.

Based on the graph Laplacian matrix $\boldsymbol{\mathcal{L}}^{c}$, we can now present the core embedding updating function of our HGNR model:

$$\begin{aligned} \mathrm{HGNR}(\boldsymbol{H})^{(l)} &= \mathrm{E}^{(l)} \\ &= \sigma\Big((\mathcal{L}^{c} + \mathbf{I})E^{(l-1)}W_o^{(l)} + \mathcal{L}^{c}E^{(l-1)} \odot E^{(l-1)}W_n^{(l)}\Big) \end{aligned}$$

where $\boldsymbol{H}$ is the input of HGNR, $\mathbf{E}^{(l)}$ is the concatenation of all users and items' embeddings obtained after $l$ graph convolutional layers, $\sigma(\cdot)$ is the LeakyReLU [7] activation function as used in other graph-based recommenders [12, 13], and $\mathbf{I}$ is the identity matrix. $\boldsymbol{W_o^{(l)}}, \boldsymbol{W_n^{(l)}} \in R^{d' \times d}$ are trainable weight matrices where $\boldsymbol{W_o^{(l)}}$ preserves the original information for each user and item's embedding and $\boldsymbol{W_n^{(l)}}$ aggregates the information from their neighbours.

Through the multi-layer GCNs, we obtain multiple embeddings from each layer, then we concatenate each $\mathbf{E}^{(l)}$, so that the final embedding collectively contains information from each GCN layer. We denote these concatenated user/item embeddings as $\boldsymbol{e_u^*}$ & $\boldsymbol{e_i^*}$. Therefore, we can predict the likelihood of an interaction for a user $u$ over a target item $i$ as a product of their concatenated embeddings:

$$\widehat{y}_{ui} = \boldsymbol{e_u^*}^{\mathrm{T}} \boldsymbol{e_i^*} \tag{3}$$

## 3.4 Model Training

Since we focus on implicit feedback, we adopt the most widely used pair-wise ranking-based loss function of BPR [11] to optimise the model's parameters. Hence, the loss function is defined as:

$$L(\Theta) = \sum_{(u,i,j)\in D_s} \ln\sigma(y_{ui} - \widehat{y}_{ui}) + \lambda \|\Theta\|_2^2 \tag{4}$$

where $\Theta$ denotes all parameters to be learnt, $D_s = \{(u,i,j)|(u,i) \in R^+, (u,j) \in R^-\}$ is the set of the training data, $R^+$ indicates the observed interactions and $R^-$ indicates the unobserved interactions, $\sigma(\cdot)$ is the sigmoid function, and $\lambda$ controls the $L_2$ regularisation.

# 4 EXPERIMENTS

We perform experiments on three public datasets: Yelp[1] (round 13), Librarything[2] and Epinions[2] to evaluate our proposed HGNR model. In this section, we describe these datasets, the experimental setup and the obtained results.

## 4.1 Datasets

The Yelp dataset is a popular venue check-in dataset, Librarything is a book review dataset, and Epinions is a general customer reviews dataset. Table 1 provides the statistics of the used datasets. For each dataset, we remove all users with only 1 interaction to avoid testing users that do not exist in the training set. Moreover, we only preserve items with at least one review since we rely on reviews to identify the links between items.

[1] https://yelp.com/dataset/challenge [2] http://cseweb.ucsd.edu/~jmcauley/datasets.html

**Table 1: Statistics of the datasets. *Regular* users are those with $\geq$ 10 historical interactions and *Cold-start* users are those with < 10 interactions.**

| | Yelp | Librarything | Epinions |
|---|---|---|---|
| Users | 215,471 | 53,246 | 27,545 |
| Items | 93,379 | 167,286 | 30,324 |
| Interactions | 3,906,039 | 1,334,648 | 91,739 |
| Social edges | 1,397,180 | 18,345 | 96,442 |
| Cold-start users | 198,017 | 36,260 | 26,608 |
| Regular users | 17,454 | 16,986 | 1,163 |
| Interaction density(%) | 0.0194 | 0.0150 | 0.003 |
| Social density(%) | 0.0030 | 0.0006 | 0.013 |

## 4.2 Experimental Setup

We compare our HGNR model with the following baselines: BPR [11], NeuMF [1], GC-MC [12], DiffNet [14] and NGCF [13]. Our experiments have two parts: Firstly we examine the social sub-graph and the review sub-graph separately, so that we can clearly demonstrate the effectiveness of each sub-graph; Secondly we compare our HGNR model with all baselines in terms of Normalised Discounted Cumulative Gain@10 (NDCG) and Hit Ratio@10 (HR).

As mentioned in Section 3.2, we use BPR [11][3] and iMF [2][4] to predict social links between users. In both cases, we use all existing social relationships as the training data. For the predicted links between items, we use the mean-tokens pooling BERT model provided by SBERT [10] to generate a 1×768 (default size) vector for each item. Based on these vector representations, the semantic relatedness between two items is computed by the cosine similarity. We define $n$ as the number of predicted links we add for each user and item where we vary $n$ from 0 to 20 at a step of 5.

We implement our HGNR model in Tensorflow. The latent dimensions and batch size are fixed to 64 and 1024 [1, 13], respectively, for all models. For each dataset, we use 20% of the interactions as a test set; of the remaining, we use 10% as a validation set, and the remainder for training. We use the Adam [3] optimiser for the model optimisation and parameters updating. To tune all hyper-parameters, we apply a grid search on the validation set, where both $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ vary amongst $\{10^0, 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}\}$; the learning rate is tuned in $\{10^{-2}, 10^{-3}, 10^{-4}\}$; and the $L_2$ normalisation in $\{10^{-1}, 10^{-2}, ..., 10^{-5}\}$. The node dropout technique is adopted in the NGCF model and our HGNR model and the ratios vary amongst $\{0.3, 0.4, ..., 0.8\}$ as suggested in [12]. For a fair comparison, we set the number of neural network layers of the models including NeuMF, NGCF and HGNR to 3. In addition, we use the win/loss ratio to demonstrate the model's effect on different groups of users.

## 4.3 Results

Figure 2 plots the performances of the social sub-graph and the review sub-graph independently in comparison with the best baseline: the NGCF model. It is clear from the figure that by incorporating the social sub-graph, our HGNR model can largely outperform NGCF in terms of NDCG@10 across all three used datasets. In particular, the blue/orange lines in Figure 2 show that adding the predicted social links from BPR and iMF does clearly enhance the recommendation performance in comparison to only using the existing social links (i.e. when $n = 0$), although this improvement tails off after

[3] https://github.com/maciejkula/spotlight [4] https://github.com/benfred/implicit

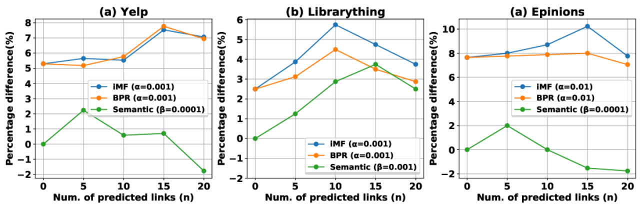


**Figure 2: Percentage comparisons between the social/review sub-graphs of the HGNR model and the NGCF model across different number of predicted links w.r.t. NDCG@10 on the 3 datasets. Similar trends are observed for HR@10.**

adding 10-15 predicted social links. On the other hand, the green lines in Figure 2 show that the review sub-graph also improves the recommendation performance, although such an improvement is not as marked as that obtained by the social sub-graph.

Table 2 reports the overall performances of our HGNR model in comparison to other baselines. The results show that by integrating the three sub-graphs, our HGNR model achieves the best overall performance, constantly and significantly outperforming all other baselines. All the evaluated models achieve their lowest performances on the Epinions dataset since this is the sparsest dataset with a sparsity of 0.003% in comparison with the Yelp dataset (0.0194%) and the Librarything dataset (0.015%). However, it is of note that our HGNR model achieves the largest percentage improvement over the best baseline, NGCF, on the Epinions dataset both in terms of HR and NDCG. This is likely because this dataset has the most abundant social information compared with the other two datasets.

Table 3 examines the performances of our HGNR model for the *cold-start* and *regular* users, respectively, in comparison to the best baseline, NGCF, in terms of NDCG@10. The results show that while all users benefit, our model benefits the *cold-start* users more than the *regular* users. Furthermore, the wins%/losses% ratios provide further details on the number of users that were given better recommendations by our HGNR model compared with NGCF. These results are overall reasonable since *regular* users have enough interaction information based on which representative user embeddings can be obtained, hence adding more auxiliary social information won't be that beneficial. However, for the *cold-start* users, their interaction information is too sparse, therefore adding more auxiliary social links will bring useful information about their possible preferences. Overall, we can conclude that our HGNR model allows to alleviate the *cold-start* problem, while ensuring effective recommendation for all users in comparison to strong baselines.

## 5 CONCLUSIONS

We proposed to alleviate the user interactions sparsity by predicting possible social links between users as well as semantic links between items based on existing social information and the reviews left by users. Furthermore, we devised a new recommendation model, HGNR, which learns users and items' embeddings by employing a multi-layer Graph Convolutional Network on a heterogeneous graph containing user-item interaction information as well as predicted side information. Extensive experiments on three public datasets showed that our HGNR model can significantly outperform competitive baselines. Moreover, the user analysis shows that by incorporating the predicted side information, our HGNR model successfully alleviates the *cold-start* problem while ensuring effective recommendations for all users.

**Table 2: Experimental results of HGNR and other baselines on the three used datasets w.r.t. HR@10 and NDCG@10. The best result is highlighted in bold; * denotes a significant difference compared to the best result (paired t-test, p<0.01).**

| | Yelp | | Librarything | | Epinions | |
|---|---|---|---|---|---|---|
| | NDCG | HR | NDCG | HR | NDCG | HR |
| BPR | 0.0367* | 0.0456* | 0.0779* | 0.0951* | 0.00606* | 0.00672* |
| NeuMF | 0.0345* | 0.0478* | 0.0788* | 0.0962* | 0.00739* | 0.00841* |
| GC-MC | 0.0354* | 0.0480* | 0.0795* | 0.0969* | 0.00794* | 0.00851* |
| DiffNet | 0.0366* | 0.0498* | 0.0785* | 0.0941* | 0.00841* | 0.00896* |
| NGCF | 0.0407* | 0.0512* | 0.0801* | 0.0977* | 0.00850* | 0.00955* |
| HGNR | **0.0442** | **0.0559** | **0.0863** | **0.1050** | **0.00945** | **0.01075** |
| %Improv. | 8.56 | 9.13 | 7.74 | 6.92 | 11.2 | 12.6 |

**Table 3: NDCG@10 of our HGNR model and the NGCF baseline (in parenthesis) across different user groups.**

| Dataset | | NDCG@10 | %Improv. | Wins%/losses% |
|---|---|---|---|---|
| Yelp | All | 0.0442 (0.0407) | 8.56 | 28.5%/19.7% |
| | Cold-start | 0.0377 (0.0342) | 10.2 | 30.5%/21.2% |
| | Regular | 0.1178 (0.1144) | 3.01 | 5.87%/2.43% |
| Librarything | All | 0.0863 (0.0801) | 7.74 | 20.0%/16.47% |
| | Cold-start | 0.0434 (0.0365) | 15.8 | 24.3%/18.7% |
| | Regular | 0.1779 (0.1733) | 2.65 | 10.8%/11.7% |
| Epinions | All | 0.0936 (0.0850) | 10.1 | 34.1%/20.3% |
| | Cold-start | 0.0920 (0.0832) | 10.6 | 34.4%/20.2% |
| | Regular | 0.1120 (0.1091) | 2.66 | 20.5%/18.7% |

## REFERENCES

[1] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proc. of WWW.*
[2] Yifan Hu, Yehuda Koren, and Chris Volinsky. 2008. Collaborative filtering for implicit feedback datasets. In *Proc. of ICDM.*
[3] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *Proc. of ICLR.*
[4] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *Proc. of ICLR.*
[5] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix Factorization Techniques for Recommender Systems. *Computer* 42 (2009).
[6] Hao Ma, Dengyong Zhou, Chao Liu, Michael R. Lyu, and Irwin King. 2011. Recommender Systems with Social Regularization. In *Proc. of WSDM.*
[7] Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. 2013. Rectifier nonlinearities improve neural network acoustic models. In *Proc. of ICML.*
[8] Jarana Manotumruksa, Craig Macdonald, and Iadh Ounis. 2016. Regularising factorised models for venue recommendation using friends and their comments. In *Proc. of CIKM.*
[9] Zaiqiao Meng, Shangsong Liang, Xiangliang Zhang, Richard McCreadie, and Iadh Ounis. 2020. Jointly Learning Representations of Nodes and Attributes for Attributed Networks. *TOIS* 38 (2020).
[10] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proc. of EMNLP.*
[11] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *Proc. of UAI.*
[12] Rianne van den Berg, Thomas N Kipf, and Max Welling. 2018. Graph Convolutional Matrix Completion. In *Proc. of KDD.*
[13] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural graph collaborative filtering. In *Proc. of SIGIR.*
[14] Le Wu, Peijie Sun, Yanjie Fu, Richang Hong, Xiting Wang, and Meng Wang. 2019. A neural influence diffusion model for social recommendation. In *Proc. of SIGIR.*
[15] Xiwang Yang, Yang Guo, Yong Liu, and Harald Steck. 2014. A survey of CF based social recommender systems. *Comput Commun* 41 (2014).
[16] Junliang Yu, Min Gao, Jundong Li, Hongzhi Yin, and Huan Liu. 2018. Adaptive implicit friends identification over heterogeneous network for social recommendation. In *Proc. of CIKM.*