Feature Disentanglement and Homogeneous Second Order Feature Propagation for Recommendation

Yaru Zhang and Xijin Tang

***Abstract*— Personalized recommendation systems mine user preference, represent item feature and thus realize recommendation by modeling user-item interaction. Because the interaction naturally forms the bipartite graph, Graph Convolution Neural Networks are used to learn representation of nodes for recommendation recently. Current approaches, however, seldom give recommendation results from causing factors. Furthermore, they may encounter over-smoothing problem when addressing dense graphs and struggle to model sparse graphs. For alleviating these problems, this paper proposes feature disentanglement and homogeneous second order feature propagation for recommendation. Features of the users and items are disentangled into two parts resulting in the final ratings. Then we improve Neighborhood Routing Algorithm via adding rating embedding so as to simultaneously learn the positive and negative effects of neighborhood nodes on the central node. Finally, we apply similar homogeneous neighbors which are statistically-validated by Bipartite Score Configuration Model to the second convolution for mitigating the problems happening on the dense graphs and sparse graphs. Experimental results on two different scale real-world datasets demonstrate the effectiveness of the proposed model.**

***Index Terms*—Feature disentanglement, negative effects, homogeneous neighbors.**

I. INTRODUCTION

RECOMMENDATION systems help users to find items of interest by modeling user-item interection. Typically, based on the interaction information traditional recommendation approaches utilize collaborative filtering [1], [2], content-based filtering [3] or hybrid filtering [4] to present the tailored items for the user. Recent years witness the development of Graph Convolution Neural Network (GCN). The GCN framework generates the node representation by aggregating its neighborhood information with a local parameter-sharing operator, and thus serves specific tasks, such as node classification, link prediction, etc. Based on user-item interaction graph and Matrix Factorization (MF), Monti et al. propose the first GCN-based approach for recommendation [5]. Since then, some GCN-based recommendation approaches have been emerging [6], [7]. In addition to users' feedback, side information, such as knowledge graph and social relationship, is used to boost recommendation performance [8]. Wang et al. design knowledge-aware path recurrent network, which generates paths composed of entities and relations for user-item interaction inference [9]. Meta-path-guided heterogeneous Graph Neural Networks leverage multi-hop meta-paths to learn comprehensive node embedding, which uncover the semantic and structural information of heterogeneous networks [10], [11]. However, knowledge-based recommendation methods need additional knowledge base, and need to extract multiple entity relations and meta-paths in advance. Social recommendation combines user-item interaction with social relation information to learn node representations, which helps to understand user preference in term of social influence [12]. Typical methods realize social recommendation, including adding social regularization constraint [13], adopting multi-layer feature integration [14], [15]. Similarly, social relation information is not available for many recommendation scenarios. If we could learn similar users and similar items from the user-item interaction graph, use the homogeneous neighborhood nodes to further update the central node representation, which may obtain similar effect to social recommendation.

Due to neural network powerful learning capability, current GCN-based recommendation methods have greater performance compared with traditional methods. But these methods usually use one hop neighbors to obtain node representation, thus, have relatively poor performance when bipartite graphs are sparse. Though DeepWalk could address sparse problem in the random walk way[16], random walk in bipartite graphs may add noise for long distance weak dependence. For dense graphs, connectivity among nodes is strong, so multiple convolutions based on one hop neighbors may make representation of nodes in the same sub-connected graph converge to a certain vector. This is the phenomenon of over-smoothing. We propose homogeneous second order feature propagation to make up aforementioned deficiencies. Specifically, statistically-validated method, Bipartite Score Configuration Model (BISCM) is utilized to identify homogeneous neighbors of the node instead of side information [17]. Then, we use one hop neighbors to carry out the first convolution operation, use homogeneous neighbors to conduct the second convolution operation. Homogeneous second order feature propagation supplements rich information from similar homogeneous neighborhood perspective, also helps to achieve diversity recommendation, avoids users falling into information cocoon room as a result of focusing on individual inherent history interests.

Some researchers assume there exists some factors causing user-item interaction, and propose recommendation approaches based on feature disentanglement or causing embed-

Yaru Zhang is with Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100190, China, and also with University of Chinese Academy of Sciences, Beijing 100049, China, email: zhangyaru@amss.ac.cn.

Xijin Tang is with Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100190, China, and also with University of Chinese Academy of Sciences, Beijing 100049, China, email: xjtang@iss.ac.cn.

Manuscript received August 20, 2021; revised October 26, 2021.

ding [18], [19]. Inspired by the idea, we expect to use two factors which cause user-item interaction to respectively represent user-item both sides. Concretely, feature of each node contains two parts, the first part features of users reflect their interest taste, and the first part features of items reflect their characteristic. The second part features of items reflect their popularity, and the second part features of users reflect the popularity level of items loved by them. Once disentanglement features are determined, we utilize improved Neighborhood Routing Algorithm to aggregate neighborhood nodes. Original Neighborhood Routing Algorithm selects positive neighbors to update the central node representation, and neglects effects of negative neighbors [20]. For rating bipartite graph, edges are assigned different weights, hence, neighborhood nodes have varying degrees of positive and negative effects on the central node. When using one hop neighbors, this paper will add rating embedding to Neighborhood Routing Algorithm in order to learn positive and negative feedback of neighbors. To summarize, this work has the following three contributions:

- We disentangle features of the users and items into two parts resulting in the final ratings. Then we improve Neighborhood Routing Algorithm via adding rating embedding to learn the positive and negative effects of neighborhood nodes on the central node.
- We apply similar homogeneous neighbors which are identified using BISCM to the second convolution, which helps to alleviate the over-smoothing problem of dense graphs and the sparse connection problem of sparse graphs.
- Experimental results on two different scale real-world datasets demonstrate the effectiveness of the proposed model.

II. Related work

In recent years, Graph Convolutional Neural Network has been applied to recommendation systems to enhance the recommendation performance. The first GCN-based recommendation system combines with Matrix Factorization to learn meaningful graph-structured patterns from user and item graphs [5]. Berg et al. propose Graph Convolutional Matrix Completion (GCMC) model [6]. Considering different labeled edges, GCMC conducts differentiable one hop neighborhood message aggregation on the user-item bipartite graph. Based on GCMC, Zhang et al. develop Stacked and Reconstructed Graph Convolutional Networks (STAR-GCN) architecture [7]. STAR-GCN masks node embeddings and reconstructs these masked embeddings with a block of graph encoder-decoder to address cold start problem. However, above methods seldom mine multi-class structure information. By incorporating social network information Ma et al. present a matrix factorization recommendation framework with social regularization [13]. Fan et al. provide a graph neural network framework for social recommendation [14]. The framework jointly models user-item interaction graph and user-user social graph, at the same time, captures the heterogeneous strengths of edges. Although above methods consider user-item interaction relation and social relation to get rich semantics, social relation is not available for many recommendation scenarios. Based on co-occurrence commodity graph Li et al. conduct multi-layer intention diffusion and aggregation process to mine user intention [21]. Sun et al. employ Bayesian Graph Convolutional Neural Network to model the uncertainty in the user-item interaction graph, and sample graphs are generated with the node copying model [22]. Both works contribute to solving user behavior sparse problem and interest weak generalization problem due to considering relevance among homogeneous nodes. But the relevance is not validated, and thus may increase noise. According to the idea which any two nodes sharing a statistically-significant number of neighbors are similar, Saracco et al. propose Bipartite Configuration Model (BICM) to obtain statistically-validated monopartite projections of bipartite networks [23]. Edges are assigned different weights in the user-item bipartite weight graph, and the weights represent users' reviews for corresponding items. Considering the weights of edges, Becatti et al. propose Bipartite Score Configuration Model to extend BICM to bipartite weight graphs [17]. In this paper, we will adopt BISCM to build similarity graphs about homogeneous nodes, and exploit the statistically-validated similarity monopartite graphs to the second convolution operation with the expectation of addressing above limitations.

Ma et al. propose a novel Neighborhood Routing Algorithm to learn disentanglement features of nodes [20]. The algorithm is capable of identifying significant latent factors which cause connection between the central node and one of its neighbors. Hu et al. apply Neighborhood Routing Algorithm to news recommendation circumstance, and introduce preference regularization to boost feature disentanglement [18]. Zheng et al. suppose that user interest and conformity two factors cause user-item interaction [19]. They mine cause-specific data, pair each positive sample with corresponding negative samples, and then learn disentanglement features of nodes through Bayesian Probabilistic Ranking (BPR) loss. Inspired by the feature disentanglement idea, considering two causing factors and neighborhood positive/negative effects this paper adjusts Neighborhood Routing Algorithm to rating prediction task on the bipartite rating graph.

III. Proposed Model

In this section, we build on the Neighborhood Routing Algorithm to develop a novel approach which takes strengths of edges into account and incorporates homogeneous second order feature propagation for rating prediction. Firstly, we learn node representation, then design loss function to train the model.

A. Node Representation Learning

In this subsection, we focus on illustrating the learning process of the user representation using improved Neighborhood Routing Algorithm. Item representation can be learned similarly, and it is not described here. The overall algorithm flow is shown in Algorithm 1 and Algorithm 2.

Algorithm 1 Improved Neighborhood Routing Algorithm: the First Order Feature Propagation

```
Input: $p_{i,k}, q_{j,k}, e_{i,j}, k = 1, 2, j \in N(i)$;
Output: $u_{i,k}, k = 1, 2$;
1: $u_{i,k} \leftarrow p_{i,k}$
2: for $T$ iterations do
3:     for $j \in N(i)$ do
4:         $s_{j,k} \leftarrow softmax(u_{i,k}^T q_{j,k}), k = 1, 2$
5:     for $k = 1, 2$ do
6:         $u_{i,k} \leftarrow Norm(p_{i,k} + \sum_{j \in N(i)} e_{i,j} \odot (s_{j,k} q_{j,k}))$
7: return $u_{i,k}$
```

Algorithm 2 Improved Neighborhood Routing Algorithm: the Second Order Feature Propagation

```
Input: $u_{i,k}, u_{j,k}, w_{i,j}, k = 1, 2, j \in H(i)$;
Output: $z_{i,k}, k = 1, 2$;
1: $z_{i,k} \leftarrow u_{i,k}$
2: for $T$ iterations do
3:     for $j \in H(i)$ do
4:         $s_{j,k} \leftarrow softmax(z_{i,k}^T u_{j,k}), k = 1, 2$
5:     for $k = 1, 2$ do
6:         $z_{i,k} \leftarrow Norm(u_{i,k} + \sum_{j \in H(i)} w_{i,j}(s_{j,k} u_{j,k}))$
7: return $z_{i,k}$
```

1) Computing the First Order Feature of the User

We assume that there are two factors that cause user-item interaction. Namely, feature of each node contains two parts which have the same dimension, the first part features of users reflect their interest taste, and the first part features of items reflect their characteristic. The second part features of items reflect their popularity, and the second part features of users reflect the popularity level of items loved by them. Formally, Let p_i, q_j be initial embeddings of $user_i$ and $item_j$. They are randomly initialized. When there is side information available for users or items, according to information category, the encoding feature could be filled into appropriate embedding section. For example, if genres of items are available, we express genres of the item as multi-hot vector, then linearly project the sparse vector into the dense vector which is viewed as $q_{j,1}$, the first part initial embedding of $item_j$, and $q_{j,2}$ is embedded with random vector. It's worth noting that for new item, we could use learned side information representation to give it more accuracy embedding for dealing with cold start problem.

Given p_i, we map the initial embedding using projection matrix W_1, and then disentangle the vector p_i into two parts, respectively normalize.

$$p_i = ReLU(W_1 p_i + b_1), \tag{1}$$

$$p_{i,1} = Norm(p_{i,1}) = \frac{p_{i,1}}{||p_{i,1}||_2}, \tag{2}$$

$$p_{i,2} = Norm(p_{i,2}). \tag{3}$$

We can construct two kinds of normalized disentanglement features of items in the same way.

$$q_j = ReLU(W_1 q_j + b_1), \tag{4}$$

$$q_{j,1} = Norm(q_{j,1}), \tag{5}$$

$$q_{j,2} = Norm(q_{j,2}). \tag{6}$$

Neighborhood Routing Algorithm infers the significant factor that mainly contributes to the final rating by computing similarities of two interaction nodes under each kind of disentanglement features [20]. The original algorithm just selects positive neighborhood nodes to update the central node. If viewing effects of positive nodes as positive feedback, then negative nodes play a negative feedback role for representation learning of the central node. Here, for the first order feature propagation, we improve Neighborhood Routing Algorithm by adding score embedding. Let $N(i)$ denote $user_i$'s fixed number of neighbors. $\{q_{j,1}|j \in N(i)\}$ and $\{q_{j,2}|j \in N(i)\}$ are normalized disentanglement feature sets of $user_i$'s neighborhood nodes. $\{e_{i,j}|j \in N(i)\}$ is score embedding set between $user_i$ and its neighbors. There are six kinds of ratings, including "0, 1, 2, 3, 4, 5". "0" represents that the user does not interact with the item, which is the padding item. The six kinds of score embeddings are learned during training. The first order feature of $user_i$ can be formalized as follows,

$$u_{i,1} = NR(p_{i,1}, \{q_{j,1}|j \in N(i)\}, \{e_{i,j}|j \in N(i)\}), \tag{7}$$

$$u_{i,2} = NR(p_{i,2}, \{q_{j,2}|j \in N(i)\}, \{e_{i,j}|j \in N(i)\}). \tag{8}$$

In detail, when executing t routing iteration, the similarity of $user_i$ and $item_j$ about feature k is

$$s_{j,k}^{(t)} = \frac{exp(u_{i,k}^{(t-1)T} q_{j,k})}{\sum_{k'=1}^{2} exp(u_{i,k'}^{(t-1)T} q_{j,k'})}, k = 1, 2. \tag{9}$$

Then we exploit Eq. 10 to update feature k of $user_i$ after t routing iteration. When aggregating information of each neighbor, we not only multiply weight about the factor, but also multiply score embedding $e_{i,j}$. The score embeddings allow the model to learn positive and negative effects of neighbors on the central node. When finishing T iterations, we obtain the first order feature of $user_i$, $[u_{i,1}, u_{i,2}]$.

$$u_{i,k}^{(t)} = \frac{p_{i,k} + \sum_{j \in N(i)} e_{i,j} \odot (s_{j,k}^{(t)} q_{j,k})}{||p_{i,k} + \sum_{j \in N(i)} e_{i,j} \odot (s_{j,k}^{(t)} q_{j,k})||_2}, k = 1, 2. \tag{10}$$

2) Computing the Second Order Feature of the User

The core of Graph Convolution Neural Network is how to aggregate features from local neighbors using neural network. To learn more accuracy node representations, GCN usually performs the second even the third convolution operation. However, when the graph is dense to some extent, deeper repetitive iterative learning could not work because it causes consistent representations of nodes in the same connected subgraph. Instead, when the user-item bipartite graph is sparse, only using closer local heterogeneous neighbors to update the central node representation could be insufficient. In addition, only considering user-item interaction could make users fall into information cocoon room. That is because items which are liked by the user could be very similar, then based on the interaction history, the model may still recommend almost the same items for the user. In other words, recommended items are lack of diversity. Furthermore, when the user do not try new types of items actively, he will fall into a self-enclosed information cocoon room.

For the issues, this paper utilizes user-user similarity graph and item-item similarity graph to supplement side information from homogeneous neighbors. BISCM [17] provides a statistically-verified method to identify similarity of homogeneous nodes, which ensures the reliability of similarity relation. The core idea of BISCM is that any two nodes having statistically enough the same high rating neighbors are similar. As Fig. 1 shows, U, I respectively represent user set and item set, dark read, light read, orange, light green, dark green edges respectively represent "5, 4, 3, 2, 1", then $user_1$ and $user_2$ maybe similar, $item_2$ and $item_3$ maybe similar according to BISCM. Instead of using explicit homogeneous graphs available, we construct the user-user graph and item-item graph by BISCM to further conduct the second order feature propagation.

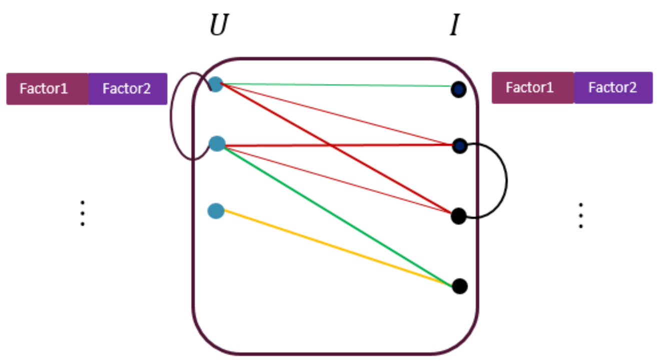

Fig. 1. The bipartite rating graph.

For the second convolution, we leverage aggregation operation of original Neighborhood Routing Algorithm. The weight of actual neighbor is 1, and the weight of padding neighbor is 0. Let $H(i)$ denote $user_i$ fixed number of homogeneous neighbors. $\{u_{j,1}|j \in H(i)\}$ and $\{u_{j,2}|j \in H(i)\}$ obtained according to Algorithm 1 are the first order disentanglement feature sets of $user_i$'s homogeneous neighbors. $\{w_{i,j}|j \in H(i)\}$ is corresponding weight set. The calculation process of $user_i$'s second order feature is as follows, and the detail is shown in Algorithm 2. Through T iterations, the second order feature of $user_i$, $[z_{i,1}, z_{i,2}]$ is obtained.

$$z_{i,1} = NR(u_{i,1}, \{u_{j,1}|j \in H(i)\}, \{w_{i,j}|j \in H(i)\}), \quad (11)$$

$$z_{i,2} = NR(u_{i,2}, \{u_{j,2}|j \in H(i)\}, \{w_{i,j}|j \in H(i)\}). \quad (12)$$

B. Model Training

1) Base Loss Function

We model the recommendation as a rating prediction task. Note that $x_{j,1}$ is the first part of $item_j$'s second order feature, and $x_{j,2}$ is the second part of $item_j$'s second order feature. Then $s^t_{i,j}$ means similarity between taste of $user_i$ and characteristic of $item_j$, and $s^p_{i,j}$ means commonality between popularity level of items which are liked by $user_i$ and the popularity of $item_j$. Both similarities contribute to the final predicted rating $\hat{r}_{i,j}$. W and b in Eq. 15 are learnable parameters.

$$s^t_{i,j} = z^T_{i,1} x_{j,1}, \quad (13)$$

$$s^p_{i,j} = z^T_{i,2} x_{j,2}, \quad (14)$$

$$\hat{r}_{i,j} = W(z^T_i x_j) + b = W(s^t_{i,j} + s^p_{i,j}) + b. \quad (15)$$

Once obtaining the predicted rating, combining with the ground truth rating $r_{i,j}$, we use Mean Square Error (MSE) to express the base loss function L_1, where O denotes training set.

$$L_1 = \frac{1}{|O|} \sum_{(user_i, item_j) \in O} (r_{i,j} - \hat{r}_{i,j})^2. \quad (16)$$

2) Additional Loss Functions

We will set two additional loss functions to promote the independence of two factors. In other words, we expect that consistencies of the user and item about two aspects respectively cause the final rating. One is likelihood loss function about classification of disentanglement features. For two kinds of disentanglement features of $user_i$ and $item_j$, we use a fully connected layer containing parameters W_c and b_c to predict category distributions $P(c|z_{i,k})$ and $P(c|x_{j,k})$. Then based on actual feature parts, including "1, 2", we give loss function L_2 to maximize corresponding likelihood.

$$P(c|z_{i,k}) = softmax(W_c z_{i,k} + b_c), k = 1, 2, \quad (17)$$

$$P(c|x_{j,k}) = softmax(W_c x_{j,k} + b_c), k = 1, 2, \quad (18)$$

$$L_2 = - \sum_{a_i \in \{z_u, x_v | (user_u, item_v) \in O\}} \sum_{k=1,2} log(P(c|a_{i,k})[k]). \quad (19)$$

The other is margin loss function about two kinds of feature similarities. For the sample $(user, item)$, we define popularity of the item as the number of rating records which the item has when the user rates. And we view records in training set as full space. For $popularity/score$, if qualitatively considering relative values of numerator and denominator, there are four kinds of value levels. We assume that if the user is interacting with the popular item, he prefers popular items. It means his second part feature is similar with that of high popularity level items. In contrast, if the user is interacting with the unpopular item, characteristic of the item may meet his taste. In short, there exists one factor mainly causing the interaction for the both circumstances. We use $y = 1$ to indicate $popularity/score < down\_threshold$, that is popularity of the item is low when the user interacts with it, but the user gives high score. Then we expect two similarities s^p and s^t learned by the proposed model satisfy $s^p < s^t$. In other words, the high score is mainly due to consistency between taste of the user and characteristic of the item. We use $y = -1$ to indicate $popularity/score > up\_threshold$, that is popularity of the item is high when the user interacts with it, but the user gives low score. Then we expect the s^p and s^i learned by the proposed model satisfy $s^p > s^t$. Namely, characteristic of the item does not match taste of the user. We utilize $y = 0$ to indicate other circumstances. The $down\_threshold$ and $up\_threshold$ are respectively assigned with the first quartile and the third quartile of the corresponding values of training set.

We exploit margin loss function L_p to learn the above inequality relations. For the sample $(user, item) \in O$, we calculate (s^t, s^p, y). Set O_1 satisfies $(s^t, s^p, y) \in O_1$, and $y = \pm 1$.

$$L_p = \frac{1}{|O_1|} \sum_{(s^t, s^p, y) \in O_1} max(0, -y(s^t - s^p)). \tag{20}$$

3) Final Loss Function

The final loss function can be expressed as

$$L = (1 - \lambda_1 - \lambda_2) L_1 + \lambda_1 L_2 + \lambda_2 L_p + \eta ||\Theta||_2. \tag{21}$$

where λ_1, λ_2 are balance coefficients, η is regularization coefficient and Θ denotes all the trainable parameters.

IV. Experiments

A. Datasets

This paper will use two different scale datasets to demonstrate the effectiveness of the proposed model. One is MovieLens 100K (ML-100K) dataset. The other is Douban dataset. Douban dataset is selected from the initial Douban dataset[1], which is collected from https://www.douban.com/ in September 2019 by Liu et al. Movies are rated as "1, 2, 3, 4, 5", and we can optionally incorporate additional item information, such as genres, publishing country, synopsis, release time, actors, etc. There are 4 million 160 thousand rating records in the initial dataset. Here, the items released after 2015 are considered, and the items with less than 5 records will be removed. Then we only consider the users who have more than 4 records. The statistics of the datasets is shown in Table I. ML-100K dataset is denser. In contrast, Douban dataset is sparser.

TABLE I
STATISTICS OF DATASETS

Dataset	Users	Items	Ratings	Density
ML-100K	943	1682	100000	0.0630
Douban	31922	10952	589534	0.0017

B. Experimental Setup

1) Experimental Setup for ML-100K Dataset

We conduct experiments on the first of the five provided data splits with 20% records for testing, that is u1.base, u1.test. There are 943 users, 1650 items in u1.base, and rating distribution is 4719: 9178: 21963: 27396: 16744. Based on u1.base, we construct the bipartite graph. We regard "4, 5" as high scores, and use corresponding high score neighborhood nodes to obtain user-user graph, item-item graph (parameter t in BISCM is set to 0.05). 10% records of the training set are randomly selected as the validation set.

[1]https://www.csuldw.com/2019/09/08/2019-09-08-moviedata-10m/

We set dropout to 0.5. The initial learning rate is set as 0.025, and gradually decreases to 0.001 with decay rate of 0.8 during training. Embedding dimension of nodes is 64 (32 for each factor), and embedding dimension of scores is 32. We set $\eta = 10^{-4}, \lambda_1 = \lambda_2 = 0.02$. The number of routing is set as 5, batch size is set as 256. The number of users' heterogeneous neighbors is 20, the number of movies' heterogeneous neighbors is 10. The number of users' homogeneous neighbors is 10, and the number of movies' homogeneous neighbors is 5. Numbers of neighbors are set according to about a quarter of the average. And we address the label leakage problem by setting rating of the target node in neighborhood nodes into "0".

2) Experimental Setup for Douban Dataset

For Douban dataset, rating records of 3481 users whose rating records are after 2019 are extracted as the challenge test set. In the remaining samples, for each user, we arrange his rating records according to rating time, and the first 60% samples are split as the training set, including 325285 rating records, 28441 users, 10289 movies, and rating distribution is 45112: 66561: 118455: 71479: 23678. The training set is used to construct the bipartite graph and calculate homogeneous neighbors. The next 20% samples are used as the validation set, and the final 20% samples are used as the test set.

The initial learning rate is set as 0.05, and gradually decreases to 0.0004 with decay rate of 0.8 during training. Embedding dimension of nodes is 128 (64 for each factor), and embedding dimension of scores is 64. We set $\eta = 10^{-5}$. Batch size is set as 1024. The number of users' heterogeneous neighbors is 10, the number of movies' heterogeneous neighbors is 30. The number of users' homogeneous neighbors is 30, and the number of movies' homogeneous neighbors is 10. Numbers of neighbors are set according to the average. Other settings are same to ML-100K's.

C. Discussion of Results

We evaluate the performance of the proposed model in term of Root Mean Square Error (RMSE). Table II gives experimental results on the two datasets. First order feature propagation denotes there is one convolution operation using heterogeneous neighbors. First order feature propagation with movie genre embedding denotes that initial item embedding contains genre information. Second order feature propagation with heterogeneous neighbors denotes that there are two convolution operations using heterogeneous neighbors. Homogeneous second order feature propagation is the model that we propose, that is the first convolution operation uses heterogeneous neighbors, and the second convolution operation uses homogeneous neighbors. sRMGCNN is proposed by Monti et al, which utilizes geometric deep learning on graph [5]. Homogeneous second order feature propagation has better performance. When embedding genre, the model shows advantage of feature disentanglement more. Compared to first order feature propagation, second order feature propagation with heterogeneous neighbors does not work, but homogeneous second order feature propagation improves prediction performance for two kinds of bipartite graphs.

TABLE II
COMPARISON OF TEST RMSE SCORES FOR DIFFERENT METHODS

Method	ML-100K	Douban
First Order Feature Propagation	0.9363	0.8696
First Order Feature Propagation with Movie Genre Embedding	0.9236	-
Second Order Feature Propagation with Heterogeneous Neighbors	0.9430	0.8722
Second Order Feature Propagation with Heterogeneous Neighbors and Movie Genre Embedding	0.9345	-
Homogeneous Second Order Feature Propagation	**0.9299**	**0.8670**
Homogeneous Second Order Feature Propagation with Movie Genre Embedding	**0.9177**	-
sRMGCNN	0.929 [5]	-

TABLE III
COMPARISON AMONG THE PROPOSED MODEL VARIANTS

Method	RMSE
First Order Feature Propagation	**0.8696**
First Order Feature Propagation w/o Likelihood Loss	0.8705
First Order Feature Propagation w/o Two Additional Losses	0.8725
First Order Feature Propagation w/o Two Additional Losses and Score Embedding	0.9321

D. Ablation Study

As the proposed model involves multiple novel components, we conduct ablation experiments on Douban dataset to explore the contribution of each component to the performance. The results are shown in Table III. First order feature propagation w/o likelihood loss denotes that we do not add likelihood loss into the final loss function. First order feature propagation w/o two additional losses and score embedding denotes that we do not add two additional losses into the final loss function, and we just use high score ("4, 5") neighbors to implement aggregation operation without score embedding. The results illustrate that learning positive and negative effects of neighbors on the central node is helpful for prediction, and using the two additional losses improves the performance of the model because they promote feature disentanglement.

E. Case Study

We randomly select 10 users and some of their homogeneous neighbors from ML-100k dataset. For each user, we assemble non-repeating sample pairs using all items rated by the user and part of items which are rated by the user's homogeneous neighbors. Next, we use homogeneous second order feature propagation with movie genre embedding model trained to predict ratings. Then, for each user, we respectively calculate the sum of multi-hot genre vectors of real medium and high score items (the corresponding score is greater than 2) and the sum of multi-hot genre vectors of predicted medium and high score items. The heat maps are shown in Fig. 2 and Fig. 3. Longitudinal "0-9" denote 10 users. Horizontal "0-18" denote 19 kinds of genres, and "0" represents "unknown genre". The average number of genres per user is 13.6 in Fig. 2, and the average number of genres per user is 14.2 in Fig. 3. Our model could recommend diversity items whose genres are not limited to genres of history high score items.

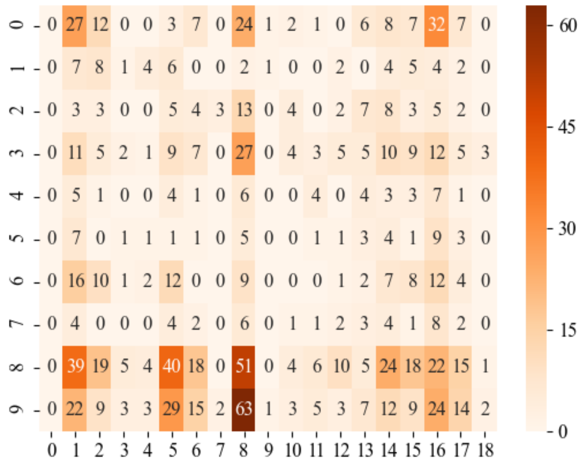

Fig. 2. The heat map about users and genres of real non-low score items.

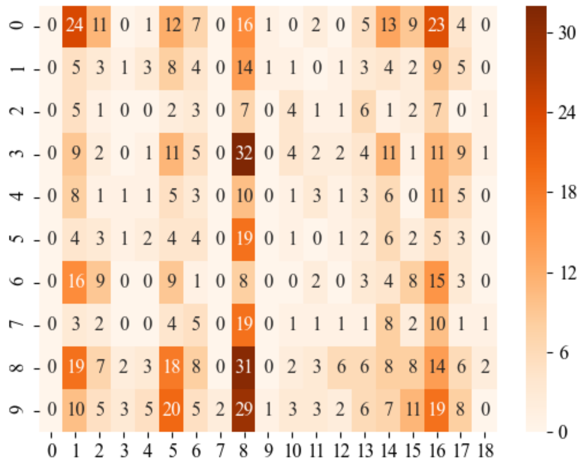

Fig. 3. The heat map about users and genres of predicted non-low score items.

V. CONCLUSIONS

In this paper, we propose the model which consists of feature disentanglement and homogeneous second order feature propagation for recommendation. First, we disentangle features of the users and items into two parts respectively causing the final rating. Then, we add score embedding into Neighbor Routing Algorithm to learn the positive and negative effects of neighborhood nodes on the central node. Last but not least, we use similar homogeneous neighbors which are identified using BISCM for secondary convolution. Homogeneous second order feature propagation alleviates the over-smoothing issue of dense graphs, lack issue of the first order neighbors of sparse graphs. Moreover, by means of homogeneous neighbors, instead of only using one hop neighbors, the model helps to achieve diversity recommendation. Experiments are performed on two different scale real-world datasets to demonstrate the effectiveness of the proposed model.

Homogeneous second order feature propagation is not only suitable for Neighborhood Routing Algorithm, but also maybe appropriate for other GCNs. We will try to apply it to other convolution operations. In the future, we will also give test results on the Douban challenge test set (all users do not appear in the training set) to further verify the model performance. For side information, we will embed initial movie popularity features using actor, movie popularity value information, and supplement movie characteristic features using movie publishing country, movie text description information.

Acknowledgment

This research was supported by National Key Research and Development Program of China under Grant No. 2016YFB1000902 and National Natural Science Foundation of China under Grant Nos. 71731002 and 71971190. The computations were partly done on the high performance computers of State Key Laboratory of Scientific and Engineering Computing, Chinese Academy of Sciences.

References

[1] J. B. Schafer, D. Frankowski, J. Herlocker, and S. Sen, “Collaborative filtering recommender systems,” in *The Adaptive Web*. Springer, 2007, pp. 291–324.

[2] N. Rao, H.-F. Yu, P. Ravikumar, and I. S. Dhillon, “Collaborative filtering with graph information: consistency and scalable methods,” in *Proceedings of the 28th International Conference on Neural Information Processing Systems*, vol. 2, 2015, pp. 2107–2115.

[3] M. J. Pazzani and D. Billsus, “Content-based recommendation systems,” in *The Adaptive Web*. Springer, 2007, pp. 325–341.

[4] P. B. Thorat, R. Goudar, and S. Barve, “Survey on collaborative filtering, content-based filtering and hybrid recommendation system,” *International Journal of Computer Applications*, vol. 110, no. 4, pp. 31–36, 2015.

[5] F. Monti, M. M. Bronstein, and X. Bresson, “Geometric matrix completion with recurrent multi-graph neural networks,” in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017, pp. 3700–3710.

[6] R. van den Berg, T. N. Kipf, and M. Welling, “Graph convolutional matrix completion,” 2018.

[7] J. Zhang, X. Shi, S. Zhao, and I. King, “Star-gcn: Stacked and reconstructed graph convolutional networks for recommender systems,” in *IJCAI*, 2019.

[8] S. Wu, W. Zhang, F. Sun, and B. Cui, “Graph neural networks in recommender systems: A survey,” *arXiv e-prints*, pp. arXiv–2011, 2020.

[9] X. Wang, D. Wang, C. Xu, X. He, Y. Cao, and T.-S. Chua, “Explainable reasoning over knowledge graphs for recommendation,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 5329–5336.

[10] S. Fan, J. Zhu, X. Han, C. Shi, L. Hu, B. Ma, and Y. Li, “Metapath-guided heterogeneous graph neural network for intent recommendation,” in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 2478–2486.

[11] J. Shi, H. Ji, C. Shi, X. Wang, Z. Zhang, and J. Zhou, “Heterogeneous graph neural network for recommendation,” *arXiv e-prints*, pp. arXiv–2009, 2020.

[12] S. Wang, L. Hu, Y. Wang, X. He, Q. Z. Sheng, M. A. Orgun, L. Cao, F. Ricci, and S. Y. Philip, “Graph learning based recommender systems: a review,” in *Proceedings of the 30th International Joint Conference on Artificial Intelligence*, 2021, pp. 4644–4652.

[13] H. Ma, D. Zhou, C. Liu, M. R. Lyu, and I. King, “Recommender systems with social regularization,” in *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining*, 2011, pp. 287–296.

[14] W. Fan, Y. Ma, Q. Li, Y. He, E. Zhao, J. Tang, and D. Yin, “Graph neural networks for social recommendation,” in *The World Wide Web Conference*, 2019, pp. 417–426.

[15] L. Wu, J. Li, P. Sun, R. Hong, Y. Ge, and M. Wang, “Diffnet++: A neural influence and interest diffusion network for social recommendation,” *IEEE Transactions on Knowledge and Data Engineering*, 2020.

[16] B. Perozzi, R. Al-Rfou, and S. Skiena, “Deepwalk: Online learning of social representations,” in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2014, pp. 701–710.

[17] C. Becatti, G. Caldarelli, and F. Saracco, “Entropy-based randomization of rating networks,” *Physical Review E*, vol. 99, no. 2, p. 022306, 2019.

[18] L. Hu, S. Xu, C. Li, C. Yang, C. Shi, N. Duan, X. Xie, and M. Zhou, “Graph neural news recommendation with unsupervised preference disentanglement,” in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 4255–4264.

[19] Y. Zheng, C. Gao, X. Li, X. He, Y. Li, and D. Jin, “Disentangling user interest and conformity for recommendation with causal embedding,” in *Proceedings of the Web Conference*, 2021, pp. 2980–2991.

[20] J. Ma, P. Cui, K. Kuang, X. Wang, and W. Zhu, “Disentangled graph convolutional networks,” in *International Conference on Machine Learning*. PMLR, 2019, pp. 4212–4221.

[21] F. Li, Z. Chen, P. Wang, Y. Ren, D. Zhang, and X. Zhu, “Graph intention network for click-through rate prediction in sponsored search,” in *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2019, pp. 961–964.

[22] J. Sun, W. Guo, D. Zhang, Y. Zhang, F. Regol, Y. Hu, H. Guo, R. Tang, H. Yuan, X. He *et al.*, “A framework for recommending accurate and diverse items using bayesian graph convolutional neural networks,” in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020, pp. 2030–2039.

[23] F. Saracco, M. J. Straka, R. Di Clemente, A. Gabrielli, G. Caldarelli, and T. Squartini, “Inferring monopartite projections of bipartite networks: An entropy-based approach,” *New Journal of Physics*, vol. 19, no. 5, p. 053022, 2017.

Yaru Zhang is currently working towards PhD in Academy of Mathematics and Systems Science, Chinese Academy of Sciences. Her research interests include natural language processing, social network analysis and knowledge management.

Xijin Tang is a full professor in Academy of Mathematics and Systems Science, Chinese Academy of Sciences. She received her BEng (1989) on computer science and engineering from Zhejiang University, MEng (1992) on management science and engineering from University of Science and Technology of China and PhD (1995) from Institute of Systems Science, CAS. During her early system research and practice, she developed several decision support systems for water resources management, weapon system evaluation, e-commerce evaluation, etc. Her recent interests are meta-synthesis and advanced modeling, social network analysis and knowledge management, opinion mining and opinion dynamics, opinion big data and societal risk perception. Now she is the secretary general of Systems Engineering Society of China. She also serves as vice president and secretary general of International Society for Knowledge and Systems Science.