Graph Neural Networks for Social Recommendation

Wenqi Fan
Department of Computer Science
City University of Hong Kong
wenqifan03@gmail.com

Yao Ma
Data Science and Engineering Lab
Michigan State University
mayao4@msu.edu

Qing Li
Department of Computing
The Hong Kong Polytechnic University
csqli@comp.polyu.edu.hk

Yuan He
JD.com
heyuan6@jd.com

Eric Zhao
JD.com
ericzhao@jd.com

Jiliang Tang
Data Science and Engineering Lab
Michigan State University
tangjili@msu.edu

Dawei Yin
JD.com
yindawei@acm.org

ABSTRACT

In recent years, Graph Neural Networks (GNNs), which can naturally integrate node information and topological structure, have been demonstrated to be powerful in learning on graph data. These advantages of GNNs provide great potential to advance social recommendation since data in social recommender systems can be represented as user-user social graph and user-item graph; and learning latent factors of users and items is the key. However, building social recommender systems based on GNNs faces challenges. For example, the user-item graph encodes both interactions and their associated opinions; social relations have heterogeneous strengths; users involve in two graphs (e.g., the user-user social graph and the user-item graph). To address the three aforementioned challenges simultaneously, in this paper, we present a novel graph neural network framework **(GraphRec)** for social recommendations. In particular, we provide a principled approach to jointly capture interactions and opinions in the user-item graph and propose the framework GraphRec, which coherently models two graphs and heterogeneous strengths. Extensive experiments on two real-world datasets demonstrate the effectiveness of the proposed framework GraphRec. Our code is available at https://github.com/wenqifan03/GraphRec-WWW19

CCS CONCEPTS

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

KEYWORDS

Social Recommendation; Graph Neural Networks; Recommender Systems; Social Network; Neural Networks

WWW '19, May 13–17, 2019, San Francisco, CA, USA

ACM ISBN 978-1-4503-6674-8/19/05.
https://doi.org/10.1145/3308558.3313488

ACM Reference Format:
Wenqi Fan, Yao Ma, Qing Li, Yuan He, Eric Zhao, Jiliang Tang, and Dawei Yin. 2019. Graph Neural Networks for Social Recommendation. In *Proceedings of the 2019 World Wide Web Conference (WWW '19), May 13–17, 2019, San Francisco, CA, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3308558.3313488

1 INTRODUCTION

The exploitation of social relations for recommender systems has attracted increasing attention in recent years [18, 28, 30]. These social recommender systems have been developed based on the phenomenon that users usually acquire and disseminate information through those around them, such as classmates, friends, or colleagues, implying that the underlying social relations of users can play a significant role in helping them filter information [23]. Hence, social relations have been proven to be helpful in boosting the recommendation performance [8, 29].

Recent years have witnessed great developments in deep neural network techniques for graph data [15]. These deep neural network architectures are known as Graph Neural Networks (GNNs) [5, 10, 19], which have been proposed to learn meaningful representations for graph data. Their main idea is how to iteratively aggregate feature information from local graph neighborhoods using neural networks. Meanwhile, node information can be propagated through a graph after transformation and aggregation. Hence, GNNs naturally integrate the node information as well as the topological structure and have been demonstrated to be powerful in representation learning [5, 7, 15]. On the other hand, data in social recommendation can be represented as graph data with two graphs. As demonstrated in Figure 1, these two graphs include a social graph denoting the relationships between users, and a user-item graph denoting interactions between users and items. Users are simultaneously involved in both graphs, who can bridge them. Moreover, the natural way of social recommendation is to incorporate the social network information into user and item latent factors learning [37]. Learning representations of items and users is the key to build social recommender systems. Thus, given their advantages, GNNs provide unprecedented opportunities to advance social recommendation.

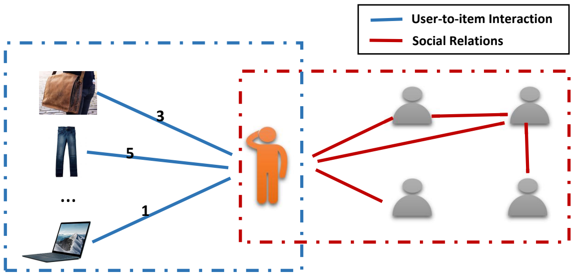

Figure 1: Graph Data in Social Recommendation. It contains two graphs including the user-item graph (left part) and the user-user social graph (right part). Note that the number on the edges of the user-item graph denotes the opinions (or rating score) of users on the items via the interactions.

Meanwhile, building social recommender systems based on GNNs faces challenges. The social graph and the user-item graph in a social recommender system provide information about users from different perspectives. It is important to aggregate information from both graphs to learn better user representations. Thus, the first challenge is how to inherently combine these two graphs. Moreover, the user-item graph not only contains interactions between users and items but also includes users' opinions on items. For example, as shown in Figure 1, the user interacts with the items of "trousers" and "laptop"; and the user likes "trousers" while disliking "laptop". Therefore, the second challenge is how to capture interactions and opinions between users and items jointly. In addition, the low cost of link formation in online worlds can result in networks with varied tie strengths (e.g., strong and weak ties are mixed together) [36]. Users are likely to share more similar tastes with strong ties than weak ties. Considering social relations equally could lead to degradation in recommendation performance. Hence, the third challenge is how to distinguish social relations with heterogeneous strengths.

In this paper, we aim to build social recommender systems based on graph neural networks. Specially, we propose a novel graph neural network **GraphRec** for social recommendations, which can address three aforementioned challenges simultaneously. Our major contributions are summarized as follows:

- We propose a novel graph neural network GraphRec, which can model graph data in social recommendations coherently;
- We provide a principled approach to jointly capture interactions and opinions in the user-item graph;
- We introduce a method to consider heterogeneous strengths of social relations mathematically; and
- We demonstrate the effectiveness of the proposed framework on various real-world datasets.

The remainder of this paper is organized as follows. We introduce the proposed framework in Section 2. In Section 3, we conduct experiments on two real-world datasets to illustrate the effectiveness of the proposed method. In Section 4, we review work related to our framework. Finally, we conclude our work with future directions in Section 5.

2 THE PROPOSED FRAMEWORK

In this section, we will first introduce the definitions and notations used in this paper, next give an overview about the proposed framework, then detail each model component and finally discuss how to learn the model parameters.

Table 1: Notation

Symbols	Definitions and Descriptions
r_{ij}	The rating value of item v_j by user u_i
$\mathbf{q}_j$	The embedding of item v_j
$\mathbf{p}_i$	The embedding of user u_i
$\mathbf{e}_r$	The opinion embedding for the rating level r, such as 5-star rating, $r \in \{1,2,3,4,5\}$
d	The length of embedding vector
$C(i)$	The set of items which user u_i interacted with
$N(i)$	The set of social friends who user u_i directly connected with
$B(j)$	The set of users who have interacted the item v_j
$\mathbf{h}_i^I$	The item-space user latent factor from item set $C(i)$ of user u_i
$\mathbf{h}_i^S$	The social-space user latent factor from the social friends $N(i)$ of user u_i
$\mathbf{h}_i$	The user latent factor of user u_i, combining from item space $\mathbf{h}_i^I$ and social space $\mathbf{h}_i^S$
$\mathbf{x}_{ia}$	The opinion-aware interaction representation of item v_a for user u_i
$\mathbf{f}_{jt}$	The opinion-aware interaction representation of user u_t for item v_j
$\mathbf{z}_j$	The item latent factor of item v_j
α_{ia}	The item attention of item v_a in contributing to $\mathbf{h}_i^I$
β_{io}	The social attention of neighboring user u_o in contributing to $\mathbf{h}_i^S$
μ_{jt}	The user attention of user u_t in contributing to $\mathbf{z}_j$
r'_{ij}	The predicted rating value of item v_j by user u_i
$\oplus$	The concatenation operator of two vectors
T	The user-user social graph
$\mathbf{R}$	The user-item rating matrix (user-item graph)
$\mathbf{W}, \mathbf{b}$	The weight and bias in neural network

2.1 Definitions and Notations

Let $U = \{u_1, u_2, ..., u_n\}$ and $V = \{v_1, v_2, ..., v_m\}$ be the sets of users and items respectively, where n is the number of users, and m is the number of items. We assume that $\mathbf{R} \in \mathbb{R}^{n \times m}$ is the user-item rating matrix, which is also called the user-item graph. If u_i gives a rating to v_j, r_{ij} is the rating score, otherwise we employ 0 to represent the unknown rating from u_i to v_j, i.e., $r_{ij} = 0$. The observed rating score r_{ij} can be seen as user u_i's opinion on the item v_j. Let $O = \left\{\left\langle u_i, v_j \right\rangle | r_{ij} \neq 0\right\}$ be the set of known ratings

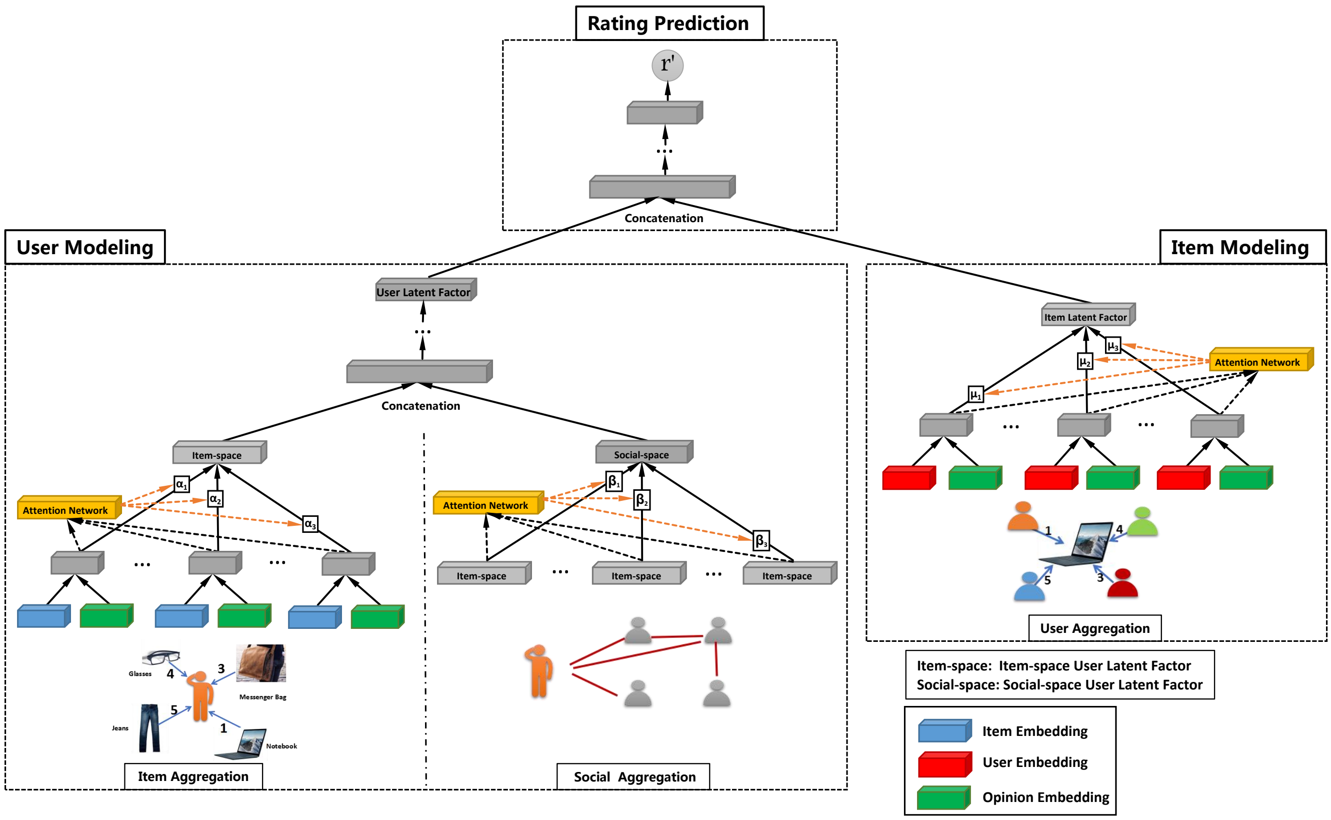

Figure 2: The overall architecture of the proposed model. It contains three major components: user modeling, item modeling, and rating prediction.

and $\mathcal{T} = \{\langle u_i, v_j \rangle | r_{ij} = 0\}$ be the set of unknown ratings. Let $N(i)$ be the set of users whom u_i directly connected with, $C(i)$ be the set of items which u_i have interacted with, and $B(j)$ be the set of users who have interacted with v_j. In addition, users can establish social relations to each other. We use $\mathrm{T} \in \mathbb{R}^{n \times n}$ to denote the user-user social graph, where $\mathrm{T}_{ij} = 1$ if u_j has a relation to u_i and zero otherwise. Given the user-item graph $\mathbf{R}$ and social graph T, we aim to predict the missing rating value in $\mathbf{R}$. Following [11], we use an embedding vector $\mathbf{p}_i \in \mathbb{R}^d$ to denote a user u_i and an embedding vector $\mathbf{q}_j \in \mathbb{R}^d$ to represent an item v_j, where d is the length of embedding vector. More details will be provided about these embedding vectors in the following subsections. The mathematical notations used in this paper are summarized in Table 1.

2.2 An Overview of the Proposed Framework

The architecture of the proposed model is shown in Figure 2. The model consists of three components: user modeling, item modeling, and rating prediction. The first component is user modeling, which is to learn latent factors of users. As data in social recommender systems includes two different graphs, i.e., a social graph and a user-item graph, we are provided with a great opportunity to learn user representations from different perspectives. Therefore, two aggregations are introduced to respectively process these two different graphs. One is item aggregation, which can be utilized to understand users via interactions between users and items in the user-item graph (or item-space). The other is social aggregation, the relationship between users in the social graph, which can help model users from the social perspective (or social-space). Then, it is intuitive to obtain user latent factors by combining information from both item space and social space. The second component is item modeling, which is to learn latent factors of items. In order to consider both interactions and opinions in the user-item graph, we introduce user aggregation, which is to aggregate users' opinions in item modeling. The third component is to learn model parameters via prediction by integrating user and item modeling components. Next, we will detail each model component.

2.3 User Modeling

User modeling aims to learn user latent factors, denoted as $\mathbf{h}_i \in \mathbb{R}^d$ for user u_i. The challenge is how to inherently combine the user-item graph and social graph. To address this challenge, we first use two types of aggregation to learn factors from two graphs, as shown in the left part in Figure 2. The first aggregation, denoted as item aggregation, is utilized to learn item-space user latent factor $\mathbf{h}_i^I \in \mathbb{R}^d$ from the user-item graph. The second aggregation is social aggregation where social-space user latent factor $\mathbf{h}_i^S \in \mathbb{R}^d$ is learned from the social graph. Then, these two factors are combined together to form the final user latent factors $\mathbf{h}_i$. Next, we will