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

WWW 2019

张峻伟
2019/7/5

Introduction

Graph Neural Networks (GNNs) can naturally integrate node information and topological structure, have been demonstrated to be powerful in learning on 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.

https://blog.csdn.net/qq_34911465/article/details/88524599

https://baijiahao.baidu.com/s?id=1625418329924881637&wfr=spider&for=pc

https://www.baidu.com/link?url=RZc1y8aq1G6sODNhhKUg1Z1mGREy8HEZR1BV-jKSvEFXUgII8xMKRCbR-IWQLV2o6icBU1463GqQfb91t-EnMxOn3hrZWyMXFr5WAmbMIN_&wd=&eqid=b2350674005c08af000000035d1e1704

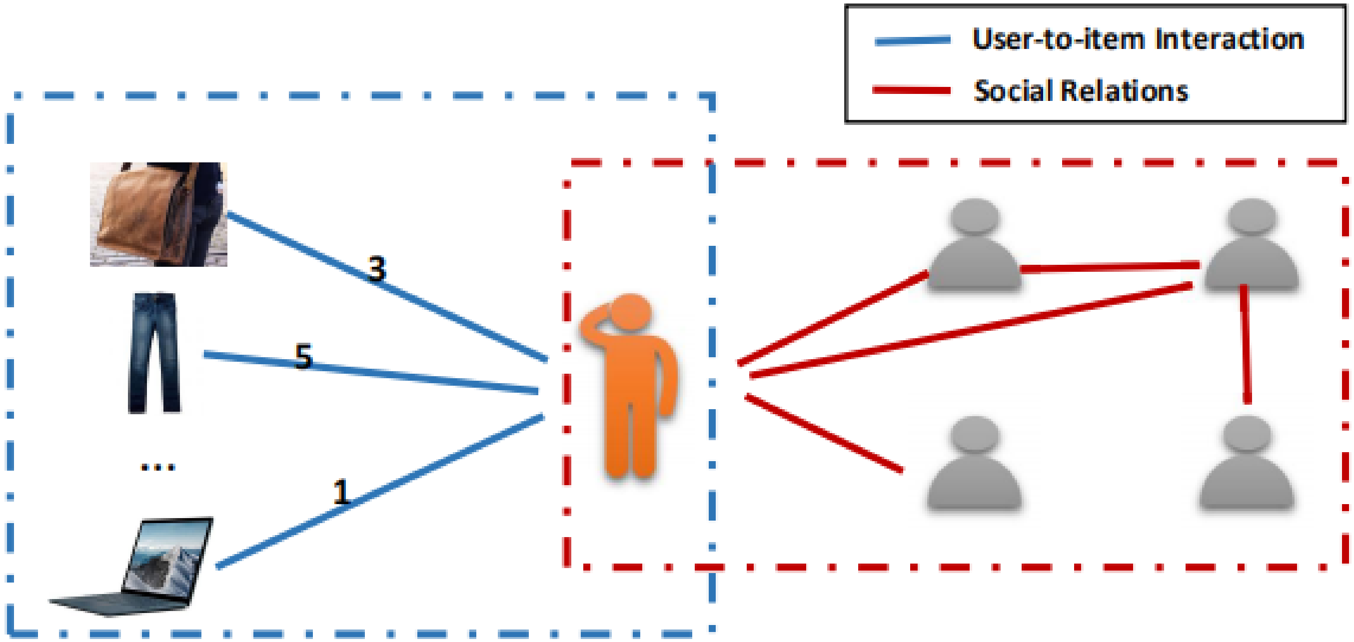


Figure 1: Graph Data in Social Recommendation. It contains two graphs including the user-item graph (left part) and the user-user social graph (right part). Note that the number on the edges of the user-item graph denotes the opinions (or rating score) of users on the items via the interactions.

The Challenge of GNNs Applied on Social Recommendation

- How to **inherently combine these two graphs;**
- How to **capture interactions and opinions between users and items jointly;**
- How to **distinguish social relations with heterogeneous strengths**(Users are likely to share more similar tastes with strong ties than weak ties)

We propose a novel graph neural network **GraphRec** for social recommendations, which can address three aforementioned challenges simultaneously

Contribution

- Propose a novel graph neural network GraphRec;
- 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**;
- We demonstrate the effectiveness of the proposed framework on various real-world datasets

The Proposed Framework

- **Definitions and Notations**

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
$\mathbf{T}$	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

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 $\mathbf{R}$.

- **Overview of the Proposed Framework**

The model consists of three components: **user modeling**, **item modeling**, and **rating prediction**.

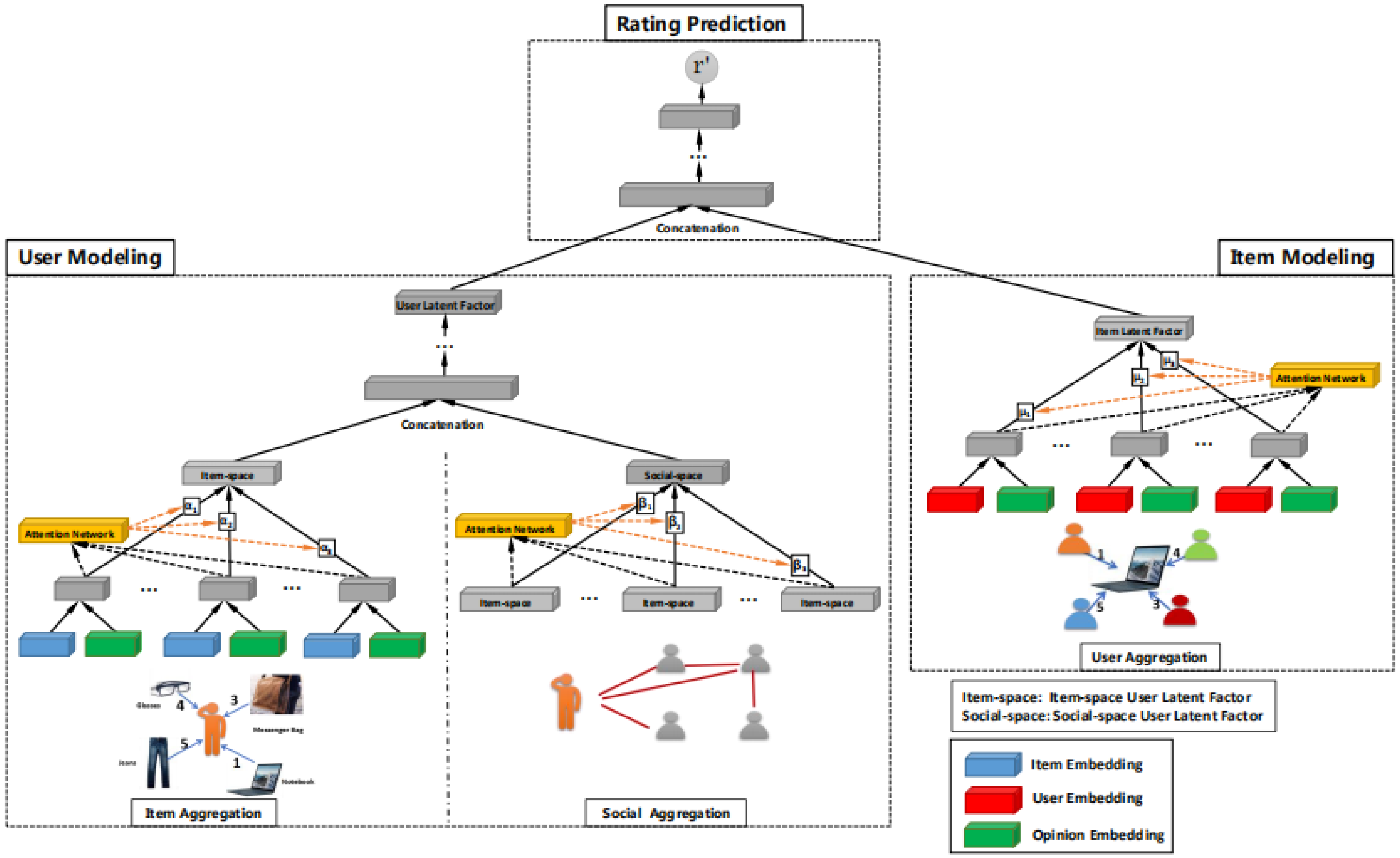
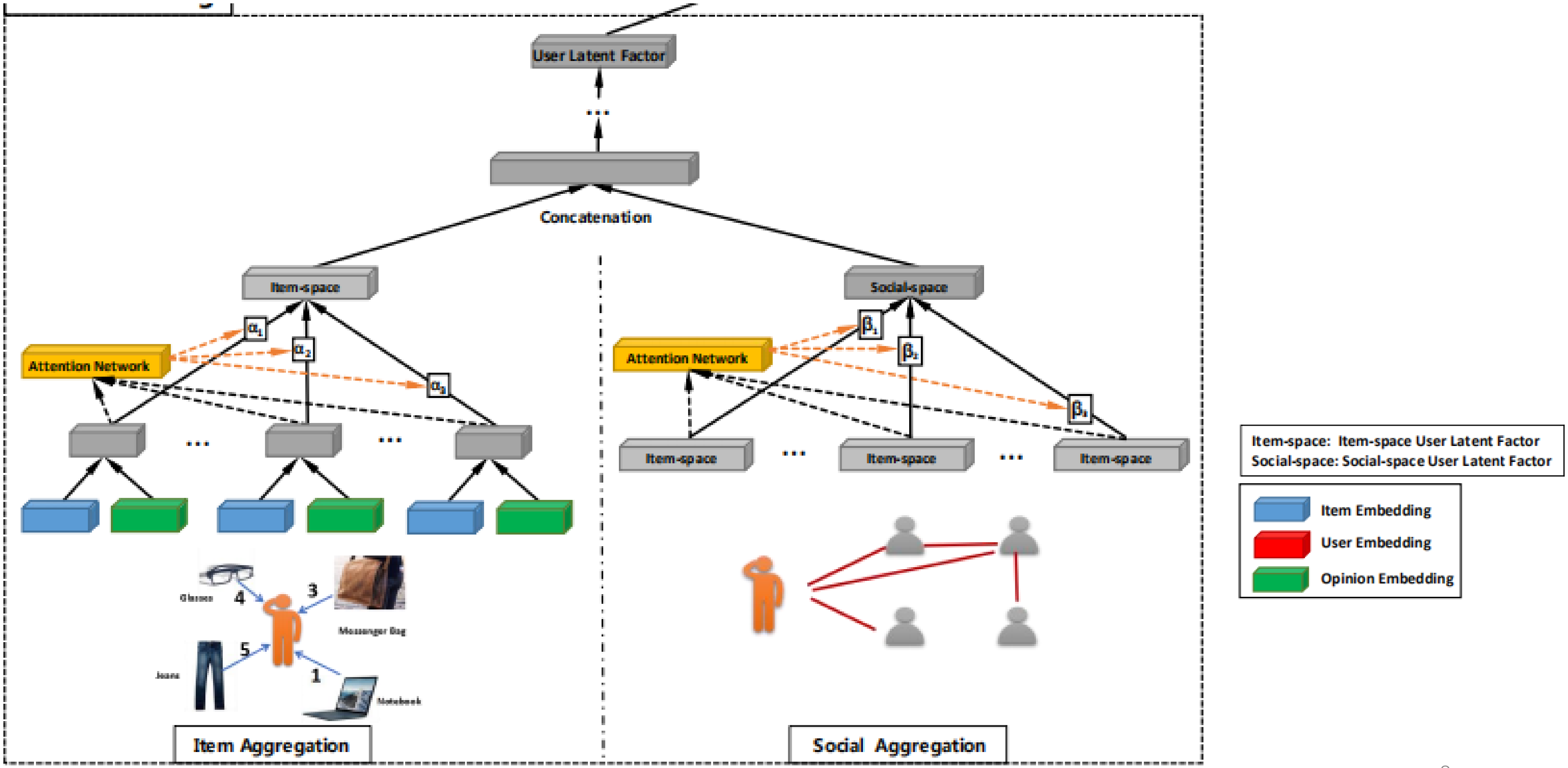


Figure 2: The overall architecture of the proposed model. It contains three major components: user modeling, item modeling, and rating prediction.

- **user modeling**



How to inherently combine the user-item graph and social graph.

Two types of aggregation to learn factors from two graphs

- **item aggregation** jointly capture interactions and opinions in the user-item graph

$$\mathbf{h}_i^I = \sigma(\mathbf{W} \cdot Aggre_{items}(\{\mathbf{x}_{ia}, \forall a \in C(i)\}) + \mathbf{b}) \qquad (1)$$

Item-space user latent factor

activation function

Items aggregation function

The set of items user ui has interacted with

The representation vector that denote opinion-aware interaction between ui and va

$$\mathbf{x}_{ia} = g_v([\mathbf{q}_a \oplus \mathbf{e}_r]) \qquad (2)$$

Multi-Layer Perceptron (MLP)

item embedding

Opinion embedding vector er for each type of opinions r

Concatenation operation

One popular aggregation function for ***Aggreitems*** is **the mean operator**

$$\mathbf{h}_i^I = \sigma(\mathbf{W} \cdot \left\{ \sum_{a \in C(i)} \alpha_i \mathbf{x}_{ia} \right\} + \mathbf{b})$$

$\frac{1}{|C(i)|}$ for all items in the mean-based aggregator.

$$\mathbf{h}_i^I = \sigma(\mathbf{W} \cdot \left\{ \sum_{a \in C(i)} \alpha_{ia} \mathbf{x}_{ia} \right\} + \mathbf{b})$$

Attention mechanisms

where α_{ia} denotes the attention weight of the interaction with v_a in contributing to user u_i's item-space latent factor

$$\alpha_{ia}^* = \mathbf{w}_2^T \cdot \sigma(\mathbf{W}_1 \cdot [\mathbf{x}_{ia} \oplus \mathbf{p}_i] + \mathbf{b}_1) + b_2$$

$$\alpha_{ia} = \frac{exp(\alpha_{ia}^*)}{\sum_{a \in C(i)} exp(\alpha_{ia}^*)}$$

□social aggregation an **attention mechanism** to select social friends

aggregate the item-space user latent factors of neighboring users from the **social graph**

$$\mathbf{h}_i^S = \sigma(\mathbf{W} \cdot Aggre_{neigbhors}(\left\{\mathbf{h}_o^I, \forall o \in N(i)\right\}) + \mathbf{b})$$

the aggregation function on user's neighbors.

$$\mathbf{h}_i^S = \sigma(\mathbf{W} \cdot \left\{\sum_{o \in N(i)} \beta_i \mathbf{h}_o^I\right\} + \mathbf{b})$$

where β_i is fixed to $\frac{1}{|N(i)|}$ for all neighbors for the mean-based aggregator. It assumes that all neighbors contribute equally to the representation of user u_i.

take the element-wise mean of the vectors

$$\mathbf{h}_i^S = \sigma(\mathbf{W} \cdot \left\{\sum_{o \in N(i)} \beta_{io} \mathbf{h}_o^I\right\} + \mathbf{b})$$

$$\beta_{io}^* = \mathbf{w}_2^T \cdot \sigma(\mathbf{W}_1 \cdot [\mathbf{h}_o^I \oplus \mathbf{p}_i] + \mathbf{b}_1) + b_2$$

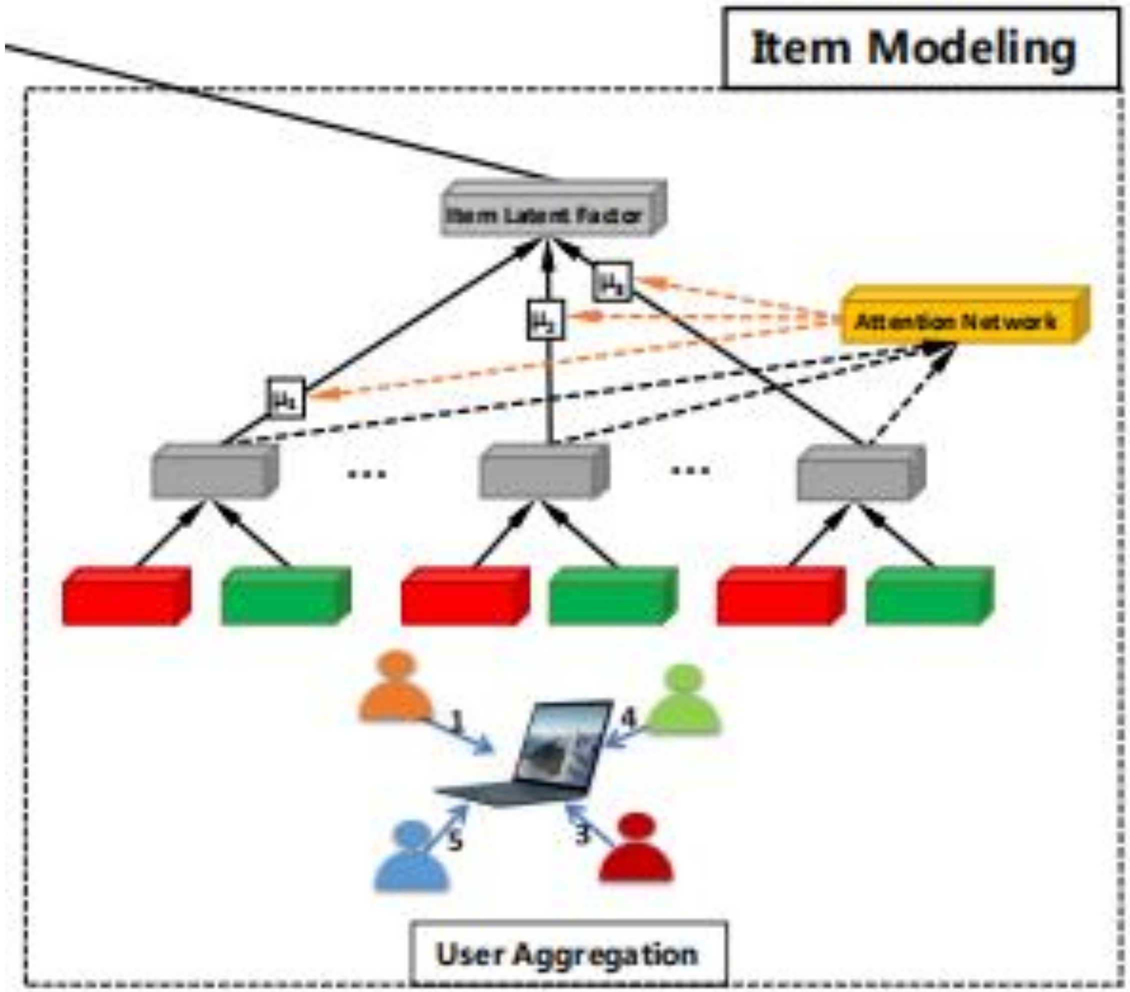
$$\beta_{io} = \frac{exp(\beta_{io}^*)}{\sum_{o \in N(i)} exp(\beta_{io}^*)}$$

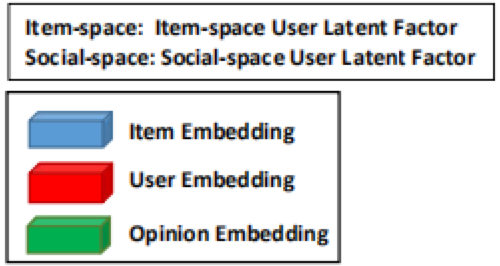
Learning User Latent Factor

We propose to combine these two latent factors to the final user latent factor

$$\mathbf{c}_1 = \left[\mathbf{h}_i^I \oplus \mathbf{h}_i^S\right]$$
$$\mathbf{c}_2 = \sigma(\mathbf{W}_2 \cdot \mathbf{c}_1 + \mathbf{b}_2)$$
$$\cdots$$
$$\mathbf{h}_i = \sigma(\mathbf{W}_l \cdot \mathbf{c}_{l-1} + \mathbf{b}_l)$$

- **item modeling**





☐ User Aggregation

For each item *vj*, we need to aggregate information from the set of users who have interacted with *vj*, **denoted as *B(j)*. Different users can capture the characteristics of the same item in different ways provided by users.**

$$\mathbf{f}_{jt} = g_u([\mathbf{p}_t \oplus \mathbf{e}_r])$$

The opinion-aware interaction user representation

MLP

Opinion embedding

Basic user embedding

$$\mathbf{z}_j = \sigma(\mathbf{W} \cdot Aggre_{users}(\{\mathbf{f}_{jt}, \forall t \in B(j)\}) + \mathbf{b})$$

$$\mathbf{z}_j = \sigma(\mathbf{W} \cdot \left\{ \sum_{t \in B(j)} \mu_{jt} \mathbf{f}_{jt} \right\} + \mathbf{b}) \quad (17)$$

$$\mu^*_{jt} = \mathbf{w}_2^T \cdot \sigma(\mathbf{W}_1 \cdot [\mathbf{f}_{jt} \oplus \mathbf{q}_j] + \mathbf{b}_1) + b_2 \quad (18)$$

capture heterogeneous influence from user-item interactions

$$\mu_{jt} = \frac{exp(\mu^*_{jt})}{\sum_{t \in B_{(j)}} exp(\mu^*_{jt})} \quad (19)$$

- **rating modeling**

With the latent factors of users and items (i.e., $\mathbf{h}_i$ and $\mathbf{z}_j$), we can first concatenate them $[\mathbf{h}_i \oplus \mathbf{z}_j]$ and then feed it into MLP for rating prediction as:

$$\begin{aligned}
\mathbf{g}_1 &= [\mathbf{h}_i \oplus \mathbf{z}_j] && (20)\\
\mathbf{g}_2 &= \sigma(\mathbf{W}_2 \cdot \mathbf{g}_1 + \mathbf{b}_2) && (21)\\
&\cdots\\
\mathbf{g}_{l-1} &= \sigma(\mathbf{W}_l \cdot \mathbf{g}_{l-1} + \mathbf{b}_l) && (22)\\
r'_{ij} &= \mathbf{w}^T \cdot \mathbf{g}_{l-1} && (23)
\end{aligned}$$

where l is the index of a hidden layer, and r'_{ij} is the predicted rating from u_i to v_j.

● Model Training

$$Loss = \frac{1}{2\,|O|} \sum_{i,j \in O} (r'_{ij} - r_{ij})^2$$

the number of observed ratings

Truth rating

Predicted rating

RMSprop as the optimizer in our implementation 微分平方加权平均数

At each time, it randomly selects a training instance and updates each model parameter towards the direction of its negative gradient. There are three embedding in our model, including item embedding q*j*, user embedding p*i*, and opinion embedding e*r* .

the **dropout strategy** has been applied to our model. The idea of dropout is to randomly drop some neurons during the training process.

https://blog.csdn.net/willduan1/article/details/78070086

EXPERIMENT

- ## Experimental Settings

  - *Datasets*

Table 2: Statistics of the datasets

Dataset	Ciao	Epinions
# of Users	7,317	18,088
# of Items	10,4975	261,649
# of Ratings	283,319	764,352
# of Density (Ratings)	0.0368%	0.0161%
# of Social Connections	111,781	355,813
# of Density (Social Relations)	0.2087%	0.1087%

  - *Evaluation Metrics*

Mean Absolute Error (MAE)
Root Mean Square Error (RMSE)

Baselines

three groups of methods including **traditional recommender systems**, **traditional social recommender** systems, and **deep neural network based recommender systems**.

- **PMF** [24]: **P**robabilistic **M**atrix **F**actorization utilizes user-item rating matrix only and models latent factors of users and items by Gaussian distributions.
- **SoRec** [17]: **So**cial **Rec**ommendation performs co-factorization on the user-item rating matrix and user-user social relations matrix.
- **SoReg** [18]: **So**cial **Reg**ularization models social network information as regularization terms to constrain the matrix factorization framework.
- **SocialMF** [13]: It considers the trust information and propagation of trust information into the matrix factorization model for recommender systems.
- **TrustMF** [37]: This method adopts matrix factorization technique that maps users into two low-dimensional spaces: truster space and trustee space, by factorizing trust networks according to the directional property of trust.
- **NeuMF** [11]: This method is a state-of-the-art matrix factorization model with neural network architecture. The original implementation is for recommendation ranking task and we adjust its loss to the squared loss for rating prediction.
- **DeepSoR** [8]: This model employs a deep neural network to learn representations of each user from social relations, and to integrate into probabilistic matrix factorization for rating prediction.
- **GCMC+SN** [1]: This model is a state-of-the-art recommender system with graph neural network architecture. In order to incorporate social network information into GCMC, we utilize the *node2vec* [9] to generate user embedding as user side information, instead of using the raw feature social connections ($T \in \mathbb{R}^{n \times n}$) directly. The reason is that the raw feature input vectors is highly sparse and high-dimensional. Using the network embedding techniques can help compress the raw input feature vector to a low-dimensional and dense vector, then the model can be easy to train.

https://www.cnblogs.com/ZeroTensor/p/10864759.html

Parameter Settings

For all neural network methods, we randomly initialized model parameters with a Gaussian distribution, where the mean and standard deviation is 0 and 0.1,

Performance Comparison of Recommender Systems

Table 3: Performance comparison of different recommender systems

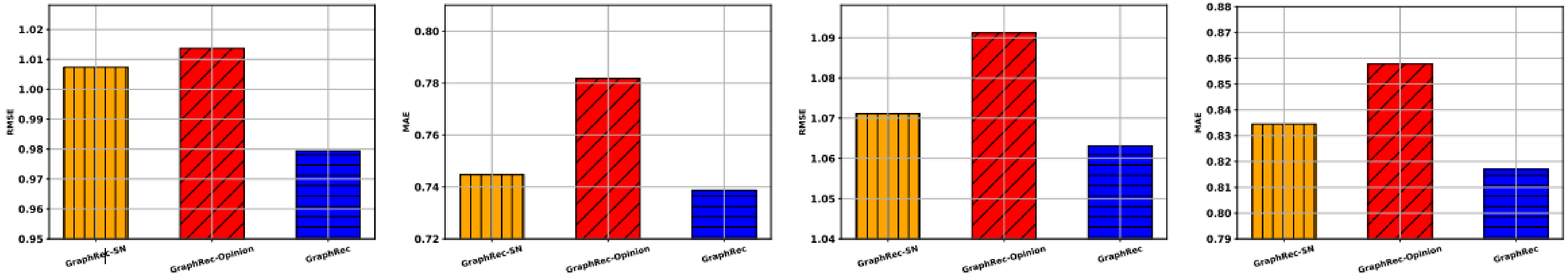
Training	Metrics	Algorithms								
		PMF	SoRec	SoReg	SocialMF	TrustMF	NeuMF	DeepSoR	GCMC+SN	**GraphRec**
Ciao (60%)	MAE	0.952	0.8489	0.8987	0.8353	0.7681	0.8251	0.7813	0.7697	0.7540
	RMSE	1.1967	1.0738	1.0947	1.0592	1.0543	1.0824	1.0437	1.0221	1.0093
Ciao (80%)	MAE	0.9021	0.8410	0.8611	0.8270	0.7690	0.8062	0.7739	0.7526	0.7387
	RMSE	1.1238	1.0652	1.0848	1.0501	1.0479	1.0617	1.0316	0.9931	0.9794
Epinions (60%)	MAE	1.0211	0.9086	0.9412	0.8965	0.8550	0.9097	0.8520	0.8602	0.8441
	RMSE	1.2739	1.1563	1.1936	1.1410	1.1505	1.1645	1.1135	1.1004	1.0878
Epinions (80%)	MAE	0.9952	0.8961	0.9119	0.8837	0.8410	0.9072	0.8383	0.8590	0.8168
	RMSE	1.2128	1.1437	1.1703	1.1328	1.1395	1.1476	1.0972	1.0711	1.0631

(1) social network information is helpful for recommendations;
(2) neural network models can boost recommendation performance
(3) the pro posed framework outperforms representative baselines.

- **Model Analysis**

  - *Effect of Social Network and User Opinions.*

    The proposed framework provides model components to (1) integrate social network information and (2) incorporate users' opinions about the interactions with items.

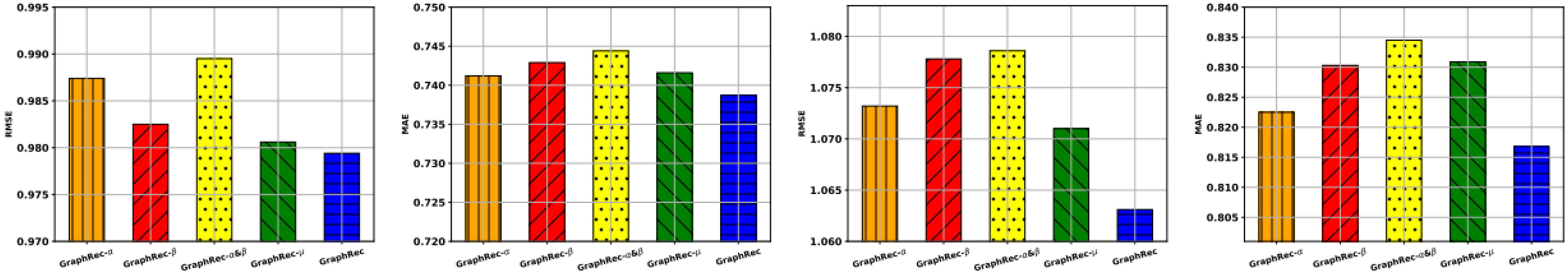


(a) Ciao-RMSE (b) Ciao-MAE (c) Epinions-RMSE (d) Epinions-MAE

Figure 3: Effect of social network and user opinions on Ciao and Epinions datasets.

Effect of Attention Mechanisms.

We further evaluate the key components of GraphRec-*Attention mechanisms*, including **item attention α**, **social attention β**, and **user attention μ**.

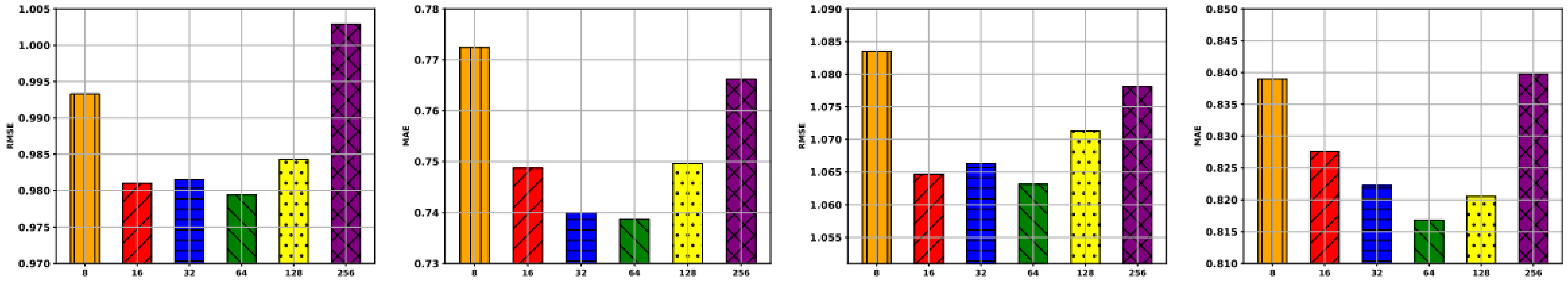


(a) Ciao-RMSE (b) Ciao-MAE (c) Epinions-RMSE (d) Epinions-MAE

Figure 4: Effect of attention mechanisms on Ciao and Epinions datasets.

GraphRec can capture the heterogeneity in aggregation operations of the proposed framework via attention mechanisms, which can boost the recommendation performance

☐ *Effect of Embedding Size.*



(a) Ciao-RMSE (b) Ciao-MAE (c) Epinions-RMSE (d) Epinions-MAE

Figure 5: Effect of embedding size on Ciao and Epinions datasets.

to analyze the effect of embedding size of **user embedding p** , **item embedding q**, and **opinion embedding e**, on the performance of our model.

CONCLUSION AND FUTURE WORK

- Our experiments reveal that the opinion information plays a crucial role in the improvement of our model performance.
- our GraphRec can differentiate the ties strengths by considering heterogeneous strengths of social relations.
- Exploring graph neural networks for recommendation with attributes would be an interesting future direction.
- we will consider building dynamic graph neural networks for social recommendations with dynamic

New Idea

- GNN和GCN的应用
- 模型中各个部分的重要性的分析
- Attention机制的使用

谢　谢