ConsisRec: Enhancing GNN for Social Recommendation via Consistent Neighbor Aggregation

Liangwei Yang [1] Zhiwei Liu [1] Yingtong Dou [1] Jing Ma [2] Philip S. Yu [1]

[1]University of Illinois at Chicago [2]Sichuan University

Background

As shown in Fig. 1, social recommendation aims to fuse social links with user-item interactions to alleviate the cold-start problem for rating prediction.

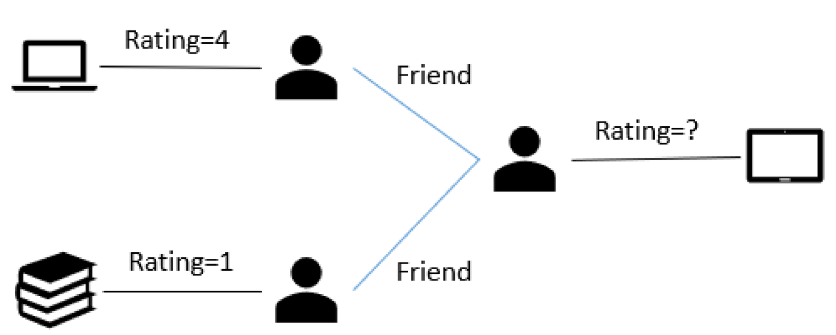

Figure 1. Illustration of social recommendation

The homophily and social influence character of social correlation indicate the usefulness of incorporating social dimension into recommender system.

- **Homophily:** People tend to be connected if they are more similar to each other.
- **Social influence:** People would influence its connections from his or her own preference

Graph neural network (GNN) learns node embedding by aggregating its neighbors' information as equation 1. GNN based models such as GCMC [1], GraphRec [2] are capturing more attention for their good performance.

$$\mathbf{h}_v^{(l)} = \mathbf{h}_v^{(l-1)} \oplus \text{AGG}^{(l)}\{\mathbf{h}_i^{(l-1)} | i \in \mathcal{N}_v\} \tag{1}$$

- AGG: Aggregator of neighbor information, such as max pooling, weighted average.
- $\oplus$: Combination of neighbor information and center node information, such as linear layer, addition.

Motivation

GNN assumes neighbors share similar features, context, and labels (smoothness), but this assumption is no longer hold in social recommendation because of the inconsistency problems.

Social inconsistency problems

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

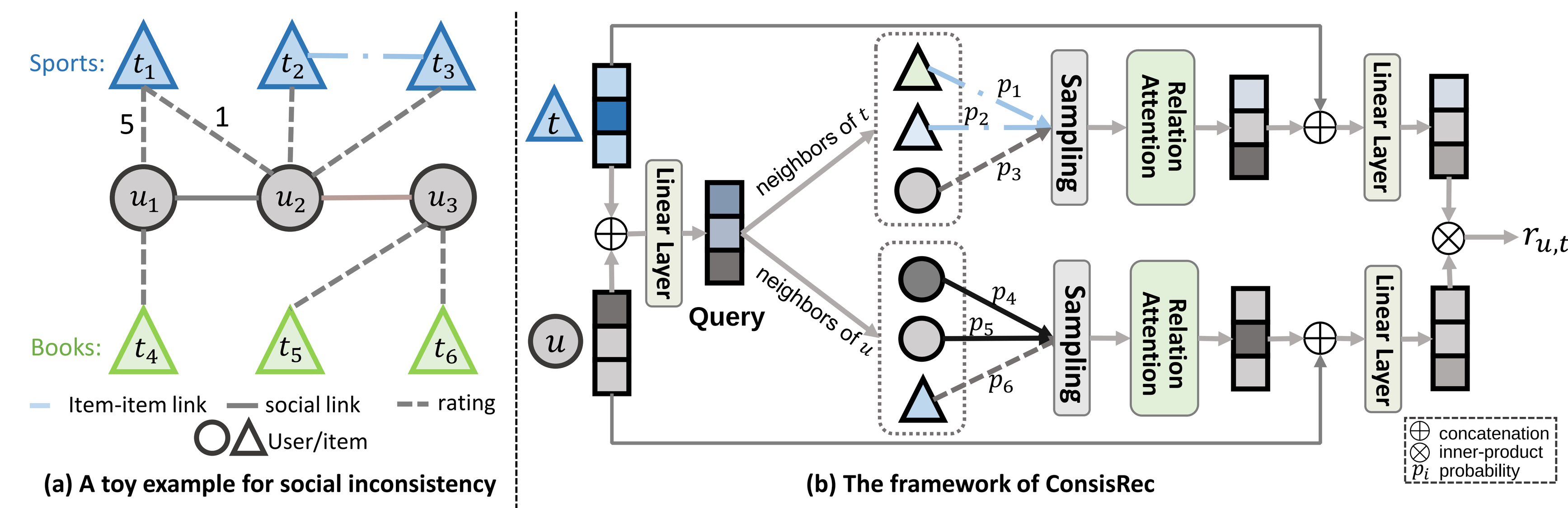

Figure 2. Social inconsistency illustration and model framework

Proposed Model (ConsisRec)

Sample consistent neighbors to cope with context level inconsistency. Use relation attention to treat different relation types.

Query layer: fuse user and item information of current rating procedure. We apply a linear layer on the concated user and item embedding as equation 2.

$$\mathbf{q}_{u,t} = \sigma\left(\mathbf{W}_q^\top(\mathbf{e}_u \oplus \mathbf{e}_t)\right), \tag{2}$$

Neighbor sampling: sample informative neighbors based on consistency score. We firstly calculate consistency score based on embedding similarity to query embedding as equation 3, and sample neighbors based on the normalized consistency score calculated by equation 4.

$$s^{(l)}(i;\mathbf{q}) = \exp(-\|\mathbf{q} - \mathbf{h}_i^{(l)}\|_2^2), \tag{3}$$

$$p^{(l)}(i;\mathbf{q}) = s^{(l)}(i;\mathbf{q}) / \sum_{j\in\mathcal{N}_v} s^{(l)}(j;\mathbf{q}). \tag{4}$$

Relation Attention: Use relation embedding to deal with various relation types. Node embedding $\mathbf{h}_i^{(l-1)}$ is firstly concated with its relation embedding $\mathbf{e}_{r_i}$. Then a linear layer is used to get the attention score in equation 5. The normalized attention score is used as neighbor embedding weight for aggregation in equation 6.

$$\alpha_i^{(l)} = \frac{\exp(\mathbf{w}_a^\top(\mathbf{h}_i^{(l-1)} \oplus \mathbf{e}_{r_i}))}{\sum_{j=1}^{Q}\exp(\mathbf{w}_a^\top(\mathbf{h}_j^{(l-1)} \oplus \mathbf{e}_{r_j}))} \tag{5}$$

$$\text{AGG}^{(l)} = \sum_{i=1}^{Q} \alpha_i^{(l)} \cdot \mathbf{h}_i^{(l-1)}, \tag{6}$$

Then we concat the center node and its aggregated neighbor's embedding. After a linear layer and a Relu activation function, we obtain the node embedding on l-th layer.

$$\mathbf{h}_v^{(l)} = \sigma\left(\mathbf{W}^{(l)\top}\left(\mathbf{h}_v^{(l-1)} \oplus \text{AGG}^{(l)}\{\mathbf{h}_i^{(l-1)} | i \in \mathcal{N}_v\}\right)\right), \tag{7}$$

Experiment Results

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

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

- GNN based methods tend to be better than matrix factorization based methods.
- ConsisRec performs the best among all the methods.

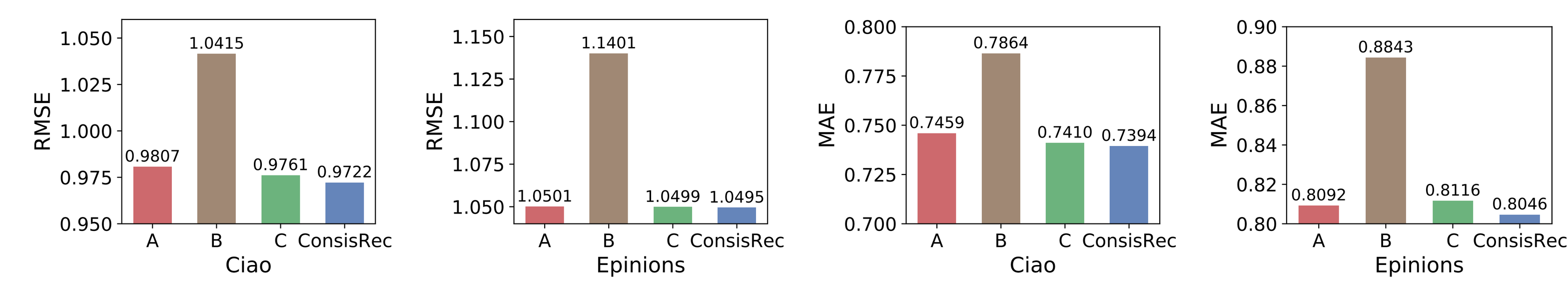

Figure 3. Ablation study of . A, B, and C are built by removing query layer, neighbor sampling and relation attention separately.

- ConsisRec always performs the best, indicting all the components are useful
- Remove neighbor sampling dramatically spoils the performance, which shows the importance of selecting consistent neighbors.

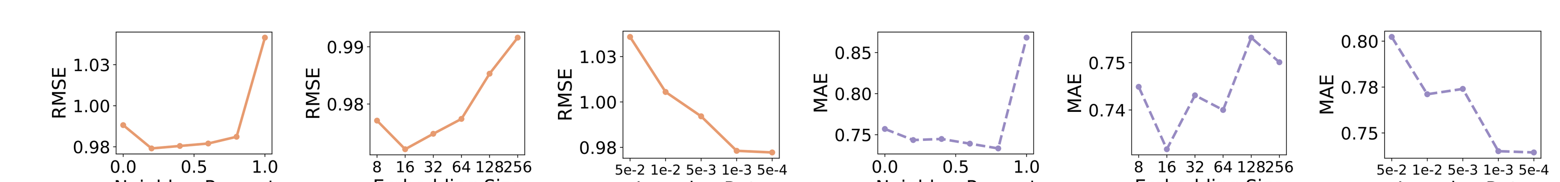

Figure 4. Parameter sensitivity on Ciao dataset.

- There is an obvious error increment when neighbor percent rising from 0.8 to 1.0, which results from aggregating inconsistent neighbors.
- Suitable embedding size should be selected.
- Learning rate has a critical impact on model performance.

References

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

[2] Wenqi Fan, Yao Ma, Qing Li, Yuan He, Eric Zhao, Jiliang Tang, and Dawei Yin. Graph neural networks for social recommendation. In *The World Wide Web Conference, 2019*, pages 417–426. ACM, 2019.