GNN for Social Recommender Systems 2

Lab Seminar (10/04)

HTET ARKAR
School of Computer Science and Engineering
Chung-Ang University

Paper Review

A Graph Neural Network Framework for Social Recommendations

(GraphRec+)

Wenqi Fan[1], Yao Ma[2], Qing Li[3], *Senior Member, IEEE*, Jianping Wang[4], Guouong Cai[2], Jiliang Tang[2], and Dawei Yin5

[1,4]CIty University of Hong Kong, [2]Michigan State University, [1,3]The Hong Kong Polytechnic University, [5]Baidu Inc.

IEEE Transactions on Knowledge and Data Engineering, 2020

Content

- Introduction
- Movitation
- Proposed Method
- Architecture
- Experimental Evaluation
- Conclusion & Future Work

Introduction

❖ Recommender Systems (RS)

- To suggest to the user a personalized list of items
  - That are likely to be clicked on or purchased

- Collaborative Filtering (CF): one of most popular techniques used
  - To model users' preference for items by utilizing the history of user-item interactions

Social Recommender Systems (SocialRS)

- **Social relations of users can play a important role**
  - In helping RS filter information
  - In profiling users' preferences towards items

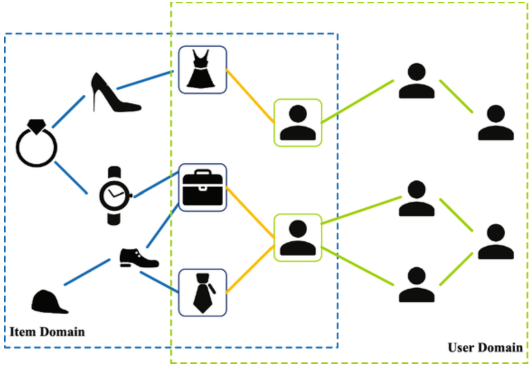
- **Utlizing Social Relations has been used in RS**
  - To greatly boost the performance



Building SocialRS with GNN

- Why Graph Neural Networks?
  - GNNs provide a powerful method for learning meaningful representations
  - By integrating node information and topological structure

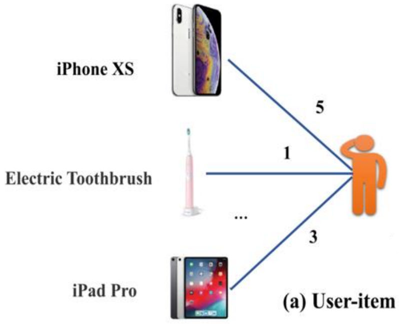
- Data Representation in Social Recommendations (SR):
  - User-Item Graph
  - Social Graph
  - Item-Item Graph



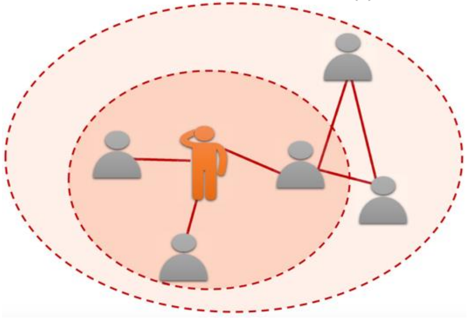
Data in Social Recommendation

- Data can be naturally represented as **graph data**
  - **User-item graph** for denoting interactions between users and items
  - **Social graph** for denoting the relationships between users

  - SR : Incorparate social network info into user and item latent factors learning



(a) User-item Graph

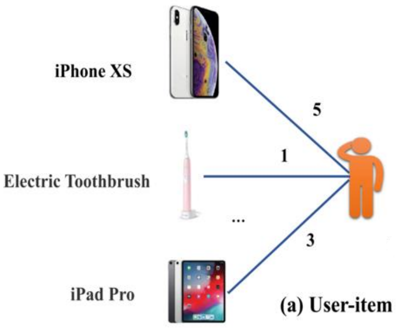

(b) User-User Social Graph

SR: social recommendation

Data in Social Recommendation (Cont'd)

- Two graphs provide info about users from different perspectives

- User-item graph contains **user-item interactions + user's opinions on items**
- **User's opinions**
  - Ratings and reviews
  - Valuable signals associated with users and items



(a) User-item Graph

W. Fan et al., "Graph neural networks for social recommendation," in Proc. World Wide Web Conf., 2019, pp. 417–426.

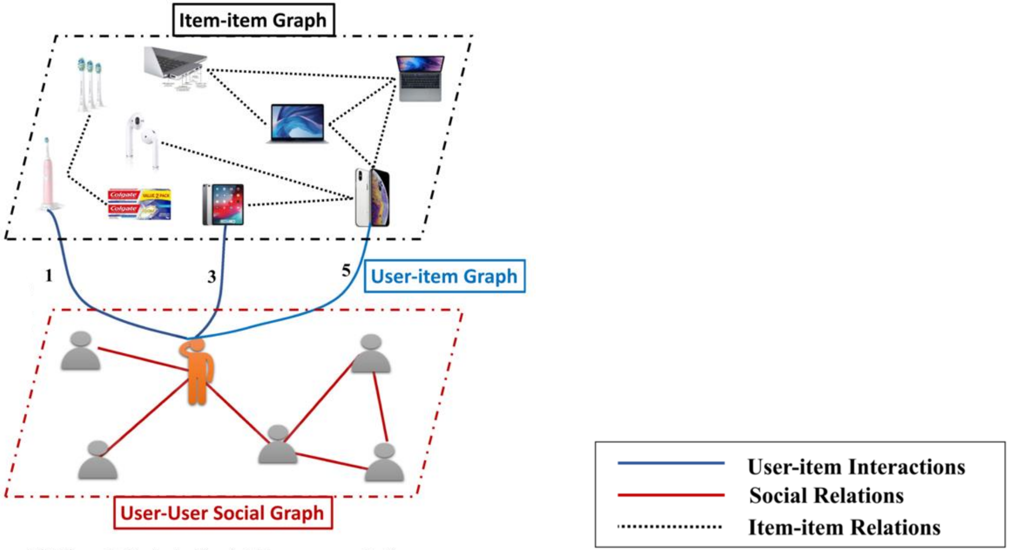
Data in Social Recommendation (cont'd)

❖ Item-Item Graph

- Relationships between items
  - ☐ Items in real-world settings (e.g., e-commerce, electronics) are often related or similar
  - ☐ E.g., users who buy an **iPhone** are likely to buy **AirPods** (both designed by Apple).
- Bundled Items for Comparison
  - ☐ When purchasing a MacBook, similar items (MacBook Air/Pro) are shown together for easy comparison
- Integrating I-I graph into the process of learning better item representations

Data in Social Recommendation (cont'd)

- It can be represented as a **heterogeneous graph**
  - Users – bridging the user-item graph and social graph
  - Items – bridging the user-item graph and I-I graph



(d) Graph Data in Social Recommendation

Motivation

- **1. How to inherently combine these graphs?**
  - Learning representations of items and users are the key to building SocialRS

$\left.\begin{array}{c}\text{U-I graph}\\ \\ \text{Social graph}\end{array}\right\}$ user representation

$\left.\begin{array}{c}\text{U-I graph}\\ \\ \text{I-I graph}\end{array}\right\}$ item representation

- **2. How to capture interactions and opinions between users & items jointly**
  - User-item interactions alone are insufficient
  - **User opinions** are crucial to fully model user preferences

- **3. How to distinguish social relations with heterogeneous strengths**
  - Scoial networks include both **strong** and **weak** ties
  - Treating all social connections equally can lead to poor performance

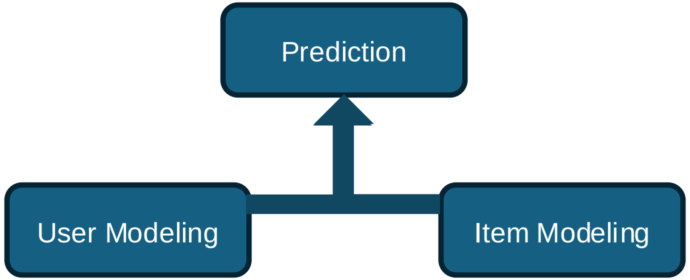
Proposed Method

- **Novel GraphRec+ Framework**:
  - Coherent modeling of multiple types of graphs
- **Joint Modeling of Interactions and Opinions:**
  - Integrating both interactions and opinions in the user-item graph
- **Heterogeneous Strengths in Social Relations**:
  - Differentiating strong and weak ties using an attention mechanism
- **Comprehensive Experiments**:
  - Demonstrating the effectiveness on three real-world datasets

Proposed Framework

❖ **Three Main Components:**

- 1. **User Modeling** : Aggregates both item and social graph information
- 2. **Item Modeling** : Uses both user-item interactions and item-item similarities
- 3. **Rating Prediction**: Combines user and item representations for final predictions



Model Architecture Overview

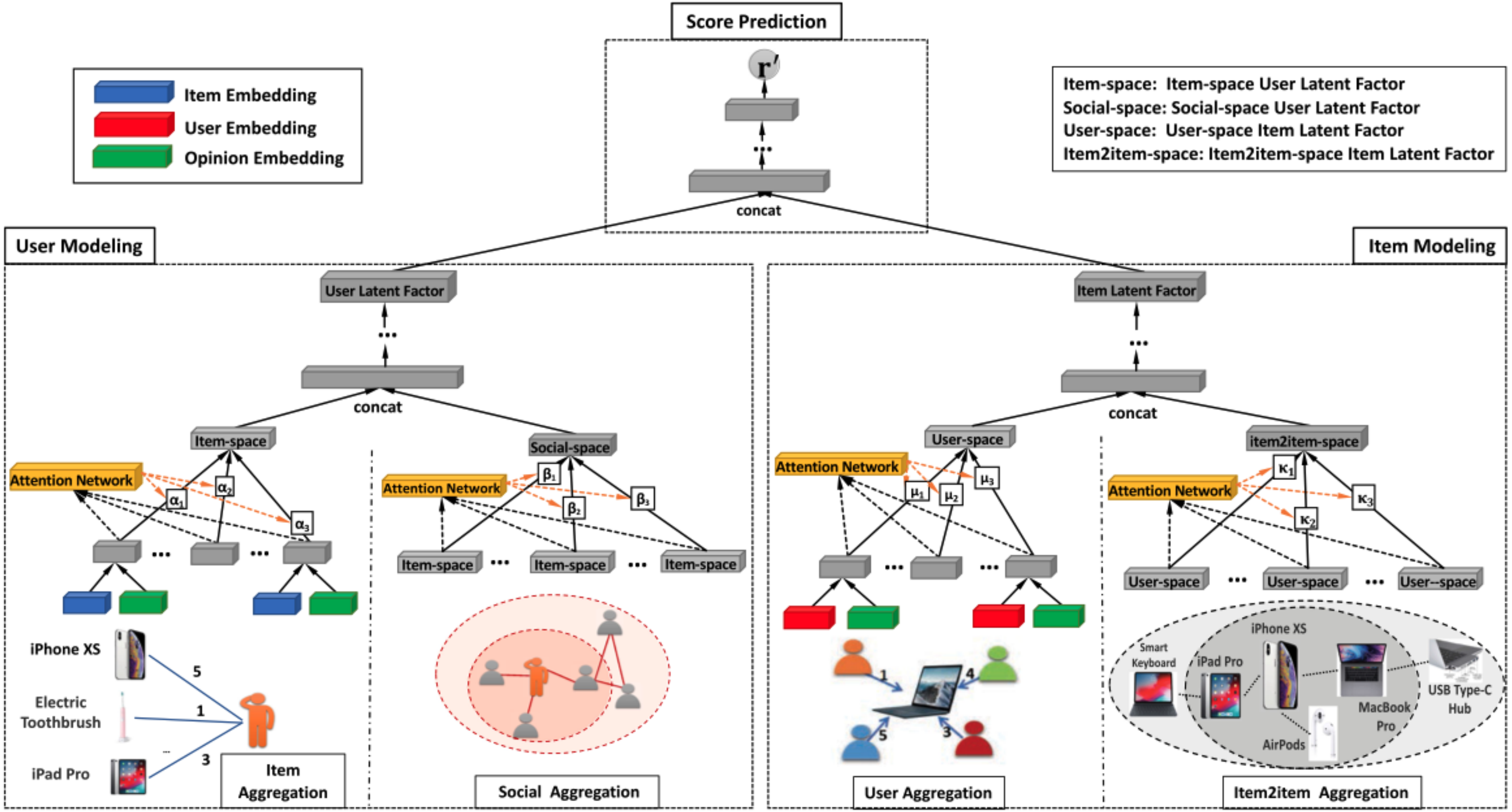


Fig. 2. The overall architecture of the proposed model. It contains three major components: User modeling, item modeling, and score prediction.

User Modeling u_i

❖ **1. Learning item-space user latent factor h_i^I**

- Item aggregation:

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

- Opinion-aware interaction representation x_{ia}

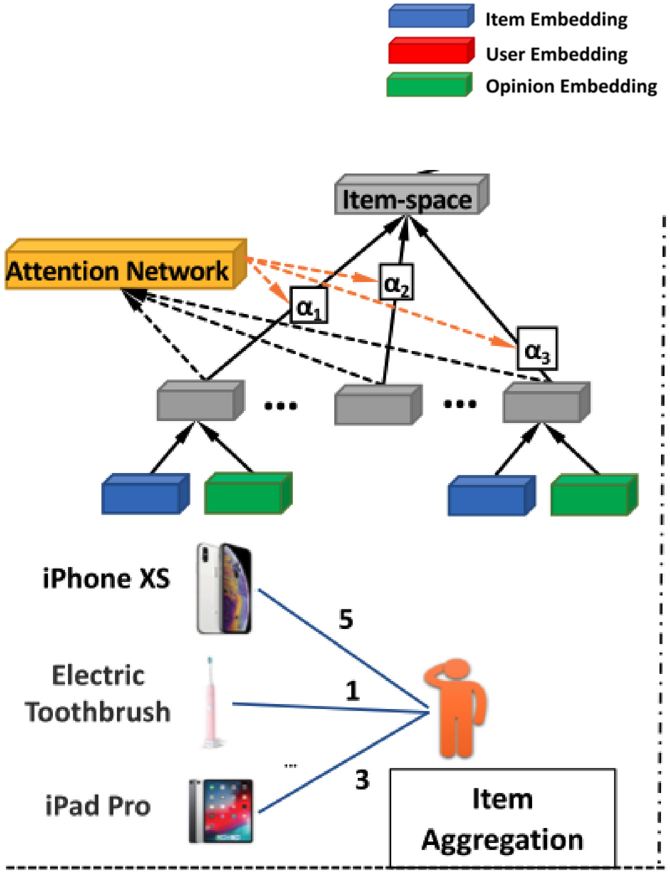
$$\mathbf{x}_{ia} = g_v([\mathbf{q}_a \oplus \mathbf{e}_r])$$

- Item-space user latent factor:

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

α_{ia} : item attention

$C(i)$: set of items, user has interacted with
g_v : Multi-Layer Perceptron (MLP)
e_r : opinion embedding vector
q_a : item embedding



DMAIS

User Modeling u_i (cont'd)

❖ **2. Learning social-space user latent factor h_i^S**

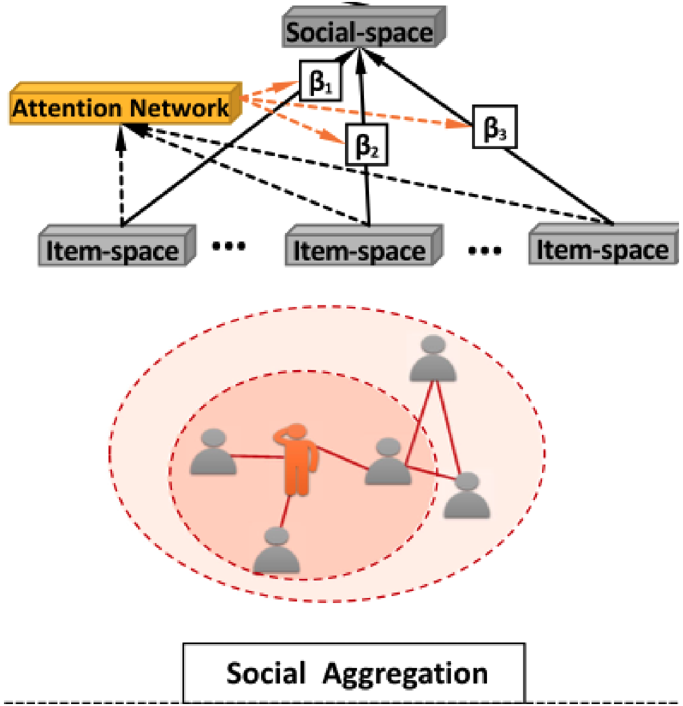
- Social aggregation:

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

- Social-space user latent factor:

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

β_{io} : social attention



$N(i)$: social neighbors

User Modeling u_i (cont'd)

❖ 3. Learning User Latent Factor h_i

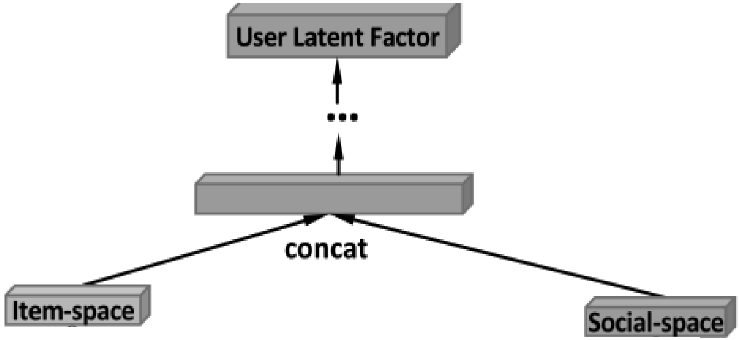
- To better learn user latent factiors
- Item-space and social-space user latent factors need to be considered together

$$\mathbf{c}_1^u = \left[\mathbf{h}_i^I \oplus \mathbf{h}_i^S\right]$$

$$\mathbf{c}_2^u = \sigma(\mathbf{W}_2 \cdot \mathbf{c}_1^u + \mathbf{b}_2)$$

$$\cdots$$

$$\mathbf{h}_i = \sigma(\mathbf{W}_l \cdot \mathbf{c}_{l-1}^u + \mathbf{b}_l)$$



Item Modeling v_j

❖ **1. Learning user-space item latent factor h_j^U**

- User aggregation:

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

- Opinion-aware interaction representation f_{jt}

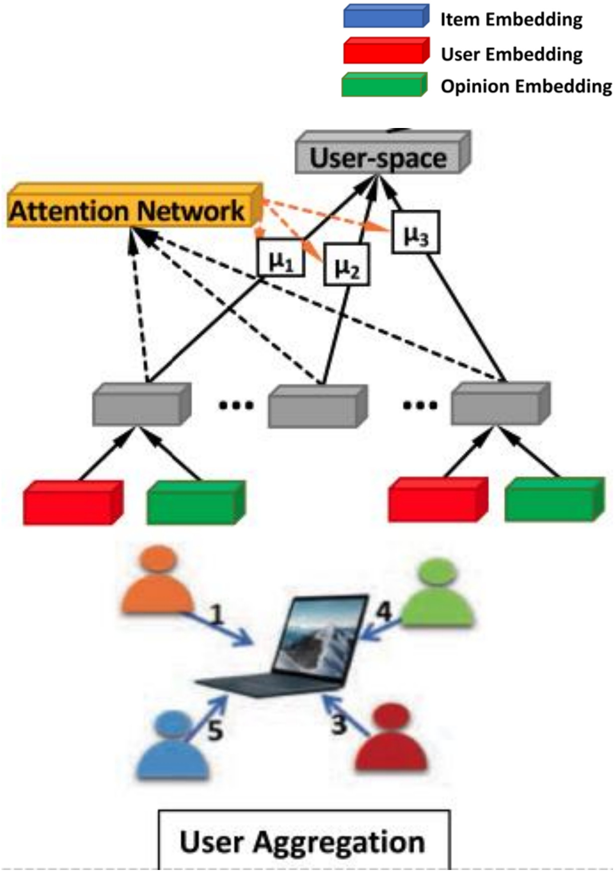
$$\mathbf{f}_{jt} = g_u([\mathbf{p}_t \oplus \mathbf{e}_r])$$

- User-space user latent factor:

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

μ_{jt} : user attention

$B(j)$: set of users who has interacted with v_j
g_u : Multi-Layer Perceptron (MLP)
e_r : opinion embedding vector
p_t : user embedding



Item Modeling v_j (cont'd)

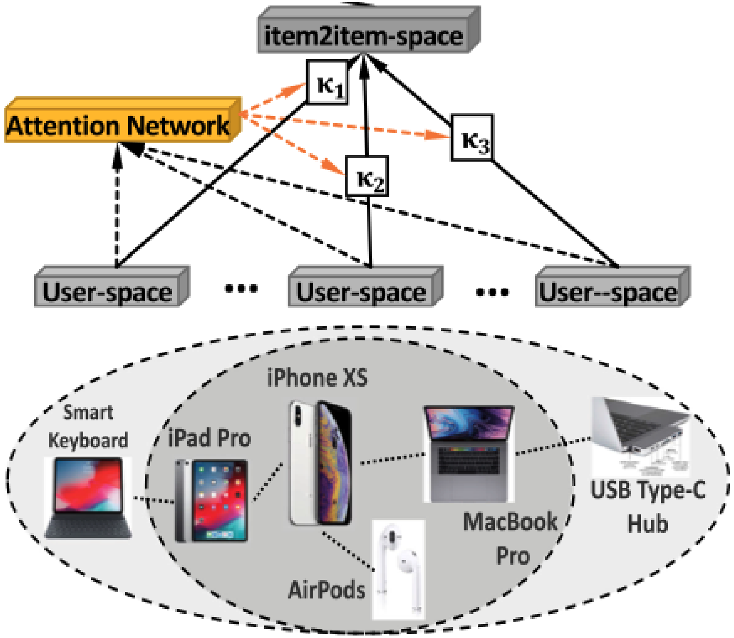
- **2. Learning item2item-space item latent factor h_j^V**
  - Item2item aggregation:

$$\mathbf{h}_j^V = \sigma(\mathbf{W} \cdot AGG_{Item2item}(\{\mathbf{h}_k^U, \forall k \in M(j)\}) + \mathbf{b})$$

  - Item2item-space item latent factor:

$$\mathbf{h}_j^V = \sigma(\mathbf{W} \cdot \left\{ \sum_{k \in M(j)} \kappa_{jk} \mathbf{h}_k^U \right\} + \mathbf{b})$$

κ_{jk} : important weight



Item2item Aggregation

$M(j)$: similar or related items

Item Modeling v_j (cont'd)

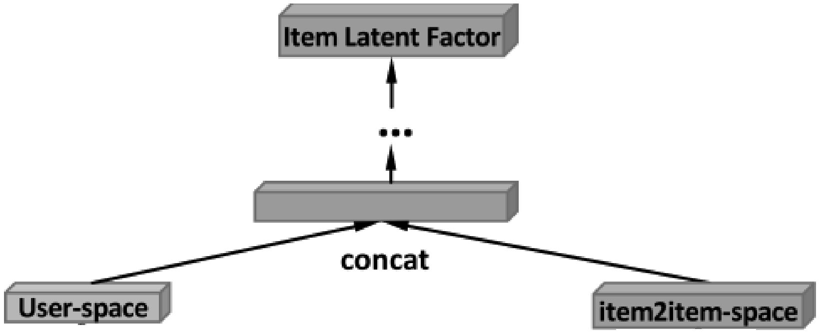
- 3. Learning Item Latent Factor z_j
  - Combine the different item representations together for final one

$$\mathbf{c}_1^v = \left[\mathbf{h}_j^U \oplus \mathbf{h}_j^V\right]$$

$$\mathbf{c}_2^v = \sigma(\mathbf{W}_2 \cdot \mathbf{c}_1^v + \mathbf{b}_2)$$

$$\ldots$$

$$\mathbf{z}_j = \sigma(\mathbf{W}_l \cdot \mathbf{c}_{l-1}^v + \mathbf{b}_l).$$



Prediction & Training

❖ Prediction

- Recommendation task: rating prediction
- Concatenate and then feed into MLP

$$\mathbf{g}_1 = [\mathbf{h}_i \oplus \mathbf{z}_j]$$

$$\mathbf{g}_2 = \sigma(\mathbf{W}_2 \cdot \mathbf{g}_1 + \mathbf{b}_2)$$

$$\ldots$$

$$\mathbf{g}_{l-1} = \sigma(\mathbf{W}_l \cdot \mathbf{g}_{l-1} + \mathbf{b}_l)$$

$$r'_{ij} = \mathbf{w}^T \cdot \mathbf{g}_{l-1},$$

❖ Model Training:

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

GraphRec (2019)

- W. Fan et al., "Graph neural networks for social recommendation," in Proc. World Wide Web Conf., 2019, pp. 417–426
- **Model:**

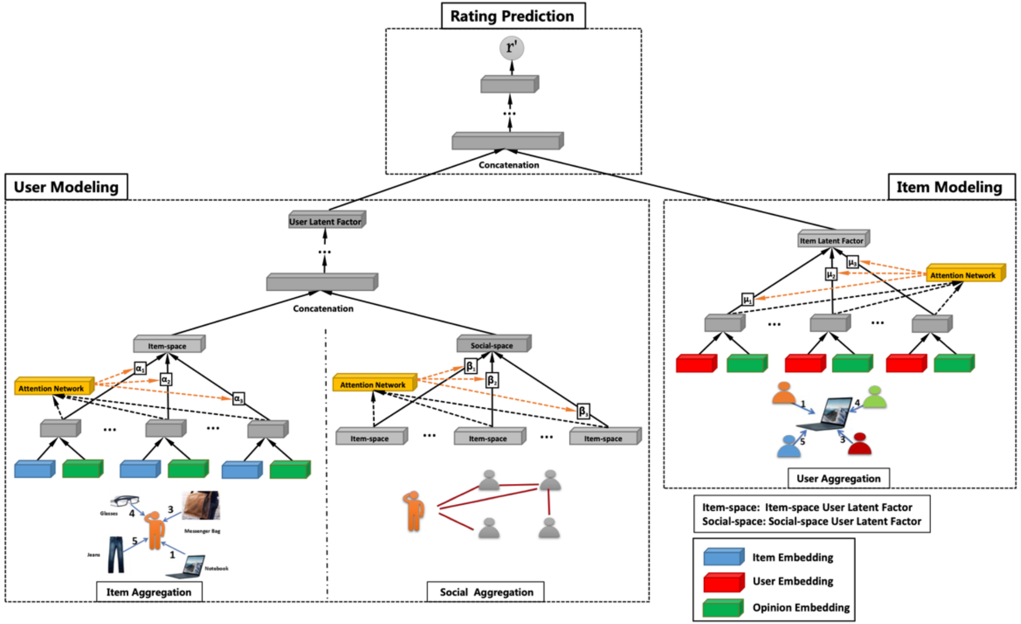


Figure 2: The overall architecture of the proposed model. It contains three major components: user modeling, item modeling, and rating prediction.

Experiment

❖ Performance Comparison of RSs

* The lower the value, the better the performance

Datasets		Ciao (80%)		Ciao (60%)		Epinions (80%)		Epinions (60%)		Flixster (80%)		Flixster (60%)	
Metrics		RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
Algorithms	PMF	1.1238	0.9021	1.1967	0.9520	1.2128	0.9952	1.2739	1.0211	1.0042	0.7559	1.0793	0.8296
	SoRec	1.0652	0.8410	1.0738	0.8489	1.1437	0.8961	1.1563	0.9086	0.9693	0.7303	0.9974	0.7537
	SoReg	1.0848	0.8611	1.0947	0.8987	1.1703	0.9119	1.1936	0.9412	0.9727	0.7415	1.0372	0.7736
	SocialMF	1.0501	0.8270	1.0592	0.8353	1.1328	0.8837	1.1410	0.8965	0.9705	0.7378	1.0060	0.7647
	TrustMF	1.0479	0.7690	1.0543	0.7681	1.1395	0.8410	1.1505	0.8550	0.9380	0.7091	0.9978	0.7480
	NeuMF	1.0617	0.8062	1.0824	0.8251	1.1476	0.9072	1.1645	0.9097	0.9698	0.7420	1.0540	0.7663
	DeepSoR	1.0316	0.7739	1.0437	0.7813	1.0972	0.8383	1.1135	0.8520	0.9548	0.7190	0.9913	0.7462
	GCMC+SN	0.9931	0.7526	1.0221	0.7697	1.0711	0.8590	1.1004	0.8602	0.9613	0.7351	0.9914	0.7606
	GraphRec	0.9794	0.7387	1.0093	0.7540	1.0631	0.8168	1.0878	0.8441	0.9452	0.7175	0.9857	0.7366
	GraphRec+	0.9787	0.733	0.9962	0.7446	1.0576	0.8093	1.0819	0.8336	0.9303	0.7047	0.9669	0.7289
Improvement* (%)		0.07%	0.77%	1.30%	1.25%	0.52%	0.92%	0.54%	1.24%	0.82%	0.63%	1.91%	1.00%

**The value indicates the percentage of improvements gained by GraphRec+ compared to the strongest baseline.*

- PMF : Utilizing user-item rating only (CF)
- SoRec : Social Recommendation (user-item rating + SN)
- SoReg : SN info as regularization terms
- SocialMF : Trust info & Propagation into MF
- TrustMF : MF, truster & trustee space

- NeuMF : CF with NN
- DeepSoR : DNN with SN
- GCMC+SN : GNN with SN
- GraphRec : GNN with SN

DMAIS

Performance Comparison of RSs

- **Traditional Matrix Factorization Models:**
  - **SoRec, SoReg, SocialMF, and TrustMF** outperform **PMF**
  - These models leverage both **rating and social network information**
  - **Social network information complements rating information** leading to better performance

Datasets		Ciao (80%)		Ciao (60%)		Epinions (80%)		Epinions (60%)		Flixster (80%)		Flixster (60%)	
Metrics		RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
Algorithms	PMF	1.1238	0.9021	1.1967	0.9520	1.2128	0.9952	1.2739	1.0211	1.0042	0.7559	1.0793	0.8296
	SoRec	1.0652	0.8410	1.0738	0.8489	1.1437	0.8961	1.1563	0.9086	0.9693	0.7303	0.9974	0.7537
	SoReg	1.0848	0.8611	1.0947	0.8987	1.1703	0.9119	1.1936	0.9412	0.9727	0.7415	1.0372	0.7736
	SocialMF	1.0501	0.8270	1.0592	0.8353	1.1328	0.8837	1.1410	0.8965	0.9705	0.7378	1.0060	0.7647
	TrustMF	1.0479	0.7690	1.0543	0.7681	1.1395	0.8410	1.1505	0.8550	0.9380	0.7091	0.9978	0.7480

* PMF uses only rating data

Performance Comparison of RSs (Cont'd)

Neural Network-based Models:

- **NeuMF** outperforms PMF significantly
- Both use only rating information, but NeuMF utilizes **neural network architecture**
- Highlighting the **power of neural networks** in RS

Advanced Models with Social Network Integration:

- **DeepSoR** and **GCMC+SN** outperform SoRec, SoReg, SocialMF, and TrustMF
- These models leverage **both rating and social network data** using neural networks
- NN-based SocialRSs show **consistent performance improvements**

NeuMF	1.0617	0.8062	1.0824	0.8251	1.1476	0.9072	1.1645	0.9097	0.9698	0.7420	1.0540	0.7663
DeepSoR	1.0316	0.7739	1.0437	0.7813	1.0972	0.8383	1.1135	0.8520	0.9548	0.7190	0.9913	0.7462
GCMC+SN	0.9931	0.7526	1.0221	0.7697	1.0711	0.8590	1.1004	0.8602	0.9613	0.7351	0.9914	0.7606

DMAIS

Performance of GraphRec+

❖ **GCMC+SN**

- Best among baselines due to its **powerful representation learning** using GNNs
- Integrate **node information and topological structure**

❖ **GraphRec+ Performance**:

- Consistently outperforms all baseline methods
- Provides advanced model components to integrate **rating, SN Info**, and **item-item relations**

❖ **Compared to GraphRec:**

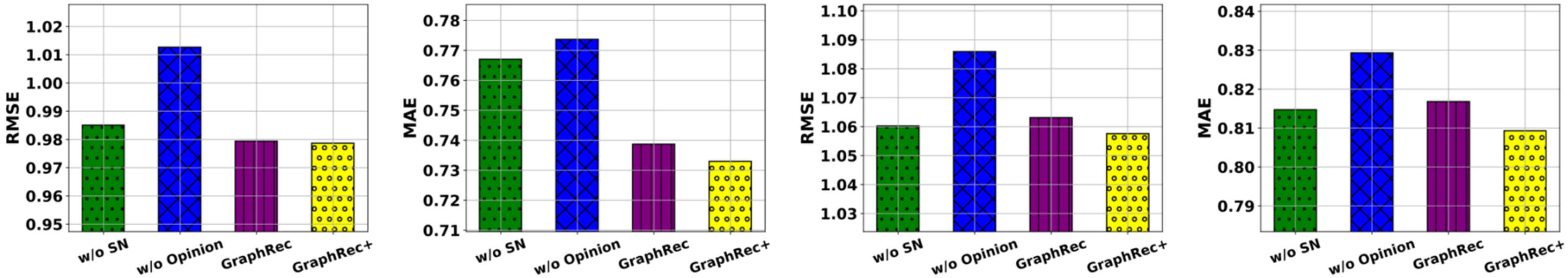
- GraphRec+ shows **better performance, especially under smaller training sets** (60% dataset condition)
- Indicates that GraphRec+ is more robust in learning representations even with less data

GraphRec	0.9794	0.7387	1.0093	0.7540	1.0631	0.8168	1.0878	0.8441	0.9452	0.7175	0.9857	0.7366
GraphRec+	0.9787	0.733	0.9962	0.7446	1.0576	0.8093	1.0819	0.8336	0.9303	0.7047	0.9669	0.7289
(%)	0.07%	0.77%	1.30%	1.25%	0.52%	0.92%	0.54%	1.24%	0.82%	0.63%	1.91%	1.00%

* GraphRec : not include item-item graph

Model Analysis

❖ Ablation Study 1



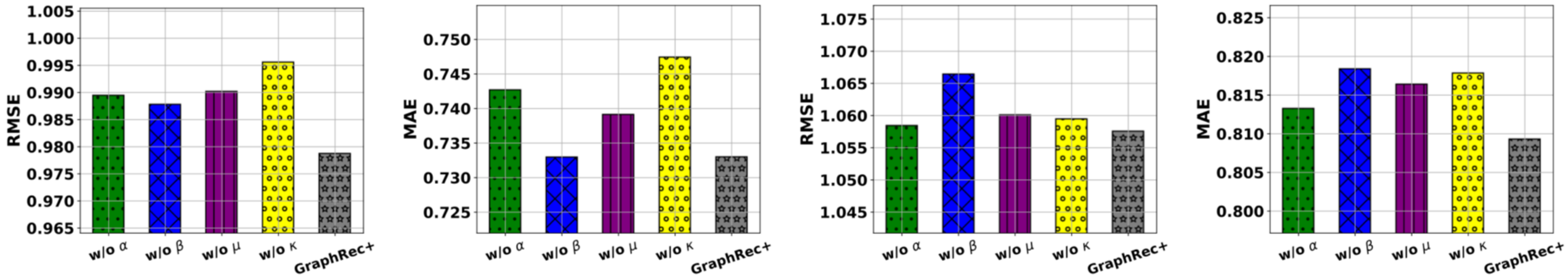
(a) Ciao-RMSE (b) Ciao-MAE (c) Epinions-RMSE (d) Epinions-MAE

- Effectiveness of Social Network Information
- Without Opinions in Interaction, performance deteriorates significantly
- Similar or related items are benefical to profile items, help improve performance

Model Analysis (Cont'd)

❖ Ablation Study 2

- Effect of attenetion mechanisms
- Differentiating the importance of each local neighbor in the graph
- Helping boost the recommendation performance

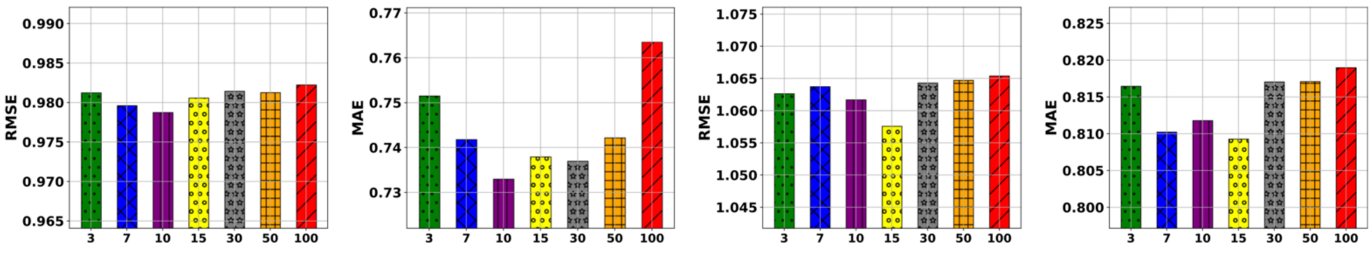


(a) Ciao-RMSE (b) Ciao-MAE (c) Epinions-RMSE (d) Epinions-MAE

Model Analysis (Cont'd)

❖ Effect of Top-k Similar/Related Items on Model Performance

- Explore the impact of different top-k values for item-item relationships
- Optimal Value: k=10 for Ciao, k = 15 for Epinions
- Performance Trend:
  - ☐ **Intitial Increase** : Performance improves as K increases
    - ○ Indicating that including more similar/related items is beneficial
  - ☐ **Performance Degradation**: When K becomes too large
    - ○ Due to noisy information being added to the item-item graph

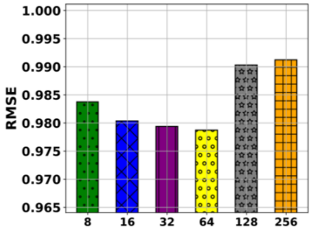


(a) Ciao-RMSE (b) Ciao-MAE (c) Epinions-RMSE (d) Epinions-MAE

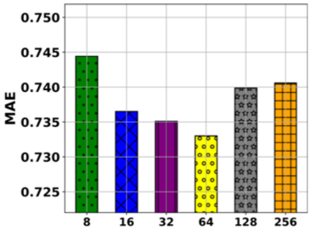
Model Analysis (Cont'd)

❖ Effect of Embedding Size on Model Performance

- Impact of different **embedding sizes** for user, item, and opinion
- Findings:
  - **Performance Improvement**:
    - Increasing embedding size from **8** to **64** leads to significant performance gains
  - **Performance Degradation**:
    - Performance starts to degrade when the embedding size is increased beyond **128**
- Reason:
  - **Smaller Embeddings**:
    - May not capture sufficient information
  - **Larger Embeddings**:
    - Introduce unnecessary complexity, leading to **overfitting** and increased computation time



(a) Ciao-RMSE



(b) Ciao-MAE

Validation on Ranking Matrics

- Importance of considering the relations among items
- Enhancing the representation learning of items for social recommendations

Performance Comparison of Different Recommender Systems on Ranking Metrics (NDCG@K)

Datasets		Ciao			Epinions			Flixster		
Metrics		NDCG@5	NDCG@10	NDCG@20	NDCG@5	NDCG@10	NDCG@20	NDCG@5	NDCG@10	NDCG@20
Algorithms	PMF	0.3044	0.3067	0.3648	0.3073	0.3627	0.3847	0.5112	0.5255	0.5321
	SoRec	0.3592	0.3690	0.3831	0.4015	0.4380	0.4568	0.5407	0.5601	0.5683
	SoReg	0.3670	0.3772	0.3951	0.4368	0.4684	0.5044	0.5504	0.5713	0.5800
	SocialMF	0.3784	0.3891	0.4119	0.4438	0.4620	0.4691	0.5565	0.5789	0.5880
	TrustMF	0.3776	0.3886	0.4103	0.4325	0.4630	0.4982	0.5662	0.5911	0.6011
	NeuMF	0.3980	0.4096	0.4342	0.4246	0.4429	0.4814	0.5344	0.5638	0.5718
	DeepSoR	0.3806	0.3938	0.4160	0.4421	0.4728	0.5046	0.5536	0.5754	0.5845
	GCMC+SN	0.3814	0.4307	0.4611	0.4475	0.4776	0.5022	0.5707	0.5928	0.6039
	GraphRec	0.3972	0.4410	0.4618	0.4580	0.4914	0.5124	0.5948	0.6245	0.6360
	GraphRec+	0.4351	0.4738	0.4966	0.5169	0.5442	0.5684	0.6385	0.6669	0.6748
Improvement* (%)		9.53	7.44	7.52	12.88	10.74	10.92	7.34	6.8	6.11

**The value indicates the percentage of improvements gained by GraphRec+ compared to the strongest baseline.*

Conclusion and Future Work

Summary

- Effectively integrates multiple graph structures for better social recommendations
- Addresses heterogeneity in relations and opinion-aware interaction modeling

Future Work

- Addressing cold-start issues
- Extending to more complex social and temporal dynamics

HTET ARKAR (hak3601@cau.ac.kr)