Data Science and Engineering Lab

IJCAI 2021 MONTREAL

Graph Neural Networks for Recommendations

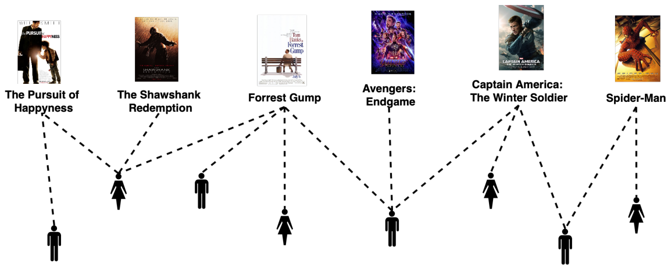
Wenqi Fan

The Hong Kong Polytechnic University

https://wenqifan03.github.io, wenqifan@polyu.edu.hk

Tutorial website: https://advanced-recommender-systems.github.io/ijcai2021-tutorial/

A General Paradigm

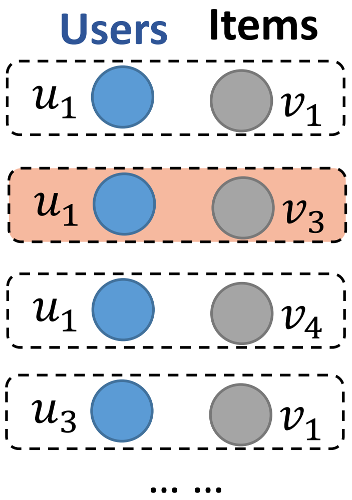


items

users

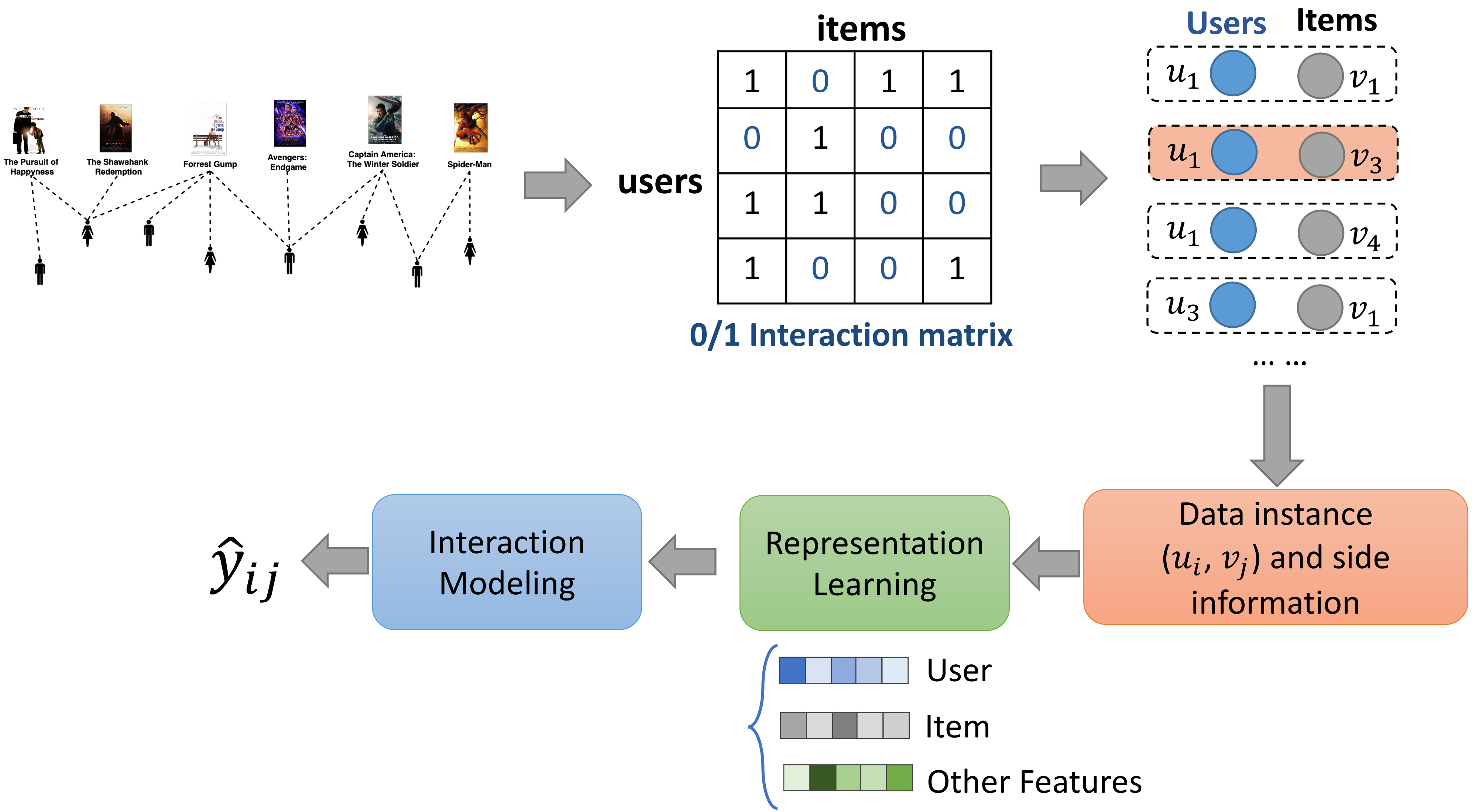
1	0	1	1
0	1	0	0
1	1	0	0
1	0	0	1

0/1 Interaction matrix

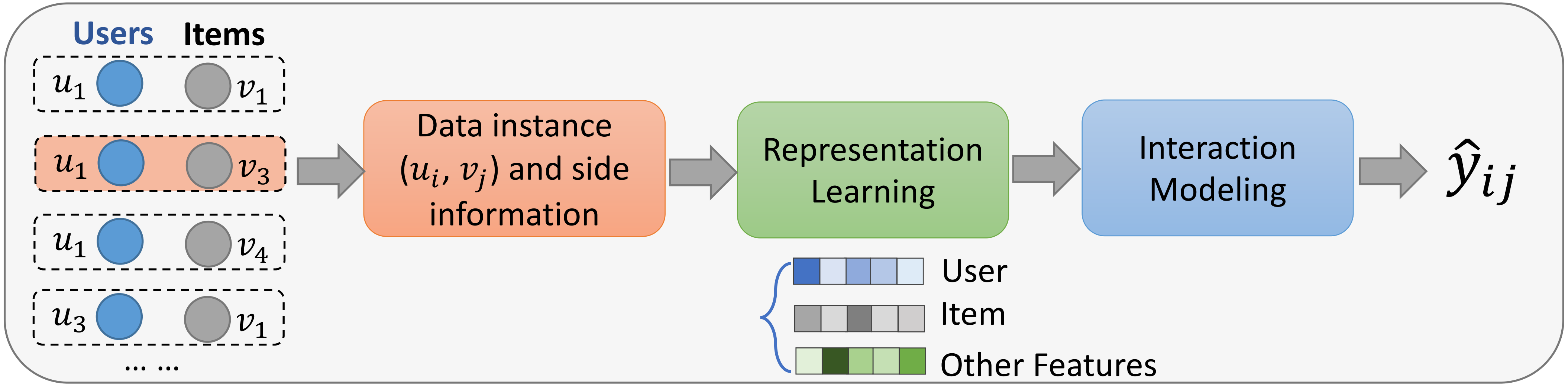


Data instance (u_i, v_j) and side information

A General Paradigm



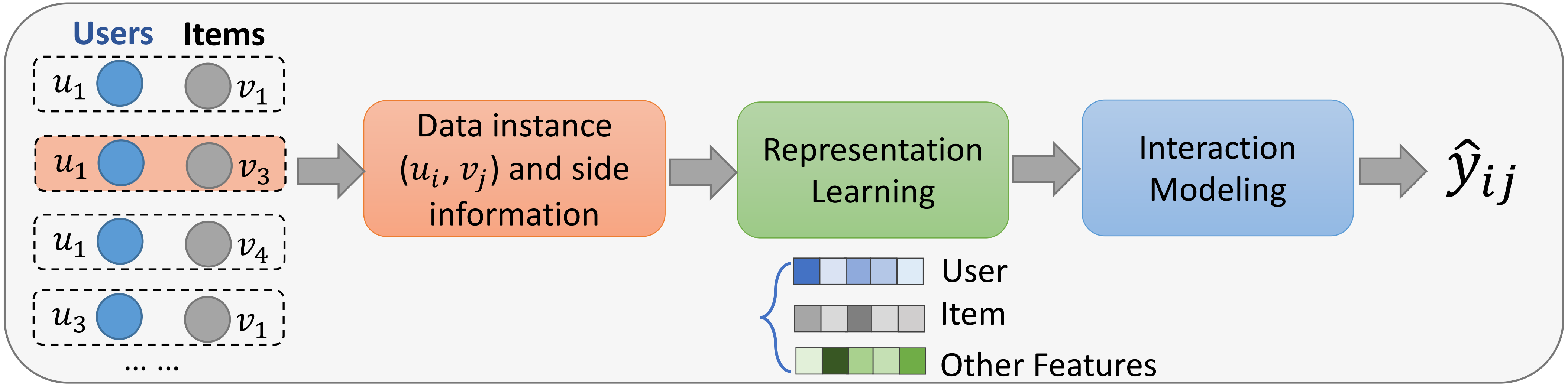
A General Paradigm



Information Isolated Island Issue

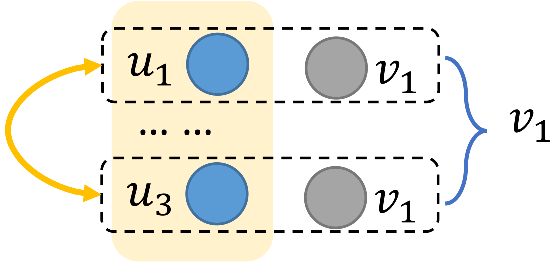
ignore implicit/explicit relationships among instances (**High-order Connectivity**)

A General Paradigm

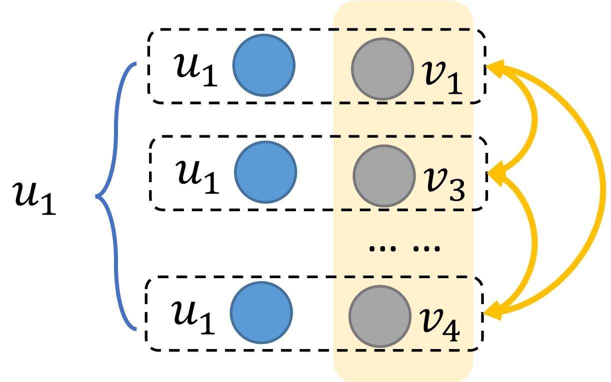


Information Isolated Island Issue

ignore implicit/explicit relationships among instances (**High-order Connectivity**)



Behavior similarity among users/items



Learning and Reasoning on Graph for Recommendation, WSDM 2020

Data as Graphs

Most of the data in RS has essentially a graph structure

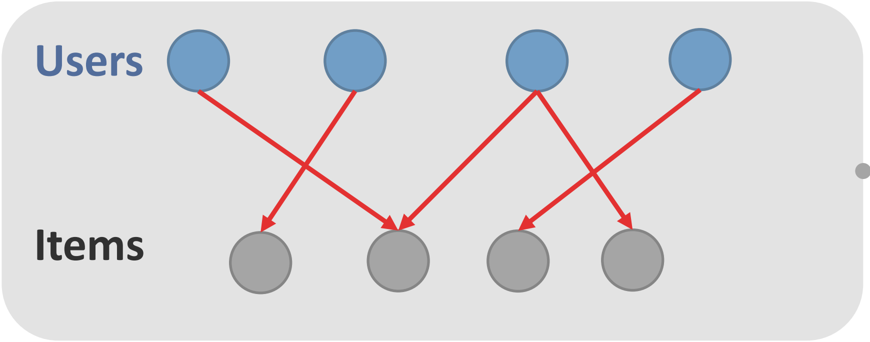
- E-commerce, Content Sharing, Social Networking ...

The world is more closely connected than you might think!

Data as Graphs

Most of the data in RS has essentially a graph structure
- E-commerce, Content Sharing, Social Networking ...

The world is more closely connected than you might think!



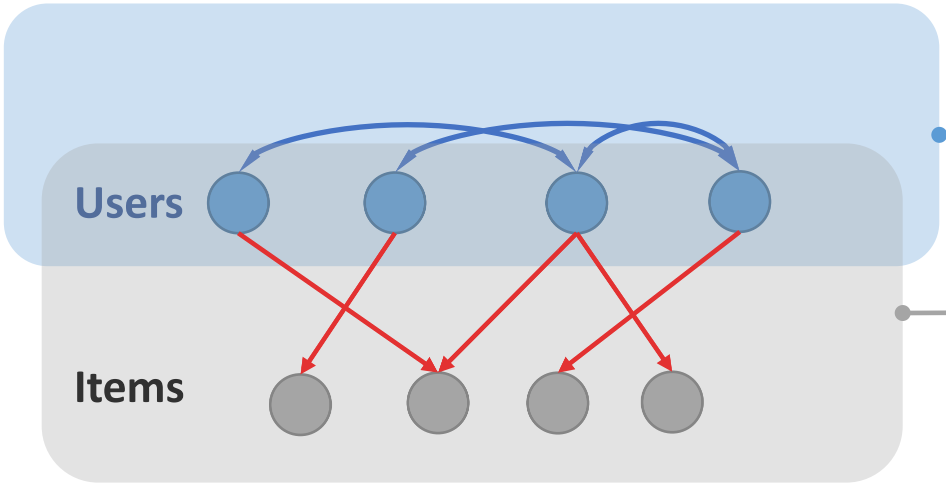
User-Item Interactions
(purchase, click, add-to-cart, etc.)

user-item graph

Data as Graphs

Most of the data in RS has essentially a graph structure
- E-commerce, Content Sharing, Social Networking ...

The world is more closely connected than you might think!



User-user Connections
(social networks)

user-user social graph

User-Item Interactions
(purchase, click, add-to-cart, etc.)

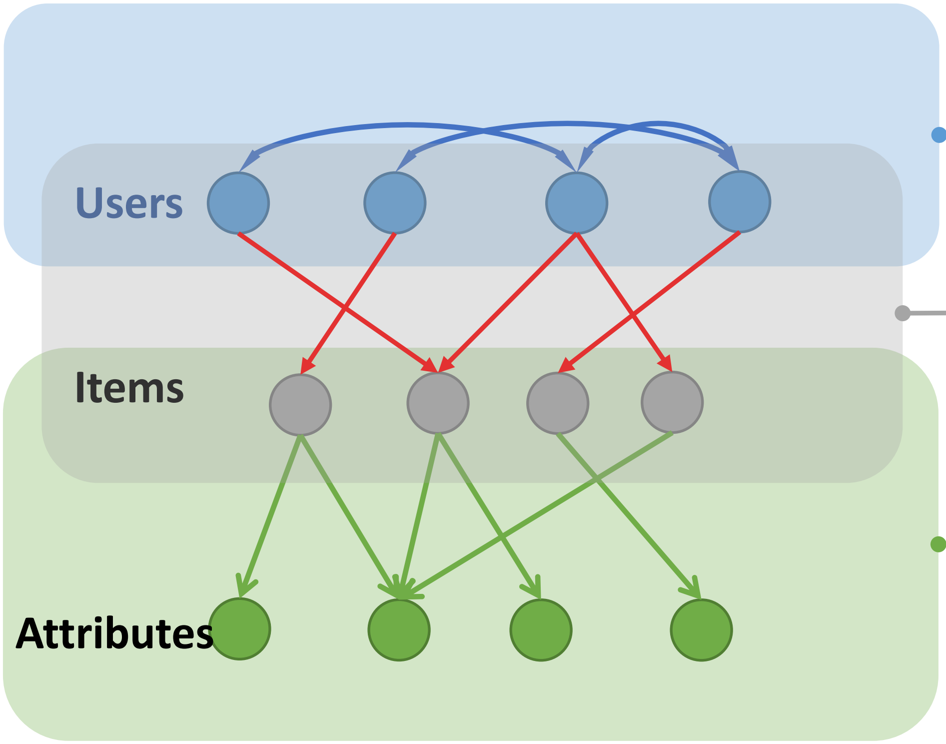
user-item graph

Data as Graphs

Most of the data in RS has essentially a graph structure

- E-commerce, Content Sharing, Social Networking ...

The world is more closely connected than you might think!

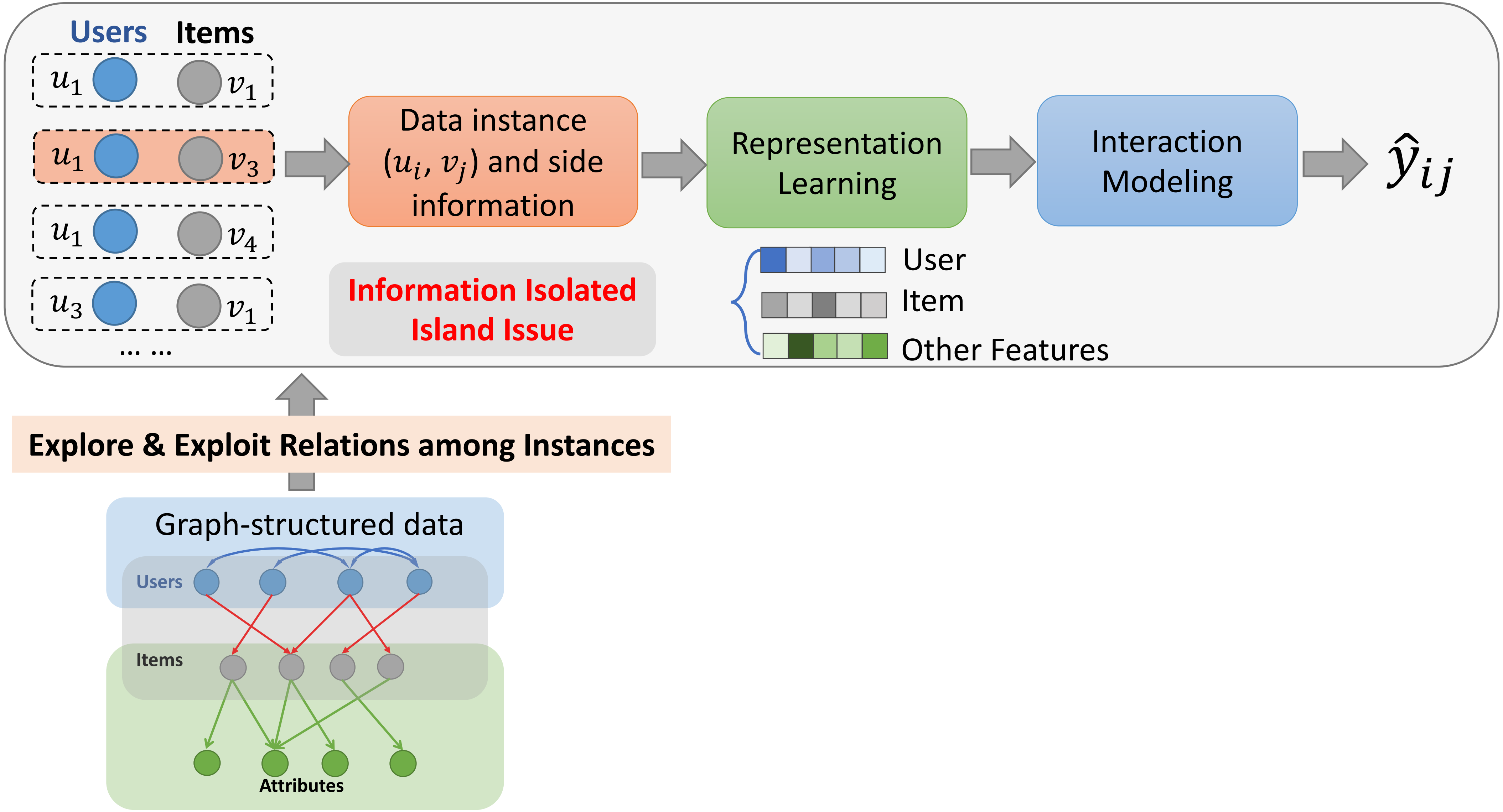


User-user Connections (social networks) — **user-user social graph**

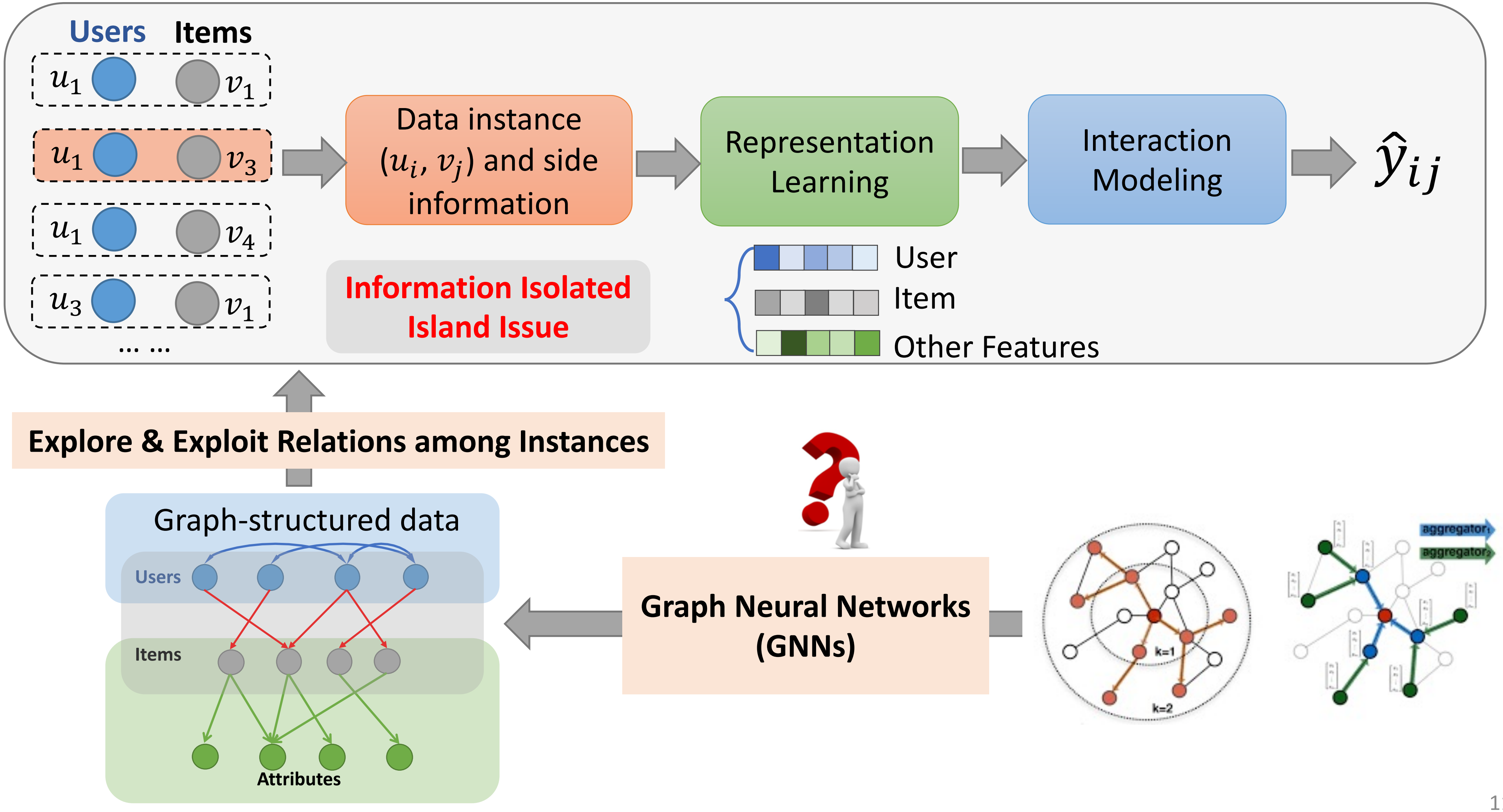
User-Item Interactions (purchase, click, add-to-cart, etc.) — **user-item graph**

Item-item Connections (substitutes/complements; external knowledge) — **item graph**

How to solve such issue?

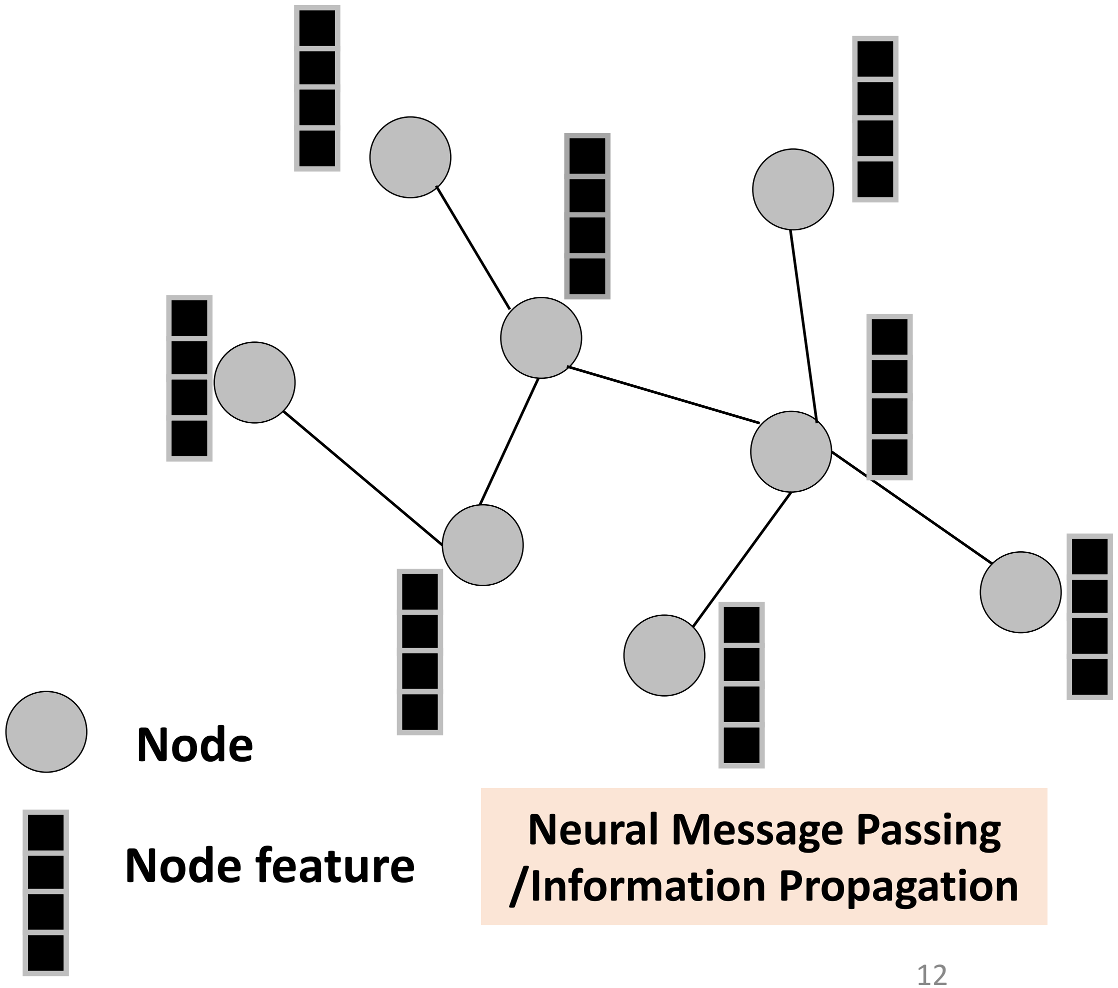


How to solve such issue?



Graph Neural Networks (GNNs)

Key idea: Generate node embeddings via using neural networks to aggregate information from local neighborhoods.

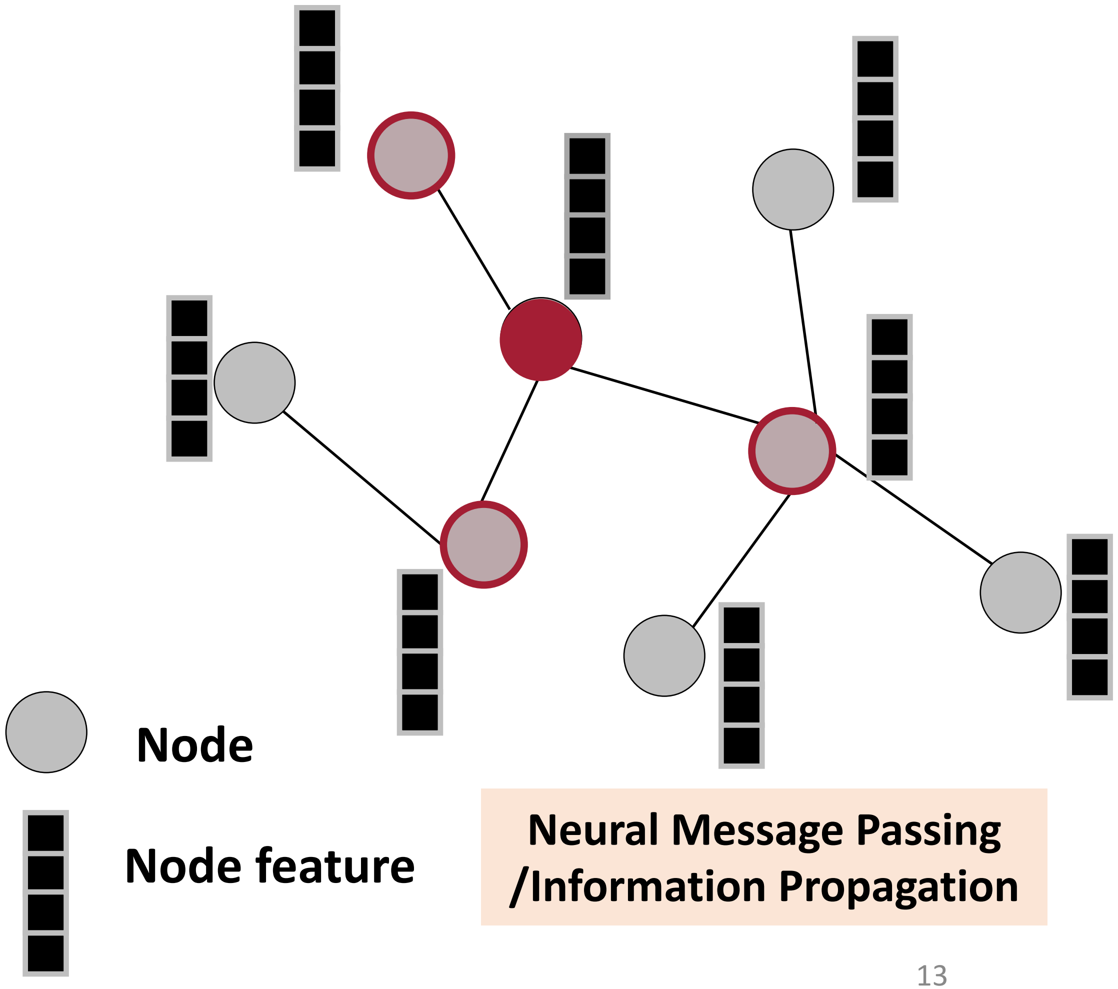


Inductive Representation Learning on Large Graphs, NeuIPS, 2017.

Graph Neural Networks (GNNs)

Key idea: Generate node embeddings via using neural networks to aggregate information from local neighborhoods.

1. Model a local structural information (neighborhood) of a node;



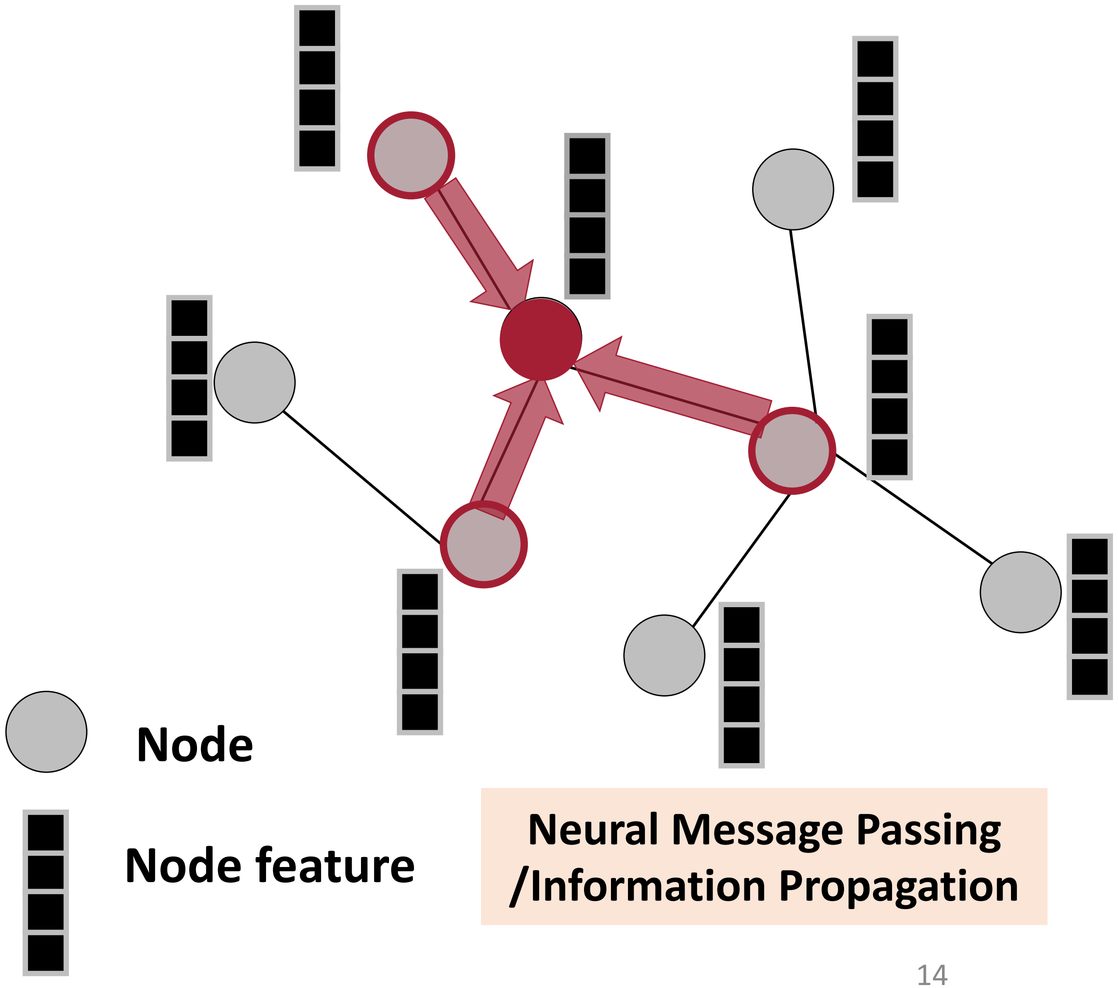
Inductive Representation Learning on Large Graphs, NeuIPS, 2017.

Graph Neural Networks (GNNs)

Key idea: Generate node embeddings via using neural networks to aggregate information from local neighborhoods.

1. Model a local structural information (neighborhood) of a node;
2. Aggregation operation;
3. Representation update.

GNNs can naturally integrate node feature and the topological structure for graph-structured data.



Neural Message Passing /Information Propagation

Inductive Representation Learning on Large Graphs, NeuIPS, 2017.

Graph Neural Networks (GNNs)

Basic approach: Average neighbor messages and apply a neural network.

$$\mathbf{h}_v^0 = \mathbf{x}_v$$

Initial 0-th layer embeddings are equal to node v's features

$$\mathbf{h}_v^k = \sigma\left(\mathbf{W}_1^k \sum_{u \in N(v)} \frac{\mathbf{h}_u^{k-1}}{\sqrt{|N(u)|}} + \mathbf{W}_2^k \mathbf{h}_v^{k-1}\right)$$

k-th layer embedding of node v

$$\mathbf{z}_v = \mathbf{h}_v^k$$

Embedding after k layers of neighborhood aggregation.

Semi-supervised Classification with Graph Convolutional Network, ICLR, 2017.

Graph Neural Networks (GNNs)

Basic approach: Average neighbor messages and apply a neural network.

$$\mathbf{h}_v^0 = \mathbf{x}_v$$

Initial 0-th layer embeddings are equal to node v's features

Non-linearity (e.g., ReLU or tanh)

trainable matrices (i.e., what we learn)

Previous layer embedding of node v

$$\mathbf{h}_v^k = \sigma\left(\mathbf{W}_1^k \sum_{u \in N(v)} \frac{\mathbf{h}_u^{k-1}}{\sqrt{|N(u)|}} + \mathbf{W}_2^k \mathbf{h}_v^{k-1}\right)$$

k-th layer embedding of node v

Average of neighbor's previous layer embeddings

$$\mathbf{z}_v = \mathbf{h}_v^k$$

Embedding after k layers of neighborhood aggregation.

Semi-supervised Classification with Graph Convolutional Network, ICLR, 2017.

Graph Neural Network (GNN)

- **Simple neighborhood aggregation:**

$$\mathbf{h}_v^k = \sigma\left(\mathbf{W}_1^k \sum_{u\in N(v)} \frac{\mathbf{h}_u^{k-1}}{\sqrt{|N(u)|}} + \mathbf{W}_2^k \mathbf{h}_v^{k-1}\right)$$

- **GraphSAGE:**

- **GAT:**

Inductive Representation Learning on Large Graphs, NeuIPS, 2017.
Graph Attention Networks, ICLR, 2018

Graph Neural Network (GNN)

- **Simple neighborhood aggregation:**

$$\mathbf{h}_v^k = \sigma\left(\mathbf{W}_1^k \sum_{u \in N(v)} \frac{\mathbf{h}_u^{k-1}}{\sqrt{|N(u)|}} + \mathbf{W}_2^k \mathbf{h}_v^{k-1}\right)$$

- **GraphSAGE:**

$$\mathbf{h}_v^k = \sigma\left([\mathbf{W}_1^k \cdot \mathrm{AGG}\left(\{\mathbf{h}_u^{k-1}, \forall_u \in N(u)\}\right), \mathbf{W}_2^k \cdot \mathbf{h}_v^k]\right)$$

Generalized Aggregation: mean, pooling, LSTM

- **GAT:**

Inductive Representation Learning on Large Graphs, NeuIPS, 2017.
Graph Attention Networks, ICLR, 2018

Graph Neural Network (GNN)

- **Simple neighborhood aggregation:**

$$\mathbf{h}_v^k = \sigma\left(\mathbf{W}_1^k \sum_{u \in N(v)} \frac{\mathbf{h}_u^{k-1}}{\sqrt{|N(u)|}} + \mathbf{W}_2^k \mathbf{h}_v^{k-1}\right)$$

- **GraphSAGE:**

$$\mathbf{h}_v^k = \sigma\left(\left[\mathbf{W}_1^k \cdot \mathrm{AGG}\left(\{\mathbf{h}_u^{k-1}, \forall_u \in N(u)\}\right), \mathbf{W}_2^k \cdot \mathbf{h}_v^k\right]\right)$$

Generalized Aggregation: mean, pooling, LSTM

- **GAT:**

$$\mathbf{h}_v^k = \sigma\left(\sum_{u \in N(v)} \alpha_{v,u} \mathbf{W}^k \, \mathbf{h}_u^{k-1}\right)$$

Learned attention weights

Inductive Representation Learning on Large Graphs, NeuIPS, 2017.
Graph Attention Networks, ICLR, 2018

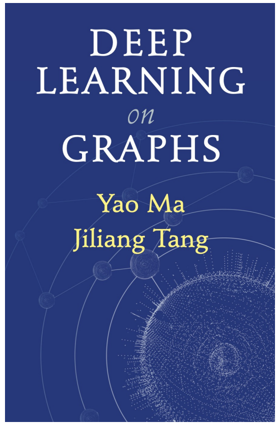
Book: Deep Learning on Graphs

Authors

English Version: Yao Ma and Jiliang Tang

Chinese Version: Yiqi Wang, Wei Jin, Yao Ma and Jiliang Tang

https://cse.msu.edu/~mayao4/dlg_book/



1. Introduction

Part One: Foundations
- 2. Foundations of Graphs
- 3. Foundations of Deep Learning

Part Two: Methods
- 4. Graph Embedding
- 5. Graph Neural Networks
- 6. Robust Graph Neural Networks
- 7. Scalable Graph Neural Networks
- 8. Graph Neural Networks for Complex Graphs
- 9. Beyond GNNs: More Deep Models for Graphs

Part Three: Applications
- 10. Graph Neural Networks in Natural Language Processing
- 11. Graph Neural Networks in Computer Vision
- 12. Graph Neural Networks in Data Mining
- 13. Graph Neural Networks in Bio-Chemistry and Healthcare

Part Four: Advances
- 14. Advanced Methods in Graph Neural Networks
- 15. Advanced Applications in Graph Neural Networks

GNNs based Recommendation

Collaborative Filtering

- Graph Convolutional Neural Networks for Web-Scale Recommender Systems (KDD’18)
- Graph Convolutional Matrix Completion (KDD’18 Deep Learning Day)
- Neural Graph Collaborative Filtering (SIGIR’19)
- LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation (SIGIR’20)
- Graph Trend Networks for Recommendations, arXiv:2108.05552, 2021

Collaborative Filtering with Side Information (Users/Items)

Social Recommendation (Users)

- Graph Neural Network for Social Recommendation (WWW’19)
- A Neural Influence Diffusion Model for Social Recommendation (SIGIR’19)
- A Graph Neural Network Framework for Social Recommendations (TKDE’20)

Knowledge-graph-aware Recommendation (Items)

- Knowledge Graph Convolutional Networks for Recommender Systems with Label Smoothness Regularization (KDD’19 and WWW’19)
- KGAT: Knowledge Graph Attention Network for Recommendation (KDD’19)

GNNs based Recommendation

Collaborative Filtering

- Graph Convolutional Neural Networks for Web-Scale Recommender Systems (KDD’18)
- Graph Convolutional Matrix Completion (KDD’18 Deep Learning Day)
- Neural Graph Collaborative Filtering (SIGIR’19)
- LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation (SIGIR’20)
- Graph Trend Networks for Recommendations, arXiv:2108.05552, 2021

Collaborative Filtering with Side Information (Users/Items)

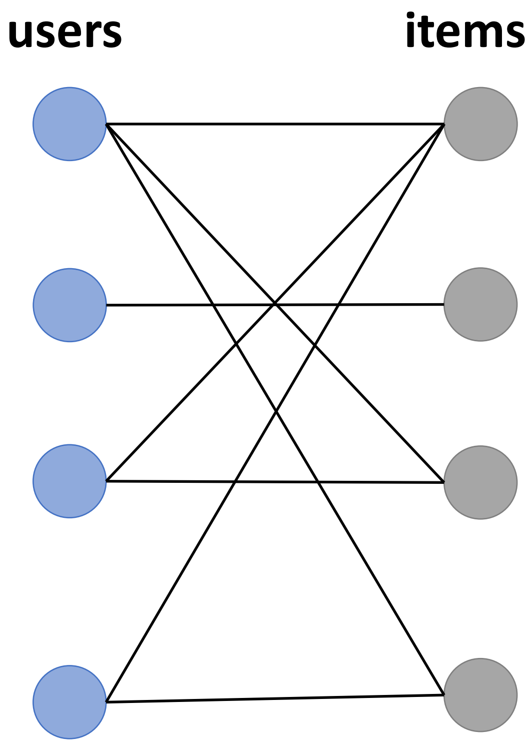
- **Social Recommendation (Users)**
  - Graph Neural Network for Social Recommendation (WWW’19)
  - A Neural Influence Diffusion Model for Social Recommendation (SIGIR’19)
  - A Graph Neural Network Framework for Social Recommendations (TKDE’20)
- **Knowledge-graph-aware Recommendation (Items)**
  - Knowledge Graph Convolutional Networks for Recommender Systems with Label Smoothness Regularization (KDD’19 and WWW’19)
  - KGAT: Knowledge Graph Attention Network for Recommendation (KDD’19)

Interactions as Bipartite Graph

items

users	1	0	1	1
	0	1	0	0
	1	1	0	0
	1	0	0	1

0/1 Interaction matrix



Bipartite Graph

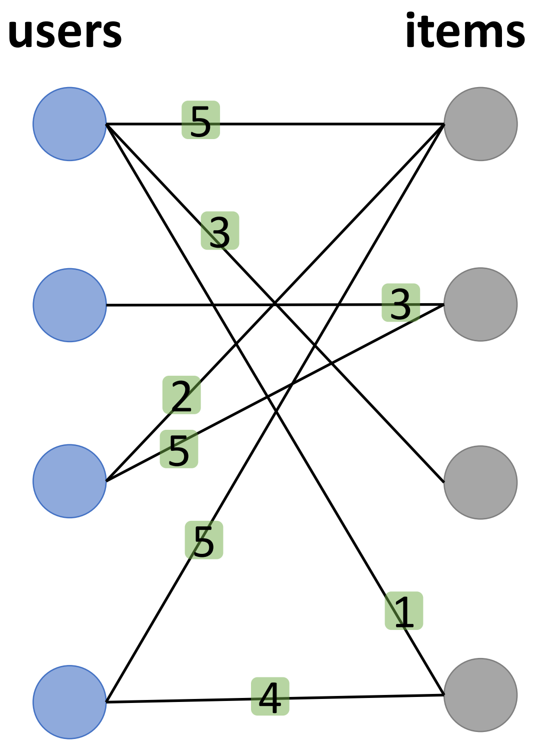
Interactions as Bipartite Graph

items

users

5	0	3	1
0	3	0	0
2	5	0	0
5	0	0	4

Weighted interaction matrix

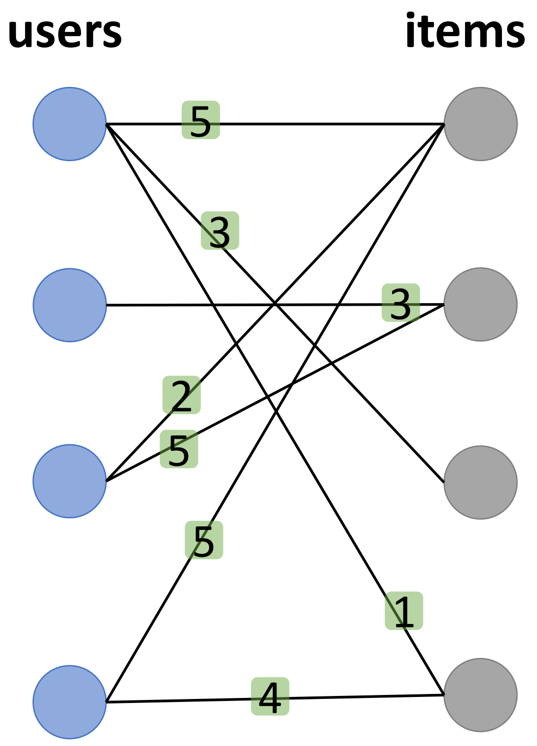


Bipartite Graph

GCMC

User representation learning

Aggregate for each rating: $\mu_{i,r} = \sum_{j \in \mathcal{N}_{i,r}} \frac{1}{c_{ij}} \boxed{W_r} x_j$



Bipartite Graph

Graph Convolutional Matrix Completion. 2017.

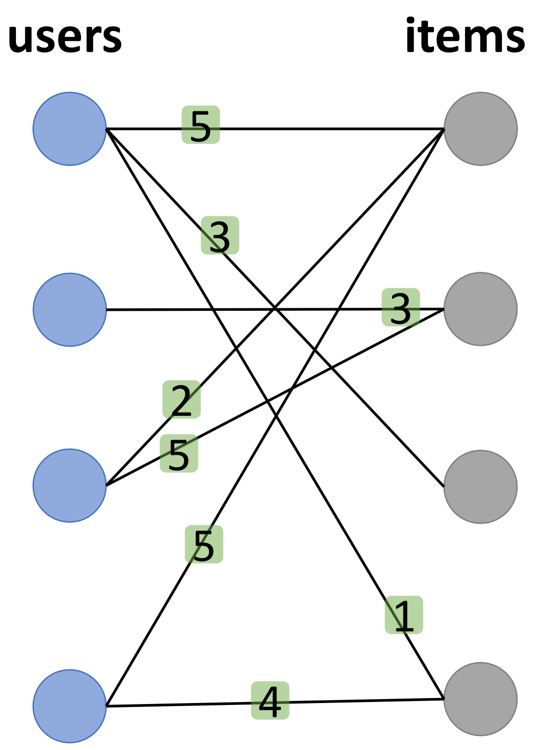
GCMC

User representation learning

Aggregate for each rating: $\mu_{i,r} = \sum_{j \in \mathcal{N}_{i,r}} \frac{1}{c_{ij}} \boxed{W_r} x_j$

$$u_i = \mathbf{W} \cdot \sigma(accum(u_{i,1}, \ldots, u_{i,R}))$$

Item representation learning in a similar way

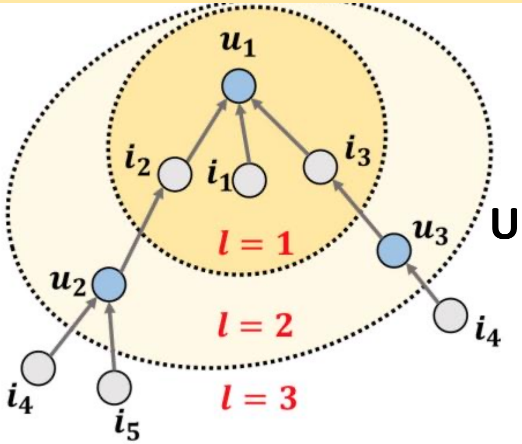


Bipartite Graph

Graph Convolutional Matrix Completion. 2017.

NGCF

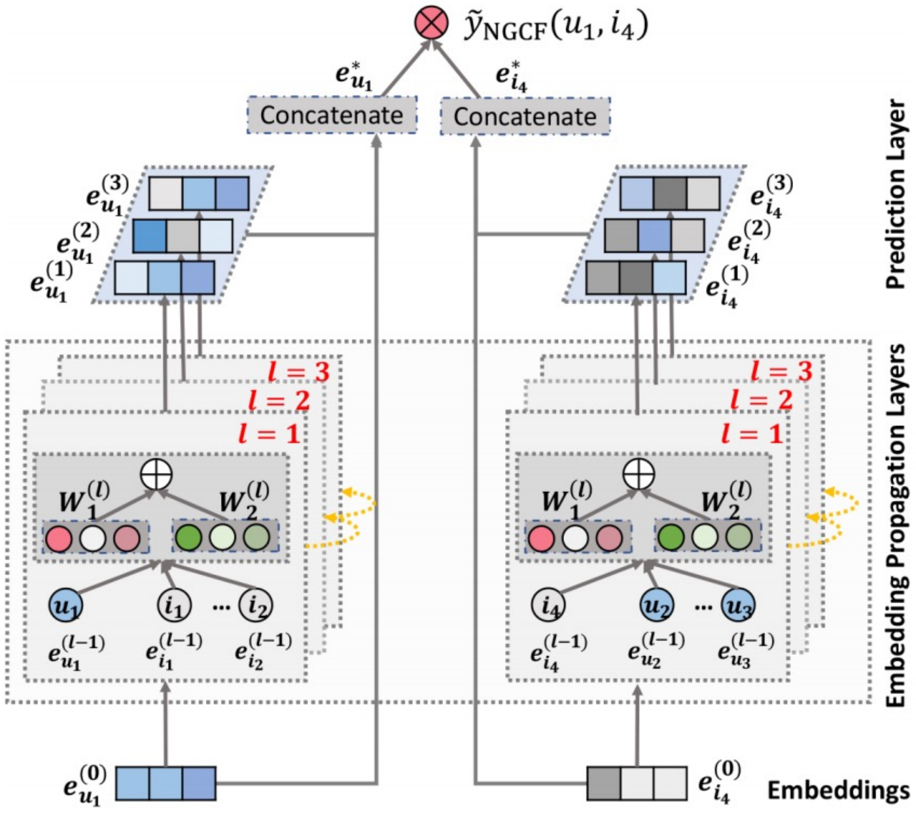
High-order Connectivity for u_1



User-item Graph

Embedding Propagation, inspired by GNNs

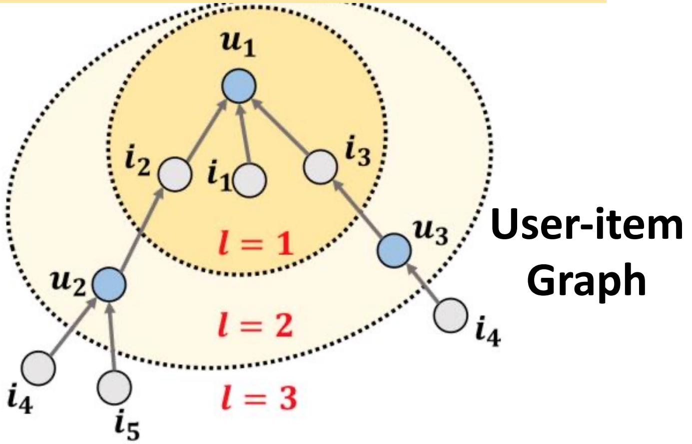
- Propagate embeddings recursively on the user-item graph
- Construct information flows in the embedding space



Neural Graph Collaborative Filtering. SIGIR 2019.

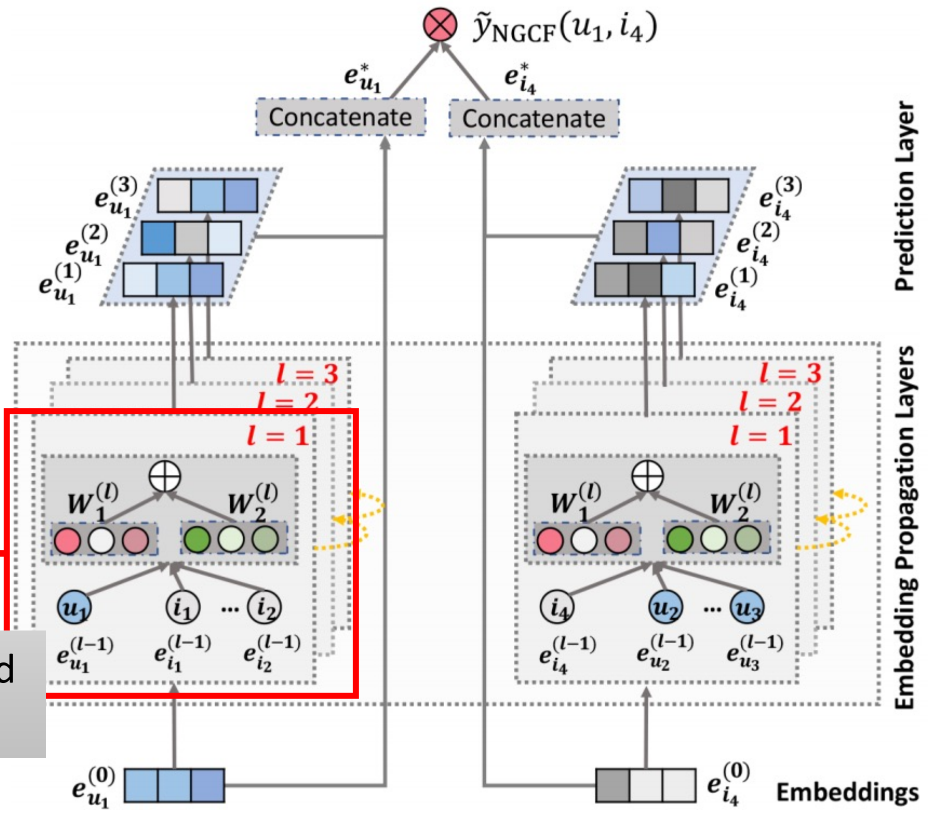
NGCF

High-order Connectivity for u_1



Embedding Propagation, inspired by GNNs

- Propagate embeddings recursively on the user-item graph
- Construct information flows in the embedding space



$$\mathbf{e}_u^{(l)} = \mathrm{LeakyReLU}\Big(\mathbf{m}_{u\leftarrow u}^{(l)} + \sum_{i\in\mathcal{N}_u} \mathbf{m}_{u\leftarrow i}^{(l)}\Big),$$

$$\begin{cases} \mathbf{m}_{u\leftarrow i}^{(l)} = p_{ui}\Big(\mathbf{W}_1^{(l)}\mathbf{e}_i^{(l-1)} + \mathbf{W}_2^{(l)}(\mathbf{e}_i^{(l-1)} \odot \mathbf{e}_u^{(l-1)})\Big) \\ \mathbf{m}_{u\leftarrow u}^{(l)} = \mathbf{W}_1^{(l)}\mathbf{e}_u^{(l-1)} \end{cases}$$

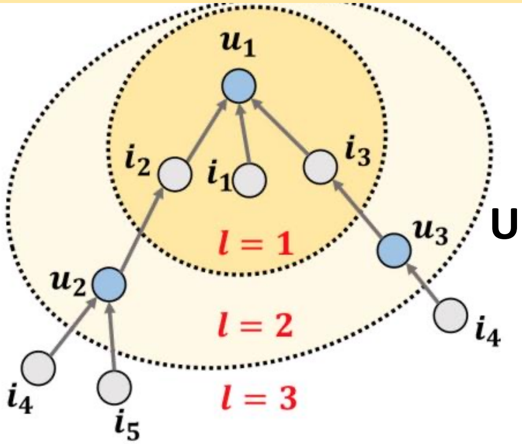
collaborative signal: message passed from interacted items to u

Self-connections

Neural Graph Collaborative Filtering. SIGIR 2019.

NGCF

High-order Connectivity for u_1



User-item Graph

Embedding Propagation, inspired by GNNs

- Propagate embeddings recursively on the user-item graph
- Construct information flows in the embedding space

Different layers

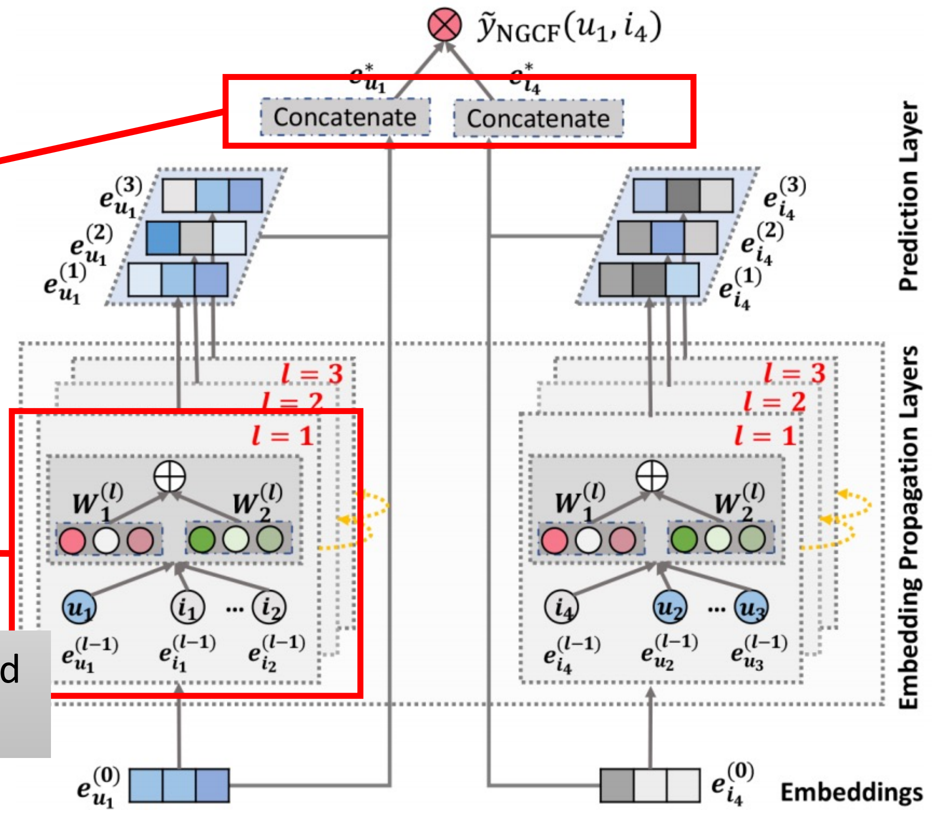
$$\mathbf{e}_u^* = \mathbf{e}_u^{(0)} \| \cdots \| \mathbf{e}_u^{(L)}, \quad \mathbf{e}_i^* = \mathbf{e}_i^{(0)} \| \cdots \| \mathbf{e}_i^{(L)},$$

$$\mathbf{e}_u^{(l)} = \text{LeakyReLU}\Big(\mathbf{m}_{u \leftarrow u}^{(l)} + \sum_{i \in \mathcal{N}_u} \mathbf{m}_{u \leftarrow i}^{(l)}\Big),$$

$$\begin{cases} \mathbf{m}_{u \leftarrow i}^{(l)} = p_{ui}\Big(\mathbf{W}_1^{(l)} \mathbf{e}_i^{(l-1)} + \mathbf{W}_2^{(l)} (\mathbf{e}_i^{(l-1)} \odot \mathbf{e}_u^{(l-1)})\Big) \\ \mathbf{m}_{u \leftarrow u}^{(l)} = \mathbf{W}_1^{(l)} \mathbf{e}_u^{(l-1)} \end{cases}$$

collaborative signal: message passed from interacted items to u

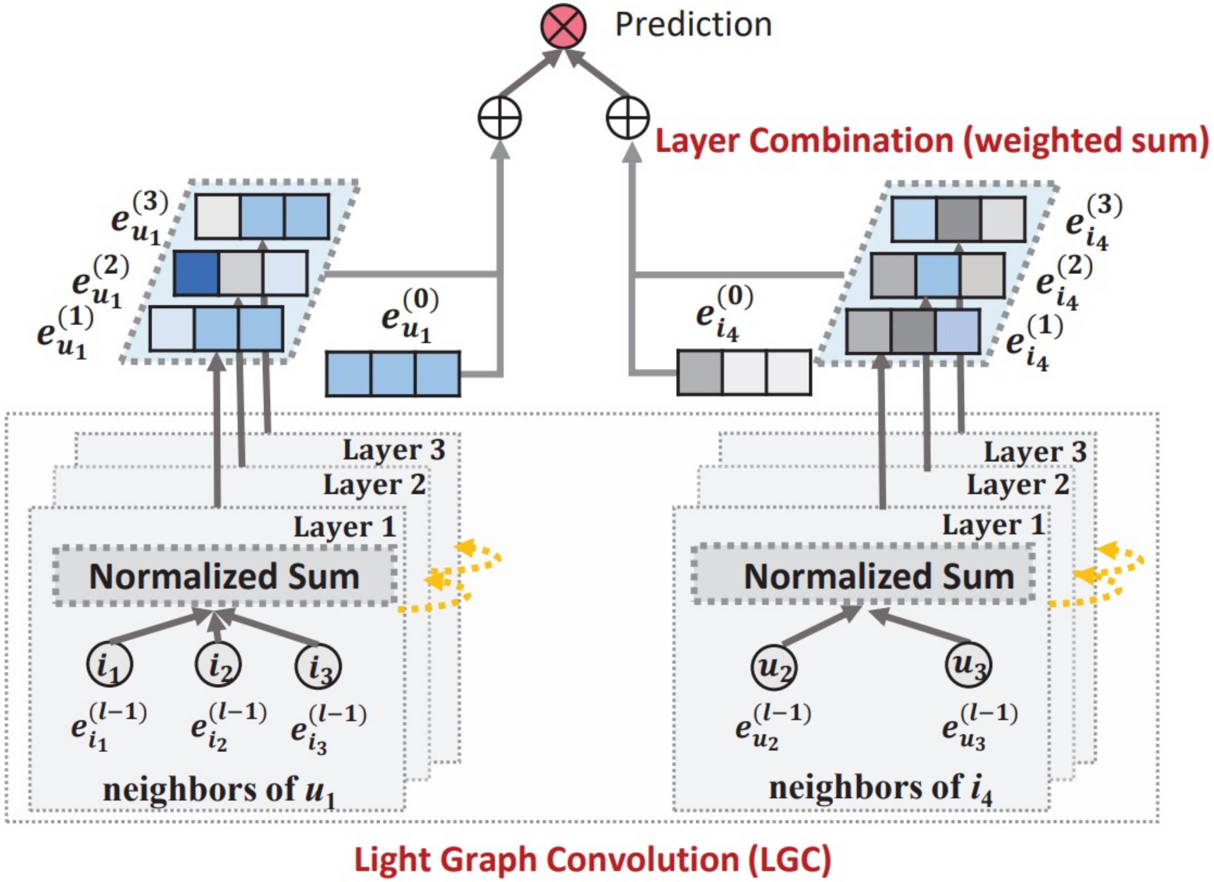
Self-connections



Neural Graph Collaborative Filtering. SIGIR 2019.

LightGCN

Simplifying GCN for recommendation



discard feature transformation and nonlinear activation

LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation. SIGIR 2020.

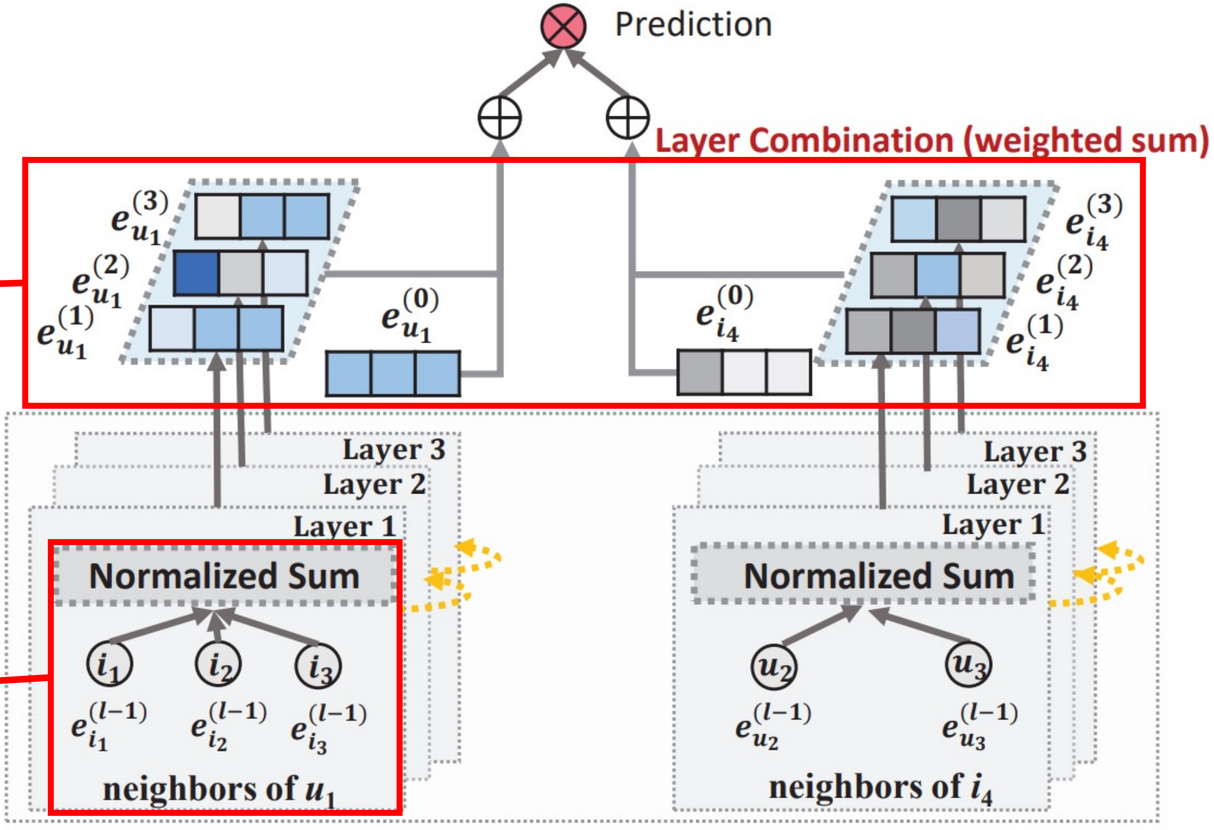
LightGCN

Simplifying GCN for recommendation

$$\mathbf{e}_u = \sum_{k=0}^{K} \alpha_k \mathbf{e}_u^{(k)}; \quad \mathbf{e}_i = \sum_{k=0}^{K} \alpha_k \mathbf{e}_i^{(k)},$$

$$\mathbf{e}_u^{(k+1)} = \sum_{i \in \mathcal{N}_u} \frac{1}{\sqrt{|\mathcal{N}_u|}\sqrt{|\mathcal{N}_i|}} \mathbf{e}_i^{(k)},$$

$$\mathbf{e}_i^{(k+1)} = \sum_{u \in \mathcal{N}_i} \frac{1}{\sqrt{|\mathcal{N}_i|}\sqrt{|\mathcal{N}_u|}} \mathbf{e}_u^{(k)}.$$



discard feature transformation and nonlinear activation

LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation. SIGIR 2020.

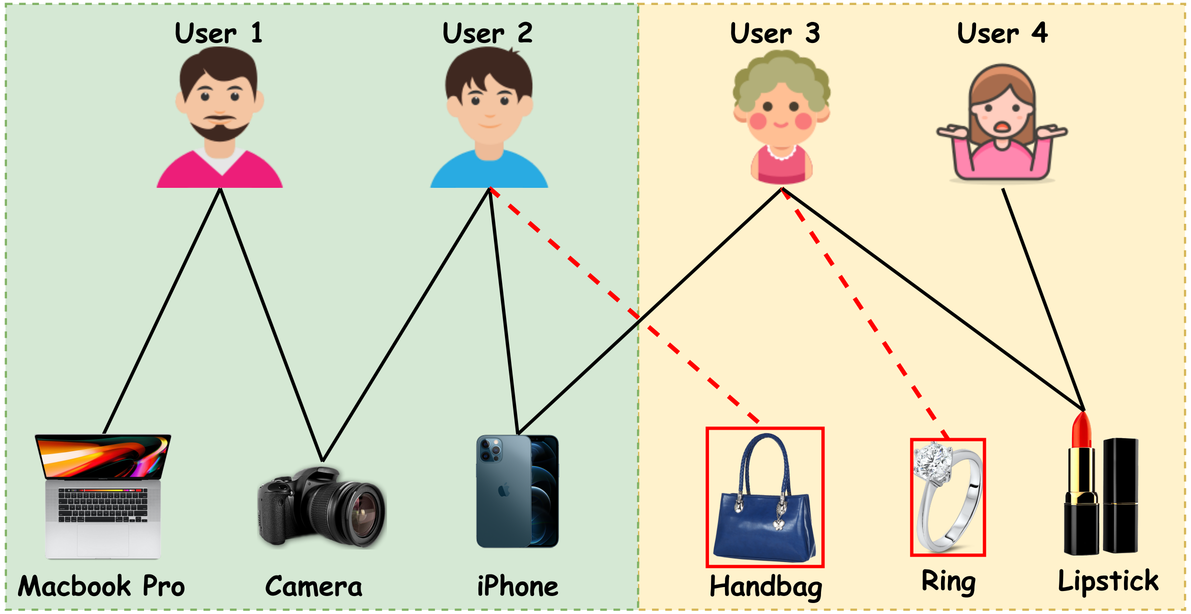
Graph Trend Networks for Recommendations

- Unreliable user-item interactions

Embedding Propagation Rule

$$\mathbf{e}_u^{k+1} = \frac{1}{\sqrt{|\mathcal{N}(u)|}} \sum_{i \in \mathcal{N}(u)} \frac{1}{\sqrt{|\mathcal{N}(i)|}} \mathbf{e}_i^k;$$

$$\mathbf{e}_i^{k+1} = \frac{1}{\sqrt{|\mathcal{N}(i)|}} \sum_{u \in \mathcal{N}(i)} \frac{1}{\sqrt{|\mathcal{N}(u)|}} \mathbf{e}_u^k$$



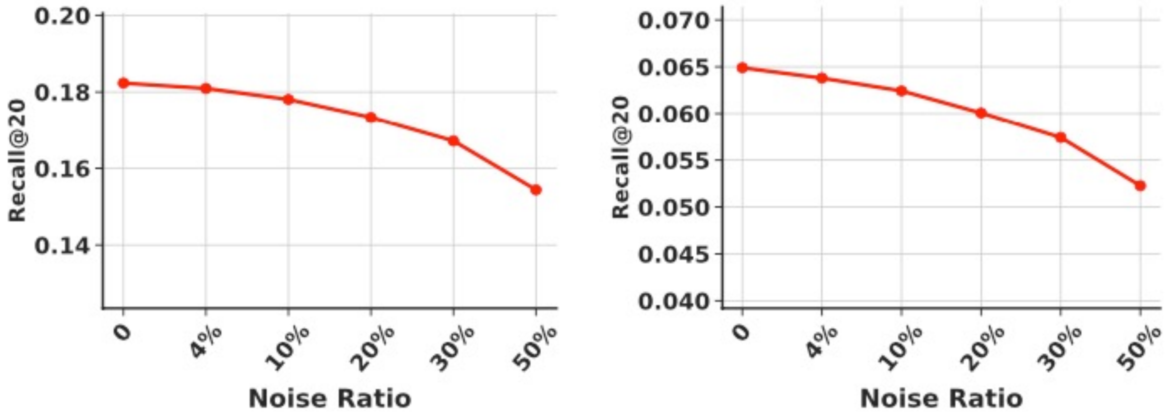
Overlook unreliable interactions (e.g., random/bait clicks) and uniformly treat all the interactions

E.g.,
(1) User 3 was affected by the click-bait issue.
(2) User 2 bought a one-time item for his mother's birthday present;

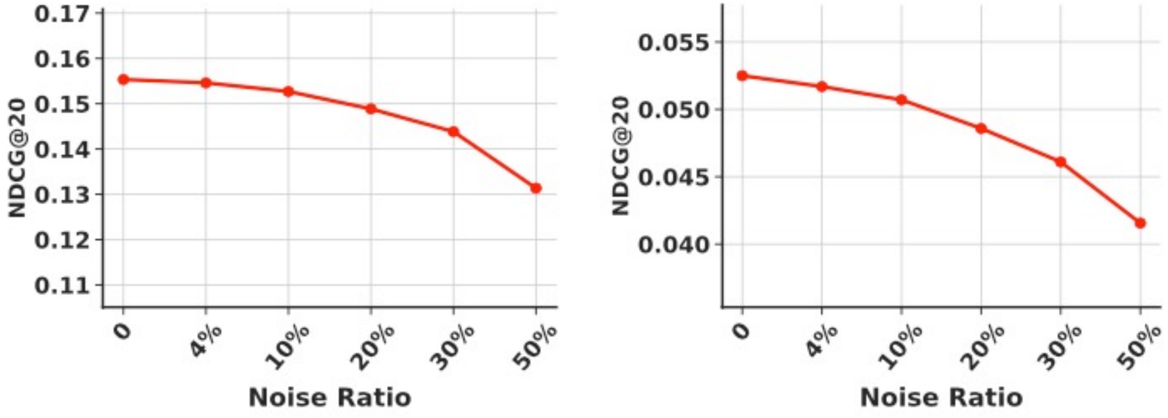
Graph Trend Networks for Recommendations. arXiv:2108.05552, 2021.

Preliminary study

- Performance of LightGCN under different perturbation rates.



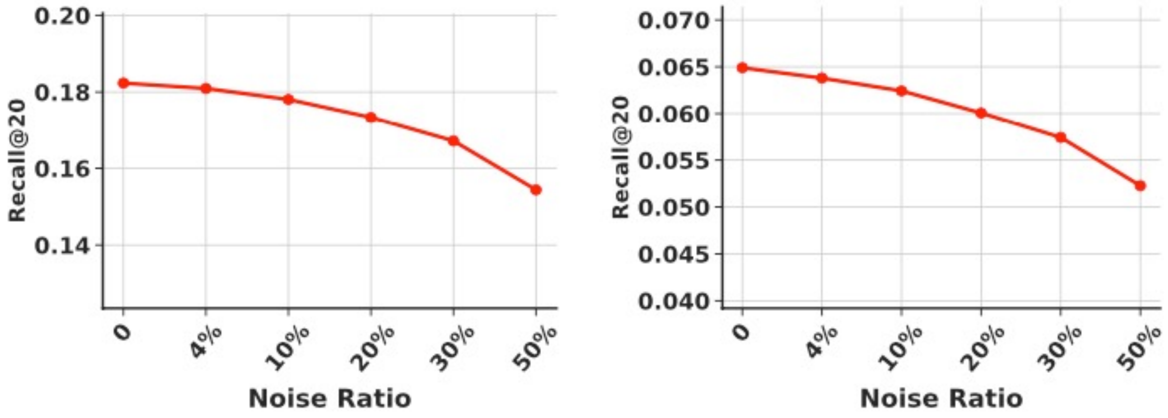
(a) Gowalla - Recall@20 (b) Yelp2018 - Recall@20



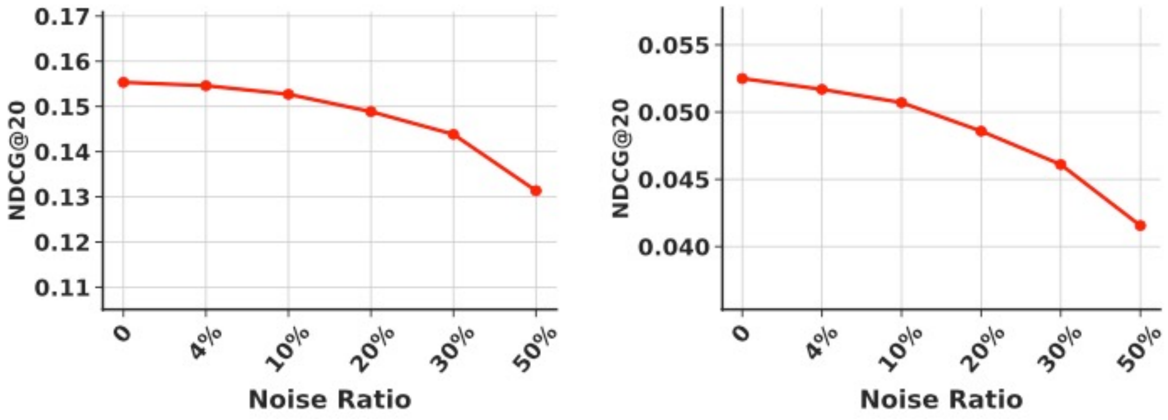
Graph Trend Networks for Recommendations. arXiv:2108.05552, 2021.

Preliminary study

- Performance of LightGCN under different perturbation rates.



(a) Gowalla - Recall@20 (b) Yelp2018 - Recall@20



- ➢ To build a more reliable and robust recommender system
- ➢ Graph Trend Networks for recommendations (GTN)

Graph Trend Networks for Recommendations. arXiv:2108.05552, 2021.

Graph Trend Networks for Recommendations

$$\mathbf{e}_u^{k+1} = \frac{1}{\sqrt{|\mathcal{N}(u)|}} \sum_{i \in \mathcal{N}(u)} \frac{1}{\sqrt{|\mathcal{N}(i)|}} \mathbf{e}_i^k$$

$$\mathbf{e}_i^{k+1} = \frac{1}{\sqrt{|\mathcal{N}(i)|}} \sum_{u \in \mathcal{N}(i)} \frac{1}{\sqrt{|\mathcal{N}(u)|}} \mathbf{e}_u^k$$

Matrix form: $\mathbf{E}^{K+1} = \tilde{\mathbf{A}}\mathbf{E}^k$

- Laplacian smoothing problem

$$\arg\min_{\mathbf{E} \in \mathbb{R}^{(n+m)\times d}} \operatorname{tr}(\mathbf{E}^\top (\mathbf{I} - \tilde{\mathbf{A}})\mathbf{E})$$

$$\operatorname{tr}(\mathbf{E}^\top (\mathbf{I} - \tilde{\mathbf{A}})\mathbf{E}) = \sum_{(i,j)\in\mathcal{E}} \left\| \frac{\mathbf{e}_i}{\sqrt{d_i + 1}} - \frac{\mathbf{e}_j}{\sqrt{d_j + 1}} \right\|_2^2$$

edge-wise form

Graph Trend Networks for Recommendations. arXiv:2108.05552, 2021.
A unified view on graph neural networks as graph signal denoising, arXiv:2010.01777, 2020.

Graph Trend Networks for Recommendations

- Design Motivation from Graph Trend Filtering
- Embedding smoothness objective:

$$\arg\min_{\mathbf{E}\in\mathbb{R}^{(n+m)\times d}} \frac{1}{2}\|\mathbf{E}-\mathbf{E}_{\text{in}}\|_F^2 + \lambda\|\tilde{\Delta}\mathbf{E}\|_1$$

Preserve the proximity

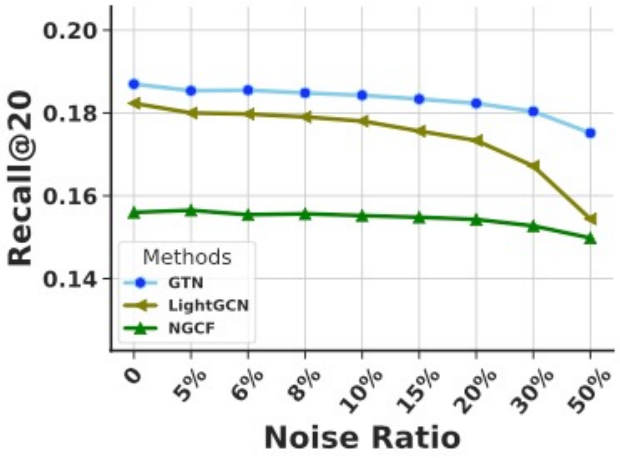
Impose embedding smoothness

$$\|\tilde{\Delta}\mathbf{E}\|_1 = \sum_{(i,j)\in\mathcal{E}} \left\|\frac{\mathbf{e}_i}{\sqrt{d_i+1}} - \frac{\mathbf{e}_j}{\sqrt{d_j+1}}\right\|_1.$$

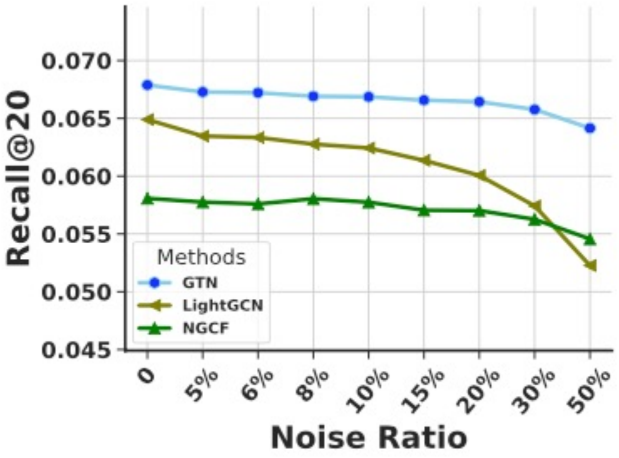
Trend filtering on graphs, *Journal of Machine Learning Research*, 2016.
Graph Trend Networks for Recommendations. arXiv:2108.05552, 2021.

GTN

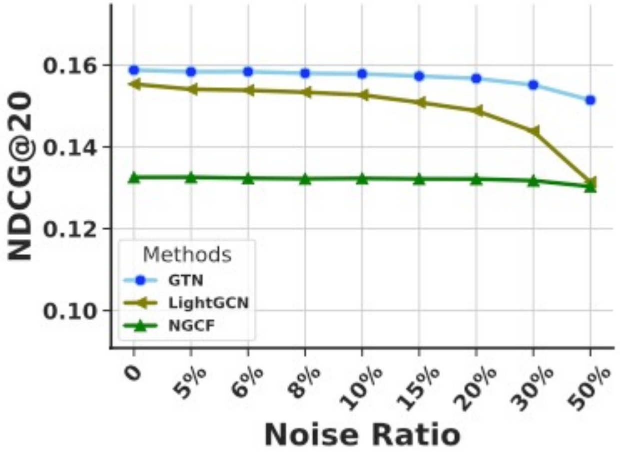
Recommendation performance under different perturbation rates.



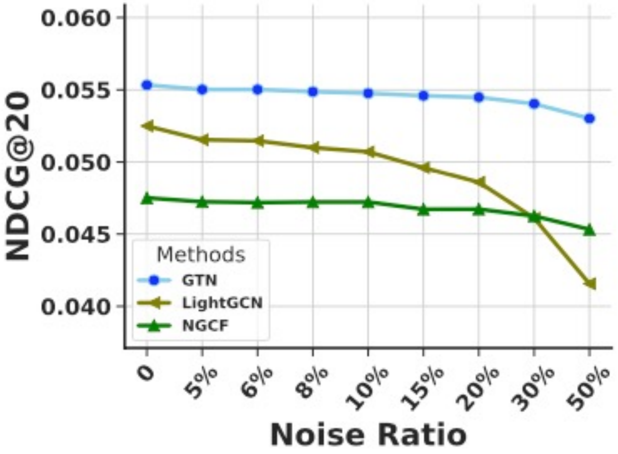
(a) Gowalla - Recall@20



(b) Yelp2018 - Recall@20



(e) Gowalla - NDCG@20



(f) Yelp2018 - NDCG@20

Graph Trend Networks for Recommendations. arXiv:2108.05552, 2021.

GNN based Recommendation

Collaborative Filtering

- Graph Convolutional Neural Networks for Web-Scale Recommender Systems (KDD’18)
- Graph Convolutional Matrix Completion (KDD’18 Deep Learning Day)
- Neural Graph Collaborative Filtering (SIGIR’19)
- LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation (SIGIR’20)

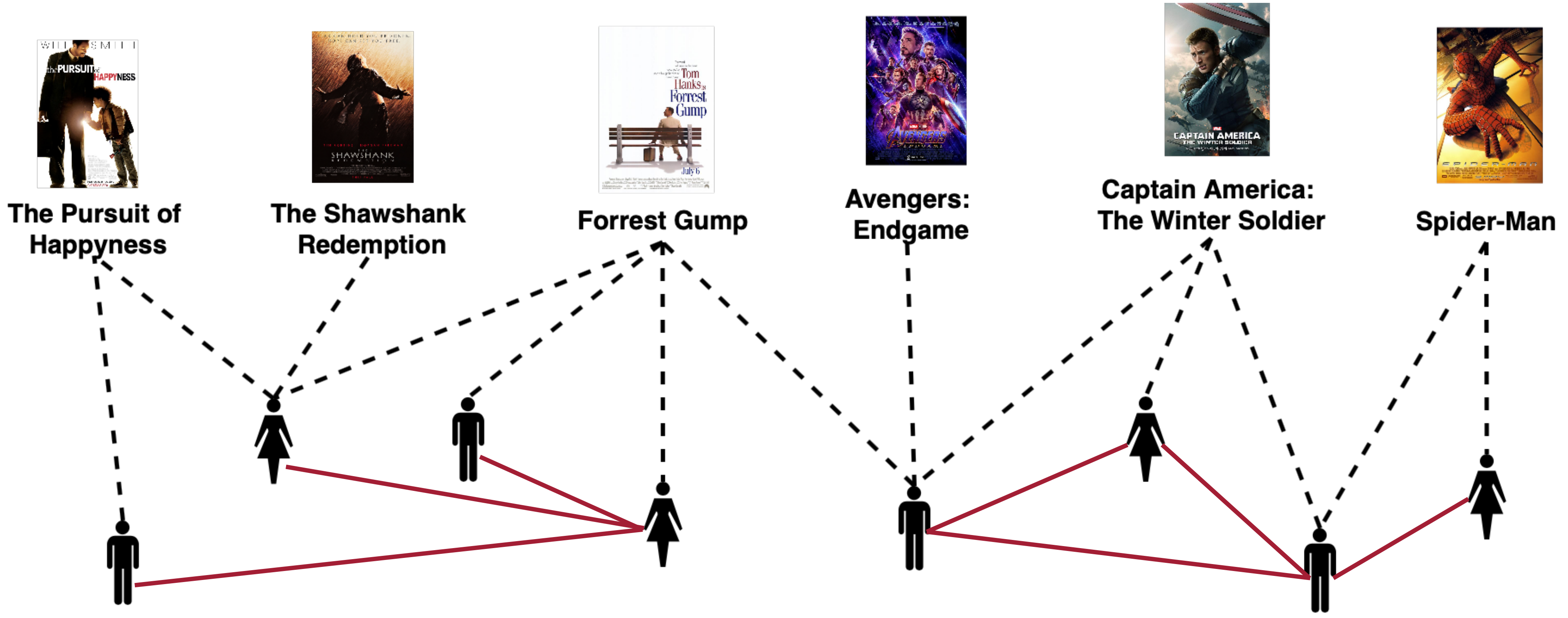
Collaborative Filtering with Side Information (Users/Items)

- **Social Recommendation (Users)**
  - Graph Neural Network for Social Recommendation (WWW’19)
  - A Neural Influence Diffusion Model for Social Recommendation (SIGIR’19)
  - A Graph Neural Network Framework for Social Recommendations (TKDE’20)
- **Knowledge-graph-aware Recommendation (Items)**
  - Knowledge Graph Convolutional Networks for Recommender Systems with Label Smoothness Regularization (KDD’19 and WWW’19)
  - KGAT: Knowledge Graph Attention Network for Recommendation (KDD’19)

Social Recommendation

Side information about users: social networks

- **Users' preferences** **are similar to or influenced by** **the people around them (nearer neighbours)** [Tang et. al, 2013]

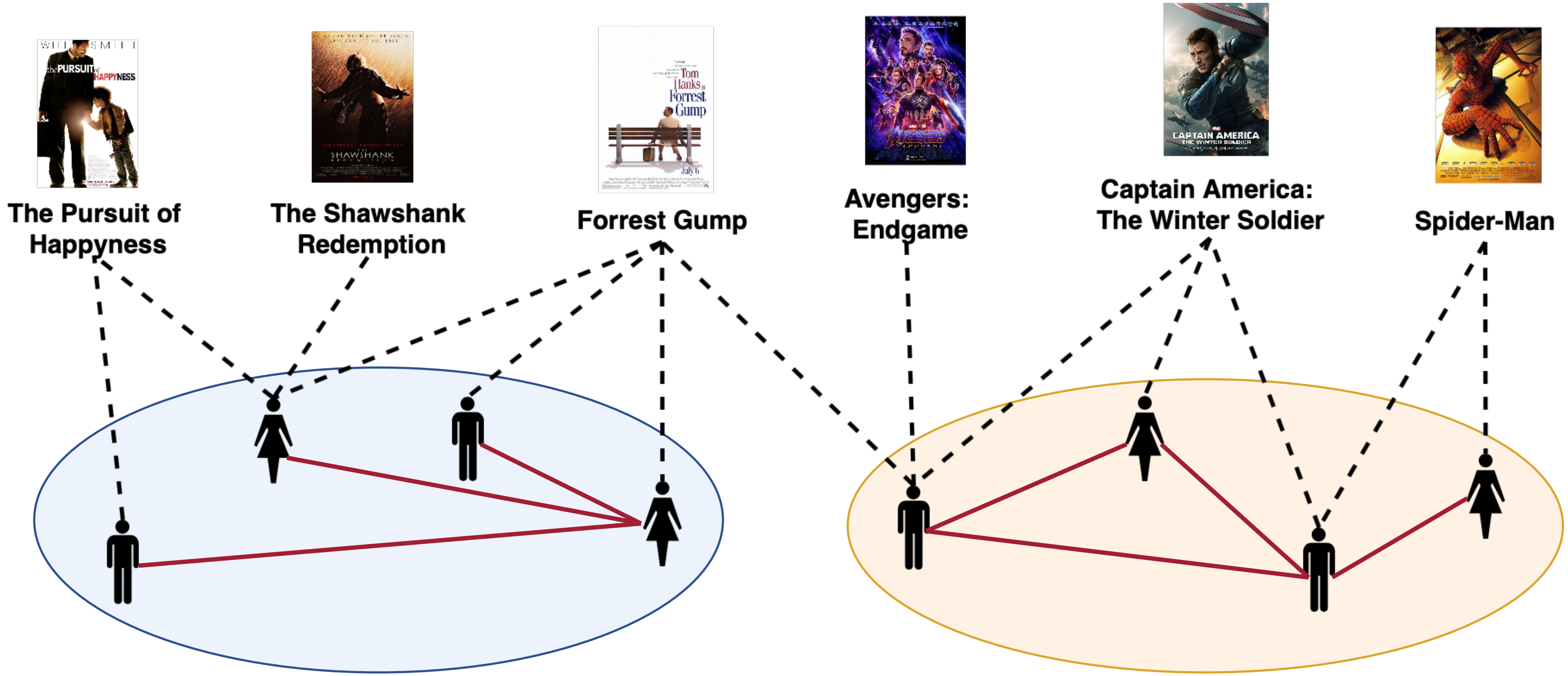


Social recommendation: a review, SNAM, 2013

Social Recommendation

Side information about users: social networks

- Users' preferences **are similar to or influenced by** the people around them (nearer neighbours) [Tang et. al, 2013]

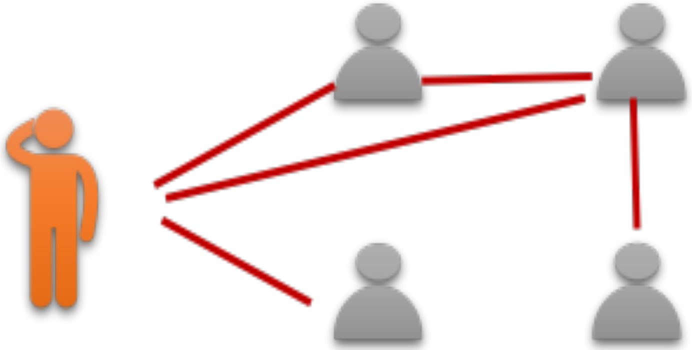


Social recommendation: a review, SNAM, 2013

GraphRec

Graph Data in Social Recommendation

User-Item Graph

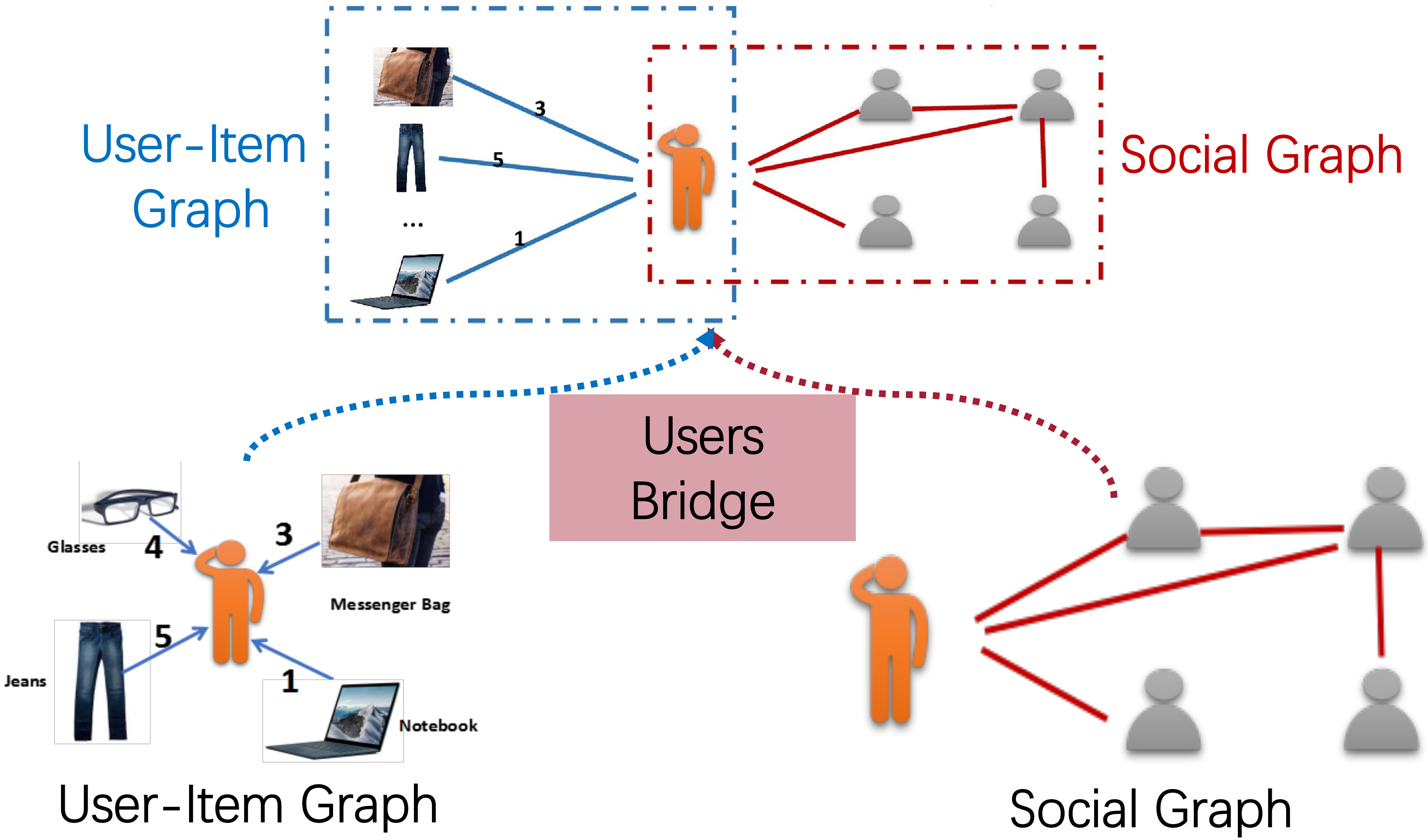


Social Graph

Graph Neural Networks for Social Recommendation. WWW 2019.

GraphRec

Graph Data in Social Recommendation

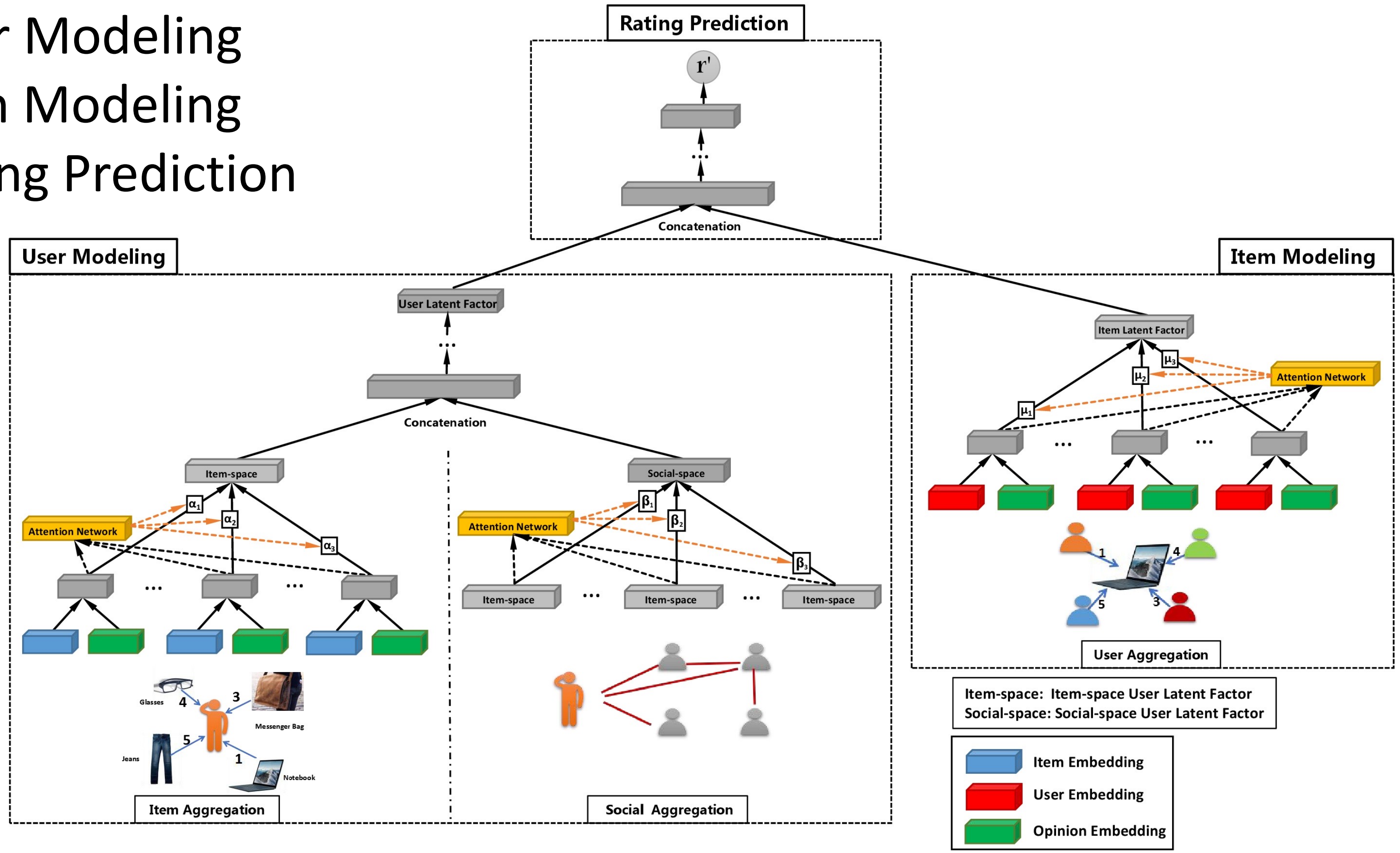


Graph Neural Networks for Social Recommendation. WWW 2019.

GraphRec

Three Components:

- User Modeling
- Item Modeling
- Rating Prediction

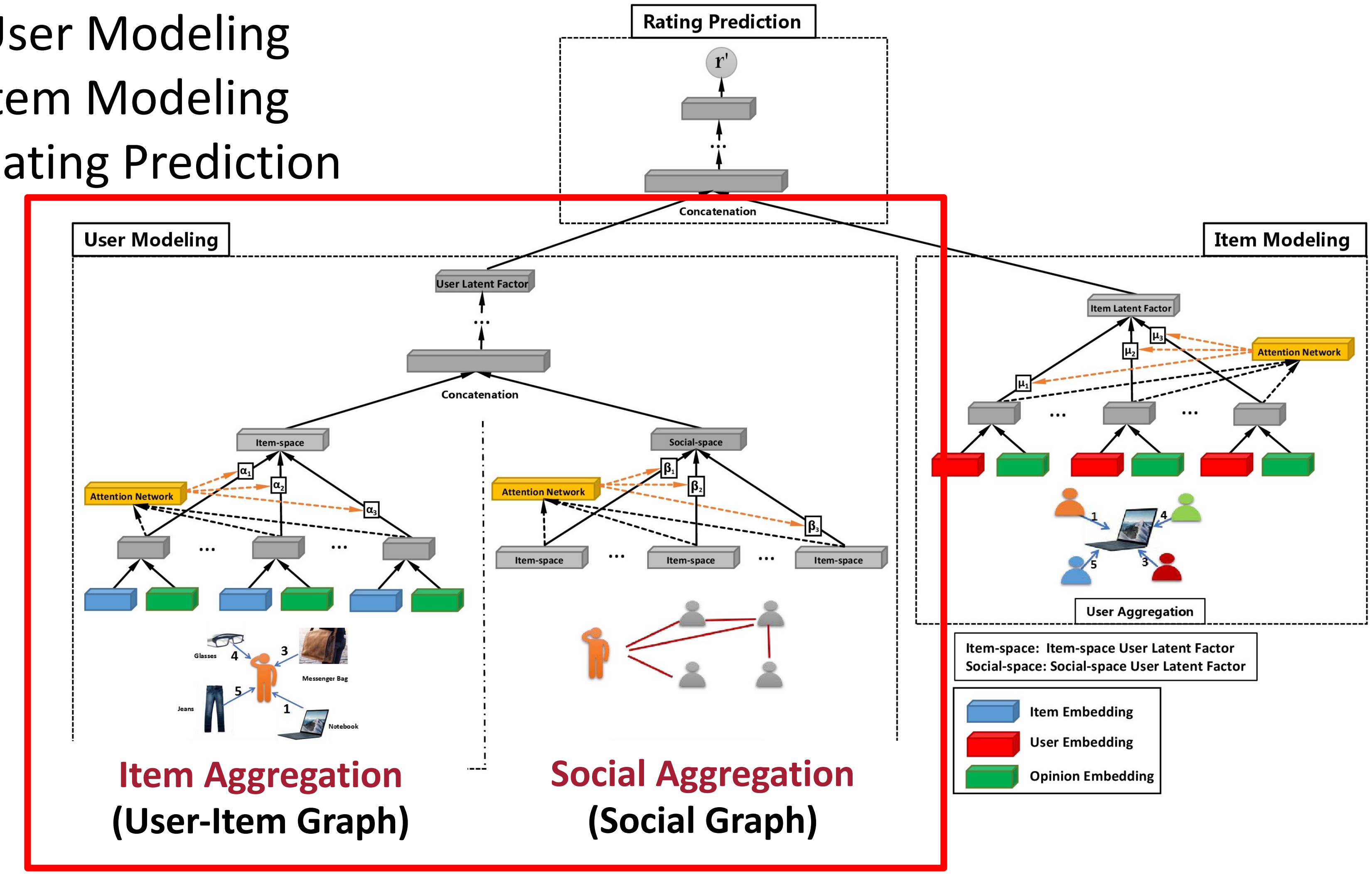


Graph Neural Networks for Social Recommendation. WWW 2019.

GraphRec

Three Components:

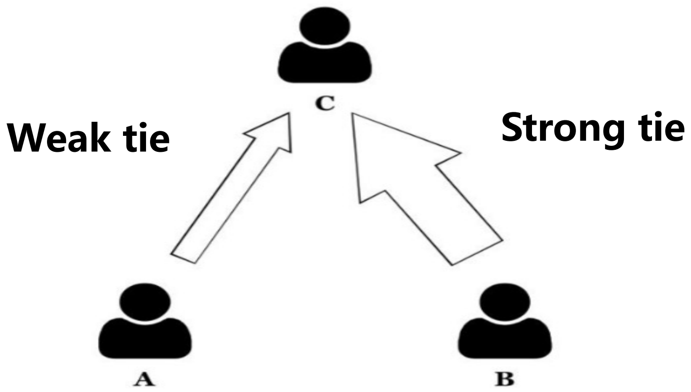
- User Modeling
- Item Modeling
- Rating Prediction



Graph Neural Networks for Social Recommendation. WWW 2019.

GraphRec: User Modeling

- Social Aggregation in user-user social graph
- Users are likely to share more similar tastes with strong ties than weak ties.

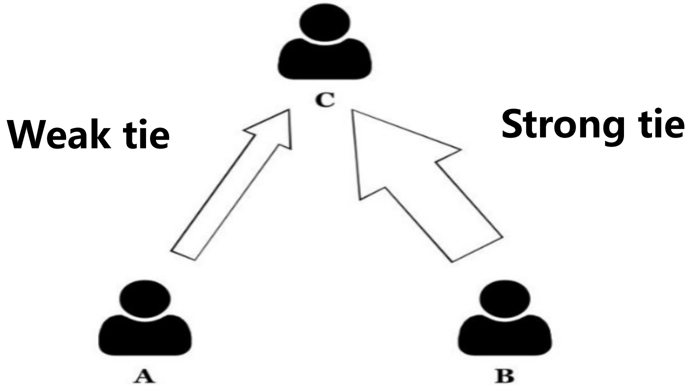


Graph Neural Networks for Social Recommendation. WWW 2019.

GraphRec: User Modeling

- Social Aggregation in user-user social graph
- Users are likely to share more similar tastes with strong ties than weak ties.

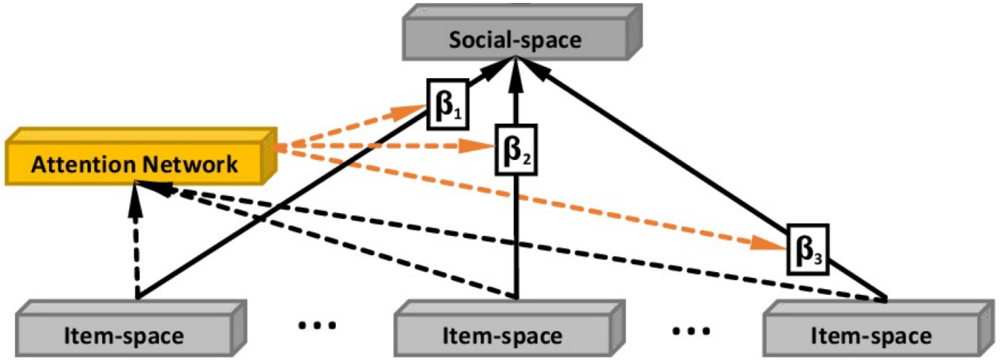
Attention network to differentiate the importance weight.



Aggregating item-space users messages from social neighbors

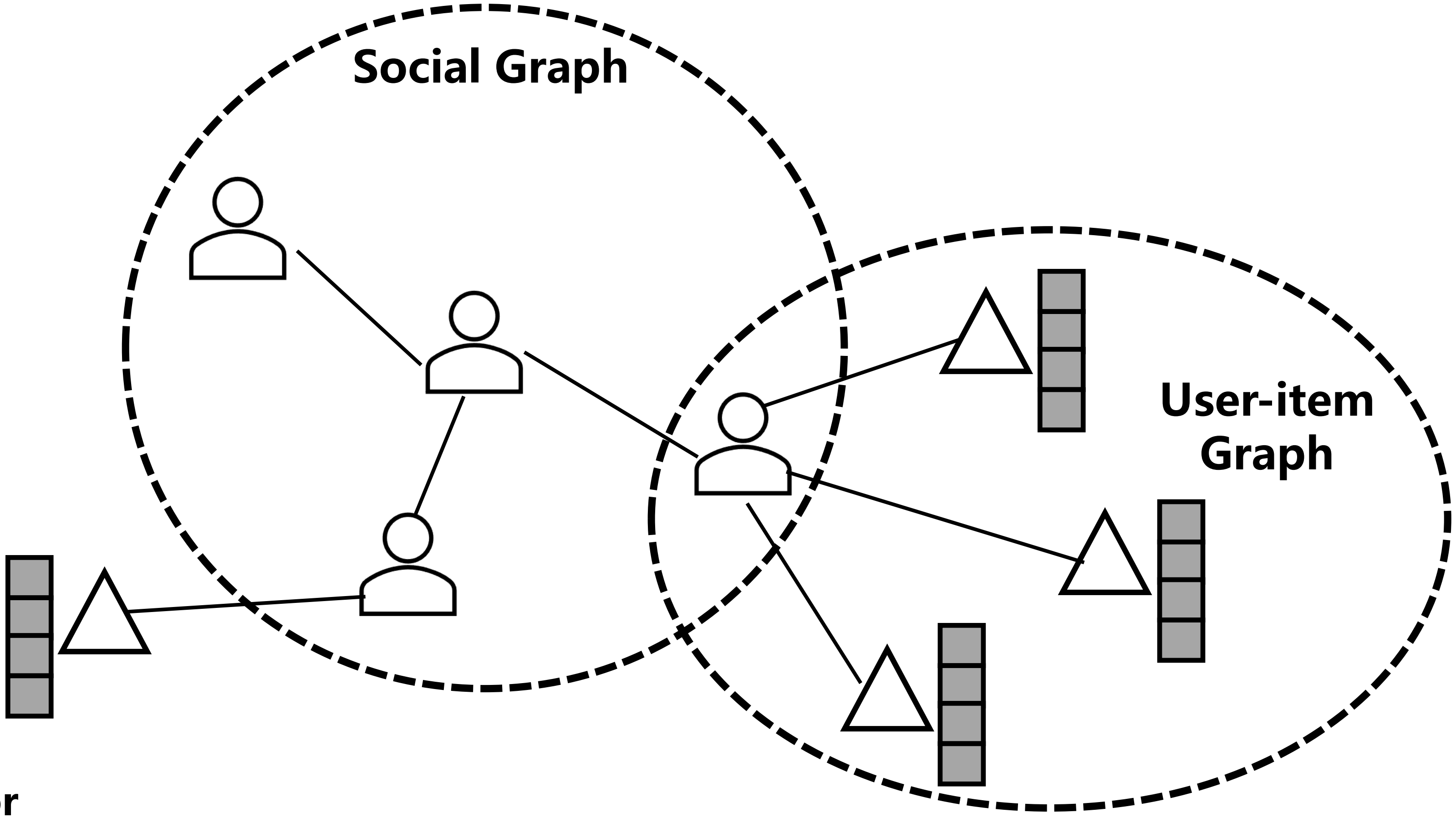
$$\mathbf{h}_i^S = \sigma(\mathbf{W} \cdot \left\{ \sum_{o \in N(i)} \boxed{\beta_{io}} \mathbf{h}_o^I \right\} + \mathbf{b})$$

attentive weight



Graph Neural Networks for Social Recommendation. WWW 2019.

User Modeling: Social Aggregation



Social-space
user latent factor

Item-space
user latent factor

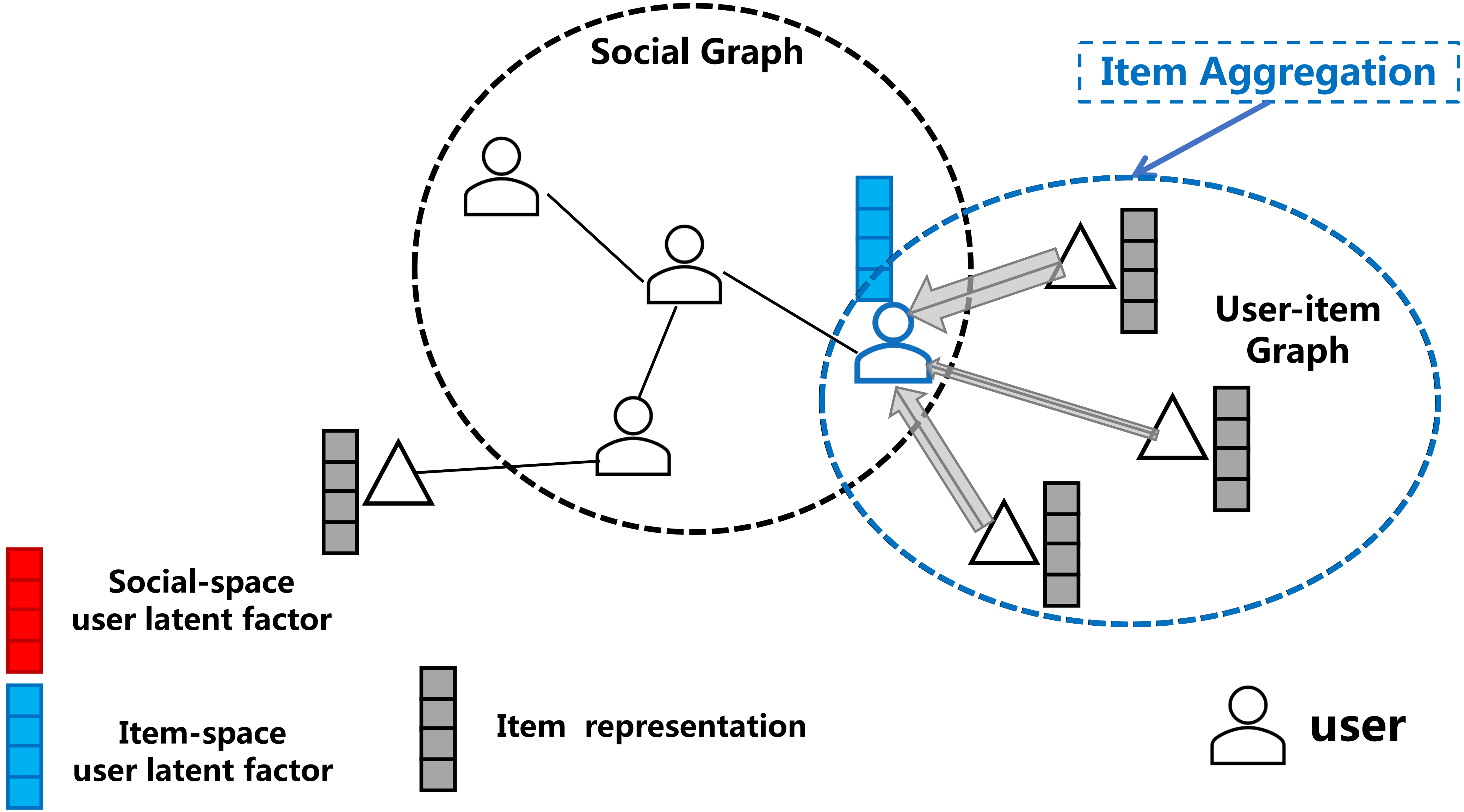
Item representation

user

item

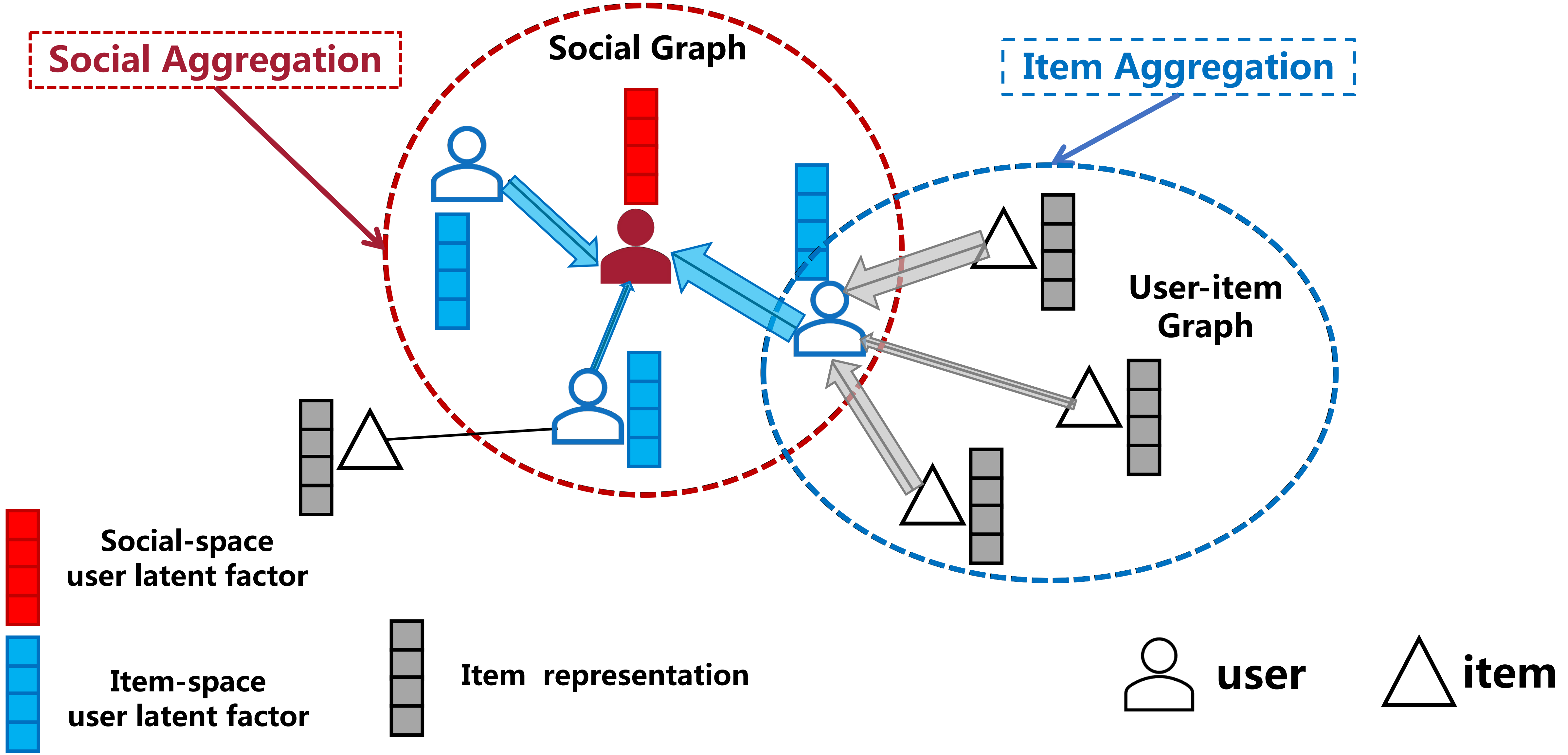
Graph Neural Networks for Social Recommendation. WWW 2019.

User Modeling: Social Aggregation



Graph Neural Networks for Social Recommendation. WWW 2019.

User Modeling: Social Aggregation



Graph Neural Networks for Social Recommendation. WWW 2019.

GNNs based Recommendation

Collaborative Filtering

- Graph Convolutional Neural Networks for Web-Scale Recommender Systems (KDD'18)
- Graph Convolutional Matrix Completion (KDD'18 Deep Learning Day)
- Neural Graph Collaborative Filtering (SIGIR'19)
- LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation (SIGIR'20)

Collaborative Filtering with Side Information (Users/Items)

Social Recommendation (Users)

- Graph Neural Network for Social Recommendation (WWW'19)
- A Neural Influence Diffusion Model for Social Recommendation (SIGIR'19)
- A Graph Neural Network Framework for Social Recommendations (TKDE'20)

Knowledge-graph-aware Recommendation (Items)

- Knowledge Graph Convolutional Networks for Recommender Systems with Label Smoothness Regularization (KDD'19 and WWW'19)
- KGAT: Knowledge Graph Attention Network for Recommendation (KDD'19)

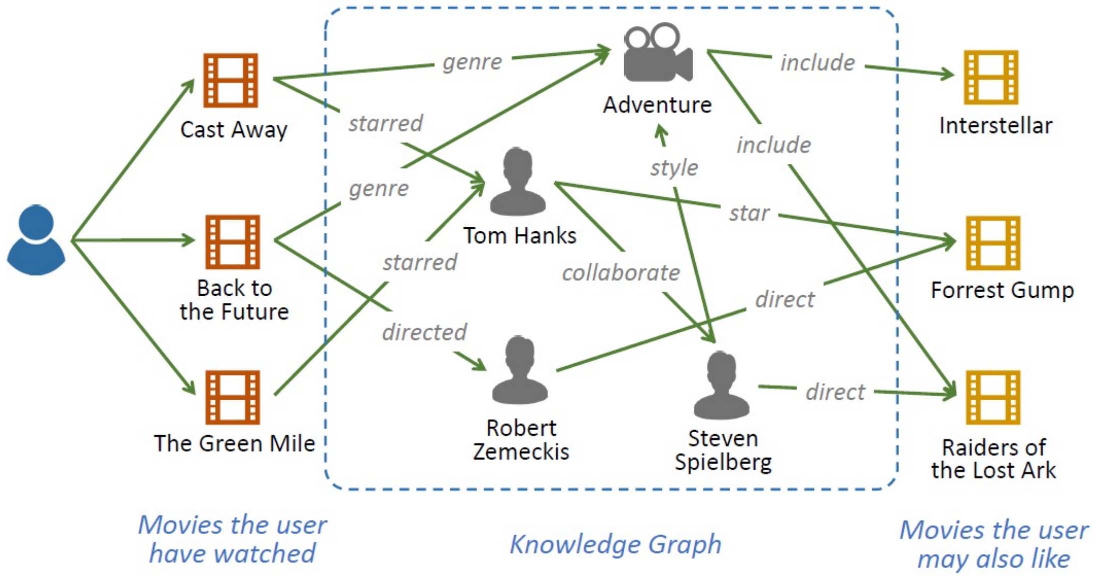
KGCN (WWW'19)

Side information about items: Knowledge Graph (KG)

Heterogeneous Graph:

- Nodes: entities (Items)
- Edges: relations

Triples: (head, relation, tail)



Knowledge Graph Convolutional Networks for Recommender Systems, WWW 2019.

KGCN (WWW'19)

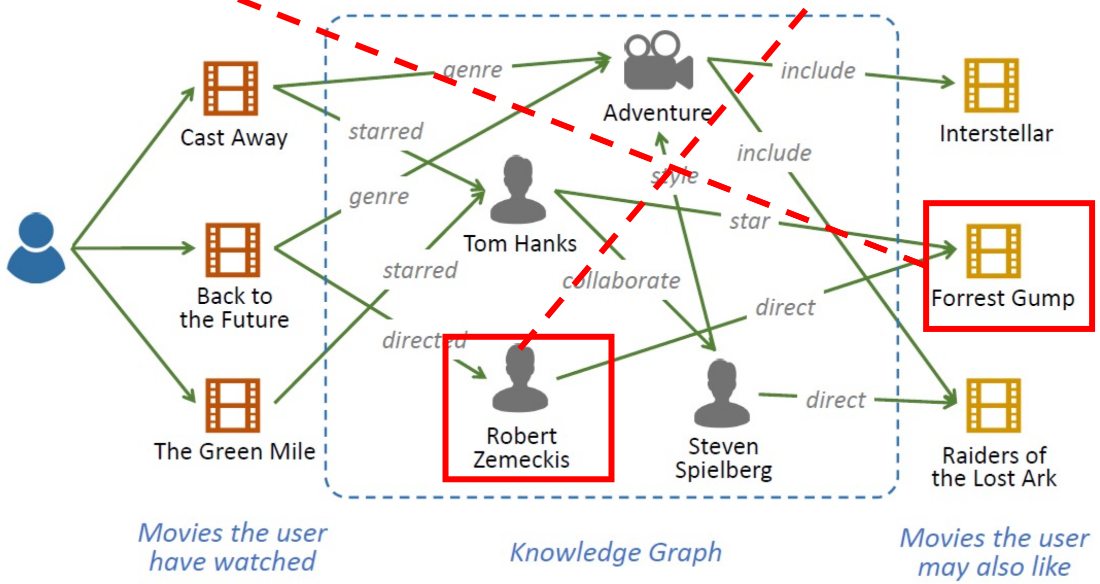
Side information about items: Knowledge Graph (KG)

Heterogeneous Graph:

- Nodes: entities (Items)
- Edges: relations

Triples: (head, relation, tail)

Forrest Gump (items) $\xrightarrow{\textbf{film.film.director}}$ **Robert Zemeckis**



Knowledge Graph Convolutional Networks for Recommender Systems, WWW 2019.

KGCN (WWW’19)

Side information about items: Knowledge Graph (KG)

Heterogeneous Graph:

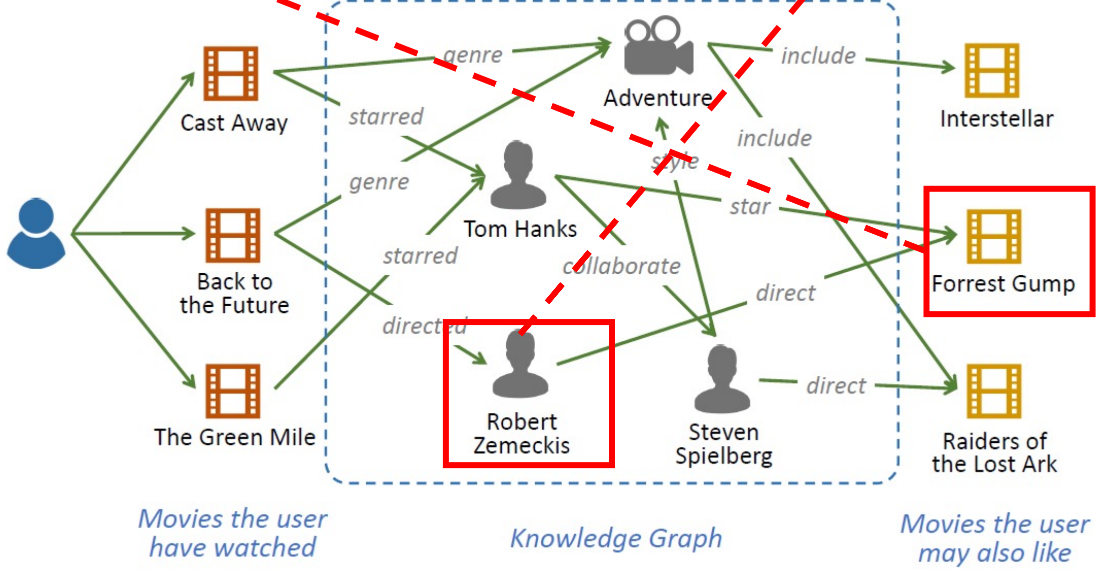
- Nodes: entities (Items)
- Edges: relations

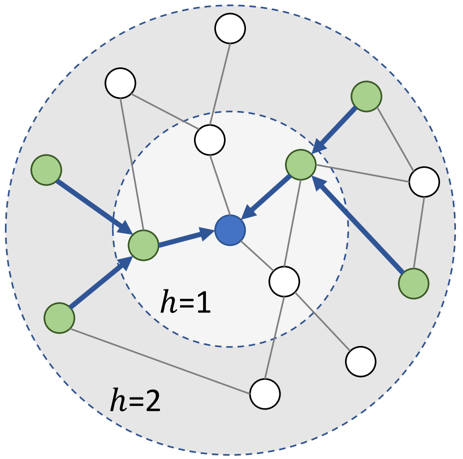
Triples: (head, relation, tail)

Forest Gump (items) $\xrightarrow{\text{film.film.director}}$ **Robert Zemeckis**

$$\hat{y}_{uv} = f(\mathbf{u}, \boxed{\mathbf{v}^u})$$

GNNs?

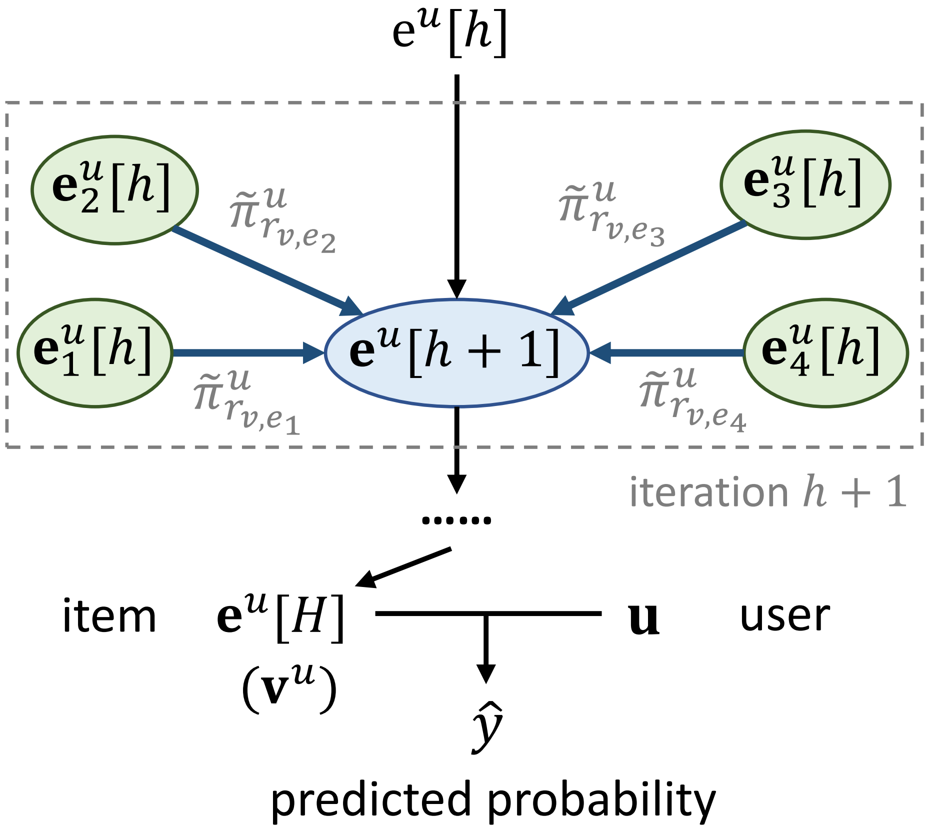




Knowledge Graph Convolutional Networks for Recommender Systems, WWW 2019.

KGCN (WWW'19)

- Representation Aggregation of neighboring entities



Transform a heterogeneous KG into a user-personalized weighted graph

Knowledge Graph Convolutional Networks for Recommender Systems. WWW 2019.

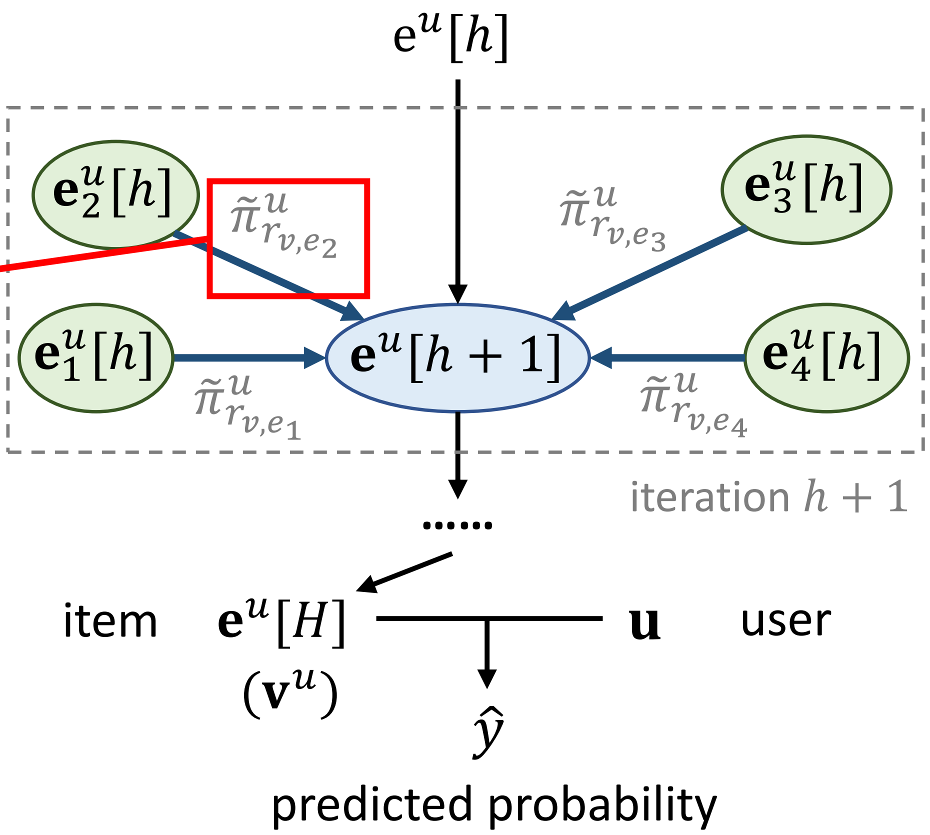
KGCN (WWW'19)

- Representation Aggregation of neighboring entities

$$\pi_r^u = g(\mathbf{u}, \mathbf{r})$$ user-specific relation (e.g., inner product)

$$\tilde{\pi}_{r_{v,e}}^u = \frac{\exp(\pi_{r_{v,e}}^u)}{\sum_{e \in \mathcal{N}(v)} \exp(\pi_{r_{v,e}}^u)}$$

Normalized



Transform a heterogeneous KG into a user-personalized weighted graph

KGCN (WWW'19)

- Representation Aggregation of neighboring entities

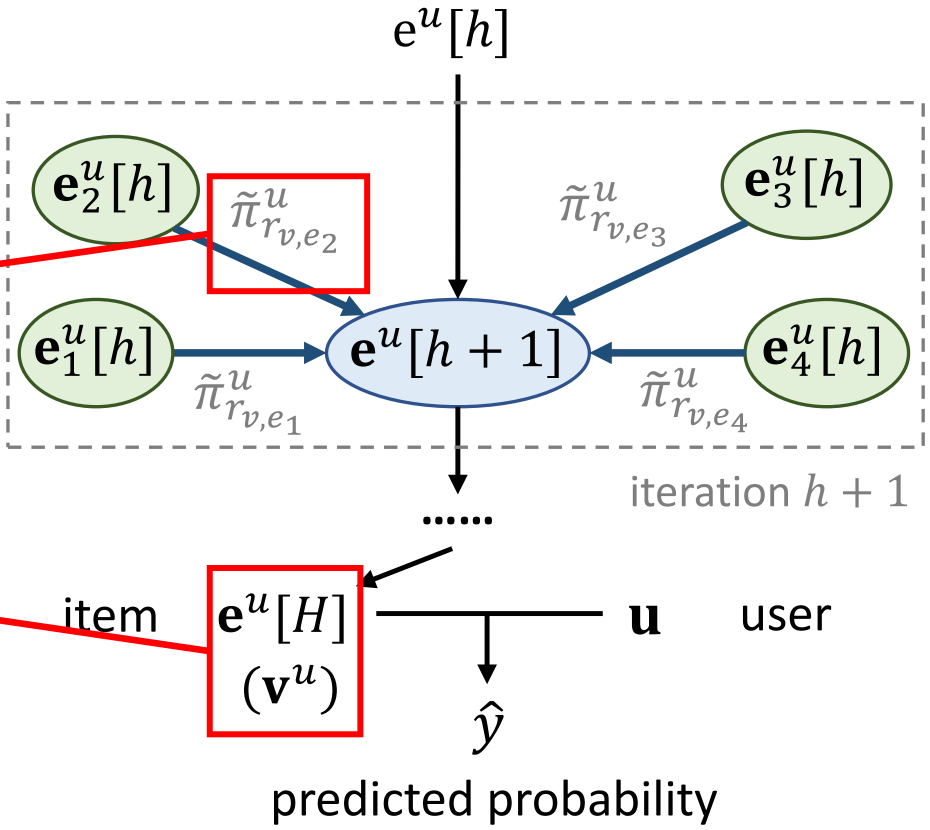
$$\pi_r^u = g(\mathbf{u}, \mathbf{r})$$ user-specific relation (e.g., inner product)

$$\tilde{\pi}_{r_{v,e}}^u = \frac{\exp(\pi_{r_{v,e}}^u)}{\sum_{e \in \mathcal{N}(v)} \exp(\pi_{r_{v,e}}^u)}$$

Normalized

$$\mathbf{v}_{\mathcal{N}(v)}^u = \sum_{e \in \mathcal{N}(v)} \tilde{\pi}_{r_{v,e}}^u \mathbf{e}$$

$$\hat{y}_{uv} = f(\mathbf{u}, \mathbf{v}^u)$$

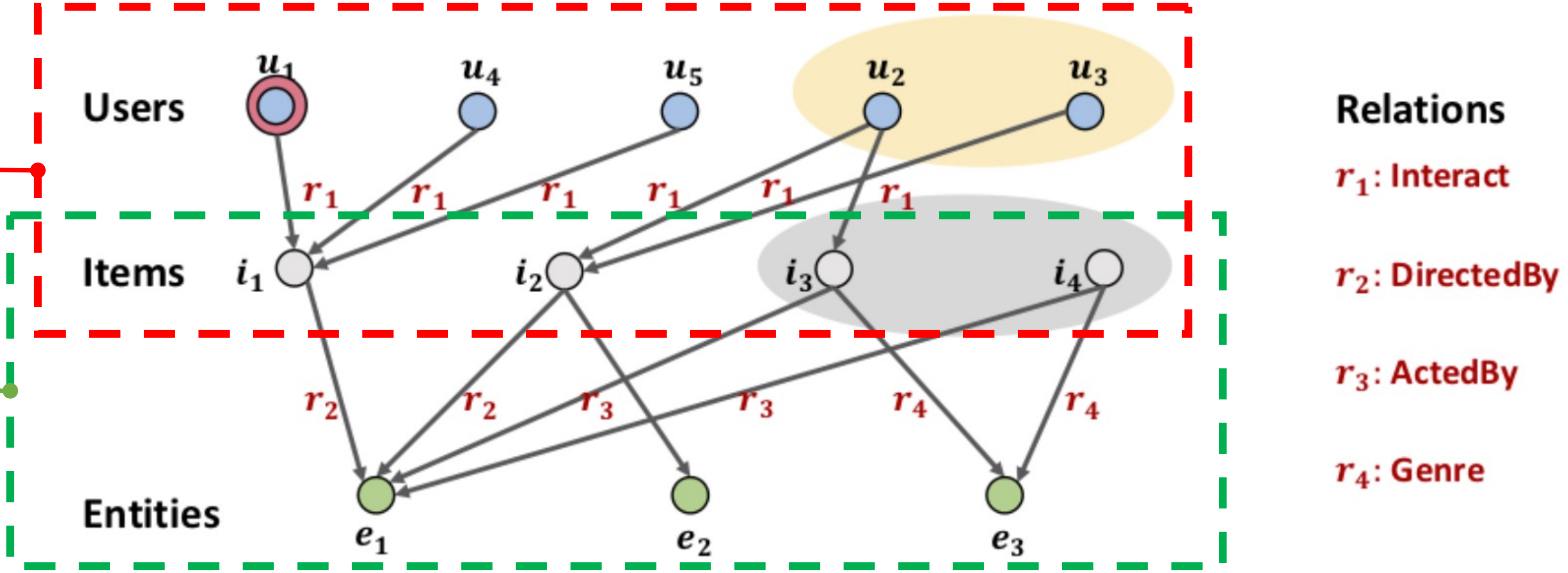


Transform a heterogeneous KG into a user-personalized weighted graph

KGAT

User-item Graph $u_1 \xrightarrow{r_1} i_1$

Knowledge Graph $i_1 \xrightarrow{r_2} e_1$



KGAT: Knowledge graph attention network for recommendation. KDD 2019.

KGAT

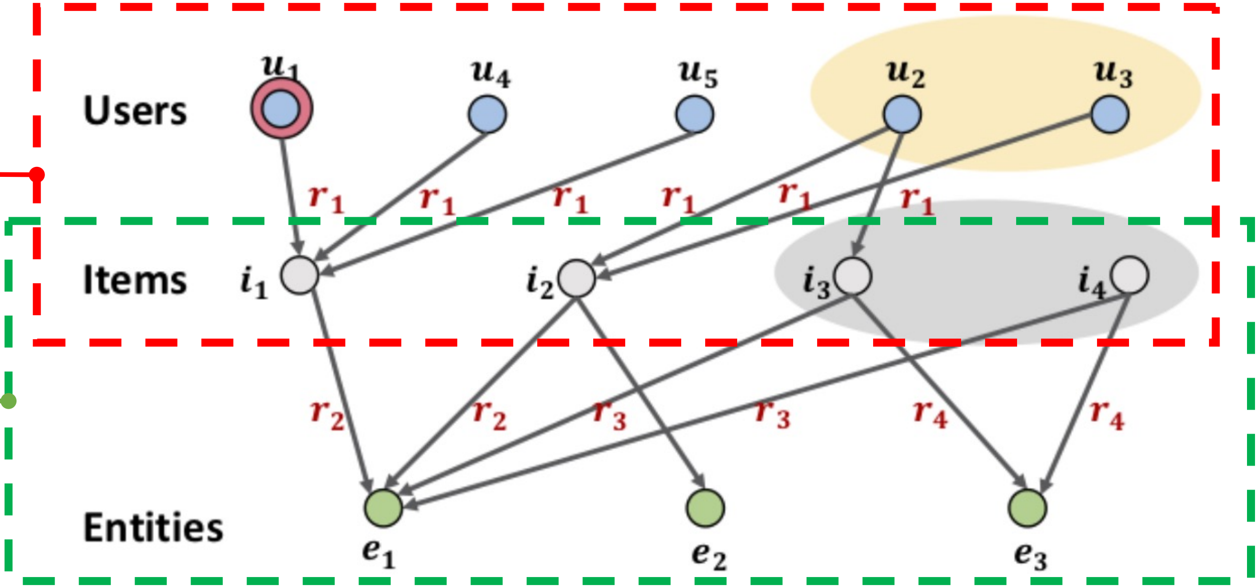
User-item Graph $u_1 \xrightarrow{r_1} i_1$

+

Knowledge Graph $i_1 \xrightarrow{r_2} e_1$

=

Collaborative Knowledge Graph (CKG)



Relations

r_1: Interact

r_2: DirectedBy

r_3: ActedBy

r_4: Genre

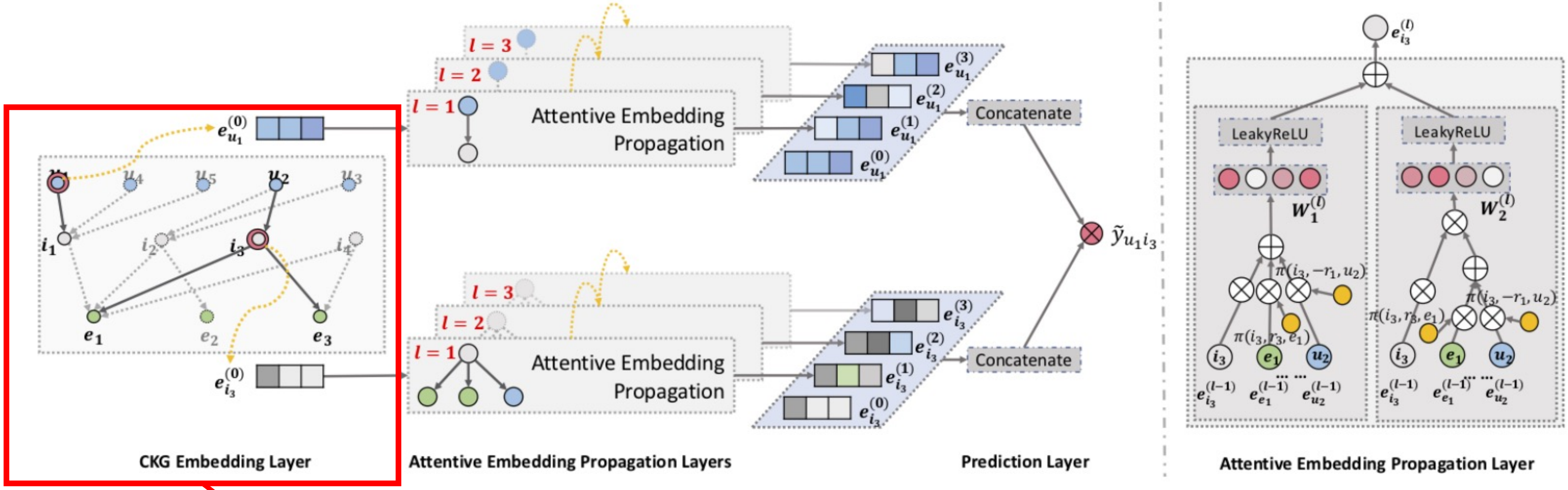
To fully exploit high-order relations in CKG
e.g., the long-range connectivities:

$$u_1 \xrightarrow{r_1} i_1 \xrightarrow{r_2} e_1 \xrightarrow{-r_2} i_2 \xrightarrow{-r_1} \{u_2, u_3\}$$

$$u_1 \xrightarrow{r_1} i_1 \xrightarrow{r_2} e_1 \xrightarrow{-r_3} \{i_3, i_4\}$$

KGAT: Knowledge graph attention network for recommendation. KDD 2019.

KGAT

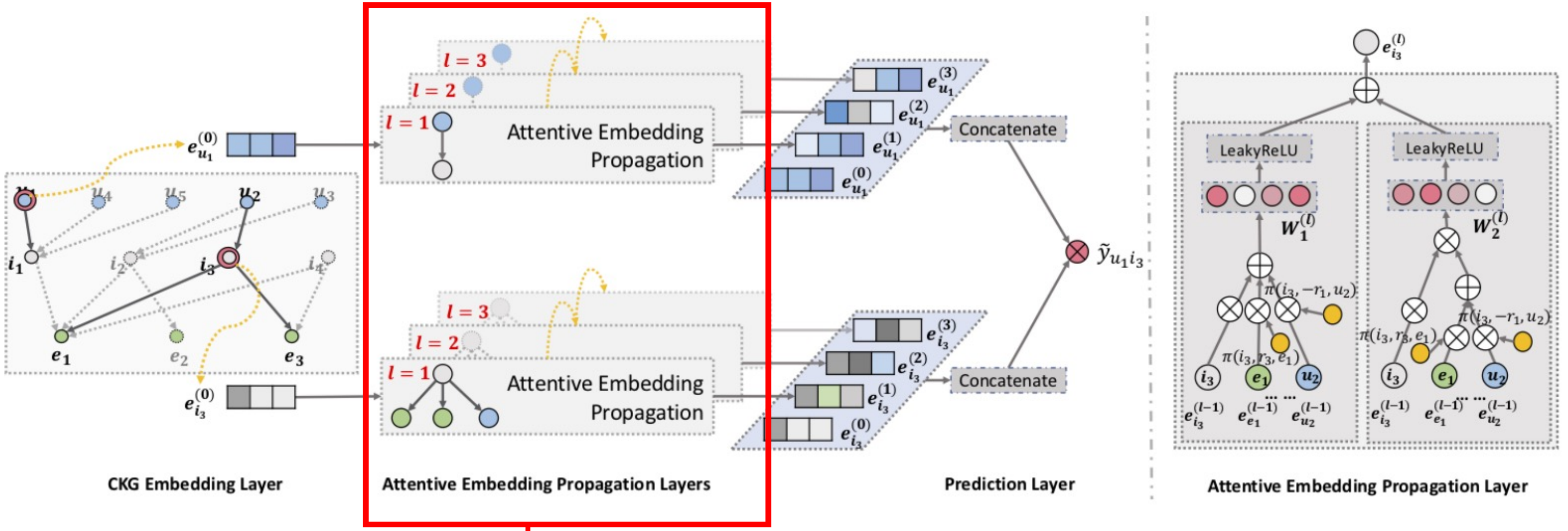


$$g(h, r, t) = \|\mathbf{W}_r \mathbf{e}_h + \mathbf{e}_r - \mathbf{W}_r \mathbf{e}_t\|_2^2$$

$$\mathcal{L}_{\mathrm{KG}} = \sum_{(h,r,t,t') \in \mathcal{T}} -\ln \sigma\Big(g(h, r, t') - g(h, r, t)\Big)$$

KGAT: Knowledge graph attention network for recommendation. KDD 2019.

KGAT



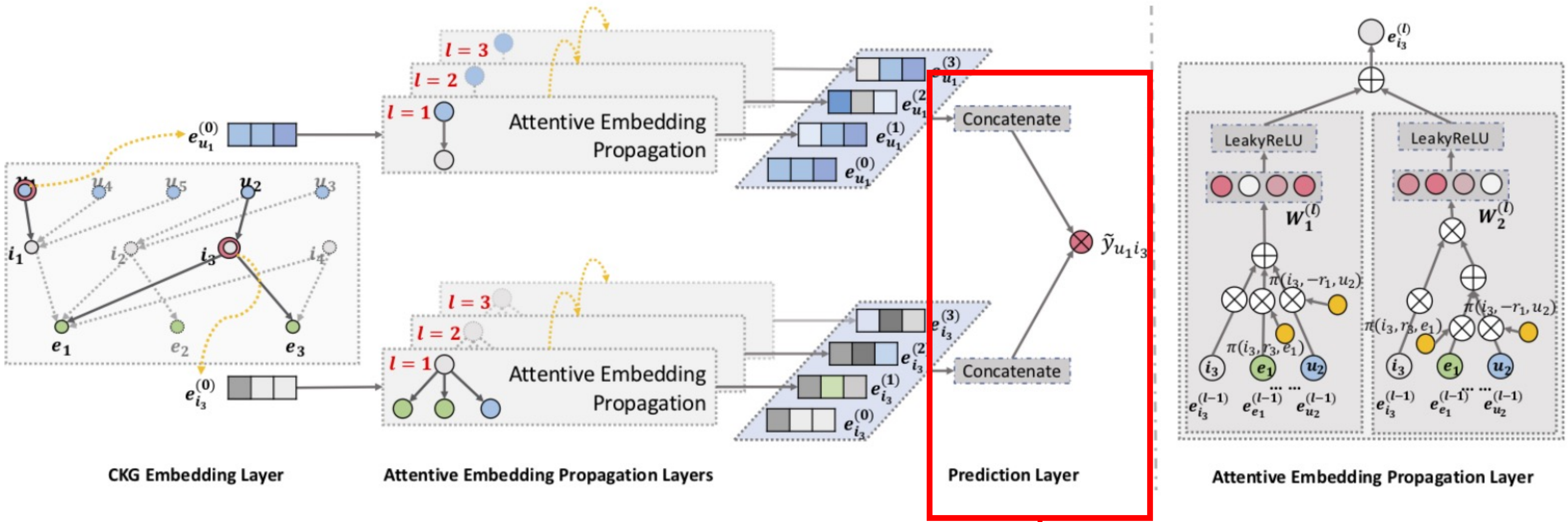
Information Propagation: $\mathbf{e}_{\mathcal{N}_h} = \sum_{(h,r,t)\in\mathcal{N}_h} \pi(h,r,t)\mathbf{e}_t$

Knowledge-aware Attention: $\pi(h,r,t) = (\mathbf{W}_r\mathbf{e}_t)^\top \tanh\big((\mathbf{W}_r\mathbf{e}_h + \mathbf{e}_r)\big)$

Information Aggregation: $f_{\text{Bi-Interaction}} = \text{LeakyReLU}\big(\mathbf{W}_1(\mathbf{e}_h + \mathbf{e}_{\mathcal{N}_h})\big) + \text{LeakyReLU}\big(\mathbf{W}_2(\mathbf{e}_h \odot \mathbf{e}_{\mathcal{N}_h})\big),$

KGAT: Knowledge graph attention network for recommendation. KDD 2019.

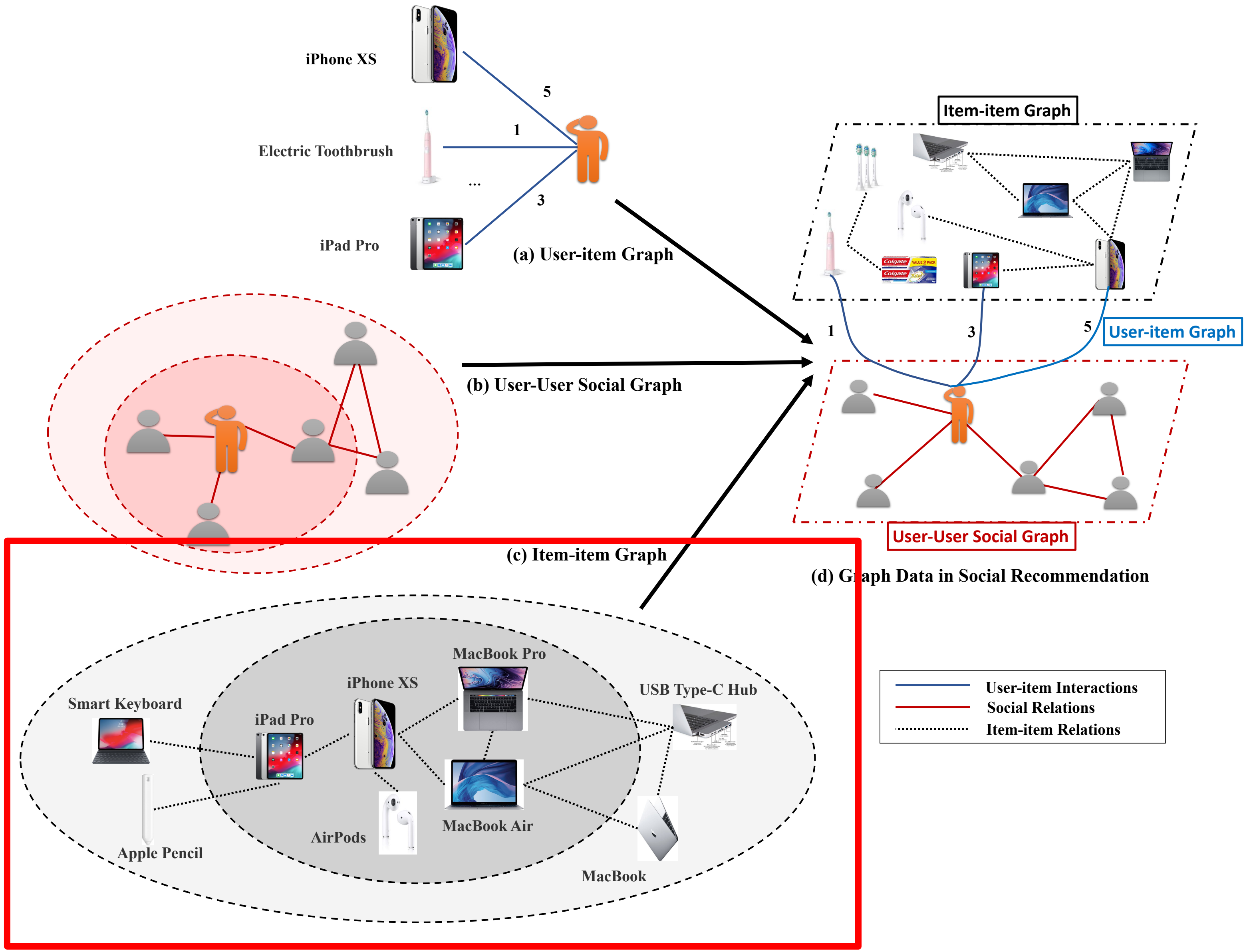
KGAT



$$\mathbf{e}_u^* = \mathbf{e}_u^{(0)} \| \cdots \| \mathbf{e}_u^{(L)}, \quad \mathbf{e}_i^* = \mathbf{e}_i^{(0)} \| \cdots \| \mathbf{e}_i^{(L)} \qquad \hat{y}(u, i) = {\mathbf{e}_u^*}^\top \mathbf{e}_i^*$$

KGAT: Knowledge graph attention network for recommendation. KDD 2019.

GraphRec+

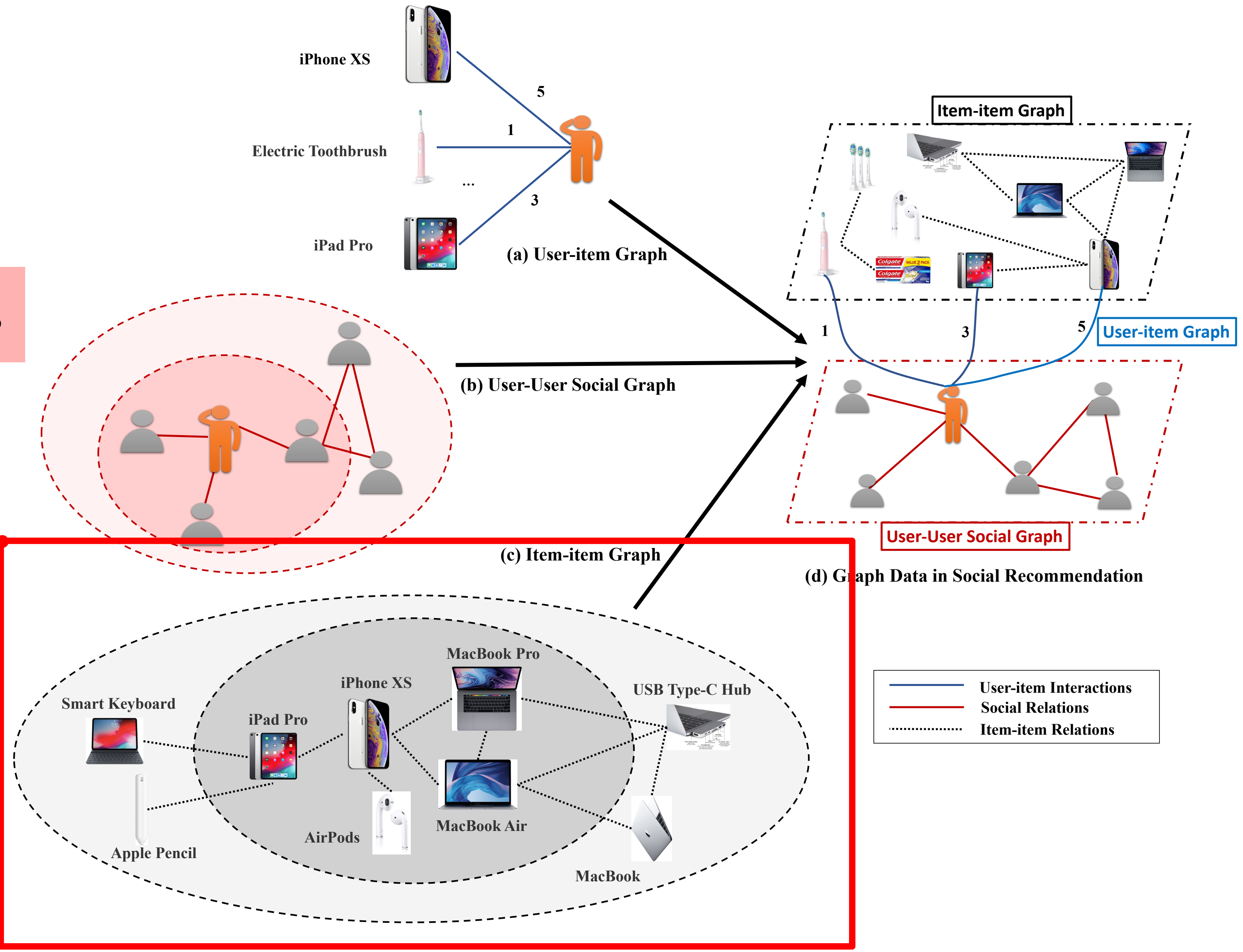


A Graph Neural Network Framework for Social Recommendations, TKDE 2020
Graph Neural Networks for Social Recommendation, WWW 2019.

GraphRec+

Item-item Graph

Substitutable and Complementary Items

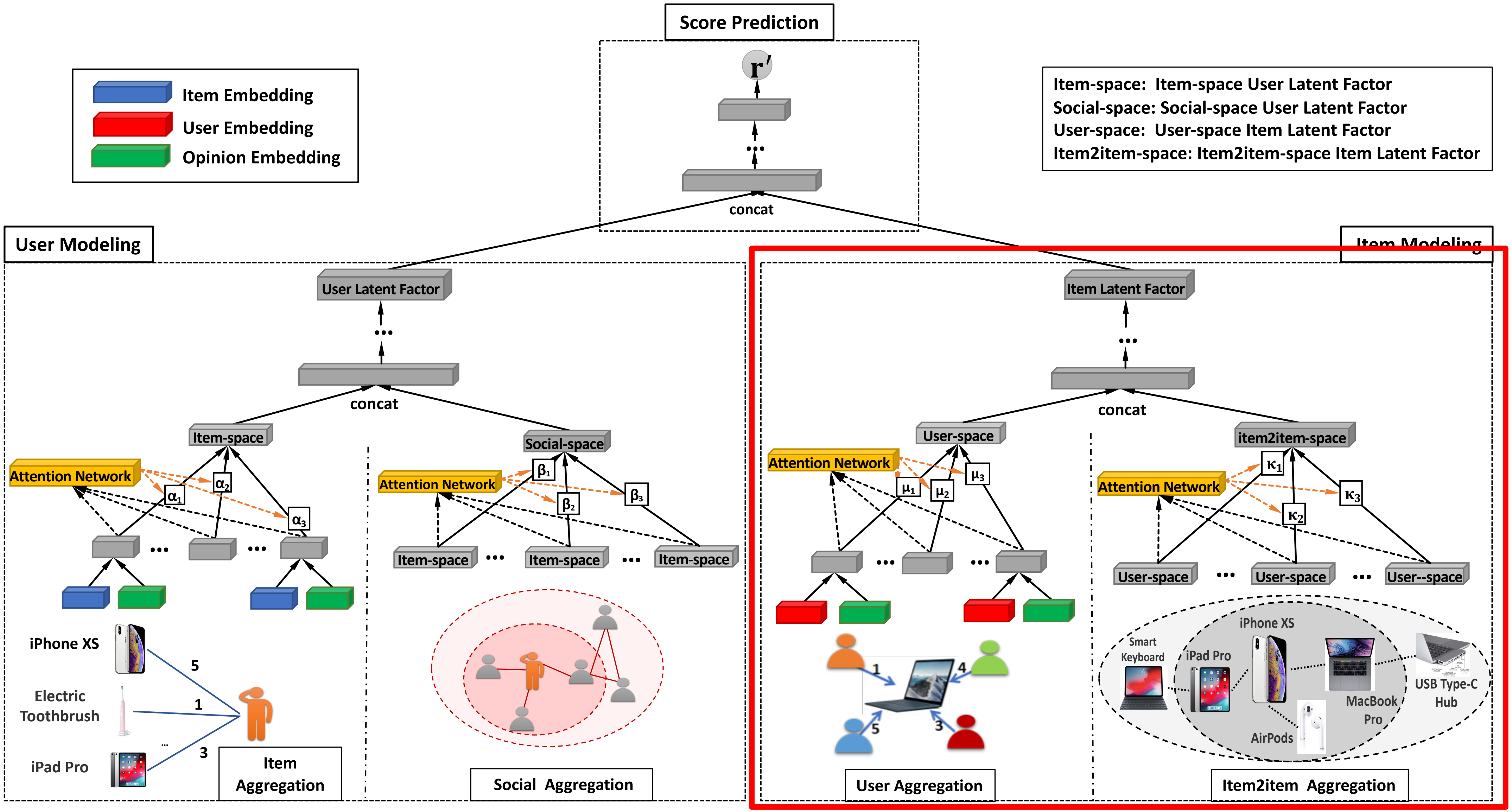


E.g.,

- ‘users who bought A also bought B’
- ‘users who viewed A also viewed B’

A Graph Neural Network Framework for Social Recommendations, TKDE 2020
Graph Neural Networks for Social Recommendation, WWW 2019.

GraphRec+



A Graph Neural Network Framework for Social Recommendations, TKDE 2020
Graph Neural Networks for Social Recommendation, WWW 2019.

Conclusion: Future Directions

- **Depth**

  When the deeper GNNs can help in recommender systems?

Conclusion: Future Directions

- **Depth**

  When the deeper GNNs can help in recommender systems?

- **Security (Data Poisoning Attack & Defense)**
  - Edges

    user-item interactions

    social relations

    knowledge graph
  - Node (users/items) Features
  - Local Graph Structure