Dual Graph Attention Networks for Deep Latent Representa of Multifaceted Social Effects in Recommender Systems

Qitian Wu[1], Hengrui Zhang[1], Xiaofeng Gao[1], Peng He[2], Paul Weng[3], Han Gao[2], Guihai Chen[1]

[1]Shanghai Key Laboratory of Scalable Computing and Systems, Department of Computer Science and Engineering, Shanghai Jiao Tong University

[2]WeChat, Tencent Inc.

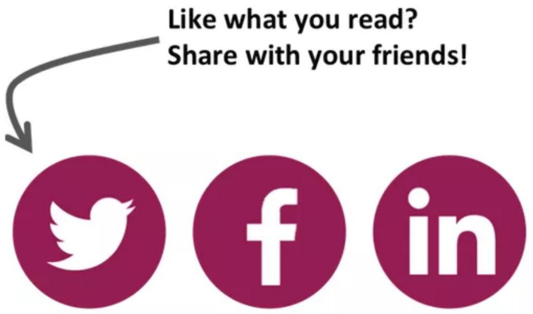
[3]UM–SJTU Joint Institute, Shanghai Jiao Tong University

上海交通大學 SHANGHAI JIAO TONG UNIVERSITY

Tencent 腾讯

UNIVERSITY OF MICHIGAN JOINT INSTITUTE 交大密西根学院

Problem



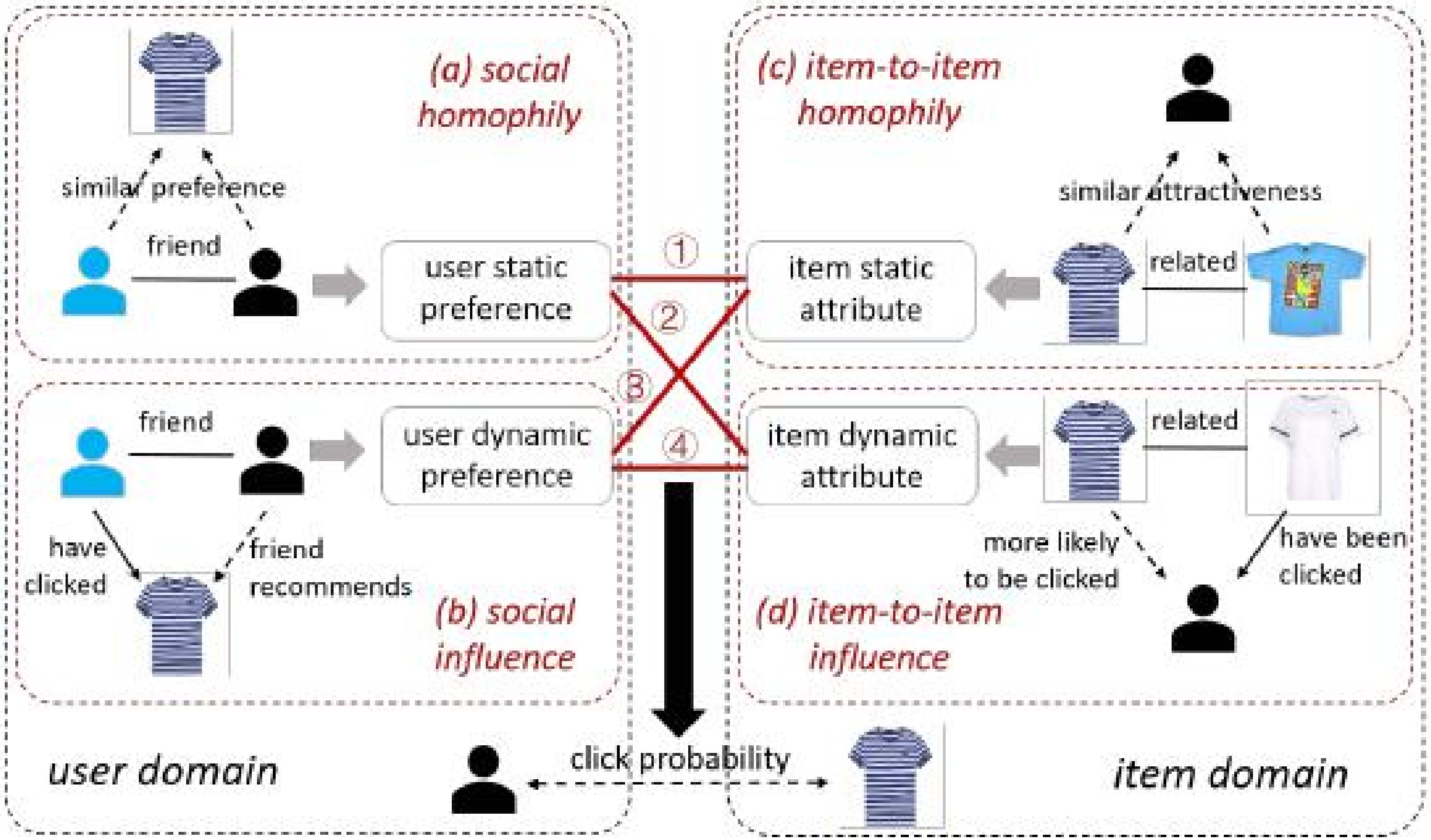
Top Stories on WeChat

(a) (b) (c)

Motivation

four–fold social effects:

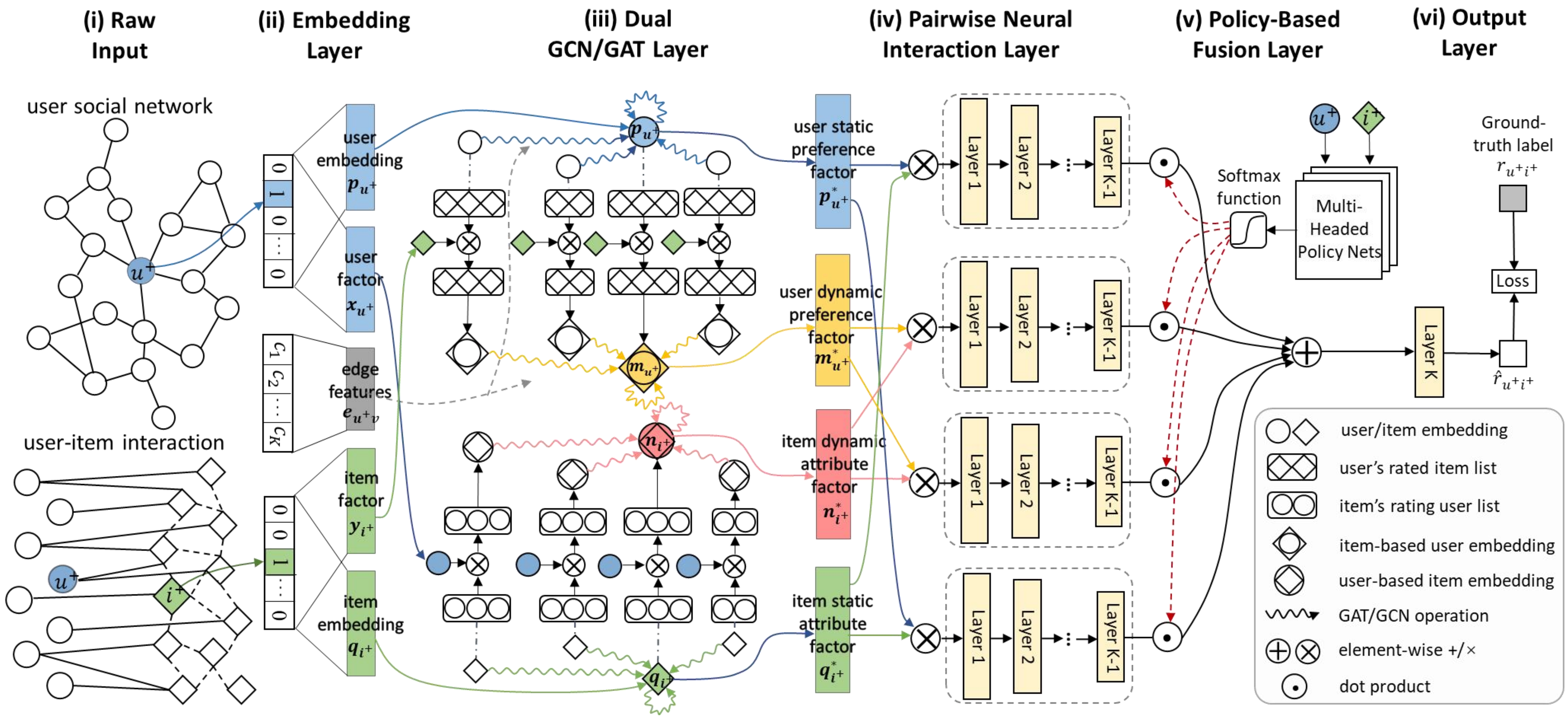
- social homophily
- social influence
- item–to–item homophily
- item–to–item



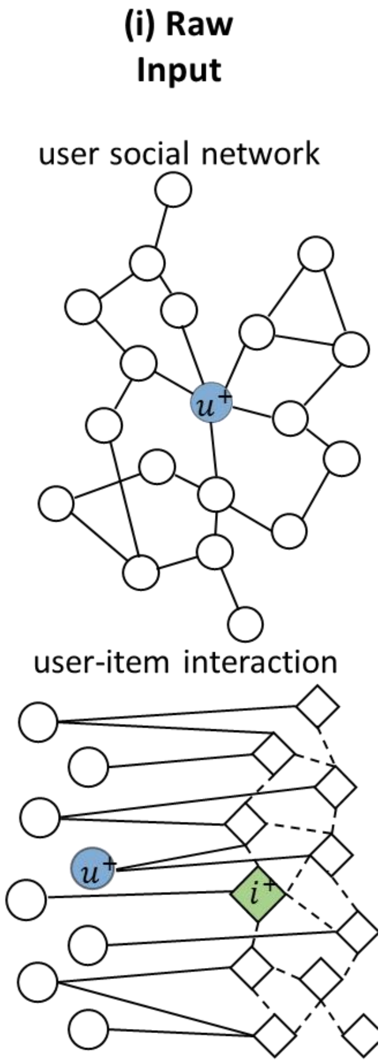
Related Works

Trust Propagation	Use the rating/clicking of friends to estimate the rating/clicking of the targeted user
Trust Matrix Factorization	Adopt latent user preference factor to retrieve the matrix of social trust
Trust Graph Regularization	Constrain the representations of neighbored users in social network to be similar
Trust Network Embedding	Treat user-item interaction network and social network as a whole network and use network embedding technique to encode the topology information as low-dimension features

Our Model:


(i) Raw Input
(ii) Embedding Layer
(iii) Dual GCN/GAT Layer
(iv) Pairwise Neural Interaction Layer
(v) Policy-Based Fusion Layer
(vi) Output Layer
user social network
user-item interaction
user embedding p_{u^+}
user factor x_{u^+}
edge features e_{u^+v}
item factor y_{i^+}
item embedding q_{i^+}
user static preference factor $p^*_{u^+}$
user dynamic preference factor $m^*_{u^+}$
item dynamic attribute factor $n^*_{i^+}$
item static attribute factor $q^*_{i^+}$
Layer 1
Layer 2
Layer K-1
Layer K
Softmax function
Multi-Headed Policy Nets
Ground-truth label $r_{u^+i^+}$
Loss
$\hat{r}_{u^+i^+}$
user/item embedding
user's rated item list
item's rating user list
item-based user embedding
user-based item embedding
GAT/GCN operation
element-wise +/×
dot product

Raw Input



user social network G_U

user-item interaction matrix R

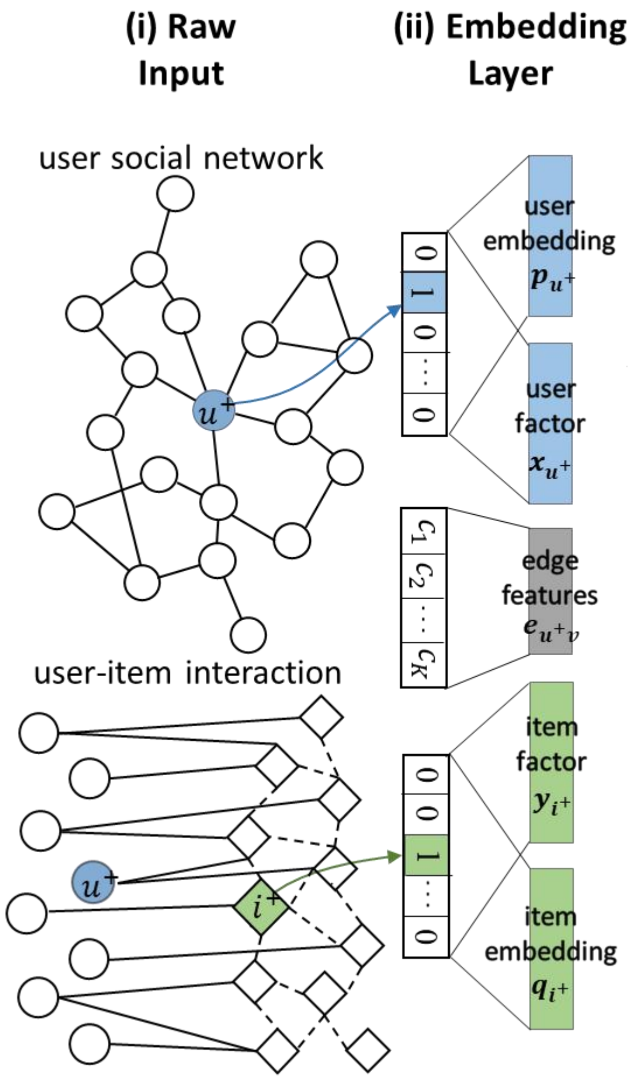
define item implicit network as graph $G_I = (V_I, E_I)$

similarity coefficient s_{ij} of item i and item j :

number of users who clicked both items

item i is related to item j if $s_{ij} > \tau$

Embedding



user embedding $\mathbf{P} = \{\mathbf{p}_u\}_{D \times M}$

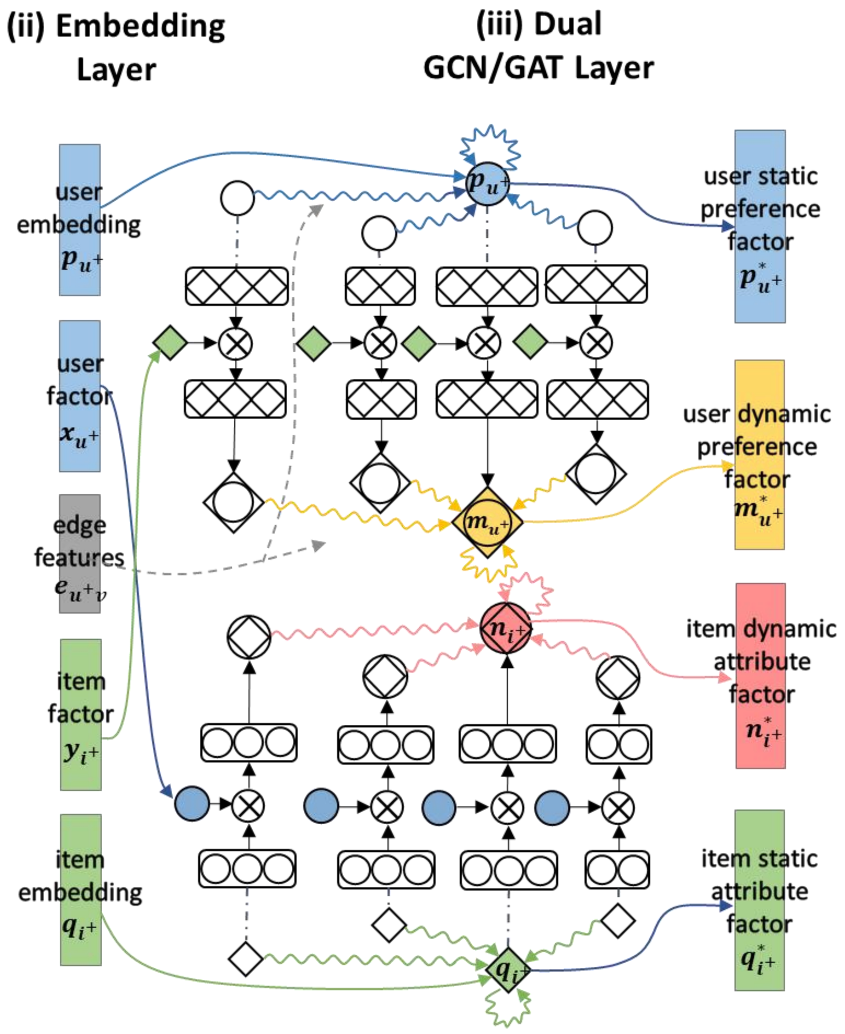
item embedding $\mathbf{Q} = \{\mathbf{q}_i\}_{D \times N}$

user factor $\mathcal{X}_i = \{\mathbf{x}_v | v \in R_U(i)\}$

item factor $\mathcal{Y}_u = \{\mathbf{y}_j | j \in R_I(u)\}$

edge features $e_{u,v}$

Dual GCN/GAT Layer



GAT to capture social influence (user domain)

$$\mathcal{Y}_u^{i+} = \{\mathbf{y}_j \otimes \mathbf{y}_{i^+} | j \in R_I(u)\}$$

$$m_{ud}^{i+} = \max_{j\in R_I(u)} \{y_{jd}\cdot y_{i^+d}\}, \forall d = 1, ..., D$$

$$M^*_{i^+} = \sigma(\mathbf{A}_M(G_U)\mathbf{M}\mathbf{W}_M^T) + \mathbf{b_M}, \mathbf{A}_M(G_U) = \{\alpha_{uv\hat{M},i^+\}_{M\times M}}\}$$

$$\alpha^M_{uv,i^+} = \frac{attn_U(\mathbf{W}_M\mathbf{m}_u^{i^+}, \mathbf{W}_M\mathbf{m}_v^{i^+}, \mathbf{W}_M\mathbf{E}\mathbf{e}_{uv})}{\sum_{w\in\Gamma_U(u)} attn_U(\mathbf{W}_M\mathbf{m}_u^{i^+}, \mathbf{W}_M\mathbf{m}_w^{i^+}, \mathbf{W}_M\mathbf{E}\mathbf{e}_{uv})}$$

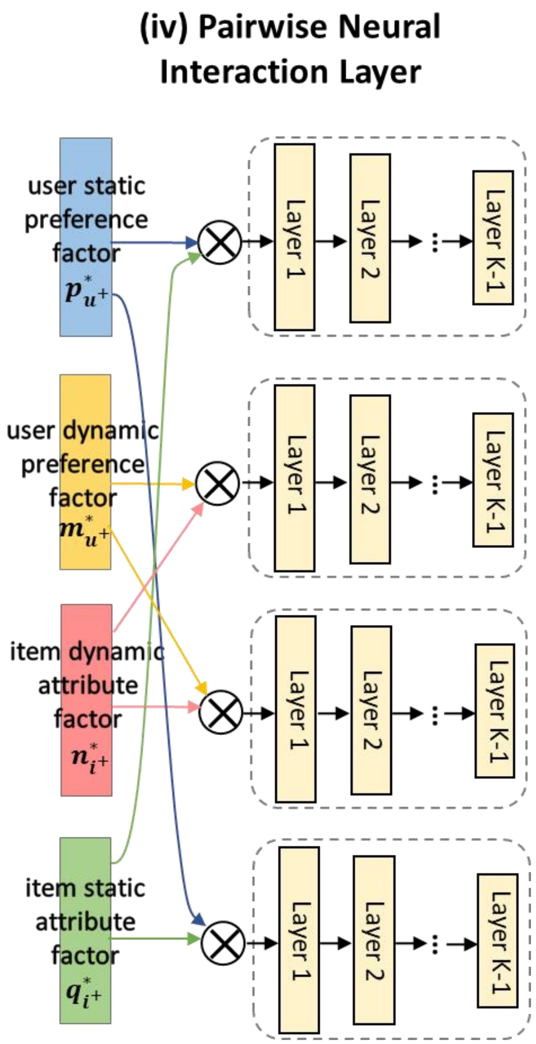
GAT to capture social homophily (user domain)

$$\mathbf{P}^* = \sigma(\mathbf{A}_P(G_U)\mathbf{P}\mathbf{W}_P^T + \mathbf{b}_p)$$

$$\alpha^P_{uv} = \frac{attn_U(\mathbf{W}_p\mathbf{p}_u, \mathbf{W}_p\mathbf{p}_v, \mathbf{W}_E\mathbf{e}_{uv})}{\sum_{w\in\Gamma_U(u)} attn_U(\mathbf{W}_p\mathbf{p}_u, \mathbf{W}_p\mathbf{p}_w, \mathbf{W}_E\mathbf{e}_{uv})}$$

Formulas of GAT in item domain are similar

Pairwise Neural Interaction Layer



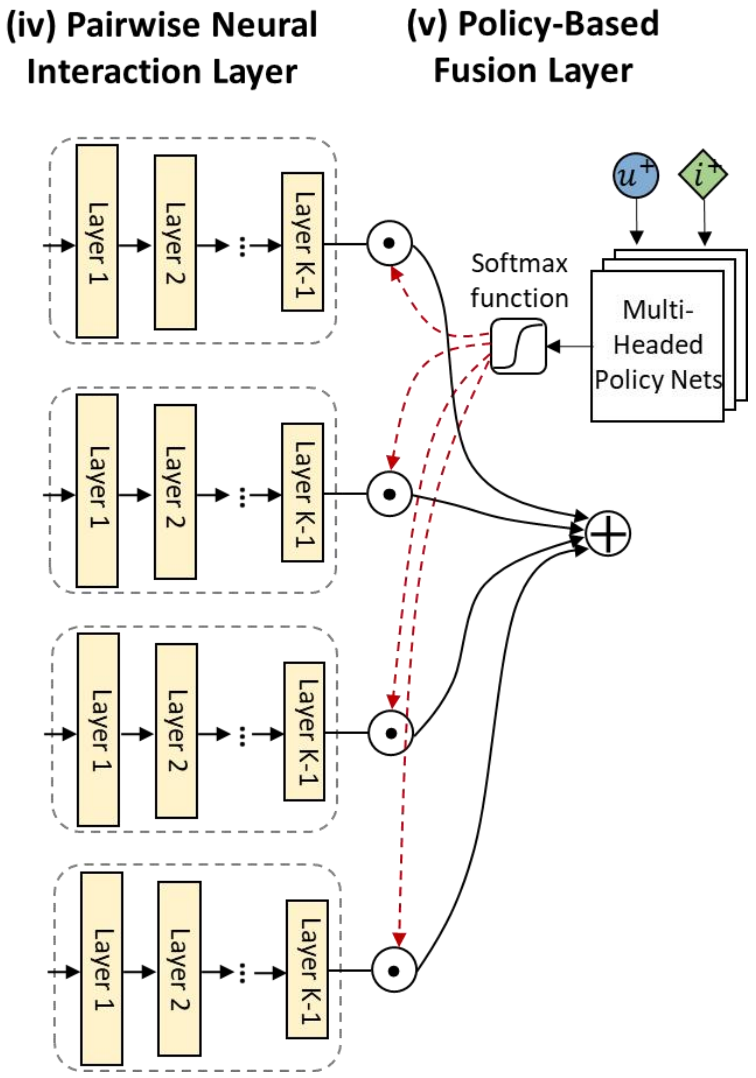
Multi-layer fully-connected neural networks

$$\mathbf{s}_a = \phi^a_K(\cdots \phi^a_2(\phi^a_1(\mathbf{z}_0[a])))$$

$$\phi^a_k(\mathbf{z}_{k-1}) = \tanh(\mathbf{W}^a_k \mathbf{z}^a_{k-1} + \mathbf{b}^a_k), k \in [1, K-1]$$

$$\mathbf{z}_0 = [\mathbf{p}^*_\mathbf{u} \oplus \mathbf{q}^*_\mathbf{i}, \mathbf{p}^*_\mathbf{u} \oplus \mathbf{n}^*_\mathbf{i}, \mathbf{m}^*_\mathbf{u} \oplus \mathbf{q}^*_\mathbf{i}, \mathbf{m}^*_\mathbf{u} \oplus \mathbf{n}^*_\mathbf{i}]$$

Policy-Based Fusion Layer

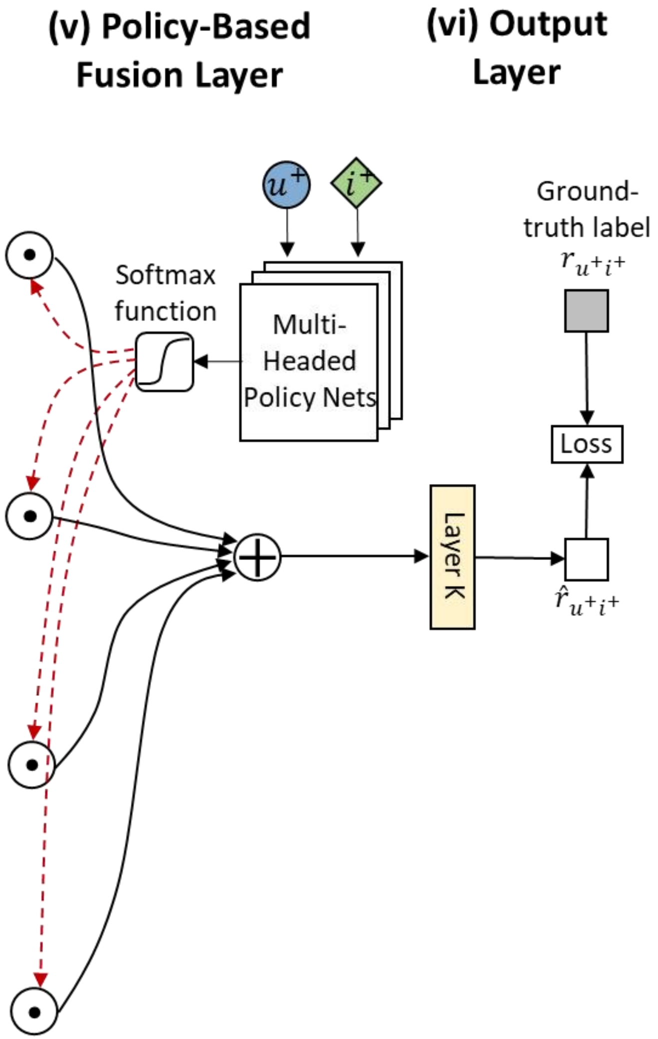


$$e_\gamma = \mathbf{W}_F^2 \tanh(\mathbf{W}_F^1(\mathbf{p}_u || \mathbf{q}_i) + \mathbf{b}_F^1) + \mathbf{b}_F^2.$$

$$p(\gamma|\mathbf{p}_u, \mathbf{q}_i) = \frac{\exp(e_\gamma)}{\sum_{a=1}^{4} \exp(e_a)}$$

$$\mathbf{s} = \mathbb{E}_{\gamma \sim p(\gamma|\mathbf{p}_u, \mathbf{q}_i)}(\mathbf{s}_\gamma) = \sum_{\gamma=1}^{4} (p(\gamma|\mathbf{p}_u, \mathbf{q}_i) \cdot \mathbf{s}_\gamma)$$

Output Layer



A fully-connected layer without activation function

implicit feedback:

$$\mathcal{L}_1 = -\sum_{(u,i)} r_{ui} \log \hat{r}_{ui} + (1 - r_{ui}) \log(1 - \hat{r}_{ui})$$

explicit feedback:

$$\mathcal{L}_1 = \sum_{(u,i)} ||\hat{r}_{ui} - r_{ui}||^2$$

Training

Mini-Batch & Sampling

B (u, i) interaction pairs for each training

$F_U(u)$ $F_I(i)$ Neighbored nodes for user and item

$R_I(v)$ $R_U(j)$ History sequence for user and item

Sample neighbored nodes

Uniformly sample F neighbored nodes for each user and item

Truncate history sequence

Truncate recent C clicked items for each user

Truncate recent C infected users for each item

$F_U(u)$ $F_I(i)$ Size $B \times F$ for one batch

$R_I(v)$ $R_U(j)$ Size $B \times F \times C$ for one batch

Local-Graph Aware Regularizatio

L1 regularization

$$\mathcal{L}_2 = \sum_u (||p_u|| + ||x_u||) + \sum_i (||q_i|| + ||y_i||).$$

$$\downarrow$$

$$\mathcal{L}_2 = \frac{1}{2} \sum_{(u,i)} \left[\|\mathbf{p}_u\| + \|\mathbf{x}_u\| \right] + | \sum_{v \in \Gamma_U(u)} \frac{1}{|F_U(v)|} (\|\mathbf{p}_v\| + \|\mathbf{x}_v\|)$$

$$+ \|\mathbf{q}_i\| + \|\mathbf{y}_i\| + \sum_{j \in \Gamma_I(j)} \frac{1}{|F_I(j)|} (\|\mathbf{q}_j\| + \|\mathbf{y}_j\|)$$

final loss function

$$\mathcal{L} = \mathcal{L}_1 + \lambda \mathcal{L}_2$$

Training

Policy Gradient

l-th policy network $p_l(\gamma|\mathbf{p}_u, \mathbf{q}_i)$

draw $\gamma \sim Multi(p_l(\gamma|\mathbf{p}_U, \mathbf{q}_i))$

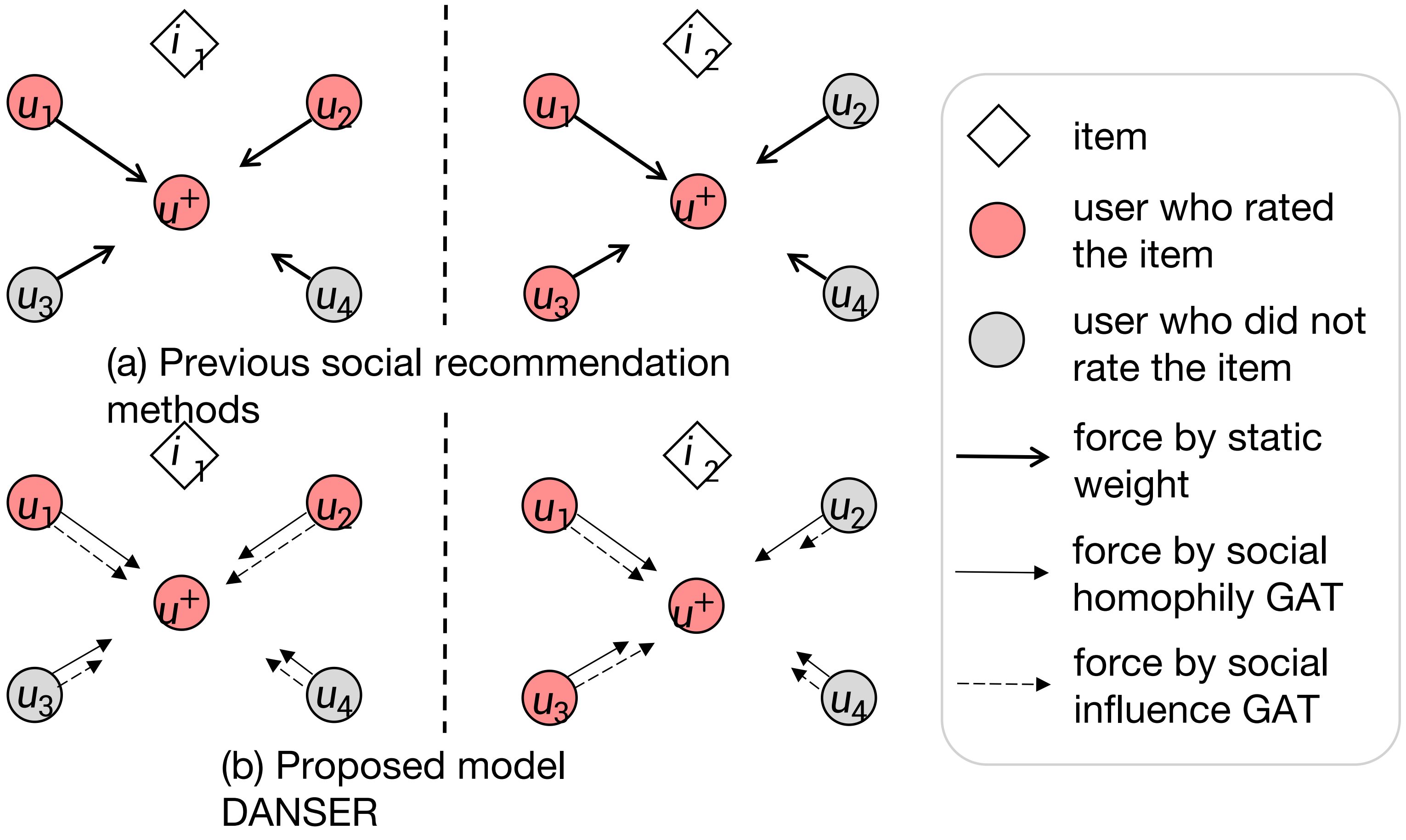
$$\mathcal{R}(p_u, q_i, \gamma) = -\mathcal{L}(\mathbf{s}_\gamma)$$

maximize the expected reward $\mathbb{E}_{\gamma \sim p(\gamma|p_u, q_i)}(\mathcal{R}(p_u, q_i, \gamma))$

calculate gradient $\nabla_\theta \mathbb{E}_{\gamma \sim p_\theta(\gamma|\mathbf{p}_u, \mathbf{q}_i)}(\mathcal{R}(\mathbf{p}_u, \mathbf{q}_i, \gamma))$

$$\simeq \frac{1}{4} \sum_{\gamma} \nabla_\theta \log p_\theta(\gamma|\mathbf{p}_u, \mathbf{q}_i) \mathcal{R}(\mathbf{p}_u, \mathbf{q}_i, \gamma)$$

Justification of Dual GATS



Experiments

Datasets

Epinion (public dataset)

WeChat (commercial dataset)

Table 1: Statistics of three datasets.

Dataset	#users	#items	#interactions	#relationships
Epinions	49,290	139,738	664,824	487,181
WeChat	~200,000	~100,000	~4,000,000	~2,000,000

Experiments

Evaluation Protoc

Data partition	For Epinions, randomly 80% for training and 20% for testing
	For Wechat, sequentially 90% for training, 10% for testing
Evaluation metrics	For Epinions, *MAPE* and *RMSE* for accuracy of 1~5 score prediction
	For Wechat, *Precision@10* and *AUC* for accuracy of 0-1 classification
Implementation	Python with Tensorflow + GTX 1080 GPU with 8G memory

Competitive Method

Collaborative filtering	SVD++	Basic collaborative filtering method
	DELF	**Dual embedding method – IJCAI'18**
Social recommendation	TrustPro	Trust propagation – SIGIR'11
	TrustMF	Trust matrix factorization – IJCAI'13
	TrustSVD	Trust matrix factorization – AAAI'15
	NSCR	**Graph regularization – SIGIR'17**
	SREPS	**Network embedding – AAAI'18**

Experiments

Table1: Comparative results for Epinions and Wechat. For MAE, RMSE, the smaller value is better and vice versa for P@10, AUC.

	Epinions		WeChat	
	MAE	RMSE	P@10	AUC
SVD++ [15]	0.8321	1.0772	0.0653	0.7304
DELF [2]	0.8115	1.0561	<u>0.0752</u>	<u>0.7818</u>
TrustPro [37]	0.9130	1.1124	0.0561	0.6482
TrustMF [36]	0.8214	1.0715	0.0625	0.7005
TrustSVD [10]	0.8144	1.0492	0.0664	0.7325
NSCR [31]	0.8044	1.0425	0.0736	0.7727
SREPS [16]	<u>0.8014</u>	<u>1.0393</u>	0.0725	0.7745
DANSER	**0.7781**	**1.0268**	**0.0823**	**0.8165**
Impv.[1]	2.87%	1.25%	9.33%	4.48%

[1] The improvement compares DANSER with the best competitor (underlined).

Table2: Ablation study of components in proposed method

	Epinions		WeChat	
	MAE	RMSE	P@10	AUC
DualEMB	.7920(1.7%)[1]	1.0363(0.7%)	.0794(3.6%)	.7992(2.2%)
DualGCN	.7840(0.7%)	1.0335(0.4%)	.0814(1.1%)	.8102(0.8%)
userGAT	.7858(0.9%)	1.0364(0.7%)	.0813(1.2%)	.8136(0.4%)
itemGAT	.7919(1.7%)	1.0335(0.4%)	.0813 (1.2%)	.8138(0.3%)
DANSER-w	.8191(4.9%)	1.0659(3.4%)	.0820(0.4%)	.8151(0.2%)
DANSER-m	.8211(5.2%)	1.0681(3.6%)	.0815(1.0%)	.8144(0.3%)
DANSER-a	.8232(5.4%)	1.0710(3.9%)	.0814 (1.1%)	.8140(0.3%)
DANSER-c	.8091(3.8%)	1.0659(3.4%)	.0809(1.7%)	.8118(0.6%)
DANSER	**0.7787**	**1.0292**	**0.0823**	**0.8165**

[1] The ratio indicates the impv. comparing DANSER with corresponding variant.

Case Study

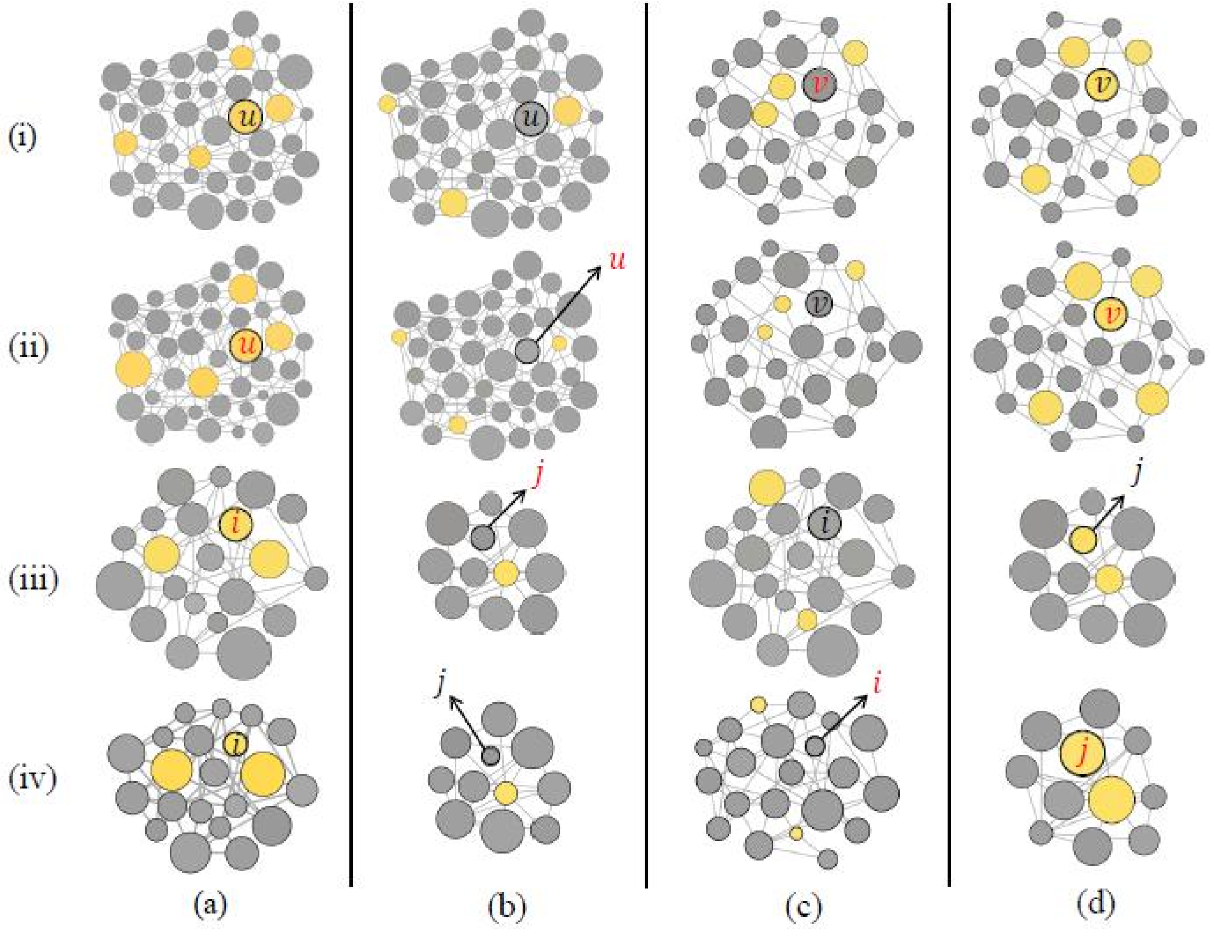


Figure3: Visualization of the four GAT's weights for a case study

Conclusions

Our contributions can be summarized as follows

i) **General Aspects**: 4 different social effects in recommendation (heterogeneous and dynamic)

ii) **Novel Methodologies**: GAT + policy fusion

iii) **Multifaceted Experiments**: comparison + ablation study + parameter sensitivity + interpretable visualization

Available sources

Poster: Thus, #284

Paper & Codes: https://github.com/echo740/DANSER

Contact us: echo740@sjtu.edu.cn