Graph Neural Network in Recommender Systems

Jiarui Qin

Apex Data & Knowledge Management Lab

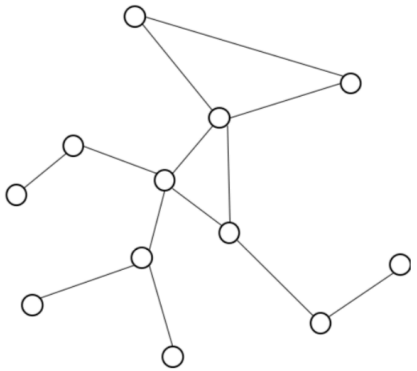
Shanghai Jiao Tong University

Contents

- **Basics of GNN**
  - Graph Convolution Networks (GCN)
  - Graph Attention Networks (GAT)
- **GNN in Social Recommendation**
- **GNN in Knowledge Graph Recommendation**
- **GNN in User-Item Bipartite Graph**

Graph Convolution Networks (GCN)

- Spectral method
- Non-spectral method:
  - Represent a node using its neighbors by an iterative manner
  - GCN
    - Aggregator: $\mathbf{h}_{\mathcal{N}_v}^t = \mathbf{h}_v^{t-1} + \sum_{k=1}^{\mathcal{N}_v} \mathbf{h}_k^{t-1}$
    - Updater: $\mathbf{h}_v^t = \sigma(\mathbf{h}_{\mathcal{N}_v}^t \mathbf{W}_L^{\mathcal{N}_v})$
  - GraphSAGE
    - Aggregator: $\mathbf{h}_{\mathcal{N}_v}^t = \mathrm{AGGREGATE}_t\left(\{\mathbf{h}_u^{t-1}, \forall u \in \mathcal{N}_v\}\right)$
    - Updater: $\mathbf{h}_v^t = \sigma\left(\mathbf{W}^t \cdot [\mathbf{h}_v^{t-1} \| \mathbf{h}_{\mathcal{N}_v}^t]\right)$
    - AGGREGATE function: avg/max pooling, LSTM



J.Zhou et al. Graph Neural Networks: A Review of Methods and Applications
D.K.Duvenaud et al. Convolutional networks on graphs for learning molecular fingerprints. NIPS 2015
W.L.Hamilton et al. Inductive representation learning on large graphs. NIPS 2017

Graph Attention Networks (GAT)

- Aggregator:

$$\alpha_{vk} = \frac{\exp\left(\text{LeakyReLU}\left(\mathbf{a}^T[\mathbf{W}\mathbf{h}_v \| \mathbf{W}\mathbf{h}_k]\right)\right)}{\sum_{j\in\mathcal{N}_v}\exp(\text{LeakyReLU}(\mathbf{a}^T[\mathbf{W}\mathbf{h}_v \| \mathbf{W}\mathbf{h}_j]))}$$

$$\mathbf{h}^t_{\mathcal{N}_v} = \sigma\left(\sum_{k\in\mathcal{N}_v}\alpha_{vk}\mathbf{W}\mathbf{h}_k\right)$$

Multi-head concatenation:

$$\mathbf{h}^t_{\mathcal{N}_v} = \Big\|_{m=1}^{M}\sigma\left(\sum_{k\in\mathcal{N}_v}\alpha^m_{vk}\mathbf{W}^m\mathbf{h}_k\right)$$

Multi-head average:

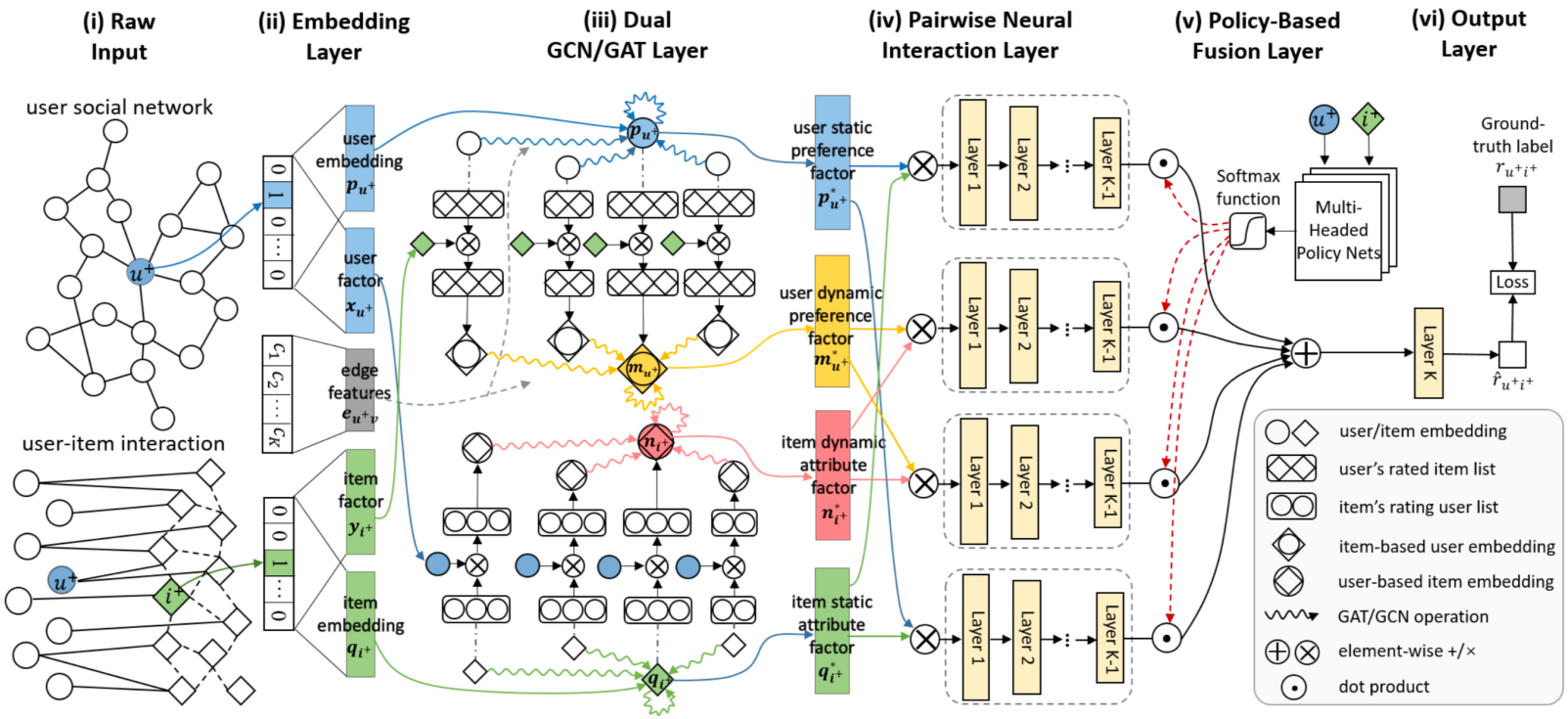
$$\mathbf{h}^t_{\mathcal{N}_v} = \sigma\left(\frac{1}{M}\sum_{m=1}^{M}\sum_{k\in\mathcal{N}_v}\alpha^m_{vk}\mathbf{W}^m\mathbf{h}_k\right)$$

- Updater:

$$\mathbf{h}^t_v = \mathbf{h}^t_{\mathcal{N}_v}$$

P. Velickovic et al. Graph attention networks. ICLR 2018

GNN in Social Recommendation (DANSER)



- Three Graphs:
  - user-item interaction;
  - user social network;
  - item-item network (attract same user)

Q. Wu et al. Dual Graph Attention Networks for Deep Latent Representation of Multifaceted Social Effects in Recommender Systems. WWW 2019

GNN in Social Recommendation (DANSER)

- GAT to capture social/item-item homophily

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

$$\alpha_{uv}^P = \frac{attn_U(\mathbf{W}_P\mathbf{p}_u, \mathbf{W}_P\mathbf{p}_v, \mathbf{W}_E\mathbf{e}_{uv})}{\sum_{w\in\Gamma_U(u)} attn_U(\mathbf{W}_P\mathbf{p}_u, \mathbf{W}_P\mathbf{p}_w, \mathbf{W}_E\mathbf{e}_{uv})}, v \in \Gamma_U(u)$$

- GAT to capture social/item-item influence

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

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

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

GNN in Social Recommendation (DANSER)

- Pairwise Neural Interaction Layer

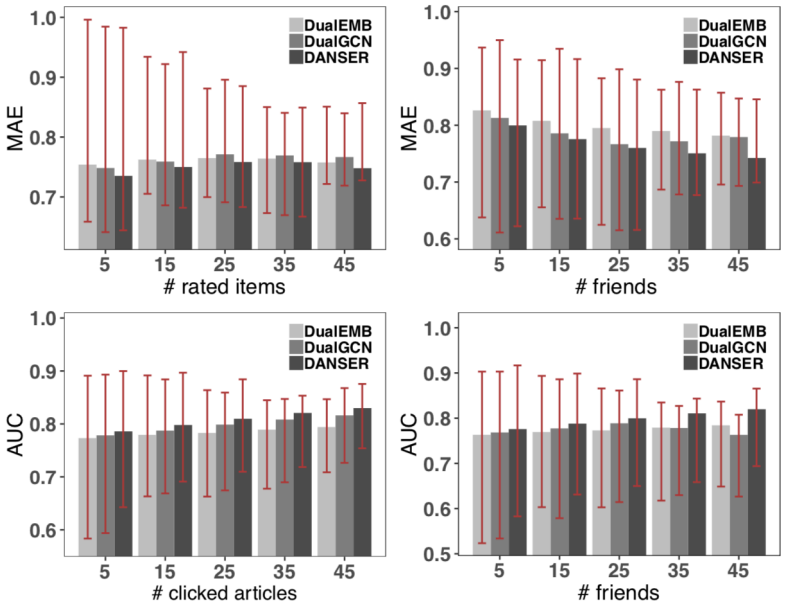
$$\mathbf{s}_a = \phi_K^a(\cdots \phi_2^a(\phi_1^a(\mathbf{z}_0[a]))),$$
$$\phi_k^a(\mathbf{z}_{k-1}) = tanh(\mathbf{W}_k^a \mathbf{z}_{k-1}^a + \mathbf{b}_k^a), k \in [1, K-1],$$
$$\mathbf{z}_0 = [\mathbf{p}_u^* \oplus \mathbf{q}_i^*, \mathbf{p}_u^* \oplus \mathbf{n}_i^*, \mathbf{m}_u^* \oplus \mathbf{q}_i^*, \mathbf{m}_u^* \oplus \mathbf{n}_i^*],$$

- Policy-Based Fusion Layer

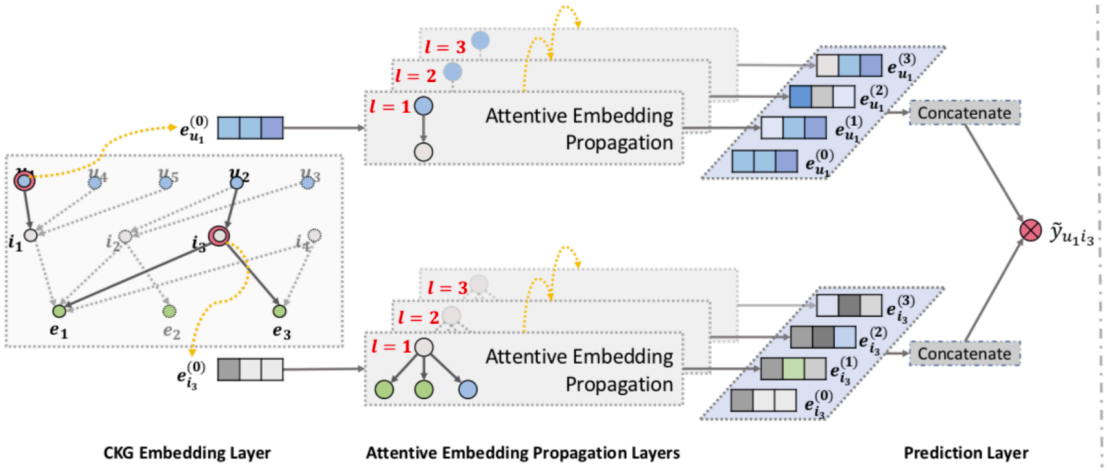
$$e_\gamma = \mathbf{W}_F^2 \tanh(\mathbf{W}_F^1(\mathbf{p}_u || \mathbf{q}_i) + \mathbf{b}_F^1) + \mathbf{b}_F^2.$$
$$p(\gamma|\mathbf{p}_u, \mathbf{q}_i) = \frac{\exp(e_\gamma)}{\sum_{a=1}^4 \exp(e_a)}.$$
$$\mathbf{s} = \mathbb{E}_{\gamma \sim p(\gamma|\mathbf{p}_u, \mathbf{q}_i)}(\mathbf{s}_\gamma) = \sum_{\gamma=1}^4 p(\gamma|\mathbf{p}_u, \mathbf{q}_i) \cdot \mathbf{s}_\gamma.$$

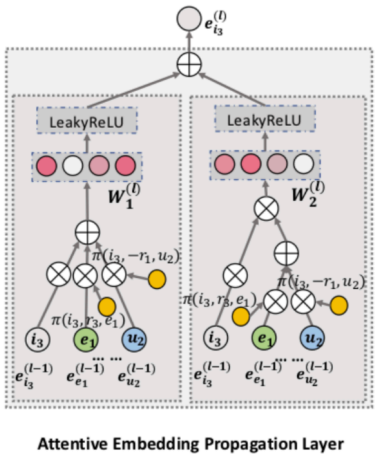
Experiments

	Epinions		WeChat	
	MAE	RMSE	P@10	AUC
SVD++ [15]	0.8321	1.0772	0.0653	0.7304
DELF [2]	0.8115	1.0561	0.0752	0.7818
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]	0.8014	1.0393	0.0725	0.7745
DANSER	**0.7781**	**1.0268**	**0.0823**	**0.8165**
Impv.[1]	2.87%	1.25%	9.33%	4.48%



GNN in Knowledge Graph Recommendation (KGAT)





- Collaborative Knowledge Graph: Add the user-item bipartite graph to the original KG
- Embedding Layer:
  - TransR: $g(h, r, t) = \|\mathbf{W}_r\mathbf{e}_h + \mathbf{e}_r - \mathbf{W}_r\mathbf{e}_t\|_2^2$ $\quad \mathcal{L}_{\mathrm{KG}} = \sum_{(h,r,t,t')\in\mathcal{T}} -\ln\sigma\Big(g(h,r,t') - g(h,r,t)\Big)$
- Attentive Embedding Propagation Layers
  - 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: $f_{\mathrm{GCN}} = \mathrm{LeakyReLU}\Big(\mathbf{W}(\mathbf{e}_h + \mathbf{e}_{\mathcal{N}_h})\Big)$

X. Wang et al. KGAT: Knowledge Graph Attention Network for Recommendation . KDD 2019

GNN in Knowledge Graph Recommendation (KGAT)

- High-order Propagation:

$$\mathbf{e}_h^{(l)} = f\left(\mathbf{e}_h^{(l-1)}, \mathbf{e}_{\mathcal{N}_h}^{(l-1)}\right),$$

$$\mathbf{e}_{\mathcal{N}_h}^{(l-1)} = \sum_{(h,r,t)\in\mathcal{N}_h} \pi(h,r,t)\mathbf{e}_t^{(l-1)},$$

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

- Model Prediction: $\hat{y}(u,i) = {\mathbf{e}_u^*}^\top \mathbf{e}_i^*$.
- Optimization: BPR ranking loss + TransR loss, optimize the two loss alternatively

$$\mathcal{L}_{\mathrm{CF}} = \sum_{(u,i,j)\in O} -\ln\sigma\left(\hat{y}(u,i) - \hat{y}(u,j)\right)$$

$$\mathcal{L}_{\mathrm{KGAT}} = \mathcal{L}_{\mathrm{KG}} + \mathcal{L}_{\mathrm{CF}} + \lambda\|\Theta\|_2^2$$

Experiments

Table 2: Overall Performance Comparison.

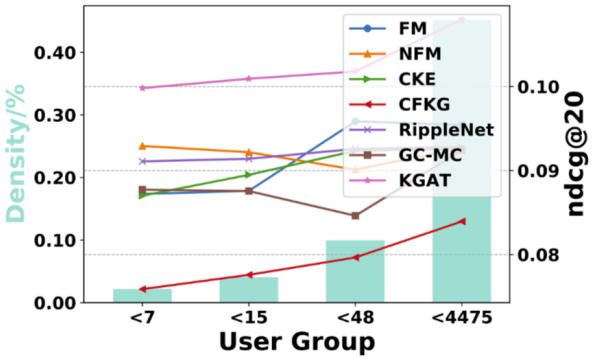
	Amazon-Book		Last-FM		Yelp2018	
	recall	ndcg	recall	ndcg	recall	ndcg
FM	0.1345	0.0886	0.0778	0.1181	0.0627	0.0768
NFM	**0.1366**	0.0913	**0.0829**	0.1214	0.0660	0.0810
CKE	0.1343	0.0885	0.0736	0.1184	0.0657	0.0805
CFKG	0.1142	0.0770	0.0723	0.1143	0.0522	0.0644
MCRec	0.1113	0.0783	-	-	-	-
RippleNet	0.1336	0.0910	0.0791	0.1238	**0.0664**	**0.0822**
GC-MC	0.1316	0.0874	0.0818	**0.1253**	0.0659	0.0790
KGAT	**0.1489***	**0.1006***	**0.0870***	**0.1325***	**0.0712***	**0.0867***
%Improv.	8.95%	10.05%	4.93%	5.77%	7.18%	5.54%

Table 3: Effect of embedding propagation layer numbers (L).

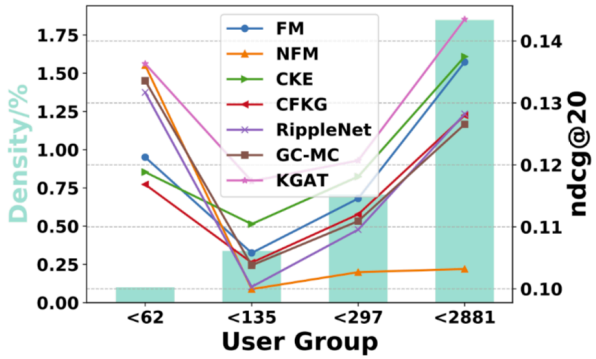
	Amazon-Book		Last-FM		Yelp2018	
	recall	ndcg	recall	ndcg	recall	ndcg
KGAT-1	0.1393	0.0948	0.0834	0.1286	0.0693	0.0848
KGAT-2	0.1464	0.1002	0.0863	0.1318	0.0714	**0.0872**
KGAT-3	0.1489	0.1006	0.0870	0.1325	0.0712	0.0867
KGAT-4	**0.1503**	**0.1015**	**0.0871**	**0.1329**	**0.0722**	0.0871

Experiments

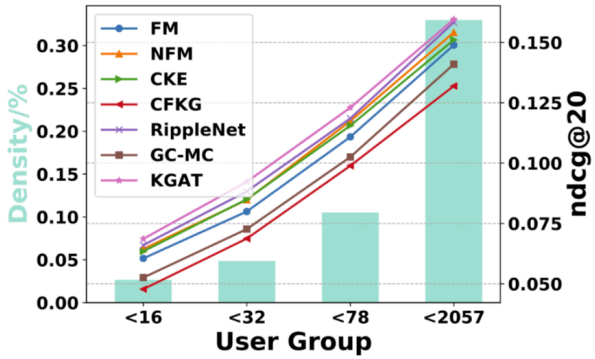
Better Performance in Sparse Recommendation



(a) ndcg on Amazon-Book



(b) ndcg on Last-FM

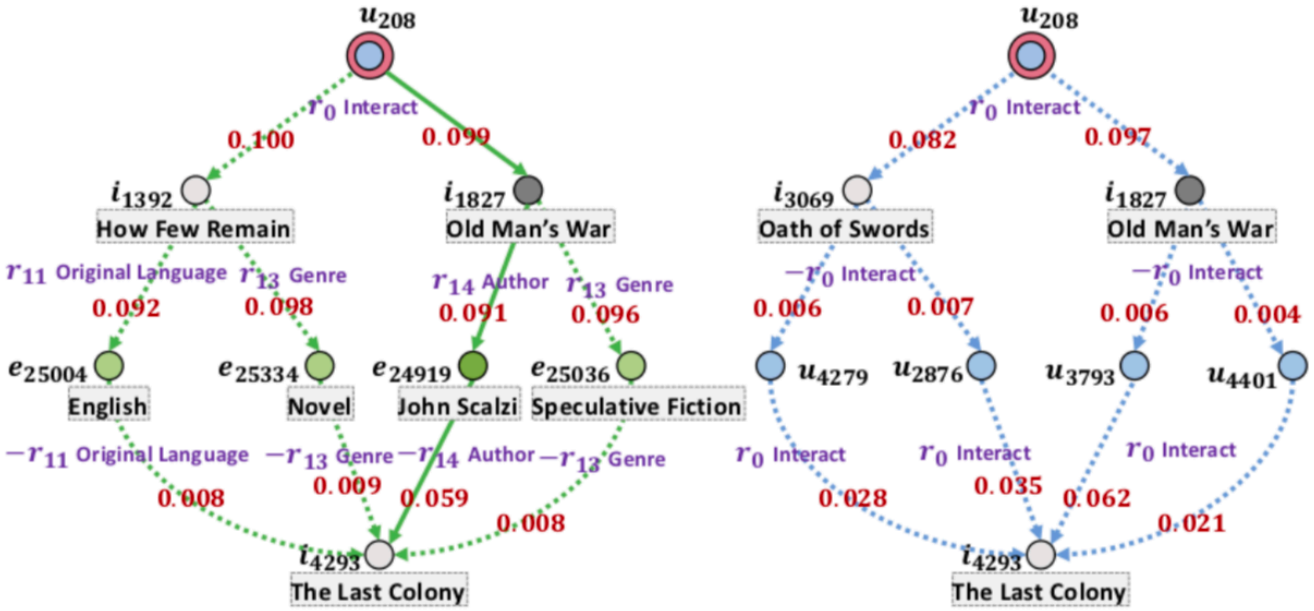


(c) ndcg on Yelp2018

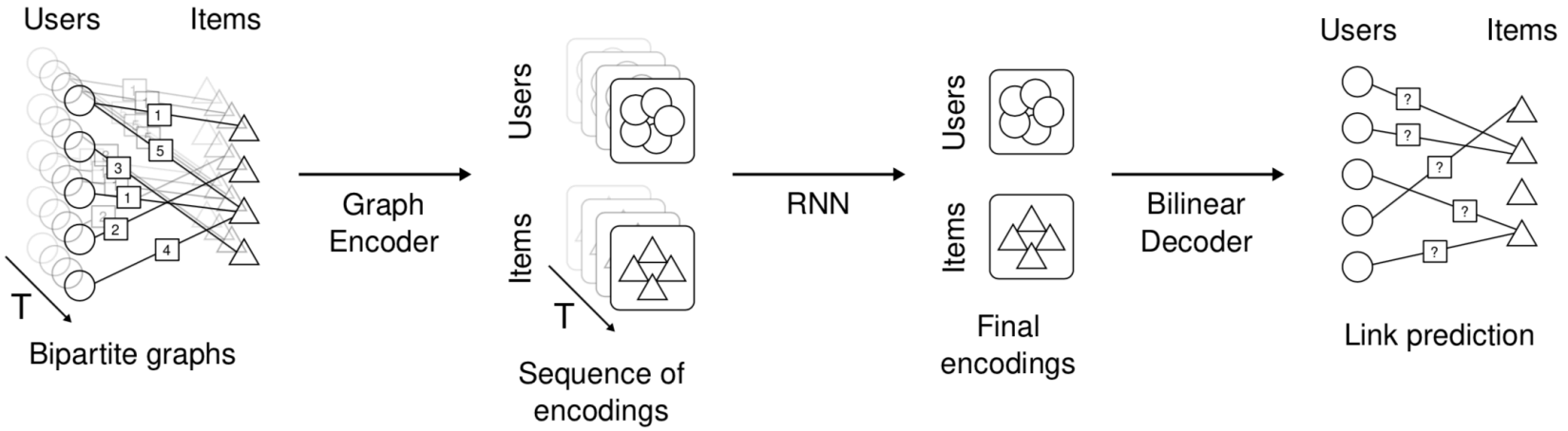
Figure 3: Performance comparison over the sparsity distribution of user groups on different datasets. The background histograms indicate the density of each user group; meanwhile, the lines demonstrate the performance *w.r.t.* ndcg@20.

Experiments

Case study: Explainability



GNN in User-Item Bipartite Graph (GCMC)



- Regard the sequential recommendation as a dynamic graph link prediction problem
- Use GCN as graph information aggregator:

$$\mathbf{h}_{u_i} = \xi\left(\text{accum}_{r\in\mathcal{R}}\left\{\sum_{j\in\mathcal{N}_{i,r}}\frac{1}{c_{ij}}W_r\mathbf{x}_{v_j}\right\}\right),$$

$$\mathbf{z}_{u_i} = \xi(W\mathbf{h}_{u_i}),$$

S.G. Fadel et al. Link Prediction in Dynamic Graphs for Recommendation. NIPS 2018

GNN in User-Item Bipartite Graph (SCoRe)

- Traditional CF models:
  - No sequential dynamics
  - Collaborative information is used in an implicit manner
- Sequential recommendation models:
  - Just use user-side temporal dynamics
  - Ignorance of collaborative information
- Sequential incremental graph models:
  - Lack of high-order propagation
  - Graph aggregator is simple and not effective enough

- Sequential Collaborative Recommender:
  - Spatial: uses collaborative information explicitly by exploiting high-order relations
  - Temporal: models both user- and item-side temporal dynamics

GNN in User-Item Bipartite Graph (SCoRe)

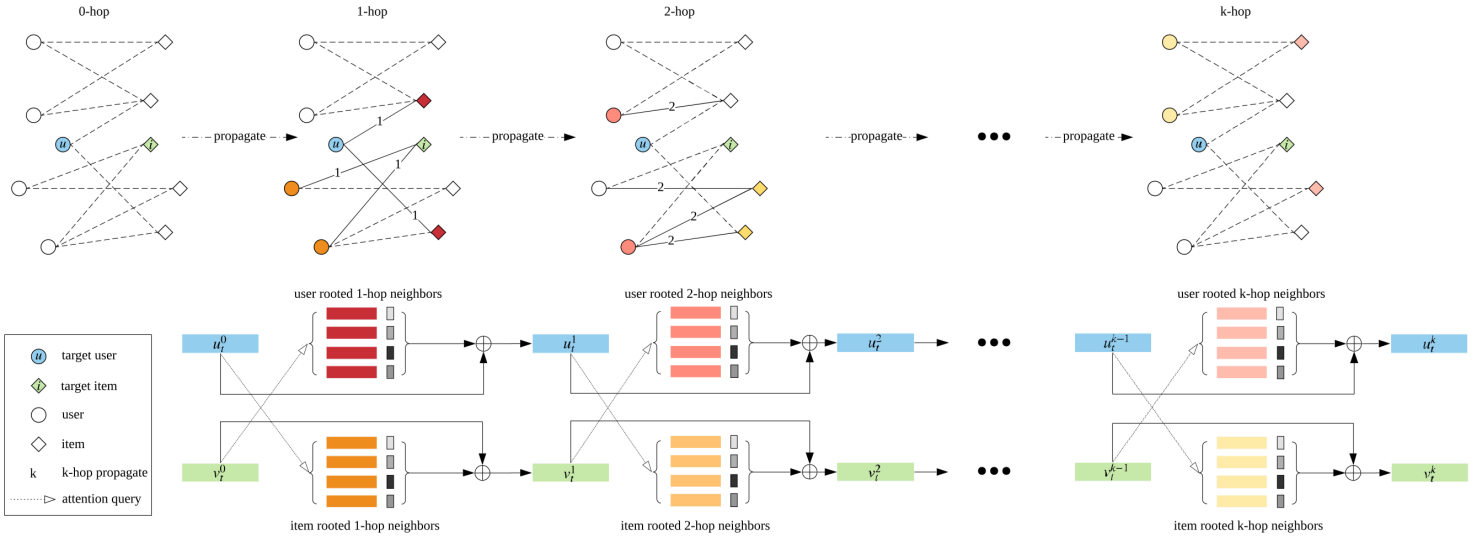


Figure 2: Multi-hop Co-Attention Graph Network calculation process.

- Co-Attention Graph Network to aggregate high-order spatial information:

$$\boldsymbol{u}_t^k = \boldsymbol{u}_t^{k-1} + \sum_i \alpha_{ui}^k e_i^k$$

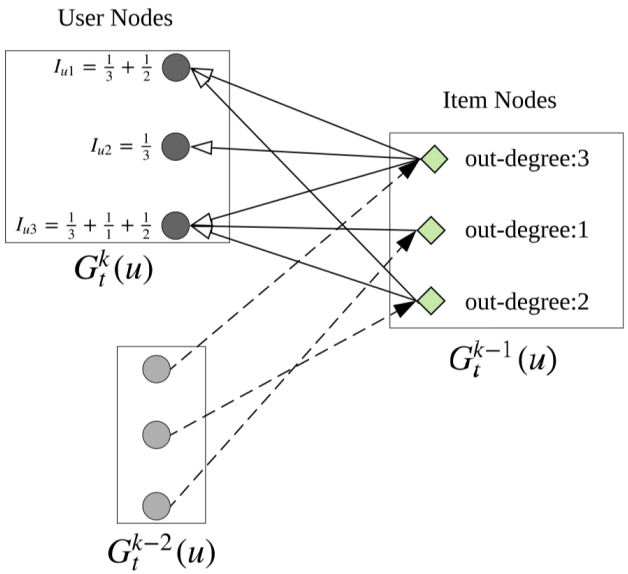
$$\alpha_{ui} = \frac{exp(LeakyReLU(\boldsymbol{a}^T [\boldsymbol{W}\boldsymbol{e}_i^k || \boldsymbol{W}\boldsymbol{v}_t^{k-1}]))}{\sum_{\boldsymbol{e}_j^k \in G^k(u)} exp(LeakyReLU(\boldsymbol{a}^T [\boldsymbol{W}\boldsymbol{e}_j^k || \boldsymbol{W}\boldsymbol{v}_t^{k-1}]))}$$

$$\boldsymbol{v}_t^k = \boldsymbol{v}_t^{k-1} + \sum_i \alpha_{vi}^k e_i^k$$

$$\alpha_{vi} = \frac{exp(LeakyReLU(\boldsymbol{a}^T [\boldsymbol{W}\boldsymbol{e}_i^k || \boldsymbol{W}\boldsymbol{u}_t^{k-1}]))}{\sum_{\boldsymbol{e}_j^k \in G^k(v)} exp(LeakyReLU(\boldsymbol{a}^T [\boldsymbol{W}\boldsymbol{e}_j^k || \boldsymbol{W}\boldsymbol{u}_t^{k-1}]))}$$

GNN in User-Item Bipartite Graph (SCoRe)

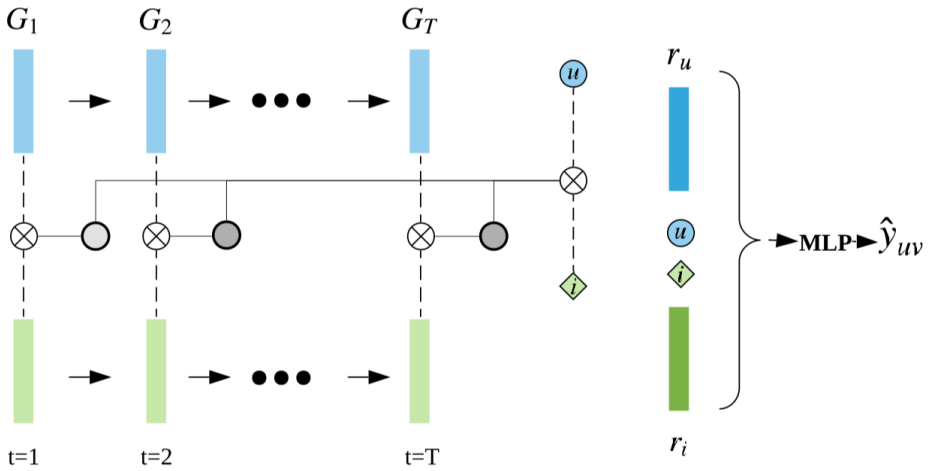
- Importance Sampling Strategy to reduce noise:



$$I_p = \sum_{q \in pre(p)} \frac{1}{out\text{-}degree(q)} \qquad Pr_p = \frac{\exp(I_p)}{\sum_{n \in G_t^k(u)} \exp(I_n)} .$$

GNN in User-Item Bipartite Graph (SCoRe)

- Temporal Interactive Dual Sequence Modeling:



$$relation_t = \sigma(\boldsymbol{Q}_2(\sigma(\boldsymbol{Q}_1((\boldsymbol{h}_t^u \odot \boldsymbol{h}_t^v)||(\boldsymbol{u} \odot \boldsymbol{v})))))$$

$$\beta_t = \frac{\exp(relation_t)}{\sum_{i=1}^{T} \exp(relation_i)}$$

$$\boldsymbol{r}_u = \sum_{t=1}^{T} \beta_t \boldsymbol{h}_t^u, \ \boldsymbol{r}_v = \sum_{t=1}^{T} \beta_t \boldsymbol{h}_t^v$$

Experiments

Dataset	Metric	Group 1		Group 2			Group 3		
		SVD++	DELF	GRU4Rec	Caser	SASRec	RRN	GCMC	SCoRe
CCMR	HR@1	0.2378	0.6536	0.6703	0.6777	0.6979	0.6633	0.6591	**0.7001**
	HR@5	0.4288	0.9087	0.8911	0.8907	0.9047	0.903	0.9063	**0.9118**
	HR@10	0.5462	0.9586	0.9268	0.9184	0.9361	0.9469	0.9508	**0.9583**
	NDCG@5	0.3356	0.7919	0.7917	0.7954	0.8123	0.7957	0.7946	**0.8161**
	NDCG@10	0.3735	0.8084	0.8034	0.8045	0.8226	0.8101	0.8092	**0.8281**
	MRR	0.3375	0.7617	0.7662	0.7699	0.7878	0.7681	0.7652	**0.7913**
Tmall	HR@1	0.1229	0.2001	0.1979	0.2029	0.3312	0.4177	0.4254	**0.4764**
	HR@5	0.3098	0.3765	0.3818	0.4037	0.6307	0.6452	0.6449	**0.6806**
	HR@10	0.3438	0.4573	0.4713	0.4925	0.7072	0.7449	0.7512	**0.7632**
	NDCG@5	0.2112	0.2975	0.2947	0.3089	0.4935	0.5368	0.5403	**0.5842**
	NDCG@10	0.2342	0.3130	0.3236	0.3376	0.5184	0.5691	0.5747	**0.6109**
	MRR	0.1373	0.2990	0.2950	0.3058	0.4676	0.5257	0.5314	**0.5734**
Taobao	HR@1	0.0705	0.1299	0.1117	0.1292	0.1897	0.2138	0.2164	**0.2431**
	HR@5	0.1539	0.2999	0.3001	0.3022	0.3871	0.4695	0.4672	**0.4991**
	HR@10	0.2063	0.4656	0.4395	0.4539	0.5331	0.6213	0.6156	**0.6467**
	NDCG@5	0.0998	0.1719	0.1667	0.1724	0.2512	0.3447	0.3449	**0.3590**
	NDCG@10	0.1082	0.2008	0.1995	0.2013	0.2788	0.3937	0.3929	**0.4112**
	MRR	0.0935	0.1203	0.1119	0.1232	0.2198	0.3402	0.3411	**0.3786**

Experiments

Table 5: Performance comparison on different size of *user/item-rooted interaction set.*

Dataset	Metric	Size of Interaction Set			
		5	10	15	20
CCMR	HR@10	**0.9583**	0.9555	0.9498	0.9423
	NDCG@10	**0.8281**	0.8239	0.8232	0.8221
	MRR	**0.7913**	0.7909	0.7895	0.7829
Tmall	HR@10	0.7589	**0.7632**	0.7619	0.7607
	NDCG@10	0.6091	**0.6109**	0.6087	0.6052
	MRR	0.5658	**0.5734**	0.5698	0.5611
Taobao	HR@10	0.6288	**0.6467**	0.6319	0.6299
	NDCG@10	0.4068	**0.4112**	0.4081	0.4027
	MRR	0.3679	**0.3786**	0.3673	0.3648

Table 6: Performance comparison of ablation study

Dataset	Metric	models				
		RIA	Single-hop	GAT	RS	SCoRe
CCMR	HR@10	0.9509	0.9296	0.9358	0.9452	**0.9583**
	NDCG@10	0.8157	0.8011	0.8104	0.8217	**0.8281**
	MRR	0.7741	0.7621	0.7724	0.7841	**0.7913**
Tmall	HR@10	0.7620	0.7632	0.7287	0.7431	**0.7632**
	NDCG@10	0.5893	0.6029	0.5866	0.6032	**0.6109**
	MRR	0.5567	0.5633	0.5543	0.5662	**0.5734**
Taobao	HR@10	0.6216	0.6172	0.6201	0.6212	**0.6467**
	NDCG@10	0.4001	0.3991	0.4017	0.4015	**0.4112**
	MRR	0.3575	0.3456	0.3498	0.3523	**0.3786**

Experiments