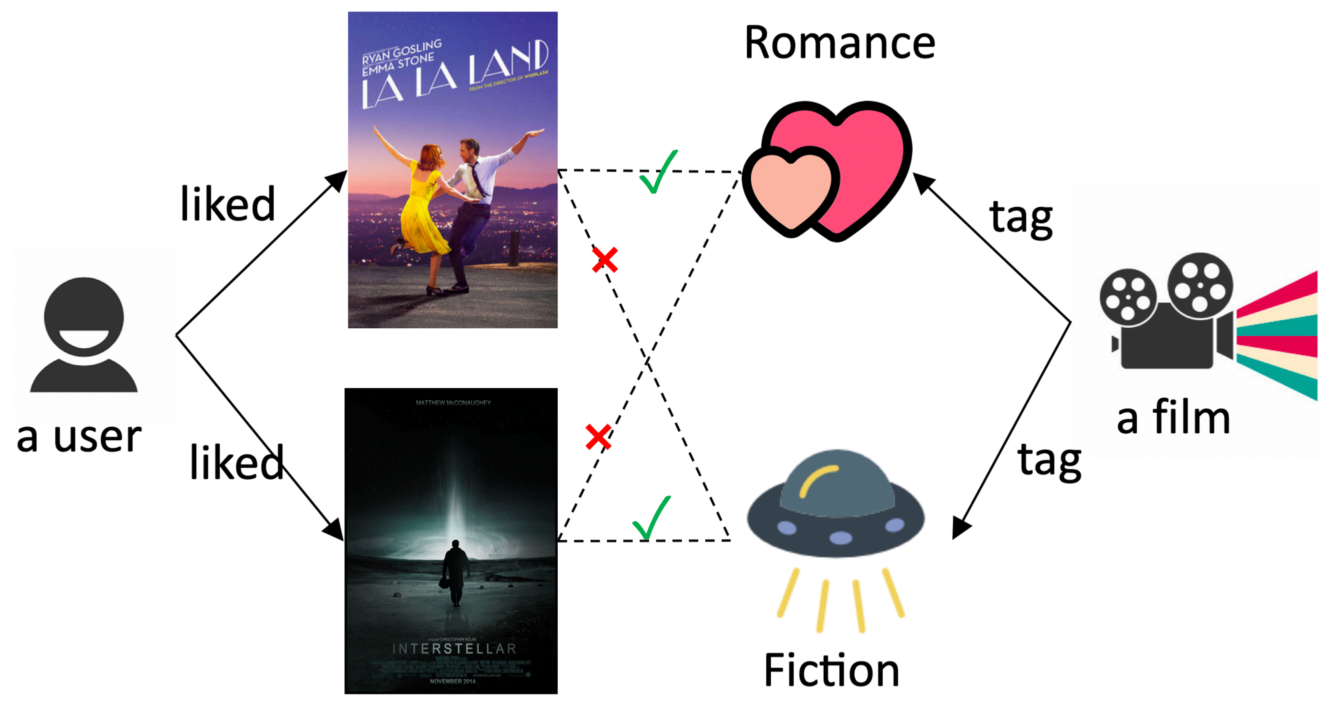
An End-to-End Neighborhood-based Interaction Model for Knowledge-enhanced Recmmendation

Yanru Qu*[1], Ting Bai*[2], Weinan Zhang[1], Jianyun Nie[3], Jian Tang[4]
[1]Shanghai Jiao Tong University, [2]Beijing University of Posts and Telecommunications, [3]Université de Montréal, [4]Mila-Quebec Institute for Learning Algorithms

Background & Motivation



Background: Recommender systems are challenged by **data sparsity** and **cold start** problems. Graph-based models facilitate recommendation with rich structural information.

Goal: Graph-based methods highly rely on estimating relations among nodes. When the relations are unknown, a system is expected to distinguish useful patterns from noise.

Challenges: There exists an "early summarization" problem in most graph-based methods, which mixes all neighborhood information together, and restricts model capacity in exploring the meticulous and valuable neighborhood structures!

Neighborhood Interaction

Graph-based model usually has an "agg" module to aggregate neighbor nodes into user/item representations, and an embedding matching layer to produce output, (a).

(a)
$$\mathbf{u} = \text{agg}(N_u)$$
$$\mathbf{v} = \text{agg}(N_v)$$
$$\hat{y}_{u,v} = \sigma(\langle \mathbf{u}, \mathbf{v} \rangle)$$

(b)
$$\text{Average: } \mathbf{u} = \frac{1}{|N_u|} \sum_{i \in N_u} \mathbf{x}_i$$
$$\text{Attention: } \alpha_{u,i} = \text{softmax}_i(\mathbf{w}^\top [\mathbf{x}_u, \mathbf{x}_i] + b)$$
$$\mathbf{u} = \sum_{i \in N_u} \alpha_{u,i} \mathbf{x}_i$$

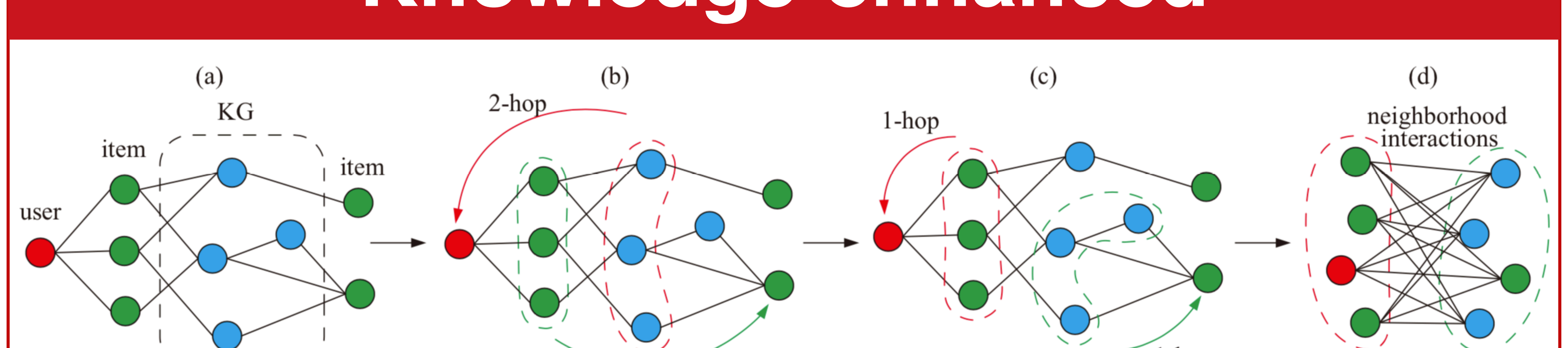
Typically, "agg" can be "average" or "attention", (b). Expanding the "agg" functions, these models have a general form, i.e., weighted sum of all pairs of nodes, (c). (detailed derivation in Sec. 2.1.3).

(c)
$$\hat{y} = \mathbf{A} \odot \mathbf{Z}$$
$$\text{s.t. } \sum_{i,j} \mathbf{A}_{i,j} = 1, \mathbf{Z}_{i,j} = \langle \mathbf{h}_i, \mathbf{h}_j \rangle$$

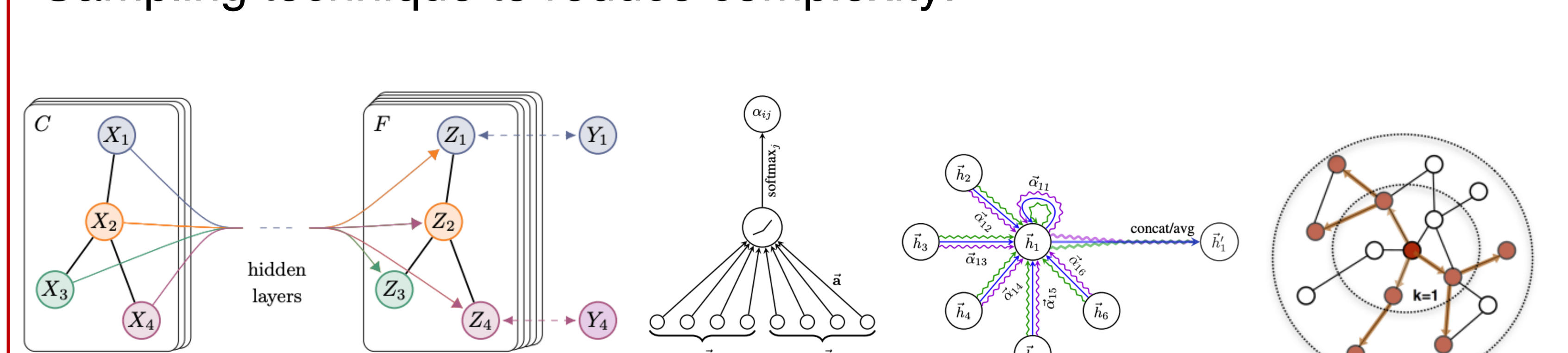
(d)
$$\alpha_{i,j} = \text{softmax}_{i,j}(\mathbf{w}^\top [\mathbf{x}_u, \mathbf{x}_i, \mathbf{x}_v, \mathbf{x}_j] + b)$$
$$\hat{y}_{u,v} = \sum_{i \in N_u} \sum_{j \in N_v} \alpha_{i,j} \langle \mathbf{x}_i, \mathbf{x}_j \rangle$$

We propose **Neighborhood Interaction** model to fully explore the shared form, (d), which can learn different weights for different terms.

Knowledge-enhanced



We build Knowledge-enhanced NI (KNI) model with external knowledge graphs. Besides, KNI Introduce GCN/GAT to learn expressive node embeddings for high-order neighborhood, and utilizes Neighbor Sampling technique to reduce complexity.



Graph Convolution Network   Graph Attention Network   Neighbor Sampling

Experiments

- Datasets: C-Book, Movie-1M, A-Book, Movie-20M

Datasets	C-Book	Movie-1M	A-Book	Movie-20M
# users	17,860	6,036	78,809	59,296
# items	14,967	2,445	32,389	11,895
# interactions	139,746	753,772	1,181,684	9,104,038
# entities	77,881	182,011	265,478	64,067
# relations	10	12	22	38
# triples	71,628	923,718	1,551,554	1,195,391

- Compared models: libFM, Wide&Deep, PER, MCRec, CKE, DKN, PinSage, RippleNet, NI (ours, w/o KG), KNI (ours, w/ KG)
- Task: CTR prediction. Each experiment is repeated 5 times.
- Absolute AUC improvements: 1.1% ~ 8.4%

Model	C-Book		Movie-1M		A-Book		Movie-20M	
	AUC	ACC	AUC	ACC	AUC	ACC	AUC	ACC
libFM	0.6850	0.6390	0.8920	0.8120	0.8300	0.7597	0.9481	0.8805
Wide&Deep	0.7110	0.6230	0.9030	0.8220	0.8401	0.7684	0.9507	0.8831
PER	0.6230	0.5880	0.7120	0.6670	0.7392	0.6939	0.8161	0.7327
MCRec	0.7250	0.6707	0.9127	0.8331	0.8708	0.7930	0.9558	0.8872
CKE	0.6760	0.6422	0.8974	0.8171	0.8572	0.7839	0.9574	0.8940
DKN	0.6488	0.6333	0.8835	0.8070	0.8455	0.7679	0.9473	0.8787
PinSage	0.7102	0.6477	0.9213	0.8443	0.8634	0.7804	0.9597	0.8960
RippleNet	0.7290	0.6630	0.9210	0.8440	0.8736	0.7975	0.9579	0.8942
NI	0.7468	0.6796	0.9401	0.8679	0.9160	0.8362	0.9693	0.9110
KNI	**0.7723***	**0.7063***	**0.9449***	**0.8721***	**0.9238***	**0.8472***	**0.9704***	**0.9120***

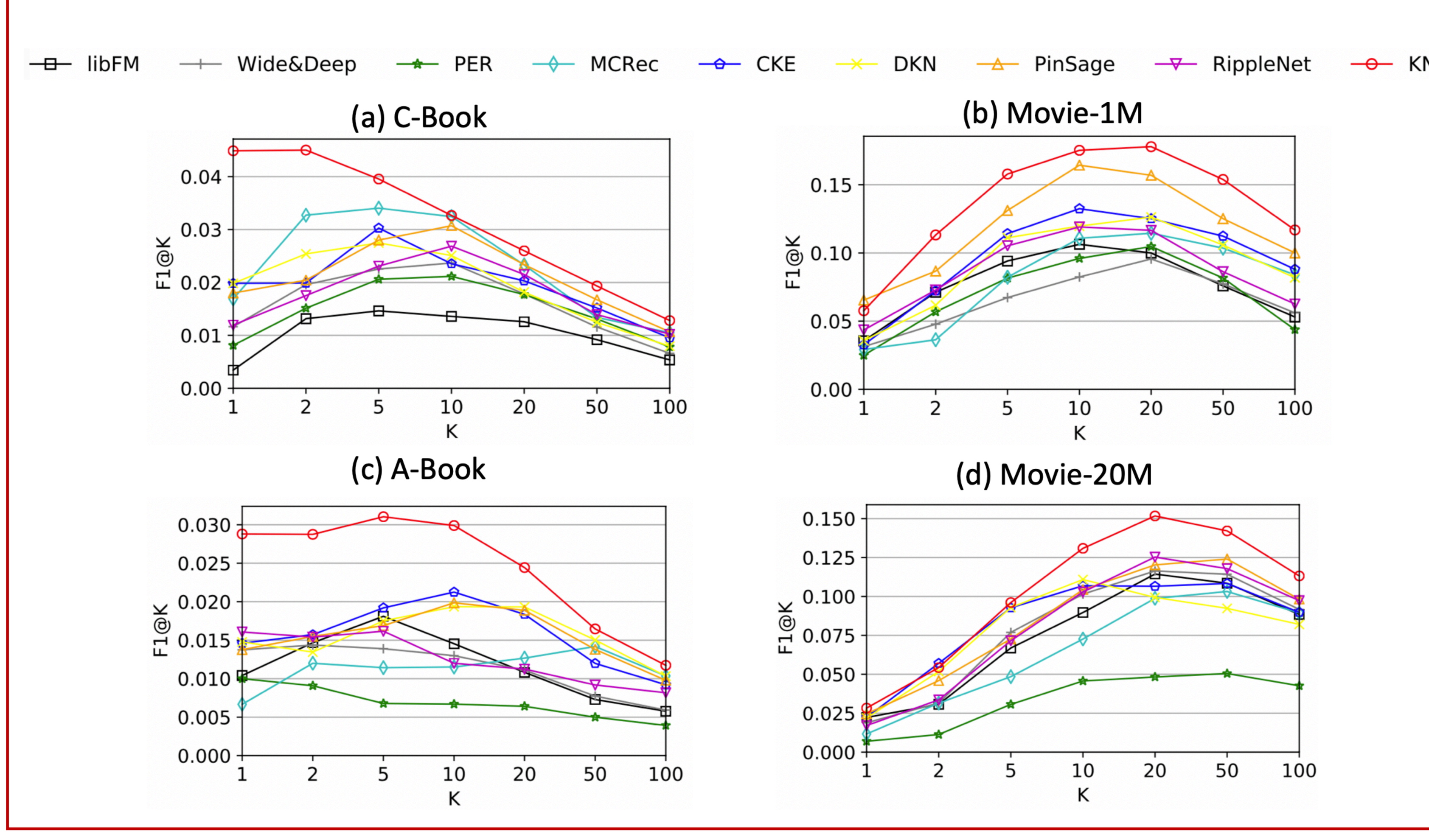
- Data Sparsity. Absolute AUC improvements over best baselines.

Datasets	1-hop	2-hop	3-hop	Sparsity	Improvement
C-Book	1	58	40	99.97%	4.33%
Movie-1M	14	42,227	35,534	97.45%	2.36%
A-Book	5	17,027	49,419	99.98%	5.02%
Movie-20M	17	40,547	14,966	99.35%	1.07%

- Training Speed. Time for 1 epoch.

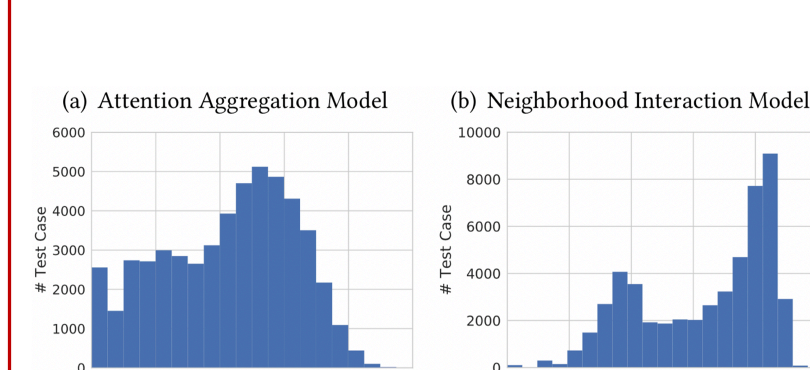
Models	C-Book	Movie-1M	A-Book	Movie-20M
RippleNet	17.75s	66.85s	120.38s	937.92s
KNI	2.05s	11.58s	21.52s	166.72s

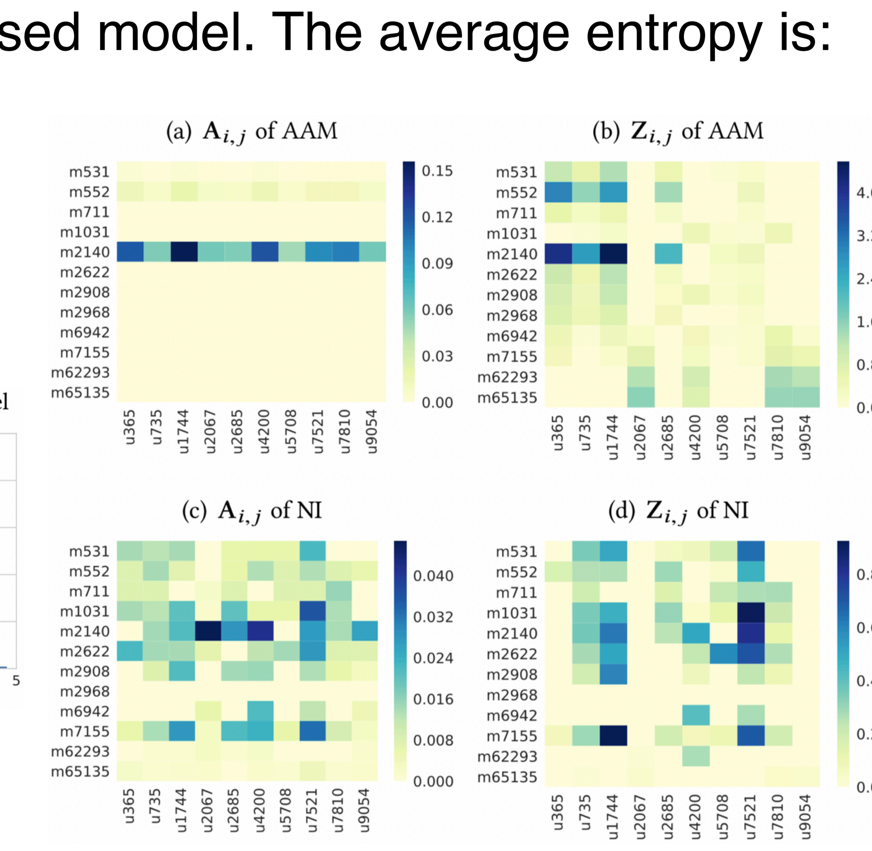
- Task: Top-N Recommendation. Experiments are conducted on best performed models in CTR prediction.



Case Study

In the general form, weight matrix A sums up to 1, we can calculate the entropy of A to measure its information. We remove KGs and compare NI with an attention-based model. The average entropy is: 3.18 (NI) > 2.12 (Attention).





We further visualize the attention matrices, which shows NI model captures more interactive patterns than attention-based model.