北京郵電大學
Beijing University of Posts and Telecommunications

Heterogeneous Graph Neural Network
Concepts, Models and Applications

Prof. Chuan Shi

shichuan@bupt.edu.cn

Beijing University of Posts and Telecommunications

Outline

- √ Basic concepts
  - Network representation
  - Heterogeneous graph
- Models
- Applications
- Conclusion and future work

ML = Representation + Objective + Optimization

Good Representation is Essential for Good Machine Learning

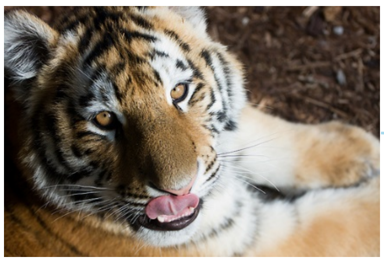
Representation Learning → Machine Learning Systems

Raw Data

Yoshua Bengio. Deep Learning of Representations. AAAI 2013 Tutorial.

Step-by-Step Representation Learning

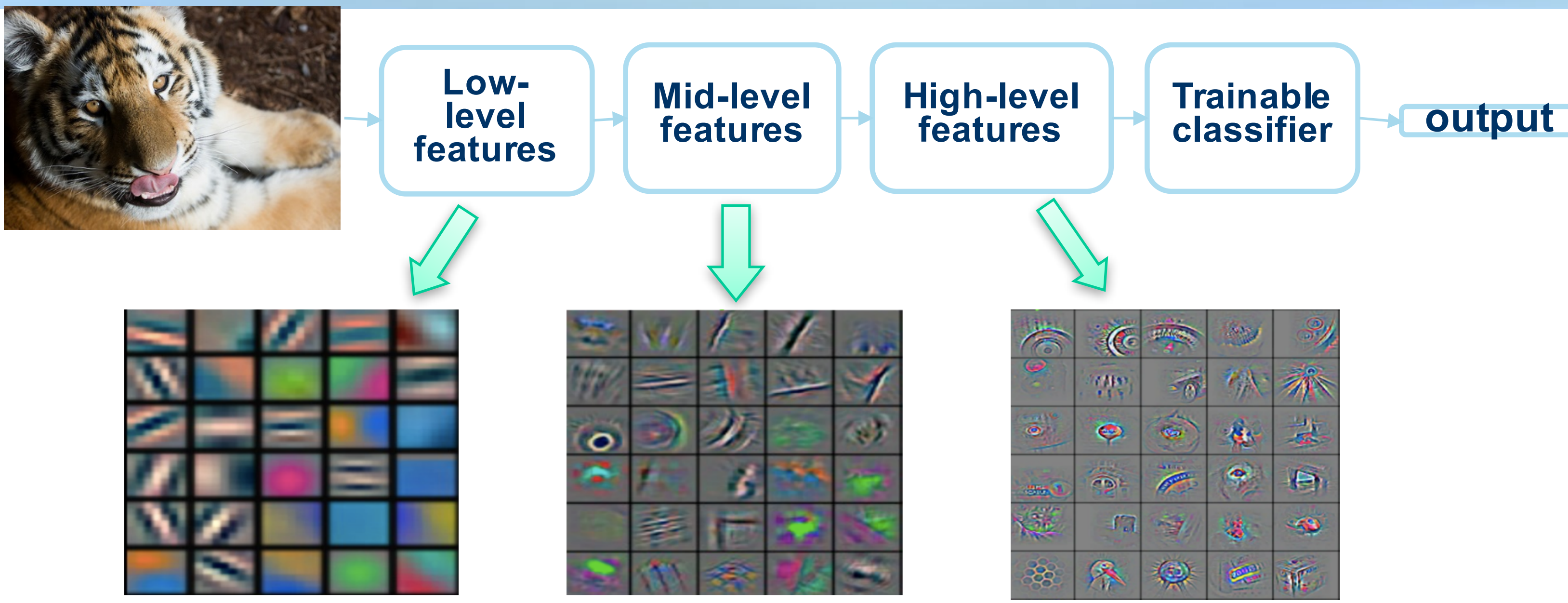
- Traditional representation learning models use hand-crafted features and relatively simple trainable classifier.



hand-crafted feature extractor → "Simple" Trainable Classifier → output

- Has the following limitations:
  - Very Tedious and Costly to develop hand-crafted features
  - Usually highly dependents on one application, and cannot be transferred easily to other applications

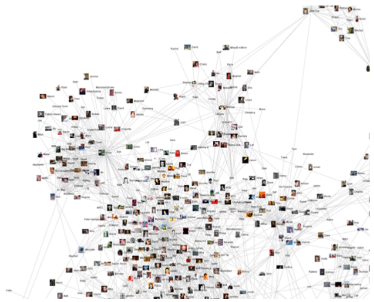
End-to-End Representation Learning



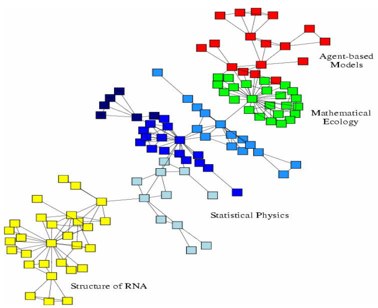
- Hierarchy of representations with increasing level of abstraction.
- Image recognition
  - Pixel → edge → texton → motif → part → object
- Text:
  - Character → word → word group → clause → sentence → story

Why do Network Representation?

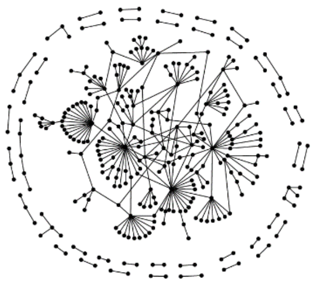
- Networks are a general language for describing and modeling complex systems



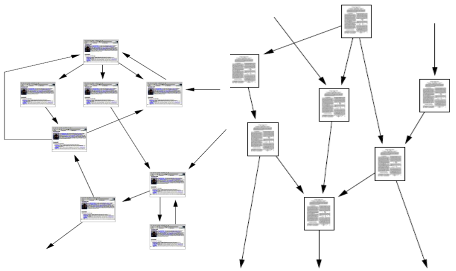
Social networks



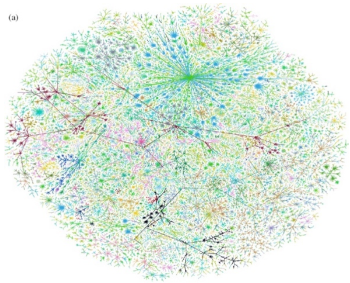
Economic networks



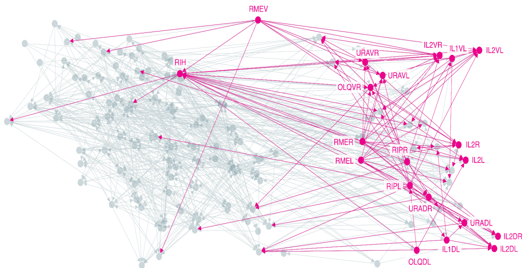
Biomedical networks



Information networks



Internet



Networks of neurons

6

Network Representation

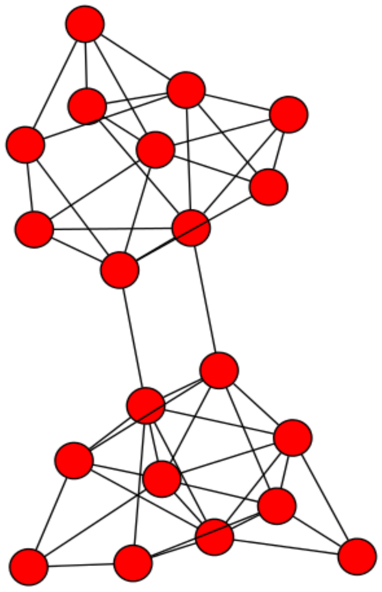
Network Representation (Network Embedding)

Embed each node of a network into

a low-dimensional vector space

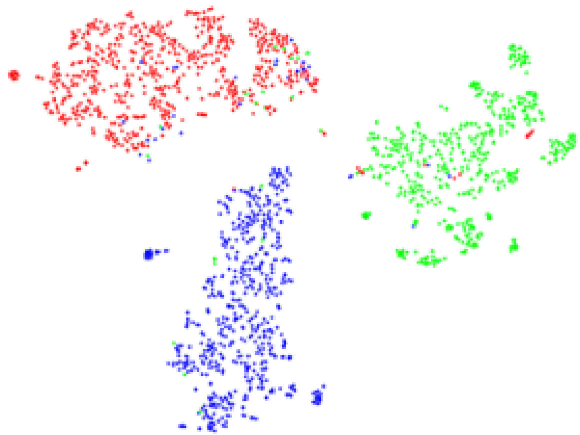
Application

- node classification
- link predication
- community detection
- network evolution
- …



generate

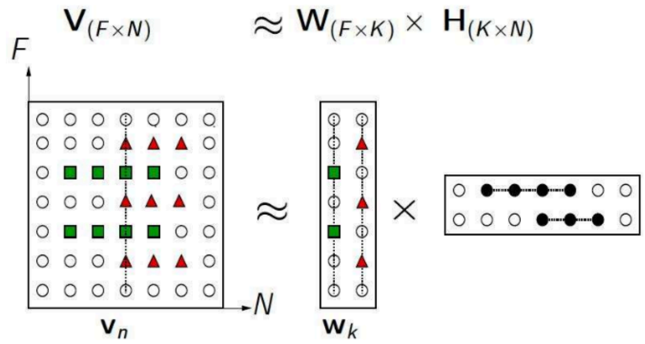
embed

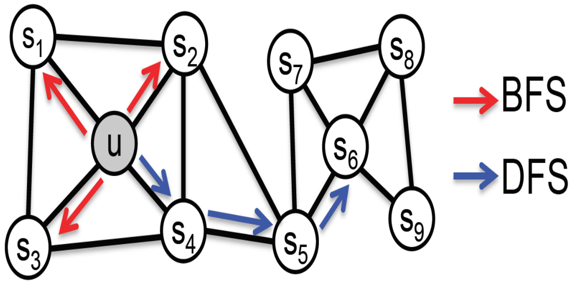


- ☐ Easy to parallel
- ☐ Can apply classical ML methods

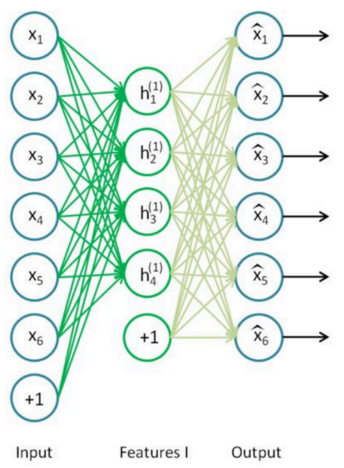
Shallow model

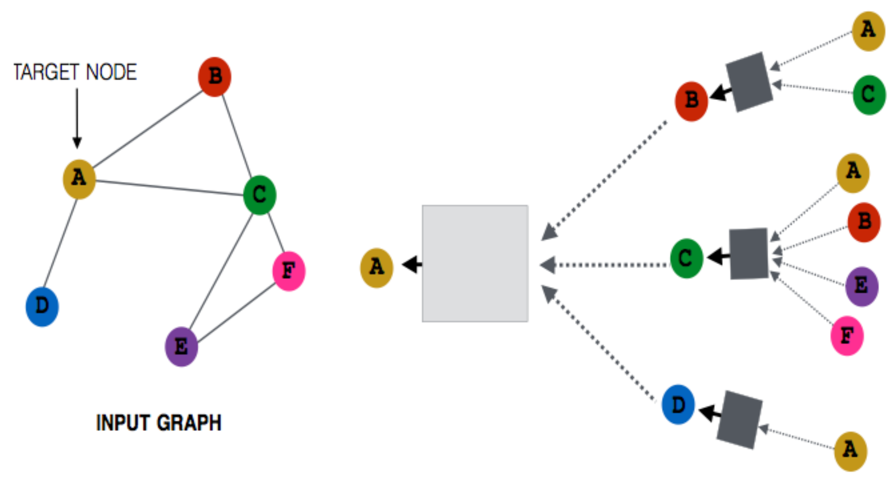
- Factorization-based approaches
  - e.g., Laplacian eigenmaps
- Random walk approaches
  - e.g., DeepWalk, node2vec





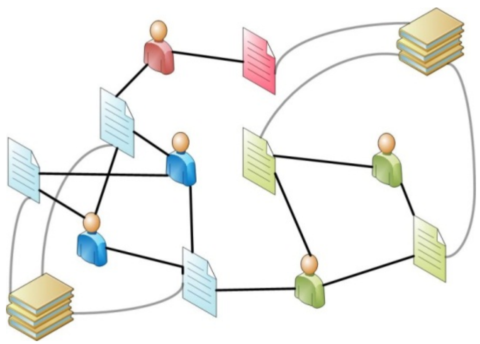
- **Deep model**
  - Apply deep neural network for graph
  - Autoencoder approaches: e.g., DNGR and SDNE
  - GNN based approaches
    - Average neighbor information and apply a neural network
    - e.g., GCN, GraphSage, GAT



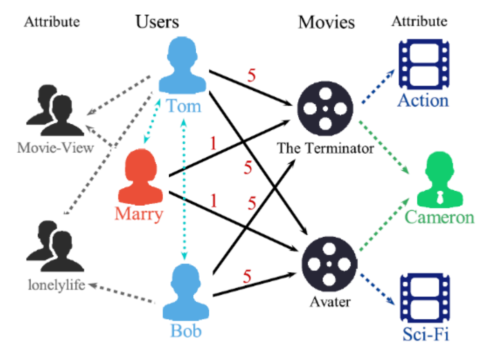


Heterogeneous Graph

- Heterogeneous Graph (HG, Heterogeneous Information Network) contain multiple object types and/or multiple link types.

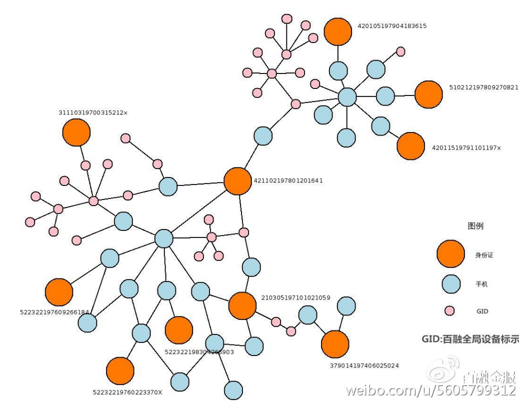


Bibliographic data



Movie data

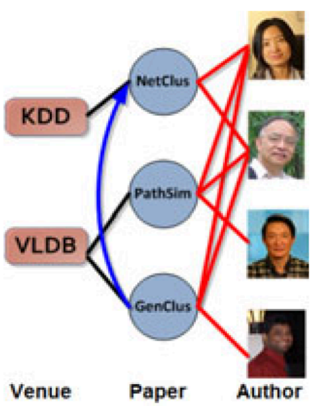
Social network data



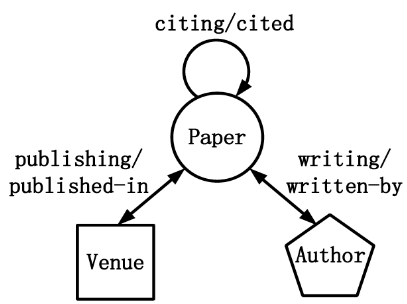
Knowledge graph

Basic Concepts in HG

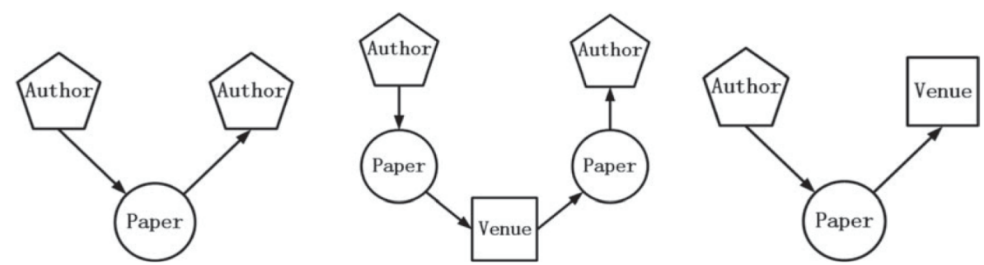
- Network schema
  - Meta-level description of a network
- Meta path (Sun VLDB2011)
  - A relation sequences connecting object pairs
  - Contain rich semantics



(a) Network instance



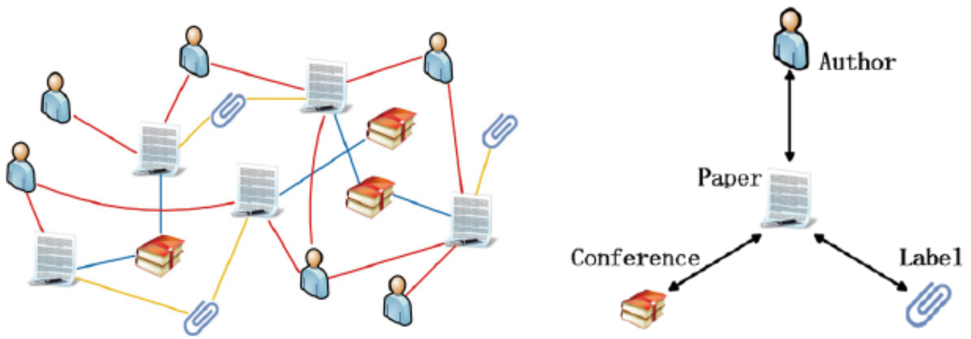
(b) Network schema



(a) APA (b) APVPA (c) APV

Yizhou Sun, Jiawei Han, Xifeng Yan, Philip S. Yu, Tianyi Wu. PathSim: Meta Path-Based Top-k Similarity Search in Heterogeneous Information Networks. VLDB pp. 992-1003, 2011.

- Constrained Meta path
  - Node constrained Meta Path (Shi, KAIS 2016)

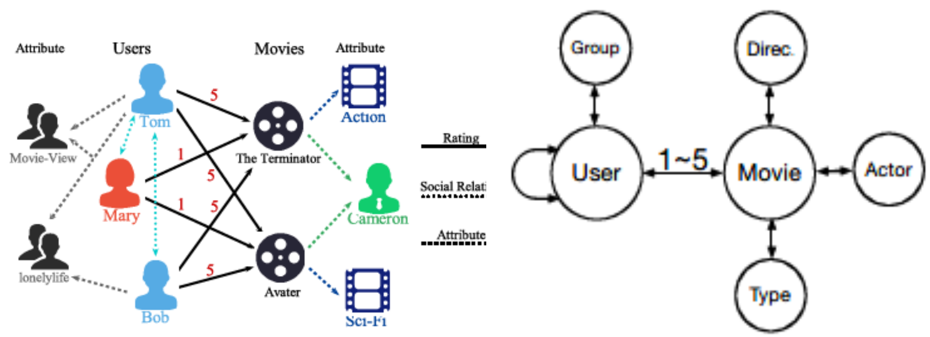


(a) Heterogeneous network (b) Network schema

constraint on objects

$$APA|P.L = \text{“}DM\text{”}$$

  - Link constrained Meta Path (Shi, CIKM 2015)



(a) Heterogeneous network (b) Network schema

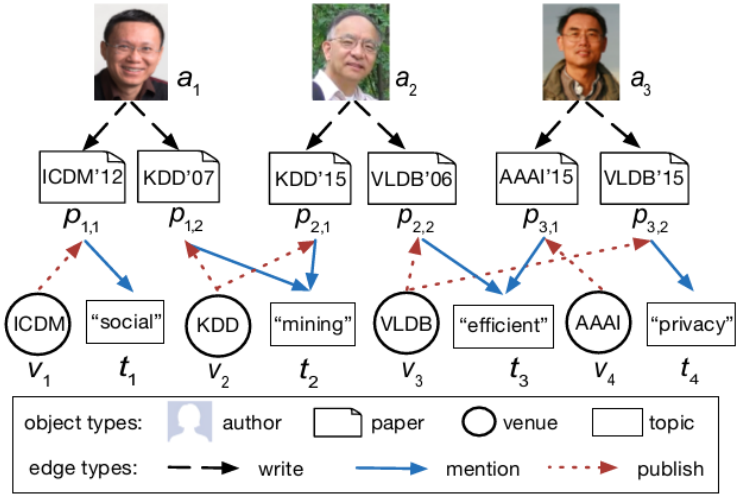
constraint on relations

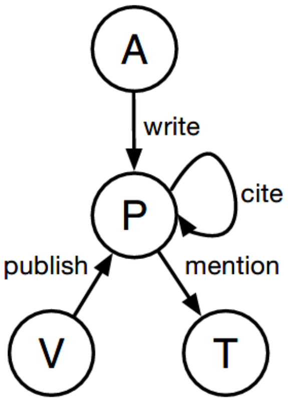
$$\mathrm{U(i)M(j)U}|i = j$$

Chuan Shi, Yitong Li, Philip S. Yu, Bin Wu. Constrained-Meta-Path based Ranking in Heterogeneous Information Network KAIS2016.
Chuan Shi, Zhiqiang Zhang, Ping Luo, Philip S. Yu, Yading Yue, Bin Wu. Semantic Path based Personalized Recommendation on Weighted Heterogeneous Information Networks. CIKM 2015.

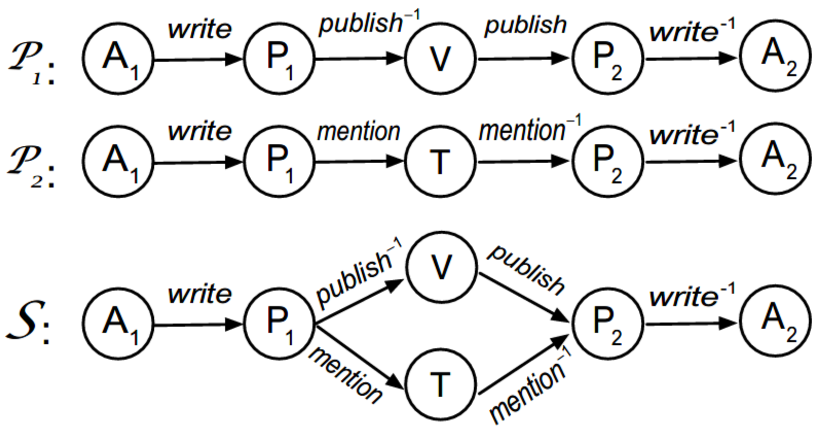
Basic Concepts in HG

- Meta structure/graph (Huang, KDD 2016; Fang, ICDE2016 Zhao, KDD 2017)





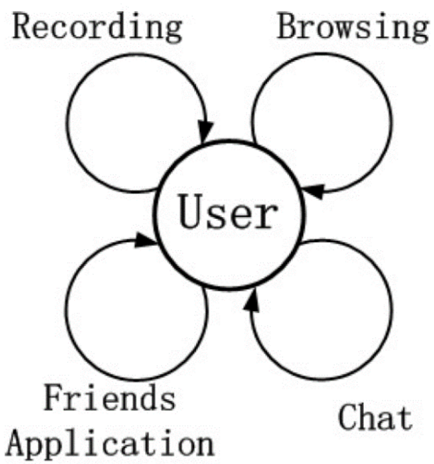
(a) Schema



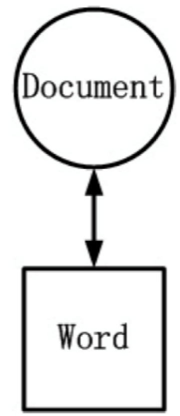
(b) Meta Path, Meta Structure

Zhipeng Huang, Yudian Zheng, Reynold Cheng, Yizhou Sun, Nikos Mamoulis, Xiang Li. Meta structure: Computing relevance in large heterogeneous information networks. KDD 2016.
Huan Zhao, Quanming Yao, Jianda Li, Yangqiu Song, Dik Lun Lee. Meta-graph based recommendation fusion over heterogeneous information networks. KDD 2017.
Yuan Fang, Wenqing Lin, Vincent Wenchen Zheng, Min Wu, Kevin Chen-Chuan Chang, Xiaoli Li. Semantic Proximity Search on Graphs with Metagraph-based Learning. ICDE 2016.

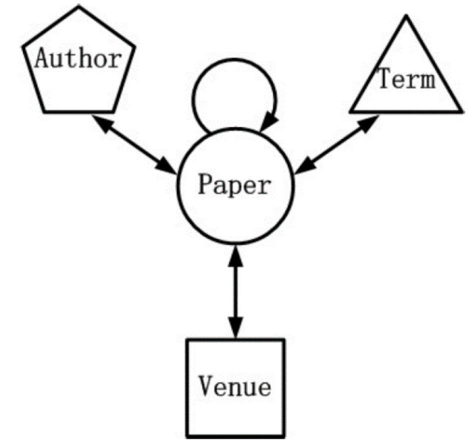
More Examples in Literatures



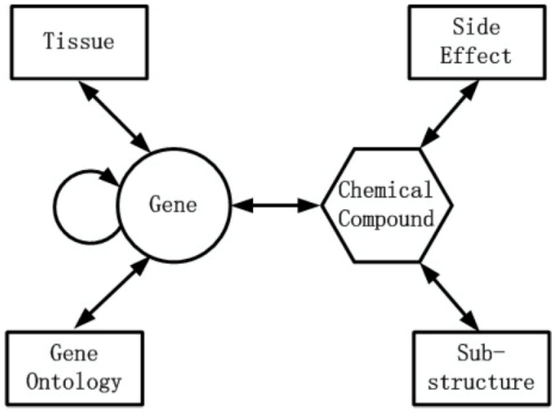
Multi-relational network



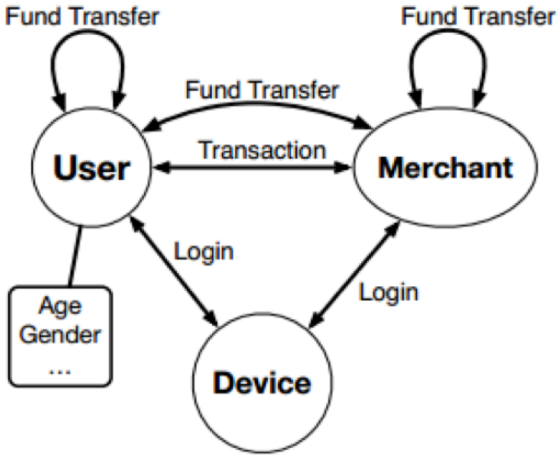
Bipartite network



Star-schema network



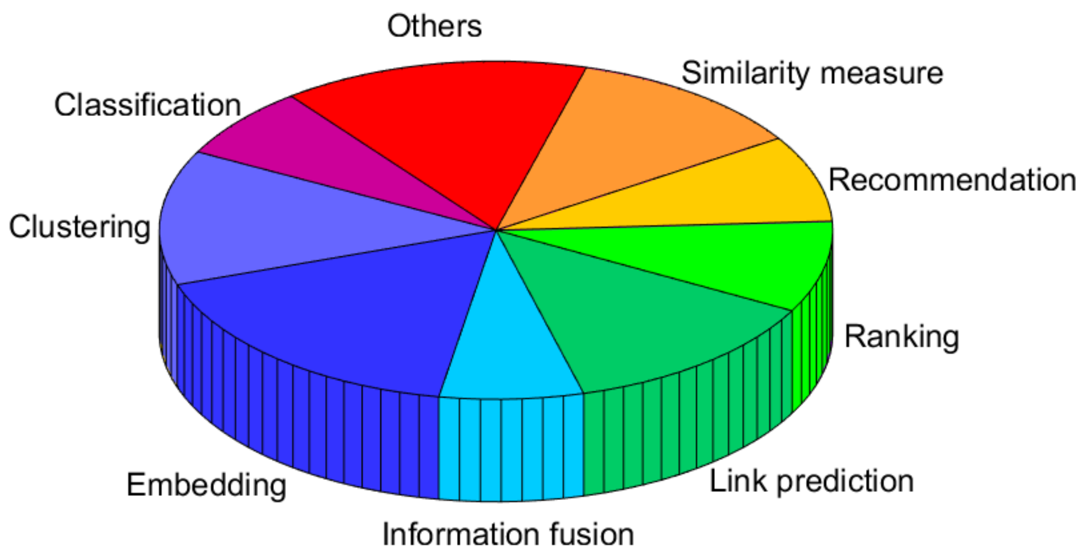
Multiple-hub network



Attributed network

Developments of HG

- Heterogeneous graph has been widely used in data mining



Chuan Shi, Yitong Li, Jiawei Zhang, Yizhou Sun, Philip S. Yu. A survey on Heterogeneous Information Network Analysis. IEEE Transactions on Knowledge and Data Engineering, 29(1), 17-37, 2017 .
Chuan Shi, Philip S. Yu. Heterogeneous Information Network Analysis and Applications. Springer, 2017

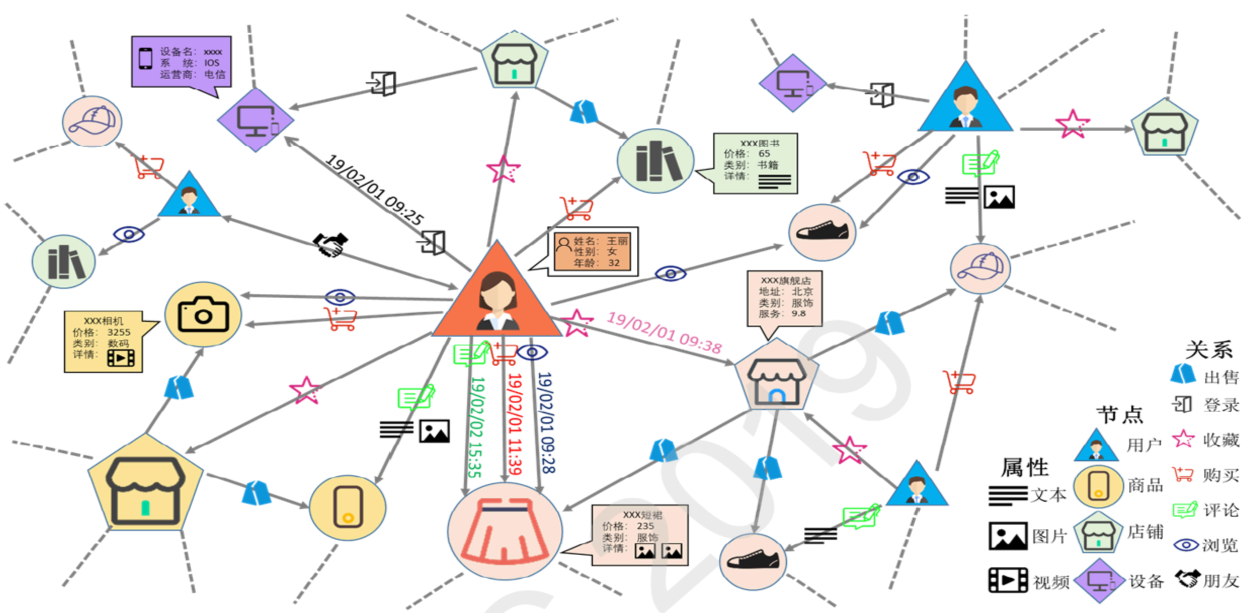
HG Representation

Why HG representation

- Heterogeneity is ubiquitous
- Information loss
- Rich semantics

Challenges

- How to handle heterogeneity
- How to fuse information
- How to capture rich semantics



Static, Topology

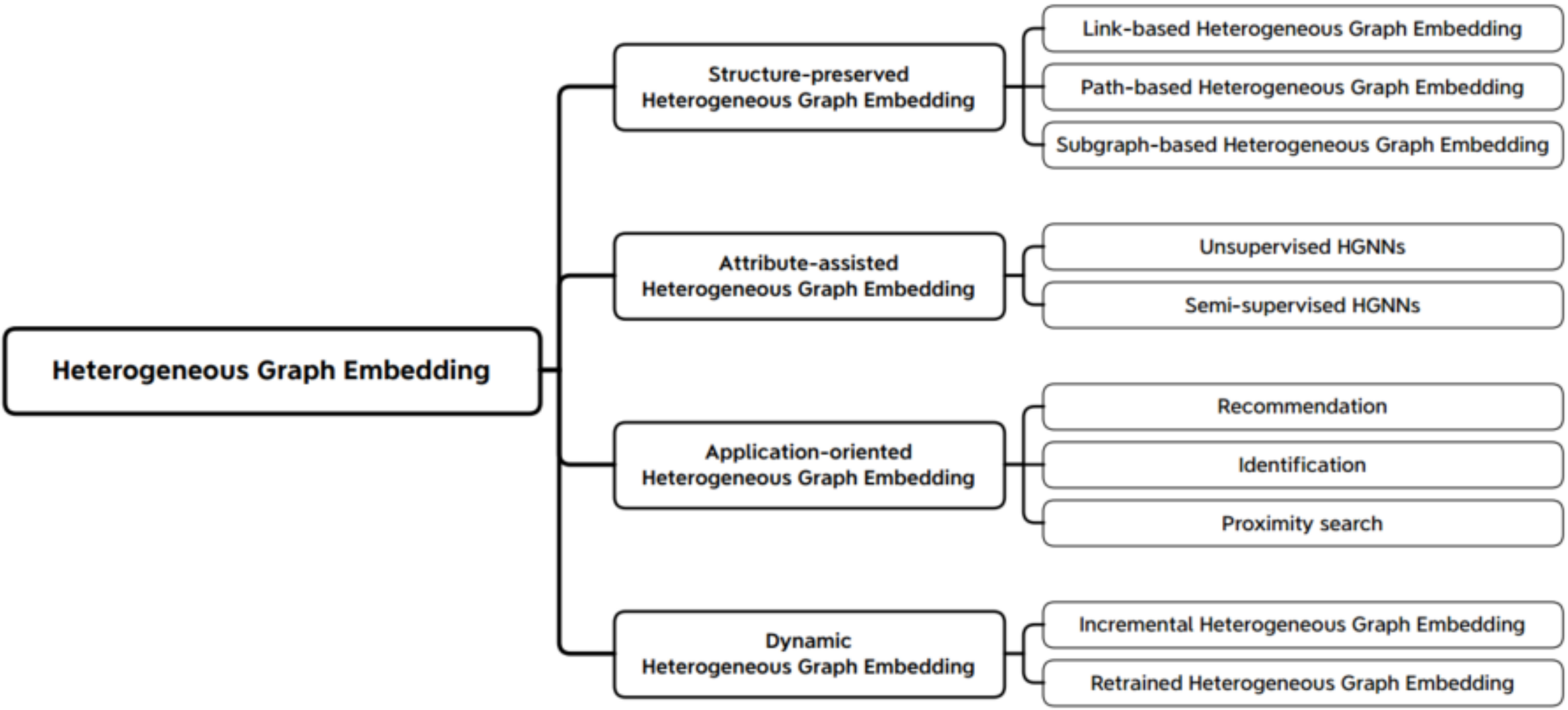
Homogeneous Graph

Dynamic, Attributed

Heterogeneous Graph

Roadmap of HG Representation Techniques

- From the perspective of used information
- From static to dynamic

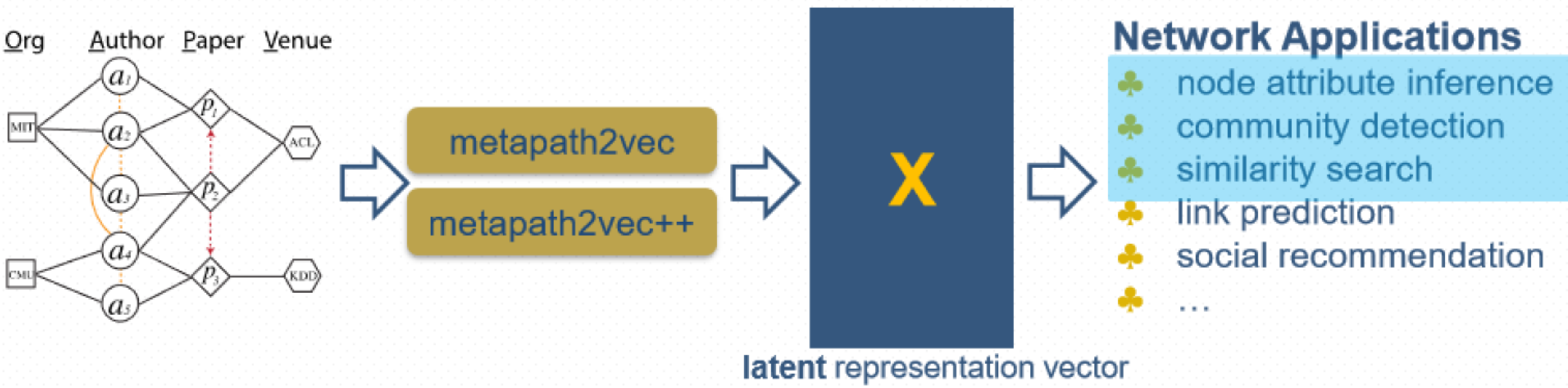


Xiao Wang, Deyu Bo, Chuan Shi, Shaohua Fan, Yanfang Ye, Philip S. Yu. A Survey on Heterogeneous Graph Embedding: Methods, Techniques, Applications and Sources. IEEE Transactions on Big Data, 2021.

Outline

- Basic concepts
- √ Models
  - √ Structure-preserved HG representation
    - MetaPath2Vec (KDD2017), HERec (TKDE2018)
    - MCRec (KDD2018), HeGAN (KDD2019)
  - Attribute-assisted HG representation(HGNN)
  - Dynamic HG representation
- Applications
- Conclusion and future work

Basic idea of Metapath2vec



YuXiao Dong, Nitesh V. Chawla, Ananthram Swami. Metapath2vec: Scalable Representation Learning for Heterogeneous Networks. KDD 2017.

Meta-Path-Based Random Walks

- Meta-Path-Based Random Walks
  - Given a meta-path

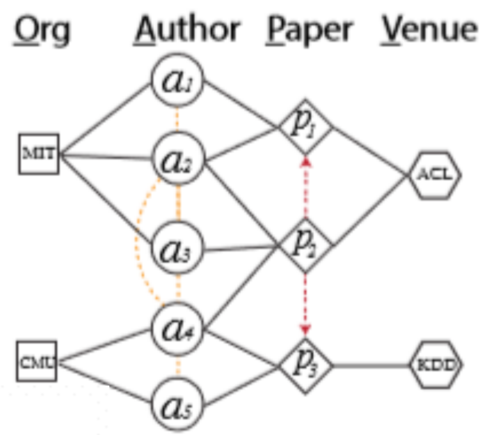
$$\mathcal{P}: V_1 \xrightarrow{R_1} V_2 \xrightarrow{R_2} \cdots V_t \xrightarrow{R_t} V_{t+1} \cdots \xrightarrow{R_{l-1}} V_l$$

  - The transition probability at step i is defined as

$$p(v^{i+1}|v_t^i, \mathcal{P}) = \begin{cases} \frac{1}{|N_{t+1}(v_t^i)|} & (v^{i+1}, v_t^i) \in E, \phi(v^{i+1}) = t+1 \\ 0 & (v^{i+1}, v_t^i) \in E, \phi(v^{i+1}) \neq t+1 \\ 0 & (v^{i+1}, v_t^i) \notin E \end{cases}$$

  - Recursive guidance for a symmetric path, i.e.,

$$p(v^{i+1}|v_t^i) = p(v^{i+1}|v_1^i), \text{ if } t = l$$



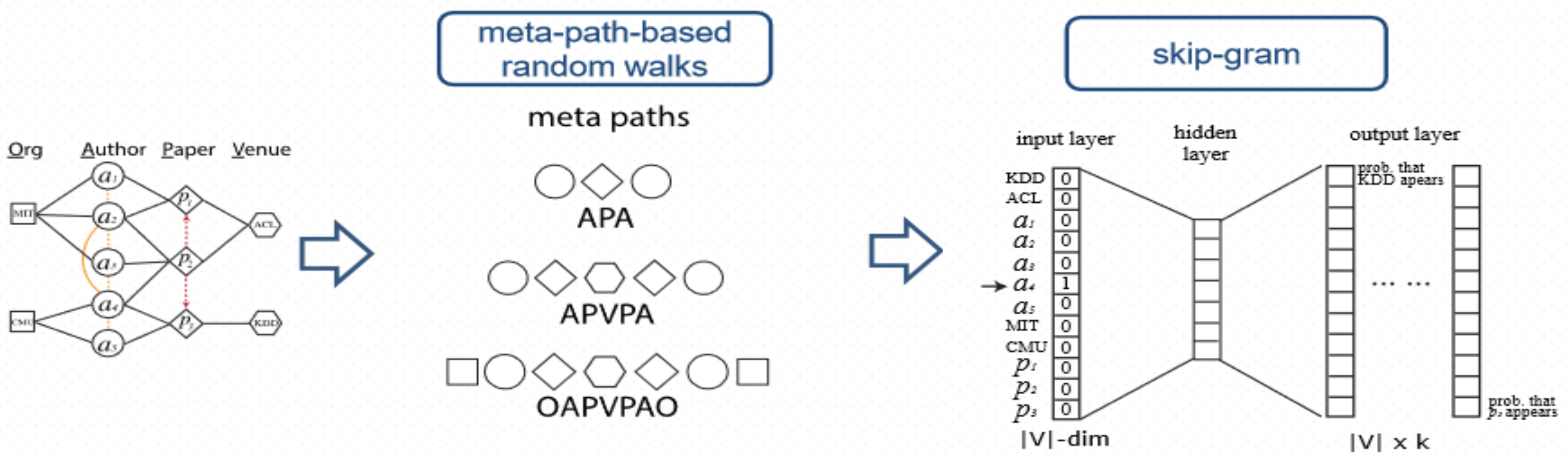
Skip-gram of metapath2vec

- Skip-gram's potential issue

$$\arg\max_{\theta} \prod_{v \in V} \prod_{c \in N(v)} p(c|v;\theta) \quad \Rightarrow \quad \arg\max_{\theta} \sum_{v \in V} \sum_{t \in T_V} \sum_{c_t \in N_t(v)} \log p(c_t|v;\theta)$$

$$p(c_t|v;\theta) = \frac{e^{X_{c_t} \cdot X_v}}{\sum_{u \in V} e^{X_u \cdot X_v}},$$

$$\log \sigma(X_{c_t} \cdot X_v) + \sum_{m=1}^{M} \mathbb{E}_{u^m \sim P(u)}[\log \sigma(-X_{u^m} \cdot X_v)]$$



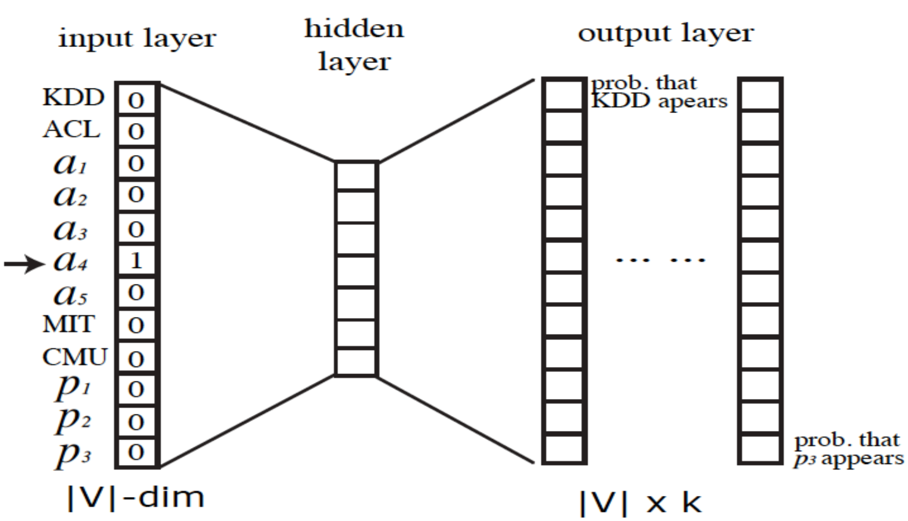
metapath2vec vs metapath2vec++

- Metapath2vec

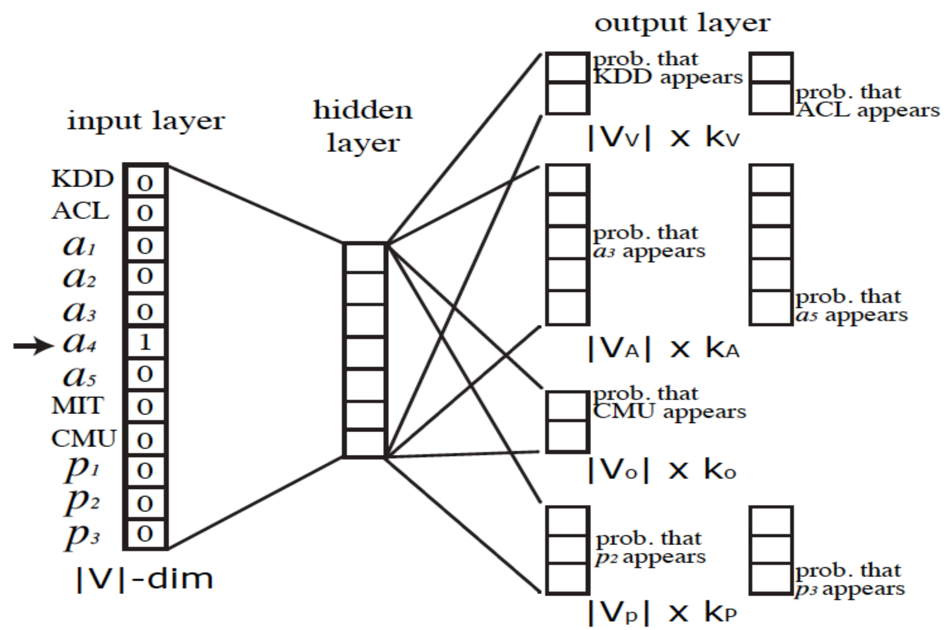
$$p(c_t|v;\theta) = \frac{e^{X_{c_t}} \cdot e^{X_v}}{\sum_{u \in V} e^{X_u} \cdot e^{X_v}}$$

metapath2vec++

$$p(c_t|v;\theta) = \frac{e^{X_{c_t}} \cdot e^{X_v}}{\sum_{u_t \in V_t} e^{X_{u_t}} \cdot e^{X_v}}$$



(b) Skip-gram in *metapath2vec*, node2vec, & DeepWalk



(c) Skip-gram in *metapath2vec++*

Effectiveness Experiments

Application 1: Multi-Class Node Classification

AMiner Academic Network

- ✓ 9 1.7 million authors
- ✓ 3800+ venues
- ✓ 8 research areas
- ✓ 3 million papers

- Baselines
  - ✓ DeepWalk
  - ✓ node2vec
  - ✓ LINE
  - ✓ PTE

Table 2: Multi-class venue node classification results in AMiner data.

Metric	Method	5%	10%	20%	30%	40%	50%	60%	70%	80%	90%
Macro-F1	DeepWalk/node2vec	0.0723	0.1396	0.1905	0.2795	0.3427	0.3911	0.4424	0.4774	0.4955	0.4457
	LINE (1st+2nd)	0.2245	0.4629	0.7011	0.8473	0.8953	0.9203	0.9308	0.9466	0.9410	0.9466
	PTE	0.1702	0.3388	0.6535	0.8304	0.8936	0.9210	0.9352	0.9505	0.9525	0.9489
	metapath2vec	0.3033	0.5247	0.8033	0.8971	0.9406	0.9532	0.9529	0.9701	0.9683	0.9670
	metapath2vec++	0.3090	0.5444	0.8049	0.8995	0.9468	0.9580	0.9561	0.9675	0.9533	0.9503
Micro-F1	DeepWalk/node2vec	0.1701	0.2142	0.2486	0.3266	0.3788	0.4090	0.4630	0.4975	0.5259	0.5286
	LINE (1st+2nd)	0.3000	0.5167	0.7159	0.8457	0.8950	0.9209	0.9333	0.9500	0.9556	0.9571
	PTE	0.2512	0.4267	0.6879	0.8372	0.8950	0.9239	0.9352	0.9550	0.9667	0.9571
	metapath2vec	0.4173	0.5975	0.8327	0.9011	0.9400	0.9522	0.9537	0.9725	0.9815	0.9857
	metapath2vec++	0.4331	0.6192	0.8336	0.9032	0.9463	0.9582	0.9574	0.9700	0.9741	0.9786

Table 3: Multi-class author node classification results in AMiner data.

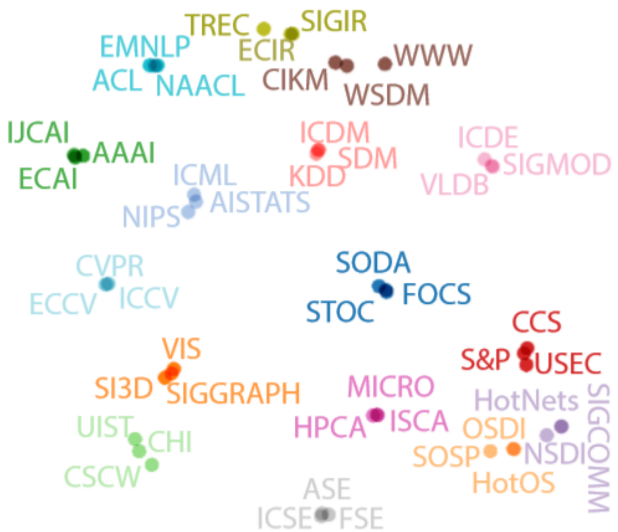
Metric	Method	5%	10%	20%	30%	40%	50%	60%	70%	80%	90%
Macro-F1	DeepWalk/node2vec	0.7153	0.7222	0.7256	0.7270	0.7273	0.7274	0.7273	0.7271	0.7275	0.7275
	LINE (1st+2nd)	0.8849	0.8886	0.8911	0.8921	0.8926	0.8929	0.8934	0.8936	0.8938	0.8934
	PTE	0.8898	0.8940	0.897	0.8982	0.8987	0.8990	0.8997	0.8999	0.9002	0.9005
	metapath2vec	0.9216	0.9262	0.9292	0.9303	0.9309	0.9314	0.9315	0.9316	0.9319	0.9320
	metapath2vec++	0.9107	0.9156	0.9186	0.9199	0.9204	0.9207	0.9207	0.9208	0.9211	0.9212
Micro-F1	DeepWalk/node2vec	0.7312	0.7372	0.7402	0.7414	0.7418	0.7420	0.7419	0.7420	0.7425	0.7425
	LINE (1st+2nd)	0.8936	0.8969	0.8993	0.9002	0.9007	0.9010	0.9015	0.9016	0.9018	0.9017
	PTE	0.8986	0.9023	0.9051	0.9061	0.9066	0.9068	0.9075	0.9077	0.9079	0.9082
	metapath2vec	0.9279	0.9319	0.9346	0.9356	0.9361	0.9365	0.9365	0.9365	0.9367	0.9369
	metapath2vec++	0.9173	0.9217	0.9243	0.9254	0.9259	0.9261	0.9261	0.9262	0.9264	0.9266

Effectiveness Experiments

Application 2: Node Clustering

Node clustering results (NMI) in AMiner data.

methods	venue	author
DeepWalk/node2vec	0.1952	0.2941
LINE (1st+2nd)	0.8967	0.6423
PTE	0.9060	0.6483
metapath2vec	0.9274	0.7470
metapath2vec++	0.9261	0.7354



Application 3: Similarity Search

Table 1: Case study of similarity search in the heterogeneous DBIS data used in [26].

Method	PathSim [26]		DeepWalk / node2vec [8, 22]		LINE (1st+2nd) [30]		PTE [29]		*metapath2vec*		*metapath2vec++*	
Input	meta-paths		heterogeneous random walk paths		heterogeneous edges		heterogeneous edges		probabilistic meta-paths		probabilistic meta-paths	
Query	PKDD	C. Faloutsos	PKDD	C. Faloutsos	PKDD	C. Faloutsos	PKDD	C. Faloutsos	PKDD	C. Faloutsos	PKDD	C. Faloutsos
1	ICDM	J. Han	R. S.	J. Pan	W. K.	C. Aggarwal	KDD	C. Aggarwal	A. S.	C. Aggarwal	KDD	R. Agrawal
2	SDM	R. Agrawal	M. N.	H. Tong	S. A.	P. Yu	ICDM	P. Yu	M. B.	J. Pei	PAKDD	J. Han
3	PAKDD	J. Pei	R. P.	H. Yang	A. B.	D. Gunopulos	SDM	Y. Tao	P. B.	P. Yu	ICDM	J. Pei
4	KDD	C. Aggarwal	G. G.	R. Filho	M. S.	N. Koudas	DMKD	N. Koudas	M. S.	H. Cheng	DMKD	C. Aggarwal
5	DMKD	H. Jagadish	F. J.	R. Chan	S. A.	M. Vlachos	PAKDD	R. Rastogi	M. K.	V. Ganti	SDM	P. Yu

Framework of HERec

Heterogeneous information network **E**mbedding for **Rec**ommendation (HERec)

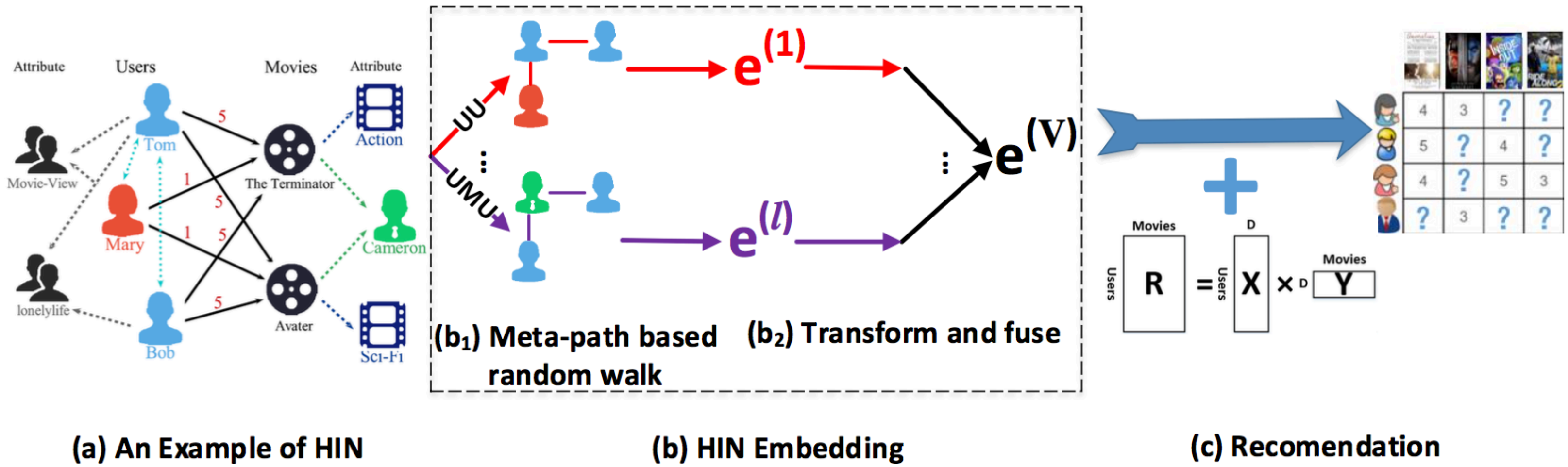


Figure 1: The schematic illustration of the proposed HERec approach.

HERec
- **Heterogeneous information network embedding**
- **Fuse embeddings into MF for recommendation**

Chuan Shi, Binbin Hu, Wayne Xin Zhao, Philip S. Yu. Heterogeneous Information Network Embedding for Recommendation. TKDE 2018

HIN Embedding – Single Meta-path

Mete-path based Random Walk

$$P(n_{t+1}=x \mid n_t=v,\rho) = \begin{cases} \dfrac{1}{|\mathcal{N}^{A_{t+1}}(v)|}, & (v,x)\in\mathcal{E} \text{ and } \phi(x)=A_{t+1} \\ 0, & \text{otherwise,} \end{cases}$$

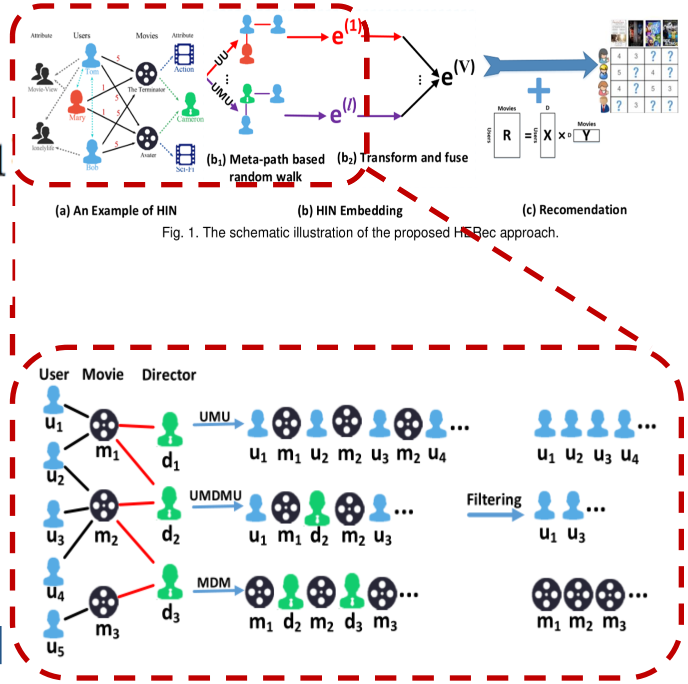


Fig. 1. The schematic illustration of the proposed HERec approach.

Type Constraints and Filtering

- Uncover heterogeneous information with homogeneous node embedding objective
- Utilize more neighbor information in a fixed length window

Embedding with Skip-Gram for Single Meta-path

$$\max_f \sum_{u\in\mathcal{V}} \log Pr(\mathcal{N}_u \mid f(u))$$

HIN Embedding – Fusion

Embedding Fusion

- Integrate information of various meta-paths
- A good fusion function should be learned according to the specific task

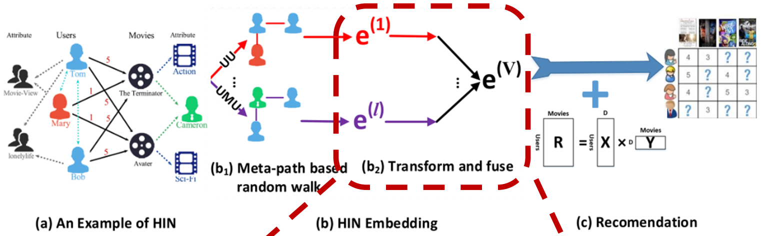


Fig. 1. The schematic illustration of the proposed HERec approach.

- Simple linear fusion.

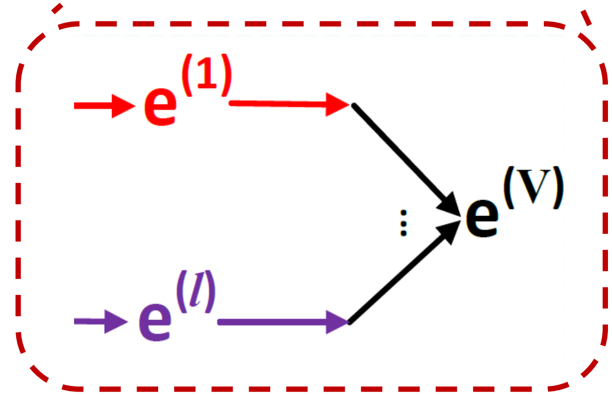
$$g(\{\boldsymbol{e}_u^{(l)}\}) = \frac{1}{|\mathcal{P}|}\sum_{l=1}^{|\mathcal{P}|}(\mathbf{M}^{(l)}\boldsymbol{e}_u^{(l)} + \boldsymbol{b}^{(l)})$$

- Personalized linear fusion.

$$g(\{\boldsymbol{e}_u^{(l)}\}) = \sum_{l=1}^{|\mathcal{P}|} w_u^{(l)}(\mathbf{M}^{(l)}\boldsymbol{e}_u^{(l)} + \boldsymbol{b}^{(l)})$$

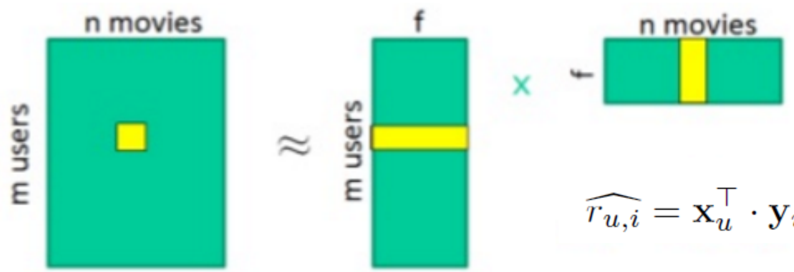
- Personalized non-linear fusion.

$$g(\{\boldsymbol{e}_u^{(l)}\}) = \sigma\left(\sum_{l=1}^{|\mathcal{P}|} w_u^{(l)}\sigma(\mathbf{M}^{(l)}\boldsymbol{e}_u^{(l)} + \boldsymbol{b}^{(l)})\right)$$



Extended MF

Basic Rating Prediction



$$\widehat{r_{u,i}} = \mathbf{x}_u^\top \cdot \mathbf{y}_i,$$

Extended Rating Prediction

- Integrated with HIN Embeddings

$$\widehat{r_{u,i}} = \mathbf{x}_u^\top \cdot \mathbf{y}_i + \alpha \cdot \mathbf{e}_u^{(U)^\top} \cdot \boldsymbol{\gamma}_i^{(I)} + \beta \cdot \boldsymbol{\gamma}_u^{(U)^\top} \cdot \mathbf{e}_i^{(I)},$$

Model Optimization Objective

$$\pounds = \sum_{\langle u,i,r_{u,i}\rangle \in \mathcal{R}} (r_{u,i} - \widehat{r_{u,i}})^2 + \lambda \sum_u (\|\mathbf{x}_u\|_2 + \|\mathbf{y}_i\|_2 + \|\boldsymbol{\gamma}_u^{(U)}\|_2 + \|\boldsymbol{\gamma}_i^{(I)}\|_2 + \|\boldsymbol{\Theta}^{(U)}\|_2 + \|\boldsymbol{\Theta}^{(I)}\|_2), \qquad (9)$$

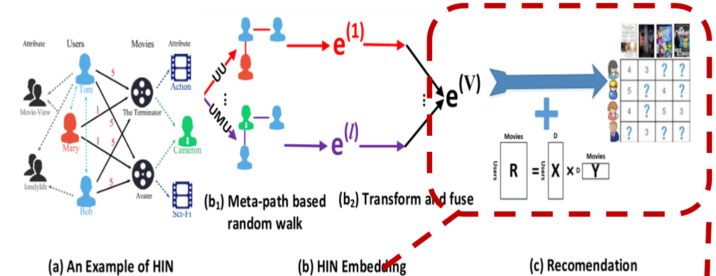
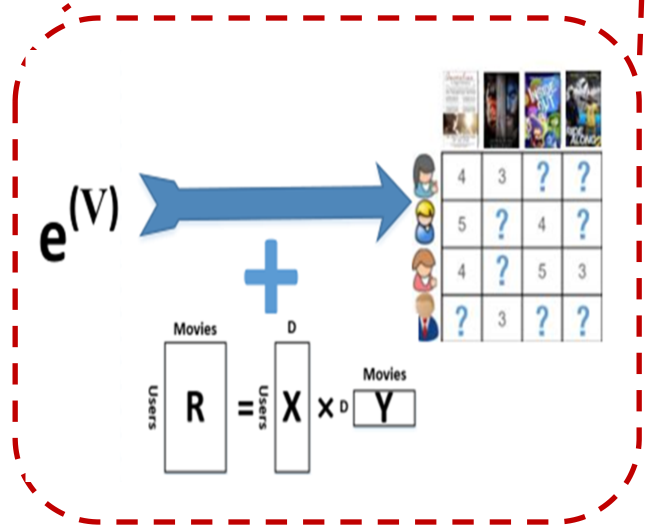


Fig. 1. The schematic illustration of the proposed HERec approach.



Experimental Setup

Dataset

Dataset (Density)	Relations (A-B)	Number of A	Number of B	Number of (A-B)	Ave. degrees of A	Ave. degrees of B	Meta-paths
Douban Movie (0.63%)	User-Movie	13,367	12,677	1,068,278	79.9	84.3	UMU, MUM UMDMU, MDM UMAMU, MAM UMTMU, MTM
	User-User	2,440	2,294	4,085	1.7	1.8	
	User-Group	13,337	2,753	570,047	42.7	207.1	
	Movie-Director	10,179	2,449	11,276	1.1	4.6	
	Movie-Actor	11,718	6,311	33,587	2.9	5.3	
	Movie-Type	12,678	38	27,668	2.2	728.1	
Douban Book (0.27%)	User-Book	13,024	22,347	792,026	60.8	35.4	UBU, BUB UBPBU, BPB UBYBU, BYB UBABU
	User-User	12,748	12,748	169,150	13.3	13.3	
	Book-Author	21,907	10,805	21,905	1.0	2.0	
	Book-Publisher	21,773	1,815	21,773	1.0	11.9	
	Book-Year	21,192	64	21,192	1.0	331.1	
Yelp (0.08%)	User-Business	16,239	14,284	198,397	12.2	13.9	UBU, BUB UBCiBU, BCiB UBCaBU, BCaB
	User-User	10,580	10,580	158,590	15.0	15.0	
	User-Compliment	14,411	11	76,875	5.3	6988.6	
	Business-City	14,267	47	14,267	1.0	303.6	
	Business-Category	14,180	511	40,009	2.8	78.3	

Metrics

- Mean absolute error **(MAE)**
- Root mean square error **(RMSE)**

Baselines

Typical methods

- PMF
- SoMF

HIN-based Methods

- FMHIN
- HeteMF
- SemRec
- DSR

Our Methods

- HERec_{dw}
- HERec_{mp}
- HERec

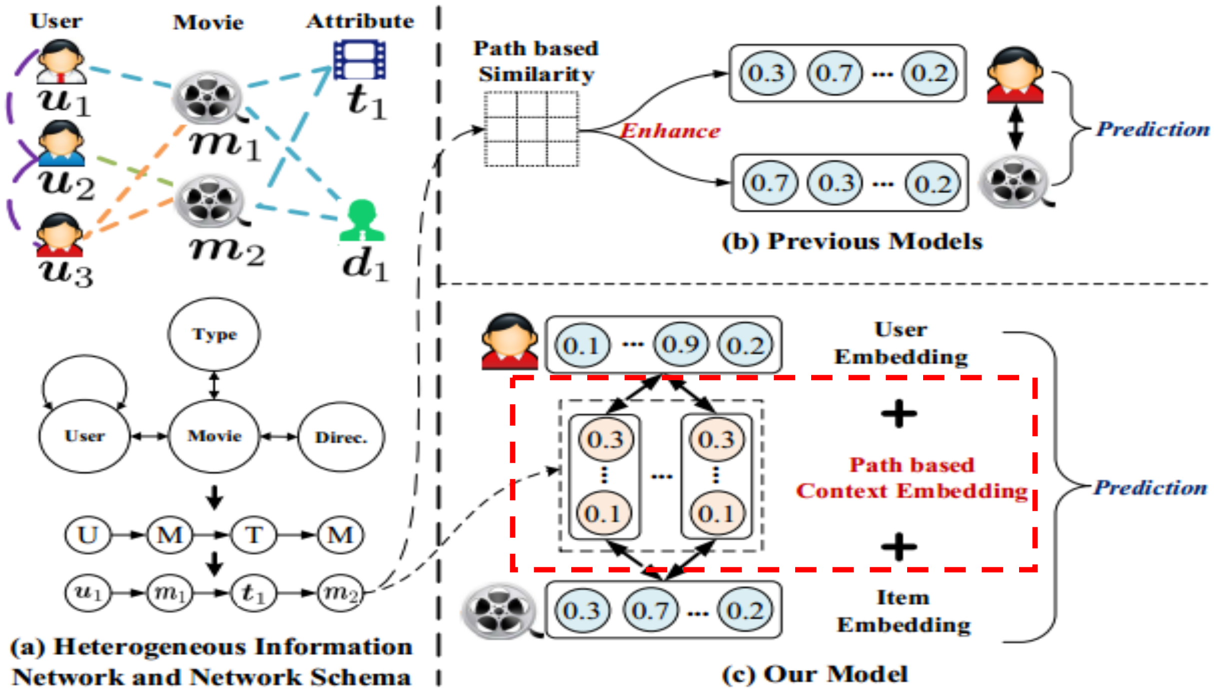
Effectiveness Experiments

Table 3: Results of effectiveness experiments on three datasets. A smaller value indicates a better performance.

Dataset	Training	Metrics	PMF	SoMF	FM_{HIN}	HeteMF	SemRec	DSR	$HERec_{dw}$	$HERec_{mp}$	$HERec_{sl}$	$HERec_{pl}$	$HERec_{pnl}$
Douban Movie	80%	MAE	0.5741	0.5817	0.5696	0.5750	0.5695	0.5681	0.5703	0.5515	0.5617	0.5523	**0.5519**
		RMSE	0.7641	0.7680	0.7248	0.7556	0.7399	0.7225	0.7446	0.7121	0.7216	**0.7024**	0.7053
	60%	MAE	0.5867	0.5991	0.5769	0.5894	0.5738	0.5831	0.5838	0.5611	0.5711	0.5606	**0.5587**
		RMSE	0.7891	0.7950	0.7342	0.7785	0.7551	0.7408	0.7670	0.7264	0.7336	**0.7142**	0.7148
	40%	MAE	0.6078	0.6328	0.5871	0.6165	0.5945	0.6170	0.6073	0.5747	0.5832	0.5732	**0.5699**
		RMSE	0.8321	0.8479	0.7563	0.8221	0.7836	0.7850	0.8057	0.7429	0.7514	0.7334	**0.7315**
	20%	MAE	0.7247	0.6979	0.6080	0.6896	0.6392	0.6584	0.6699	0.6063	0.5953	0.5965	**0.5900**
		RMSE	0.9440	0.9852	0.7878	0.9357	0.8599	0.8345	0.9076	0.7877	0.7916	0.7674	**0.7660**
Douban Book	80%	MAE	0.5774	0.5756	0.5716	0.5740	0.5675	0.5740	0.5875	0.5591	0.5578	0.5556	**0.5502**
		RMSE	0.7414	0.7302	0.7199	0.7360	0.7283	0.7206	0.7450	0.7081	0.7079	0.7093	**0.6811**
	60%	MAE	0.6065	0.5603	0.5812	0.5823	0.5833	0.6020	0.6203	0.5666	0.5690	0.5669	**0.5600**
		RMSE	0.7908	0.7518	0.7319	0.7466	0.7505	0.7552	0.7905	0.7318	0.7251	0.7274	**0.7123**
	40%	MAE	0.6800	0.6161	0.6028	0.5982	0.6025	0.6271	0.6976	0.5954	0.5838	**0.5638**	0.5774
		RMSE	0.9203	0.7936	0.7617	0.7779	0.7751	0.7730	0.9022	0.7703	0.7490	0.7549	**0.7400**
	20%	MAE	1.0344	0.6327	0.6396	0.6311	0.6481	0.6300	1.0166	0.6785	**0.6232**	0.6347	0.6450
		RMSE	1.4414	0.8236	0.8188	0.8304	0.8350	0.8200	1.3205	0.8869	**0.8168**	0.8382	0.8581
Yelp	90%	MAE	1.0412	1.0095	0.9013	0.9487	0.9043	0.9054	1.0388	0.8822	0.8643	0.8506	**0.8395**
		RMSE	1.4268	1.3392	1.1417	1.2549	1.1637	1.1186	1.3581	1.1309	1.1204	1.0948	**1.0907**
	80%	MAE	1.0791	1.0373	0.9038	0.9654	0.9176	0.9098	1.0750	0.8953	0.8789	0.8578	**0.8475**
		RMSE	1.4816	1.3782	1.1497	1.2799	1.1771	1.1208	1.4075	1.1516	1.1403	1.1139	**1.1117**
	70%	MAE	1.1170	1.0694	0.9108	0.9975	0.9407	0.9429	1.1196	0.9043	0.8889	0.8650	**0.8580**
		RMSE	1.5387	1.4201	1.1651	1.3229	1.2108	1.1582	1.4632	1.1639	1.1599	**1.1229**	1.1256
	60%	MAE	1.1778	1.1135	0.9435	1.0368	0.9637	1.0043	1.1691	0.9257	0.9042	0.8818	**0.8759**
		RMSE	1.6167	1.4748	1.2039	1.3713	1.2380	1.2257	1.5182	1.1887	1.1817	**1.1451**	1.1488
Average Rank			10.33	8.79	5.25	7.83	6.54	6.13	9.5	4.17	3.2	2.4	**1.79**

Basic idea

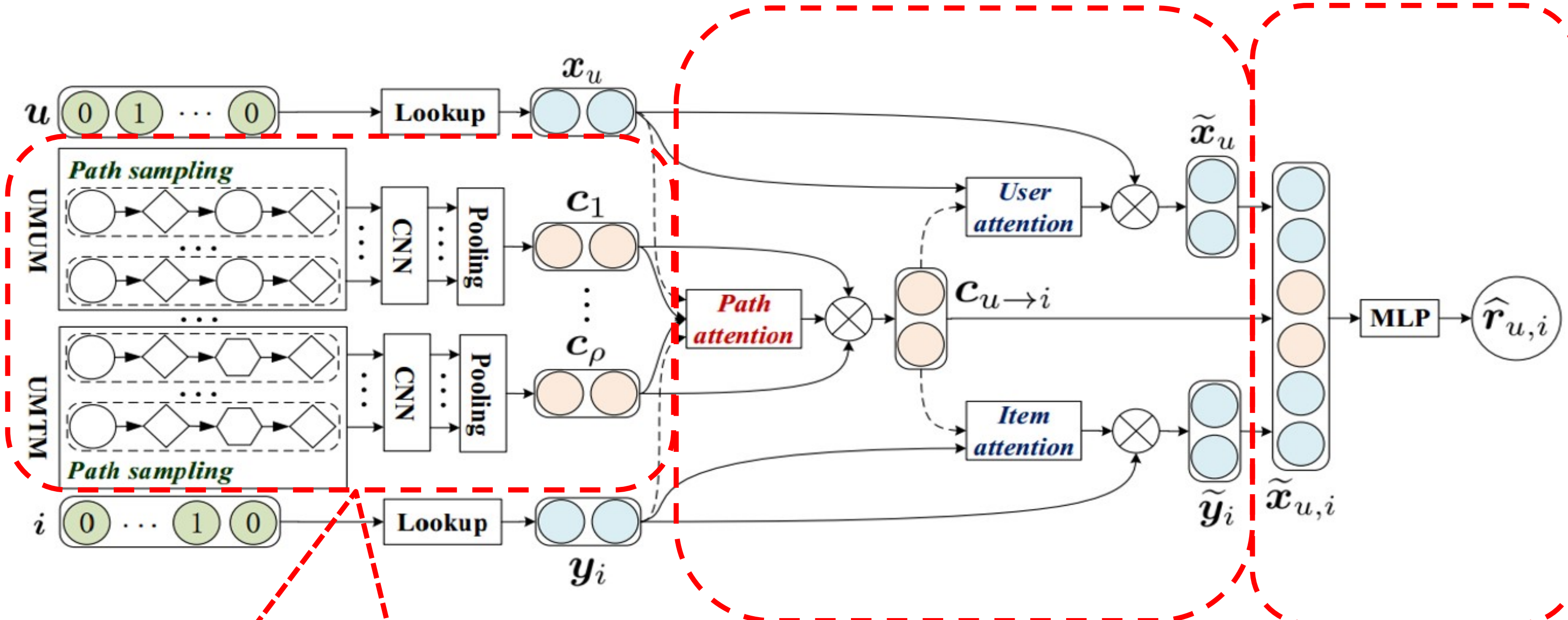
- Learn explicit representations for meta-path based context tailored for the recommendation task
- Characterize a three-way interaction ⟨user, meta-path, item⟩



Binbin Hu, Chuan Shi, Wayne Xin Zhao, Philip S. Yu. Leveraging Meta-path based Context for Top-N Recommendation with A Neural Co-Attention Model. KDD 2018

Framework of MCRec

Meta-path based Context for Recommendation (MCRec)



Interpretability
Meta-path based context embedding

Mutual Effect
Neural co-attention mechanism

Rank
Ranking predication model

Experimental Setup

Dataset

Datasets	Relations (A-B)	#A	#B	#A-B	Meta-paths
Movielens	*User-Movie*	943	1,682	100,000	UMUM
	User-User	943	943	47,150	UMGM
	Movie-Movie	1,682	1,682	82,798	UUUM
	Movie-Genre	1,682	18	2861	UMMM
LastFM	*User-Artist*	1,892	17,632	92,834	UATA
	User-User	1,892	1,892	18,802	UAUA
	Artist-Artist	17,632	17,632	153,399	UUUA
	Artist-Tag	17,632	11,945	184,941	UUA
Yelp	*User-Business*	16,239	14,284	198,397	UBUB
	User-User	16,239	16,239	158,590	UBCaB
	Business-City (Ci)	14,267	47	14,267	UUB
	Business-Category (Ca)	14,180	511	40,009	UBCiB

Metrics

- Perc@10
- Recall@10
- NDCG@10

Baselines

- **CF based Methods**
  - ItemKNN
  - BPR
  - MF
  - NeuMF
- **HIN based Methods**
  - SVDFeature_{hete}
  - SVDFeature_{mp}
  - HeteRS
  - FMG_{rank}

Methods

- MCRec_{rand}
- MCRec_{avg}
- MCRec_{mp}
- MCRec

Effectiveness Experiments

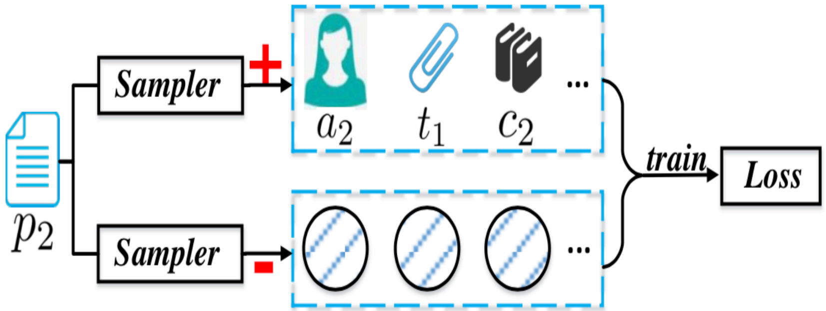
Model	Movielens			LastFM			Yelp		
	Prec@10	Recall@10	NDCG@10	Prec@10	Recall@10	NDCG@10	Prec@10	Recall@10	NDCG@10
ItemKNN	0.2578	0.1536	0.5692	0.4160	0.4513	0.7981	0.1386	0.5421	0.5378
BRP	0.3010	0.1946	0.6459	0.4129	0.4492	0.8099	0.1474	0.5504	0.5549
MF	0.3247	0.2053	0.6511	0.4364	0.4634	0.7921	0.1503	0.5350	0.5322
NeuMF	0.3293*	0.2090	0.6587	0.4540	0.4678	0.8104	0.1504	0.5857	0.5713
SVDFeature_{hete}	0.3171	0.2021	0.6445	0.4576	0.4841	0.8290*	0.1404	0.5613	0.5289
SVDFeature_{mp}	0.3109	0.1929	0.6536	0.4391	0.4651	0.8116	0.1524	0.5932	0.5974*
HeteRS	0.2485	0.1674	0.5967	0.4276	0.4489	0.8026	0.1423	0.5613	0.5600
FMG_{rank}	0.3256	0.2165*	0.6682*	0.4630*	0.4916*	0.8263	0.1538*	0.5951*	0.5861
MCRec_{rand}	0.3223	0.2104	0.6650	0.4540	0.4795	0.8002	0.1510	0.5842	0.5718
MCRec_{avg}	0.3270	0.2111	0.6631	0.4645	0.4914	0.8311	0.1595	0.5933	0.6021
MCRec_{mp}	0.3401	0.2200	0.6828	0.4662	0.4924	0.8428	0.1655	0.6303	0.6228
MCRec	**0.3451#**	**0.2256#**	**0.6900#**	**0.4807#**	**0.5068#**	**0.8526#**	**0.1686#**	**0.6326#**	**0.6301#**

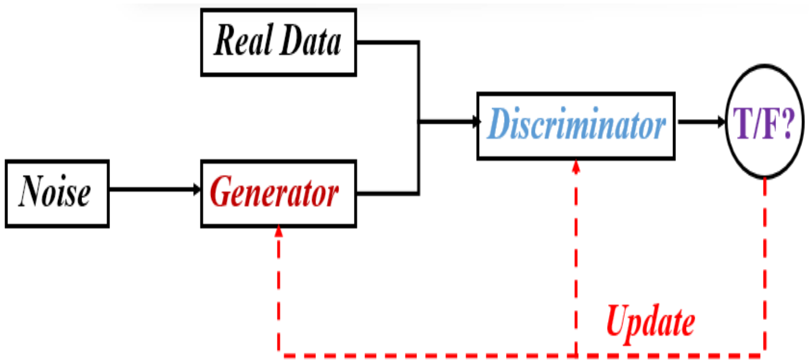
MCRec significantly outperforms CF, NN, and HIN based recommendations

Motivation of HeGAN

Negative samples are a basic function in HG embedding

Adversarial learning can provide better samples





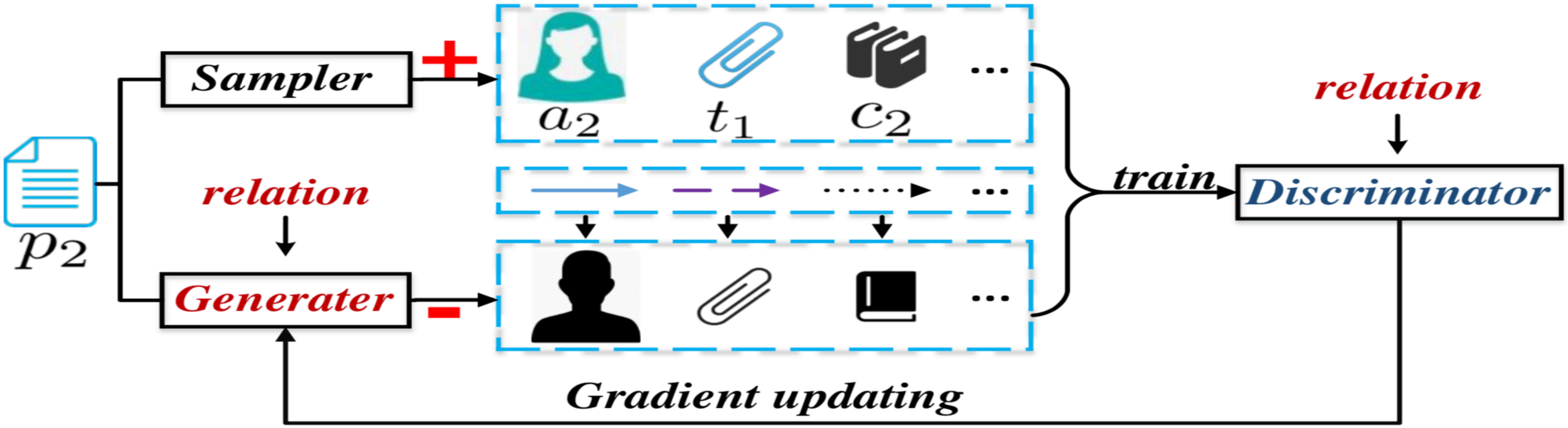
Two Challenges

- How to capture the semantics of multiple types of nodes and relations?
- How to generate fake samples efficiently and effectively?

Binbin Hu, Yuan Fang, Chuan Shi. Adversarial Learning on Heterogeneous Information Networks. KDD 2019.

Basic idea

HIN Embedding with GAN based Adversarial Learning (HeGAN)



Relation-aware Generator and Discriminator

Challenge 1

(i) Discriminator tells whether a node pair is real or fake w.r.t relation
(ii) Generator produces fake node pairs that mimic real pairs w.r.t relation

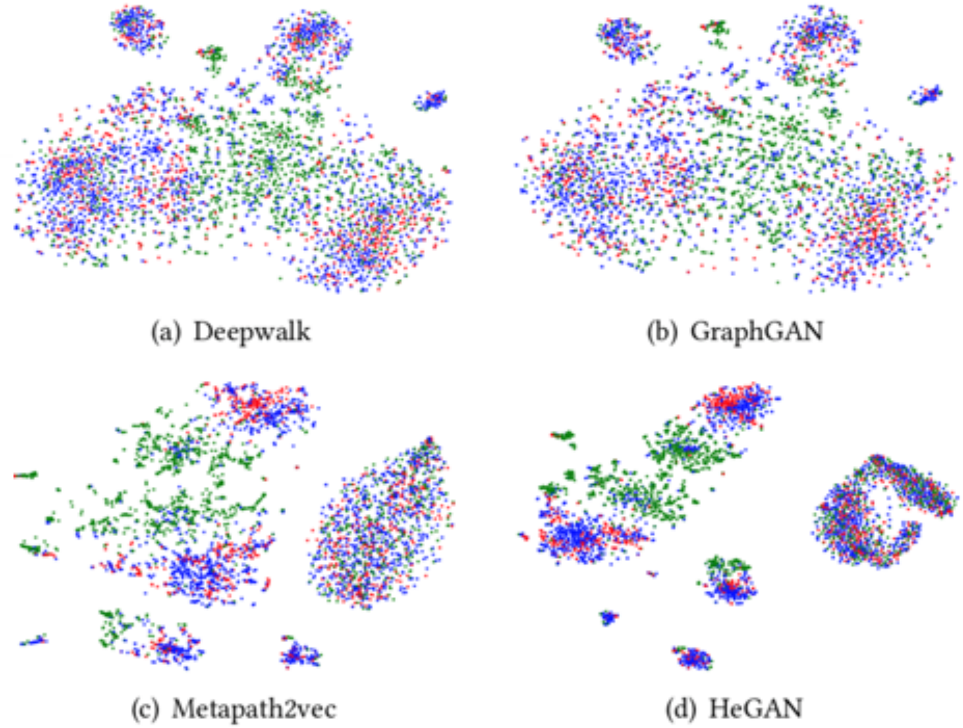
Generalized Generator

Challenge 2

(i) Sample latent nodes from a continuous distribution
(ii) no softmax computation and fake samples are not restricted to the existing nodes

Experiments

Methods	DBLP	Yelp	AMiner
Deepwalk	0.7398	0.3306	<u>0.4773</u>
LINE-1st	0.7412	0.3556	0.3518
LINE-2nd	0.7336	0.3560	0.2144
GraphGAN	0.7409	0.3413	-
ANE	0.7138	0.3145	0.4483
HERec-HNE	0.7274	0.3476	0.4635
HIN2vec	0.7204	0.3185	0.2812
Metapath2vec	<u>0.7675</u>	<u>0.3672</u>	0.4726
HeGAN	**0.7920****	**0.4037****	**0.5052****



(a) Deepwalk (b) GraphGAN (c) Metapath2vec (d) HeGAN

HeGAN learns semantic-preserving representations in a robust manner

HeGAN has a more crisp boundary and denser clusters

Outline

- Basic concepts
- √ Models
  - Structure-preserved HG representation
  - √ Attribute-assisted HG representation (HGNN)
    - HetGNN (KDD2019), HAN(WWW2019), MAGNN(WWW20)
    - NSHE(IJCAI20), HeCo(KDD21), PT-HGNN(KDD21)
  - Dynamic HG representation
- Applications
- Conclusion and future work

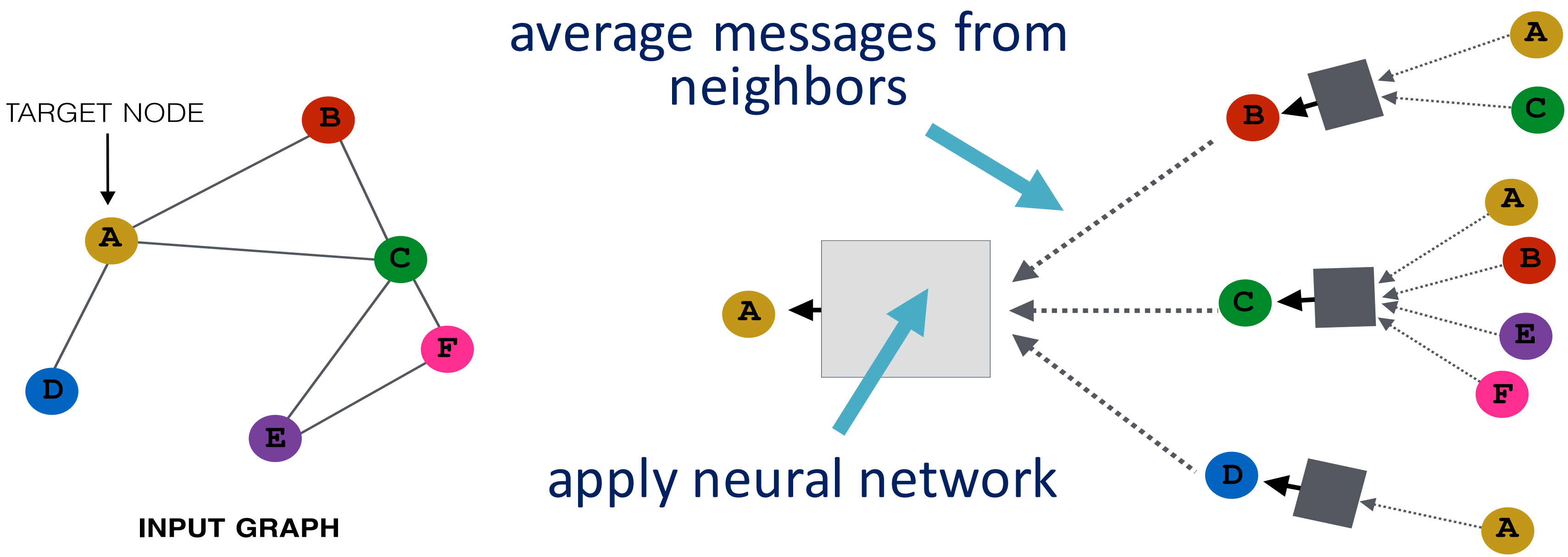
Attribute-assisted HG Representation

- Various types of nodes and links in HG bring different graph structures and rich attributes (i.e., heterogeneity).
- Attribute-assisted heterogeneous graph embedding incorporate more information beyond structure, e.g., node and edge attributes, into embedding technology, so as to utilize the neighborhood information more effectively.

Graph neural networks provide a powerful tool to fuse attributes and topology

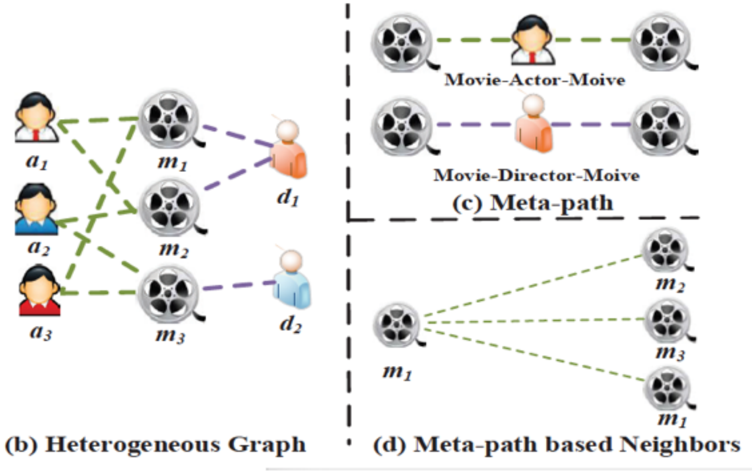
Graph Neural Network

- Graph Neural Network
  - Neural networks for processing graph-structured inputs
  - Average neighbor information with a neural network
  - Provide a powerful ability to fuse features and topology



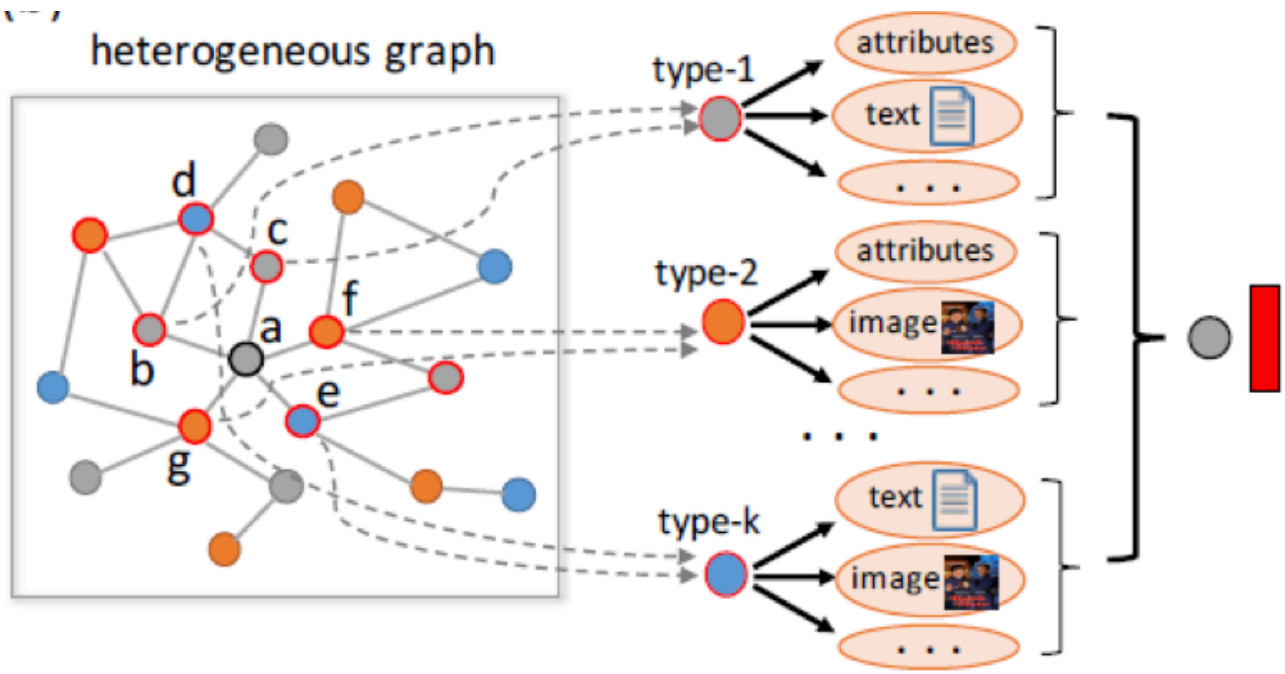
Challenges in HG Neural Network

- Current GNNs focus on homogeneous network
  - Directly fuse the attributes of neighbors to update node embeddings
- Challenges of HGNN
  - Heterogeneity. Cannot directly fuse, due to different types of nodes and attributes
  - Neighbors. Different neighbors under different meta paths.



Motivation of HetGNN

- Can we jointly consider heterogeneous structural (graph) information as well as heterogeneous contents information for node embedding effectively in HG？

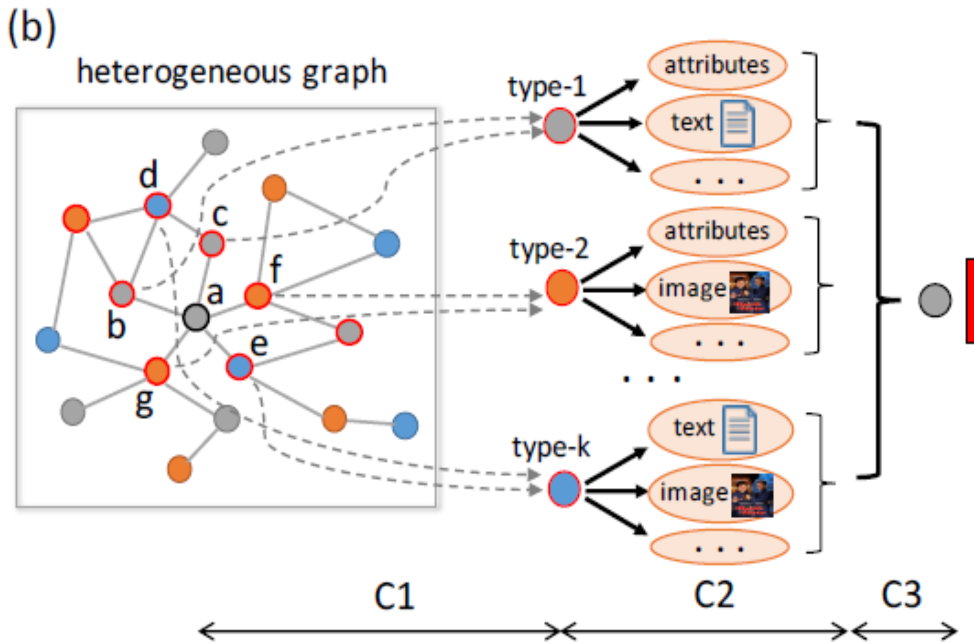


Chuxu Zhang, Dongjin Song, Chao Huang, Ananthram Swami, Nitesh V. Chawla. Heterogeneous Graph Neural Network. KDD 2019.

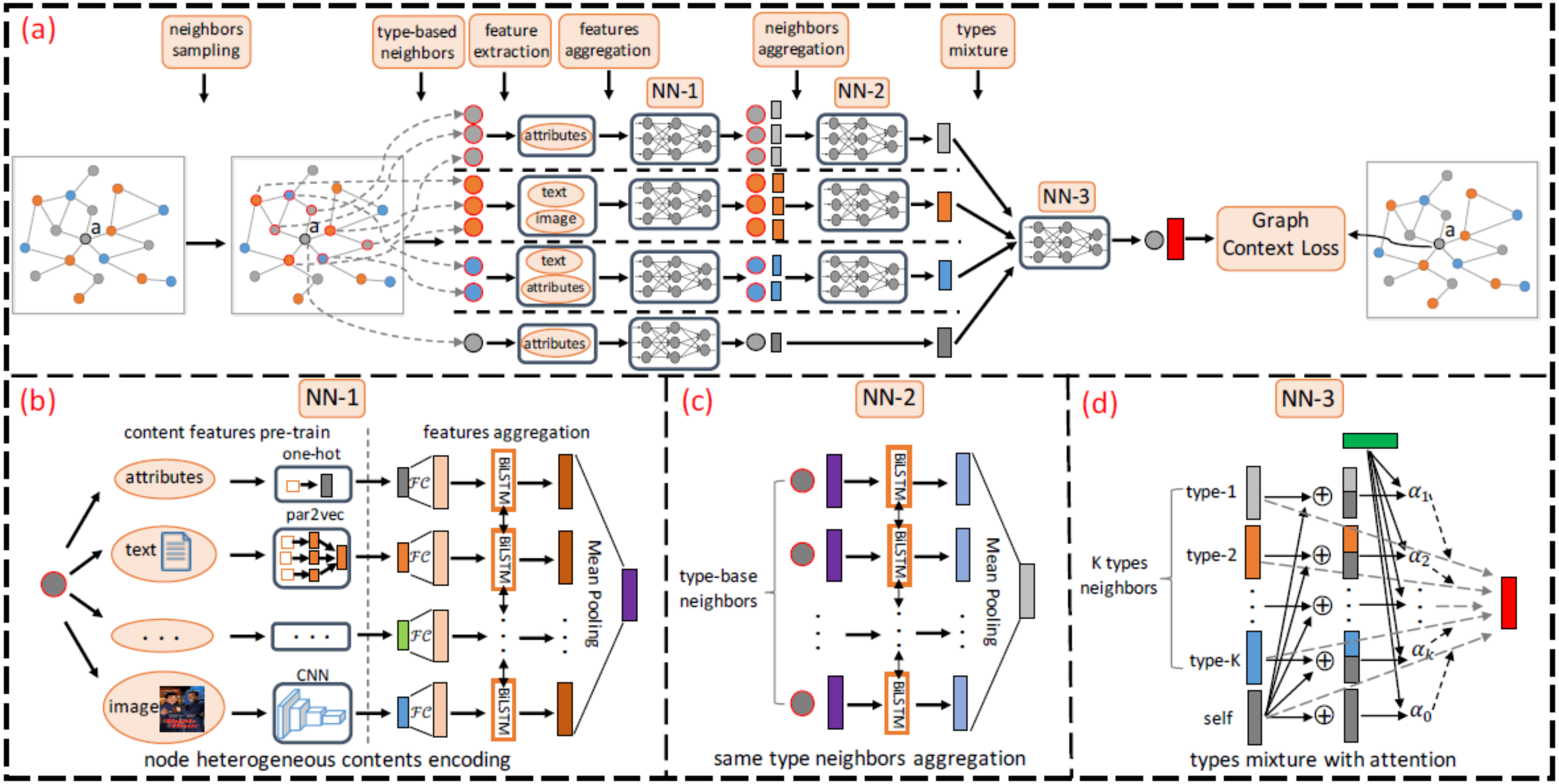
Challenges

Requirements of heterogeneous graph neural network

- C1: Sampled heterogeneous neighbors are correlated to embedding generation
- C2: Design node content encoder for content heterogeneity
- C3: Aggregate heteroge-neous neighbors by considering node types



Heterogeneous Graph Neural Network(HetGNN)



(a) The overall architecture of HetGNN (b) NN-1: node heterogeneous contents encoder; (c) NN-2: type-based neighbors aggregator; (d) NN-3: heterogeneous types combination.

Effectiveness Experiments

Link prediction task

$Data_{split}$	Metric	MP2V [4]	ASNE [15]	SHNE [34]	GSAGE [7]	GAT [31]	HetGNN
A-I_{2003} (type-1)	AUC	0.636	0.683	0.696	0.694	0.701	**0.714**
	F1	0.435	0.584	0.597	0.586	0.606	**0.620**
A-I_{2003} (type-2)	AUC	0.790	0.794	0.781	0.790	0.821	**0.837**
	F1	0.743	0.774	0.755	0.746	0.792	**0.815**
A-I_{2002} (type-1)	AUC	0.626	0.667	0.688	0.681	0.691	**0.710**
	F1	0.412	0.554	0.590	0.567	0.589	**0.615**
A-I_{2002} (type-2)	AUC	0.808	0.782	0.795	0.806	0.837	**0.851**
	F1	0.770	0.753	0.761	0.772	0.816	**0.828**
A-II_{2013} (type-1)	AUC	0.596	0.689	0.683	0.695	0.678	**0.717**
	F1	0.348	0.643	0.639	0.615	0.613	**0.669**
A-II_{2013} (type-2)	AUC	0.712	0.721	0.695	0.714	0.732	**0.767**
	F1	0.647	0.713	0.674	0.664	0.705	**0.754**
A-II_{2012} (type-1)	AUC	0.586	0.671	0.672	0.676	0.655	**0.701**
	F1	0.318	0.615	0.612	0.573	0.560	**0.642**
A-II_{2012} (type-2)	AUC	0.724	0.726	0.706	0.739	0.750	**0.775**
	F1	0.664	0.737	0.692	0.706	0.715	**0.757**
R-$I_{5:5}$	AUC	0.634	0.623	0.651	0.661	0.683	**0.749**
	F1	0.445	0.551	0.586	0.542	0.665	**0.735**
R-$I_{7:3}$	AUC	0.701	0.656	0.695	0.716	0.706	**0.787**
	F1	0.595	0.613	0.660	0.688	0.702	**0.776**
R-$II_{5:5}$	AUC	0.678	0.655	0.685	0.677	0.712	**0.736**
	F1	0.541	0.582	0.593	0.565	0.659	**0.701**
R-$II_{7:3}$	AUC	0.737	0.695	0.728	0.721	0.742	**0.772**
	F1	0.660	0.648	0.685	0.653	0.713	**0.749**

Effectiveness Experiments

Recommendation

$Data_{split}$	Metric	MP2V [4]	ASNE [15]	SHNE [34]	GSAGE [7]	GAT [31]	HetGNN
$A\text{-}I_{2003}$	Rec	0.158	0.201	0.298	0.263	0.275	**0.319**
	Pre	0.044	0.060	0.081	0.077	0.079	**0.094**
	F1	0.069	0.092	0.127	0.120	0.123	**0.145**
$A\text{-}I_{2002}$	Rec	0.144	0.152	0.279	0.231	0.274	**0.293**
	Pre	0.046	0.050	0.086	0.073	0.087	**0.093**
	F1	0.070	0.075	0.134	0.112	0.132	**0.141**
$A\text{-}II_{2013}$	Rec	0.516	0.419	0.608	0.540	0.568	**0.625**
	Pre	0.207	0.174	0.241	0.219	0.230	**0.252**
	F1	0.295	0.333	0.345	0.312	0.327	**0.359**
$A\text{-}II_{2012}$	Rec	0.468	0.382	0.552	0.512	0.518	**0.606**
	Pre	0.204	0.171	0.233	0.224	0.227	**0.264**
	F1	0.284	0.236	0.327	0.312	0.316	**0.368**

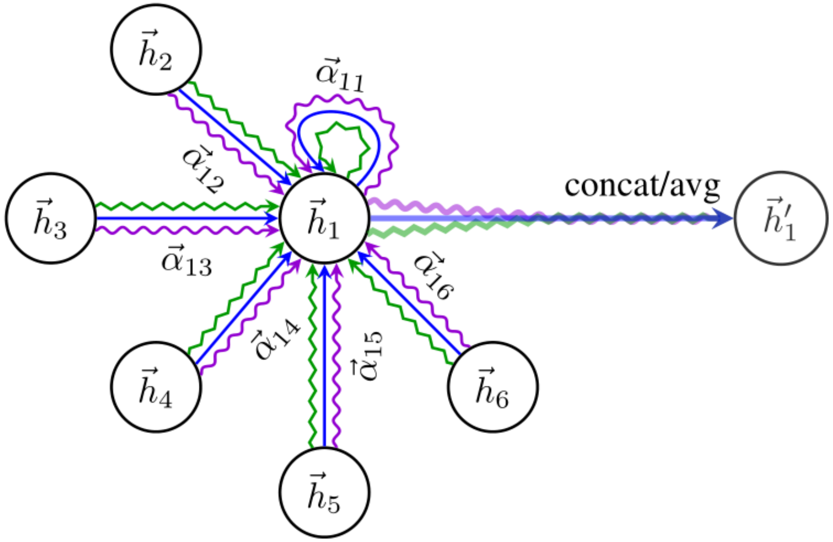
Multi-label Classification

Task	Metric	MP2V [4]	ASNE [15]	SHNE [34]	GSAGE [7]	GAT [31]	HetGNN
MC (10%)	Macro-F1	0.972	0.965	0.939	**0.978**	0.962	**0.978**
	Micro-F1	0.973	0.967	0.940	0.978	0.963	**0.979**
MC (30%)	Macro-F1	0.975	0.969	0.939	0.979	0.965	**0.981**
	Micro-F1	0.975	0.970	0.941	0.980	0.965	**0.982**
NC	NMI	0.894	0.854	0.776	**0.914**	0.845	0.901
	ARI	0.933	0.898	0.813	**0.945**	0.882	0.932

Inductive Multi-label Classification

Task	Metric	GSAGE [7]	GAT [31]	HetGNN
IMC (10%)	Macro-F1	0.938	0.954	**0.962**
	Micro-F1	0.945	0.958	**0.965**
IMC (30%)	Macro-F1	0.949	0.956	**0.964**
	Micro-F1	0.955	0.960	**0.968**
INC	NMI	0.714	0.765	**0.840**
	ARI	0.764	0.803	**0.894**

Motivation of HAN



A dog is standing on a hardwood floor.

GAT
for Homogeneous graph
(Petar et al, ICLR2018)

Attention Mechanism

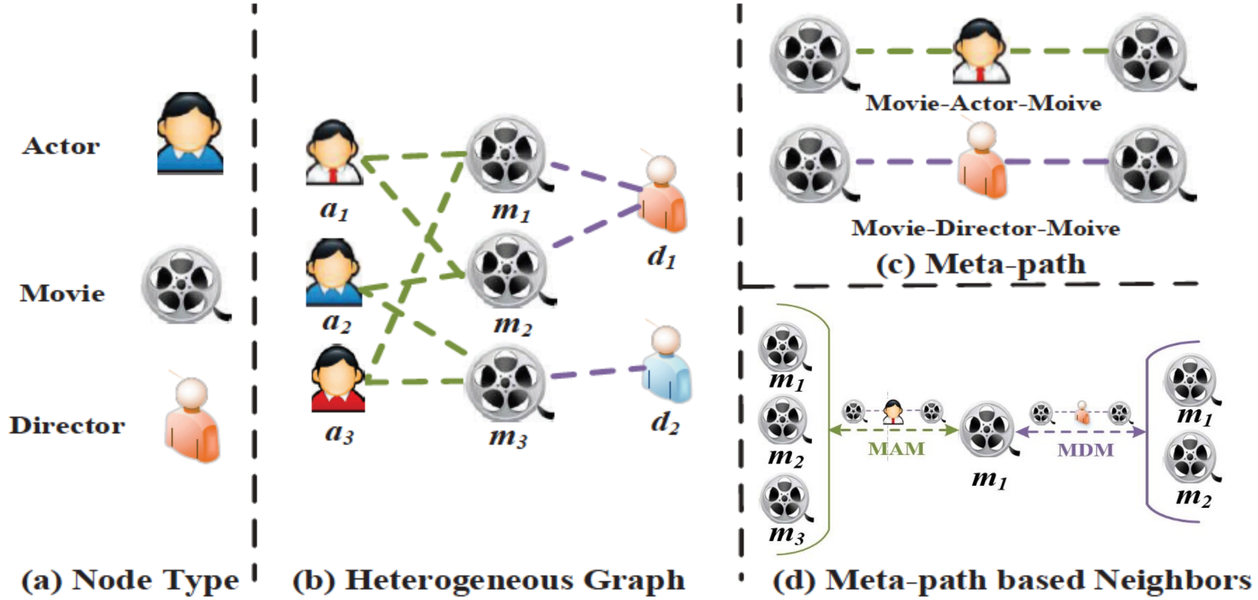
Can we design attention mechanism for heterogeneous graph?

Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Peng Cui, Philip S. Yu, Yanfang Ye. Heterogeneous Graph Attention Network. WWW 2019.

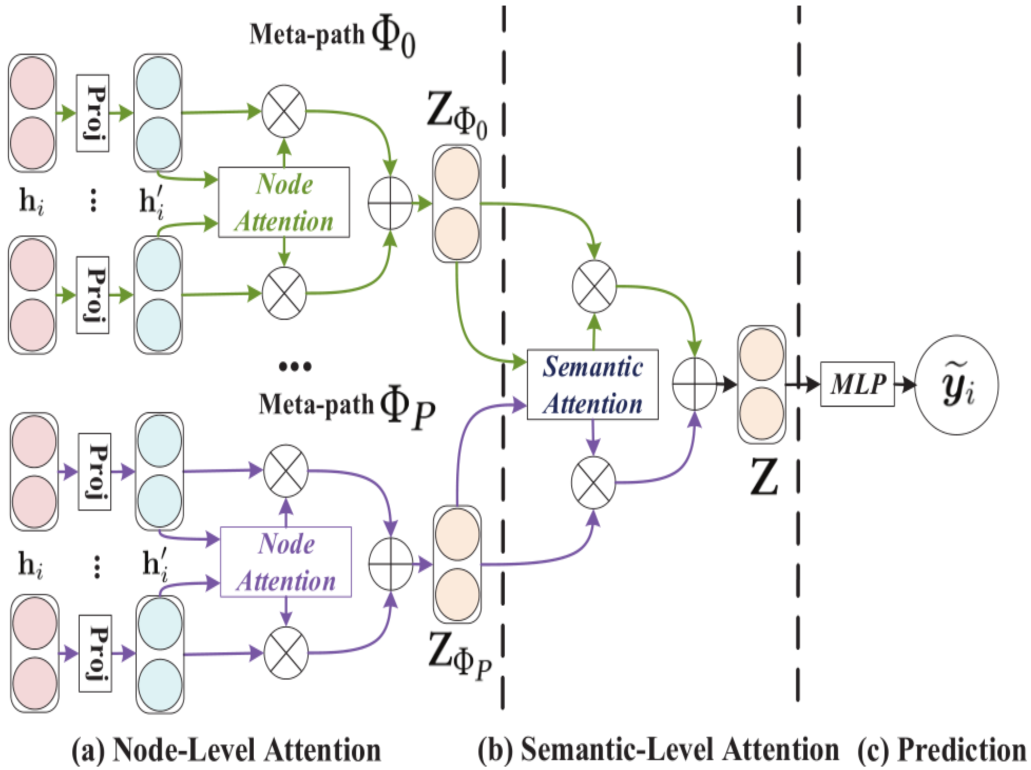
Challenges

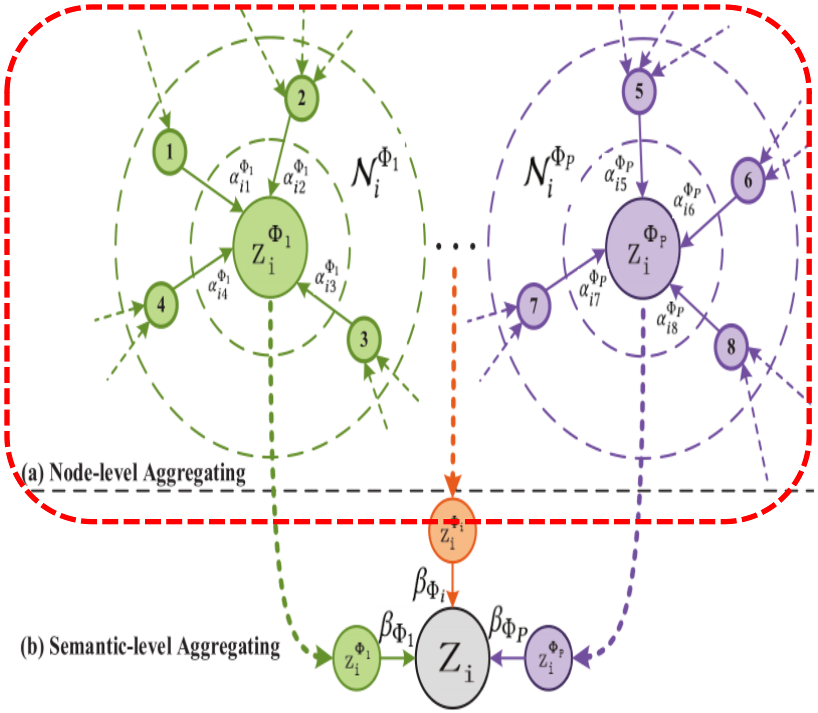
Requirements of Heterogeneous Graph Neural Network

- Heterogeneity of graph
- Node-level attention
- Semantic-level attention



Heterogeneous Graph Attention Network(HAN)





Classification

Datasets	Metrics	Training	DeepWalk	ESim	metapath2vec	HERec	GCN	GAT	HAN_{nd}	HAN_{sem}	HAN
ACM	Macro-F1	20%	77.25	77.32	65.09	66.17	86.81	86.23	88.15	89.04	**89.40**
		40%	80.47	80.12	69.93	70.89	87.68	87.04	88.41	89.41	**89.79**
		60%	82.55	82.44	71.47	72.38	88.10	87.56	87.91	**90.00**	89.51
		80%	84.17	83.00	73.81	73.92	88.29	87.33	88.48	90.17	**90.63**
	Micro-F1	20%	76.92	76.89	65.00	66.03	86.77	86.01	87.99	88.85	**89.22**
		40%	79.99	79.70	69.75	70.73	87.64	86.79	88.31	89.27	**89.64**
		60%	82.11	82.02	71.29	72.24	88.12	87.40	87.68	**89.85**	89.33
		80%	83.88	82.89	73.69	73.84	88.35	87.11	88.26	89.95	**90.54**
DBLP	Macro-F1	20%	77.43	91.64	90.16	91.68	90.79	90.97	91.17	92.03	**92.24**
		40%	81.02	92.04	90.82	92.16	91.48	91.20	91.46	92.08	**92.40**
		60%	83.67	92.44	91.32	92.80	91.89	90.80	91.78	92.38	**92.80**
		80%	84.81	92.53	91.89	92.34	92.38	91.73	91.80	92.53	**93.08**
	Micro-F1	20%	79.37	92.73	91.53	92.69	91.71	91.96	92.05	92.99	**93.11**
		40%	82.73	93.07	92.03	93.18	92.31	92.16	92.38	93.00	**93.30**
		60%	85.27	93.39	92.48	93.70	92.62	91.84	92.69	93.31	**93.70**
		80%	86.26	93.44	92.80	93.27	93.09	92.55	92.69	93.29	**93.99**
IMDB	Macro-F1	20%	40.72	32.10	41.16	41.65	45.73	49.44	49.78	**50.87**	50.00
		40%	45.19	31.94	44.22	43.86	48.01	50.64	52.11	50.85	**52.71**
		60%	48.13	31.68	45.11	46.27	49.15	51.90	51.73	52.09	**54.24**
		80%	50.35	32.06	45.15	47.64	51.81	52.99	52.66	51.60	**54.38**
	Micro-F1	20%	46.38	35.28	45.65	45.81	49.78	55.28	54.17	55.01	**55.73**
		40%	49.99	35.47	48.24	47.59	51.71	55.91	56.39	55.15	**57.97**
		60%	52.21	35.64	49.09	49.88	52.29	56.44	56.09	56.66	**58.32**
		80%	54.33	35.59	48.81	50.99	54.61	56.97	56.38	56.49	**58.51**

Clustering & Visualization

Datasets	Metrics	DeepWalk	ESim	metapath2vec	HERec	GCN	GAT	HAN_{nd}	HAN_{sem}	HAN
ACM	NMI	41.61	39.14	21.22	40.70	51.40	57.29	60.99	61.05	**61.56**
	ARI	35.10	34.32	21.00	37.13	53.01	60.43	61.48	59.45	**64.39**
DBLP	NMI	76.53	66.32	74.30	76.73	75.01	71.50	75.30	77.31	**79.12**
	ARI	81.35	68.31	78.50	80.98	80.49	77.26	81.46	83.46	**84.76**
IMDB	NMI	1.45	0.55	1.20	1.20	5.45	8.45	9.16	10.31	**10.87**
	ARI	2.15	0.10	1.70	1.65	4.40	7.46	7.98	9.51	**10.01**

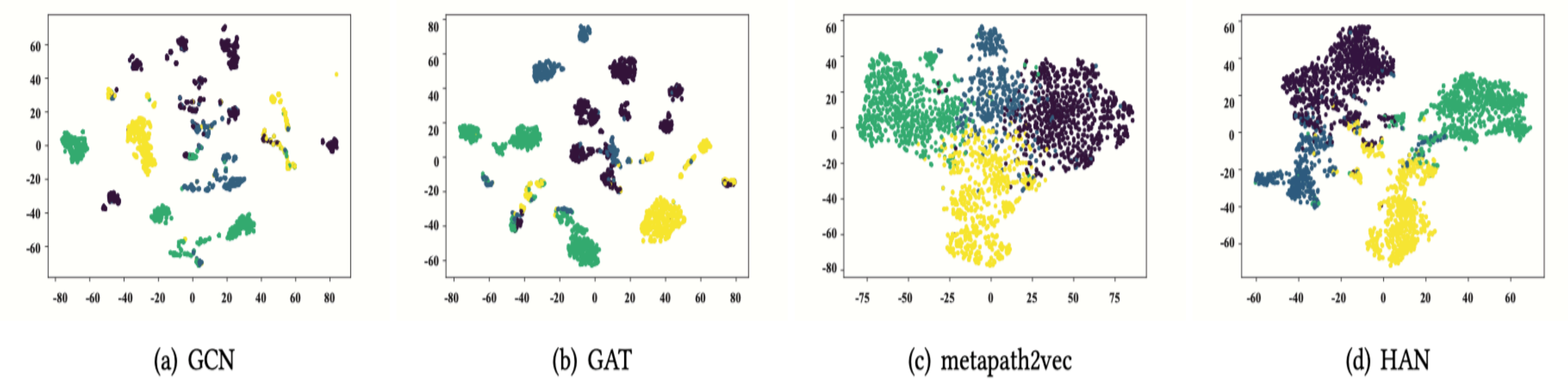


Figure 6: Visualization embedding on DBLP. Each point indicates one author and its color indicates the research area.

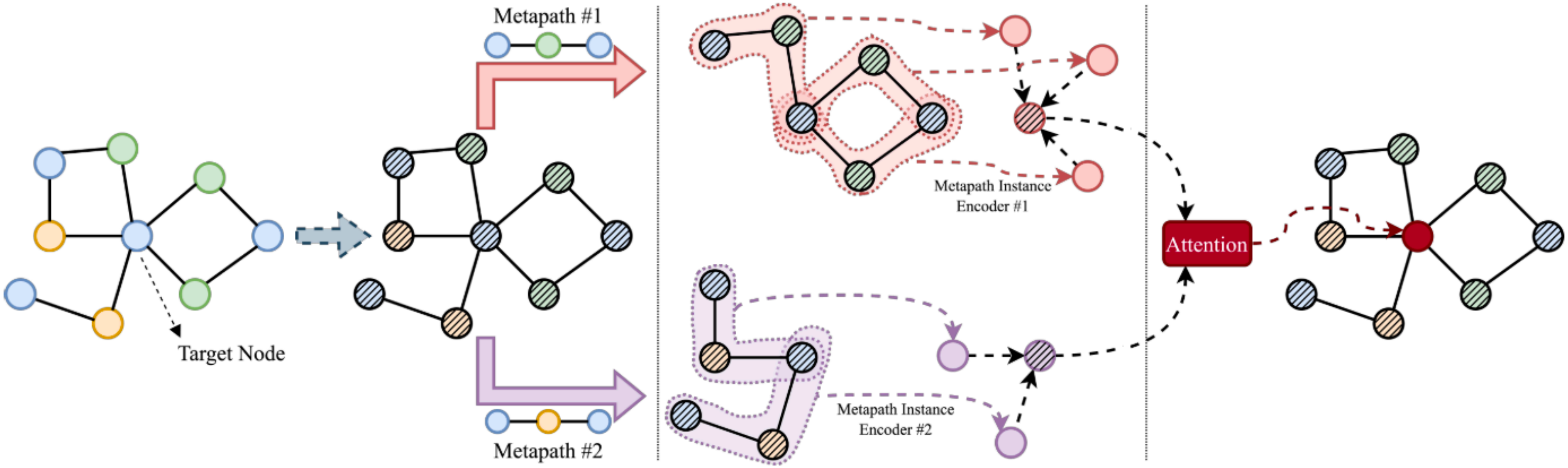
Motivation of MAGNN

Exsiting metapath-based HGNN:

Limitions		Solutions
• Does not leverage node content features	➡	• Node content transformation
• Discards intermediate nodes along the metapath	➡	• Intra-metapath aggregation
• Relies on a single metapath	➡	• Inter-metapath aggregation

Xinyu Fu, Jiani Zhang, Ziqiao Meng, Irwin King. MAGNN: Metapath Aggregated Graph Neural Network for Heterogeneous Graph Embedding. WWW 2020.

MAGNN



(a) Node Content Transformation

$$\mathbf{h}'_v = \mathbf{W}_A \cdot \mathbf{x}_v^A$$

(b) Intra-metapath Aggregation

Metapath Instance Encoder:

$$\mathbf{h}_{P(v,u)} = f_\theta\left(P(v,u)\right) = f_\theta\left(\mathbf{h}'_v, \mathbf{h}'_u, \left\{\mathbf{h}'_t, \forall t \in \{m^{P(v,u)}\}\right\}\right)$$

Metapath Instance-Level Attention:

$$e_{vu}^P = \text{LeakyReLU}\left(\mathbf{a}_P^\top \cdot \left[\mathbf{h}'_v \| \mathbf{h}_{P(v,u)}\right]\right)$$

$$\alpha_{vu}^P = \frac{\exp\left(e_{vu}^P\right)}{\sum_{s \in \mathcal{N}_v^P} \exp\left(e_{vs}^P\right)},$$

$$\mathbf{h}_v^P = \sigma\left(\sum_{u \in \mathcal{N}_v^P} \alpha_{vu}^P \cdot \mathbf{h}_{P(v,u)}\right).$$

Multi Head Attention:

$$\mathbf{h}_v^P = \Big\Vert_{k=1}^{K} \sigma\left(\sum_{u \in \mathcal{N}_v^P} \left[\alpha_{vu}^P\right]_k \cdot \mathbf{h}_{P(v,u)}\right).$$

(c) Inter-metapath Aggregation

Metapath-Level Attention:

$$\mathbf{s}_{P_i} = \frac{1}{|\mathcal{V}_A|} \sum_{v \in \mathcal{V}_A} \tanh\left(\mathbf{M}_A \cdot \mathbf{h}_v^{P_i} + \mathbf{b}_A\right)$$

$$e_{P_i} = \mathbf{q}_A^\top \cdot \mathbf{s}_{P_i},$$

$$\beta_{P_i} = \frac{\exp\left(e_{P_i}\right)}{\sum_{P \in \mathcal{P}_A} \exp\left(e_P\right)},$$

$$\mathbf{h}_v^{\mathcal{P}_A} = \sum_{P \in \mathcal{P}_A} \beta_P \cdot \mathbf{h}_v^P,$$

$$\mathbf{h}_v = \sigma\left(\mathbf{W}_o \cdot \mathbf{h}_v^{\mathcal{P}_A}\right)$$

Experimental Results

- Node Classification

Dataset	Metrics	Train %	Unsupervised					Semi-supervised			
			LINE	node2vec	ESim	metapath2vec	HERec	GCN	GAT	HAN	MAGNN
IMDb	Macro-F1	20%	44.04	49.00	48.37	46.05	45.61	52.73	53.64	56.19	**59.35**
		40%	45.45	50.63	50.09	47.57	46.80	53.67	55.50	56.15	**60.27**
		60%	47.09	51.65	51.45	48.17	46.84	54.24	56.46	57.29	**60.66**
		80%	47.49	51.49	51.37	49.99	47.73	54.77	57.43	58.51	**61.44**
	Micro-F1	20%	45.21	49.94	49.32	47.22	46.23	52.80	53.64	56.32	**59.60**
		40%	46.92	51.77	51.21	48.17	47.89	53.76	55.56	57.32	**60.50**
		60%	48.35	52.79	52.53	49.87	48.19	54.23	56.47	58.42	**60.88**
		80%	48.98	52.72	52.54	50.50	49.11	54.63	57.40	59.24	**61.53**
DBLP	Macro-F1	20%	87.16	86.70	90.68	88.47	90.82	88.00	91.05	91.69	**93.13**
		40%	88.85	88.07	91.61	89.91	91.44	89.00	91.24	91.96	**93.23**
		60%	88.93	88.69	91.84	90.50	92.08	89.43	91.42	92.14	**93.57**
		80%	89.51	88.93	92.27	90.86	92.25	89.98	91.73	92.50	**94.10**
	Micro-F1	20%	87.68	87.21	91.21	89.02	91.49	88.51	91.61	92.33	**93.61**
		40%	89.25	88.51	92.05	90.36	92.05	89.22	91.77	92.57	**93.68**
		60%	89.34	89.09	92.28	90.94	92.66	89.57	91.97	92.72	**93.99**
		80%	89.96	89.37	92.68	91.31	92.78	90.33	92.24	93.23	**94.47**

Experimental Results

- Node Clustering

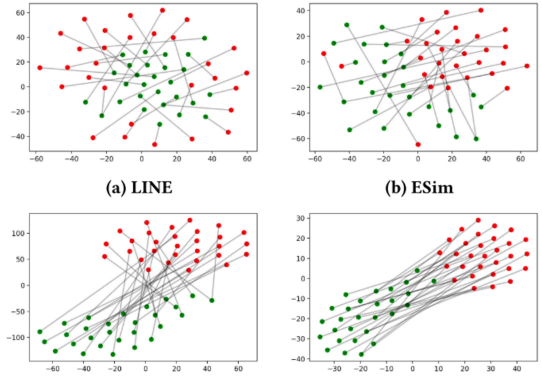
Dataset	Metrics	Unsupervised					Semi-supervised			
		LINE	node2vec	ESim	metapath2vec	HERec	GCN	GAT	HAN	MAGNN
IMDb	NMI	1.13	5.22	1.07	0.89	0.39	7.46	7.84	10.79	**15.58**
	ARI	1.20	6.02	1.01	0.22	0.11	7.69	8.87	11.11	**16.74**
DBLP	NMI	71.02	77.01	68.33	74.18	69.03	73.45	70.73	77.49	**80.81**
	ARI	76.52	81.37	72.22	78.11	72.45	77.50	76.04	82.95	**85.54**

- Link Prediction

Dataset	Metrics	LINE	node2vec	ESim	metapath2vec	HERec	GCN	GAT	GATNE	HAN	MAGNN
Last.fm	AUC	85.76	67.14	82.00	92.20	91.52	90.97	92.36	89.21	93.40	**98.91**
	AP	88.07	64.11	82.19	90.11	89.47	91.65	91.55	88.86	92.44	**98.93**

- Visualization

  - *randomly select 30 user-artist pairs from the positive testing set of the Last.fm dataset*



(a) LINE (b) ESim (c) GCN (d) MAGNN

Motivation of NSHE

Existing HG embeddings are mostly based on meta-path (MP)

- Domain knowledge to select MPs
- Supervision to fuse different semantics

HGNN without Meta-Path?

Network Schema based HINE

- Unique to each HG
- Provide a uniform embedding
- Preserve the overall semantics

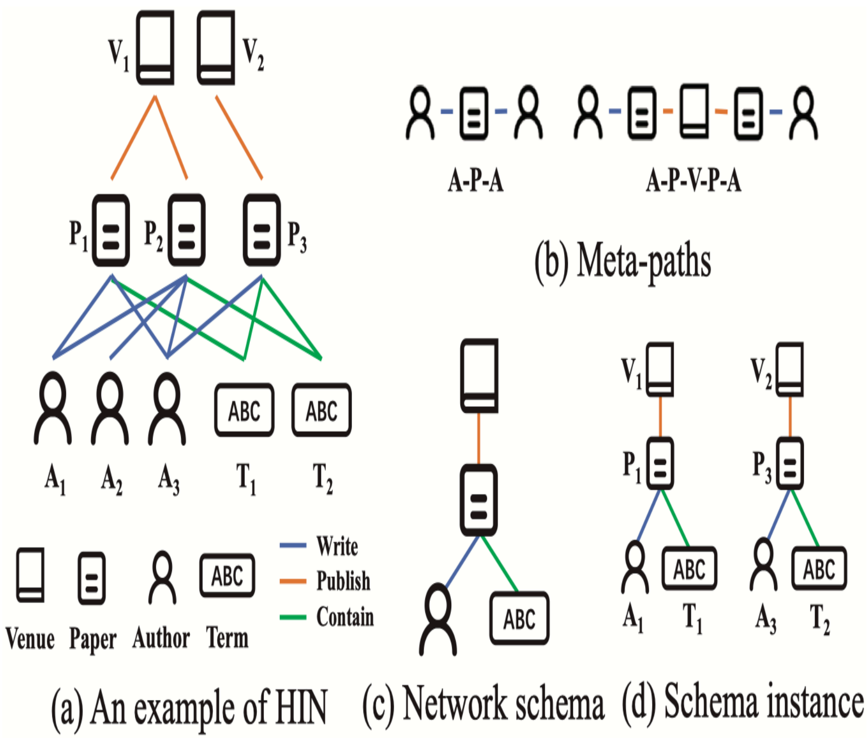
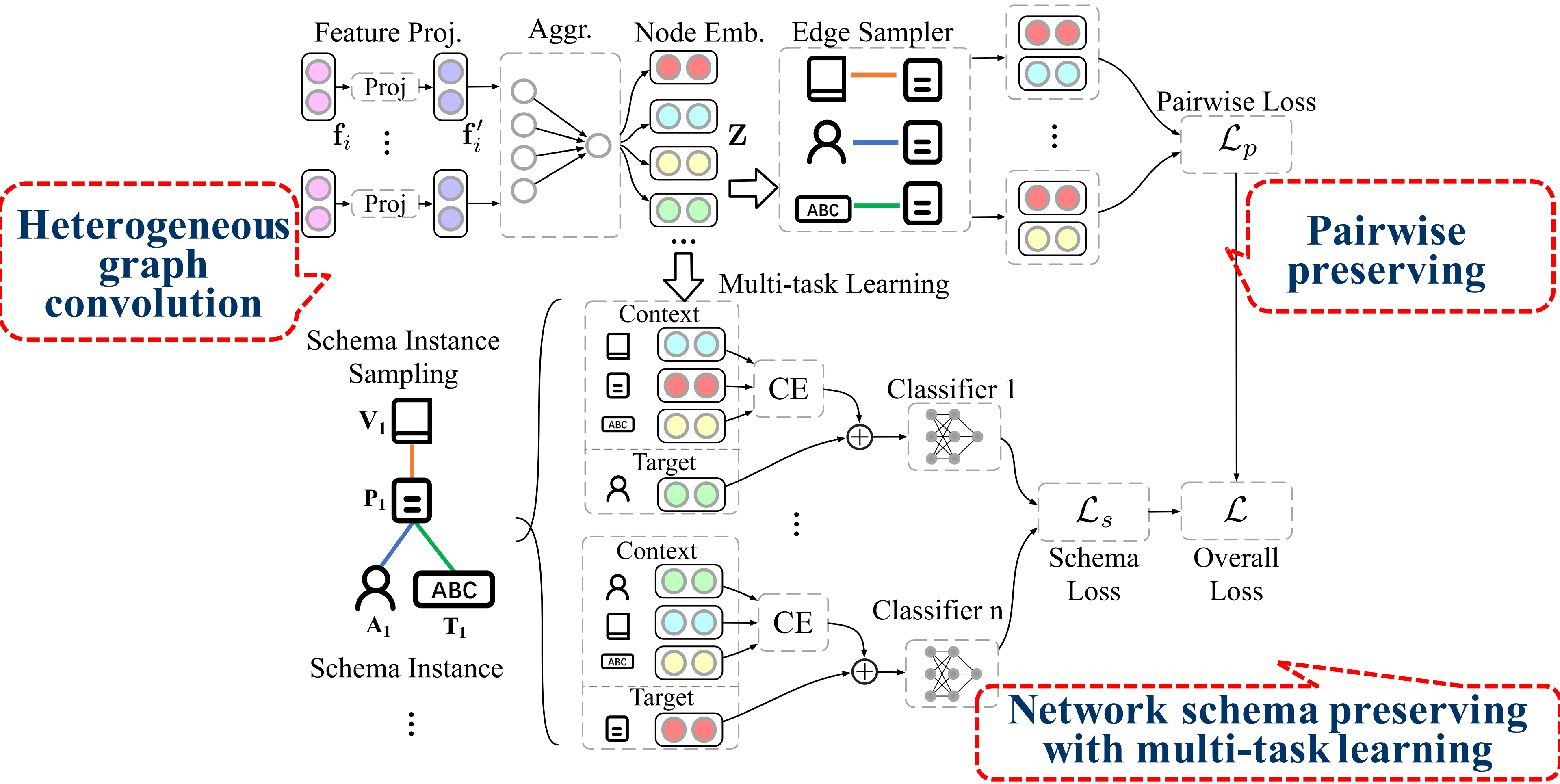


Figure 1: A toy example of an HIN on bibliographic data.

Jianan Zhao, Xiao Wang, Chuan Shi, Zekuan Liu, and Yanfang Ye. Network Schema Preserving Heterogeneous Information Network Embedding. IJCAI 2020

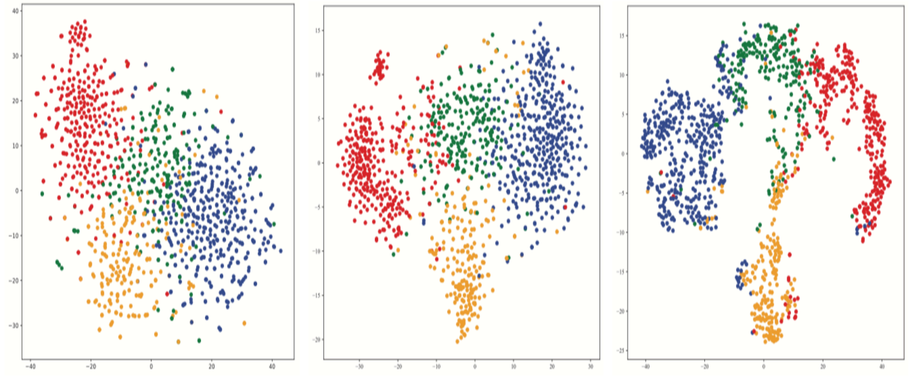
Framework of NSHE



Experiments

	DBLP-P		DBLP-A		IMDB		ACM	
	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1
DeepWalk	90.12	89.45	89.44	88.48	56.52	55.24	82.17	81.82
LINE-1st	81.43	80.74	82.32	80.20	43.75	39.87	82.46	82.35
LINE-2nd	84.76	83.45	88.76	87.35	40.54	33.06	82.21	81.32
DHNE	85.71	84.67	73.30	67.61	38.99	30.53	65.27	62.31
Metapath2Vec	92.86	92.44	89.36	87.95	51.90	50.21	83.61	82.77
HIN2Vec	83.81	83.85	90.30	89.46	48.02	46.24	54.30	48.59
HERec	90.47	87.50	86.21	84.55	54.48	53.46	81.89	81.74
HeGAN	88.79	83.81	90.48	89.27	58.56	57.12	83.09	82.94
NSHE	**95.24**	**94.76**	**93.10**	**92.37**	**59.21**	**58.35**	**84.12**	**83.27**

	DBLP-P	DBLP-A	IMDB	ACM
DeepWalk	46.75	66.25	0.41	**48.81**
LINE-1st	42.18	29.98	0.03	37.75
LINE-2nd	46.83	61.11	0.03	41.80
DHNE	35.33	21.00	0.05	20.25
Metapath2Vec	56.89	68.74	0.09	42.71
HIN2Vec	30.47	65.79	0.04	42.28
HERec	39.46	24.09	0.51	40.70
HeGAN	60.78	68.95	6.56	43.35
NSHE	**65.54**	**69.52**	**7.58**	44.32

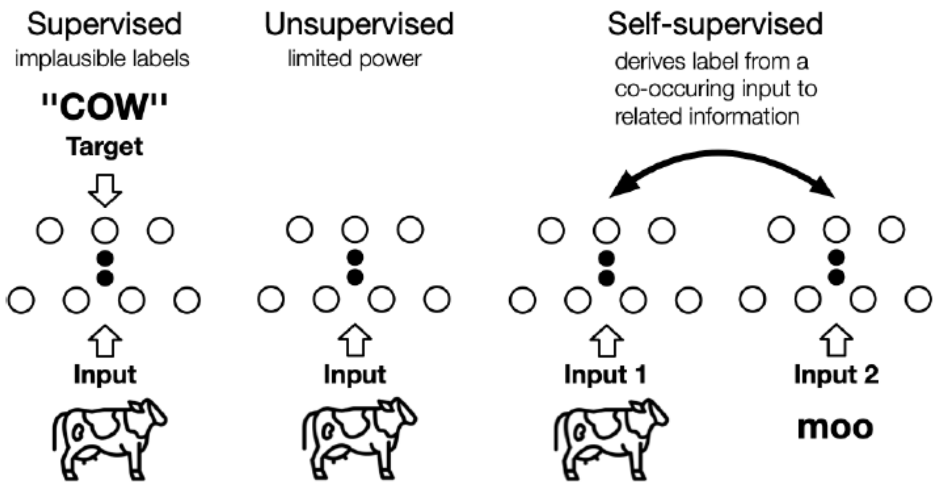


(a) DeepWalk (b) Metapath2Vec (c) NSHE

Self-supervised GNN

- Most HGNNs belong to the semi-supervised settings, while it’s very expensive to obtain enough labels.

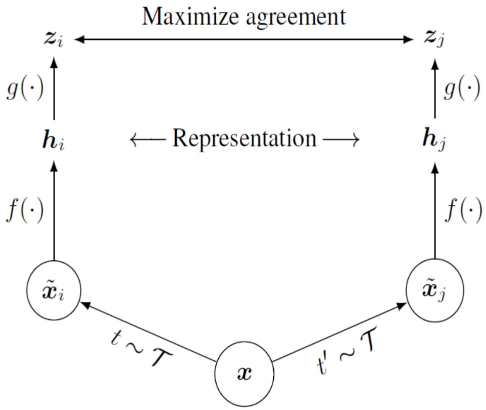
- A promising solution: Self-supervised Learning
  - Spontaneously find supervised signals from the data itself.



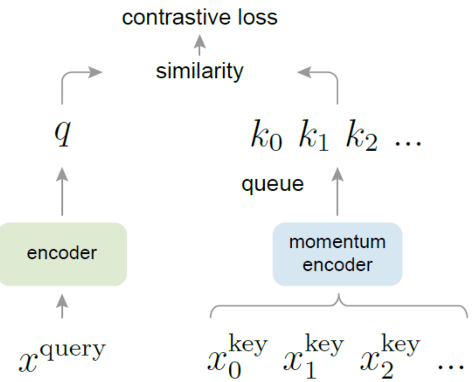
Xiao Wang, Nian Liu, Hui Han, Chuan Shi. Self-supervised Heterogeneous Graph Neural Network with Co-contrastive Learning. KDD 2021.

Motivation of HeCo

- Typical supervised learning: Contrastive learning
  - Push forward positives, push away negatives.
  - It can learn discriminative embeddings even without labels.
  - It has been widely used in CV and NLP, but not in HG.



SimCLR



MoCo

How to conduct heterogeneous contrastive learning on a HG?

YaChen T, Kornblith S, Norouzi M, et al. A simple framework for contrastive learning of visual representations. ICML, 2020.
He K, Fan H, Wu Y, et al. Momentum contrast for unsupervised visual representation learning. CVPR, 2020.

Challenges

- **Three fundamental challenges**
  - **C1: How to design a heterogeneous contrastive mechanism**
    - complex structures, multiple views → view-invariant factors cross views
  - **C2: How to select proper views in a HG**
    - cover both of the local and high-order structure

      network schema **meta-path**
  - **C3: How to set a difficult contrastive task**
    - too similar views → too weak signals
    - information diversity or harder negative samples

Self-supervised Heterogeneous Graph Neural Network with Co-contrastive Learning (HeCo)

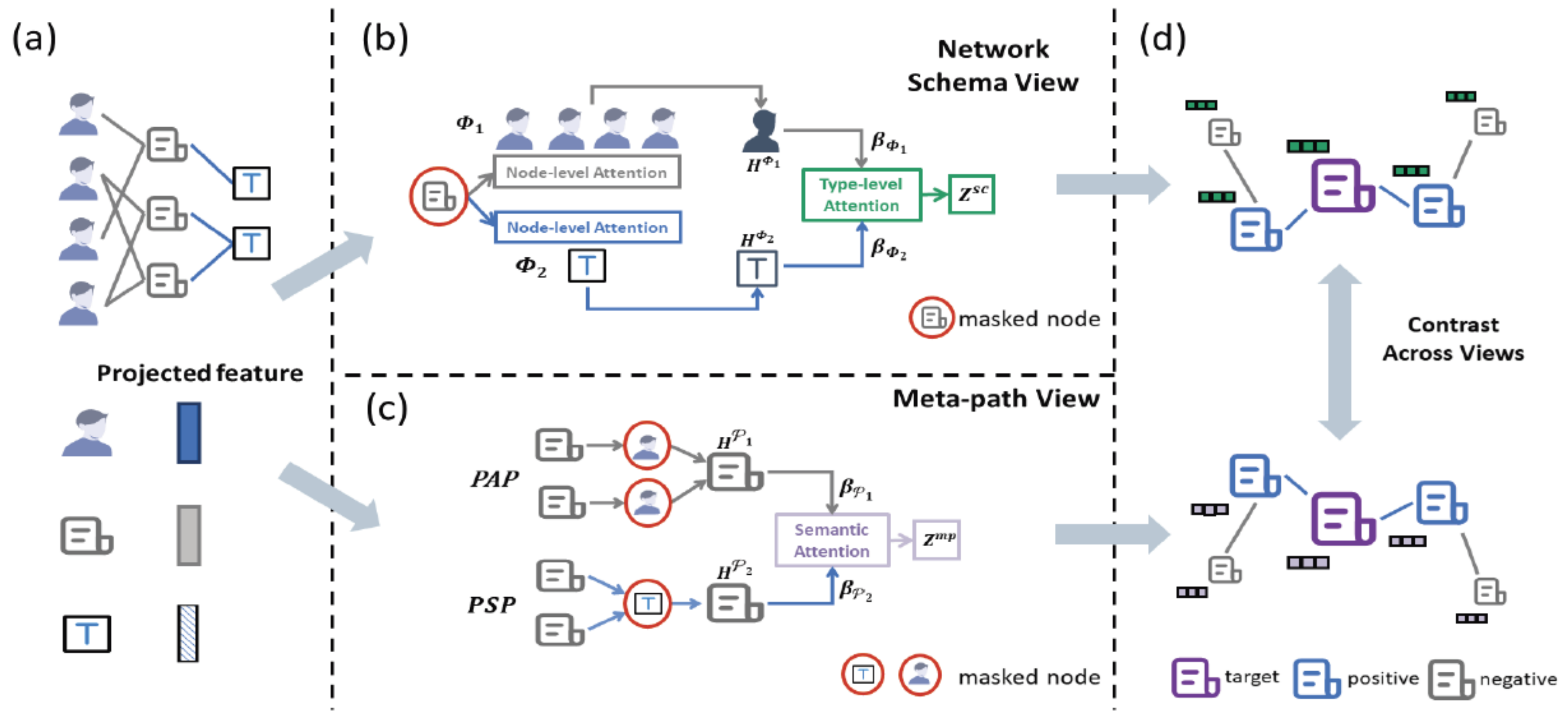
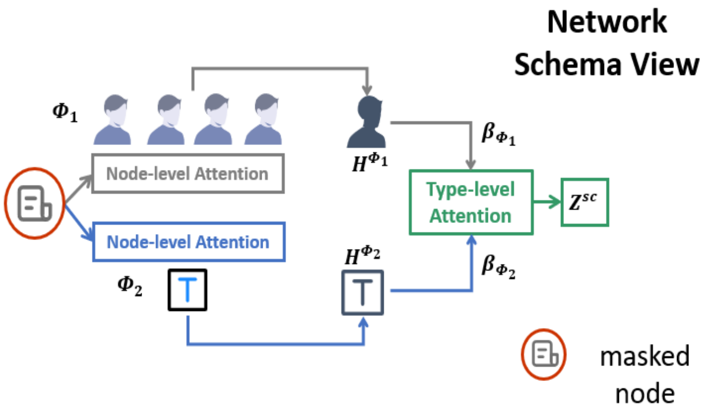


Figure 2: The overall architecture of our proposed HeCo.

(b) Network Schema View Guided Encoder

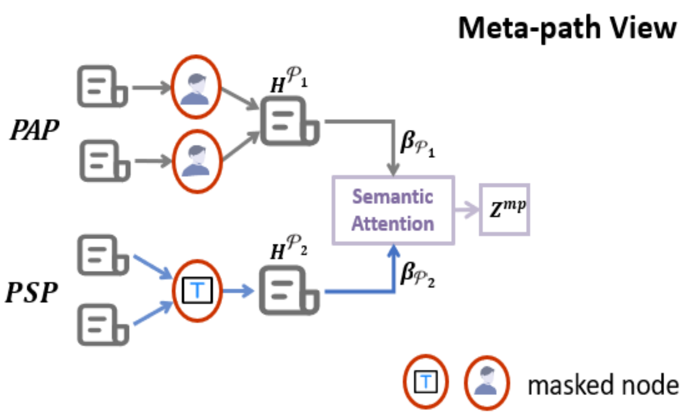


- **Node-level attention**
  randomly sample different type of neighbors, and then aggregate them
- **Type-level attention**

$$w_{\Phi_m} = \frac{1}{|V|} \sum_{i \in V} \mathbf{a}_{sc}^{\top} \cdot \tanh\left(\mathbf{W}_{sc} h_i^{\Phi_m} + \mathbf{b}_{sc}\right),$$

$$\beta_{\Phi_m} = \frac{\exp\left(w_{\Phi_m}\right)}{\sum_{i=1}^{S} \exp\left(w_{\Phi_i}\right)}, \qquad z_i^{sc} = \sum_{m=1}^{S} \beta_{\Phi_m} \cdot h_i^{\Phi_m}.$$

(c) Meta-path View Guided Encoder

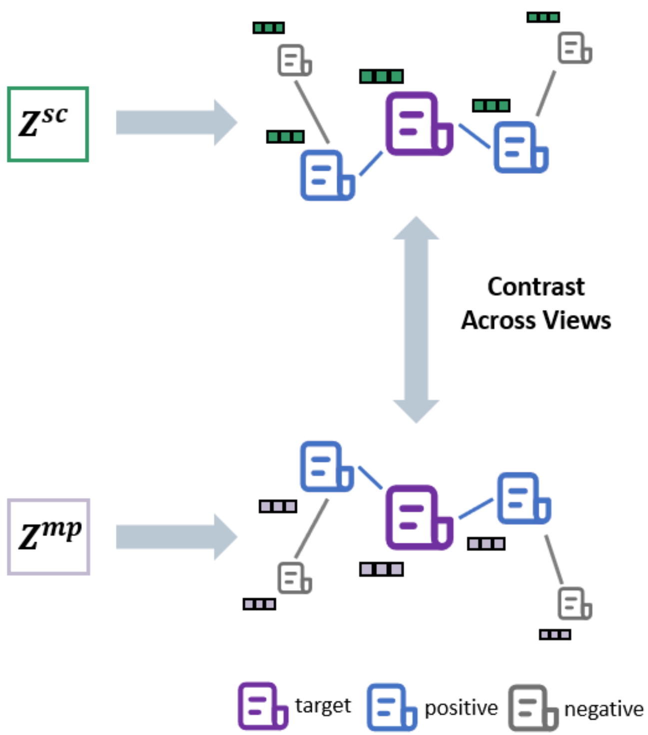


- **Meta-path specific GCN**
  find meta-path based neighbors, and fuse them
- **Semantic-level aggregation**

$$w_{\mathcal{P}_n} = \frac{1}{|V|} \sum_{i \in V} \mathbf{a}_{mp}^{\top} \cdot \tanh\left(\mathbf{W}_{mp} h_i^{\mathcal{P}_n} + \mathbf{b}_{mp}\right),$$

$$\beta_{\mathcal{P}_n} = \frac{\exp\left(w_{\mathcal{P}_n}\right)}{\sum_{i=1}^{M} \exp\left(w_{\mathcal{P}_i}\right)}, \qquad z_i^{mp} = \sum_{n=1}^{M} \beta_{\mathcal{P}_n} \cdot h_i^{\mathcal{P}_n},$$

(d) Collaboratively Contrastive Optimization



- **How to define positives in HG?**

  Principle: If two nodes are connected by many meta-paths, they are positives.
  Method: count #meta-paths, select top k as positives, and others as negatives

- **Contrastive loss:**

$$\mathcal{L}_i^{sc} = -\log \frac{\sum_{j\in\mathbb{P}_i} exp\left(sim\left(z_i^{sc}\_proj, z_j^{mp}\_proj\right)/\tau\right)}{\sum_{k\in\{\mathbb{P}_i \bigcup \mathbb{N}_i\}} exp\left(sim\left(z_i^{sc}\_proj, z_k^{mp}\_proj\right)/\tau\right)},$$

$\mathbb{P}_i$: positive set $\mathbb{N}_i$: negative set

- **Overall objective:**

$$\mathcal{J} = \frac{1}{|V|}\sum_{i\in V}\left[\lambda\cdot\mathcal{L}_i^{sc} + (1-\lambda)\cdot\mathcal{L}_i^{mp}\right],$$

- **View mask mechanism**
  - enhance information diversity

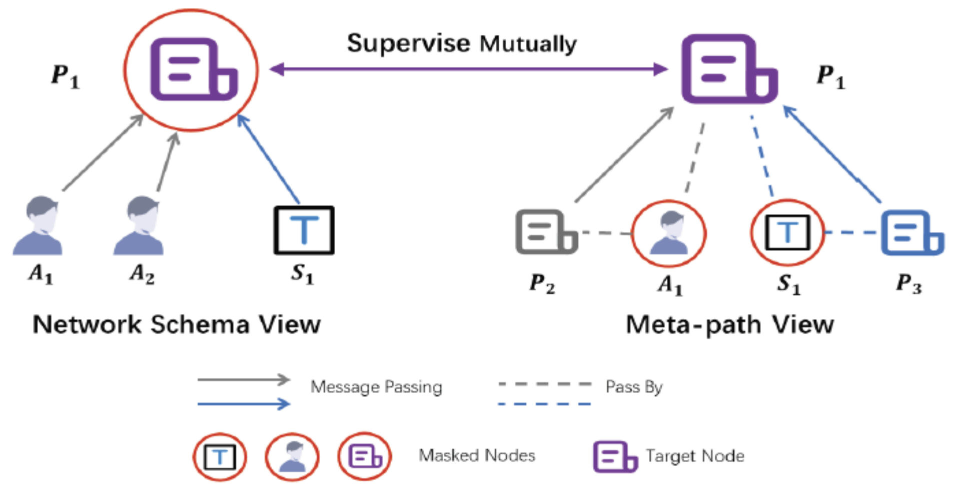


Figure 3: A schematic diagram of view mask mechanism.

- **Model extensions**
  - generate harder negative samples

I. **HeCo_GAN**: generate negative samples from a Gaussian distribution

II. **HeCo_MU**: randomly add k hardest negatives to create new negatives

- Node Classification

Table 3: Quantitative results (%±σ) on node classification.

Datasets	Metric	Split	GraphSAGE	GAE	Mp2vec	HERec	HetGNN	HAN	DGI	DMGI	HeCo
ACM	Ma-F1	20	47.13±4.7	62.72±3.1	51.91±0.9	55.13±1.5	72.11±0.9	85.66±2.1	79.27±3.8	87.86±0.2	**88.56±0.8**
		40	55.96±6.8	61.61±3.2	62.41±0.6	61.21±0.8	72.02±0.4	87.47±1.1	80.23±3.3	86.23±0.8	**87.61±0.5**
		60	56.59±5.7	61.67±2.9	61.13±0.4	64.35±0.8	74.33±0.6	88.41±1.1	80.03±3.3	87.97±0.4	**89.04±0.5**
	Mi-F1	20	49.72±5.5	68.02±1.9	53.13±0.9	57.47±1.5	71.89±1.1	85.11±2.2	79.63±3.5	87.60±0.8	**88.13±0.8**
		40	60.98±3.5	66.38±1.9	64.43±0.6	62.62±0.9	74.46±0.8	87.21±1.2	80.41±3.0	86.02±0.9	**87.45±0.5**
		60	60.72±4.3	65.71±2.2	62.72±0.3	65.15±0.9	76.08±0.7	88.10±1.2	80.15±3.2	87.82±0.5	**88.71±0.5**
	AUC	20	65.88±3.7	79.50±2.4	71.66±0.7	75.44±1.3	84.36±1.0	93.47±1.5	91.47±2.3	**96.72±0.3**	96.49±0.3
		40	71.06±5.2	79.14±2.5	80.48±0.4	79.84±0.5	85.01±0.6	94.84±0.9	91.52±2.3	96.35±0.3	**96.40±0.4**
		60	70.45±6.2	77.90±2.8	79.33±0.4	81.64±0.7	87.64±0.7	94.68±1.4	91.41±1.9	**96.79±0.2**	96.55±0.3
DBLP	Ma-F1	20	71.97±8.4	90.90±0.1	88.98±0.2	89.57±0.4	89.51±1.1	89.31±0.9	87.93±2.4	89.94±0.4	**91.28±0.2**
		40	73.69±8.4	89.60±0.3	88.68±0.2	89.73±0.4	88.61±0.8	88.87±1.0	88.62±0.6	89.25±0.4	**90.34±0.3**
		60	73.86±8.1	90.08±0.2	90.25±0.1	90.18±0.3	89.56±0.5	89.20±0.8	89.19±0.9	89.46±0.6	**90.64±0.3**
	Mi-F1	20	71.44±8.7	91.55±0.1	89.67±0.1	90.24±0.4	90.11±1.0	90.16±0.9	88.72±2.6	90.78±0.3	**91.97±0.2**
		40	73.61±8.6	90.00±0.3	89.14±0.2	90.15±0.4	89.03±0.7	89.47±0.9	89.22±0.5	89.92±0.4	**90.76±0.3**
		60	74.05±8.3	90.95±0.2	91.17±0.1	91.01±0.3	90.43±0.6	90.34±0.8	90.35±0.8	90.66±0.5	**91.59±0.2**
	AUC	20	90.59±4.3	98.15±0.1	97.69±0.0	98.21±0.2	97.96±0.4	98.07±0.6	96.99±1.4	97.75±0.3	**98.32±0.1**
		40	91.42±4.0	97.85±0.1	97.08±0.0	97.93±0.1	97.70±0.3	97.48±0.6	97.12±0.4	97.23±0.2	**98.06±0.1**
		60	91.73±3.8	98.37±0.1	98.00±0.0	98.49±0.1	97.97±0.2	97.96±0.5	97.76±0.5	97.72±0.4	**98.59±0.1**

- Node Classification

Table 3: Quantitative results (%±σ) on node classification.

Datasets	Metric	Split	GraphSAGE	GAE	Mp2vec	HERec	HetGNN	HAN	DGI	DMGI	HeCo
Freebase	Ma-F1	20	45.14±4.5	53.81±0.6	53.96±0.7	55.78±0.5	52.72±1.0	53.16±2.8	54.90±0.7	55.79±0.9	**59.23±0.7**
		40	44.88±4.1	52.44±2.3	57.80±1.1	59.28±0.6	48.57±0.5	59.63±2.3	53.40±1.4	49.88±1.9	**61.19±0.6**
		60	45.16±3.1	50.65±0.4	55.94±0.7	56.50±0.4	52.37±0.8	56.77±1.7	53.81±1.1	52.10±0.7	**60.13±1.3**
	Mi-F1	20	54.83±3.0	55.20±0.7	56.23±0.8	57.92±0.5	56.85±0.9	57.24±3.2	58.16±0.9	58.26±0.9	**61.72±0.6**
		40	57.08±3.2	56.05±2.0	61.01±1.3	62.71±0.7	53.96±1.1	63.74±2.7	57.82±0.8	54.28±1.6	**64.03±0.7**
		60	55.92±3.2	53.85±0.4	58.74±0.8	58.57±0.5	56.84±0.7	61.06±2.0	57.96±0.7	56.69±1.2	**63.61±1.6**
	AUC	20	67.63±5.0	73.03±0.7	71.78±0.7	73.89±0.4	70.84±0.7	73.26±2.1	72.80±0.6	73.19±1.2	**76.22±0.8**
		40	66.42±4.7	74.05±0.9	75.51±0.8	76.08±0.4	69.48±0.2	77.74±1.2	72.97±1.1	70.77±1.6	**78.44±0.5**
		60	66.78±3.5	71.75±0.4	74.78±0.4	74.89±0.4	71.01±0.5	75.69±1.5	73.32±0.9	73.17±1.4	**78.04±0.4**
AMiner	Ma-F1	20	42.46±2.5	60.22±2.0	54.78±0.5	58.32±1.1	50.06±0.9	56.07±3.2	51.61±3.2	59.50±2.1	**71.38±1.1**
		40	45.77±1.5	65.66±1.5	64.77±0.5	64.50±0.7	58.97±0.9	63.85±1.5	54.72±2.6	61.92±2.1	**73.75±0.5**
		60	44.91±2.0	63.74±1.6	60.65±0.3	65.53±0.7	57.34±1.4	62.02±1.2	55.45±2.4	61.15±2.5	**75.80±1.8**
	Mi-F1	20	49.68±3.1	65.78±2.9	60.82±0.4	63.64±1.1	61.49±2.5	68.86±4.6	62.39±3.9	63.93±3.3	**78.81±1.3**
		40	52.10±2.2	71.34±1.8	69.66±0.6	71.57±0.7	68.47±2.2	76.89±1.6	63.87±2.9	63.60±2.5	**80.53±0.7**
		60	51.36±2.2	67.70±1.9	63.92±0.5	69.76±0.8	65.61±2.2	74.73±1.4	63.10±3.0	62.51±2.6	**82.46±1.4**
	AUC	20	70.86±2.5	85.39±1.0	81.22±0.3	83.35±0.5	77.96±1.4	78.92±2.3	75.89±2.2	85.34±0.9	**90.82±0.6**
		40	74.44±1.3	88.29±1.0	88.82±0.2	88.70±0.4	83.14±1.6	80.72±2.1	77.86±2.1	88.02±1.3	**92.11±0.6**
		60	74.16±1.3	86.92±0.8	85.57±0.2	87.74±0.5	84.77±0.9	80.39±1.5	77.21±1.4	86.20±1.7	**92.40±0.7**

Experiments

- Node clustering

Table 2: Quantitative results on node clustering.

Datasets	ACM		DBLP		Freebase		AMiner	
Metrics	NMI	ARI	NMI	ARI	NMI	ARI	NMI	ARI
GraphSage	29.20	27.72	51.50	36.40	9.05	10.49	15.74	10.10
GAE	27.42	24.49	72.59	77.31	19.03	14.10	28.58	20.90
Mp2vec	48.43	34.65	73.55	77.70	16.47	17.32	30.80	25.26
HERec	47.54	35.67	70.21	73.99	19.76	19.36	27.82	20.16
HetGNN	41.53	34.81	69.79	75.34	12.25	15.01	21.46	26.60
DGI	51.73	41.16	59.23	61.85	18.34	11.29	22.06	15.93
DMGI	51.66	46.64	70.06	75.46	16.98	16.91	19.24	20.09
HeCo	**56.87**	**56.94**	**74.51**	**80.17**	**20.38**	**20.98**	**32.26**	**28.64**

- Model extensions

Table 4: Evaluation of extended models on various tasks using ACM (Task 1: Classification; Task 2: Clustering).

Task 1		DMGI	HeCo	HeCo_MU	HeCo_GAN
Ma	20	87.86±0.2	88.56±0.8	88.65±0.8	**89.22±1.1**
	40	86.23±0.8	87.61±0.5	87.78±1.7	**88.61±1.6**
	60	87.97±0.4	89.04±0.5	**89.83±0.5**	89.55±1.3
Mi	20	87.60±0.8	88.13±0.8	88.39±0.9	**88.92±0.9**
	40	86.02±0.9	87.45±0.5	87.66±1.7	**88.48±1.7**
	60	87.82±0.5	88.71±0.5	**89.52±0.5**	89.29±1.4
AUC	20	96.72±0.3	96.49±0.3	96.38±0.5	**96.91±0.3**
	40	96.35±0.3	96.40±0.4	96.54±0.5	**97.13±0.5**
	60	96.79±0.2	96.55±0.3	96.67±0.7	**97.12±0.4**
Task 2		DMGI	HeCo	HeCo_MU	HeCo_GAN
NMI		51.66	56.87	58.17	**59.34**
ARI		46.64	56.94	59.41	**61.48**

Existing GNN training:

Limitations		Solutions
• Require more abundant data for better results	➡	• Pre-training for graph
• Existing pre-training models designed for homogeneous graph	➡	• Design specific heterogeneous pre-training tasks

Xunqiang Jiang, Tianrui Jia, Yuan Fang, Chuan Shi, Zhe lin, and Hui Wang. Pre-training on Large-Scale Heterogeneous Graph. KDD2021

Challenges

Two fundamental problems

- 1. How to capture the **semantic** and **structural** properties on a heterogeneous graph during pre-training
  - Preserve the inherent semantic and structural properties

    Node and Network Schema-level pretrain tasks

- 2. How to **efficiently** pre-train GNNs on a large-scale heterogeneous graph
  - Real-word heterogeneous graphs：billions of nodes and edges

    Edge sparsification

Preserve heterogeneous properties as transferable knowledge, and sparsify large-scale heterogeneous graph for efficient pre-training

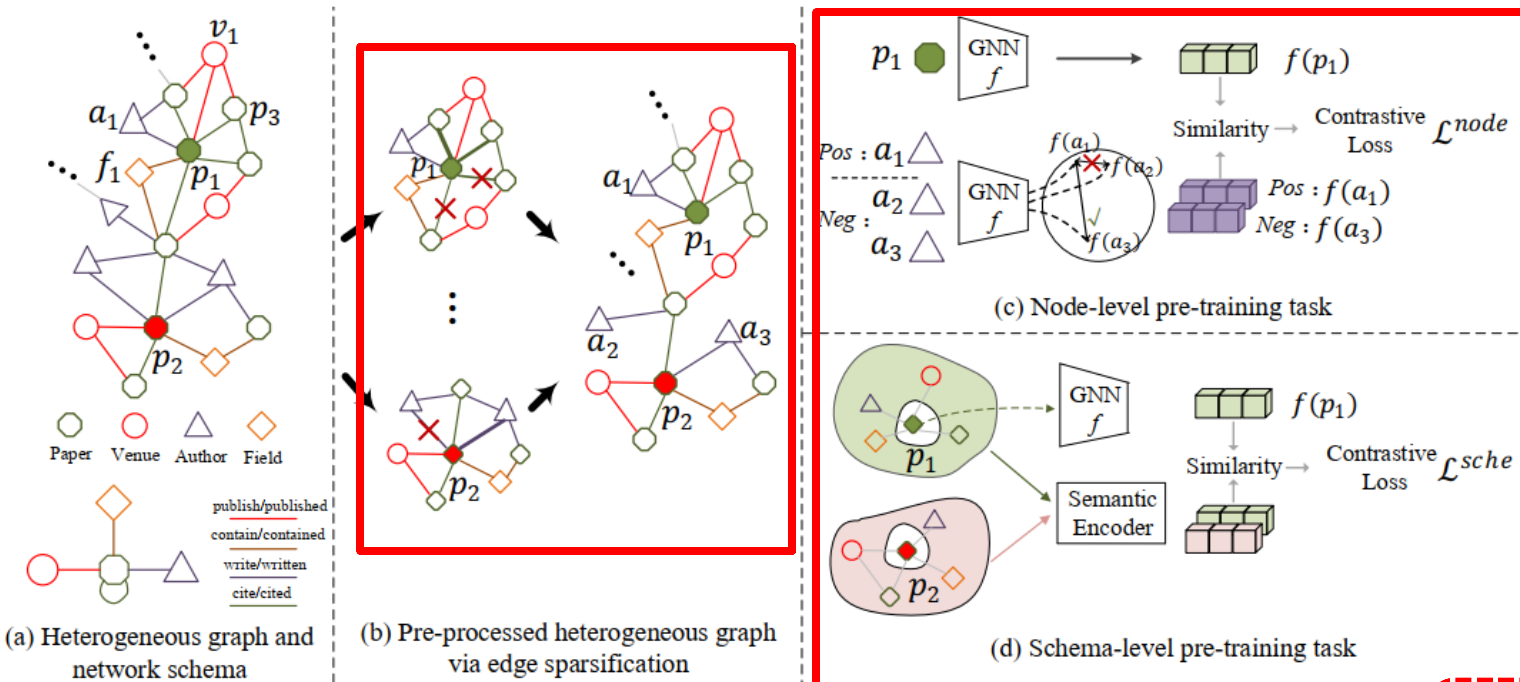


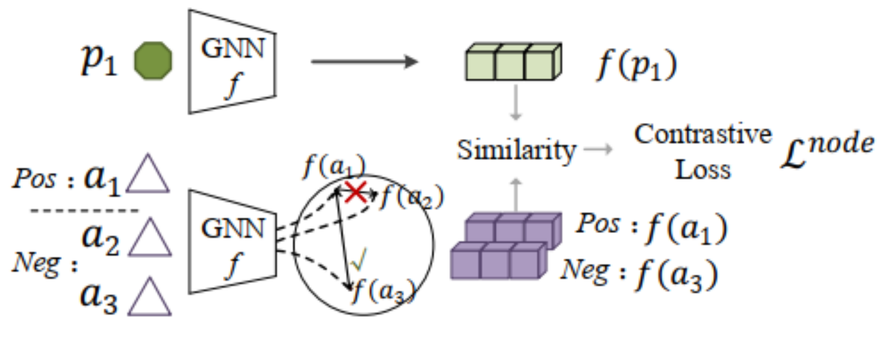
Figure 1: The overall framework of PT-HGNN.

Edge Sparsification

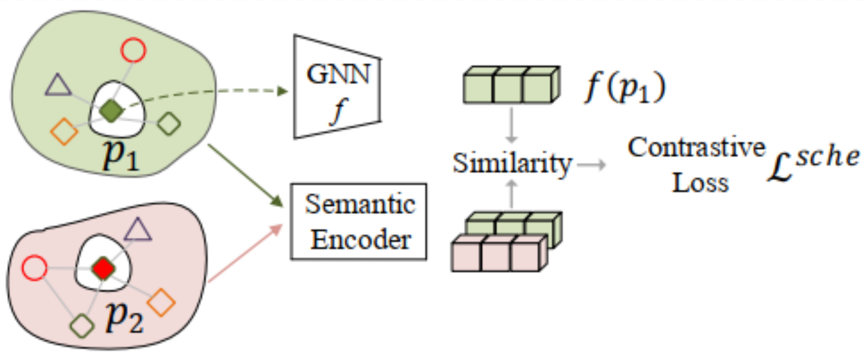
Pre-training Tasks

PT-HGNN

Design the node- and schema-level pre-training tasks



(c) Node-level pre-training task



(d) Schema-level pre-training task

Node-level:

Model **pairwise relations** between different types of node

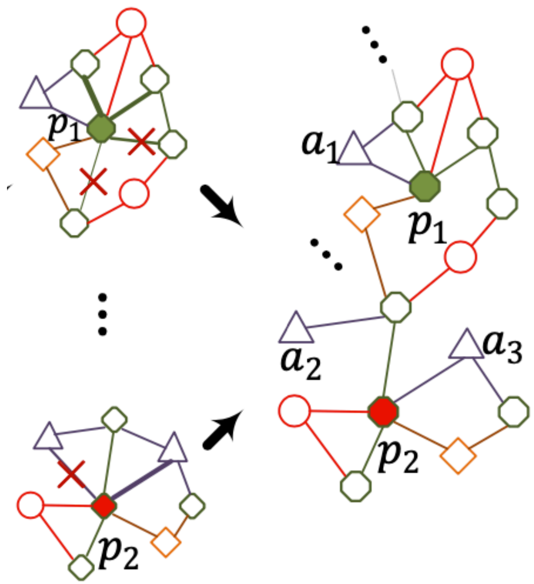
$$\mathcal{L}_{u,R}^{node} = -\log \frac{\exp\left(\mathbf{h}_u^\top \mathbf{W}_R \mathbf{h}_v / \tau\right)}{\sum_{i \in \{v\} \cup \{w \mid \langle u,R,w\rangle \in \mathcal{N}_{\langle u,R,v\rangle}^{node}\}} \exp\left(\mathbf{h}_u^\top \mathbf{W}_R \mathbf{h}_i / \tau\right)},$$

Schema-level:

Model the relation between **center node** with **context nodes**

$$\mathcal{L}_u^{sche} = \sum_{\mathbf{s}^+ \in \mathcal{P}_u^{sche}} \log \frac{\exp\left(\mathbf{h}_u^\top \mathbf{c}^{\mathbf{s}+} / \tau\right)}{\sum_{\mathbf{s} \in \{\mathbf{s}+\} \cup \mathcal{N}_u^{sche}} \exp\left(\mathbf{h}_u^\top \mathbf{c}^{\mathbf{s}} / \tau\right)},$$

Relation-based sparsification for efficiency



Why edge Sparsification:

- Preserve more **meaningful** edges (lower noise in graphs)
- Improve the **time efficiency** on large graph

Method : Relation based Personalized PageRank

Acceleration :

Random-Walk Formulation (Forward Search)

+ Top-K Entries

OAG Dataset Statistics

Open Academic Graph (OAG) unifies two academic graphs: Microsoft Academic Graph (MAG) and Aminer, is the largest existing knowledge graph

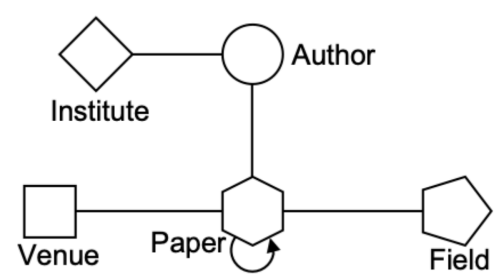
Dataset	#nodes	#edges	#venues	#papers	#fields	#authors	#institutes	#P-V	#P-P	#P-F	#P-A	#A-I
CS	11,918,983	107,263,811	27,433	5,597,605	289,930	5,985,759	18,256	5,597,606	31,441,552	47,462,559	15,571,614	7,190,480
Mater	4,552,941	42,161,581	15,141	2,442,235	79,305	2,005,362	10,898	2,442,235	13,011,272	19,119,947	5,582,765	2,005,362
Engin	5,191,920	36,146,719	19,867	3,239,504	99,444	1,819,100	14,005	3,239,504	4,848,158	22,498,822	3,741,135	1,819,100
Chem	12,158,967	159,537,437	19,142	7,193,321	183,782	4,748,812	13,910	7,193,321	74,018,600	57,162,528	16,414,176	4,748,812
OAG	178,663,987	2,236,196,802	53,073	89,606,257	615,288	88,364,081	25,288	89,606,258	1,021,237,518	657,049,405	300,853,688	167,449,933

Tasks:

- Ordinary experiment:
  - Paper-Field prediction
  - Paper-Venue prediction
  - Author Name disambiguation

  Node classification

  Link prediction
- Transfer Experiment
- Efficiency experiment



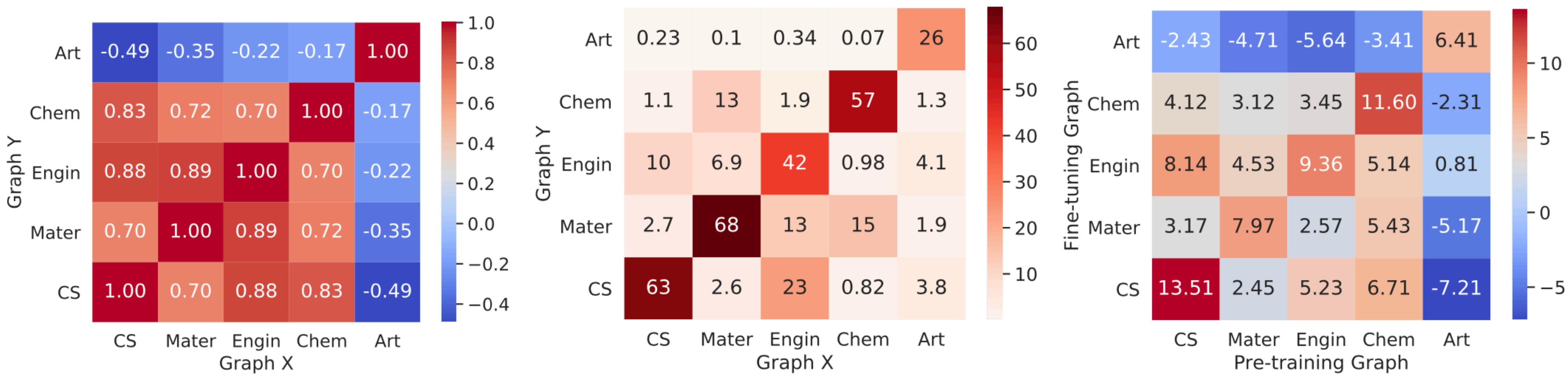
Network Schema of OAG

Experiment

Node Classification and Link Prediction

Dataset	Downstream Task		No pre-train	EdgePred	DGI	ContextPred	GraphCL	GPT-GNN	PT-HGNN	Improv.
CS	Paper–Field	NDCG	27.42±0.42	31.37±0.32	32.82±0.67	33.15±0.71	32.64±0.65	35.24±0.47	**36.04**±0.37	2.27%
		MRR	23.17±0.45	32.13±0.52	33.43±0.81	33.24±0.57	33.24±0.67	33.57±0.71	**37.76**±0.42	12.48%
	Paper–Venue	NDCG	27.76±0.56	35.77±0.59	34.23±0.71	34.30±0.92	32.11±0.69	36.15±0.53	**38.81**±0.51	7.35%
		MRR	11.39±0.37	16.34±0.47	16.21±0.62	17.66±0.81	16.29±0.49	19.13±0.65	**21.19**±0.45	10.76%
	Author ND	NDCG	76.27±0.53	79.41±0.68	81.38±0.93	79.22±0.72	79.95±0.89	80.20±0.51	**82.19**±0.60	2.48%
		MRR	54.82±0.49	59.06±0.74	58.98±0.79	60.23±0.83	60.55±0.74	60.94±0.52	**63.38**±0.38	4.00%
OAG	Paper–Field	NDCG	32.33±0.36	38.03±0.33	37.12±0.42	38.40±0.49	39.32±0.30	40.76±0.40	**42.33**±0.62	3.85%
		MRR	28.15±0.48	44.23±0.56	42.96±0.43	43.15±0.55	45.65±0.60	45.70±0.41	**47.29**±0.49	3.48%
	Paper–Venue	NDCG	42.28±0.50	43.25±0.61	44.23±0.53	43.07±0.74	42.66±0.66	44.05±0.75	**47.13**±0.68	6.56%
		MRR	22.76±0.37	23.40±0.35	24.38±0.35	24.12±0.42	25.03±0.48	25.19±0.45	**26.75**±0.57	6.19%
	Author ND	NDCG	76.52±1.13	78.01±0.86	77.98±0.93	77.88±0.72	78.11±0.93	79.33±0.87	**79.99**±0.92	0.83%
		MRR	54.65±0.53	58.00±0.63	58.30±0.48	57.49±0.60	58.22±0.53	59.08±0.52	**61.32**±0.55	3.79%

Transfer Experiment



(a) Correlation metric computed with a series of graph properties.

(b) The citation coefficient: the percentage of publications in graph Y that have citations in graph X

(c) MRR gain (%) of the proposed method over the method with no pre-training

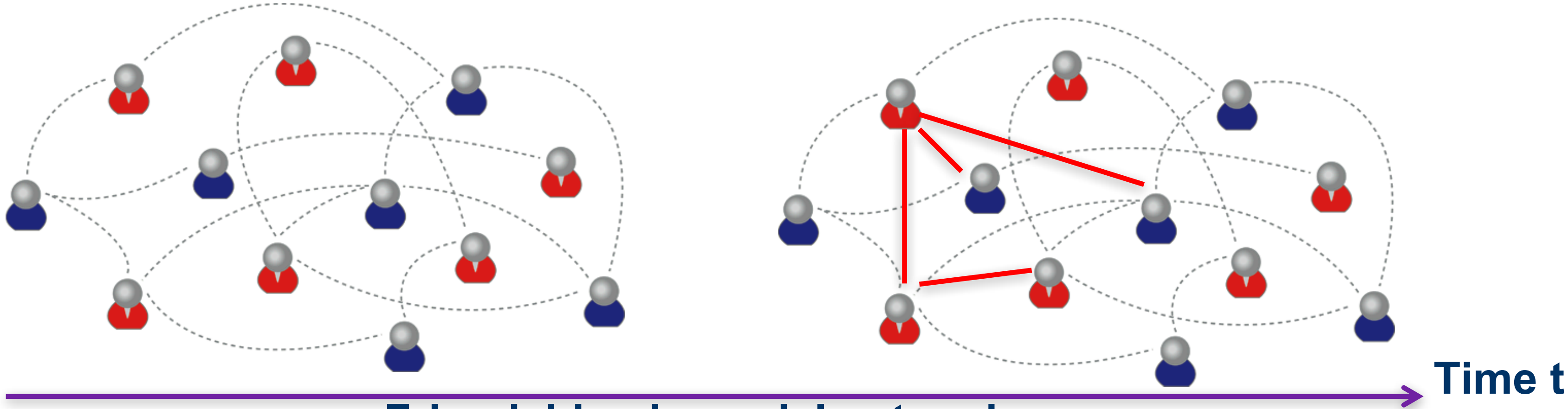
- **Knowledge transferring from pre-training to fine-tuning does not guarantee a gain in performance**
- **Positive correlation value between graphs results in positive transferring and vice versa**

Outline

- Basic concepts
- √ Models
  - Structure-preserved HG representation
  - Attribute-assisted HG representation(HGNN)
  - √ Dynamic HG representation
    - DyHNE (TKDE2020), SHCF(SDM2021)
- Applications
- Conclusion and future work

Networks are dynamic in nature

- Networks are dynamically evolving.

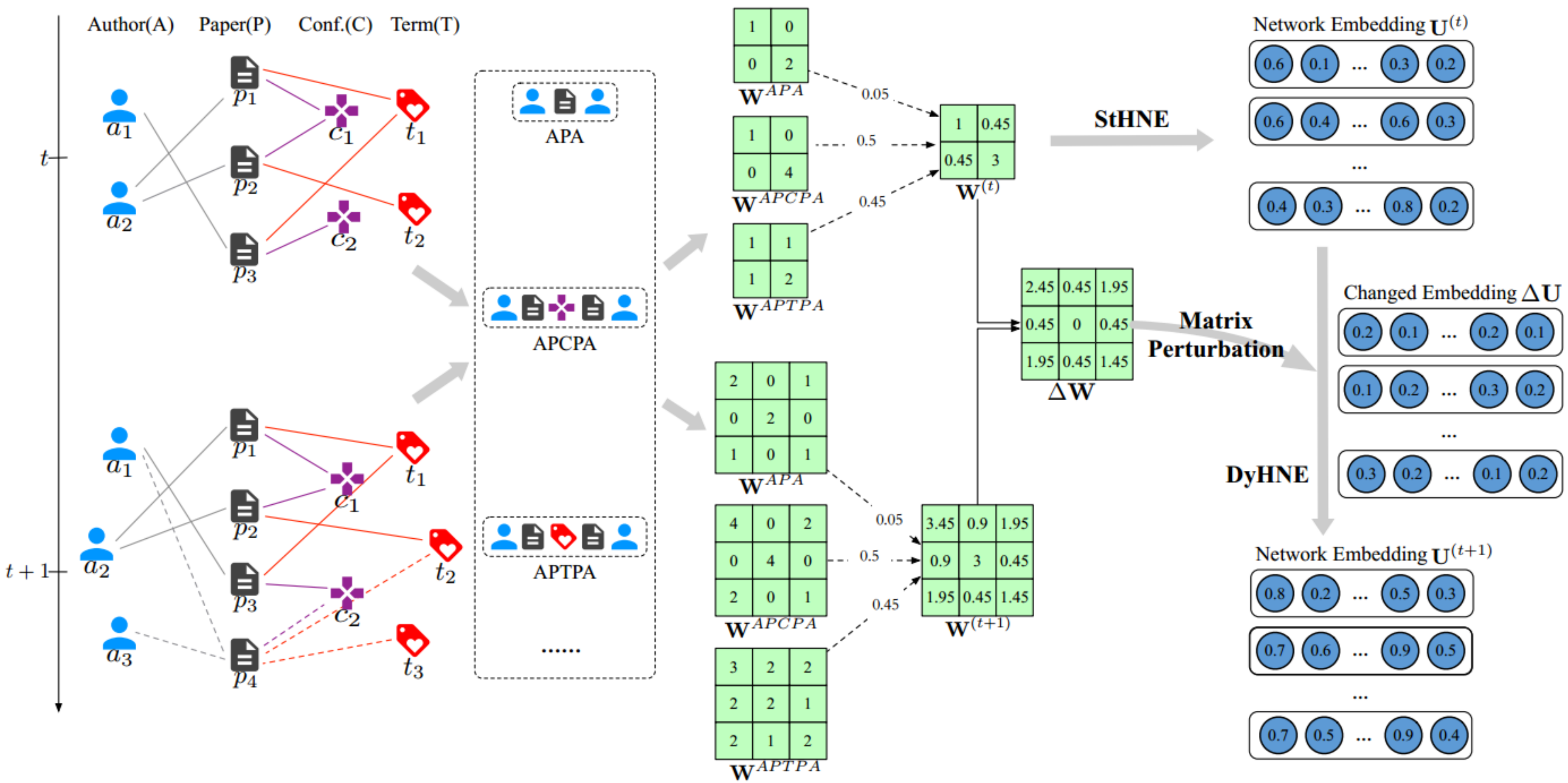


Friendships in social network

- Do we have to re-train network embedding model every time?
- The task is more challenging for HIN
  - Effective: capture meta-path structure
  - Efficient: incrementally update node embedding

Xiao Wang, Yuanfu Lu, Chuan Shi, Ruijia Wang, Peng Cui, Shuai Mou. Dynamic Heterogeneous Information Network Embedding with Meta-path based Proximity. IEEE TKDE, 2020.

☐ StHNE, Matrix perturbation, DyHNE



Static model: StHNE

- Static HIN embedding
  - Meta-path based first-order proximity

$$\mathcal{L}_1^m = \sum_{v_i, v_j \in \mathcal{V}} w_{ij}^m ||\mathbf{u}_i - \mathbf{u}_j||_2^2.$$

  - Meta-path based second-order proximity

$$\mathcal{L}_2^m = \sum_{v_p \in \mathcal{V}} ||\mathbf{u}_p - \sum_{v_q \in \mathcal{N}(v_p)^m} w_{pq}^m \mathbf{u}_q||_2^2.$$

  - The unified model

$$\arg \min_{\mathbf{U}^\top \mathbf{D} \mathbf{U} = \mathbf{I}} \sum_{m \in \mathcal{M}} \theta_m (\mathcal{L}_1^m + \gamma \mathcal{L}_2^m)$$

  - Generalized eigenvalue problem

$$(\mathbf{L} + \gamma \mathbf{H}) \mathbf{U} = \mathbf{D} \mathbf{\Lambda} \mathbf{U}$$

From time t to t+1, update $\mathbf{U}^{(t)}$ to $\mathbf{U}^{(t+1)}$

Dynamic model: DyHNE

- When HIN changes, the changes of L and H are

$$\Delta \mathbf{L} = \Delta \mathbf{D} - \Delta \mathbf{W},$$

$$\Delta \mathbf{H} = \Delta \mathbf{W}^\top \Delta \mathbf{W} - (\mathbf{I} - \mathbf{W})^\top \Delta \mathbf{W} - \Delta \mathbf{W}^\top (\mathbf{I} - \mathbf{W}).$$

  Then the optimization becomes

$$\begin{aligned}&(\mathbf{L} + \Delta \mathbf{L} + \gamma \mathbf{H} + \gamma \Delta \mathbf{H})(\mathbf{u}_i + \Delta \mathbf{u}_i)\\ &= (\lambda_i + \Delta \lambda_i)(\mathbf{D} + \Delta \mathbf{D})(\mathbf{u}_i + \Delta \mathbf{u}_\mathbf{i}).\end{aligned}$$

  Other operations also re-designed

- Finally, we have

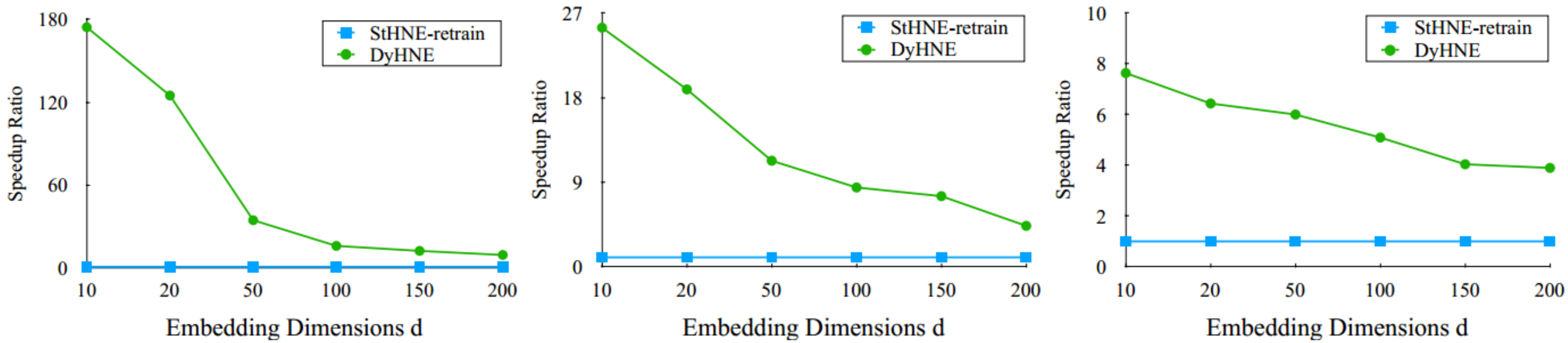
$$\mathbf{\Lambda}^{(t+1)} = \mathbf{\Lambda}^{(t)} + \Delta \mathbf{\Lambda}, \qquad \mathbf{U}^{(t+1)} = \mathbf{U}^{(t)} + \Delta \mathbf{U}.$$

- More acceleration details, please see the paper.

Node classification on DyHNE

Datasets	Metric	Tr.Ratio	DeepWalk	LINE-1st	LINE-1st	ESim	metapath2vec	StHNE	DANE	DHPE	DHNE	DyHNE
Yelp	Macro-F1	40%	0.5840	0.5623	0.5248	0.6463	0.5765	0.6118	0.6102	0.5412	0.6293	**0.6459**
		60%	0.5962	0.5863	0.5392	0.6642	0.6192	0.6644	0.6342	0.5546	0.6342	**0.6641**
		80%	0.6044	0.6001	0.6030	0.6744	0.6285	0.6882	0.6471	0.5616	0.6529	**0.6893**
	Micro-F1	40%	0.6443	0.6214	0.5901	0.6932	0.6457	0.6826	0.6894	0.5823	0.6689	**0.6933**
		60%	0.6558	0.6338	0.5435	0.6941	0.6656	0.7074	0.6921	0.5981	0.6794	**0.6998**
		80%	0.6634	0.6424	0.6297	0.7104	0.6722	0.7281	0.6959	0.6034	0.6931	**0.7298**
DBLP	Macro-F1	40%	0.9269	0.9266	0.9147	0.9372	0.9162	0.9395	0.8862	0.8893	0.9302	**0.9434**
		60%	0.9297	0.9283	0.9141	0.9369	0.9253	0.9461	0.8956	0.8946	0.9351	**0.9476**
		80%	0.9322	0.9291	0.9217	0.9376	0.9302	0.9502	0.9051	0.9087	0.9423	**0.9581**
	Micro-F1	40%	0.9375	0.9310	0.9198	0.9383	0.9254	0.9438	0.8883	0.8847	0.9352	**0.9467**
		60%	0.9346	0.9245	0.9192	0.9404	0.9281	0.9496	0.8879	0.8931	0.9404	**0.9505**
		80%	0.9371	0.9297	0.9261	0.9415	0.9354	0.9543	0.9071	0.9041	0.9489	**0.9617**
AMiner	Macro-F1	40%	0.8197	0.8219	0.8282	0.8797	0.8673	0.8628	0.7642	0.7694	0.8903	**0.9014**
		60%	0.8221	0.8218	0.8323	0.8807	0.8734	0.8651	0.7704	0.7735	0.9011	**0.9131**
		80%	0.8235	0.8238	0.8351	0.8821	0.8754	0.8778	0.7793	0.7851	0.9183	**0.9212**
	Micro-F1	40%	0.8157	0.8189	0.8323	0.8729	0.8652	0.8563	0.7698	0.7633	0.8992	**0.9117**
		60%	0.8175	0.8182	0.8361	0.8734	0.8693	0.8574	0.7723	0.7698	0.9045	**0.9178**
		80%	0.8191	0.8201	0.8298	0.8751	0.8725	0.8728	0.7857	0.7704	0.9132	**0.9203**

Efficiency compared to StHNE



(a) Yelp (b) DBLP (c) AMiner

Approximation error analysis

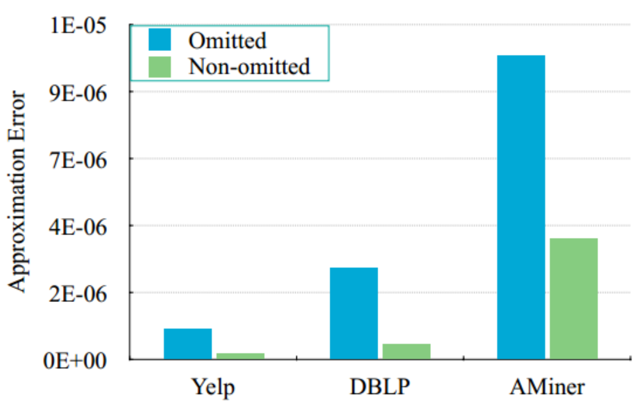


Fig. 7: The approximation error of matrix eigendecomposition in DyHNE, with respect to one timestamp update.

TABLE 6: The approximation error that whether to omit higher-order terms in DyHNE, w.r.t. node classification and relationship prediction.

Task	Model	Yelp	DBLP	AMiner
Classification (Mircro-F1)	Non-omitted	0.6922	0.9611	0.9521
	Omitted	0.6893	0.9581	0.9212
	Error	0.004189	0.003124	0.032454
LinkPrediction (AUC)	Non-omitted	0.8364	0.9385	0.8939
	Omitted	0.8346	0.9278	0.8821
	Error	0.002152	0.011401	0.013201

HIN based Sequential Recommendation

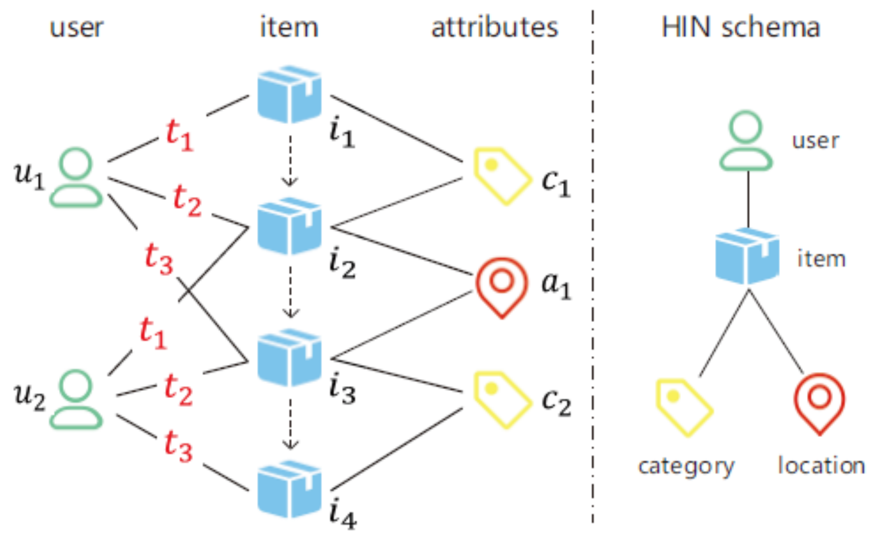
◻ HIN-based Recommendation

Neglecting sequential pattern while people have dynamic preference.

Can we jointly consider sequential information as well as high-order heterogeneous information?

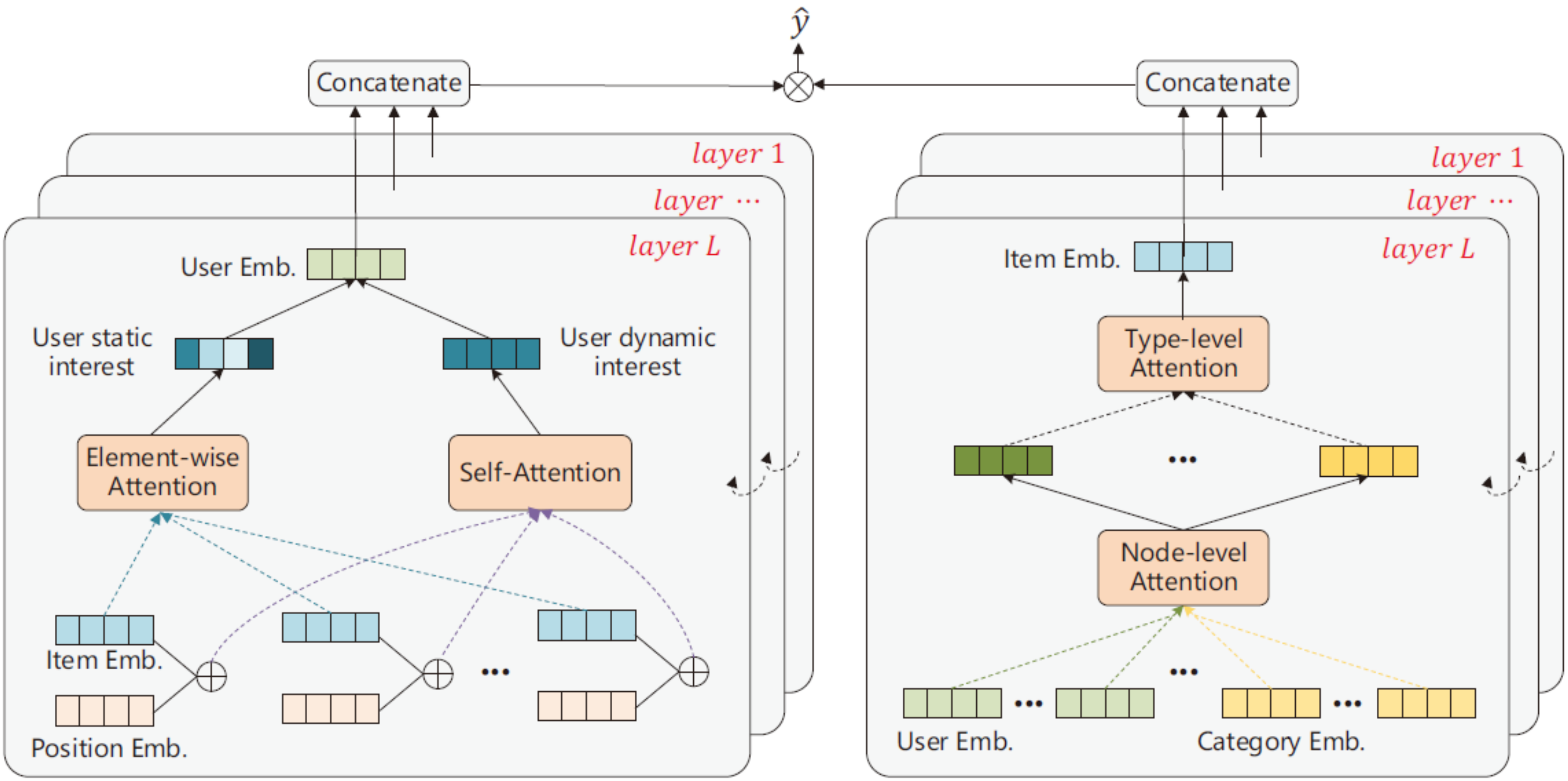
◻ Sequential Recommendation

Suffering from cold-start problem when there are few user interaction behaviors.



Chen Li, Linmei Hu, Chuan Shi, Guojie Song and Yuanfu Lu. Sequence-aware Heterogeneous Graph Neural Collaborative Filtering. SDM 2021

SHCF Model



$\hat{y}$
Concatenate
Concatenate
layer 1
layer ···
layer L
User Emb.
User static interest
User dynamic interest
Element-wise Attention
Self-Attention
Item Emb.
Position Emb.
layer 1
layer ···
layer L
Item Emb.
Type-level Attention
Node-level Attention
User Emb.
Category Emb.

☐ Sequence-aware Self-attention

$$\hat{\mathbf{I}}_u = \begin{bmatrix} \mathbf{i}_t + \mathbf{p}_1 \\ \mathbf{i}_{t-1} + \mathbf{p}_2 \\ \cdots \\ \mathbf{i}_1 + \mathbf{p}_t \end{bmatrix}, \quad \text{ATTENTION}(\mathbf{Q},\mathbf{K},\mathbf{V}) = softmax(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d}}) \cdot \mathbf{V},$$

$$\mathbf{u}_d = \mathop{\Big\|}_{h=1}^{H} \text{ATTENTION}(\hat{\mathbf{I}}_u \mathbf{W}^Q, \hat{\mathbf{I}}_u \mathbf{W}^K, \hat{\mathbf{I}}_u \mathbf{W}^V)$$

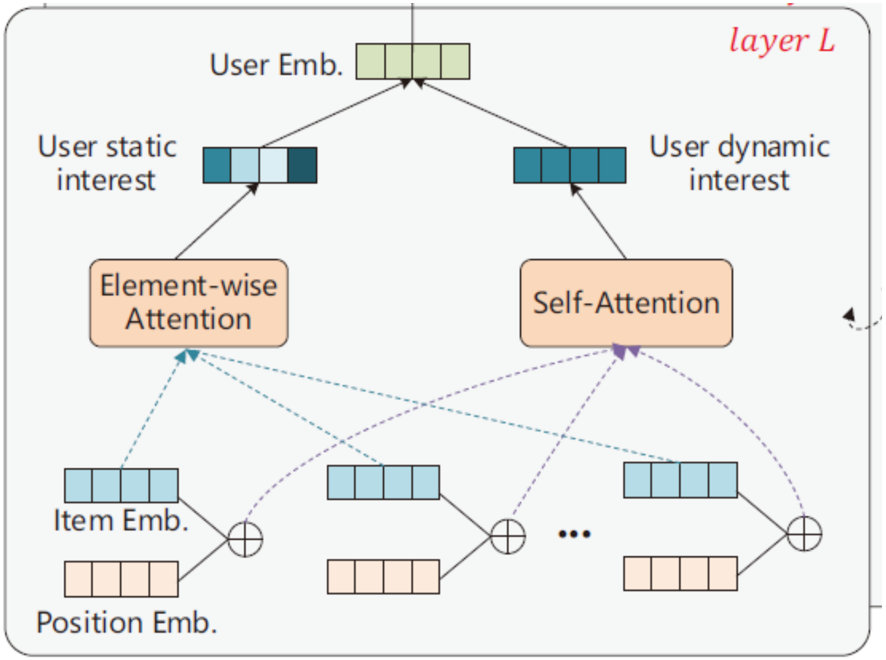
☐ Element-wise Attention

$$\gamma_j = \tanh(\mathbf{W}_u \cdot \mathbf{i}_j + b),$$

$$\mathbf{u}_s = \sum_{j \in \mathcal{S}_u} \gamma_j \odot \mathbf{i}_j .$$

☐ User Modeling

$$\tilde{\mathbf{u}} = \lambda \mathbf{u}_d + (1 - \lambda)\mathbf{u}_s$$



Node-level Attention

$$\alpha_{vv'} = \frac{\exp(\sigma(\mathbf{a}_{\tau'}^{\top} \cdot [\mathbf{i}_v || \mathbf{h}_{v'}]))}{\sum_{k \in \mathcal{N}_v^{\tau'}} \exp(\sigma(\mathbf{a}_{\tau'}^{\top} \cdot [\mathbf{i}_v || \mathbf{h}_k]))},$$

$$\mathbf{g}_v^{\tau'} = \sigma(\sum_{v' \in \mathcal{N}_v^{\tau'}} \alpha_{vv'} \cdot \mathbf{h}_{v'}).$$

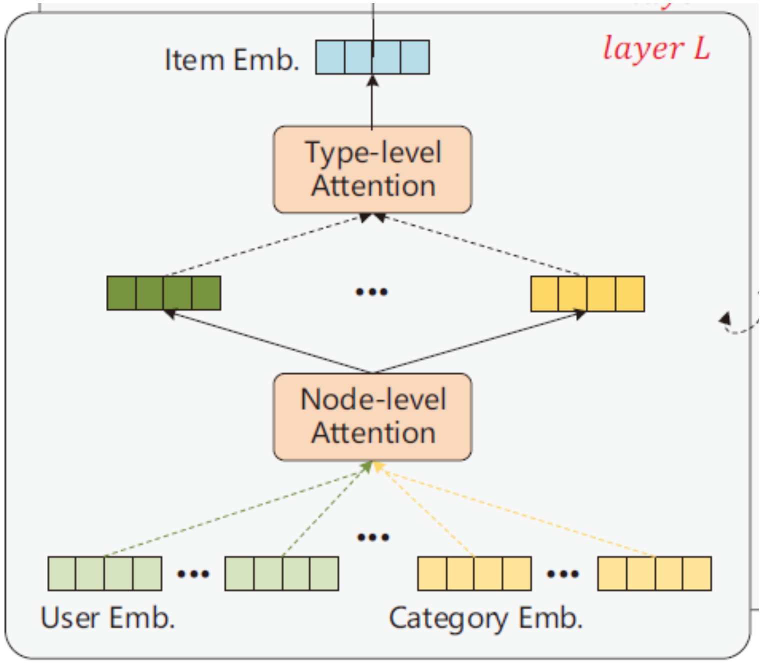
Type-level Attention

$$m_v^{\tau'} = V \cdot \tanh(\mathbf{w} \cdot \mathbf{g}_v^{\tau'} + b),$$

$$\beta_v^{\tau'} = \frac{\exp(m_v^{\tau'})}{\sum_{\tau \in \mathcal{T}} \exp(m_v^{\tau})}.$$

Item Modeling

$$\tilde{\mathbf{i}}_v = \sigma(\sum_{\tau' \in \mathcal{T}} \beta_v^{\tau'} \cdot \mathbf{g}_v^{\tau'}).$$

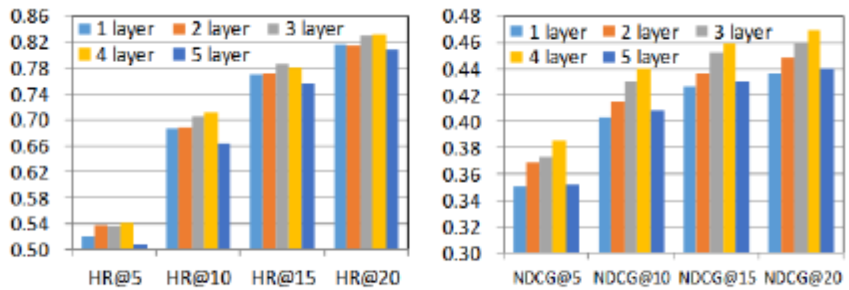


Experiments

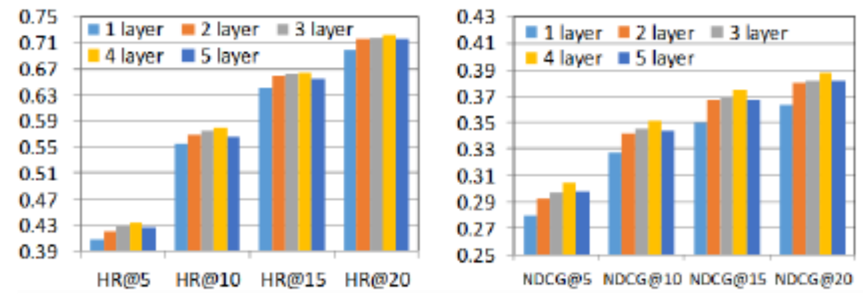
Table 2: Recommendation performance of different models. The best result in each row is bold and the second best result is underlined. The improvements of our method over the second best models are shown in the last column.

Dataset	Metrics	BPR-MF	NeuMF	NGCF	NeuACF	HeRec	NARM	SR-GNN	SHCF	Improve
ML100K	HR@5	0.4030	0.4057	0.4274	0.4337	0.4255	0.5228	0.5010	**0.5414**	3.56%
	NDCG@5	0.2747	0.2676	0.2889	0.2874	0.2798	0.3659	0.3510	**0.3859**	5.47%
	HR@10	0.5801	0.5689	0.5864	0.6034	0.6012	0.6723	0.6660	**0.7108**	5.73%
	NDCG@10	0.3312	0.3127	0.3402	0.3420	0.3325	0.4142	0.4048	**0.4401**	6.25%
	HR@15	0.6787	0.6706	0.6649	0.7084	0.6981	0.7529	0.7598	**0.7817**	2.88%
	NDCG@15	0.3573	0.3462	0.3611	0.3697	0.3521	0.4354	0.4298	**0.4592**	5.47%
	HR@20	0.7455	0.7595	0.7434	0.7720	0.7524	0.8038	0.8048	**0.8324**	3.43%
	NDCG@20	0.3731	0.3672	0.3796	0.3847	0.3721	0.4475	0.4396	**0.4693**	4.87%
ML1M	HR@5	0.4921	0.5092	0.5017	0.5050	0.4923	0.6713	0.6634	**0.6927**	3.18%
	NDCG@5	0.3376	0.3511	0.3437	0.3508	0.3455	0.5201	0.5233	**0.5299**	1.26%
	HR@10	0.6577	0.6803	0.6688	0.6684	0.6601	0.7603	0.7699	**0.7964**	3.19%
	NDCG@10	0.3910	0.4066	0.3977	0.4038	0.3982	0.5565	0.5580	**0.5639**	1.06%
	HR@15	0.7551	0.7761	0.7587	0.7593	0.7403	0.8230	0.8184	**0.8503**	3.32%
	NDCG@15	0.4168	0.4320	0.4216	0.4279	0.4194	0.5687	0.5709	**0.5785**	1.33%
	HR@20	0.8159	0.8369	0.8167	0.8232	0.8105	0.8602	0.8584	**0.8844**	2.81%
	NDCG@20	0.4311	0.4463	0.4353	0.4430	0.4328	0.5723	0.5803	**0.5968**	2.84%
Yelp	HR@5	0.3077	0.3571	0.4097	0.4094	0.3982	0.3490	0.3754	**0.4421**	7.91%
	NDCG@5	0.2086	0.2419	0.2855	0.2844	0.2765	0.2373	0.2565	**0.3100**	8.58%
	HR@10	0.4325	0.5018	0.5584	0.5553	0.5505	0.4900	0.5208	**0.5878**	5.27%
	NDCG@10	0.2488	0.2885	0.3335	0.3314	0.3311	0.2828	0.3035	**0.3572**	7.11%
	HR@15	0.5084	0.6006	0.6434	0.6504	0.6423	0.5851	0.6054	**0.6725**	3.40%
	NDCG@15	0.2689	0.3146	0.3544	0.3565	0.3499	0.3080	0.3259	**0.3796**	6.48%
	HR@20	0.5643	0.6717	0.7005	0.7138	0.6923	0.6496	0.6684	**0.7260**	1.71%
	NDCG@20	0.2821	0.3315	0.3686	0.3715	0.3603	0.3232	0.3408	**0.3922**	5.57%

☐ Effect of the number of message passing layers

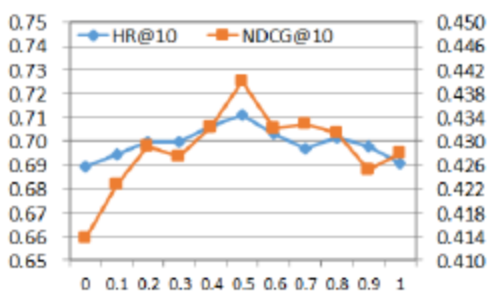


(a) ML100K

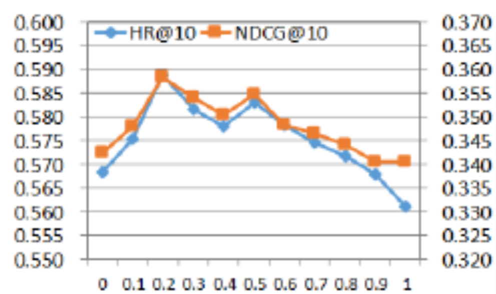


(b) Yelp

☐ Effect of the balance coefficient λ



(a) ML100K



(b) Yelp

Outline

- Basic concepts
- Models

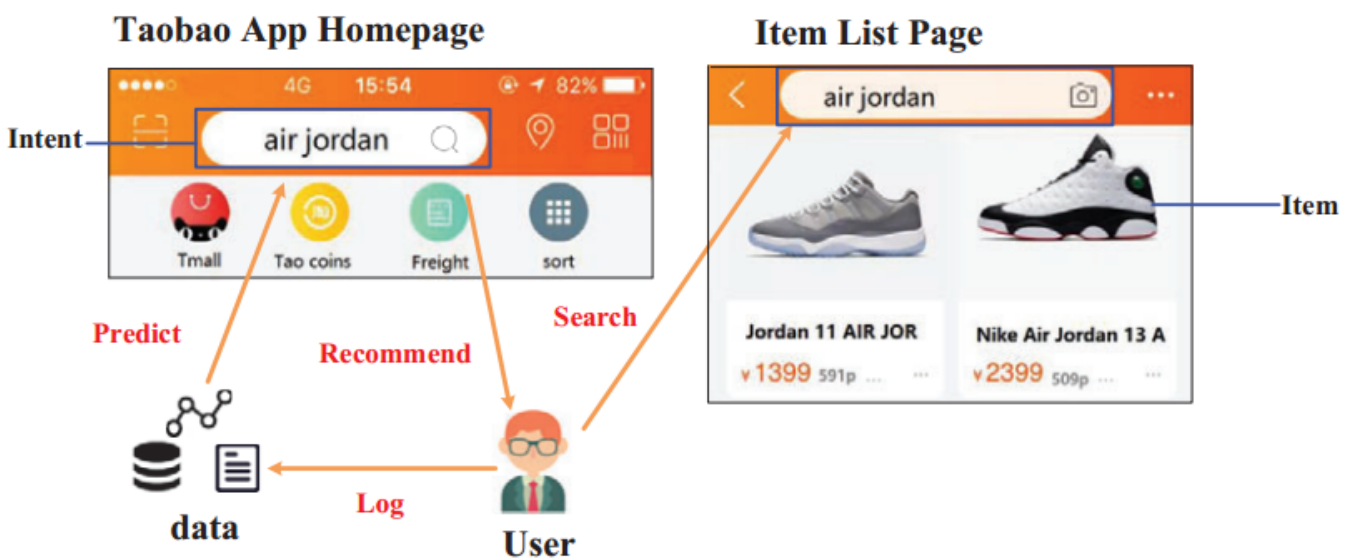
√ Applications

  √ MEIRec (KDD2019), HGAT(EMNLP2019), HACUD (AAAI2019),

  √ HGSRec(AAAI21), SIAN(PKDD20), IPRec(SIGIR21)

- Conclusion and future work

- **Intent recommendation**
  - A new recommendation service in many mobile e-commerce Apps.
  - Automatically recommend a personalized intent for a user according to his/her historical behaviors without query input.



Shaohua Fan, Junxiong Zhu, Xiaotian Han, Chuan Shi, Linmei Hu, Biyu Ma, Yongliang Li, Metapath-guided Heterogeneous Graph Neural Network for Intent Recommendation. KDD 2019.

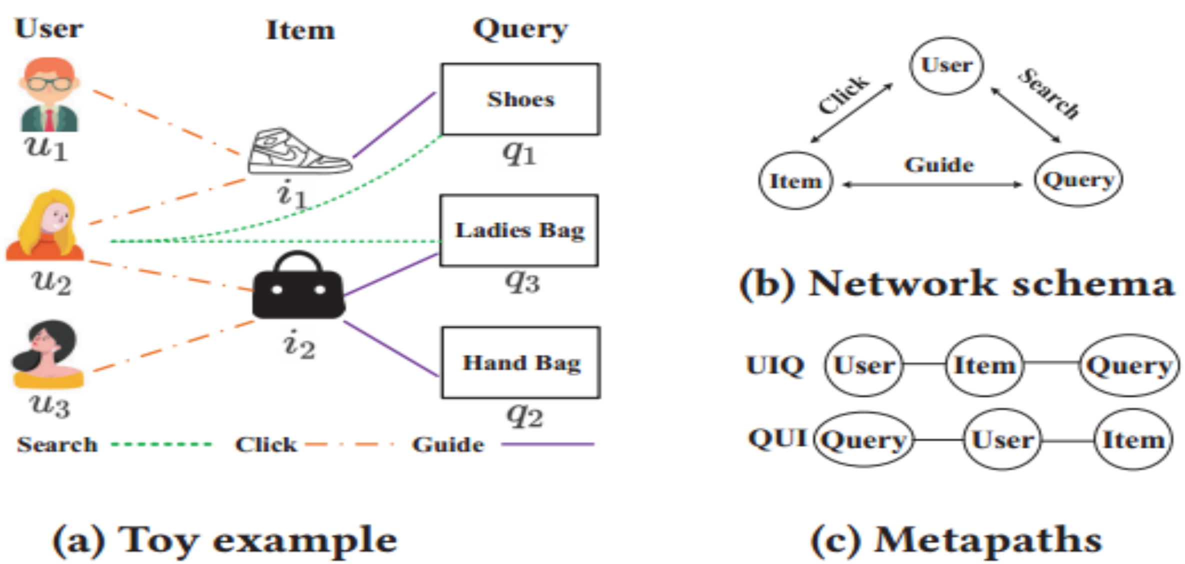
Background

Existing methods used in industry

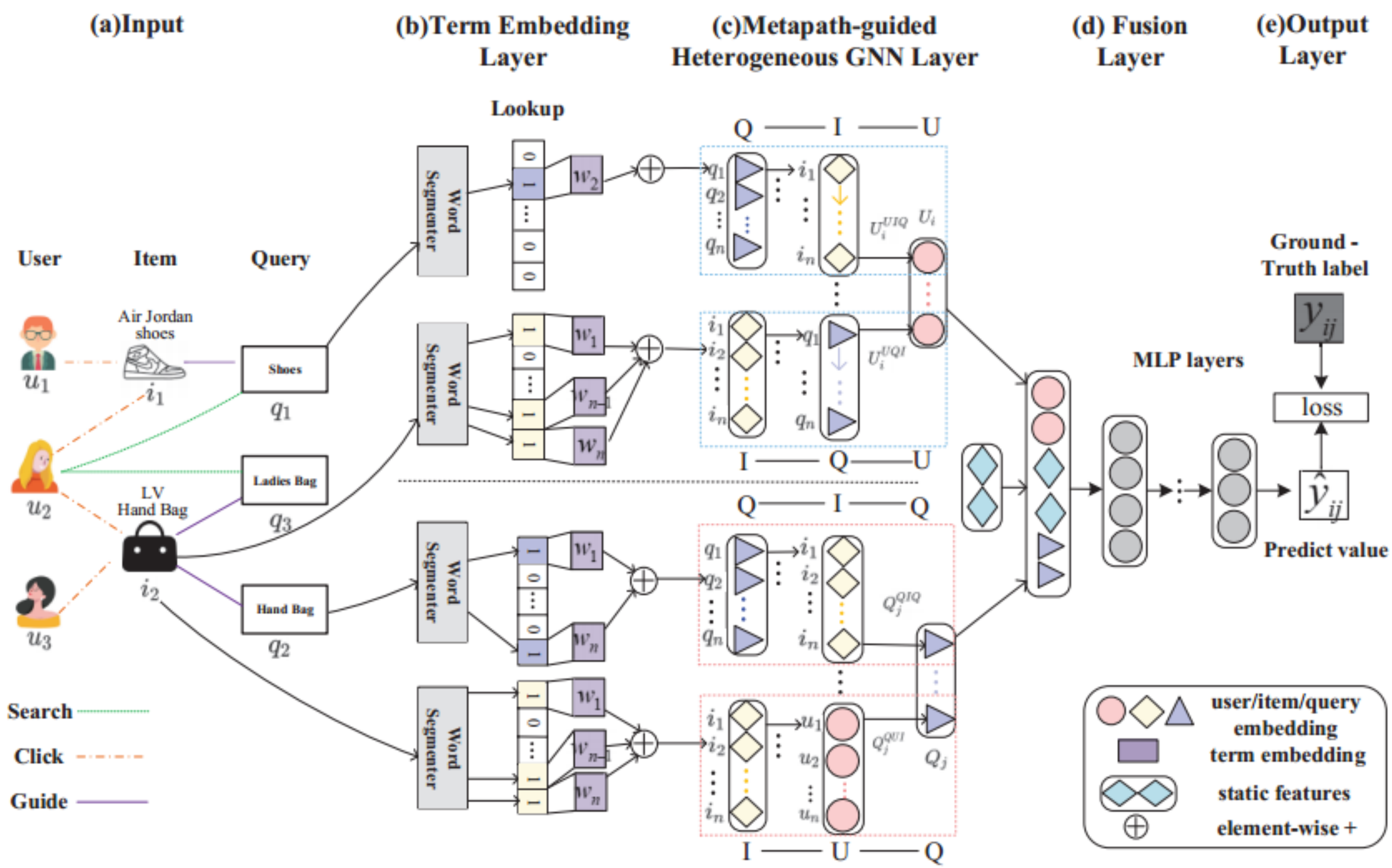
- **Classification method**
  - heavily rely on domain knowledge and need laboring feature engineering
  - fail to take full advantage of rich interaction information
- **Item recommendation method**
  - Only consider binary interactions between users and items
  - Only consider atomic and static items, intent always dynamic change.

MEIRec

- **Model intent recommendation with a HIN**
  - Flexibly exploit rich interaction
- **Heterogeneous Graph Neural Network**
  - Learn structural feature representations of users and queries
  - A uniform term embedding mechanism to handle large-scale and dynamic data

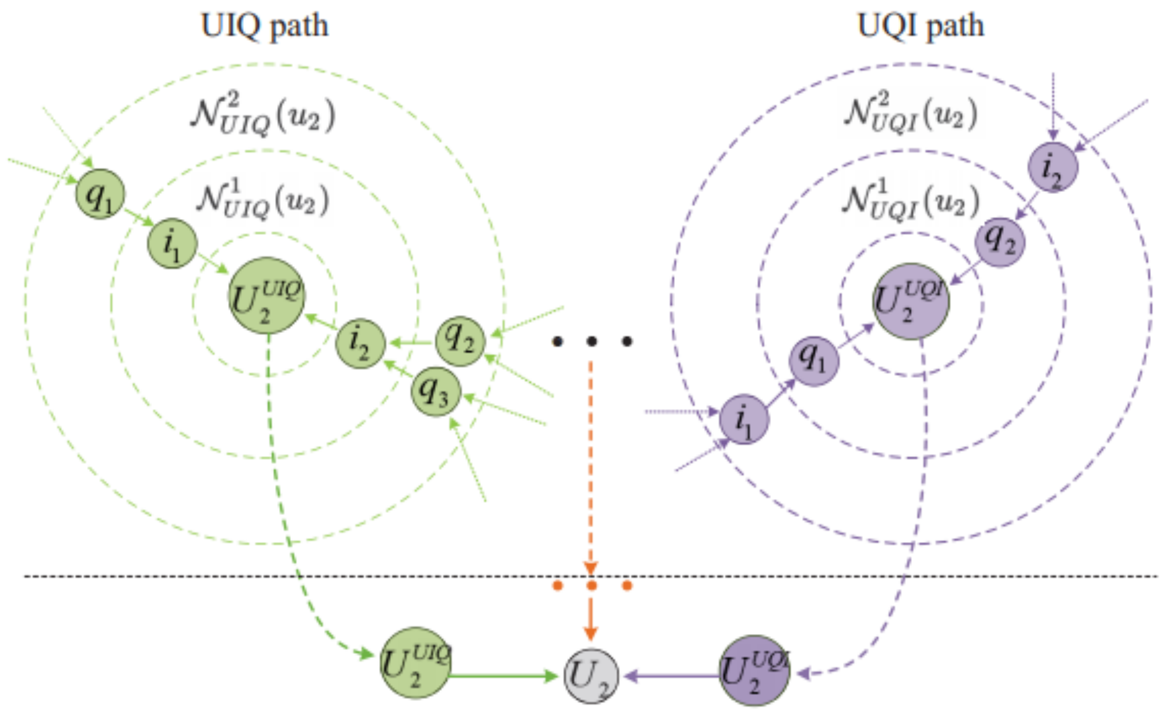


MEIRec Method



The framework of MEIRec

Metapath-guided Neighbor Aggregation



- Fuse heterogeneous information
  - Leverage metapaths to obtain different-step neighbors of an object
  - Different aggregation functions are designed for different types of neighboring information
  - More information can be added by expanding metapaths

Offline experiments

Method	1-day				3-day				5-day			
	40%	60%	80%	100%	40%	60%	80%	100%	40%	60%	80%	100%
NeuMF	0.6014	0.6066	0.6136	0.6143	0.6168	0.6218	0.6249	0.6291	0.6172	0.6224	0.6246	0.6295
LR	0.6854	0.6838	0.6884	0.6889	0.6844	0.6863	0.6857	0.6865	0.6817	0.6831	0.6827	0.6836
LR+DW	0.6878	0.6904	0.6898	0.6930	0.6888	0.6896	0.6898	0.6900	0.6838	0.6842	0.6863	0.6867
LR+MP	0.6918	0.6936	0.6950	0.6969	0.6919	0.6930	0.6933	0.6933	0.6874	0.6890	0.6898	0.6899
DNN	0.6939	0.6981	0.6991	0.6997	0.6966	0.6985	0.6999	0.7008	0.6996	0.7011	0.7017	0.7029
DNN+DW	0.6962	0.6980	0.7003	0.7024	0.7005	0.7017	0.7024	0.7030	0.7017	0.7029	0.7040	0.7047
DNN+MP	0.6984	0.6992	0.7024	0.7057	0.7025	0.7040	0.7051	0.7057	0.7017	0.7044	0.7060	0.7069
GBDT	0.7071	0.7071	0.7067	0.7073	0.7070	0.7071	0.7072	0.7071	0.7067	0.7068	0.7072	0.7066
GBDT+DW	0.7114	0.7119	0.7112*	0.7118*	0.7109	0.7106	0.7106	0.7104	0.7109	0.7112	0.7109	0.7114
GBDT+MP	0.7122*	0.7127*	0.7110	0.7111	0.7123*	0.7122*	0.7122*	0.7124*	0.7118*	0.7114*	0.7114*	0.7120*
MEIRec	**0.7273**	**0.7302**	**0.7339**	**0.7346**	**0.7352**	**0.7369**	**0.7380**	**0.7390**	**0.7372**	**0.7401**	**0.7409**	**0.7425**
Improvement	2.1%	2.5%	3.2%	3.2%	3.2%	3.5%	3.6%	3.7%	3.6%	4.0%	4.1%	4.3%

MEIRec significantly outperforms GBDT, DNN, and MF based methods

Online experiments

Table 3: Online A/B testing experiments results.

Data	Methods	CTR	Unique Click	UCTR
Android	GBDT	1.746%	256,116	13.939%
	MEIRec	1.758%	260,634	14.229%
	Improvement	0.70%	1.76%	2.07%
IOS	GBDT	0.7687%	62,462	5.2579%
	MEIRec	0.8056%	65,895	5.5436%
	Improvement	4.79%	5.50%	5.43%
Total	GBDT	1.4035%	318,578	10.5252%
	MEIRec	1.4252%	326,529	10.8052%
	Improvement	1.54%	2.50%	2.66%

MEIRec significantly improves key metrics considered by the platform and attracts more new users to search the recommended query

Background

- Short texts are very popular in social media
  - Tweets
  - Reviews
  - News

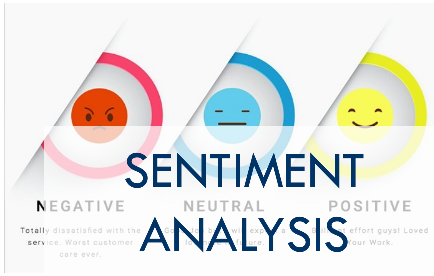
The hard truth about the United States is that the money other countries spend on health and infrastructure, we spend on war.

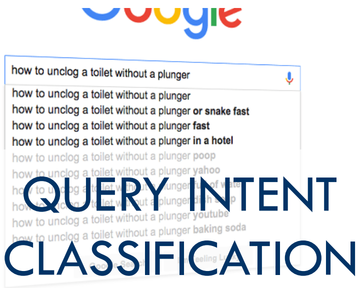
★★★★★ **Home secured.**
By M. Wolff on January 25, 2014
I purchased this along with 4 turtles and a rat.

18 years worth of karate lessons later, I finally feel safe to leave my house at night.

Trump claims he's suing 'various people' for violating confidentiality agreements
By Veronica Stracqualursi and Pamela Brown, CNN
Updated 4:37 PM ET, Sat August 31, 2019

- Short text classification is the foundation of many applications
- Semi-supervised short text classification is a pressing need, due to limited labeled data





Linmei Hu, Tianchi Yang, Chuan Shi, Houye Ji, Xiaoli Li. Heterogeneous Graph Attention Networks for Semi-supervised Short Text Classification. EMNLP 2019.

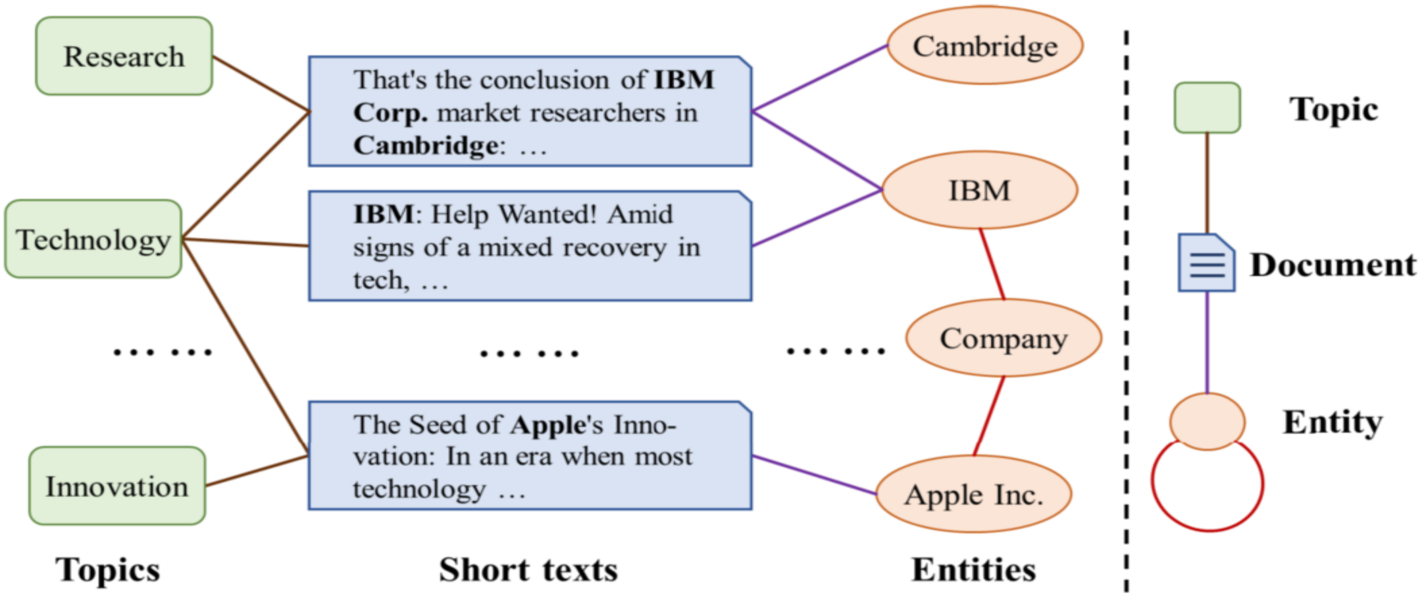
Motivation

semi-supervised short text classification

Challenges	Solutions
• Semantically sparse and ambiguous	• HIN flexibly integrates information
• Different importances of integrated information	• Attention mechanism can weight importances
• Limited labeled training data	• Information propagation in network

Heterogeneous Graph ATtention networks (HGAT) for short text classification

HIN Modeling



(a) HIN for short texts (b) HIN schema

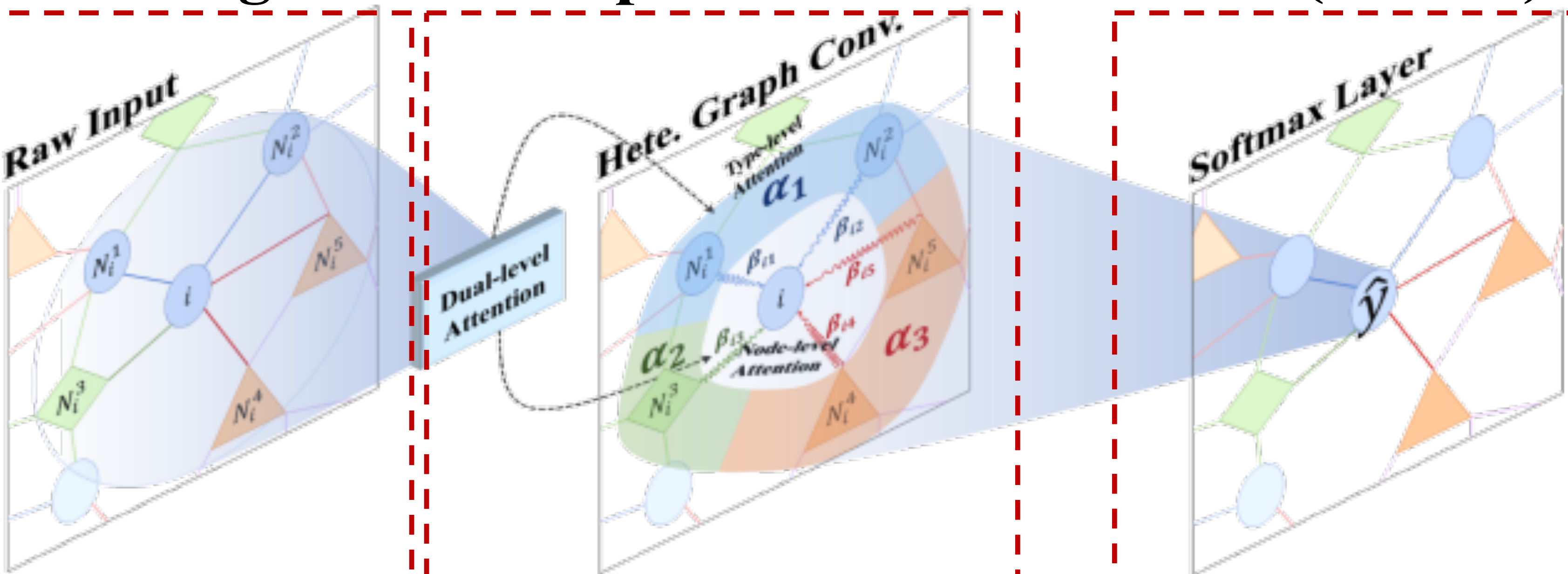
Nodes

- **Text**'s fea.: TF-IDF vectors of content
- **Topic**' fea.: word distributions on vocabulary
- **Entity**'s fea.: TF-IDF vectors of entity description

Edges

- **Text-Topic**: text assigned to top P topics
- **Text-Entity**: text contains entity
- **Entity-Entity**: similarity above a threshold

Heterogeneous Graph ATtention Network (HGAT)



Input:

HIN for short texts

Graph Convolution:

dual-level attention

differentiately aggregate

different-typed neighbors

Output:

Predict the document

label by softmax layer

Experimental Setting

Datasets

	#docs	#tokens	#entities	#classes
AGNews	6,000	18.4	0.9 (72%)	4
Snippets	12,340	14.5	4.4 (94%)	8
Ohsumed	7,400	6.8	3.1 (96%)	23
TagMyNews	32,549	5.1	1.9 (86%)	7
MR	10,662	7.6	1.8 (76%)	2
Twitter	10,000	3.5	1.1 (63%)	2

Metrics

- Accuracy (**ACC**)

Baselines

Typical methods

- SVM+TF-IDF
- SVM+LDA

Deep Methods

- CNN-rand
- CNN-pretrain
- LSTM-rand
- LSTM-pretrain

Graph Methods

- PTE
- TextGCN
- HAN

Our Variants

- GCN-HIN
- HGAT w/o ATT
- HGAT-Type
- HGAT-Node
- HGAT

Experimental Results

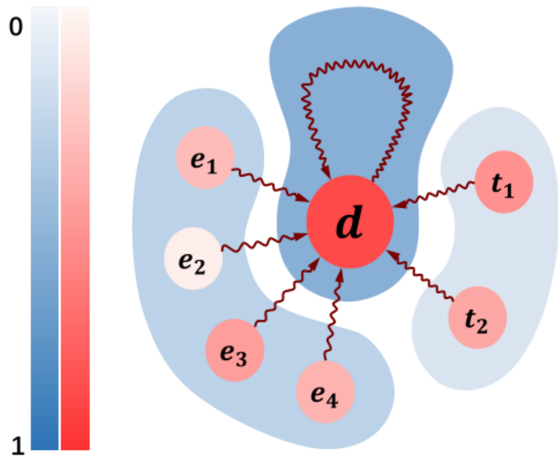
Dataset	SVM +TFIDF	SVM +LDA	CNN -rand	CNN -pretrain	LSTM -rand	LSTM -pretrain	PTE	TextGCN	HAN	HGAT
AGNews	57.73	65.16	32.65	67.24	31.24	66.28	36.00	<u>67.61</u>	62.64	**72.10***
Snippets	63.85	63.91	48.34	77.09	26.38	75.89	63.10	<u>77.82</u>	58.38	**82.36***
Ohsumed	41.47	31.26	35.25	32.92	19.87	28.70	36.63	<u>41.56</u>	36.97	**42.68***
TagMyNews	42.90	21.88	28.76	57.12	25.52	<u>57.32</u>	40.32	54.28	42.18	**61.72***
MR	56.67	54.69	54.85	58.32	52.62	<u>60.89</u>	54.74	59.12	57.11	**62.75***
Twitter	54.39	50.42	52.58	56.34	54.80	<u>60.28</u>	54.24	60.15	53.75	**63.21***

Table 2: Test accuracy (%) of different models on six standard datasets. The second best results are underlined. The note * means our model significantly outperforms the baselines based on t-test ($p < 0.01$).

Dataset	GCN -HIN	HGAT w/o ATT	HGAT -Type	HGAT -Node	HGAT
AGNews	70.87	70.97	71.54	71.76	**72.10***
Snippets	76.69	80.42	81.68	81.93	**82.36***
Ohsumed	40.25	41.31	41.95	42.17	**42.68***
TagMyNews	56.33	59.41	60.78	61.29	**61.72***
MR	60.81	62.13	62.27	62.31	**62.75***
Twitter	61.59	62.35	62.95	62.45	**63.21***

Table 3: Test accuracy (%) of our variants.

- ## Case Study



Short Text *d*

Shawn Green (Entity e_1) hit two home runs, as **Los Angeles (Entity e_2)** defeated the **Atlanta Braves (Entity e_3)** 7-4 in a battle of National League division leaders at **Dodger Stadium (Entity e_4)**.

Topic t_1:

game	sox	red	beat	team
clubs	season	win	astros	run

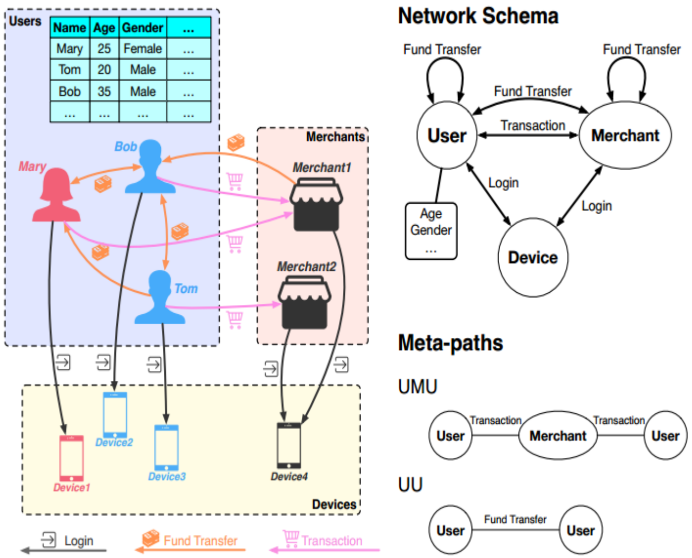
Topic t_2:

wins	awards	prix	star	prize
greek	china	grand	british	olympics

Figure 5: Visualization of the dual-level attention including node-level attention (shown in red) and type-level attention (shown in blue). Each topic t is represented by top 10 words with highest probabilities.

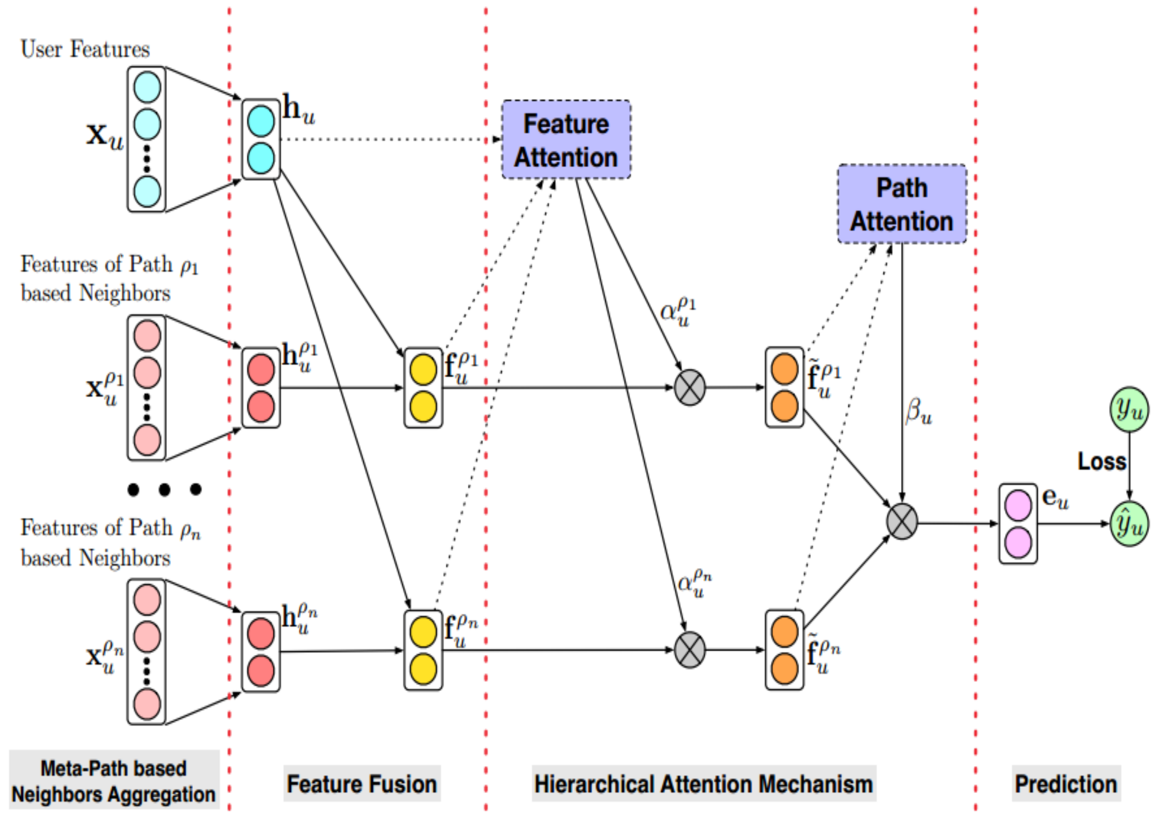
Cash-out User Detection

- Predict whether a user will do cash-out transactions



(a) Scenario of credit payment service

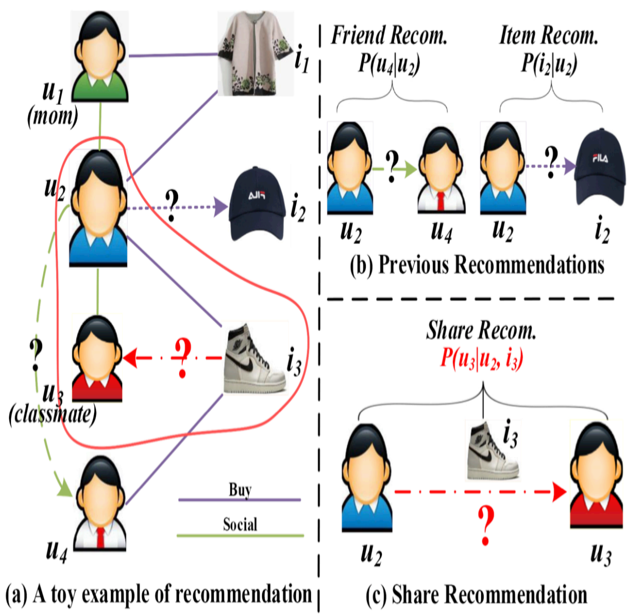
(b) Network schema and meta-path examples

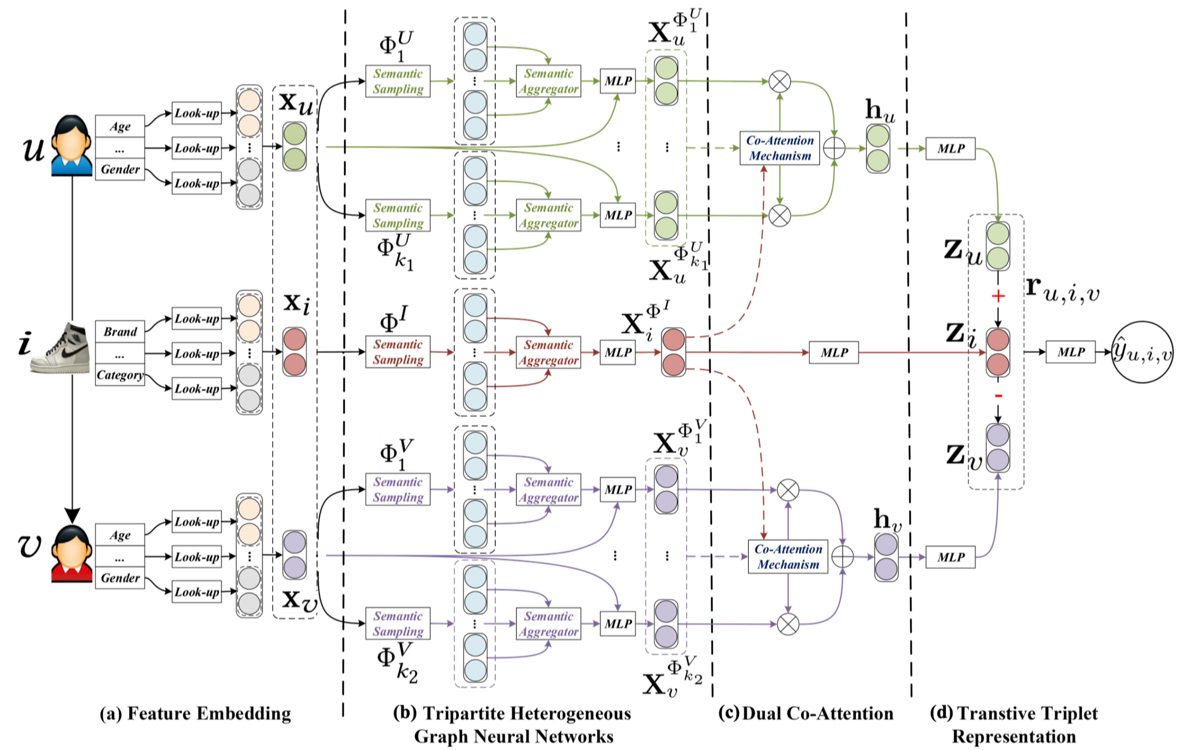


Binbin Hu, Zhiqiang Zhang, Chuan Shi, Jun Zhou, Xiaolong Li, Yuan Qi. Cash-out User Detection based on Attributed Heterogeneous Information Network with a Hierarchical Attention Mechanism. AAAI 2019.

Share Recommendation

- Predict whether a user will share an item with his friend *P(Friend|User, Item)*.



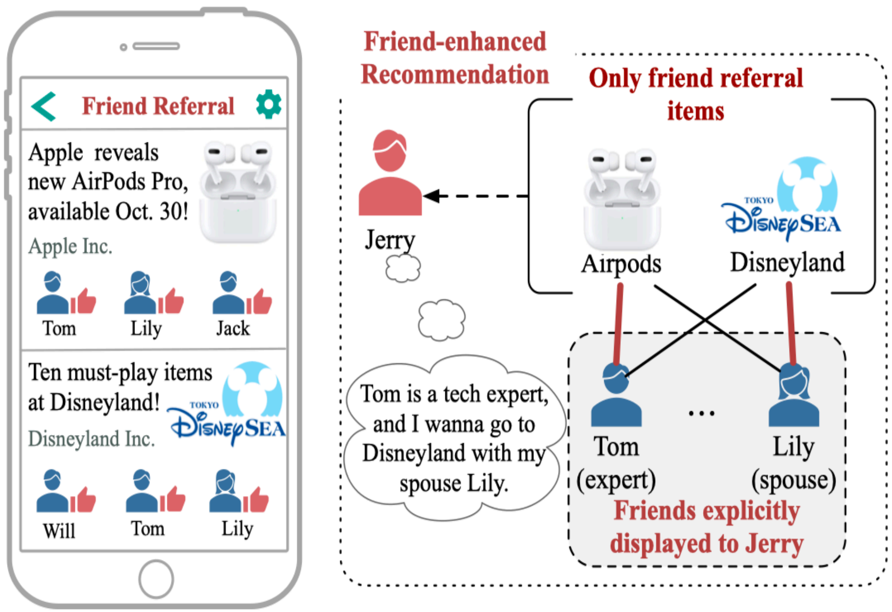


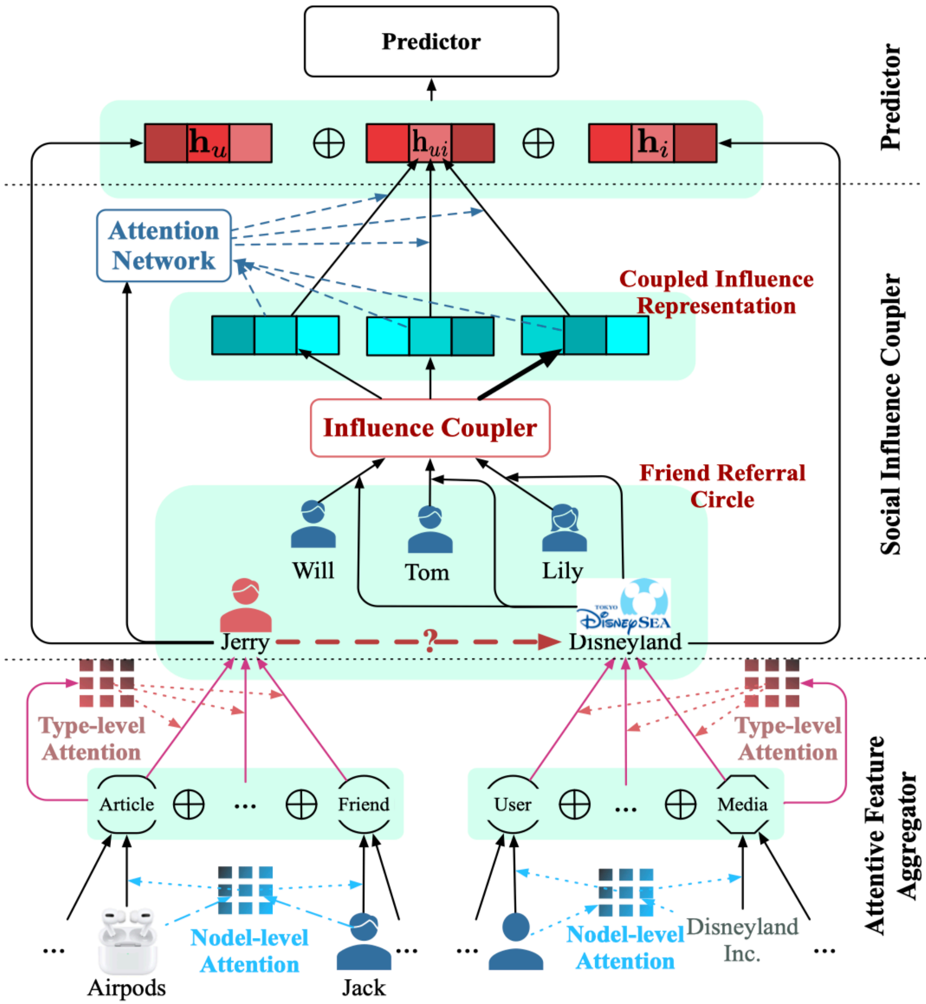
Houye Ji, Junxiong Zhu, Xiao Wang, Chuan Shi, Bai Wang, Xiaoye Tan, Yanghua Li, Shaojian He. Who You Would Like to Share With? A Study of Share Recommendation in Social E-commerce. AAAI2021.

Friend-enhanced Recommendation

Friend-enhanced Recommendation

- Only recommends items that friends have interacted with

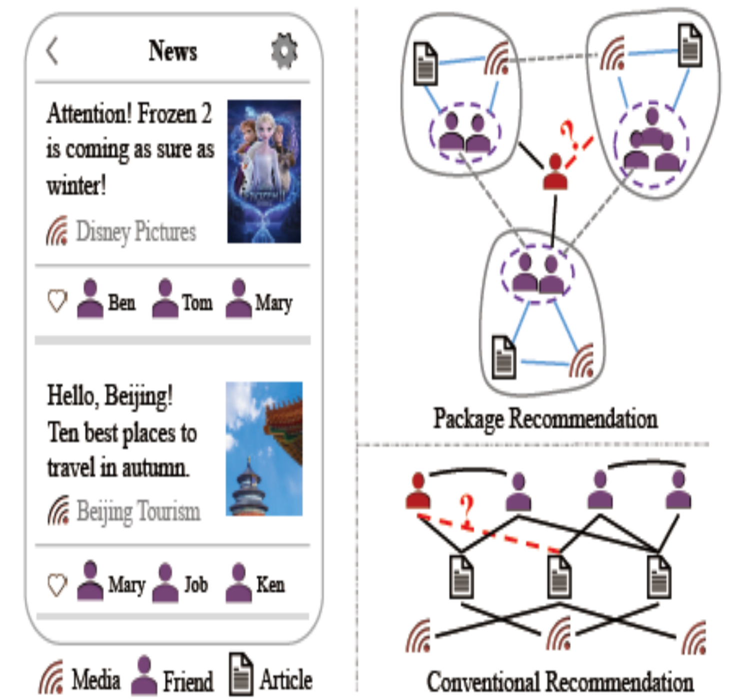




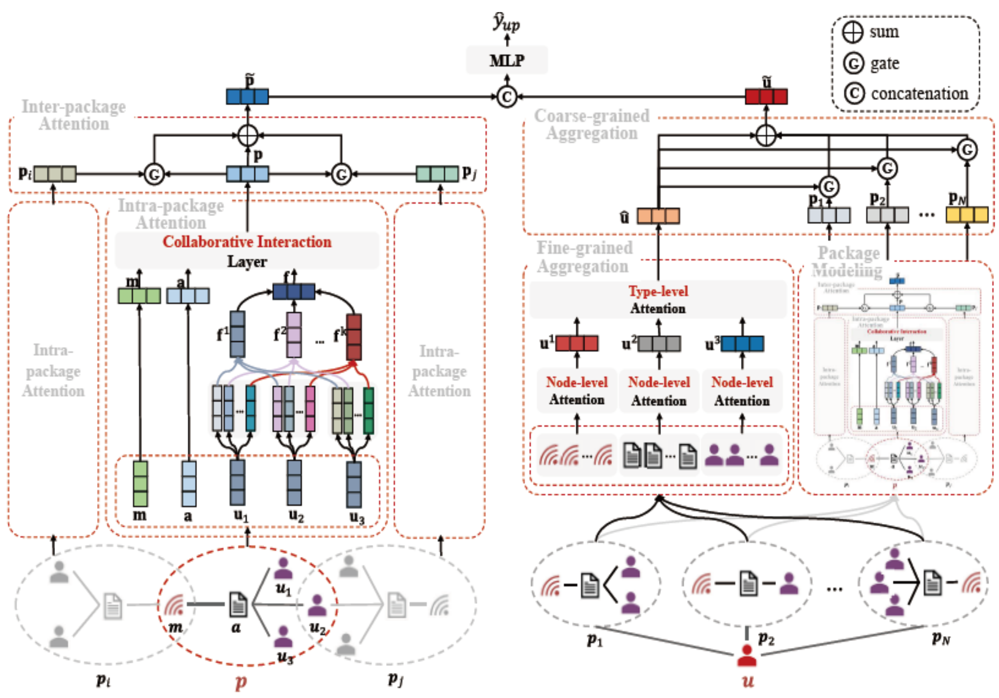
Yuanfu Lu, Ruobing Xie, Chuan Shi, Yuan Fang, et al. Social Influence Attentive Neural Network for Friend-Enhanced Recommendation. ECML-PKDD 2020.

Package recommendation

- Recommends a combination of heterogeneous and diverse objects (i.e. package) rather than a single item or item list



(a) Example of package recommendation (b) Comparison of recommendation paradigm



(a) Package Modeling (b) User Modeling

Chen Li, Yuanfu Lu, Wei Wang, Chuan Shi, Ruobing Xie, Haili Yang, Cheng Yang, Xu Zhang and Leyu Lin. Package Recommendation with Intra- and Inter-Package Attentive Networks. SIGIR 2021.

Outline

- Basic concepts
- Models
- Applications

√ Conclusion and future work

Technique summary

- Shallow model
  - Random walk-based
    - Semantic-aware random walk, e.g., metapath2vec, HHNE
    - Metagraph-guided random walks, e.g., metagraph2vec, spacey
  - Decomposition-based
    - Subgraph based, e.g., JERec, PME, PTE, HEBE
- Deep model
  - Message passing-based
    - Semantic-aware aggregation function, e.g., HAN, HetGNN, GTN
  - Encoder-decoder-based
    - Property-preserved autoencoder, e.g., HNE, Camel, DHNE
  - Adversarial-based
    - Relation-aware GAN, e.g., HeGAN
    - Adversarial completion, e.g., MV-ACM

Technique summary

TABLE 2: Typical heterogeneous graph embedding methods.

Method	Inductive	Label	Information	Task	Technique	Characteristic
mp2vec [8]			Strcuture	Embedding	Random walk (Shallow model)	• Easy to parallelize • Two-stage training • High memory cost Complexity: $\mathcal{O}(\tau \cdot l \cdot k \cdot n_s \cdot d \cdot \lvert\mathcal{V}\rvert)$
Spacey [59]						
JUST [60]						
BHIN2vec [61]						
HHNE [62]						
mg2vec [41]						
HeRec [2]		√	Strcuture+Task	Recommendation		
PME [17]			Strcuture	Embedding	Decomposition (Shallow model)	• Easy to parallelize • Two-stage training • High memory cost Complexity: $\mathcal{O}(\lvert\mathcal{E}\rvert \cdot d)$
EOE [50]						
HEER [53]						
MNE [57]						
PTE [17]						
RHINE [63]						
HAN [15]	√	√	Structure+Attribute		Message passing (Deep model)	• End-to-End training • Encoding structures and attributes • Semantic fusion • High training cost Complexity: $\mathcal{O}(\lvert\mathcal{V}\rvert \cdot d_1 + \lvert\mathcal{R}\rvert \cdot d_2)$
MAGNN [74]	√	√				
HetSANN [75]	√	√				
HGT [76]	√	√				
HetGNN [16]	√					
GATNE [72]	√					
GTN [79]		√				
RSHN [80]		√				
RGCN [81]	√	√				
IntentGC [20]	√	√	Strcuture +Attribute+Task	Recommendation		
MEIRec [19]	√	√				
GNUD [5]	√	√				
Player2vec [95]	√	√		Identification		
AHIN2vec [96]	√	√				
Vendor2vec [97]	√	√				
HIN2vec [9]			Strcuture	Embedding	Encoder-decoder (Deep model)	• End-to-End training • Flexible goal-orientation Complexity: $\mathcal{O}(\lvert\mathcal{V}\rvert \cdot d_1 + \lvert\mathcal{E}\rvert \cdot d_2)$
DHNE [65]						
HNE [69]	√	√	Structure+Attribute			
SHNE [70]		√				
NSHE [78]						
PAHNE [44]		√	Strcuture +Attribute+Task	Identification		
Camel [93]		√				
TaPEm [94]		√				
HeGAN [18]			Strcuture	Embedding	Adversarial (Deep model)	• Robustness • High complexity Complexity: $\mathcal{O}(\lvert\mathcal{V}\rvert \cdot \lvert\mathcal{R}\rvert \cdot n_s \cdot d)$
MV-ACM [120]						
Rad-HGC [24]		√	Strcuture+Task	Malware detection		

Xiao Wang, Deyu Bo, Chuan Shi, Shaohua Fan, Yanfang Ye, Philip S. Yu. A Survey on Heterogeneous Graph Embedding: Methods, Techniques, Applications and Sources. IEEE Transactions on Big Data, 2021.

Future directions

- Preserving HG structures and properties
  - Motif, Network schema – beyond metapath and metagraph
  - Dynamic
  - Uncertainty e.g., Gaussian distribution
- Deep graph learning on HG data
  - Over-smoothing, e.g., deep HGNN
  - Self-supervised learning
  - Pre-training, e.g., transfer ability
- Making HG embedding reliable
  - Fairness or debias, e.g., age and gender
  - Robust
  - Explainable, e.g., disentangled learning

Future directions

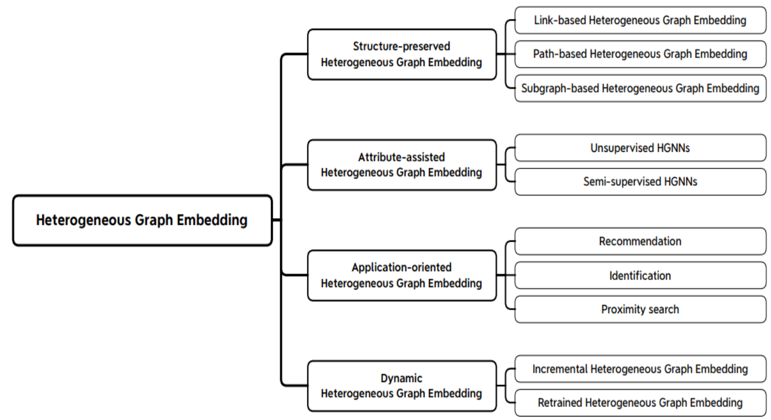
- Real-world applications
  - Software engineering
  - Biological system
  - Large-scale industrial scenarios
- Others
  - Non-Euclidean space embedding
    - e.g., Hyperbolic embedding
  - Heterogeneous graph structure learning
  - Connections with knowledge graph

More materials

A Survey on Heterogeneous Graph Embedding: Methods, Techniques, Applications and Sources

Xiao Wang, Deyu Bo, Chuan Shi†, *Member, IEEE,* Shaohua Fan, Yanfang Ye, *Member, IEEE,* and Philip S. Yu, *Fellow, IEEE*

Abstract—Heterogeneous graphs (HGs) also known as heterogeneous information networks have become ubiquitous in real-world scenarios; therefore, HG embedding, which aims to learn representations in a lower-dimension space while preserving the heterogeneous structures and semantics for downstream tasks (e.g., node/graph classification, node clustering, link prediction), has drawn considerable attentions in recent years. In this survey, we perform a comprehensive review of the recent development on HG embedding methods and techniques. We first introduce the basic concepts of HG and discuss the unique challenges brought by the heterogeneity for HG embedding in comparison with homogeneous graph representation learning; and then we systemically survey and categorize the state-of-the-art HG embedding methods based on the information they used in the learning process to address the challenges posed by the HG heterogeneity. In particular, for each representative HG embedding method, we provide detailed introduction and further analyze its pros and cons; meanwhile, we also explore the transformativeness and applicability of different types of HG embedding methods in the real-world industrial environments for the first time. In addition, we further present several widely deployed systems that have demonstrated the success of HG embedding techniques in resolving real-world application problems with broader impacts. To facilitate future research and applications in this area, we also summarize the open-source code, existing graph learning platforms and benchmark datasets. Finally, we explore the additional issues and challenges of HG embedding and forecast the future research directions in this field.

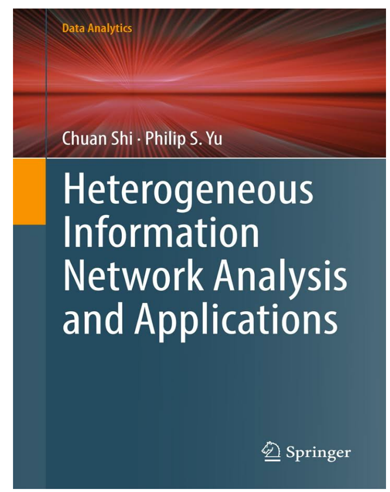


A Survey of Heterogeneous Information Network Analysis

Chuan Shi, *Member, IEEE,* Yitong Li, Jiawei Zhang, Yizhou Sun, *Member, IEEE,* and Philip S. Yu, *Fellow, IEEE*

Abstract—Most real systems consist of a large number of interacting, multi-typed components, while most contemporary researches model them as homogeneous information networks, without distinguishing different types of objects and links in the networks. Recently, more and more researchers begin to consider these interconnected, multi-typed data as heterogeneous information networks, and develop structural analysis approaches by leveraging the rich semantic meaning of structural types of objects and links in the networks. Compared to widely studied homogeneous information network, the heterogeneous information network contains richer structure and semantic information, which provides plenty of opportunities as well as a lot of challenges for data mining. In this paper, we provide a survey of heterogeneous information network analysis. We will introduce basic concepts of heterogeneous information network analysis, examine its developments on different data mining tasks, discuss some advanced topics, and point out some future research directions.

Index Terms—Heterogeneous information network, data mining, semi-structural data, meta path



- More materials in my webpage: www.shichuan.org

References

- Xiao Wang, Meiqi Zhu, Deyu Bo, Peng Cui, Chuan Shi, Jian Pei.AM-GCN: Adaptive Multi-channel Graph Convolutional Networks. KDD 2020.
- Meiqi Zhu, Xiao Wang, Chuan Shi, Houye Ji, Peng Cui. Interpreting and Unifying Graph Neural Networks with An Optimization Framework. Under Review.
- Deyu Bo, Xiao Wang, Chuan Shi, Huawei Shen. Frequency Adaptation Graph Convolutional Networks. AAAI 2021.
- Chuan Shi, Philip S. Yu. Heterogeneous Information Network Analysis and Applications. Springer. ISBN 978-3-319-56211-7. 2017.
- Chuan Shi, Yitong Li, Jiawei Zhang, Yizhou Sun, Philip S. Yu. A survey on Heterogeneous Information Network Analysis. IEEE Transactions on Knowledge and Data Engineering, 29(1), 17-37, 2017.
- Chuan Shi, Binbin Hu, Wayne Xin Zhao, Philip S. Yu. Heterogeneous Information Network Embedding for Recommendation. IEEE Transactions on Knowledge and Data Engineering, 2018.
- Binbin Hu, Chuan Shi, Xin Zhao, Philip S. Yu. Leveraging Meta-path based Context for Top-N Recommendation with A Neural Co-Attention Model. KDD 2018.
- Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Peng Cui, Philip S. Yu, Yanfang Ye. Heterogeneous Graph Attention Network. WWW 2019.
- Jianan Zhao, Xiao Wang, Chuan Shi, Zekuan Liu, and Yanfang Ye. Network Schema Preserving Heterogeneous Information Network Embedding. IJCAI 2020.
- Yuanfu Lu, Yuan Fang, Chuan Shi. Meta-learning on Heterogeneous Information Networks for Cold-start Recommendation. KDD 2020.
- Jianan Zhao, Xiao Wang, Chuan Shi, Binbin Hu, Guojie Song and Yanfang Ye. Heterogeneous Graph Structure Learning for Graph Neural Networks. AAAI 2021.
- Yuanfu Lu, Xiao Wang, Chuan Shi, Philip S. Yu, Yanfang Ye. Temporal network embedding with micro- and macro-dynamics. CIKM 2019.
- Xiao Wang, Yuanfu Lu, Chuan Shi, Ruijia Wang, Peng Cui, Shuai Mou. Dynamic Heterogeneous Information Network Embedding with Meta-path based Proximity. TKDE 2020.
- Yugang Ji, MingYang Yin, Yuan Fang, Hongxia Yang, Xiangwei Wang, Tianrui Jia and Chuan Shi. Temporal Heterogeneous Interaction Graph Embedding For Next-Item Recommendation. ECML-PKDD 2020.
- Deyu Bo, Xiao Wang, Chuan Shi, Meiqi Zhu, Emiao Lu, Peng Cui.Structural Deep Clustering Network. WWW 2020.
- Xiao Wang, Ruijia Wang, Chuan Shi, Guojie Song, Qingyong Li. Multi-Component Graph Convolutional Collaborative Filtering. AAAI 2020.

References

- Ji S, Pan S, Cambria E, et al. A Survey on Knowledge Graphs: Representation, Acquisition and Applications[C]. arXiv preprint arXiv:2002.00388, 2020
- Xia C, Zhang C, Yang T, et al. Multi-grained named entity recognition[C]. ACL, 2019, pp.1430–1440.
- Huang H, Heck L, Ji H. Leveraging deep neural networks and knowledge graphs for entity disambiguation. arXiv preprint arXiv:1504.07678, 2015.
- Ren X, He W, Qu M, et al. Label noise reduction in entity typing by heterogeneous partial-label embedding[C]. SIGKDD, 2016, pp. 1825–1834.
- Zhang Y, Qi P, Manning C D. Graph Convolution over Pruned Dependency Trees Improves Relation Extraction[A]. Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing[C]. Association for Computational Linguistics, 2018: 2205–2215.
- Hu L, Ding J, Shi C, et al. Graph neural entity disambiguation[J]. Knowledge-Based Systems, 2020, 195: 105620.
- Zhu H, Xie R, Liu Z., et al. Iterative entity alignment via joint knowledge embeddings[C]. IJCAI, 2017, pp. 4258–4264.
- Zhang Z, Zhuang F, Zhu H, et al. Relational Graph Neural Network with Hierarchical Attention for Knowledge Graph Completion[J]. Proceedings of the AAAI Conference on Artificial Intelligence, 2020, 34(05): 9612–9619
- Bordes A, Usunier N, Garcia-Duran A, et al. Translating Embeddings for Modeling Multi-relational Data[C]. NeurIPS, 2013: 2787-2795.
- Schlichtkrull M, Kipf T N, Bloem P, et al. Modeling Relational Data with Graph Convolutional Networks[A]. European Semantic Web Conference[C]. Springer, 2018: 593–607.
- Hu L, Zhang M, Shi J, et al. Text-Graph Enhanced Knowledge Graph Representation Learning. Under Review. 2020.
- Lin B Y, Chen X, Chen J, et al. KagNet: Knowledge-Aware Graph Networks for Commonsense Reasoning[C]// Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, 2019: 2829–2839.
- Hu L, Yang T, Shi C, et al. Heterogeneous Graph Attention Networks for Semi-supervised Short Text Classification[A]. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing[C]. Association for Computational Linguistics, 2019: 4821–4830.
- Wang H, Zhao M, Xie X, et al. Knowledge Graph Convolutional Networks for Recommender Systems[A]. Proceedings of the 2019 World Wide Web Conference[C]. ACM Press, 2019: 3307–3313.

Thanks !

GAMMA Lab:

Xiao Wang, Cheng Yang, Bai Wang, Binbin Hu, Houye Ji, Yuanfu Lu, Yugang Ji, Shaohua Fan, Jianan Zhao, Deyu Bo, Ruijia Wang, Meiqi Zhu, Tianchi Yang, Nian Liu, Xunqiang Zhang, Chen Li

Collaborators:

Philip Yu, Yuan Fang, Fanny Ye, Xin Zhao, Peng Cui, Guojie Song, Zhiyuan Liu, Zhiqiang Zhang, Ruobin Xie, Junxiong Zhu

图数据挖掘和机器学习

▶ 扫码关注我们 ◀

www.shichuan.org