Learning Heterogeneous Information Networks for Link Prediction

Speaker: Dr. Hongxu Chen
Host: Dr. Yulei Sui
16/10/2020

My basic information

Hongxu CHEN. 陈红旭

Education Background

- 2009.9 – 2013.6, GUET, B.Sc.
- 2015.2 - 2015.12, The University of Queensland, Master of Computer Science.
- 2016.4 - 2019.10, The University of Queensland, Ph.D. in Computer Science.
  - Supervisor: Dr. Hongzhi Yin & Prof. Xue Li

Education Background

- 2019. 11 – now: Postdoc Research Fellow in Network Science Lab, UTS

My basic information

Professional Services:

- Publicity Co-Chair: BSEC 2020
- Program Committee Member：ADMA20, ICONIP 2020, CCL 2020, EMERGING 2020, ADMA19, RSBSD2019, ADMA2018, BDASC'18
- Invited Reviewer (conference): KDD20, IJCAI20, SIGIR20, VLDB20, CIKM19, ICDM19, KDD19, VLDB19, SIGIR19, AAAI19, PAKDD19, ICDM8, CIKM18, ICDM17, CIKM17, WISE17, WISE18.
- Invited Journal Reviewer:
  - ❑ Editorial Board of Complexity Journal.
  - ❑ *IEEE Transactions on Knowledge and Data Engineering (TKDE)*
  - ❑ *WWW Journal*
  - ❑ *VLDB Journal*
  - ❑ *IEEE Transactions on Systems, Man and Cybernetics: Systems*
  - ❑ *Journal of Complexity*
  - ❑ *ACM Transactions on Data Science.*
  - ❑ *Journal of Computer Science & Technology.*
  - ❑ *IEEE Access*

My basic information

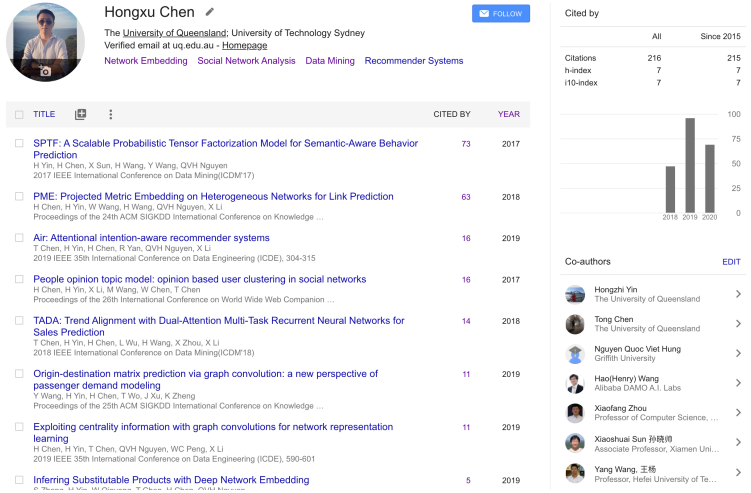
Research profile:

Homepage:

https://sites.google.com/view/hxchen

Google Scholar:

https://scholar.google.com.au/citations?user=W3CtDGQAAAAJ&hl=en



Hongxu Chen

FOLLOW

The University of Queensland; University of Technology Sydney
Verified email at uq.edu.au - Homepage

Network Embedding Social Network Analysis Data Mining Recommender Systems

TITLE	CITED BY	YEAR
SPTF: A Scalable Probabilistic Tensor Factorization Model for Semantic-Aware Behavior Prediction H Yin, H Chen, X Sun, H Wang, Y Wang, QVH Nguyen 2017 IEEE International Conference on Data Mining(ICDM'17)	73	2017
PME: Projected Metric Embedding on Heterogeneous Networks for Link Prediction H Chen, H Yin, W Wang, H Wang, QVH Nguyen, X Li Proceedings of the 24th ACM SIGKDD International Conference on Knowledge ...	63	2018
Air: Attentional intention-aware recommender systems T Chen, H Yin, H Chen, R Yan, QVH Nguyen, X Li 2019 IEEE 35th International Conference on Data Engineering (ICDE), 304-315	16	2019
People opinion topic model: opinion based user clustering in social networks H Chen, H Yin, X Li, M Wang, W Chen, T Chen Proceedings of the 26th International Conference on World Wide Web Companion ...	16	2017
TADA: Trend Alignment with Dual-Attention Multi-Task Recurrent Neural Networks for Sales Prediction T Chen, H Yin, H Chen, L Wu, H Wang, X Zhou, X Li 2018 IEEE International Conference on Data Mining(ICDM'18)	14	2018
Origin-destination matrix prediction via graph convolution: a new perspective of passenger demand modeling Y Wang, H Yin, H Chen, T Wo, J Xu, K Zheng Proceedings of the 25th ACM SIGKDD International Conference on Knowledge ...	11	2019
Exploiting centrality information with graph convolutions for network representation learning H Chen, H Yin, T Chen, QVH Nguyen, WC Peng, X Li 2019 IEEE 35th International Conference on Data Engineering (ICDE), 590-601	11	2019
Inferring Substitutable Products with Deep Network Embedding	5	2019

Cited by

	All	Since 2015
Citations	216	215
h-index	7	7
i10-index	7	7

Co-authors EDIT

- Hongzhi Yin, The University of Queensland
- Tong Chen, The University of Queensland
- Nguyen Quoc Viet Hung, Griffith University
- Hao(Henry) Wang, Alibaba DAMO A.I. Labs
- Xiaofang Zhou, Professor of Computer Science, ...
- Xiaoshuai Sun 孙晓帅, Associate Professor, Xiamen Uni...
- Yang Wang, 王杨, Professor, Hefei University of Te...

Dr. Hongxu CHEN 陈红旭

ABOUT:

Dr. Hongxu Chen is now working as a Postdoctoral Research Fellow in Network Science Lab at University of Technology Sydney (UTS). Hongxu completed his Ph.D. at The University of Queensland (UQ), Australia, under the supervision of Dr. Hongzhi Yin and Prof. Xue Li In Data Science Research Group , School of Information Tech. and Electrical Engineering (ITEE).

His research interests include:

- `Network Science`
- `Complex Network Mining`
- `Network Embedding`
- `Recommender Systems`
- `Social Networks Modelling and Analytics`

Office: 11.07.112 UTS, 123 Broadway, Ultimo, Sydney.

E-mail: hongxu.chen[*at*]uts.edu.au

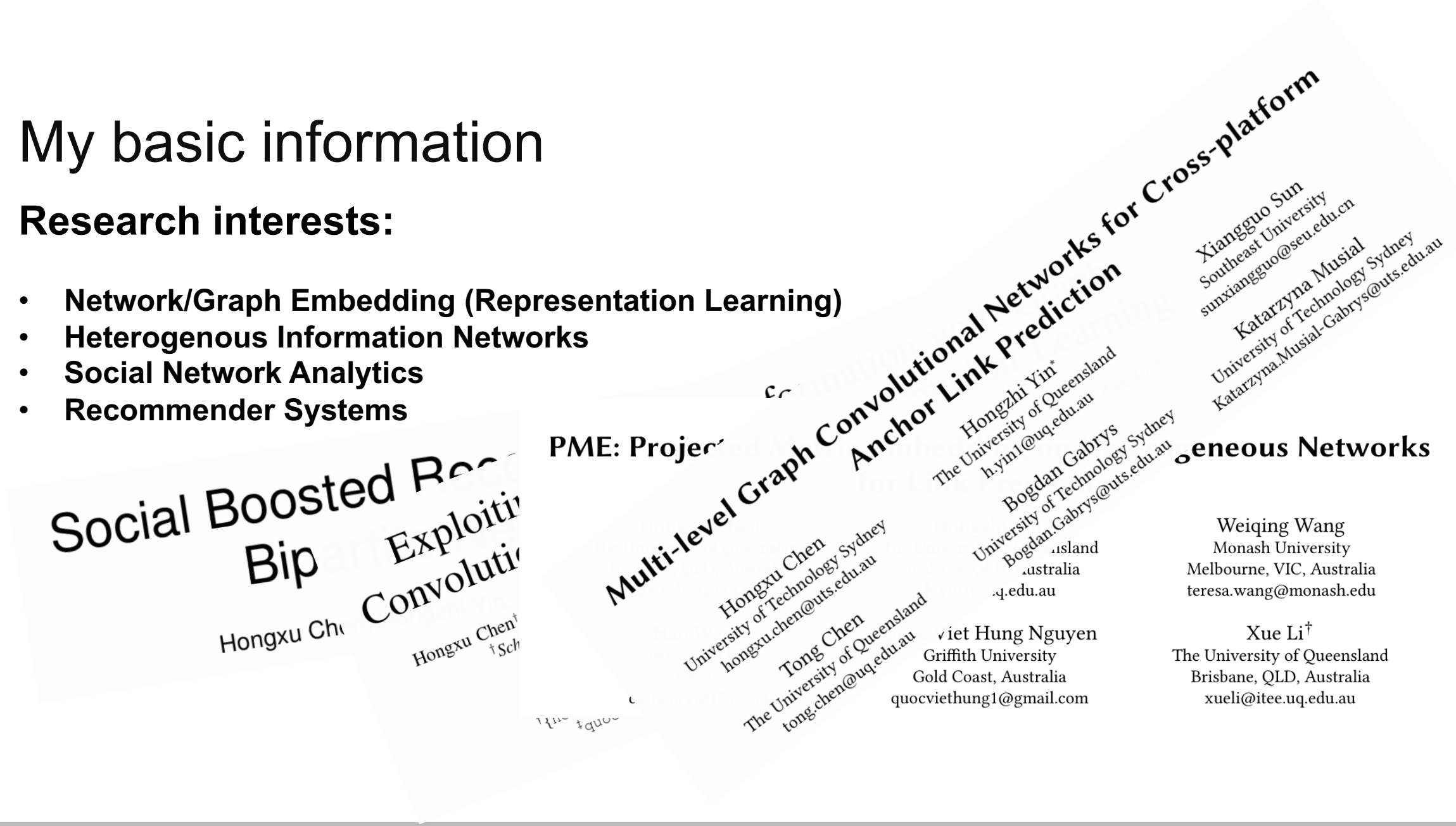
Professional Services:

- Editorial Board of Complexity Journal.
- Publicity Co-Chair: BSEC 2020
- Program Committee Member： ADMA20, ICONIP 2020, CCL 2020, EMERGING 2020, ADMA19, RSBSD2019, ADMA2018, BDASC'18
- Invited Reviewer (conference): KDD20, IJCAI20, SIGIR20, VLDB20, CIKM19, ICDM19, KDD19, VLDB19, SIGIR19, AAAI19, PAKDD19, ICDM8, CIKM18, ICDM17, CIKM17, WISE17, WISE18.
- Invited Journal Reviewer:

My basic information

Research interests:

- **Network/Graph Embedding (Representation Learning)**
- **Heterogenous Information Networks**
- **Social Network Analytics**
- **Recommender Systems**



Outline

- Network Embedding
- Link prediction on Heterogenous Information Networks.
  - Projected Metric Embedding (KDD18)
- Anchor link prediction across different platforms.
  - Multi-level GCN (KDD20)

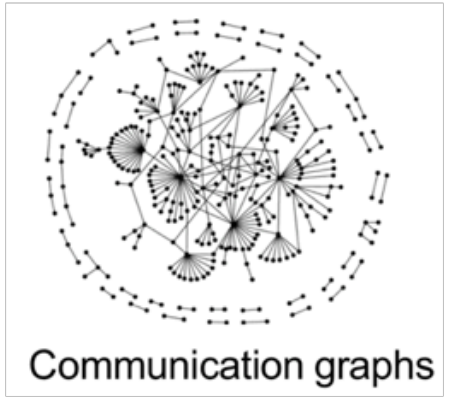
Outline

- Network Embedding
- Link prediction on Heterogenous Information Networks.
  - Projected Metric Embedding (KDD18)
- Anchor link prediction across different platforms.
  - Multi-level GCN (KDD20)

Networks are ubiquitous

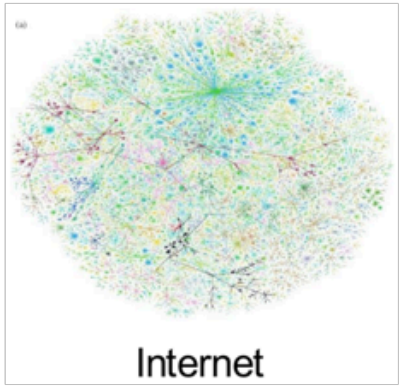
Social networks

Road networks

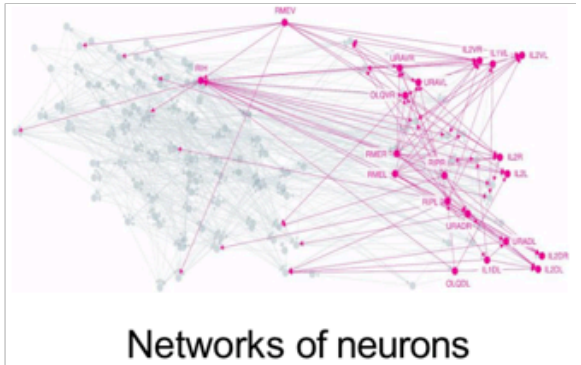


Communication graphs

Information networks:
Web & citations



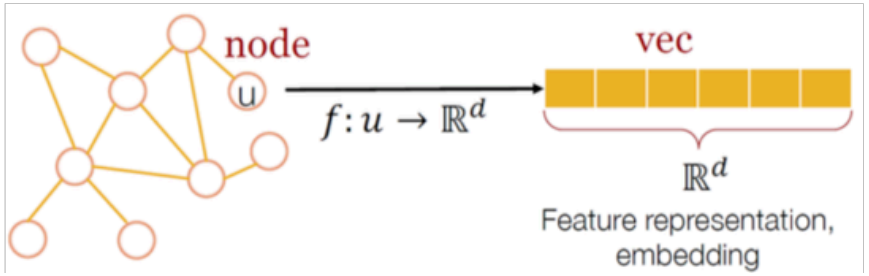
Internet



Networks of neurons

Represent networks by vectors

- Graph Representation Learning.
  - Also known as Graph Embedding or Network Embedding.
  - Low-dimensional vector for vertices.
  - Effectively preserve network structure.



  - Downstream data mining tasks on graphs:
    - ✓ Link prediction.
    - ✓ Node classification.
    - ✓ Recommendation.
    - ✓ …

Outline

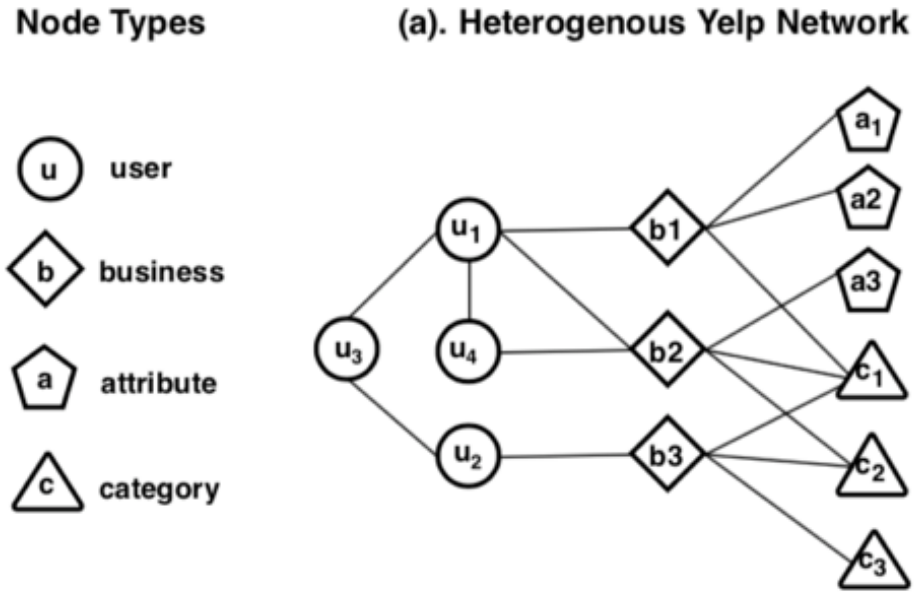
- Network Embedding
- Link prediction on Heterogenous Information Networks.
  - Projected Metric Embedding (KDD18)
- Anchor link prediction across different platforms.
  - Multi-level GCN (KDD20)

Projected Metric Embedding (KDD18)

Homogenous Networks

- Single-Typed Nodes.
- Single-Typed Links.

Projected Metric Embedding (KDD18)



The Real World: Heterogenous Networks

- Multiple object types.
- Multiple link types.

E.g., vertex u1 is close to both b2 and u3, but these relationships have different semantics. b2 is a business visited by user u1, while u3 is a friend of u1.

Projected Metric Embedding (KDD18)

We argue that –

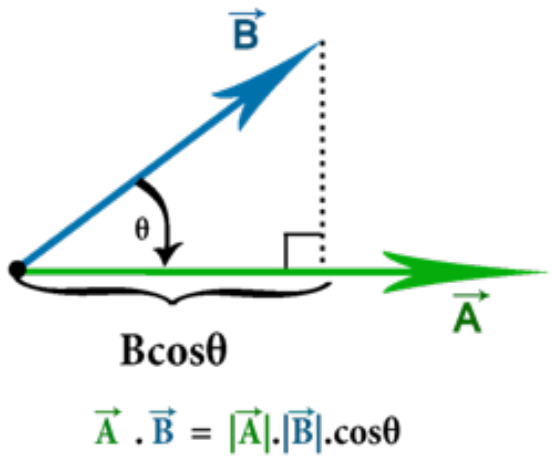
Homogenous networks are information loss projection of heterogenous networks.

- ❑ However, most of existing methods only focus on Homogenous Network Embedding that equally treats each type of nodes and links.
- ❑ Directly mining information-richer heterogenous networks.

- ❑ Compared to homogenous network embedding

  The proximity between objects in a HIN is:
  - Not just a measure of closeness or distance.
  - But it is also based on semantics.

Projected Metric Embedding (KDD18)



- Previous attempts.
  To model semantic-specific relationships:
  - Metapath2vec (Dong et. al., KDD 2017)
  - EOE (Xu et. al., WSDM 2017)
  - Dot product is used to compute the proximity between different types of nodes
  - Not a metric based distance
  - Violates the crucial triangle inequality

- Node A is close to Node C
- Node B is close to Node C

} Node A is also close to Node B

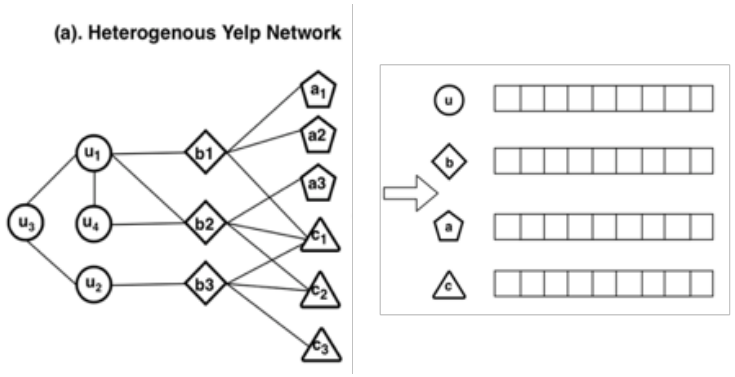
Therefore, existing HIN embedding methods (e.g., Metapath2vec and EOE)

- Can only capture local structures (both A and B are close to C)
- But fail to capture the second-order proximities. (A and B are also close)

Projected Metric Embedding (KDD18)

Our proposed method

- ❑ Projected Metric Embedding for HIN
  - ❖ Simultaneously preserves:
    - ➢ The first-order proximities between nodes
    - ➢ And the second-order proximities between nodes



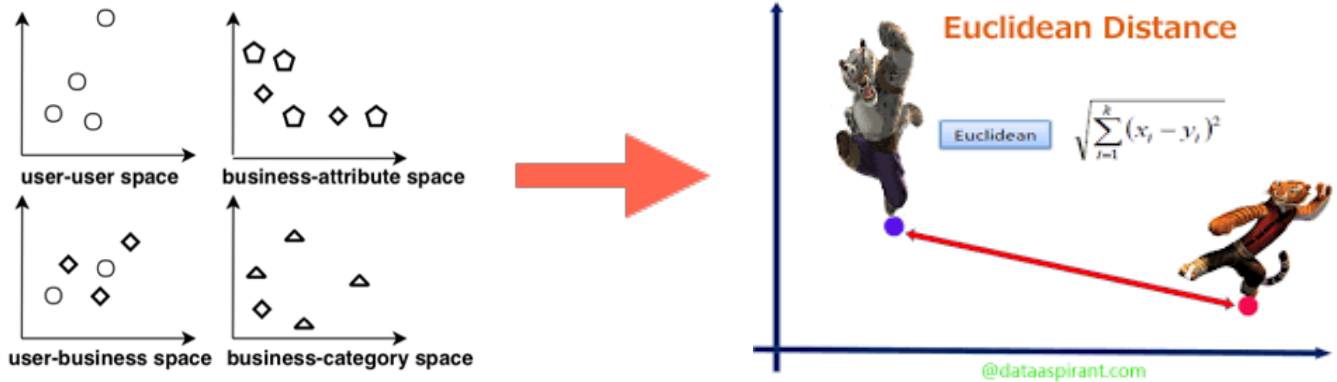
Such that, in the latent space,

$$D(u,v) < D(u,k); e_{uv} \in E, e_{u,k} \notin E$$

Projected Metric Embedding (KDD18)

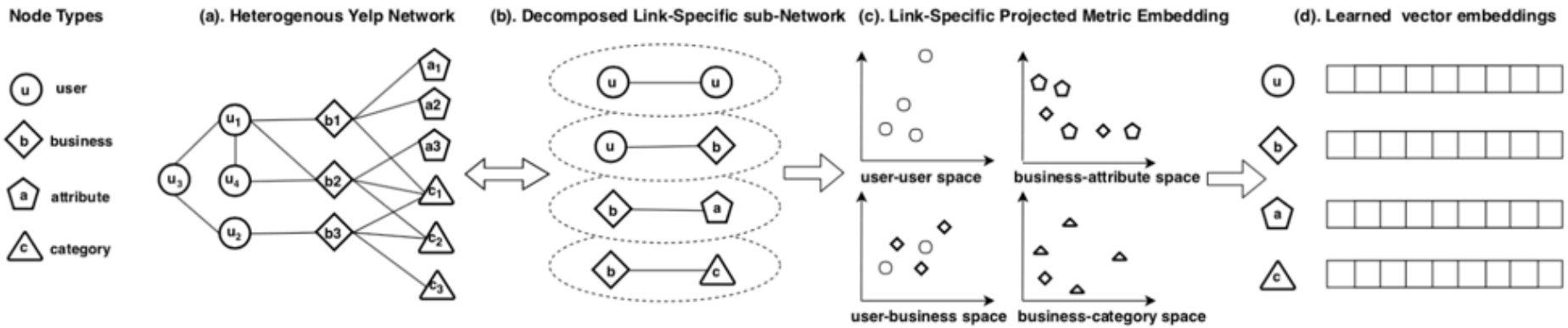
- However, directly applying the Euclidean distance as a metric will be problematic !
  - Mathematically, It is geometrically restrictive and an ill-posed algebraic system.
  - On the other hand, one object may have multiple aspects.

- To address these issues:
  - PME introduces relation-specific projection embedding matrices.
  - Model objects and relations in distinct spaces.
    - One shared object space.
    - Multiple relation spaces. (relation-specific object spaces).



Hence, it is possible that some objects are far away from each other in the object space, but are close to each other in the corresponding relation spaces.

Projected Metric Embedding (KDD18)



- Decompose the HIN to link-specific sub-networks.
- Metric learning on each link-specific space.

$$\mathbf{v}_i^r = \mathbf{M_r}\mathbf{v_i} \qquad d_r(v_i, v_j) = \|\mathbf{M_r}\mathbf{v_i} - \mathbf{M_r}\mathbf{v_j}\|, r \in \mathcal{R}$$

- Learn shared node embeddings and relation-specific space.

$$\min_{\mathbf{v}_*, \mathbf{M}_*} \sum_{r \in \mathcal{R}} \sum_{(v_i, v_j) \in D_r} \sum_{(v_i, v_k) \notin D_r} [m + f_r(v_i, v_j)^2 - f_r(v_i, v_k)^2]_+$$
$$s.t. \quad \|\mathbf{v}_*\| <= 1 \quad and \quad \|\mathbf{M}_*\| <= 1$$

Projected Metric Embedding (KDD18)

Model optimization:

- Bi-directional Negative Sampling Strategy

$$\min_{\mathbf{v}_*, \mathbf{M}_*} \sum_{r \in \mathcal{R}} \sum_{(v_i, v_j) \in D_r} \sum_{(v_i, v_k) \notin D_r} [m + f_r(v_i, v_j)^2 - f_r(v_i, v_k)^2]_+$$
$$s.t. \quad \|\mathbf{v}_*\| <= 1 \quad and \quad \|\mathbf{M}_*\| <= 1$$

- Directly optimizing this equation is expensive!

  Inspired by the negative sampling techniques

$$O = \sum_{r \in \mathcal{R}} \sum_{(v_i, v_j) \in D_r} \Big(\sum_{k=1}^{K} E_{v_k \sim p_n(v)} [m + f_r(v_i, v_j)^2 - f_r(v_i, v_k)^2]_+ + \sum_{k=1}^{K} E_{v_k \sim p_n(v)} [m + f_r(v_i, v_j)^2 - f_r(v_k, v_j)^2]_+ \Big)$$

first fix vertex v_i and edge type r, then generate K negative vertices v_k

then fix right side of e_{ijr} , and sample K negative vertex from the left side

Projected Metric Embedding (KDD18)

Model optimization:

- ❑ Loss-aware Adaptive Positive Sampling Strategy

1. **A sequence of the losses for each sub-network.**

$$L = (l_1, l_2, l_3, ..., l_{\|\mathcal{R}\|})$$

2. **Simply calculate the sum of the losses.**

$$L_{sum} = \sum_{r \in \mathcal{R}} l_r$$

3. **Draw a random value within the range of [0,1].**

$$x \sim Uniform(0,1)$$

4. **To see which interval the random fails into.**

$$[\sum_{j=0}^{r-1} \frac{l_j}{L_{sum}}, \sum_{j=0}^{r} \frac{l_j}{L_{sum}})$$

Algorithm 1 Training PME model

Input: A heterogeneous network $G(V, E, W, \mathcal{R})$, number of stochastic gradient steps, N, number of negative samples for each positive sample, K;
Output: Embeddings for network vertices and relation-specific projection matrix. (i.e., $\mathbf{v}, M_r$);

```
iter ← 0 :
while iter < N do
    if iter = 0 then
        Initialize the positive sampling probability as proportional to the original link distribution from G;
    else
        Sample M positive examples based on adaptive positive sampling strategy;
    End if
    For each sampled positive edge, sample K negative vertices from both sides of the edge;
    Compute gradients and update parameters;
    Censor the norm of v and projection matrix M_r;
    Compute relation-specific subgraph loss, and update the positive sampling probability;
    iter ← iter + 1 :
end
```

Projected Metric Embedding (KDD18)

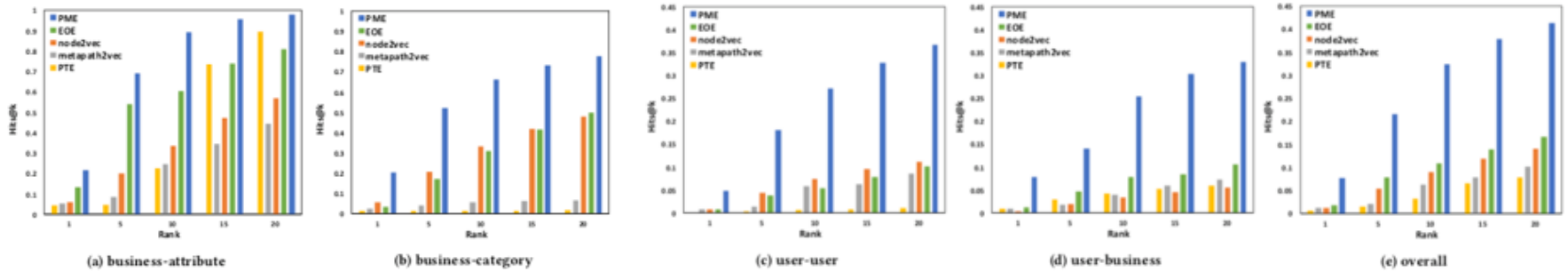
Experimental Results

- ❖ Binary Link Classification (Yelp challenge dataset)

Table 4: AUC scores on NV network

	PME	node2vec	PTE	EOE	metapath2vec
Overall	**0.9618**	0.8789	0.7494	0.8562	0.6232
user-user	**0.9672**	0.8909	0.6347	0.9033	0.5141
user-business	**0.9590**	0.8835	0.8615	0.9129	0.8179
business-attribute	**0.9376**	0.7522	0.8944	0.9201	0.5653
business-category	**0.9896**	0.9233	0.9652	0.9819	0.7725

- ❖ Prediction accuracy (Hit ratio) $Hits@k = \frac{\#hit@k}{\|D^{+}_{test}\|}$



(a) business-attribute (b) business-category (c) user-user (d) user-business (e) overall

Projected Metric Embedding (KDD18)

Code: https://www.dropbox.com/s/o9h7pgkovurtyud/KDD18-Hongxu.zip?dl=0&file_subpath=/KDD18-Hongxu

References:

[1]. Dong, Yuxiao, Nitesh V. Chawla, and Ananthram Swami. "metapath2vec: Scalable representation learning for heterogeneous networks." In Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining, pp. 135-144. 2017.

[2]. Xu, Linchuan, Xiaokai Wei, Jiannong Cao, and Philip S. Yu. "Embedding of Embedding (EOE) Joint Embedding for Coupled Heterogeneous Networks." In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, pp. 741-749. 2017.

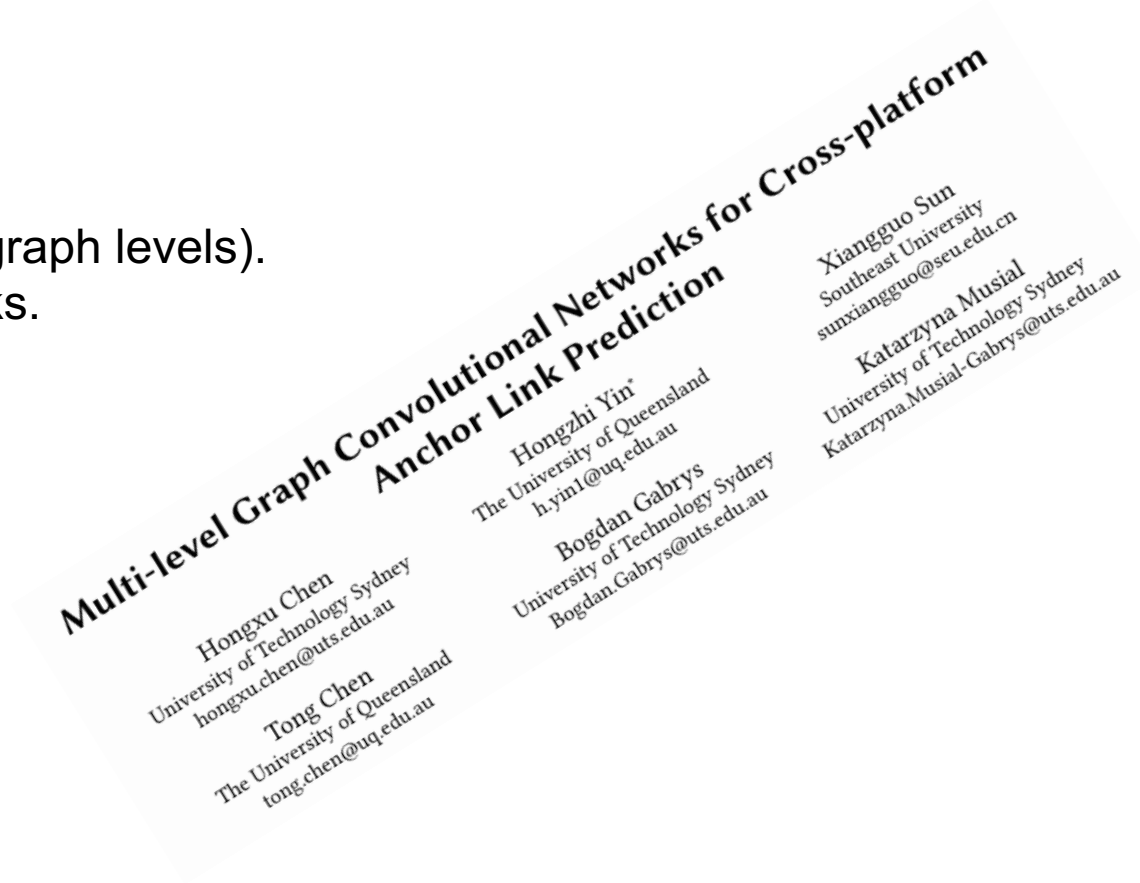
Outline

- Network Embedding
- Link prediction on Heterogenous Information Networks.
  - Projected Metric Embedding (KDD18)
- Anchor link prediction across different platforms.
  - Multi-level GCN (KDD20)

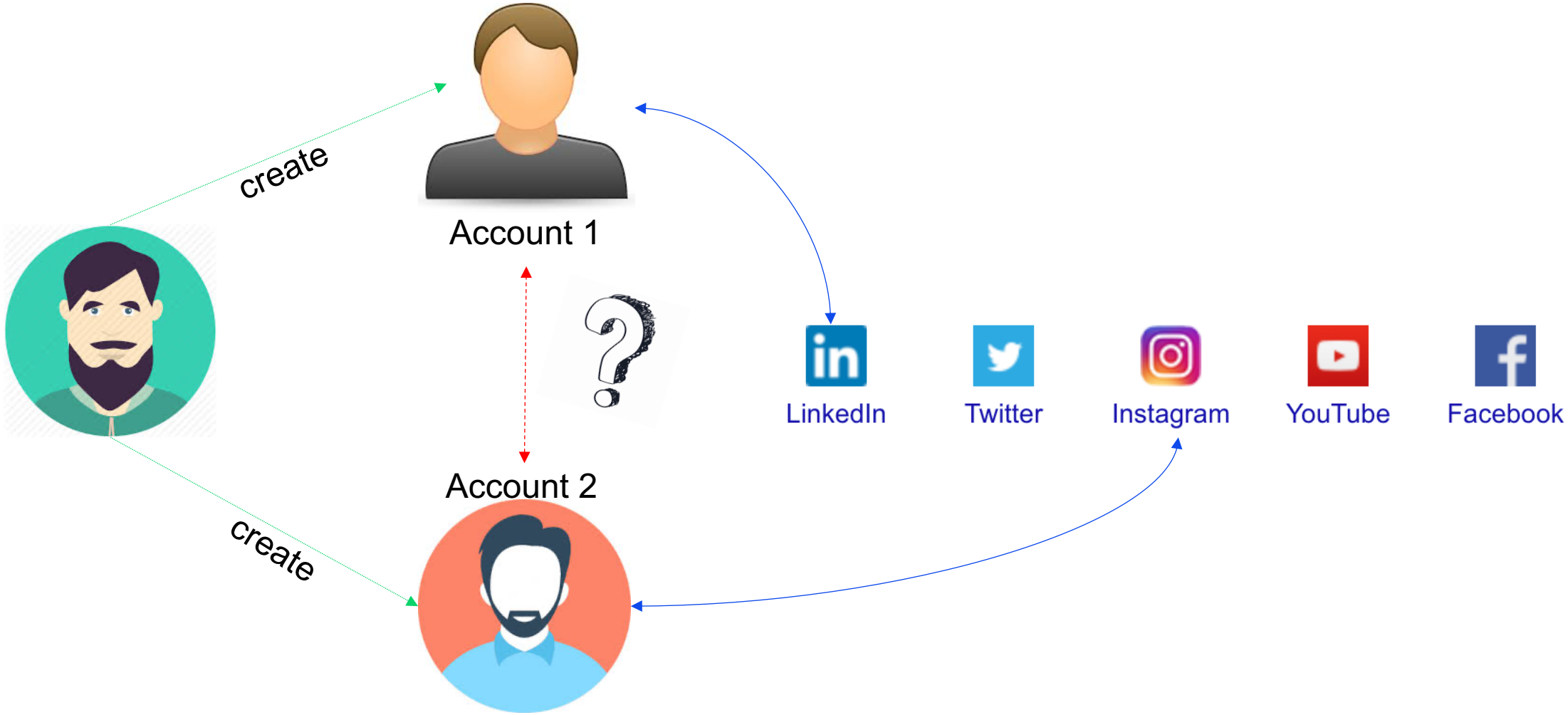
Multi-level Graph Convolutional Networks (KDD20)

Highlights:

- ❑ Cross-platform account matching.
- ❑ Multi-level GCN framework. (Local and Hypergraph levels).
- ❑ Treatments for adapting to large-scale networks.
  - ❖ Network partitioning.
  - ❖ Sub-space reconciliation.



Multi-level Graph Convolutional Networks (KDD20)



Multi-level Graph Convolutional Networks (KDD20)

- Cross-platform account matching:
  - a.k.a. Account Matching, Social Network De-anonymization, Social Anchor Link Prediction.
  - Beneficial to wide range of applications.

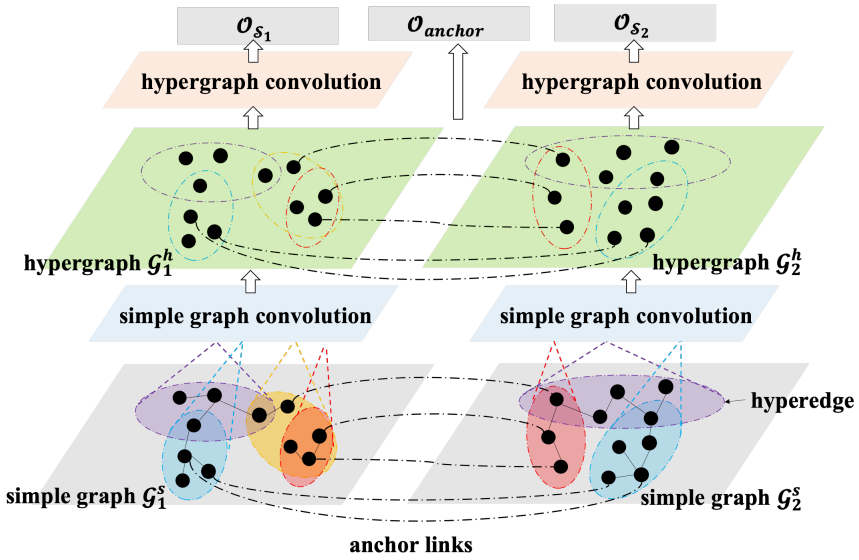
- Current Challenges:
  - Privacy issue and correctness.
    - Traditional methods by using profiles (self-generated), demographic info (name, pics, location, gender, etc.)
  - Data-insufficiency.
    - NE based methods need abundant network information.

Multi-level Graph Convolutional Networks (KDD20)

- Our method:
  - Hypergraph integration. (data enrichment)
    - Non-pairwise relations can be captured.
  - Multi-level GCNs.
    - Simple graph GCN.
      e.g., friendship, followers.
    - Hypergraph GCN.
      e.g., N-hop neighbours of a user – friends circle.
      Centrality-based hypergraphs represent different social levels.

Multi-level Graph Convolutional Networks (KDD20)

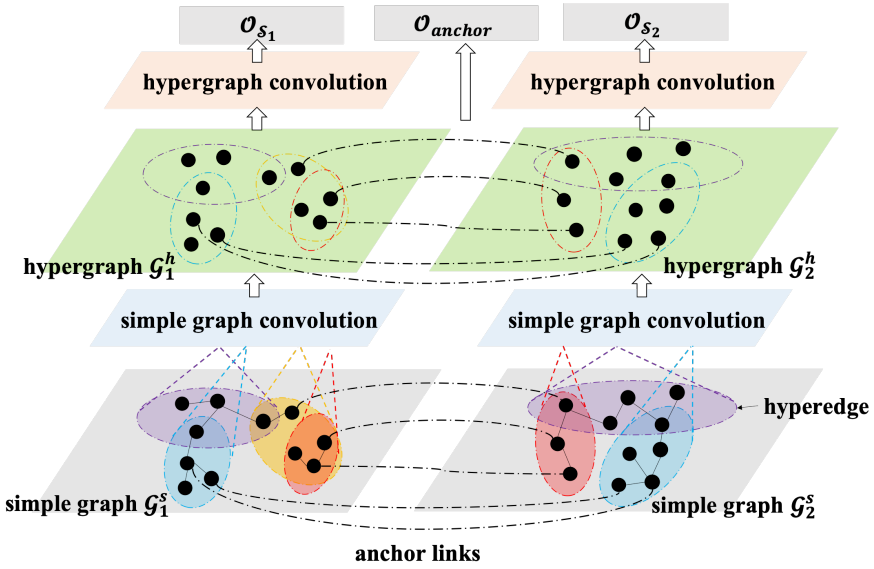
❑ Overview



❑ Rationale of exploiting and integrating hypergraphs.

- ❖ hypergraphs provide a more flexible network representation. - *It can contain additional and richer information compared to individual, single graph GCNs on local network topology.*
- ❖ GCNs on simple graph are only able to capture the local information. - i.e., *optimal number of GCN layers is always set to two in most cases because adding more layers cannot significantly improve the performance.*

Multi-level Graph Convolutional Networks (KDD20)



- Convolution on Simple Graphs:

$$\mathbf{X}_e^{k+1} = \sigma(\mathbf{A}_e \mathbf{X}_e^k \mathbf{W}^k)$$

$$\mathbf{S}_e(v_i, v_j) = \begin{cases} p(v,e), & \text{if } v_i = v_j, v_i \in e \\ 0, & \text{otherwise} \end{cases}$$

$$\mathbf{A}_e = \mathbf{S}_e \hat{\mathbf{A}} \mathbf{S}_e$$

$$\mathbf{X}_{simple}^{k+1} = f(\oplus_{e \in \mathcal{E}^h} \mathbf{X}_e^{k+1})$$

- Convolution on Hypergraphs:

$$\begin{aligned} \mathbf{X}^{k+1} &= \sigma\left(\left(\mathbf{I}_{|\mathcal{V}|} + \mathbf{D}_n^{-\frac{1}{2}} \mathbf{A}^h \mathbf{D}_n^{-\frac{1}{2}}\right) \mathbf{X}^k \mathbf{W}^k\right) \\ &= \sigma\left(\left(\mathbf{I}_{|\mathcal{V}|} + \mathbf{D}_n^{-\frac{1}{2}} \left(\mathbf{H}\mathbf{H}^\top - \mathbf{D}_n\right) \mathbf{D}_n^{-\frac{1}{2}}\right) \mathbf{X}^k \mathbf{W}^k\right) \\ &= \sigma\left(\mathbf{D}_n^{-\frac{1}{2}} \mathbf{H}\mathbf{H}^\top \mathbf{D}_n^{-\frac{1}{2}} \mathbf{X}^k \mathbf{W}^k\right) \end{aligned}$$

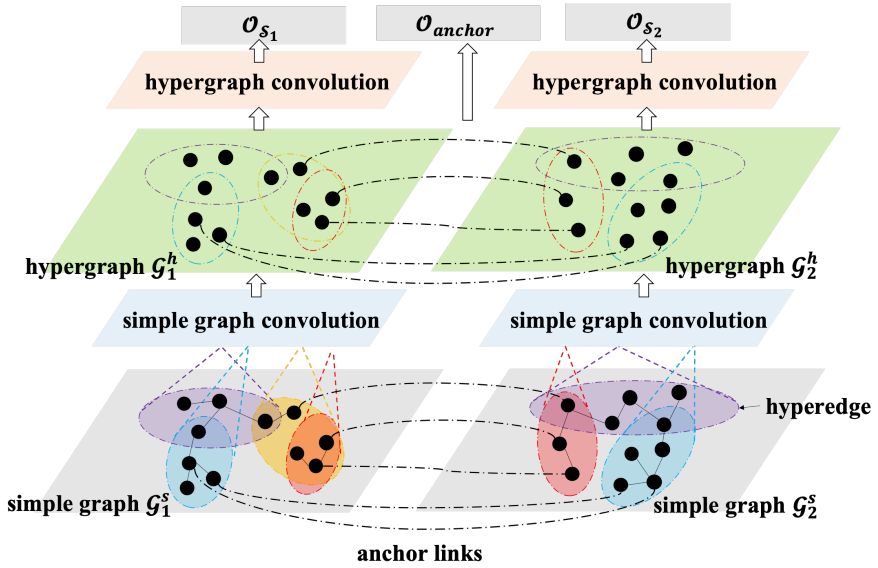
Multi-level Graph Convolutional Networks (KDD20)

- Learning Network Embedding:

$$O_{embedding} = \sum_{(v_i, v_j) \in \mathcal{E}} \log \eta(\mathbf{x}_i^{K\top} \mathbf{x}_j^K).$$
$$+ \sum_{k=1}^{M} E_{v_k \propto P(v)} \left[\log \left(1 - \sigma(\mathbf{x}_i^{K\top} \mathbf{x}_k^K)\right) \right]$$
$$+ \sum_{k=1}^{M} E_{v_k \propto P(v)} \left[\log \left(1 - \sigma(\mathbf{x}_j^{K\top} \mathbf{x}_k^K)\right) \right]$$

- Anchor Link Prediction:

$$O_{anchor} = \sum_{(v,u) \in \mathcal{S}_{anchor}} \|\mathbf{X}_1^K[v,:] - \phi(\mathbf{X}_2^K[u,:] | \Theta, \mathbf{b})\|^2$$



Multi-level Graph Convolutional Networks (KDD20)

- Handling Large-scale Networks.
  - Network Partition.
  - Reconcile Latent Embedding Spaces.

Algorithm 1: Graph Partitioning

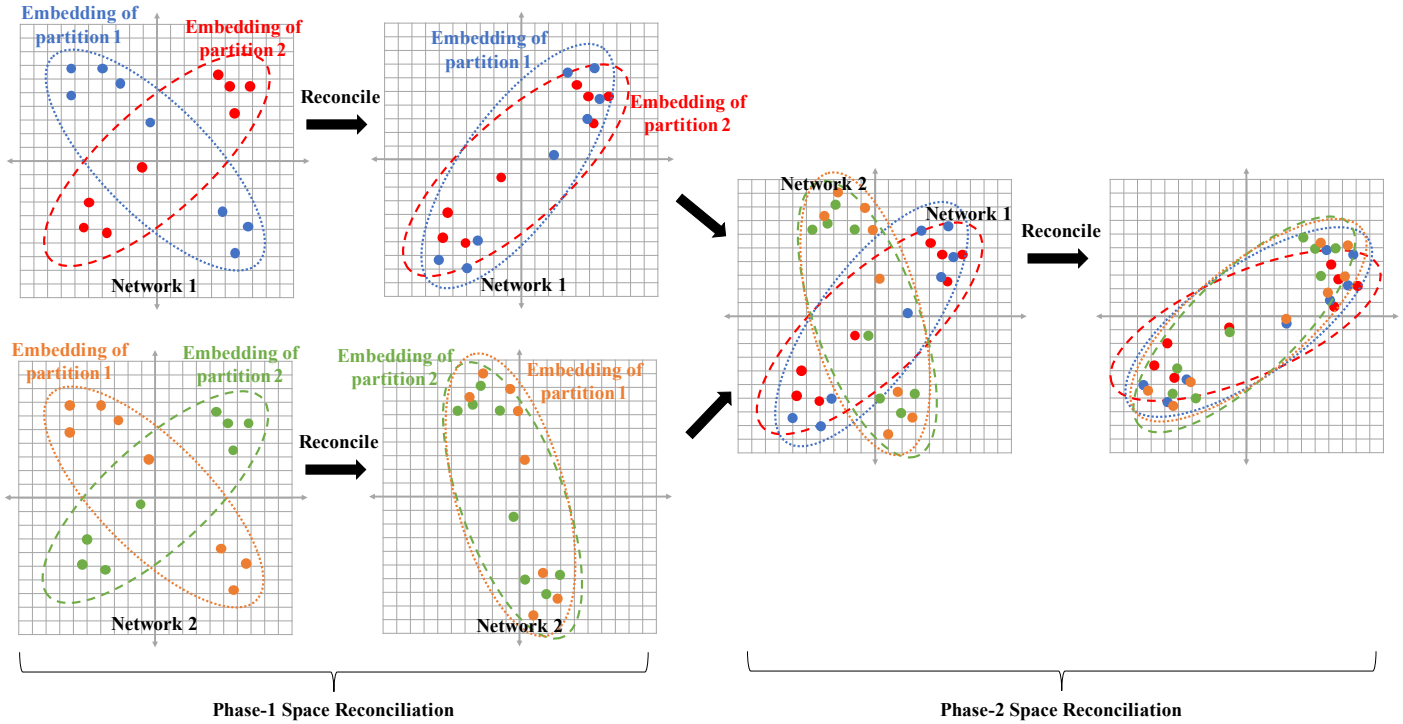
Input: $\mathcal{G}(\mathcal{V}, \mathcal{E})$, N_{max}, N_{min}, iteration T.
Output: partitions $P = \{\mathcal{G}_1(\mathcal{V}_1, \mathcal{E}_1), \cdots, \mathcal{G}_n(\mathcal{V}_n, \mathcal{E}_n)\}$.

```
P = Louvain(G) //Generating partitions P from G
  according to Louvain algorithm[4].
for iter from 1 to T do
    for partition G' ∈ P do
        if |V'| < N_min then
            add nodes of V' into other partitions, delete G'.
        else if N_min < |V'| <= N_max then
            continue
        else
            P_t = Louvain(G') //Generating partitions P_t
              from G' according to Louvain algorithm [4].
            P = P ∪ P_t
        end
    end
end
return P
```

Maximises the modularity using the Louvain algorithm.

Multi-level Graph Convolutional Networks (KDD20)

- Handling Large-scale Networks.
  - Network Partition.
  - Reconcile Latent Embedding Spaces.



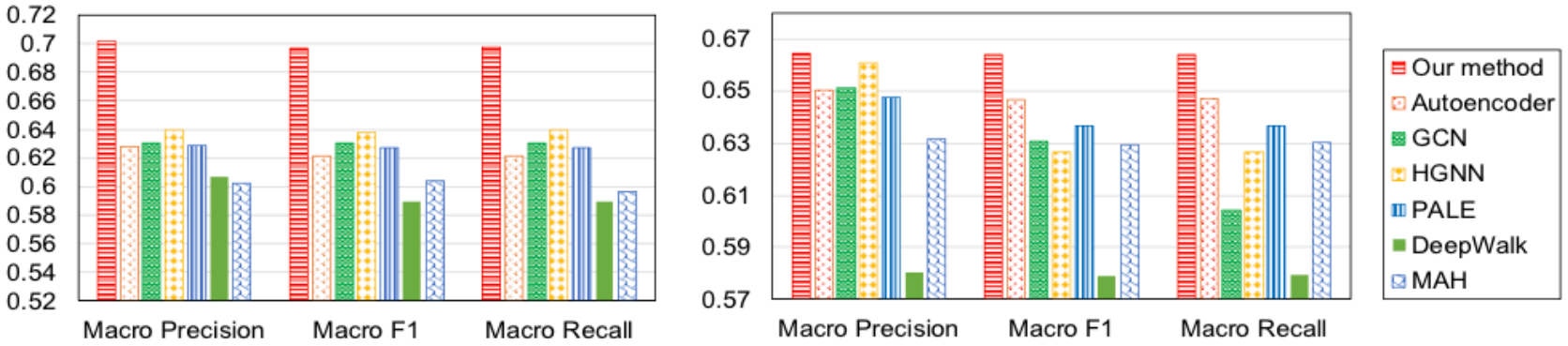
$$O_{partition} = \sum_{p=2}^{P} \sum_{v_i \in \mathcal{V}_{shared}} \log \sigma\left((f_p(\mathbf{x}_i^{(p)}))^\top \mathbf{x}_i^{(1)}\right)$$

Multi-level Graph Convolutional Networks (KDD20)

- Experiments
  - Datasets
    - Facebook-Twitter dataset.
    - Douban-Weibo dataset.
  - Baselines
    - Autoencoder
    - MAH
    - Deepwalk
    - GCN
    - HGCN
  - Evaluation Metrics:
    - Macro Precession, Recall, and F1. (Treat this task as a binary classification problem)

Multi-level Graph Convolutional Networks (KDD20)

- Experiments
  - Performance on Anchor Link Prediction

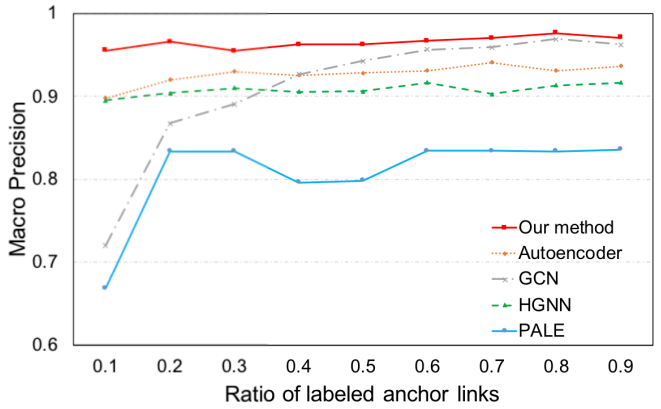


(a) Anchor Link Prediction on Facebook-Twitter

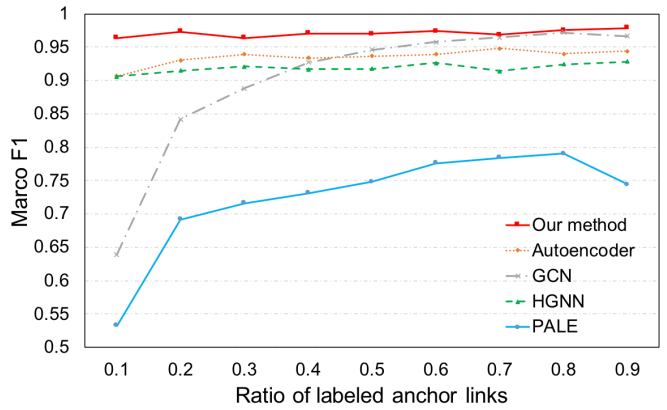
(b) Anchor Link Prediction on Douban-Weibo

Multi-level Graph Convolutional Networks (KDD20)

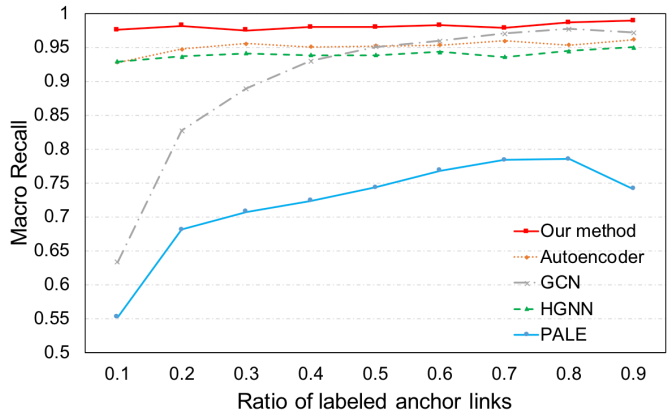
- Experiments
  - Analysis on Model Robustness
    - Results w.r.t. observed anchor link percentage.



(a) Macro Precision



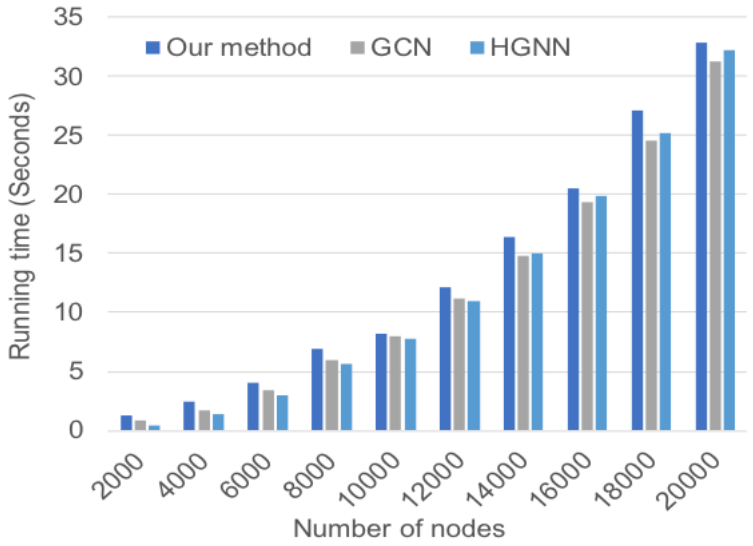
(b) Macro F1



(c) Macro Recall

Multi-level Graph Convolutional Networks (KDD20)

- Experiments
  - Analysis on Model Efficiency.
    - Forward propagation time w.r.t. network scales.



Multi-level Graph Convolutional Networks (KDD20)

- Preprint paper at:
  - https://arxiv.org/pdf/2006.01963.pdf

- Code available at:
  - https://github.com/sunxiangguo/MGCN

Thanks! Questions?