Network Embedding under Partial Monitoring for Evolving Networks

Yu Han[1], Jie Tang[1] and Qian Chen[2]

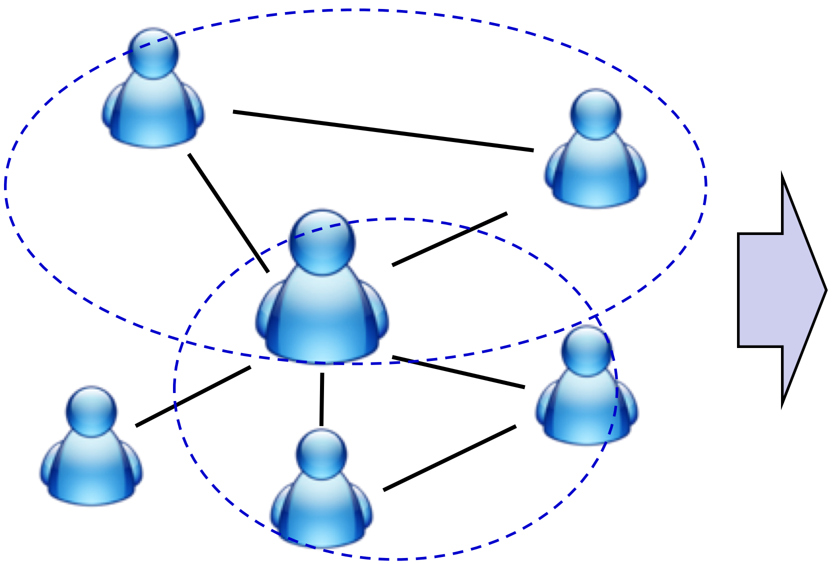
[1]Department of Computer Science and Technology
Tsinghua University

[2]Tencent Corporation

The slides can be downloaded at http://keg.cs.tsinghua.edu.cn/jietang

Motivation

Network/Graph Embedding
Representation Learning

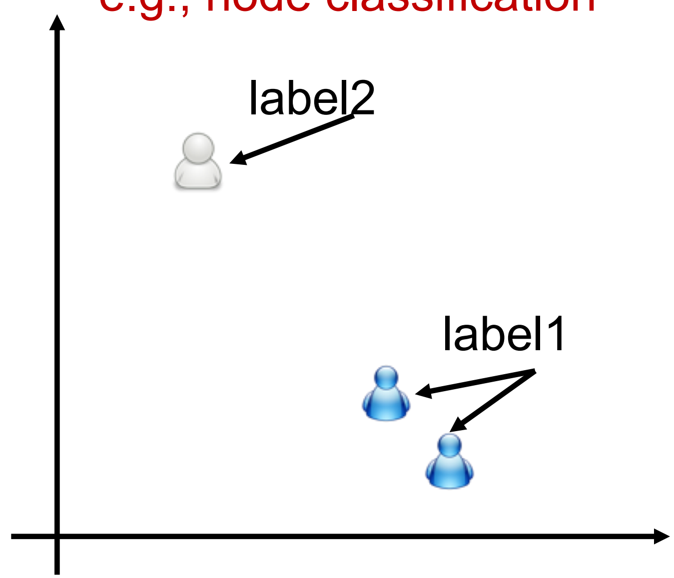


d-dimensional vector, $d \ll |V|$

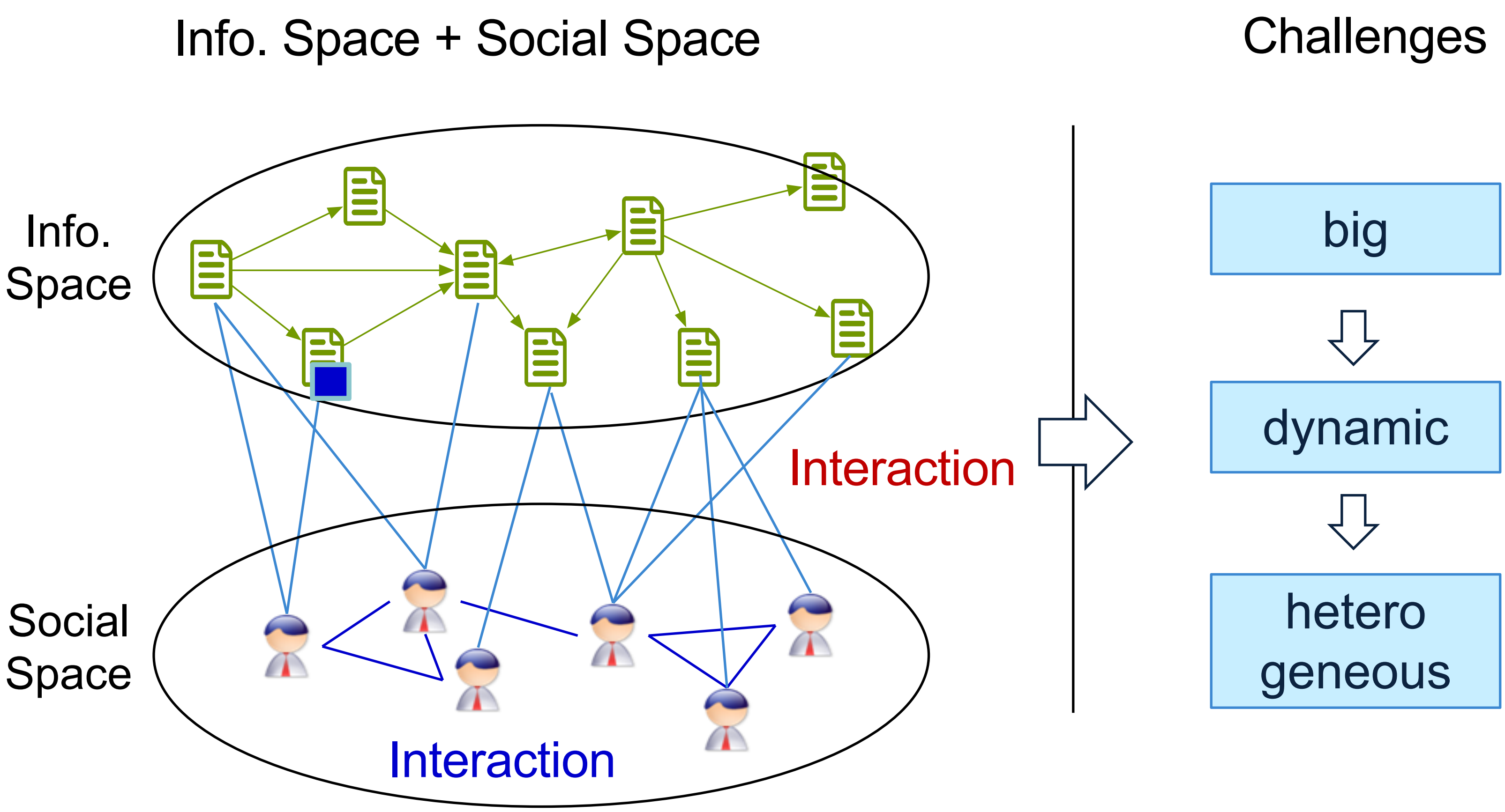
0.8	0.2	0.3	…	0.0	0.0

Users with the same label are located in the *d*-dimensional space closer than those with different labels

e.g., node classification



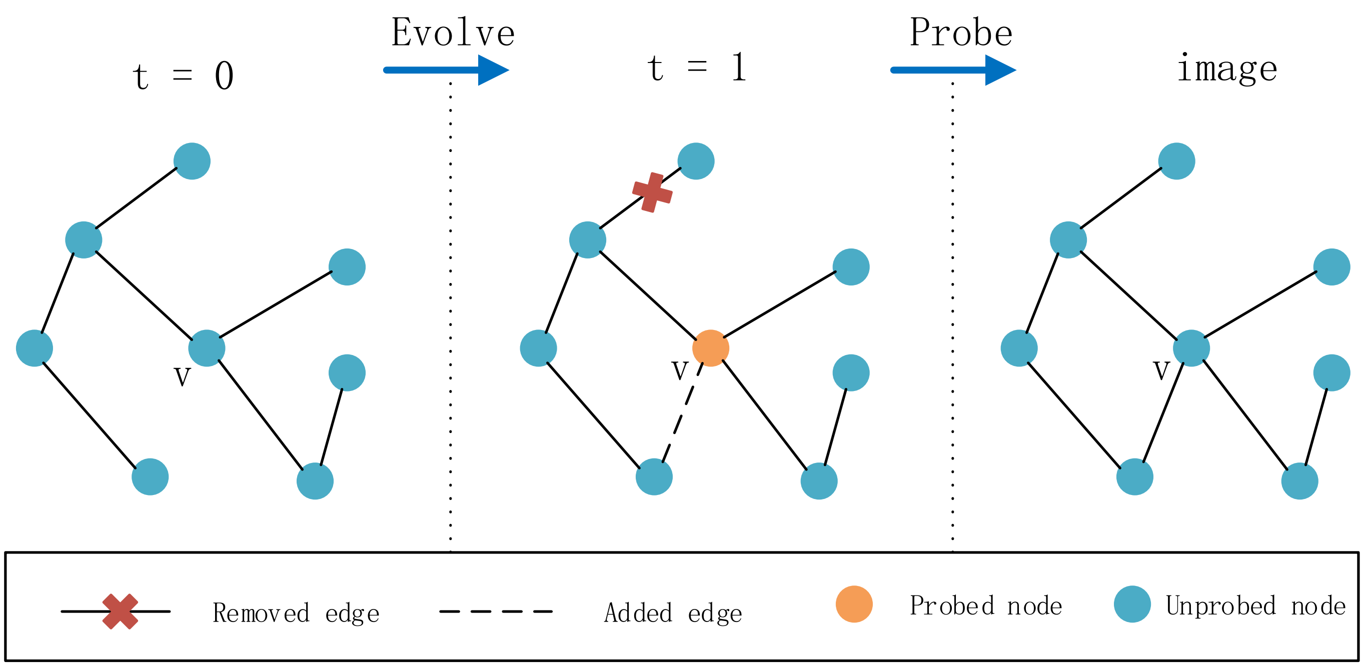
Challenges



1. J. Scott. (1991, 2000, 2012). Social network analysis: A handbook.
2. D. Easley and J. Kleinberg. Networks, crowds, and markets: Reasoning about a highly connected world. Cambridge University Press, 2010.

Problem: partial monitoring

What is network embedding under partial monitoring?



We can only probe part of the nodes to perceive the change of the network!

Revisit NE: distributional hypothesis of harris

- Words in similar contexts have similar meanings (skip-gram in word embedding)

- Nodes in similar structural contexts are similar (Deepwalk, LINE in network embedding)

- Problem: Representation Learning
  - Input: a network $G = (\mathcal{V}, \mathcal{E})$
  - Output: node embeddings $\mathbf{V} \in \mathbb{R}^{|\mathcal{V}| \times d}, d \ll |\mathcal{V}|$

Network Embedding

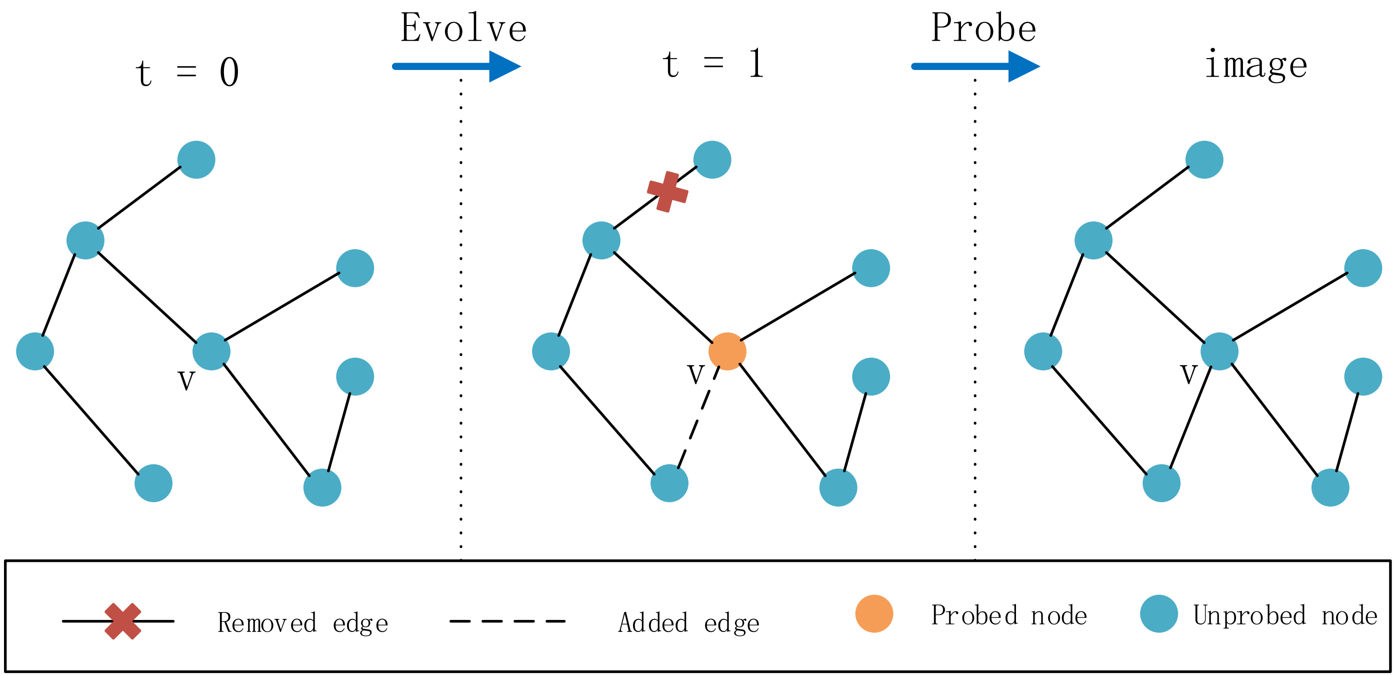
- We define the proximity matrix M, which is an $N \times N$ matrix, and $M_{i,j}$ represents the value of the corresponding proximity from node v_i to v_j.
- Given proximity matrix M, we need to minimize the objective function $\mathcal{O}_{NE} = \min_{X,Y} \|M - XY^T\|_F$, where X is the embedding table, Y is the embedding table when the nodes act as context.
- We can perform network embedding with SVD:
$$X = U\Sigma^{\frac{1}{2}}, \quad Y = \Sigma^{\frac{1}{2}}W^T.$$

1. Qiu et al. Network embedding as matrix factorization: unifying deepwalk, line, pte, and node2vec. *WSDM'18.* **The most cited paper in WSDM'18 as of May 2019**

Proximity Matrix

- Given graph $G = (V, A)$, any kinds of proximity can be exploited by network embedding models, such as:
  - Adjacency Proximity $M^{(AP)} = A.$
  - Jaccard's Coefficient Proximity $M_{i,j}^{(JC)} = \frac{|nbr(v_i) \cap nbr(v_j)|}{|nbr(v_i) \cup nbr(v_j)|}$
  - Katz Proximity
  - Adamic-Adar Proximity
  - SimRank Proximity
  - Preferential Attachment Proximity
  - ...

Problem



If we can only probe part of the nodes to perceive the change of the network, how to select the nodes to make the embeddings as accurate as possible?

Problem

- We formally define our problem

  In a network, given a time stamps sequence $< 0,1, \ldots, T >$, the starting time stamp (say t_0), the proximity and the dimension, we need to figure out a strategy π, to choose at most $K < N$ nodes to probe at each following time stamp, so that it minimizes the discrepancy between the approximate distributed representation, denoted as $\widehat{f_t}(V)$, and the potentially best distributed representation $f_t^*(V)$, as described by the following objective function.

$$\mathcal{O} = \min_{\pi} \sum_{t=1}^{T} Discrepancy(f_t^*(V), \hat{f}_t(V))$$

- The Key point: How to figure out the strategy to select the nodes.

Problem

- It is a sequential decision problem
- Obviously, the best strategy is to capture as much “change” as possible with limited “probing budget”.

Credit Probing Network Embedding

- Based on a kind of reinforcement learning problem, namely Multi-armed Bandit (MAB)
- Choose the “productive” nodes according to their historical “rewards”.
- At each time stamp t_j , we maintain a “credit” for each node v_i, which is the consideration for selecting the nodes.
- The “credit” should make a trade-off between exploitation and exploration.

Credit Probing Network Embedding

- The "credit" for each node v_i at time stamp t_j can be defined as:

Exploration

Exploitation

$$C_{v_i,t_j} = \hat{\mu}_{v_i,t_j} + \lambda\sqrt{\frac{\ln \boxed{t_j}}{\boxed{T_{v_i}}}}$$

Current time stamp

Empirical mean of v_i's historical rewards $||M||_F$, which refer to the the change it bring to the proximity matrix M from the last time stamp.

Hyperparameter to make a trade off between exploration and exploitation

Times that v_i has been probed

Credit Probing Network Embedding

- How to evaluate the difference between two embeddings X and X*?
- Obviously, it makes no sense to measure their concrete values with $||X-X^*||_F$.
- So we define two metrics: Magnitude Gap and Angle Gap from their geometric meanings.

Credit Probing Network Embedding

- Magnitude Gap

$$MG = \|S^* - \hat{S}\|_2$$

- Angle Gap

$$\begin{aligned} AG &= \sqrt{\|\sin\Theta\|_F^2 + \|\sin\Phi\|_F^2} \\ &= \sqrt{\frac{\|P_{U^*} - P_{\hat{U}}\|_F^2 + \|P_{V^*} - P_{\hat{V}}\|_F^2}{2}}, \end{aligned}$$

where P_{U^*} is the orthogonal projection operator of U^*, which can be achieved by $P_{U^*} = U^*U^{*\dagger} = U^*(U^{*T}U^*)^{-1}U^{*T}$, in which $(\cdot)^{-1}$ is the inverse of the corresponding matrix and $(\cdot)^\dagger$ is Moore-Penrose pseudoinverse. In the same way, we can get $P_{\hat{U}}$, P_{V^*} and $P_{\hat{V}}$ with $\hat{U}$, V^* and $\hat{V}$ respectively.

Credit Probing Network Embedding

- We prove the error bound for loss of magnitude gap and angle gap with matrix perturbation theory and combinatorial multi-armed bandit theory:

$$L_{MG} \leq \sum_{v_i \in V} \frac{4\lambda^2 N^2 \ln T}{\Delta_{v_i}^{min}} + (1 + \sum_{d=1}^{\infty} d^{1-2\lambda^2}) N \Delta^{max}$$

$$L_{AG} \leq \frac{\sqrt{\sum_{v_i \in V} \frac{8\lambda^2 N^2 \ln T}{\Delta_{v_i}^{min}} + 2(1 + \sum_{d=1}^{\infty} d^{1-2\lambda^2}) N \Delta^{max}}}{\delta}$$

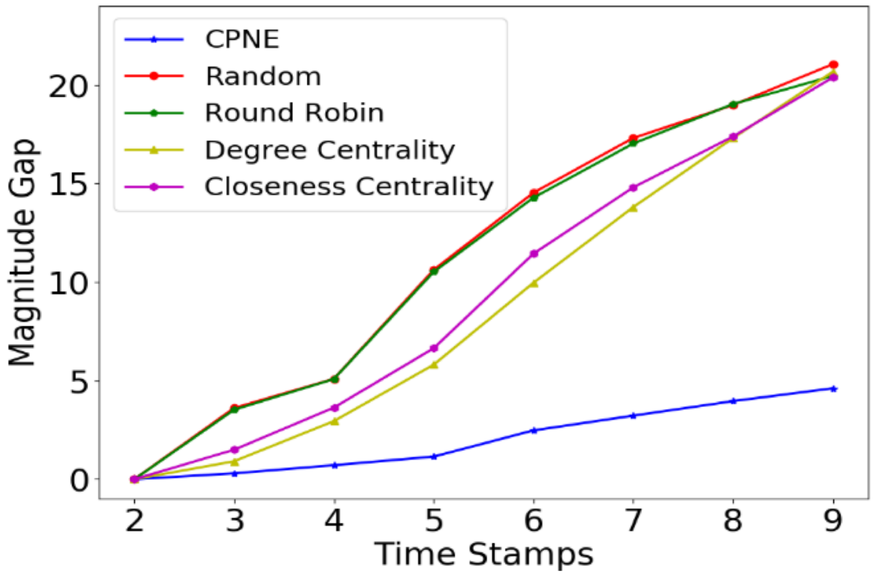
Experimental Setting

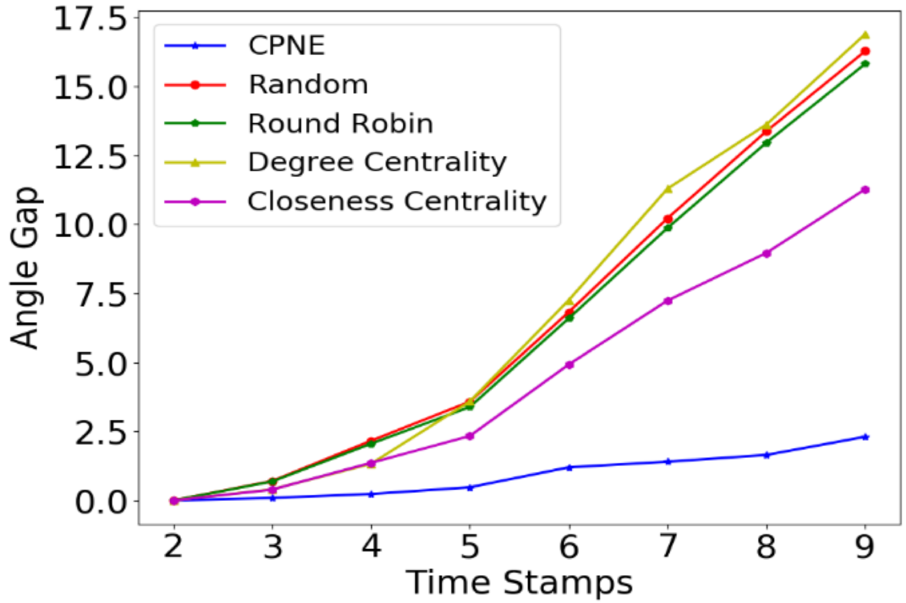
- Approaching the Potential Optimal Values
  - Datasets: AS
  - Baselines: Random, Round Robin, Degree Centrality, Closeness Centrality
  - Metrics: Magnitude Gap, Angle Gap
- Link Prediction
  - Datasets: WeChat
  - Baselines: BCGD[1] with the four settings
  - Metrics: AUC

1. Zhu et al. Scalable temporal latent space inference for link prediction in dynamic social networks. TKDE, 28(10):2765–2777, 2016

Experimental Results

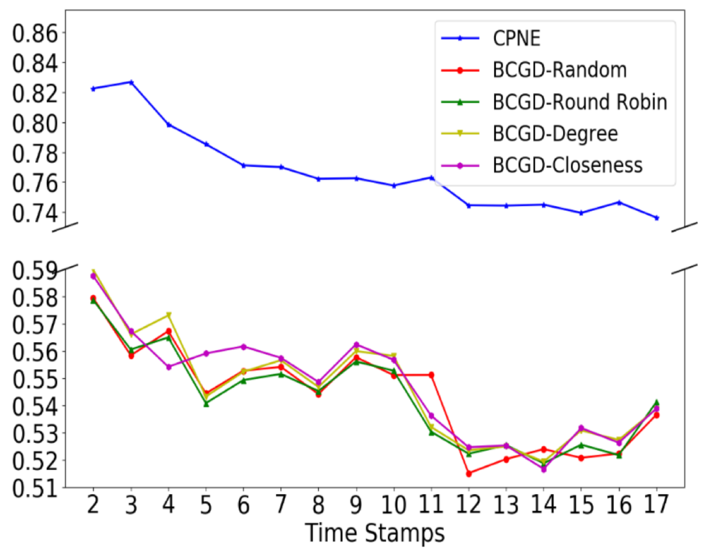
- Approaching the Potential Optimal Values



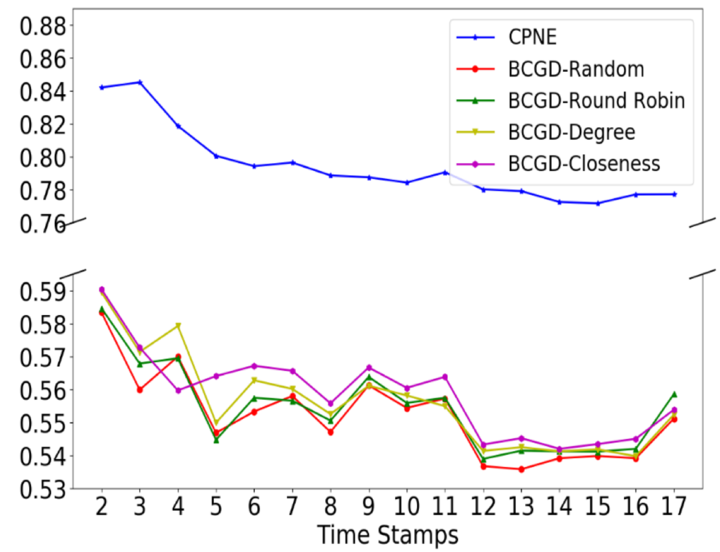


Experimental Results

- Link Prediction



K = 500



K = 1000

Further Consideration

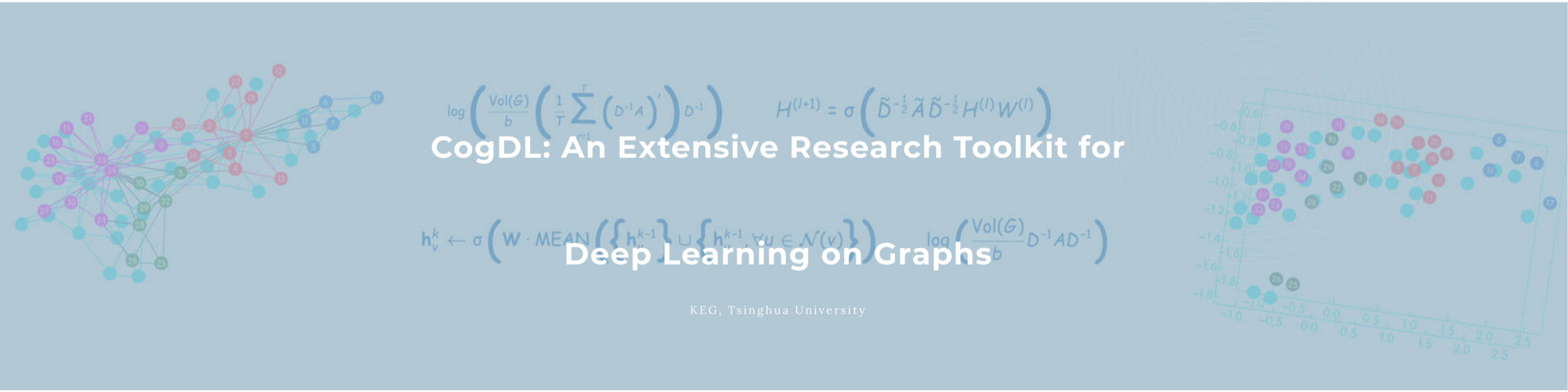
- Trying other reinforcement learning algorithms to solve such problems.
- Trying deep models to learning embedding values in such a setting.

CogDL

—A Toolkit for Deep Learning on Graphs

COGDL TOOLKIT

HOME DOCUMENTATION LEADERBOARD METHODS DATASETS FAQ

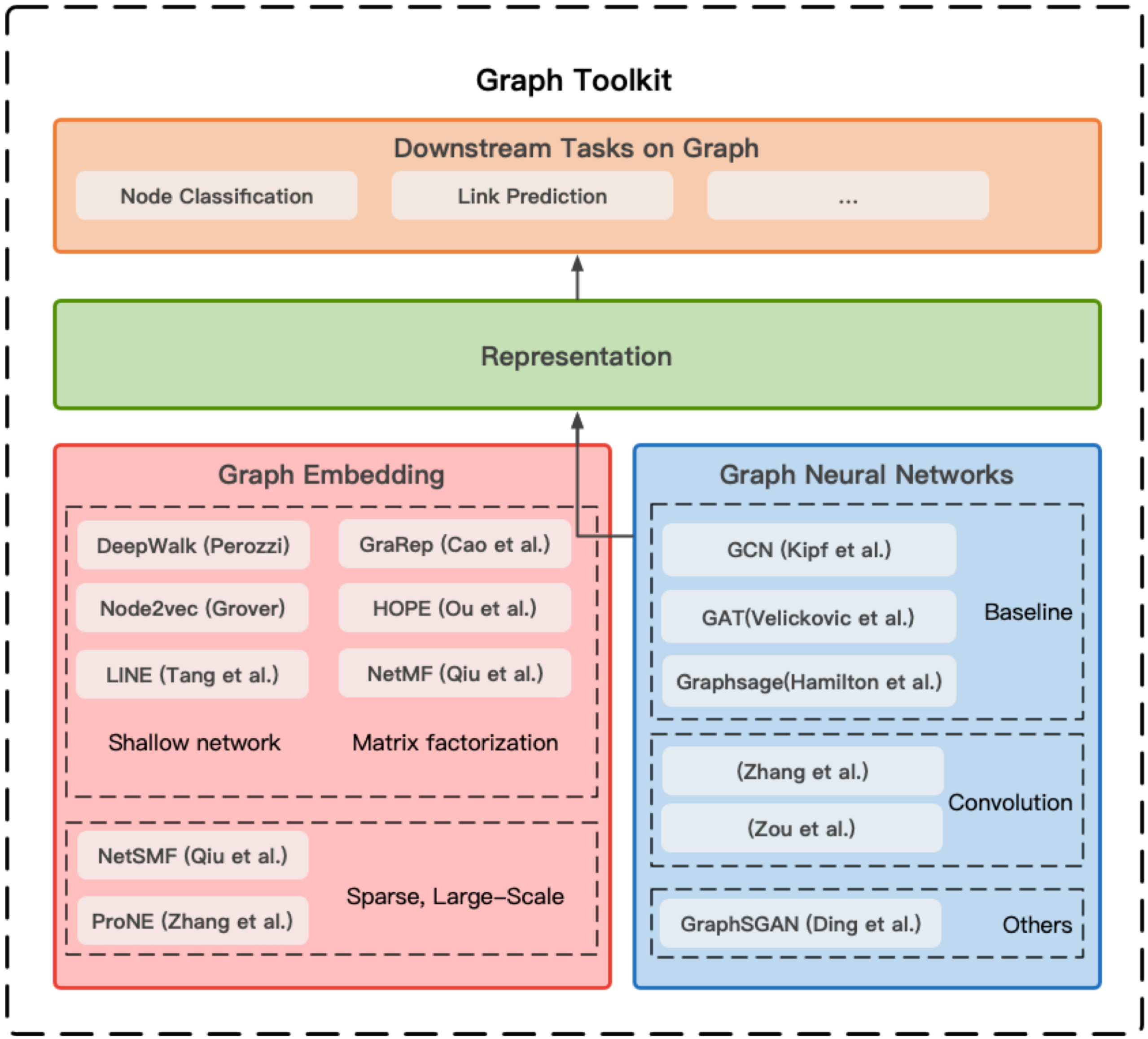


What is CogDL?

CogDL is a graph representation learning toolkit that allows researchers and developers to easily train and compare baseline or custom models for node classification, link prediction and other tasks on graphs. It provides implementations of many popular models, including: non-GNN Baselines like Deepwalk, LINE, NetMF, GNN Baselines like GCN, GAT, GraphSAGE.

** Code available at https://keg.cs.tsinghua.edu.cn/cogdl/

CogDL
—A Toolkit for Deep Learning on Graphs



Leaderboards: Link Prediction

Rank	Method	PPI			Wikipedia			Blogcatalog		
		ROC_AUC	PR_AUC	F1	ROC_AUC	PR_AUC	F1	ROC_AUC	PR_AUC	F1
1	ProNE (Zhang et al, IJCAI'19)	**95.14**	**94.21**	**89.44**	83.15	82.33	74.82	**89.63**	**86.74**	**81.92**
2	NetMF (Qiu et al, WSDM'18)	92.99	93.49	86.16	**85.86**	**86.85**	**77.73**	85.05	84.68	77.37
3	Node2vec (Grover et al, KDD'16)	92.13	93.27	84.93	84.18	86.53	75.09	84.41	85.06	75.98
4	DeepWalk (Perozzi et al, KDD'14)	92.05	93.19	84.85	83.94	86.36	74.91	84.45	85.10	75.99
5	LINE (Tang et al, WWW'15)	91.80	91.88	84.62	77.95	77.14	71.21	77.25	71.70	70.25
6	Hope (Ou et al, KDD'16)	92.77	91.12	86.44	69.16	62.94	63.54	80.99	79.77	74.07
7	NetSMF (Qiu et at, WWW'19)	75.16	75.67	68.64	47.66	75.66	68.64	68.14	64.03	61.62
8	GraRep (Cao et al, CIKM'15):	79.21	80.07	71.09	29.55	38.77	33.56	49.24	48.31	48.03

http://keg.cs.tsinghua.edu.cn/cogdl/link-prediction.html

Join us

- Feel free to join us with the three following ways:
  - ✓ add your data into the leaderboard
  - ✓ add your result into the leaderboard
  - ✓ add your algorithm into the toolkit

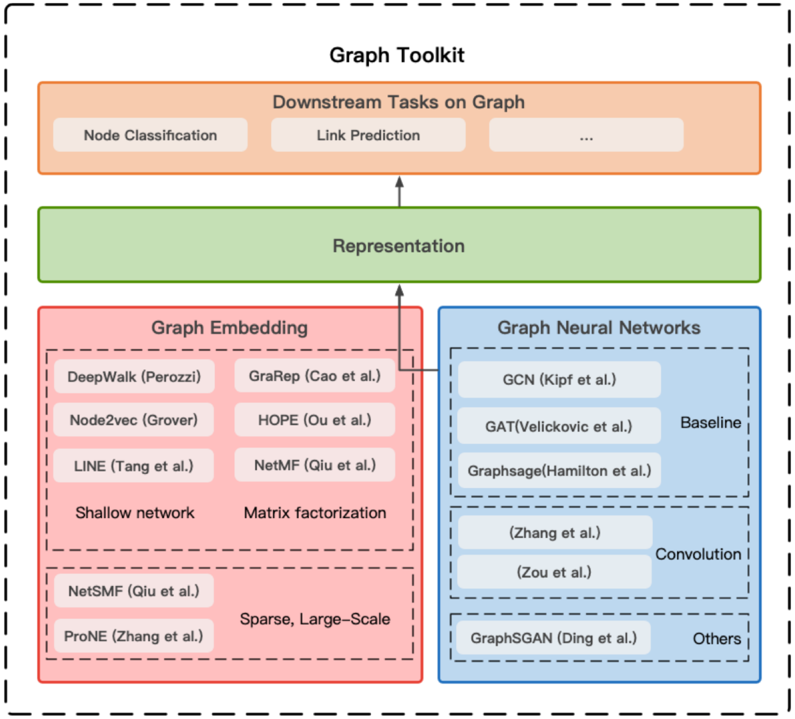
Rank	Method	PPI			Wikipedia			Blogcatalog		
		ROC_AUC	PR_AUC	F1	ROC_AUC	PR_AUC	F1	ROC_AUC	PR_AUC	F1
1	ProNE (Zhang et al, IJCAI'19)	**95.14**	**94.21**	**89.44**	83.15	82.33	74.82	**89.63**	**86.74**	**81.92**
2	NetMF (Qiu et al, WSDM'18)	92.99	93.49	86.16	**85.86**	**86.85**	**77.73**	85.05	84.68	77.37
3	Node2vec (Grover et al, KDD'16)	92.13	93.27	84.93	84.18	86.53	75.09	84.41	85.06	75.98
4	DeepWalk (Perozzi et al, KDD'14)	92.05	93.19	84.85	83.94	86.36	74.91	84.45	85.10	75.99
5	LINE (Tang et al, WWW'15)	91.80	91.88	84.62	77.95	77.14	71.21	77.25	71.70	70.25
6	Hope (Ou et al, KDD'16)	92.77	91.12	86.44	69.16	62.94	63.54	80.99	79.77	74.07
7	NetSMF (Qiu et at, WWW'19)	75.16	75.67	68.64	47.66	75.66	68.64	68.14	64.03	61.62
8	GraRep (Cao et al, CIKM'15):	79.21	80.07	71.09	29.55	38.77	33.56	49.24	48.31	48.03

Related Publications

- Yu Han, Jie Tang, and Qian Chen. Network Embedding under Partial Monitoring for Evolving Networks. IJCAI'19.
- Jie Zhang, Yuxiao Dong, Yan Wang, Jie Tang, and Ming Ding. ProNE: Fast and Scalable Network Representation Learning. IJCAI'19.
- Yukuo Cen, Xu Zou, Jianwei Zhang, Hongxia Yang, Jingren Zhou and Jie Tang. Representation Learning for Attributed Multiplex Heterogeneous Network. KDD'19.
- Fanjin Zhang, Xiao Liu, Jie Tang, Yuxiao Dong, Peiran Yao, Jie Zhang, Xiaotao Gu, Yan Wang, Bin Shao, Rui Li, and Kuansan Wang. OAG: Toward Linking Large-scale Heterogeneous Entity Graphs. KDD'19.
- Yifeng Zhao, Xiangwei Wang, Hongxia Yang, Le Song, and Jie Tang. Large Scale Evolving Graphs with Burst Detection. IJCAI'19.
- Ming Ding, Chang Zhou, Qibin Chen, Hongxia Yang, and Jie Tang. Cognitive Graph for Multi-Hop Reading Comprehension at Scale. ACL'19.
- Jiezhong Qiu, Yuxiao Dong, Hao Ma, Jian Li, Chi Wang, Kuansan Wang, and Jie Tang. NetSMF: Large-Scale Network Embedding as Sparse Matrix Factorization. WWW'19.
- Jiezhong Qiu, Jian Tang, Hao Ma, Yuxiao Dong, Kuansan Wang, and Jie Tang. DeepInf: Modeling Influence Locality in Large Social Networks. KDD'18.
- Jiezhong Qiu, Yuxiao Dong, Hao Ma, Jian Li, Kuansan Wang, and Jie Tang. Network Embedding as Matrix Factorization: Unifying DeepWalk, LINE, PTE, and node2vec. WSDM'18.
- Jie Tang, Jing Zhang, Limin Yao, Juanzi Li, Li Zhang, and Zhong Su. ArnetMiner: Extraction and Mining of Academic Social Networks. KDD'08.

For more, please check here http://keg.cs.tsinghua.edu.cn/jietang

Thank you !



Jie Tang, KEG, Tsinghua U http://keg.cs.tsinghua.edu.cn/jietang

Download all data & Codes https://keg.cs.tsinghua.edu.cn/cogdl/