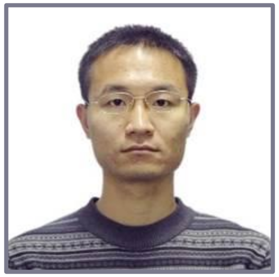
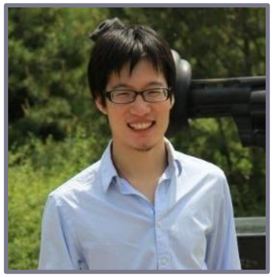
SFU
SIMON FRASER UNIVERSITY
ENGAGING THE WORLD

manLab
MEDIA AND NETWORK

Asymmetric Transitivity Preserving Graph Embedding

Mingdong Ou	Peng Cui	Jian Pei	**Ziwei Zhang**	Wenwu Zhu
Tsinghua U	Tsinghua U	Simon Fraser U	**Tsinghua U**	Tsinghua U


Graph Embedding

Graph Data

- Social Network
- Citation/ Collaboration
- Image/Video Tag/Caption
- Web Hyperlink
- Etc.

Graph Data Representation

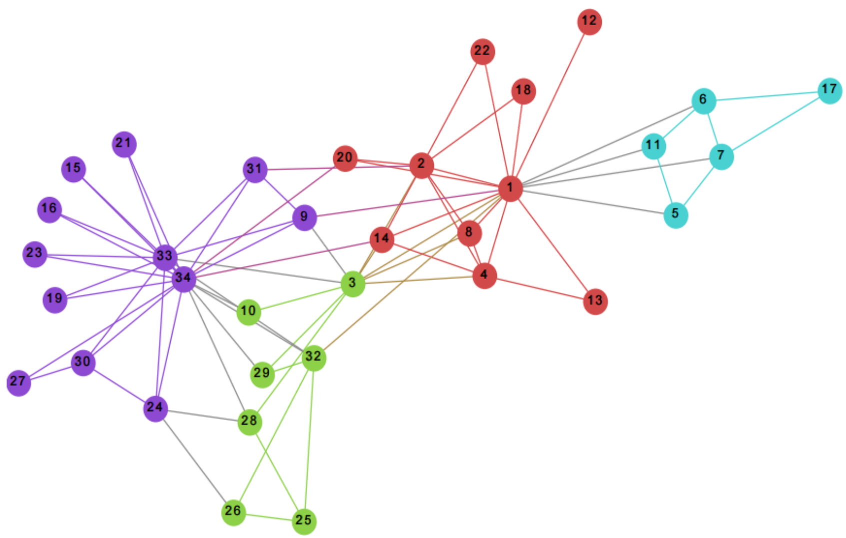
Graph Embedding

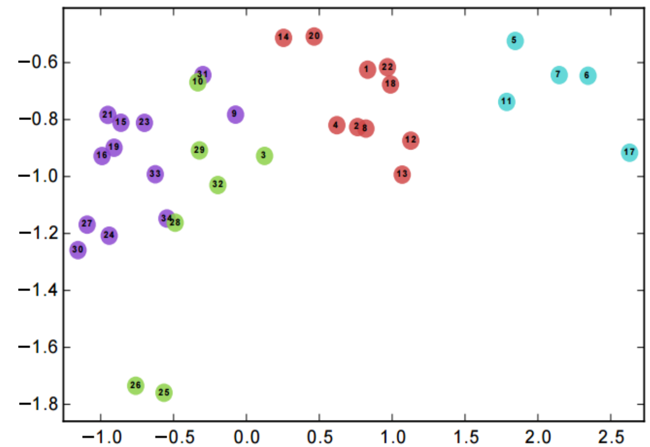
Application

- Similarity Measure
- Link Prediction
- Clustering/ Classification
- Visualization
- Etc.

Graph Embedding

- **Graph Embedding:**





Input graph/network → Low dimensional space

☐ **Advantages**:

- ☐ Fast computation of nodes similarity
- ☐ Utilization of vector-based machine learning techniques
- ☐ Facilitating parallel computing

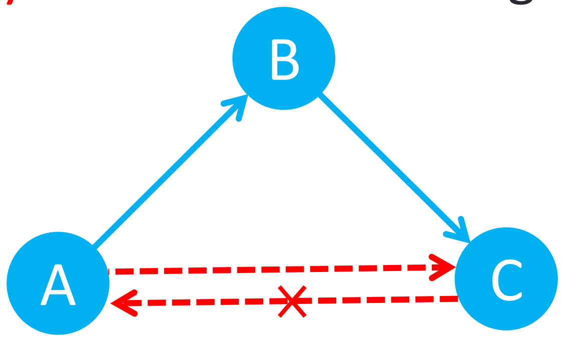
Existing graph embedding methods

- **Existing work:**
  - LINE(Tang J, et al. WWW 2015): explicitly preserves first-order and second-order proximity
  - DeepWalk(Perozzi B, et al. KDD 2014): random walk on graphs + SkipGram Model from NLP
  - GraRep(Cao S, et al. CIKM 2015)
  - SDNE(Daixin W, et al. KDD 2016)
- **Most methods focus on undirected graph**

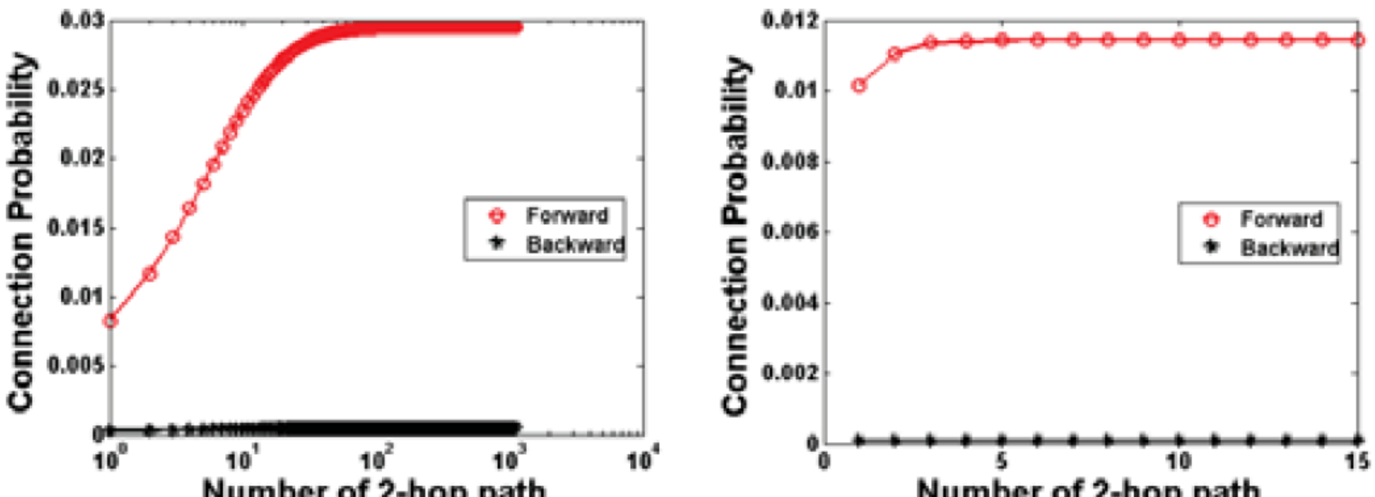
Directed Graph

Critical property in directed graph: **Asymmetric Transitivity**

- Transitivity is Asymmetric in directed graph:



- Key in graph inference.
- Data Validation: Tencent Weibo and Twitter



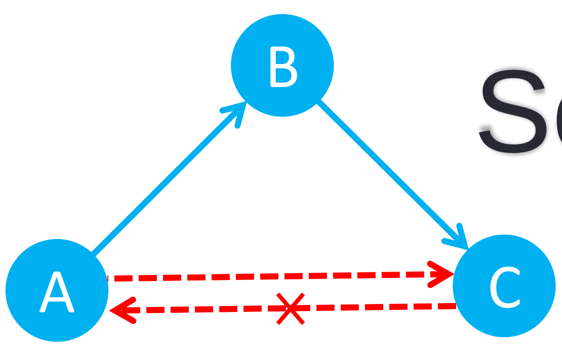
- **Asymmetric transitivity** is important!

Asymmetric Transitivity → Graph Embedding

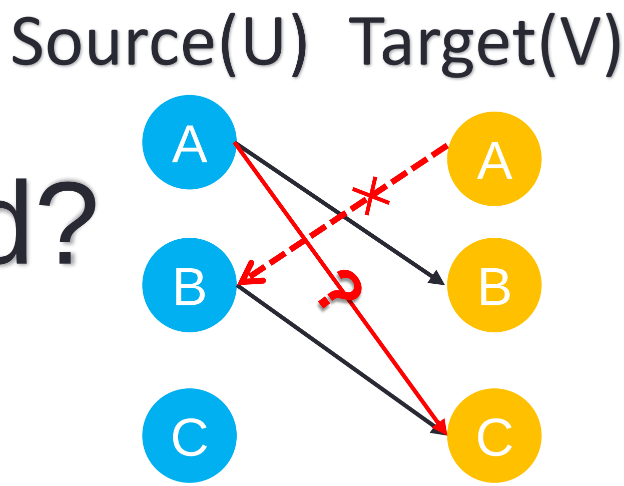
- Challenge: incorporate asymmetric transitivity in graph embedding
- Problem: metric space is **symmetric**

Asymmetric Transitivity → Graph Embedding

- **Directed graph embedding:** use two vectors to represent each node
  - LINE(Tang J, et al. WWW 2015): second-order proximity is directed
  - PPE(Song H H, et al. SIGCOMM 2009): using sub-block of the proximity matrix



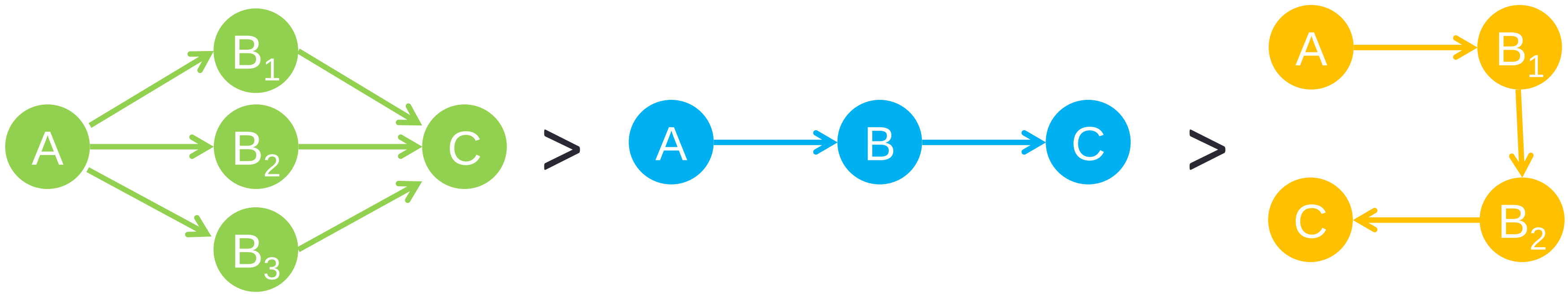
Solved?



- **Asymmetric: YES; Transitive: NO!**

Similarity metric with asymmetric transitivity

- Asymmetric transitivity:
  - Asymmetry: not symmetric in directed graph
  - Transitivity:
    - More directed paths, larger similarity
    - Shorter paths, larger similarity
  - Compare A -> C similarity:



High-order Proximity!

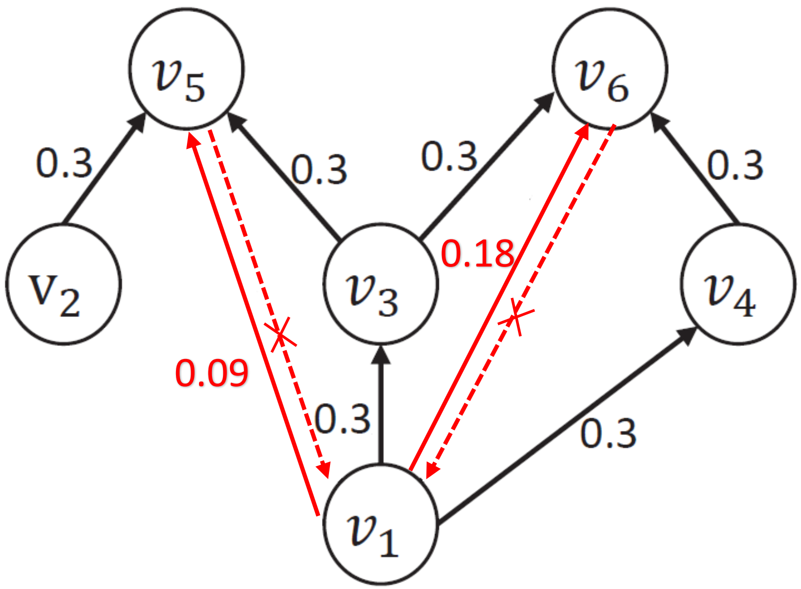
(E.g. Katz, Rooted PageRank)

High-Order Proximity

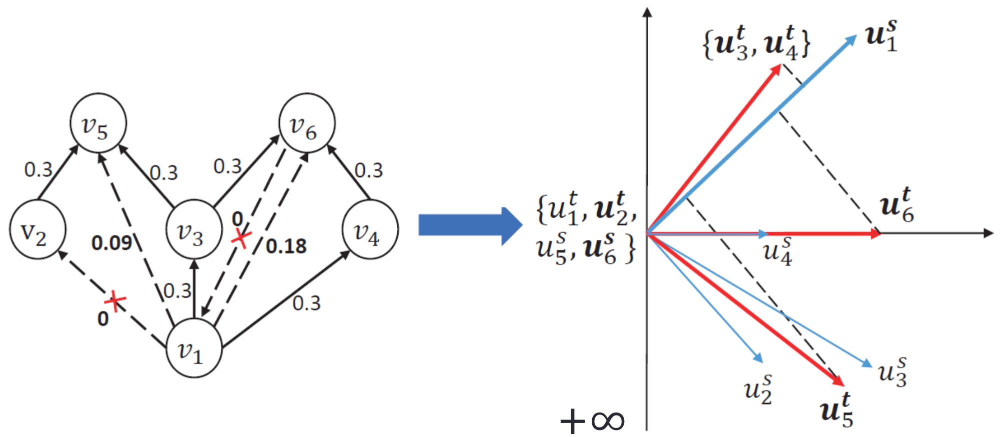
- Solution: directly model transitivity using high-order proximity
  - Example: Katz Index
  - A: adjacency matrix, β: decaying constant

$$\mathbf{S}^{Katz} = \sum_{l=1}^{+\infty} (\beta \cdot A)^l$$

  - Example: when $\beta = 1$



Preserve high-order proximity embedding



Naive Solution: **SVD**?

$$S^{Katz} = \sum_{l=1}^{+\infty} (\beta \cdot A)^l$$

S^{Katz} = U^s × U^t

□ **Time and space complexity: $O(N^3)$**, N: node number

High-Order Proximity: a general form

- Katz Index:

$$S^{Katz} = \sum_{l=1}^{+\infty} (\beta \cdot A)^l = (I - \beta \cdot A)^{-1} \cdot (\beta \cdot A)$$

- **General Form**

$$\boldsymbol{M_g^{-1} \cdot M_l}$$

where M_g, M_l are polynomial of adjacency matrix or its variants

General Formulation for High-Order Proximity measurements

Proximity Measurement	$\mathbf{M}_g$	$\mathbf{M}_l$
Katz	$\mathbf{I} - \beta \cdot \mathbf{A}$	$\beta \cdot \mathbf{A}$
Personalized Pagerank	$\mathbf{I} - \alpha\mathbf{P}$	$(1-\alpha) \cdot \mathbf{I}$
Common neighbors	$\mathbf{I}$	$\mathbf{A}^2$
Adamic-Adar	$\mathbf{I}$	$\mathbf{A} \cdot \mathbf{D} \cdot \mathbf{A}$

The Power of General Form

$$\min_{U_s, U_t} \|\boldsymbol{S} - U_s \cdot U_t^T\|_F^2$$

$$\boldsymbol{S} = M_g^{-1} \cdot M_l$$

Generalized SVD (Singular Value Decomposition) theorem

If we have the singular value decomposition of the general formulation

$$\mathbf{M}_g^{-1} \cdot \mathbf{M}_l = \mathbf{V}^s \Sigma \mathbf{V}^{t\top}$$

where $\mathbf{V}^t$ *and* $\mathbf{V}^s$ *are two orthogonal matrices,*

$$\Sigma = diag(\sigma_1, \sigma_2, \cdots, \sigma_N)$$

Then, there exists a nonsingular matrix $\mathbf{X}$ *and two diagonal matrices, i.e.* Σ^l *and* Σ^g*, satisfying that*

$$\mathbf{V}^{t\top} \mathbf{M}_l^\top \mathbf{X} = \Sigma^l \qquad \mathbf{V}^{s\top} \mathbf{M}_g^\top \mathbf{X} = \Sigma^g$$

, where

$$\Sigma^l = diag(\sigma_1^l, \sigma_2^l, \cdots, \sigma_N^l) \quad \sigma_1^l \geq \sigma_2^l \geq \cdots \geq \sigma_K^l \geq 0$$

$$\Sigma^g = diag(\sigma_1^g, \sigma_2^g, \cdots, \sigma_N^g) \quad 0 \leq \sigma_1^g \leq \sigma_2^g \leq \cdots \leq \sigma_K^g$$

$$\forall i \qquad {\sigma_i^l}^2 + {\sigma_i^g}^2 = 1$$

The Power of General Form

$$\min_{U_s, U_t} \|\boldsymbol{S} - U_s \cdot U_t^T\|_F^2$$

$$\boldsymbol{S} = M_g^{-1} \cdot M_l$$

- Generalized SVD: decompose $\boldsymbol{S}$ **without** actually calculating it
- JDGSVD: Time Complexity

$$O(K^2 L \cdot m)$$

Embedding Dimension (constant) | Iteration (constant) | Edge number

- **Linear complexity** w.r.t. the volume of data (i.e. edge number)

--> Scalable algorithm, suitable for large-scale data

Theoretical Guarantee

- Approximation Error Upper Bound:

THEOREM 2. *Given the proximity matrix,* $\mathbf{S}$*, of a directed graph, and the embedding vectors,* $\mathbf{U}^s$ *and* $\mathbf{U}^t$*, learned by* HOPPE. *Then the approximation error is*

$$\|\mathbf{S} - \mathbf{U}^s \cdot \mathbf{U}^t\|_F^2 = \sum_{i=K+1}^{N} \sigma_i^2$$

, and the relative approximation error is:

$$\frac{\|\mathbf{S} - \mathbf{U}^s \cdot \mathbf{U}^t\|_F^2}{\|\mathbf{S}\|_F^2} = \frac{\sum_{i=K+1}^{N} \sigma_i^2}{\sum_{i=1}^{N} \sigma_i^2} \tag{22}$$

where $\{\sigma_i\}$ *are the singular values of* $\mathbf{S}$ *in descend order.*

HOPE: HIGH-ORDER PROXIMITY PRESERVED EMBEDDING

HOPE: High-Order Proximity preserved Embedding

Algorithm framework:

Algorithm 1 High-order Proximity preserved Embedding

Require: adjacency matrix $\mathbf{A}$, embedding dimension K, parameters of high-order proximity measurementθ.

Ensure: embedding source vectors $\mathbf{U}^s$ and target vectors $\mathbf{U}^t$.

1: calculate $\mathbf{M}_g$ and $\mathbf{M}_l$.

2: perform JDGSVD with $\mathbf{M}_g$ and $\mathbf{M}_l$, and obtain the generalized singular values $\{\sigma_1^l, \cdots, \sigma_K^l\}$ and $\{\sigma_1^g, \cdots, \sigma_K^g\}$, and the corresponding singular vectors, $\{\mathbf{v}_1^s, \cdots, \mathbf{v}_K^s\}$ and $\{\mathbf{v}_1^t, \cdots, \mathbf{v}_K^t\}$.

3: calculate singular values $\{\sigma_1, \cdots, \sigma_K\}$ according to Equation (21).

4: calculate embedding matrices $\mathbf{U}^s$ and $\mathbf{U}^t$ according to Equation (19) and (20).

Experiment Setting: Datasets

Datasets:

- Synthetic (Syn): generate using Forest Fire Model
- Cora[1]: citation network of academic papers
- SN-Twitter[2]: Twitter Social Network
- SN-TWeibo[3]: Tencent Weibo Social Network

Statistics of datasets. $|V|$ denotes the number of vertexes and $|E|$ denotes the number of edges.

	Syn	Cora	SN-Twitter	SN-TWeibo
$\|\mathbf{V}\|$	10,000	23166	465,017	1,944,589
$\|\mathbf{E}\|$	144,555	91500	834,797	50,655,143

[1]http://konect.uni-koblenz.de/networks/subelj_cora
[2]http://konect.uni-koblenz.de/networks/munmun_twitter_social
[3]http://www.kddcup2012.org/c/kddcup2012-track1/data

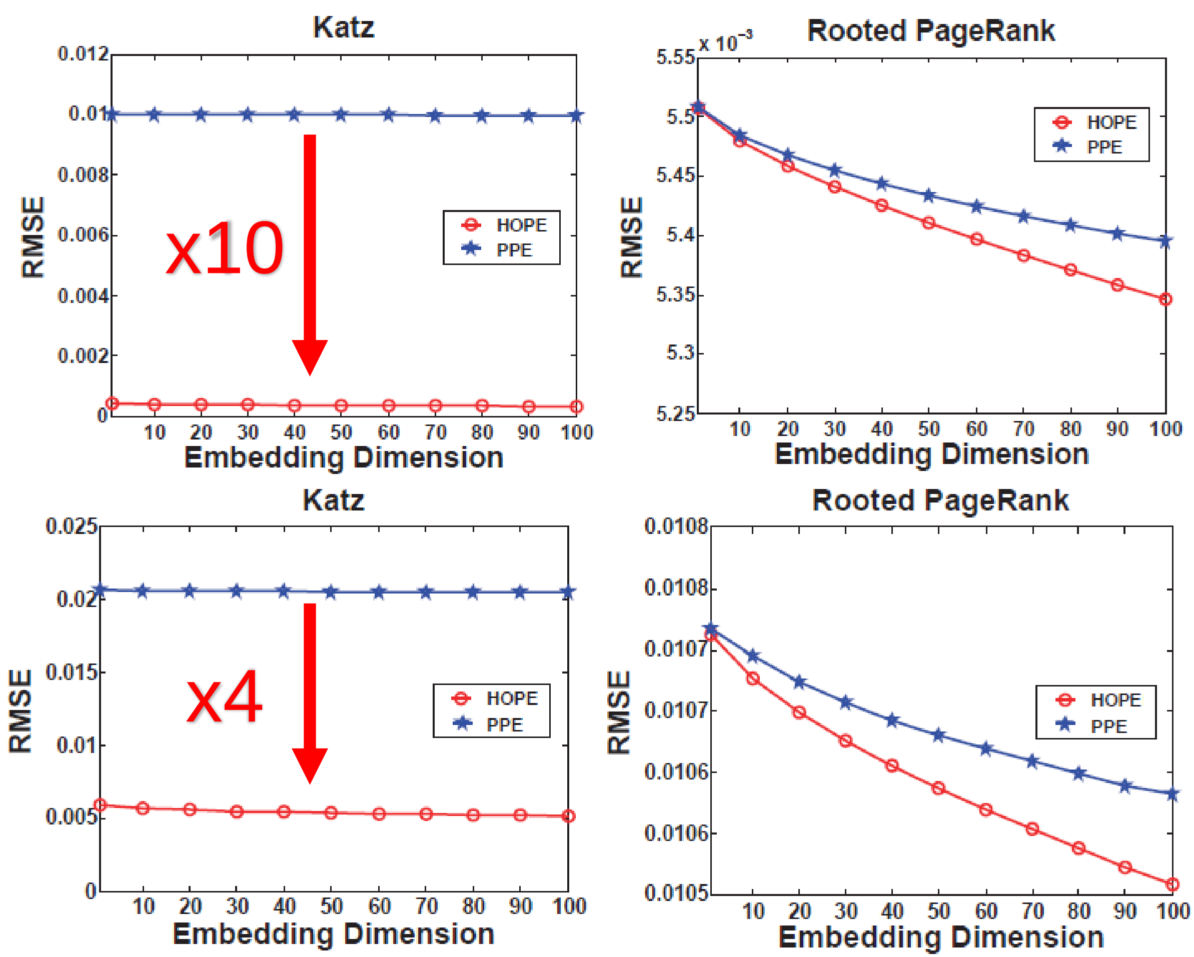
Experiment Setting: Task

- Approximation accuracy
  - **High-order proximity approximation**: how well can embedded vectors approximate high-order proximity
- Reconstruction
  - **Graph Reconstruction**: how well can embedded vectors reconstruct training sets
- Inference:
  - **Link Prediction**: how well can embedded vectors predict missing edges
  - **Vertex Recommendation:** how well can embedded vectors recommend vertices for each node

Experiment Setting: Baseline

- **Graph embedding**
  - **PPE**: approximate high-order proximity by selecting landmarks and using sub-block of the proximity matrix
  - **LINE**: preserves first-order and second-order proximity, called LINE1 and LINE2 respectively
  - **DeepWalk**: random walk on graphs + SkipGram Model
- **Task Specific:**
  - **Common Neighbors**: used for link prediction and vertex recommendation task
  - **Adamic-Adar**: used for link prediction and vertex recommendation task

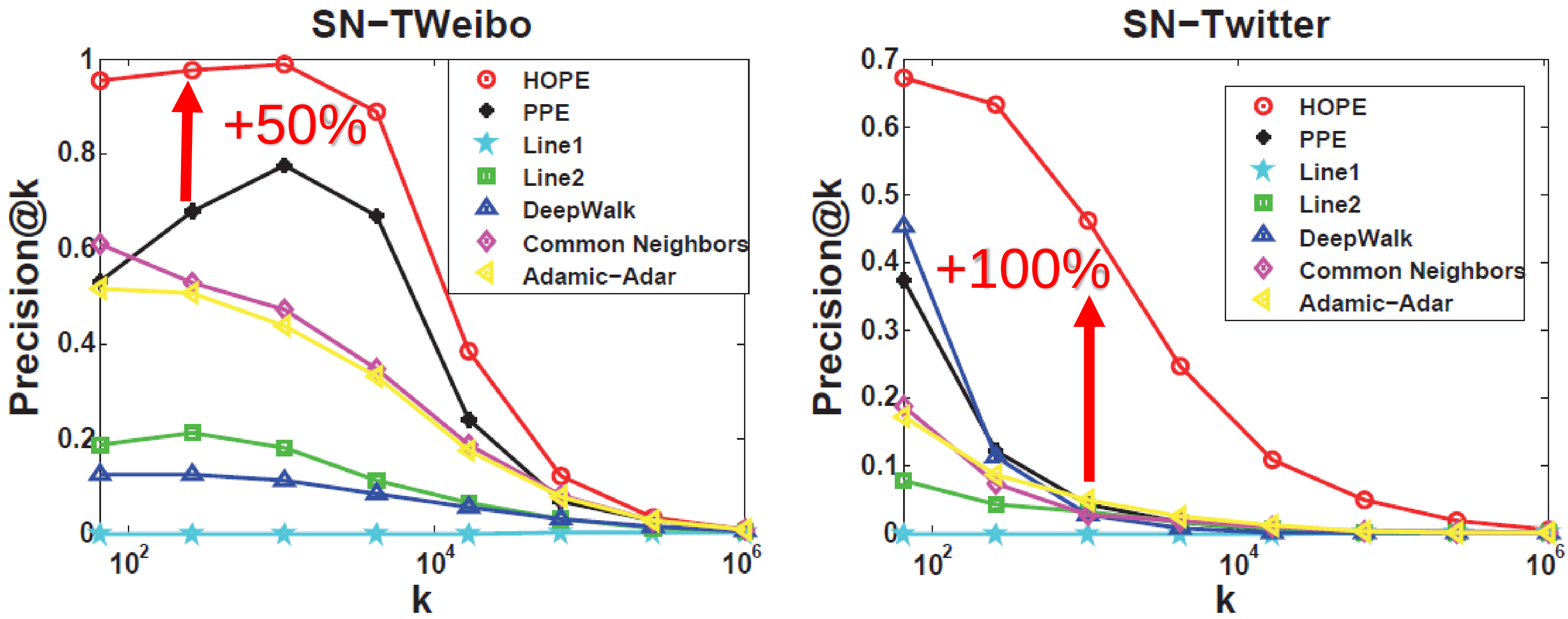
Experiment result: high-order Proximity Approximation



Conclusion: HOPE achieves much smaller RMSE error

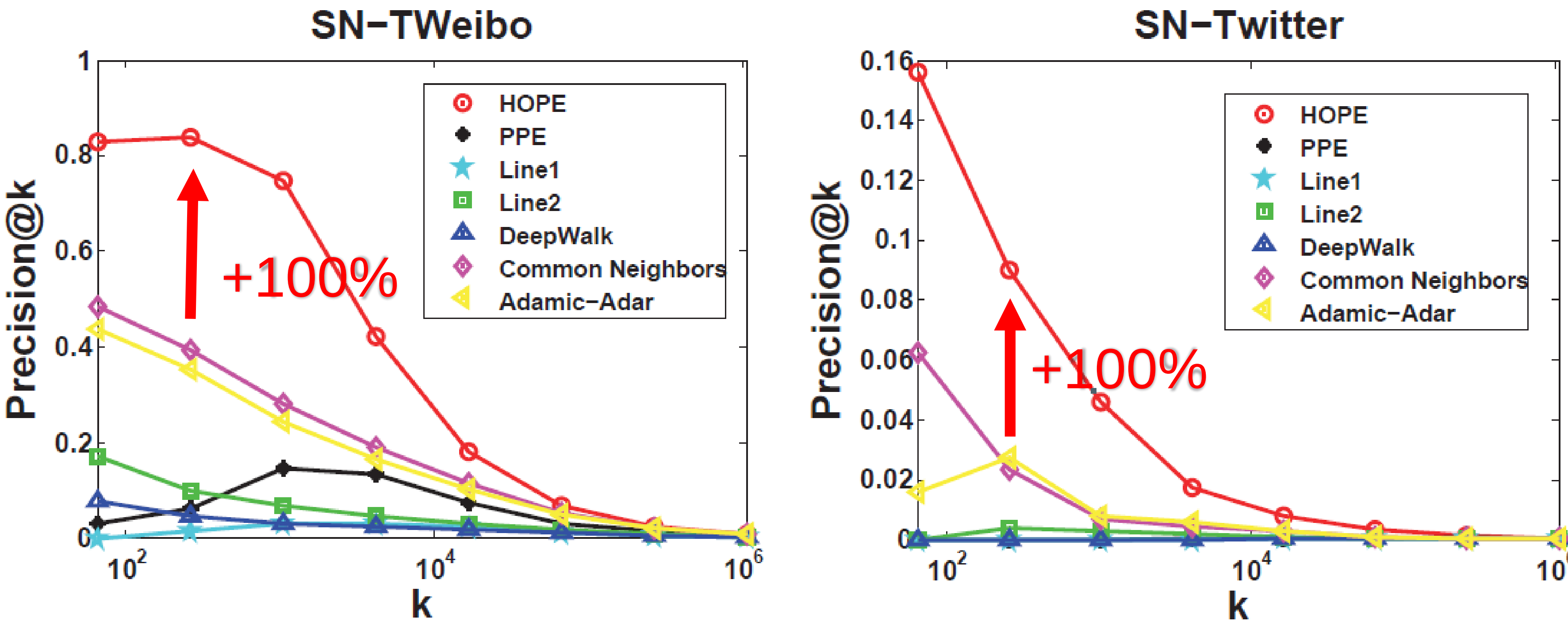
-> generalized SVD achieves a good approximation

Experiment result: Graph Reconstruction



Conclusion: HOPE successfully capture the information of training sets

Experiment result: Link Prediction



Conclusion: HOPE has good inference ability

-> based on asymmetric transitivity

Experiment result: Vertex Recommendation

+88% improvement

+81% improvement

Method	SN-TWebio			SN-Twitter		
	MAP@10	MAP@50	MAP@100	MAP@10	MAP@50	MAP@100
HOPE	**0.2295**	**0.1869**	**0.169**	**0.1000**	**0.0881**	**0.0766**
PPE	0.0928	0.0845	0.077	0.0061	0.0077	0.0081
LINE1	0	0	0.005	0.0209	0.0221	0.0221
LINE2	0.051	0.051	0.048	0.0044	0.0043	0.0035
DeepWalk	0.0635	0.0583	0.004	0.0006	0.0008	0.001
Common Neighbors	0.1217	0.1031	0.155	0.0394	0.0379	0.0369
Adamic-Adar	0.1173	0.0990	0.156	0.0455	0.0442	0.0423

Conclusion: HOPE significantly outperforms all state-of-the-art baselines on all these experiments

Conclusion

- Directed graph embedding:
  - **High-order Proximity → Asymmetric Transitivity**
- Derivation of a **general form** for high-order proximities, and solution with **generalized SVD**
  - Covering multiple commonly used high order proximity
  - **Time complexity linear** w.r.t. graph size
  - **Theoretically guaranteed** accuracy.
- Extensive experiments on several datasets
  - Outperforming all baselines in various applications.
  - **x4/x10** smaller approximation error for Katz
  - **+50% improvement** in reconstruction and inference

manLab
MEDIA AND NETWORK

Thanks!

Ziwei Zhang, Tsinghua University

zw-zhang16@mails.tsinghua.edu.com

https://cn.linkedin.com/in/zhangziwei