Microsoft

Graph Representation Learning with Hierarchical Structure and Domain Adaptation

Speaker： **Lun Du**

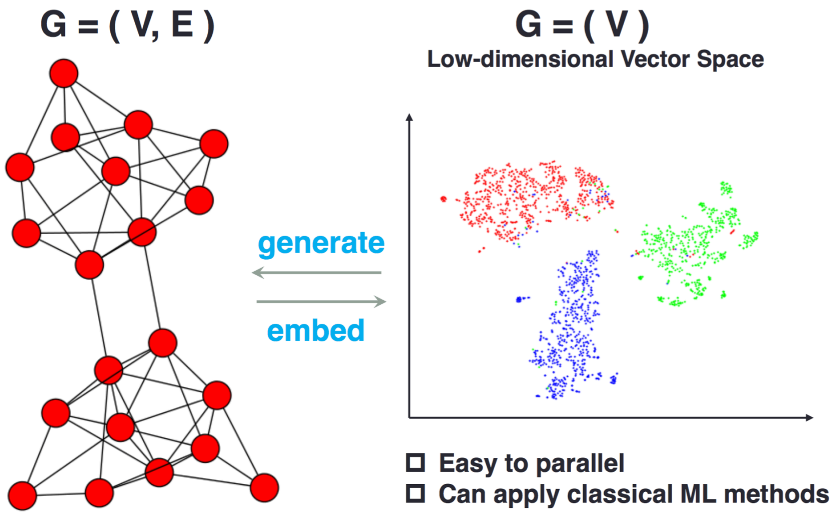
Affiliation: **Microsoft Research Asia**

Main Contents

- **Background**
- Graph embedding with hierarchical community structure
- Domain adaptive graph embedding
- Future works

Background

- **Graph Embedding** tries to map graph vertices into a low-dimensional vector space under the condition of preserving different types of graph properties.



- Node classification
- Link Prediction
- Network Visualization
- Community detection
- ……

Background

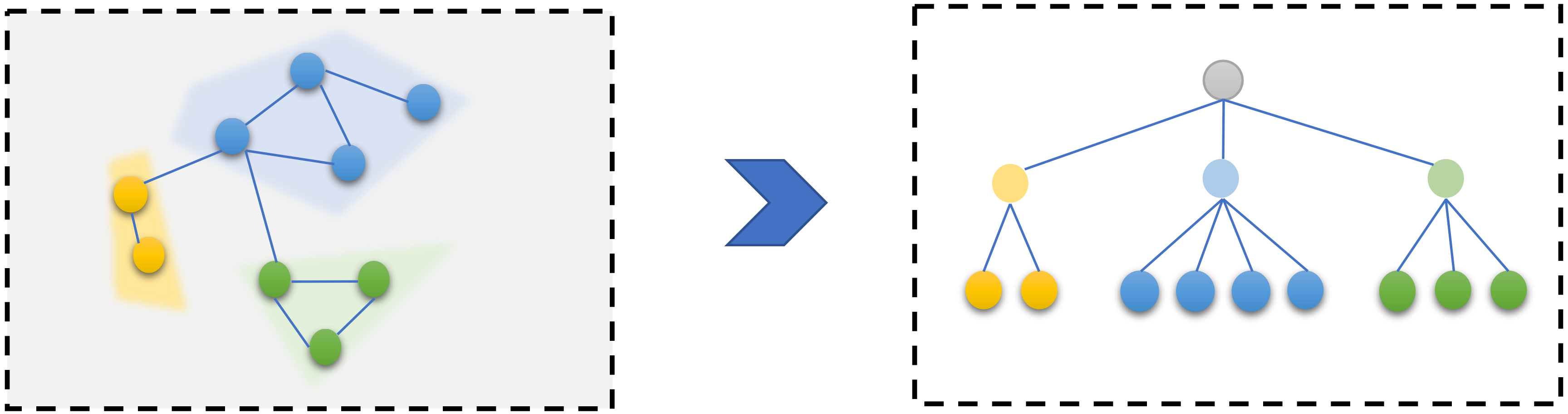
- Unsupervised vs. Supervised
  - DeepWalk, LINE, node2vec, etc.
  - GCN, GraphSAGE, etc.
- Euclidean vs. Non-Euclidean
  - Hyperbolic space (Tag2Vec, WWW’19)
- Vector vs. Distribution
  - Using variance to model uncertainty of semantic

Main Contents

- Background
- **Graph embedding with hierarchical community structure**
- Domain adaptive graph embedding
- Future works

Outline

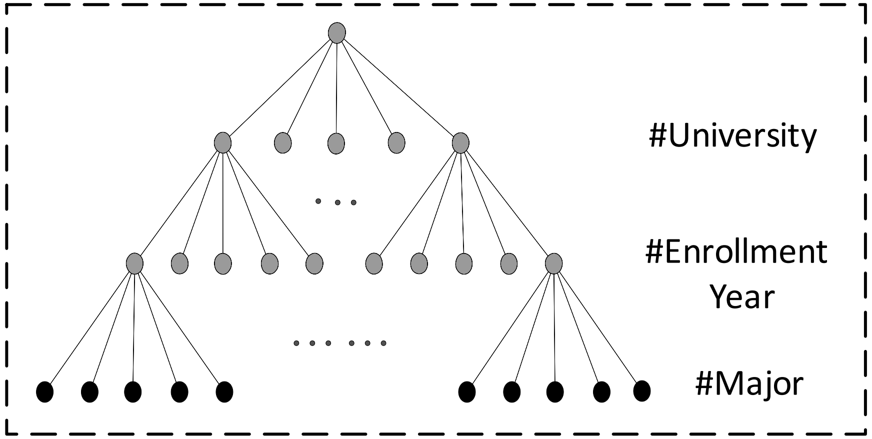
- Conceptually, complex networks have **hierarchical community** in real world.
  - E.g. social networks, air transportation networks, and metabolic networks, etc.



Outline

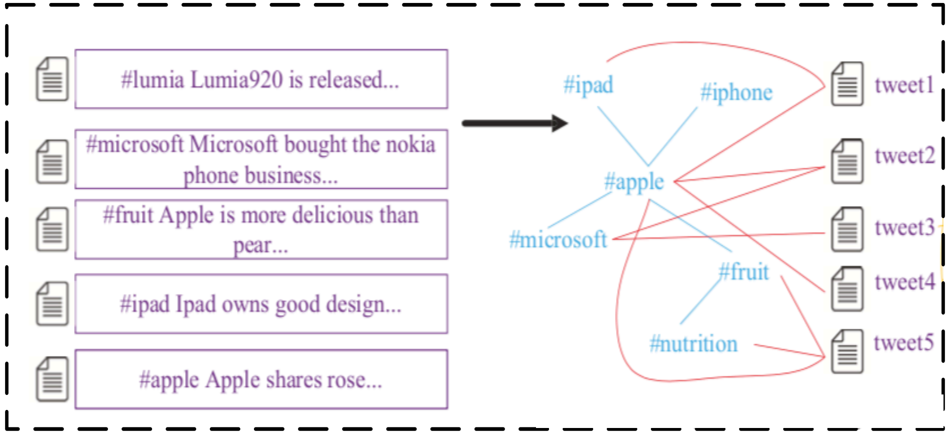
- Hierarchical Info can be observed to a certain extent in online networks.

Explicit hierarchy with attributes



Facebook Network

Implicit hierarchy with tags



Twitter Network

Outline

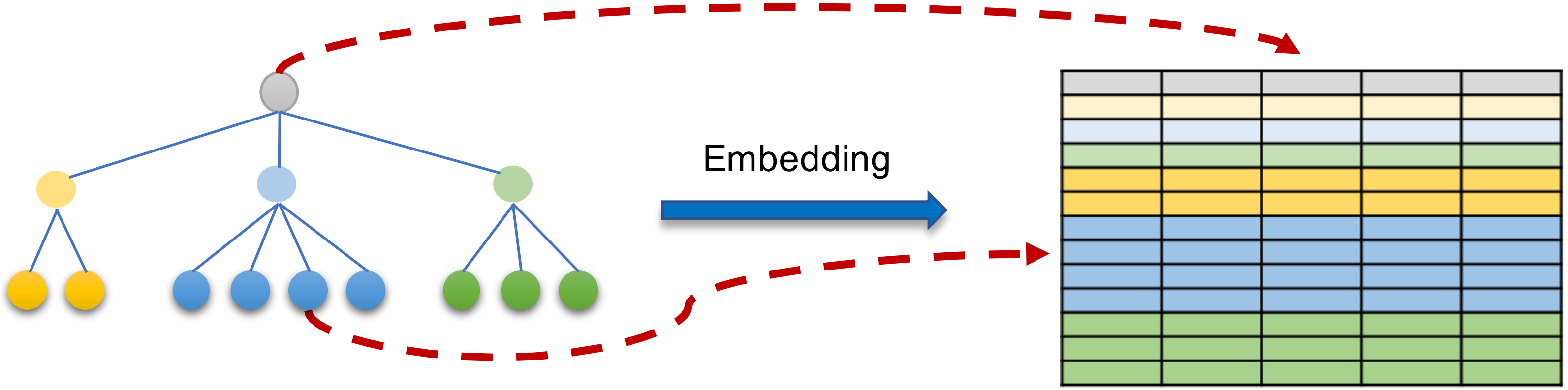
- Goal:
  - Encoding the rich hierarchical structural information
- Main Challenges:
  - How to represent nodes or tags?
  - How to learn the representations effectively and efficiently?

Galaxy Network Embedding: A Hierarchical Community Structure Preserving Approach

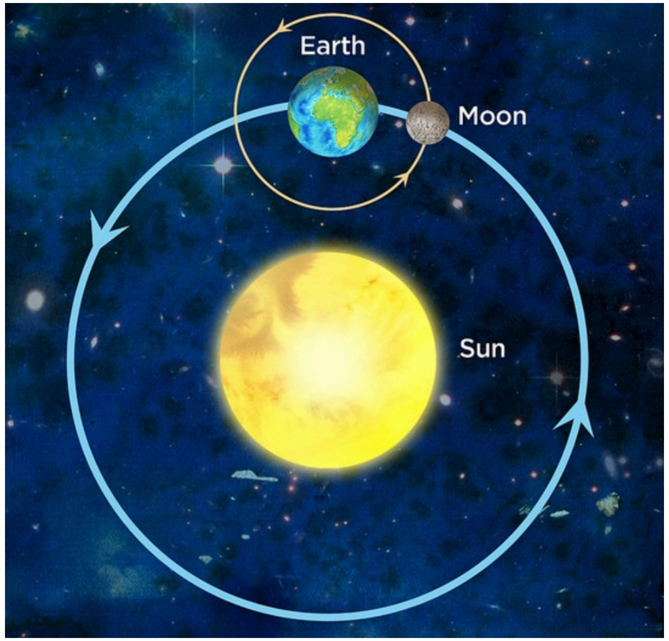
Lun Du, *Zhicong Lu, Yun Wang, Guojie Song*[ʄ] *, Yiming Wang, Wei Chen.* Galaxy Network Embedding: A Hierarchical Community Structure Preserving Approach. *In Proceedings of IJCAI, 2018.*

How to Represent?

- Inspired by galaxy structure
- Embedding nodes and communities simultaneously
  - Easy to analyze the network at different scales.



(The representations of nodes in tree)



How to Learn?

- Formulate the **hierarchical community preserving** network embedding
  - One is the local information, i.e. pairwise nodes similarity in the same community.
  - The other is the hierarchical structure property, i.e. horizontal relationship and vertical relationship.
- Implement and optimize efficiently the embedding method.

Hierarchical Preserving Network Embedding

- Pairwise Proximity Preservation

$$\min_{\Phi,\Phi'} O_k^{(l-1)} = - \sum_{c_i^l, c_j^l \in Ch(c_k^{l-1})} S_{i,j}^l \log P(\Phi'(c_j^l) \mid \Phi(c_i^l))$$

$$S_{i,j}^l = \frac{1}{|c_i^l||c_j^l|} \sum_{u \in c_i^l} \sum_{v \in c_j^l} \frac{A_u^T A_v}{\sqrt{||A_u||_1 ||A_v||_1}},$$

$$P(\Phi'(c_j^l) \mid \Phi(c_i^l)) = \frac{\exp(\Phi'(c_j^l) \cdot \Phi(c_i^l))}{\sum_{c_t^l \in Ch(c_k^{l-1})} \exp(\Phi'(c_t^l) \cdot \Phi(c_i^l))},$$

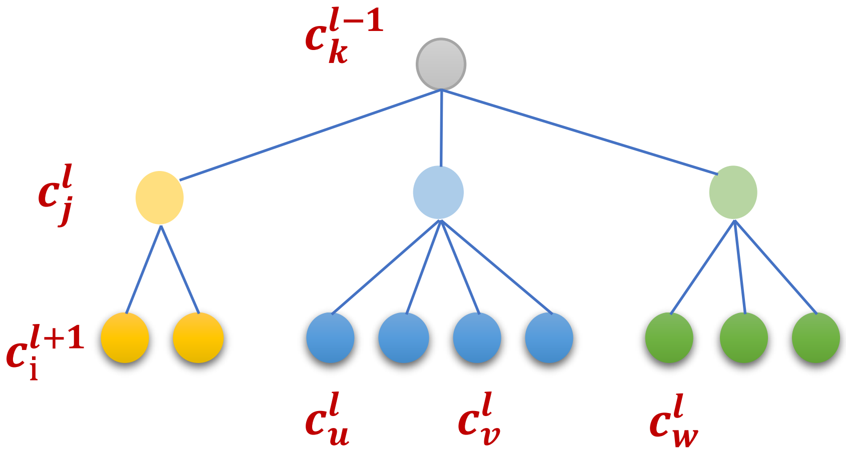
Hierarchical Preserving Network Embedding

- Hierarchical Structure Preservation
  - Horizontal relationship:

$$\|\Phi(c_u^l) - \Phi(c_v^l)\| < \|\Phi(c_u^l) - \Phi(c_w^l)\|, \qquad (3)$$

  - Vertical relationship:

$$\|\Phi(c_i^{l+1}) - \Phi(c_j^l))\| < \|\Phi(c_j^l) - \Phi(c_k^{l-1})\|. \qquad (4)$$



Galaxy Network Embedding

Objective

$$\min_{\Phi,\Phi'} O_k^{(l-1)} = - \sum_{c_i^l, c_j^l \in Ch(c_k^{l-1})} S_{i,j}^l \log P(\Phi'(c_j^l) \mid \Phi(c_i^l))$$
$$s.t. \quad \forall c_i^l \in Ch(c_k^{l-1}), \quad \|\Phi(c_i^l) - \Phi(c_k^{l-1})\|_2 = r_k^{l-1}. \tag{5}$$

where,

$$r_i^l = \eta \cdot d_k^{l-1}, \quad \eta < \frac{1}{6}$$
$$d_k^{l-1} = \min_{c_i^l, c_j^l \in Ch(c_k^{l-1}), i \neq j} Dist\left(\Phi(c_i^l), \Phi(c_j^l)\right), \tag{6}$$
$$Dist(x, y) = \|x - y\|,$$

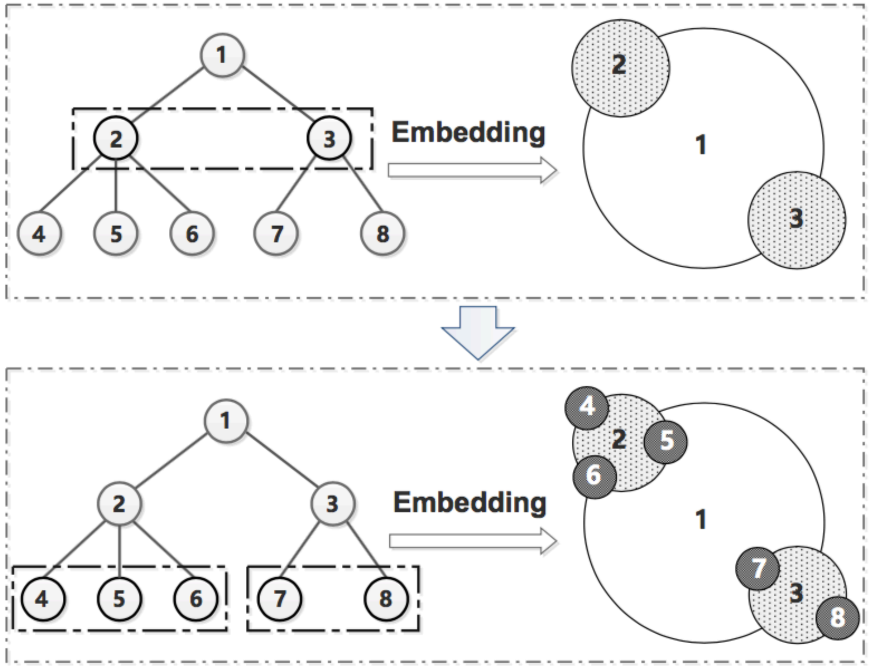


Figure 2: Structure of GNE

Proof

Galaxy Network Embedding

$$\min_{\Phi,\Phi'} O_k^{(l-1)} = - \sum_{c_i^l, c_j^l \in Ch(c_k^{l-1})} S_{i,j}^l \log P(\Phi'(c_j^l) \mid \Phi(c_i^l))$$
$$s.t. \quad \forall c_i^l \in Ch(c_k^{l-1}), \quad \|\Phi(c_i^l) - \Phi(c_k^{l-1})\|_2 = r_k^{l-1}. \tag{5}$$

where,

$$r_i^l = \eta \cdot d_k^{l-1}, \quad \eta < \frac{1}{6}$$
$$d_k^{l-1} = \min_{c_i^l, c_j^l \in Ch(c_k^{l-1}), i \neq j} Dist\left(\Phi(c_i^l), \Phi(c_j^l)\right), \tag{6}$$
$$Dist(x, y) = \|x - y\|,$$

?

=>

Hierarchical Preserving Network Embedding

- Pairwise Proximity Preservation

$$\min_{\Phi,\Phi'} O_k^{(l-1)} = - \sum_{c_i^l, c_j^l \in Ch(c_k^{l-1})} S_{i,j}^l \log P(\Phi'(c_j^l) \mid \Phi(c_i^l))$$

- Hierarchical Structure Preservation
  - Horizontal relationship:

$$\|\Phi(c_u^l) - \Phi(c_v^l)\| < \|\Phi(c_u^l) - \Phi(c_w^l)\|, \tag{3}$$

  - Vertical relationship:

$$\|\Phi(c_i^{l+1}) - \Phi(c_j^l))\| < \|\Phi(c_j^l) - \Phi(c_k^{l-1})\|. \tag{4}$$

Proof

Lemma 1

The community representations learned from recursively optimizing the objective Eq.(5) with the strategy Eq.(6) preserve the constraints Eq.(3) and Eq.(4).

$$r_i^l = \eta \cdot d_k^{l-1}, \quad \eta < \frac{1}{6}$$

$$d_k^{l-1} = \min_{c_i^l, c_j^l \in Ch(c_k^{l-1}), i \neq j} Dist\left(\Phi(c_i^l), \Phi(c_j^l)\right),$$

$$Dist(x, y) = \|x - y\|,$$

$\Longrightarrow$

Horizontal relationship:

$$\|\Phi(c_u^l) - \Phi(c_v^l)\| < \|\Phi(c_u^l) - \Phi(c_w^l)\|,$$

Vertical relationship:

$$\|\Phi(c_i^{l+1}) - \Phi(c_j^l))\| < \|\Phi(c_j^l) - \Phi(c_k^{l-1})\|.$$

Experiment

Dataset

- Facebook social network datasets:
  - Amherst College
  - Hamilton University
  - Georgetown University
- Hierarchical random graphs (HRG):
  - syn_with_125nodes
  - syn_with_1800nodes
  - syn_with_2560nodes
  - syn_with_3750nodes

Baselines

- Spectral Clustering [Tang and Liu, 2011]
- DeepWalk [Perozzi *et al*, 2014]
- Node2vec [Grover and Leskovec, 2016]
- LINE [Tang *et al.*, 2017]
- GraRep [Cao *et al.*, 2015]
- MNMF [Wang et al., 2017]

Experiment

Hierarchical Community Detection

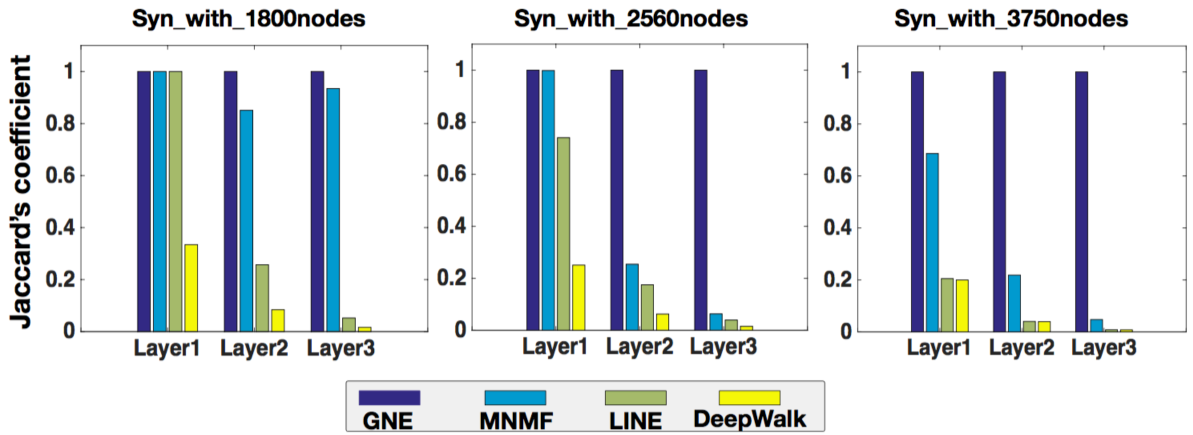


Figure 3: The comparison of hierarchical community preservation on different models. Three different structures of HRG with the same number of layers are used.

Experiment

Network Visualization

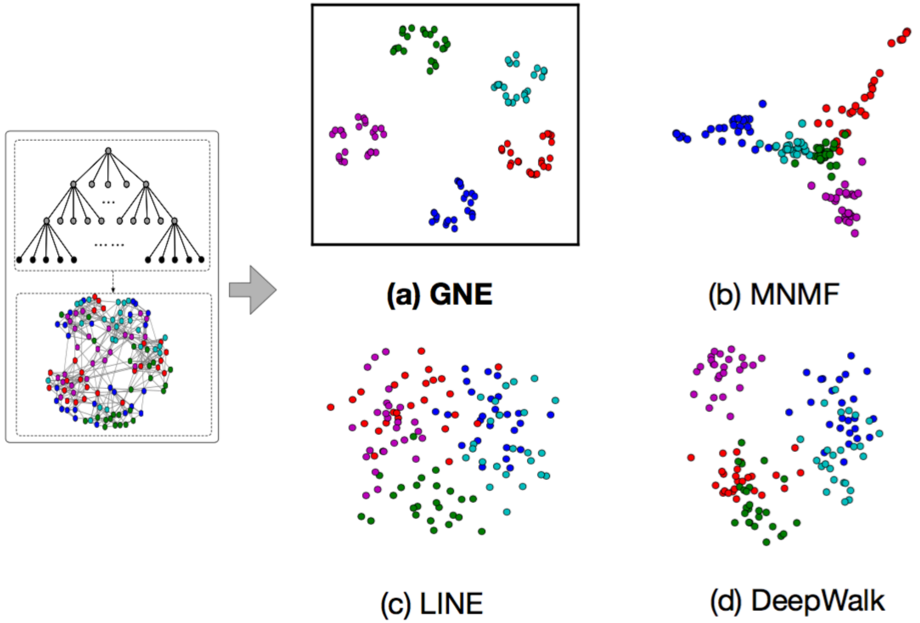


Figure 4: The visualization of vertex representations on different models

Experiment

Vertex Classification

Model	Amherst					Hamilton					Georgetown				
	10%	30%	50%	70%	90%	10%	30%	50%	70%	90%	10%	30%	50%	70%	90%
GNE	**93.57**	**93.31**	**93.33**	**93.18**	**92.89**	**94.83**	**94.53**	**94.11**	**94.17**	**93.86**	53.22	**53.80**	53.55	**52.20**	**51.88**
SpectralClustering	72.89	73.49	73.94	74.32	72.82	78.16	77.60	77.21	77.59	74.92	49.26	50.87	50.79	50.60	48.53
DeepWalk	90.62	91.65	91.32	91.13	90.41	92.89	92.33	92.52	92.18	91.55	54.07	53.79	53.35	51.69	50.92
Node2Vec	91.29	91.24	91.04	90.44	90.02	92.09	91.03	91.18	90.06	89.56	52.86	53.73	53.16	52.70	51.28
LINE	90.76	91.82	91.48	91.09	89.42	92.33	92.72	92.52	92.62	91.73	54.64	53.45	53.81	52.71	51.28
GraRep	92.13	92.25	91.78	91.56	91.48	93.67	93.04	92.30	92.40	91.00	**54.80**	53.24	**53.95**	51.87	51.74
MNMF	89.82	89.06	88.04	86.43	78.44	91.42	90.32	89.12	87.02	81.19	53.43	52.63	52.10	51.52	50.35

Table 1: The multi-label classification results on different percentages of test datasets

Hierarchical Community Structure Preserving Network Embedding: A Subspace Approach

Lun Du[*], *Qingqing Long*[*], *Yiming Wang*[*], *Guojie Song*[ʄ], *YiLun jin*, *Wei Lin*. Hierarchical Community Structure Preserving Network Embedding: A Subspace Approach. *Accepted by CIKM, 2019.*

Drawbacks of GNE

- Failed when embedding deeper communities
  - Radii shrink exponentially
  - Data sparsity in a deeper community
- Probably overvalued hierarchical information
  - Vertices across community are exponentially distant than those within the same community.

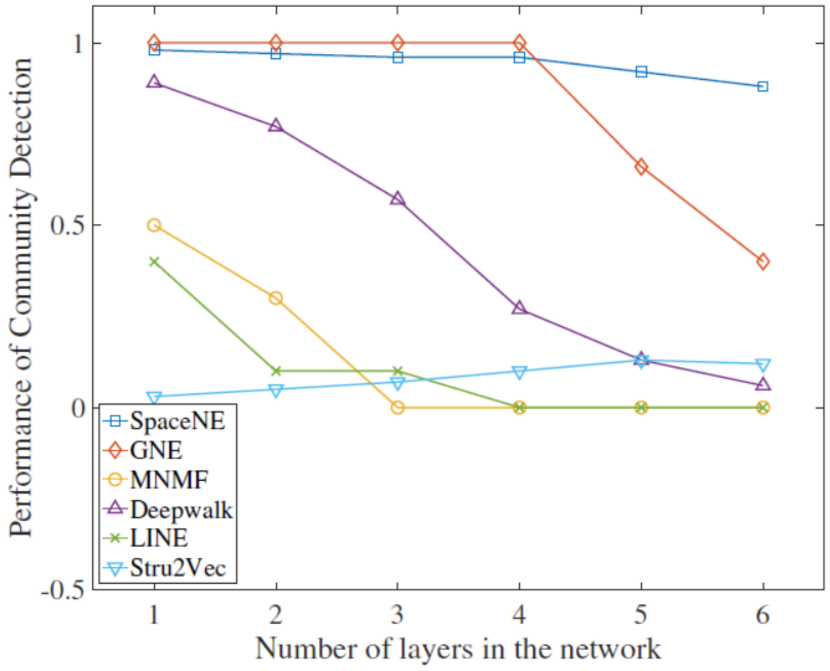
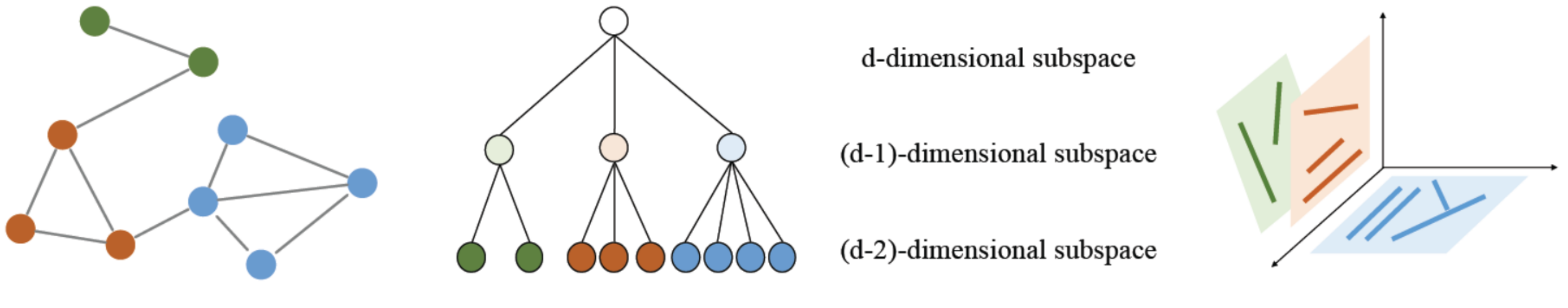


Figure 5: The comparison of hierarchical community preservation on different models. A 6-layer generated hierarchical networks is used.

How to Represent?

- Subspace
  - Natural hierarchical structure
  - Deeper community corresponding to lower dimensional subspace



(a) Network (b) Hierarchical community structure (c) 3-d hierarchical subspace

Figure 1: The correspondence between the community hierarchy and the subspace hierarchy

How to Learn?

- Formulating the problem into an optimization problem with subspace constraints
  - Modeling community affiliation by subspace
  - Reducing the representation dim by constraining the rank of base vectors
- Designing efficient learning algorithm
  - From global to layer-wise optimization
  - From discrete to differentiable optimization

Hierarchical Structure Preserved

- Preservation of Structure within Individual Communities

$$\mathcal{L}_1 = \sum_{(i,j)\in E} \log \sigma(||\overrightarrow{u_j}^{(0)} - \overrightarrow{u_i}^{(0)}||) + k \cdot \mathbb{E}_{v_n \sim P_n} [\log \sigma(-||\overrightarrow{u_n}^{(0)} - \overrightarrow{u_i}^{(0)}||)]$$

Where, $\overrightarrow{u_i}^{(l)} = S_{f_i^l} \overrightarrow{u_i}^{(l-1)}$ for $l = 1 \ldots L, v_i \in V$

- Preservation of Structure among Communities

$$\mathcal{L}_2 = \sum_{l=0}^{L} \sum_{i=1}^{|C_l|} \sum_{j=i+1}^{|C_l|} (\Delta_{ij}^l - \Gamma_{ij})^2$$

- Low Rank Constraints

$$\mathcal{L}_3 = \sum_{l=0}^{L} \sum_{i=1}^{|C_l|} rank(S_i^l)$$

Experiment

Vertex Classification

Model	Amherst				Georgetown				UC			
	30%	50%	70%	90%	30%	50%	70%	90%	30%	50%	70%	90%
SpaceNE	92.52	93.11	**93.74**	**95.09**	**56.12**	**56.42**	**56.92**	**56.54**	**88.69**	**89.02**	**89.23**	**90.07**
GNE	**93.17**	**93.33**	93.26	93.52	52.19	53.53	53.75	53.12	87.78	88.42	88.42	87.57
MNMF	87.11	88.04	89.23	89.96	51.52	51.69	51.60	53.25	87.89	87.95	88.09	88.10
DeepWalk	91.09	91.26	91.71	92.03	51.45	53.25	53.76	54.03	88.35	88.42	88.51	88.63
LINE	91.11	91.53	91.89	91.67	51.35	51.93	52.18	52.38	87.71	87.88	87.95	87.53
Struc2Vec	72.72	73.35	73.92	77.23	46.85	47.44	48.33	47.59	87.96	87.89	88.11	88.25
SpectralClustering	72.88	73.51	73.89	74.41	49.67	50.02	50.79	51.23	84.23	84.35	84.31	84.21

Table 2: The multi-label classification results on different percentages of train datasets

Experiment

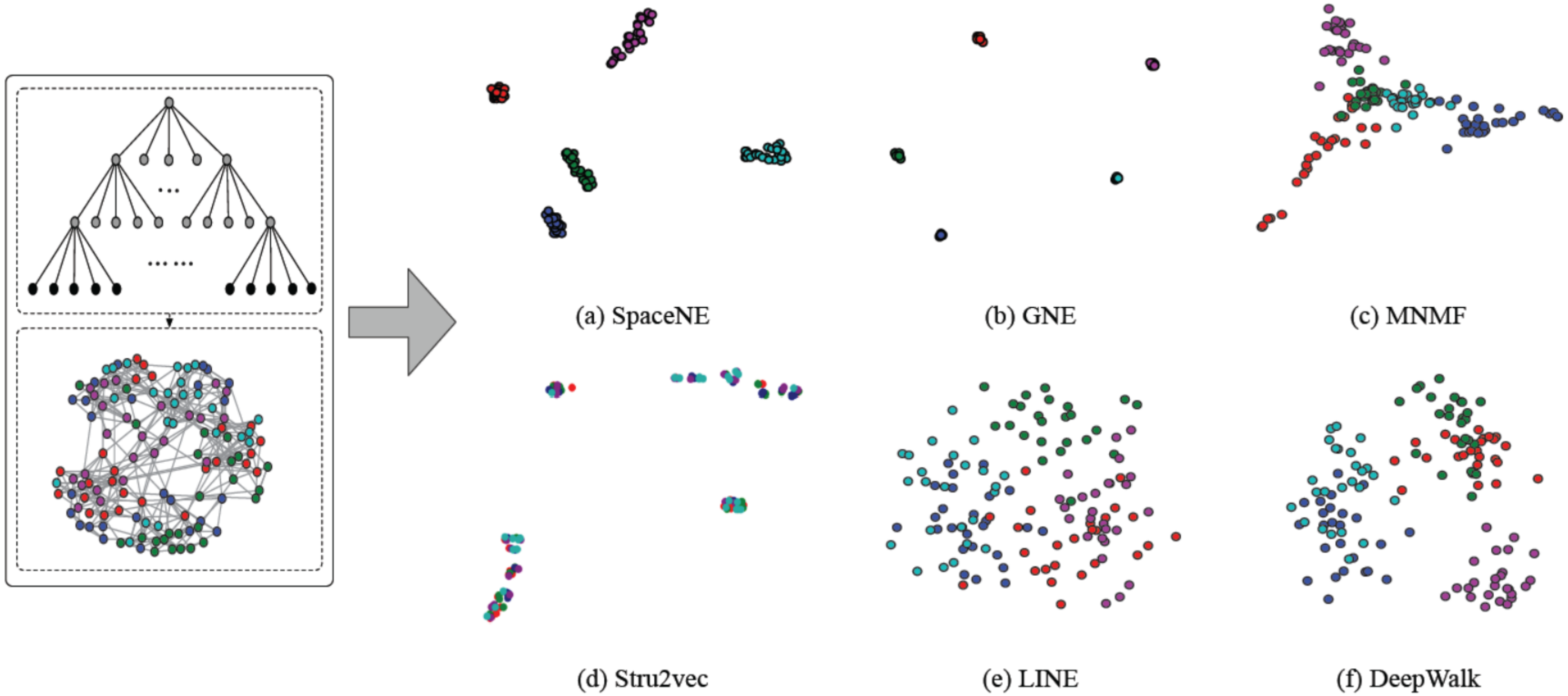
Link Prediction

Model	Amherst	Georgetown	UC
SpaceNE	85.61	89.28	91.32
GNE	62.07	68.97	51.25
MNMF	48.89	49.76	50.05
DeepWalk	86.40	89.16	91.39
LINE	74.37	76.58	71.22
Struc2Vec	51.77	49.94	46.83
SpectralClustering	37.76	40.63	38.68

Table 4: The link prediction results on different datasets.

Experiment

Network Visualization



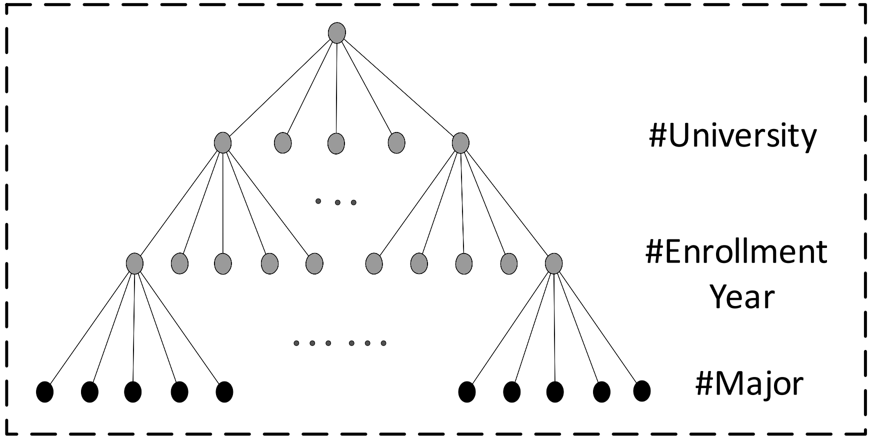
(a) SpaceNE (b) GNE (c) MNMF

(d) Stru2vec (e) LINE (f) DeepWalk

Outline

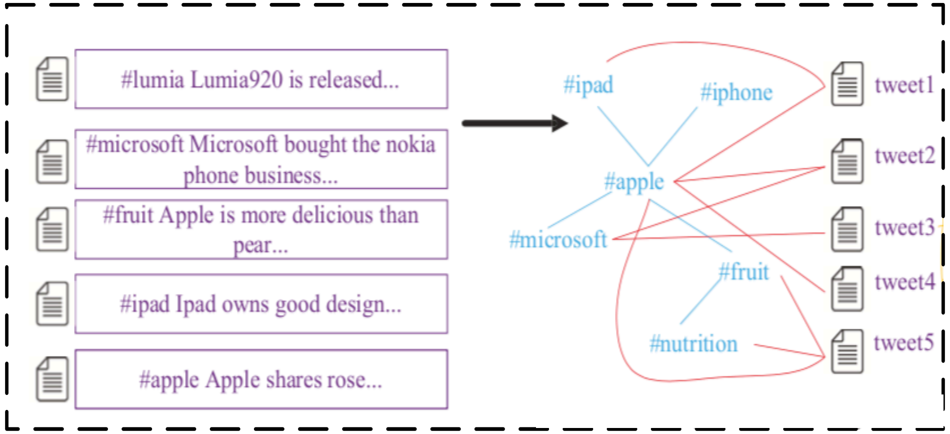
- Hierarchical Info can be observed to a certain extent in online networks.

Explicit hierarchy with attributes



Facebook Network

Implicit hierarchy with tags



Twitter Network

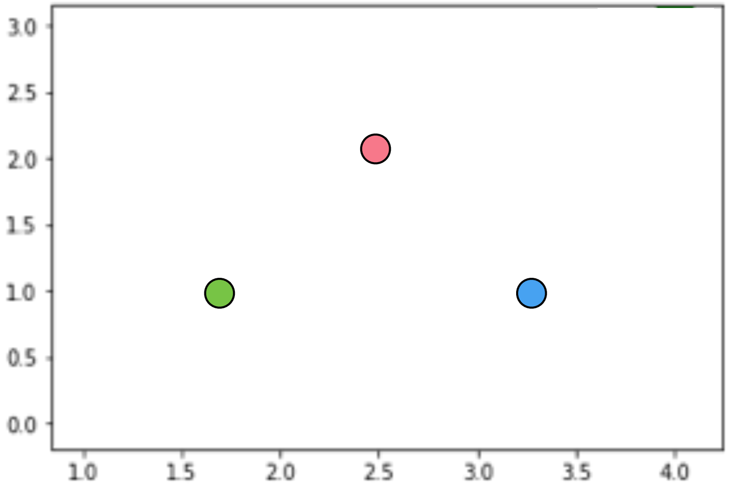
Microsoft

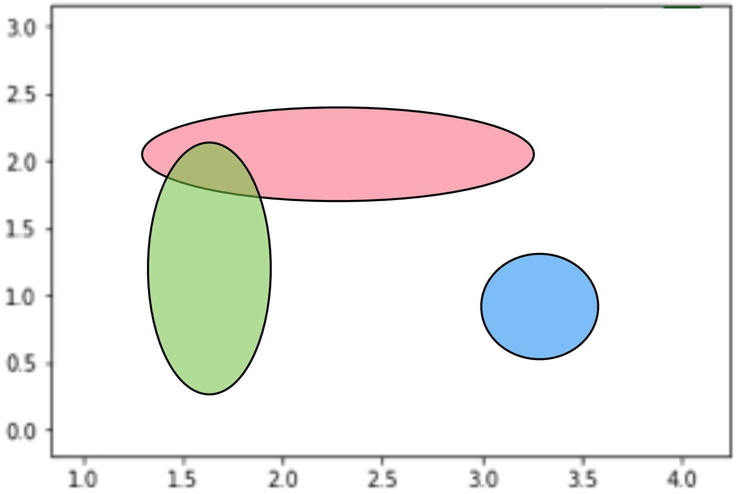
Tag2Gauss: Learning Tag Representations via Gaussian Distribution in Tagged Networks

Lun Du[*], *Yun Wang*[*], *Guojie Song*[†], *Xiao Ma, Lichen Jin, Wei Lin, Fei Sun*. Tag2Gauss: Learning Tag Representations via Gaussian Distribution in Tagged Networks. *In Proceedings of IJCAI, 2019.*

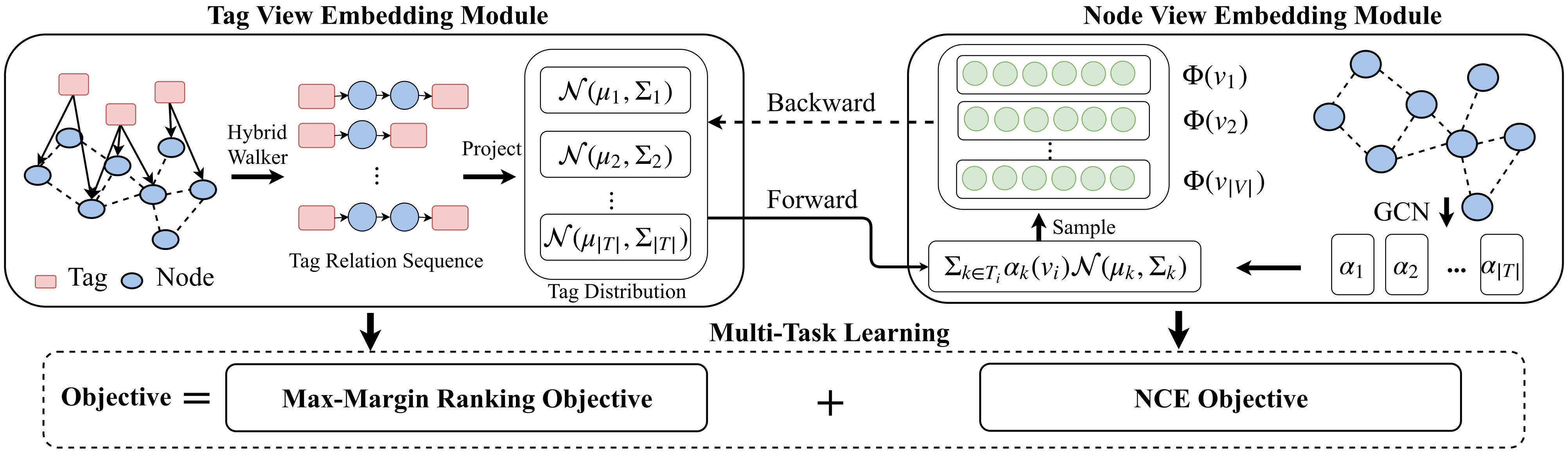
How to Represent?

- Represent tags and nodes simultaneously
- Tags represent node communities with intricate overlapping relationships
- Distribution: Tag; Sample from distributions: Node





How to learn?



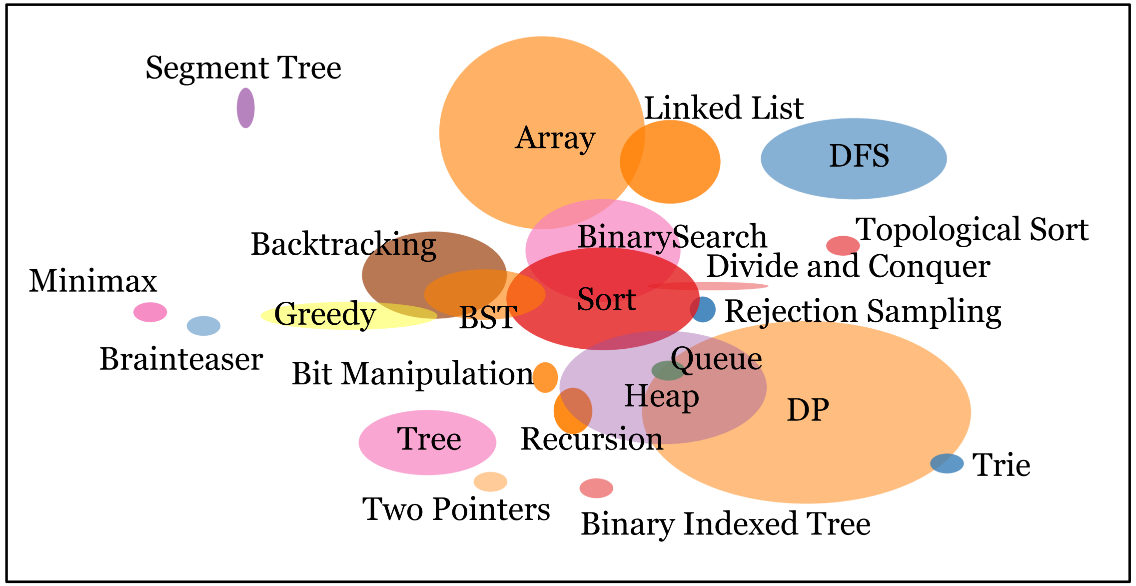
- Tag2Gauss Framework:
  - Tag-view Embedding
  - Node-view Embedding
  - Multi-task Learning

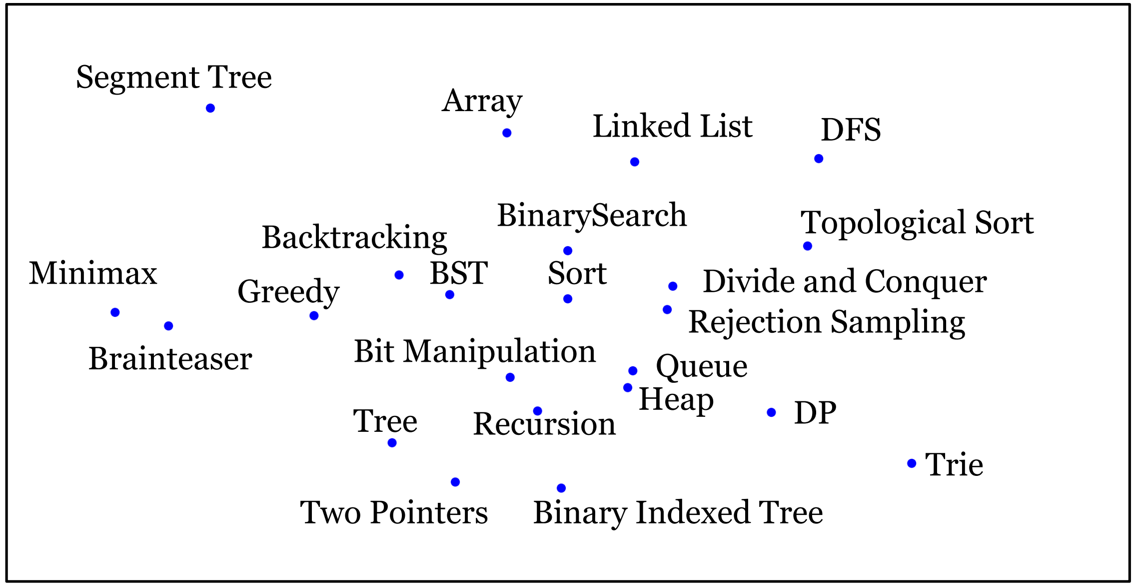
Experiments

- Datasets:
  - Leetcode (652 nodes, 1096 edges, 34 tags, 3 labels)
  - Bilibili (11727 nodes, 187148 edges, 151 tags, 10 labels)
  - Cora. (2707 nodes, 5429 edges, 1433 tags, 7 labels)
- Baselines
  - DeepWalk (KDD’14)
  - Node2vec (KDD’16)
  - Hybrid Deepwalk (Naive Design)
  - GraphSage (NIPS’17)

Experiment

The Advantage of Distribution Representations





Experiment

Node Classification

Model	Leetcode					Bilibili					Cora				
	10%	30%	50%	70%	90%	10%	30%	50%	70%	90%	10%	30%	50%	70%	90%
Node2Vec	36.37%	36.37%	38.68%	37.63%	39.68%	48.19%	48.19%	45.36%	45.36%	42.88%	57.12%	57.40%	57.40%	50.84%	48.84%
LINE	34.41%	38.59%	35.89%	33.66%	40.46 %	6.55%	7.21%	7.65%	8.30%	9.28%	49.00%	49.96%	46.23%	45.48%	39.13%
GraphSage	34.00%	37.37%	36.65%	39.77%	44.37%	61.48%	60.81%	60.52%	59.02%	54.26%	50.95%	51.63%	49.10%	45.70%	34.15%
Tag2Gauss	**42.27%**	**42.68%**	**43.70 %**	**44.04%**	**45.03%**	**61.65%**	**61.23%**	**60.83%**	**60.58%**	**56.85%**	**68.45%**	**67.21%**	**66.56%**	**64.87%**	**63.26%**

Table 1: The comparison of node classification measured by Macro-F_1 on different models and different training size.

Main Contents

- Background
- Graph embedding with hierarchical community structure
- **Domain adaptive graph embedding**
- Future works

DANE: Domain Adaptive Network Embedding

Yizhou Zhang, Guojie Song[†], **Lun Du**, *Shuwen Yang, Yilun Jin.* DANE: Domain Adaptive Network Embedding. *In Proceedings of IJCAI, 2019.*

Motivation

- Domain adaptation
  - Transferring machine learning models across different datasets to handle the same task
- Domain adaptation on networks is significant:
  - Reduce the cost of training downstream machine learning models by enabling models to be reused on other networks
  - Handle the scarcity of labeled data by transferring models trained well on a labeled network to unlabeled networks
- It is important to design a network embedding algorithm that can support domain adaptation.

Challenges

- Embedding space alignment
  - Structurally similar nodes should have similar representations in the embedding space, even if they are from different networks.
- Distribution alignment
  - Embedding vectors of different networks should have similar distribution in embedding space.
  - Most machine learning models perform as guaranteed only when they work on data with similar distribution as their training data.

Technique Framework: Overall

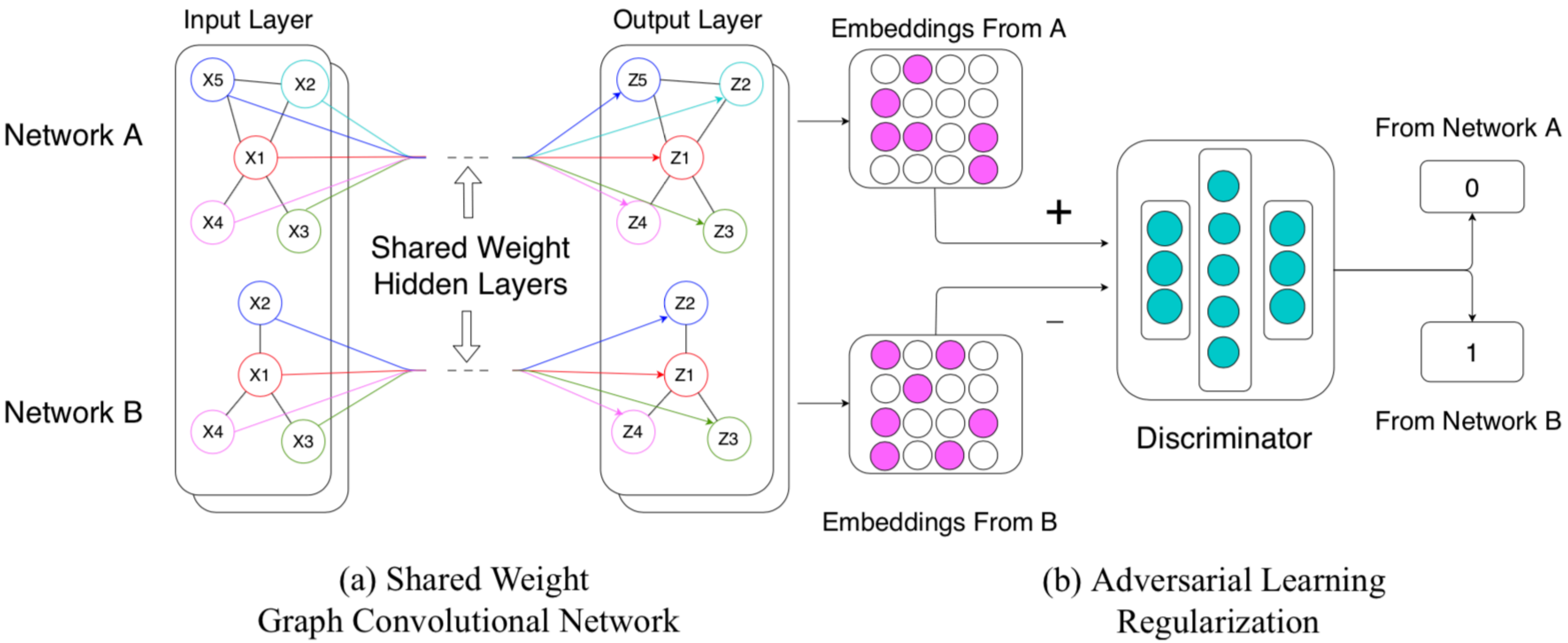


Figure 1: An overview of DANE. DANE consists of two major components: (a) shared weight graph convolutional network (SWGCN) projects the nodes from two networks into a shared embedding space and preserve cross-network similarity; (b) adversarial learning regularization is a two-player game where the first player is a discriminator trained to distinguish which network a representation vector is from and the second player is the SWGCN trying to generate embeddings that can confuse the discriminator.

Adversarial Learning Regularization

- Discriminator to avoid the instability of adversarial learning:

$$L_D = \mathbb{E}_{x \in V_{src}}[(D(x) - 0)^2] + \mathbb{E}_{x \in V_{tgt}}[(D(x) - 1)^2]$$

- Adversarial training loss function to confuse the discriminator is:

$$L_{adv} = \mathbb{E}_{x \in V_{src}}[(D(x) - 1)^2] + \mathbb{E}_{x \in V_{tgt}}[(D(x) - 0)^2]$$

- Overall loss function

$$L = L_{gcn} + \lambda L_{adv}$$

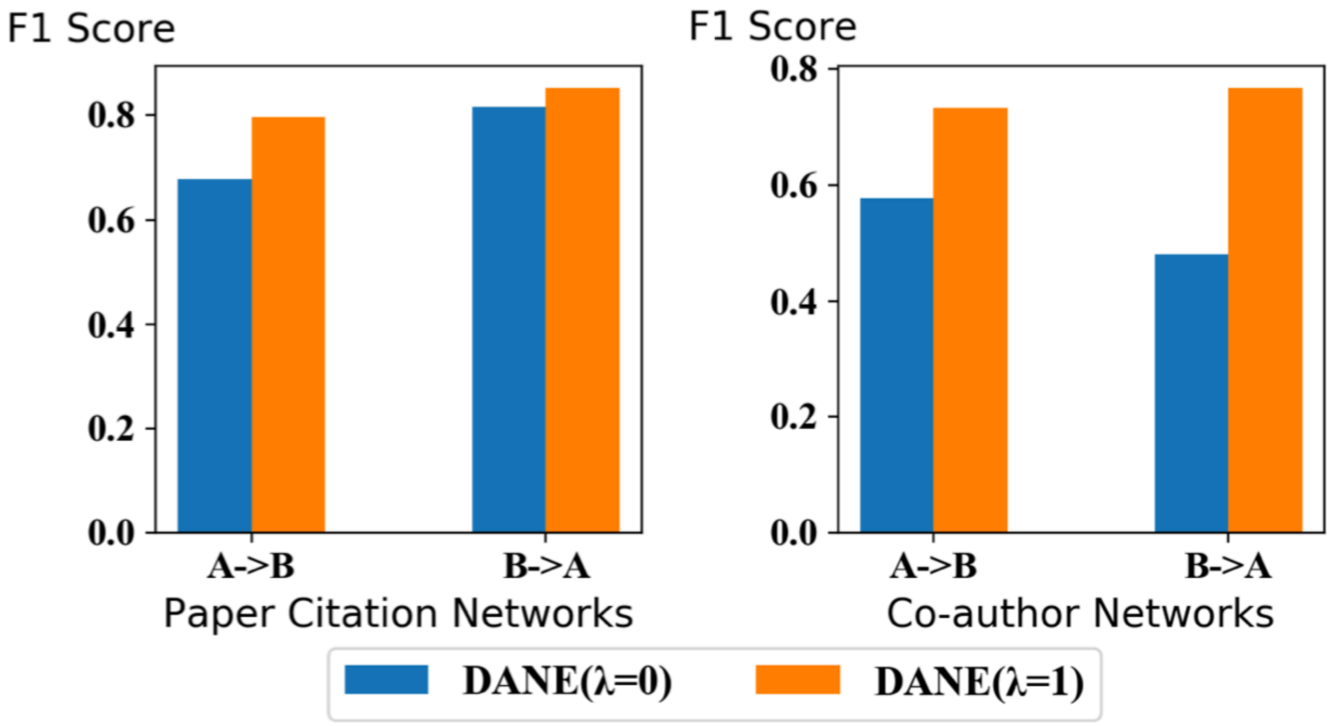
Experiment

Comparison with Baselines

Methods	Paper Citation Network				Co-author Network			
	A→B		B→A		A→B		B→A	
	Macro F1	Accuracy	Macro F1	Accuracy	Macro F1	Accuracy	Macro F1	Accuracy
DeepWalk	0.282	0.381	0.22	0.32	0.238	0.250	0.267	0.280
LINE	0.156	0.214	0.175	0.272	0.232	0.261	0.262	0.262
Node2vec	0.147	0.196	0.248	0.32	0.283	0.294	0.264	0.273
GraphSAGE Unsup	0.671	0.703	**0.861**	0.853	0.631	0.650	0.680	0.678
DANE	**0.797**	**0.803**	0.852	**0.872**	**0.732**	**0.742**	**0.767**	**0.774**

Experiment

Comparison with the Variant without adversarial learning



Main Contents

- Background
- Graph embedding with hierarchical community structure
- Domain adaptive graph embedding
- **Future works**

Future Works

- Understanding of graph neural networks
  - Why does it work?
  - What kind of graph is it more effective?
- Customized GNN for different kinds of graphs
- Applications
  - Semi-structured data mining
  - Source code analytics

Q & A

Welcome to collaboration or internship!

lun.du@microsoft.com