Inductive and Unsupervised Representation Learning on Graph Structured Objects

Lichen Wang[1] , Bo Zong[2], Qianqian Ma[3], Wei Cheng[2], Jingchao Ni[2], Wenchao Yu[2], Yanchi Liu[2], Dongjin Song[2], Haifeng Chen[2], Yun Fu[1]

[1]Northeastern University
[2]NEC Laboratories America
[3]Boston University

SmileLab website: https://web.northeastern.edu/smilelab/
Email: wanglichenxj@gmail.com

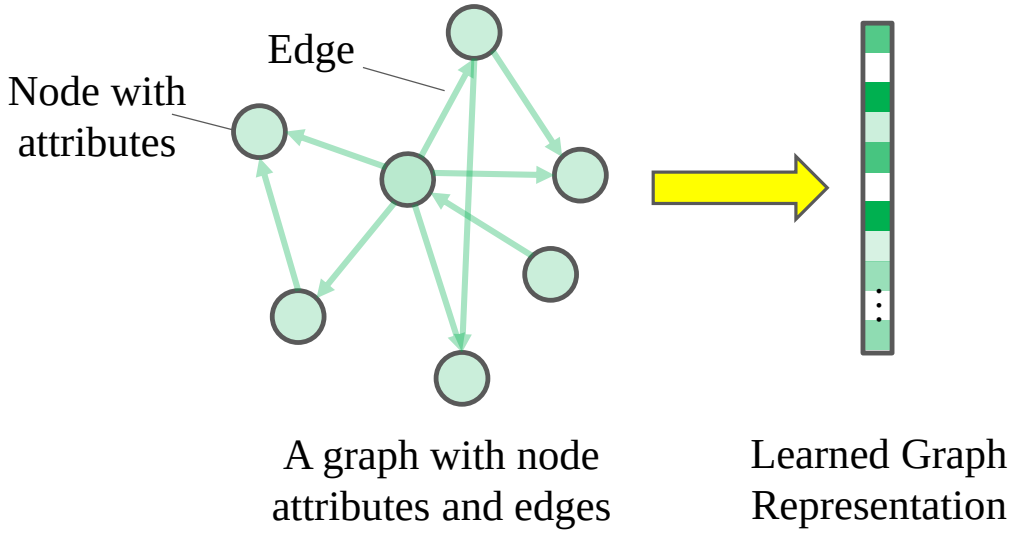
Problem Setting

- Input:

  Graph with node attributes and edge attributes

- Output:

  Graph representation as a vector



Concept of Graph Representation Learning

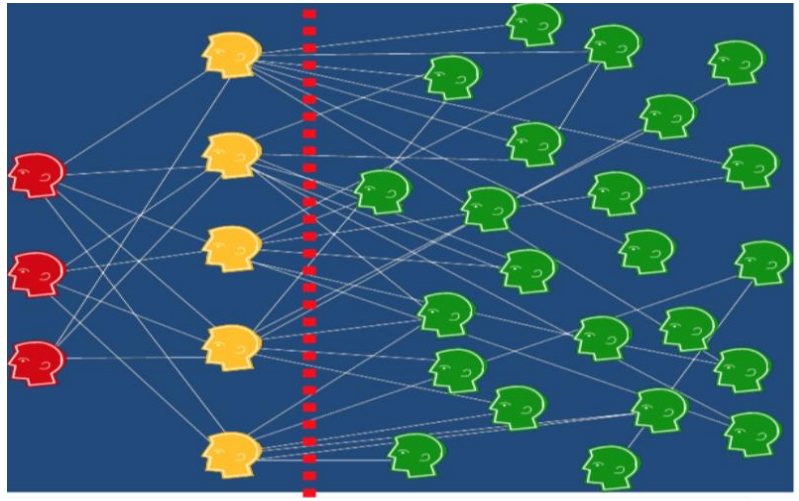
Why Inductive and Unsupervised are Important?

Graph objects have a wide range of potential applications [1]:

- Social Network
  - Facebook, Twitter, WhatsApp
- Finance
  - Credit card fraud, Money laundry
- Logistics Industry:
  - eBay, Amazon, FedEx

Problems:

- Not enough labeled samples
- Learned model should be generalized to unseen data



Fake Social Account

Credit Fraud

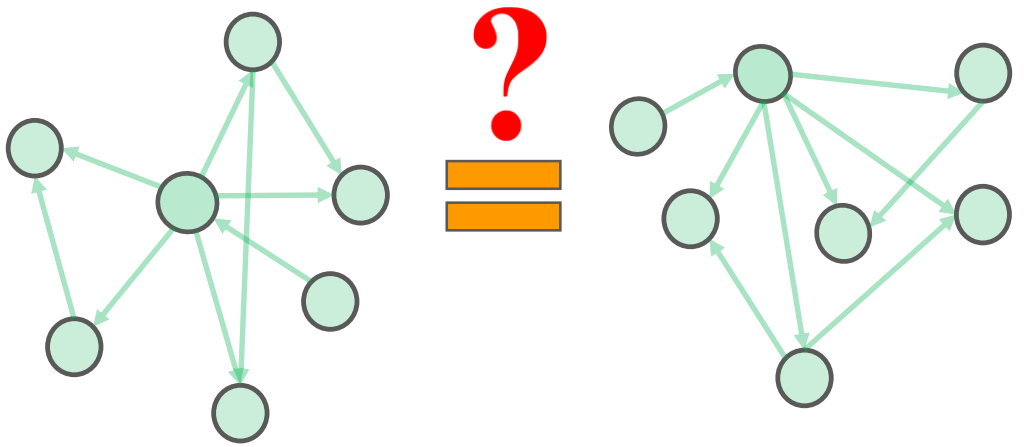
Computer Hack

[1] Chau, Duen Horng, Shashank Pandit, and Christos Faloutsos. "Detecting fraudulent personalities in networks of online auctioneers." PKDD, 2006
[2] https://datafloq.com/read/will-analytics-technology-end-credit-card-fraud/2121

Challenges

- Existing approaches are in transductive setting
  - Difficult to handle unseen graphs
- Reconstruction-based approach
  - How similar of two graphs?
  - Graph Isomorphism is hard and rigid
  - Computational costly

We proposed a framework that addresses the practical need for graph representation learning in real-life applications



Isomorphism test is a necessary but hard and computational cost in graph representation learning

The proposed Framework: SEED (1)

SEED: **S**ampling, **E**ncoding, and **E**mbedding **D**istributions

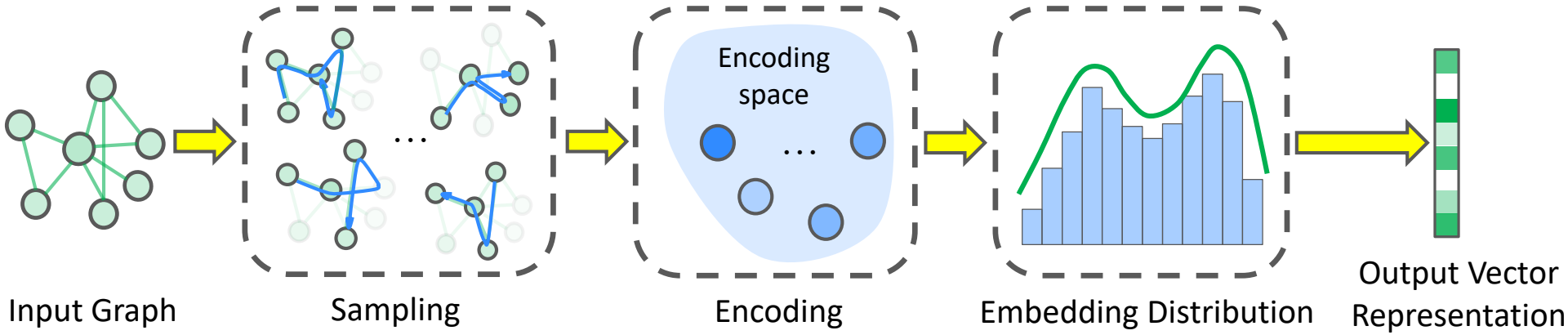
- **Sampling**: Random walk-based subgraph sampling from the input graph

  - Difficult to directly get whole graph representations

  - Could be easier to obtain representations for walks

- **Encoding**: Subgraph encoding via earliest visiting time

  - Make the process efficient and the representations effective



Framework of our SEED approach

The proposed Framework: SEED (2)

SEED: **S**ampling, **E**ncoding, and **E**mbedding **D**istributions

- **Embedding Distributions**:

  We encode a vector distribution into a single vector, which should preserve the similarity between vector distributions.

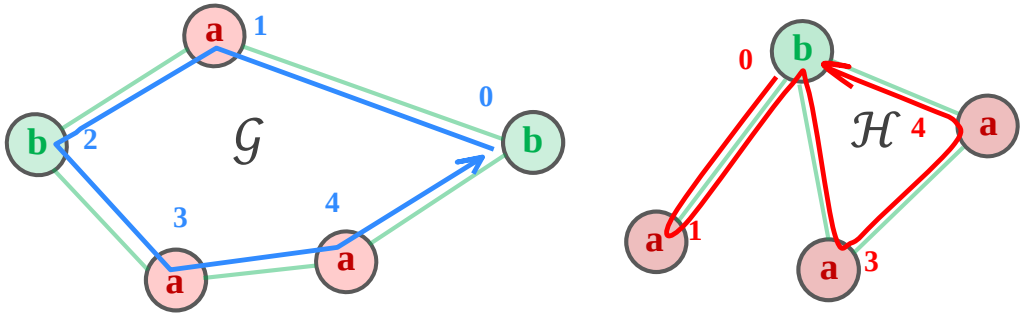
  - Each input graph is reduced into a set of vectors, each of which is the representation for a sampled subgraph.

  - Given that we have sampled a sufficient number of subgraphs, if two input graphs are similar, their vector distributions should be similar

Sampling & Encoding

WEAVE: Random **W**alk with **EA**rliest **V**isit tim**E**).

- Random walk (RW) in graphs
- Revisit information: earliest visiting time
- Advantages:
  - RW: easy to reconstruct, but no loop info preserved
  - RW + revisit: easy to reconstruct with loop info
  - RW with revisit contains more structural info



Vanilla random walk:
b-a-b-a-a-b
b-a-b-a-a-b

WEAVE:
$\begin{bmatrix} b \\ 0 \end{bmatrix}$-$\begin{bmatrix} a \\ 1 \end{bmatrix}$-$\begin{bmatrix} a \\ 2 \end{bmatrix}$-$\begin{bmatrix} a \\ 3 \end{bmatrix}$-$\begin{bmatrix} a \\ 4 \end{bmatrix}$-$\begin{bmatrix} b \\ 5 \end{bmatrix}$
$\begin{bmatrix} b \\ 0 \end{bmatrix}$-$\begin{bmatrix} a \\ 1 \end{bmatrix}$-$\begin{bmatrix} a \\ 0 \end{bmatrix}$-$\begin{bmatrix} a \\ 3 \end{bmatrix}$-$\begin{bmatrix} a \\ 4 \end{bmatrix}$-$\begin{bmatrix} b \\ 0 \end{bmatrix}$

Encoding results of Vanilla random walk and WEAVE. WEAVE could distinguish the difference of the two graphs.

Embedding Distribution

- Insight: Walk distribution representation similarity $\Longrightarrow$ graph similarity
- Theoretical: as proved, distribution $R_{\mathcal{g}} = R_{\mathcal{H}}$ if graph $\mathcal{g}$ and $\mathcal{H}$ are isomorphic
- Option 1: Identity kernel

  - We assume $r_{\mathcal{g}} \sim N(\mu_1, I)$ and $r_{\mathcal{H}} \sim N(\mu_2, I)$, it is simple but surprisingly effective.

$$\hat{\mu}_{\mathcal{G}} = \frac{1}{s}\sum_{i=1}^{s}\mathbf{z}_i \qquad \hat{\mu}_{\mathcal{H}} = \frac{1}{s}\sum_{i=1}^{s}\mathbf{h}_i$$

- Option 2: Commonly adopted kernels

$$\hat{\mu}'_{\mathcal{G}} = \frac{1}{s}\sum_{i=1}^{s}\hat{\phi}(\mathbf{z}_i;\theta_m) \quad \hat{\mu}'_{\mathcal{H}} = \frac{1}{s}\sum_{i=1}^{s}\hat{\phi}(\mathbf{h}_i;\theta_m) \quad D(P_{\mathcal{G}}, P_{\mathcal{H}}) = \|\hat{\mu}'_{\mathcal{G}} - \hat{\mu}'_{\mathcal{H}}\|_2^2$$

Theoretical Insights

Theorem: Given graphs $\mathcal{g}$ and $\mathcal{H}$, distribution $R_{\mathcal{g}} = R_{\mathcal{H}}$ if graph $\mathcal{g}$ and $\mathcal{H}$ are isomorphic

The theorem holds for the situations:

- Graphs without any attributes
- Graphs with node attributes
- Graphs with node and edge attributes

Experiments (1)

- Seven graph datasets
- Two down-stream tasks:
  - Clustering
  - Classification
- Our approach obtains the highest performance.
  - Up to 10% improvements

Setting	Datasets	Methods / Metric	SAGE	GIN	GMN	SEED	SAGE	GIN	GMN	SEED
			Node Feature **Excluded**				Node Feature **Included**			
Clustering	Dezzer	ACC	0.3853	0.4913	0.4924	**0.4927**	0.3840	**0.4930**	0.4808	0.4810
		NMI	0.0079	0.0958	0.0726	**0.1277**	0.0003	**0.0893**	0.0651	0.0566
	MUTAG	ACC	0.6649	0.4997	0.4990	**0.8014**	0.6649	0.4963	0.4910	**0.7260**
		NMI	0.0150	0.0946	0.0825	**0.3214**	0.0070	0.0933	0.0917	**0.1567**
	NCI1	ACC	0.5098	0.5221	0.5022	**0.5510**	0.5070	0.5204	0.5005	**0.5441**
		NMI	0.0003	0.0015	0.0034	**0.0073**	0.0002	0.0013	0.0042	**0.0089**
	PROTEINS	ACC	0.5657	0.5957	**0.5966**	0.5957	0.5657	0.5957	0.5957	**0.5957**
		NMI	0.0013	0.0038	0.0117	**0.0518**	0.0004	0.0034	0.0067	**0.0689**
	COLLAB	ACC	0.5208	0.5458	0.5173	**0.5973**	-	-	-	-
		NMI	0.0025	0.0729	0.0193	**0.2108**	-	-	-	-
	IMDB-BINARY	ACC	0.5069	0.6202	0.5010	**0.5776**	-	-	-	-
		NMI	0.0002	0.0459	0.0093	**0.0241**	-	-	-	-
	IMDB-MULTI	ACC	0.3550	3607	0.3348	**0.3816**	-	-	-	-
		NMI	0.0019	0.0185	0.0112	**0.0214**	-	-	-	-
Classification	Dezzer	ACC	0.3775	0.5094	0.5427	**0.6327**	0.3754	0.5270	0.5627	**0.7451**
	MUTAG	ACC	0.6778	0.6778	0.6889	**0.8112**	0.6889	0.6778	0.6889	**0.8222**
	NCI1	ACC	0.5410	0.5571	0.5123	**0.6105**	0.5328	0.5231	0.5133	**0.6151**
	PROTEINS	ACC	0.6846	**0.7387**	0.6216	0.7207	0.7027	0.7207	0.6357	**0.7462**
	COLLAB	ACC	0.5650	0.6170	0.5460	**0.6720**	-	-	-	-
	IMDB-BINARY	ACC	0.5400	0.7310	0.5140	**0.7660**	-	-	-	-
	IMDB-MULTI	ACC	0.3866	0.3843	0.3478	**0.4466**	-	-	-	-

Clustering & Classification Performance

Experiments (2)

How parameters impact the output quality?

- Subgraph extraction with different sampling number and walk length.
  - Quantitative performance
  - t-SNE[1] visualization

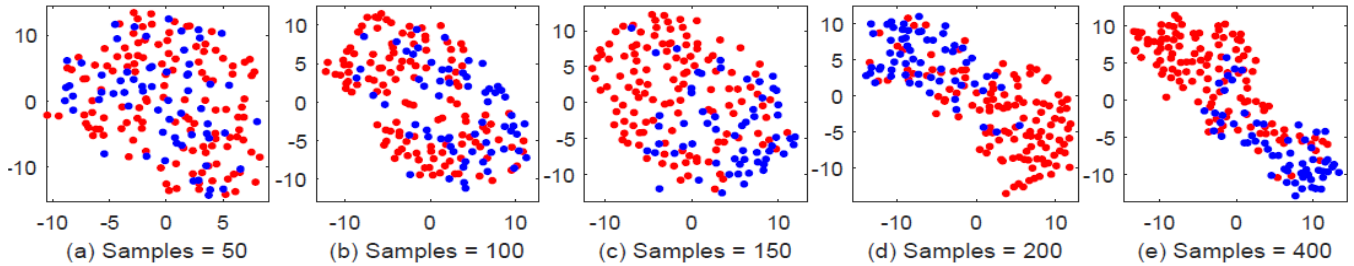
Summary

- More sampling number and walk length could improve the learned representation quality

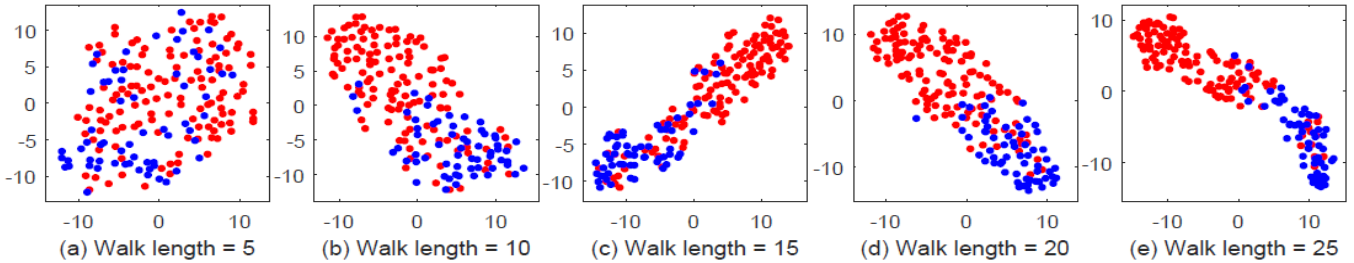
Sampling Number	Classification Accuracy	Clustering ACC	Clustering NMI
25	0.6832	0.6649	0.0031
50	0.6778	0.6649	0.0005
100	0.7778	0.6649	0.0537
150	0.7889	0.6968	0.1081
200	0.7778	0.7633	**0.2100**
300	0.7833	0.7502	0.1995
400	**0.8389**	0.7628	0.1928
800	0.8111	**0.7660**	0.1940

Walk Length	Classification Accuracy	Clustering ACC	Clustering NMI
5	0.7278	0.6649	0.0534
10	0.7778	0.7633	0.2100
15	0.8167	0.7723	0.2495
20	0.8778	0.8245	0.3351
25	0.8722	0.8218	**0.3380**
30	**0.8743**	**0.8285**	0.3321

Classification & clustering performance



t-SNE visualization with different sampling numbers



t-SNE visualization with different work length

[1]] Maaten, Laurens van der, and Geoffrey Hinton. "Visualizing data using t-SNE." Journal of machine learning research 9, no. Nov (2008): 2579-2605.

Northeastern University

Smile lab
Synergetic Media Learning Lab

NEC Laboratories America
Relentless passion for innovation

BOSTON UNIVERSITY

Thank you!

Welcome to contact: **wanglichenxj@gmail.com** for questions