Self-Paced Network Embedding

Hongchang Gao
Department of Electrical and Computer Engineering
University of Pittsburgh
Pittsburgh, USA
hongchanggao@gmail.com

Heng Huang*
Department of Electrical and Computer Engineering
University of Pittsburgh
Pittsburgh, USA
heng.huang@pitt.edu

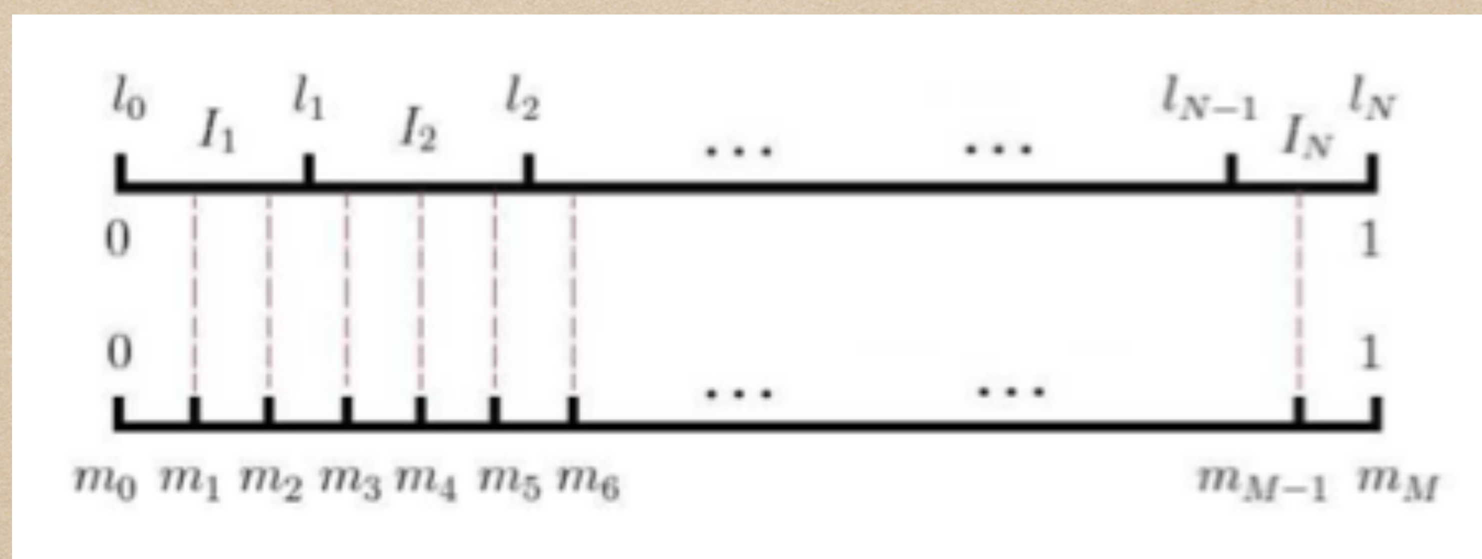
KDD '18, August 19–23, 2018, London, United Kingdom

ACM ISBN 978-1-4503-5552-0/18/08…$15.00
https://doi.org/10.1145/3219819.3220041

Network Embedding

$$\max \prod_{i=1}^{n} \prod_{j \in c_i} p(v_j | v_i)\,, \qquad (1)$$

Negative Sampling



- **Pre-designed distribution—Intuition**
- **ignores that the less connected nodes may also be informative in practice.**
- **the sampling distribution does not change during training.**

Problem Definition

Given a network $\mathcal{G} = \{\mathcal{V}, E\}$ where $\mathcal{V} = \{v_i\}_{i=1}^{n}$ denotes a set of n nodes and $E = [e_{ij}] \in \Re^{n \times n}$ denotes the adjacency matrix, if there exists an edge between node v_i and v_j, $e_{ij} = 1$. Otherwise, $e_{ij} = 0$. The embedding of each node v_i is a low-dimension vector $u_i \in \Re^d$.

employ negative sampling method to accelerate it as follows:

$$\max \log p(v_p | v_i) + \sum_{j \in \mathcal{N}_{v_i}} \log(1 - p(v_j | v_i)),$$

- The popularity-based sampling method cannot really reflect the informativeness of a node. Some less connected nodes can also be much informative in practice.
- The informativeness of a node is usually changing with the training process going on. But the predefined sampling method fails to reflect this change.

Informativeness of Node

Traditional:

$$p(v_j|v_i) = \sigma(u_j^T u_i) = \frac{1}{1 + \exp(-u_j^T u_i)}, \quad (4)$$

easy negative context node → **conditional probability is low**

difficult negative context node → **conditional probability is high**

↓

vj is more informative to the node vi .

informativeness-aware negative sampling distribution

$$p_{ij} = \frac{\exp(u_j^T u_i)}{\sum_{j \in N_{v_i}} \exp(u_j^T u_i)}, \quad (5)$$

Self-Paced Negative Sampling

$$p_{ij} = \frac{\exp(u_j^T u_i)}{\sum_{j \in N_{v_i}} \exp(u_j^T u_i)}, \quad (5)$$

It always selects the difficult negative context node

From easy samples to difficult samples

SeedNE

With the training process going on, l(μ) is increasing difficult negative context nodes will be included gradually.

$$\max \log p(v_p|v_i) + \sum_{j \overset{p'_{ij}}{\sim} N_{v_i}} \log(1 - p(v_j|v_i)) + l(\mu)$$

$$s.t. \quad p'_{ij} = \begin{cases} p_{ij}, & p_{ij} < l(\mu) \\ 0, & otherwise \end{cases},$$

Extension: Adversarial Self-Paced Network Embedding.

$$\min_{\theta} \max_{\phi} E_{v_j \sim p_d(v_j|v_i)} \log p_\phi(v_j|v_i) + E_{v_j \sim G_\theta(v_j|v_i)}[\log(1 - p_\phi(v_j|v_i))]$$

$p_d(v_j|v_i)$ denotes the sampling distribution for the positive context node of node v_i, $G_\theta(v_j|v_i)$ denotes the sampling distribution for the negative context node, which is constructed from the generator.

Discriminator $\max_{\phi} E_{v_j \sim p_d(v_j|v_i)} \log p_\phi(v_j|v_i) + E_{v_j \sim G_\theta(v_j|v_i)}[\log(1 - p_\phi(v_j|v_i))]$,

Generator $\min_{\theta} E_{v_j \sim G_\theta(v_j|v_i)}[\log(1 - p_\phi(v_j|v_i))]$ $G_\theta(v_j|v_i) = \frac{\exp(u_j'^T u_i')}{\sum_j \exp(u_j'^T u_i')}$,

ASeedNE

ignores the easy negative context node

$$\min_{\theta}\max_{\phi} E_{v_j\sim p_d(v_j|v_i)} \log p_\phi(v_j|v_i) + l(\mu)$$
$$+ E_{v_j\sim G'_\theta(v_j|v_i)}[\log(1 - p_\phi(v_j|v_i))]$$
$$s.t. \quad G'_\theta(v_j|v_i) = \begin{cases} G_\theta(v_j|v_i), & G_\theta(v_j|v_i) < l(\mu) \\ 0, & otherwise\ . \end{cases}$$

Policy Gradient

$$\begin{aligned}
&\nabla_\theta E_{v_j\sim G_\theta(v_j|v_i)}[\log(1 - p_\phi(v_j|v_i))] \\
&= \sum_{j=1}^{K} \nabla_\theta G_\theta(v_j|v_i) \log(1 - p_\phi(v_j|v_i)) \\
&= \sum_{j=1}^{K} G_\theta(v_j|v_i) \nabla_\theta \log[G_\theta(v_j|v_i)] \log(1 - p_\phi(v_j|v_i)) \\
&= E_{v_j\sim G_\theta(v_j|v_i)} \nabla_\theta \log[G_\theta(v_j|v_i)] \log(1 - p_\phi(v_j|v_i))\ .
\end{aligned}$$

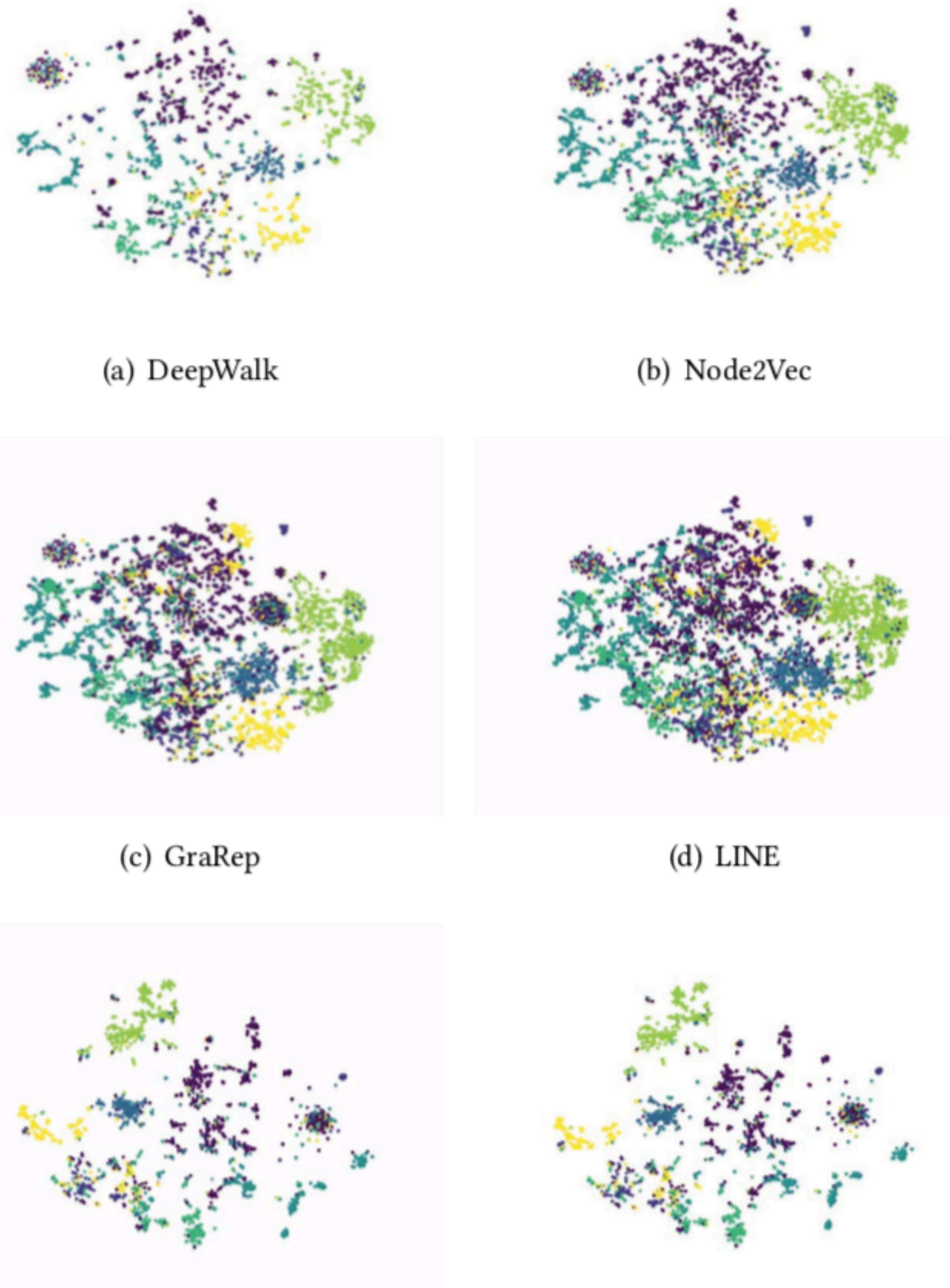
EXPERIMENTS

Table 1: Description of Benchmark Datasets

Dataset	#Nodes	#Edges	#Labels
Wiki	4,777	184,812	40
PPI	3,890	76,584	50
Cora	2,708	5,278	7
Citeseer	3,312	4,660	6
BlogCatalog	10,312	333,983	39
Facebook	4,039	88,234	-
GR-QC	5,242	14,496	-

Baselines

DeepWalk, Node2Vec, GraRep, LINE.



(a) DeepWalk (b) Node2Vec

(c) GraRep (d) LINE

(e) SeedNE (f) ASeedNE

Figure 1: The visualization of Cora dataset.

Table 7: Link Prediction Accuracy

Method	Facebook	GR-QC
DeepWalk	0.9050	0.8354
Node2Vec	0.8900	0.7949
GraRep	0.9445	0.8899
LINE	0.9329	0.8847
SeedNE	**0.9532**	**0.9208**
ASeedNE	**0.9545**	**0.9230**

Method	50%	
	Micro-F1	Macro-F1
DeepWalk	0.3927	0.2556
Node2Vec	0.3965	0.2582
GraRep	0.3674	0.2039
LINE	0.3518	0.1786
SeedNE	**0.4095**	**0.2646**
ASeedNE	**0.4106**	**0.2680**

Thanks and QA