Network Embedding: from Theoretical Understanding to System Design

Jiezhong Qiu

Tsinghua University

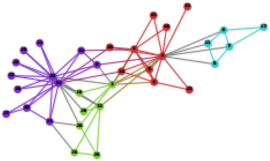
August 13, 2021

About Me

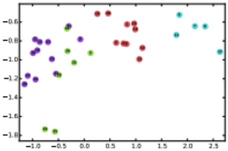
- A 6th year PhD student at Department of Computer Science and Technology of Tsinghua University, under the supervision of Prof. Jie Tang.
- My research interests include algorithm design for large-scale information networks and representation learning for graph-structured data.
- Nomination Award of the 2018 MSRA Fellowship and Nomination Award of the 2020 WAIC Youth Outstanding Paper.
- Google Scholar Citation 1,290.

What is Graph Representation Learning and Network Embedding?

- Network Embedding (NE)

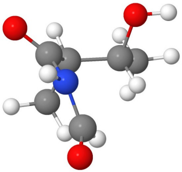


(a) Input: Karate Graph

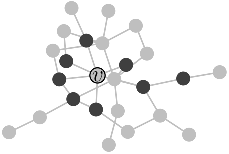


(b) Output: Representation

- Graph Neural Networks (GNN)

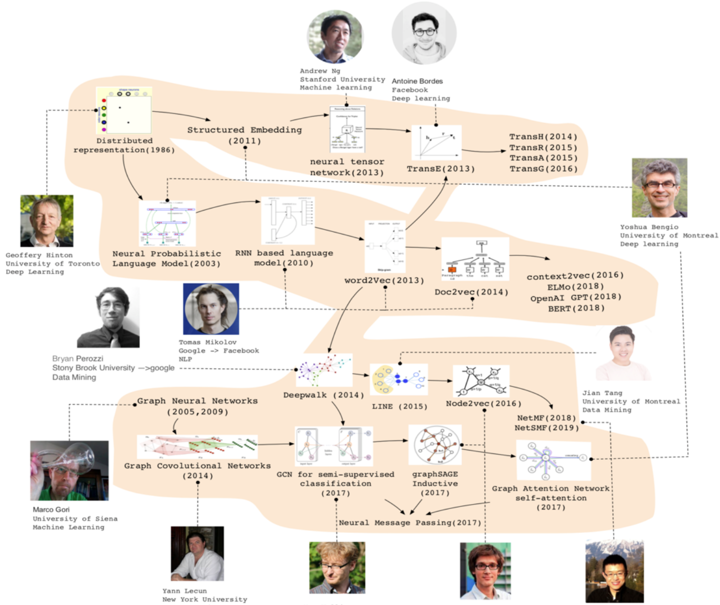


(a) A molecule



(b) An ego-network

History of Graph Representation Learning



About Today's Talk

Three Questions

1. Theory of Network Embedding.
2. Sample complexity of Network embedding algorithms?
3. Design better and scalable network embedding system.

- Network Embedding as Matrix Factorization: Unifying DeepWalk, LINE, PTE, and node2vec (WSDM'18 most cited, Nomination Award of the 2020 WAIC Youth Outstanding Paper)
- A Matrix Chernoff Bound for Markov Chains and Its Application to Co-occurrence Matrices (NeurIPS'20)
- NetSMF: Large-Scale Network Embedding As Sparse Matrix Factorization (WWW '19); LightNE: A Lightweight Graph Processing System for Network Embedding (SIGMOD'21)

Contents

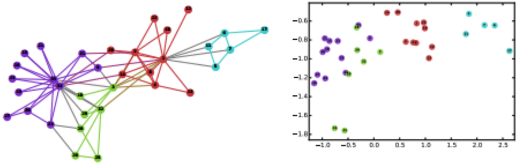
Motivation and Problem Formulation

Problem Formulation

Give a network $G = (V, E)$, aim to learn a function $f : V \to \mathbb{R}^p$ to capture neighborhood similarity and community membership.

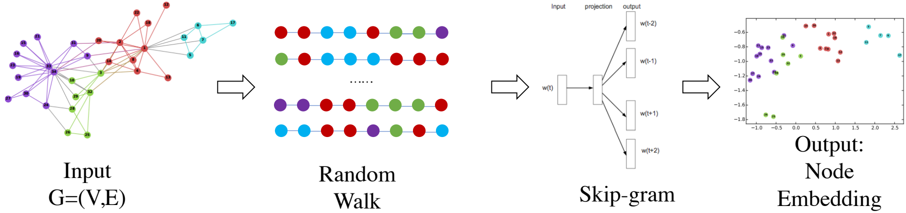
Applications:

- link prediction
- community detection
- label classification



(a) Input: Karate Graph (b) Output: Representation

DeepWalk and Three Questions We Want to Answer



Questions

1. Theory behind DeepWalk.
2. Sample complexity of DeepWalk?
3. Design better and more scalable network embedding systems.

Notations

Consider an undirected weighted graph $G = (V, E)$, where $|V| = n$ and $|E| = m$.

- Adjacency matrix $\boldsymbol{A} \in \mathbb{R}_+^{n \times n}$:

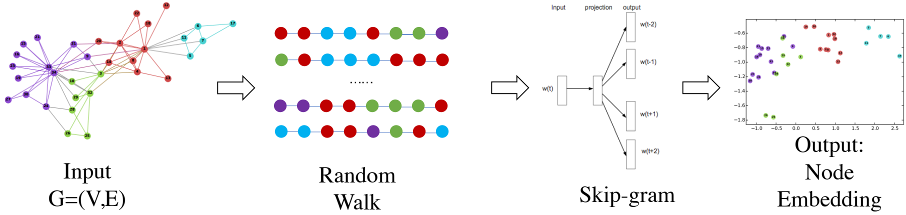
$$\boldsymbol{A}_{i,j} = \begin{cases} a_{i,j} > 0 & (i,j) \in E \\ 0 & (i,j) \notin E \end{cases}.$$

- Degree matrix $\boldsymbol{D} = \mathrm{diag}(d_1, \cdots, d_n)$, where d_i is the generalized degree of vertex i.
- Volume of the graph G: $\mathrm{vol}(G) = \sum_i \sum_j \boldsymbol{A}_{i,j}$.

Assumption

$G = (V, E)$ is connected, undirected, and not bipartite, which makes $P(w) = \frac{d_w}{\mathrm{vol}(G)}$ a unique stationary distribution.

DeepWalk: Random Walk on Graph + Skip-gram

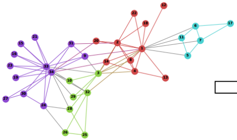


The objective of skip-gram model:

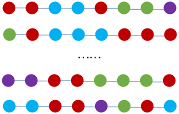
$$\mathcal{L}(\boldsymbol{X}, \boldsymbol{Y}) = |\mathcal{D}| \sum_{w} \sum_{c} \left(\frac{\#(w,c)}{|\mathcal{D}|} \log g(\boldsymbol{x}_w^\top \boldsymbol{y}_c) + b \frac{\#(w)}{|\mathcal{D}|} \frac{\#(c)}{|\mathcal{D}|} \log g(-\boldsymbol{x}_w^\top \boldsymbol{y}_c) \right).$$

$\#(w,c)$ indicates the number of times the pair (w,c) appears in a size-T sliding window.

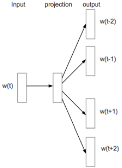
Skip-gram as PMI Matrix Factorization



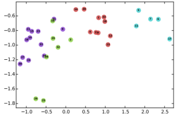
Input
G=(V,E)



Random
Walk



Skip-gram



Output:
Node
Embedding

Levy & Goldberg (NIPS 14)

$$\log\left(\frac{\#(w,c)\,|\mathcal{D}|}{b\#(w)\#(c)}\right)$$

b Number of negative samples

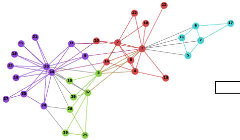
$\#(w,c)$ Co-occurrence of w and c

$\#(w)$ Occurrence of word w

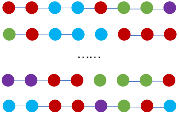
$|\mathcal{D}|$ Total number of word-context pairs

$\#(c)$ Occurrence of context c

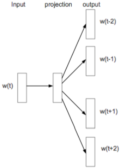
PMI Matrix of Random Walks on a Graph



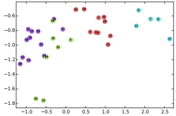
Input
G=(V,E)



Random
Walk



Skip-gram



Output:
Node
Embedding

Levy & Goldberg (NIPS 14)

$$\log\left(\frac{\#(w,c)\,|\mathcal{D}|}{b\#(w)\#(c)}\right)$$

?

b Number of negative samples

$\#(w,c)$ Co-occurrence of w and c

$|\mathcal{D}|$ Total number of word-context pairs

$\#(w)$ Occurrence of word w

$\#(c)$ Occurrence of context c

Our Results

Denote $\boldsymbol{P} = \boldsymbol{D}^{-1}\boldsymbol{A}$, which is the transition probability matrix. Also note that we want to analyze $\frac{\#(w,c)|\mathcal{D}|}{\#(w)\cdot\#(c)} = \frac{\frac{\#(w,c)}{|\mathcal{D}|}}{\frac{\#(w)}{|\mathcal{D}|}\cdot\frac{\#(c)}{|\mathcal{D}|}}$

Theorem

When the length of random walk $L \to \infty$, we have

$$\frac{\#(w,c)}{|\mathcal{D}|} \xrightarrow{p} \frac{1}{2T}\sum_{r=1}^{T}\left(\frac{d_w}{\mathrm{vol}(G)}\left(\boldsymbol{P}^r\right)_{w,c} + \frac{d_c}{\mathrm{vol}(G)}\left(\boldsymbol{P}^r\right)_{c,w}\right).$$

Consequently, $\frac{\#(w)}{|\mathcal{D}|} \xrightarrow{p} \frac{d_w}{\mathrm{vol}(G)}$ and $\frac{\#(c)}{|\mathcal{D}|} \xrightarrow{p} \frac{d_c}{\mathrm{vol}(G)}$, as $L \to \infty$.

Theorem

For DeepWalk, when the length of random walk $L \to \infty$,

$$\frac{\#(w,c)\,|\mathcal{D}|}{\#(w)\cdot\#(c)} \xrightarrow{p} \frac{\mathrm{vol}(G)}{2T}\left(\frac{1}{d_c}\sum_{r=1}^{T}\left(\boldsymbol{P}^r\right)_{w,c} + \frac{1}{d_w}\sum_{r=1}^{T}\left(\boldsymbol{P}^r\right)_{c,w}\right).$$

Our Results

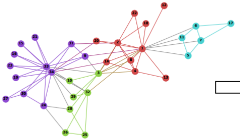
Theorem

DeepWalk is asymptotically and implicitly factorizing

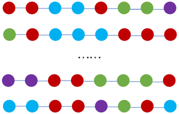
$$\log^{\circ}\left(\frac{\mathrm{vol}(G)}{b}\left(\frac{1}{T}\sum_{r=1}^{T}\left(\boldsymbol{D}^{-1}\boldsymbol{A}\right)^{r}\right)\boldsymbol{D}^{-1}\right).$$

Detailed proof

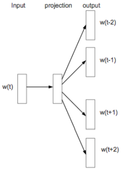
DeepWalk as Matrix Factorization



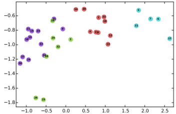
Input
G=(V,E)



Random
Walk



Skip-gram



Output:
Node
Embedding

Levy & Goldberg (NIPS 14)

$$\log\left(\frac{\mathrm{vol}(G)}{b}\left(\frac{1}{T}\sum_{r=1}^{T}\left(\boldsymbol{D}^{-1}\boldsymbol{A}\right)^{r}\right)\boldsymbol{D}^{-1}\right) \Longleftarrow \log\left(\frac{\#(w,c)\,|\mathcal{D}|}{b\#(w)\#(c)}\right)$$

$\boldsymbol{A}$ Adjacency matrix $\quad \mathrm{vol}(G) = \sum_i \sum_j \boldsymbol{A}_{i,j}$

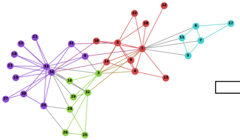
$\boldsymbol{D}$ Degree matrix $\quad b$ Number of negative samples

Summary

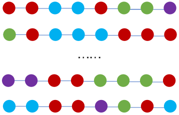
Table 1: The matrices that are implicitly approximated and factorized by DeepWalk, LINE, PTE, and node2vec.

Algorithm	Matrix
DeepWalk	$\log^\circ\left(\mathrm{vol}(G)\left(\frac{1}{T}\sum_{r=1}^{T}(\boldsymbol{D}^{-1}\boldsymbol{A})^r\right)\boldsymbol{D}^{-1}\right)-\log b$
LINE	$\log^\circ\left(\mathrm{vol}(G)\boldsymbol{D}^{-1}\boldsymbol{A}\boldsymbol{D}^{-1}\right)-\log b$
PTE	$\log^\circ\left(\begin{bmatrix}\alpha\,\mathrm{vol}(G_{\mathsf{ww}})(\boldsymbol{D}_{\mathsf{row}}^{\mathsf{ww}})^{-1}\boldsymbol{A}_{\mathsf{ww}}(\boldsymbol{D}_{\mathsf{col}}^{\mathsf{ww}})^{-1}\\ \beta\,\mathrm{vol}(G_{\mathsf{dw}})(\boldsymbol{D}_{\mathsf{row}}^{\mathsf{dw}})^{-1}\boldsymbol{A}_{\mathsf{dw}}(\boldsymbol{D}_{\mathsf{col}}^{\mathsf{dw}})^{-1}\\ \gamma\,\mathrm{vol}(G_{\mathsf{lw}})(\boldsymbol{D}_{\mathsf{row}}^{\mathsf{lw}})^{-1}\boldsymbol{A}_{\mathsf{lw}}(\boldsymbol{D}_{\mathsf{col}}^{\mathsf{lw}})^{-1}\end{bmatrix}\right)-\log b$
node2vec	$\log^\circ\left(\frac{\frac{1}{2T}\sum_{r=1}^{T}\left(\sum_u \boldsymbol{X}_{w,u}\underline{\boldsymbol{P}}_{c,w,u}^{r}+\sum_u \boldsymbol{X}_{c,u}\underline{\boldsymbol{P}}_{w,c,u}^{r}\right)}{\left(\sum_u \boldsymbol{X}_{w,u}\right)\left(\sum_u \boldsymbol{X}_{c,u}\right)}\right)-\log b$

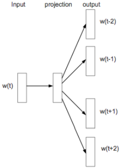
Contents



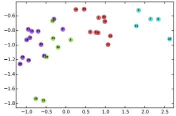
Input
G=(V,E)



Random
Walk



Skip-gram



Output:
Node
Embedding

Levy & Goldberg (NIPS 14)

$$\log\left(\frac{\mathrm{vol}(G)}{b}\left(\frac{1}{T}\sum_{r=1}^{T}\left(\boldsymbol{D}^{-1}\boldsymbol{A}\right)^r\right)\boldsymbol{D}^{-1}\right) \Longleftarrow \log\left(\frac{\#(w,c)\,|\mathcal{D}|}{b\#(w)\#(c)}\right)$$

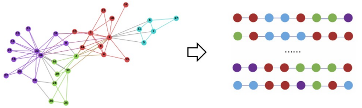
Question

- NetMF studies the convergence of co-occurrence matrices of random walk on graphs in the limit $(L \to \infty)$.
- How about the convergence rate?

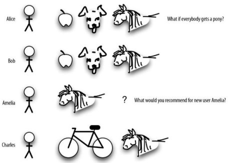
Applications of Co-occurrence Matrices

counts	I	like	enjoy	deep	learning	NLP	flying	.
I	0	2	1	0	0	0	0	0
like	2	0	0	1	0	1	0	0
enjoy	1	0	0	0	0	0	1	0
deep	0	1	0	0	1	0	0	0
learning	0	0	0	1	0	0	0	1
NLP	0	1	0	0	0	0	0	1
flying	0	0	1	0	0	0	0	1
.	0	0	0	0	1	1	1	0

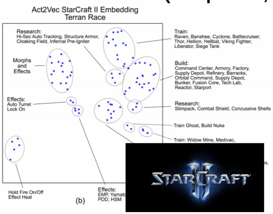
NLP
(LDA, Word2vec, Glove)



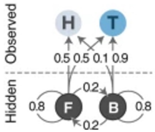
Graph Learning
(DeepWalk, node2vec, metapath2vec)



Recommendation System
(Pin2Vec, Item2vec)



Reinforcement Learning
(Act2Vec)



Hidden Markov Models
(Emission Co-occurrence)

Question:

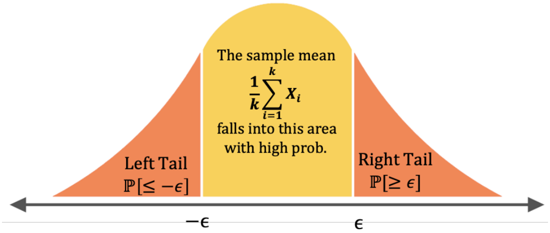
What is the #samples required to estimate a good co-occurrence matrix?

Chernoff Bound

Theorem

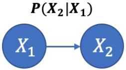
If $X_1, \cdots, X_k$ are k independent zero-mean scalar-valued random variables with $|X_i| \leq 1$. Then for $\epsilon \in (0,1)$:

$$\mathbb{P}\left[\left|\frac{1}{k}\sum_{i=1}^{k} X_i\right| \geq \epsilon\right] \leq 2\exp\left(-k\epsilon^2/4\right).$$

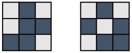


Our Matrix Chernoff Bound for Markov Chains

~~Independence~~
Markov Dependence



~~Scalar-valued Random Variables~~
Matrix-valued Random Variables

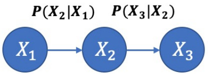


$f(X_1)$ $f(X_2)$

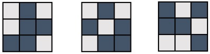
Sample Mean Matrix $\frac{1}{2}(f(X_1) + f(X_2))$

Our Matrix Chernoff Bound for Markov Chains

~~Independence~~
Markov Dependence



~~Scalar-valued Random Variables~~
Matrix-valued Random Variables



$f(X_1)$ $f(X_2)$ $f(X_3)$

Sample Mean Matrix

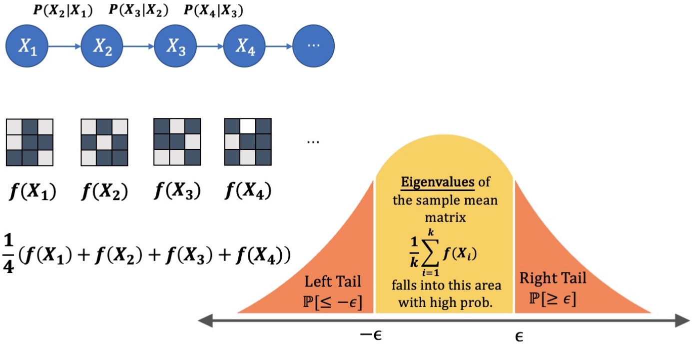
$$\frac{1}{3}(f(X_1) + f(X_2) + f(X_3))$$

Our Matrix Chernoff Bound for Markov Chains

~~Independence~~
Markov Dependence

~~Scalar-valued Random Variables~~
Matrix-valued Random Variables

Sample Mean Matrix



Comparison to Previous Results

We want bound the tail probability of extreme eigenvalues:

$$\mathbb{P}\left[\lambda_{\min}\left(\frac{1}{k}\sum_{i=1}^{k} f(X_i)\right)\right] \text{ and } \mathbb{P}\left[\lambda_{\max}\left(\frac{1}{k}\sum_{i=1}^{k} f(X_i)\right)\right].$$

Comparison	X	f	Tail Prob.
Chernoff'52	i.i.d scalars	identity	$\exp(-\Omega(k\epsilon^2))$
Tropp'12	i.i.d $d \times d$ matrices	identity	$d \exp(-\Omega(k\epsilon^2))$
CLLM'12	random walk on a regular Markov chain with spectral expansion λ	$[N] \to \boldsymbol{C}$	$\exp(-\Omega(k(1-\lambda)\epsilon^2))$
GLSS'18	random walk on an undirected regular graph with 2nd eigenvalue λ	$[N] \to \boldsymbol{C}^{d\times d}$	$d \exp(-\Omega(k(1-\lambda)\epsilon^2))$
Ours	random walk on a regular Markov chain with spectral expansion λ	$[N] \to \boldsymbol{C}^{d\times d}$	$d \exp(-\Omega(k(1-\lambda)\epsilon^2))$

Matrix Chernoff Bound for Ergodic Markov Chains

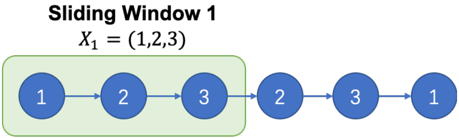
Theorem

Let $\boldsymbol{P}$ be an ergodic Markov chain with state space $[N]$, stationary distribution $\boldsymbol{\pi}$ and spectral norm λ. Let $f : [N] \to \mathbb{R}^{d \times d}$ be a function such that (1) $\forall v \in [N]$, $f(v)$ is symmetric and $\|f(v)\|_2 \leq 1$; (2) $\sum_{v \in [N]} \pi_v f(v) = 0$. Let $(v_1, \cdots, v_k)$ denote a k-step random walk on $\boldsymbol{P}$ starting from a distribution ϕ on $[N]$. Then given $\epsilon \in (0, 1)$,

$$\mathbb{P}\left[\lambda_{\max}\left(\frac{1}{k}\sum_{j=1}^{k} f(v_j)\right) \geq \epsilon\right] \leq \|\boldsymbol{\phi}\|_{\boldsymbol{\pi}} d^2 \exp\left(-(\epsilon^2(1-\lambda)k/72)\right)$$

$$\mathbb{P}\left[\lambda_{\min}\left(\frac{1}{k}\sum_{j=1}^{k} f(v_j)\right) \leq -\epsilon\right] \leq \|\boldsymbol{\phi}\|_{\boldsymbol{\pi}} d^2 \exp\left(-(\epsilon^2(1-\lambda)k/72)\right).$$

Co-occurrence Matrix of Sequential Data

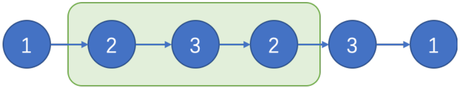


$$\boldsymbol{C} = \frac{1}{4}\begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}$$

Co-occurrence Matrix of Sequential Data

Sliding Window 2

$X_2 = (2,3,2)$

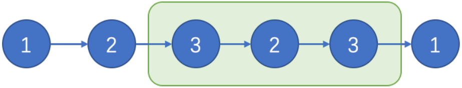


$$\boldsymbol{C} = \frac{1}{2}\left[\frac{1}{4}\begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix} + \frac{1}{4}\begin{pmatrix} 0 & 0 & 0 \\ 0 & 2 & 1 \\ 0 & 1 & 0 \end{pmatrix}\right]$$

Co-occurrence Matrix of Sequential Data

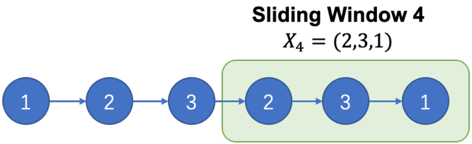
Sliding Window 3

$X_3 = (3,2,3)$



$$\boldsymbol{C} = \frac{1}{3}\left[\frac{1}{4}\begin{pmatrix}0 & 1 & 1\\ 1 & 0 & 0\\ 1 & 0 & 0\end{pmatrix} + \frac{1}{4}\begin{pmatrix}0 & 0 & 0\\ 0 & 2 & 1\\ 0 & 1 & 0\end{pmatrix} + \frac{1}{4}\begin{pmatrix}0 & 0 & 0\\ 0 & 0 & 1\\ 0 & 1 & 2\end{pmatrix}\right]$$

Co-occurrence Matrix of Sequential Data



$$\boldsymbol{C} = \frac{1}{4}\left[\frac{1}{4}\begin{pmatrix}0 & 1 & 1\\ 1 & 0 & 0\\ 1 & 0 & 0\end{pmatrix} + \frac{1}{4}\begin{pmatrix}0 & 0 & 0\\ 0 & 2 & 1\\ 0 & 1 & 0\end{pmatrix} + \frac{1}{4}\begin{pmatrix}0 & 0 & 0\\ 0 & 0 & 1\\ 0 & 1 & 2\end{pmatrix} + \frac{1}{4}\begin{pmatrix}0 & 1 & 0\\ 1 & 0 & 1\\ 0 & 1 & 0\end{pmatrix}\right]$$
$$= \frac{1}{4}\left(\boldsymbol{f}(\boldsymbol{X}_1) + \boldsymbol{f}(\boldsymbol{X}_2) + \boldsymbol{f}(\boldsymbol{X}_3) + \boldsymbol{f}(\boldsymbol{X}_4)\right)$$

Observations

- Co-occurrence matrix can be written as sample mean $\boldsymbol{C} = \frac{1}{k}\sum_{i=1}^{k} f(X_i)$.
- If the input sequence is a Markov Chain, then sliding window X_i is also a Markov Chain.

Convergence Rate of Co-occurrence Matrices

Theorem (Convergence Rate of Co-occurrence Matrices)

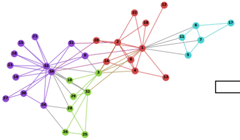
Let $\boldsymbol{P}$ be a regular Markov chain with state space $[n]$, stationary distribution $\boldsymbol{\pi}$ and mixing time τ. Let $(v_1, \cdots, v_L)$ denote a L-step random walk on $\boldsymbol{P}$ starting from a distribution $\boldsymbol{\phi}$ on $[n]$. Given step weight coefficients $(\alpha_1, \cdots, \alpha_T)$ s.t. $\sum_{r=1}^{T} |\alpha_r| = 1$, and $\epsilon \in (0,1)$, the probability that the co-occurrence matrix $\boldsymbol{C}$ deviates from its asymptotic expectation $\mathbb{AE}[\boldsymbol{C}]$ (in 2-norm) is bounded by:

$$\mathbb{P}\left[\left\|\boldsymbol{C} - \mathbb{AE}[\boldsymbol{C}]\right\|_2 \geq \epsilon\right] \leq 2\left(\tau + T\right) \left\|\boldsymbol{\phi}\right\|_{\boldsymbol{\pi}} n^2 \exp\left(-\frac{\epsilon^2 (L-T)}{576\left(\tau + T\right)}\right).$$

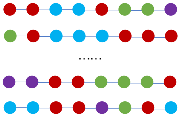
Roughly, one needs $L = O\left(\tau(\log n + \log \tau)/\epsilon^2\right)$ samples to guarantee good estimation to the co-occurrence matrix.

Contents

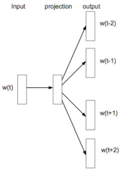
Q3: How to Design Better NE Systems



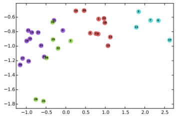
Input
G=(V,E)



Random
Walk



Skip-gram



Output:
Node
Embedding

Levy & Goldberg (NIPS 14)

$$\log\left(\frac{\#(w,c)\,|\mathcal{D}|}{b\#(w)\#(c)}\right)$$

$$\log\left(\frac{\mathrm{vol}(G)}{b}\left(\frac{1}{T}\sum_{r=1}^{T}\left(\boldsymbol{D}^{-1}\boldsymbol{A}\right)^{r}\right)\boldsymbol{D}^{-1}\right)$$

$\boldsymbol{A}$ Adjacency matrix $\quad \mathrm{vol}(G)=\sum_i\sum_j \boldsymbol{A}_{i,j}$

$\boldsymbol{D}$ Degree matrix $\quad b$ Number of negative samples

- DeepWalk: SGD on $O(|V|d)$ parameters.
- NetMF: the matrix is dense.

Random-walk Matrix Polynomial Sparsification

Theorem ([CCL+15])

For random-walk matrix polynomial $\boldsymbol{L} = \boldsymbol{D} - \sum_{r=1}^{T} \alpha_r \boldsymbol{D} \left(\boldsymbol{D}^{-1}\boldsymbol{A}\right)^r$*, where* $\sum_{r=1}^{T} \alpha_r = 1$ *and* α_r *non-negative, one can construct, in time* $O(T^2 m \epsilon^{-2} \log^2 n)$*, a* $(1+\epsilon)$*-spectral sparsifier,* $\widetilde{\boldsymbol{L}}$*, with* $O(n \log n \epsilon^{-2})$ *non-zeros. For unweighted graphs, the time complexity can be reduced to* $O(T^2 m \epsilon^{-2} \log n)$*.*

Definition

Suppose $G = (V, E, \boldsymbol{A})$ and $\widetilde{G} = (V, \widetilde{E}, \widetilde{\boldsymbol{A}})$ are two weighted undirected networks. Let $\boldsymbol{L} = \boldsymbol{D}_G - \boldsymbol{A}$ and $\widetilde{\boldsymbol{L}} = \boldsymbol{D}_{\widetilde{G}} - \widetilde{\boldsymbol{A}}$ be their Laplacian matrices, respectively. We define G and $\widetilde{G}$ are $(1+\epsilon)$-spectrally similar if

$$\forall \boldsymbol{x} \in \mathbb{R}^n, (1-\epsilon) \cdot \boldsymbol{x}^\top \widetilde{\boldsymbol{L}} \boldsymbol{x} \le \boldsymbol{x}^\top \boldsymbol{L} \boldsymbol{x} \le (1+\epsilon) \cdot \boldsymbol{x}^\top \widetilde{\boldsymbol{L}} \boldsymbol{x}.$$

NetSMF

Setting $\alpha_1 = \cdots = \alpha_T = \frac{1}{T}$ in $\boldsymbol{L} = \boldsymbol{D} - \sum_{r=1}^{T} \alpha_r \boldsymbol{D} \left(\boldsymbol{D}^{-1}\boldsymbol{A}\right)^r$, we can observe the tight connection between random walk matrix polynomial and NetMF:

$$
\begin{aligned}
&\log^\circ \left(\frac{\mathrm{vol}(G)}{b} \left(\frac{1}{T} \sum_{r=1}^{T} \left(\boldsymbol{D}^{-1}\boldsymbol{A}\right)^r \right) \boldsymbol{D}^{-1} \right) \\
=& \log^\circ \left(\frac{\mathrm{vol}(G)}{b} \boldsymbol{D}^{-1} (\boldsymbol{D} - \boldsymbol{L}) \boldsymbol{D}^{-1} \right) \\
\approx& \log^\circ \left(\frac{\mathrm{vol}(G)}{b} \boldsymbol{D}^{-1} (\boldsymbol{D} - \widetilde{\boldsymbol{L}}) \boldsymbol{D}^{-1} \right)
\end{aligned}
$$

Idea

Rewrite the NetMF matrix in terms of random walk matrix polynomial $\boldsymbol{L}$, and further approximate $\boldsymbol{L}$ with $\widetilde{\boldsymbol{L}}$.

NetSMF—Algorithm

The proposed NetSMF algorithm consists of three steps:

- Construct a random walk matrix polynomial sparsifier, $\widetilde{\boldsymbol{L}}$, by calling algorithm proposed in [CCL+15].
- Construct a NetMF matrix sparsifier.

$$\mathrm{trunc\_log}^{\circ}\left(\frac{\mathrm{vol}(G)}{b}\boldsymbol{D}^{-1}(\boldsymbol{D}-\widetilde{\boldsymbol{L}})\boldsymbol{D}^{-1}\right)$$

- Truncated randomized singular value decomposition.

Detailed Algorithm

Experiments

Label Classification:

- BlogCatelog, PPI, Flickr, YouTube, OAG.
- Logistic Regression
- DeepWalk, LINE, node2vec, NetMF[1] $(T = 10)$, NetSMF[2] $(T = 10)$.

Table 2: Statistics of Datasets.

Dataset	**BlogCatalog**	**PPI**	**Flickr**	**YouTube**	**OAG**
$\lvert V\rvert$	10,312	3,890	80,513	1,138,499	67,768,244
$\lvert E\rvert$	333,983	76,584	5,899,882	2,990,443	895,368,962
#Labels	39	50	195	47	19

[1] Qiu et al. Network Embedding as Matrix Factorization: Unifying DeepWalk, LINE, PTE, and node2vec. WSDM'18

[2] Qiu et al. NetSMF: Large-Scale Network Embedding as Sparse Matrix Factorization. WWW'19

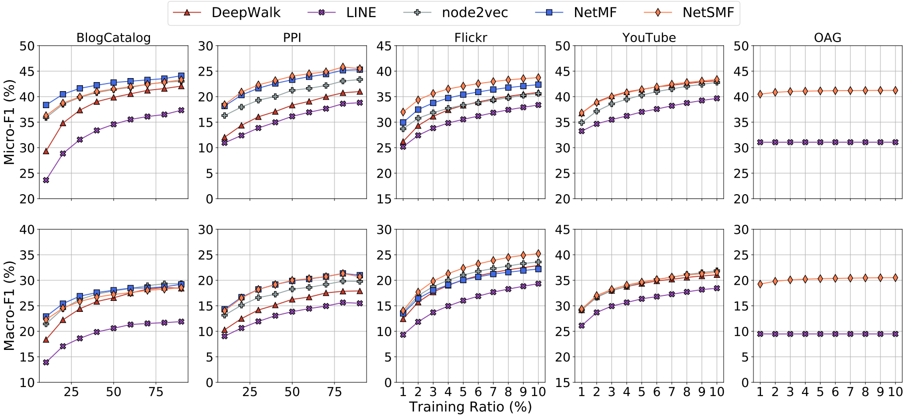
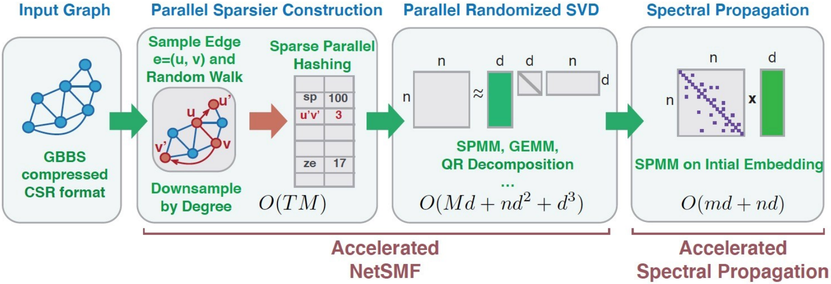


Figure 1: Predictive performance on varying the ratio of training data. The x-axis represents the ratio of labeled data (%), and the y-axis in the top and bottom rows denote the Micro-F1 and Macro-F1 scores respectively.

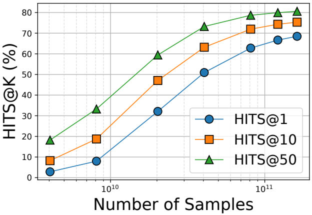
Q3: Even More Scalable?



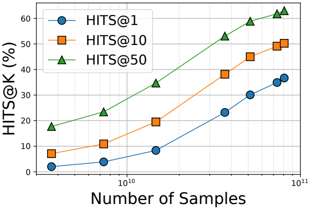
- Scalable: Embed graphs with 1B edges within 1.5 hours.
- Lightweight: Occupy hardware costs below 100 dollars measured by cloud rent to process 1B to 100B edges.
- Accurate: Achieve the highest accuracy in downstream tasks under the same time budget and similar resources.

LightNE on Very Large Graphs!

	ClueWeb	Hyperlink2014
$\lvert V\rvert$	978,408,098	1,724,573,718
$\lvert E\rvert$	74,744,358,622	124,141,874,032



(a) ClueWeb-Sym



(b) Hyperlink2014-Sym

Figure 2: HITS@K ($K = 1, 10, 50$) of LightNE w.r.t. the number of samples.

None of the existing network embedding systems can handle such large graphs in a single machine!

Thanks.

www.jiezhongqiu.com

- Network Embedding as Matrix Factorization: Unifying DeepWalk, LINE, PTE, and node2vec (WSDM'18 most cited)
- A Matrix Chernoff Bound for Markov Chains and Its Application to Co-occurrence Matrices (NeurIPS'20)
- NetSMF: Large-Scale Network Embedding As Sparse Matrix Factorization (WWW '19)
- LightNE: A Lightweight Graph Processing System for Network Embedding (SIGMOD '21)

Q&A

Contents

References I

Dehua Cheng, Yu Cheng, Yan Liu, Richard Peng, and Shang-Hua Teng, *Spectral sparsification of random-walk matrix polynomials*, arXiv preprint arXiv:1502.03496 (2015).