Fast Network Embedding Enhancement via High Order Proximity Approximation

报告人:王昕毅

1.Introduction

In this paper:

1)we summarize most existing NRL methods into a unified two-step framework, including proximity matrix construction and dimension reduction

2)We propose Network Embedding Update(NEU) algorithm which implicitly approximates higher order proximities with theoretical approximation bound and can be applied on any NRL methods to enhance their performances. NEU can make a consistent and significant improvement over a number of NRL methods with almost negligible running time on all three publicly available datasets. We focus on the most general case where NRL methods use only network topology in this paper. In this section, we focus on the first step and study how to define a good proximity matrix for NRL. The study of different dimension reduction methods, e.g. SVD decomposition, will be left as future work.

Why we come up with this work:

Modeling higher order and accurate proximity matrix can improve the quality of network representation. However, Accurate computation of high order proximity matrix is not feasible for large-scale networks.

Define adjacency matrix:

$\widetilde{A} \in \mathbb{R}^{|V|\times|V|}$ as $\widetilde{A}_{ij} = 1$ if $(v_i, v_j) \in E$ and $\widetilde{A}_{ij} = 0$ otherwise.

Define Diagonal matrix:

$D \in \mathbb{R}^{|V|\times|V|}$ where $D_{ii} = d_i$ represents the degree of vertex v_i.

A is normalized adjacency matrix where the summation of each row equals to 1:

$$A = D^{-1}\widetilde{A}$$

K-order Proximity:

the second-order proximity can be characterized by the number of common neighbors between vertices. As an alternative view, the second-order proximity between vi and vj can also be modeled by the probability that a 2-step random walk from vi reaches vj. In the probabilistic setting based on random walk, we can easily generalize it to k-order proximity. A is the transition probability matrix of a single step random walk

k-order proximity matrix:

$$A^k = \underbrace{A \cdot A \ldots A}_{k},$$

2.NRL Framework

We summarize NRL methods as a two-step framework:

Step 1: Proximity Matrix Construction. proximity matrix M encodes the information of k-order proximity matrix where k = 1,2...,K. $M = \frac{1}{K}A + \frac{1}{K}A^2 \cdots + \frac{1}{K}A^K$ Stands for an average combination of k-order proximity matrix for k = 1,2...,K. The proximity matrix M is usually represented by a polynomial of normalized adjacency matrix A of degree K.

Step 2: Dimension Reduction. Find network embedding matrix R and context embedding C so that the product $R \cdot C^T$ approximates proximity matrix M. Here different algorithms may employ different distance functions to minimize the distance between M and $R \cdot C^T$. For example, we can naturally use the norm of matrix $M - R \cdot C^T$ to measure the distance and minimize it.

Example：

Deep walk generates random walks and employs Skip-gram model for representation learning. Deep Walk implicitly factorizes a matrix M into the product of $R \cdot C^T$, where

$$M = \log \frac{A + A^2 + \cdots + A^w}{w}$$

w is the window size used in Skip-gram model. Matrix M characterizes the average of the first-order, second-order, ..., w-order proximities. Deep Walk algorithm approximates high-order proximity by Monte Carlo sampling based on random walk generation without calculating the k-order proximity matrix directly.

GraRep

accurately calculates k-order proximity matrix Ak for k = 1,2...,K, computes a specific representation for each k, and concatenates these embeddings.

Specifically, GraRep reduces the dimension of k-order proximity matrix A^k for k-order representation via SVD decomposition

3 Observation and Problem Formalization

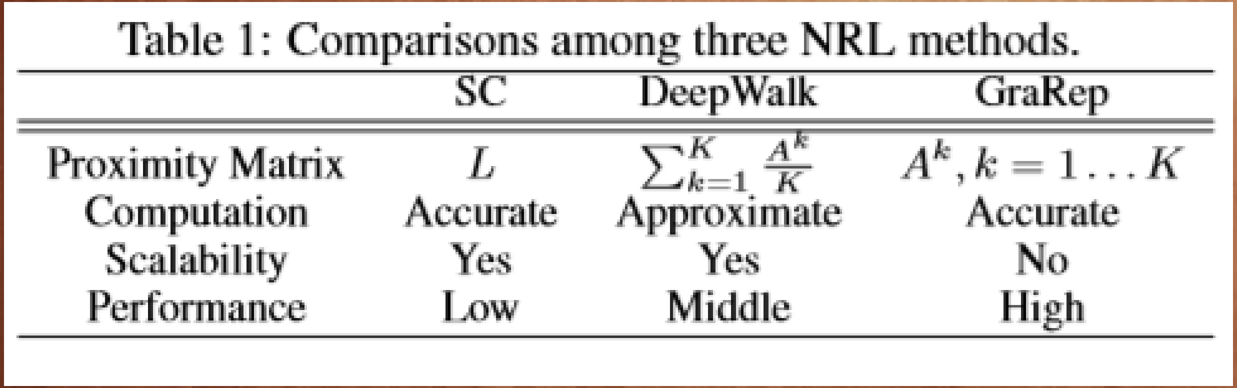


Table 1: Comparisons among three NRL methods.

	SC	DeepWalk	GraRep
Proximity Matrix	L	$\sum_{k=1}^{K} \frac{A^k}{K}$	$A^k, k = 1 \dots K$
Computation	Accurate	Approximate	Accurate
Scalability	Yes	Yes	No
Performance	Low	Middle	High

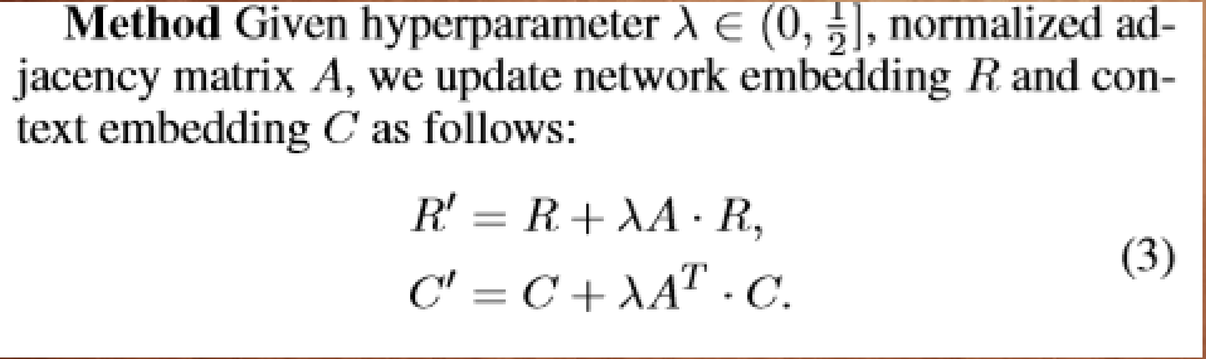
Observation 1 Modeling higher order and accurate proximity matrix can improve the quality of network representation. In other words, NRL could benefit if we explore a polynomial proximity matrix f(A) of a higher degree.

Observation 2 Accurate computation of high order proximity matrix is not feasible for large-scale networks. The major drawback of GraRep is the computation complexity of calculating the accurate k-order proximity matrix. In fact, the computation of high order proximity matrix takes $O(|V|^2)$ time and the time complexity of SVD decomposition also increases as k-order proximity matrix gets dense when k grows.

Problem Formalization:

Assume that we have normalized adjacency matrix A as the first-order proximity matrix, network embedding R and context embedding C. Suppose that the embeddings R and C are learned by the above NRL framework which indicates that the product $R \cdot C^T$ approximates a polynomial proximity matrix f(A) of degree K. Our goal is to learn a better representation R′ and C′ which approximates a polynomial proximity matrix g(A) with higher degree than f(A). Also, the algorithm should be efficient in the linear time of |V|.

4 Approximation Algorithm

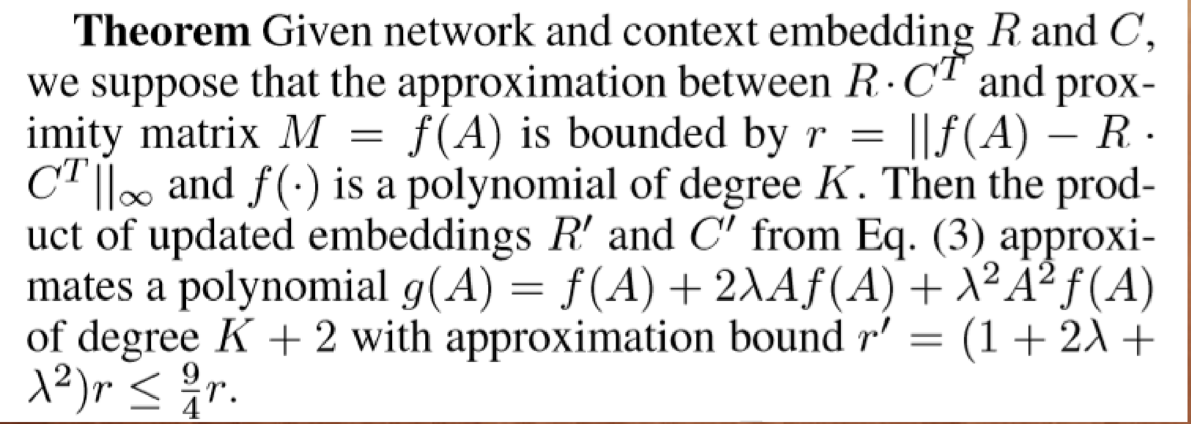


Method Given hyperparameter $\lambda \in (0, \frac{1}{2}]$, normalized adjacency matrix A, we update network embedding R and context embedding C as follows:

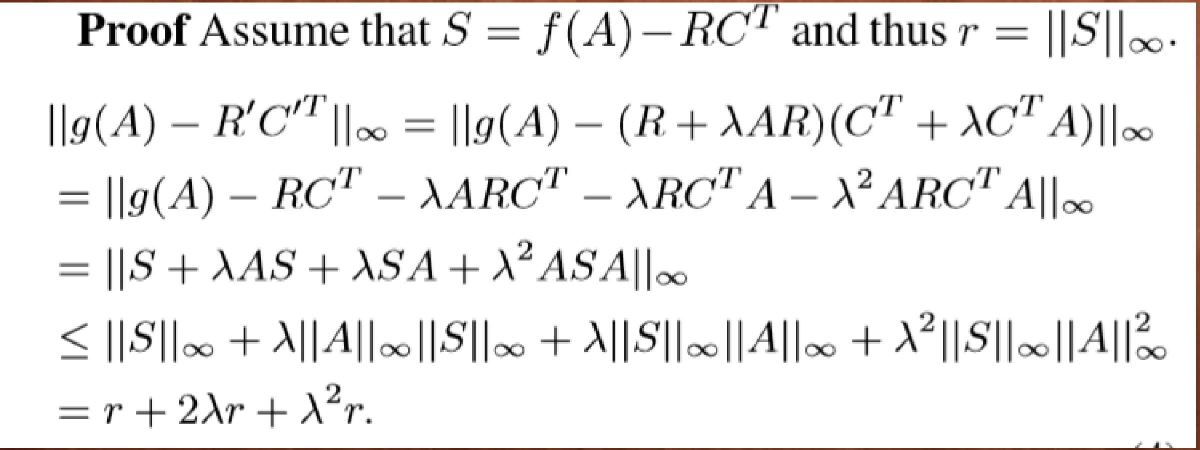
$$\begin{aligned} R' &= R + \lambda A \cdot R, \\ C' &= C + \lambda A^T \cdot C. \end{aligned} \tag{3}$$

The time complexity of computing $A \cdot R$ and $A^T \cdot C$ is $O(|V|d)$ because matrix A is sparse and has $O(|V|)$ nonzero entries. Thus the overall time complexity of one iteration of operation (3) is $O(|V|d)$.

Recall that product of previous embedding R and C approximates polynomial proximity matrix $f(A)$ of degree K. Now we prove that the algorithm can learn better embeddings R' and C' where the product $R' \cdot C'^T$ approximates a polynomial proximity matrix $g(A)$ of degree $K + 2$ bounded by matrix infinite norm.



Theorem Given network and context embedding R and C, we suppose that the approximation between $R \cdot C^T$ and proximity matrix $M = f(A)$ is bounded by $r = ||f(A) - R \cdot C^T||_\infty$ and $f(\cdot)$ is a polynomial of degree K. Then the product of updated embeddings R' and C' from Eq. (3) approximates a polynomial $g(A) = f(A) + 2\lambda A f(A) + \lambda^2 A^2 f(A)$ of degree $K + 2$ with approximation bound $r' = (1 + 2\lambda + \lambda^2)r \leq \frac{9}{4}r$.



Proof Assume that $S = f(A) - RC^T$ and thus $r = ||S||_\infty$.

$$
\begin{aligned}
&||g(A) - R'C'^T||_\infty = ||g(A) - (R + \lambda AR)(C^T + \lambda C^T A)||_\infty \\
&= ||g(A) - RC^T - \lambda ARC^T - \lambda RC^T A - \lambda^2 ARC^T A||_\infty \\
&= ||S + \lambda AS + \lambda SA + \lambda^2 ASA||_\infty \\
&\leq ||S||_\infty + \lambda||A||_\infty||S||_\infty + \lambda||S||_\infty||A||_\infty + \lambda^2||S||_\infty||A||_\infty^2 \\
&= r + 2\lambda r + \lambda^2 r.
\end{aligned}
$$

In our experimental settings, we assume that the weight of lower order proximities should be larger than higher order proximities because they are more directly related to the original network. Therefore, given $g(A) = f(A) + 2\lambda A f(A) + \lambda^2 A^2 f(A)$, we have $1 \geq 2\lambda \geq \lambda^2 > 0$ which indicates that $\lambda \in (0, \frac{1}{2}]$. The proof indicates that the updated embedding can implicitly approximate a polynomial $g(A)$ of 2 more degrees within $\frac{9}{4}$ times matrix infinite norm of previous embeddings. QED.

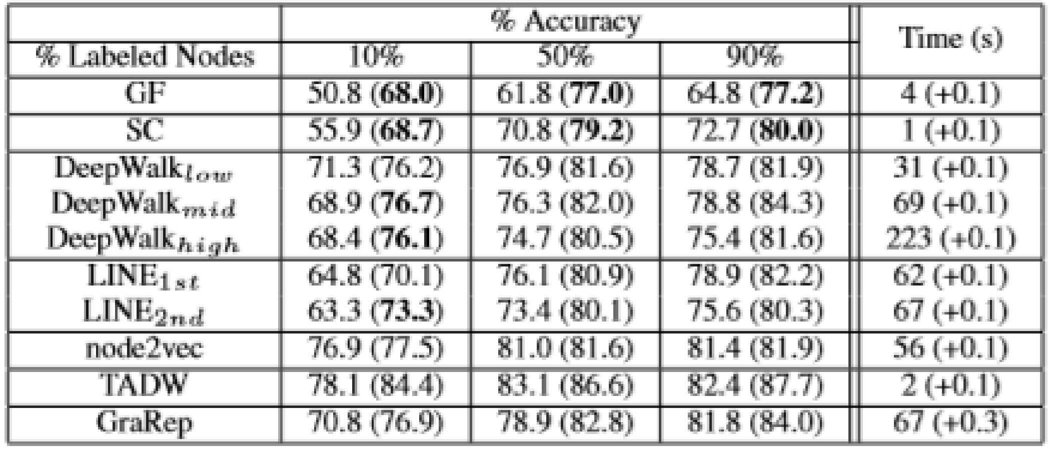
Algorithm The update Eq. (3) can be further generalized in two directions. First we can update embeddings R and C according to Eq. (5):

$$\begin{aligned} R' &= R + \lambda_1 A \cdot R + \lambda_2 A \cdot (A \cdot R), \\ C' &= C + \lambda_1 A^T \cdot C + \lambda_2 A^T \cdot (A^T \cdot C). \end{aligned} \tag{5}$$

The time complexity is still O(|V|d) but Eq. (5) can obtain higher proximity matrix approximation than Eq. (3) in one iteration. Another direction is that the update equation can be processed for T rounds to obtain higher proximity approximation. However, the approximation bound would grow exponentially as the number of rounds T grows and thus the update cannot be done infinitely.

5.Experiments

Table 2: Classification results on Cora dataset.



	% Accuracy			Time (s)
% Labeled Nodes	10%	50%	90%	
GF	50.8 (**68.0**)	61.8 (**77.0**)	64.8 (**77.2**)	4 (+0.1)
SC	55.9 (**68.7**)	70.8 (**79.2**)	72.7 (**80.0**)	1 (+0.1)
DeepWalk$_{low}$	71.3 (76.2)	76.9 (81.6)	78.7 (81.9)	31 (+0.1)
DeepWalk$_{mid}$	68.9 (**76.7**)	76.3 (82.0)	78.8 (84.3)	69 (+0.1)
DeepWalk$_{high}$	68.4 (**76.1**)	74.7 (80.5)	75.4 (81.6)	223 (+0.1)
LINE$_{1st}$	64.8 (70.1)	76.1 (80.9)	78.9 (82.2)	62 (+0.1)
LINE$_{2nd}$	63.3 (**73.3**)	73.4 (80.1)	75.6 (80.3)	67 (+0.1)
node2vec	76.9 (77.5)	81.0 (81.6)	81.4 (81.9)	56 (+0.1)
TADW	78.1 (84.4)	83.1 (86.6)	82.4 (87.7)	2 (+0.1)
GraRep	70.8 (76.9)	78.9 (82.8)	81.8 (84.0)	67 (+0.3)

We have four main observations over the experimental results of two evaluation tasks:

1)NEU consistently and significantly improves the performance of various network embeddings using almost negligible running time on both evaluation tasks. The absolute improvements on Flickr are not as significant as that on Cora and Blog Catalog because Flickr dataset has an average degree of 147. But for Cora dataset where the average degree is 4, NEU has very significant improvement as high-order proximity plays an important role for sparse networks.

2) NEU facilitates NRL method to converge fast and stably. We can see that the performances of DeepWalklow+NEU and DeepWalkmid+NEU are comparable and even better than that of DeepWalkmid and DeepWalkhigh respectively and the former ones use much less time.

3) NEU also works for NRL algorithms which don’t follow our two-step NRL framework, i.e. node2vec. NEU doesn’t harm the performance of node2vec and even improve a little bit.

4) NEU can serve as a standard preprocessing step for evaluations of future NRL methods. In other words, network embeddings will be evaluated only after enhanced by NEU as NEU won’t increase time and space complexity at all.

Thank you