Network Embedding as Matrix Factorization: Unifying DeepWalk, LINE, PTE, and node2vec

Jiezhong Qiu

Tsinghua University

February 21, 2018

Joint work with Yuxiao Dong (MSR), Hao Ma (MSR), Jian Li (IIIS, Tsinghua), Kuansan Wang (MSR), Jie Tang (DCST, Tsinghua)
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

Figure 1: A toy example (Figure from DeepWalk).
History of Network Embedding

1973
- Fiedler Vector [Fiedler]
- Spectral Partitioning [Donath, Hoffman]

1996
- Image Segmentation [Shi & Malik]
- Spectral Clustering [Ng et al.]

2000
- A large body of literature
  - [Pothen et al.]
  - [Simon]
  - [Bolla]
  - [Hagen & Kahng]
  - [Hendrickson & Leland]
  - [Van Driessche & Roose]
  - [Barnard et al.]
  - [Spielman & Teng]
  - [Guattery & Miller]

2002
- word2vec (skip-gram) [Mikolov et al.]

2005
- Spectral Clustering v.s. Kernel k-means [Dhillon et al.]

2009
- SocDim [Tang & Liu]

2013
- Spectral Clustering v.s. Kernel k-means [Dhillon et al.]

2014
- DeepWalk [Perozzi et al.]

2015
- LINE & PTE [Tang et al.]

2016
- node2vec [Grover & Leskovec]

2017
- metapath2vec [Dong et al.]

2018
- A large body of literature

2019
- node2vec [Grover & Leskovec]
Contents

Preliminaries

Main Theoretic Results
   Notations
   DeepWalk (KDD’14)
   LINE (WWW’15)
   PTE (KDD’15)
   node2vec (KDD’16)

NetMF
   NetMF for a Small Window Size $T$
   NetMF for a Large Window Size $T$
Experiments
Consider an undirected weighted graph \( G = (V, E) \), where \( |V| = n \) and \( |E| = m \).

▶ Adjacency matrix \( A \in \mathbb{R}^{n \times n}_{+} \):
\[
A_{i,j} = \begin{cases} 
    a_{i,j} > 0 & (i, j) \in E \\
    0 & (i, j) \notin E 
\end{cases}
\]

▶ Degree matrix \( D = \text{diag}(d_1, \cdots, d_n) \), where \( d_i \) is the generalized degree of vertex \( i \).

▶ Volume of the graph \( G \): \( \text{vol}(G) = \sum_i \sum_j A_{i,j} \).

**Assumption**

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

Input
\( G = (V, E) \)

Random Walk

Skip-gram

Output:
Node Embedding
Algorithm 1: DeepWalk

1. for $n = 1, 2, \ldots, N$ do
2.    Pick $w_1^n$ according to a probability distribution $P(w_1)$;
3.    Generate a vertex sequence $(w_1^n, \ldots, w_L^n)$ of length $L$ by a random walk on network $G$;
4.    for $j = 1, 2, \ldots, L - T$ do
5.        for $r = 1, \ldots, T$ do
6.            Add vertex-context pair $(w_j^n, w_{j+r}^n)$ to multiset $D$;
7.            Add vertex-context pair $(w_{j+r}^n, w_j^n)$ to multiset $D$;
8.    Run SGNS on $D$ with $b$ negative samples.
DeepWalk — Roadmap

Input:
\[ G = (V, E) \]

Random Walk

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

- \#(w, c): Co-occurrence of w and c
- |\mathcal{D}|: Total number of word-context pairs
- b: Number of negative samples
- \#(w): Occurrence of word w
- \#(c): Occurrence of context c

Output:
Node Embedding

Levy & Goldberg (NIPS 14)
SGNS maintains a multiset $\mathcal{D}$ which counts the occurrence of each word-context pair $(w, c)$.

Objective:

$$\mathcal{L} = \sum_w \sum_c \left( \#(w, c) \log g \left( x_w^\top y_c \right) + \frac{b\#(w)\#(c)}{|\mathcal{D}|} \log g \left( -x_w^\top y_c \right) \right),$$

where $x_w, y_c \in \mathbb{R}^d$, $g$ is the sigmoid function, and $b$ is the number of negative samples for SGNS.

For sufficiently large dimensionality $d$, equivalent to factorizing PMI matrix (Levy & Goldberg, NIPS’14):

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

Input \( G=(V,E) \) \( \Downarrow \) Random Walk

\( \Downarrow \) Levy & Goldberg (NIPS 14)

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

Output: Node Embedding

\( b \) Number of negative samples
\( \#(w, c) \) Co-occurrence of w and c
\( \#(w) \) Occurrence of word w
\( \#(c) \) Occurrence of context c

Total number of word-context pairs

\(|D|\)
Question
Suppose the multiset $\mathcal{D}$ is constructed based on random walk on graph, can we interpret $\log \left( \frac{\#(w,c) |\mathcal{D}|}{b \#(w) \#(c)} \right)$ with graph theory terminologies?
Question
Suppose the multiset $\mathcal{D}$ is constructed based on random walk on graph, can we interpret $\log \left( \frac{\#(w,c)|\mathcal{D}|}{b\#(w)\#(c)} \right)$ with graph theory terminologies?

Challenge
We mix so many things together, i.e., direction and distance.
Question
Suppose the multiset $\mathcal{D}$ is constructed based on random walk on graph, can we interpret $\log \left( \frac{\#(w,c)|\mathcal{D}|}{b\#(w)\#(c)} \right)$ with graph theory terminologies?

Challenge
We mix so many things together, i.e., direction and distance.

Solution
Let’s distinguish them!
Partition the multiset $\mathcal{D}$ into several sub-multisets according to the way in which vertex and its context appear in a random walk sequence. More formally, for $r = 1, \cdots, T$, we define

$$
\mathcal{D}_r = \{(w, c) : (w, c) \in \mathcal{D}, w = w_j^n, c = w_{j+r}^n\},
$$

$$
\mathcal{D}_{\leftarrow r} = \{(w, c) : (w, c) \in \mathcal{D}, w = w_{j+r}^n, c = w_j^n\}.
$$
Some observations

Observation 1:

\[
\log \left( \frac{\#(w, c) |D|}{b \#(w) \cdot \#(c)} \right) = \log \left( \frac{\#(w, c)}{|D|} \cdot \frac{|D|}{b \#(w) \cdot \#(c)} \right)
\]

Observation 2:

\[
\frac{\#(w, c)}{|D|} = \frac{1}{2T} \sum_{r=1}^{T} \left( \frac{\#(w, c) \rightarrow}{|D \rightarrow|} + \frac{\#(w, c) \leftarrow}{|D \leftarrow|} \right).
\]

Sufficient to characterize \( \frac{\#(w, c) \rightarrow}{|D \rightarrow|} \) and \( \frac{\#(w, c) \leftarrow}{|D \leftarrow|} \).
DeepWalk — Theorems

**Theorem**
Denote \( P = D^{-1} A \), when the length of random walk \( L \to \infty \),

\[
\frac{\#(w, c) \rightarrow \hat{\tau}}{|D_{\rightarrow}|} \xrightarrow{p} \frac{d_w}{\text{vol}(G)} (P^r)_{w,c} \quad \text{and} \quad \frac{\#(w, c) \leftarrow \hat{\tau}}{|D_{\leftarrow}|} \xrightarrow{p} \frac{d_c}{\text{vol}(G)} (P^r)_{c,w}.
\]

**Theorem**
When the length of random walk \( L \to \infty \), we have

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

**Theorem**
For DeepWalk, when the length of random walk \( L \to \infty \),

\[
\frac{\#(w, c) |D|}{\#(w) \cdot \#(c)} \xrightarrow{p} \frac{\text{vol}(G)}{2T} \left( \frac{1}{d_c} \sum_{r=1}^{T} (P^r)_{w,c} + \frac{1}{d_w} \sum_{r=1}^{T} (P^r)_{c,w} \right).
\]
DeepWalk — Conclusion

Theorem

*DeepWalk is asymptotically and implicitly factorizing*

\[
\log \left( \frac{\text{vol}(G')}{b} \left( \frac{1}{T} \sum_{r=1}^{T} (D^{-1} A)^r \right) D^{-1} \right).
\]
DeepWalk — Roadmap

Input
\[ G = (V, E) \]

\[ \log \left( \frac{\text{vol}(G)}{b} \left( \frac{1}{T} \sum_{r=1}^{T} \left( D^{-1} A \right)^r \right) D^{-1} \right) \]

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

\[ \frac{\text{vol}(G) = \sum_i \sum_j A_{i,j}} \]

\[ b \text{ Number of negative samples} \]
Objective of LINE:

\[
\mathcal{L} = \sum_{i=1}^{|V|} \sum_{j=1}^{|V|} \left( A_{i,j} \log g (x_i^T y_j) + \frac{b d_i d_j}{\text{vol}(G)} \log g (-x_i^T y_j) \right).
\]

Align it with the Objective of SGNS:

\[
\mathcal{L} = \sum_{w} \sum_{c} \left( \#(w, c) \log g (x_w^T y_c) + \frac{b \#(w) \#(c)}{|\mathcal{D}|} \log g (-x_w^T y_c) \right).
\]

LINE is actually factorizing

\[
\log \left( \frac{\text{vol}(G)}{b} D^{-1} AD^{-1} \right)
\]

Recall DeepWalk’s matrix form:

\[
\log \left( \frac{\text{vol}(G)}{b} \left( \frac{1}{T} \sum_{r=1}^{T} (D^{-1} A)^r \right) D^{-1} \right).
\]

Observation LINE is a special case of DeepWalk \((T = 1)\).
Figure 2: Heterogeneous Text Network.

- word-word network $G_{ww}$, $A_{ww} \in \mathbb{R}^{\#\text{word} \times \#\text{word}}$.
- document-word network $G_{dw}$, $A_{dw} \in \mathbb{R}^{\#\text{doc} \times \#\text{word}}$.
- label-word network $G_{lw}$, $A_{lw} \in \mathbb{R}^{\#\text{label} \times \#\text{word}}$. 
PTE as Implicit Matrix Factorization

\[
\log \left( \begin{bmatrix}
\alpha \text{vol}(G_{ww})(D_{row}^{ww})^{-1} A_{ww}(D_{col}^{ww})^{-1} \\
\beta \text{vol}(G_{dw})(D_{row}^{dw})^{-1} A_{dw}(D_{col}^{dw})^{-1} \\
\gamma \text{vol}(G_{lw})(D_{row}^{lw})^{-1} A_{lw}(D_{col}^{lw})^{-1}
\end{bmatrix} \right) - \log b,
\]

- The matrix is of shape (\#word + \#doc + \#label) × \#word.
- \(b\) is the number of negative samples in training.
- \(\{\alpha, \beta, \gamma\}\) are hyper-parameters to balance the weights of the three networks. In PTE, \(\{\alpha, \beta, \gamma\}\) satisfy

\[
\alpha \text{vol}(G_{ww}) = \beta \text{vol}(G_{dw}) = \gamma \text{vol}(G_{lw})
\]
node2vec — 2nd Order Random Walk

\[ T_{u,v,w} = \begin{cases} 
\frac{1}{p} & (u,v) \in E, (v,w) \in E, u = w; \\
1 & (u,v) \in E, (v,w) \in E, u \neq w, (w,u) \in E; \\
\frac{1}{q} & (u,v) \in E, (v,w) \in E, u \neq w, (w,u) \notin E; \\
0 & \text{otherwise.}
\end{cases} \]

\[ P_{u,v,w} = \text{Prob}(w_{j+1} = u | w_j = v, w_{j-1} = w) = \frac{T_{u,v,w}}{\sum_u T_{u,v,w}}. \]

**Stationary Distribution**

\[ \sum_w P_{u,v,w} X_{v,w} = X_{u,v} \]

Existence guaranteed by Perron-Frobenius theorem, but may not be unique.
node2vec as Implicit Matrix Factorization

**Theorem**
node2vec is asymptotically and implicitly factorizing a matrix whose entry at \( w \)-th row, \( c \)-th column is

\[
\log \left( \frac{\frac{1}{2T} \sum_{r=1}^{T} \left( \sum_u X_{w,u} P_{c,w,u}^r + \sum_u X_{c,u} P_{w,c,u}^r \right)}{b \left( \sum_u X_{w,u} \right) \left( \sum_u X_{c,u} \right)} \right)
\]
Contents

Preliminaries

Main Theoretic Results
Notations
DeepWalk (KDD’14)
LINE (WWW’15)
PTE (KDD’15)
ode2vec (KDD’16)

NetMF
NetMF for a Small Window Size $T$
NetMF for a Large Window Size $T$
Experiments
\[
\log \left( \frac{\text{vol}(G)}{b} \left( \frac{1}{T} \sum_{r=1}^{T} (D^{-1} A)^r \right) D^{-1} \right)
\]

\[
\log \left( \frac{\#(w, c) |\mathcal{D}|}{b \#(w) \#(c)} \right)
\]
Factorize the DeepWalk matrix:

\[
\log \left( \frac{\text{vol}(G)}{b} \left( \frac{1}{T} \sum_{r=1}^{T} (D^{-1} A)^r \right) D^{-1} \right).
\]

For numerical reason, we use truncated logarithm —
\[\tilde{\log}(x) = \log(\max(1, x))\]
Algorithm 2: NetMF for a Small Window Size $T$

1. Compute $P^1, \ldots, P^T$;
2. Compute $M = \frac{\text{vol}(G)}{bT} \left( \sum_{r=1}^{T} P^r \right) D^{-1}$;
3. Compute $M' = \max(M, 1)$;
4. Rank-$d$ approximation by SVD: $\log M' = U_d \Sigma_d V_d^\top$;
5. return $U_d \sqrt{\Sigma_d}$ as network embedding.
NetMF for a Large Window Size $T$ — Observations

We want to factorize
\[
\log \left( \frac{\text{vol}(G)}{b} \left( \frac{1}{T} \sum_{r=1}^{T} (D^{-1} A)^{r} \right) D^{-1} \right).
\]

We know the property of normalized graph Laplacian
\[
D^{-1/2} A D^{-1/2} = U \Lambda U^\top
\]
where $\Lambda = \text{diag}(\lambda_1, \cdots, \lambda_n)$ and $\forall \lambda_i \in [-1, 1]$.

\[
\left( \frac{1}{T} \sum_{r=1}^{T} (D^{-1} A)^{r} \right) D^{-1} = \left( D^{-1/2} \right) \left( \frac{1}{T} \sum_{r=1}^{T} (D^{-1/2} A D^{-1/2})^{r} \right) \left( D^{-1/2} \right)
\]

\[
= \left( D^{-1/2} \right) \left( U \left( \frac{1}{T} \sum_{r=1}^{T} \Lambda^r \right) U^\top \right) \left( D^{-1/2} \right)
\]

where $\Lambda$ is a polynomial.
NetMF for a Large Window Size $T$ — Observations

![Figure 4: $f(\lambda) = \frac{1}{T} \sum_{r=1}^{T} \lambda^r$](image)

Idea

This polynomial implicitly filters out negative eigenvalues and small positive eigenvalues, why not do it explicitly.
Algorithm 3: NetMF for a Large Window Size $T$

1. Eigen-decomposition $D^{-1/2} A D^{-1/2} \approx U_h \Lambda_h U_h^\top$;
2. Approximate $M$ with
   \[
   \hat{M} = \frac{\text{vol}(G)}{b} D^{-1/2} U_h \left( \frac{1}{T} \sum_{r=1}^{T} \Lambda^r_h \right) U_h^\top D^{-1/2};
   \]
3. Compute $\hat{M}' = \max(\hat{M}, 1)$;
4. Rank-$d$ approximation by SVD: $\log \hat{M}' = U_d \Sigma_d V_d^\top$;
5. return $U_d \sqrt{\Sigma_d}$ as network embedding.
Setup

Label Classification:

- BlogCatalog, PPI, Wikipedia, Flickr
- Logistic Regression
- NetMF ($T = 1$) v.s. LINE
- NetMF ($T = 10$) v.s. DeepWalk

**Table 1**: Statistics of Datasets.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>BlogCatalog</th>
<th>PPI</th>
<th>Wikipedia</th>
<th>Flickr</th>
</tr>
</thead>
<tbody>
<tr>
<td>$</td>
<td>V</td>
<td>$</td>
<td>10,312</td>
<td>3,890</td>
</tr>
<tr>
<td>$</td>
<td>E</td>
<td>$</td>
<td>333,983</td>
<td>76,584</td>
</tr>
<tr>
<td>#Labels</td>
<td>39</td>
<td>50</td>
<td>40</td>
<td>195</td>
</tr>
</tbody>
</table>
Figure 5: 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.
Conclusion

Table 2: The matrices that are implicitly approximated and factorized by DeepWalk, LINE, PTE, and node2vec.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Matrix</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeepWalk</td>
<td>$\log \left( \frac{\text{vol}(G)}{\frac{1}{T} \sum_{r=1}^{T} \left( D^{-1} A \right)^{r}} \right) - \log b$</td>
</tr>
<tr>
<td>LINE</td>
<td>$\log \left( \frac{\text{vol}(G)}{D^{-1} AD^{-1}} \right) - \log b$</td>
</tr>
<tr>
<td>PTE</td>
<td>$\log \left( \begin{bmatrix} \alpha \frac{\text{vol}(G_{ww})}{\left( D_{row}^{ww} \right)^{-1}} &amp; \alpha \frac{\text{vol}(G_{ww})}{\left( D_{col}^{ww} \right)^{-1}} \ \beta \frac{\text{vol}(G_{dw})}{\left( D_{row}^{dw} \right)^{-1}} &amp; \beta \frac{\text{vol}(G_{dw})}{\left( D_{col}^{dw} \right)^{-1}} \ \gamma \frac{\text{vol}(G_{lw})}{\left( D_{row}^{lw} \right)^{-1}} &amp; \gamma \frac{\text{vol}(G_{lw})}{\left( D_{col}^{lw} \right)^{-1}} \end{bmatrix} \right) - \log b$</td>
</tr>
<tr>
<td>node2vec</td>
<td>$\log \left( \frac{1}{2T} \sum_{r=1}^{T} \left( \frac{\sum_{u} X_{w,u} P_{c,w,u}^{r} + \sum_{u} X_{c,u} P_{w,c,u}^{r}}{\left( \sum_{u} X_{w,u} \right) \left( \sum_{u} X_{c,u} \right)} \right) \right) - \log b$</td>
</tr>
</tbody>
</table>
Thanks.

Standing on the shoulders of giants
— Isaac Newton

Code available at github.com/xptree/NetMF

Q&A
Theorem
Denote \( P = D^{-1} A \), when \( L \to \infty \), we have

\[
\frac{\#(w, c) \to}{|D \to|} \overset{p}{\to} \frac{d_w}{\text{vol}(G)} (P^r)_{w,c} \quad \text{and} \quad \frac{\#(w, c) \leftarrow}{|D \leftarrow|} \overset{p}{\to} \frac{d_c}{\text{vol}(G)} (P^r)_{c,w}.
\]

Proof.
Consider the special case when \( N = 1 \), thus we only have one vertex sequence \( w_1, \cdots, w_L \) generated by random walk. Let \( Y_j \ (j = 1, \cdots, L - T) \) be the indicator function for event that \( w_j = w \) and \( w_{j+r} = c \).
Proof (Con’t)

Observation

\[\mathbb{E}[Y_j] = \text{Prob}(w_j = w, w_{j+r} = c) \to \frac{d_w}{\text{vol}(G)} (P^r)_{w,c}.\]

\[\frac{\#(w,c) \rightarrow r}{|D \rightarrow r|} = \frac{1}{L-T} \sum_{j=1}^{L-T} Y_j.\]

\[\text{Cov}(Y_i, Y_j) \to 0 \text{ as } |i - j| \to \infty.\]

Lemma

(S.N. Bernstein Law of Large Numbers) Let \(Y_1, Y_2 \cdots\) be a sequence of random variables with finite expectation \(\mathbb{E}[Y_j]\) and variance \(\text{Var}(Y_j) < K, j \geq 1\), and covariances are s.t.
\(\text{Cov}(Y_i, Y_j) \to 0 \text{ as } |i - j| \to \infty\). Then the law of large numbers (LLN) holds.

\[\frac{\#(w,c) \rightarrow r}{|D \rightarrow r|} = \frac{1}{L-T} \sum_{j=1}^{L-T} Y_j \xrightarrow{p} \frac{1}{L-T} \sum_{j=1}^{L-T} \mathbb{E}(Y_j) \to \frac{d_w}{\text{vol}(G)} (P^r)_{w,c}\]
Time Complexity

- Eigen-Decomposition (Implicitly Restarted Lanczos Method) $O(mhI + nh^2I + h^3I)$.
- Reconstruction $O(n^2h)$
- Element-wise logarithm $O(n^2)$.
- SVD (a naive implementation with eigen-decomposition): $O(n^2dI + nd^2I + d^3I)$. 

Future Work

- Comprehend high-order cases, e.g., node2vec.

$$\log \left( \frac{1}{2T} \sum_{r=1}^{T} \left( \sum_u X_{w,u} P_{c,w,u}^r + \sum_u X_{c,u} P_{w,c,u}^r \right) \right)$$

- Design scalable algorithm (e.g., using spectral sparsification of random-walk polynomials).

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

- Connection with graph convolutional networks (Kipf & Welling, ICLR’17).