SepNE: Bringing Separability to Network Embedding

Ziyao Li, Liang Zhang and Guojie Song

Presented by: Ziyao Li
Jan. 2019

Contents

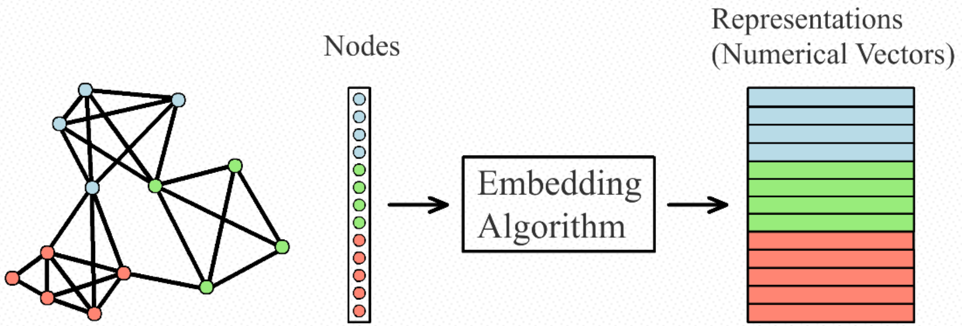
Part 01

Motivation

The motivation of bringing up *separability*.

What is Network Embedding?

- **Network Embedding (NE):** Finding network data representations which concisely represent networks, in order to achieve efficiency in various downstream applications such as pattern discovery, analysis and prediction.



Network Embedding

Current Embedding Methods

- Traditional Methods: LLE[1]; Isomap[2]; Laplace Eigenmaps[3];
- Random-Walk Based: DeepWalk[4]; node2vec[5];
- Edge-Reconstruction Based: LINE[6];
- Neural Network Based: GCN[7]; GraphSAGE[8].

Inefficiencies

- Almost all current embedding algorithms learn representations for entire networks, even when only a small proportion of nodes are requested. This leads to:

- 1. Vain efforts in embedding unrequested nodes.

- 2. A limitation over the maximum network scale that such methods can handle.

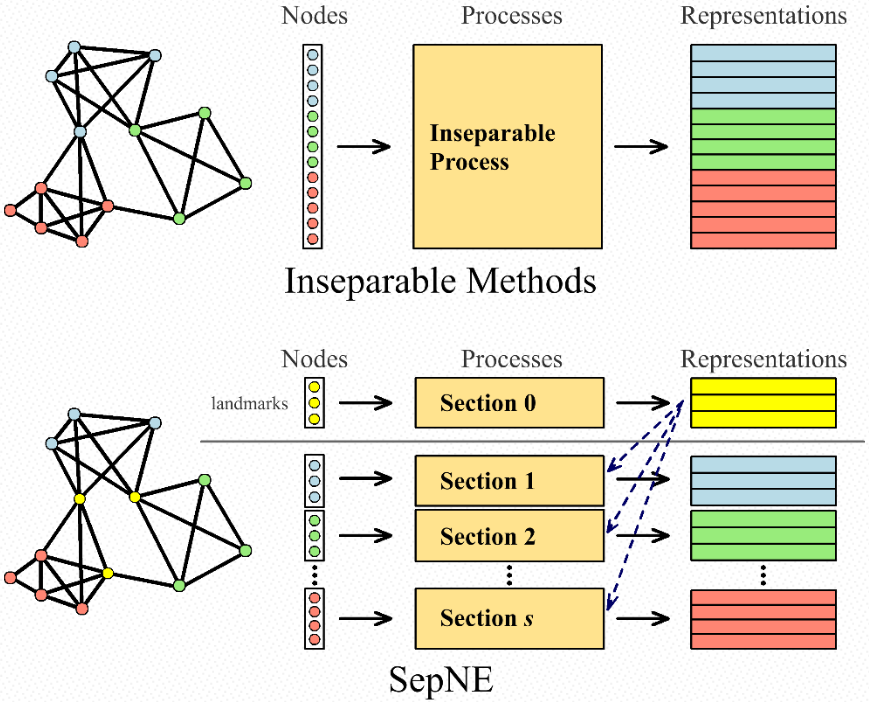
Improvement

- An impossibility: it is impossible to directly solve the efficiency problem on super-scale networks, as the running time of algorithms inevitably grows linearly to problem scales.
- An observation: often, it is unnecessary to derive representations for all of the nodes in a network.
- An attempt: to learn representations for different subsets of nodes--very small compared to the collectivity--while preserving information of the entire network.

Separability

- **Separability**: a separable algorithm divides the original problem into different self-standing sub-problems and separately solves each, and the solution to the sub-problems are directly usable answers instead of intermediate results.

SepNE v.s. Inseparable algorithms



A Inseparability

Very clumsy when handling super-large networks.

B SepNE

Flexible, fast and parallelizable.

Part 02

Separated Matrix Factorization

Basic idea, approximation and optimization goal.

Matrix Factorization

- **Matrix Factorization (MF):** finding two matrices W and C that both satisfy given constraints and minimize the residuals of reconstructing original matrix M.

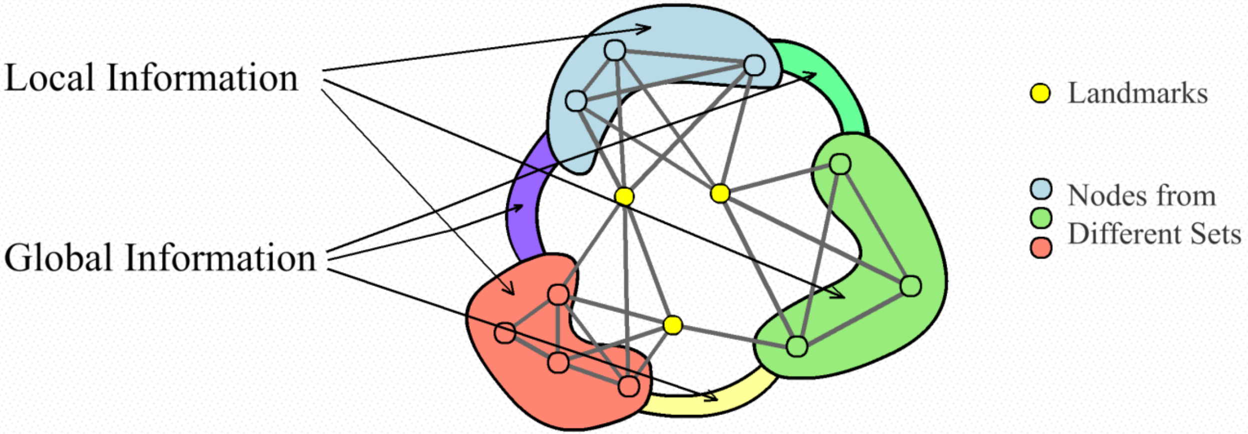
- Denoted in formula:

$$\min_{W,C} \|M - W^T C\|$$

- **MF based Network Embedding:** Factorizing a somehow normalized proximity matrix to derive node representations.

Local and Global Information

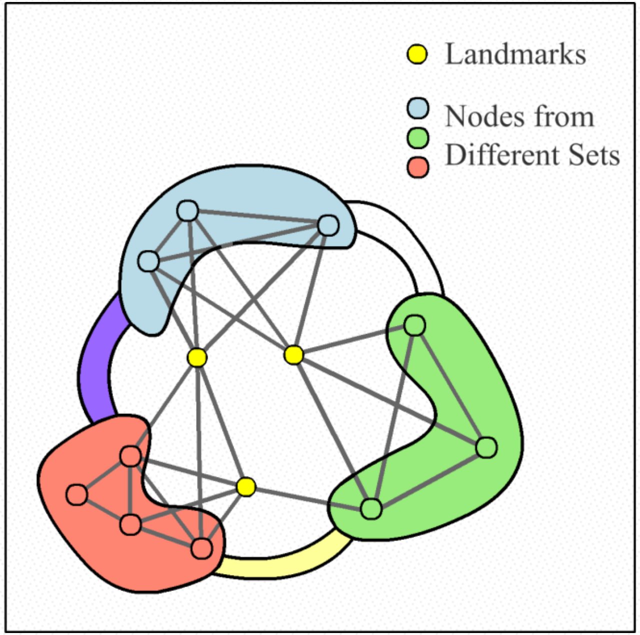
- Local Information: the proximity within a given set of nodes.
- Global Information: the proximity between two given set of nodes.

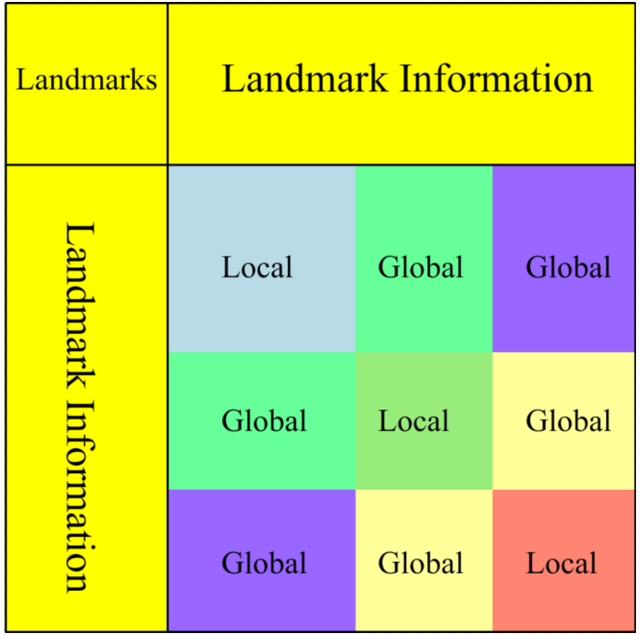


Landmark Information

- Landmark Information: the proximity between a given set of nodes and the selected landmarks.
- Why landmarks? To maintain comparability of representation vectors of different sets.

From Graph to Matrix





Preserving Different Information

- Local reconstructing:

$$Loss^{lc} = \|M_{ii} - W_i^T C_i\|, i = 0,1,\cdots,s$$

- Landmark reconstructing

$$Loss^{lm} = \|M_{i0} - W_i^T C_0\| + \|M_{0i} - W_0^T C_i\|, i = 1,2,\cdots,s$$

- Global reconstructing

$$Loss^{gb} = \|M_{i\bar{i}} - W_i^T C_{\bar{i}}\| + \|M_{\bar{i}i} - W_{\bar{i}}^T C_i\|, i = 1,2,\cdots,s$$

Approximating Global Information via Landmarks

- A transformation:

$$\Phi = W_0;\ \Psi = C_0$$
$$W_i = \Phi A_i; C_i = \Psi B_i$$

- Global approximation:

$$\begin{aligned} Loss^{gb} &= \|M_{i\bar{i}} - W_i^T C_{\bar{i}}\| + \|M_{\bar{i}i} - W_{\bar{i}}^T C_i\| \\ &= \|M_{i\bar{i}} - A_i^T \Phi^T \Psi B_{\bar{i}}\| + \|M_{\bar{i}i} - A_{\bar{i}}^T \Phi^T \Psi B_i\| \\ &\approx \|M_{i\bar{i}} - A_i^T M_{0\bar{i}}\| + \|M_{\bar{i}i} - M_{\bar{i}0} B_i\| \end{aligned}$$

Final Loss of SMF

- Global approximation:

$$Loss = Loss^{lc} + Loss^{lm} + \lambda Loss^{gl} + \eta(\|A\|_F^2 + \|B\|_F^2),$$

where

$$Loss^{lc} = \frac{1}{2}\left\|M_{ii} - A_i^T \Phi^T \Psi B_i\right\|_F^2$$

$$Loss^{lm} = \frac{1}{2}\left(\left\|M_{i0} - A_i^T \Phi^T \Psi\right\|_F^2 + \|M_{0i} - \Phi^T \Psi B_i\|_F^2\right)$$

$$Loss^{gb} = \frac{1}{2}\left(\left\|M_{i\bar{i}} - A_i^T M_{0\bar{i}}\right\|_F^2 + \|M_{\bar{i}i} - M_{\bar{i}0} B_i\|_F^2\right)$$

Part 03

SepNE

Algorithm details.

Algorithm

- 1. Graph partitioning;
- 2. Landmark selecting;
- 3. SVD over landmark local information (derive Φ, Ψ);
- 4. Optimizing SMF over different sets.

Preparation Stage

Optimization Stage

Graph Partitioning

- Structure-based setup: partitioning the graph according to network communities; Louvain[9] is used in the paper.

- Random partition: randomly partition the network into subsets.

- Request-based setup: only extract requested nodes as one or more sets.

Landmark Selecting

- Degree-based methods: selecting landmarks according to node degrees, probabilistically or deterministically.

- GDS: Greedy Dominating Set approach, choosing in each step the node with highest degree which is un-dominated.

Optimization & Complexity

- Using an iterative optimization method:

$$A_i^{(t+1)} = \min_A Loss(A, B_i^{(t)})$$

$$B_i^{(t+1)} = \min_B Loss(A_i^{(t+1)}, B)$$

- Complexity: preparation - $O(n \log n + k^3)$
  optimization - $O(k \times (deg + iter \times k) \times n_i)$

- The optimization complexity is irrelevant to the network scale.

Part 04

Experiments & Discussion

Experimental results, analysis and discussion

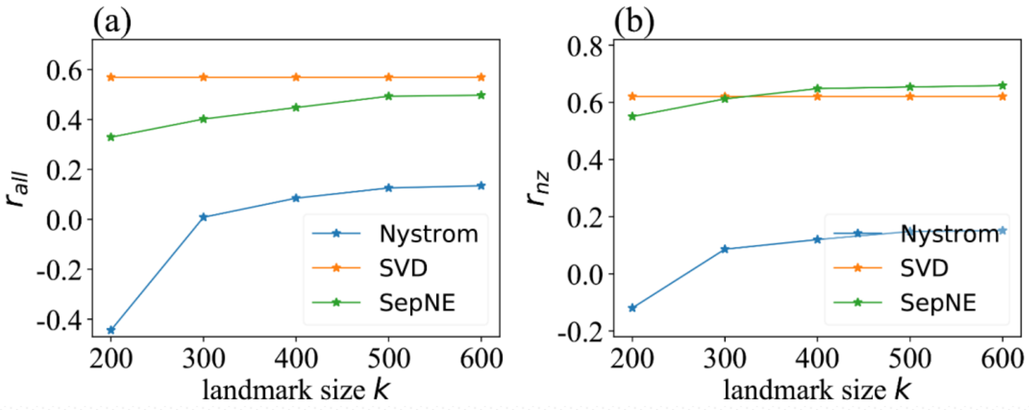
Running Time

Algorithm	Running Time
SepNE-IO(Interested Only)	6.2min
SepNE-RP(Random Partition)	43.8min
SepNE-LP(Louvain Partition)	68.8min
LINE(1st)	138.1min
DeepWalk	>24hrs

Running Time over Flickr Data (1.7M nodes, 70k requested).

- SepNE shows much better speed than current embedding methods.

Matrix Reconstruction



Reconstruction loss of Nystrom[10], SVD and SepNE.

- Compared with other matrix factorization methods (Nystrom), SepNE shows comparable reconstruction ability to SVD.

Accuracy – Document Datasets

	Wiki		Cora		Citeseer	
%train	10%	90%	10%	90%	10%	90%
LINE(1st)	0.4488	0.5937	0.4657	0.6009	0.3206	0.4259
LINE(2nd)	0.3298	0.4787	0.2637	0.3297	0.2221	0.2561
DeepWalk	0.5737	0.6893	0.7509	0.8187	0.5086	0.5813
SepNE	0.5764	0.6867	0.7365	0.822	0.5157	0.6072

Micro F1 over document Datasets.

- SepNE shows competent performance over small datasets.

Accuracy – Flickr

%train	1%	3%	5%	10%	20%	30%	50%	90%
LINE(1st)	0.3683	0.4118	0.4165	0.4219	0.4270	0.4273	0.4296	0.4274
LINE(2nd)	0.3450	0.3824	0.3955	0.3973	0.4032	0.4056	0.4069	0.4068
DeepWalk	0.4072	0.4353	0.4433	0.4481	0.4518	0.4564	0.4585	0.4592
SepNE-IO	0.4065	0.4341	0.4477	0.4562	0.4582	0.4607	0.4630	0.4622
SepNE-RP	0.4061	0.4388	0.4502	0.4601	0.4628	0.4634	0.4636	0.4658
SepNE-LP	0.4269	0.4468	0.4562	0.4623	0.4645	0.4656	0.4674	0.4677

Micro F1 over Flickr Datasets.

- On Flickr, SepNE even shows significantly better performance than DeepWalk and LINE.

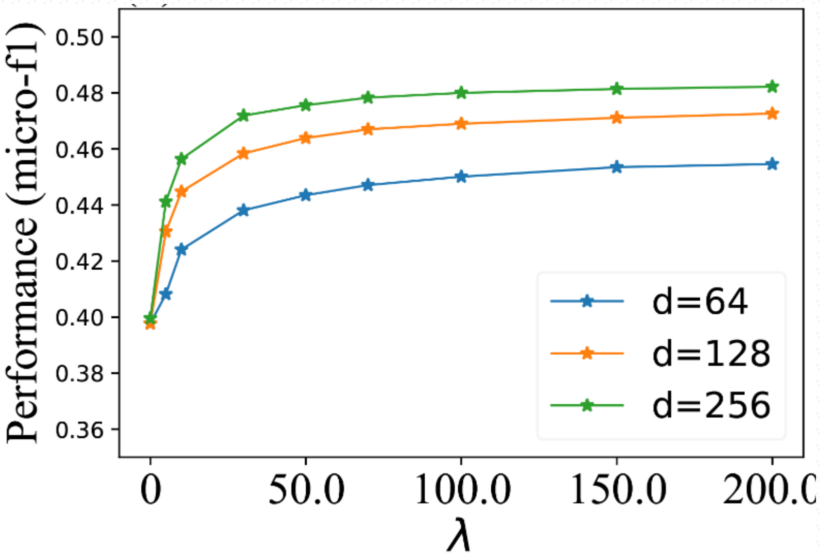
Accuracy – Flickr

%train	1%	3%	5%	10%	20%	30%	50%	90%
LINE(1st)	0.1031	0.2322	0.2745	0.3141	0.3410	0.3520	0.3594	0.3673
LINE(2nd)	0.0782	0.1839	0.2158	0.2643	0.2987	0.3159	0.3280	0.3350
DeepWalk	0.2037	0.3397	0.3739	0.4105	0.4355	0.4438	0.4501	0.4556
SepNE-IO	0.2035	0.3325	0.3574	0.3885	0.4041	0.4129	0.4170	0.4216
SepNE-RP	0.2256	0.3355	0.3633	0.3920	0.4115	0.4157	0.4214	0.4273
SepNE-LP	0.2253	0.3361	0.3620	0.3882	0.4118	0.4170	0.4218	0.4277

Micro F1 over Flickr Datasets.

- On Youtube, SepNE also shows competent accuracy.

Discussion: Parameter λ



- Larger λ leads to better performance, converging with λ >100.
- The higher performances of larger λs, particularly compared with λ=0, show the effectiveness of the elaborated global loss.

Conclusion

- With comparable accuracy, SepNE significantly outperforms state-of-the-art baselines in running times and flexibility.

- SMF reduced the complexity of MF from $O(n^3)$ to $O(n \log n)$. Should there be a theoretical proof of a lower bound over the loss in matrix reconstruction, SMF can be generated to all MF-based algorithms.

Possible Future Work

- More efficiently exploring higher-order proximities in SepNE;
- Adapting SepNE to dynamic network scenarios;
- Finding a theoretical proof of a lower bound over the loss of SMF.

THANK YOU

References

- [1] Roweis, S., and Saul, L. 2000. Nonlinear dimensionality reduction by locally linear embedding. Science 2323–2326.
- [2] Tenenbaum, J. B. et al. 2000. A global geometric framework for nonlinear dimensionality reduction. Science 2319–2323.
- [3] Belkin, M., and Niyogi, P. 2002. Laplacian eigenmaps and spectral techniques for embedding and clustering. In Advances in NIPS 14. 585–591.

References

- [4] Perozzi, B. et al. 2014. Deepwalk: online learning of social representations. In Proceedings of KDD-14, 701–710.
- [5] Grover, A., and Leskovec, J. 2016. node2vec: Scalable feature learning for networks. In Proceedings of KDD-16, 855–864.
- [6] Tang, J. et al. 2015. Line: large-scale information network embedding. In Proceedings of WWW-15, 1067–1077.
- [7] Kipf, T. N., and Welling, M. 2016. Semi-supervised classification with graph convolutional networks. CoRR abs/1609.02907.

References

- [8] Hamilton, W. et al. 2017. Inductive representation learning on large graphs. In Advances in NIPS 30. 1024–1034.
- [9] Blondel, V. D. et al. 2008. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment* 2008(10):P10008.
- [10] Drineas, P., and Mahoney, M. 2005. On the nystrm method for approximating a gram matrix for improved kernel-based learning. *Journal of Machine Learning Research* 6:2153–2175.