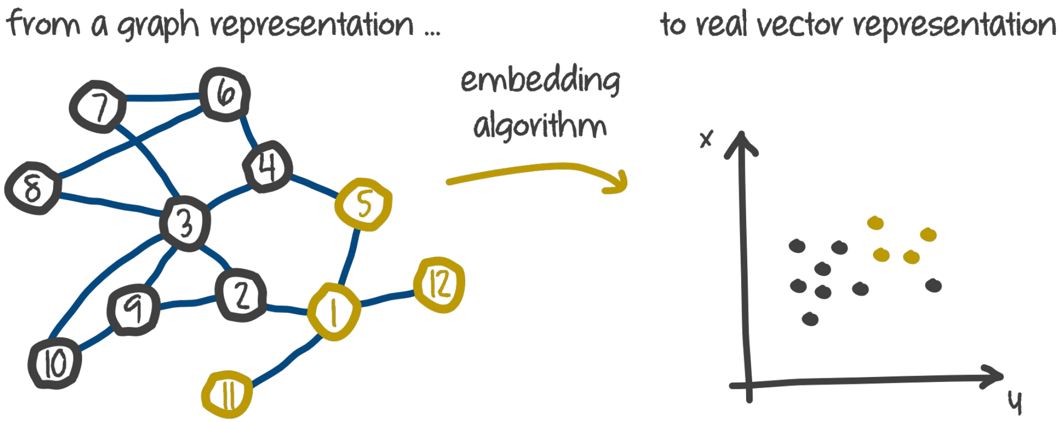
Community Preserving Network Embedding

Xiao Wang, Peng Cui, Jing Wang, Jian Pei, Wenwu Zhu, Shiqiang Yang

Presented by: Ben, Ashwati, SK

What is Network Embedding

1. Representation of a node in an m-dimensional vector space s.t. its properties are preserved.
2. # dimensions << # vertices (generally)
3. Preserve one or all of
   - $1 \sim k^{th}$ order proximities
   - Community information

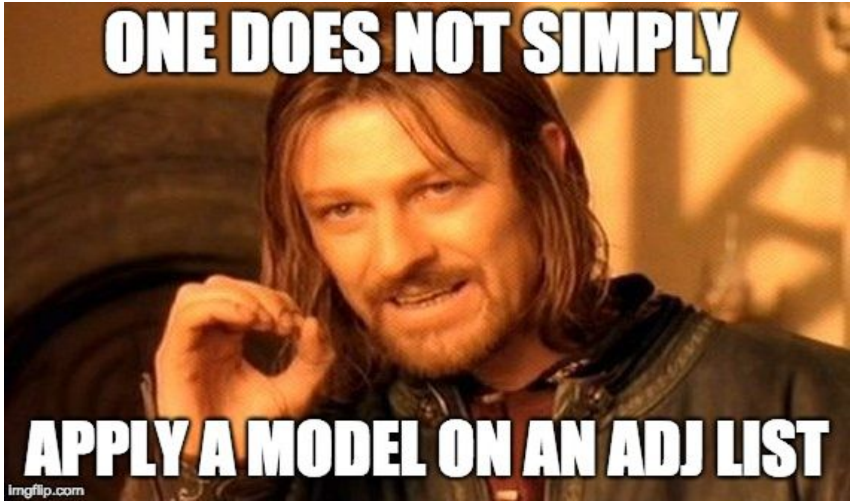


http://gear.github.io/img/mage_example.png

[Goyal and Ferrara '17]

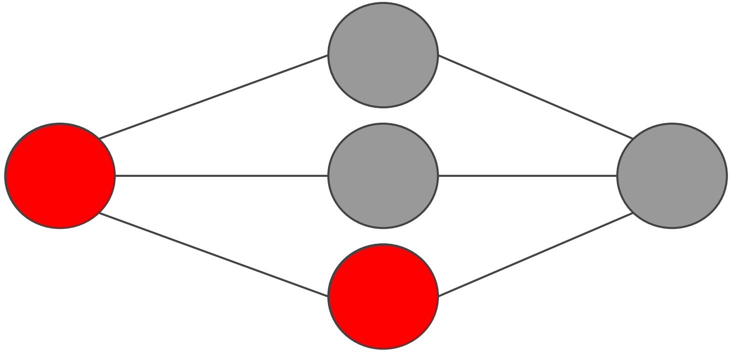
Motivation - Why Network Embedding?

- Graphs usually represented as adjacency list/matrix
- Machine Learning algorithms are not friendly to such adjacency representations
- Continuous valued vectors are much more model friendly
- Applications
  - Network Visualization
  - Network Compression
  - Link Prediction
  - Node Classification
  - Clustering



Goals of Network Embedding:

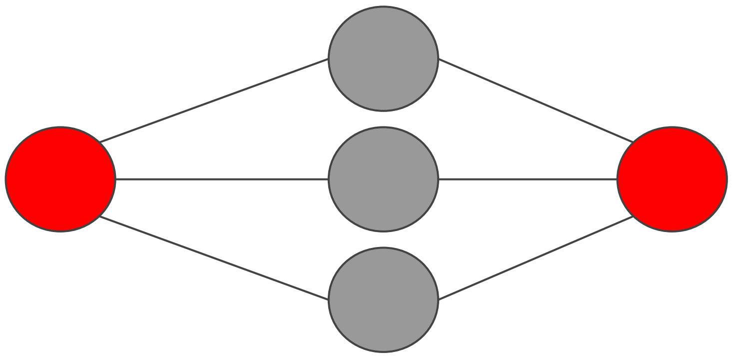
- Preserve 1st-order proximity (neighboring nodes have similar embeddings)



- Accomplished by: most network embedding techniques

Goals of Network Embedding:

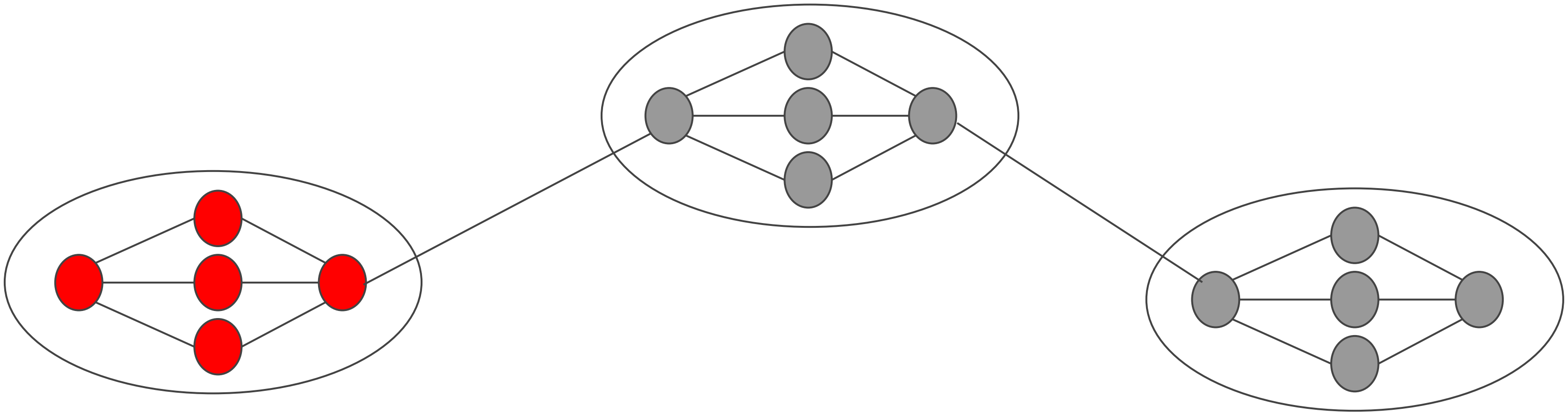
- Preserve 2nd-order proximity (nodes with similar sets of neighbors have similar embeddings)



- Accomplished by: various network embedding techniques, including LINE, DeepWalk, etc.

Goals of Network Embedding:

- Preserve community structure (nodes in same community have similar embeddings)



- Accomplished by: Modularized non-negative matrix factorization (this paper)

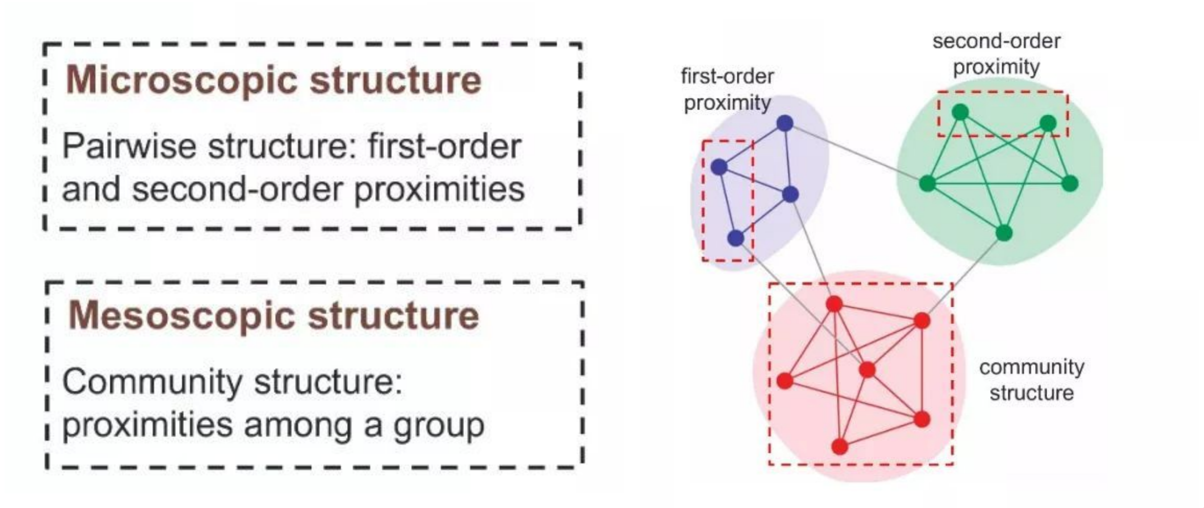
Techniques

Category	Year	Published	Method	Time Complexity	Properties preserved
Factorization	2000	Science[26]	LLE	$O(\|E\|d^2)$	1^{st} order proximity
	2001	NIPS[25]	Laplacian Eigenmaps	$O(\|E\|d^2)$	
	2013	WWW[21]	Graph Factorization	$O(\|E\|d)$	
	2015	CIKM[27]	GraRep	$O(\|V\|^3)$	$1-k^{th}$ order proximities
	2016	KDD[24]	HOPE	$O(\|E\|d^2)$	
Random Walk	2014	KDD[28]	DeepWalk	$O(\|V\|d)$	
	2016	KDD[29]	*node2vec*	$O(\|V\|d)$	$1-k^{th}$ order proximities, structural equivalence
Deep Learning	2016	KDD[23]	SDNE	$O(\|V\|\|E\|)$	1^{st} and 2^{nd} order proximities
	2016	AAAI[30]	DNGR	$O(\|V\|^2)$	$1-k^{th}$ order proximities
	2017	ICLR[31]	GCN	$O(\|E\|d^2)$	$1-k^{th}$ order proximities
Miscellaneous	2015	WWW[22]	LINE	$O(\|E\|d)$	1^{st} and 2^{nd} order proximities

[Goyal and Ferrara '17]

Community Preserving Network Embedding

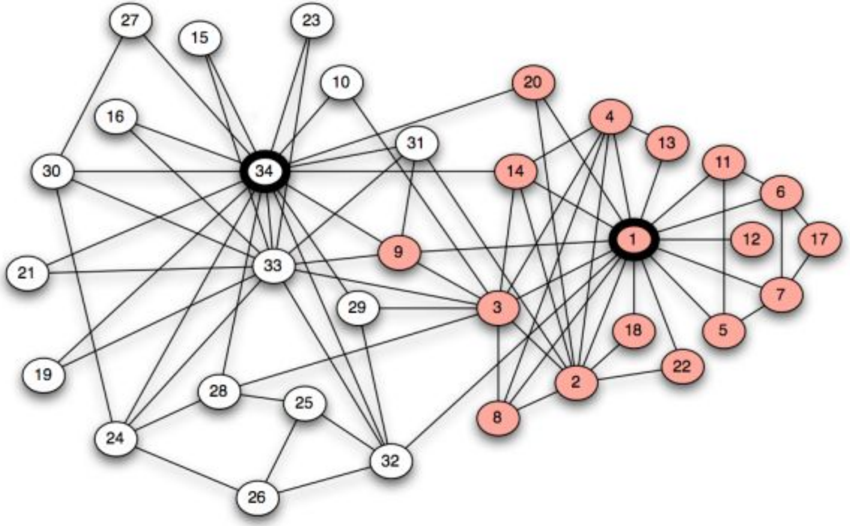
- Preserves node's community affiliation and proximities
- Representation of nodes within same community more similar



http://zhuanzhiimg.b0.upaiyun.com/images/wx/fde7a2ac92b77564bdf08af297e31ff1

Motivation - Why preserve communities?

- Existing methods focus on microscopic structures
- Community structure of the network ignored
- For majority of networks, communities highlight organizational structure and functional components
- Two weakly connected nodes of same community should be closer in the vector space



34 person Zachary's Karate Club Network

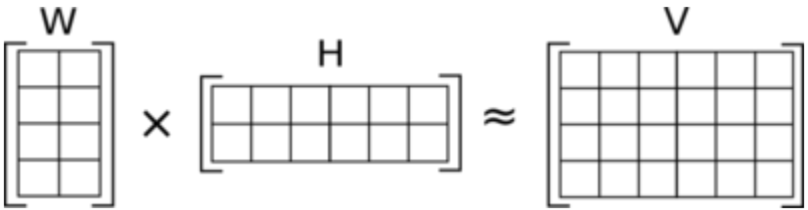
Image Courtesy: Srinivasan Parthasarathy, OSU CSE5245: Introduction to Network Science, Lecture 1

Technique developed

- Novel method that considers both proximity and community structure
  - Modularized Non Negative Matrix Factorization
- Pairwise similarity using a combination of first and second order proximities
- Communities detected using modularity maximization

Non-negative matrix factorization (NMF or NNMF)

- Matrix V factorized into W and H s.t. the matrices have no negative elements
- Non-negativity inherent to the data being considered
- Problem not exactly solvable, thus approximated numerically.



https://en.wikipedia.org/wiki/Non-negative_matrix_factorization

Modularized Nonnegative Matrix Factorization

- Modularized NMF

Entire eq = non - convex optimization
Break it down and solve each part individually

$$\min_{\mathbf{M},\mathbf{U},\mathbf{H},\mathbf{C}} \|\mathbf{S} - \mathbf{M}\mathbf{U}^T\|_F^2 + \alpha\|\mathbf{H} - \mathbf{U}\mathbf{C}^T\|_F^2 - \beta tr(\mathbf{H}^T\mathbf{B}\mathbf{H})$$

$$s.t., \mathbf{M} \geqslant 0, \mathbf{U} \geqslant 0, \mathbf{H} \geqslant 0, \mathbf{C} \geqslant 0, tr(\mathbf{H}^T\mathbf{H}) = n,$$

$\|\mathbf{S} - \mathbf{M}\mathbf{U}^T\|_F^2$ Preserving the first and second order proximity

$\beta tr(\mathbf{H}^T\mathbf{B}\mathbf{H})$ Identifying the community structure

$\alpha\|\mathbf{H} - \mathbf{U}\mathbf{C}^T\|_F^2$ Linking up the node embedding with the community structure

Terminology

S = Similarity Matrix
M = Basis Matrix
U = Representation Matrix
H = Community Membership Indicator
C = Community Representation Matrix
B = Modularity Matrix
α/β = Control parameters

http://nrl.thumedia.org/community-preserving-network-embedding

Optimization

- Objective function not convex
  - Optimization separated to 4 subproblems
  - Iteratively optimize each subproblem till converges to local minima
- Overall computational complexity

$$\mathcal{O}(n^2m + n^2k)$$

  - n = # nodes, m = dimensionality, k = # communities
  - Comparable to other graph factorization algorithms
    - But slower than random-walk-based approaches

Experimental Evaluation

- Two tasks
  - Node clustering
  - Node classification
- Real Networks for testing w/ ground truth
  - WebKB Network with 4 subnetworks
  - Polblogs
  - Four Facebook networks at different US universities
- Baseline comparison
  - DeepWalk
  - LINE1 - first order proximity
  - LINE2 - second order proximity
  - GraRep
  - Node2Vec

Node Clustering: k-means clustering

- Green: Significantly better
- Red: Slight improvement

Table 1: Accuracy (%) of node clustering (bold numbers represent the best results).

Methods	DeepWalk	LINE1	LINE2	GraRep	Node2Vec	M-NMF0	M-NMF
Cornell	32.82	35.38	42.56	33.85	34.36	40.00	**43.05**
Texas	37.97	40.64	55.61	35.29	50.27	47.06	**63.10**
Washington	35.65	38.70	53.48	36.52	41.74	55.65	**59.57**
Wisconsin	34.34	35.09	43.77	36.60	35.47	42.64	**45.66**
Polblogs	52.68	57.38	63.88	53.42	**84.83**	72.75	82.82
Amherst	10.34	42.36	44.38	46.41	41.66	43.54	**47.25**
Hamilton	10.15	33.47	31.30	38.81	35.41	38.34	**42.49**
Mich	11.66	15.58	14.63	**35.12**	14.05	29.66	31.50
Rochester	7.94	17.88	16.86	33.80	18.00	30.35	**38.09**

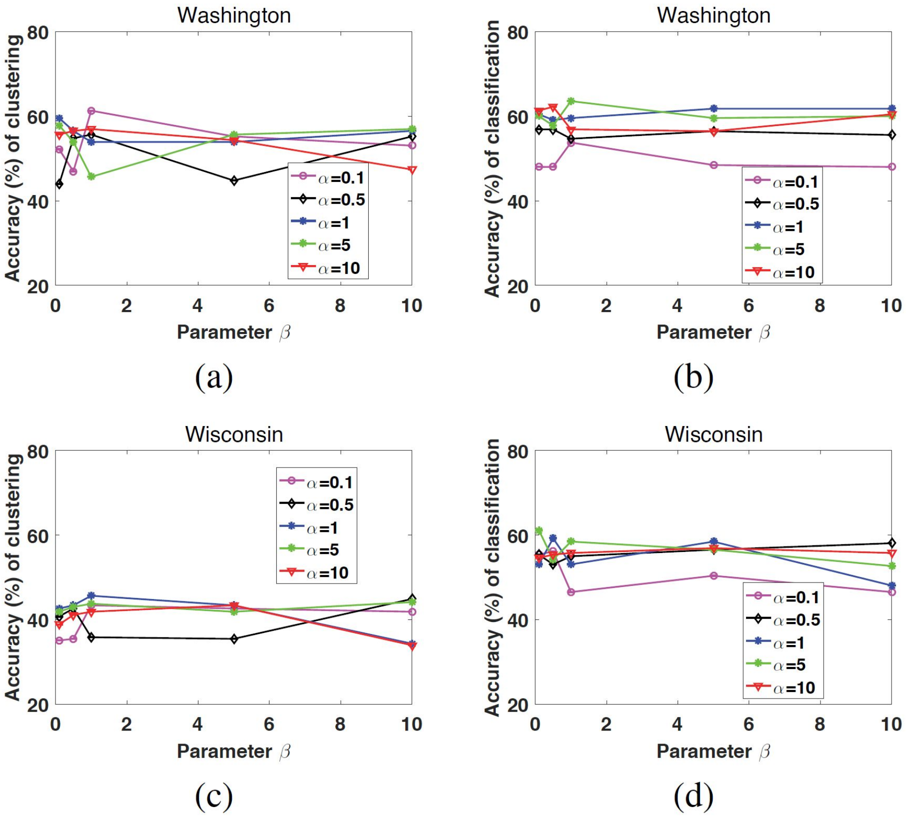
Node Classification: Linear Classifier

- Significantly better
- Slight improvement

Table 2: Accuracy (%) of node classification (bold numbers represent the best results).

Methods	DeepWalk	LINE1	LINE2	GraRep	Node2Vec	M-NMF0	M-NMF
Cornell	24.10	27.69	44.62	45.38	38.46	27.69	**47.18**
Texas	22.63	34.21	**73.16**	68.42	51.05	47.89	70.00
Washington	24.44	25.33	50.22	52.00	53.78	54.67	**63.56**
Wisconsin	26.15	28.46	51.54	59.62	44.62	39.62	**61.15**
Polblogs	64.77	83.02	80.87	89.60	84.03	80.20	**90.67**
Amherst	41.59	91.51	87.99	91.46	89.73	87.74	**92.00**
Hamilton	39.95	91.64	87.27	91.64	91.45	89.36	**92.92**
Mich	25.44	62.09	60.75	60.79	61.98	58.15	**62.26**
Rochester	34.78	87.04	84.23	85.47	83.65	84.28	**87.18**

How to pick alpha and beta



- Accuracies of clustering and classification does not change much with varying α and β

Summary

Technique

Pros:

- Clear benefit from preserving community structure (M-NMF vs. M-NMF0)
- Including community info does not increase computational complexity
- Extensible to include higher order proximities

Cons:

- Computationally complex
- Performance often comparable/poorer than state-of-the-arts
- Matrix factorization less intuitive than other approaches (random walks)

Paper

Pros:

- Clear motivation
- Analysis of parameters alpha and beta

Cons:

- Only accuracy considered
- Not tried on synthetic datasets
- Need more discussion on how/why including community info improves results
- Labels in classification are community dependent
- Effects of dimension not included
- Contribution of community preservation to performance in other tasks is unclear
- Using m for dimensionality in complexity analysis is confusing
- Abbreviation is a mouthful.. MNMNMNNN??

Thank You

Questions??