

Attributed network embedding

- Motivations & challenges
- Mining attributed networks with shallow embedding
 - Coupled spectral embedding
 - Coupled matrix & tri-factorization
 - Random walk based embedding
- Mining attributed networks with deep embedding
- Human-centric network analysis

Coupled spectral embedding

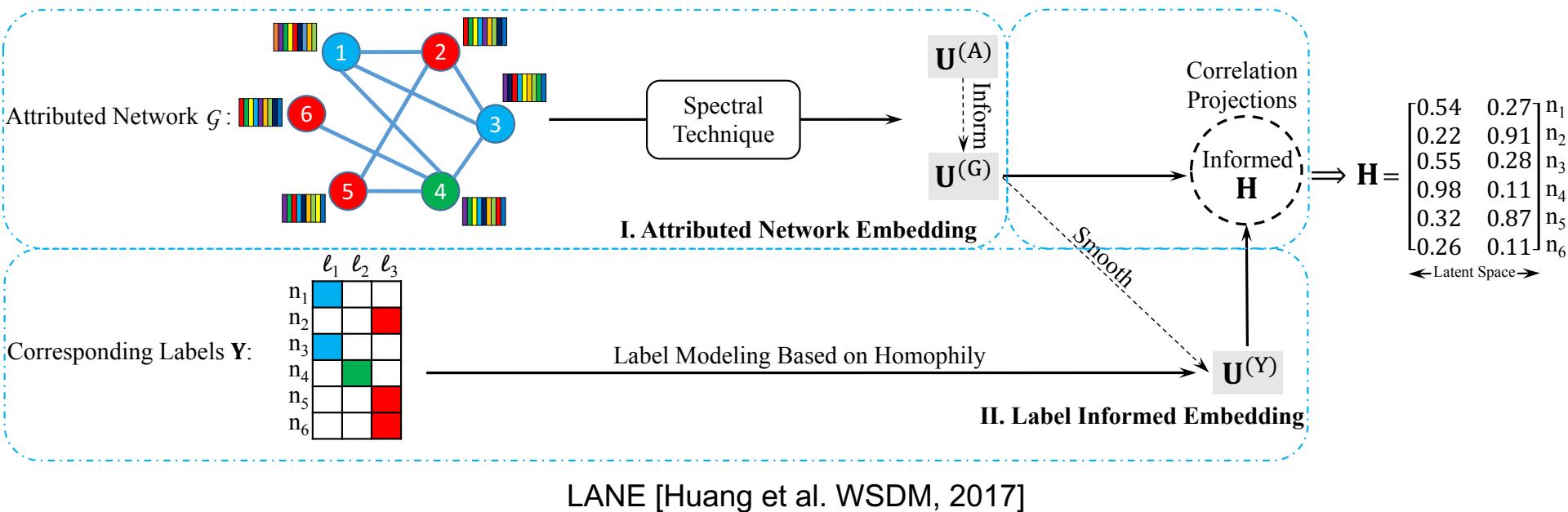
- Spectral embedding on plain networks:

$$\underset{\mathbf{U}}{\text{minimize}} \quad \frac{1}{2} \sum_{i,j=1}^n g_{ij} \left\| \frac{\mathbf{u}_i}{\sqrt{d_i}} - \frac{\mathbf{u}_j}{\sqrt{d_j}} \right\|_2^2 = \text{Trace}[\mathbf{U}^\top (\mathbf{I} - \mathbf{D}^{-\frac{1}{2}} \mathbf{G} \mathbf{D}^{-\frac{1}{2}}) \mathbf{U}]$$

Normalized Graph Laplacian

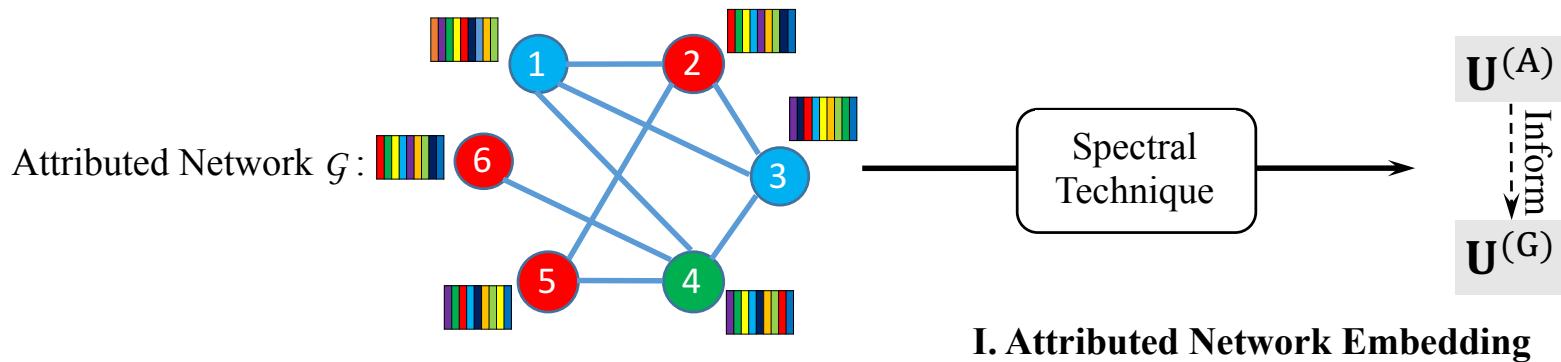
- For each pair of nodes i and j , larger g_{ij} tends to make their vector representations more similar
- **Spectral Graph Theory:** Eigenvalues are strongly connected to almost all key invariants of a graph
- How to extend spectral embedding to attributed networks?
 - Challenges: Heterogeneity & Large Scale

Label informed attributed network embedding



- **Goal:** embed nodes with similar network structure, attribute proximity, or same label into similar vector representations

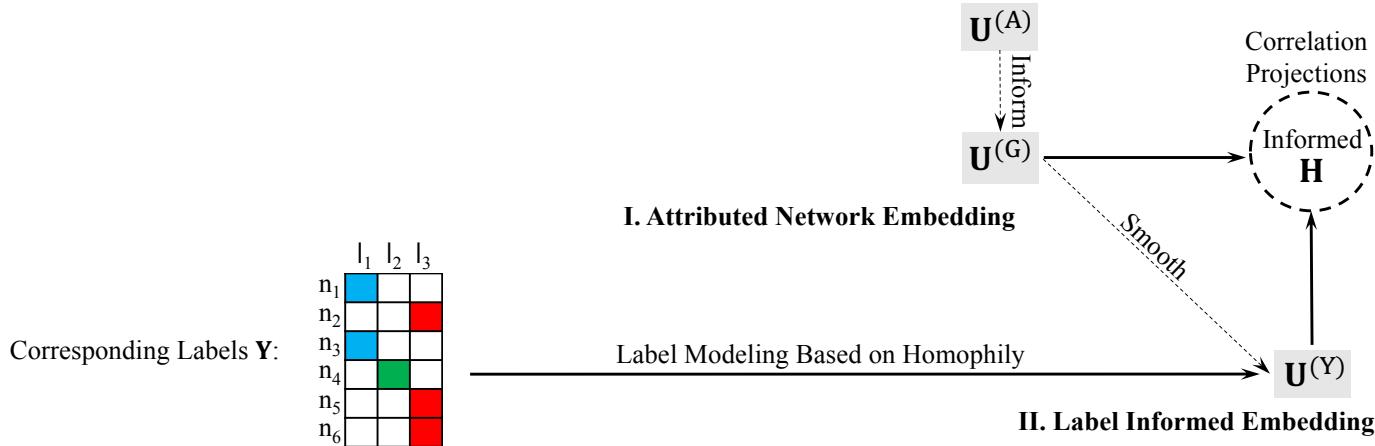
Couple embedding via correlation projection



- Though network \mathbf{G} , node attributes \mathbf{A} , labels \mathbf{Y} are heterogeneous, node proximities defined by $\mathbf{G}, \mathbf{A}, \mathbf{Y}$ are homogeneous
- We map the node proximities in network and node attributes into two latent representations $\mathbf{U}^{(G)}$ and $\mathbf{U}^{(A)}$ via spectral embedding and fuse them by extracting their correlations

$$\underset{\mathbf{U}^{(G)}, \mathbf{U}^{(A)}}{\text{maximize}} \quad \text{Tr}(\mathbf{U}^{(G)}^\top \mathcal{L}^{(G)} \mathbf{U}^{(G)} + \alpha \mathbf{U}^{(A)}^\top \mathcal{L}^{(A)} \mathbf{U}^{(A)} + \alpha \mathbf{U}^{(A)}^\top \mathbf{U}^{(G)} \mathbf{U}^{(G)\top} \mathbf{U}^{(A)})$$

Uniform projections



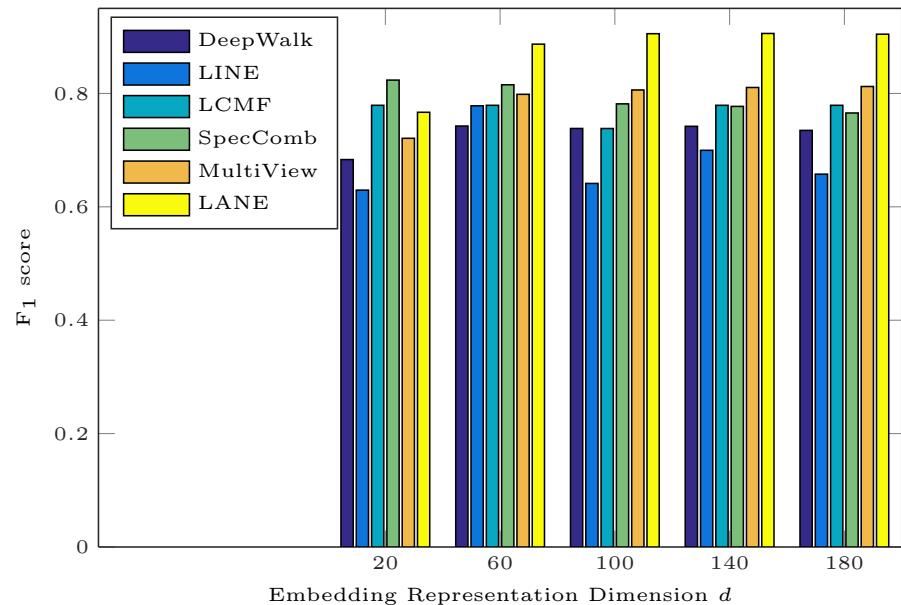
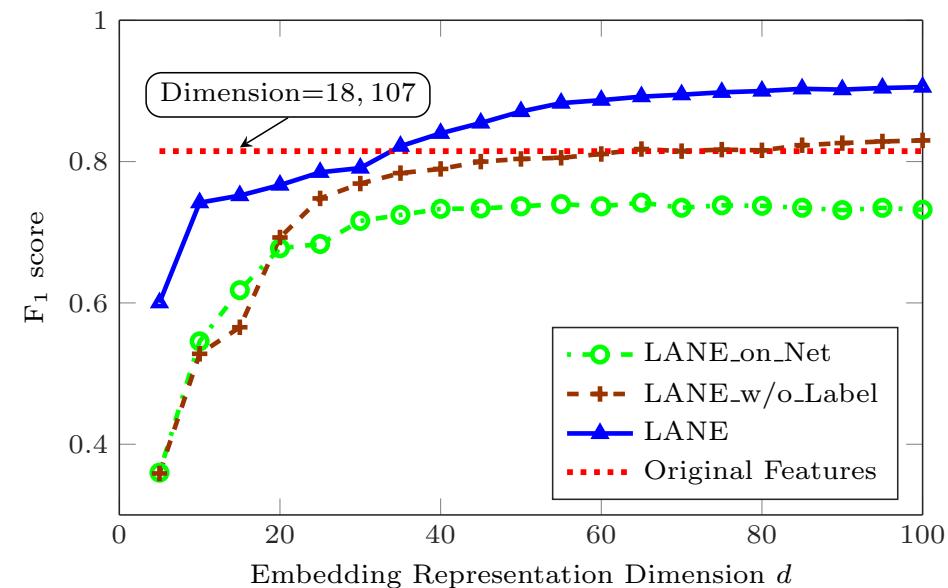
- Consider nodes with the same label as a clique, and employ the learned network proximity to smooth the label information

$$\underset{\mathbf{U}^{(G)}, \mathbf{U}^{(Y)}}{\text{maximize}} \quad \text{Tr} \left(\mathbf{U}^{(Y)\top} (\mathcal{L}^{YY} + \mathbf{U}^{(G)} \mathbf{U}^{(G)\top}) \mathbf{U}^{(Y)} \right)$$

- Uniformly project all of the learned latent representations into \mathbf{H}

$$\underset{\mathbf{U}^{(G)}, \mathbf{U}^{(A)}, \mathbf{U}^{(Y)}, \mathbf{H}}{\text{maximize}} \quad \text{Tr} \left(\mathbf{H}^\top (\mathbf{U}^{(G)} \mathbf{U}^{(G)\top} + \mathbf{U}^{(A)} \mathbf{U}^{(A)\top} + \mathbf{U}^{(Y)} \mathbf{U}^{(Y)\top}) \mathbf{H} \right)$$

Experimental results



- LANE and its variation outperform Original Features
- LANE achieves significantly better performance than the state-of-the-art embedding algorithms

Summary of coupled spectral embedding

- I. Convert node attributes into a network by computing the affinity matrix and couple multiple spectral embedding

- Label informed attributed network embedding, WSDM 2017
 - Co-regularized multi-view spectral clustering, NIPS 2011

$$\underset{\mathbf{U}^{(G)}, \mathbf{U}^{(A)}}{\text{maximize}} \quad \text{Tr}(\mathbf{U}^{(G)^\top} \mathcal{L}^{(G)} \mathbf{U}^{(G)} + \alpha \mathbf{U}^{(A)^\top} \mathcal{L}^{(A)} \mathbf{U}^{(A)} + \alpha \mathbf{U}^{(A)^\top} \mathbf{U}^{(G)} \mathbf{U}^{(G)^\top} \mathbf{U}^{(A)})$$

- ANE for learning in a dynamic environment, CIKM 2017

- Initialization:

$$\underset{\mathbf{p}, \mathbf{q}}{\text{maximize}} \quad \mathbf{p}^\top \mathbf{U}^{(G)^\top} \mathbf{U}^{(G)} \mathbf{p} + \mathbf{p}^\top \mathbf{U}^{(G)^\top} \mathbf{U}^{(A)} \mathbf{q} + \mathbf{q}^\top \mathbf{U}^{(A)^\top} \mathbf{U}^{(G)} \mathbf{p} + \mathbf{q}^\top \mathbf{U}^{(A)^\top} \mathbf{U}^{(A)} \mathbf{q}$$

- Joint representations:

$$\mathbf{H} = [\mathbf{U}^{(G)}, \mathbf{U}^{(A)}] \times [\mathbf{P}, \mathbf{Q}]$$

Summary of coupled spectral embedding

II. Leverage spectral embedding to handle networks and couple with other low-rank approximations, including matrix factorization

- Exploring context and content links in social media, TPAMI 2012

$$\underset{\mathbf{H}}{\text{minimize}} \quad \|\mathbf{A} - \mathbf{H}\|_F^2 + \lambda \text{Trace}[\mathbf{H}^\top (\mathbf{D} - \mathbf{G})\mathbf{H}] + \gamma \|\mathbf{H}\|_*$$

- Attributed signed network embedding, CIKM 2017

- Use spectral embedding to encode node attribute affinity matrix

III. Spectral filters in graph neural networks

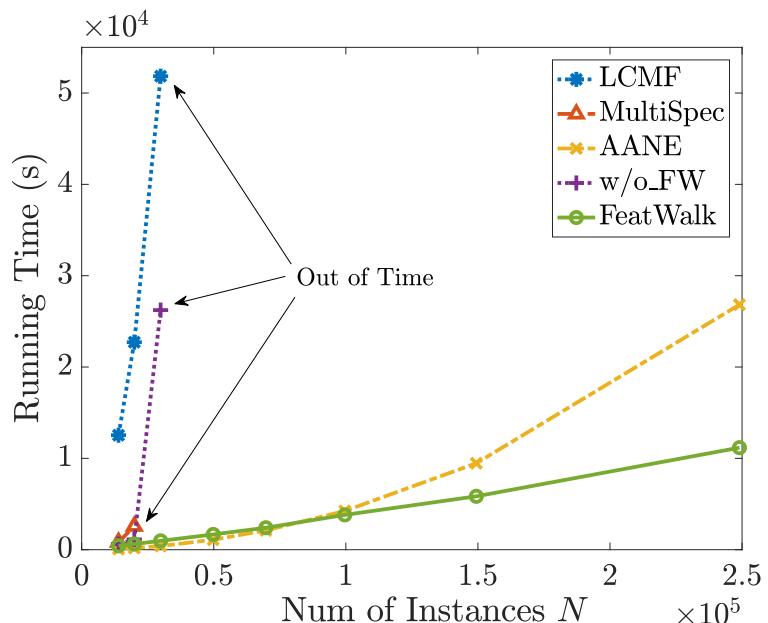
- Eigenvalues & Eigenvectors are identified as the frequencies of graph & graph Fourier modes
- CNN on graphs with fast localized spectral filtering, NIPS 2016
- Semi-supervised classification with graph convolutional networks, 2016
- GCN networks with complex rational spectral filters, 2019

Coupled matrix & tri- factorization

- Learning a unified representation from two matrices is trivial

$$\min_{\mathbf{H}, \mathbf{U}, \mathbf{V}}$$

$$\|\mathbf{G} - \mathbf{H}\mathbf{U}\|_F^2 + \alpha \|\mathbf{A} - \mathbf{H}\mathbf{V}\|_F^2$$



- Intuitive solutions:

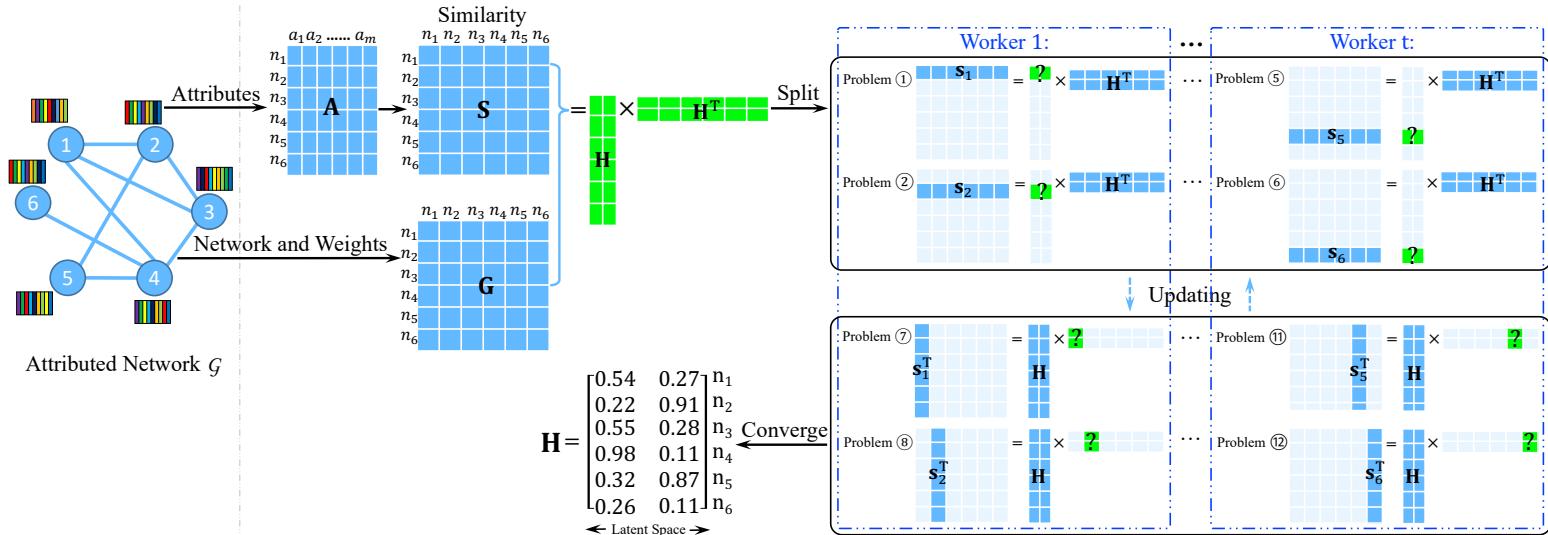
- Combining Content and Link for Classification using Matrix Factorization, 2007 (LCMF)

$$\min_{\mathbf{H}, \mathbf{U}, \mathbf{V}} \|\mathbf{G} - \mathbf{H}\mathbf{U}\mathbf{H}^\top\|_F^2 + \alpha \|\mathbf{A} - \mathbf{H}\mathbf{V}\|_F^2 + \gamma \|\mathbf{U}\|_F^2 + \beta \|\mathbf{V}\|_F^2$$

- Focuses:

- Factorizing networks
- Improving efficiency

Accelerated attributed network embedding



AANE [Huang et al. SDM, 2017]

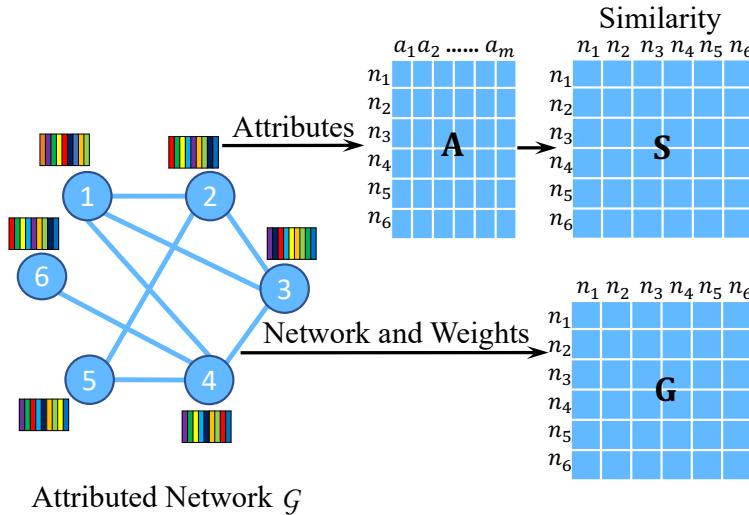
- **Goal:** Preserve the network & node attributes into a unified latent representation, in an efficient way
- AANE accelerates the optimization by decomposing it into low complexity sub-problems

Network structure modeling

- Objective function: $\min_{\mathbf{H}} \quad \mathcal{J} = \|\mathbf{S} - \mathbf{HH}^\top\|_F^2 + \lambda \sum_{(i,j) \in \mathcal{E}} g_{ij} \|\mathbf{h}_i - \mathbf{h}_j\|_2$

Network Lasso
- Network lasso [Hallac et al. KDD, 2015]:
 - If we use squared norms, it would reduce to Laplacian regularization
 - A generalization of group lasso, encouraging $h_i = h_j$ across the edge
 - For each edge i to j , set $\{(h_{i1} - h_{j1}), (h_{i2} - h_{j2}), \dots\}$ as a group
 - Group lasso: $\min_{\boldsymbol{\beta}} \quad \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda \sum_{\mathcal{I}=1, \dots, I} \|\boldsymbol{\beta}_{\mathcal{I}}\|_2$
- λ adjusts the size of clustering groups
- ℓ_2 -norm alleviates the impacts from outliers and missing data

Incorporating node attribute affinities



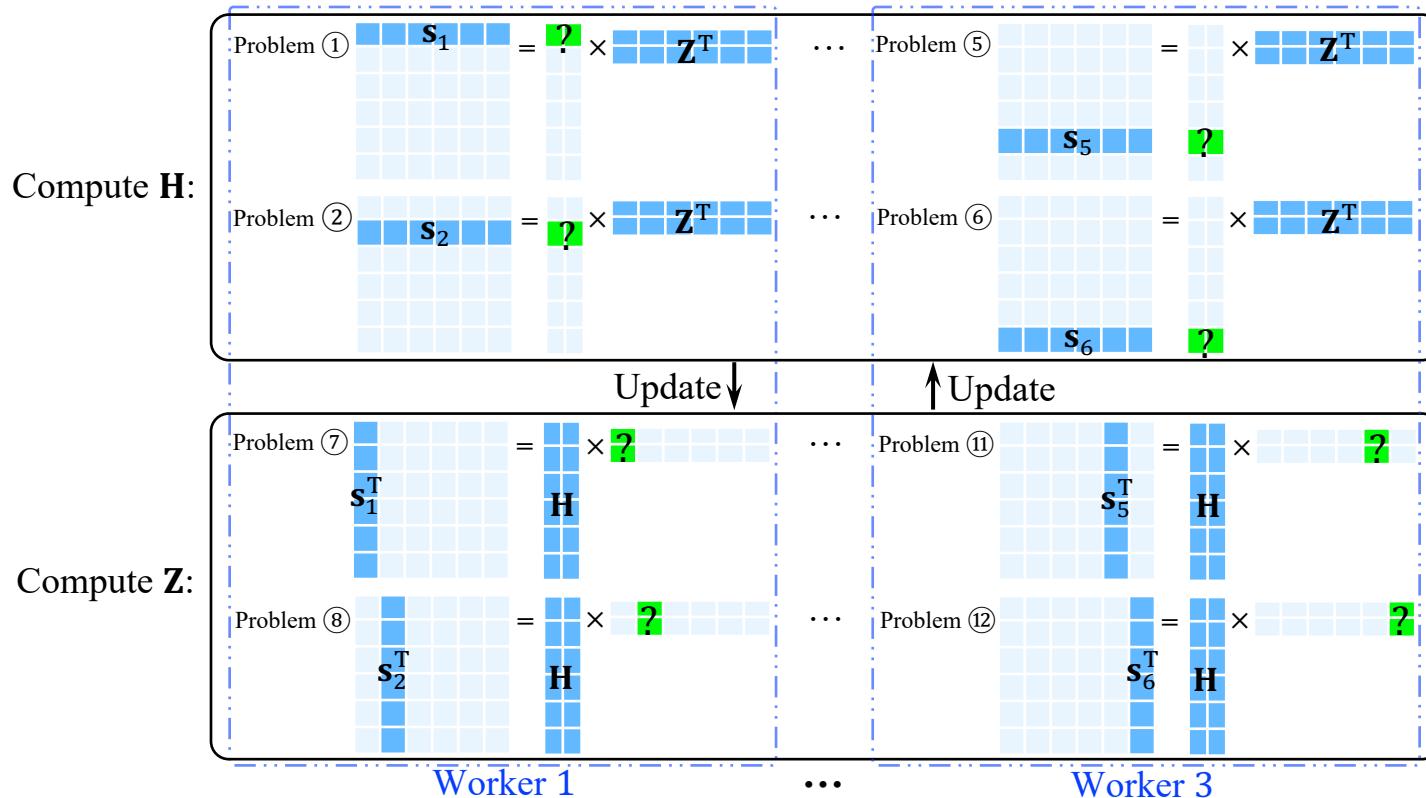
Objective functions:

$$\min_{\mathbf{H}} \quad \mathcal{J} = \|\mathbf{S} - \mathbf{H}\mathbf{H}^\top\|_F^2 + \lambda \sum_{(i,j) \in \mathcal{E}} g_{ij} \|\mathbf{h}_i - \mathbf{h}_j\|_2$$

Network Lasso

- Though network & node attributes are heterogeneous info, node proximity defined by attributes is homogenous with network
- Based on the decomposition of similarities defined by attributes and penalty of embedding difference between connected nodes

Acceleration via distributed optimization



- Make sub-problems independent to each other to allow parallel computation

Low-complexity independent sub-problems

- Make a copy of \mathbf{H} , named \mathbf{Z}
- Reformulate objective function into a linearly constrained problem

$$\min_{\mathbf{H}} \quad \sum_{i=1}^n \|\mathbf{s}_i - \mathbf{h}_i \mathbf{Z}^\top\|_2^2 + \lambda \sum_{(i,j) \in \mathcal{E}} g_{ij} \|\mathbf{h}_i - \mathbf{z}_j\|_2,$$

subject to $\mathbf{h}_i = \mathbf{z}_i, \quad i = 1, \dots, n.$

- Given fixed \mathbf{H} , all the row \mathbf{z}_i could be calculated independently
- Each sub-problem only needs row \mathbf{s}_i , not the entire \mathbf{S}
- Time complexity of updating \mathbf{h}_i is $\mathcal{O}(d^3 + dn + d|N(i)|)$, with space complexity $\mathcal{O}(n)$

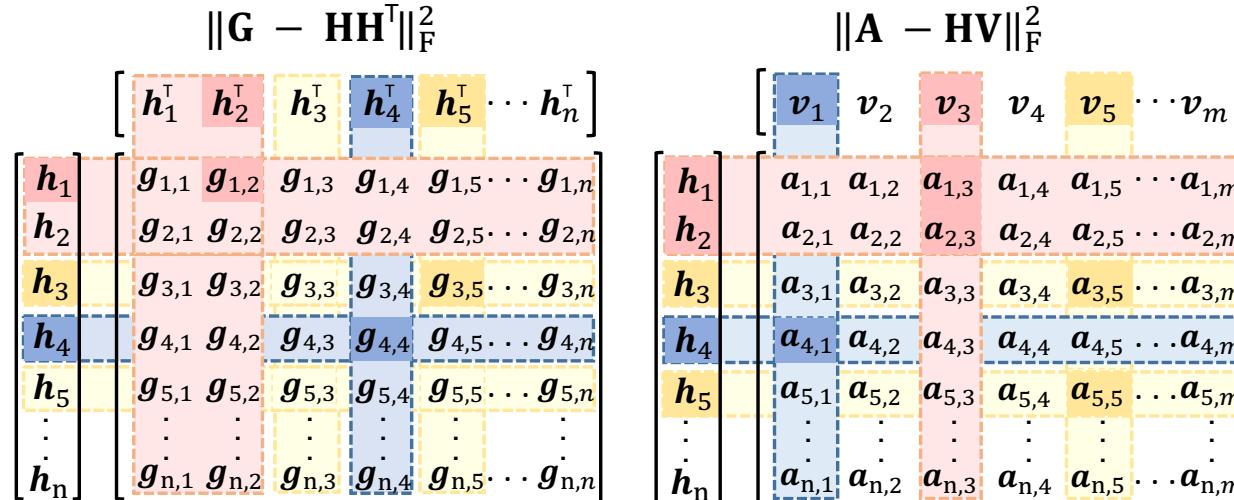
Summary of coupled matrix & tri- factorization

I. Accelerate coupled matrix factorization via distributed optimizations

- Accelerated attributed network embedding, SDM 2017
- Accelerated local anomaly detection via resolving AN, IJCAI 2017

■ $\min_{\mathbf{H}, \mathbf{V}} \|\mathbf{G} - \mathbf{H}\mathbf{H}^\top\|_F^2 + \alpha \|\mathbf{A} - \mathbf{H}\mathbf{V}\|_F^2 + \gamma(\|\mathbf{H}\|_F^2 + \|\mathbf{V}\|_F^2)$

■ A parallel mini-batch SGD to accelerate the optimization



Summary of coupled matrix & tri-factorization

II. Modeling networks via matrix tri-factorization

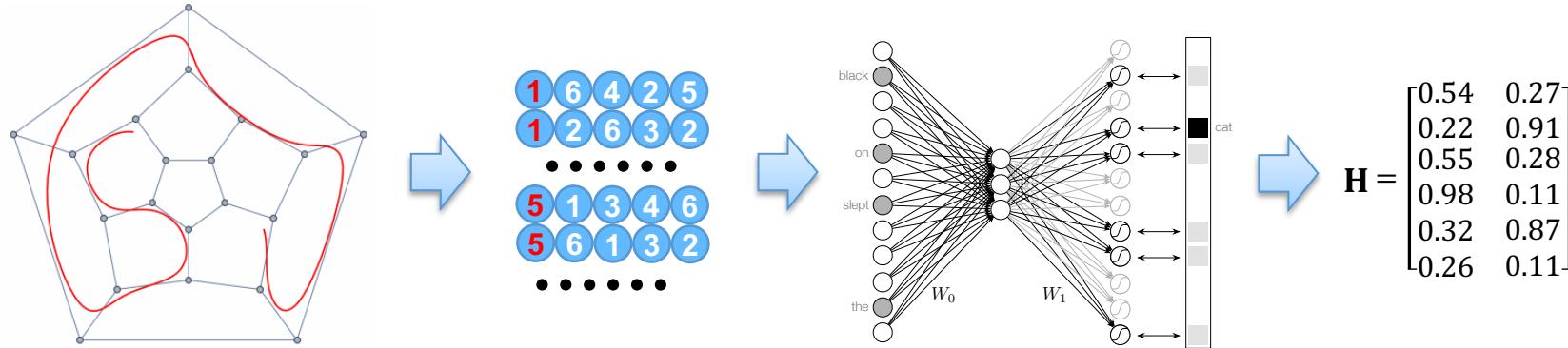
- Network Representation Learning with Rich Text Information, IJCAI 2015
 - Let \mathbf{T} be the transition matrix of the PageRank on \mathbf{G} , and $\mathbf{M} = (\mathbf{T} + \mathbf{T}^2)/2$
 - $\min_{\mathbf{H}, \mathbf{V}} \|\mathbf{M} - \mathbf{H}\mathbf{V}\mathbf{A}^\top\|_F^2 + \frac{\lambda}{2}(\|\mathbf{H}\|_F^2 + \|\mathbf{V}\|_F^2)$
- Preserving Proximity and Global Ranking for Network Embedding, 2017
 - **Lemma:** Matrix tri-factorization $\mathbf{H}^\top \mathbf{V} \mathbf{H} \approx \mathbf{M}^{\text{PMI}}$ preserves the second-order proximity, where (shifted) pointwise mutual information is defined as follows

$$\mathbf{M}^{\text{PMI}} = \begin{cases} \max\{0, \log \frac{p_{s,t}(i,j)}{p_s(i)p_t(j)} - \log \alpha\}, & \text{if } (i, j) \in \mathcal{E} \\ 0, & \text{otherwise} \end{cases}$$

- $p_{s,t}(i, j) = \frac{1}{|\mathcal{E}|}$, $p_s(i) = \frac{\text{degree}_{\text{out}}^i}{|\mathcal{E}|}$, $p_t(j) = \frac{\text{degree}_{\text{in}}^j}{|\mathcal{E}|}$

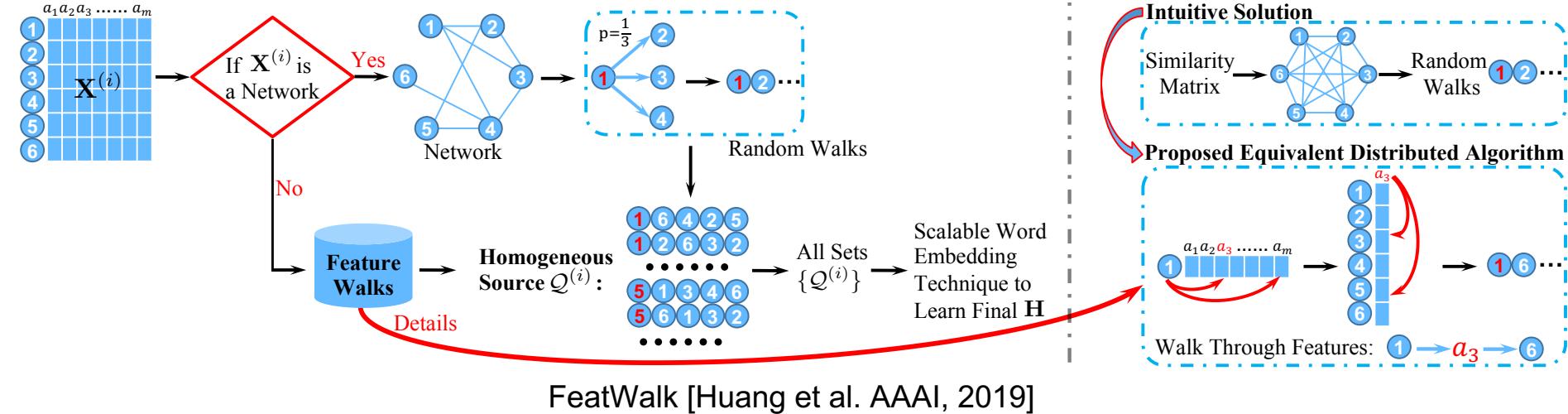
- Negative values are filtered since less informative [Levy and Goldberg, 2014]

Random walk based embedding



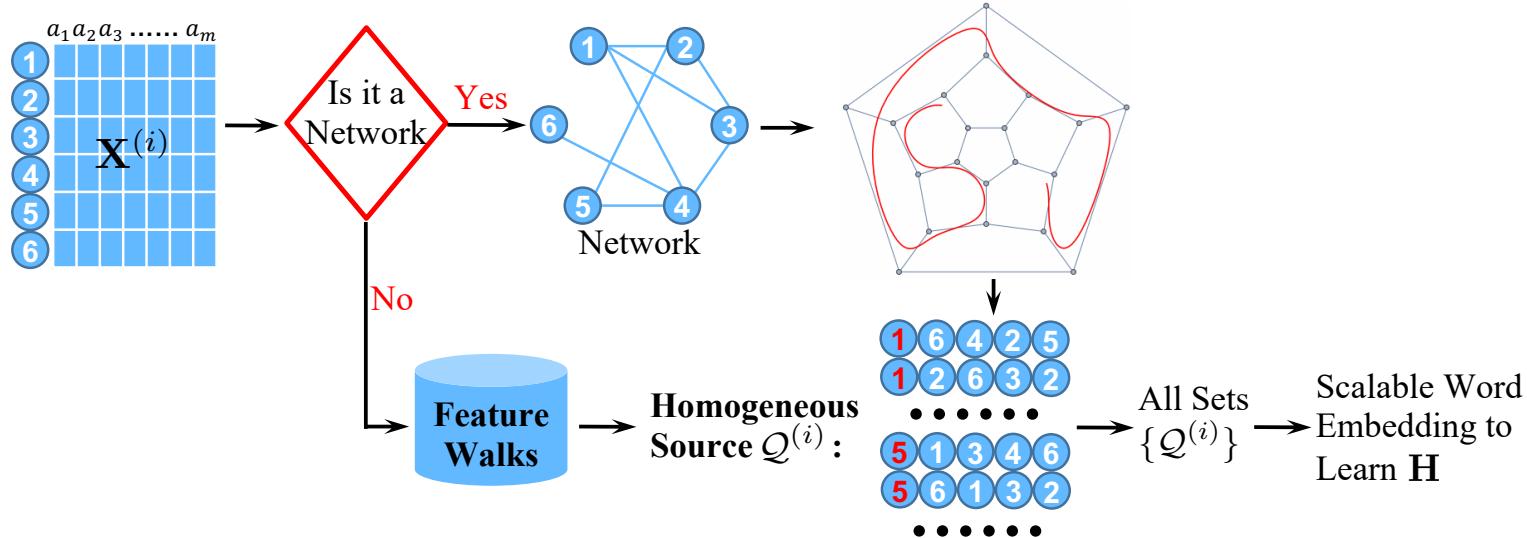
- Random walks on plain networks:
 - Conduct random walks on a network and record the walking trajectories
 - Treat nodes as words and sequences as sentences to learn embedding
- Nodes' co-occurrence probabilities \approx linking probabilities
- It converts geometric structures into structured sequences while alleviating the issues of sparsity and curse of dimensionality
- Random walks on attributed networks? (Heterogeneity)

Large-scale heterogeneous feature embedding



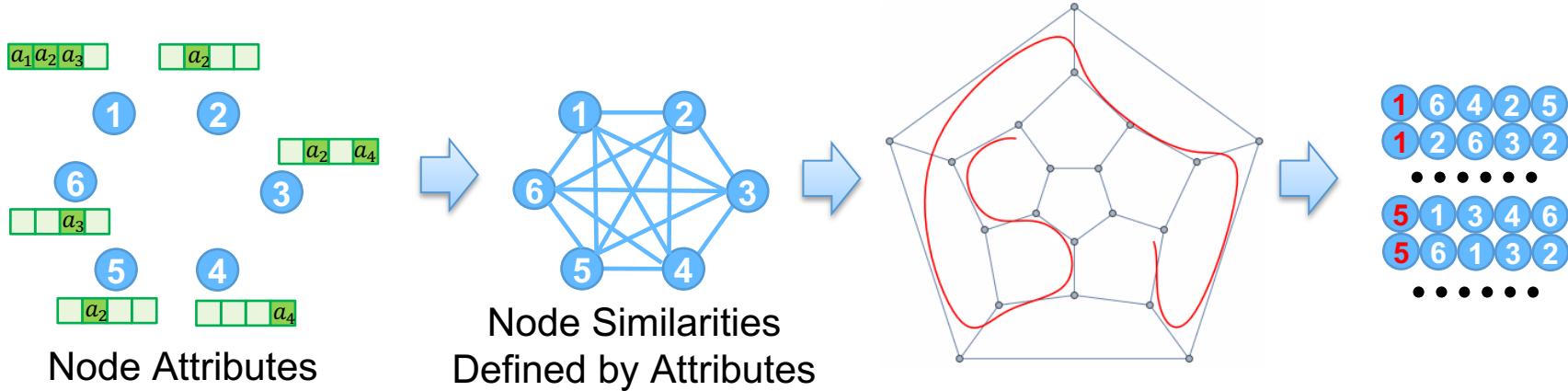
- **Goal:** Incorporate multiple networks & multiple types of high-dimensional node attributes into a unified latent representation
- E.g., amazon products have product info, customer reviews, etc.
Networks: customer purchase record, & customer viewing history

Learn node proximities to handle heterogeneity



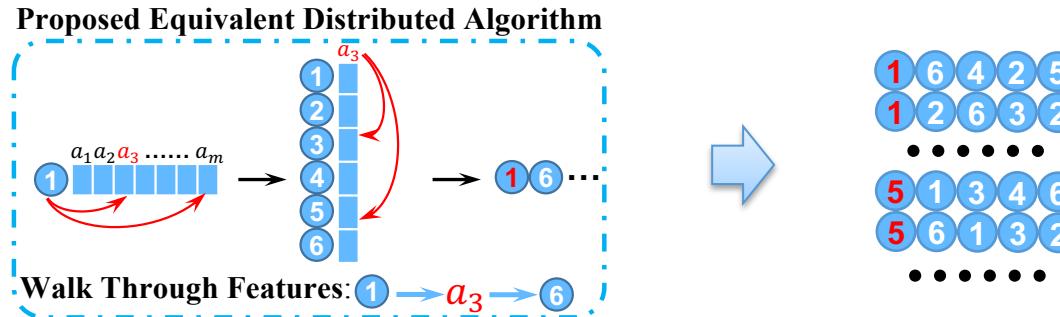
- **Node proximity:** Similarities between nodes defined by links or attributes of nodes, i.e., rows of each $\mathbf{X}^{(i)}$
- Node proximities learned from different $\{\mathbf{X}^{(i)}\}$ are homogeneous
- FeatWalk projects each node proximity into a set of node sequences $\mathcal{Q}^{(i)}$, and learns \mathbf{H} from all $\{\mathcal{Q}^{(i)}\}$

The intuitive solution



- To learn $\mathcal{Q}^{(i)}$, intuitive solution is to compute node similarity matrix S based on $A^{(i)}$, and perform random walks on S
- Random Walks: In $\mathcal{Q}^{(i)}$, a sequence of node indices, probability of i follows j approaches their similarity in S
- Expensive: S is dense with $n \times n$ dimensions

Equivalent similarity-based random walks



- **Theorem 1.** Probability of walking from i to j via FeatWalk is equal to the one via random walks on \mathbf{S} , where
$$\mathbf{S} = \mathbf{YDT}^\top$$
- \mathbf{Y} is the node attribute matrix after special normalizations
- FeatWalk learns the same sequences as the intuitive solution, while avoiding the computation of node similarities \mathbf{S}

FeatWalk walks via features

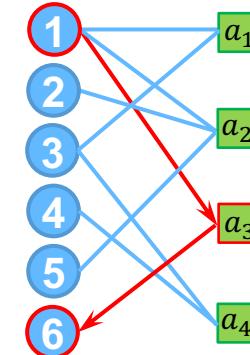
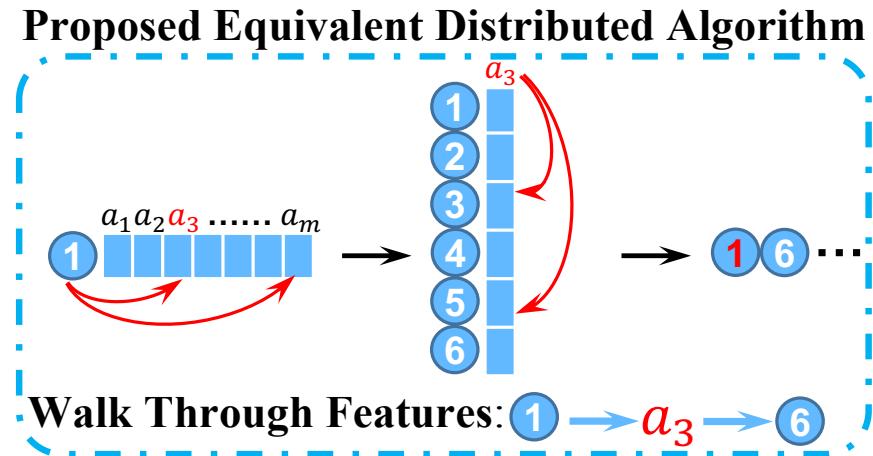
- Given the initial i , we walk to the m^{th} attribute category with probability

$$P(i \rightarrow a_m) = \frac{\hat{x}_{im}}{\sum_{p=1}^M \hat{x}_{ip}}$$

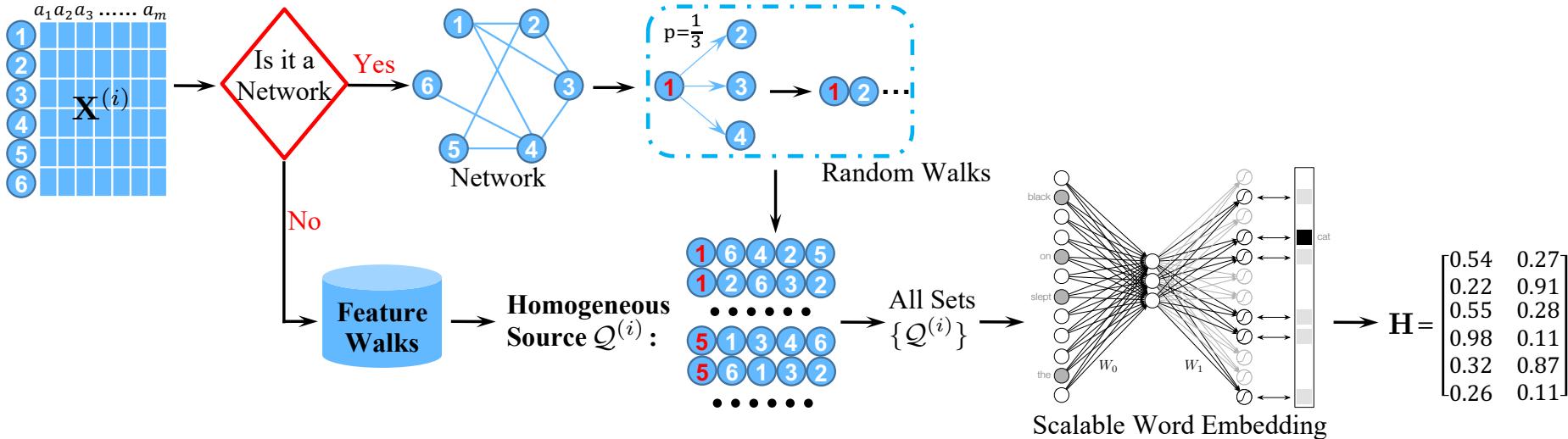
- We focus on the m^{th} attribute category and walk from a_m to j with probability

$$P(a_m \rightarrow j) = \frac{y_{jm}}{\sum_{n=1}^N y_{nm}}$$

- \hat{x}_{im} and y_{jm} are normalized node attributes

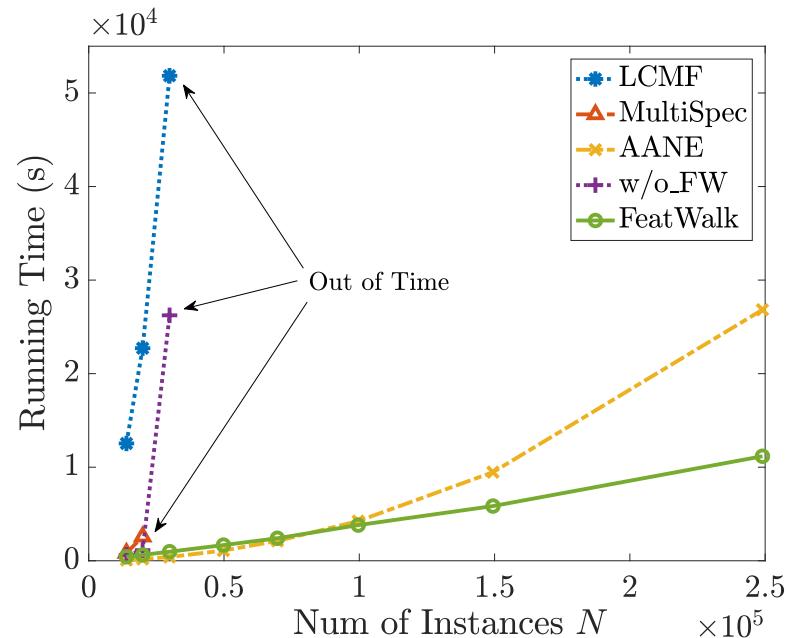
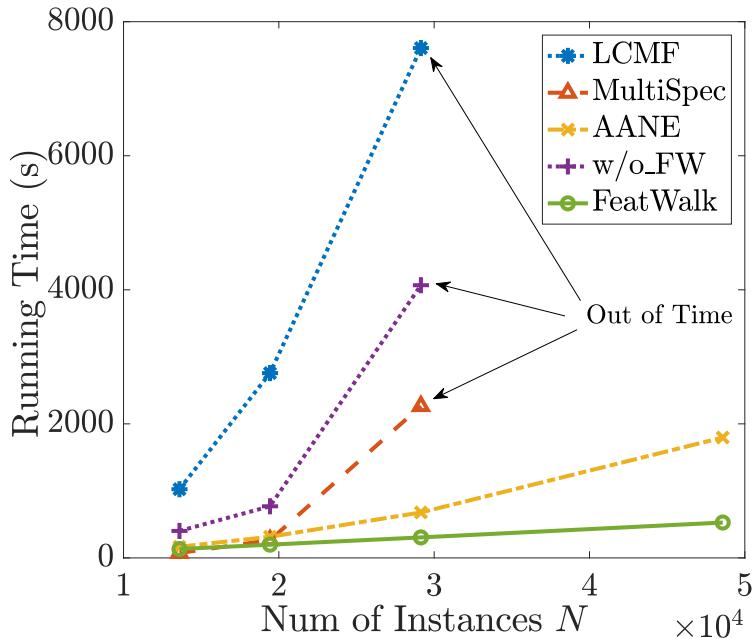


Summary of FeatWalk



- Project each node proximity into a set of node sequence $\mathcal{Q}^{(i)}$
- Consider nodes as words and truncated sequences as sentences
- Apply a scalable word embedding technique to all $\{\mathcal{Q}^{(i)}\}$ to learn a joint embedding representation \mathbf{H}

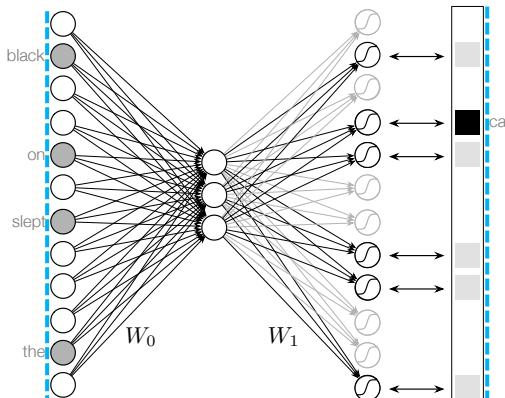
Efficiency evaluation



- Running time of FeatWalk is almost linear to N
- FeatWalk achieves a significant acceleration compared to the intuitive solution w/o_FW

Summary of random walk based embedding

Skip-Gram Model & Negative Sampling



Word2vec: Distributed Representations of Words and Phrases and their Compositionality
DeepWalk: Online Learning of Social Representations
FeatWalk: Large-Scale Heterogeneous Feature Embedding
TriDNR: Tri-Party Deep Network Representation
Gat2vec: Representation Learning for Attributed Graphs

Word2vec: words → surrounding words [2013]

DeepWalk: nodes → neighbors [2014]

FeatWalk: nodes → neighbors defined by edges
nodes with same attributes [2019]

TriDNR: nodes
Gat2vec: attributes → neighbors defined by edges
nodes with same attributes
nodes with same labels [2016]
[2019] 38

Mining attributed networks with shallow embedding

- **Focuses:**

Joint learning, embedding networks, & accelerating optimization

- **Methods:**

Coupled spectral embedding

Coupled matrix & tri-factorization

Random walk based embedding

- **Techniques:**

Spectral graph theory, Coupling,
distributed optimization, joint
random walks, etc.

