Representation Learning on Networks

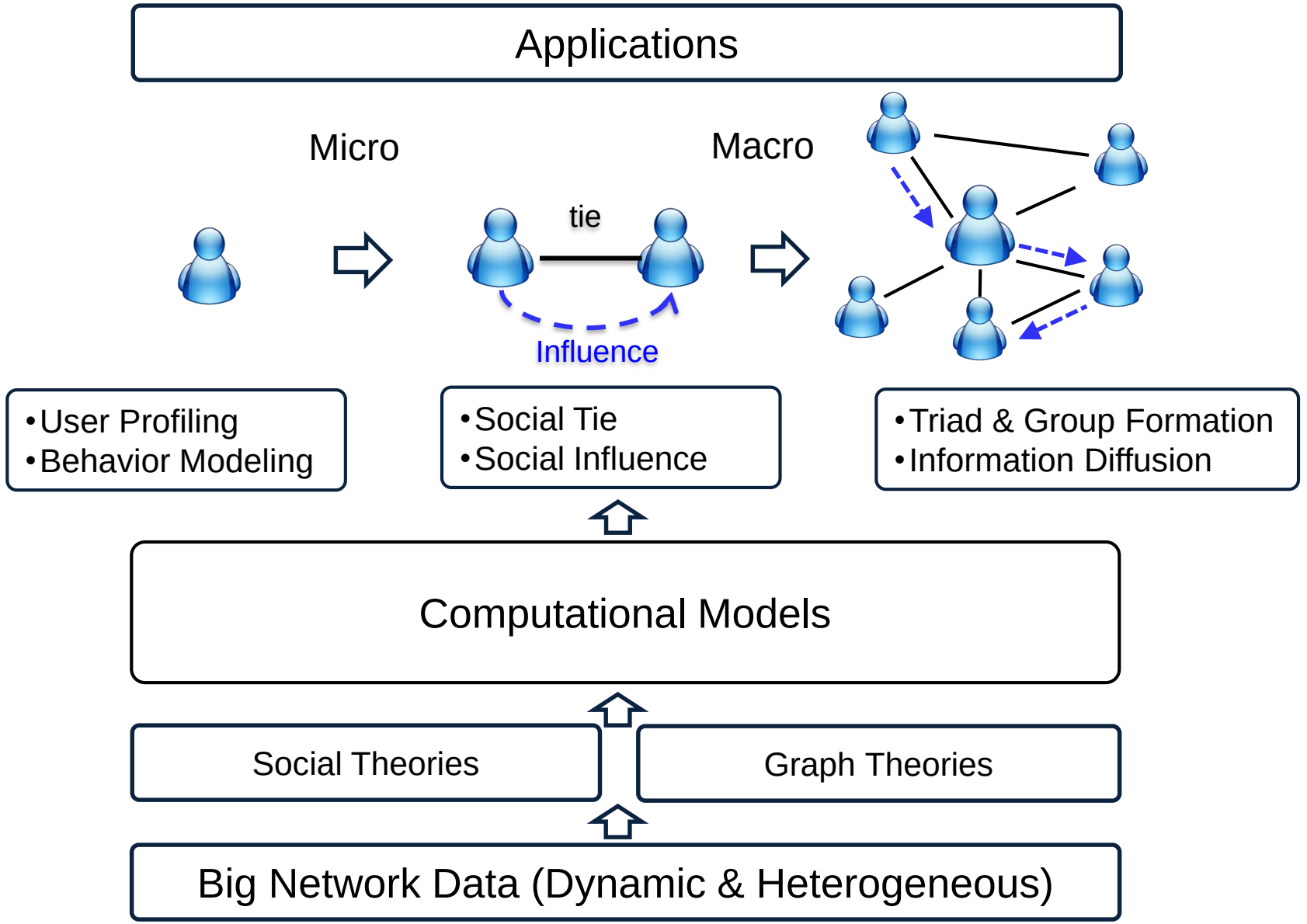
Algorithms, Theory, and Applications

Jie Tang
Tsinghua University

Yuxiao Dong
Microsoft Research, Redmond

Microsoft Research

Social & Information Network Analysis

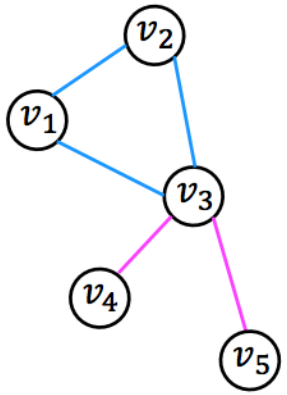


Representation Learning on Networks

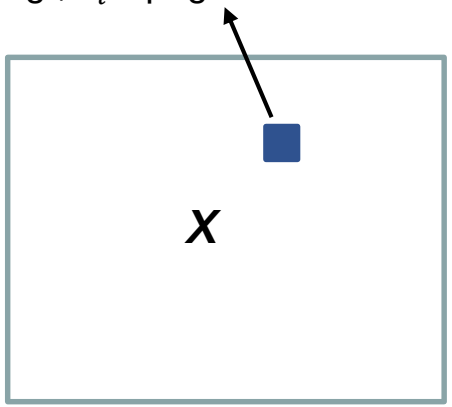
- **The first part:**
  - Conventional network analysis
    - Node classification
    - Social tie & link prediction
  - Network embeddings
    - Embedding models
    - Theoretical understanding
    - Large-scale embedding

- **The second part:**
  - Graph neural networks
    - Graph convolution
    - Graph GAN
    - Dynamic Representation
    - Heterogeneous Representation
  - Large-scale applications
    - Knowledge graph linking
    - Recommendation in E-commerce
    - Online-to-offline recommendations
    - Social influence in gaming

The network analysis paradigm



x_{ij}: node v_i's j^{th} feature,
e.g., v_i's pagerank value



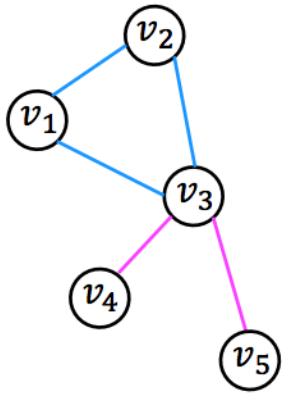
hand-crafted feature matrix

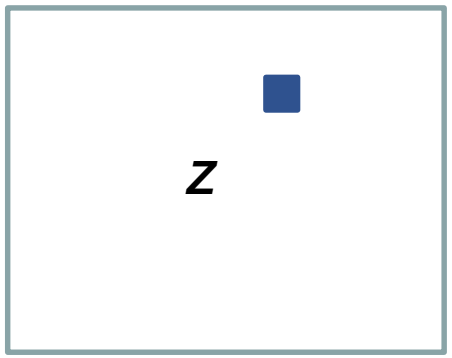
Graph & network applications
- Node label inference;
- Link prediction;
- User behavior… …

feature **engineering**

machine learning models

Representation learning for networks?





~~hand-crafted~~ **latent** feature matrix

Graph & network applications
- Node label inference;
- Node clustering;
- Link prediction;
- … …

Feature ~~engineering~~ **learning**

machine learning models

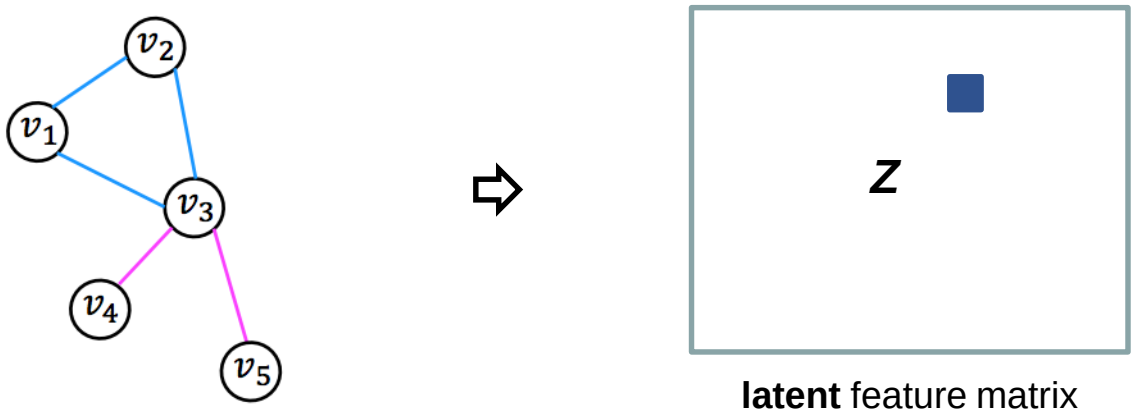
- Bengio, Courville, Vincent. Representation learning: A review and new perspectives. *IEEE TPAMI* 2013.
- LeCun, Bengio, Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

Representation learning for networks

Problem (Graph representation learning, network embedding, graph embedding)

- Input: a network $G = (V, E)$
- Output: $\boldsymbol{Z} \in R^{|V| \times k}, k \ll |V|$, k-dim vector $\boldsymbol{Z}_v$ for each node *v*.

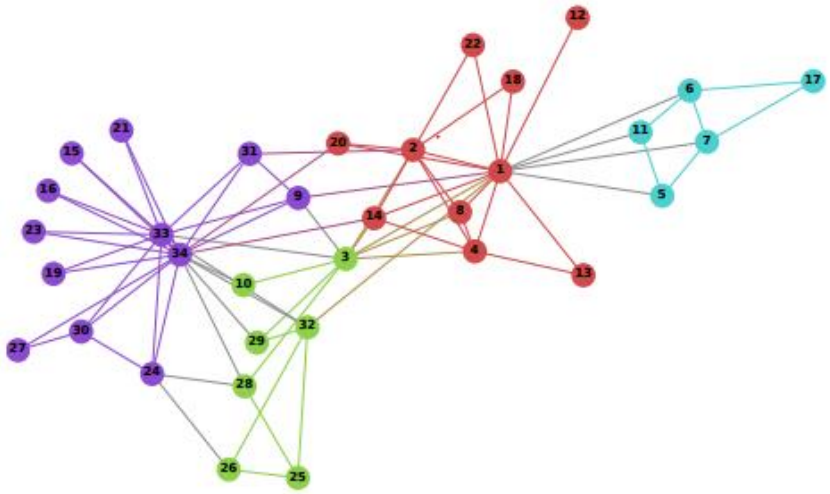
The goal is to map each node into a latent low-dimension space such that network structure information is encoded into distributed node representations



latent feature matrix

Feature **learning**

Graph representation learning



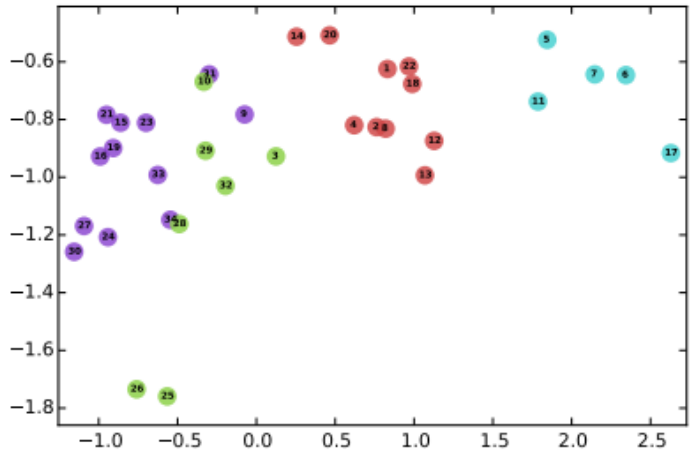
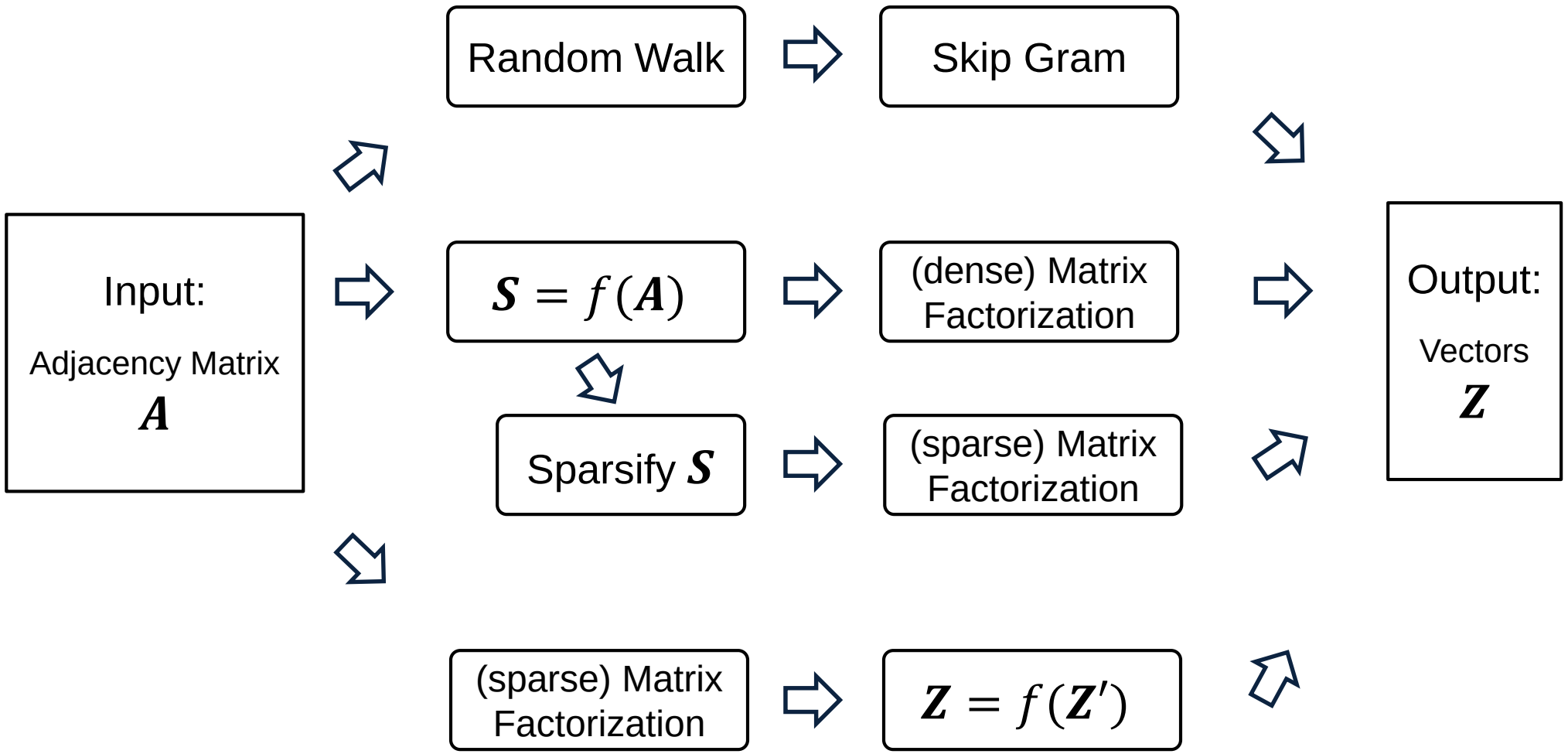
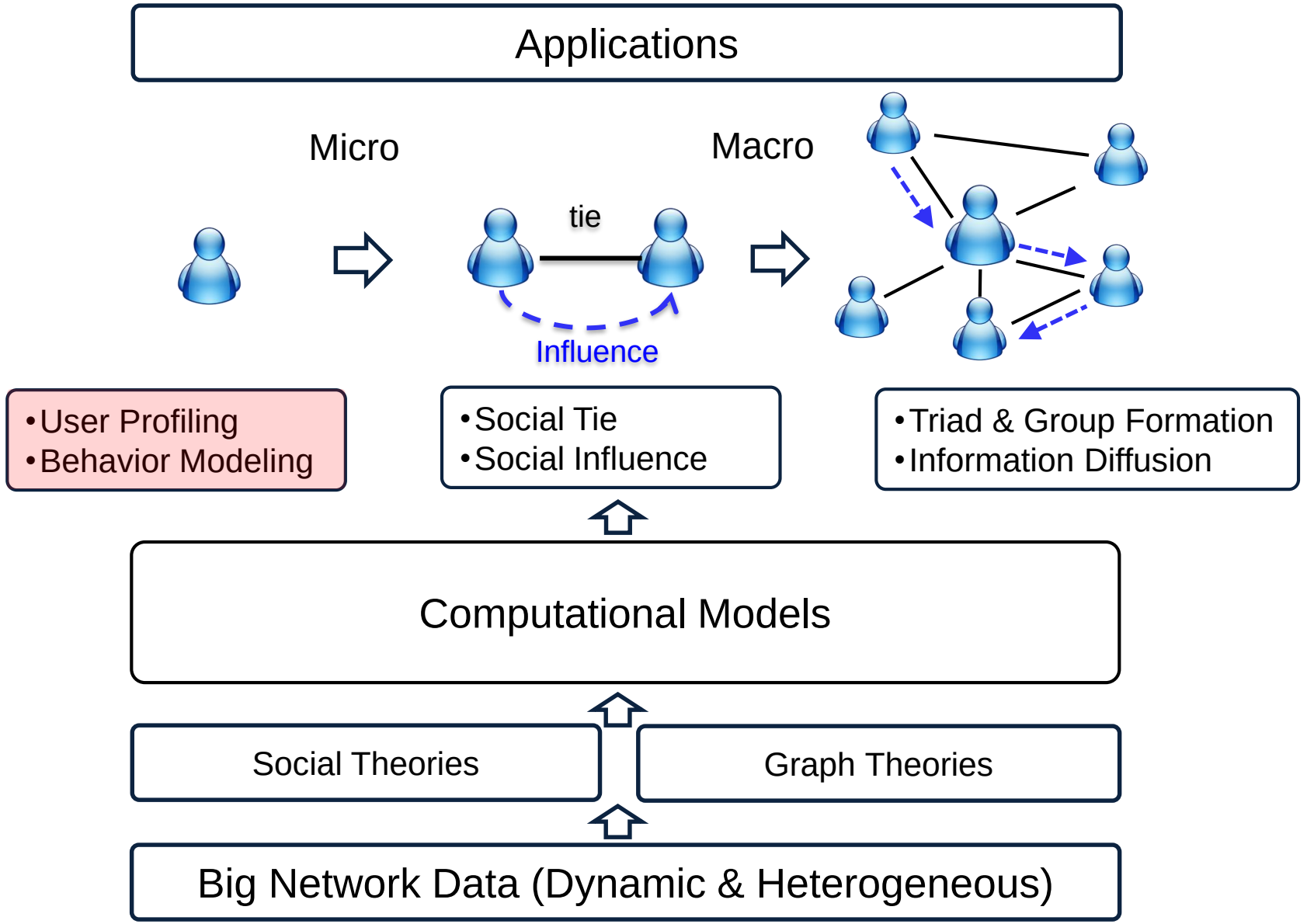


Image Credit: Perozzi et al. DeepWalk: Online learning of social representations. In *KDD' 14*, pp. 701–710.

Network Embedding



Social & Information Network Analysis



Node Classification

Inferring user demographics in social networks

Dong et al. Inferring user demographics and social strategies in mobile social networks. In *KDD 2014*.

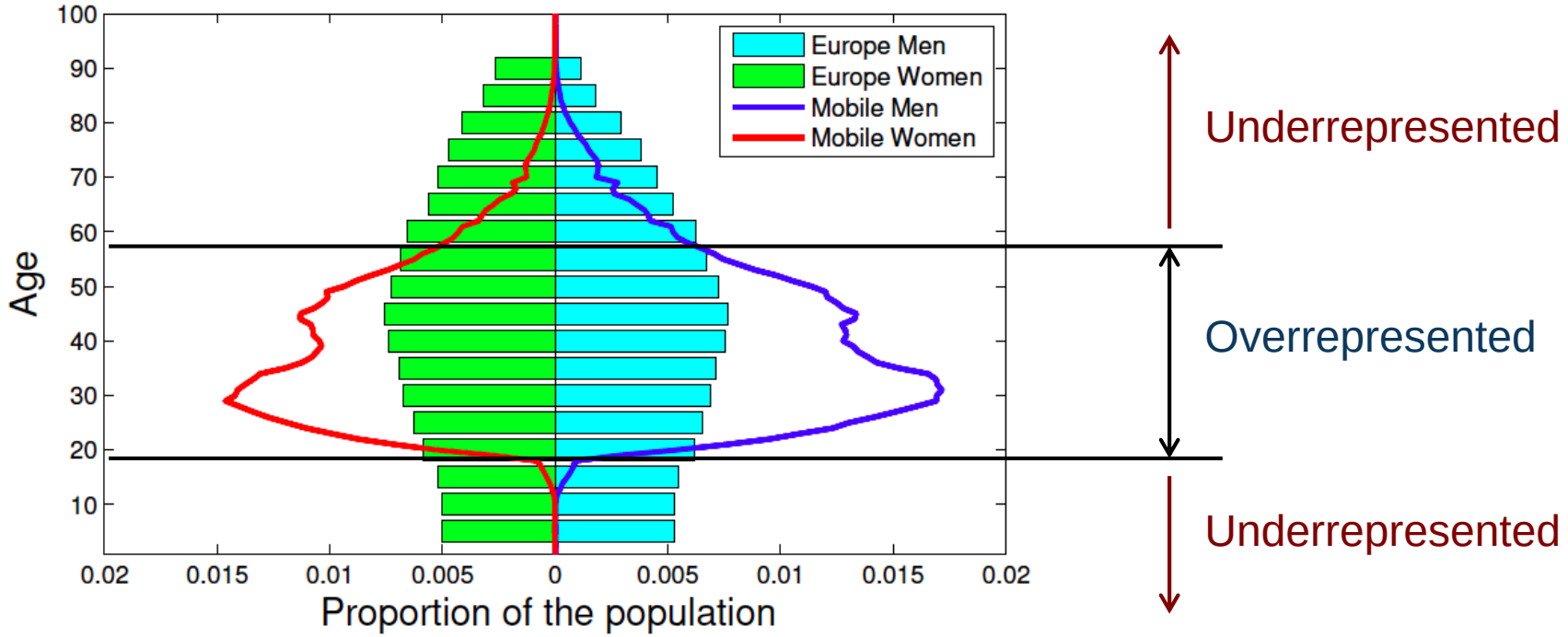
User Profiling on Demographics

Knowledge discovery & feature engineering

How do people of different gender and age connect & interact with each other?

Big Mobile Network Data

- A **nation-wide** large mobile communication data
  - Over 7 million users: male 55% / Female 45%
  - Over 1 billion call & message records between Aug. and Sep. 2008
  - Reciprocal, undirected, and weighted networks: CALL & SMS

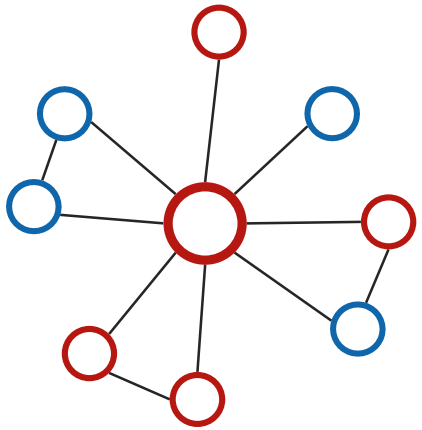


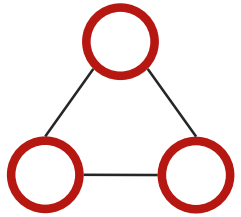
Europe and Mobile (CALL) population pyramids.

Structural Features

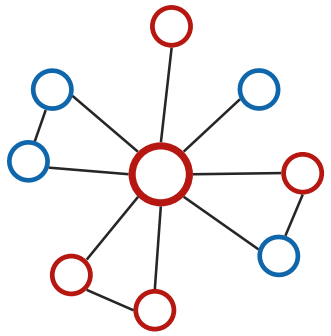
- Given one node *v* and its ego network:
  - Individual feature:
    - Individual attribute: degree, neighbor connectivity, clustering coefficient, embeddedness and weighted degree.
  - Link feature:
    - Friend attribute: # of connections to female/male, young/young-adult/middle-age/senior friends (from labeled friends).
    - Dyadic factor: both labeled and unlabeled friends for social tie structures in v's ego network.
  - Circle feature:
    - Circle attribute: # of demographic triads, i.e., v-FF, v-FM, v-MM; v-AA, v-AB, v-AC, v-AD, v-BB, v-BC, v-BD, v-CC, v-CD, v-DD. (A/B/C/C denote the young/young-adult/middle-age/senior)
    - Triadic factor: both labeled and unlabeled friends for social triad structures in v's ego network.
- LCR/SVM/NB/RF/Bag/RBF:
  - Individual/Friend/Circle Attributes
- FGM/DFG
  - Individual/Friend/Circle Attributes
  - Structure feature: Dyadic factors
  - Structure feature: Triadic factors

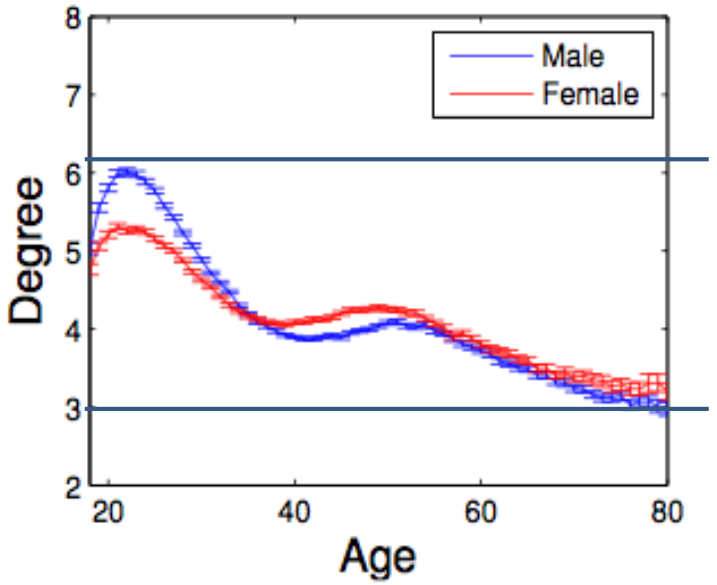
Micro: Ego, Social Tie, & Triad



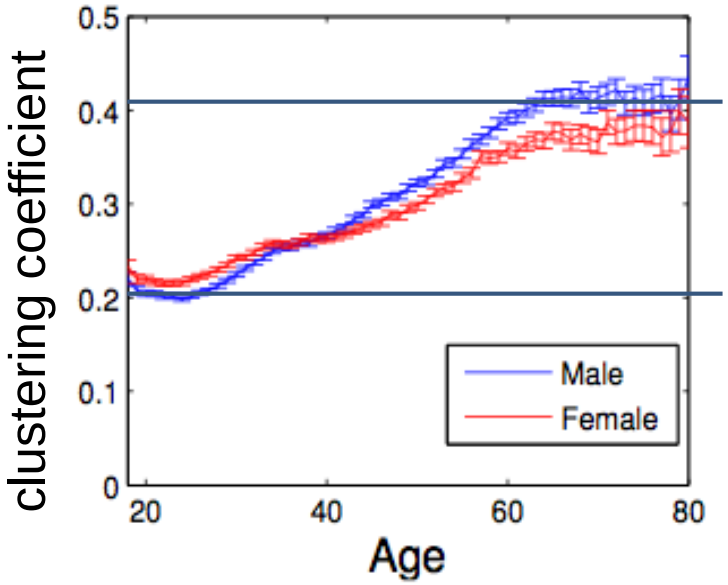


Ego Networks





(a) Degree Centrality

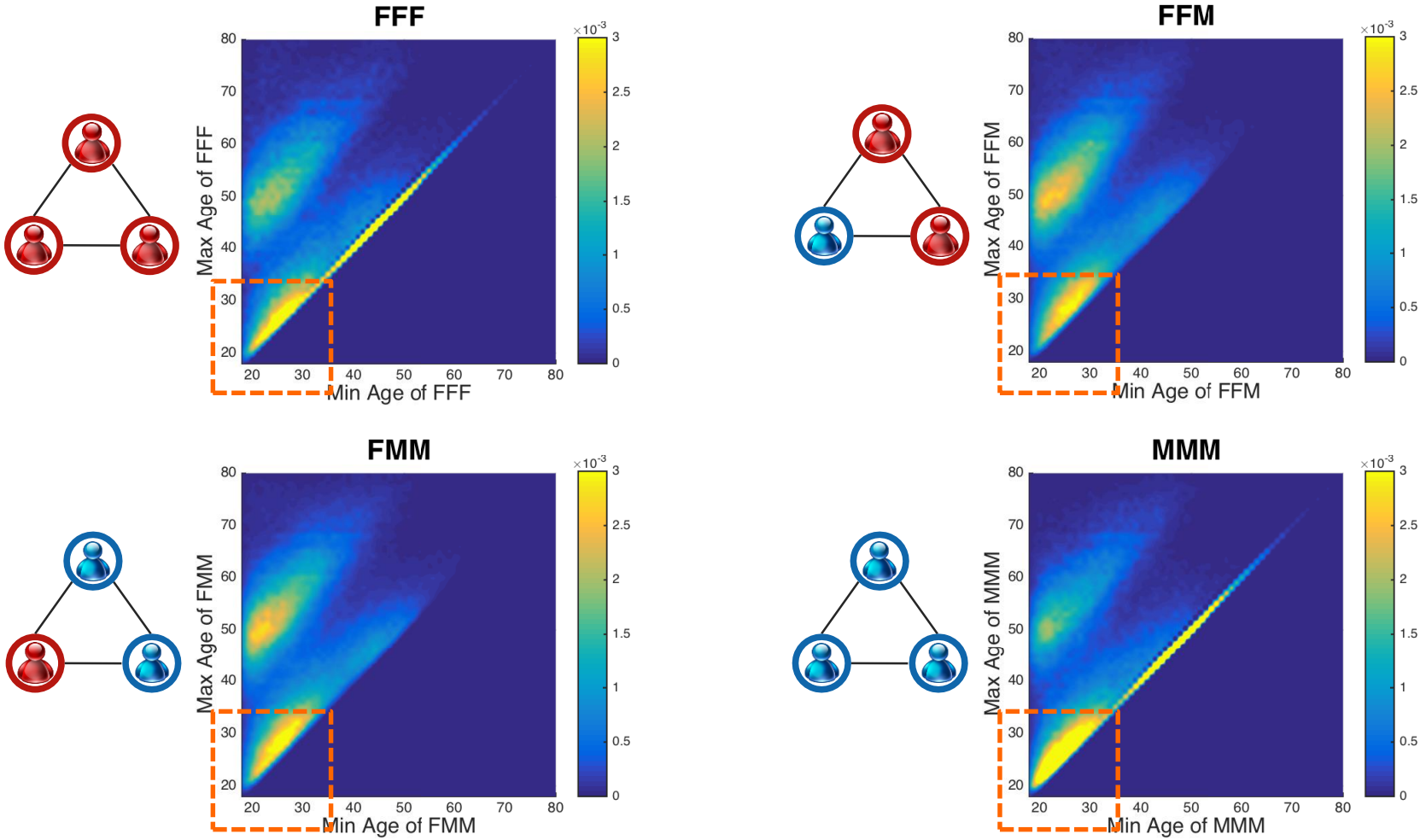


(b) Triadic Closure

- Younger people are active in broadening their social circles, while older people tend to maintain smaller but more closed connections.

Results in the CALL network, and similar observations are also found from SMS.

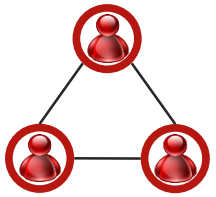
How many different triadic social circles do we have?

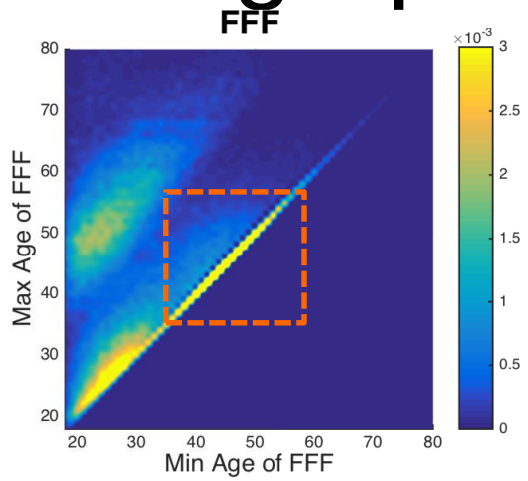


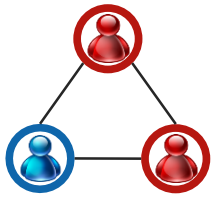
- People expand both same-gender and opposite-gender social groups.

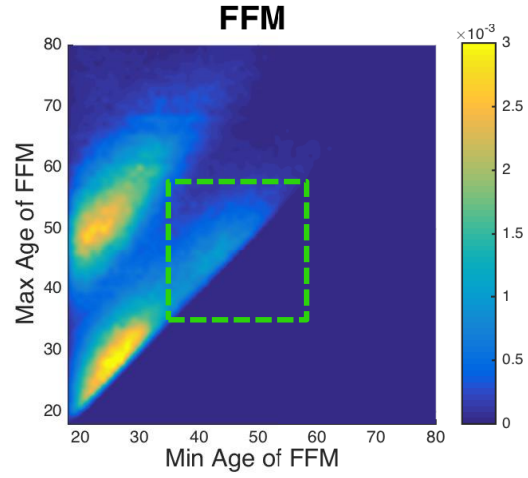
Results in the CALL network, and similar observations are also found from SMS.

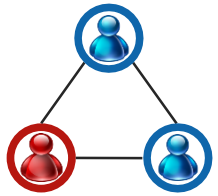
Demographic Triad Distribution

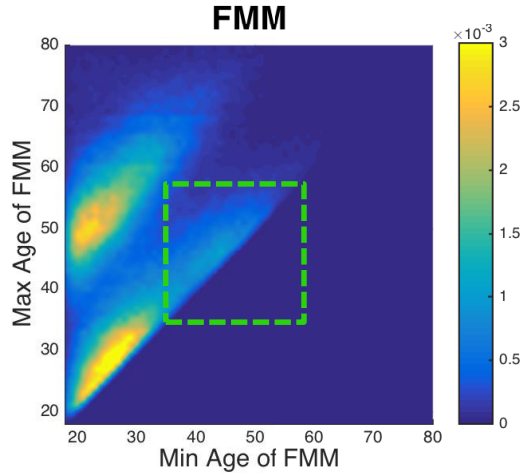


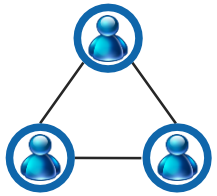


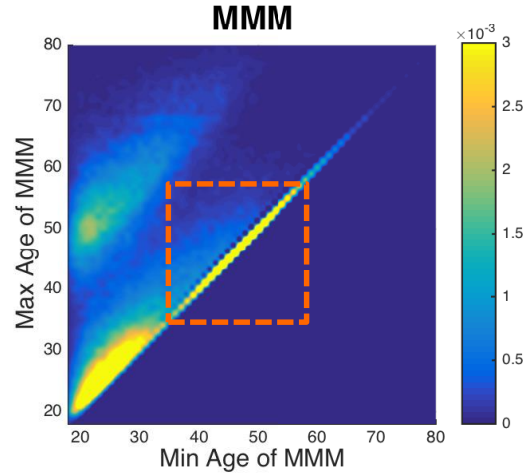












VS.

- ♣ The opposite-gender social groups disappear.
- ♣ The same-gender social groups last for a lifetime.

Results in the CALL network, and similar observations are also found from SMS.

Null Model

- Users' gender and age are randomly shuffled
- Randomly shuffle 10,000 times

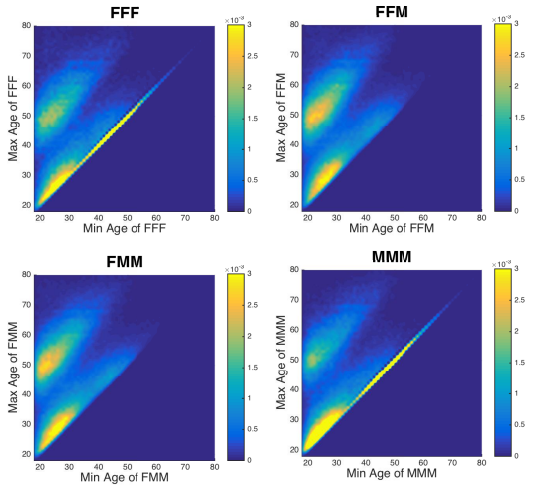
- x: empirical result from real data
- $\tilde{x}$: shuffled results
- $\mu(\tilde{x})$: the average of shuffled data
- $\sigma(\tilde{x})$: the standard deviation of shuffled data

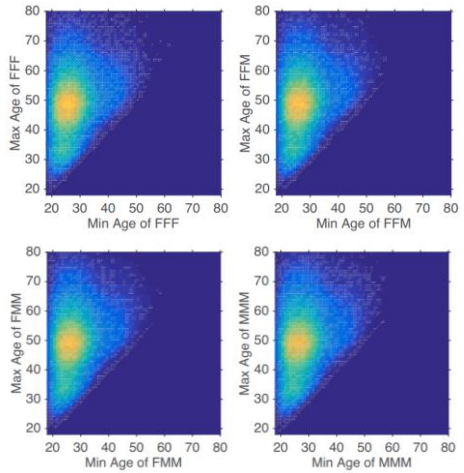
- $z(x)$: *z-score* $\quad z(x) = \dfrac{x - \mu(\tilde{x})}{\sigma(\tilde{x})}$

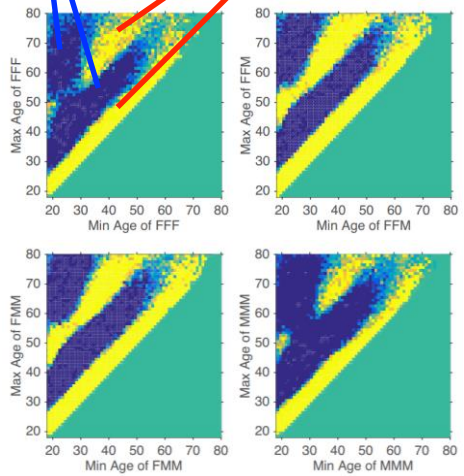
Demographic Triad Distribution

$z < -3.3$
underrepresented

$z > 3.3$
overrepresented





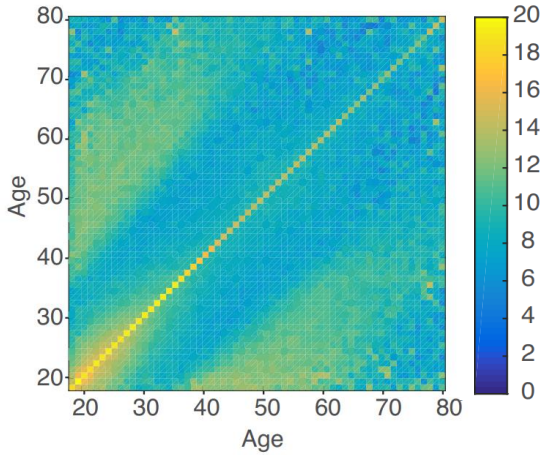


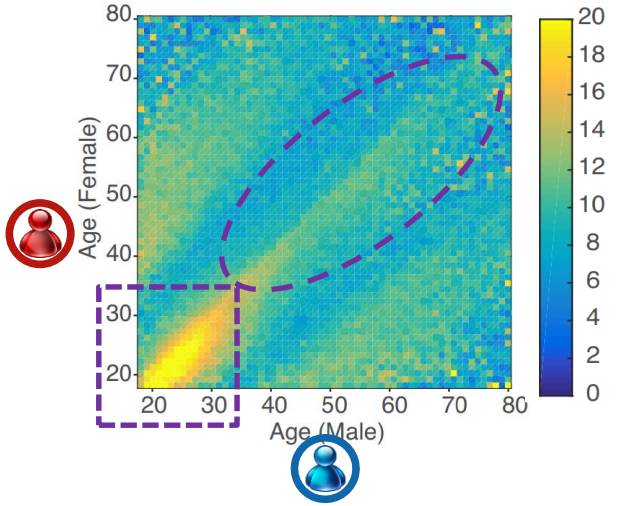
- x: empirical result from **real** data
- $\mu(\tilde{x})$: the average of **shuffled** data
- $z(x)$: *z-score*

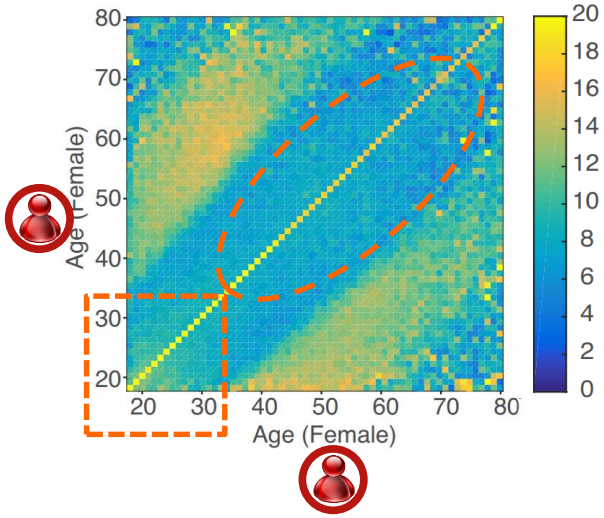
- The results are statistically significant

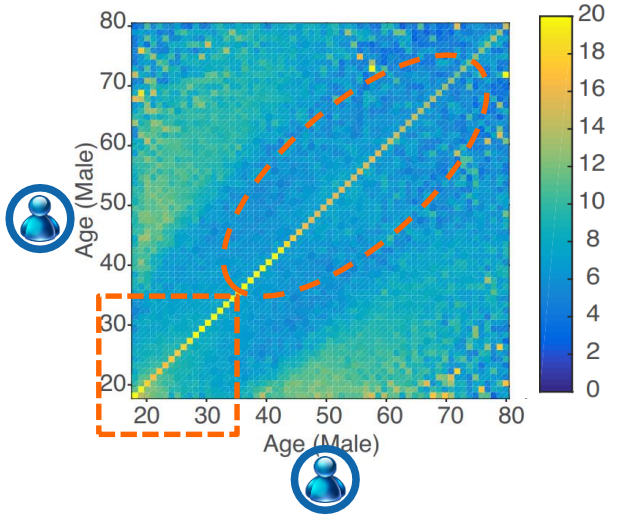
Results in the CALL network, and similar observations are also found from SMS.

How frequently do you call your mom vs. your significant other?









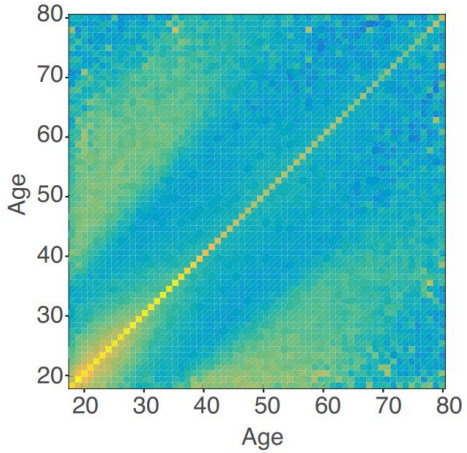
VS.

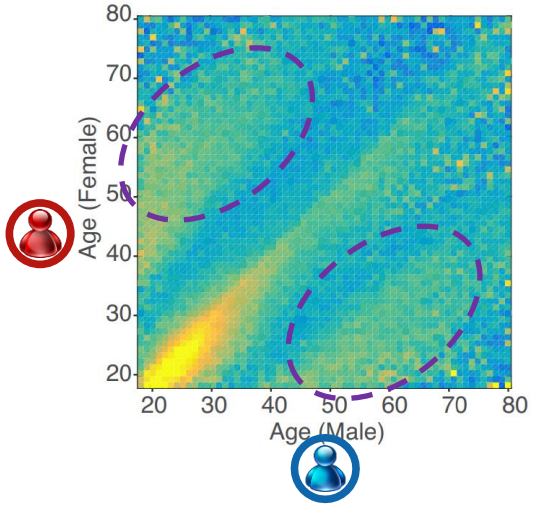
Color:
#calls/per month

- Interactions between young girls and boys are much more frequent than those between two girls or two boys.

Results in the CALL network, and similar observations are also found from SMS.

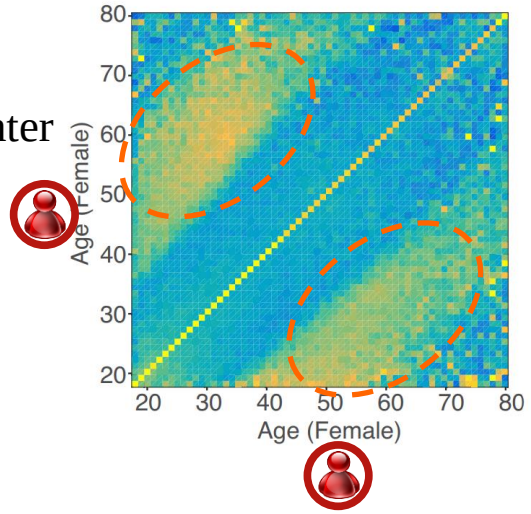
Social Tie Strength

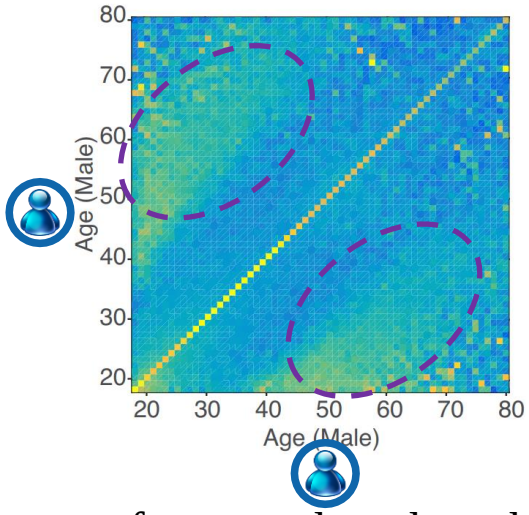




e.g., mom--son
dad--daughter

e.g., mom--daughter





e.g., dad--son

- Cross-generation interactions between two females are more frequent than those between two males or one male and one female.

Results in the CALL network, and similar observations are also found from SMS.

Social Strategies across the Lifespan

Younger → Older

more friends { same-gender, opposite-gender

fewer friends { only same-gender, closed circles

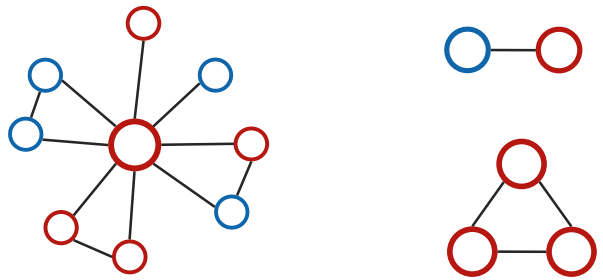
Computational models

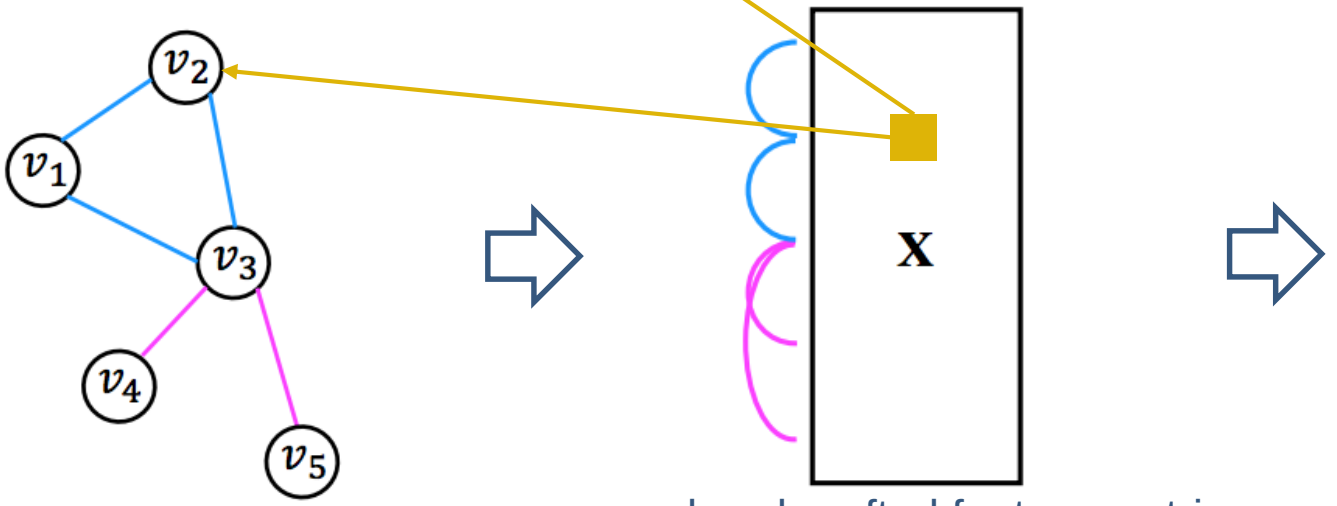
Can we know who we are based on our social networks?

Network Mining and Learning Paradigm

Node Centralities:

- degree
- betweenness
- clustering coefficient
- PageRank
- Eigenvector
- …





hand-crafted feature matrix

feature engineering

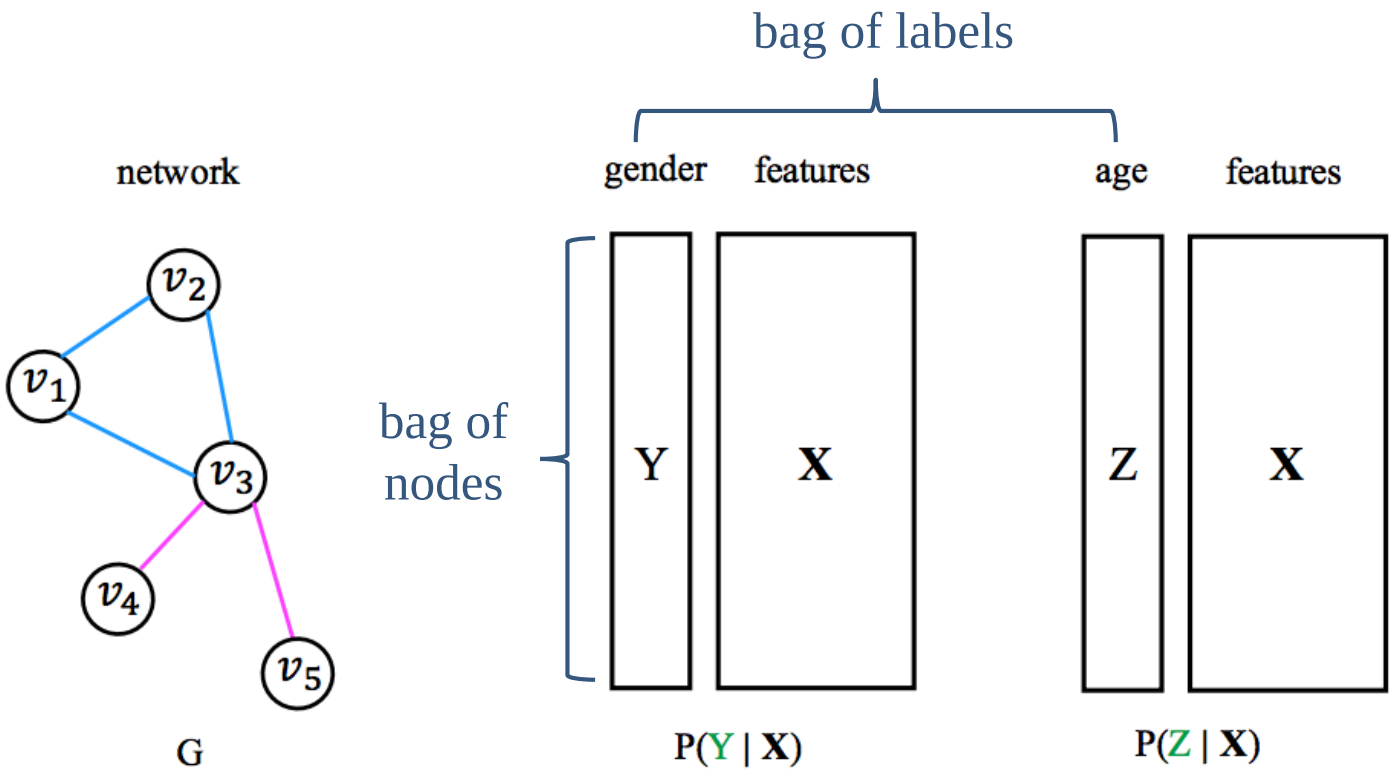
machine learning models

Network Mining Tasks

- **node attribute inference**
- community detection
- similarity search
- link prediction
- social recommendation
- …

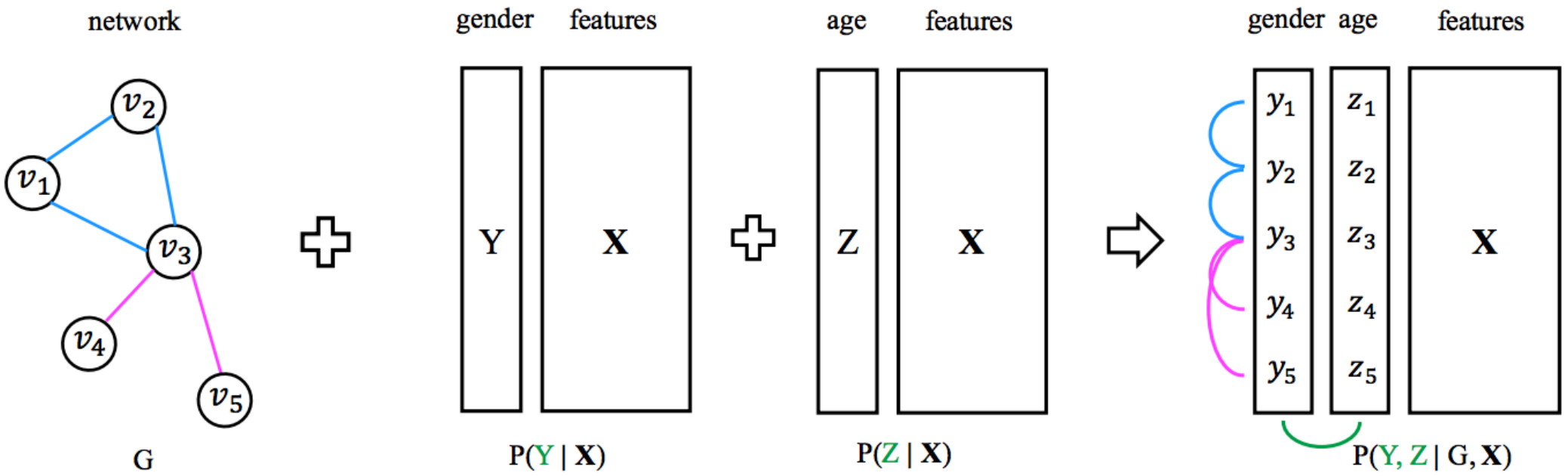
Predicting User Demographic Attributes

- Infer Users' Gender Y and Age Z Separately.
  - Model correlations between gender Y and attributes $\mathbf{X}$;
  - Model correlations between age Z and attributes $\mathbf{X}$;

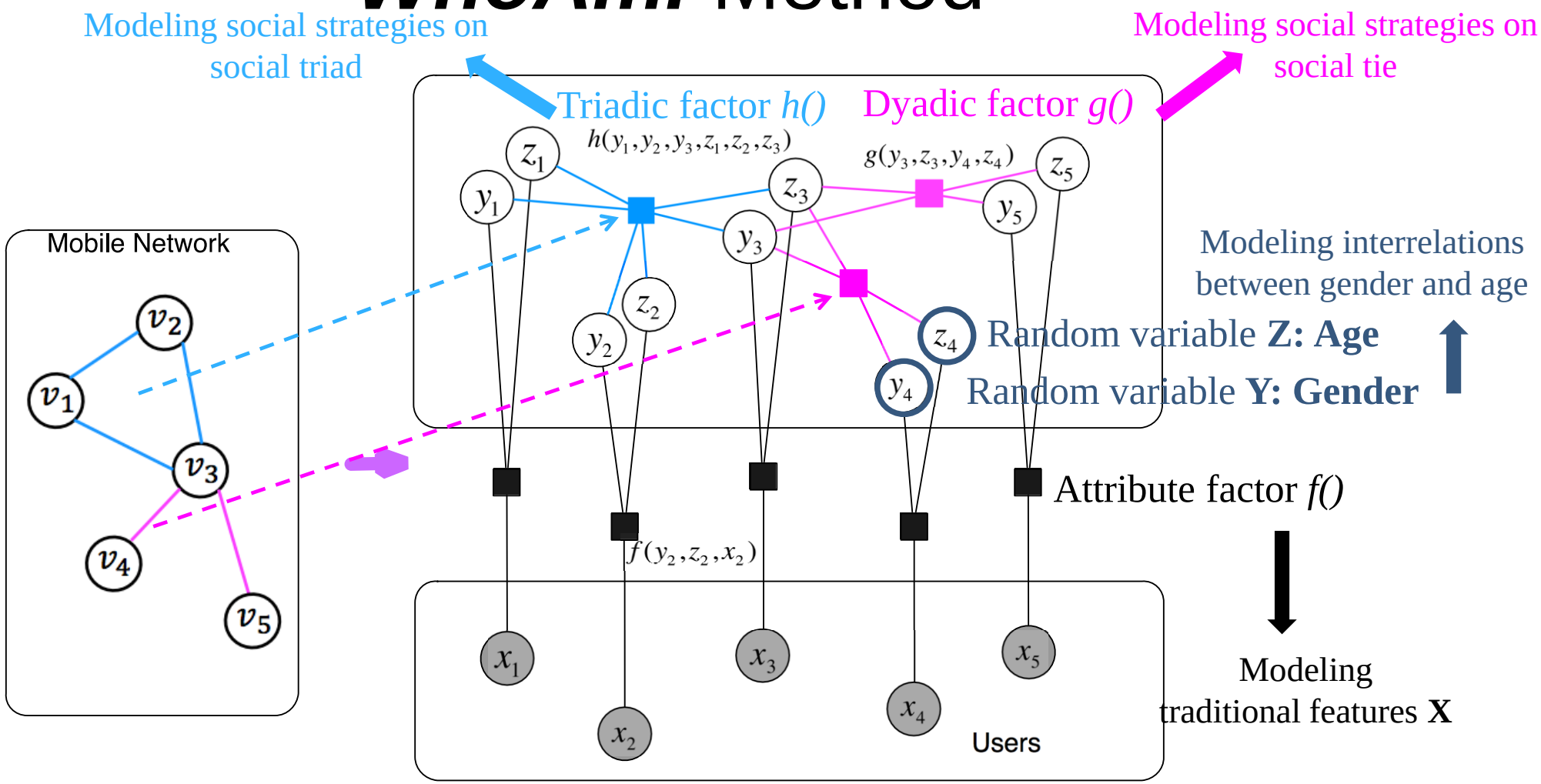


Demographic Prediction

- Infer Users' Gender Y and Age Z Simultaneously.
  - Model correlations between gender Y and attributes $\mathbf{X}$, Network G and Y;
  - Model correlations between age Z and attributes $\mathbf{X}$, Network G and Z;
  - Model interrelations between Y and Z;



WhoAmI Method



Joint Distribution:

$$P(Y, Z|G, \mathbf{X}) = \prod_{v_i \in V} f(y_i, z_i, \mathbf{x}_i) \prod_{e_{ij} \in E} [g(\mathbf{y}_e, \mathbf{z}_e)] \prod_{c_{ijk} \in G} [h(\mathbf{y}_c, \mathbf{z}_c)]$$

Code is available at: http://arnetminer.org/demographic

WhoAmI: Objective Function

Objective function:

$$\mathcal{O}(\alpha,\beta,\gamma)=\sum_{v_i\in V}\alpha_{y_i z_i}\mathbf{x}_i+\sum_{e_{ij}\in E}\sum_{p=1}^{6}\beta_p g'_p(\cdot)$$
$$+\sum_{c_{ijk}\in G}\sum_{q=1}^{20}\gamma_q h'_q(\cdot)-\log W$$

Model learning: gradient descent

$$\frac{\partial\mathcal{O}(\theta)}{\partial\alpha}=\boldsymbol{E}[\sum_{v_i\in V}\mathbf{x}_i]-\boldsymbol{E}_{P_\alpha(Y,Z|X)}[\sum_{v_i\in V}\mathbf{x}_i]$$
$$\frac{\partial\mathcal{O}(\theta)}{\partial\beta}=\boldsymbol{E}[\sum_{e_{ij}\in E}g'(\cdot)]-\boldsymbol{E}_{P_\beta(Y,Z|\mathbf{x},G)}[\sum_{e_{ij}\in E}g'(\cdot)]$$
$$\frac{\partial\mathcal{O}(\theta)}{\partial\gamma}=\boldsymbol{E}[\sum_{c_{ijk}\in G}h'(\cdot)]-\boldsymbol{E}_{P_\gamma(Y,Z|\mathbf{x},G)}[\sum_{c_{ijk}\in G}h'(\cdot)]$$

Circles?
Loopy Belief Propagation

K. P. Murphy, Y. Weiss, M. I. Jordan. Loopy Belief Propagation for Approximate Inference: Am Empirical Study. In UAI'99

Code is available at: http://arnetminer.org/demographic

Experiments: Feature Definition

- Given one node *v* and its ego network:
  - Individual feature:
    - Individual attribute: degree, neighbor connectivity, clustering coefficient, embeddedness and weighted degree.
  - Friend feature:
    - Friend attribute: # of connections to female/male, young/young-adult/middle-age/senior friends (from labeled friends).
    - Dyadic factor: both labeled and unlabeled friends for social tie structures in v's ego network.
  - Circle feature:
    - Circle attribute: # of demographic triads, i.e., v-FF, v-FM, v-MM; v-AA, v-AB, v-AC, v-AD, v-BB, v-BC, v-BD, v-CC, v-CD, v-DD. (A/B/C/C denote the young/young-adult/middle-age/senior)
    - Triadic factor: both labeled and unlabeled friends for social triad structures in v's ego network.
- LCR/SVM/NB/RF/Bag/RBF:
  - Individual/Friend/Circle Attributes
- FGM/DFG
  - Individual/Friend/Circle Attributes
  - Structure feature: Dyadic factors
  - Structure feature: Triadic factors

WhoAmI: Experiments

Network	Method	Gender			Age		
		wPrecision	wRecall/Accu	wF1-Measure	wPrecision	wRecall/Accu	wF1-Measure
CALL	LRC						
	SVM						
	NB						
	RF						
	Bag						
	RBF						
	FGM						
	DFG						
SMS	LRC						
	SVM						
	NB						
	RF						
	Bag						
	RBF						
	FGM						
	DFG						

♣ Data: mobile phone users

- >1.09 million users in CALL
- >304 thousand users in SMS
- 50% as training data
- 50% as test data

♣ Evaluation Metrics:

- Weighted Precision
- Weighted Recall
- Weighted F1 Measure
- Accuracy

♣ Baselines:

- LRC: Logistic Regression
- SVM: Support Vector Machine
- NB: Naïve Bayes
- RF: Random Forest
- BAG: Bagged Decision Tree
- RBF: Gaussian Radial Basis NN
- FGM: Factor Graph Model
- ***DFG (WhoAmI)***

Demographic Predictability

Network	Method	Gender			Age		
		wPrecision	wRecall/Accu	wF1-Measure	wPrecision	wRecall/Accu	wF1-Measure
CALL	LRC						
	SVM						
	NB						
	RF						
	Bag						
	RBF						
	FGM						
	DFG						
SMS	LRC						
	SVM						
	NB						
	RF						
	Bag						
	RBF						
	FGM						
	DFG						

♣ Predictability of User Demographic Profiles

- The proposed *WhoAmI* (DFG) outperforms baselines by up to 10% in terms of F1-Measure.

- We can infer 80% of users' gender from the CALL network
- We can infer 73% of users' age from the SMS network

- The phone call behavior reveals more user gender than text messaging
- The text messaging behavior reveals more user age than phone call

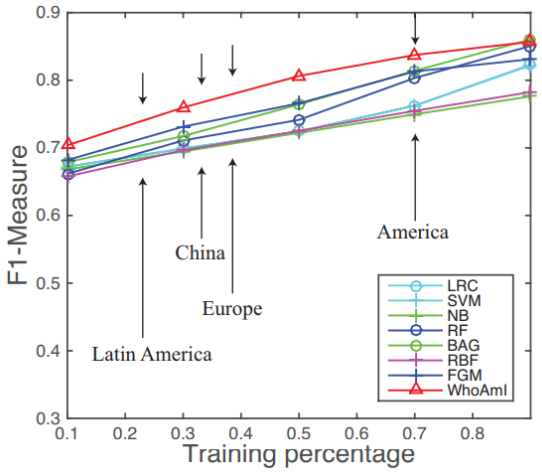
Application 1: Postpaid → Prepaid

- ***Postpaid*** mobile users are required to create an account by providing detailed demographic information (e.g., name, age, gender, etc.).

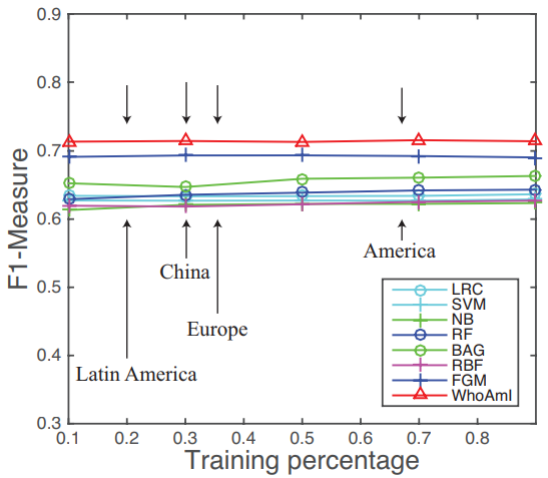
- ***Prepaid*** services (pay-as-you-go) allow users to be anonymous --- no need to provide any user-specific information.
  - 95% of mobile users in India
  - 80% of mobile users in Latin America
  - 70% of mobile users in China
  - 65% of mobile users in Europe
  - 33% of mobile users in the United States

- Train the model on postpaid users and infer prepaid users' demographics

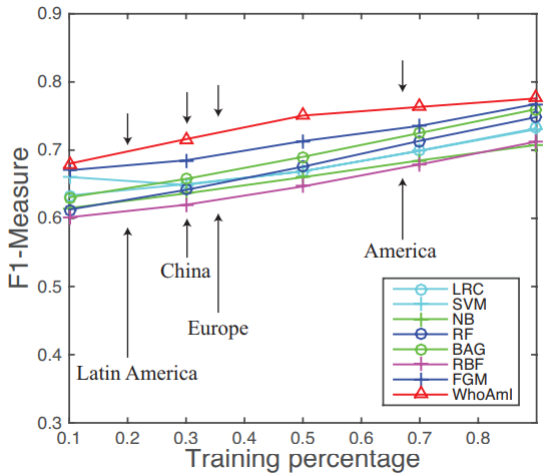
Application 1: Postpaid → Prepaid



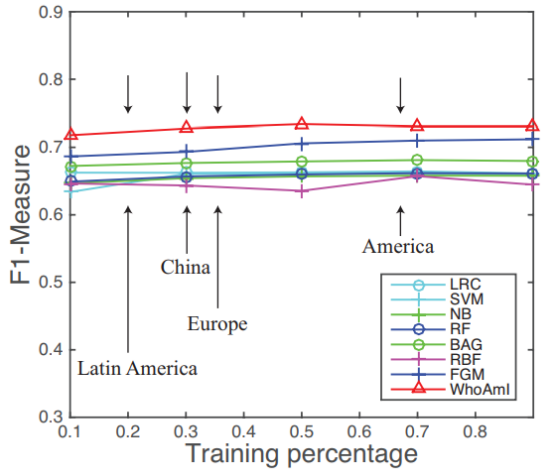
CALL Gender



CALL Age



SMS Gender



SMS Age

- Slide the training ratio to match proportion of postpaid users per country
- Train the model on postpaid users and infer prepaid users' demographics

Application 2: Coupled Networks

2015.08.08 10:30

2015.08.08 10:48

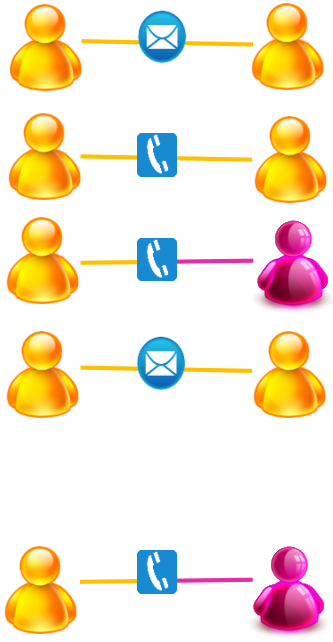
2015.08.08 11:01

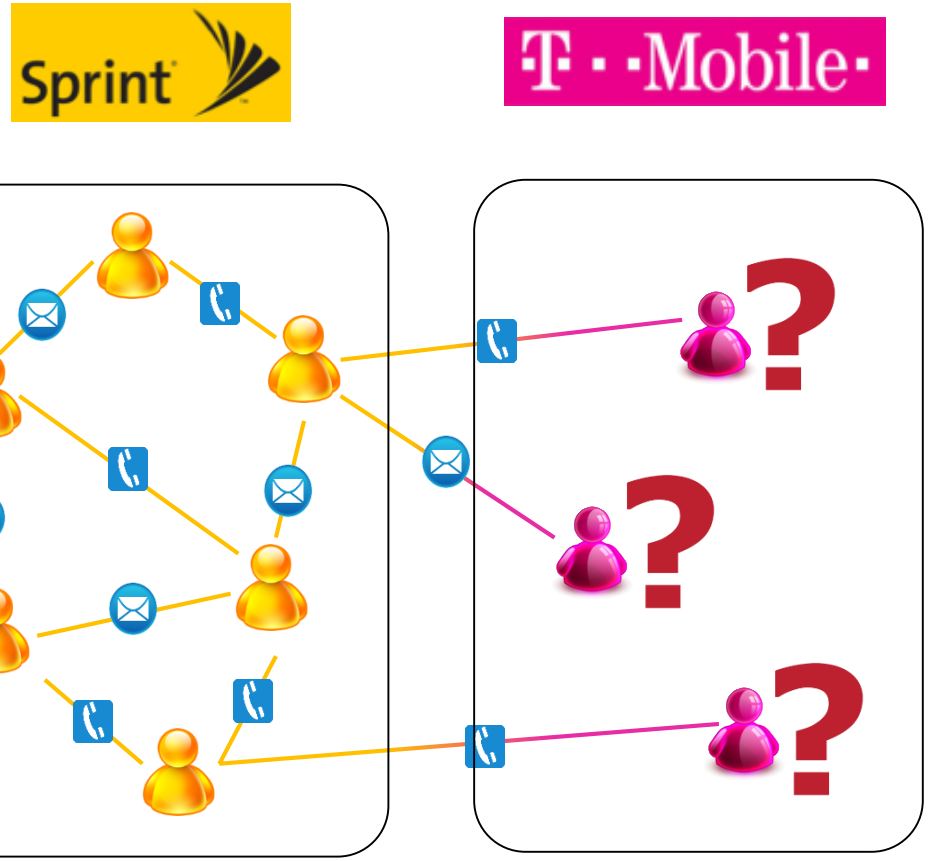
2015.08.08 11:29

……

2016.01.01 00:00

……





Coupled Demographic Prediction

Coupled Network Data

- Real-world large mobile communication data
  - Over 1 billion call & message records between Aug. to Sep. 2008
  - Undirected and weighted networks
  - Three major mobile operators E_a, E_b, E_c

	E_a	E_b	E_c	$E_a \leftrightarrow E_b$	$E_a \leftrightarrow E_c$	$E_b \leftrightarrow E_c$
#Nodes	2,531,187	655,755	354,166	1,912,933	1,255,046	625,379
#Links	3,355,197	649,322	311,432	1,844,342	1,131,593	507,894
k	2.65	1.98	1.75	1.92	1.80	1.62
cc	0.0457	0.0366	0.0317	0	0	0
ac	0.2848	0.2693	0.2806	0.0231	-0.0305	0.1113

k: average degree
cc: clustering coefficient
ac: associative coefficient

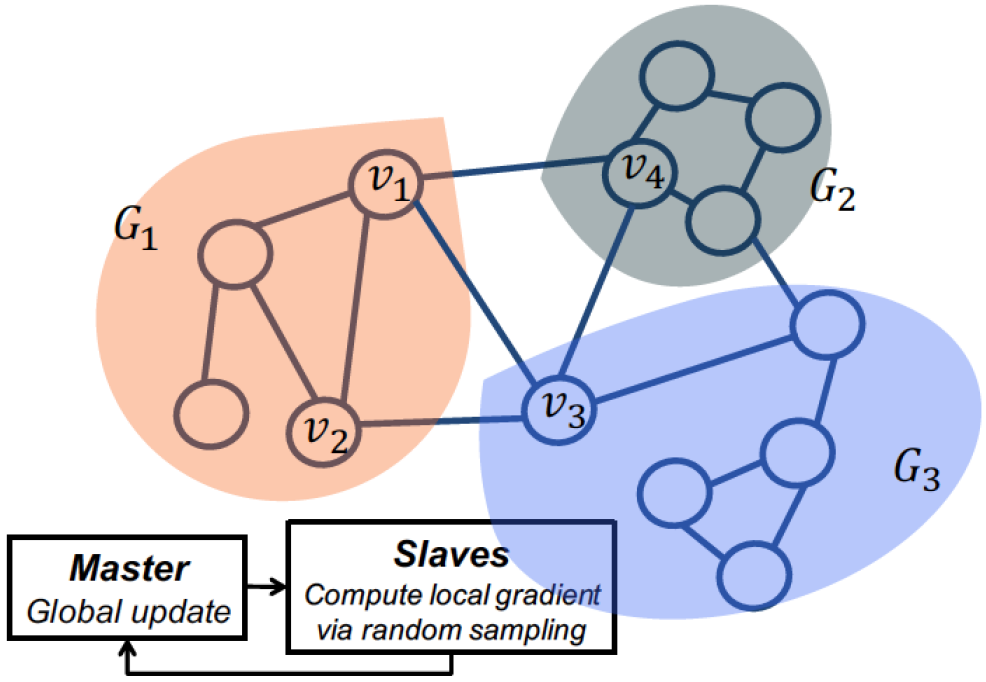
WhoAmI: Distributed Coupled Learning

ALGORITHM 1: Distributed CoupledMFG Learning Algorithm.

Input: The source network G^S, the cross network G^C, the node set V^T of the target network G^T, and the learning rate η

Output: Parameters $\theta = (\alpha^S, \alpha^T, \beta, \gamma)$

Master initializes $\theta \leftarrow 0$;
Master constructs the coupled factor graph according to Eq. 4.12 with G^S, G^C, V^T;
Master partitions the input mobile network into K subgraphs of relatively equal size;
Master completes the broken structural factors with virtual nodes;
Master forwards all subgraphs to slaves [Communication];
repeat
  Master broadcasts θ to Slaves [Communication];
  for $k = 1 \rightarrow K$ **do**
    Slave k computes local belief according to Eqs. 4.9 and 4.10;
    Slave k sends the local belief to Master [Communication];
  end
  Master calculates the marginal distribution for each variable according to Eq. 4.11;
  Master calculates the gradient for each parameter according to Eq. 4.7;
  Master updates the parameters according to Eq. 4.8;
until *Convergence*;



MPI based

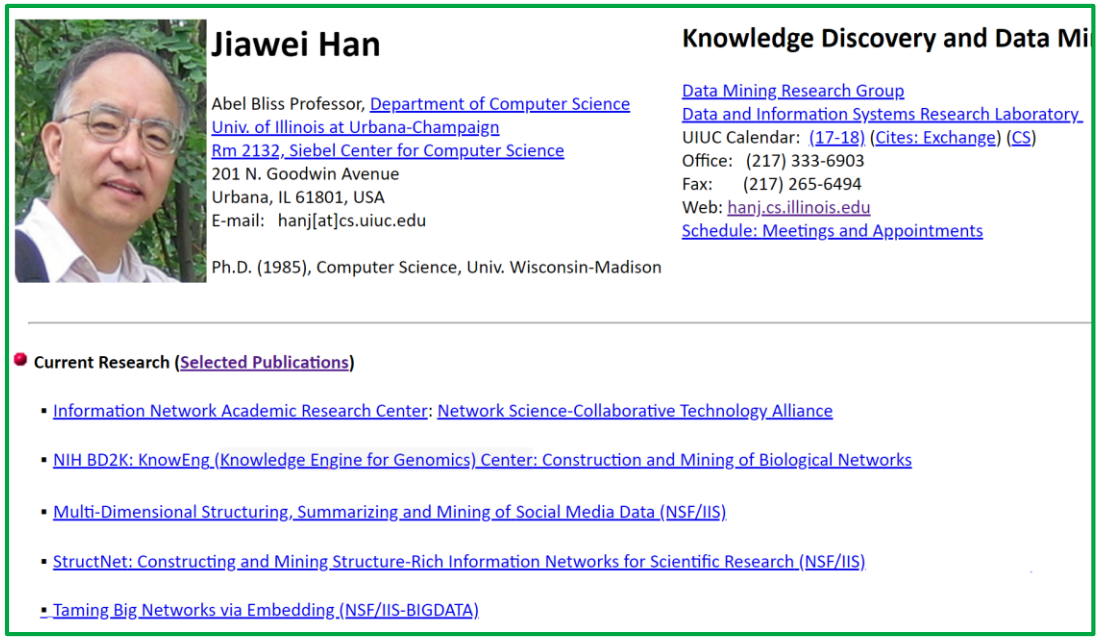
Coupled Demographic Prediction

Network	Method	Gender			Age		
		wPrecision	wRecall	wF1-Measure	wPrecision	wRecall	wF1-Measure
CALL	E_a to E_b	0.7870	0.7800	0.7807	0.7075	0.7087	0.7039
	E_a to E_c	0.7936	0.7939	0.7818	0.7100	0.7140	0.7085
	E_b to E_a	0.7404	0.7403	0.7396	0.6986	0.6801	0.6696
	E_b to E_c	0.7986	0.7979	0.7982	0.7160	0.7167	0.7094
	E_c to E_a	0.7325	0.7282	0.7251	0.6900	0.6758	0.6622
	E_c to E_b	0.7810	0.7794	0.7768	0.7147	0.7090	0.6981
SMS	E_a to E_b	0.7217	0.7222	0.7219	0.7172	0.7168	0.7049
	E_a to E_c	0.7329	0.7326	0.7327	0.7240	0.7259	0.7143
	E_b to E_a	0.6737	0.6713	0.6721	0.6897	0.6734	0.6540
	E_b to E_c	0.7347	0.7288	0.7285	0.7272	0.7245	0.7095
	E_c to E_a	0.6831	0.6846	0.6798	0.6885	0.6729	0.6497
	E_c to E_b	0.7232	0.7201	0.7143	0.7191	0.7152	0.6964

- Train the model on my own users and infer the demographics of my competitor' users.
- Infer 73~79% of gender information and 66~70% of age of a competitor's users.

Generation to Other Networks?

Profiling Web Users: Gender



Jiawei Han

Abel Bliss Professor, Department of Computer Science
Univ. of Illinois at Urbana-Champaign
Rm 2132, Siebel Center for Computer Science
201 N. Goodwin Avenue
Urbana, IL 61801, USA
E-mail: hanj[at]cs.uiuc.edu

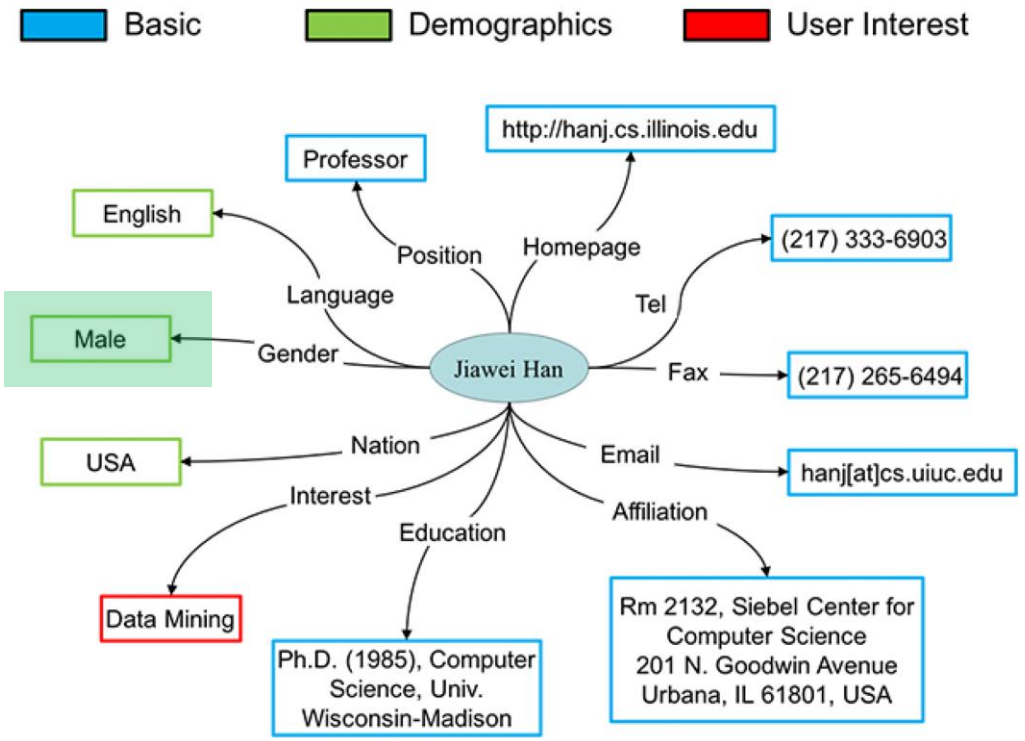
Ph.D. (1985), Computer Science, Univ. Wisconsin-Madison

Knowledge Discovery and Data Mi

Data Mining Research Group
Data and Information Systems Research Laboratory
UIUC Calendar: (17-18) (Cites: Exchange) (CS)
Office: (217) 333-6903
Fax: (217) 265-6494
Web: hanj.cs.illinois.edu
Schedule: Meetings and Appointments

Current Research (Selected Publications)

- Information Network Academic Research Center: Network Science-Collaborative Technology Alliance
- NIH BD2K: KnowEng (Knowledge Engine for Genomics) Center: Construction and Mining of Biological Networks
- Multi-Dimensional Structuring, Summarizing and Mining of Social Media Data (NSF/IIS)
- StructNet: Constructing and Mining Structure-Rich Information Networks for Scientific Research (NSF/IIS)
- Taming Big Networks via Embedding (NSF/IIS-BIGDATA)



Welcome Mr./Mrs.?

Scholar Gender Prediction

Name jie tang

Affiliation tsinghua university

Reset Submit

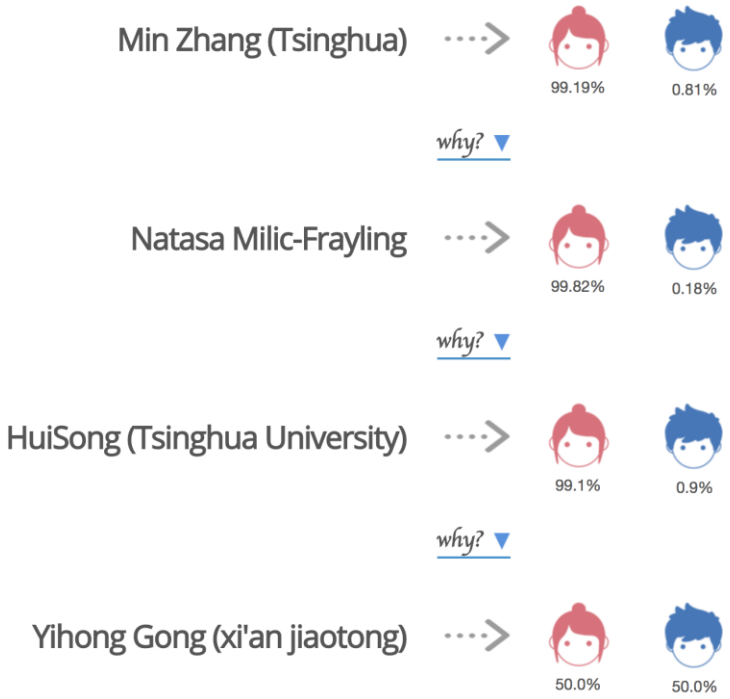
P.S. English supported only

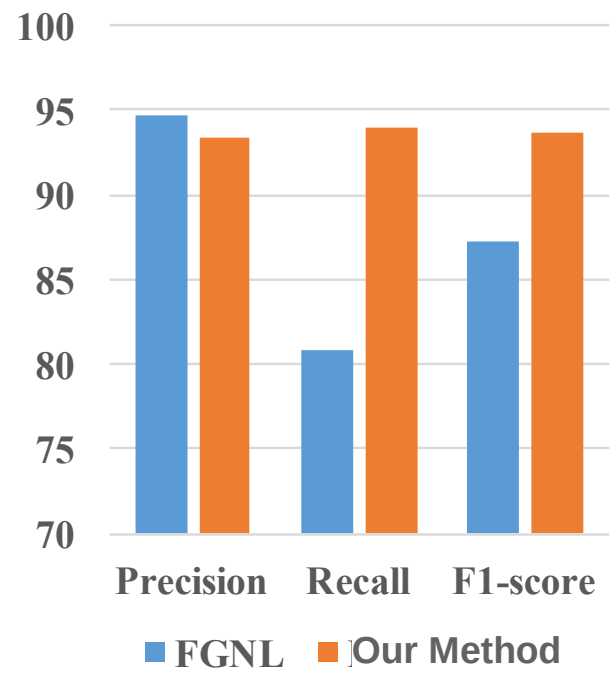
99.17%

jie tang (tsinghua university)

why?

Inferring Gender





FGNL is a baseline method

https://aminer.org/gender

Gu et al. Web User Profiling using Data Redundancy. **ASONAM'16**. (**Best Student Paper Runner-up**)

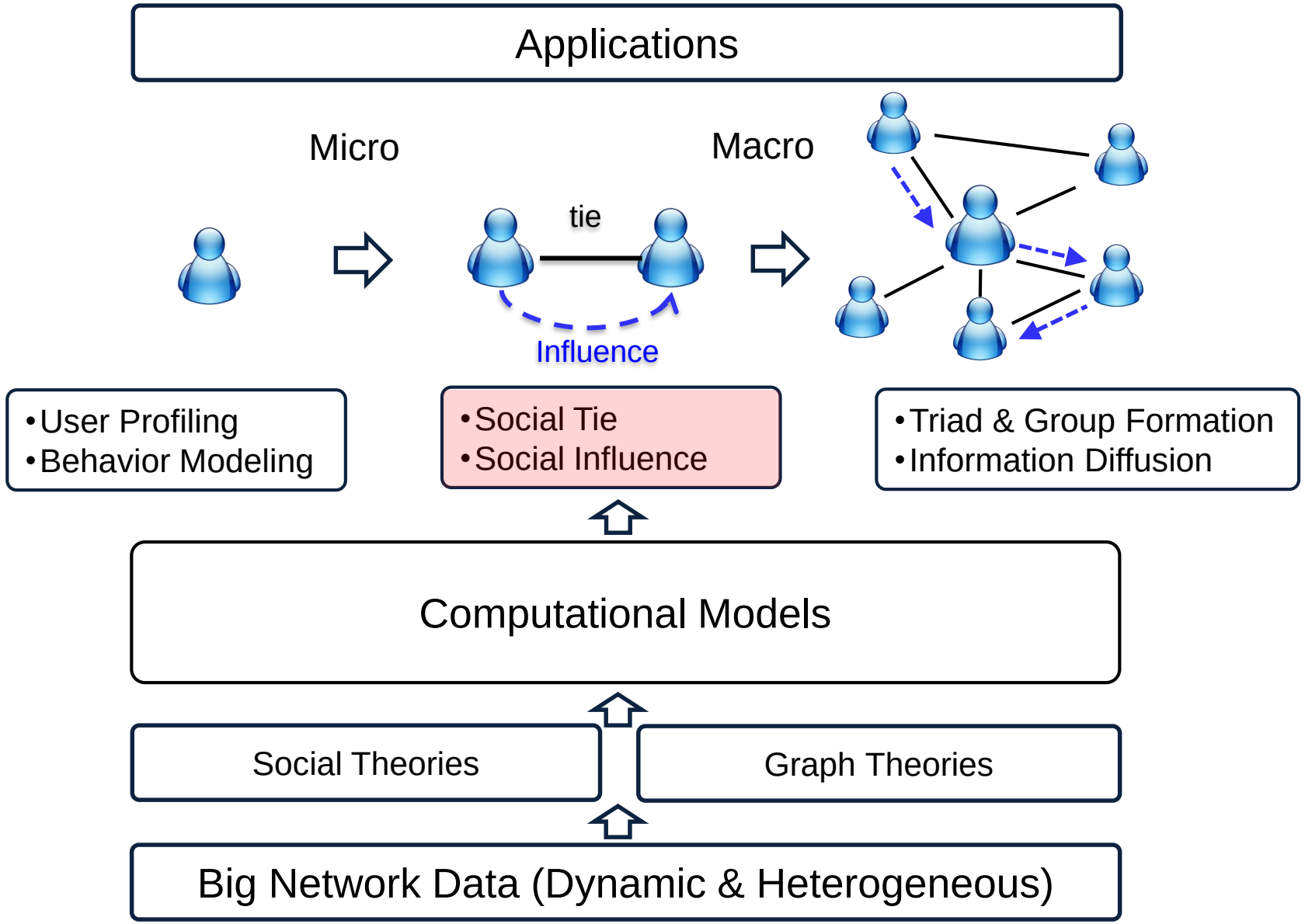
Structural features + graphical models

- The discovery of the evolution of social strategies across lifespan
- A Probabilistic Graphical Model---Multi-Label Factor Graph (***WhoAmI***)---for node attribute prediction in networks
- The predictability of users' gender and age from mobile communication networks & two applications in telecommunications.

Feature Definition & Construction

- Given one node *v* and its ego network:
  - Individual feature:
    - Individual attribute: degree, neighbor connectivity, clustering coefficient, embeddedness and weighted degree.
  - Friend feature:
    - Friend attribute: # of connections to female/male, young/young-adult/middle-age/senior friends (from labeled friends).
    - Dyadic factor: both labeled and unlabeled friends for social tie structures in v's ego network.
  - Circle feature:
    - Circle attribute: # of demographic triads, i.e., v-FF, v-FM, v-MM; v-AA, v-AB, v-AC, v-AD, v-BB, v-BC, v-BD, v-CC, v-CD, v-DD. (A/B/C/C denote the young/young-adult/middle-age/senior)
    - Triadic factor: both labeled and unlabeled friends for social triad structures in v's ego network.
- LCR/SVM/NB/RF/Bag/RBF:
  - Individual/Friend/Circle Attributes
- FGM/DFG
  - Individual/Friend/Circle Attributes
  - Structure feature: Dyadic factors
  - Structure feature: Triadic factors

This Tutorial:



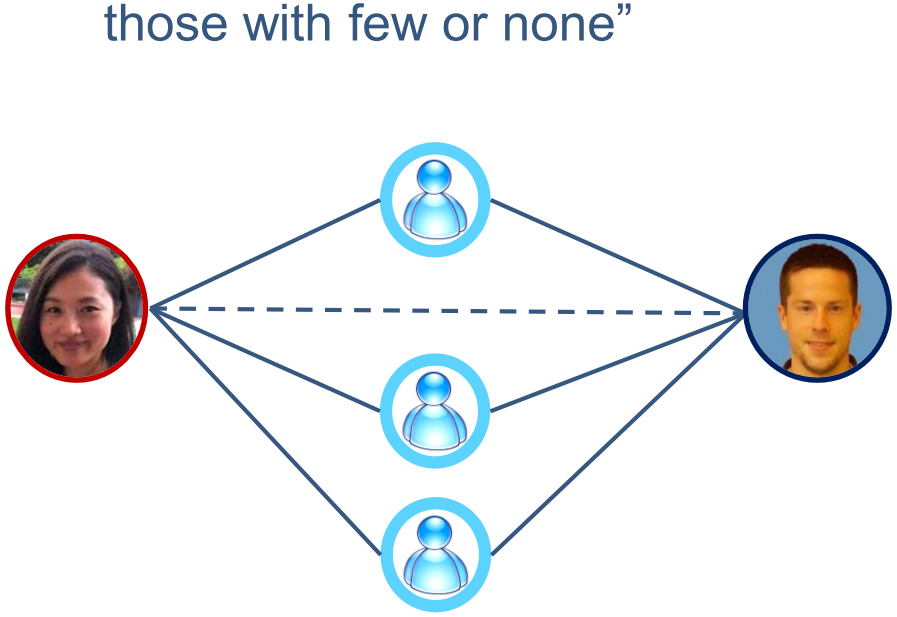
Link Prediction and Social Tie

- Dong et al., Structural Diversity and Homophily: A Study Across More Than One Hundred Big Networks. In *KDD 2017*.
- Tang et al., Transfer learning to infer social ties across heterogeneous networks. In TOIS 2016.

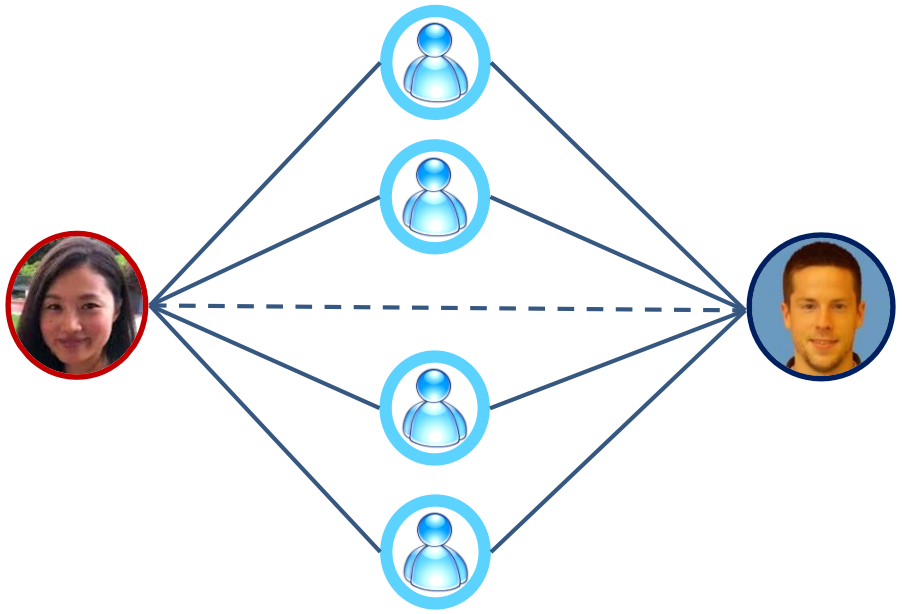
Link Prediction

"Love those who are like themselves" ---*Aristotle*

"People with many common friends are more likely to become acquainted than those with few or none"



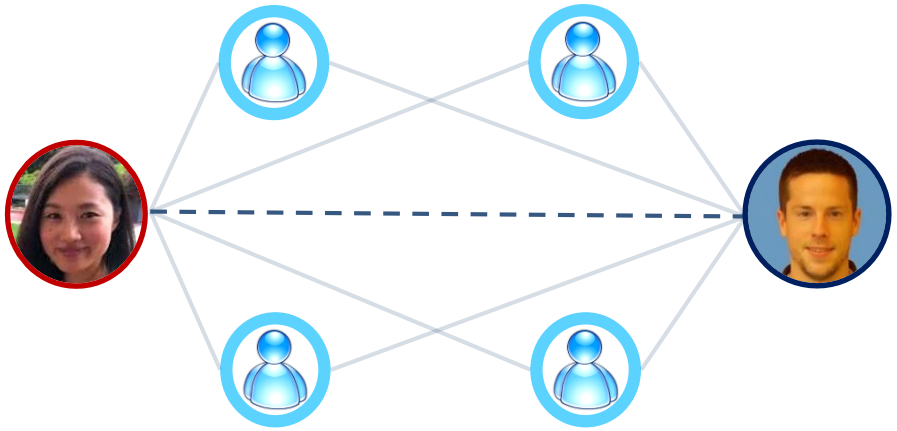
P_1 < P_2

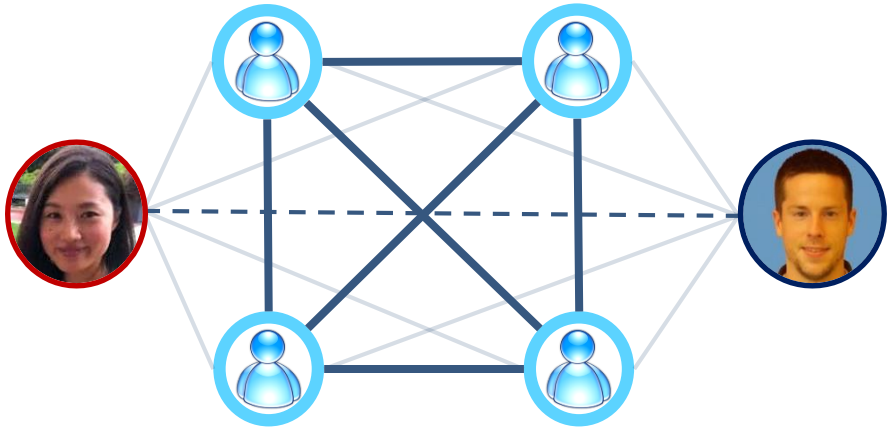


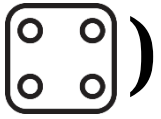
- M. E. J. Newman. Clustering and preferential attachment in growing networks. **Phys. Rev. E**. 2001.
- M. McPherson, L. Smith-Lovin, J. M. Cook. Birds of a feature: homophily in social networks. **Annual Review of Sociology**. 2001.

Common Neighbor (CN) Subgraph

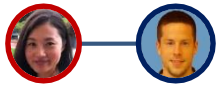
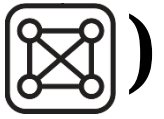
P(connect | common-neighbor-subgraph)





P_1( | )

?

P_2( | )

more diverse

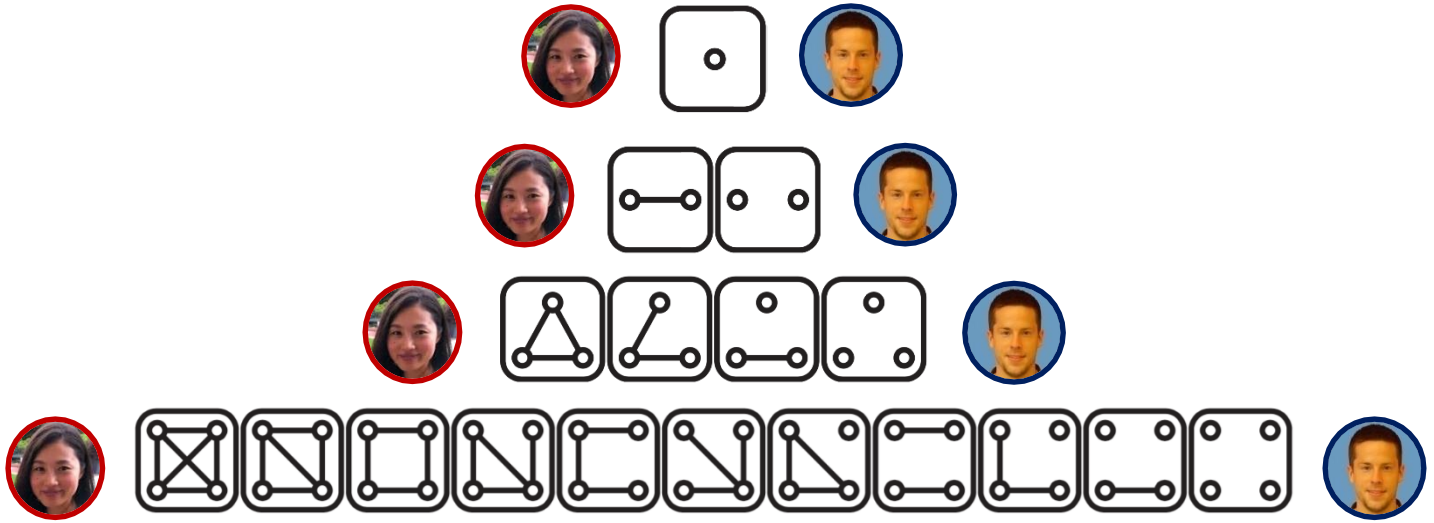
less diverse

Structural Diversity: #components of a common neighbor subgraph

- M. Granovetter. Problems of explanation in economic sociology. Networks and organizations: Structure, form, and action, 25:56, 1992.
- B. Uzzi. Social structure and competition in interfirm networks: the paradox of embeddedness. **Administrative science quarterly**. 1997.
- J. Ugander, L. Backstrom, C. Marlow, and J. Kleinberg. Structural diversity in social contagion. PNAS, 109(16):5962–5966, 2012.

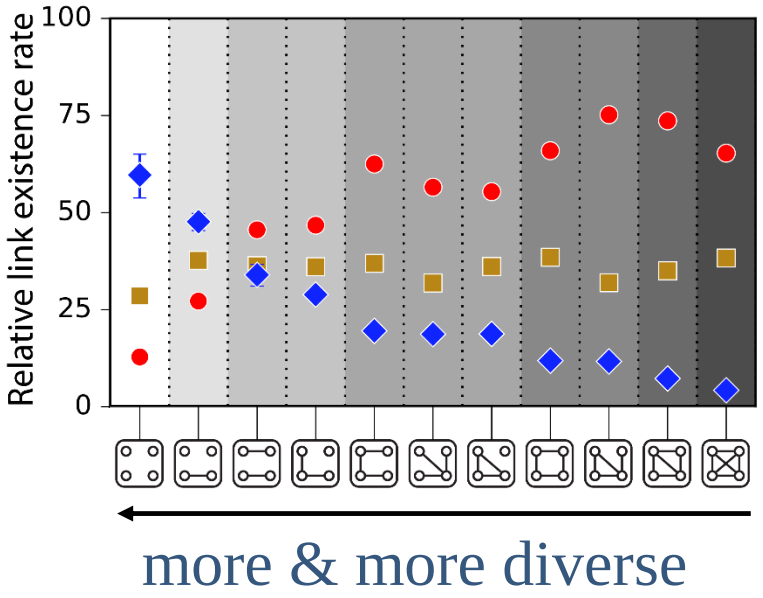
Common Neighbor (CN) Subgraph

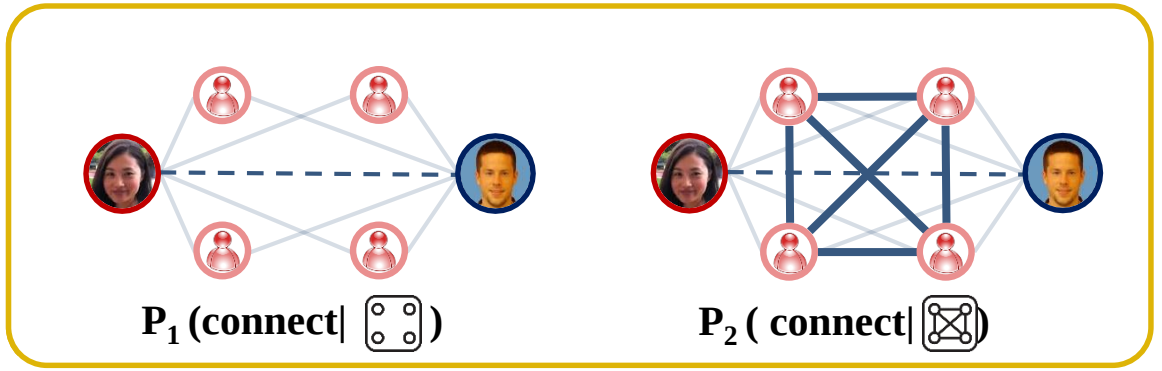
P(connect | common-neighbor-subgraph)



Network	# nodes	# edges	# pairs with ≥1 CN	Data source
Friendster	65,608,366	1,806,067,135	546 billion	SNAP
BlogCatalog	88,784	2,093,195	612 million	ASU
YouTube	1,134,890	2,987,624	1 billion	MPI-SWS

Structural Diversity of CN Subgraph Affects Link Existence



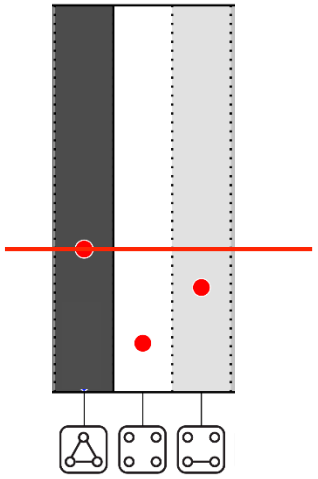


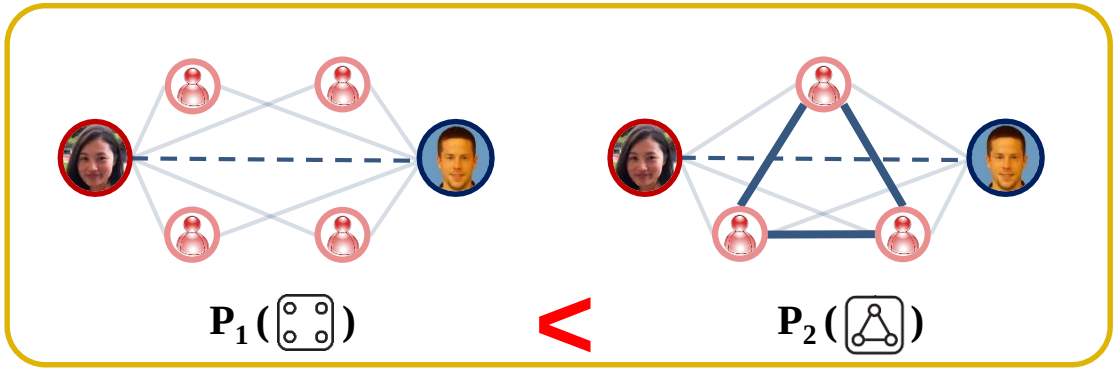
$$\frac{P_1(\text{connect} \mid \square)}{P_2(\text{connect} \mid \boxtimes)} \approx \frac{1}{5}$$

- YouTube
- Friendster
- BlogCatalog

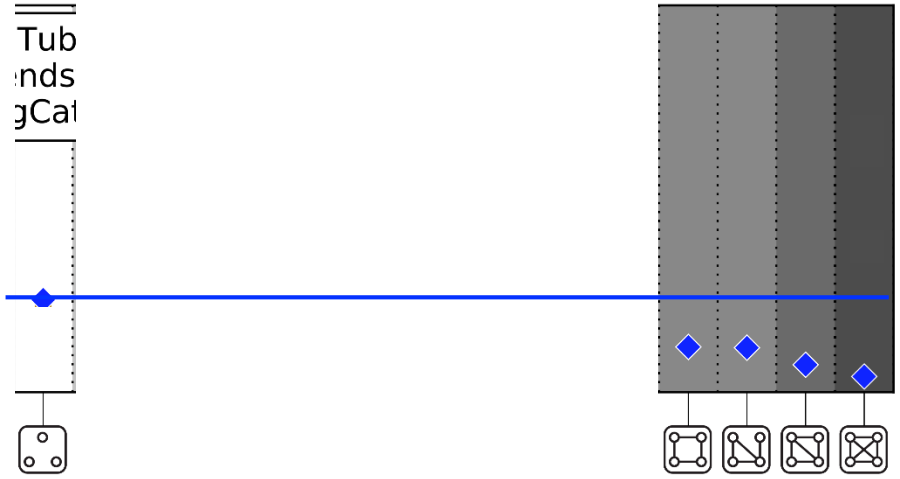
friendster

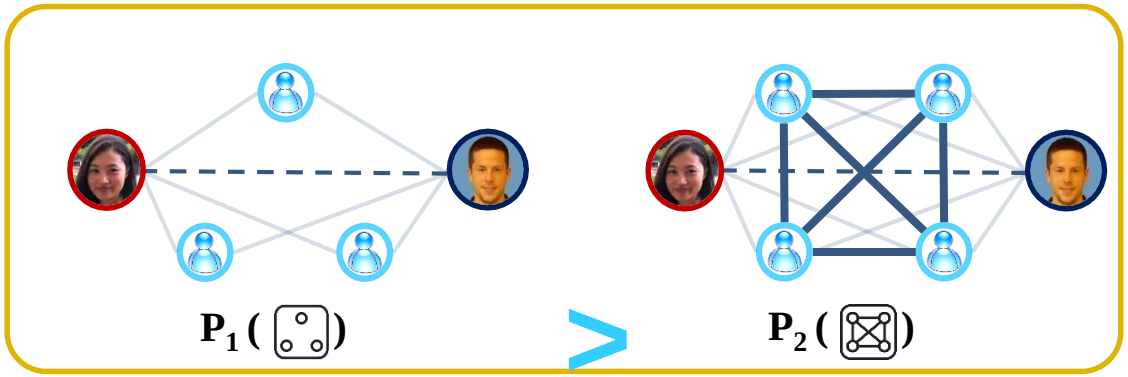
The Violation of Structural Homophily





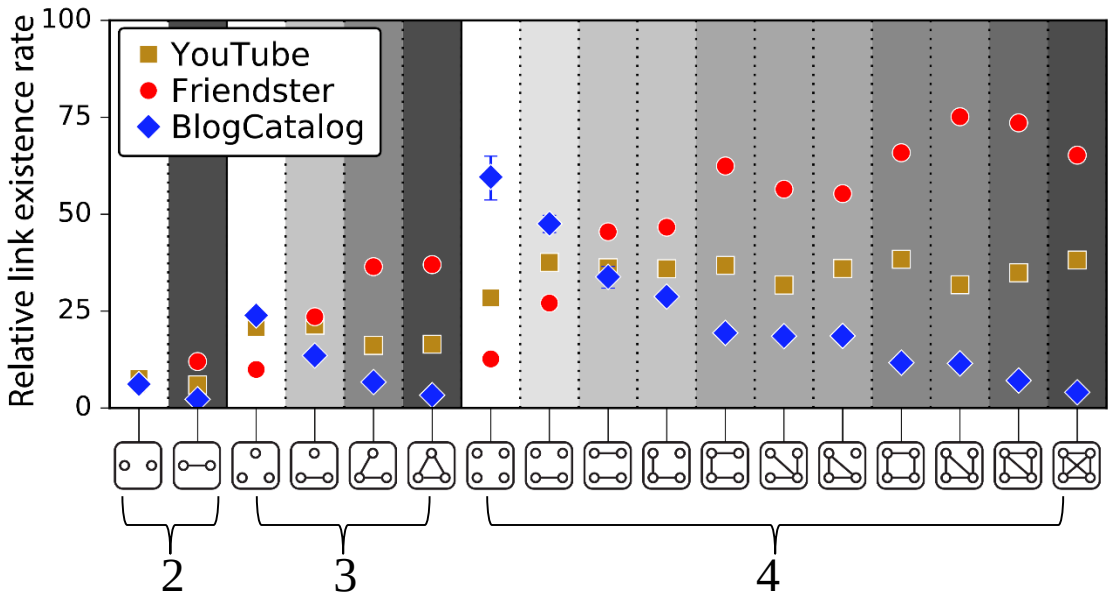
friendster

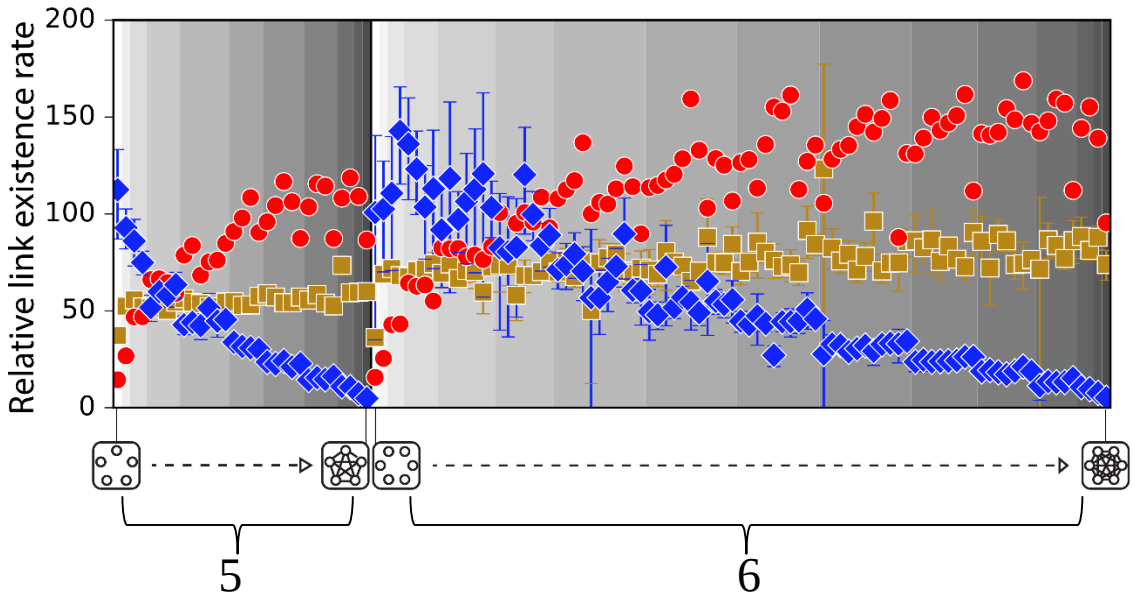




blogcatalog

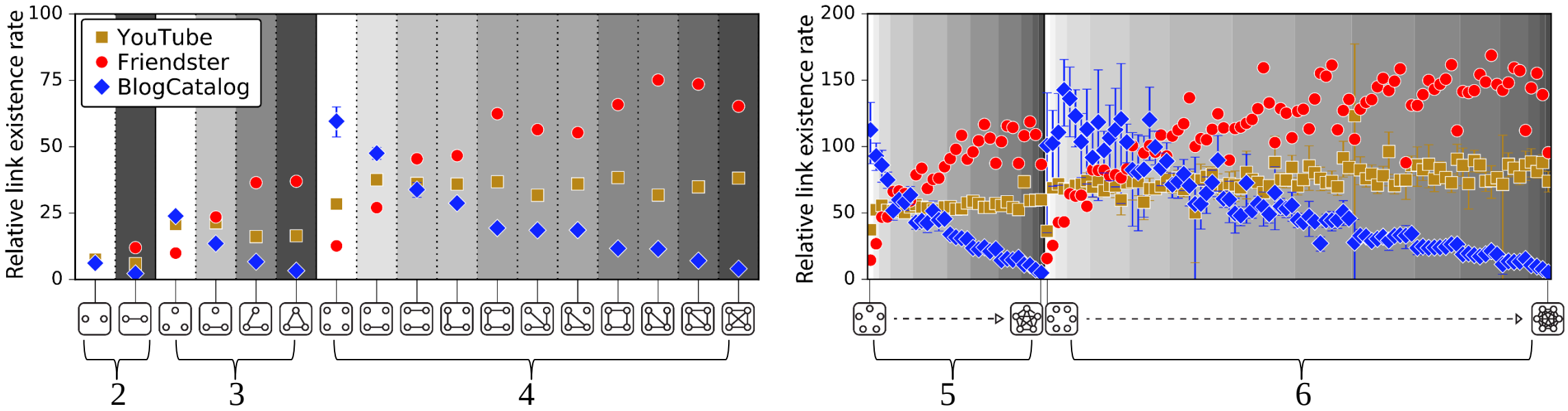
Structural Diversity of Common Neighborhood





- The diversity of common neighborhood affects link formation and also violates the principle of homophily.

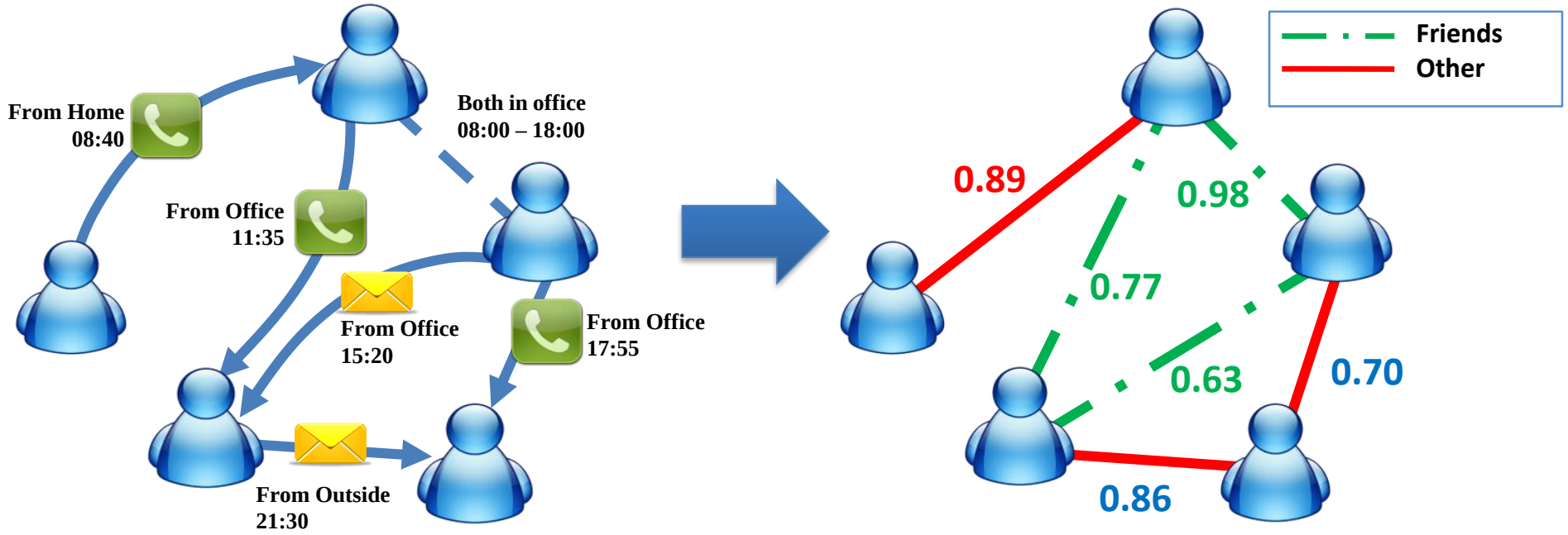
Common Neighborhood Signature



$$\left[\, y_2^1, y_2^2,\ y_3^1, y_3^2, y_3^3, y_3^4, y_4^1, y_4^2 \ \ldots\ y_4^{11}, y_5^1 \ \ldots\ \ldots \,\right]$$

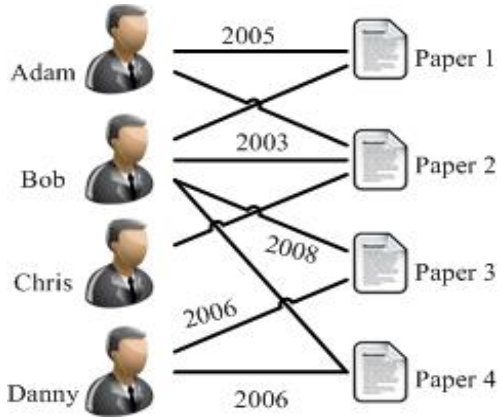
Metric	Method	Friendster	BlogCatalog	YouTube
Data	#Pairs	67,033,108,105	224,786,028	118,635,122
	%Positive	0.91830%	0.09430%	0.50820%
AUPR	Homophily	0.02230	0.00178	0.01524
	Diversity	0.03499	0.00279	0.01532
AUROC	Homophily	0.68539	0.66259	0.69371
	Diversity	0.71722	0.70239	0.68401

Social Tie

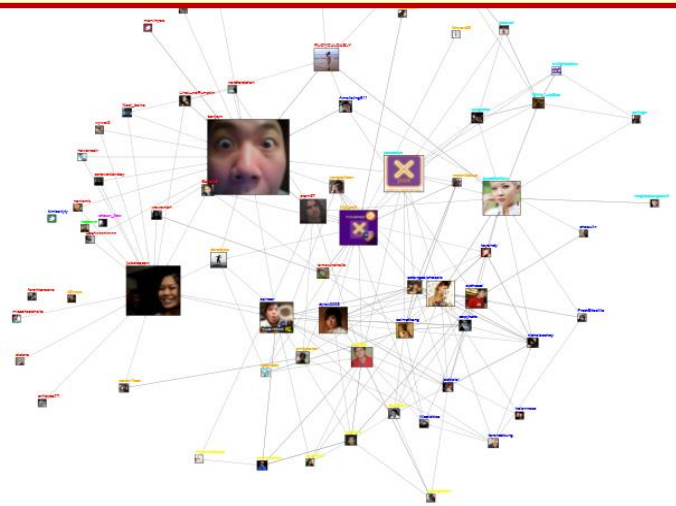

From Home
08:40
Both in office
08:00 – 18:00
From Office
11:35
From Office
15:20
From Office
17:55
From Outside
21:30
Friends
Other
0.89
0.98
0.77
0.63
0.70
0.86

Challenges

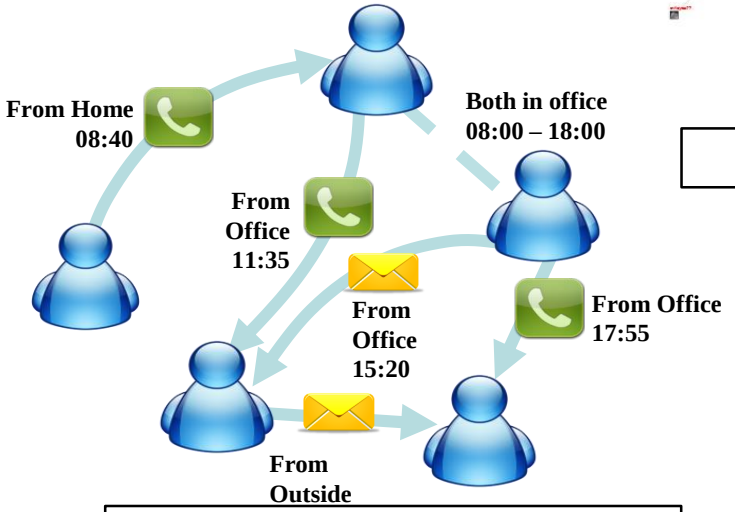
- What are the fundamental forces behind?
- Can we automatically infer the type of social ties?



Publication network



Twitter's following network



Mobile communication network

Networks

- **Epinions** a network of product reviewers: 131,828 nodes (users) and 841,372 edges
  - trust relationships between users
- **Slashdot**: 82,144 users and 59,202 edges
  - “friend” relationships between users
- **Mobile**: 107 mobile users and 5,436 edges
  - to infer friendships between users

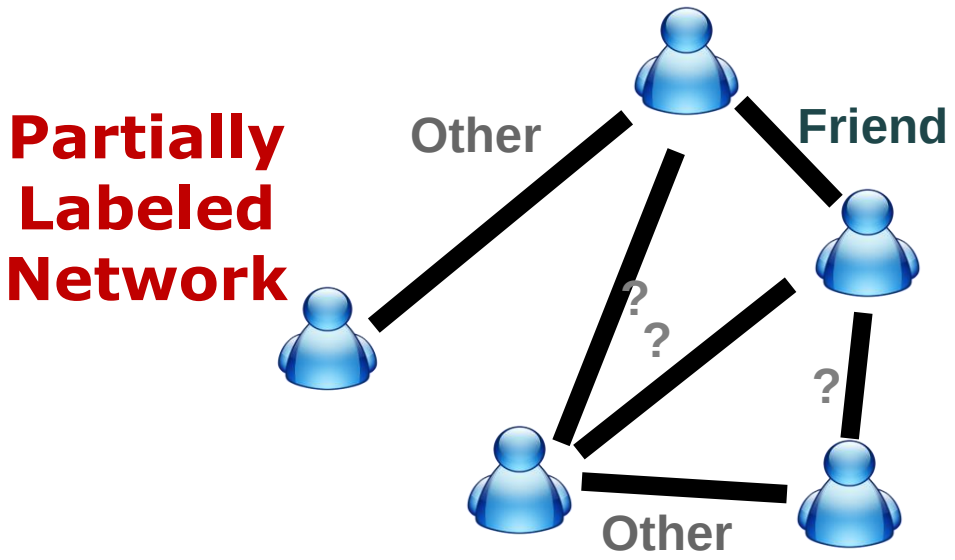
Undirected network

- **Coauthor**: 815,946 authors and 2,792,833 coauthor relationships
  - to infer advisor-advisee relationships between coauthors
- **Enron**: 151 Enron employees and 3572 edges
  - to infer manager-subordinate relationships between users.

Directed network

Problem Formulation

Input: $G=(V,E^L,E^U,R^L,W)$



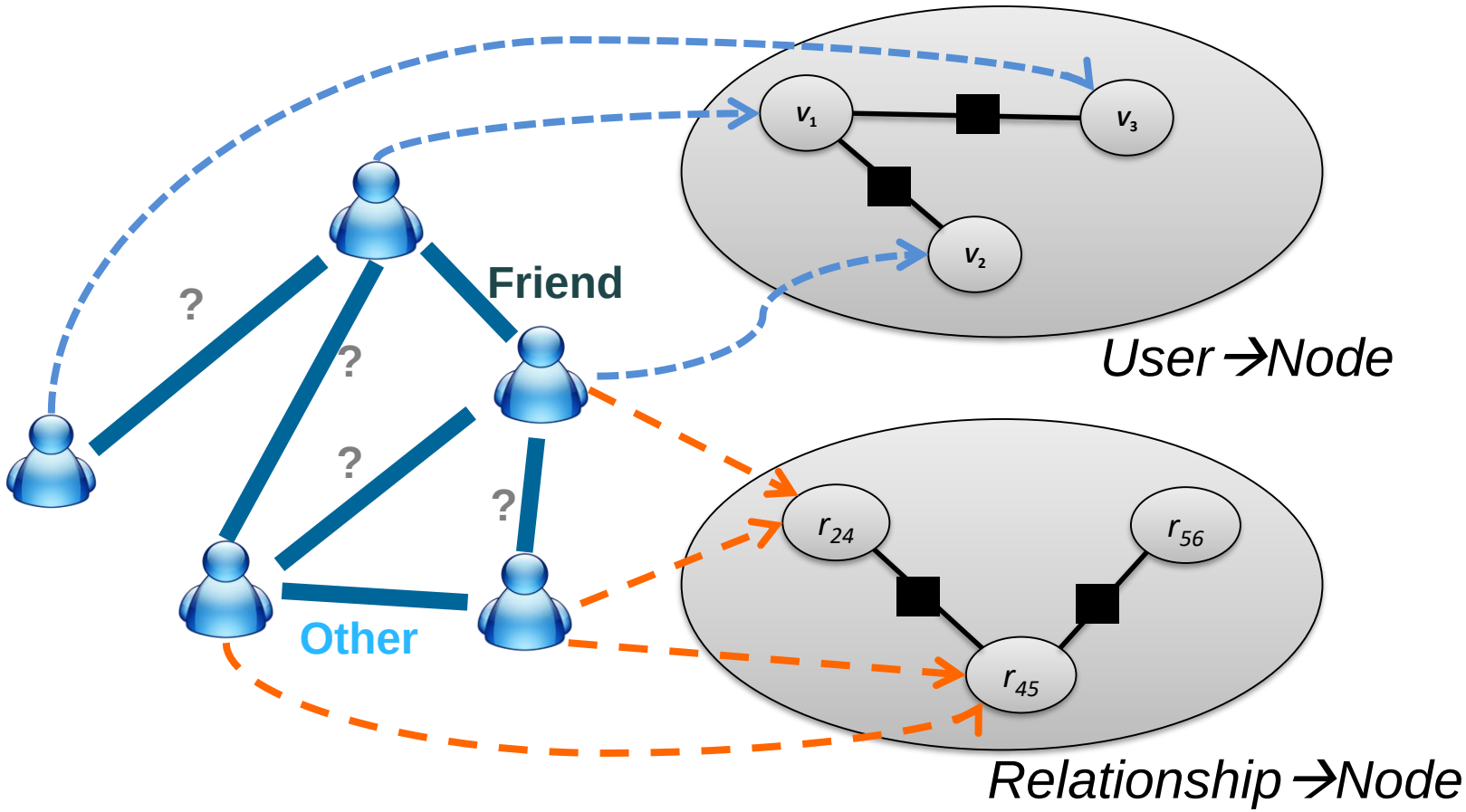
V: Set of Users

E^L,R^L: Labeled relationships

E^U: Unlabeled relationships

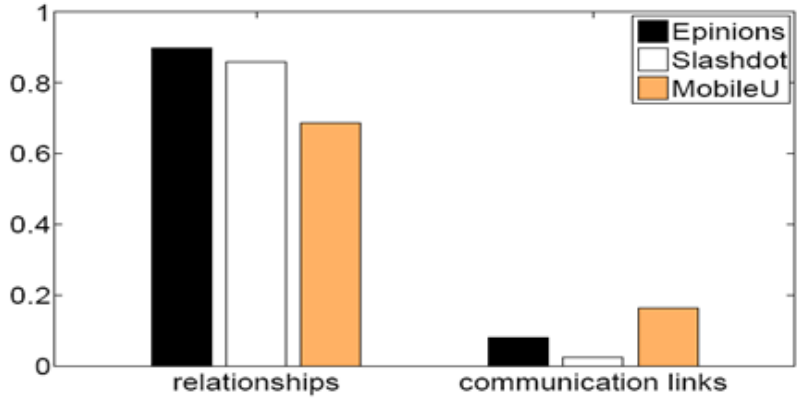
Input: $G=(V,E^L,E^U,R^L,W)$ → Output: $f: G \rightarrow R$

Basic Idea

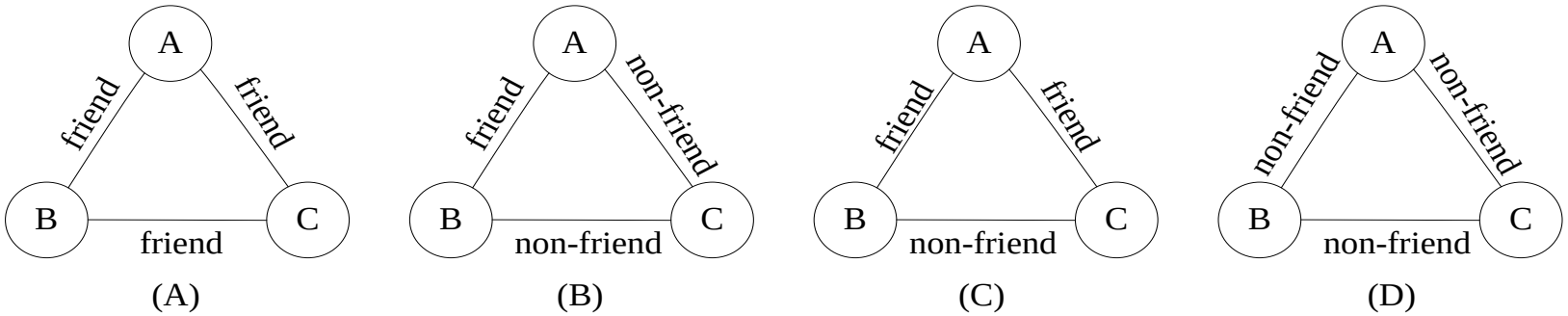

Friend
?
?
?
?
Other
v_1
v_3
v_2
User→Node
r_{24}
r_{56}
r_{45}
Relationship→Node

Social Theories

- Social balance theory
- Structural hole theory
- Social status theory
- Two-step-flow theory

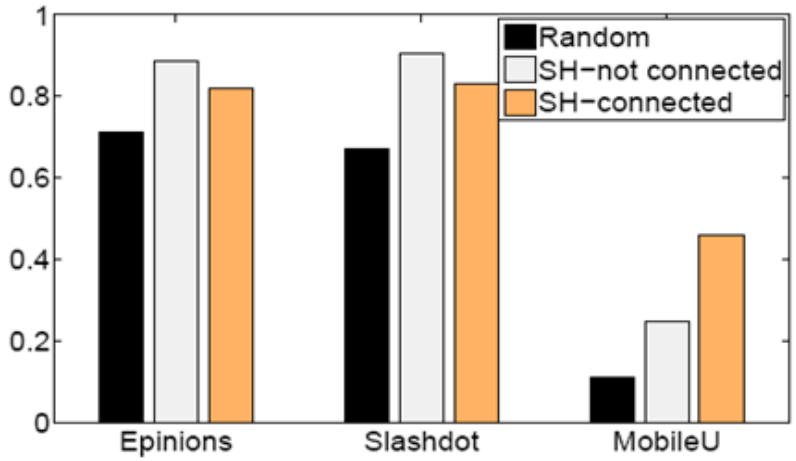


Observations:
(1) The underlying networks are unbalanced;
(2) While the friendship networks are balanced.

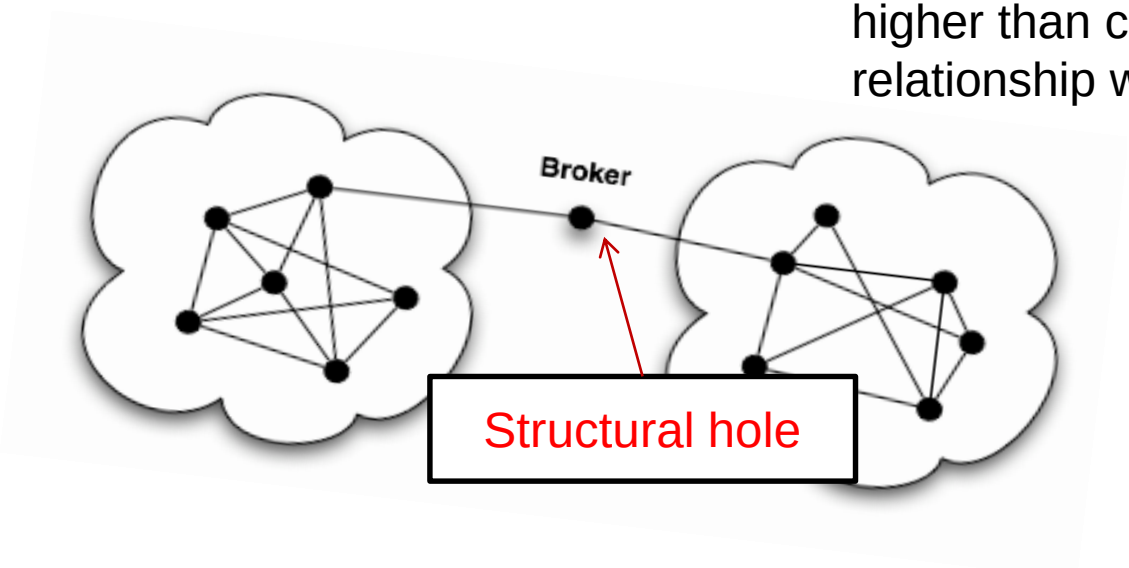


Social Theories—Structural hole

- Social balance theory
- Structural hole theory
- Social status theory
- Two-step-flow theory

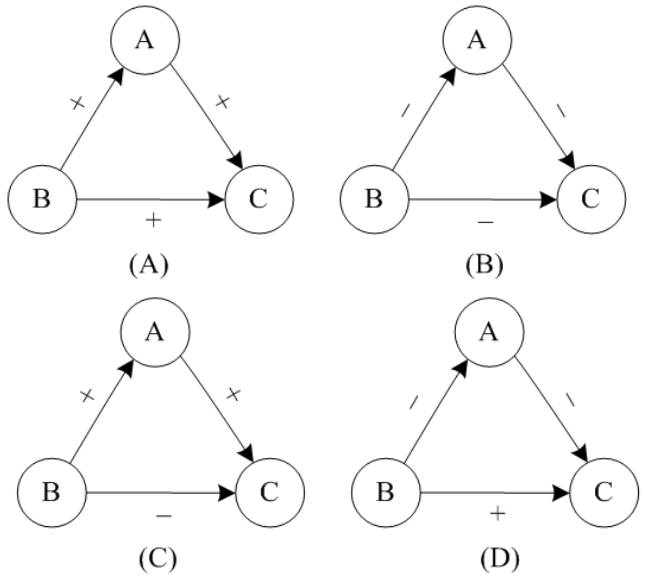


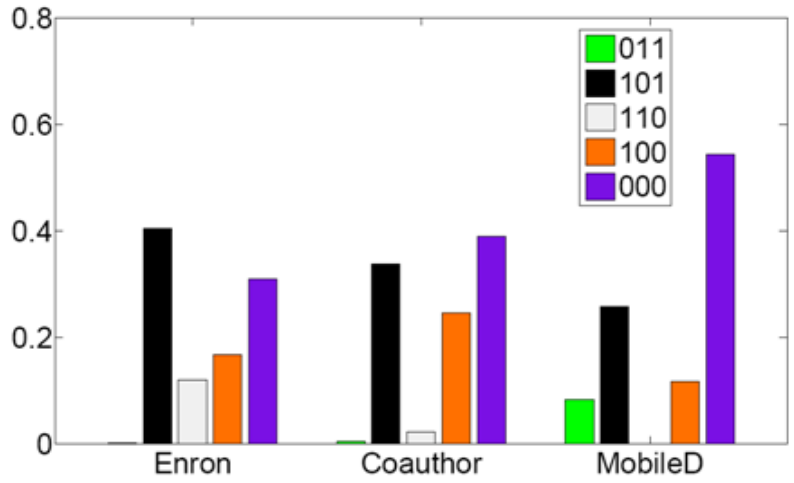
Observations: Users are more likely (+25-150% higher than change) to have the same type of relationship with C if C **spans structural holes**



Social Theories—Social status

- Social balance theory
- Structural hole theory
- Social status theory
- Two-step-flow theory



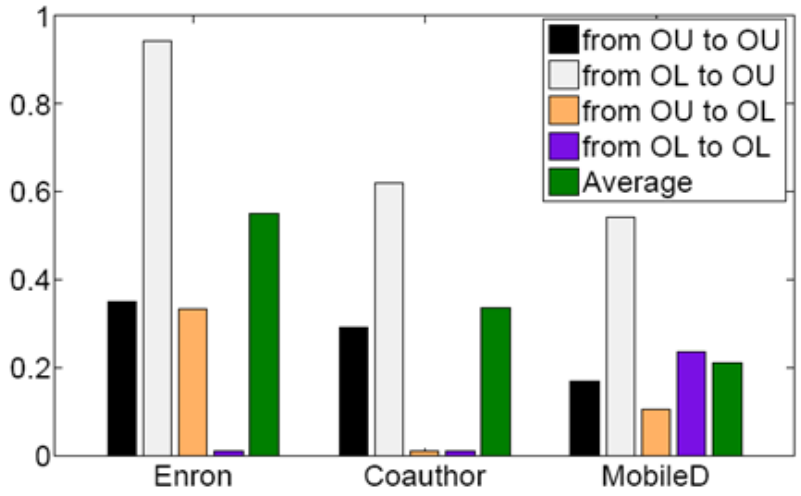


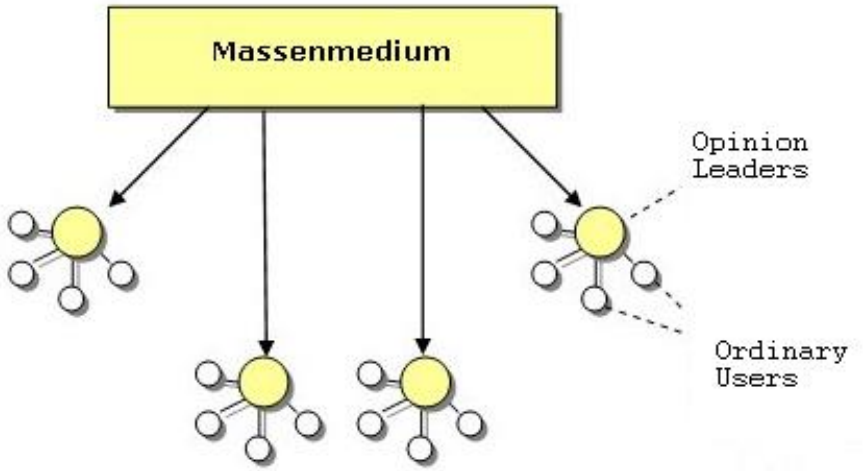
Observations: 99% of triads in the networks satisfy the social status theory

Note: Given a triad (A,B,C), let us use 1 to denote the advisor-advisee relationship and 0 colleague relationship. Thus the number 011 to denote A and B are colleagues, B is C's advisor and A is C's advisor.

Social Theories—Two-step-flow

- Social balance theory
- Structural hole theory
- Social status theory
- Two-step-flow theory



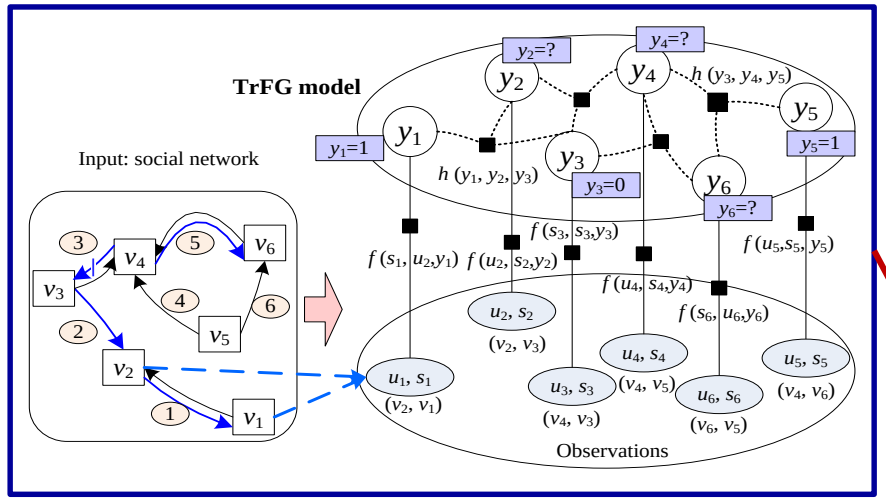


OL : Opinion leader;
OU : Ordinary user.

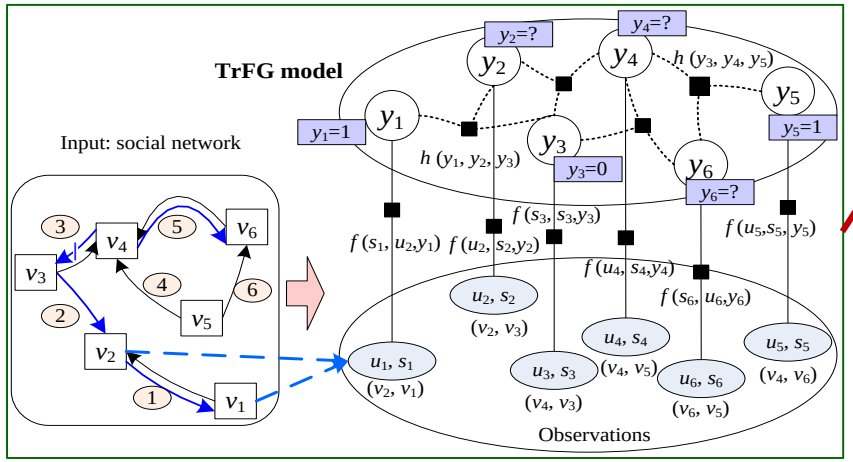
Observations: Opinion leaders are more likely (+71%-84% higher than chance) to have a higher social-status than ordinary users.

Transfer Factor Graph Model

Coauthor network

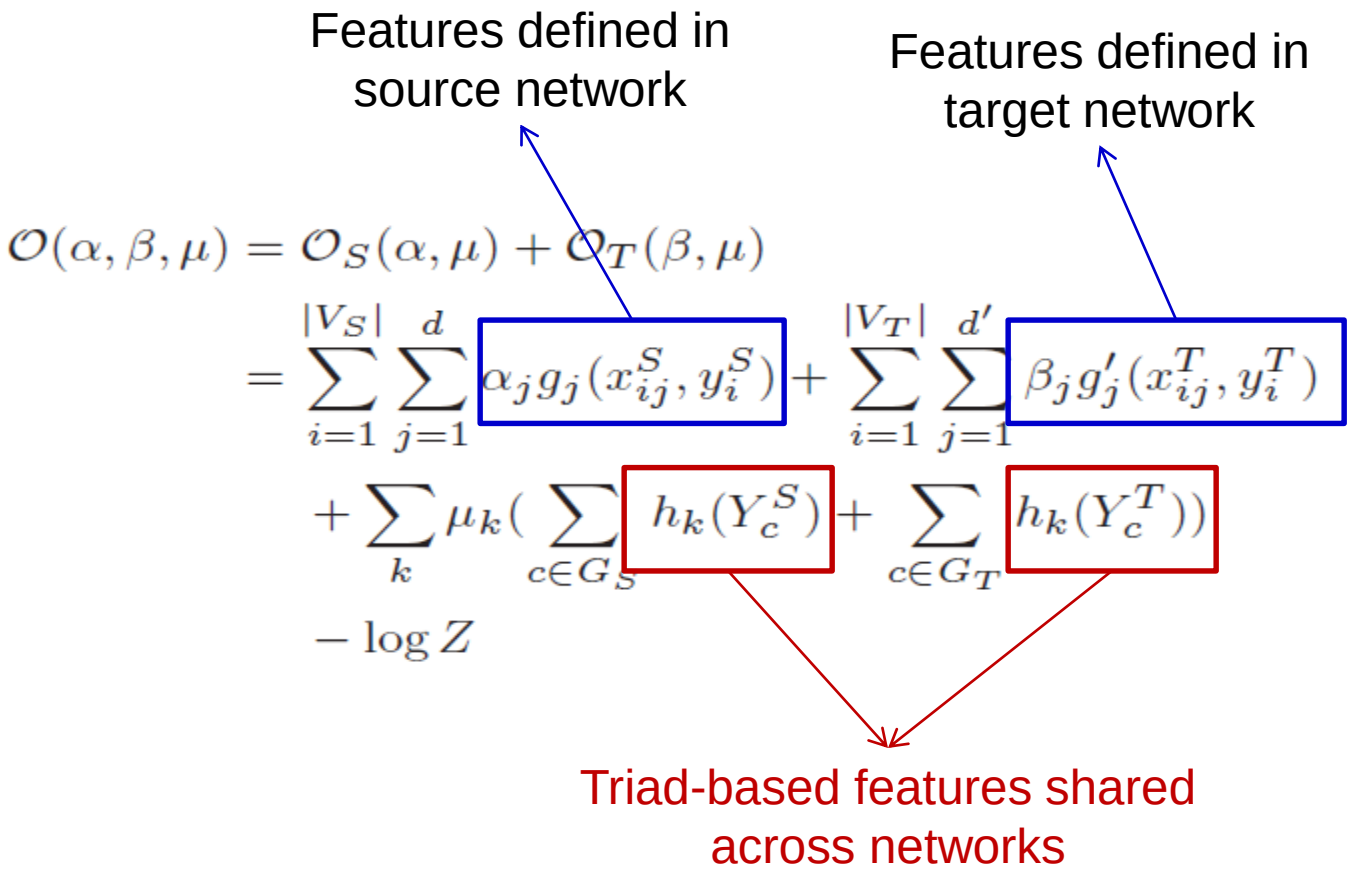


mobile



Bridge via social theories

Mathematical Formulation



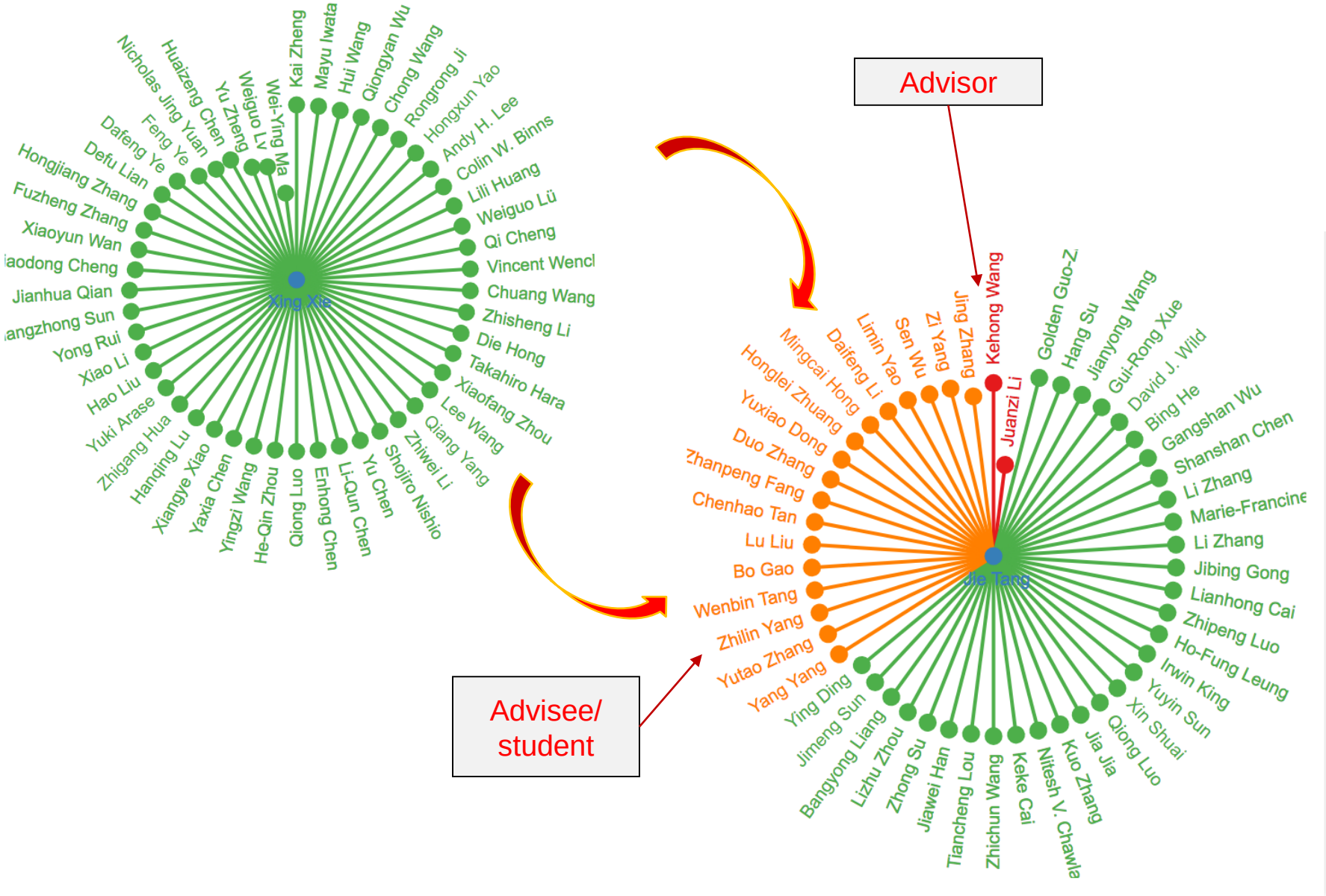
Jie Tang, Tiancheng Lou, and Jon Kleinberg. Inferring Social Ties across Heterogeneous Networks. In WSDM'2012. pp. 743-752.

Results – undirected networks

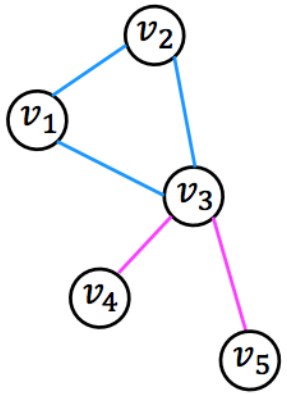
- **SVM** and **CRF** are two baseline methods
- **PFG** is the proposed partially-labeled factor graph model
- **TranFG** is the proposed transfer–based factor graph model.

Data Set	Method	Prec.	Rec.	F1-score
Epinions (S) to Slashdot (T) (40%)	SVM	0.7157	**0.9733**	0.8249
	CRF	0.8919	0.6710	0.7658
	PFG	0.9300	0.6436	0.7607
	TranFG	**0.9414**	0.9446	**0.9430**
Slashdot (S) to Epinions (T) (40%)	SVM	0.9132	**0.9925**	0.9512
	CRF	0.8923	0.9911	0.9393
	PFG	**0.9954**	0.9787	**0.9870**
	TranFG	**0.9954**	0.9787	**0.9870**
Epinions (S) to Mobile (T) (40%)	SVM	0.8983	0.5955	0.7162
	CRF	0.9455	0.5417	0.6887
	PFG	**1.0000**	0.5924	0.7440
	TranFG	0.8239	**0.8344**	**0.8291**
Slashdot (S) to Mobile (T) (40%)	SVM	0.8983	0.5955	0.7162
	CRF	0.9455	0.5417	0.6887
	PFG	**1.0000**	0.5924	0.7440
	TranFG	0.7258	**0.8599**	**0.7872**

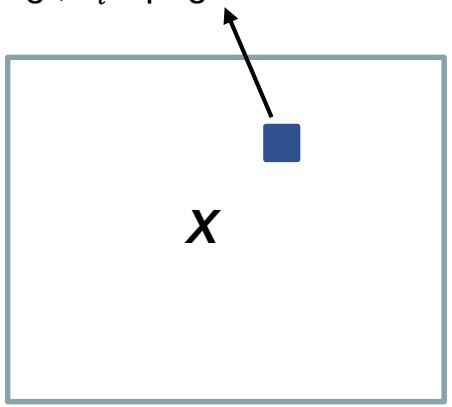
Inferring Social Ties in AMiner



The network analysis paradigm so far



x_{ij}: node v_i's j^{th} feature,
e.g., v_i's pagerank value



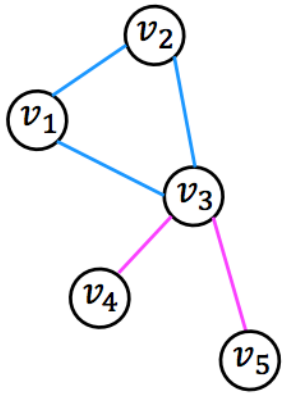
hand-crafted feature matrix

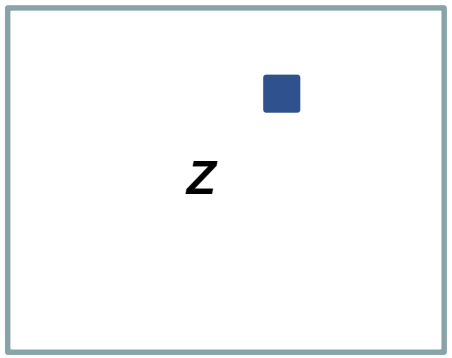
Graph & network applications
- Node label inference;
- Link prediction;
- User behavior… …

feature **engineering**

machine learning models

Representation learning for networks?





~~hand-crafted~~ **latent** feature matrix

Graph & network applications
- Node label inference;
- Node clustering;
- Link prediction;
- … …

Feature ~~engineering~~ **learning**

machine learning models

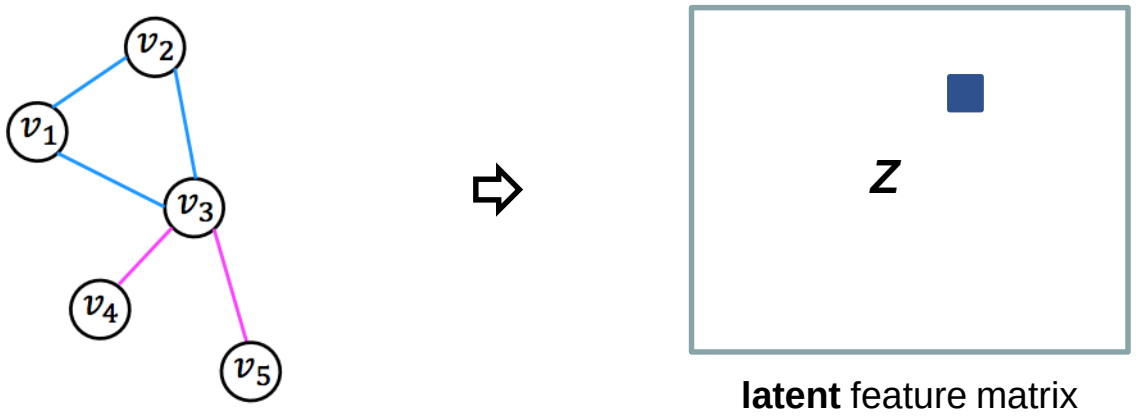
- Bengio, Courville, Vincent. Representation learning: A review and new perspectives. *IEEE TPAMI* 2013.
- LeCun, Bengio, Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

Representation learning for networks?

Problem (Graph representation learning, network embedding, graph embedding)

- Input: a network $G = (V, E)$
- Output: $\boldsymbol{Z} \in R^{|V| \times k}, k \ll |V|$, k-dim vector $\boldsymbol{Z}_v$ for each node *v*.

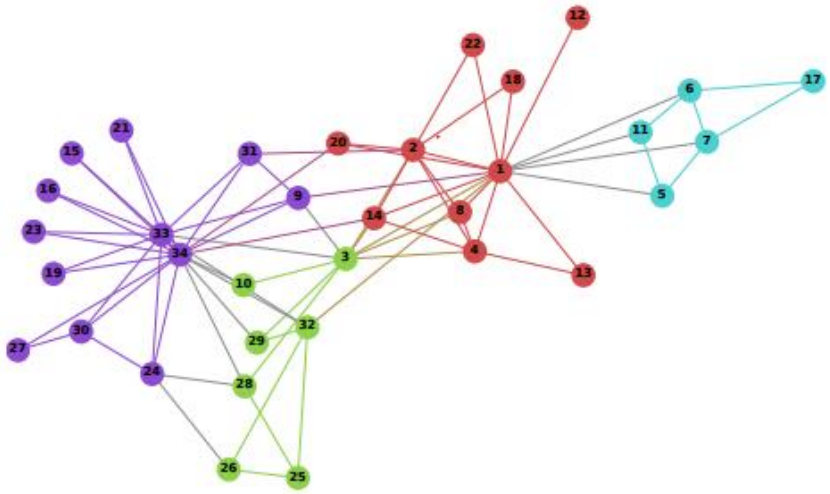
The goal is to map each node into a latent low-dimension space such that network structure information is encoded into distributed node representations



latent feature matrix

Feature **learning**

Graph representation learning



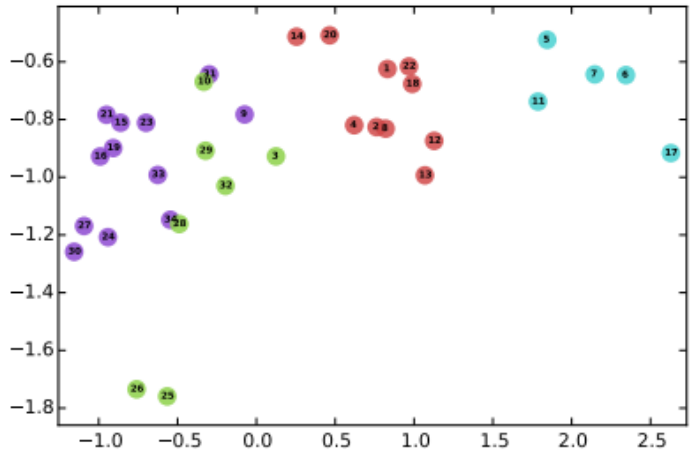
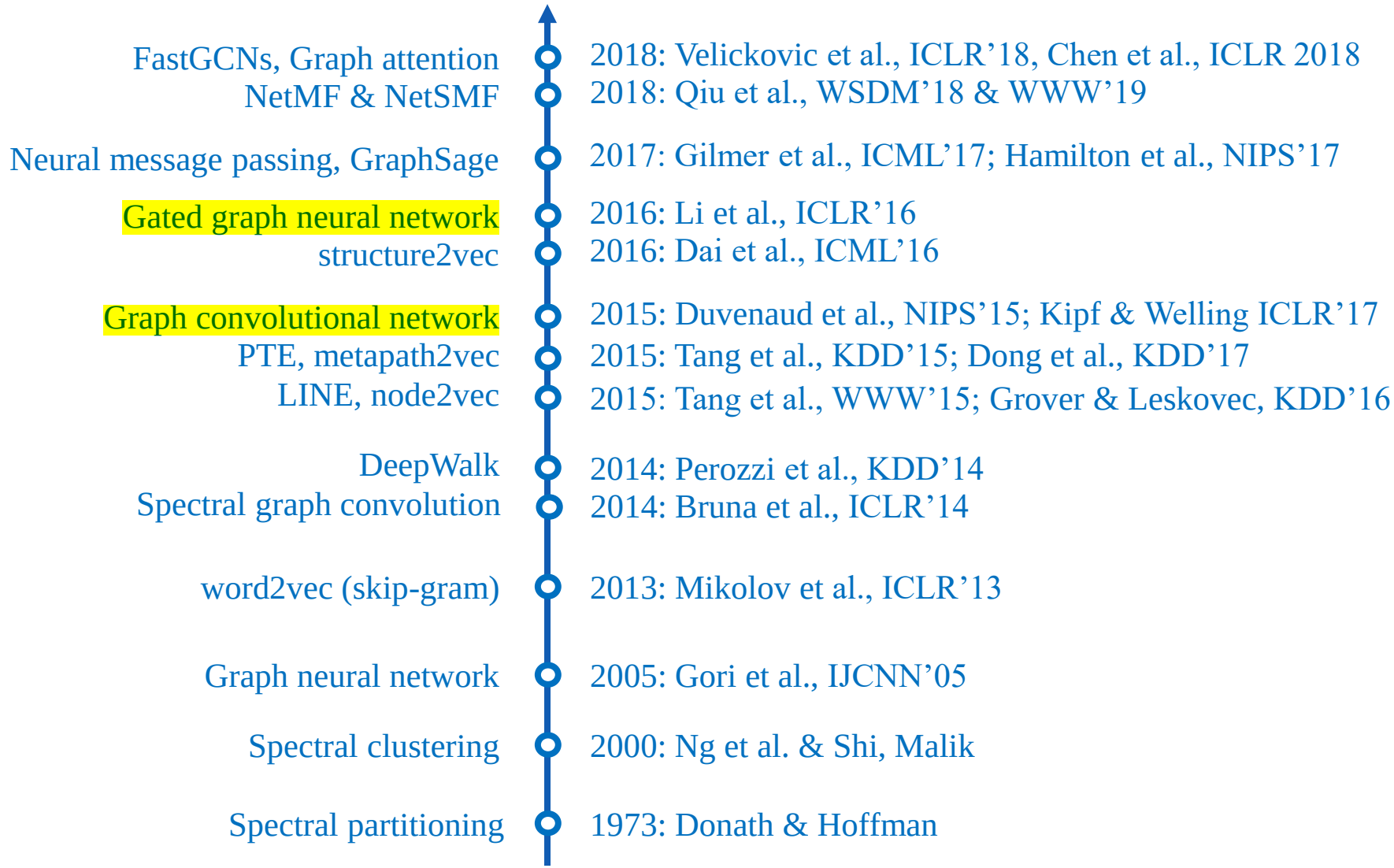


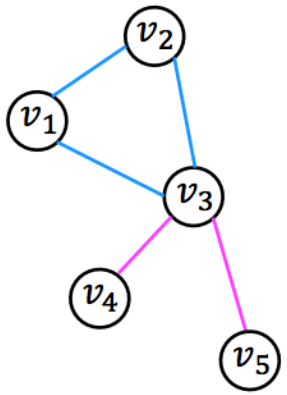
Image Credit: Perozzi et al. DeepWalk: Online learning of social representations. In *KDD' 14*, pp. 701–710.

Network Representation Learning / Network Embedding



Network Representation Learning

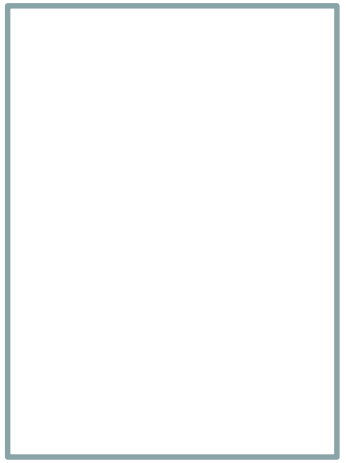
- Input: a graph $G = (V, E, \boldsymbol{A})$
- Output: $\boldsymbol{Z} \in R^{|V| \times k}, k \ll |V|$, k-dim vector $\boldsymbol{Z}_v$ for each node v.



⇨

?

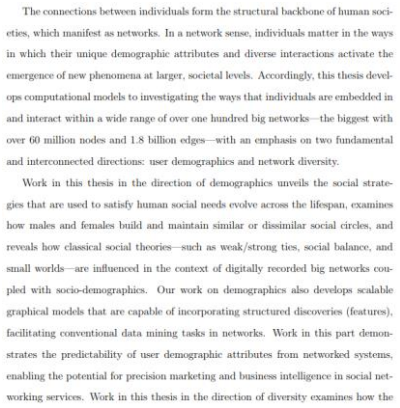
Feature **learning**



latent feature matrix

Word embedding in NLP

- Input: a text corpus $D = \{W\}$
- Output: $\boldsymbol{X} \in R^{|W| \times d}, d \ll |W|$, d-dim vector $\boldsymbol{X}_w$ for each word *w.*

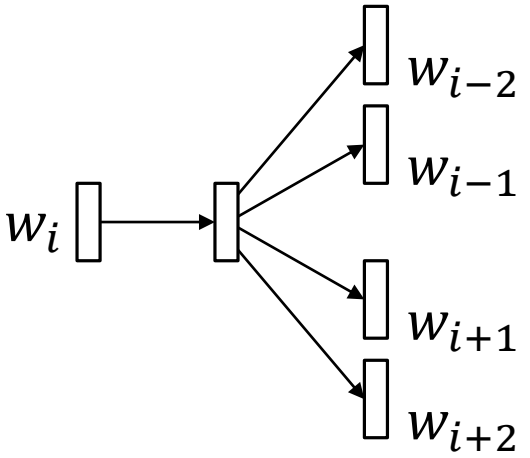


The connections between individuals form the structural backbone of human societies, which manifest as networks. In a network sense, individuals matter in the ways in which their unique demographic attributes and diverse interactions activate the emergence of new phenomena at larger, societal levels. Accordingly, this thesis develops computational models to investigating the ways that individuals are embedded in and interact within a wide range of over one hundred big networks—the biggest with over 60 million nodes and 1.8 billion edges—with an emphasis on two fundamental and interconnected directions: user demographics and network diversity.

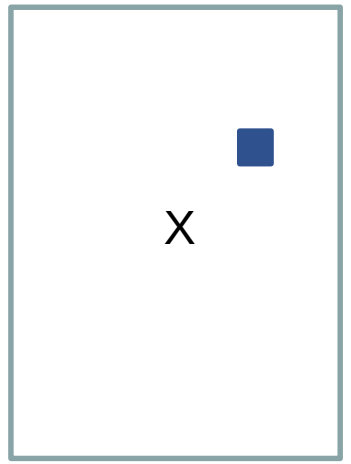
Work in this thesis in the direction of demographics unveils the social strategies that are used to satisfy human social needs evolve across the lifespan, examines how males and females build and maintain similar or dissimilar social circles, and reveals how classical social theories—such as weak/strong ties, social balance, and small worlds—are influenced in the context of digitally recorded big networks coupled with socio-demographics. Our work on demographics also develops scalable graphical models that are capable of incorporating structured discoveries (features), facilitating conventional data mining tasks in networks. Work in this part demonstrates the predictability of user demographic attributes from networked systems, enabling the potential for precision marketing and business intelligence in social networking services. Work in this thesis in the direction of diversity examines how the

- Computational lens on big social and information networks.
- The connections between individuals form the structural …
- In a network sense, individuals matters in the ways in which ...
- Accordingly, this thesis develops computational models to investigating the ways that ...
- We study two fundamental and interconnected directions: user demographics and network diversity
- ... ...

sentences



Word embedding models



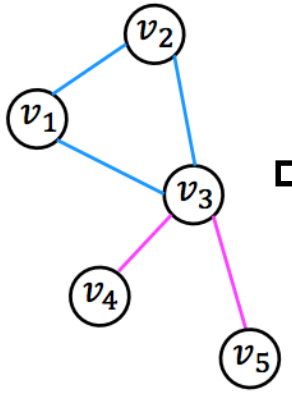
latent feature matrix

- Harris' distributional hypothesis: words in similar contexts have similar meanings.
- Key idea: try to predict the words that surrounding each one.

1. Bengio, et al. Representation learning: A review and new perspectives. In *IEEE TPAMI 2013.*
2. Mikolov, et al. Efficient estimation of word representations in vector space. In *ICLR 2013*.

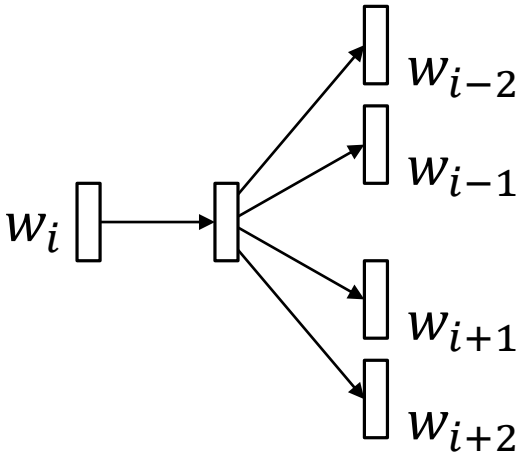
Network Representation Learning

- Input: a network $G = (V, E)$
- Output: $\boldsymbol{X} \in R^{|V| \times k}, k \ll |V|$, k-dim vector $\boldsymbol{X}_v$ for each node v.



sentences

- Computational lens on big social and information networks.
- The connections between individuals form the structural …
- In a network sense, individuals matters in the ways in which ...
- Accordingly, this thesis develops computational models for investigating the ways that ...



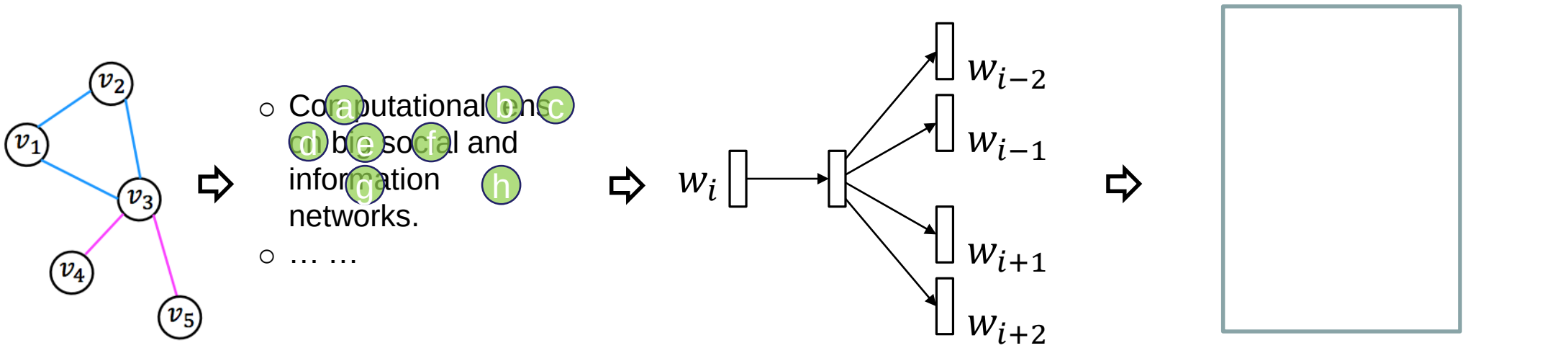
skip-gram

latent feature matrix

Feature **learning**

Network Representation Learning

- Input: a network $G = (V, E)$
- Output: $\boldsymbol{X} \in R^{|V| \times k}, k \ll |V|$, k-dim vector $\boldsymbol{X}_v$ for each node v.

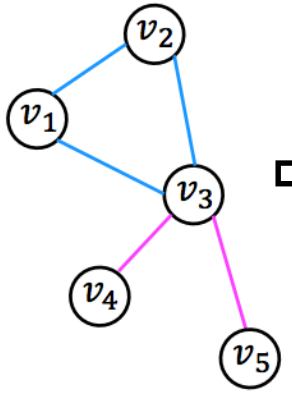


sentences skip-gram ~~hand-crafted~~ **latent** feature matrix

Feature ~~engineering~~ **learning**

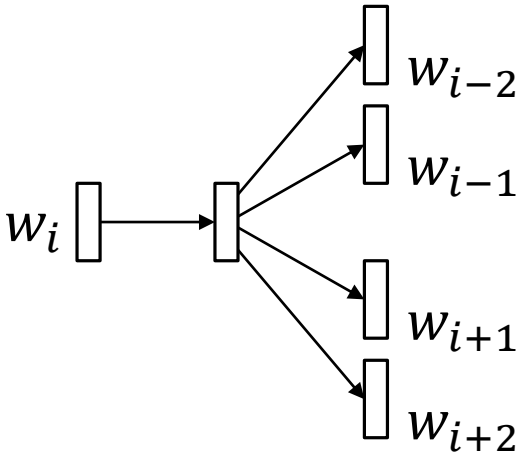
Network Representation Learning

- Input: a network $G = (V, E)$
- Output: $\boldsymbol{X} \in R^{|V| \times k}, k \ll |V|$, k-dim vector $\boldsymbol{X}_v$ for each node v.

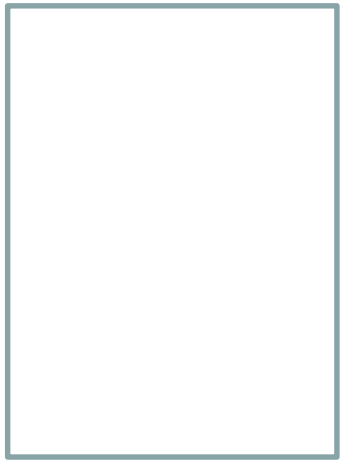


sentences

- Computational lens on big social and information networks.
- … …



skip-gram

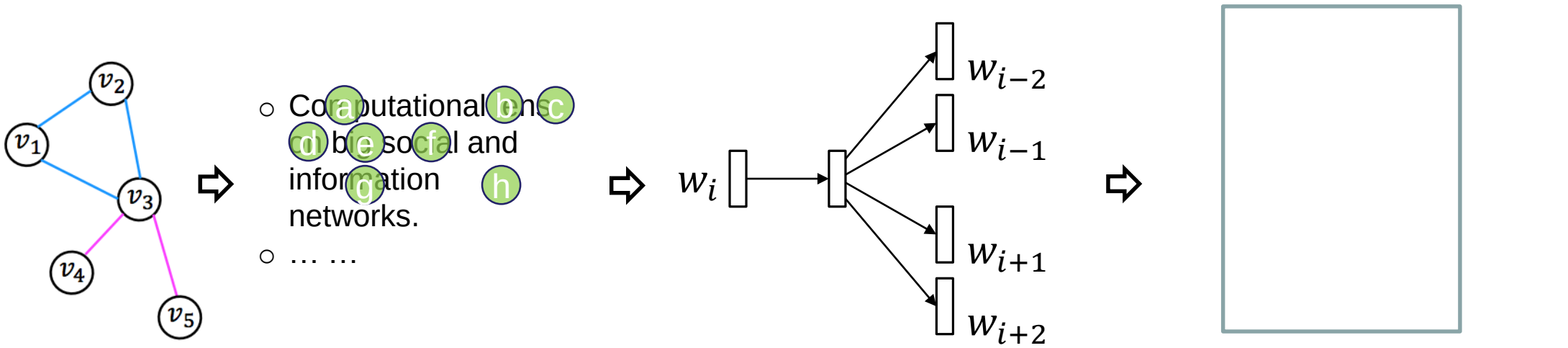


~~hand-crafted~~ **latent** feature matrix

Feature ~~engineering~~ **learning**

Network Representation Learning

- Input: a network $G = (V, E)$
- Output: $\boldsymbol{X} \in R^{|V| \times k}, k \ll |V|$, k-dim vector $\boldsymbol{X}_v$ for each node v.

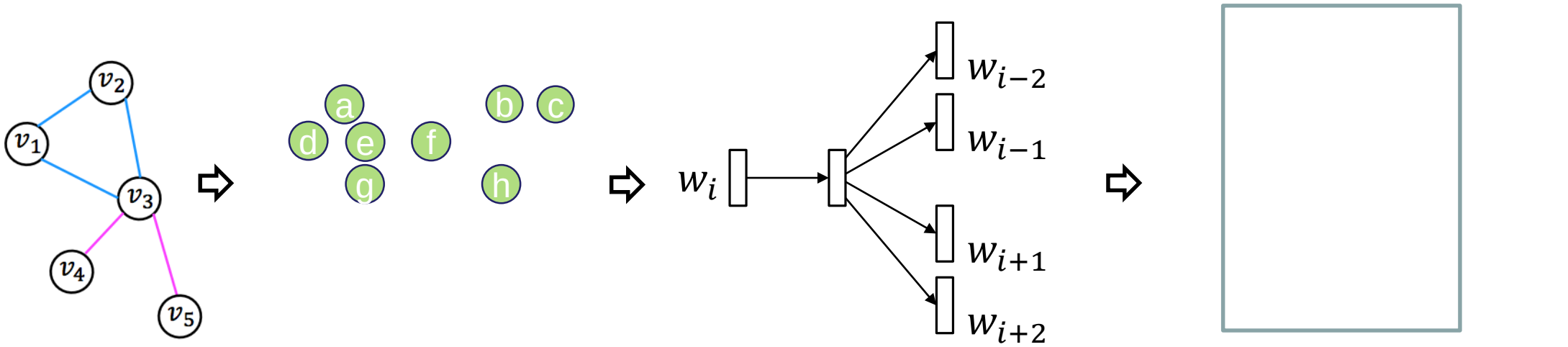


sentences skip-gram ~~hand-crafted~~ **latent** feature matrix

Feature ~~engineering~~ **learning**

Network Representation Learning

- Input: a network $G = (V, E)$
- Output: $\boldsymbol{X} \in R^{|V| \times k}, k \ll |V|$, k-dim vector $\boldsymbol{X}_v$ for each node v.

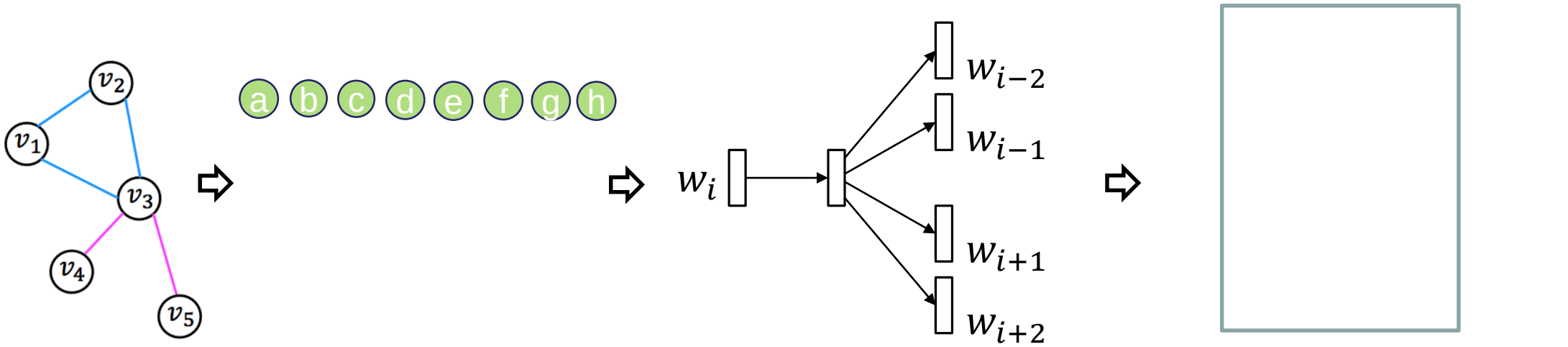


sentences skip-gram ~~hand-crafted~~ **latent** feature matrix

Feature ~~engineering~~ **learning**

Network embedding: DeepWalk

- Input: a network $G = (V, E)$
- Output: $\boldsymbol{X} \in R^{|V| \times k}, k \ll |V|$, k-dim vector $\boldsymbol{X}_v$ for each node v.



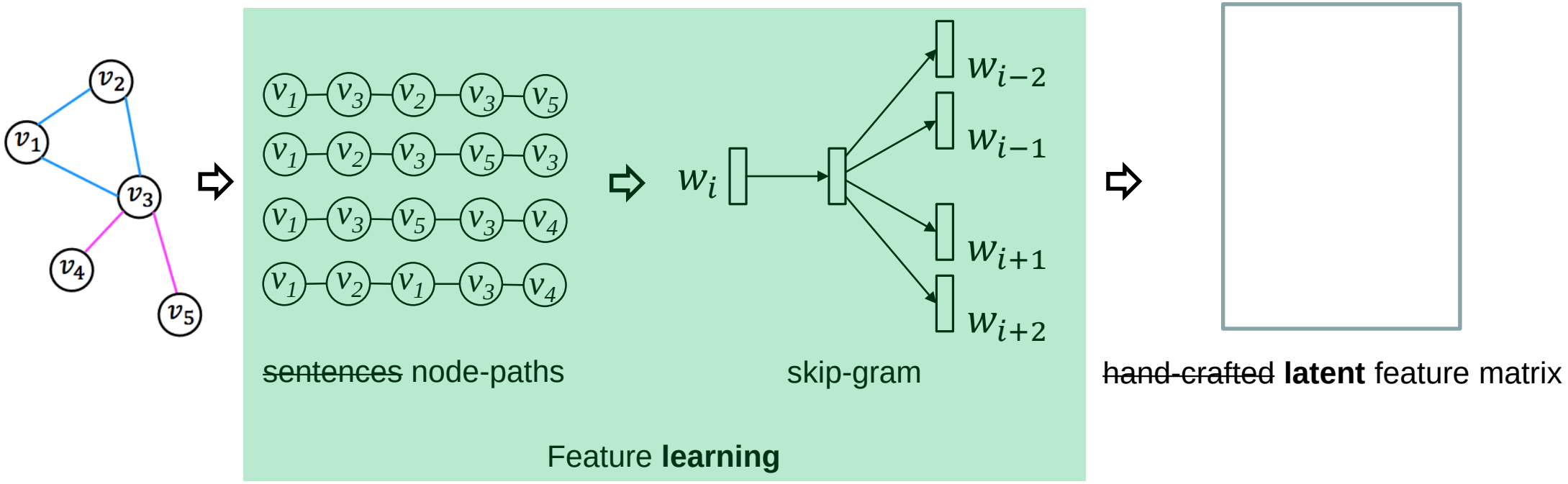
~~sentences~~ node-paths skip-gram ~~hand-crafted~~ **latent** feature matrix

Feature ~~engineering~~ **learning**

1. Perozzi et al. DeepWalk: Online learning of social representations. In *KDD'14*, pp. 701–710. **Most Cited Paper in KDD'14.**

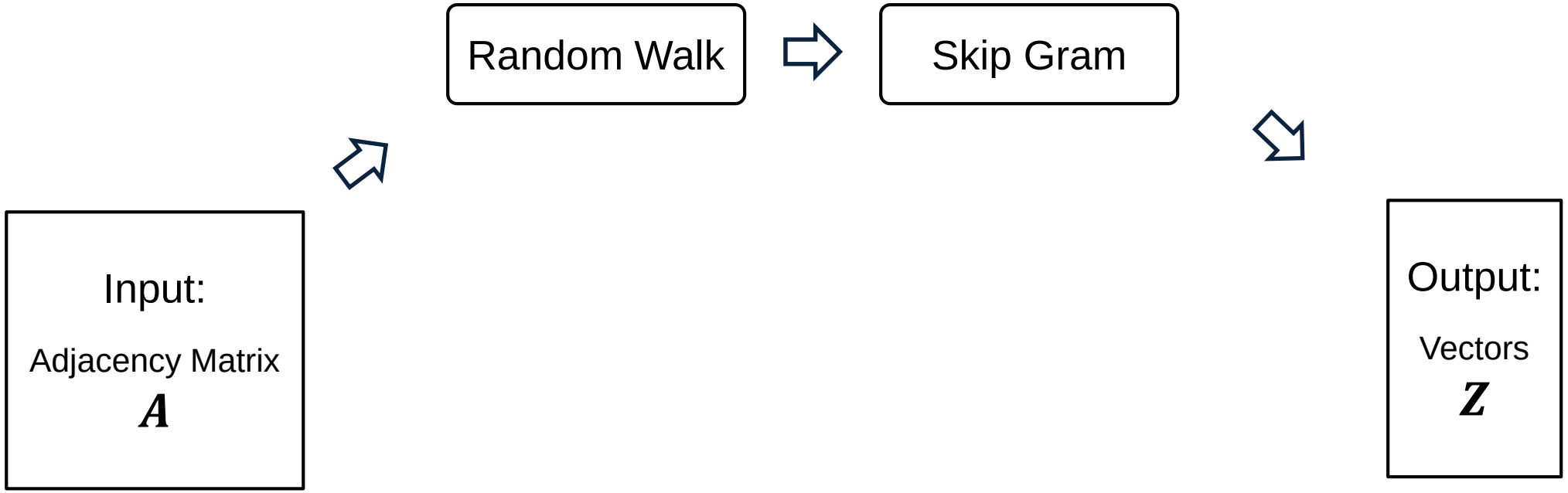
Network embedding: DeepWalk

- Input: a network $G = (V, E)$
- Output: $\boldsymbol{X} \in R^{|V| \times k}, k \ll |V|$, k-dim vector $\boldsymbol{X}_v$ for each node v.

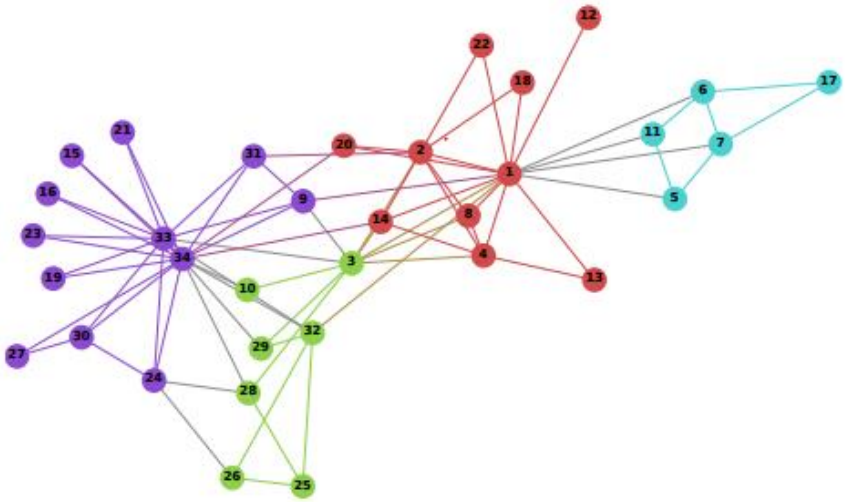


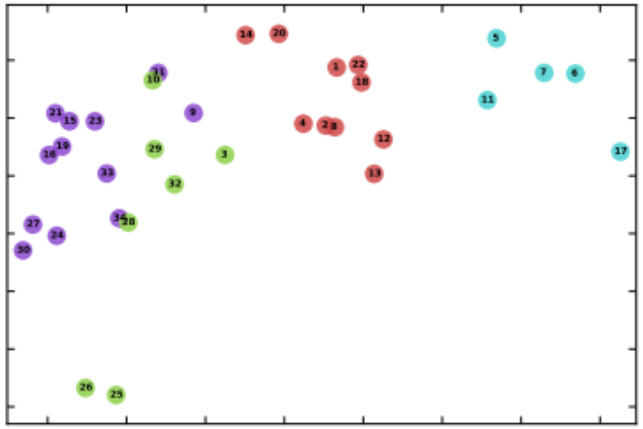
1. Perozzi et al. DeepWalk: Online learning of social representations. In *KDD'14*, pp. 701–710. **Most Cited Paper in KDD'14.**

Network Embedding



Network embedding: DeepWalk





Graph & network applications

- Node label inference;
- Node clustering;
- Link prediction;
- … …

Image Credit: Perozzi et al. DeepWalk: Online learning of social representations. In *KDD' 14*, pp. 701–710.

Distributional Hypothesis of Harris

- **Word embedding**: words in similar contexts have similar meanings (e.g., skip-gram in word embedding)

- **Node embeddings**: nodes in similar structural contexts are similar
  - DeepWalk: structural contexts are defined by co-occurrence over random walk paths

The learning problem

$$\mathcal{L} = \sum_{v \in V} \sum_{c \in N_{rw}(v)} -\log(P(c|v))$$

- To maximize the likelihood of node co-occurrence on a random walk path

The learning problem

$$\mathcal{L} = \sum_{v \in V} \sum_{c \in N_{rw}(v)} -\log(P(c|v)) \quad \Leftarrow \quad p(c|v) = \frac{\exp(\boldsymbol{z}_v^\top \boldsymbol{z}_c)}{\sum_{u \in V} \exp(\boldsymbol{z}_v^\top \boldsymbol{z}_u)}$$

word2vec addresses the $O(|V|^2)$ complexity of skip-gram by

- Hierarchical softmax (used in DeepWalk)
- Negative sampling (used in node2vec/LINE)

The main idea behind

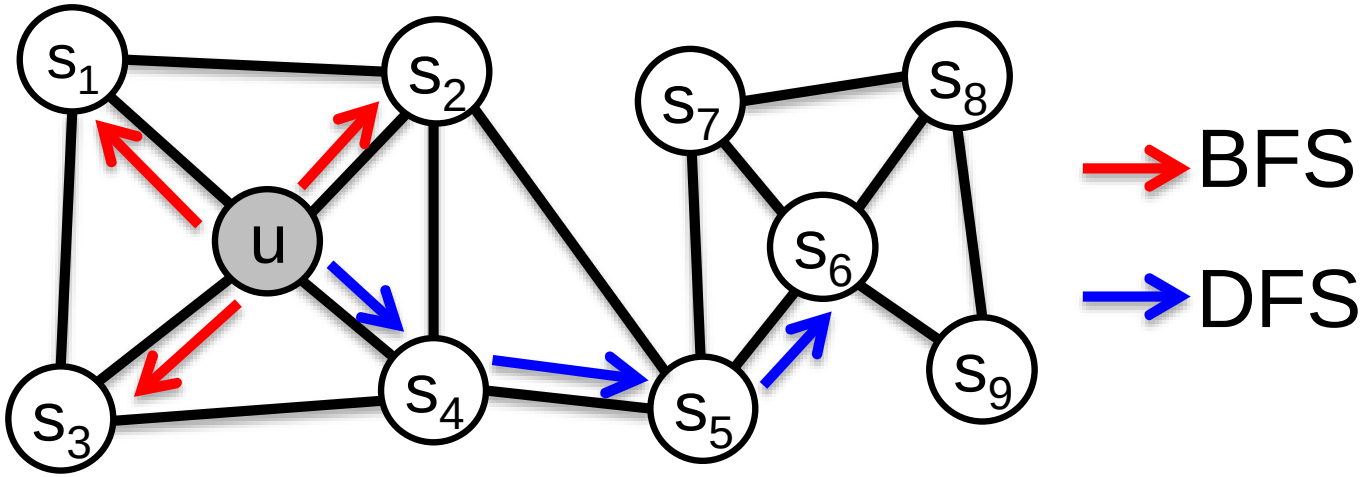
$$\mathcal{L} = \sum_{v \in V} \sum_{c \in N_{rw}(v)} -\log(P(c|v)) \quad \Leftarrow \quad p(c|v) = \frac{\exp(\boldsymbol{z}_v^\top \boldsymbol{z}_c)}{\sum_{u \in V} \exp(\boldsymbol{z}_v^\top \boldsymbol{z}_u)}$$

$\boldsymbol{z}_v^\top \boldsymbol{z}_c$ → the probability that node v and context c appear on a random walk path

Random Walk Strategies

- Random Walk
  - DeepWalk
- Biased Random Walk
  - 2nd order Random Walk
    - node2vec
  - Metapath guided Random Walk
    - metapath2vec

node2vec



$$N_{BFS}(u) = \{ s_1, s_2, s_3 \}$$

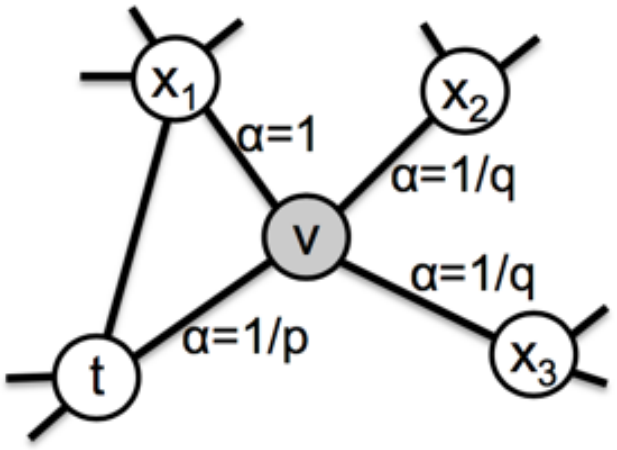
$$N_{DFS}(u) = \{ s_4, s_5, s_6 \}$$

A. Grover and J. Leskovec. 2016. node2vec: Scalable feature learning for networks. *KDD*, 855–864.

node2vec

Biased random walk R that given a node v generates random walk neighborhood $N_{rw}(v)$

- Return parameter p:
  - Return back to the previous node
- In-out parameter q:
  - Moving outwards (DFS) vs. inwards (BFS)

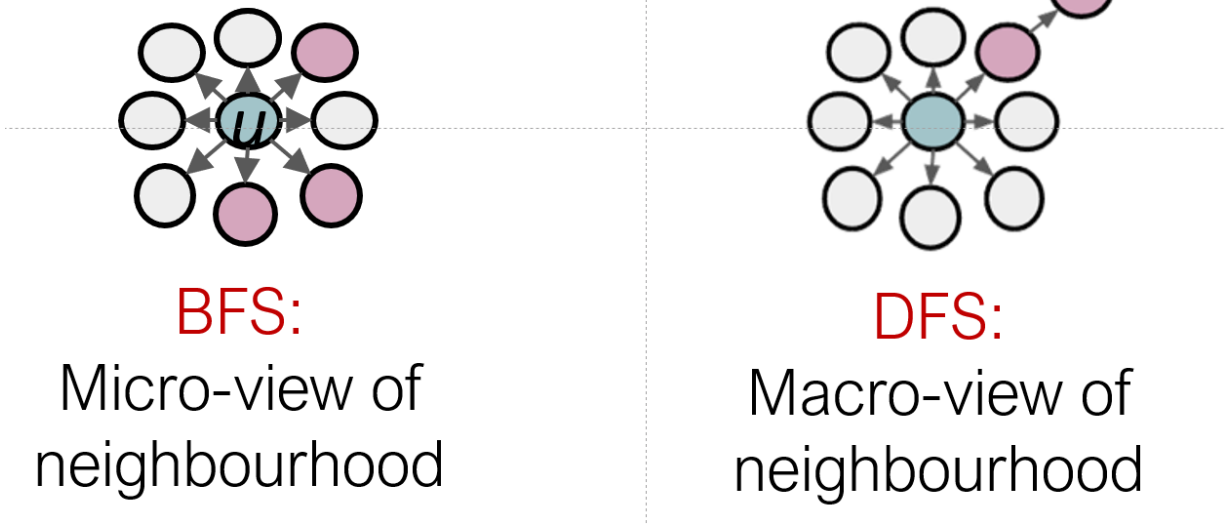


Picture snipped from Leskovec

node2vec

Biased random walk R that given a node v generates random walk neighborhood $N_{rw}(v)$

- Return parameter p:
  - Return back to the previous node
- In-out parameter q:
  - Moving outwards (DFS) vs. inwards (BFS)



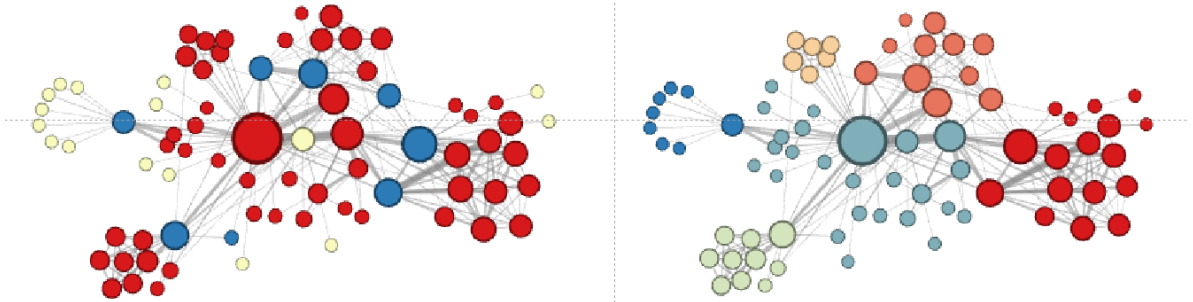
Picture snipped from Leskovec

node2vec

Biased random walk R that given a node v generates random walk neighborhood $N_{rw}(v)$

- Return parameter p:
  - Return back to the previous node
- In-out parameter q:
  - Moving outwards (DFS) vs. inwards (BFS)

Interactions of characters in a novel:



p=1, q=2
Microscopic view of the network neighbourhood

p=1, q=0.5
Macroscopic view of the network neighbourhood

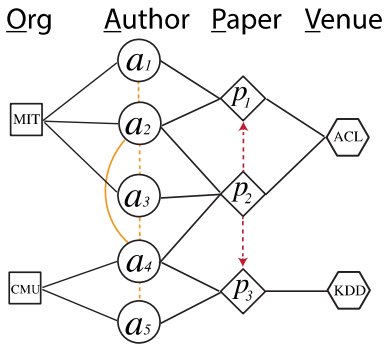
Picture snipped from Leskovec

Random Walk Strategies

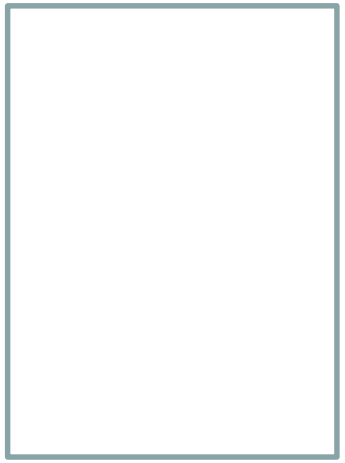
- Random Walk
  - DeepWalk
- Biased Random Walk
  - 2nd order Random Walk
    - node2vec
  - Metapath guided Random Walk
    - metapath2vec

Heterogeneous random walk

- Input: a heterogeneous graph $G = (V, E)$
- Output: $\boldsymbol{X} \in R^{|V| \times k}, k \ll |V|$, k-dim vector $\boldsymbol{X}_v$ for each node *v.*



- How do we random walk over heterogeneous networks?
- How do we apply skip-gram over different types of nodes?



~~hand-crafted~~ **latent** feature matrix

Feature **learning**

Microsoft Academic Graph

219,352,601
Papers

664,190
Topics

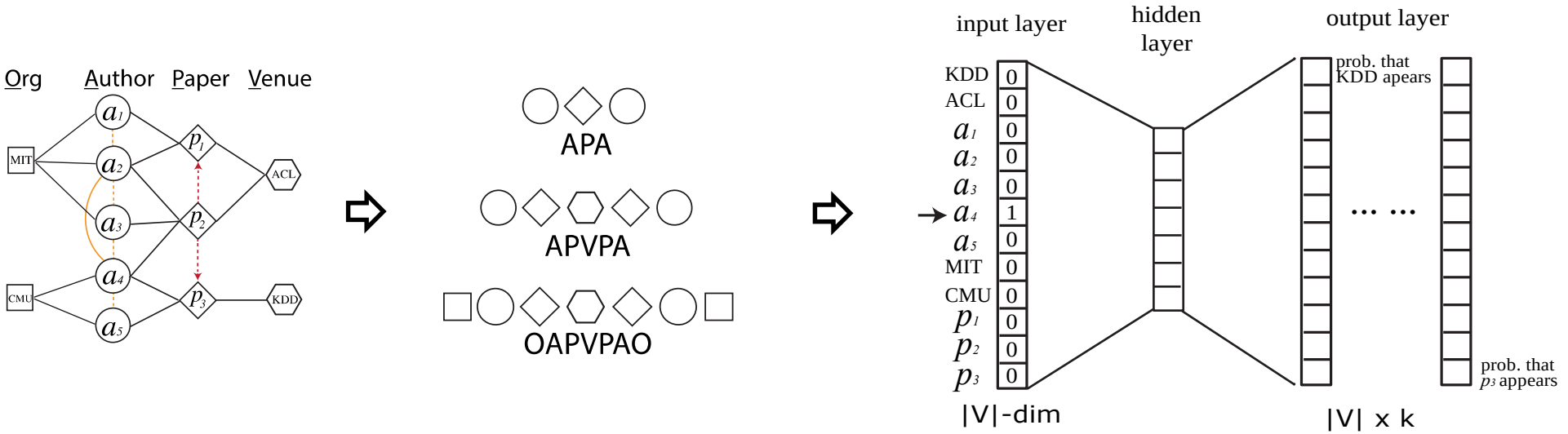
48,731
Journals

239,952,453
Authors

4,388
Conferences

25,509
Institutions

Heterogeneous graph embedding

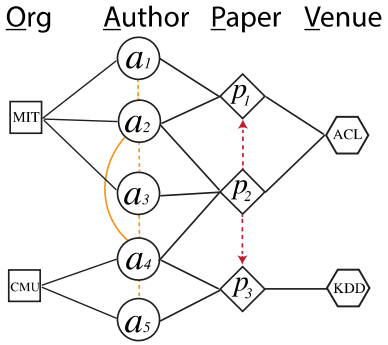


meta-path-based random walks

skip-gram

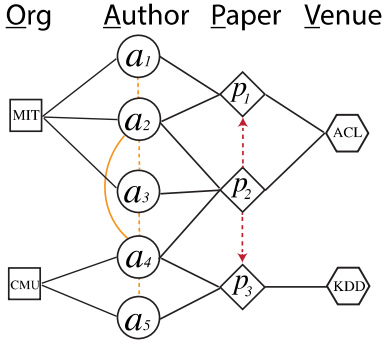
1. Sun and Han. Mining heterogeneous information networks: Principles and Methodologies. Morgan & Claypool Publishers, 2012.
2. Dong et al. metapath2vec: scalable representation learning for heterogeneous networks. In *ACM KDD* 2017. **The most cited paper in KDD'17 as of Aug 2018.**

Heterogeneous graph embedding



Goal: to generate paths that are able to capture both the semantic and structural correlations between different types of nodes, facilitating the transformation of heterogeneous network structures into skip-gram.

Heterogeneous graph embedding: Meta-Path-Based Random Walks



- Given a meta-path scheme

$$\mathcal{P}: V_1 \xrightarrow{R_1} V_2 \xrightarrow{R_2} \cdots V_t \xrightarrow{R_t} V_{t+1} \cdots \xrightarrow{R_{l-1}} V_l$$

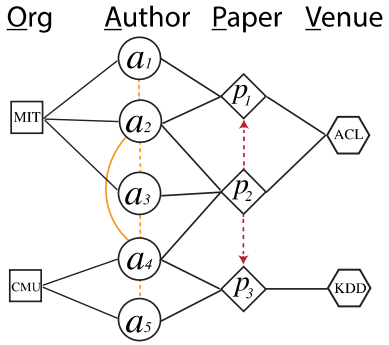
- The transition probability at step *i* is defined as

$$p(v^{i+1}|v_t^i, \mathcal{P}) = \begin{cases} \frac{1}{|N_{t+1}(v_t^i)|} & (v^{i+1}, v_t^i) \in E, \phi(v^{i+1}) = t+1 \\ 0 & (v^{i+1}, v_t^i) \in E, \phi(v^{i+1}) \neq t+1 \\ 0 & (v^{i+1}, v_t^i) \notin E \end{cases}$$

- Recursive guidance for random walkers, i.e.,

$$p(v^{i+1}|v_t^i) = p(v^{i+1}|v_1^i), \text{ if } t = l$$

Heterogeneous graph embedding: Meta-Path-Based Random Walks



- Given a meta-path scheme (Example)

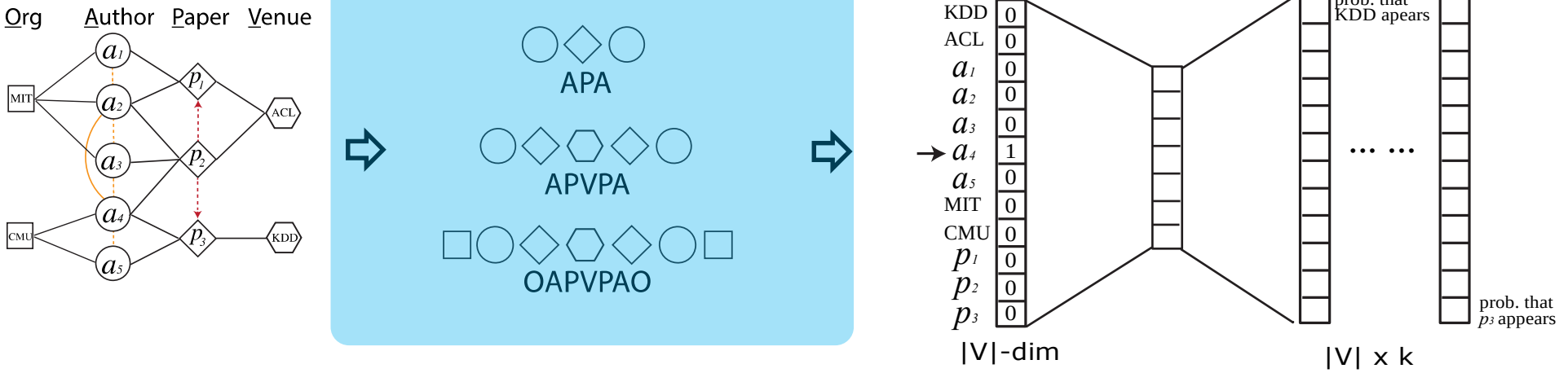
  OAPVPAO

- In a traditional random walk procedure, in the toy example, the next step of a walker on node a_4 transitioned from node O_{CMU} can be all types of nodes surrounding it— a_2, a_3, a_5, p_2, p_3 and O_{CMU}.

- Under the meta-path scheme ‘OAPVPAO’, for example, the walker is biased towards paper nodes (P) given its previous step on an organization node O_{CMU}(O), following the semantics of this meta-path.

Heterogeneous graph embedding

- Input: a heterogeneous graph $G = (V, E)$
- Output: $\boldsymbol{X} \in R^{|V| \times k}, k \ll |V|$, k-dim vector $\boldsymbol{X}_v$ for each node v.

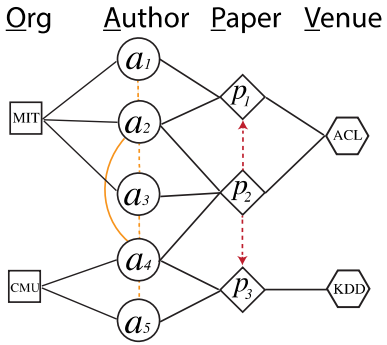


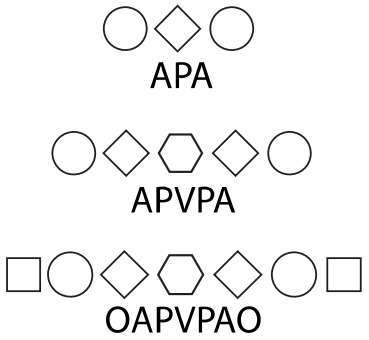
meta-path-based random walks

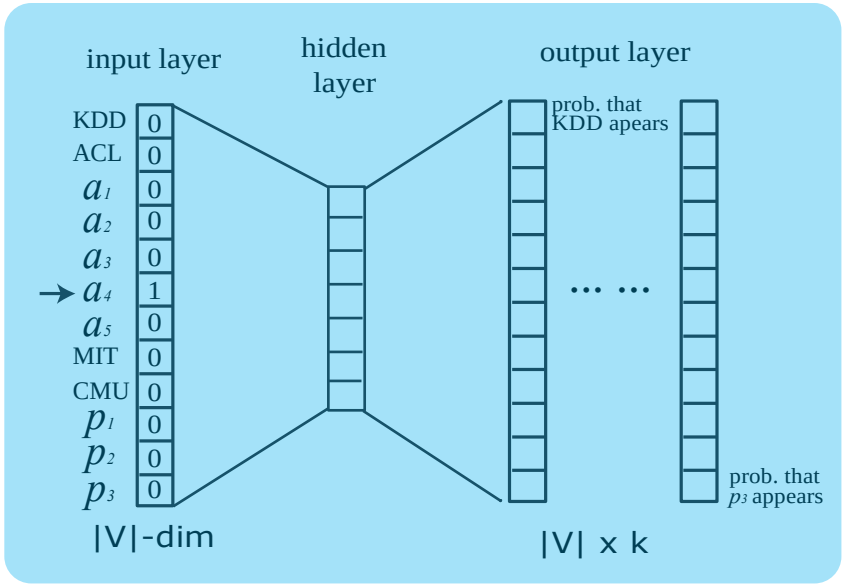
skip-gram

Heterogeneous graph embedding

- Input: a heterogeneous graph $G = (V, E)$
- Output: $\boldsymbol{X} \in R^{|V| \times k}, k \ll |V|$, k-dim vector $\boldsymbol{X}_v$ for each node v.





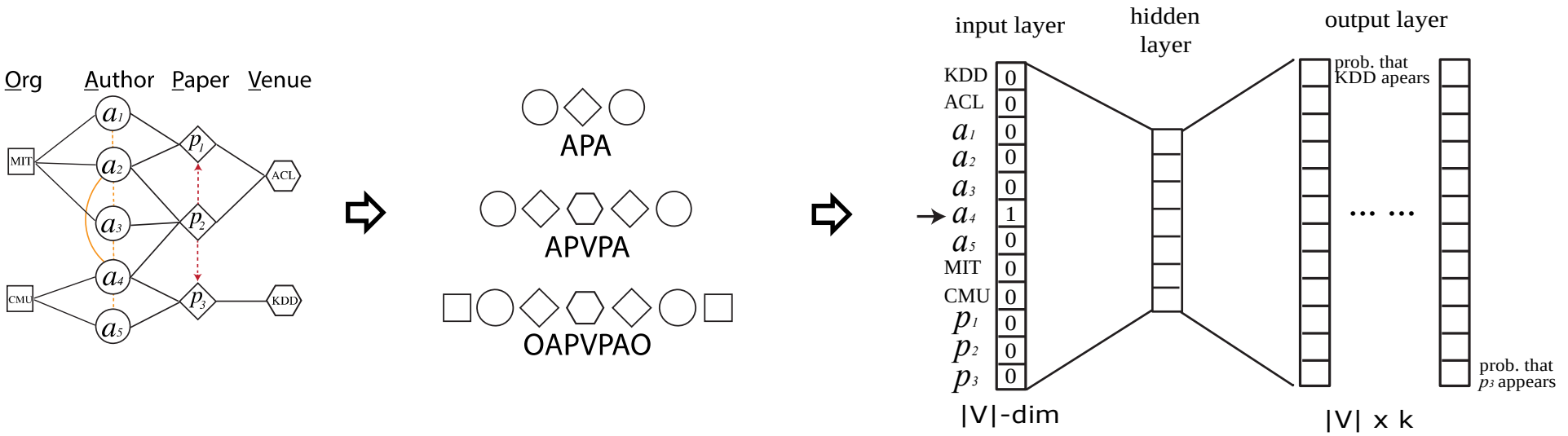


meta-path-based random walks

skip-gram

Heterogeneous graph embedding

- Input: a heterogeneous graph $G = (V, E)$
- Output: $\boldsymbol{X} \in R^{|V| \times k}, k \ll |V|$, k-dim vector $\boldsymbol{X}_v$ for each node v.

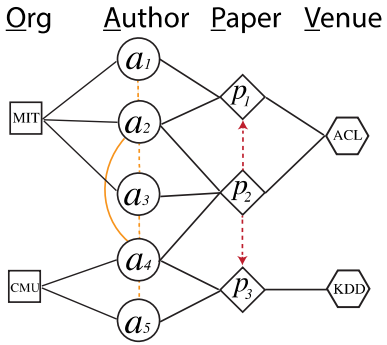


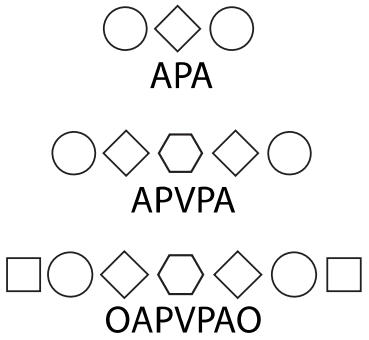
meta-path-based random walks

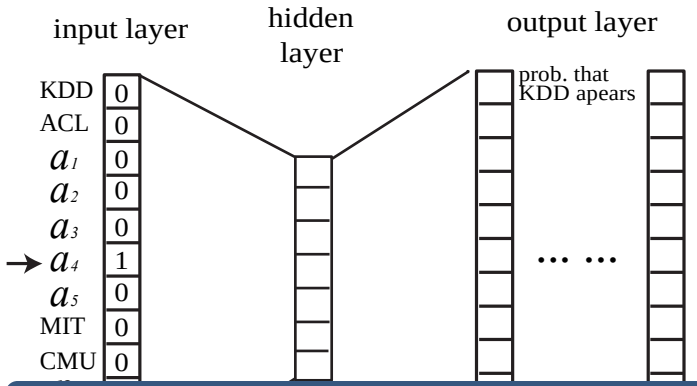
skip-gram

Heterogeneous graph embedding

- Input: a heterogeneous graph $G = (V, E)$
- Output: $\boldsymbol{X} \in R^{|V| \times k}, k \ll |V|$, k-dim vector $\boldsymbol{X}_v$ for each node v.





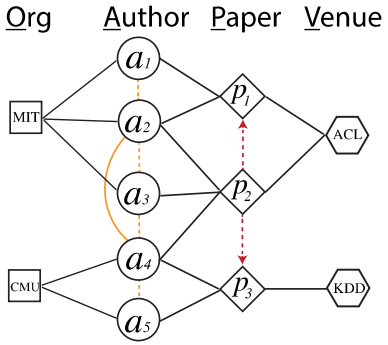


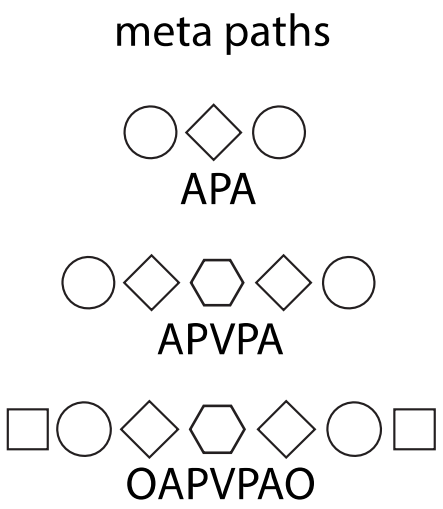
meta-path-based random walks

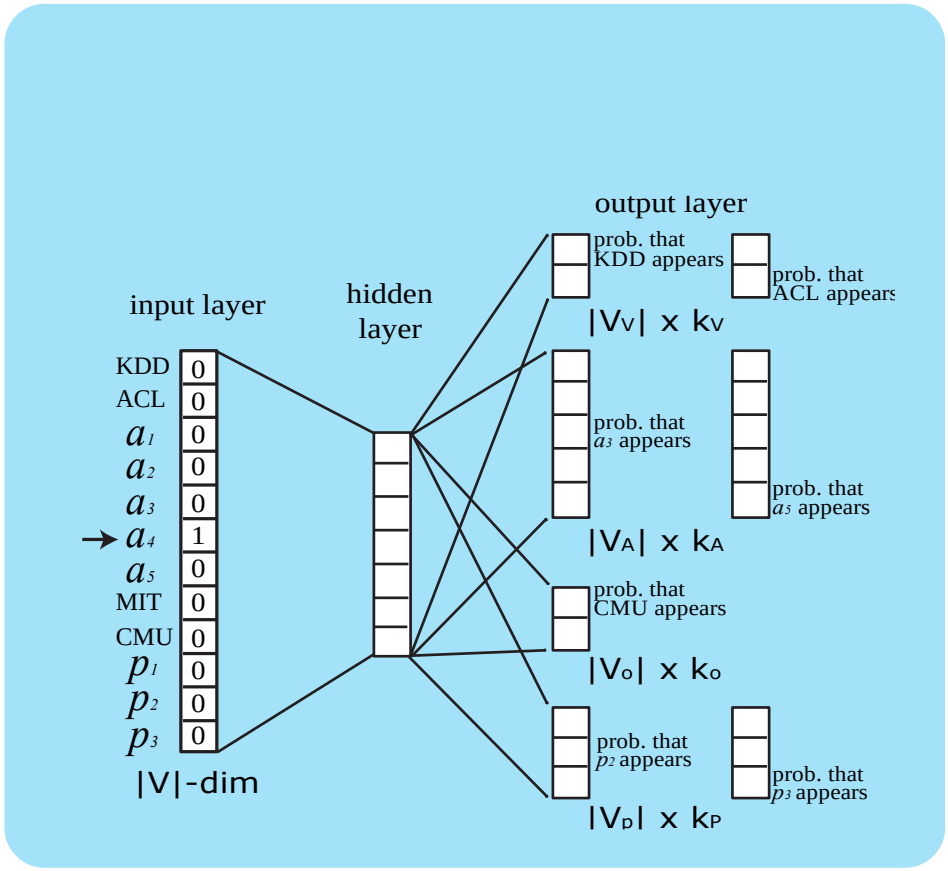
The potential issue of skip-gram for heterogeneous network embedding:

To predict the context node c_t (type *t*) given a node *v*, *metapath2vec* encourages all types of nodes to appear in this context position

Heterogeneous graph embedding



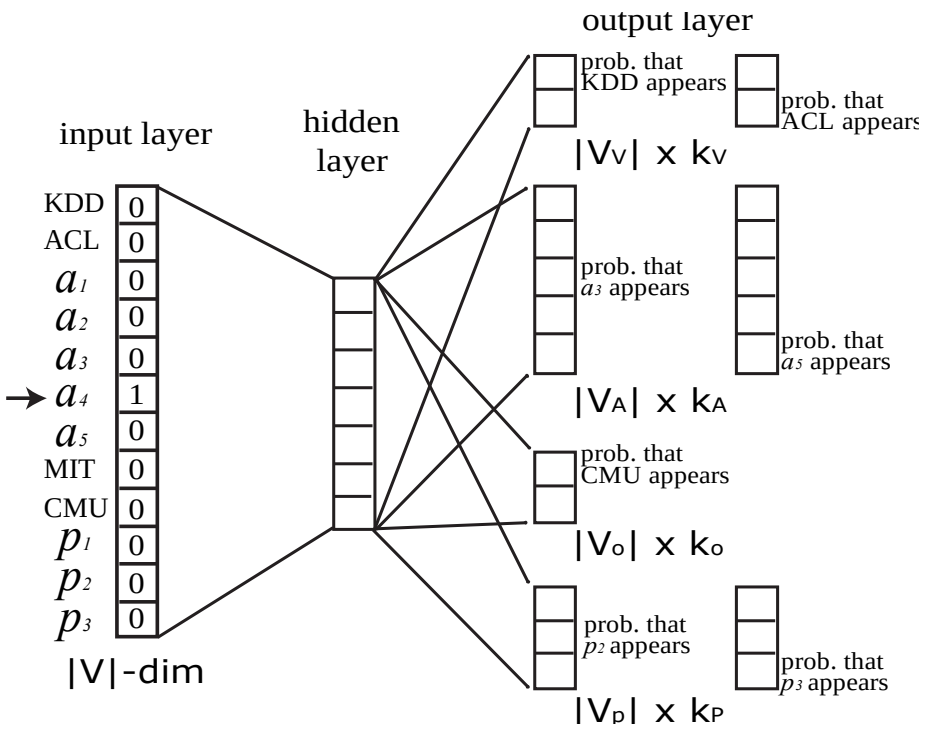




meta-path-based random walks

heterogeneous skip-gram

Heterogeneous Skip-Gram

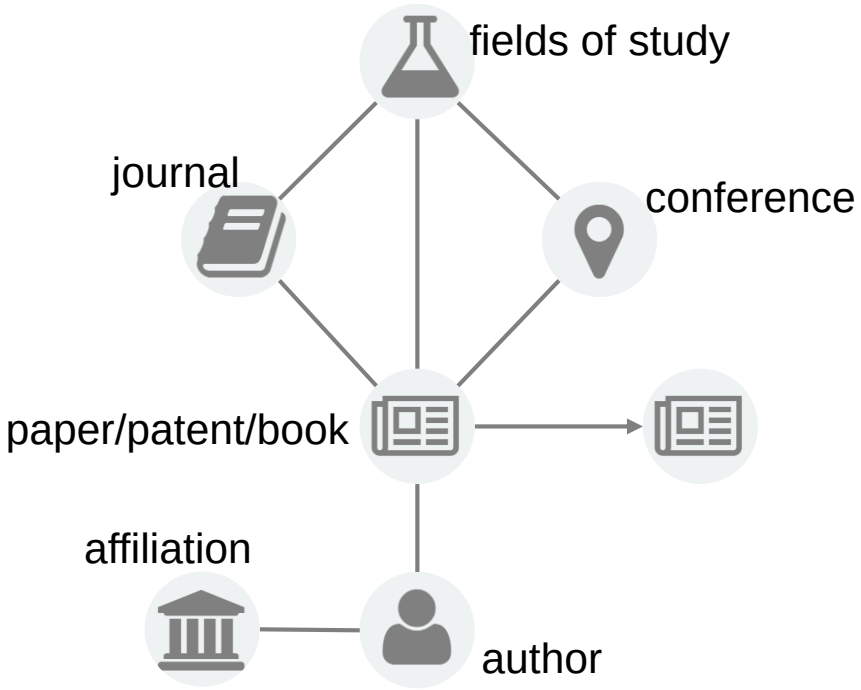


$$\mathcal{O}(\mathbf{X}) = \log \sigma(X_{c_t} \cdot X_v) + \sum_{k=1}^{K} \mathbb{E}_{u_t^k \sim P_t(u_t)} [\log \sigma(-X_{u_t^k} \cdot X_v)]$$

Heterogeneous graph embedding

1. Sun and Han. Mining heterogeneous information networks: Principles and Methodologies. Morgan & Claypool Publishers, 2012.
2. Dong et al. metapath2vec: scalable representation learning for heterogeneous networks. In *ACM KDD* 2017. **The most cited paper in KDD'17 as of Aug 2018.**

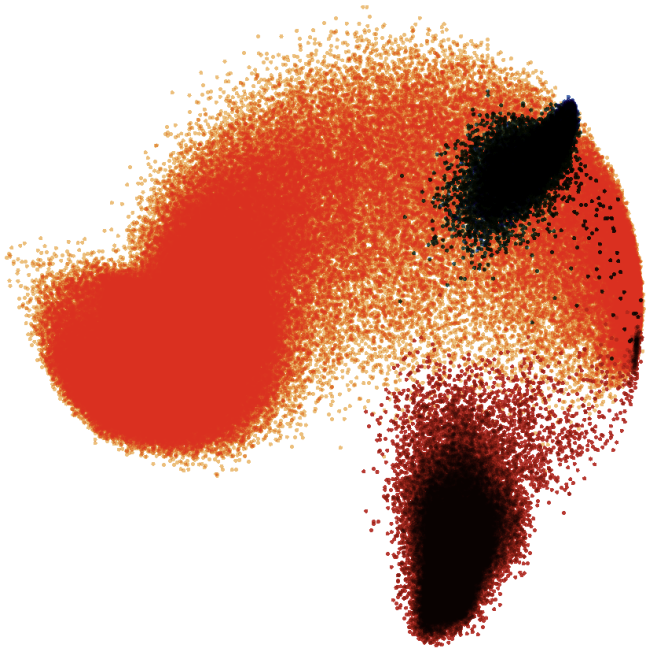
Application: Embedding Heterogeneous Academic Graph



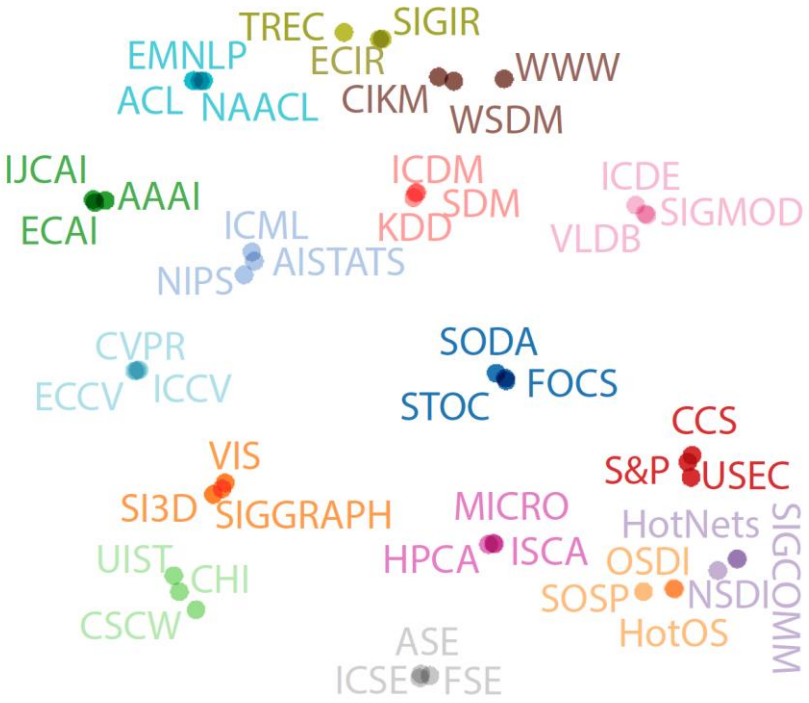
Microsoft Academic Graph
&
AMiner

metapath2vec

metapath2vec++



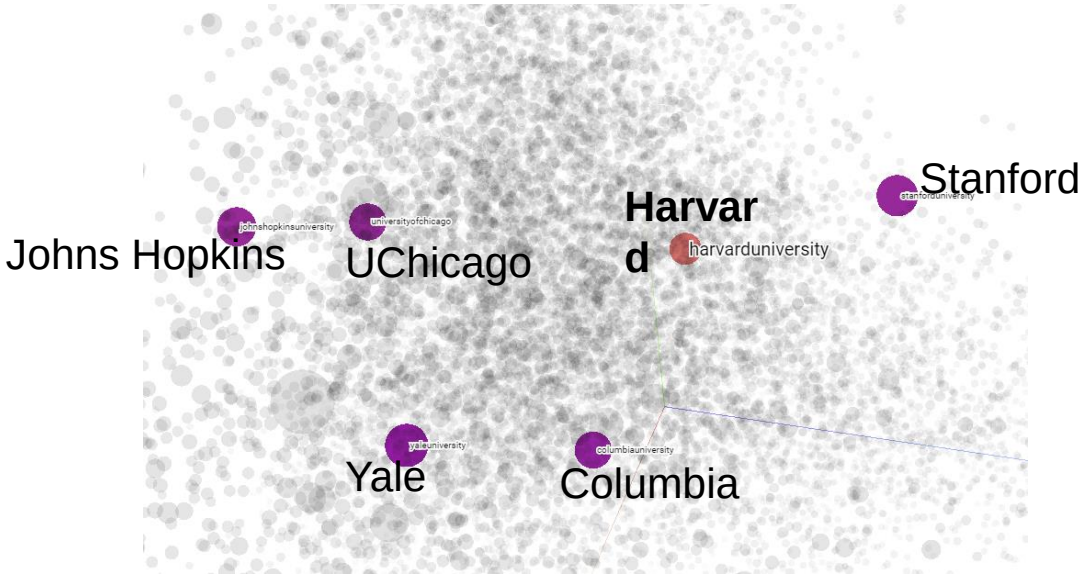
Application 2: Node Clustering

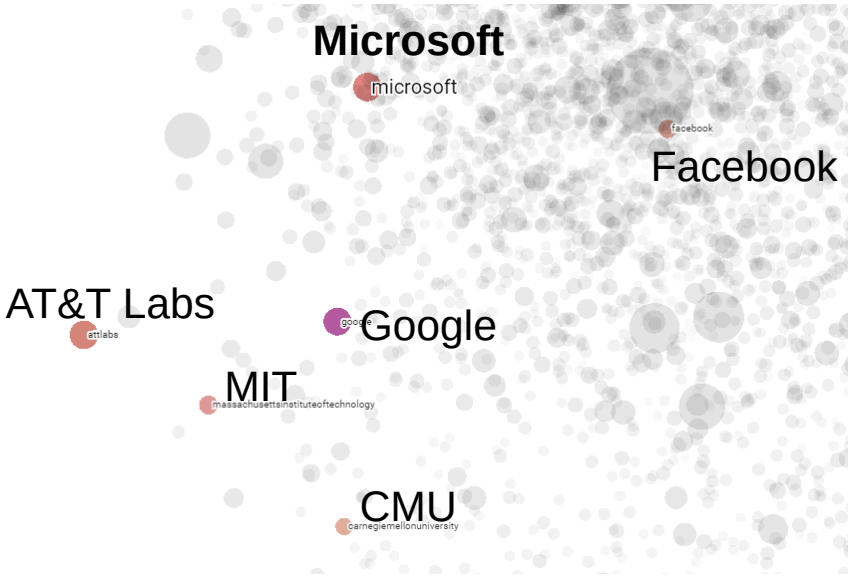


Application 3: Similarity Search (Conference)

Rank	ACL	NIPS	IJCAI	CVPR	FOCS	SOSP	ISCA	S&P	ICSE	SIGGRAPH	SIGCOMM	CHI	KDD	SIGMOD	SIGIR	WWW
0	ACL	NIPS	IJCAI	CVPR	FOCS	SOSP	ISCA	S&P	ICSE	SIGGRAPH	SIGCOMM	CHI	KDD	SIGMOD	SIGIR	WWW
1	EMNLP	ICML	AAAI	ECCV	STOC	TOCS	HPCA	CCS	TOSEM	TOG	CCR	CSCW	SDM	PVLDB	ECIR	WSDM
2	NAACL	AISTATS	AI	ICCV	SICOMP	OSDI	MICRO	NDSS	FSE	SI3D	HotNets	TOCHI	TKDD	ICDE	CIKM	CIKM
3	CL	JMLR	JAIR	IJCV	SODA	HotOS	ASPLOS	USENIX S	ASE	RT	NSDI	UIST	ICDM	DE Bull	IR J	TWEB
4	CoNLL	NC	ECAI	ACCV	A-R	SIGOPS E	PACT	ACSAC	ISSTA	CGF	CoNEXT	DIS	DMKD	VLDBJ	TREC	ICWSM
5	COLING	MLJ	KR	CVIU	TALG	ATC	ICS	JCS	E SE	NPAR	IMC	HCI	KDD E	EDBT	SIGIR F	HT
6	IJCNLP	COLT	AI Mag	BMVC	ICALP	NSDI	HiPEAC	ESORICS	MSR	Vis	TON	MobileHCI	WSDM	TODS	ICTIR	SIGIR
7	NLE	UAI	ICAPS	ICPR	ECCC	OSR	PPOPP	TISS	ESEM	JGT	INFOCOM	INTERACT	CIKM	CIDR	WSDM	KDD
8	ANLP	KDD	CI	EMMCVPR	TOC	ASPLOS	ICCD	ASIACCS	A SE	VisComp	PAM	GROUP	PKDD	SIGMOD R	TOIS	TIT
9	LREC	CVPR	AIPS	T on IP	JAlG	EuroSys	CGO	RAID	ICPC	GI	MobiCom	NordiCHI	ICML	WebDB	IPM	WISE
10	EACL	ECML	UAI	WACV	ITCS	SIGCOMM	ISLPED	CSFW	WICSA	CG	IPTPS	UbiComp	PAKDD	PODS	AIRS	WebSci

Application 3: Similarity Search (Institution)





Application 3: Related Venues

https://academic.microsoft.com/#/detail/137773608

Microsoft Academic | Nature

Nature

Description: Nature is a British multidisciplinary scie
the Science Edition of the 2010 Journal Citation Rep
of the few remaining academic journals that publish

Websites: www.nature.com, en.wikipedia.org,

Related Journals

- Science
- Proceedings of the National Academy of Sciences of the United States of America
- Nature Communications
- PLOS Biology
- Philosophical Transactions of the Royal Society B
- Current Biology
- BioEssays
- Nature Methods
- EMBO Reports
- PLOS ONE

Secure | https://academic.microsoft.com/#/detail/41523882

Microsoft Academic | ACM Transactions on Knowledge Discovery From Data

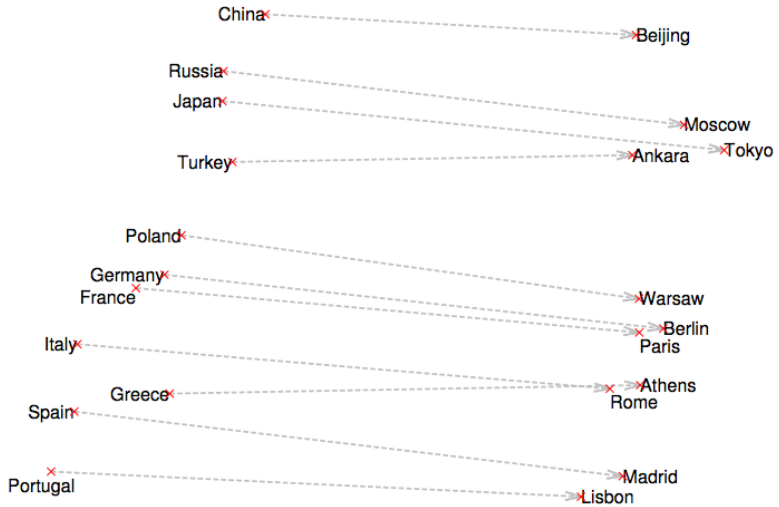
ACM Transactions on Knowledge Discovery From Data

Websites: bing.com

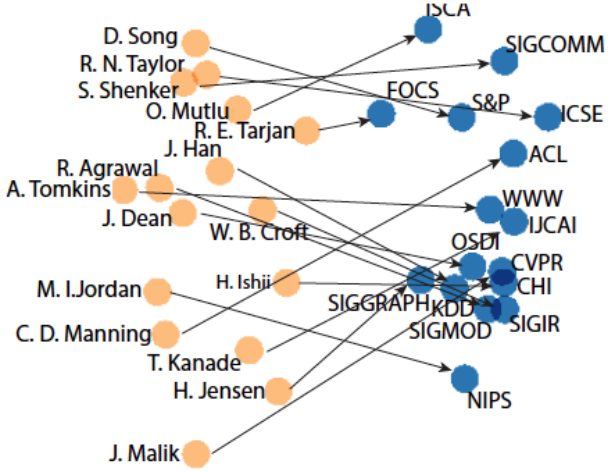
Related Journals

- Data Mining and Knowledge Discovery
- IEEE Transactions on Knowledge and Data Engineering
- Knowledge and Information Systems
- Sigkdd Explorations
- Proceedings of The Vldb Endowment
- World Wide Web
- ACM Transactions on Intelligent Systems and Technology
- Statistical Analysis and Data Mining
- arXiv: Learning
- The Vldb Journal

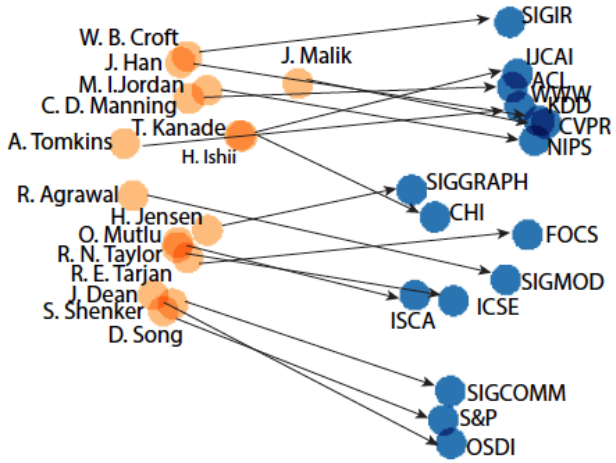
Visualization



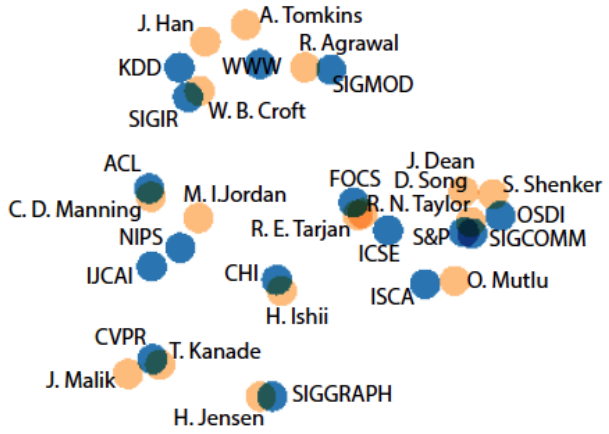
word2vec [Mikolov, 2013]



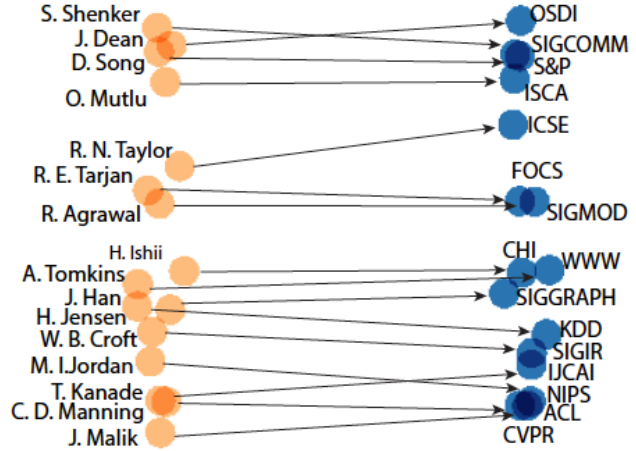
(a) DeepWalk/node2vec



(b) PTE



(c) *metapath2vec*



(d) *metapath2vec++*

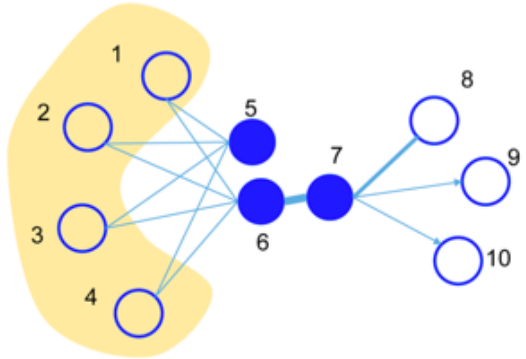
http://projector.tensorflow.org/

Random Walk Strategies

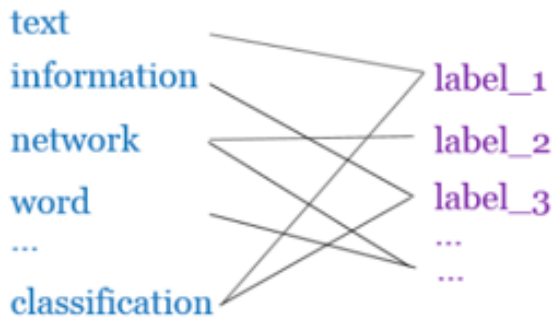
- Random Walk
  - DeepWalk
- Biased Random Walk
  - 2nd order Random Walk
    - node2vec
  - Metapath guided Random Walk
    - metapath2vec

LINE and PTE

- **LINE:** explicitly preserves both first-order and second-order proximities.

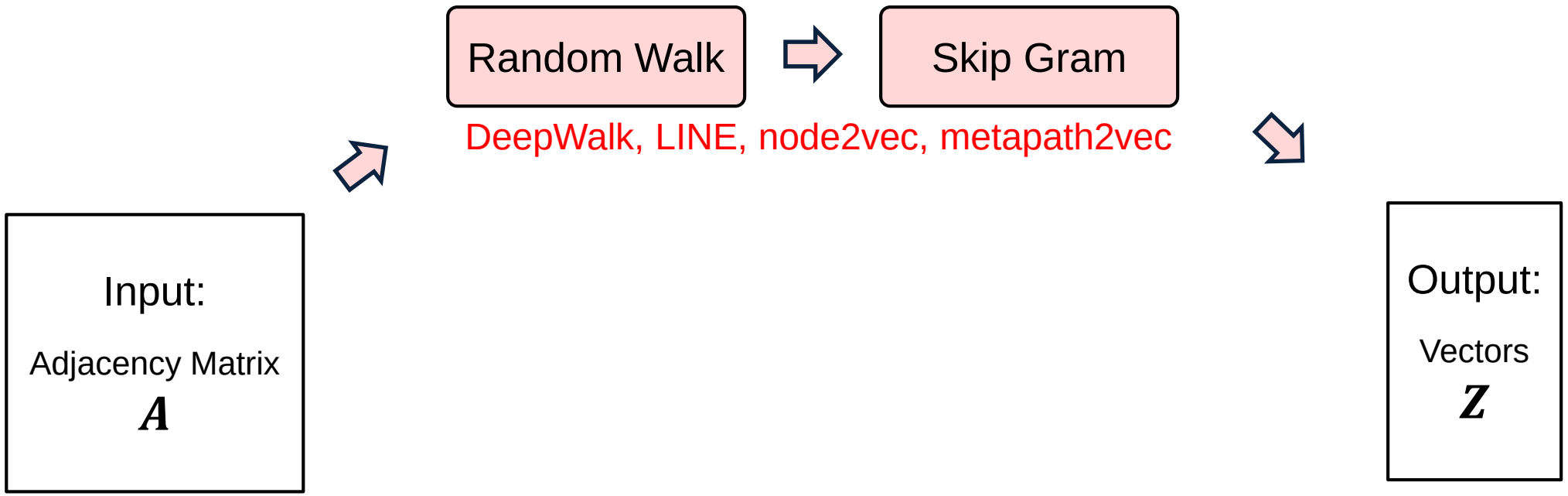


- **PTE:** learn heterogeneous text network embedding via a semi-supervised manner.



J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei. 2015. Line: Large-scale information network embedding. *WWW*. ***Most Cited Paper in WWW'15.***
J. Tang, M. Qu, and Q. Mei. 2015. Pte: Predictive text embedding through large-scale heterogeneous text networks. *KDD*, 1165–1174.

Network Embedding



What are the fundamentals

underlying skip-gram based network embedding models?

Network embedding as matrix factorization: unifying deepwalk, line, pte, and node2vec

Qiu et al., Network embedding as matrix factorization: unifying deepwalk, line, pte, and node2vec. In *WSDM'18.*
The most cited paper in WSDM'18 as of May 2019

Skip-gram based network embedding

- Input: an undirected weighted network $G = (V, E)$ with $|V| = n$ & $|E| = \mathrm{m}$
  - Adjacency matrix $\boldsymbol{A} \in \mathbb{R}_{+}^{n \times \mathrm{n}}$

$$A_{i,j} = \begin{cases} a_{i,j} > 0 & (i,j) \in E \\ 0 & (i,j) \notin E \end{cases}$$

  - Degree matrix $\boldsymbol{D} = diag(d_1, d_2, \cdots, d_n)$
  - Volume of G: $vol(G) = \sum_i \sum_j A_{ij}$
- Output: for each node, its k-dimension latent feature representation vector $\mathbf{Z}^{n \times k}$
  - Latent feature embedding matrix $\mathbf{Z} \in \mathbb{R}^{n \times k}$

Unifying DeepWalk, LINE, PTE, & node2vec as Matrix Factorization

- DeepWalk $\log\left(\frac{\mathrm{vol}(G)}{b}\left(\frac{1}{T}\sum_{r=1}^{T}\left(\boldsymbol{D}^{-1}\boldsymbol{A}\right)^{r}\right)\boldsymbol{D}^{-1}\right)$
- LINE $\log\left(\frac{\mathrm{vol}(G)}{b}\boldsymbol{D}^{-1}\boldsymbol{A}\boldsymbol{D}^{-1}\right)$
- PTE $\log\left(\begin{bmatrix}\alpha\,\mathrm{vol}(G_{\mathsf{ww}})(\boldsymbol{D}_{\mathsf{row}}^{\mathsf{ww}})^{-1}\boldsymbol{A}_{\mathsf{ww}}(\boldsymbol{D}_{\mathsf{col}}^{\mathsf{ww}})^{-1}\\ \beta\,\mathrm{vol}(G_{\mathsf{dw}})(\boldsymbol{D}_{\mathsf{row}}^{\mathsf{dw}})^{-1}\boldsymbol{A}_{\mathsf{dw}}(\boldsymbol{D}_{\mathsf{col}}^{\mathsf{dw}})^{-1}\\ \gamma\,\mathrm{vol}(G_{\mathsf{lw}})(\boldsymbol{D}_{\mathsf{row}}^{\mathsf{lw}})^{-1}\boldsymbol{A}_{\mathsf{lw}}(\boldsymbol{D}_{\mathsf{col}}^{\mathsf{lw}})^{-1}\end{bmatrix}\right)-\log b$
- node2vec $\log\left(\frac{\frac{1}{2T}\sum_{r=1}^{T}\left(\sum_u \boldsymbol{X}_{w,u}\underline{\boldsymbol{P}}_{c,w,u}^{r}+\sum_u \boldsymbol{X}_{c,u}\underline{\boldsymbol{P}}_{w,c,u}^{r}\right)}{b\left(\sum_u \boldsymbol{X}_{w,u}\right)\left(\sum_u \boldsymbol{X}_{c,u}\right)}\right)$

$\boldsymbol{A}$ Adjacency matrix

$\boldsymbol{D}$ Degree matrix

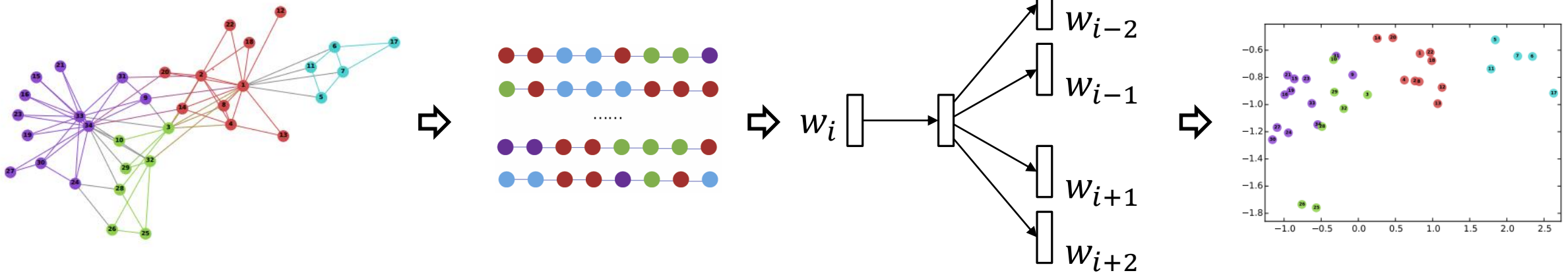
b: #negative samples

T: context window size

$$vol(G)=\sum_i\sum_j A_{ij}$$

1. Qiu et al. Network embedding as matrix factorization: unifying deepwalk, line, pte, and node2vec. In *WSDM'18.* **The most cited paper in WSDM'18 as of May 2019**

Understanding random walk + skip gram



Skip gram with negative sampling

Skip-gram with negative sampling (SGNS)

- SGNS maintains a multiset $\mathcal{D}$ that counts the occurrence of each word-context pair (w, c)
- Objective

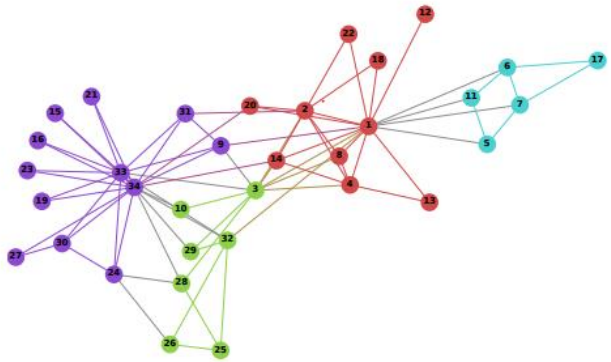
$$\mathcal{L} = \sum_{w} \sum_{c} (\#(w,c) \log g(x_w^T x_c) + \frac{b\#(w)\#(c)}{|\mathcal{D}|} \log g(-x_w^T x_c))$$

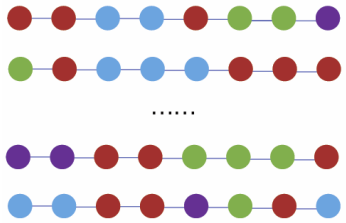
- For sufficiently large dimension *d*, the objective above is equivalent to factorizing the PMI matrix

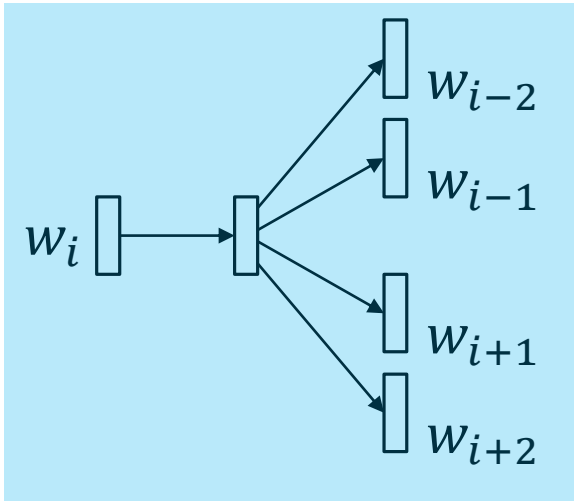
$$\log \frac{\#(w,c)|\mathcal{D}|}{b\#(w)\#(c)}$$

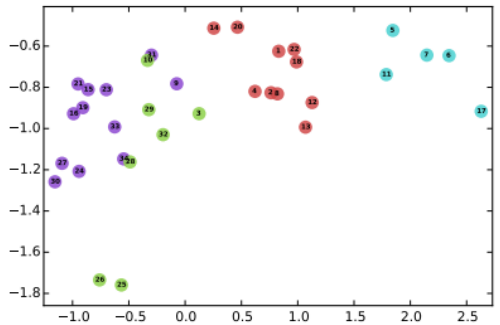
Levy and Goldberg. Neural word embeddings as implicit matrix factorization. In *NIPS 2014*

Understanding random walk + skip gram









$G = (V, E)$

- Adjacency matrix $\boldsymbol{A}$
- Degree matrix $\boldsymbol{D}$
- Volume of G: $vol(G)$

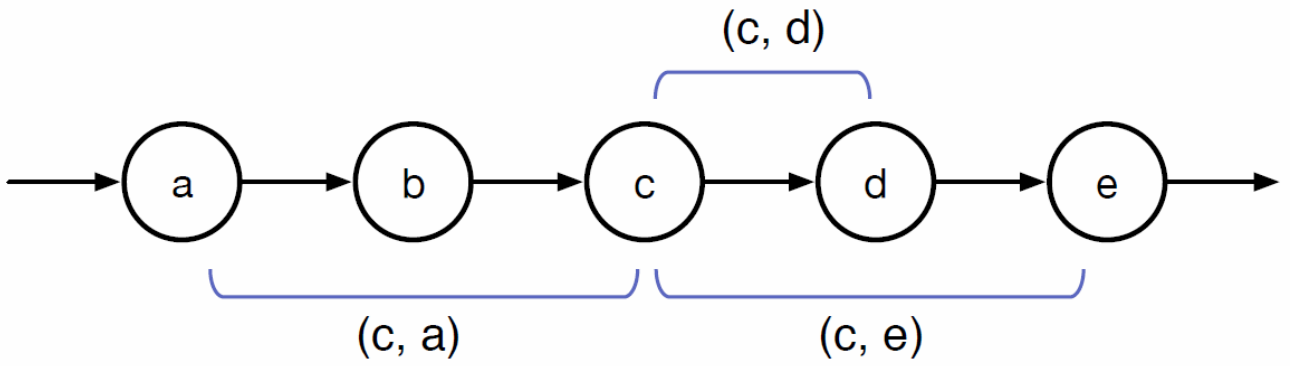
?

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

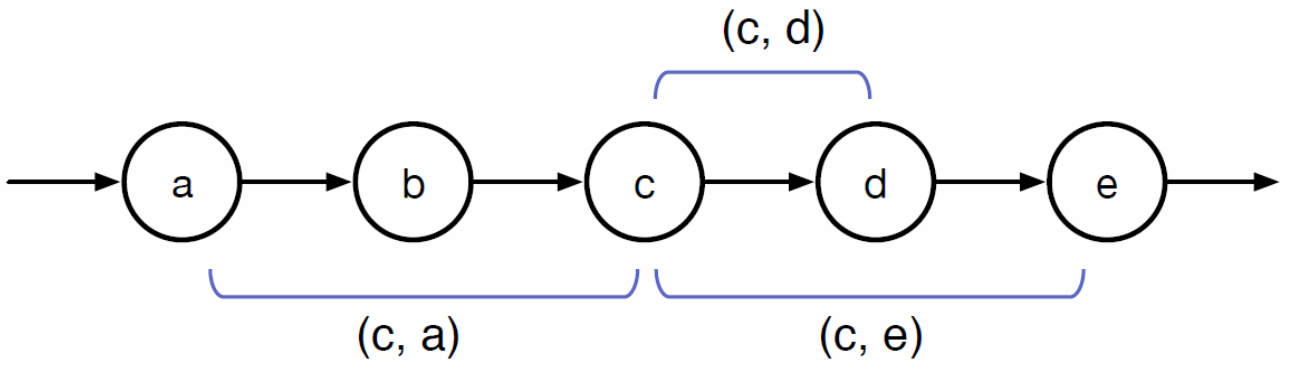
Understanding random walk + skip gram

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

Understanding random walk + skip gram

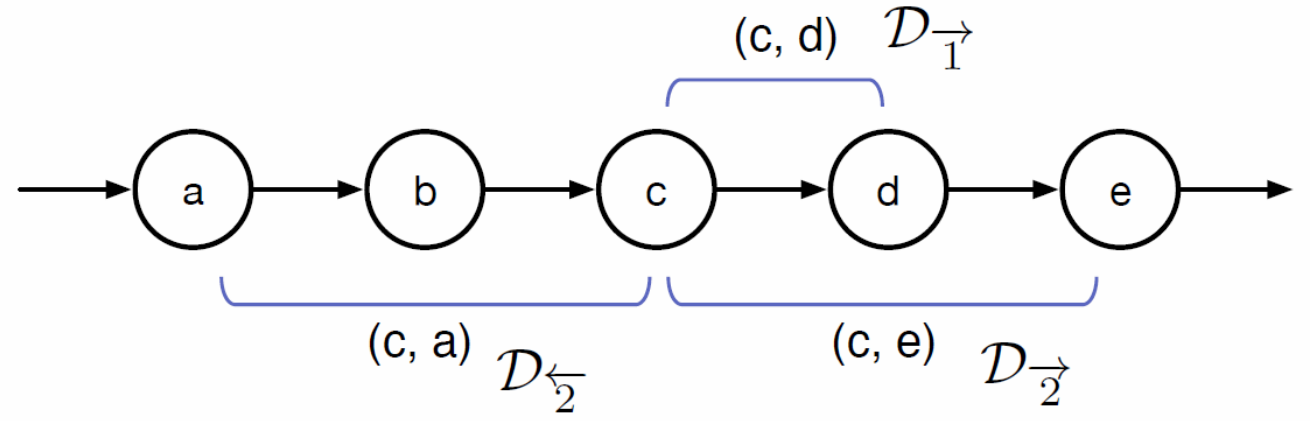


Understanding random walk + skip gram



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

Understanding random walk + skip gram



- Partition the multiset $\mathcal{D}$ into several sub-multisets according to the way in which each node and its context appear in a random walk node sequence.
- More formally, for $r = 1, 2, \cdots, T$, we define

$$\mathcal{D}_{\overrightarrow{r}} = \left\{(w, c) : (w, c) \in \mathcal{D}, w = w_j^n, c = w_{j+r}^n\right\}$$
$$\mathcal{D}_{\overleftarrow{r}} = \left\{(w, c) : (w, c) \in \mathcal{D}, w = w_{j+r}^n, c = w_j^n\right\}$$

Distinguish direction and distance

Understanding random walk + skip gram

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

Understanding random walk + skip gram

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

$$\frac{\#(w,c)}{|\mathcal{D}|}=\frac{1}{2T}\sum_{r=1}^{T}\left(\frac{\#(w,c)_{\overrightarrow{r}}}{|\mathcal{D}_{\overrightarrow{r}}|}+\frac{\#(w,c)_{\overleftarrow{r}}}{|\mathcal{D}_{\overleftarrow{r}}|}\right)$$

Understanding random walk + skip gram

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

the length of random walk $L \to \infty$

$$\frac{\#(w,c)}{|\mathcal{D}|}=\frac{1}{2T}\sum_{r=1}^{T}\left(\frac{\#(w,c)_{\overrightarrow{r}}}{|\mathcal{D}_{\overrightarrow{r}}|}+\frac{\#(w,c)_{\overleftarrow{r}}}{|\mathcal{D}_{\overleftarrow{r}}|}\right)$$

$$\frac{\#(w,c)_{\overrightarrow{r}}}{|\mathcal{D}_{\overrightarrow{r}}|}\xrightarrow{p}\frac{d_w}{\mathrm{vol}(G)}\left(\boldsymbol{P}^r\right)_{w,c}$$

$$\frac{\#(w,c)_{\overleftarrow{r}}}{|\mathcal{D}_{\overleftarrow{r}}|}\xrightarrow{p}\frac{d_c}{\mathrm{vol}(G)}\left(\boldsymbol{P}^r\right)_{c,w}$$

$$\boldsymbol{P}=\boldsymbol{D}^{-1}\boldsymbol{A}$$

Understanding random walk + skip gram

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

the length of random walk $L\to\infty$

$$\frac{\#(w,c)}{|\mathcal{D}|}=\frac{1}{2T}\sum_{r=1}^{T}\left(\frac{\#(w,c)_{\overrightarrow{r}}}{|\mathcal{D}_{\overrightarrow{r}}|}+\frac{\#(w,c)_{\overleftarrow{r}}}{|\mathcal{D}_{\overleftarrow{r}}|}\right)$$

$$\frac{\#(w,c)_{\overrightarrow{r}}}{|\mathcal{D}_{\overrightarrow{r}}|}\xrightarrow{p}\frac{d_w}{\mathrm{vol}(G)}\left(\boldsymbol{P}^r\right)_{w,c}$$

$$\frac{\#(w,c)_{\overleftarrow{r}}}{|\mathcal{D}_{\overleftarrow{r}}|}\xrightarrow{p}\frac{d_c}{\mathrm{vol}(G)}\left(\boldsymbol{P}^r\right)_{c,w}$$

$$\frac{\#(w,c)}{|\mathcal{D}|}\xrightarrow{p}\frac{1}{2T}\sum_{r=1}^{T}\left(\frac{d_w}{\mathrm{vol}(G)}\left(\boldsymbol{P}^r\right)_{w,c}+\frac{d_c}{\mathrm{vol}(G)}\left(\boldsymbol{P}^r\right)_{c,w}\right)$$

$$\boldsymbol{P}=\boldsymbol{D}^{-1}\boldsymbol{A}$$

Understanding random walk + skip gram

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

the length of random walk $L\to\infty$

$$\frac{\#(w,c)}{|\mathcal{D}|}=\frac{1}{2T}\sum_{r=1}^{T}\left(\frac{\#(w,c)_{\overrightarrow{r}}}{|\mathcal{D}_{\overrightarrow{r}}|}+\frac{\#(w,c)_{\overleftarrow{r}}}{|\mathcal{D}_{\overleftarrow{r}}|}\right)$$

$$\frac{\#(w,c)_{\overrightarrow{r}}}{|\mathcal{D}_{\overrightarrow{r}}|}\xrightarrow{p}\frac{d_w}{\mathrm{vol}(G)}\left(\boldsymbol{P}^r\right)_{w,c}$$

$$\frac{\#(w,c)_{\overleftarrow{r}}}{|\mathcal{D}_{\overleftarrow{r}}|}\xrightarrow{p}\frac{d_c}{\mathrm{vol}(G)}\left(\boldsymbol{P}^r\right)_{c,w}$$

$$\frac{\#(w,c)}{|\mathcal{D}|}\xrightarrow{p}\frac{1}{2T}\sum_{r=1}^{T}\left(\frac{d_w}{\mathrm{vol}(G)}\left(\boldsymbol{P}^r\right)_{w,c}+\frac{d_c}{\mathrm{vol}(G)}\left(\boldsymbol{P}^r\right)_{c,w}\right)$$

$$\boldsymbol{P}=\boldsymbol{D}^{-1}\boldsymbol{A}$$

$$\frac{\#(w)}{|\mathcal{D}|}\xrightarrow{p}\frac{d_w}{\mathrm{vol}(G)}$$

$$\frac{\#(c)}{|\mathcal{D}|}\xrightarrow{p}\frac{d_c}{\mathrm{vol}(G)}$$

Understanding random walk + skip gram

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

the length of random walk $L \to \infty$

$$\frac{\#(w,c)}{|\mathcal{D}|} \xrightarrow{p} \frac{1}{2T}\sum_{r=1}^{T}\left(\frac{d_w}{\mathrm{vol}(G)}\left(\boldsymbol{P}^r\right)_{w,c} + \frac{d_c}{\mathrm{vol}(G)}\left(\boldsymbol{P}^r\right)_{c,w}\right)$$

$$\boldsymbol{P} = \boldsymbol{D}^{-1}\boldsymbol{A}$$

$$\frac{\#(w)}{|\mathcal{D}|} \xrightarrow{p} \frac{d_w}{\mathrm{vol}(G)} \qquad \frac{\#(c)}{|\mathcal{D}|} \xrightarrow{p} \frac{d_c}{\mathrm{vol}(G)}$$

$$\frac{\#(w,c)\,|\mathcal{D}|}{\#(w)\cdot\#(c)} = \frac{\frac{\#(w,c)}{|\mathcal{D}|}}{\frac{\#(w)}{|\mathcal{D}|}\cdot\frac{\#(c)}{|\mathcal{D}|}} \xrightarrow{p} \frac{\frac{1}{2T}\sum_{r=1}^{T}\left(\frac{d_w}{\mathrm{vol}(G)}\left(\boldsymbol{P}^r\right)_{w,c} + \frac{d_c}{\mathrm{vol}(G)}\left(\boldsymbol{P}^r\right)_{c,w}\right)}{\frac{d_w}{\mathrm{vol}(G)}\cdot\frac{d_c}{\mathrm{vol}(G)}}$$

$$= \frac{\mathrm{vol}(G)}{2T}\left(\frac{1}{d_c}\sum_{r=1}^{T}\left(\boldsymbol{P}^r\right)_{w,c} + \frac{1}{d_w}\sum_{r=1}^{T}\left(\boldsymbol{P}^r\right)_{c,w}\right)$$

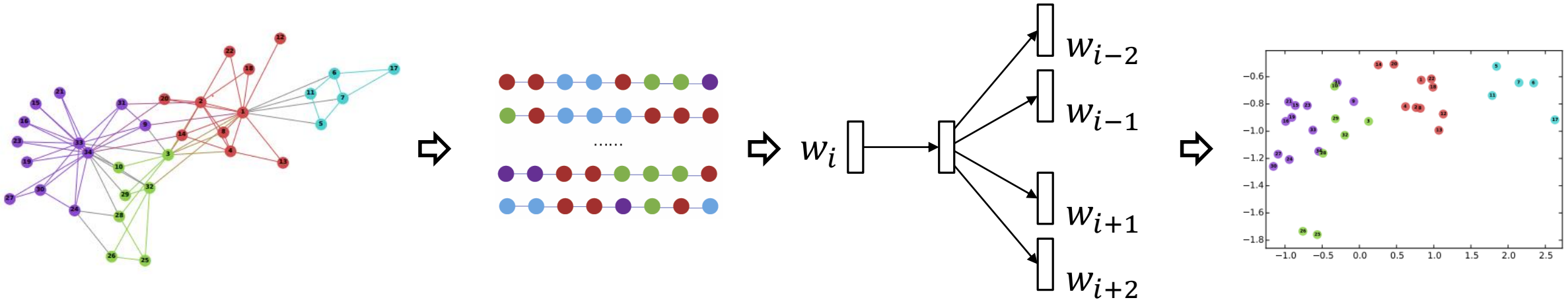
$$\frac{\#(w,c)\,|\mathcal{D}|}{\#(w)\cdot\#(c)} \xrightarrow{p} \frac{\mathrm{vol}(G)}{2T}\left(\frac{1}{d_c}\sum_{r=1}^{T}\left(\boldsymbol{P}^r\right)_{w,c}+\frac{1}{d_w}\sum_{r=1}^{T}\left(\boldsymbol{P}^r\right)_{c,w}\right)$$

$$\frac{\mathrm{vol}(G)}{2T}\left(\sum_{r=1}^{T}\boldsymbol{P}^r\boldsymbol{D}^{-1}+\sum_{r=1}^{T}\boldsymbol{D}^{-1}\left(\boldsymbol{P}^r\right)^{\top}\right)$$

$$=\frac{\mathrm{vol}(G)}{2T}\left(\sum_{r=1}^{T}\underbrace{\boldsymbol{D}^{-1}\boldsymbol{A}\times\cdots\times\boldsymbol{D}^{-1}\boldsymbol{A}}_{r\text{ terms}}\boldsymbol{D}^{-1}+\sum_{r=1}^{T}\boldsymbol{D}^{-1}\underbrace{\boldsymbol{A}\boldsymbol{D}^{-1}\times\cdots\times\boldsymbol{A}\boldsymbol{D}^{-1}}_{r\text{ terms}}\right)$$

$$=\frac{\mathrm{vol}(G)}{T}\sum_{r=1}^{T}\underbrace{\boldsymbol{D}^{-1}\boldsymbol{A}\times\cdots\times\boldsymbol{D}^{-1}\boldsymbol{A}}_{r\text{ terms}}\boldsymbol{D}^{-1}=\mathrm{vol}(G)\left(\frac{1}{T}\sum_{r=1}^{T}\boldsymbol{P}^r\right)\boldsymbol{D}^{-1}.$$

Understanding random walk + skip gram



DeepWalk is asymptotically and implicitly factorizing

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

$\boldsymbol{A}$ Adjacency matrix

$\boldsymbol{D}$ Degree matrix

$vol(G) = \sum_i \sum_j A_{ij}$

b: #negative samples

T: context window size

1. Qiu et al. Network embedding as matrix factorization: unifying deepwalk, line, pte, and node2vec. In *WSDM'18.* **The most cited paper in WSDM'18 as of May 2019**

Understanding LINE

- Objective of LINE:

$$\mathcal{L} = \sum_{i=1}^{|V|} \sum_{j=1}^{|V|} \left(\boldsymbol{A}_{i,j} \log g\left(\boldsymbol{x}_i^\top \boldsymbol{y}_j\right) + \frac{b d_i d_j}{\mathrm{vol}(G)} \log g\left(-\boldsymbol{x}_i^\top \boldsymbol{y}_j\right) \right).$$

- Align it with the Objective of SGNS:

$$\mathcal{L} = \sum_{w} \sum_{c} \left(\#(w,c) \log g\left(\boldsymbol{x}_w^\top \boldsymbol{y}_c\right) + \frac{b \#(w) \#(c)}{|\mathcal{D}|} \log g\left(-\boldsymbol{x}_w^\top \boldsymbol{y}_c\right) \right)$$

- LINE is actually factorizing

$$\log^\circ \left(\frac{\mathrm{vol}(G)}{b} \boldsymbol{D}^{-1} \boldsymbol{A} \boldsymbol{D}^{-1} \right)$$

J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei. 2015. Line: Large-scale information network embedding. *WWW.* ***Most Cited Paper in WWW'15.***

Understanding PTE

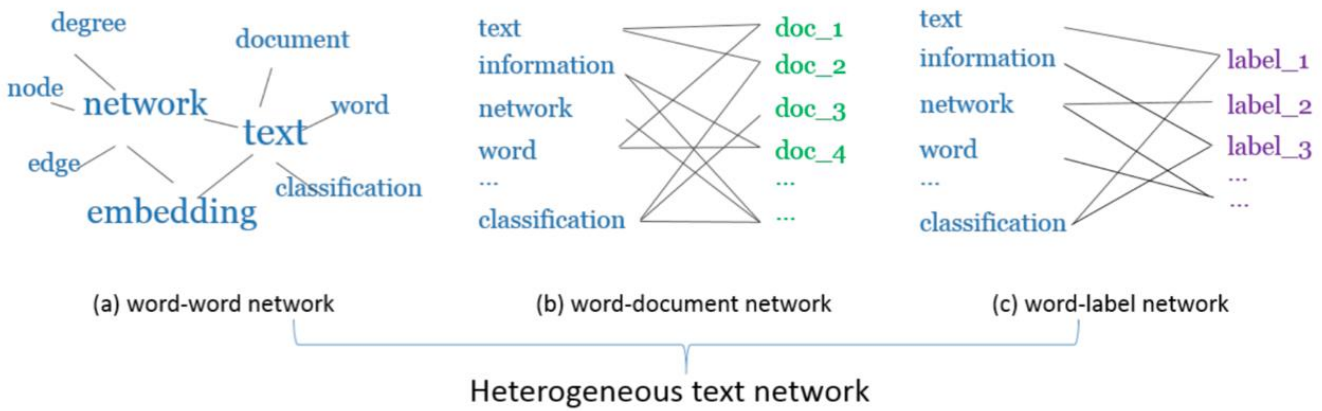


Figure 2: Heterogeneous Text Network.

- word-word network G_{ww}, $\boldsymbol{A}_{\text{ww}} \in \mathbb{R}^{\#\text{word} \times \#\text{word}}$.
- document-word network G_{dw}, $\boldsymbol{A}_{\text{dw}} \in \mathbb{R}^{\#\text{doc} \times \#\text{word}}$.
- label-word network G_{lw}, $\boldsymbol{A}_{\text{lw}} \in \mathbb{R}^{\#\text{label} \times \#\text{word}}$.

J. Tang, M. Qu, and Q. Mei. 2015. Pte: Predictive text embedding through large-scale heterogeneous text networks. *KDD*, 1165–1174.

Understanding PTE

$$\log\left(\begin{bmatrix}\alpha\,\mathrm{vol}(G_{\mathsf{ww}})(\boldsymbol{D}_{\mathsf{row}}^{\mathsf{ww}})^{-1}\boldsymbol{A}_{\mathsf{ww}}(\boldsymbol{D}_{\mathsf{col}}^{\mathsf{ww}})^{-1}\\ \beta\,\mathrm{vol}(G_{\mathsf{dw}})(\boldsymbol{D}_{\mathsf{row}}^{\mathsf{dw}})^{-1}\boldsymbol{A}_{\mathsf{dw}}(\boldsymbol{D}_{\mathsf{col}}^{\mathsf{dw}})^{-1}\\ \gamma\,\mathrm{vol}(G_{\mathsf{lw}})(\boldsymbol{D}_{\mathsf{row}}^{\mathsf{lw}})^{-1}\boldsymbol{A}_{\mathsf{lw}}(\boldsymbol{D}_{\mathsf{col}}^{\mathsf{lw}})^{-1}\end{bmatrix}\right)-\log b,$$

- The matrix is of shape $(\#\mathsf{word}+\#\mathsf{doc}+\#\mathsf{label})\times\#\mathsf{word}$.
- 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

$$\alpha\,\mathrm{vol}(G_{\mathsf{ww}})=\beta\,\mathrm{vol}(G_{\mathsf{dw}})=\gamma\,\mathrm{vol}(G_{\mathsf{lw}})$$

J. Tang, M. Qu, and Q. Mei. 2015. Pte: Predictive text embedding through large-scale heterogeneous text networks. *KDD*, 1165–1174.

Understanding node2vec

$$\underline{\boldsymbol{T}}_{u,v,w} = \begin{cases} \frac{1}{p} & (u,v) \in E, (v,w) \in E, u = w; \\ 1 & (u,v) \in E, (v,w) \in E, u \neq w, (w,u) \in E; \\ \frac{1}{q} & (u,v) \in E, (v,w) \in E, u \neq w, (w,u) \notin E; \\ 0 & \text{otherwise.} \end{cases}$$

$$\underline{\boldsymbol{P}}_{u,v,w} = \text{Prob}\left(w_{j+1} = u | w_j = v, w_{j-1} = w\right) = \frac{\underline{\boldsymbol{T}}_{u,v,w}}{\sum_u \underline{\boldsymbol{T}}_{u,v,w}}.$$

Stationary Distribution

$$\sum_w \underline{\boldsymbol{P}}_{u,v,w} \boldsymbol{X}_{v,w} = \boldsymbol{X}_{u,v}$$

Existence guaranteed by Perron-Frobenius theorem, but may not be unique.

A. Grover and J. Leskovec. 2016. node2vec: Scalable feature learning for networks. *KDD*, 855–864.

Understanding node2vec

Theorem

node2vec is asymptotically and implicitly factorizing a matrix whose entry at w-th row, c-th column is

$$\log\left(\frac{\frac{1}{2T}\sum_{r=1}^{T}\left(\sum_u \boldsymbol{X}_{w,u}\underline{\boldsymbol{P}}^r_{c,w,u}+\sum_u \boldsymbol{X}_{c,u}\underline{\boldsymbol{P}}^r_{w,c,u}\right)}{b\left(\sum_u \boldsymbol{X}_{w,u}\right)\left(\sum_u \boldsymbol{X}_{c,u}\right)}\right)$$

A. Grover and J. Leskovec. 2016. node2vec: Scalable feature learning for networks. *KDD*, 855–864.

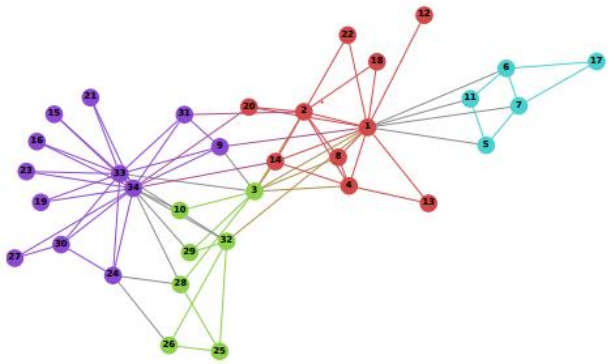
Unifying DeepWalk, LINE, PTE, & node2vec as Matrix Factorization

- DeepWalk $\log\left(\frac{\mathrm{vol}(G)}{b}\left(\frac{1}{T}\sum_{r=1}^{T}\left(\boldsymbol{D}^{-1}\boldsymbol{A}\right)^{r}\right)\boldsymbol{D}^{-1}\right)$
- LINE $\log\left(\frac{\mathrm{vol}(G)}{b}\boldsymbol{D}^{-1}\boldsymbol{A}\boldsymbol{D}^{-1}\right)$
- PTE $\log\left(\begin{bmatrix}\alpha\,\mathrm{vol}(G_{\mathsf{ww}})(\boldsymbol{D}^{\mathsf{ww}}_{\mathsf{row}})^{-1}\boldsymbol{A}_{\mathsf{ww}}(\boldsymbol{D}^{\mathsf{ww}}_{\mathsf{col}})^{-1}\\ \beta\,\mathrm{vol}(G_{\mathsf{dw}})(\boldsymbol{D}^{\mathsf{dw}}_{\mathsf{row}})^{-1}\boldsymbol{A}_{\mathsf{dw}}(\boldsymbol{D}^{\mathsf{dw}}_{\mathsf{col}})^{-1}\\ \gamma\,\mathrm{vol}(G_{\mathsf{lw}})(\boldsymbol{D}^{\mathsf{lw}}_{\mathsf{row}})^{-1}\boldsymbol{A}_{\mathsf{lw}}(\boldsymbol{D}^{\mathsf{lw}}_{\mathsf{col}})^{-1}\end{bmatrix}\right)-\log b$
- node2vec $\log\left(\frac{\frac{1}{2T}\sum_{r=1}^{T}\left(\sum_u \boldsymbol{X}_{w,u}\underline{\boldsymbol{P}}^{r}_{c,w,u}+\sum_u \boldsymbol{X}_{c,u}\underline{\boldsymbol{P}}^{r}_{w,c,u}\right)}{b\left(\sum_u \boldsymbol{X}_{w,u}\right)\left(\sum_u \boldsymbol{X}_{c,u}\right)}\right)$

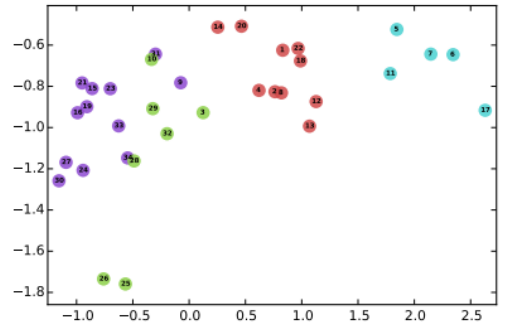
1. Qiu et al. Network embedding as matrix factorization: unifying deepwalk, line, pte, and node2vec. In *WSDM'18.* **The most cited paper in WSDM'18 as of May 2019**

Can we directly factorize the derived matrices for learning embeddings?

NetMF: explicitly factorizing the DW matrix



Matrix Factorization



DeepWalk is asymptotically and implicitly factorizing

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

1. Qiu et al. Network embedding as matrix factorization: unifying deepwalk, line, pte, and node2vec. In *WSDM'18.* **The most cited paper in WSDM'18 as of May 2019**

NetMF

- DeepWalk is implicitly factorizing

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

- NetMF is explicitly factorizing

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

NetMF

- DeepWalk is asymptotically and implicitly factorizing

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

Recall that in random walk + skip gram based network embedding models:
$\boldsymbol{z}_v^{\top}\boldsymbol{z}_c$ → the probability that node v and context c appear on a random walk path

- NetMF is explicitly factorizing

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

$\boldsymbol{z}_v^{\top}\boldsymbol{z}_c$ → the similarity score $\boldsymbol{M}_{vc}$ between node v and context c defined by this matrix

The NetMF Algorithm

- Factorize the DeepWalk matrix:

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

- For numerical reason, we use truncated logarithm — $\widetilde{\log}(x) = \log\left(\max(1, x)\right)$

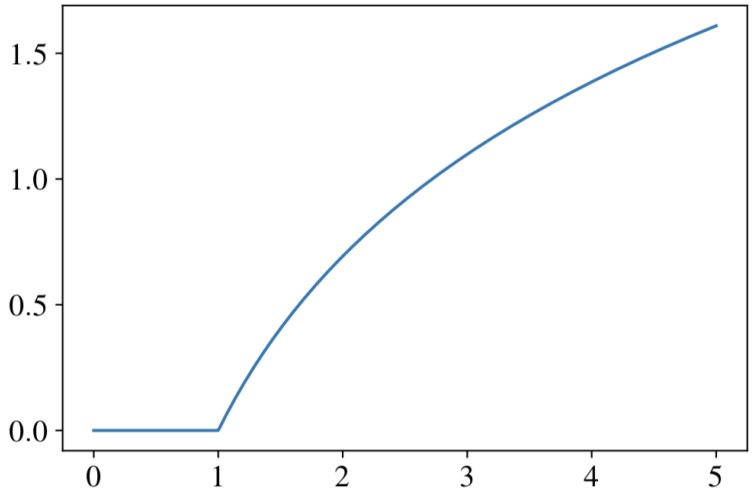


Figure 3: Truncated Logarithm

NetMF for a small window size T

Algorithm 2: NetMF for a Small Window Size T

Compute $\boldsymbol{P}^1, \cdots, \boldsymbol{P}^T$;

Compute $\boldsymbol{M} = \frac{\mathrm{vol}(G)}{bT} \left(\sum_{r=1}^{T} \boldsymbol{P}^r \right) \boldsymbol{D}^{-1}$;

Compute $\boldsymbol{M}' = \max(\boldsymbol{M}, 1)$;

Rank-d approximation by SVD: $\log \boldsymbol{M}' = \boldsymbol{U}_d \boldsymbol{\Sigma}_d \boldsymbol{V}_d^\top$;

return $\boldsymbol{U}_d \sqrt{\boldsymbol{\Sigma}_d}$ as network embedding.

NetMF for a LARGE window size T

Algorithm 2: NetMF for a Small Window Size T

1 Compute $\boldsymbol{P}^1, \cdots, \boldsymbol{P}^T$;

2 Compute $\boldsymbol{M} = \frac{\mathrm{vol}(G)}{bT} \left(\sum_{r=1}^{T} \boldsymbol{P}^r \right) \boldsymbol{D}^{-1}$;

3 Compute $\boldsymbol{M}' = \max(\boldsymbol{M}, 1)$;

4 Rank-d approximation by SVD: $\log \boldsymbol{M}' = \boldsymbol{U}_d \boldsymbol{\Sigma}_d \boldsymbol{V}_d^\top$;

5 **return** $\boldsymbol{U}_d \sqrt{\boldsymbol{\Sigma}_d}$ as network embedding.

Expensive

NetMF for a large window size *T*

- Factorize the DeepWalk matrix explicitly, e.g., using singular-value decomposition (SVD)

$$\log\left(\frac{\mathrm{vol}(G)}{b}\left(\frac{1}{T}\sum_{r=1}^{T}\left(\boldsymbol{D}^{-1}\boldsymbol{A}\right)^{r}\right)\boldsymbol{D}^{-1}\right)$$ This may be computationally challenging with a large T

- But we know the property of normalized graph Laplacian $\boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{D}^{-\frac{1}{2}} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^{\top}$ where $\boldsymbol{\Lambda} = diag(\lambda_1, \lambda_2, \cdots, \lambda\_n)$ and $\forall\lambda_i \in [-1, 1]$.

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

$$= \left(\boldsymbol{D}^{-1/2}\right)\left(\boldsymbol{U}\underbrace{\left(\frac{1}{T}\sum_{r=1}^{T}\boldsymbol{\Lambda}^{r}\right)}_{\text{A polynomial}}\boldsymbol{U}^{\top}\right)\left(\boldsymbol{D}^{-1/2}\right)$$

NetMF for a large window size T

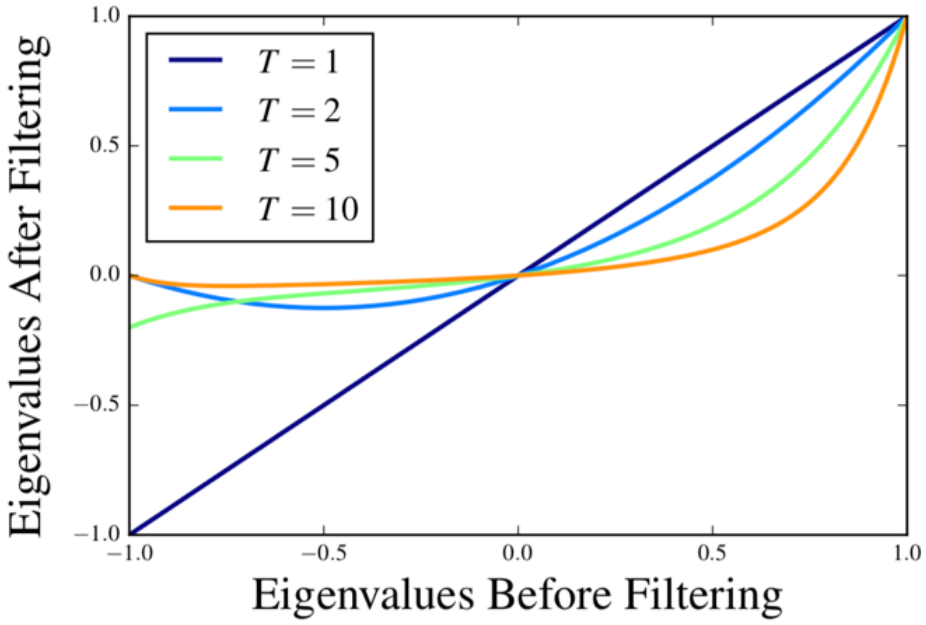


Figure 4: $f(\lambda) = \frac{1}{T}\sum_{r=1}^{T}\lambda^r$

- This polynomial implicitly filters out negative eigenvalues and small positive eigenvalues
- We can do it explicitly for efficiency

NetMF for a large window size T

1 Eigen-decomposition $\boldsymbol{D}^{-1/2}\boldsymbol{A}\boldsymbol{D}^{-1/2} \approx \boldsymbol{U}_h\boldsymbol{\Lambda}_h\boldsymbol{U}_h^\top$;

2 Approximate M with

$$\hat{\boldsymbol{M}} = \frac{\mathrm{vol}(G)}{b}\boldsymbol{D}^{-1/2}\boldsymbol{U}_h\left(\frac{1}{T}\sum_{r=1}^{T}\boldsymbol{\Lambda}_h^r\right)\boldsymbol{U}_h^\top\boldsymbol{D}^{-1/2};$$

3 Compute $\hat{\boldsymbol{M}}' = \max(\hat{\boldsymbol{M}}, 1)$;

4 Rank-d approximation by SVD: $\log \hat{\boldsymbol{M}}' = \boldsymbol{U}_d\boldsymbol{\Sigma}_d\boldsymbol{V}_d^\top$;

5 **return** $\boldsymbol{U}_d\sqrt{\boldsymbol{\Sigma}_d}$ as network embedding.

Approximate $\boldsymbol{D}^{-1/2}\boldsymbol{A}\boldsymbol{D}^{-1/2}$ with its top-h eigenpairs $\boldsymbol{U}_h\boldsymbol{\Lambda}_h\boldsymbol{U}_h^\top$

The Arnoldi algorithm [1] for significant time reduction

[1] R. Lehoucq, D. Sorensen, and C.Yang. 1998. ARPACK users' guide: solution of large-scale eigenvalue problems with implicitly restarted Arnoldi methods. SIAM.

Error Bound for NetMF for a large window size *T*

- According to Frobenius norm's property

$$\left|\log \boldsymbol{M}'_{i,j} - \log \hat{\boldsymbol{M}}'_{i,j}\right| = \log \frac{\hat{\boldsymbol{M}}'_{i,j}}{\boldsymbol{M}'_{i,j}} = \log\left(1 + \frac{\hat{\boldsymbol{M}}'_{i,j} - \boldsymbol{M}'_{i,j}}{\boldsymbol{M}'_{i,j}}\right)$$

$$\leq \frac{\hat{\boldsymbol{M}}'_{i,j} - \boldsymbol{M}'_{i,j}}{\boldsymbol{M}'_{i,j}} \leq \hat{\boldsymbol{M}}'_{i,j} - \boldsymbol{M}'_{i,j} = \left|\hat{\boldsymbol{M}}'_{i,j} - \boldsymbol{M}'_{i,j}\right|$$

- and because $M'_{i,j}=\max(M_{i,j}, 1)>=1$, we have

$$\left|\boldsymbol{M}'_{i,j} - \hat{\boldsymbol{M}}'_{i,j}\right| = \left|\max(\boldsymbol{M}_{i,j}, 1) - \max(\hat{\boldsymbol{M}}_{i,j}, 1)\right| \leq \left|\boldsymbol{M}_{i,j} - \hat{\boldsymbol{M}}_{i,j}\right|$$

- Also because the property of NGL,

$$\sigma_s\left(\left(\frac{1}{T}\sum_{r=1}^{T}\boldsymbol{P}^r\right)\boldsymbol{D}^{-1}\right) \leq \frac{\left|\frac{1}{T}\sum_{r=1}^{T}\lambda_{p_s}^r\right|}{d_{q_1}} \Rightarrow \left\|\boldsymbol{M} - \hat{\boldsymbol{M}}\right\|_F \leq \frac{\mathrm{vol}(G)}{b d_{\min}}\sqrt{\sum_{j=k+1}^{n}\left|\frac{1}{T}\sum_{r=1}^{T}\lambda_j^r\right|^2}$$

Experimental Setup

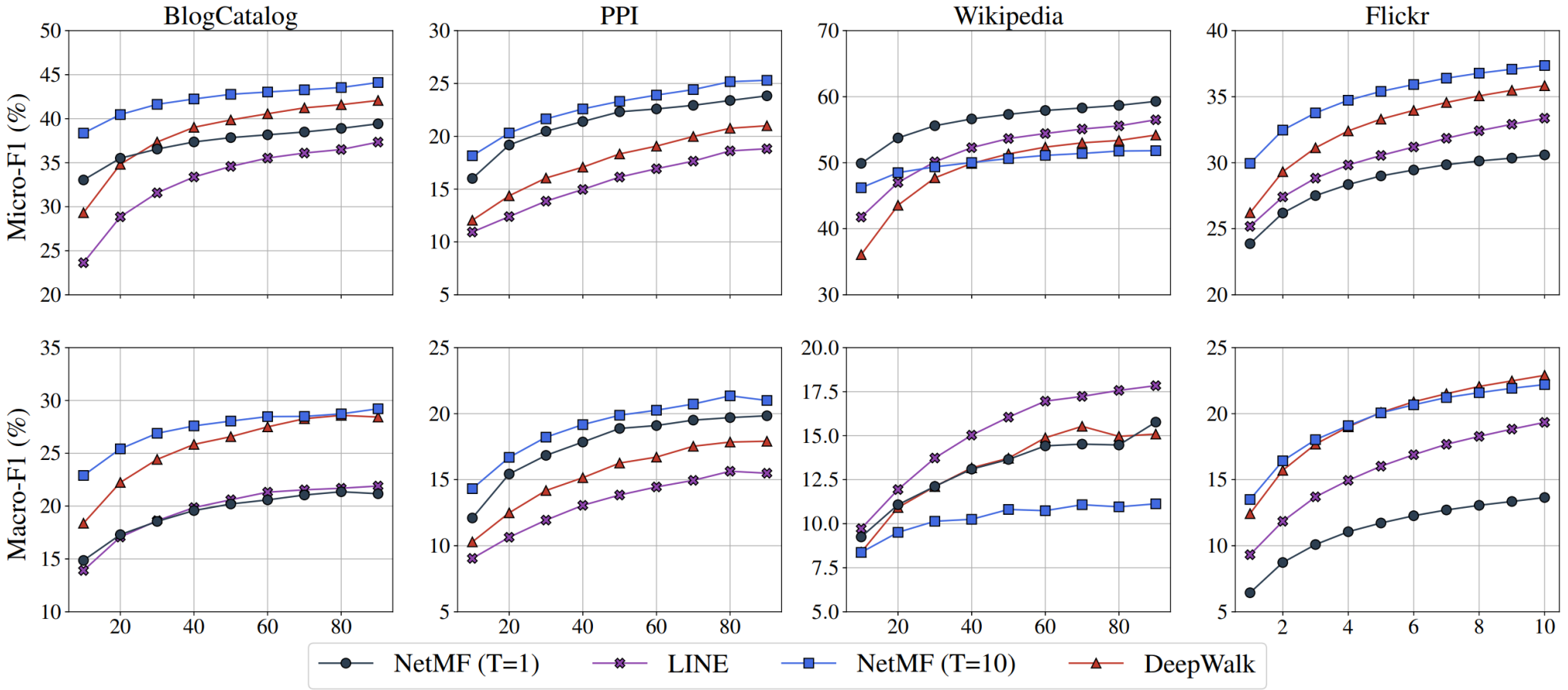
Label Classification:

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

Table 1: Statistics of Datasets.

Dataset	BlogCatalog	PPI	Wikipedia	Flickr
$\|V\|$	10,312	3,890	4,777	80,513
$\|E\|$	333,983	76,584	184,812	5,899,882
#Labels	39	50	40	195

Experimental Results



Predictive performance on varying the ratio of training data;

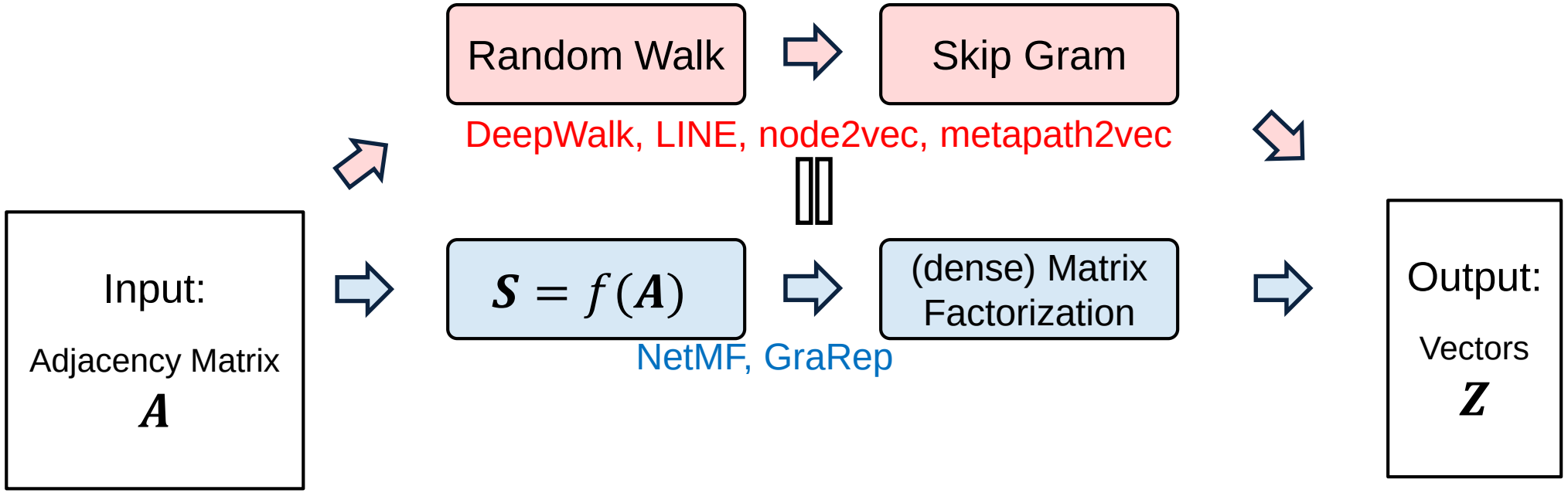
The *x*-axis represents the ratio of labeled data (%)

Network Embedding as Matrix Factorization

- DeepWalk $\log\left(\frac{\mathrm{vol}(G)}{b}\left(\frac{1}{T}\sum_{r=1}^{T}\left(\boldsymbol{D}^{-1}\boldsymbol{A}\right)^{r}\right)\boldsymbol{D}^{-1}\right)$
- LINE $\log\left(\frac{\mathrm{vol}(G)}{b}\boldsymbol{D}^{-1}\boldsymbol{A}\boldsymbol{D}^{-1}\right)$
- PTE $\log\left(\begin{bmatrix}\alpha\,\mathrm{vol}(G_{\mathsf{ww}})(\boldsymbol{D}_{\mathsf{row}}^{\mathsf{ww}})^{-1}\boldsymbol{A}_{\mathsf{ww}}(\boldsymbol{D}_{\mathsf{col}}^{\mathsf{ww}})^{-1}\\ \beta\,\mathrm{vol}(G_{\mathsf{dw}})(\boldsymbol{D}_{\mathsf{row}}^{\mathsf{dw}})^{-1}\boldsymbol{A}_{\mathsf{dw}}(\boldsymbol{D}_{\mathsf{col}}^{\mathsf{dw}})^{-1}\\ \gamma\,\mathrm{vol}(G_{\mathsf{lw}})(\boldsymbol{D}_{\mathsf{row}}^{\mathsf{lw}})^{-1}\boldsymbol{A}_{\mathsf{lw}}(\boldsymbol{D}_{\mathsf{col}}^{\mathsf{lw}})^{-1}\end{bmatrix}\right)-\log b$
- node2vec $\log\left(\frac{\frac{1}{2T}\sum_{r=1}^{T}\left(\sum_u \boldsymbol{X}_{w,u}\underline{\boldsymbol{P}}^{r}_{c,w,u}+\sum_u \boldsymbol{X}_{c,u}\underline{\boldsymbol{P}}^{r}_{w,c,u}\right)}{b\left(\sum_u \boldsymbol{X}_{w,u}\right)\left(\sum_u \boldsymbol{X}_{c,u}\right)}\right)$

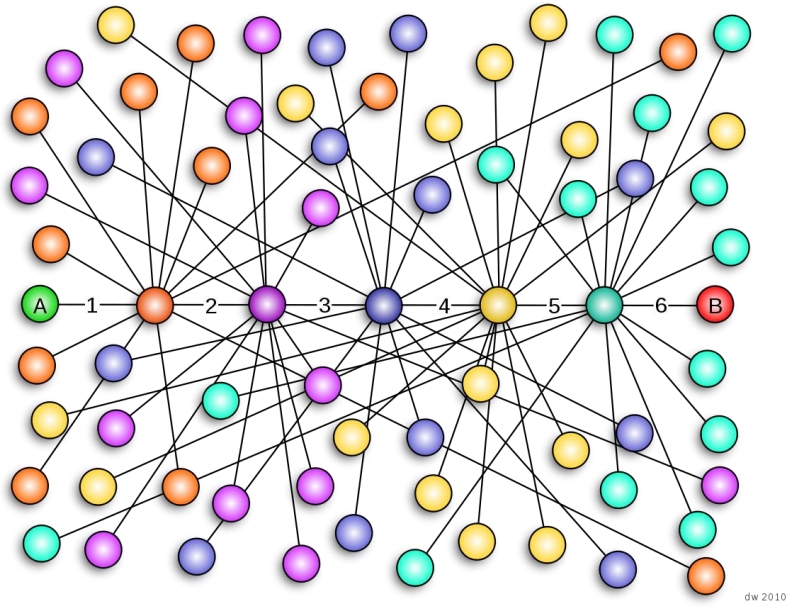
1. Qiu et al. Network embedding as matrix factorization: unifying deepwalk, line, pte, and node2vec. In *WSDM'18.* **The most cited paper in WSDM'18 as of Aug 2018**

Network Embedding



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

Challenges



$$\Rightarrow \quad \boldsymbol{S} = \log\left(\frac{\mathrm{vol}(G)}{b}\left(\frac{1}{T}\sum_{r=1}^{T}\left(\boldsymbol{D}^{-1}\boldsymbol{A}\right)^{r}\right)\boldsymbol{D}^{-1}\right)$$

dense

NetMF is not practical for very large networks

Recall NetMF for a large window size T

1 Eigen-decomposition $\boldsymbol{D}^{-1/2}\boldsymbol{A}\boldsymbol{D}^{-1/2} \approx \boldsymbol{U}_h\boldsymbol{\Lambda}_h\boldsymbol{U}_h^\top$;

2 Approximate $\boldsymbol{M}$ with

$$\hat{\boldsymbol{M}} = \frac{\mathrm{vol}(G)}{b}\boldsymbol{D}^{-1/2}\boldsymbol{U}_h\left(\frac{1}{T}\sum_{r=1}^{T}\boldsymbol{\Lambda}_h^r\right)\boldsymbol{U}_h^\top\boldsymbol{D}^{-1/2};$$

3 Compute $\hat{\boldsymbol{M}}' = \max(\hat{\boldsymbol{M}}, 1)$;

4 Rank-d approximation by SVD: $\log \hat{\boldsymbol{M}}' = \boldsymbol{U}_d\boldsymbol{\Sigma}_d\boldsymbol{V}_d^\top$;

5 **return** $\boldsymbol{U}_d\sqrt{\boldsymbol{\Sigma}_d}$ as network embedding.

Approximate $\boldsymbol{D}^{-1/2}\boldsymbol{A}\boldsymbol{D}^{-1/2}$ with its top-h eigenpairs $\boldsymbol{U}_h\boldsymbol{\Lambda}_h\boldsymbol{U}_h^\top$

The Arnoldi algorithm [1] for significant time reduction

[1] R. Lehoucq, D. Sorensen, and C.Yang. 1998. ARPACK users' guide: solution of large-scale eigenvalue problems with implicitly restarted Arnoldi methods. SIAM.

How can we solve this issue?

1. Construction
2. Factorization

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

1. Qiu et al. NetSMF: Network embedding as sparse matrix factorization. In **WWW 2019**

How can we solve this issue?

1. **Sparse** Construction
2. **Sparse** Factorization

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

1. Qiu et al. NetSMF: Network embedding as sparse matrix factorization. In **WWW 2019**

NetSMF: Network embedding as sparse matrix factorization

Qiu et al., NetSMF: Network embedding as sparse matrix factorization. In WWW 2019.

Sparsify $\boldsymbol{S}$

For random-walk matrix polynomial $\boldsymbol{L} = \boldsymbol{D} - \sum_{r=1}^{T} \alpha_r \boldsymbol{D} \left(\boldsymbol{D}^{-1}\boldsymbol{A}\right)^r$

where $\sum_{r=1}^{T} \alpha_r = 1$ and α_r non-negative

One can construct a $(1+\epsilon)$-spectral sparsifier $\tilde{\boldsymbol{L}}$ with $O(n \log n \epsilon^{-2})$ non-zeros

in time $O(T^2 m \epsilon^{-2} \log^2 n)$

$O(T^2 m \epsilon^{-2} \log n)$ for undirected graphs

- Dehua Cheng, Yu Cheng, Yan Liu, Richard Peng, and Shang-Hua Teng, Efficient Sampling for Gaussian Graphical Models via Spectral Sparsification, COLT 2015.
- Dehua Cheng, Yu Cheng, Yan Liu, Richard Peng, and Shang-Hua Teng. Spectral sparsification of random-walk matrix polynomials. arXiv:1502.03496.

Sparsify $\boldsymbol{S}$

For random-walk matrix polynomial $\boldsymbol{L} = \boldsymbol{D} - \sum_{r=1}^{T} \alpha_r \boldsymbol{D} \left(\boldsymbol{D}^{-1}\boldsymbol{A}\right)^r$

where $\sum_{r=1}^{T} \alpha_r = 1$ and α_r non-negative

One can construct a $(1+\epsilon)$-spectral sparsifier $\tilde{\boldsymbol{L}}$ with $O(n \log n \epsilon^{-2})$ non-zeros

in time $O(T^2 m \epsilon^{-2} \log^2 n)$

Suppose $G = (V, E, \boldsymbol{A})$ and $\widetilde{G} = (V, \widetilde{E}, \widetilde{\boldsymbol{A}})$ are two weighted undirected networks. Let $\boldsymbol{L} = \boldsymbol{D}_G - \boldsymbol{A}$ and $\widetilde{\boldsymbol{L}} = \boldsymbol{D}_{\widetilde{G}} - \widetilde{\boldsymbol{A}}$ be their Laplacian matrices, respectively. We define G and $\widetilde{G}$ are $(1+\epsilon)$-spectrally similar if

$$\forall \boldsymbol{x} \in \mathbb{R}^n, (1-\epsilon) \cdot \boldsymbol{x}^\top \widetilde{\boldsymbol{L}} \boldsymbol{x} \leq \boldsymbol{x}^\top \boldsymbol{L} \boldsymbol{x} \leq (1+\epsilon) \cdot \boldsymbol{x}^\top \widetilde{\boldsymbol{L}} \boldsymbol{x}.$$

Sparsify $\boldsymbol{S}$

For random-walk matrix polynomial $\boldsymbol{L} = \boldsymbol{D} - \sum_{r=1}^{T} \alpha_r \boldsymbol{D} \left(\boldsymbol{D}^{-1}\boldsymbol{A}\right)^r$

where $\sum_{r=1}^{T} \alpha_r = 1$ and α_r non-negative

One can construct a $(1+\epsilon)$-spectral sparsifier $\tilde{\boldsymbol{L}}$ with $O(n \log n \epsilon^{-2})$ non-zeros

in time $O(T^2 m \epsilon^{-2} \log^2 n)$

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

Sparsify $\boldsymbol{S}$

For random-walk matrix polynomial $\boldsymbol{L} = \boldsymbol{D} - \sum_{r=1}^{T} \alpha_r \boldsymbol{D} \left(\boldsymbol{D}^{-1}\boldsymbol{A}\right)^r$

where $\sum_{r=1}^{T} \alpha_r = 1$ and α_r non-negative

One can construct a $(1+\epsilon)$-spectral sparsifier $\tilde{\boldsymbol{L}}$ with $O(n \log n \epsilon^{-2})$ non-zeros

in time $O(T^2 m \epsilon^{-2} \log^2 n)$

$$\alpha_1 = \cdots = \alpha_T = \tfrac{1}{T} \quad \Rightarrow$$

$$\begin{aligned}
&\log^{\circ}\left(\frac{\mathrm{vol}(G)}{b}\left(\frac{1}{T}\sum_{r=1}^{T}\left(\boldsymbol{D}^{-1}\boldsymbol{A}\right)^r\right)\boldsymbol{D}^{-1}\right)\\
=&\log^{\circ}\left(\frac{\mathrm{vol}(G)}{b}\boldsymbol{D}^{-1}(\boldsymbol{D}-\boldsymbol{L})\boldsymbol{D}^{-1}\right)\\
\approx&\log^{\circ}\left(\frac{\mathrm{vol}(G)}{b}\boldsymbol{D}^{-1}(\boldsymbol{D}-\tilde{\boldsymbol{L}})\boldsymbol{D}^{-1}\right)
\end{aligned}$$

NetSMF --- Sparse

- Construct a random walk matrix polynomial sparsifier, $\widetilde{\boldsymbol{L}}$
- Construct a NetMF matrix sparsifier.

$$\text{trunc\_log}^{\circ}\left(\frac{\text{vol}(G)}{b}\boldsymbol{D}^{-1}(\boldsymbol{D}-\widetilde{\boldsymbol{L}})\boldsymbol{D}^{-1}\right)$$

- Factorize the constructed matrix

NetSMF --- Sparse

Input : A social network $G = (V, E, \boldsymbol{A})$ which we want to learn network embedding; The number of non-zeros M in the sparsifier; The dimension of embedding d.

Output: An embedding matrix of size $n \times d$, each row corresponding to a vertex.

$\widetilde{G} \leftarrow (V, \emptyset, \widetilde{\boldsymbol{A}} = \boldsymbol{0})$

/* Create an empty network with $E = \emptyset$ and $\widetilde{\boldsymbol{A}} = 0$. */

for $i \leftarrow 1$ **to** M **do**

  Uniformly pick an edge $e = (u, v) \in E$

  Uniformly pick an integer $r \in [1 : T]$

  $u', v', Z \leftarrow$ PathSampling(*e*, *r*)

  Add an edge $\left(u', v', \frac{2rm}{MZ}\right)$ to $\widetilde{G}$

  /* Parallel edges will be merged into one edge, with their weights summed up together. */

end

Compute $\widetilde{\boldsymbol{L}}$ to be the unnormalized graph Laplacian of $\widetilde{G}$

Compute $\widetilde{\boldsymbol{M}} = \boldsymbol{D}^{-1}\left(\boldsymbol{D} - \widetilde{\boldsymbol{L}}\right)\boldsymbol{D}^{-1}$

$\boldsymbol{U}_d, \boldsymbol{\Sigma}_d, \boldsymbol{V}_d \leftarrow$ RandomizedSVD$\left(\text{trunc_log}^{\circ}\left(\frac{\text{vol}(G)}{b}\widetilde{\boldsymbol{M}}\right), d\right)$

return $\boldsymbol{U}_d\sqrt{\boldsymbol{\Sigma}_d}$ as network embeddings

Algorithm 5: PathSampling algorithm as described in [CCL+15].

Procedure `PathSampling`$(e = (u, v),\ r)$

- Uniformly pick an integer $k \in [1 : r]$
- Perform $(k-1)$-step random walk from u to u_0
- Perform $(r-k)$-step random walk from v to u_r
- Keep track of $Z(\boldsymbol{p}) = \sum_{i=1}^{r} \frac{2}{\boldsymbol{A}_{u_{i-1},u_i}}$ along the length-r path $\boldsymbol{p}$ between u_0 and u_r
- **return** $u_0, u_r, Z(\boldsymbol{p})$

- Dehua Cheng, Yu Cheng, Yan Liu, Richard Peng, and Shang-Hua Teng, Efficient Sampling for Gaussian Graphical Models via Spectral Sparsification, COLT 2015.
- Dehua Cheng, Yu Cheng, Yan Liu, Richard Peng, and Shang-Hua Teng. Spectral sparsification of random-walk matrix polynomials. arXiv:1502.03496.

	Time	Space
Step 1	$O(MT \log n)$ for weighted networks $O(MT)$ for unweighted networks	$O(M + n + m)$
Step 2	$O(M)$	$O(M + n)$
Step 3	$O(Md + nd^2 + d^3)$	$O(M + nd)$

NetSMF---bounded approximation error

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

$$=\log^\circ\left(\frac{\mathrm{vol}(G)}{b}\boldsymbol{D}^{-1}(\boldsymbol{D}-\boldsymbol{L})\boldsymbol{D}^{-1}\right)\longrightarrow \boldsymbol{M}$$

$$\approx\log^\circ\left(\frac{\mathrm{vol}(G)}{b}\boldsymbol{D}^{-1}(\boldsymbol{D}-\widetilde{\boldsymbol{L}})\boldsymbol{D}^{-1}\right)\longrightarrow \widetilde{\boldsymbol{M}}$$

Theorem

The singular value of $\widetilde{\boldsymbol{M}}-\boldsymbol{M}$ satisfies

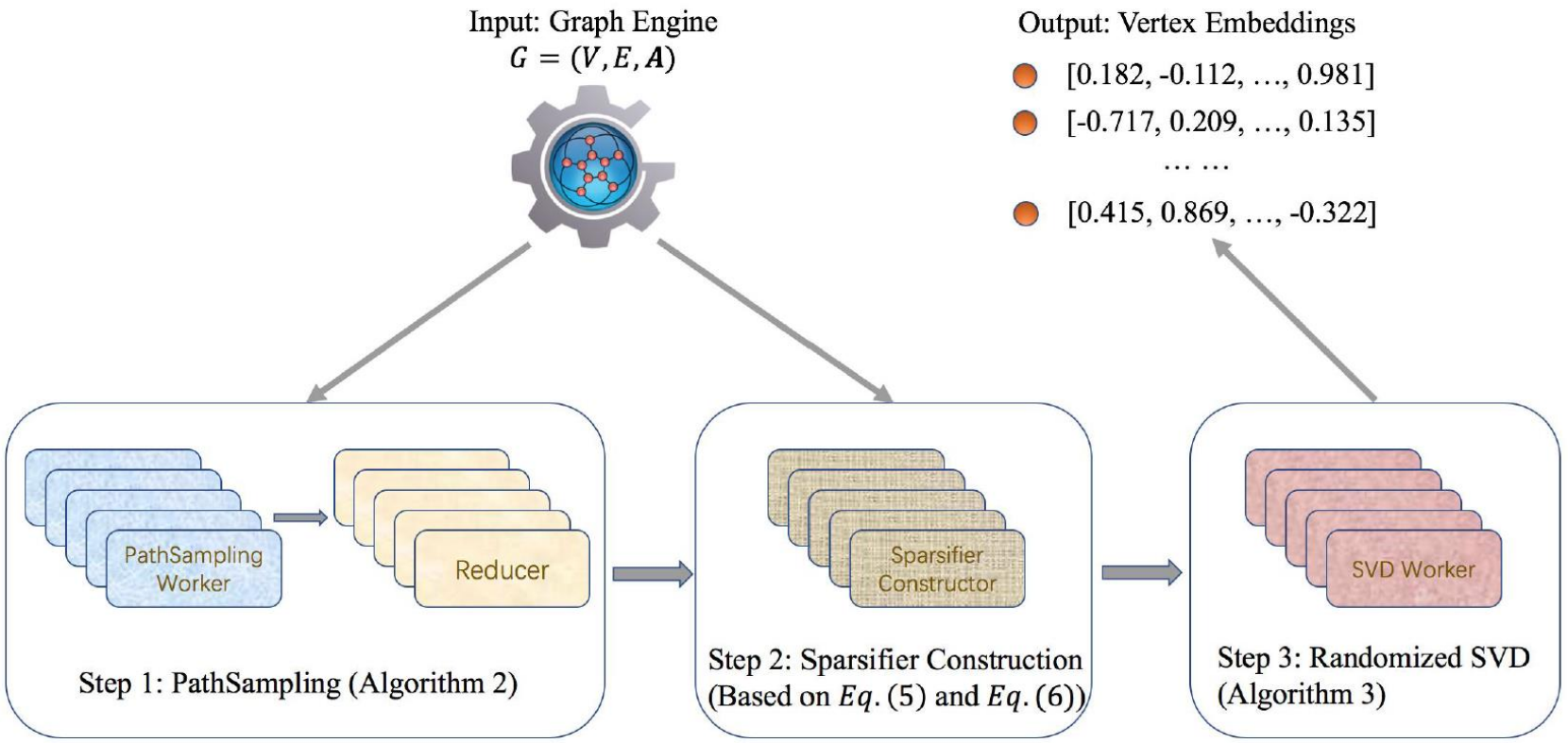
$$\sigma_i(\widetilde{\boldsymbol{M}}-\boldsymbol{M})\leq\frac{4\epsilon}{\sqrt{d_i d_{\min}}},\forall i\in[n].$$

Theorem

Let $\|\cdot\|_F$ be the matrix Frobenius norm. Then

$$\left\|\mathrm{trunc\_log}^\circ\left(\frac{\mathrm{vol}(G)}{b}\widetilde{\boldsymbol{M}}\right)-\mathrm{trunc\_log}^\circ\left(\frac{\mathrm{vol}(G)}{b}\boldsymbol{M}\right)\right\|_F\leq\frac{4\epsilon\,\mathrm{vol}(G)}{b\sqrt{d_{\min}}}\sqrt{\sum_{i=1}^{n}\frac{1}{d_i}}.$$

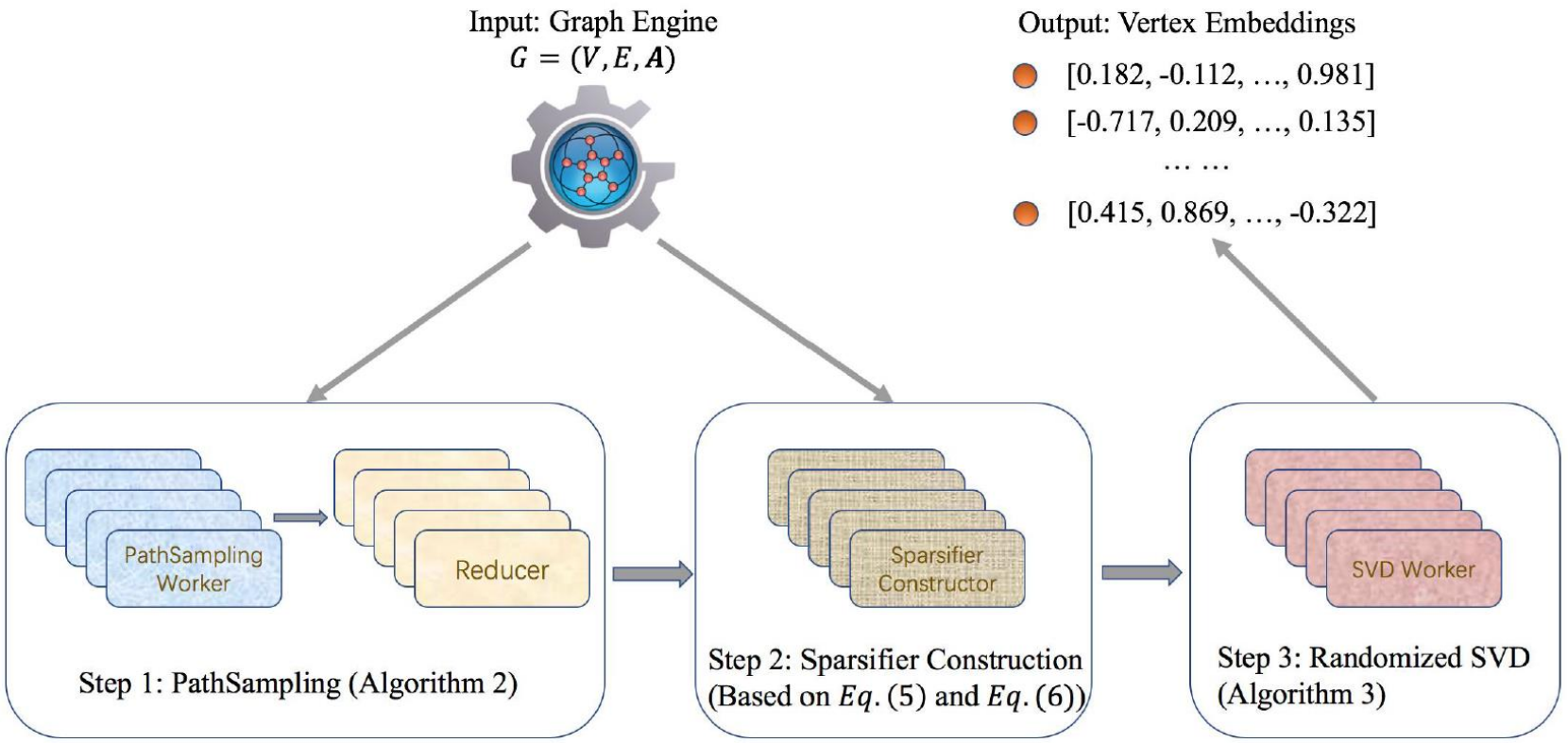
Distributed system design

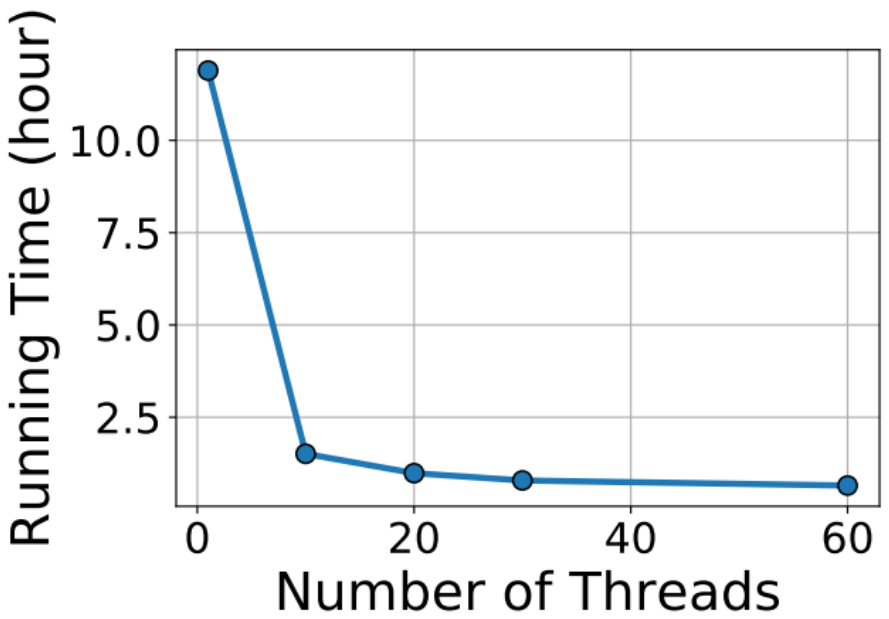


Dataset	BlogCatalog	PPI	Flickr	YouTube	OAG
$\|V\|$	10,312	3,890	80,513	1,138,499	67,768,244
$\|E\|$	333,983	76,584	5,899,882	2,990,443	895,368,962
#labels	39	50	195	47	19

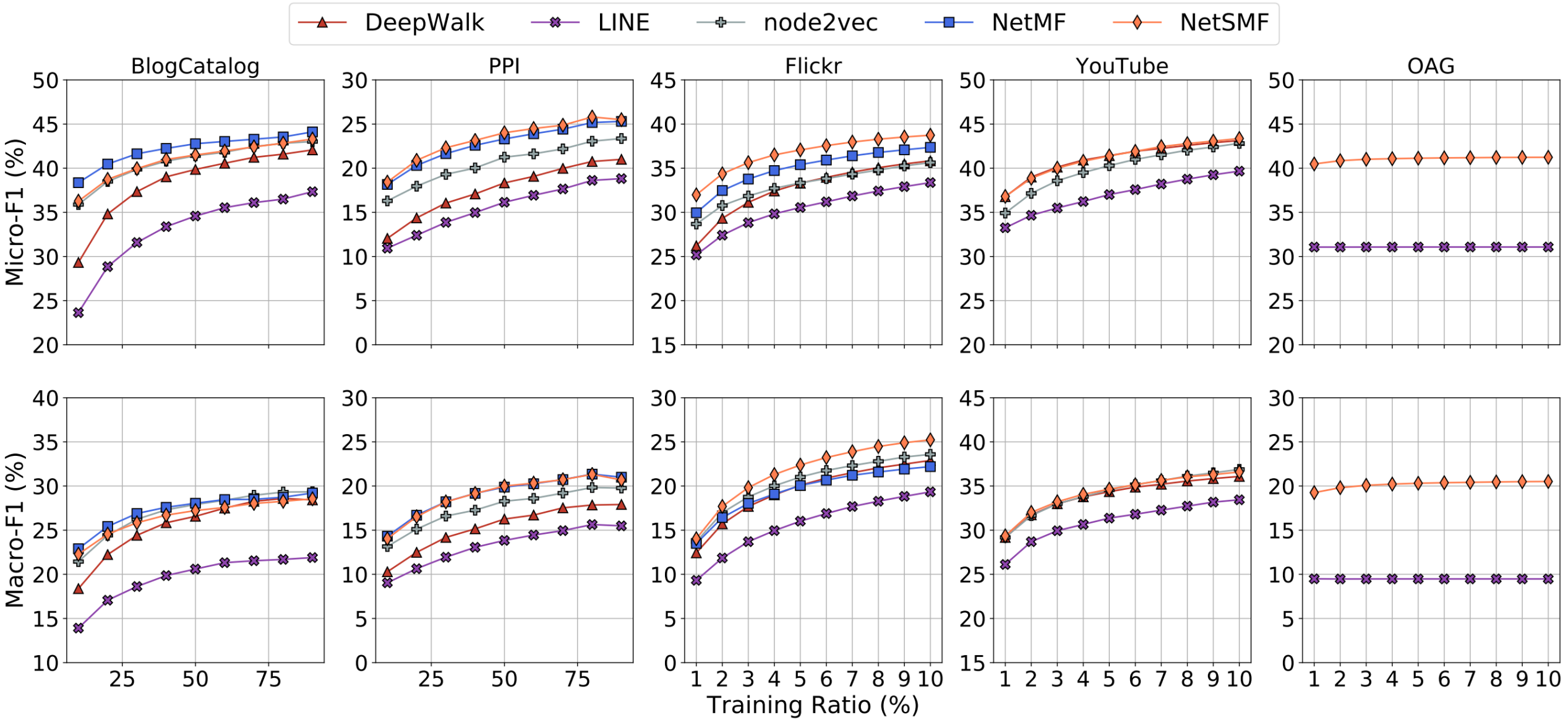
	LINE	DeepWalk	node2vec	NetMF	NetSMF
BlogCatalog	40 mins	12 mins	56 mins	19 mins	13 mins
PPI	41 mins	4 mins	4 mins	1 min	10 secs
Flickr	42 mins	2.2 hours	21 hours	5 days	48 mins
YouTube	46 mins	4.3 hours	4 days	×	4.1 hours
OAG	2.6 hours	–	–	×	24 hours

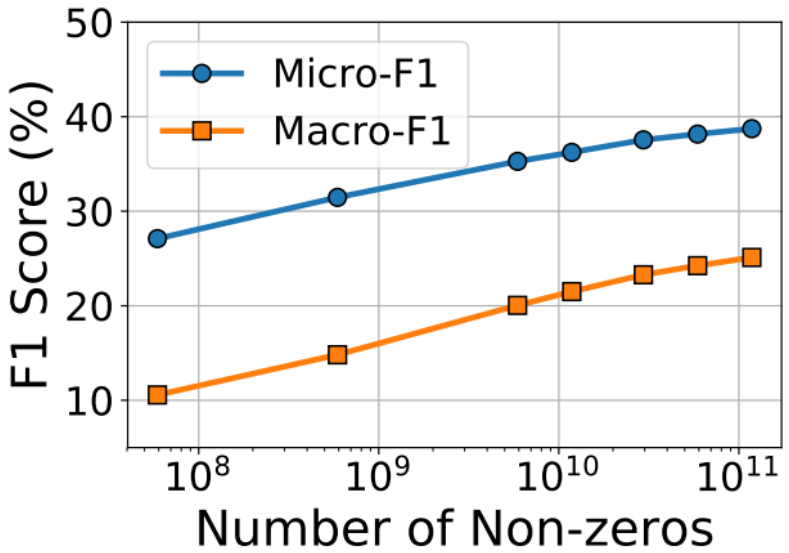
Distributed system design



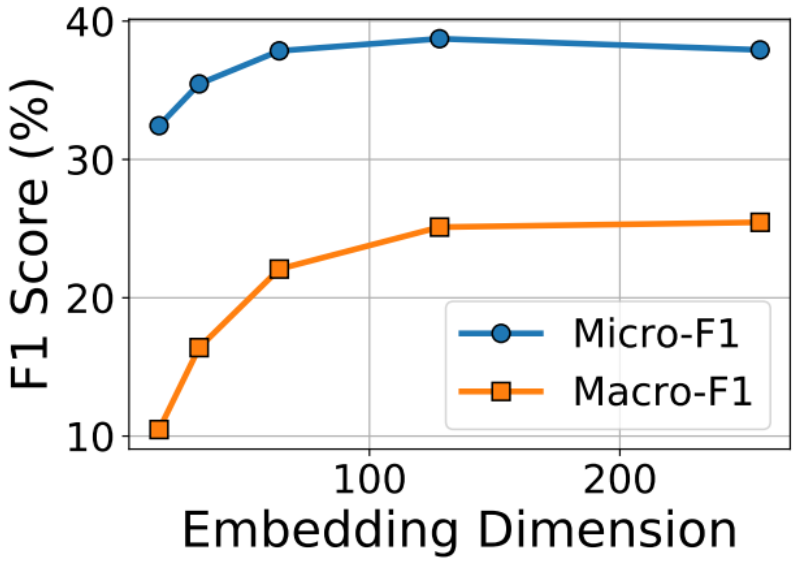


(d)

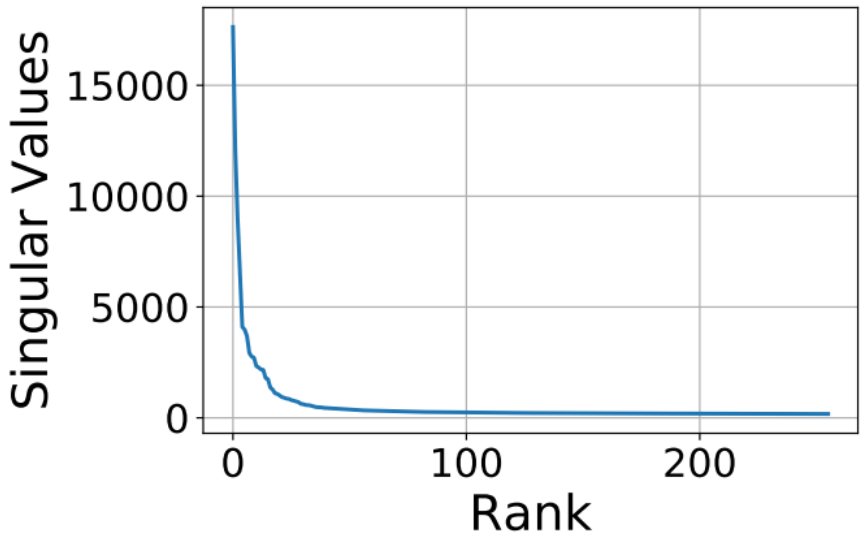

DeepWalk
LINE
node2vec
NetMF
NetSMF
BlogCatalog
PPI
Flickr
YouTube
OAG
Micro-F1 (%)
Macro-F1 (%)
Training Ratio (%)


50
40
30
20
10
F1 Score (%)
Micro-F1
Macro-F1
10^8
10^9
10^{10}
10^{11}
Number of Non-zeros

(c)

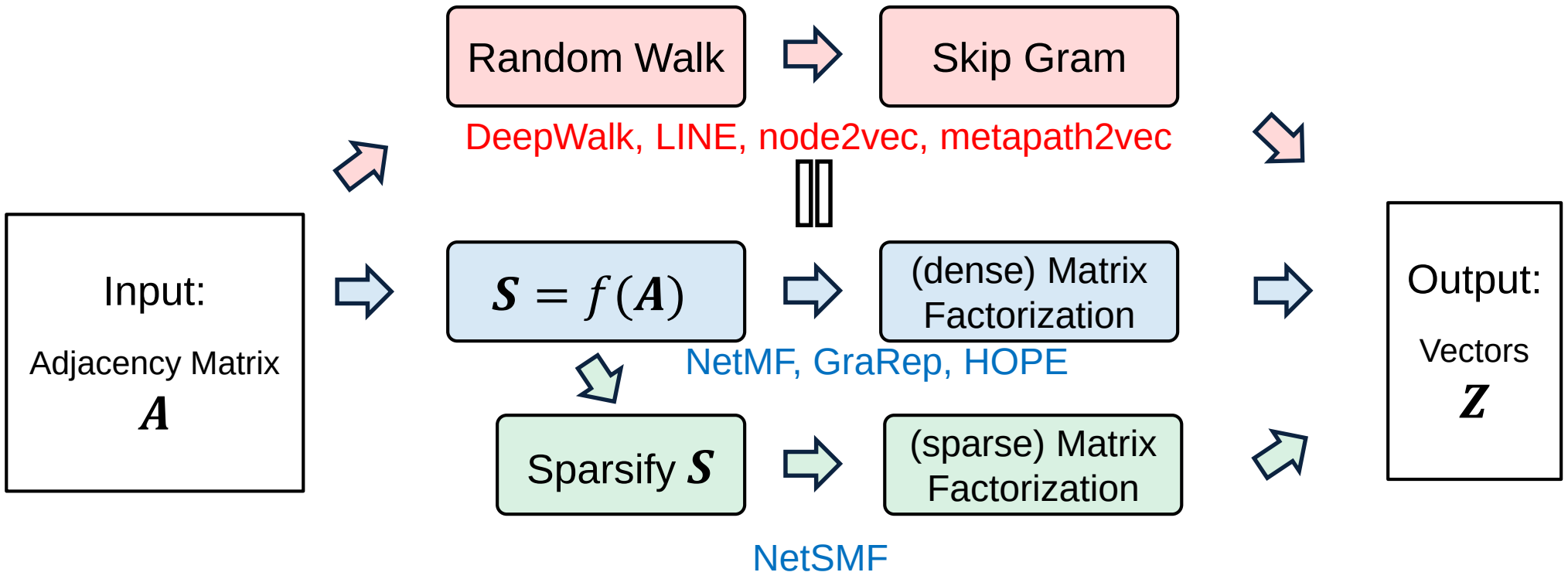

40
30
20
10
F1 Score (%)
100
200
Embedding Dimension
Micro-F1
Macro-F1

(a)


15000
10000
5000
0
Singular Values
0
100
200
Rank

(b)

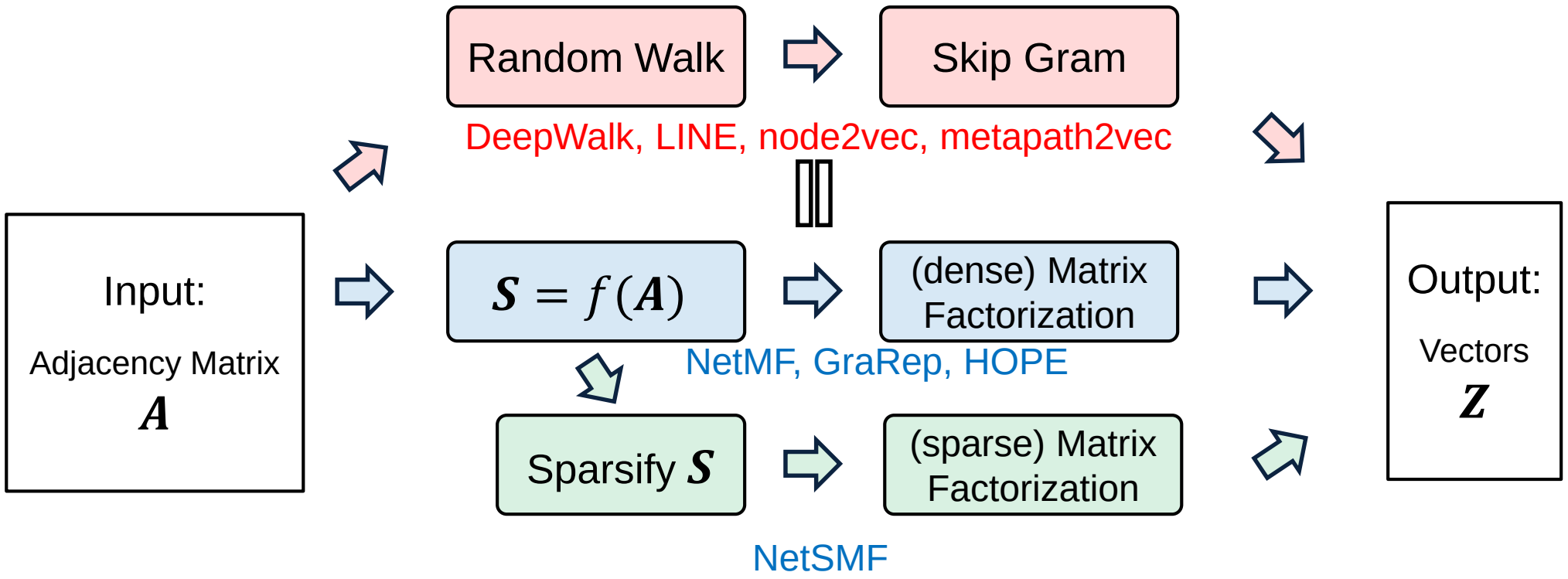
Network Embedding



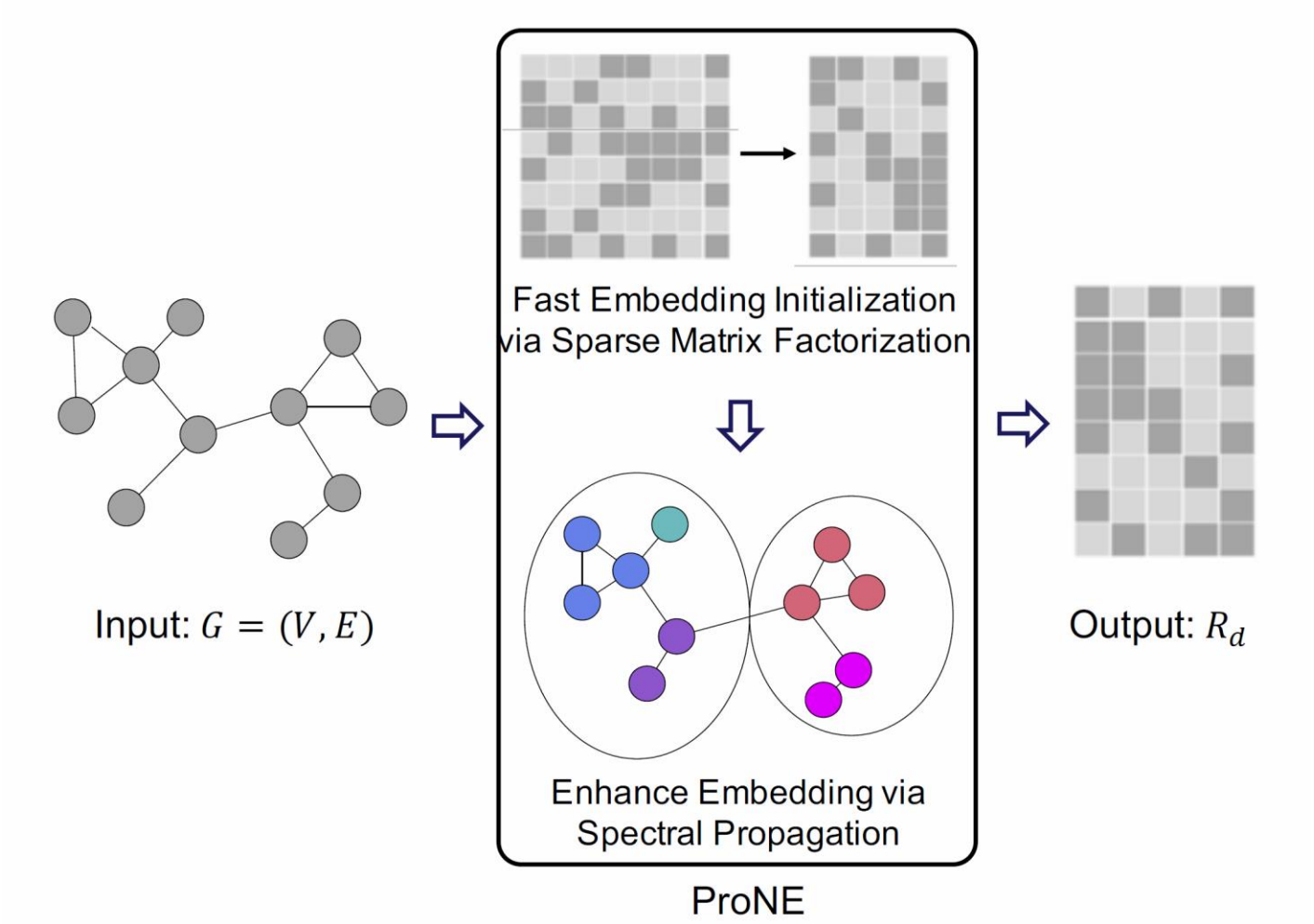
Much More Fast and Scalable Network Representation Learning

Zhang et al., ProNE: Fast and Scalable Network Representation Learning. In IJCAI 2019

Network Embedding



ProNE: More fast & scalable network embedding



Distributional Hypothesis of Harris

- **Word embedding**: words in similar contexts have similar meanings (e.g., skip-gram in word embedding)
- **Node embeddings**: nodes in similar structural contexts are similar
  - RW + Skip-Gram: structural contexts are defined by co-occurrence over random walk paths

Network embedding as sparse matrix factorization

- The occurrence probability of context v_j given node v_i as

$$\hat{p}_{i,j} = \sigma(r_i^T c_j)$$

- Objective (weighted sum of logloss)

$$l = -\sum_{(i,j)\in\mathcal{D}} [p_{i,j} \ln \hat{p}_{i,j}]$$

$$= -\sum_{(i,j)\in\mathcal{D}} [p_{i,j} \ln \sigma(r_i^T c_j)]$$

$$p_{ij} = A_{ij}/D_{ii}$$

$$\mathcal{D} = E$$

sparsity

Network embedding as sparse matrix factorization

- To avoid the trivial solution $(r_i = c_j, r_i^T c_j \to \infty, s.t. \hat{p} \to 1)$
  - Negative samples are drawn from $P_{\mathcal{D},j} \propto \sum_{i:(i,j)\in\mathcal{D}} p_{i,j}$
  - Negative samples are only from the local neighborhood (sparse!)
- Updated loss (sum over the edge → sparse!)

$$l = -\sum_{(i,j)\in\mathcal{D}} [p_{i,j} \ln \sigma(r_i^T c_j) + \lambda P_{\mathcal{D},j} \ln \sigma(-r_i^T c_j)]$$

Network embedding as sparse matrix factorization

- Node similarity in vector space

$$r_i^T c_j = \ln p_{i,j} - \ln(\lambda P_{D,j}), \quad (v_i, v_j) \in \mathcal{D}$$

- Sparse similarity matrix:

$$M_{i,j} = \begin{cases} \ln p_{i,j} - \ln(\lambda P_{D,j}) & , (v_i, v_j) \in \mathcal{D} \\ 0 & , (v_i, v_j) \notin \mathcal{D} \end{cases}$$

- Network embedding as sparse matrix factorization
  - tSVD

$$M \approx U_d \Sigma_d V_d^T$$

$$R_d \leftarrow U_d \Sigma_d^{1/2}$$

  - Randomized tSVD

$$M \approx QQ^T M = (QS_d)\Sigma_d V_d^T$$

$$R_d = QS_d \Sigma_d^{1/2}$$

$O(|E|)$ time

NE as sparse matrix factorization

- Compared with the matrix factorization version of DeepWalk (i.e., NetMF)

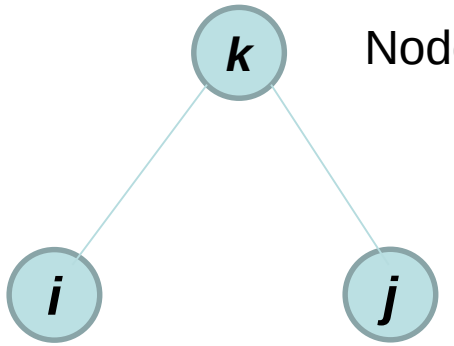
$$\log\left(\frac{\mathrm{vol}(G)}{b}\left(\frac{1}{T}\sum_{r=1}^{T}\left(\boldsymbol{D}^{-1}\boldsymbol{A}\right)^{r}\right)\boldsymbol{D}^{-1}\right)$$

$$M_{i,j}=\begin{cases}\ln p_{i,j}-\ln(\lambda P_{D,j}) & ,(v_i,v_j)\in\mathcal{D}\\ 0 & ,(v_i,v_j)\notin\mathcal{D}\end{cases}$$

- Sparsity (local structure and local negative samples)
  - leads to much faster and scalable learning
  - but may lose high-order structure information!

Embedding enhancement

- Basic idea: again, distributional hypothesis of Harris
- An intuitive example---similarity is amplified
- How to propagate the embedding?



Node *k* as the context node for *i* and *j*

$$(R_i + R_k)^T (R_j + R_k) = R_i^T R_j + R_i^T R_k + R_i^T R_k + R_k^T R_k$$

Embedding enhancement

How to propagate the embeddings?

Embedding enhancement

- Graph Spectral
  - Normalized graph Laplacian

    $L=I_n-D^{-1}A$

  - Decomposition

    $L=U\Lambda U^T$

    $\Lambda=diag([\lambda_1,...,\lambda_n])$

    $0=\lambda_1 \leq \cdots \leq \lambda_n$

- Graph Partition
  - Cheeger constant (conductance)

    For a partition $S \subseteq V$, $\phi(S)=\frac{|E(S)|}{min\{vol(S),vol(V-S)\}}$

  - k-way Cheeger constant

    $\rho_G(k)=min\{max\{\phi(S_i):S_1,S_2,...,S_k \subseteq V\ disjoint\}\}$

Higher-order Cheeger's inequality

$$\frac{\lambda_k}{2} \leq \rho_G(k) \leq O(k^2)\sqrt{\lambda_k}$$

Embedding enhancement

Higher-order Cheeger's inequality

$$\frac{\lambda_k}{2} \leq \rho_G(k) \leq O(k^2)\sqrt{\lambda_k}$$

- Low eigenvalues control the global information
- High eigenvalues control the local information
  - Example: $\lambda_2 = 0 \iff$ Graph is disconnected
- Spectral propagation: propagate the initial network embedding in the spectrally modulated/partitioned network!

Embedding enhancement via spectral propagation

$$R_d \leftarrow D^{-1}A(I_n - \tilde{L})\, R_d$$

$\widetilde{L} = Ug(\Lambda)U^T$ is the spectral filter of $L = I_n - D^{-1}A$

$D^{-1}A(I_n - \tilde{L})$ is $D^{-1}A$ modulated by the filter in the spectrum

Embedding Enhancement via Spectral Propagation

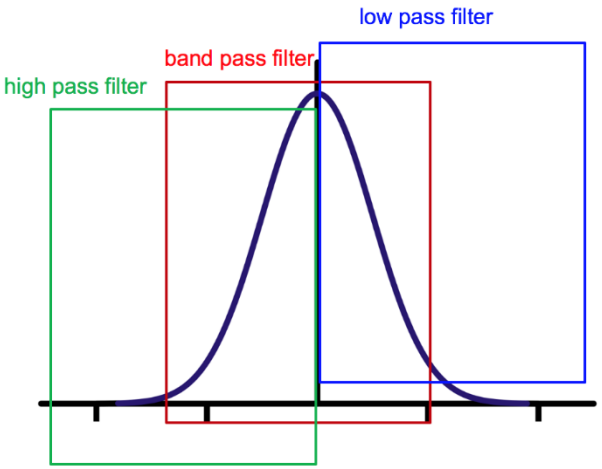
$$R_d \leftarrow D^{-1}A(I_n - \tilde{L})\, R_d$$

$$\widetilde{L} = U g(\Lambda) U^T$$ is the spectral filter of $L = I_n - D^{-1}A$

$$= U diag([g(\lambda_1), ..., g(\lambda_n)]) U^T$$

$$g(\lambda) = e^{-\frac{1}{2}[(\lambda-\mu)^2-1]\theta}$$

- Band-pass (low-pass, high-pass)



• Chebyshev Expansion for Efficiency

- Chebyshev expansion

$$T_{i+1}(x) = 2xT_i(x) - T_{i-1}(x) \text{ with } T_0(x) = 1, T_1(x) = x$$

to avoid explicit eigendecomposition and Fourier transform

$$\widetilde{L} = U diag([g(\lambda_1), ..., g(\lambda_n)])U^T$$

$$\approx U \sum_{i=0}^{k-1} c_i(\theta) T_i(\bar{\Lambda}) U^T$$

$$= \sum_{i=0}^{k-1} c_i(\theta) T_i(\bar{L})$$

$$\widetilde{L} \approx B_0(\theta)T_0(\bar{L}) + 2\sum_{i=1}^{k-1}(-)^i B_i(\theta)T_i(\bar{L})$$

Embedding enhancement via spectral propagation

$$R_d \leftarrow D^{-1}A(I_n - \tilde{L})\, R_d$$

ProNE Complexity

- Time complexity of the SMF step

  $O(|V|d^2 + |E|)$

- Time complexity of the Enhancement step

  $O(k|E|)$

- Time complexity

  $O(|V|d^2 + k|E|)$

- Space complexity

  $O(|V|d + k|E|)$

- Parallelizability

  $\frac{p}{\log(p)} \times$ speedup

Efficiency

20 Threads

Dataset	*DeepWalk*	*LINE*	*node2vec*
PPI	272	70	828
Wiki	494	87	939
BlogCatalog	1,231	185	3,533
DBLP	3,825	1,204	4,749
Youtube	68,272	5,890	>5days

19hours 98mins

1.1M nodes

Efficiency

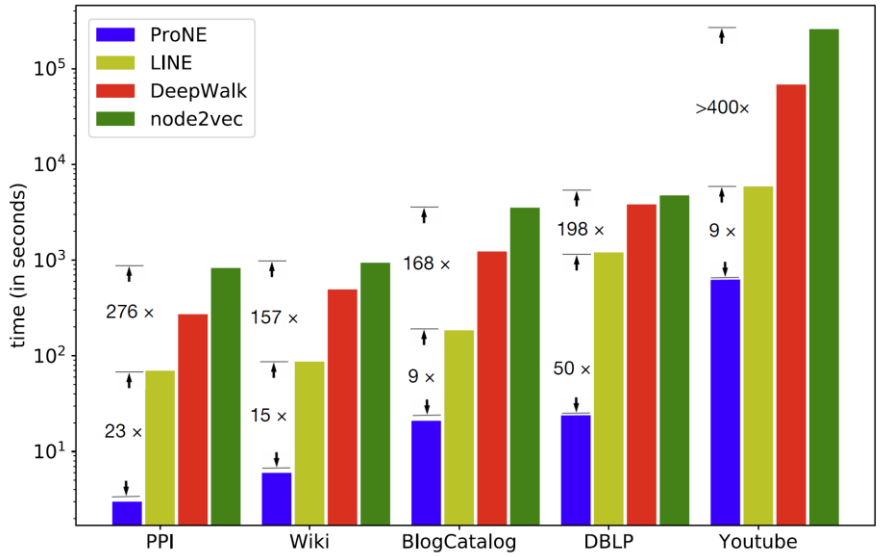
20 Threads

1 Thread

Dataset	*DeepWalk*	*LINE*	*node2vec*	*ProNE*
PPI	272	70	828	**3**
Wiki	494	87	939	**6**
BlogCatalog	1,231	185	3,533	**21**
DBLP	3,825	1,204	4,749	**24**
Youtube	68,272	5,890	>5days	**627**

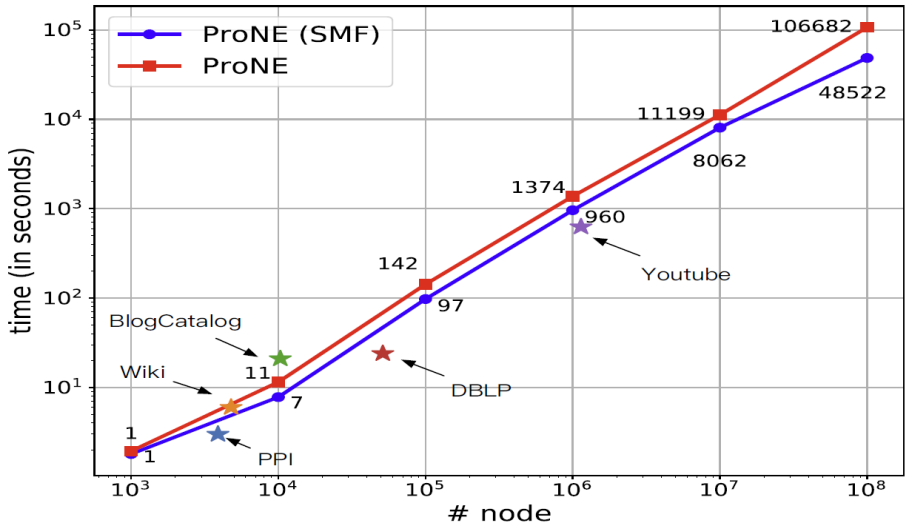
19hours 98mins 10mins

1.1M nodes



ProNE offers 10-400X speedups (1 thread vs 20 threads)

Efficiency

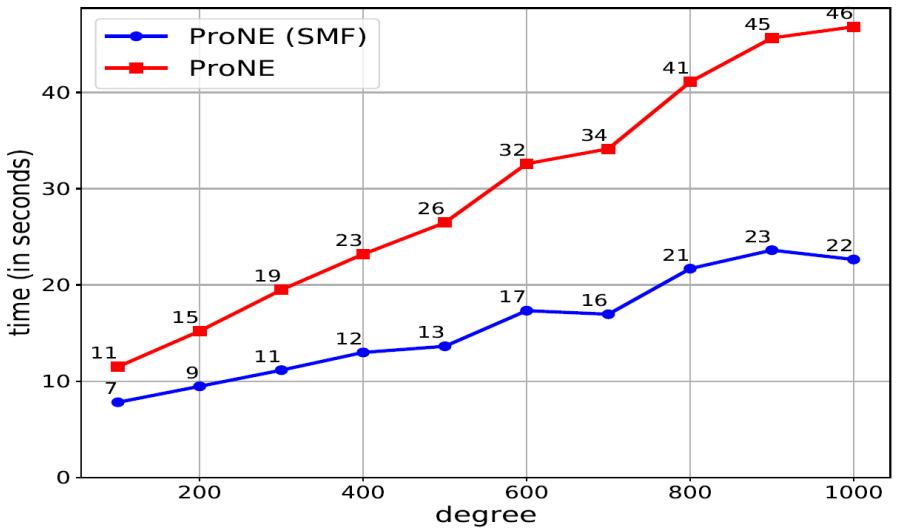


(a) The node degree is fixed to 10 and #nodes grows

Embed 100,000,000 nodes by one thread:
29 hours

$\frac{p}{\log(p)} \times$ speedup

Efficiency



(b) #nodes is fixed to 10, 000 and the node degree grows

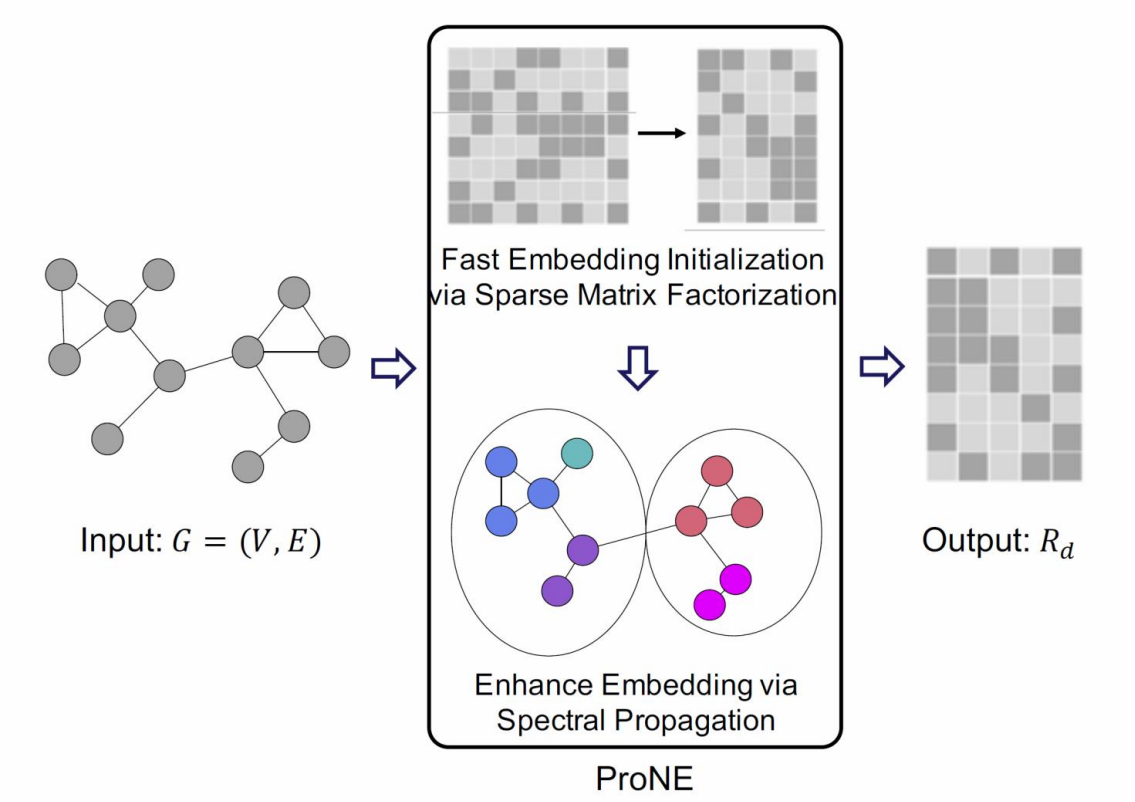
The efficiency of ProNE is linearly correlated with network density

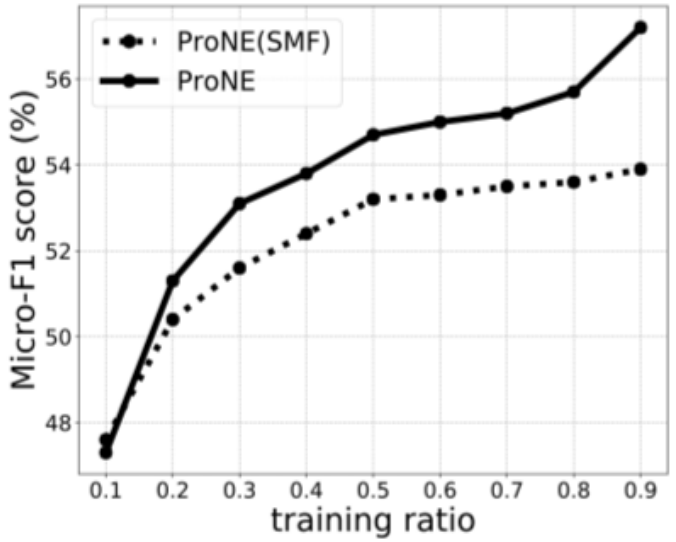
Effectiveness

Dataset	training ratio	0.1	0.3	0.5	0.7	0.9
PPI	DeepWalk	16.4	19.4	21.1	22.3	22.7
	LINE	16.3	20.1	21.5	22.7	23.1
	node2vec	16.2	19.7	21.6	23.1	24.1
	GraRep	15.4	18.9	20.2	20.4	20.9
	HOPE	16.4	19.8	21.0	21.7	22.5
	ProNE (SMF)	15.8	20.6	22.7	23.7	24.2
	ProNE	**18.2**	**22.7**	**24.6**	**25.4**	**25.9**
	($\pm\sigma$)	($\pm$0.5)	($\pm$0.3)	($\pm$0.7)	($\pm$1.0)	($\pm$1.1)
Wiki	DeepWalk	40.4	45.9	48.5	49.1	49.4
	LINE	**47.8**	50.4	51.2	51.6	52.4
	node2vec	45.6	47.0	48.2	49.6	50.0
	GraRep	47.2	49.7	50.6	50.9	51.8
	HOPE	38.5	39.8	40.1	40.1	40.1
	ProNE (SMF)	47.6	51.6	53.2	53.5	53.9
	ProNE	47.3	**53.1**	**54.7**	**55.2**	**57.2**
	($\pm\sigma$)	($\pm$0.7)	($\pm$0.4)	($\pm$0.8)	($\pm$0.8)	($\pm$1.3)
BlogCatalog	DeepWalk	36.2	39.6	40.9	41.4	42.2
	LINE	28.2	30.6	33.2	35.5	36.8
	node2vec	**36.3**	39.7	41.1	42.0	42.1
	GraRep	34.0	32.5	33.3	33.7	34.1
	HOPE	30.7	33.4	34.3	35.0	35.3
	ProNE (SMF)	34.6	37.6	38.6	39.3	39.0
	ProNE	36.2	**40.0**	**41.2**	**42.1**	**42.7**
	($\pm\sigma$)	($\pm$0.5)	($\pm$0.3)	($\pm$0.6)	($\pm$0.7)	($\pm$1.2)

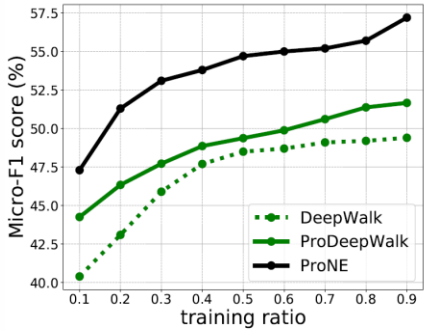
Dataset	training ratio	0.01	0.03	0.05	0.07	0.09
DBLP	DeepWalk	49.3	55.0	57.1	57.9	58.4
	LINE	48.7	52.6	53.5	54.1	54.5
	node2vec	48.9	55.1	57.0	58.0	58.4
	GraRep	50.5	52.6	53.2	53.5	53.8
	HOPE	**52.2**	55.0	55.9	56.3	56.6
	ProNE (SMF)	50.8	54.9	56.1	56.7	57.0
	ProNE	48.8	**56.2**	**58.0**	**58.8**	**59.2**
	($\pm\sigma$)	($\pm$1.0)	($\pm$0.5)	($\pm$0.2)	($\pm$0.2)	($\pm$0.1)
Youtube	DeepWalk	38.0	40.1	41.3	42.1	42.8
	LINE	33.2	35.5	37.0	38.2	39.3
	ProNE (SMF)	36.5	40.2	41.2	41.7	42.1
	ProNE	**38.2**	**41.4**	**42.3**	**42.9**	**43.3**
	($\pm\sigma$)	($\pm$0.8)	($\pm$0.3)	($\pm$0.2)	($\pm$0.2)	($\pm$0.2)

Embedding enhancement

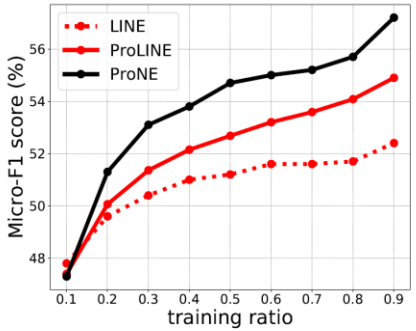




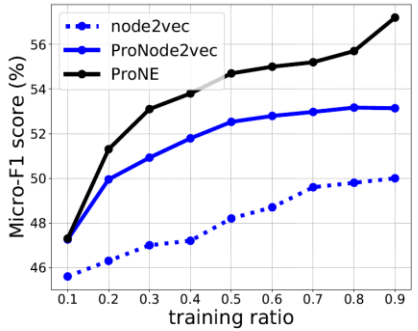
A general embedding enhancement framework



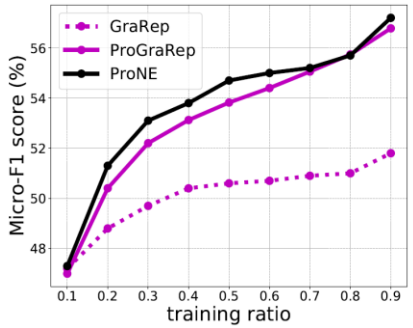
(a) ProDeepWalk



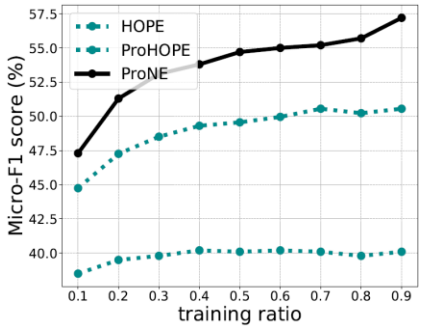
(b) ProLINE



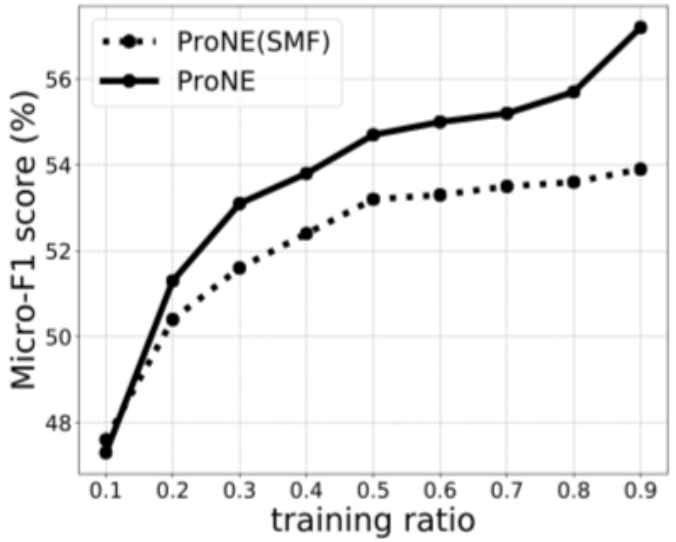
(c) ProNode2vec



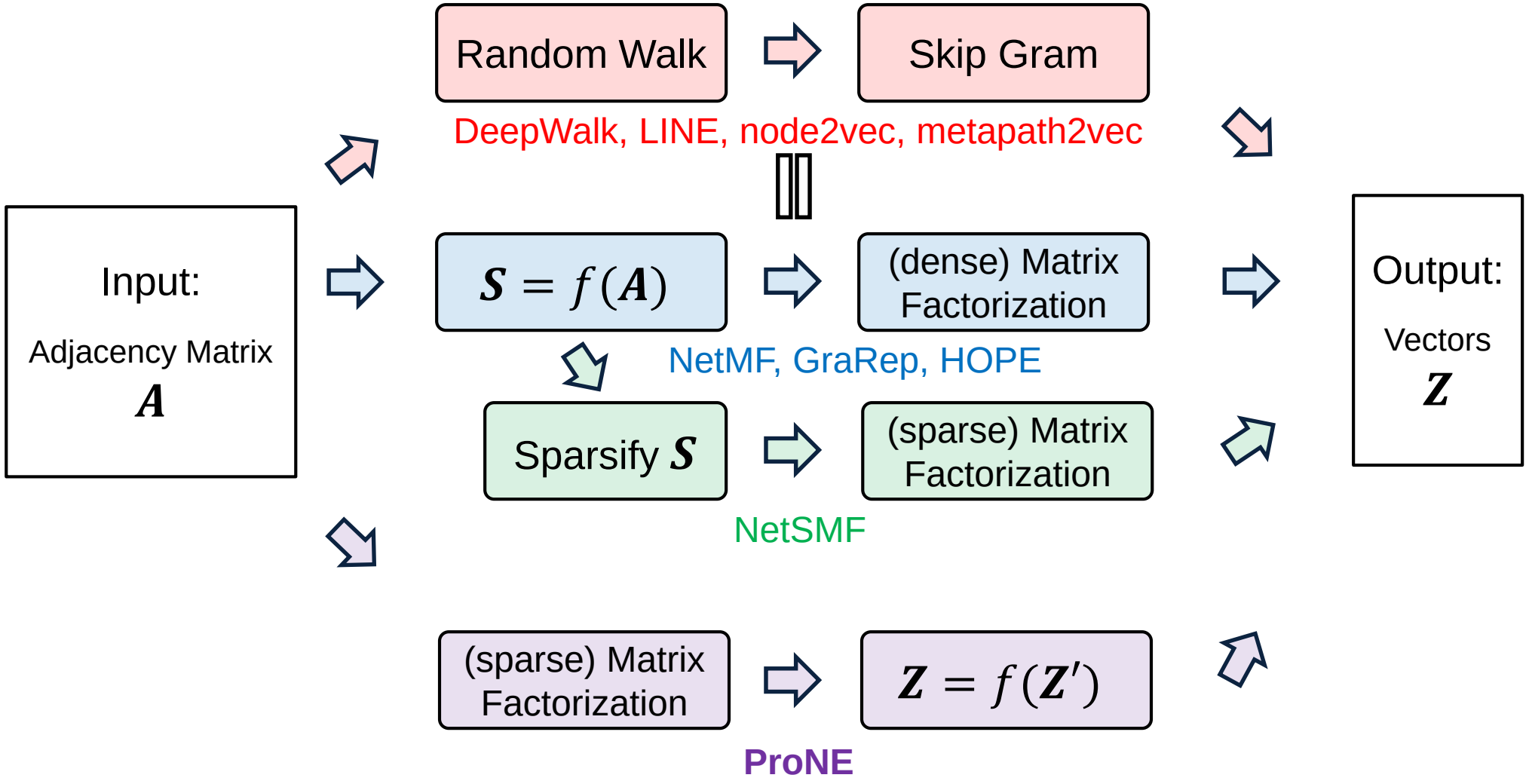
(d) ProGraRep



(e) ProHOPE



Network Embedding

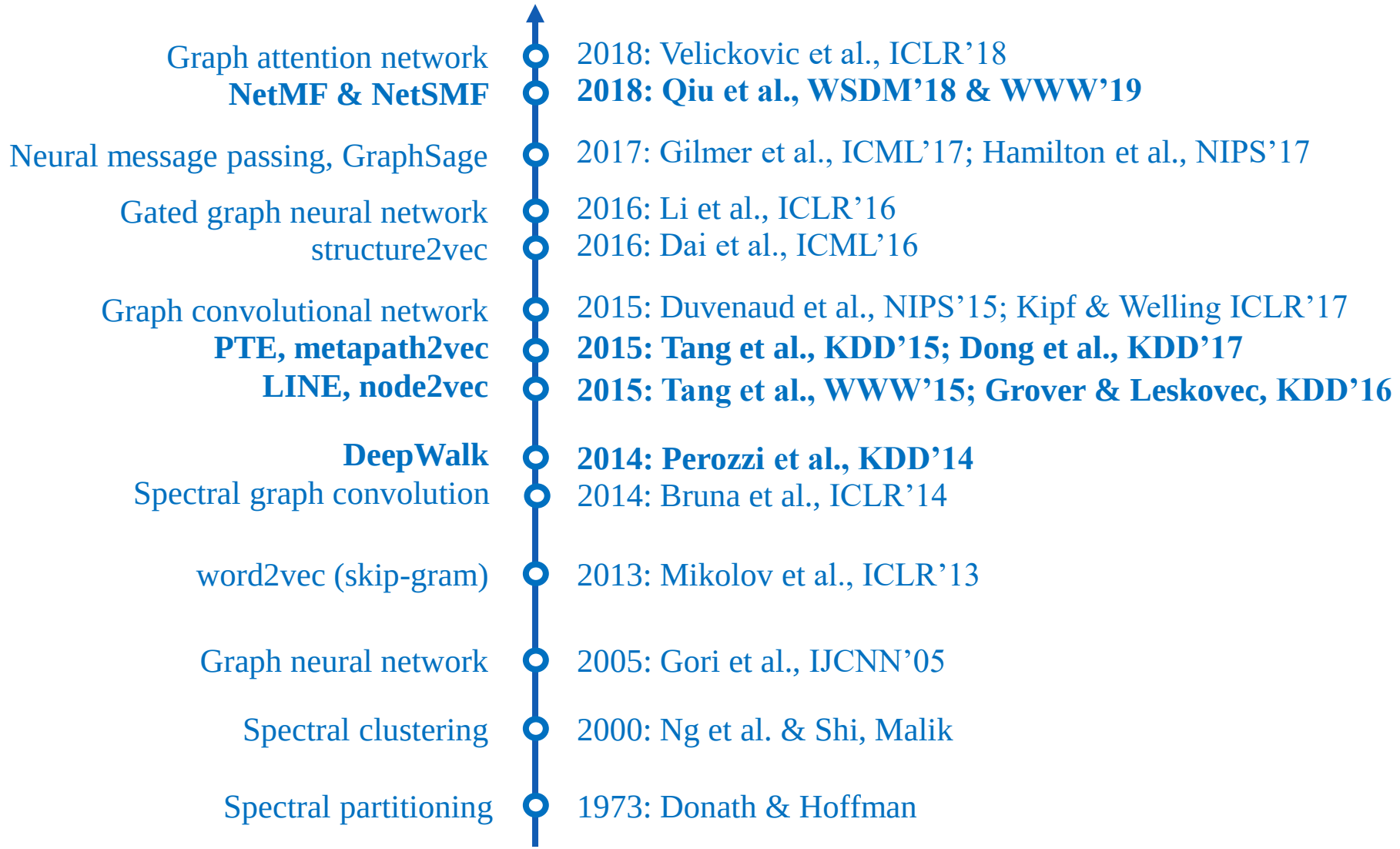


Different perspectives of network embedding

Type	Method	Decoder	Similarity measure	Loss function (ℓ)
Matrix factorization	Laplacian Eigenmaps [4]	$\|\mathbf{z}_i - \mathbf{z}_j\|_2^2$	general	$\text{DEC}(\mathbf{z}_i, \mathbf{z}_j) \cdot s_{\mathcal{G}}(v_i, v_j)$
	Graph Factorization [1]	$\mathbf{z}_i^\top \mathbf{z}_j$	$\mathbf{A}_{i,j}$	$\|\text{DEC}(\mathbf{z}_i, \mathbf{z}_j) - s_{\mathcal{G}}(v_i, v_j)\|_2^2$
	GraRep [9]	$\mathbf{z}_i^\top \mathbf{z}_j$	$\mathbf{A}_{i,j}, \mathbf{A}_{i,j}^2, ..., \mathbf{A}_{i,j}^k$	$\|\text{DEC}(\mathbf{z}_i, \mathbf{z}_j) - s_{\mathcal{G}}(v_i, v_j)\|_2^2$
	HOPE [45]	$\mathbf{z}_i^\top \mathbf{z}_j$	general	$\|\text{DEC}(\mathbf{z}_i, \mathbf{z}_j) - s_{\mathcal{G}}(v_i, v_j)\|_2^2$
Random walk	DeepWalk [47]	$\frac{e^{\mathbf{z}_i^\top \mathbf{z}_j}}{\sum_{k \in \mathcal{V}} e^{\mathbf{z}_i^\top \mathbf{z}_k}}$	$p_{\mathcal{G}}(v_j \mid v_i)$	$-s_{\mathcal{G}}(v_i, v_j) \log(\text{DEC}(\mathbf{z}_i, \mathbf{z}_j))$
	node2vec [28]	$\frac{e^{\mathbf{z}_i^\top \mathbf{z}_j}}{\sum_{k \in \mathcal{V}} e^{\mathbf{z}_i^\top \mathbf{z}_k}}$	$p_{\mathcal{G}}(v_j \mid v_i)$ (biased)	$-s_{\mathcal{G}}(v_i, v_j) \log(\text{DEC}(\mathbf{z}_i, \mathbf{z}_j))$

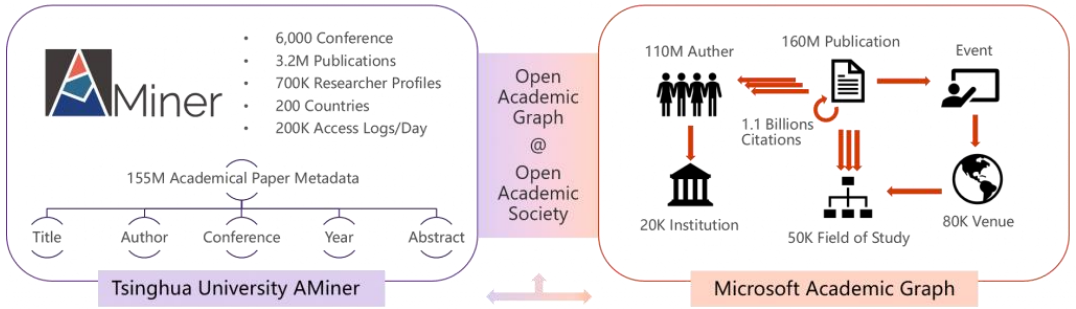
Hamilton, Ying, Leskovec. Representation learning on graphs: methods and applications. IEEE Data Engineering Bulletin. 2017

Network Representation Learning / Network Embedding



OAG: Open Academic Graph

https://www.openacademic.ai/oag/



Data set	#Pairs/Venues	Date
Linking relations	29,841	2018.12
AMiner venues	69,397	2018.07
MAG venues	52,678	2018.11

Table 1: statistics of OAG venue data

Data set	#Pairs/Papers	Date
Linking relations	91,137,597	2018.12
AMiner papers	172,209,563	2019.01
MAG papers	208,915,369	2018.11

Table 2: statistics of OAG paper data

Data set	#Pairs/Authors	Date
Linking relations	1,717,680	2019.01
AMiner authors	113,171,945	2018.07
MAG authors	253,144,301	2018.11

Table 3: statistics of OAG author data

Open Academic Graph

Open Academic Graph (OAG) is a large knowledge graph unifying two billion-scale academic graphs: Microsoft Academic Graph (MAG) and AMiner. In mid 2017, we published OAG v1, which contains 166,192,182 papers from MAG and 154,771,162 papers from AMiner (see below) and generated 64,639,608 linking (matching) relations between the two graphs. This time, in OAG v2, author, venue and newer publication data and the corresponding matchings are available.

Overview of OAG v2

The statistics of OAG v2 is listed as the three tables below. The two large graphs are both evolving and we take MAG November 2018 snapshot and AMiner July 2018 or January 2019 snapshot for this version.

Thank you !

Collaborators: John Hopcroft, Jon Kleinberg, Chenhao Tan (**Cornell**)

Jiawei Han (**UIUC**), Philip Yu (**UIC**)

Jian Pei (**SFU**), Hanghang Tong (**ASU**)

Tiancheng Lou (**Google&Baidu**), Jimeng Sun (**GIT**)

Wei Chen, Ming Zhou, Long Jiang, Chi Wang, Kuansan Wang (**Microsoft**)

Hongxia, Jingren Zhou, Chang Zhou (**Alibaba**)

Jiezhong Qiu, Jie Zhang, Fanjin Zhang, Qibin Chen, Yukuo Cen, et al. (**THU**)

Jie Tang, KEG, Tsinghua U,
Download all data & Codes,

http://keg.cs.tsinghua.edu.cn/jietang
http://arnetminer.org/data
http://arnetminer.org/data-sna