Network Embedding

Yuanfu Lu

BUPT

June 17, 2019

Outline

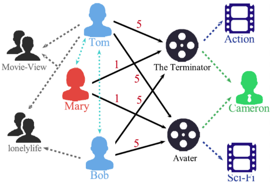
- Problem
- Methodology
- Application
- Conclusion

Outline

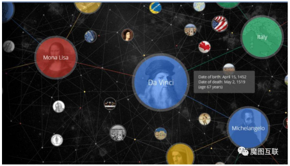
- Problem
- Methodology
- Application
- Conclusion

Problem

(a) Social Network



(b) Movie Network

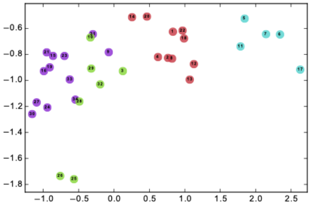


(c) Knowledge Graph

- Homogeneous Network (a)
  - 1 type of node and edge
- Heterogeneous Network (b)
  - $\geq$ 2 types of node and edge
- Knowledge Graph (c)
  - triad (h, r, t)

Problem

(a) Input: Karate Graph



(b) Output: Representation

- Input: a network/graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$
- Output: the representation of the network $\mathbf{U} \in \mathbb{R}^{n \times d}$, $d \ll |\mathcal{V}|$, d-dim vector $\mathbf{u}_i$ for each node v_i.

Goal: learn a mapping function $f : v_i \to \mathbf{u}_i \in \mathbb{R}^d$

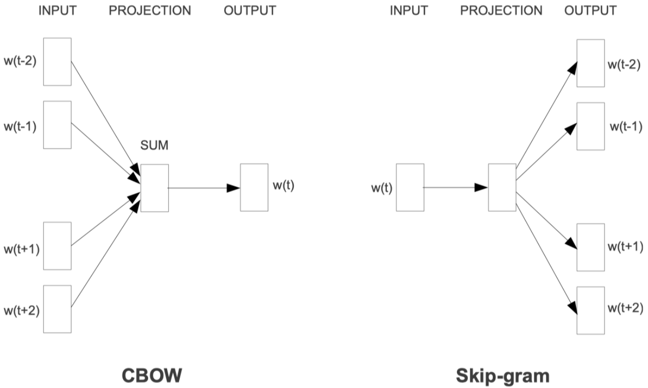
Outline

- Problem
- Methodology
- Application
- Conclusion

Methodology

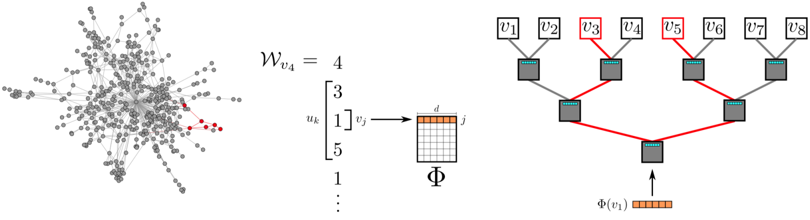
HAN	2019: Wang et al. WWW
RHINE	2019: Lu et al. AAAI
GAT	2018: Velickovic et al. ICLR
DynamicTriad, HTNE	2018: Lekui et al. AAAI, Yuan et al. KDD
PME, HERec	2018: Chen et al. KDD, Shi et al. TKDE
DHPE	2018: Dingyuan Zhu et al. TKDE
GCN, GraphSage	2017: Kipf et al. ICLR, Jure et al. NIPS
EOE, HIN2Vec	2017: Xu et al. WSDM, Fu et al. CIKM
mtapath2vec	2017: Dong et al. KDD
Esim	2016: Shang et al. arXiv
SDNE, HOPE	2016: Cui et al. KDD
node2vec	2016: Leskovec et al. KDD
GraRep	2015: Cao et al. CIKM
LINE, PTE	2015: Tang et al. WWW, KDD
DeepWalk	2014: Perozzi et al. KDD
Word2vec	2013: Mikolov et al. ICLR

Method: word2vec(ICLR'13)



The CBOW architecture predicts the current word based on the context, and the Skip-gram predicts surrounding words given the current word.

Method: DeepWalk (KDD'14)

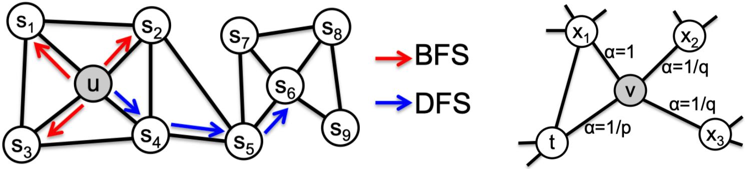


(a) Random walk generation. (b) Representation mapping. (c) Hierarchical Softmax.

$$\Pr\left(\{v_{i-w}, \cdots, v_{i+w}\} \setminus v_i | \Phi(v_i)\right) = \prod_{\substack{j=i-w \\ j \neq i}}^{i+w} \Pr\left(v_j | \Phi(v_i)\right)$$

Maximize the cooccurrence probability among the nodes that appear within a window w, in a random walk.

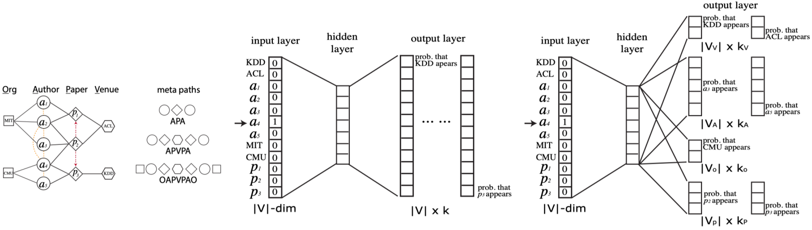
Method: node2vec (KDD'16)



$$\max_f \sum_{u \in V} \left(-\log \sum_{v \in V} \exp(f(u) \cdot f(v)) + \sum_{n_i \in N_S(u)} f(n_i) \cdot f(u) \right)$$

- **BFS**. Immediate neighbors of the source node.
- **DFS**. Increasing distances from the source node.

Method: metapath2vec (KDD'17)



(a) An academic network (b) Skip-gram in *metapath2vec*, node2vec, & DeepWalk (c) Skip-gram in *metapath2vec++*

- meta-path based random walk

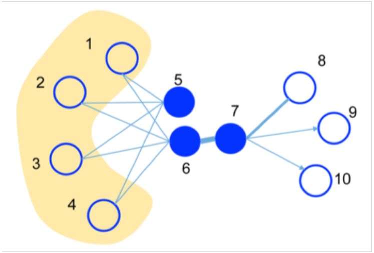
$$p\left(v^{i+1} | v_t^i, \mathcal{P}\right) = \begin{cases} \frac{1}{\left|N_{t+1}\left(v_t^i\right)\right|} & \left(v^{i+1}, v_t^i\right) \in E, \phi\left(v^{i+1}\right) = t+1 \\ 0 & \left(v^{i+1}, v_t^i\right) \in E, \phi\left(v^{i+1}\right) \neq t+1 \\ 0 & \left(v^{i+1}, v_t^t\right) \notin E \end{cases}$$

- (heterogeneous) negative sample

$$p\left(c_t | v; \theta\right) = \frac{e^{X_{c_t} \cdot X_v}}{\sum_{u \in V} e^{X_u \cdot X_v}}, \quad p\left(c_t | v; \theta\right) = \frac{e^{X_{c_t} \cdot X_v}}{\sum_{u_t \in V_t} e^{X_{u_t} \cdot X_v}}$$

$$\log \sigma\left(X_{c_t} \cdot X_v\right) + \sum_{m=1}^{M} \mathbb{E}_{u^m \sim P(u)}\left[\log \sigma\left(-X_u^m \cdot X_v\right)\right]$$

Method: LINE (WWW'15)



First-order Proximity

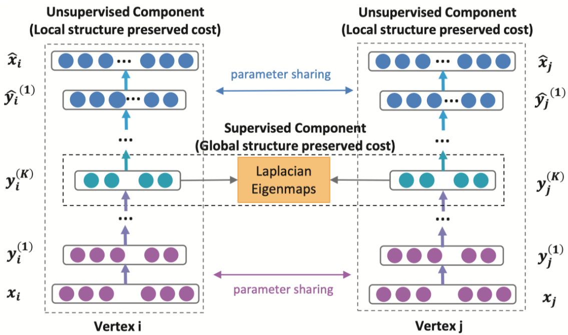
$$p_1(v_i, v_j) = \frac{1}{1 + \exp(-\vec{u}_i^T \cdot \vec{u}_j)}$$

Second-order Proximity

$$p_2(v_j|v_i) = \frac{\exp(\vec{u}_j'^T \cdot \vec{u}_i)}{\sum_{k=1}^{|V|} \exp(\vec{u}_k'^T \cdot \vec{u}_i)}$$

$$\log \sigma({\vec{u}'_j}^T \cdot \vec{u}_i) + \sum_{i=1}^{K} E_{v_n \sim P_n(v)}[\log \sigma(-{\vec{u}'_n}^T \cdot \vec{u}_i)]$$

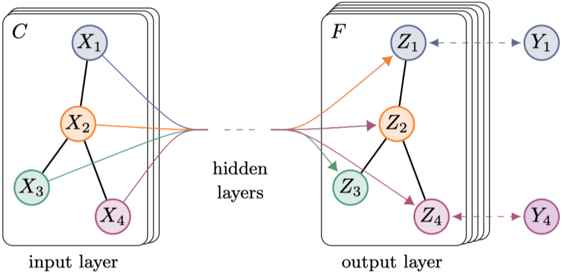
Method: SDNE (KDD'16)



$$\mathcal{L}_{1st} = \sum_{i,j=1}^{n} s_{i,j} \left\| \mathbf{y}_i^{(K)} - \mathbf{y}_j^{(K)} \right\|_2^2 = \sum_{i,j=1}^{n} s_{i,j} \left\| \mathbf{y}_i - \mathbf{y}_j \right\|_2^2$$

$$\mathcal{L}_{2nd} = \sum_{i=1}^{n} \left\| (\hat{\mathbf{x}}_i - \mathbf{x}_i) \odot \mathbf{b_i} \right\|_2^2 = \left\| (\hat{X} - X) \odot B \right\|_F^2$$

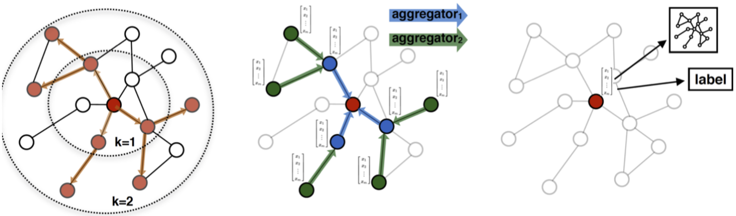
Method: GCN (ICLR'17)



$$Z = f(X, A) = \text{softmax}\left(\hat{A}\,\text{ReLU}\left(\hat{A}XW^{(0)}\right)W^{(1)}\right)$$

$$\mathcal{L} = -\sum_{l \in \mathcal{Y}_L} \sum_{f=1}^{F} Y_{lf} \ln Z_{lf}$$

Method: GraphSage (NIPS'17)



1. Sample neighborhood
2. Aggregate feature information from neighbors
3. Predict graph context and label using aggregated information

- Mean aggregator

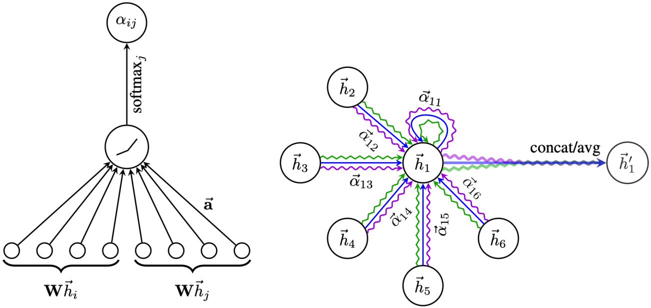
$$\mathbf{h}_v^k \leftarrow \sigma\left(\mathbf{W} \cdot \mathrm{MEAN}\left(\left\{\mathbf{h}_v^{k-1}\right\} \cup \left\{\mathbf{h}_u^{k-1}, \forall u \in \mathcal{N}(v)\right\}\right)\right)$$

- LSTM aggregator
- Pooling aggregator

$$\mathbf{h}_v^k = \max\left(\left\{\sigma\left(\mathbf{W}_{\mathrm{pool}}\mathbf{h}_{u_i}^k + \mathbf{b}\right), \forall u_i \in \mathcal{N}(v)\right\}\right)$$

$$J_{\mathcal{G}}(\mathbf{z}_u) = -\log\left(\sigma\left(\mathbf{z}_u^\top \mathbf{z}_v\right)\right) - Q \cdot \mathbb{E}_{v_n \sim P_n(v)} \log\left(\sigma\left(-\mathbf{z}_u^\top \mathbf{z}_{v_n}\right)\right)$$

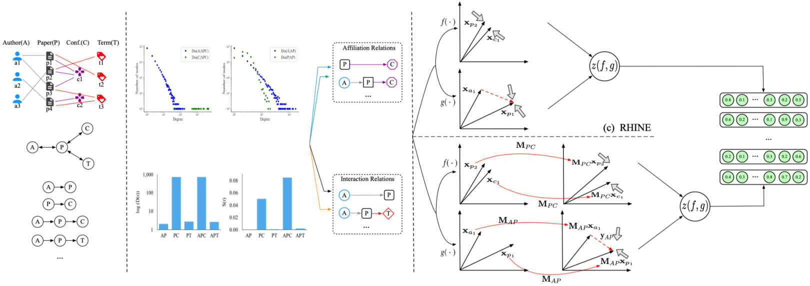
Method: GAT (ICLR'18)



$$\alpha_{ij} = \frac{\exp\left(\mathrm{LeakyReLU}\left(\vec{\mathbf{a}}^T\left[\mathbf{W}\vec{h}_i \| \mathbf{W}\vec{h}_j\right]\right)\right)}{\sum_{k\in\mathcal{N}_i}\exp\left(\mathrm{LeakyReLU}\left(\vec{\mathbf{a}}^T\left[\mathbf{W}\vec{h}_i \| \mathbf{W}\vec{h}_k\right]\right)\right)}, \vec{h}'_i = \sigma\left(\sum_{j\in\mathcal{N}_i}\alpha_{ij}\mathbf{W}\vec{h}_j\right)$$

$$\vec{h}'_i = \sigma\left(\frac{1}{K}\sum_{k=1}^{K}\sum_{j\in\mathcal{N}_i}\vec{\alpha}_{ij}^{k}\mathbf{W}^k\vec{h}_j\right)$$

Method: RHINE (AAAI'19)

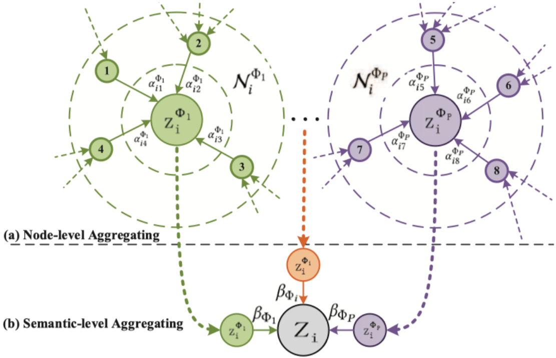


(a) An example of HIN (b) Structural characteristics analysis of relations in an HIN (d) RHINE-M (e) Node embeddings

$$f(p,q) = w_{pq} \left\| \mathbf{x}_p - \mathbf{x}_q \right\|_2^2$$
$$g(u,v) = w_{uv} \left\| \mathbf{x}_u + \mathbf{y}_r - \mathbf{x}_v \right\|$$

$$L = L_{EuAR} + L_{TrIR}$$
$$= \sum_{s \in R_{AR}\langle p,s,q\rangle \in P_{AR}\langle p',s,q'\rangle \in P'_{AR}} \max\left[0, \gamma + f(p,q) - f(p',q')\right]$$
$$+ \sum_{r \in R_{IR}\langle u,r,v\rangle \in P_{IR}\langle u',r,v'\rangle \in P'_{IR}} \max\left[0, \gamma + g(u,v) - g(u',v')\right]$$

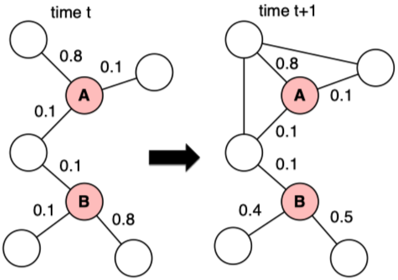
Method: HAN (WWW'19)



$$\mathbf{h}_i' = \mathbf{M}_{\phi_i} \cdot \mathbf{h}_i, \quad \alpha_{ij}^{\Phi} = \frac{\exp\left(\sigma\left(\mathbf{a}_{\Phi}^{\mathrm{T}} \cdot \left[\mathbf{h}_i' \| \mathbf{h}_j'\right]\right)\right)}{\sum_{k \in \mathcal{N}_i^{\Phi}} \exp\left(\sigma\left(\mathbf{a}_{\Phi}^{\mathrm{T}} \cdot \left[\mathbf{h}_i' \| \mathbf{h}_k'\right]\right)\right)}, \quad \mathbf{z}_i^{\Phi} = \sigma\left(\sum_{j \in \mathcal{N}_i^{\Phi}} \alpha_{ij}^{\Phi} \cdot \mathbf{h}_j'\right)$$

$$w_{\Phi_i} = \frac{1}{|\mathcal{V}|} \sum_{i \in \mathcal{V}} \mathbf{q}^{\mathrm{T}} \cdot \tanh\left(\mathbf{W} \cdot \mathbf{z}_i^{\Phi} + \mathbf{b}\right), \quad \beta_{\Phi_i} = \frac{\exp\left(w_{\Phi_i}\right)}{\sum_{i=1}^{P} \exp\left(w_{\Phi_i}\right)}, \quad \mathbf{Z} = \sum_{i=1}^{P} \beta_{\Phi_i} \cdot \mathbf{Z}_{\Phi_i}$$

Method: DynamicTrias (AAAI'18)



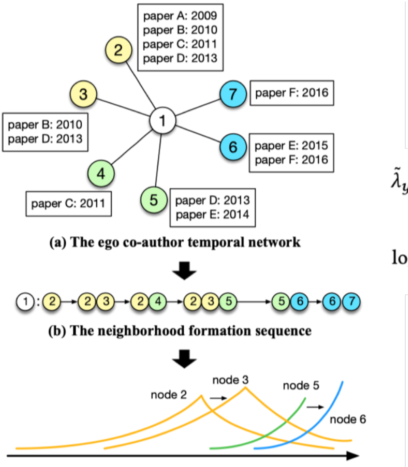
- Triadic closure process

$$P_{\mathrm{tr}}^t(i,j,k) = \frac{1}{1+\exp\left(-\left\langle \boldsymbol{\theta}, \boldsymbol{x}_{ijk}^t \right\rangle\right)}$$

- Social homophily and temporal smoothness

$$g^t(j,k) = \left\| \boldsymbol{u}_j^t - \boldsymbol{u}_k^t \right\|_2^2, \quad L_{\mathrm{smooth}}^t = \begin{cases} \sum_{i=1}^N \left\| \boldsymbol{u}_i^t - \boldsymbol{u}_i^{t-1} \right\|_2^2 & t > 1 \\ 0 & t = 1 \end{cases}$$

Method: HTNE (KDD'18)

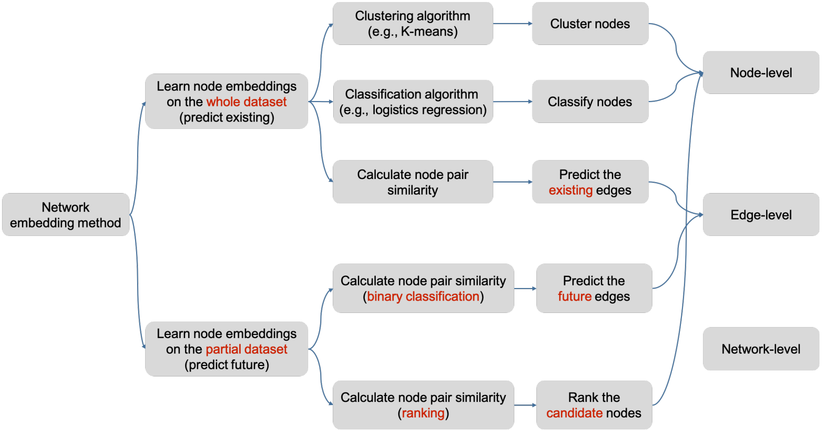


(c) The arrival rate of several target neighbours in the sequence

$$\tilde{\lambda}_{y|x}(t) = \mu_{x,y} + \sum_{t_h<t} \alpha_{h,y}\kappa(t - t_h),$$

$$\log \sigma(\tilde{\lambda}_{y|x}(t)) + \sum_{k=1}^{K} \mathbb{E}_{v^k \sim P_n(v)}[-\log \sigma(\tilde{\lambda}_{v^k|x}(t))],$$

Application



Application: Node Clustering

- Setting

Learn node embeddings → Clustering algorithm (e.g., K-means) → Cluster nodes

- Evaluation

$$NMI\,(C, C') = \frac{MI(C,C')}{\max(H(C),H(C'))}$$

$H(C)$ is the entropy of C, and $MI(C, C')$ is the mutual information metric of C and C' .

Methods	DBLP	Yelp	AMiner
DeepWalk	0.3884	0.3043	0.5427
LINE-1st	0.2775	0.3103	0.3736
LINE-2nd	0.4675	0.3593	0.3862
PTE	0.3101	0.3527	0.4089
ESim	0.3449	0.2214	0.3409
HIN2Vec	0.4256	0.3657	0.3948
metapath2vec	0.6065	0.3507	0.5586
HERec	0.5893	0.3313	0.5123
RHINE	0.7204	0.3882	0.6024
RHINE-M	**0.7323**	**0.3934**	**0.6152**

Yuanfu Lu et al. Relation Structure-Aware Heterogeneous Information Network Embedding. AAAI2019.

Application: Node Classification

- Setting

Learn node embeddings → Classification algorithm (e.g., logistics regression) → Classify nodes

- Evaluation

Datasets	Metrics	Training	DeepWalk	ESim	metapath2vec	HERec	GCN	GAT	HAN_{nd}	HAN_{sem}	HAN
ACM	Macro-F1	20%	77.25	77.32	65.09	66.17	86.81	86.23	88.15	89.04	**89.40**
		40%	80.47	80.12	69.93	70.89	87.68	87.04	88.41	89.41	**89.79**
		60%	82.55	82.44	71.47	72.38	88.10	87.56	87.91	**90.00**	89.51
		80%	84.17	83.00	73.81	73.92	88.29	87.33	88.48	90.17	**90.63**
	Micro-F1	20%	76.92	76.89	65.00	66.03	86.77	86.01	87.99	88.85	**89.22**
		40%	79.99	79.70	69.75	70.73	87.64	86.79	88.31	89.27	**89.64**
		60%	82.11	82.02	71.29	72.24	88.12	87.40	87.68	**89.85**	89.33
		80%	83.88	82.89	73.69	73.84	88.35	87.11	88.26	89.95	**90.54**
DBLP	Macro-F1	20%	77.43	91.64	90.16	91.68	90.79	90.97	91.17	92.03	**92.24**
		40%	81.02	92.04	90.82	92.16	91.48	91.20	91.46	92.08	**92.40**
		60%	83.67	92.44	91.32	92.80	91.89	90.80	91.78	92.38	**92.80**
		80%	84.81	92.53	91.89	92.34	92.38	91.73	91.80	92.53	**93.08**
	Micro-F1	20%	79.37	92.73	91.53	92.69	91.71	91.96	92.05	92.99	**93.11**
		40%	82.73	93.07	92.03	93.18	92.31	92.16	92.38	93.00	**93.30**
		60%	85.27	93.39	92.48	93.70	92.62	91.84	92.69	93.31	**93.70**
		80%	86.26	93.44	92.80	93.27	93.09	92.55	92.69	93.29	**93.99**
IMDB	Macro-F1	20%	40.72	32.10	41.16	41.65	45.73	49.44	49.78	**50.87**	50.00
		40%	45.19	31.94	44.22	43.86	48.01	50.64	52.11	50.85	**52.71**
		60%	48.13	31.68	45.11	46.27	49.15	51.90	51.73	52.09	**54.24**
		80%	50.35	32.06	45.15	47.64	51.81	52.99	52.66	51.60	**54.38**
	Micro-F1	20%	46.38	35.28	45.65	45.81	49.78	55.28	54.17	55.01	**55.73**
		40%	49.99	35.47	48.24	47.59	51.71	55.91	56.39	55.15	**57.97**
		60%	52.21	35.64	49.09	49.88	52.29	56.44	56.09	56.66	**58.32**
		80%	54.33	35.59	48.81	50.99	54.61	56.97	56.38	56.49	**58.51**

Xiao Wang et al. Heterogeneous Graph Attention Network. WWW2019.

Application: Network Reconstruction

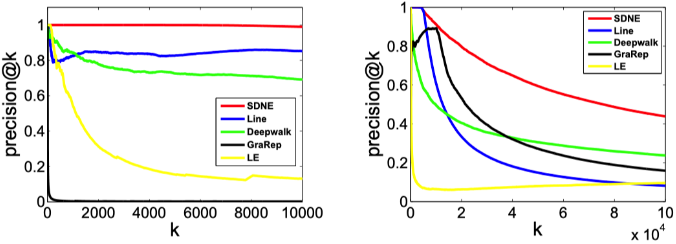
- Setting

Learn node embeddings → Calculate node pair similarity → Predict the existing edges

- Evaluation

$$\text{Precision@k} = \frac{|\{(i,j)|(i,j) \in \mathbf{E}_p \cap \mathbf{E}_o\}|}{|\mathbf{E}_p|}$$

$\mathbf{E}_p$ is the set of top-k predicted links, $\mathbf{E}_o$ is the set of observed links.



(a) ARXIV GR-QC (b) BLOGCATALOG

Daxin Wang et al. Structural Deep Network Embedding. KDD2016.

Application: Link Prediction

- Setting

Learn node embeddings → Calculate node pair similarity (binary classification) → Predict the future edges

- Evaluation

	Amazon			YouTube			Twitter			Alibaba-S		
	ROC-AUC	PR-AUC	F1	ROC-AUC	PR-AUC	F1	ROC-AUC	PR-AUC	F1	ROC-AUC	PR-AUC	F1
DeepWalk	94.20	94.03	87.38	71.11	70.04	65.52	69.42	72.58	62.68	59.39	60.62	56.10
node2vec	94.47	94.30	87.88	71.21	70.32	65.36	69.90	73.04	63.12	62.26	63.40	58.49
LINE	81.45	74.97	76.35	64.24	63.25	62.35	62.29	60.88	58.18	53.97	54.65	52.85
metapath2vec	94.15	94.01	87.48	70.98	70.02	65.34	69.35	72.61	62.70	60.94	61.40	58.25
ANRL	71.68	70.30	67.72	75.93	73.21	70.65	70.04	67.16	64.69	58.17	55.94	56.22
PMNE(n)	95.59	95.48	89.37	65.06	63.59	60.85	69.48	72.66	62.88	62.23	63.35	58.74
PMNE(r)	88.38	88.56	79.67	70.61	69.82	65.39	62.91	67.85	56.13	55.29	57.49	53.65
PMNE(c)	93.55	93.46	86.42	68.63	68.22	63.54	67.04	70.23	60.84	51.57	51.78	51.44
MVE	92.98	93.05	87.80	70.39	70.10	65.10	72.62	73.47	67.04	60.24	60.51	57.08
MNE	90.28	91.74	83.25	82.30	82.18	75.03	91.37	91.65	84.32	62.79	63.82	58.74
GATNE-T	**97.44**	**97.05**	**92.87**	**84.61**	81.93	**76.83**	**92.30**	91.77	**84.96**	66.71	67.55	62.48
GATNE-I	96.25	94.77	91.36	84.47	**82.32**	**76.83**	92.04	**91.95**	84.38	**70.87**	**71.65**	**65.54**

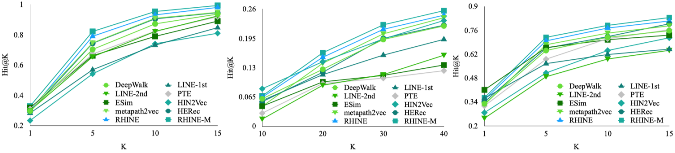
Yukuo Cen et al. Representation Learning for Attributed Multiplex Heterogeneous Network. KDD2019.

Application: Recommendation

- Setting

Learn node embeddings → Calculate node pair similarity (ranking) → Rank the candidate nodes

- Evaluation



(a) DBLP (b) Yelp (c) Aminer

Method	SN-TWebio			SN-Twitter		
	MAP@10	MAP@50	MAP@100	MAP@10	MAP@50	MAP@100
HOPE	**0.2295**	**0.1869**	**0.169**	**0.1000**	**0.0881**	**0.0766**
PPE	0.0928	0.0845	0.077	0.0061	0.0077	0.0081
LINE1	0	0	0.005	0.0209	0.0221	0.0221
LINE2	0.051	0.051	0.048	0.0044	0.0043	0.0035
DeepWalk	0.0635	0.0583	0.004	0.0006	0.0008	0.001
Common Neighbors	0.1217	0.1031	0.155	0.0394	0.0379	0.0369
Adamic-Adar	0.1173	0.0990	0.156	0.0455	0.0442	0.0423

Mingdong Ou et al. Asymmetric Transitivity Preserving Graph Embedding. KDD2016.

Outline

- Problem
- Methodology
- Application
- Conclusion

Conclusion

- Existing methods summary
- Existing methods problem
- Future work

- Hongyun Cai et al. A Comprehensive Survey of Graph Embedding: Problems, Techniques and Applications. TKDE2017.
- Peng Cui et al. A Survey on Network Embedding. TKDE2018.
- Palash Goyal et al. Graph Embedding Techniques, Applications, and Performance: A Survey. arXiv2017.
- Mikolov et al. Efcient Estimation of Word Representations in Vector Space. ICLR2013.
- Bryan Perozzi et al. DeepWalk: Online Learning of Social Representations. KDD2014.
- Jian Tang et al. LINE: Large-scale Information Network Embedding. WWW2015.
- Jian Tang et al. PTE: Predictive Text Embedding through Large-scale Heterogeneous Text Networks. KDD2015.
- Shaosheng Cao et al. GraRep: Learning Graph Representations with Global Structural Information. CIKM2015.
- Aditya Grover et al. node2vec: Scalable Feature Learning for Networks. KDD2016.
- Mingdong Ou et al. Asymmetric Transitivity Preserving Graph Embedding. KDD2016.
- Daixin Wang et al. Structural Deep Network Embedding. KDD2016.
- Yuxiao Dong et al. metapath2vec: Scalable Representation Learning for Heterogeneous Networks. KDD2017.
- Linchuan Xu et al. Embedding of Embedding (EOE) : Joint Embedding for Coupled Heterogeneous Networks. WSDM2017.
- Tao-yang Fu et al. HIN2Vec: Explore Meta-paths in Heterogeneous Information Networks for Representation Learning. CIKM2017.
- Thomas N. Kipf et al. Semi-supervised Classification with Graph Convolution Networks. ICLR2017.
- William L. Hamilton et al. Inductive Representation Learning on Large Graphs. NIPS2017.
- Dingyuan Zhu et al. High-order Proximity Preserved Embedding For Dynamic Networks. TKDE2018.
- Hongxu Chen et al. PME: Projected Metric Embedding on Heterogeneous Networks for Link Prediction. KDD2018.
- Chuan Shi et al. Heterogeneous Information Network Embedding for Recommendation. TKDE2018.
- Lekui Zhou et al. Dynamic Network Embedding by Modeling Triadic Closure Process. AAAI2018.
- Yuan Zuo et al. Embedding Temporal Network via Neighborhood Formation. KDD2018.
- Petar Velickovic et al. Graph Attention Networks. ICLR2018.
- Yuanfu Lu et al. Relation Structure-Aware Heterogeneous Information Network Embedding. AAAI2019.
- Xiao Wang et al. Heterogeneous Graph Attention Network. WWW2019.

Thank You
Q&A