A Broader Picture of Random-walk Based Graph Embedding

KDD'21

Zexi Huang[1], Arlei Silva[2], Ambuj Singh[1]

[1]Department of Computer Science, University of California, Santa Barbara

[2]Department of Computer Science, Rice University

Research funded by NSF IIS 1817046 and HDTRA1-19-1-0017.

Most Popular

Downloaded | **Cited**

1. August 2016
XGBoost: A Scalable Tree Boosting System
KDD '16: Proceedings of the 22nd ACM SIGKDD International Conference on...
Tianqi Chen, Carlos Guestrin
Cited 4,504 times

2. August 2003
Maximizing the spread of influence through a social network
KDD '03: Proceedings of the ninth ACM SIGKDD international conference on...
David Kempe, Jon Kleinberg, Éva Tardos
Cited 3,740 times

3. August 2004
Mining and summarizing customer reviews
KDD '04: Proceedings of the tenth ACM SIGKDD international conference on...
Minqing Hu, Bing Liu
Cited 2,859 times

4. August 2014
DeepWalk: online learning of social representations
KDD '14: Proceedings of the 20th ACM SIGKDD international conference on...
Bryan Perozzi, Rami Al-Rfou, Steven Skiena
Cited 2,244 times

5. August 2016
node2vec: Scalable Feature Learning for Networks
KDD '16: Proceedings of the 22nd ACM SIGKDD International Conference on...
Aditya Grover, Jure Leskovec
Cited 2,155 times

6. July 2002
Optimizing search engines using clickthrough data
KDD '02: Proceedings of the eighth ACM SIGKDD international conference on...
Thorsten Joachims
Cited 2,021 times

7. August 2008
Factorization meets the neighborhood: a multifaceted collaborative filtering model
KDD '08: Proceedings of the 14th ACM SIGKDD international conference on...
Yehuda Koren
Cited 1,804 times

8. August 2016
"Why Should I Trust You?": Explaining the Predictions of Any Classifier
KDD '16: Proceedings of the 22nd ACM SIGKDD International Conference on...
Marco Tulio Ribeiro, Sameer Singh, + 1
Cited 1,698 times

Research papers

- [PARS14] DeepWalk: online learning of social representations
- [GL16] node2vec: scalable feature learning for networks
- ... (with many more items omitted)
- [CM20] InfiniteWalk: deep network embeddings as ...
- [ZLHK21] Node proximity is all you need: ...

Research papers

- [PARS14] DeepWalk: online learning of social representations
- [GL16] node2vec: scalable feature learning for networks
- ... (with many more items omitted)
- [CM20] InfiniteWalk: deep network embeddings as ...
- [ZLHK21] Node proximity is all you need: ...

Surveys

- [HYL17] Representation learning on graphs: methods and applications
- ...
- [CAEHP$^+$20] Machine learning on graphs: a model and comprehensive taxonomy

Research papers

- [PARS14] DeepWalk: online learning of social representations
- [GL16] node2vec: scalable feature learning for networks
- ... (with many more items omitted)
- [CM20] InfiniteWalk: deep network embeddings as ...
- [ZLHK21] Node proximity is all you need: ...

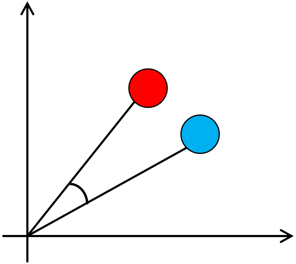
Surveys

- [HYL17] Representation learning on graphs: methods and applications
- ...
- [CAEHP+20] Machine learning on graphs: a model and comprehensive taxonomy

Difficult to compare existing methods and to design novel ones

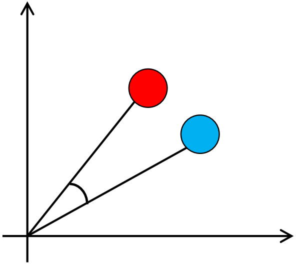
Question: link prediction

Question: link prediction

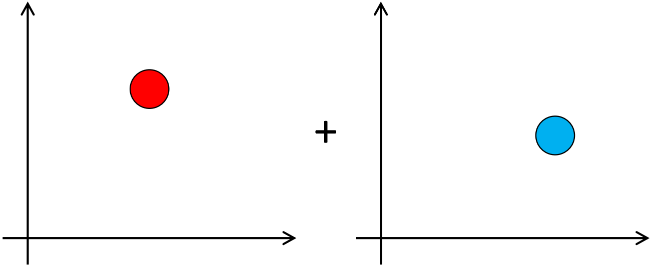


(a) Dot product
[OCP+16, WCZ16, ZCW+18]

Question: link prediction

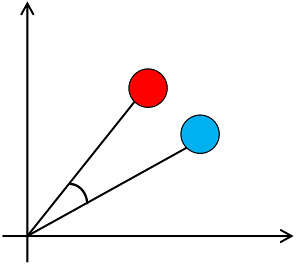


(a) Dot product
[OCP+16, WCZ16, ZCW+18]

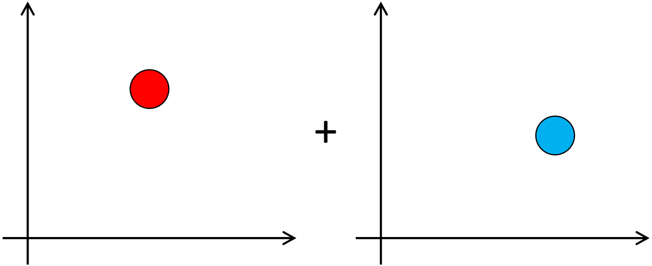


(b) Classification based on combined embeddings
[GL16, NLR+18, JDE+20]

Question: link prediction



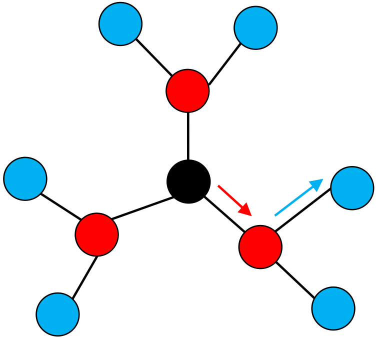
(a) Dot product
[OCP+16, WCZ16, ZCW+18]



(b) Classification based on combined embeddings
[GL16, NLR+18, JDE+20]

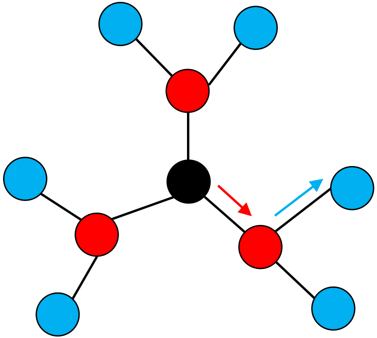
How should embeddings be used for link prediction?

Question: multiscale

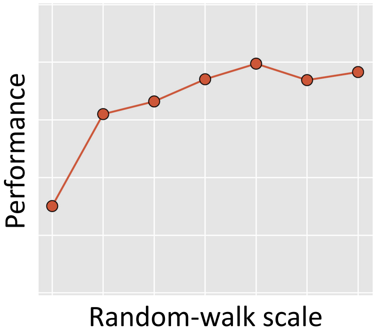


(a) Random-walks capture multiple structural scales [DYB10]

Question: multiscale

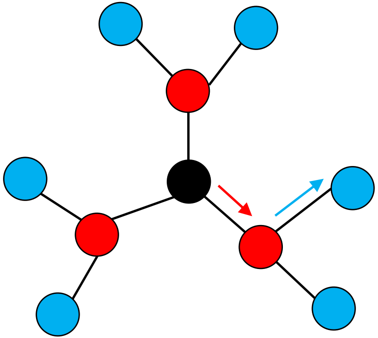


(a) Random-walks capture multiple structural scales [DYB10]

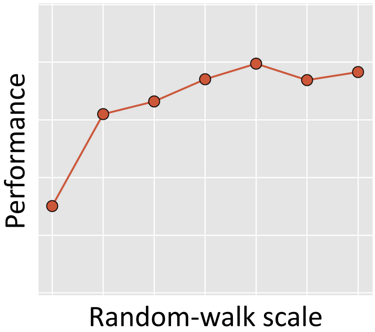


(b) Embedding performance [GL16] insensitve to random-walk scales

Question: multiscale



(a) Random-walks capture multiple structural scales [DYB10]



(b) Embedding performance [GL16] insensitve to random-walk scales

How do embeddings capture different structural scales?

Our analytical framework

random-walk process + similarity metric + embedding algorithm

Our analytical framework

	random-walk process	similarity metric	embedding algorithm
DeepWalk	standard	PMI	sampling

Our analytical framework

	random-walk process	similarity metric	embedding algorithm
DeepWalk	standard	PMI	sampling
node2vec	*biased*	PMI	sampling

Our analytical framework

	random-walk process	similarity metric	embedding algorithm
DeepWalk	standard	PMI	sampling
node2vec	*biased*	PMI	sampling
Multiscale	standard	*autocovariance*	*factorization*
...	...	...	...

Random-walk process

M: transition matrix

π, Π: stationary distribution

Random-walk process

M: transition matrix

π, Π: stationary distribution

Process

Random-walk based embedding methods

Standard

DeepWalk, LINE
Multiscale, NetMF
NetSMF, InfiniteWalk

Non-standard

node2vec, APP
NetMF, NERD

Similarity metric

PMI[1]: $R = \log(\Pi M^{\tau}) - \log(\pi\pi^{T})$

[1] [LG14] Levy and Goldberg. Neural word embedding as implicit matrix factorization. NeurIPS'14.

Similarity metric

PMI[1]: $R = \log(\Pi M^\tau) - \log(\pi\pi^T)$ Autocovariance[2]: $R = \Pi M^\tau - \pi\pi^T$

[1] [LG14] Levy and Goldberg. Neural word embedding as implicit matrix factorization. NeurIPS'14.
[2] [DYB10] Delvenne, Yalikari, and Barahona. Stability of graph communities across time scales. PNAS'10.

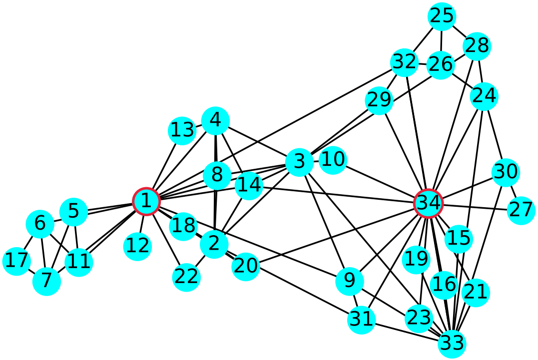
Similarity metric

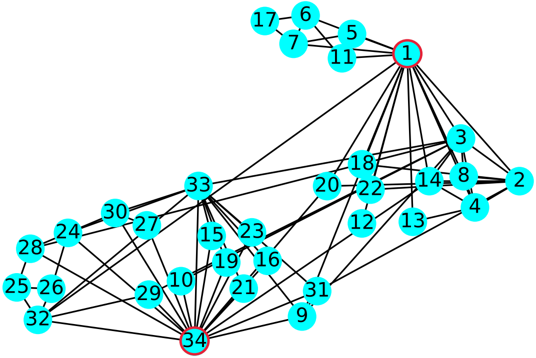
PMI[1]: $R = \log(\Pi M^\tau) - \log(\pi \pi^T)$ Autocovariance[2]: $R = \Pi M^\tau - \pi \pi^T$

[1][LG14] Levy and Goldberg. Neural word embedding as implicit matrix factorization. NeurIPS'14.
[2][DYB10] Delvenne, Yalikari, and Barahona. Stability of graph communities across time scales. PNAS'10.

Similarity metric

PMI[1]: $R = \log(\Pi M^{\tau}) - \log(\pi\pi^{T})$ Autocovariance[2]: $R = \Pi M^{\tau} - \pi\pi^{T}$

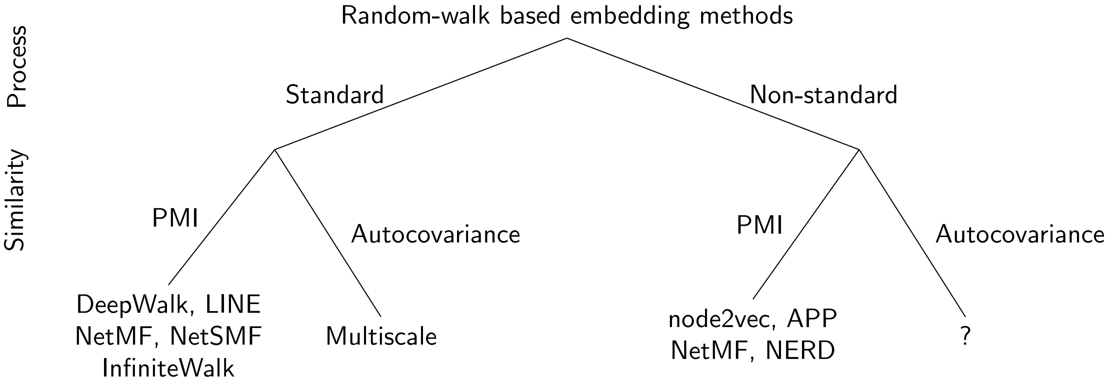




[1] [LG14] Levy and Goldberg. Neural word embedding as implicit matrix factorization. NeurIPS'14.
[2] [DYB10] Delvenne, Yalikari, and Barahona. Stability of graph communities across time scales. PNAS'10.

Similarity metric

PMI[1]: $R = \log(\Pi M^{\tau}) - \log(\pi \pi^{T})$ Autocovariance[2]: $R = \Pi M^{\tau} - \pi \pi^{T}$



[1][LG14] Levy and Goldberg. Neural word embedding as implicit matrix factorization. NeurIPS'14.
[2][DYB10] Delvenne et al. Stability of graph communities across time scales. PNAS'10.

Embedding algorithm

Explicit: Factorization (SVD)

$$\min \|UU^T - R\|_F^2$$

Implicit: Sampling (SGD)

$$\max \sum_{u,v} \log \Pr((u,v) \in \mathcal{D} | \mathbf{u}_u, \mathbf{v}_v)$$

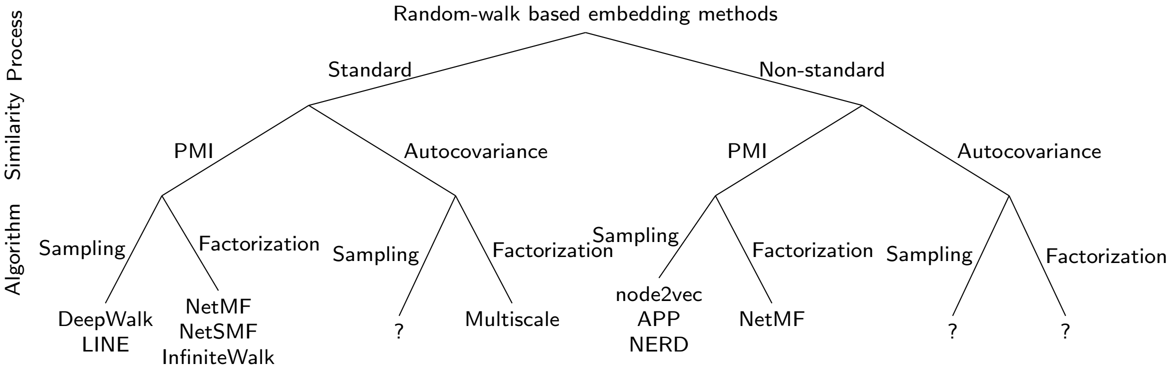
Embedding algorithm

Explicit: Factorization (SVD)

$$\min \|UU^T - R\|_F^2$$

Implicit: Sampling (SGD)

$$\max \textstyle\sum_{u,v} \log \Pr((u,v) \in \mathcal{D} | \mathbf{u}_u, \mathbf{v}_v)$$



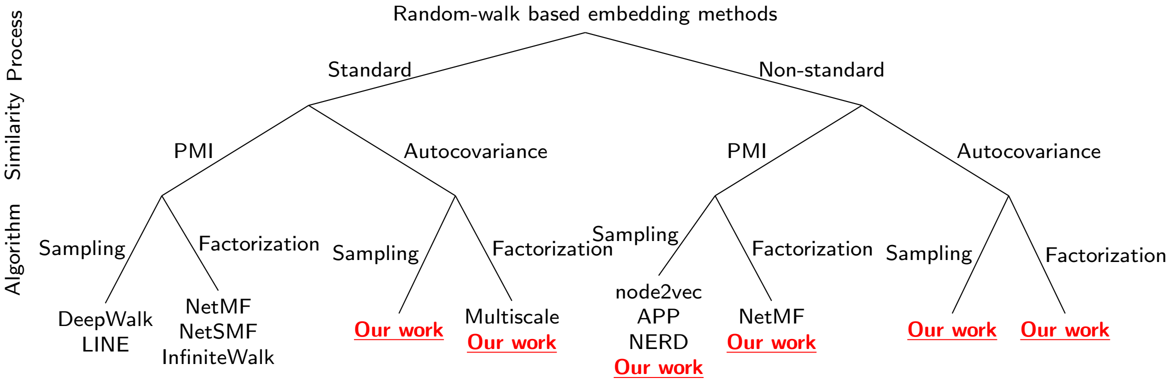
Embedding algorithm

Explicit: Factorization (SVD)

$$\min \|UU^T - R\|_F^2$$

Implicit: Sampling (SGD)

$$\max \sum_{u,v} \log \Pr((u,v) \in \mathcal{D} | \mathbf{u}_u, \mathbf{v}_v)$$



Datasets

	$\lvert\mathcal{V}\rvert$	$\lvert\mathcal{E}\rvert$	labels
BlogCatalog	10,312	333,983	interests
Airport	3,158	18,606	countries/continents
Wiki-words	4,777	92,157	tags
PoliticalBlogs	1,222	16,717	ideologies

Downstream tasks

- Node classification (Micro/Macro-F1)
- Link prediction (precision@k)
- Community detection (NMI)

Embedding dimension = 128

Comparing similarity metrics: node classification

PMI: $R = \log(\Pi M^{\tau}) - \log(\pi\pi^{T})$ Autocovariance: $R = \Pi M^{\tau} - \pi\pi^{T}$

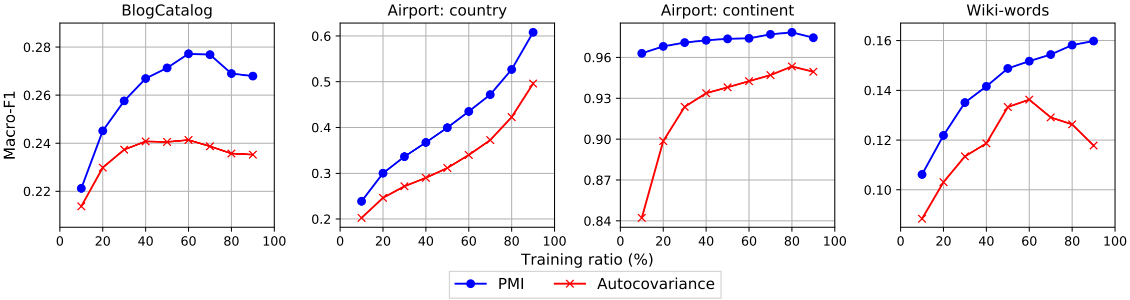


Figure: PMI consistently outperforms autocovariance in node classification.

Comparing similarity metrics: link prediction

PMI: $R = \log(\Pi M^\tau) - \log(\pi\pi^T)$ Autocovariance: $R = \Pi M^\tau - \pi\pi^T$

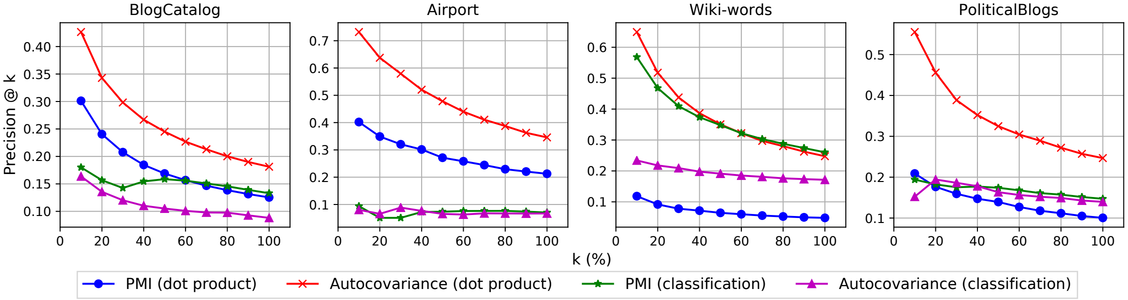


Figure: Autocovariance with dot product ranking consistently outperforms PMI (with either ranking scheme) in link prediction.

Understanding the difference

predicted degree $\propto$ embedding norm $\|\mathbf{u}\|$

Understanding the difference

$$\text{predicted degree} \propto \text{embedding norm } \|\mathbf{u}\| \propto \begin{cases} \text{actual degree} & \text{for autocov.} \\ \text{constant} & \text{for PMI} \end{cases}$$

Understanding the difference

$$\text{predicted degree} \propto \text{embedding norm } \|\mathbf{u}\| \propto \begin{cases} \text{actual degree} & \text{for autocov.} \\ \text{constant} & \text{for PMI} \end{cases}$$

Autocovariance captures heterogeneous degree distribution in graphs!

Understanding the difference

$$\text{predicted degree} \propto \text{embedding norm } \|\mathbf{u}\| \propto \begin{cases} \text{actual degree} & \text{for autocov.} \\ \text{constant} & \text{for PMI} \end{cases}$$

Autocovariance captures heterogeneous degree distribution in graphs!

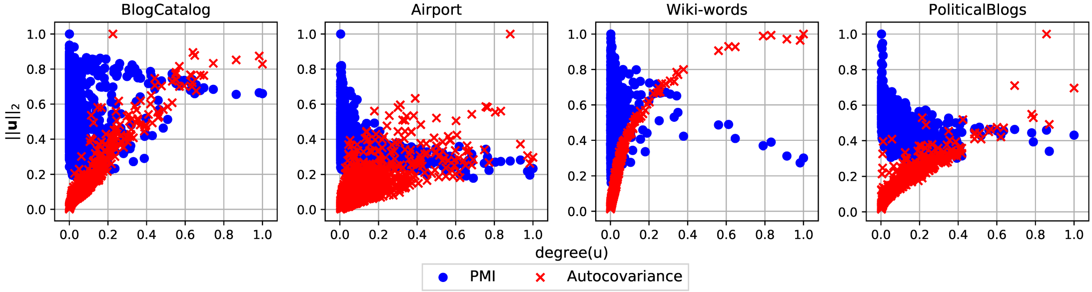


Figure: Autocovariance embedding norms correlate with actual degrees, but not PMI.

Understanding the difference

$$\text{predicted degree} \propto \text{embedding norm } \|\mathbf{u}\| \propto \begin{cases} \text{actual degree} & \text{for autocov.} \\ \text{constant} & \text{for PMI} \end{cases}$$

(a) PMI (b) Autocovariance

Figure: Autocovariance predicts more edges connecting to the hubs than PMI.

Multiscale

PMI: $R = \log(\Pi M^{\tau}) - \log(\pi\pi^{T})$ Autocovariance: $R = \Pi M^{\tau} - \pi\pi^{T}$

Multiscale

PMI: $\widetilde{R} = \log(\Pi \frac{1}{\tau} \sum_{t=1}^{\tau} M^t) - \log(\pi\pi^T)$

Autocovariance: $R = \Pi M^{\tau} - \pi\pi^T$

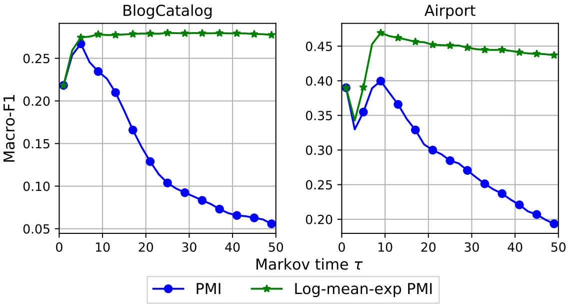


Figure: Node classification performance for PMI can be improved by smooth-averaging across multiple Markov times.

Multiscale

PMI: $\widetilde{R} = \log(\Pi \frac{1}{\tau} \sum_{t=1}^{\tau} M^t) - \log(\pi \pi^T)$

Autocovariance: $R = \Pi M^{\tau} - \pi \pi^T$

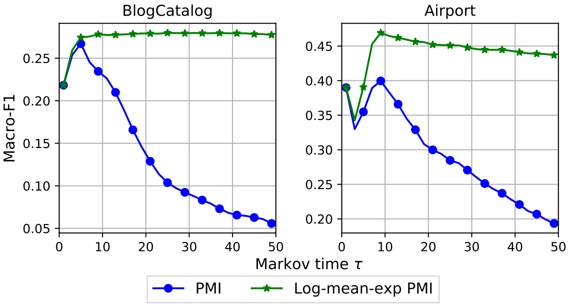


Figure: Node classification performance for PMI can be improved by smooth-averaging across multiple Markov times.

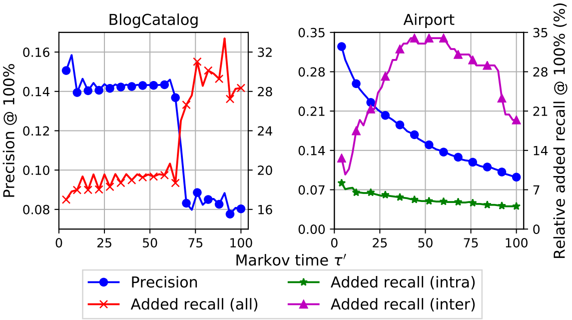


Figure: Prediction of edges of specific structural scales can be improved with different Markov times for autocovariance.

Problems:

1. How can one compare existing methods and to advance the SOTA?
2. How should embeddings be used for link prediction?
3. How do embeddings capture different structural scales?

Contributions:

1. A unified view of different processes, similarities, and algorithms.
2. Autocovariance embedding is significantly better for link prediction.
3. Ways to exploit multiscale similarity for optimized performance.

References I

[CAEHP+20] Ines Chami, Sami Abu-El-Haija, Bryan Perozzi, Christopher Ré, and Kevin Murphy.
Machine learning on graphs: A model and comprehensive taxonomy.
arXiv:2005.03675, 2020.

[CM20] Sudhanshu Chanpuriya and Cameron Musco.
Infinitewalk: Deep network embeddings as laplacian embeddings with a nonlinearity.
In *SIGKDD*, 2020.

[DYB10] J-C Delvenne, Sophia N Yaliraki, and Mauricio Barahona.
Stability of graph communities across time scales.
PNAS, 107(29):12755–12760, 2010.

[GL16] Aditya Grover and Jure Leskovec.
node2vec: Scalable feature learning for networks.
In *SIGKDD*, 2016.

[HYL17] William L Hamilton, Rex Ying, and Jure Leskovec.
Representation learning on graphs: Methods and applications.
arXiv:1709.05584, 2017.

References II

[JDE+20] Amin Javari, Tyler Derr, Pouya Esmailian, Jiliang Tang, and Kevin Chen-Chuan Chang.
Rose: Role-based signed network embedding.
In *WebConf*, 2020.

[LG14] Omer Levy and Yoav Goldberg.
Neural word embedding as implicit matrix factorization.
In *NeurIPS*, 2014.

[NLR+18] Giang Hoang Nguyen, John Boaz Lee, Ryan A Rossi, Nesreen K Ahmed, Eunyee Koh, and Sungchul Kim.
Continuous-time dynamic network embeddings.
In *WebConf*, 2018.

[OCP+16] Mingdong Ou, Peng Cui, Jian Pei, Ziwei Zhang, and Wenwu Zhu.
Asymmetric transitivity preserving graph embedding.
In *SIGKDD*, 2016.

[PARS14] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena.
Deepwalk: Online learning of social representations.
In *SIGKDD*, 2014.

References III

[WCZ16] Daixin Wang, Peng Cui, and Wenwu Zhu.
Structural deep network embedding.
In *SIGKDD*, 2016.

[ZCW+18] Ziwei Zhang, Peng Cui, Xiao Wang, Jian Pei, Xuanrong Yao, and Wenwu Zhu.
Arbitrary-order proximity preserved network embedding.
In *SIGKDD*, 2018.

[ZLHK21] Jing Zhu, Xingyu Lu, Mark Heimann, and Danai Koutra.
Node proximity is all you need: Unified structural and positional node and graph embedding.
In *SDM*, 2021.