VERSE: **Vers**atile Graph **E**mbeddings from Similarity Measures

Anton Tsitsulin[1] Davide Mottin[1] Panagiotis Karras[2] Emmanuel Müller[1]

Hasso Plattner Institut
1

AARHUS UNIVERSITY
2

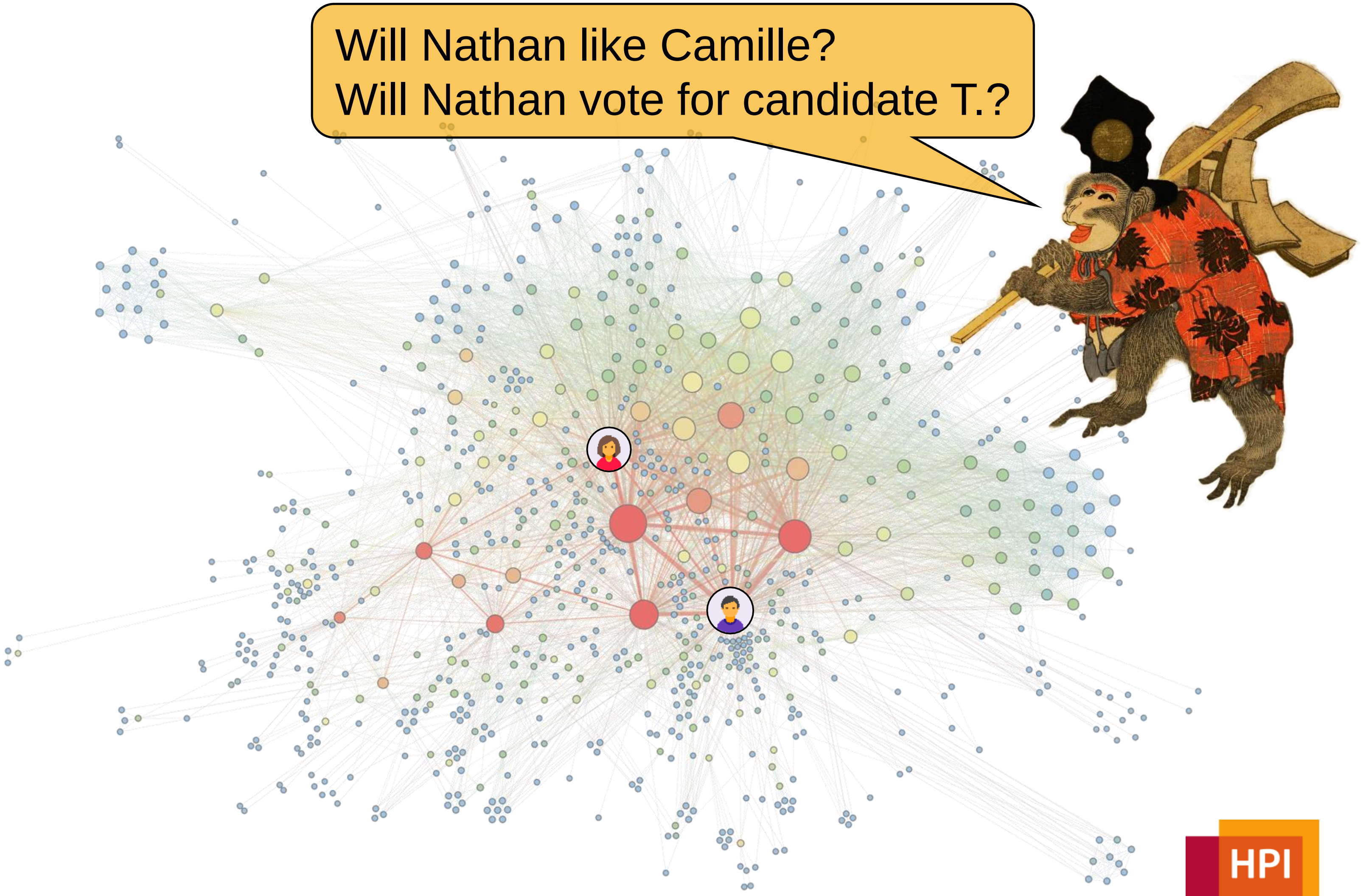
Graphs × tasks = problems

Different domains:

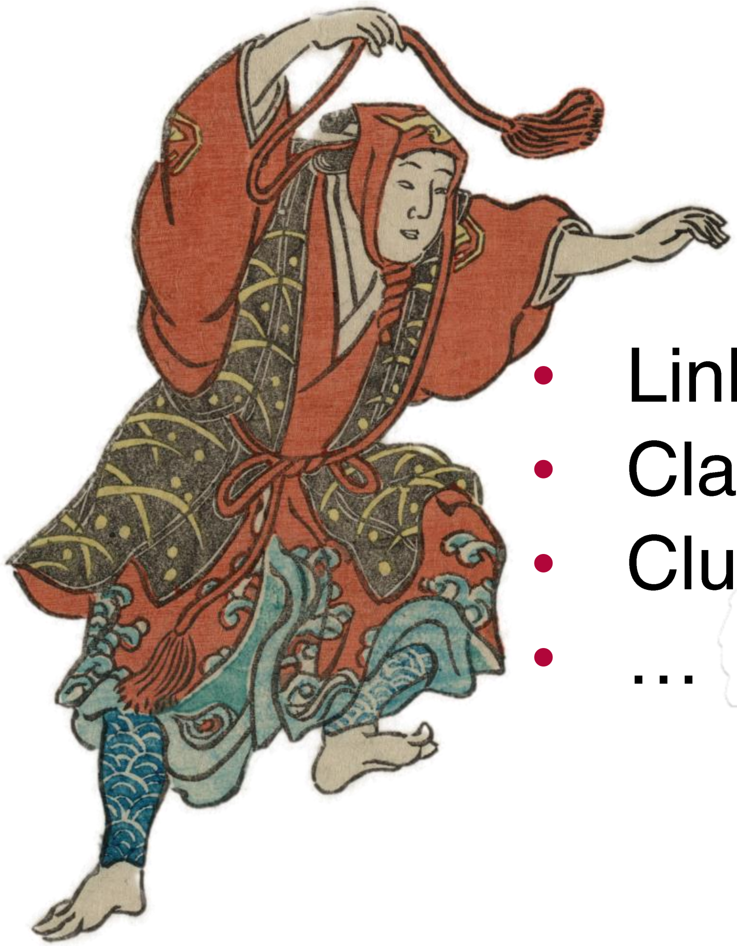
- Social nets
- Biology
- WWW

Different tasks:

- Classification
- Clustering
- Link prediction

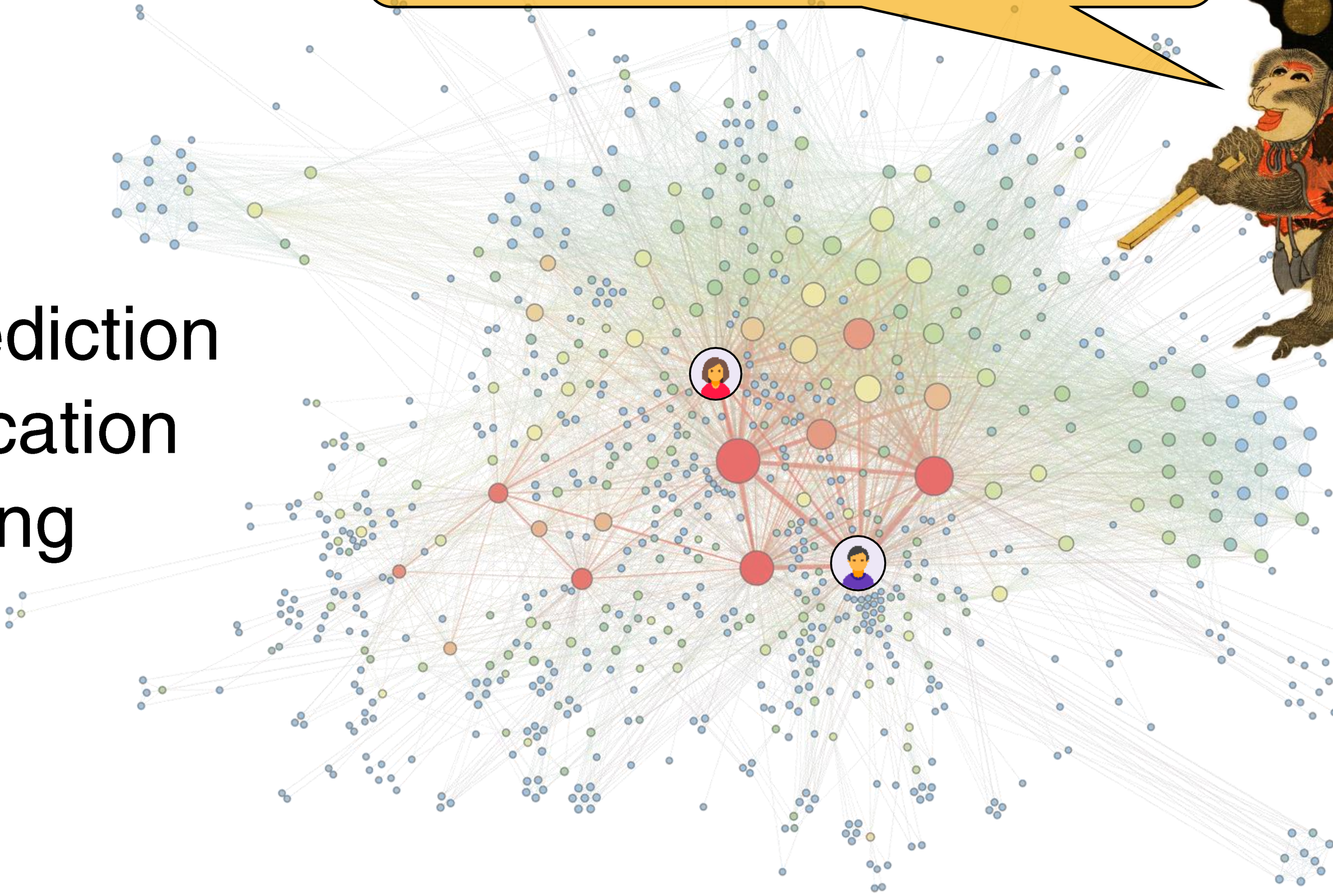


Graph mining ↔ domain experts?



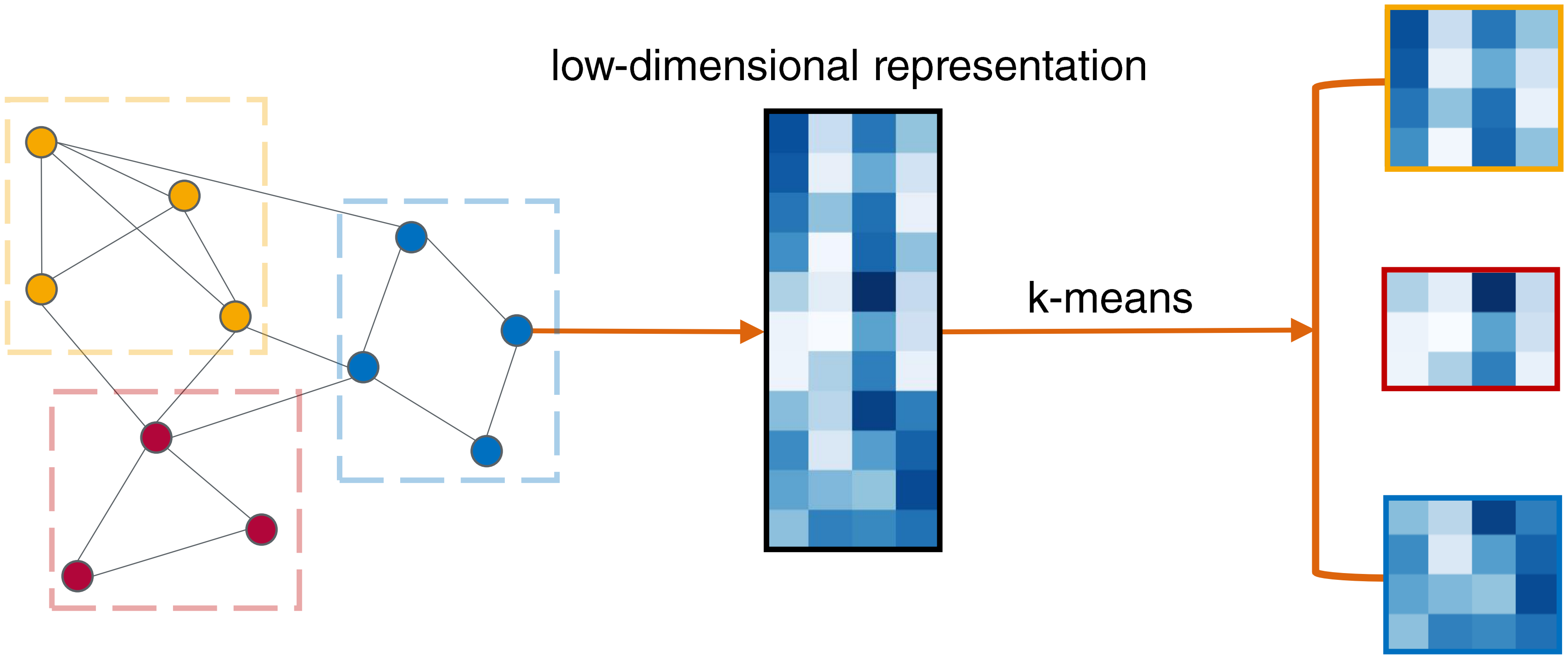
- Link prediction
- Classification
- Clustering
- …

Will Nathan like Camille?
Will Nathan vote for candidate T.?



Representations to the rescue

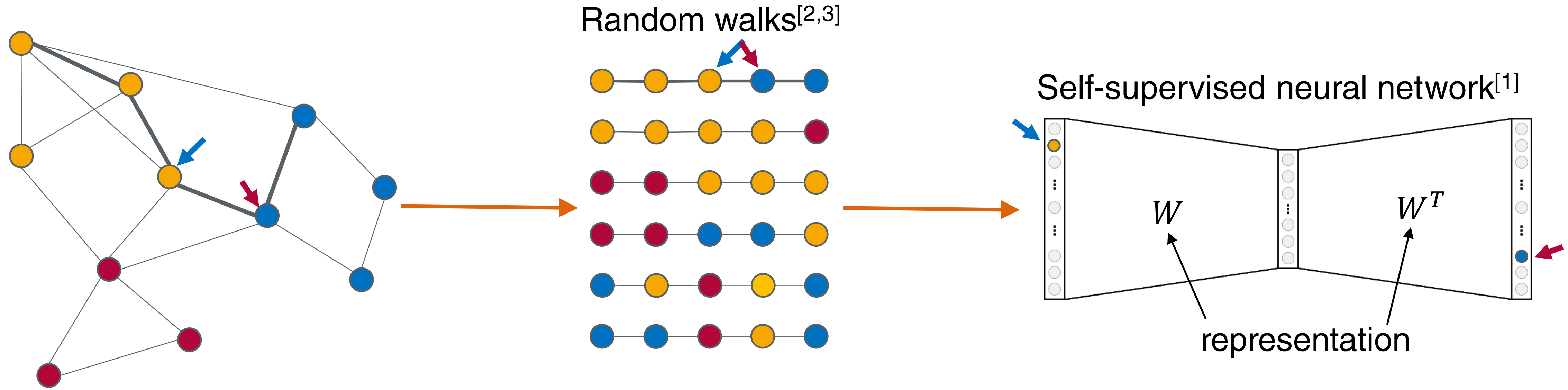
We have fast algorithms for mining vector data…



Q: How to represent a graph's nodes as vectors?

Neural node representations

Nodes in random walks ≈ words in sentences → word2vec



[1] Efficient Estimation of Word Representations in Vector Space, Mikolov et al., NIPS 2013
[2] DeepWalk: Online Learning of Social Representations, Perozzi et al., KDD 2014
[3] node2vec: Scalable Feature Learning for Networks, Grover & Leskovec, KDD 2016

Wait, what?

Do we know what do these walks <u>mean</u>?

- What do parameters change?
- What does the model optimize?

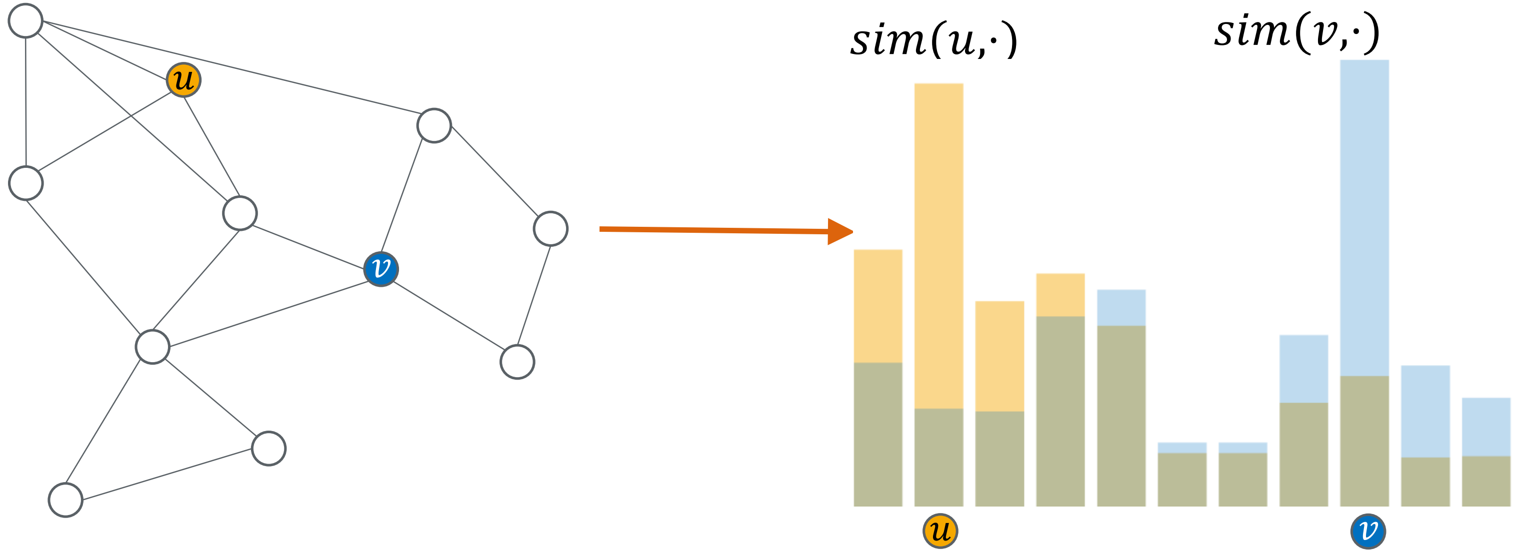
word2vec can be understood as matrix factorization!

Yes, but the assumptions are too strict!

dimensionality = number of nodes

Key observation

Random walks define node similarity distributions!

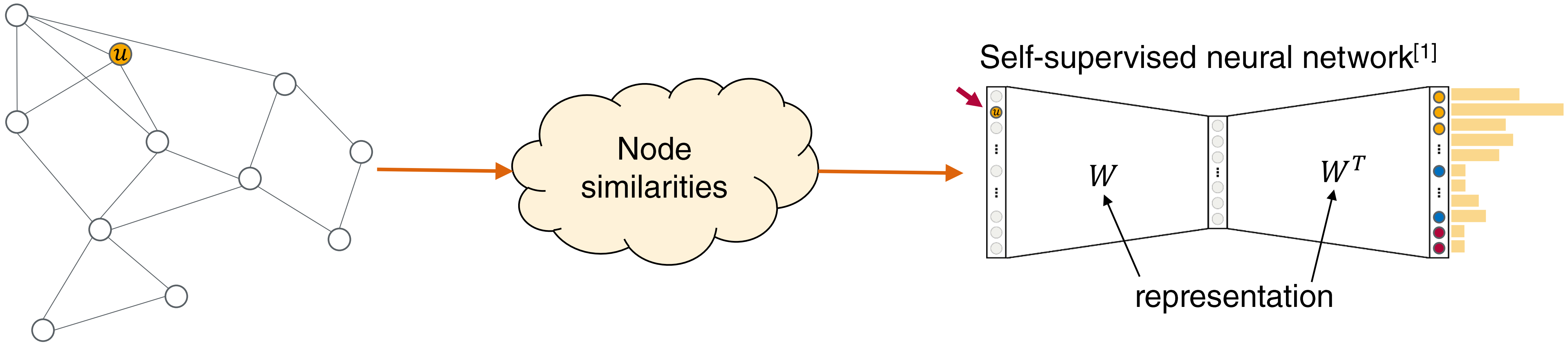


Random walks converge to Personalized PageRank

Q: Can we inject similarities fully into the model?

Yes, we can!

VERSE can learn similarity distributions

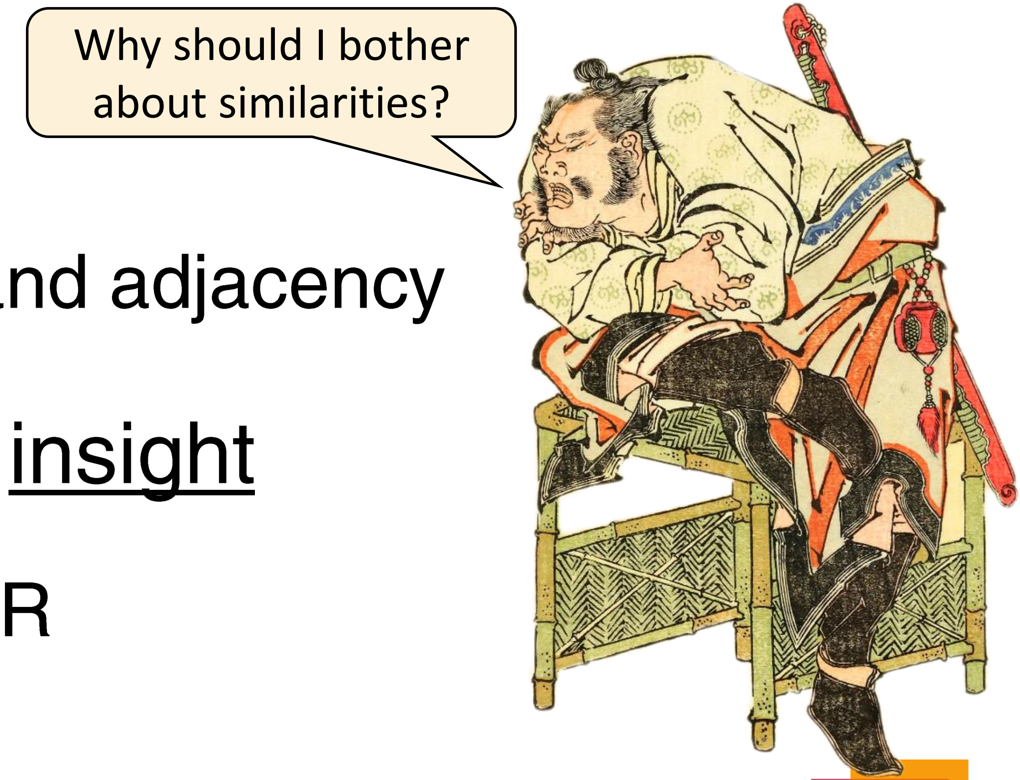


Q1: Which similarities can we possibly represent?

Q2: What other methods have to do with similarities?

Why similarities?

- We can measure quality explicitly
- We can easily change the similarity
  - We test VERSE with PPR, SimRank, and adjacency
- Thinking about similarities provides insight
  - We show DeepWalk & node2vec ≈ PPR
  - VERSE uses 1 parameter instead of 5

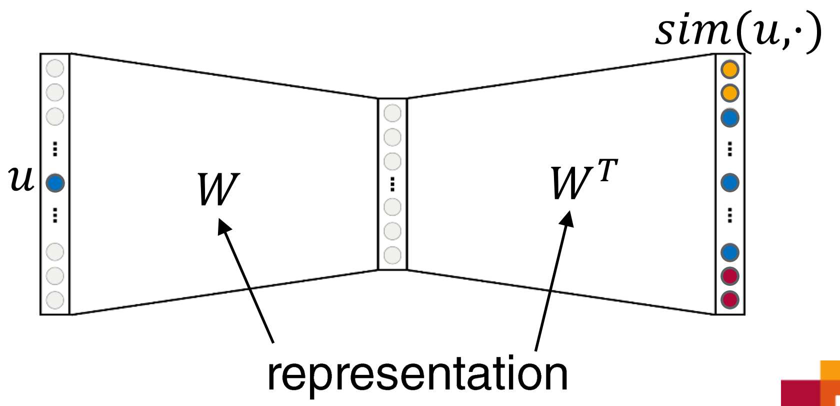


VERSE graph embedding

Algorithm for given $sim(u,\cdot)$:

1. Initialize $W \sim \mathcal{N}(0,1)$
2. For $u \in V$ optimize W for softmax $sim(u,\cdot)$ by gradient descent

Full updates are <u>too expensive</u> - $O(n^2)$



We make it faster with sampling!

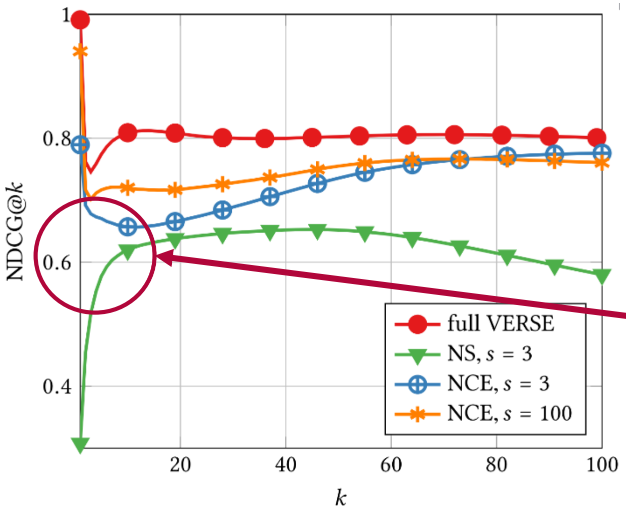
VERSE graph embedding: sampling

We use Noise Contrastive Estimation

$$\mathcal{L}_{NCE} = \sum_{\substack{u \sim \mathcal{P} \\ v \sim sim_{\mathrm{G}}(u,\cdot)}} \Big[\log \Pr_W(D = 1 | sim_{\mathrm{E}}(u, v)) +$$

$$k \mathbb{E}_{\widetilde{v} \sim \mathcal{Q}(u)} \log \Pr_W(D = 0 | sim_{\mathrm{E}}(u, \widetilde{v})) \Big]$$

Why not just using Negative Sampling?



Negative Sampling does not preserve similarities!

Experimental setting

- Graphs × tasks = problems
- PPR as a default, HSVERSE with tuned similarity
- Max one day on a 10-core server
  - No GPU farms, everything fits in desktop RAM
- Reimplement everything for performance
- Code and data available (bit.ly/www-verse)

Experiments: social graph × link prediction

			edge representation		
method	Average	Concat	Hadamard	L1	L2
VERSE	73.78	73.66	79.71	74.11	74.56
DeepWalk	70.05	69.92	69.79	78.38	77.37
LINE	75.17	75.13	72.54	63.77	64.47
HOPE	71.89	71.90	70.22	71.22	70.63
hsVERSE	74.14	74.02	80.26	73.04	73.53
Node2vec	71.29	71.22	72.43	78.38	78.66
Feature Eng.			78.84		

Link prediction

(accuracy)

Experiments: graphs × node clustering

method	CoCit	CoAuthor	VK	YouTube	Orkut
VERSE	69.43	79.25	45.78	67.63	42.64
DeepWalk	70.04	73.83	43.30	58.08	44.66
LINE	60.02	71.58	39.65	63.40	42.59
GraRep	67.61	77.40	—	—	—
HOPE	42.45	69.57	21.70	37.94	—
hsVERSE	69.81	79.31	45.84	69.13	—
Node2vec	70.06	75.78	44.27	—	—
Louvain	72.05	84.29	46.60	71.06	—

Node clustering

(modularity)

Experiments: web graph × node classification

method	*labelled nodes, %*				
	1%	3%	5%	7%	9%
VERSE	17.92	22.26	24.07	25.07	25.99
DeepWalk	18.16	21.55	22.89	23.64	24.54
LINE	13.71	17.36	18.69	19.84	20.64
HOPE	9.22	13.80	15.09	16.18	16.78
HSVERSE	18.16	22.84	25.40	27.38	29.09

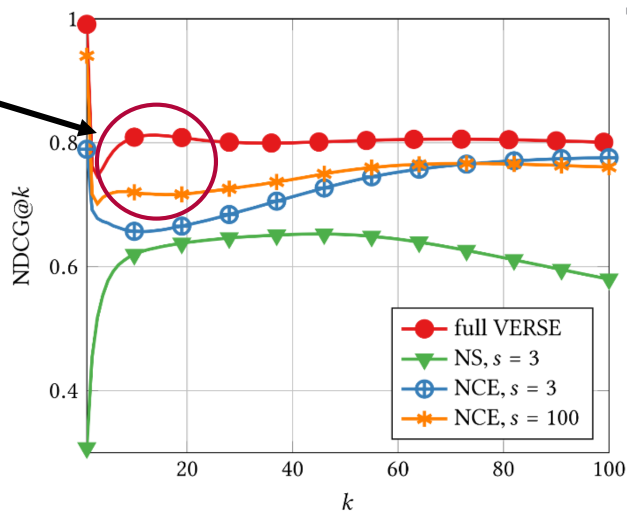
Classification

(accuracy)

HPI

Conclusion

1. We provide new <u>useful abstraction</u>: node similarities
2. We create VERSE to <u>explicitly</u> work with similarities
3. We develop a scalable approximation via NCE
4. There is a room for improvement!



Thank you
for your attention!

bit.ly/www-verse
github.com/xgfs/verse

HPI