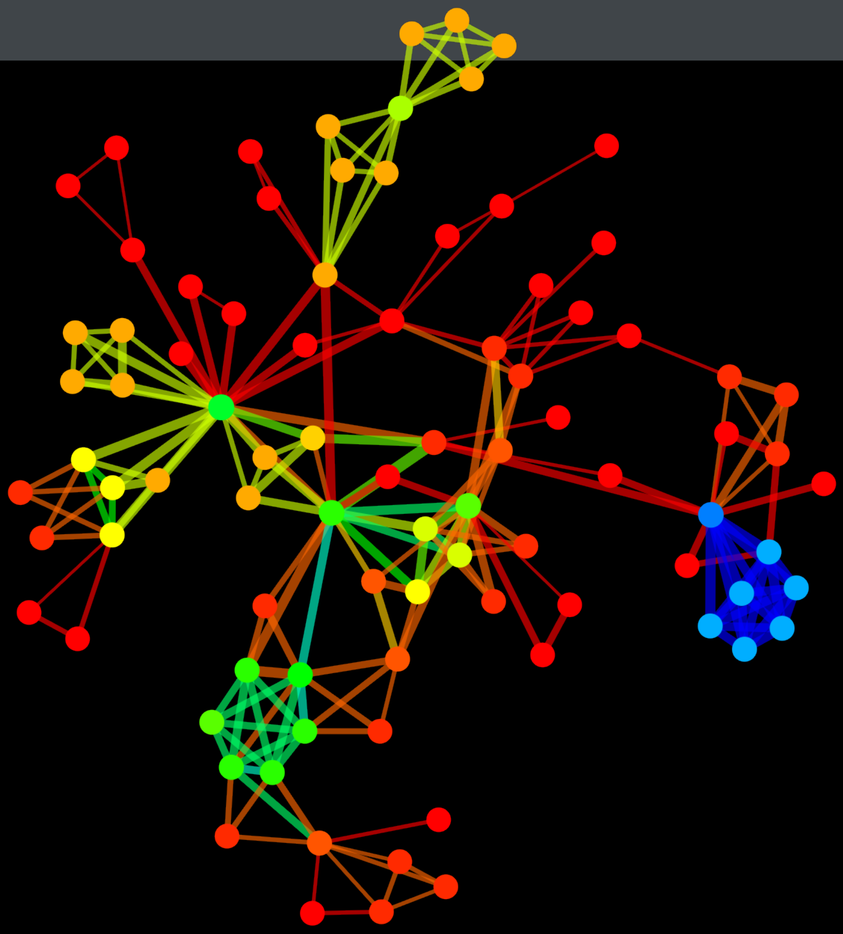
A Structural Graph Representation Learning Framework

Ryan A. Rossi (Adobe Research)

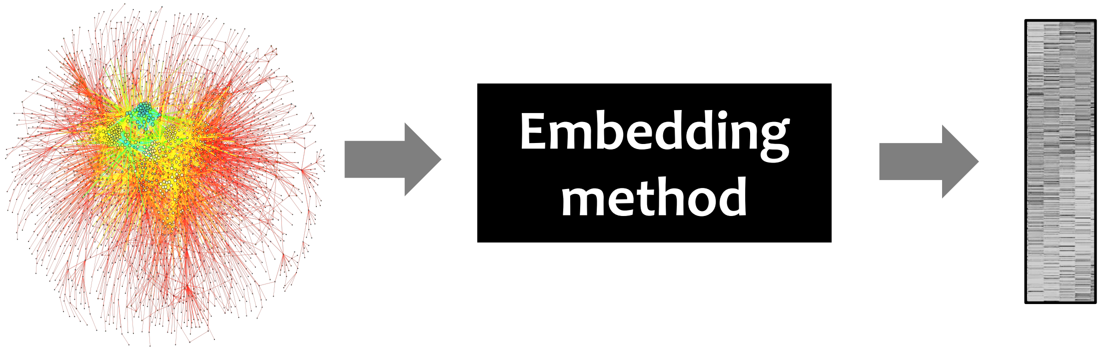
Joint work with:

Nesreen K. Ahmed (Intel Labs)
Eunyee Koh, Sungchul Kim, and Anup Rao (Adobe Research)
Yasin Abbasi-Yadkori (VinAI)



Motivation

- Success of ML algorithms depend on data representation



- Existing methods have many limitations
  - Most focus on proximity/community-based embeddings
  - Embeds nodes that are close to one another in the graph in a similar fashion
  - Embeddings do not generalize across different graphs
  - Unable to capture higher-order motif-based network structures

  …

Motivation

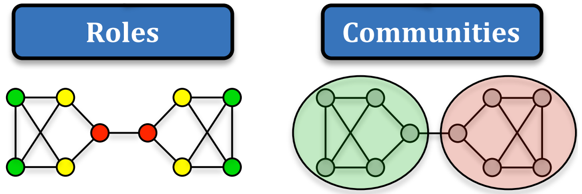
- Two complementary notions of embeddings in graphs:

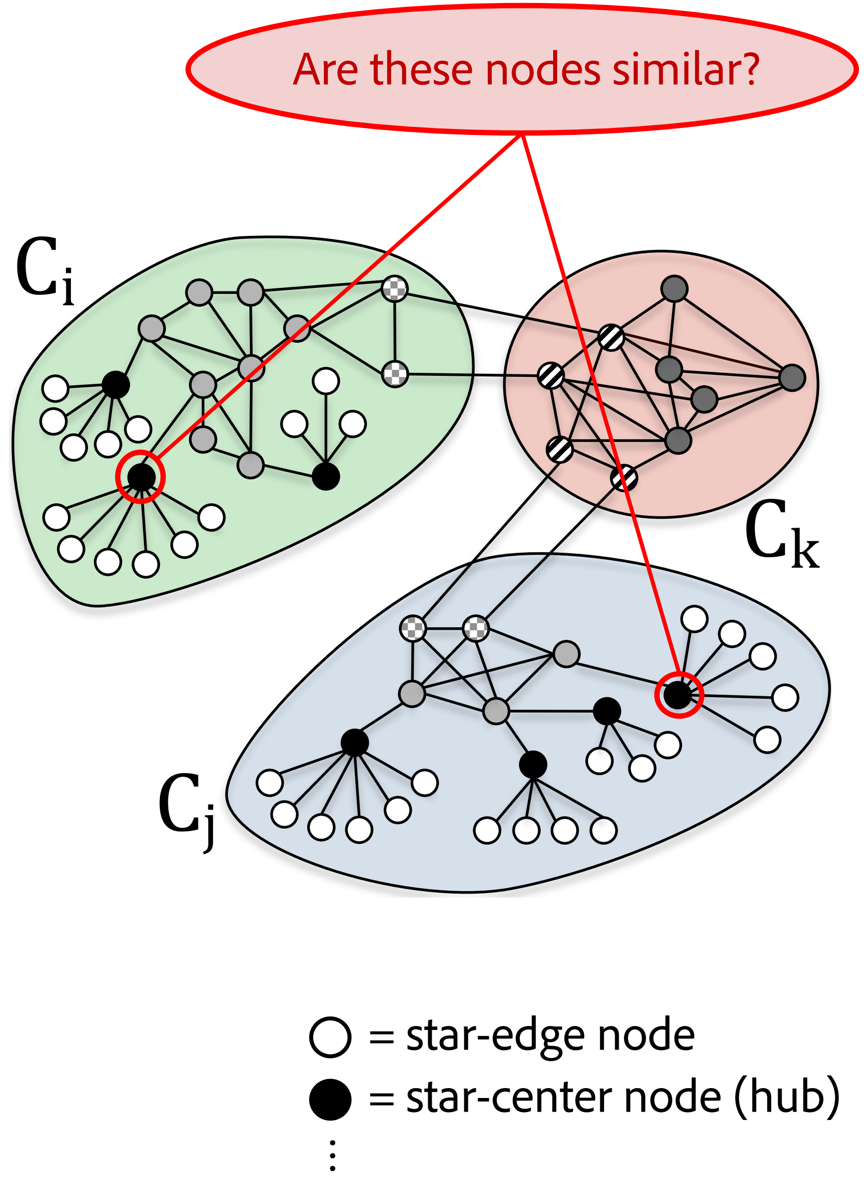
  proximity/community-based

  and

  structural role-based embeddings

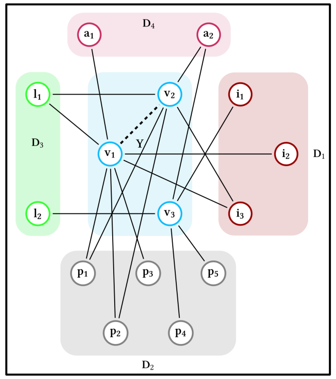
- Structural role-based embeddings = based on *structural similarity* [Rossi TKDE15]
  - Independent of distance/proximity
  - Generalize across-networks
  - Roles = key structural patterns/behaviors of nodes





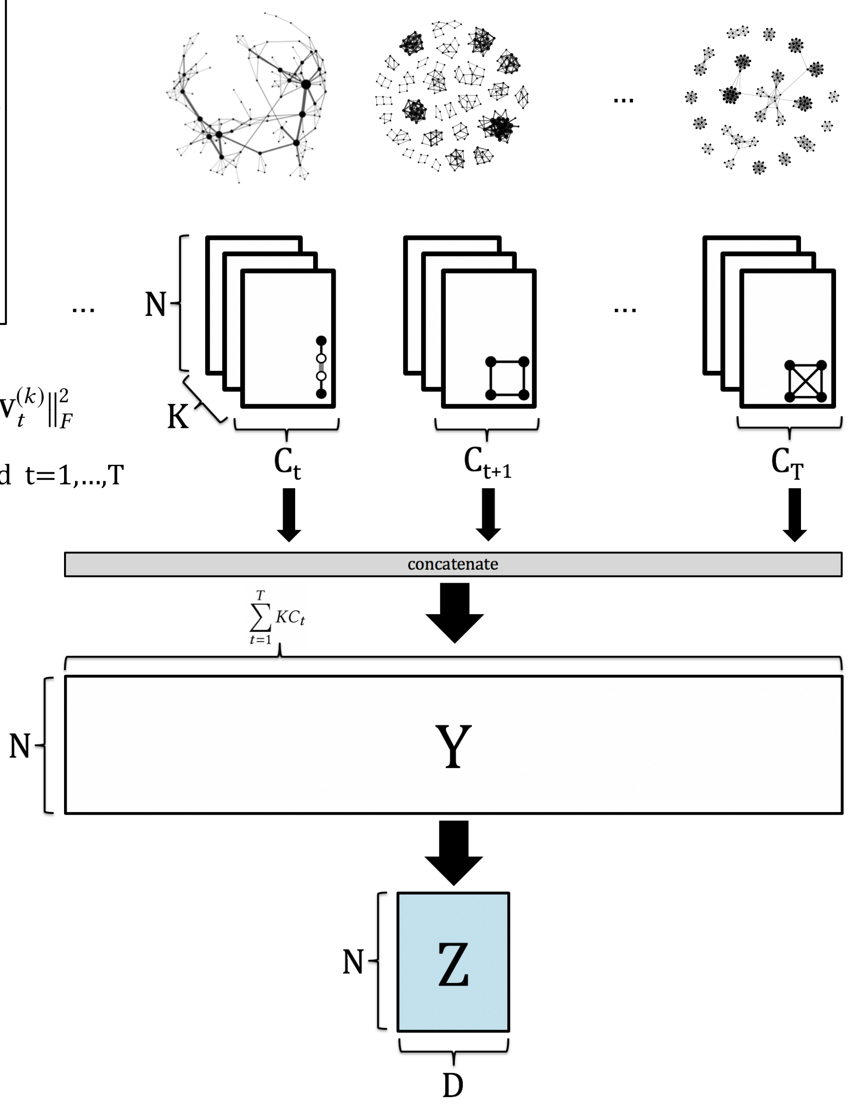
HONE Overview

1. Derive network motifs
2. Form motif adjacency matrices
3. Motif matrix functions
4. Derive k-step motif matrix
5. For each k-step motif matrix, learn node embeddings
6. Learn global higher-order embedding



$$\min_{U_t^{(k)},V_t^{(k)}} \frac{1}{2}\|S_t^{(k)} - U_t^{(k)}V_t^{(k)}\|_F^2$$

for k=1,...,K and t=1,...,T

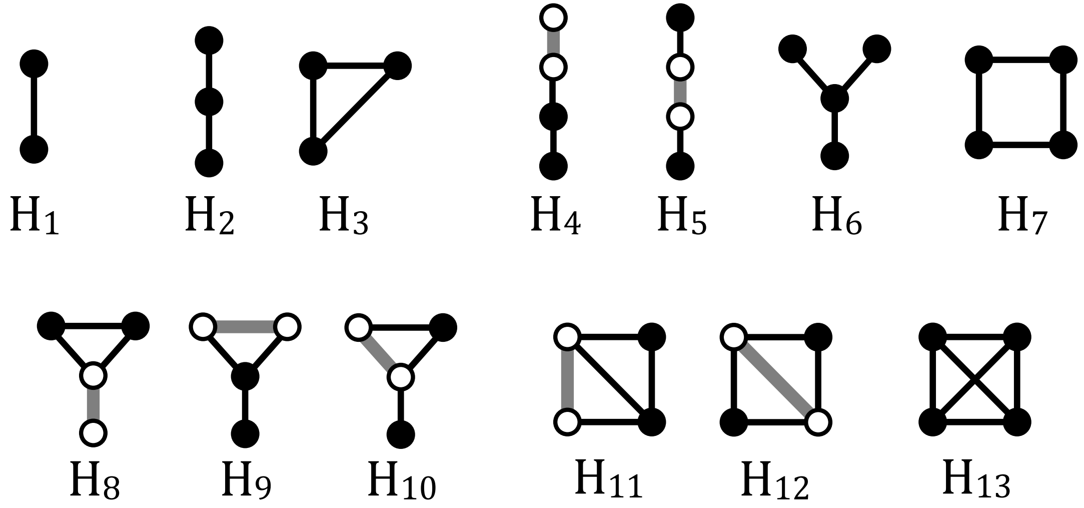


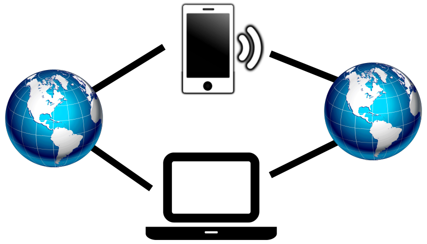
HIGHER ORDER NETWORK EMBEDDING (HONE):
Given a network G = (V, E), a set of network motifs $\mathcal{H} = \{H_1, \ldots, H_T\}$ the goal of higher-order network embedding (HONE) is to learn a function f : V → R^D that maps nodes to D-dimensional **structural node embeddings** using network motifs.

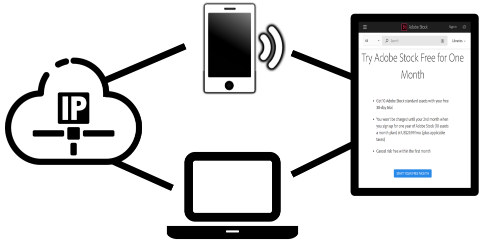
Step 1: Derive network motif counts

- Network motifs => **graphlets** (small induced subgraphs) or **orbits** (graphlet automorphisms)
- Graphlet decomposition

Basis for capturing structural behavior





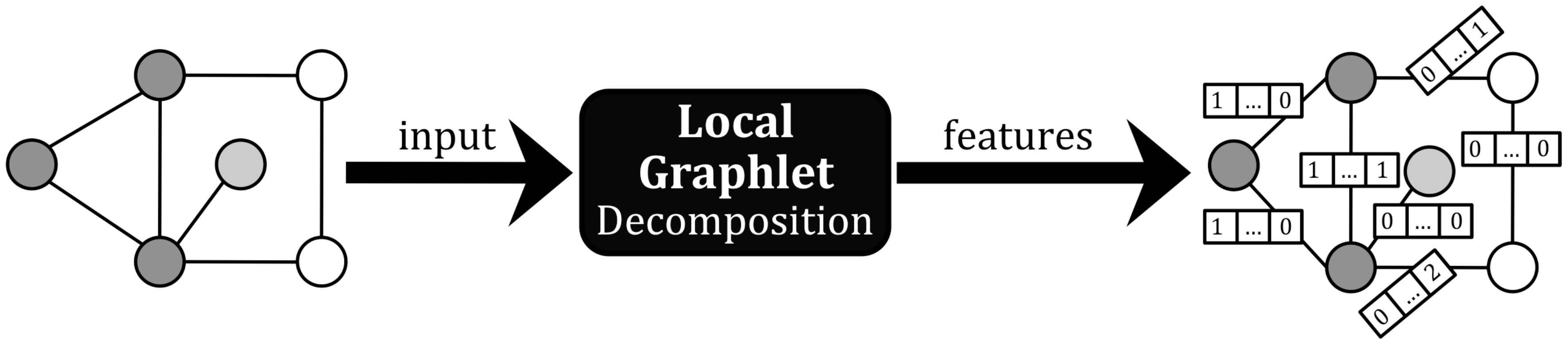


Applied to food, biologcal, genetic, neural, web, social and other networks

Network Motifs: Simple Building Blocks of Complex Networks – [Milo et. al – Science 2002]
The Structure and Function of Complex Networks – [Newman – Siam Review 2003]

Step 1: Derive network motif counts

- Network motifs => graphlets (small induced subgraphs) or orbits (graphlet automorphisms)



Exact algorithms: count cliques & cycles, and use combinatorial relationships to derive others in o(1) time [ICDM 2015]

Estimation methods: Motif estimators with provable error bounds [TNNLS18, BigData 2016]

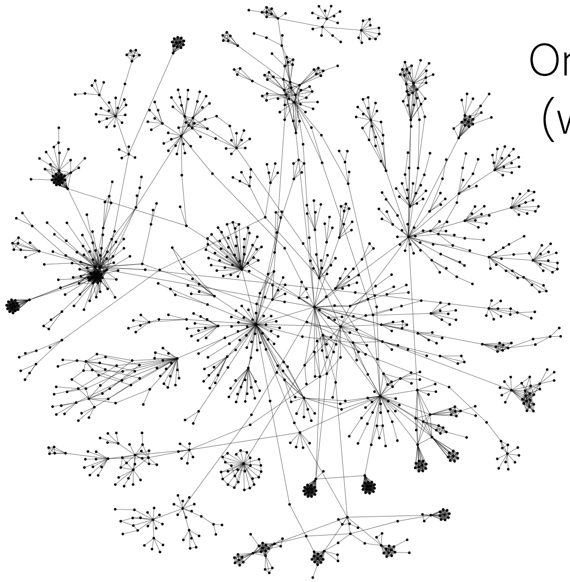
Step 2: Form weighted motif adjacency matrices

Given:

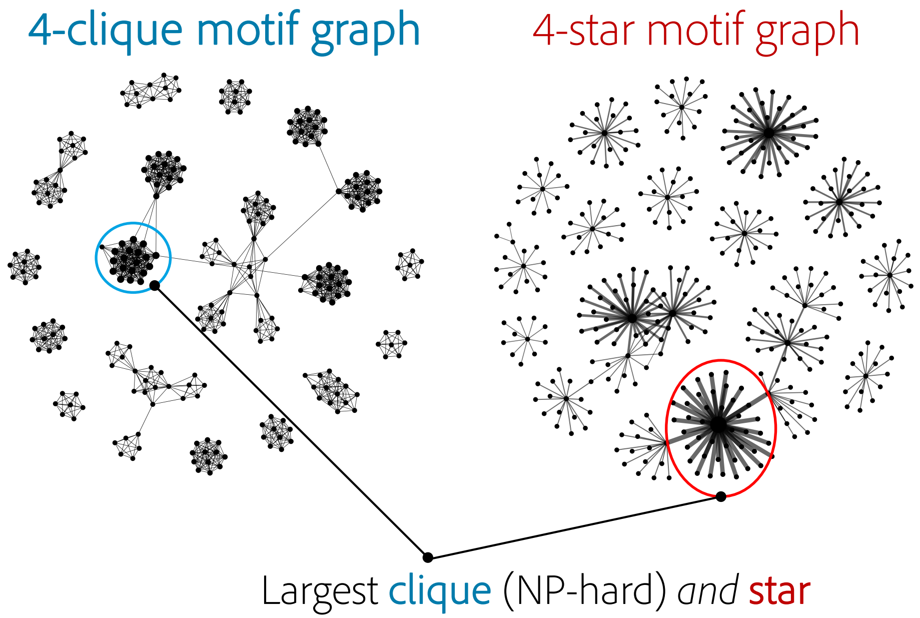
- a graph **G=(V, E)** and a set $\mathcal{H} = \{H_1, \ldots, H_T\}$ of T network motifs, we form the (weighted) motif adjacency matrices

$$\mathcal{W} = \{\mathbf{W}_1, \mathbf{W}_2, \ldots, \mathbf{W}_T\}$$

$$(\mathbf{W}_t)_{ij} = \text{\# of instances of motif } H_t \text{ that contain nodes i and j}$$



Original graph (web-google)



Step 3: Motif matrix functions

To generalize HONE for any motif-based matrix, we define a function $\Psi : \mathbb{R}^{N \times N} \rightarrow \mathbb{R}^{N \times N}$

A few motif matrix functions investigated:

- Motif Transition Matrix $\Psi : \mathbf{W} \rightarrow \mathbf{D}^{-1}\mathbf{W}$

$$\mathbf{P} = \mathbf{D}^{-1}\mathbf{W}$$

$$\sum_j P_{ij} = \mathbf{p}_i^T \mathbf{e} = 1$$

$$P_{ij} = \frac{W_{ij}}{w_i}$$

diagonal motif degree matrix:

$$\mathbf{D} = \mathtt{diag}(\mathbf{We})$$

$$\mathbf{e} = \begin{bmatrix} 1 & 1 & \cdots & 1 \end{bmatrix}^T$$

- Weighted Motif Laplacian

$$\mathbf{L} = \mathbf{D} - \mathbf{W}$$

can use in/out/total motif degree

- Normalized Weighted Motif Laplacian

$$\widehat{\mathbf{L}} = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}$$

$$\widehat{L}_{ij} = \begin{cases} 1 - \dfrac{W_{ij}}{w_j} & \text{if } i = j \text{ and } w_j \neq 0 \\ -\dfrac{W_{ij}}{\sqrt{w_i w_j}} & \text{if } i \text{ and } j \text{ are adjacent} \\ 0 & \text{otherwise} \end{cases}$$

$w_i = \sum_j W_{ij}$ is the motif degree of node i

- RW Norm. Weighted Motif Laplacian

$$\widehat{\mathbf{L}}_{\mathrm{rw}} = \mathbf{I} - \mathbf{D}^{-1}\mathbf{W}$$

Other interesting motif matrix formulations can also be used!

Step 4: Derive k-step motif-based matrices

- Given the motif matrix function Ψ and the set $\mathcal{W}$ of motif adjacency matrices, we derive all k-step motif-based matrices for all T motifs and K steps

Captures important dependencies further away

$$\mathbf{S}_t^{(k)} = \Psi(\mathbf{W}_t^k), \quad \text{for } k = 1, \ldots, K \text{ and } t = 1, \ldots, T$$

The number of paths weighted by motif counts from node i to node j in k-steps is

$$(\mathbf{W}^k)_{ij} = (\underbrace{\mathbf{W} \cdots \mathbf{W}}_{k})_{ij}$$

The probability (weighted by motif counts) of transitioning from node i to node j in k-steps is given by

$$(\mathbf{P}^k)_{ij} = (\underbrace{\mathbf{P} \cdots \mathbf{P}}_{k})_{ij}$$

non-uniform random walk that selects subsequent nodes with prob. proportional to the edge's motif count.

Step 5: Local k-step Motif Embeddings

$$\mathbf{S}_t^{(k)} \approx \Phi\langle \mathbf{U}_t^{(k)} \mathbf{V}_t^{(k)} \rangle$$

- We find low-rank "local" node embeddings for each motif and k-step matrix by solving:

$$\underset{\mathbf{U}_t^{(k)}, \mathbf{V}_t^{(k)} \in C}{\arg\min} \; \mathbb{D}\big(\mathbf{S}_t^{(k)} \,\|\, \Phi\langle \mathbf{U}_t^{(k)} \mathbf{V}_t^{(k)} \rangle \big), \quad \text{for k=1,...,K and t=1,...,T}$$

- Concatenate all k-step node embedding for all T motifs and K steps

$$\mathbf{Y} = \Big[\underbrace{\mathbf{U}_1^{(1)} \cdots \mathbf{U}_T^{(1)}}_{\text{1-step}} \cdots \underbrace{\mathbf{U}_1^{(K)} \cdots \mathbf{U}_T^{(K)}}_{K\text{-steps}} \Big]$$

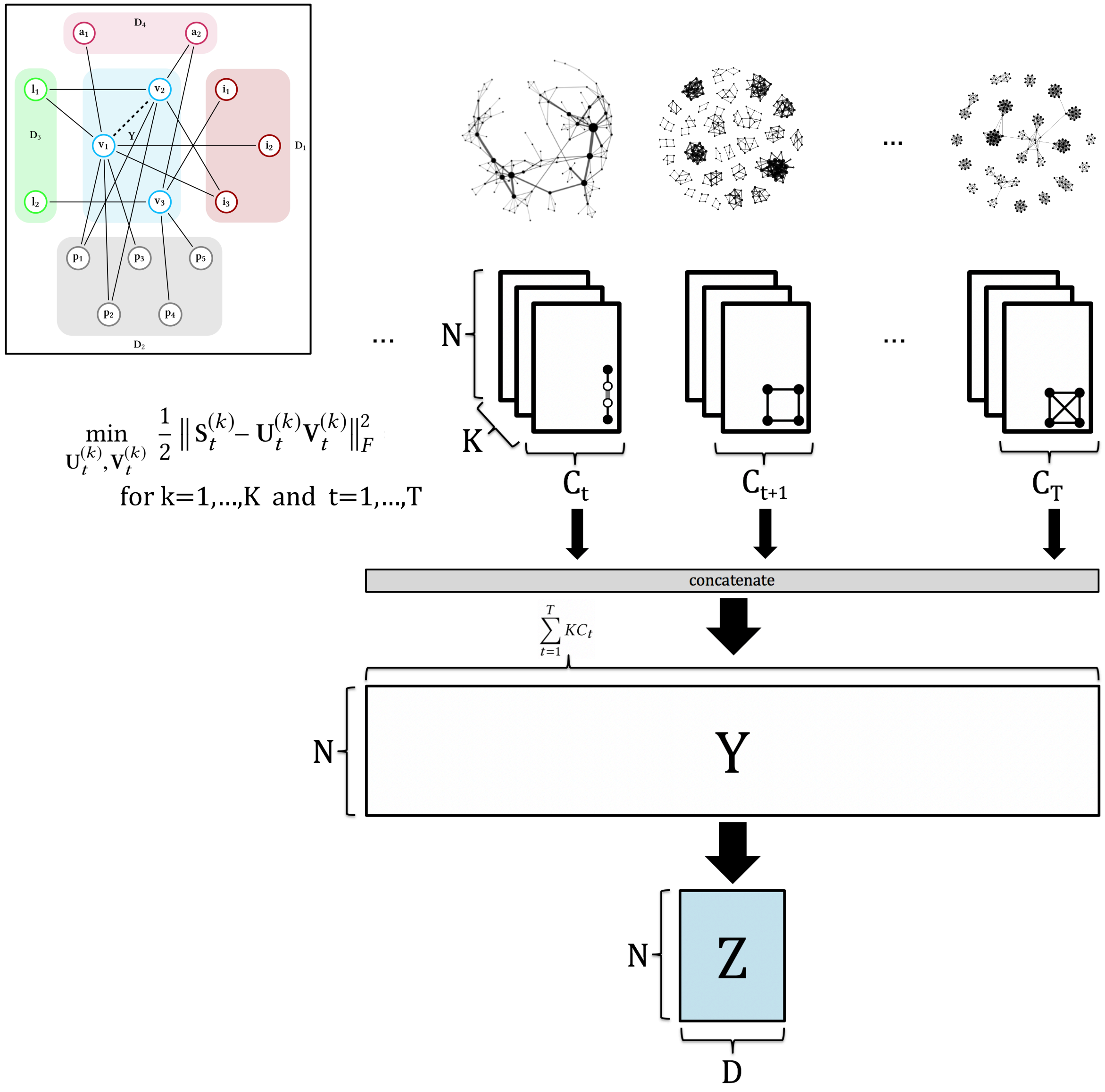
$$\mathbf{Y} \text{ is } N \times TKD_\ell$$

Step 6: Global higher-order node embeddings

- Given $\mathbf{Y} = \left[\, \mathbf{U}_1^{(1)} \;\cdots\; \mathbf{U}_T^{(1)} \;\cdots\; \mathbf{U}_1^{(K)} \;\cdots\; \mathbf{U}_T^{(K)} \,\right]$, find low-rank "global" higher-order node embeddings by solving

$$\underset{\mathbf{Z},\mathbf{H}\in \mathcal{C}}{\arg\min}\;\; \mathbb{D}\big(\,\mathbf{Y} \,\|\, \Phi\langle \mathbf{Z}\mathbf{H}\rangle\big)$$

$$\min_{\mathbf{Z},\mathbf{H}} \; \frac{1}{2}\,\|\mathbf{Y} - \mathbf{Z}\mathbf{H}\|_F^2 \;=\; \frac{1}{2}\sum_{ij}\big(\mathbf{Y}_{ij} - (\mathbf{Z}\mathbf{H})_{ij}\big)^2$$



Extensions

- Attribute diffusion

$$\bar{\mathbf{X}}_t^{(0)} \leftarrow \mathbf{X}$$

$$\bar{\mathbf{X}}_t^{(k)} = \Psi\big(\mathbf{W}_t^{(k)}\big)\bar{\mathbf{X}}_t^{(k-1)}, \quad \text{for } k = 1, 2, \dots, K \quad \text{OR} \quad \bar{\mathbf{X}}^{(k)} = (1-\theta)\mathbf{L}\bar{\mathbf{X}}^{(k-1)} + \theta\mathbf{X}, \quad \text{for } k = 1, 2, \dots$$

- Accumulation motif variants

$$\bar{\mathbf{S}}^{(k)} = \sum_{\ell=1}^{k} \alpha^{\ell}\Psi(\mathbf{W}^{\ell}) = \alpha\Psi(\mathbf{W}) + \alpha^2\Psi(\mathbf{W}^2) + \cdots + \alpha^k\Psi(\mathbf{W}^k)$$

- Weighted and combined motif matrix

$$\mathbf{W} = \sum_{t=1}^{T} \beta_t \mathbf{W}_t \qquad \beta_t \geq 0$$

$$\bar{\mathbf{W}}^{(k)} = \sum_{\ell=1}^{k} \mathbf{W}^{\ell} = \mathbf{W} + \mathbf{W}^2 + \cdots + \mathbf{W}^k$$

where $\mathbf{W}_k$ is a weighted graph that counts the number of paths of length up to k.

Results

Experimental setup

- 10-fold cross-validation, repeated for 10 random trials
- Used all 2-4 node connected orbits
- D=128, Dl = 16 for the local motif embeddings
- Edge embedding derived via $(\mathbf{z}_i + \mathbf{z}_j)/2$
- Predict link existence via logistic regression (LR)
- # steps K selected via grid search over $K \in \{1, 2, 3, 4\}$

	soc-hamster	*rt-twitter-cop*	*soc-wiki-Vote*	*tech-routers-rf*	*facebook-PU*	*inf-openflights*	*soc-bitcoinA*	RANK
HONE-W (Eq. 2)	0.841	0.843	0.811	0.862	0.726	0.910	0.979	**1**
HONE-P (Eq. 5)	0.840	0.840	0.812	0.863	0.724	0.913	0.980	**2**
HONE-L (Eq. 7)	0.829	0.841	0.808	0.858	0.722	0.906	0.975	**3**
HONE-$\widehat{\mathbf{L}}$ (Eq. 8)	0.829	0.836	0.803	0.862	0.722	0.908	0.976	**5**
HONE-$\widehat{\mathbf{L}}_{\mathrm{rw}}$ (Eq. 9)	0.831	0.834	0.808	0.863	0.723	0.909	0.976	**4**
Node2Vec [19]	0.810	0.635	0.721	0.804	0.701	0.844	0.894	**6**
DeepWalk [34]	0.796	0.621	0.710	0.796	0.696	0.837	0.863	**7**
LINE [47]	0.752	0.706	0.734	0.800	0.630	0.837	0.780	**8**
GraRep [9]	0.805	0.672	0.743	0.829	0.702	0.898	0.559	**9**
Spectral [48]	0.561	0.699	0.593	0.602	0.516	0.606	0.629	**10**

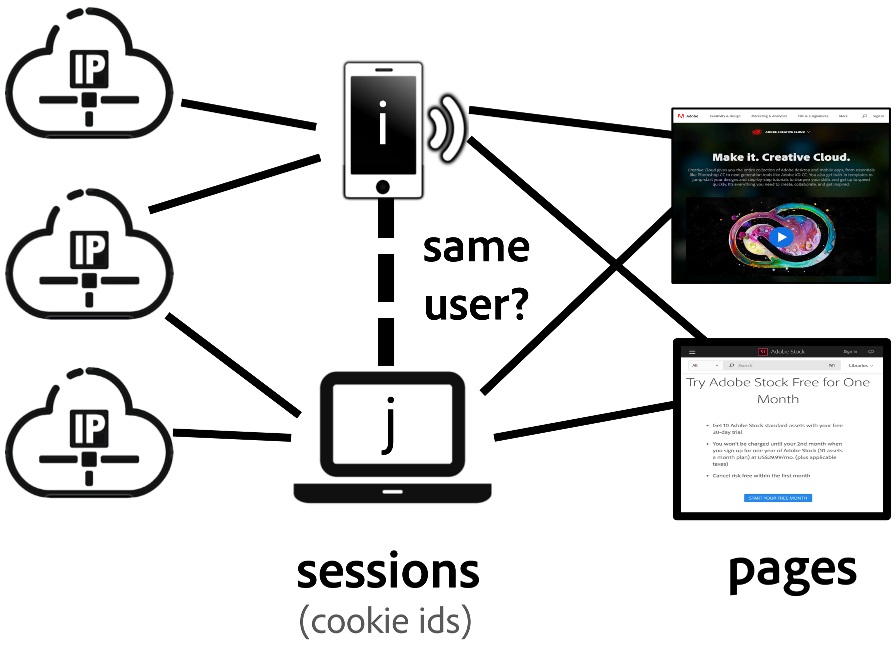
Main findings:

- Mean Gain in AUC of **19.24%** (& up to 75.21%)

Visitor Stitching of Web Logs

Problem: Given web browser logs, the goal is to predict the sessions (cookie ids) that correspond to the same user.

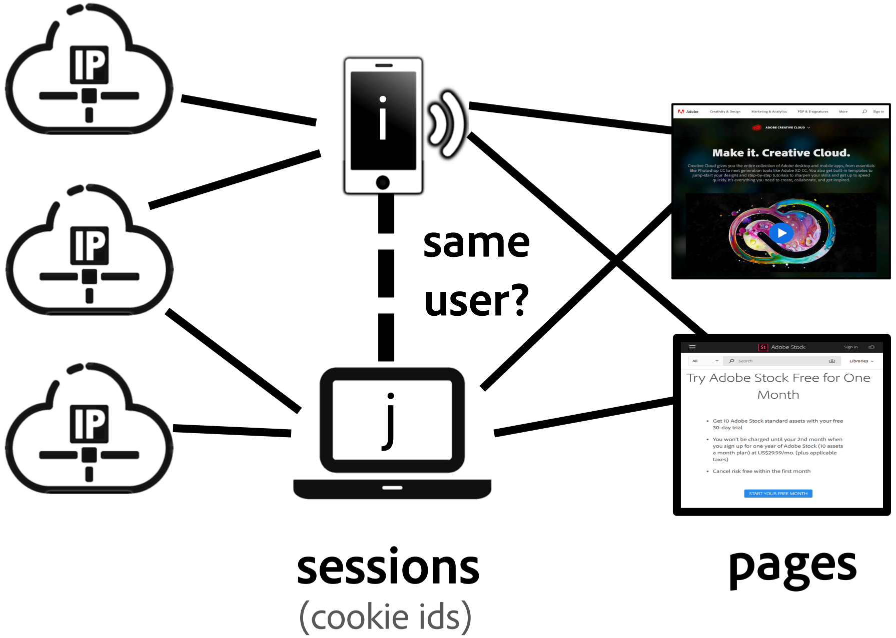
- Core to many products
- Utility/perf. of downstream applications relies on it



Tasks	Graph	$\|V\|$	$\|E\|$	d_{avg}
Visitor Stitching	Comp-A (web logs)	8.9M	55.2M	6.2
	Comp-B (web logs)	22.8M	61.3M	2.7

Visitor Stitching of Web Logs

Problem: Given web browser logs, the goal is to predict the sessions (cookie ids) that correspond to the same user.



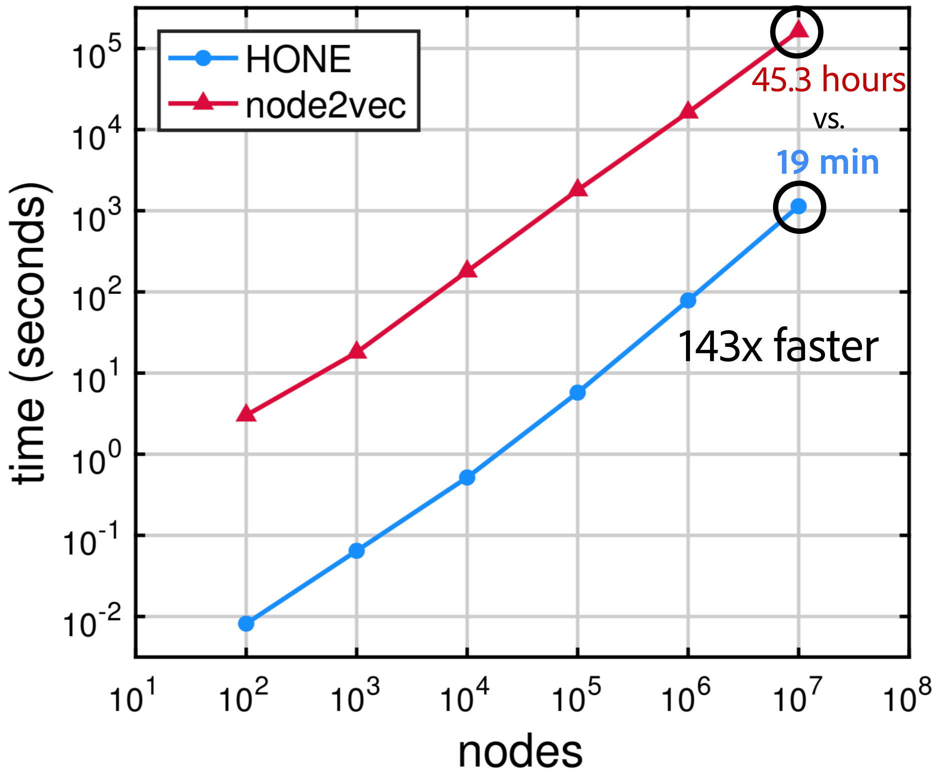
		N2V	DW	LINE	GraRep	Spec.	HONE	HONE+$\bar{X}$
COMP.-A	**Prec.**	ETL	ETL	0.8947	0.9924	0.7404	**0.9999**	0.9810
	Rec.	ETL	ETL	0.6254	0.4388	0.4907	0.6676	**0.8777**
	F1	ETL	ETL	0.7362	0.6085	0.5902	0.8007	**0.9265**
	AUC	ETL	ETL	0.7968	0.7215	0.5730	0.8572	**0.9304**
COMP.-B	**Prec.**	ETL	ETL	0.8362	0.9945	0.7731	**0.9923**	0.9885
	Rec.	ETL	ETL	0.3985	0.0937	0.2768	0.7336	**0.7736**
	F1	ETL	ETL	0.5397	0.1713	0.4077	0.8249	**0.8534**
	AUC	ETL	ETL	0.6843	0.5447	0.5259	0.8626	**0.8811**

*N2V=Node2Vec, DW=DeepWalk

Runtime & Scalability

- State-of-the-art method takes 1.8 days (45.3 hours) for 10 million nodes (avg. degree of 10)
- HONE finishes in only **19 minutes** (10M nodes, 100M edges)
  - Results from laptop
- 143x faster

Erdös-Rényi graphs (from 100 to 10 million nodes) w/ avg. degree 10

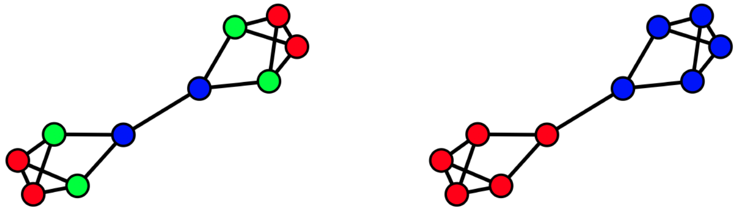


Results for Attribute/Feature-Diffusion Variants

Mean gain of the HONE methods with attribute diffusion relative to each of the original HONE methods

	HONE-W	**HONE-P**	**HONE-L**	**HONE-$\widehat{\mathbf{L}}$**	**HONE-$\widehat{\mathbf{L}}_{\mathrm{rw}}$**
HONE-W + $\bar{\mathbf{X}}$	0.76%	1.30%	1.38%	1.24%	1.08%
HONE-P + $\bar{\mathbf{X}}$	1.58%	2.12%	2.20%	2.06%	1.90%
HONE-L + $\bar{\mathbf{X}}$	0.62%	1.15%	1.23%	1.09%	0.93%
HONE-$\widehat{\mathbf{L}}$ + $\bar{\mathbf{X}}$	1.37%	1.91%	1.99%	1.85%	1.69%
HONE-$\widehat{\mathbf{L}}_{\mathrm{rw}}$ + $\bar{\mathbf{X}}$	1.27%	1.81%	1.88%	1.74%	1.58%

Structural Role-based Embeddings

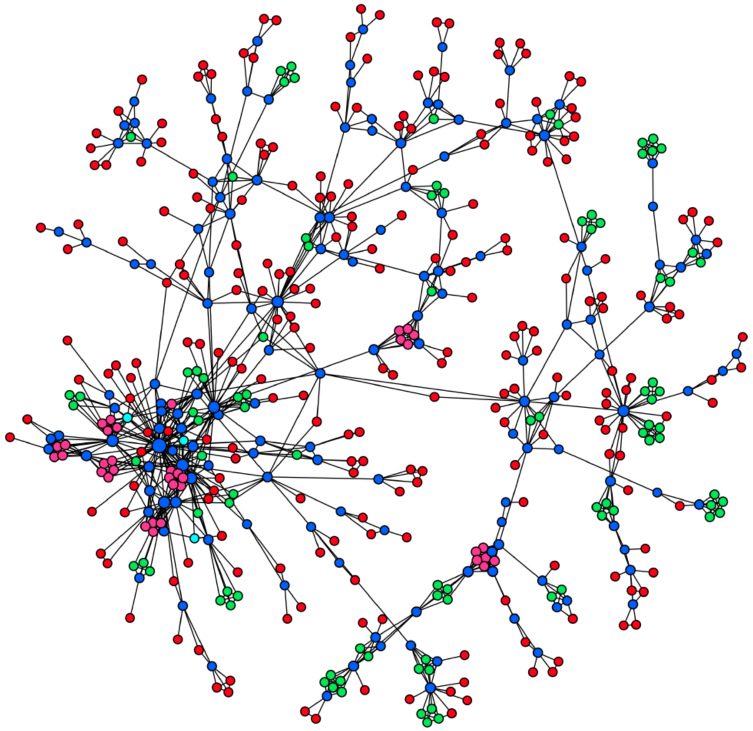


(a) HONE (b) node2vec

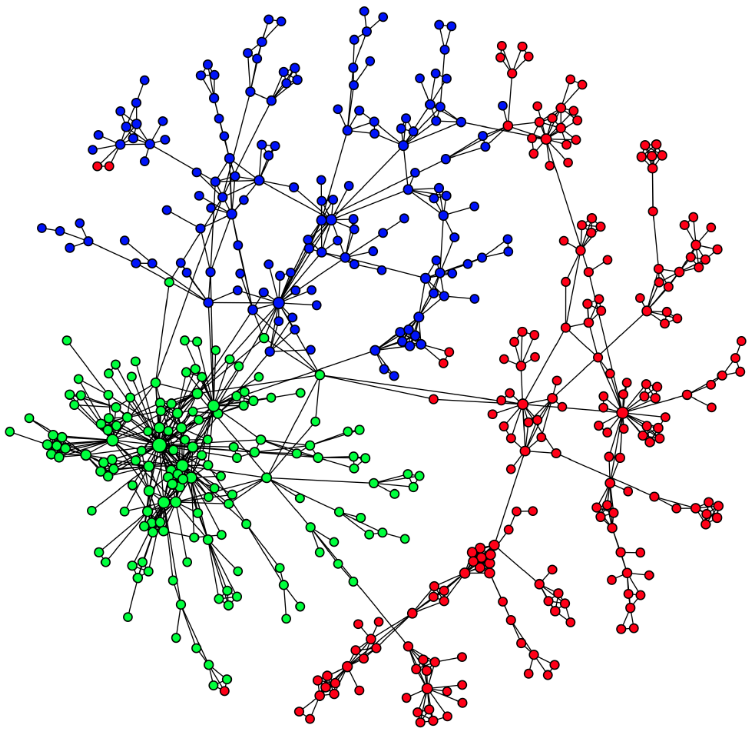
Validation of HONE's ability to capture roles on graphs with known ground-truth.

Structural Role-based Embeddings

diseasome network



(a) HONE



(b) node2vec

Summary & Key Contributions

- Introduced Higher-Order motif-based Network Embeddings (HONE)
- Described a computational framework for computing them
- Demonstrated the effectiveness of higher-order network embeddings for link prediction and visitor stitching

Future work should investigate other HONE variants, matrix motif formulations, etc.

Thanks!

Questions?

Data accessible online:

http://networkrepository.com

Adobe Research