Modeling Dynamic Behavior in Large Evolving Graphs

R. Rossi, J. Neville, B. Gallagher, and K. Henderson

Presented by: Doaa Altarawy

Outline

- Motivation
- Proposed Model
- Definitions
- Modeling dynamic graphs
- Results

Introduction

Activity Networks:

Structure change over time

Examples:

- Personal communications (email, phone)
- Social networks (Twitter, Facebook)
- Web traffic

Motivation: Why modeling Dynamic Graphs?

1. Identify dynamic **patterns** in node behavior

Motivation: Why modeling Dynamic Graphs?

1. Identify dynamic **patterns** in node behavior

2. Predict **future** structural changes

Motivation: Why modeling Dynamic Graphs?

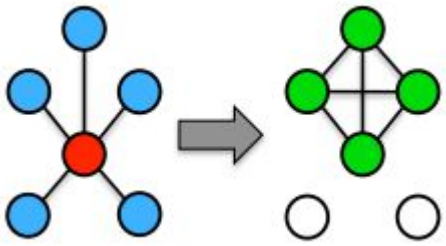
1. Identify dynamic **patterns** in node behavior

2. Predict **future** structural changes

3. Detect **unusual** transitions in behavior

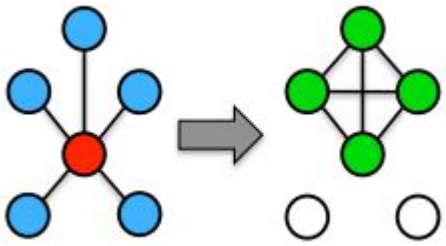
Proposed work

- Goal: Modeling behavioral roles of nodes and their evolution over time



Proposed work

- Goal: Modeling behavioral roles of nodes and their evolution over time



- Dynamic behavioral mixed-membership model (DBMM)
  - Discovers **graph features** for all timesteps
  - Learns behavioral **“roles”** for nodes at each timestep

The concept of Roles

Communities: set of nodes with more connections inside the set set than outside

The concept of Roles

Communities: set of nodes with more connections inside the set set than outside

Roles: set of nodes that are more structurally similar to nodes inside the set than outside

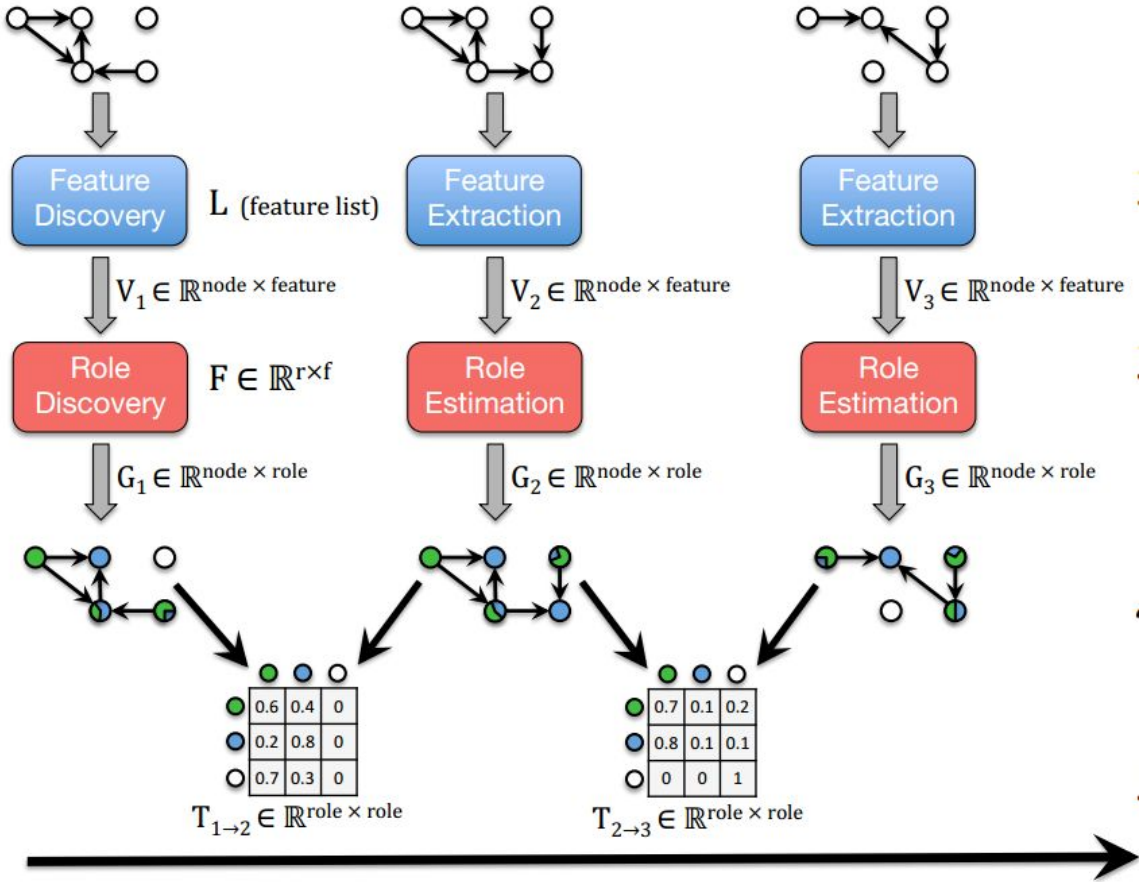
Preliminary

Dynamic network $D = (n , E)$

- n is the set of nodes and E is the set of edges in D

Network snapshot $S_t = (n_t , E_t)$

- a subgraph of D
- E_t active edges at time t
- n_t active nodes at time t



from (http://www.ryanrossi.com/talks/wsdm13-dbmm-rossi.pdf)

Modeling Steps

1- Learn set of features

2- Extract the features of each snapshot $\rightarrow V_1, V_2, \ldots, V_t$

3- Learn roles from features using NMF

4- Extract roles from each feature matrix $\rightarrow G_1, G_2, \ldots, G_t$

5- Use NMF to estimate transition model

Feature discovery

- Represent each **active node** with a set of features
- Uses the method in (Henderson and Keith et al., 2011) to create a **feature matrix** for each snapshot:

$$\boldsymbol{V} = \{\boldsymbol{V}_t : t = 1, ..., t_{max}\}.$$

$$\boldsymbol{V}_t \quad \text{is} \quad n_t \times f$$

1. Constructs measures (degrees, clustering coeff, ..),
2. aggregates using sum (or mean), creating recursive features,
3. prune correlated features,
4. proceed aggregation recursively

Role discovery

- automatically discover groups of nodes (representing common behavior) based on their features.
- use Non-negative Matrix Factorization (NMF) to extract roles as in (Henderson and Keith et al., 2012)
- minimize the function:

$$f(\mathbf{G}_t, \mathbf{F}) = \frac{1}{2}||\mathbf{V}_t - \mathbf{G}_t\mathbf{F}||_F^2$$

where

$$\mathbf{G}_t \in \mathbb{R}^{n_t \times r} \text{ and } \mathbf{F} \in \mathbb{R}^{r \times f}$$

Result:

$$\boldsymbol{G} = \{\boldsymbol{G}_t : t = 1, ..., t_{max}\}$$

Behavioral Transition Model

- learn a transition matrix T that approximates the change in behavior from time t-1 to t

- ***T*** is estimated using NMF such that $\boldsymbol{G}_{t-1}\boldsymbol{T} \approx \boldsymbol{G}_t$

Behavioral Transition Model

- learn a transition matrix T that approximates the change in behavior from time t-1 to t

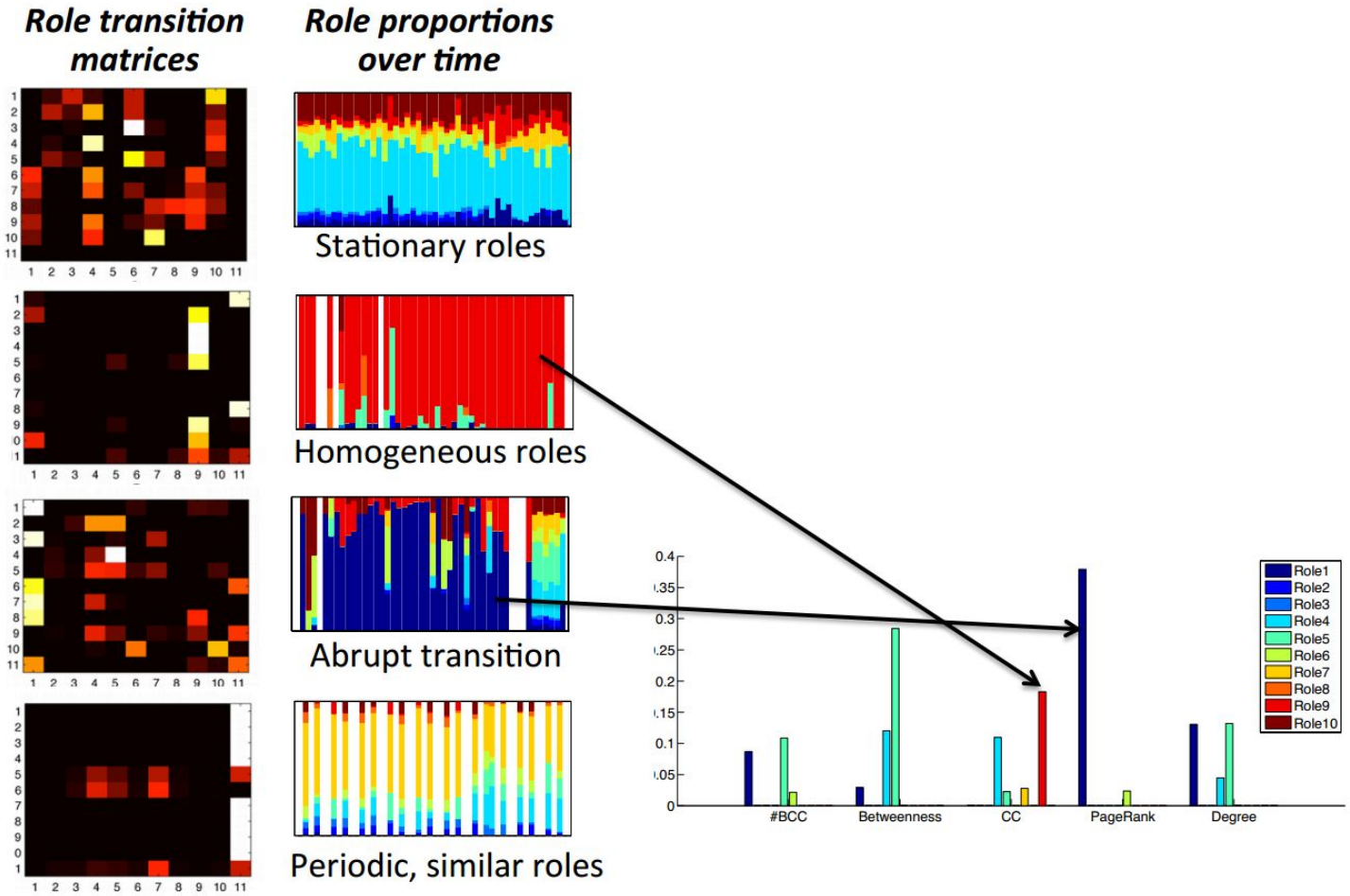
- ***T*** is estimated using NMF such that $\boldsymbol{G}_{t-1}\boldsymbol{T} \approx \boldsymbol{G}_t$

- To predict future behavior:

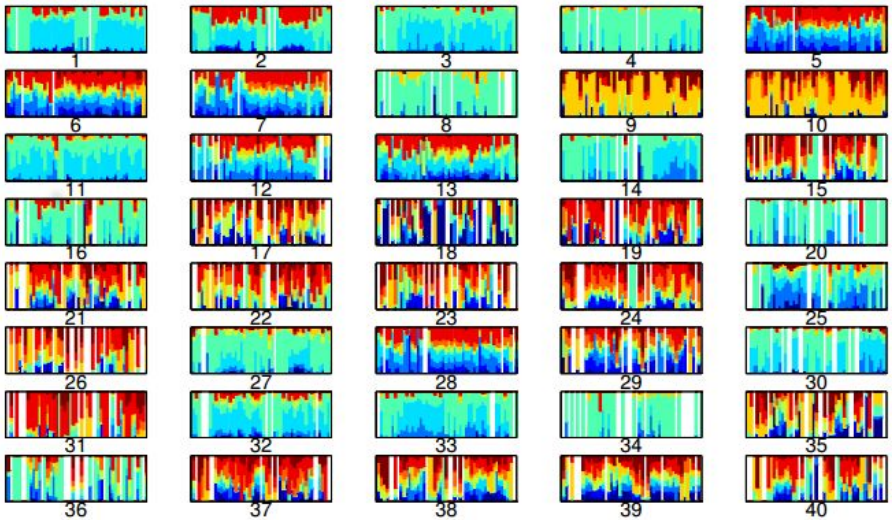
$$\mathbf{G}'_{\mathbf{t+1}} = \mathbf{G}_{\mathbf{t}}\mathbf{T}$$

Dataset	Feat.	Roles	|V|	|E|	|T|	length
TWITTER	1325	12	310K	4M	41	1 day
TWITTER-COP	150	5	8.5K	27.8K	112	3 hours
FACEBOOK	161	9	46.9K	183K	18	1 day
EMAIL-UNIV	652	10	116K	1.2M	50	60 min
NETWORK-TRA	268	11	183K	1.6M	49	15 min
INTERNET AS	30	2	37.6K	505K	28	3 months
ENRON	173	6	151	50.5K	82	2 weeks
IMDB	45	3	21.2K	296K	28	1 year
REALITY	99	5	97	31.6K	46	1 month

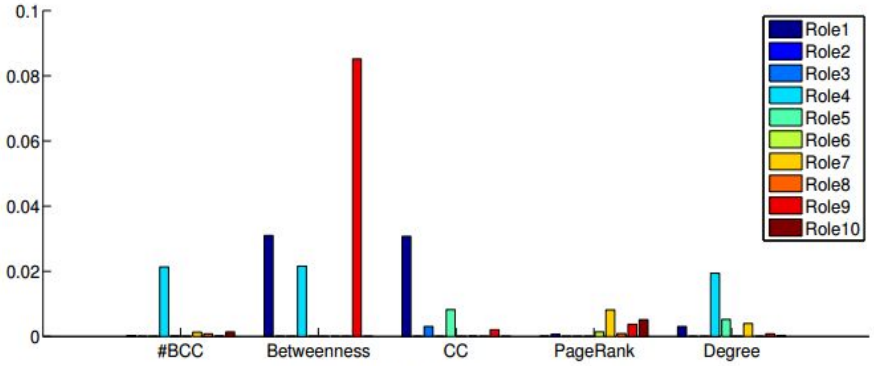
Datasets



Applying DBMM to a large IP trace network (http://www.ryanrossi.com/talks/wsdm13-dbmm-rossi.pdf)



(a) Time-evolving Mixed-Memberships (Email)

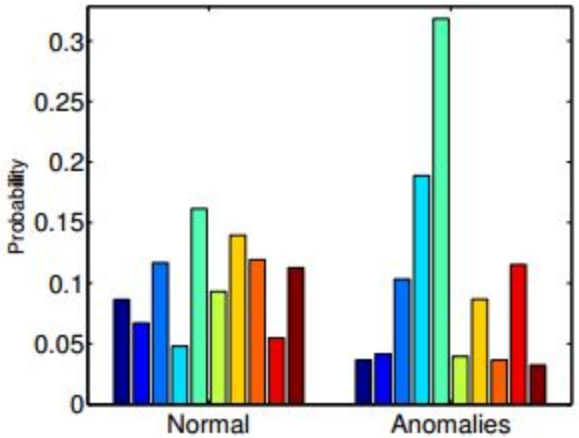


(b) Email Role Interpretation

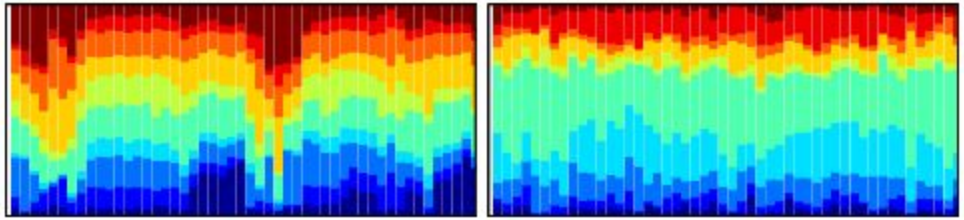
DBMM model for email network

Anomaly detector

Anomaly detector captures
in an email network



(a) Anomaly-Role Patterns



(b) Network Memberships (c) Anomaly Memberships

References

- Rossi, Ryan A., et al. "Modeling dynamic behavior in large evolving graphs."*Proceedings of the sixth ACM international conference on Web search and data mining*. ACM, 2013.
- Henderson, Keith, et al. "It's who you know: graph mining using recursive structural features." *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2011.
- Henderson, Keith, et al. "Rolx: structural role extraction & mining in large graphs." *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2012.

Mining Unstable Communities from Network Ensembles

Ahsanur Rahman, Steve Jan, Hyunju Kim, B. Aditya Prakash and T. M. Murali

Presented by: Doaa Altarawy

Outline

- Proposed work
- Definition: Unstable Community
- Definition: Subgraph divergence
- Mining Unstable Community
- Results

Proposed work

- This paper studies the opposite problem of community detection
- Propose to discover maximally **variable regions** of the graphs.
- Capture the main structural variations of the given set of networks

  → called **unstable community** (structural variations)

- Applications: in contact networks, communication networks and citation networks

Contribution

- Formalizing the concept of unstable communities (UC), a new class of problems

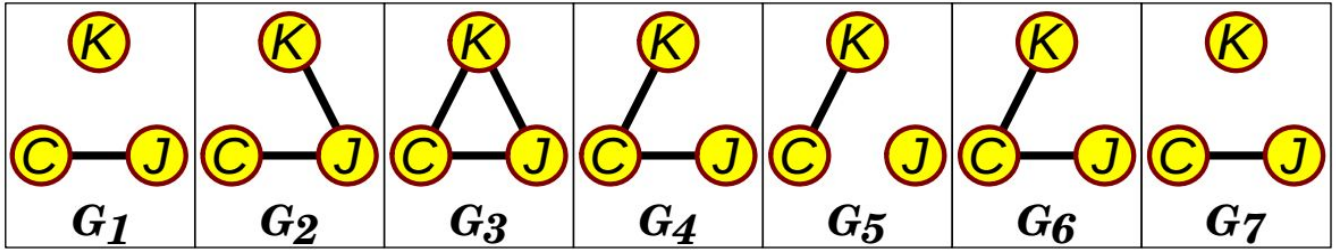
- Algorithm to find unstable communities.

- Shows how to use UC to summarize structural variations in phone calls, citations and communication networks

Definition: Unstable Community (UC)

- how many times each distinct graph among the nodes appears as a subgraph in a given ensemble of graphs (close to a uniform distribution)

- a set of nodes is a UC if the **relative entropy** between the subgraph probabilities of these nodes and the uniform distribution is at most a user-specified threshold.



Dentition: Relative entropy

- Relative entropy between p(X) and q(X):

$$R(p(X)||q(X)) = \sum_{\substack{\vec{x}\in\Omega_X, p(X=\vec{x})\neq 0 \\ q(X=\vec{x})\neq 0}} p(X=\vec{x}) \log_2 \left(\frac{p(X=\vec{x})}{q(X=\vec{x})} \right)$$

- When q(X) is uniform, i.e., : $q(X) = \frac{1}{2^{|X|}}$

$$R(p(X)) = |X| + \sum_{\substack{\vec{x}\in\Omega_X \\ p(X=\vec{x})\neq 0}} p(X=\vec{x}) \log_2 p(X=\vec{x}).$$

Subgraph divergence (SD)

- a way to measure the difference between the observed distribution of subgraphs and the uniform distribution

Subgraph divergence (SD)

- Let $\mathcal{G}$ be a set of n undirected and unweighted graphs.
- Let **G(U)** denote the subgraph of G induced by a set of nodes U.
- Let $\mathcal{G}$**(U)** ={G(U) | G ∈ $\mathcal{G}$} be multiset of subgraphs induced by U in each of the graphs in $\mathcal{G}$.
- Let **p$\mathcal{G}$(G)** be the probability for G to be present in $\mathcal{G}$(U) (the number of times *G* is a member of $\mathcal{G}$(*U*) divided by |$\mathcal{G}$|)
- Let ***P*(*U*)** denote the set of 2^{kC2} possible subgraphs on the nodes in *U*.

Subgraph divergence (SD) of U in $\mathcal{G}$, as the **relative entropy** of the probability distribution {p$\mathcal{G}$(G)), G ∈ P(U)} from the **uniform** distribution, i.e.,

$$S_{\mathcal{G}}(U) = R(p_{\mathcal{G}}(G)) = \binom{|U|}{2} + \sum_{\substack{G \in \mathcal{P}(U) \\ p_{\mathcal{G}}(G) \neq 0}} p_{\mathcal{G}}(G) \log_2 p_{\mathcal{G}}(G)$$

Scaled subgraph divergence (SSD)

Subgraph divergence depends on the size of the subgraph

$$0 \leq S_{\mathcal{G}}(U) \leq \binom{|U|}{2}$$

Alternative: scaled SD:

$$T_{\mathcal{G}}(U) = \frac{S_{\mathcal{G}}(U)}{\binom{|U|}{2}}$$

Unstable Communities using SD and SSD

1- SD-UC

- a set of nodes U is a ρ-SD-UC if its subgraph divergence $S_{\mathcal{G}}(U) \leq \rho$.

2- SSD-UC

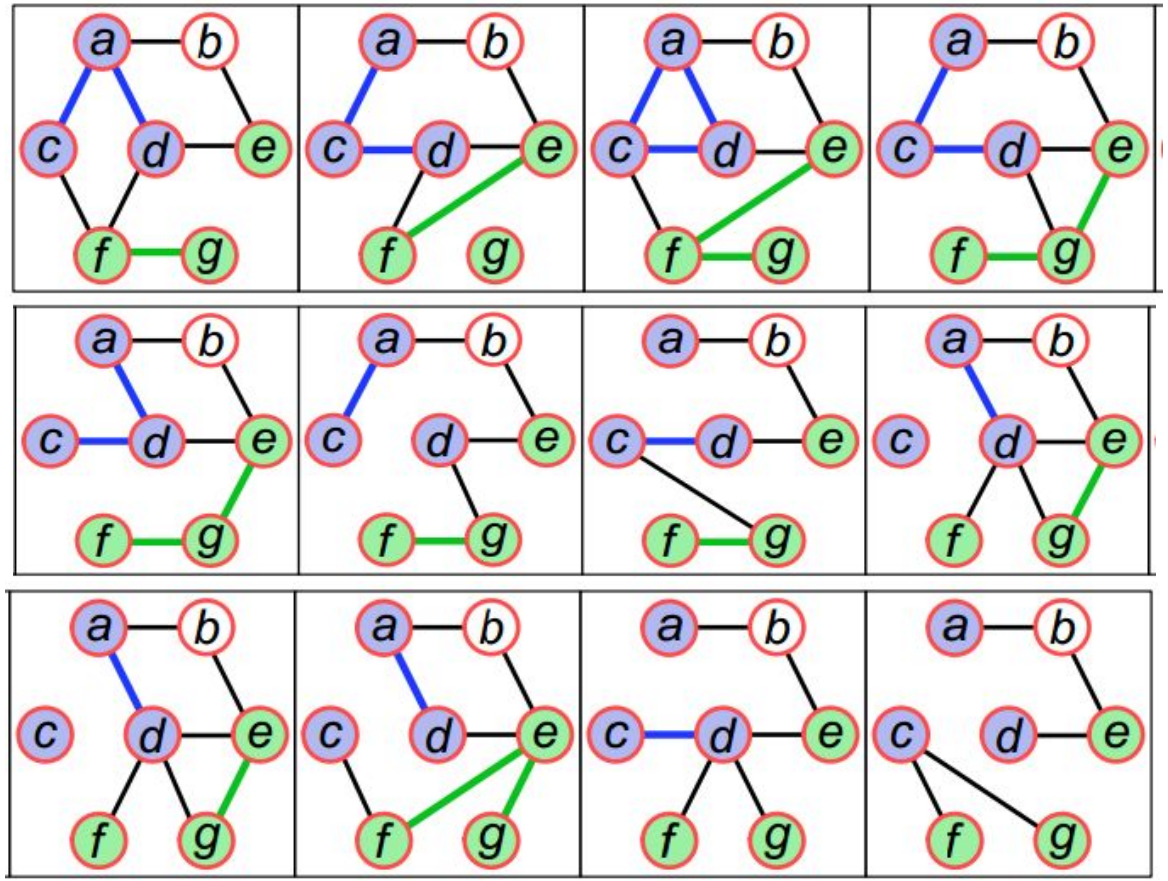
- a set of nodes U is a σ-SSD-UC if
  a. its scaled subgraph divergence $T_{\mathcal{G}}(U) \leq \sigma$ and
  b. every subset of U is a σ-SSD-UC (to be anti-monotone)

Maximal SD-UC: if no proper superset of U is a ρ-SD-UC.

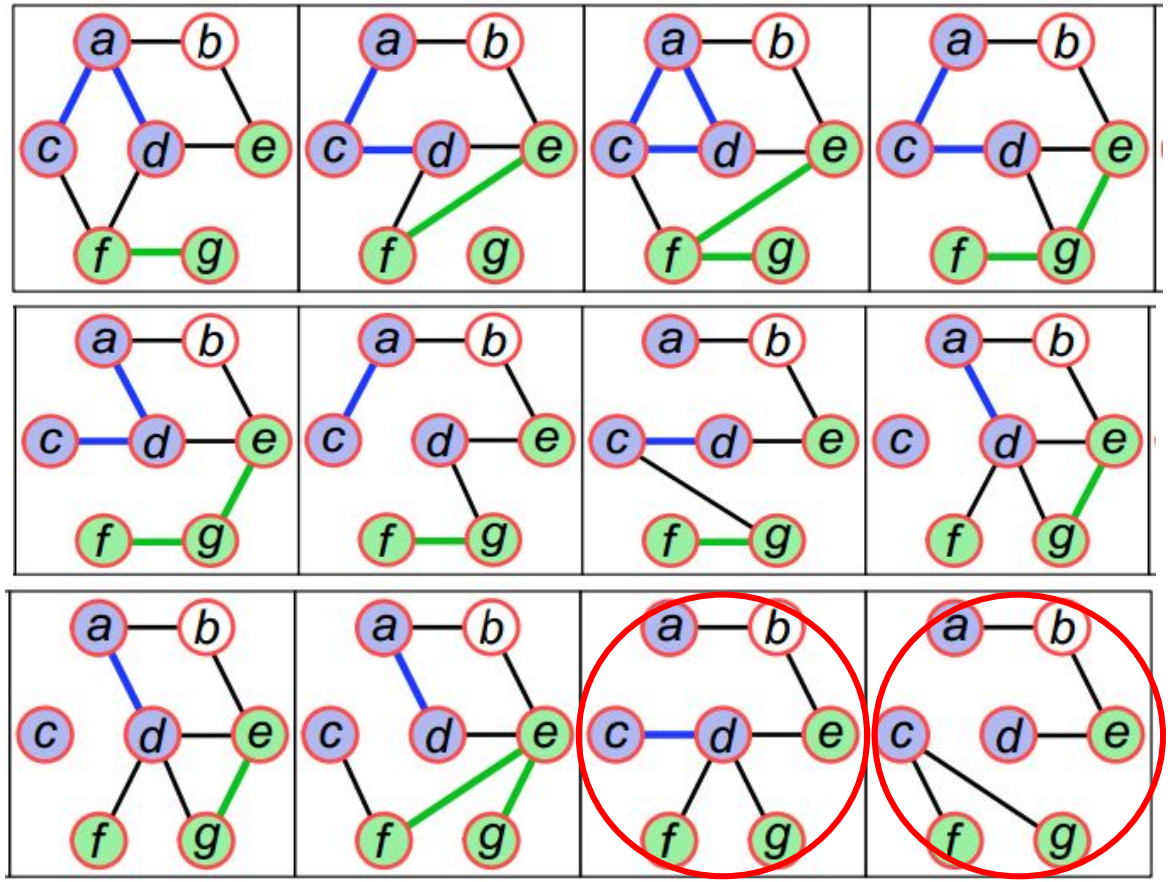
Bad SD-UC

ρ-SD-UC U is bad if it has a subset W ⊂ U such that

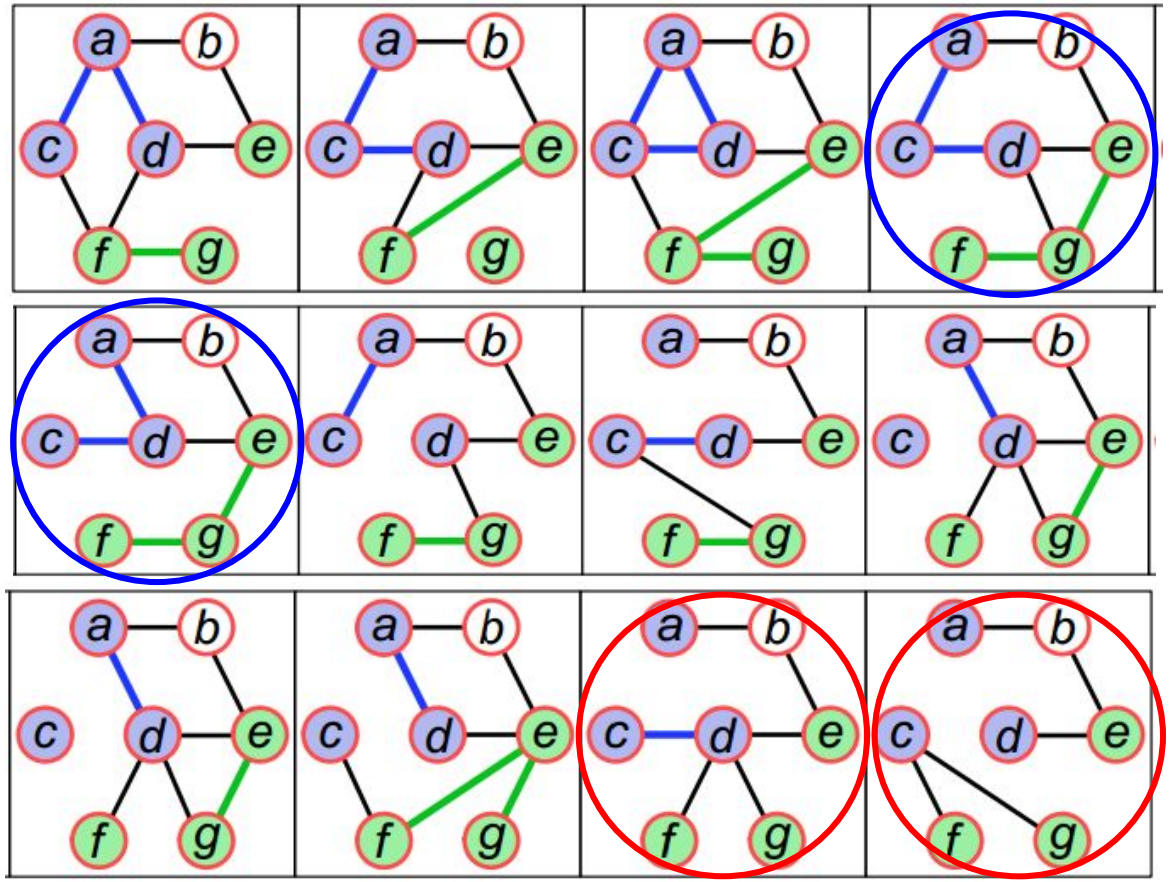
$$T_{\mathcal{G}}(W) > \rho/\binom{|U|}{2} \geq T_{\mathcal{G}}(U)$$



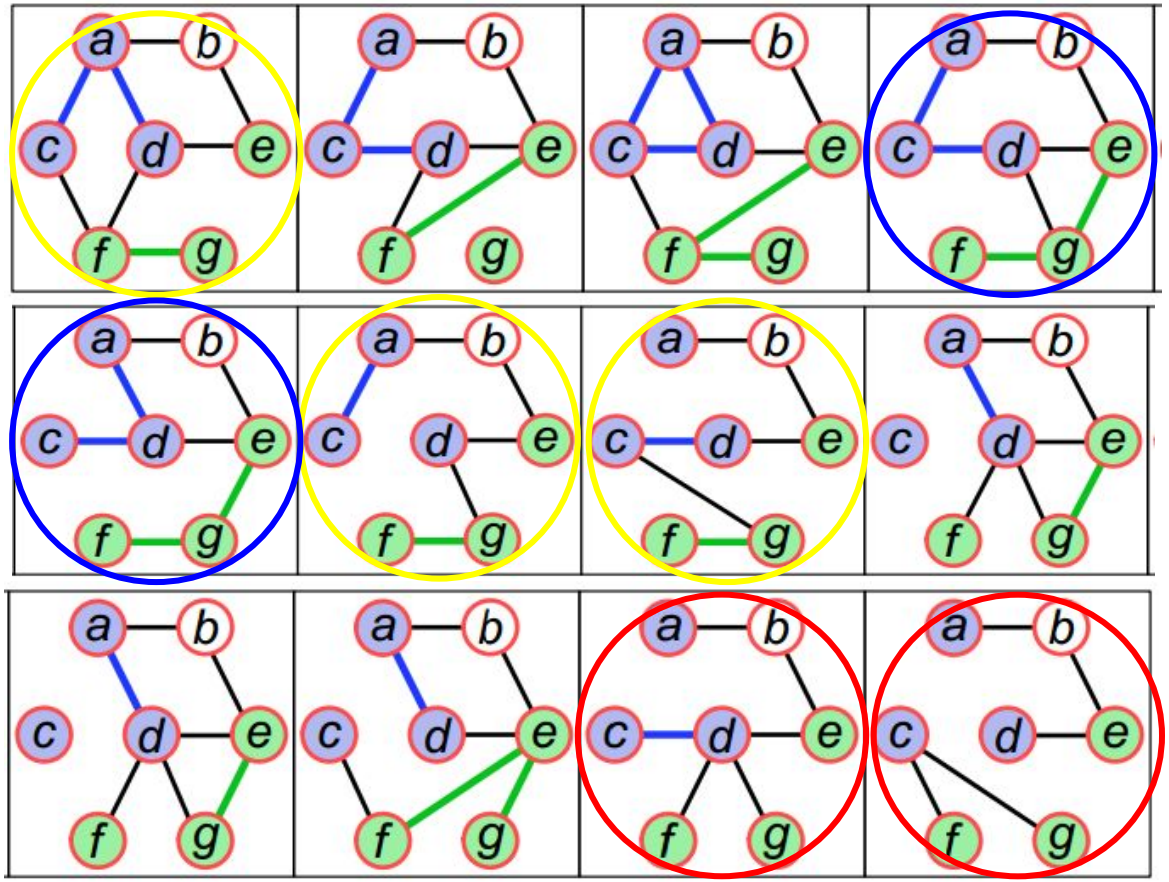
U = {e, f, g}



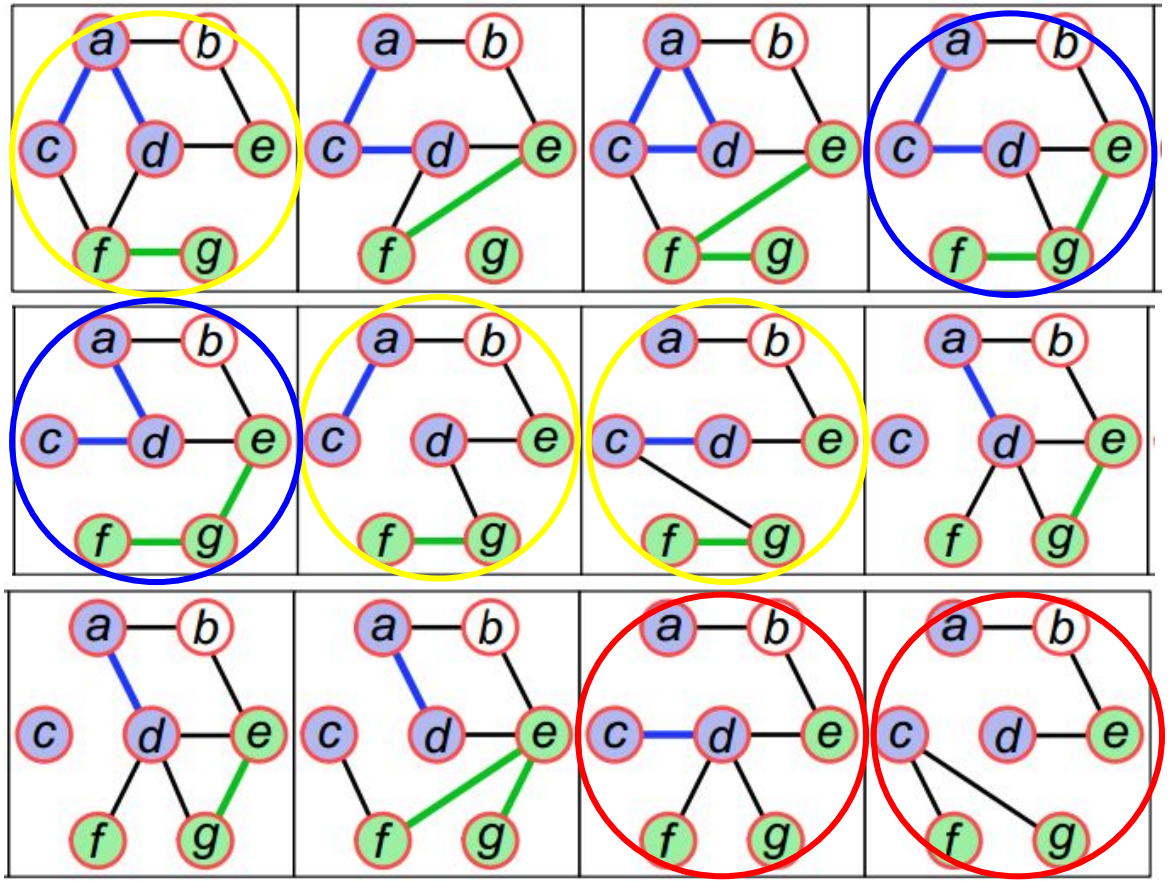
U = {e, f, g}



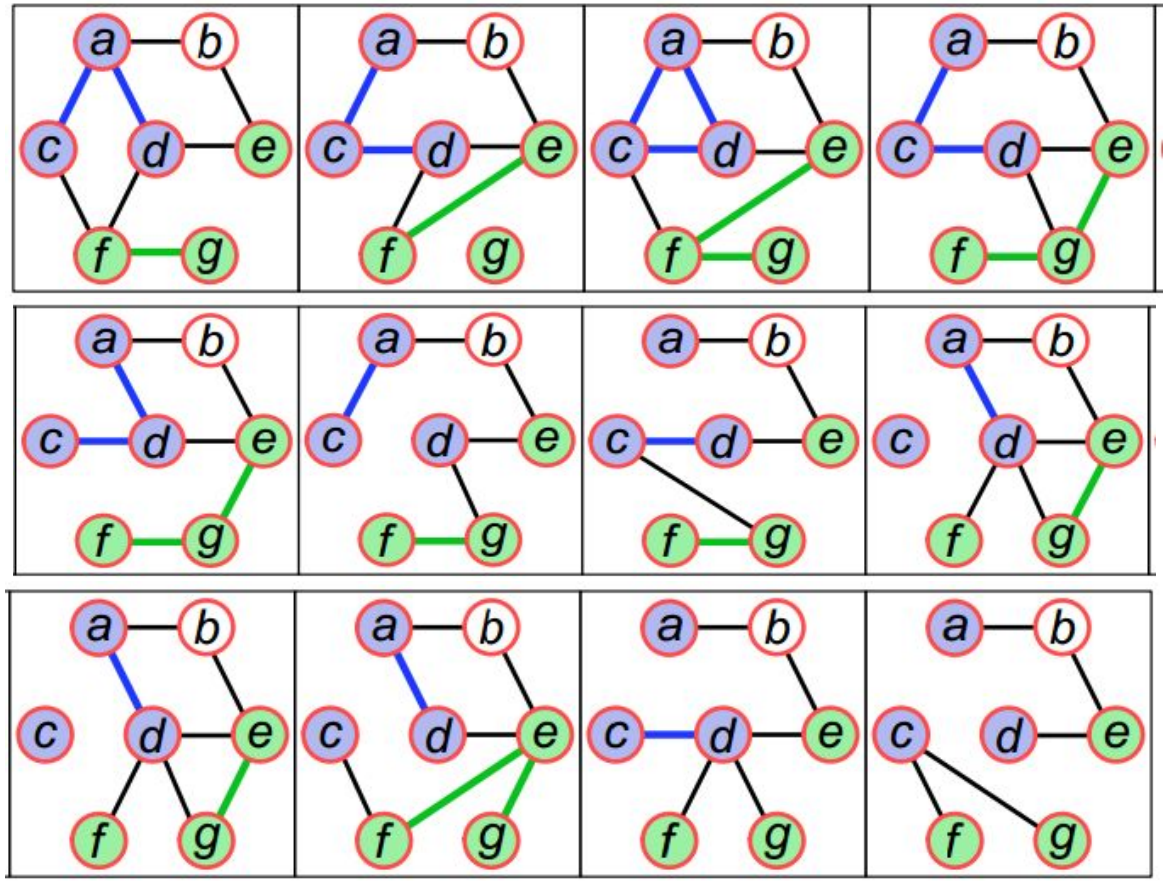
U = {e, f, g}



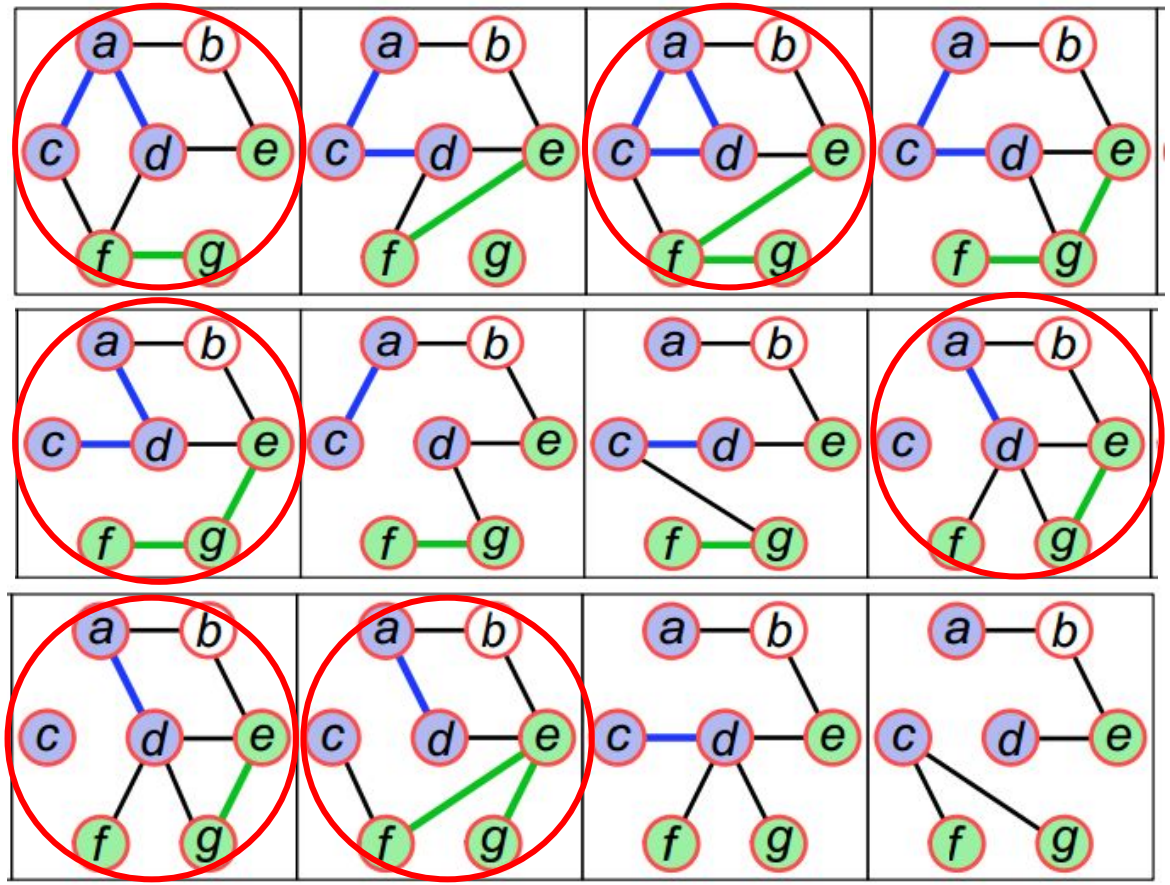
U = {e, f, g}



U = {e, f, g} is: 0.1446-SD-UC and 0.0482-SSD-UC



A = {a, b, d, e}



A = {a, b, d, e} , A is a bad 5-SD-UC (B = {a, b, e})

Mining UC

Problem 1:

Given a set of graphs $\mathcal{G}$ and a parameter $\rho \geq 0$,

enumerate all maximal ρ-SD-UCs.

Problem 2:

Given a set of graphs $\mathcal{G}$ and a parameter $0 \leq \sigma < 1$,

enumerate all maximal σ-SSD-UCs.

Lemma 1

"Let $\mathcal{G}$ be a set of graphs and U be a set of nodes.

For every node a in U, we have $S_{\mathcal{G}}(U \setminus \{a\}) \leq S_{\mathcal{G}}(U)$"

$\rightarrow$ removing a node from U does not increase its subgraph divergence. (SD-UC is anti-monotone)

Lemma 2

"Let G be a set of graphs and Q be a set of unordered node pairs. For every node pair $\{a, b\} \in Q$, $S_{\mathcal{G}}(Q \setminus \{a, b\}) \leq S_{\mathcal{G}}(Q)$"

$\rightarrow$ removing a node pair from Q does not increase its subgraph divergence.

Computing all maximal ρ-SD-UCs

Algorithm 1 COMPUTESDUCS $(\mathcal{G}, \rho)$

Require: A set $\mathcal{G}$ of graphs, $0 \leq \rho$.
Ensure: All ρ-SD-UCs.

```
S ← {(u, v) ∈ V × V | S_G({u, v}) ≤ ρ}
while S is not empty do
    T ← φ
    for every set U ∈ S do
        Compute S_G(U)
        if S_G(U) ≤ ρ then
            Output U
            Insert U into T
    S ← GENERATE-CANDIDATES(T)
```

Computing all maximal ρ-SD-UCs

Algorithm 1 COMPUTESDUCS $(\mathcal{G}, \rho)$

Require: A set $\mathcal{G}$ of graphs, $0 \leq \rho$.
Ensure: All ρ-SD-UCs.

```
S ← {(u, v) ∈ V × V | S_G({u, v}) ≤ ρ}
while S is not empty do
    T ← φ
    for every set U ∈ S do
        Compute S_G(U)
        if S_G(U) ≤ ρ then
            Output U
            Insert U into T
    S ← GENERATE-CANDIDATES(T)
```

Increase the size of the candidates in S by one

Computing all maximal ρ-SD-UCs

Algorithm 1 COMPUTESDUCS $(\mathcal{G}, \rho)$

Require: A set $\mathcal{G}$ of graphs, $0 \leq \rho$.
Ensure: All ρ-SD-UCs.

```
S ← {(u, v) ∈ V × V | S_G({u, v}) ≤ ρ}
while S is not empty do
    T ← φ
    for every set U ∈ S do
        Compute S_G(U)
        if S_G(U) ≤ ρ then
            Output U
            Insert U into T
    S ← GENERATE-CANDIDATES(T)
```

mark all parents of the UC U for deletion

Datasets

Social Evolution (SE-Prox and SE-Phone):

timestamped records of MIT reality mining repository: phone communications and proximity, 8 networks

Hospital:

temporal proximity between patients and/or staff in France, 97 networks

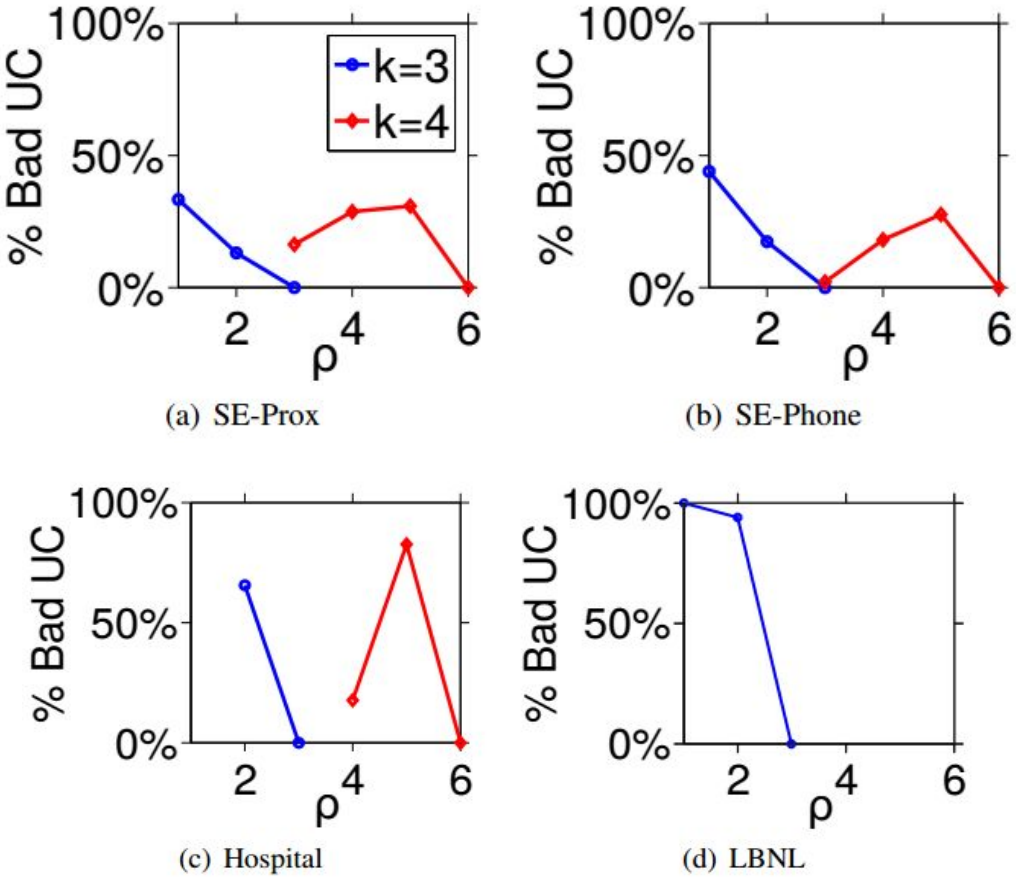
Citation network (HEP-PH):

from arxiv.org, 11 years, ~20k nodes

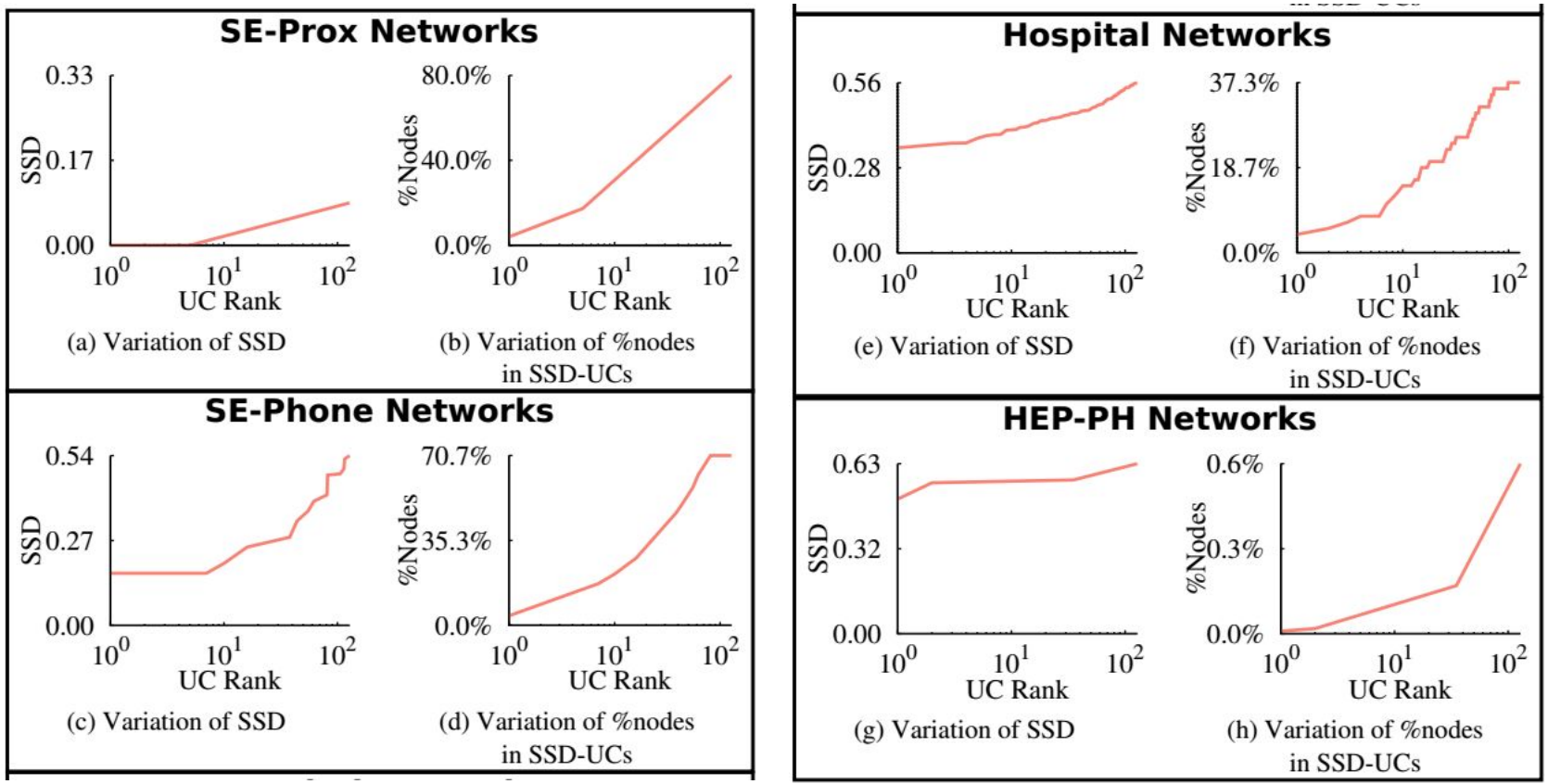
TCP (LBNL):

source and destination over time, creating 61 networks, ~2.7k nodes

Due to this weakness of SD-UCs, they focus on SSD-UCs



Percentages of bad k-node SD-UCs for different values of ρ



Capturing structural variations

Thanks
Questions?