CentraleSupélec

ESSEC BUSINESS SCHOOL

Network Science Analytics

Option Applied Math and M.Sc. in DSBA

Lecture 6B

Graph sampling and summarization

Fragkiskos Malliaros

Friday, March 9, 2018

Acknowledgements

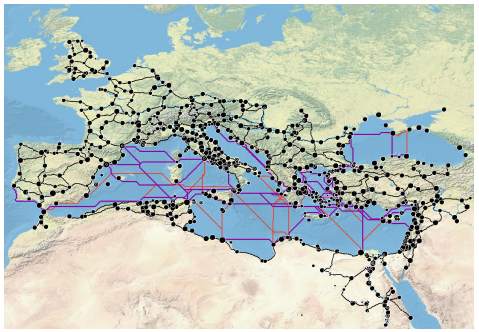
- The lecture is partially based on material by
  - Jure Leskovec, Stanford University
  - Aaron Clauset, CU Boulder
  - Manos Papaggelis, York University
  - Gonzalo Mateos, University of Rochester
  - Albert-László Barabási, Northeastern University
  - Christos Faloutsos, CMU
  - Panagiotis Tsaparas, University of Ioannina
  - Danai Koutra, University of Michigan
  - Mohammad Al Hasan (IUPUI), Nesreen K. Ahmed (Intel Labs), Jennifer Neville (Purdue University)
  - Srinivasan Parthasarathy, Ohio State University
  - R. Zafarani, M. A. Abbasi, and H. Liu, Social Media Mining: An Introduction, Cambridge University Press, 2014. Free book and slides at http://socialmediamining.info/

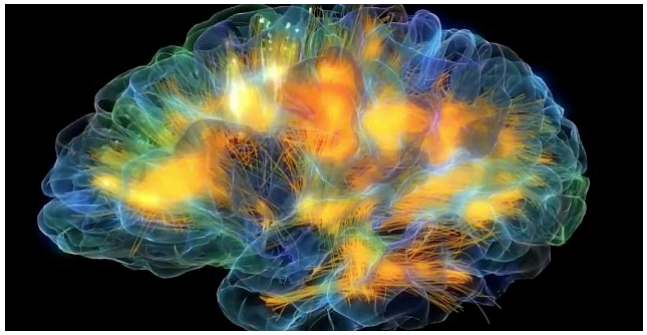
Thank you!

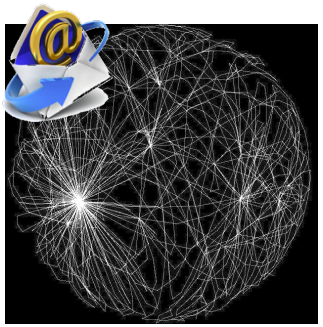
In This Lecture

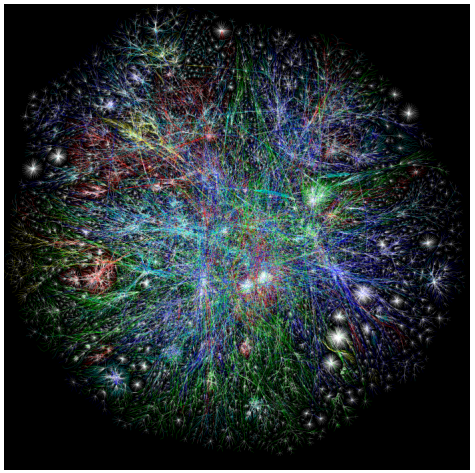
- Graph sampling
- Graph sparsification for community detection
- Graph summarization

Graph Data

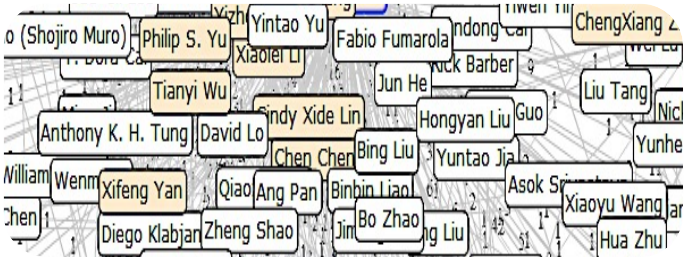




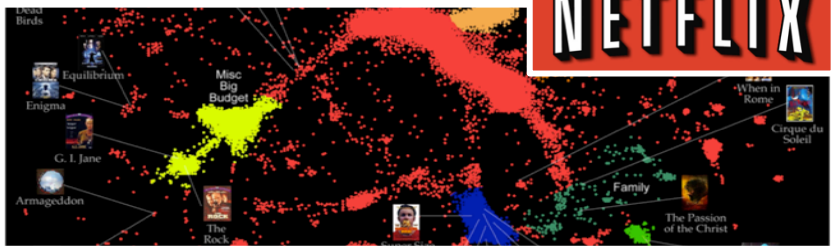




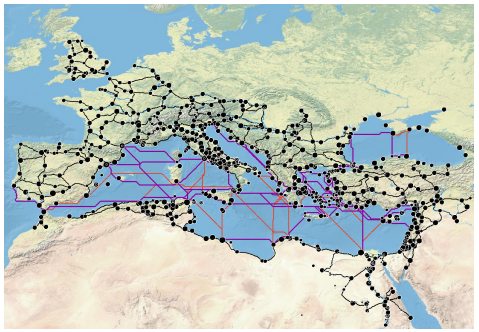
dblp

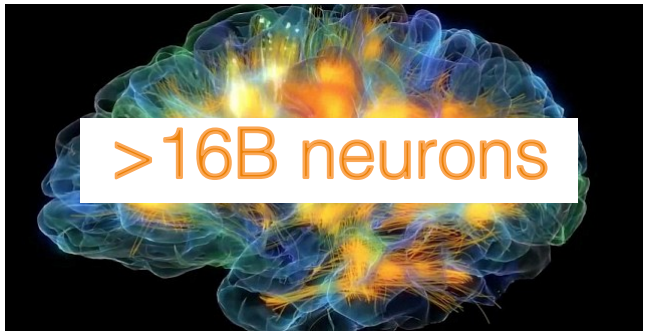


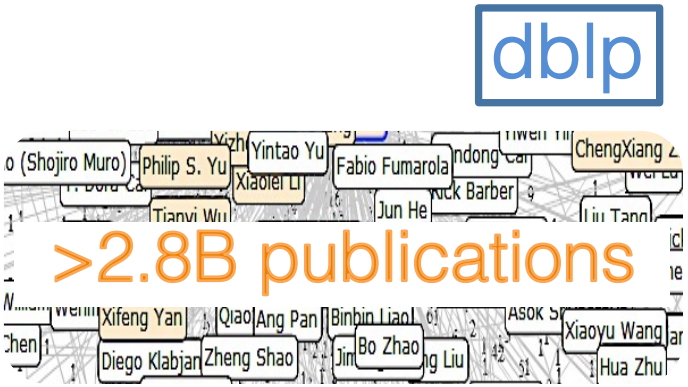
NETFLIX



Large-Scale Graph Data







An Example on Facebook

- 1.8 Billion users; avg. number of friends: 130
- It costs > 1TB memory to simply save the raw graph data (without attributes, labels or content)
- This can cause problems for information extraction, processing, and analysis
  - Computing centrality measures: $O(nm)$
  - Community detection using the Girvan-Newman algorithm: $O(m^2n)$
  - Triangle counting: $O(m^{1.41})$
  - Graphlet counting for size k: $O(n^k)$
  - Eigenvector computation: $O(n^3)$

Two possible solutions:
sampling and summarization

Overview of Sampling

- We can sample a set of nodes (or edges) and estimate the related (node or edge) properties of the original network
  - E.g., average degree and degree distribution
- Instead of analyzing the whole graph, we can sample a small subgraph similar to the original network
  - The goal is to maintain global structural characteristics of the graph
  - E.g., degree distribution, clustering coefficient, community structure
- We can also sample local substructures from the network to estimate their relative frequencies or counts
  - In many tasks, we just need to know the frequency of local structures
  - E.g., triangles for computing the clustering coefficient of the graph

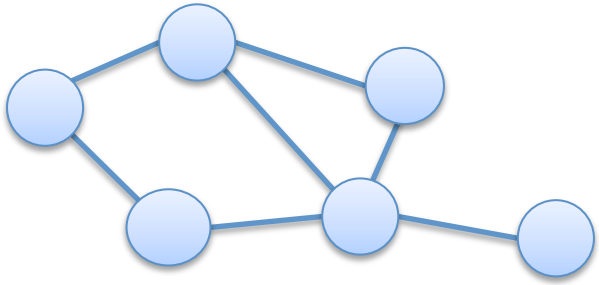
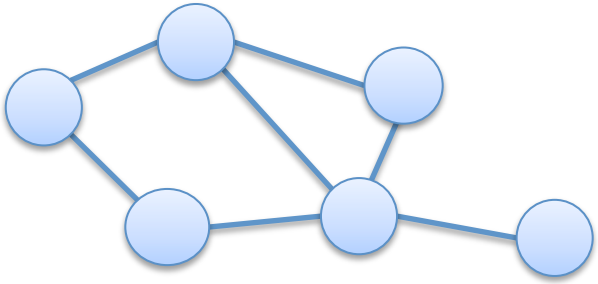
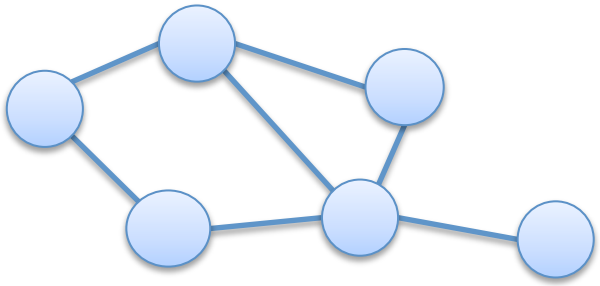
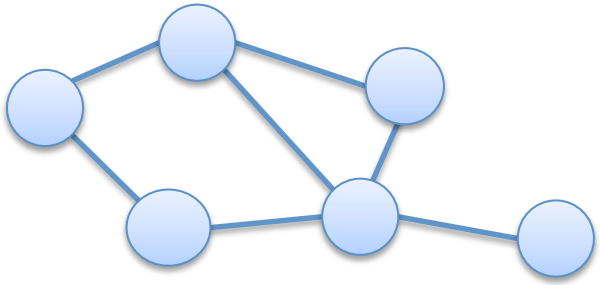
[Al Hasan, Ahmed and Neville, Tutorial at KDD '13]

Challenges

- In many cases, we have access to the entire graph
  - We can perform sampling to estimate properties of the graph (e.g., avg. degree)

- Online sampling
  - The graph is not known apriori (e.g., streaming, crawling)
  - How can one efficiently identify the most important or influential individuals without complete access to the entire network?
    - Sampling by exploration
  - This scenario arises when the network is too large

Sampling Graphs - Overview

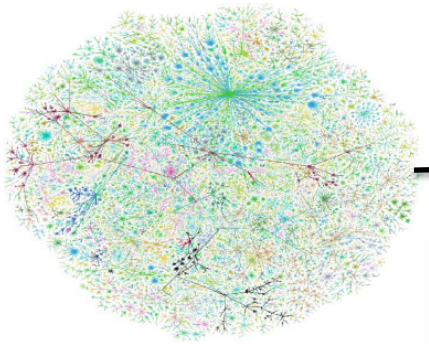
Node/edge sampling (e.g., random sampling)	Sampling by exploration
vertex sampling	BFS sampling
	
edge sampling	random walk sampling
	

Task 1

Estimating node/edge properties of the original network

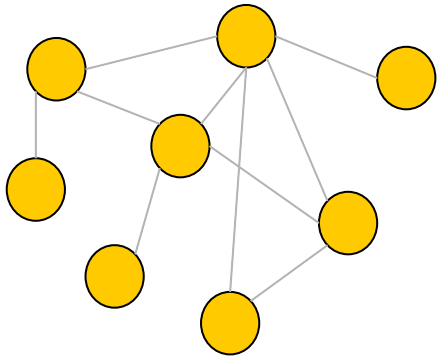
Task 1: Sample a Set of Nodes/Edges

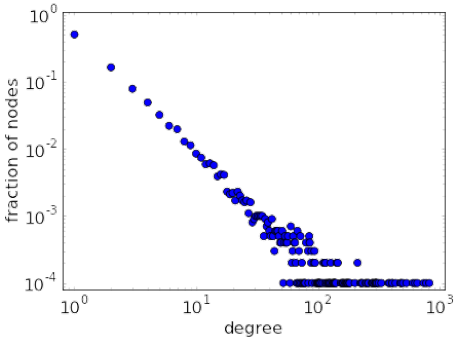
Original graph G



Sampling

Sample of nodes/edges





- Estimate network characteristics by sampling nodes or edges of the original network
- The population is the entire node set (for node sampling) and the entire edge set (for edge sampling
- Sampling with or without replacement

Sampling

- Let $G^* = (V^*, E^*)$ be the sampled graph
- Suppose a given graph characteristic or summary $\eta(G)$ is of interest
  - E.g., order $n=|V|$, size $e=|E|$, degree k_v, clustering coefficient $cl(G)$
- Typically impossible to recover $\eta(G)$ exactly from G^*
  - Q: Can we still form a useful estimate $\hat{\eta} = \hat{\eta}(G^*)$ of $\eta(G)$
- Plug-in estimator $\hat{\eta} = \eta(G^*)$
  - Many familiar estimators are of this type
  - E.g., sample means, standard deviations, covariance, quantiles

Example – Estimating Avg. Degree

- Let $G=(V, E)$ be a network of protein interactions
  - Characteristic of interest is the average degree

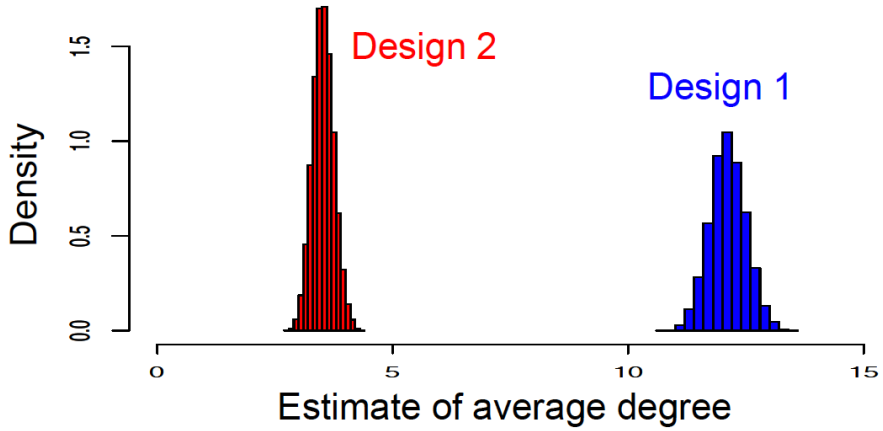
$$\eta(G) = \frac{1}{n} \sum_{i \in V} k_i$$

  - Here $n = 5{,}151$, $m = 31{,}201$ --> $\eta(G) = 12.115$
- Consider two sampling designs to obtain G^*
  - First sample c vertices $V^*=\{i_1, \ldots, i_c\}$
  - Design 1: For each $i \in V^*$, observe incident edges $(i,j) \in E$
  - Design 2: Observe edge (i, j) only when both $i, j \in V^*$
- Estimate $\eta(G)$ by averaging the observed degree sequence $\{k_i^*\}_{i \in V^*}$

$$\eta(G^*) = \frac{1}{c} \sum_{i \in V^*} k_i^*$$

Example – Estimating Avg. Degree

- Random sample of c = 1,500 nodes; Designs 1 and 2 for edges
  - Process repeated for 10,000 trials -> histogram of η(G*)



- Under-estimate η(G) for Design 2, but Design 1 performs well. Why?
  - Design 1: sample node degree explicitly, i.e., $k_i^* = k_i$
  - Design 2: (implicitly) sample node degree with bias, i.e., $k_i^* \approx \frac{c}{n} k_i$

Sampling Methods - Overview

- Random node sampling (RN)
  - Uniformly select a set of nodes
  - A sample is then a graph induced by the set of nodes
  - It has been shown that RN does not retain the power-law degree distr.
- Non uniform node sampling
  - Random degree node sampling (RDN)
    - Bias towards high degree nodes
  - Random PageRank node sampling (RPN)
    - Probability of selection a node is proportional to its PageRank score
- Random edge sampling (RE)
  - Select edges uniformly at random
- Random node-edge sampling (RNE)
  - Uniformly at random pick a node and then uniformly at random pick an edge incident to the node
- Hybrid approach (HYB)
  - With prob. p perform a step of RNE; with prob. 1-p perform a step of RE sampling

Random Node Selection (RN)

- In this strategy, a node is selected uniformly and independently from the set of all nodes
  - The sampling task is trivial if the network is fully accessible
- It provides unbiased estimates for any node property
  - E.g., Average degree and degree distribution

Random Degree Node Sampling (RDN)

- In RDN sampling, the selection of a node is proportional to its degree
  - The probability of selection node u is $\pi(u) = k(u) / 2m$
  - High degree nodes have higher chances to be selected
  - Average degree estimation is higher than actual, and the degree distribution is biased towards high-degree nodes
  - Any node property estimate is biased towards high-degree nodes

Random PageRank Node Sampling (RPN)

- RPN samples in proportion to PageRank score
  - PageRank: Visit a neighbor following an outgoing link with probability α (typically set to 0.85), and jump to a random node (uniformly) with probability 1-α
  - Node u is sampled with probability proportional to

$$\pi(u) = \alpha \frac{k^{in}(u)}{m} + \frac{1-\alpha}{n}$$

  - When α=1, the sampling is similar to Random Degree Node (RDN) sampling for the directed graph
  - When α=0, the sampling is similar to RN
  - Nodes with high in-degree have higher chances to be selected

Random Edge Selection (RE)

- Uniformly select a set of edges and the sampled networks is assumed to comprise of those edges
- A node is selected in proportion to its degree
  - If r = $|E_S|/|E|$, the probability to select node u is

$$\pi(u) = 1 - (1 - r)^{k_u}$$

r: prob. of selecting edges
Pr{not selecting node u} = $(1-r)^{k_u}$

  - When r -> 0, the probability is $\pi(u) = r \cdot k_u$
  - If we have a large sample, the degree bias is reduced
  - The selection of the nodes is not independent, as both endpoints of an edge are selected
- The properties of the nodes will be biased to high-degree nodes
- The edge statistics are unbiased due to the uniform edge selection

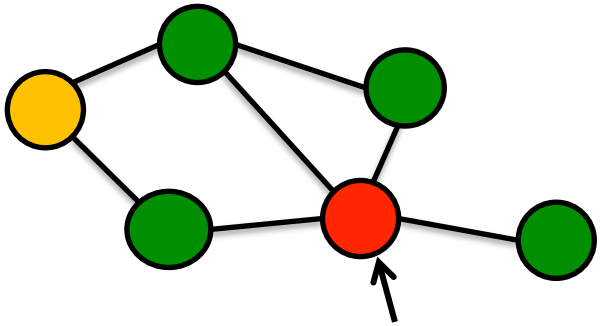
Sampling Under Restricted Access

- Assumptions
  - The network is connected and hidden, however it supports crawling
  - We have access to one seed node or a collection of seed nodes
- Methods
  - Graph traversal techniques (exploration without replacement)
    - Breadth-First Search (BFS) or Depth-First Search (DFS)
    - Forest Fire (FF)
  - Random walk techniques (exploration with replacement)
    - Random walks
    - Markov Chain Monte Carlo (MCMC) using the Metropolis-Hastings algorithm
    - Random walk with restart (RWR) or with random jump (RJ)

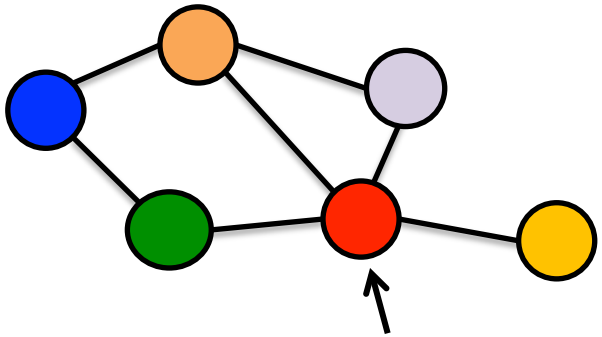
During the traversal, the visited nodes are collected in a sample set and are used to estimate network parameters

[Leskovec and Faloutsos '06]

BFS vs. DFS Sampling



BFS sampling



DFS sampling

- BFS sampling: discovers all vertices that are at distance d before discovering any vertices that is at distance d+1
- DBS sampling: discovers farthest vertex along a chain, then backtracks

Forest Fire Sampling

- Forest Fire (FF) sampling is a randomized version of BFS
  - Recall the Forest Fire model for temporal graph evolution
- Every neighbor of the current node is visited with probability p
  - If a link gets burned, the node at the other endpoint gets a chance to burn its own links, and so on recursively
- For p=1, FF sampling becomes BFS sampling
- The performance of FF sampling is similar to the one of BFS sampling

Random Walk Sampling (RWS)

- At each iteration, one of the neighbors of currently visiting node is selected to be visited
- For a selected node, the node and all its neighbors are discovered (not yet explored though)
- The sampling follows depth-first search pattern
- This sampling is biased as high-degree nodes have higher change to be sampled

Summary of Exploration-based Sampling

- Exploration-based sampling is biased toward high degree nodes
- Uniform sampling by exploration: Can we perform random walk over the nodes while ensuring that we sample each node uniformly?

- Challenges:
  - We have no knowledge about the sample space
  - At any state, only the currently visiting nodes and its neighbors are accessible

- Solution
  - Use random walk with the Metropolis-Hastings correction to accept or reject a new node

Metropolis-Hastings Random Walk Sampling

- Similar to random walk sampling (RWS), but it applies a correction so that the high degree bias of RWS is eliminated
- MH uses a distribution Q to choose one of the neighbors j of the current node i
  - If Q is uniform, $g_{ij} = 1/k_i$

This distribution defines the transition matrix of the random walk

- Then, it accepts or rejects the sampled node j with probability

$$\alpha_{ij} = \min\left\{\frac{q_{ji}}{q_{ij}}, 1\right\}$$

- Thus, if $k_j \leq k_i$, the choice is accepted with probability 1

$$\alpha_{ij} = \min\left\{\frac{k_i}{k_j}, 1\right\}$$

The random walk moves to node j with probability α_{ij}

We always accept the move towards a node of smaller degree, and reject some of the moves towards higher degree nodes

[Gjoka et al. '10]

Evaluation – Estimate Properties

Static graphs

- In- and out-degree distribution
- The distribution of sizes of weakly connected components (wcc)
- The distribution of sizes of strongly connected components (scc)
- Hop-plot: the number P(h) of reachable pairs of nodes at distance h or less
- The distribution of the first left singular vector of the graph adjacency matrix versus the rank
- The distribution of singular values of the graph adjacency matrix versus the rank
- The distribution of the clustering coefficient C_d

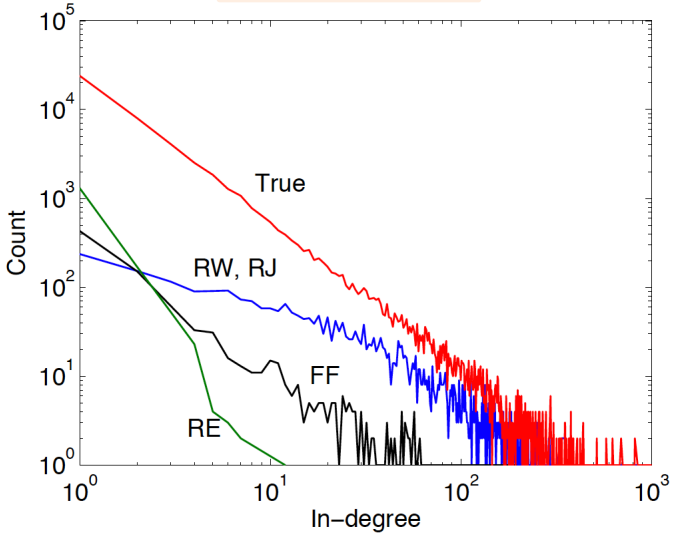
Temporal graphs

- Densification power-law (DPL)
- Diameter over time
- The normalized size of the largest connected component (CC) over time
- The largest singular value of graph adjacency matrix over time
- Average clustering coefficient C over time (C at time t is the average C_v of all nodes v in graph at time t)

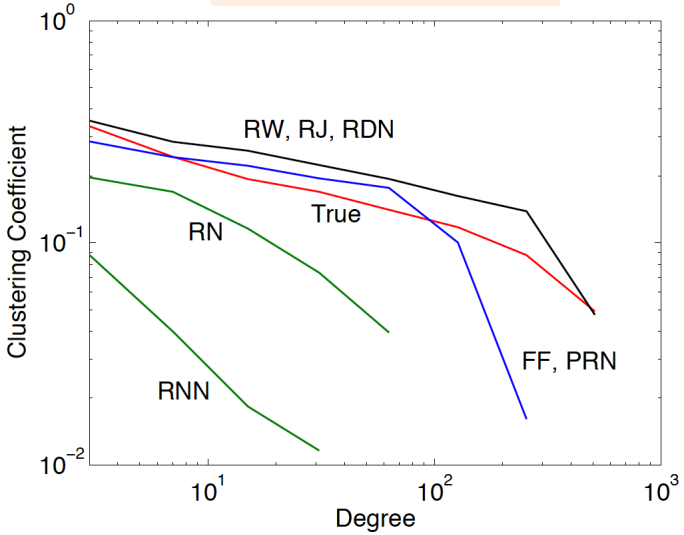
[Leskovec and Faloutsos '06]

Experimental Evaluation (1/3)

In-degree

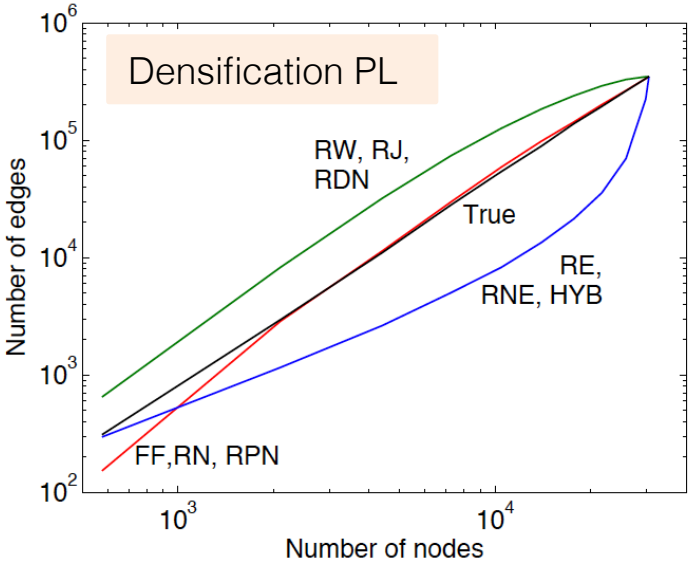


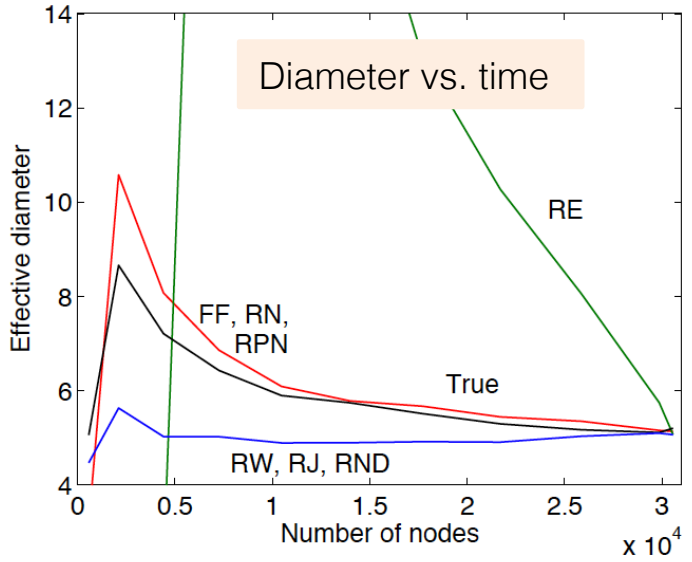
Clustering Coefficient

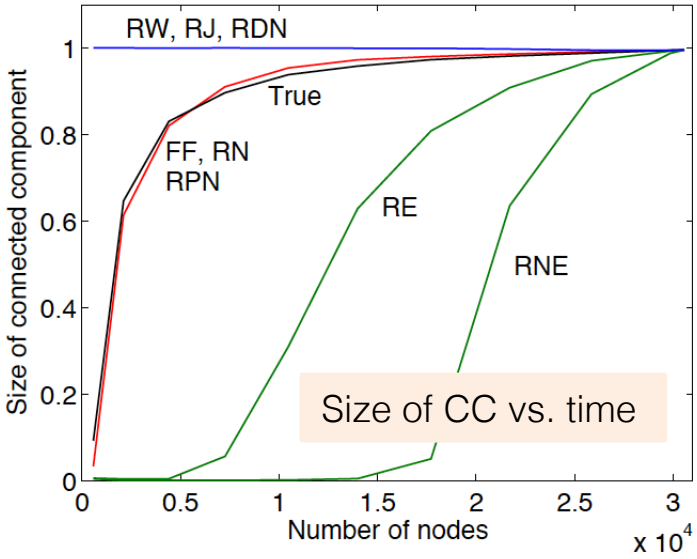


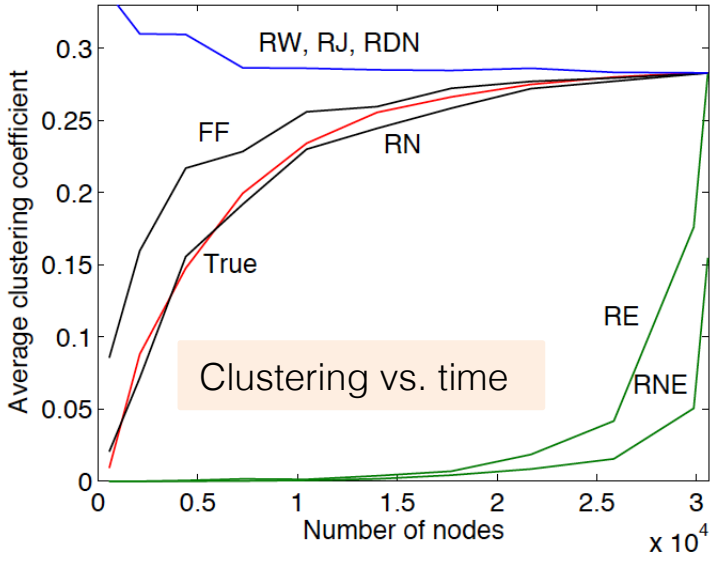
- Random node degree (RDN), random jump (RJ) and random walk (RW) have very similar behavior
  - Biased towards high degree nodes and densely connected parts of the graph
  - The tail of degree distribution is over estimated at the cost of under represented nodes with small degrees
- Forest Fire (FF), Random PageRank node sampling (RPN) and Random node (RN)
  - Not biased towards high degree nodes; similar shape for the degree distribution
  - Able to capture the shape of the clustering coeff. curve

Experimental Evaluation (2/3)









Experimental Evaluation (3/3)

	Static graph patterns								Temporal graph patterns				
	in-deg	out-deg	wcc	scc	hops	sng-val	sng-vec	clust	diam	cc-sz	sng-val	clust	**AVG**
RN	0.084	0.145	0.814	0.193	0.231	0.079	0.112	0.327	0.074	0.570	0.263	0.371	0.272
RPN	**0.062**	**0.097**	0.792	0.194	**0.200**	0.048	0.081	0.243	0.051	0.475	0.162	0.249	0.221
RDN	0.110	0.128	0.818	0.193	0.238	0.041	0.048	0.256	0.052	0.440	**0.097**	0.242	0.222
RE	0.216	0.305	**0.367**	0.206	0.509	0.169	0.192	0.525	0.164	0.659	0.355	0.729	0.366
RNE	0.277	0.404	0.390	0.224	0.702	0.255	0.273	0.709	0.370	0.771	0.215	0.733	0.444
HYB	0.273	0.394	0.386	0.224	0.683	0.240	0.251	0.670	0.331	0.748	0.256	0.765	0.435
RNN	0.179	0.014	0.581	0.206	0.252	0.060	0.255	0.398	0.058	0.463	0.200	0.433	0.258
RJ	0.132	0.151	0.771	0.215	0.264	0.076	0.143	**0.235**	0.122	0.492	0.161	**0.214**	0.248
RW	0.082	0.131	0.685	0.194	0.243	0.049	**0.033**	**0.243**	**0.036**	**0.423**	**0.086**	0.224	**0.202**
FF	0.082	0.105	0.664	0.194	**0.203**	**0.038**	0.092	**0.244**	0.053	0.434	0.140	**0.211**	**0.205**

Each entry of the table is obtained by averaging the D-statistic (distance between distributions) over 10 runs, 5 datasets, and 20 sample sizes per dataset

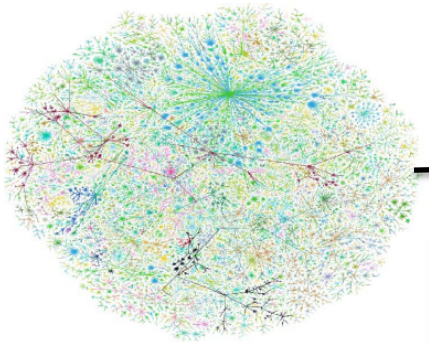
- Distribution of weakly connected components is best matched by edge selection techniques
- Random walks (RW) and Forest Fire (FF) sampling perform well, in general
- No single perfect answer to graph sampling
  - Choose the appropriate method depending the application domain and the properties of interest

Task 2

Representative subgraph sampling

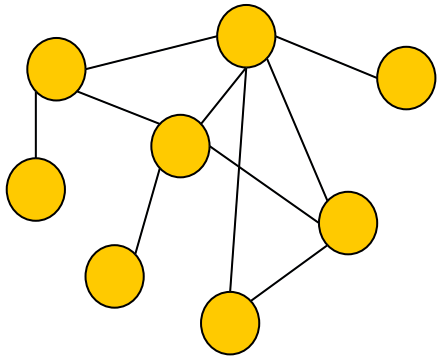
Task 2: Representative Subgraph Sampling

Original graph G



Sampling

Subgraph S



- Most of the previously presented techniques can also be applied here
  - Sample nodes/edges and take the induced subgraph
- We are mostly interested for subgraphs that are representative of the structure of the graph
  - Capture the community structure

The sampled subgraph should be representative of the whole graph

Motivation

- Traditional graph sampling approaches would lead to random sample of nodes
  - The induced subgraph on the sampled nodes may be a collection of disconnected isolated singletons
  - Essentially useless for any meaningful analysis

- Task: sample a subgraph that is representative of the community structure in the original network
  - The sample will contain nodes from the "major" communities of the graph
  - Why is this useful? The subgraph samples can be used to infer the community affiliation of nodes not present in the sample

Expansion Sampling – The Idea

- The expansion factor X(S) of a sample S is defined as

$$X(S) = \frac{|N(S)|}{|S|}$$

N(S): neighbors of set S

  - It captures similar structural properties as the measures of conductance and cut
  - Recall that, in community detection we are interested in subsets S that minimize the expansion X(S)
  - Samples with good expansion properties tend to be more representative of the community structure

- Here we are interested in finding the sample S of size k with the maximum expansion factor

$$\arg\max_{S:|S|=k} = \frac{|N(S)|}{|S|}$$

[Maiya and Berger-Wolf '10]

Expansion Sampling – The Algorithm

- Input: Graph G = (V, E)
  k, the sample size

1. S = 0
2. V = *random* (v)
3. $S = S \cup \{v\}$
4. while |S| ≤ k do
5. Select new node $v \in N(S)$ based on the maximization of

$$|N(\{v\}) - (N(S) \cup S)|$$

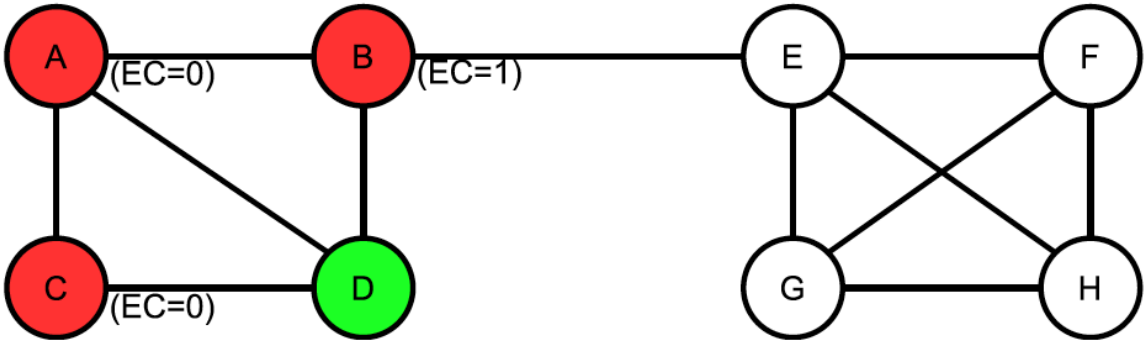
6. $S = S \cup \{v\}$
7. end while
8. Return S

Choose first node v at random

How node v contributes to the expansion factor of the currently constructed sample S

[Maiya and Berger-Wolf '10]

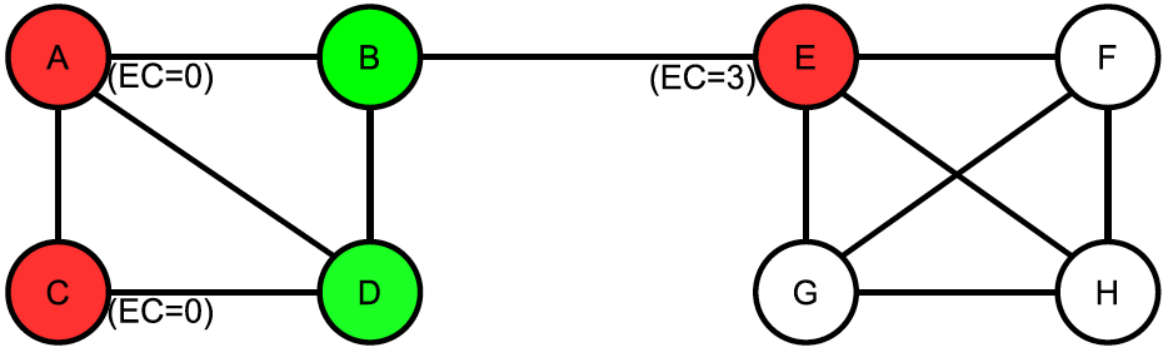
Example



Step 1:

- D is added to S
- S = {D}, N(S) = {A, B, C}

EC: Expansion contribution



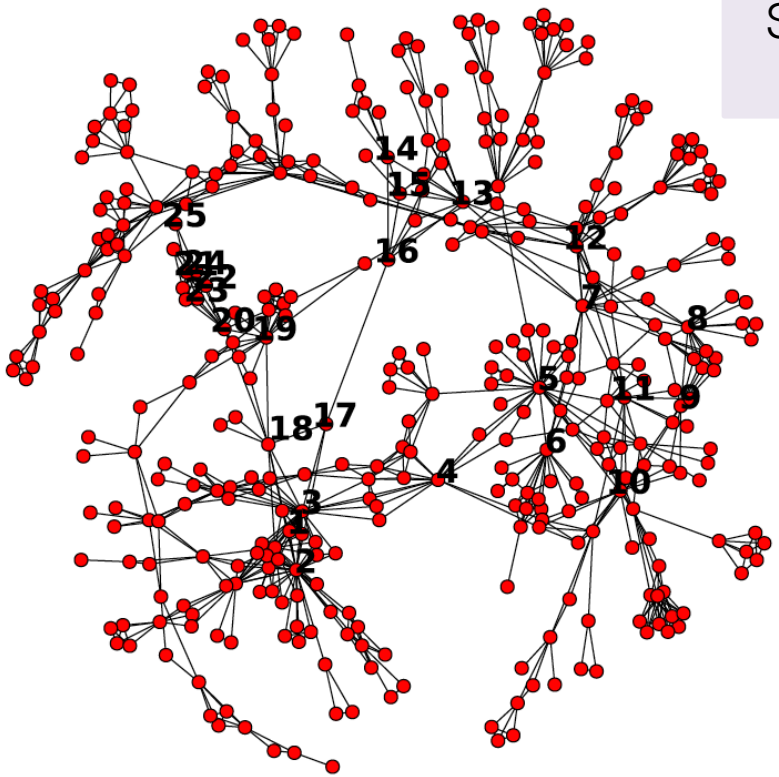
Step 2:

- B is added to S
- S = {D, B},
  N(S) = {A, C, E}

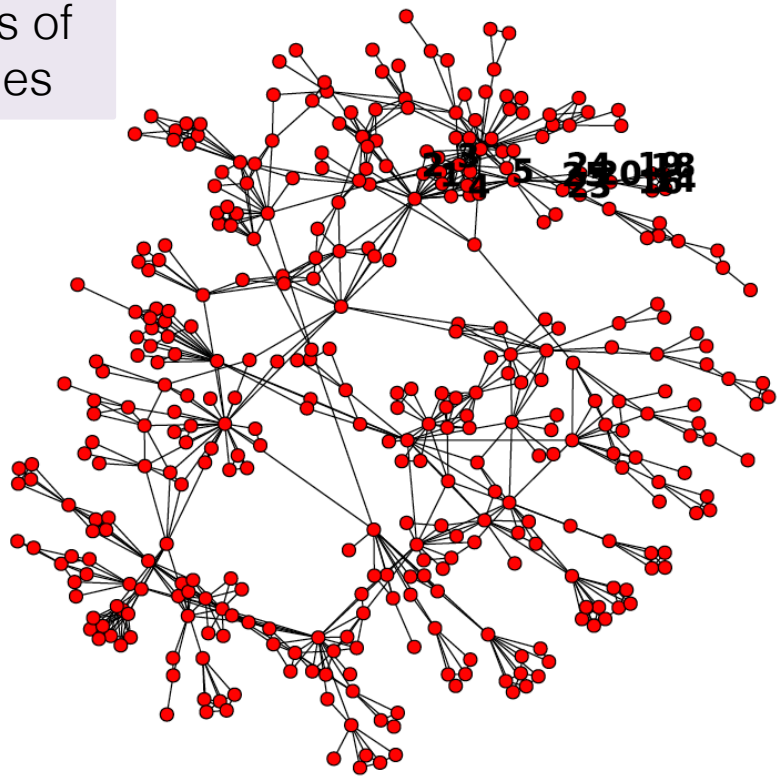
[Maiya and Berger-Wolf '10]

Expansion vs. Random Walk Sampling

Samples of 25 nodes



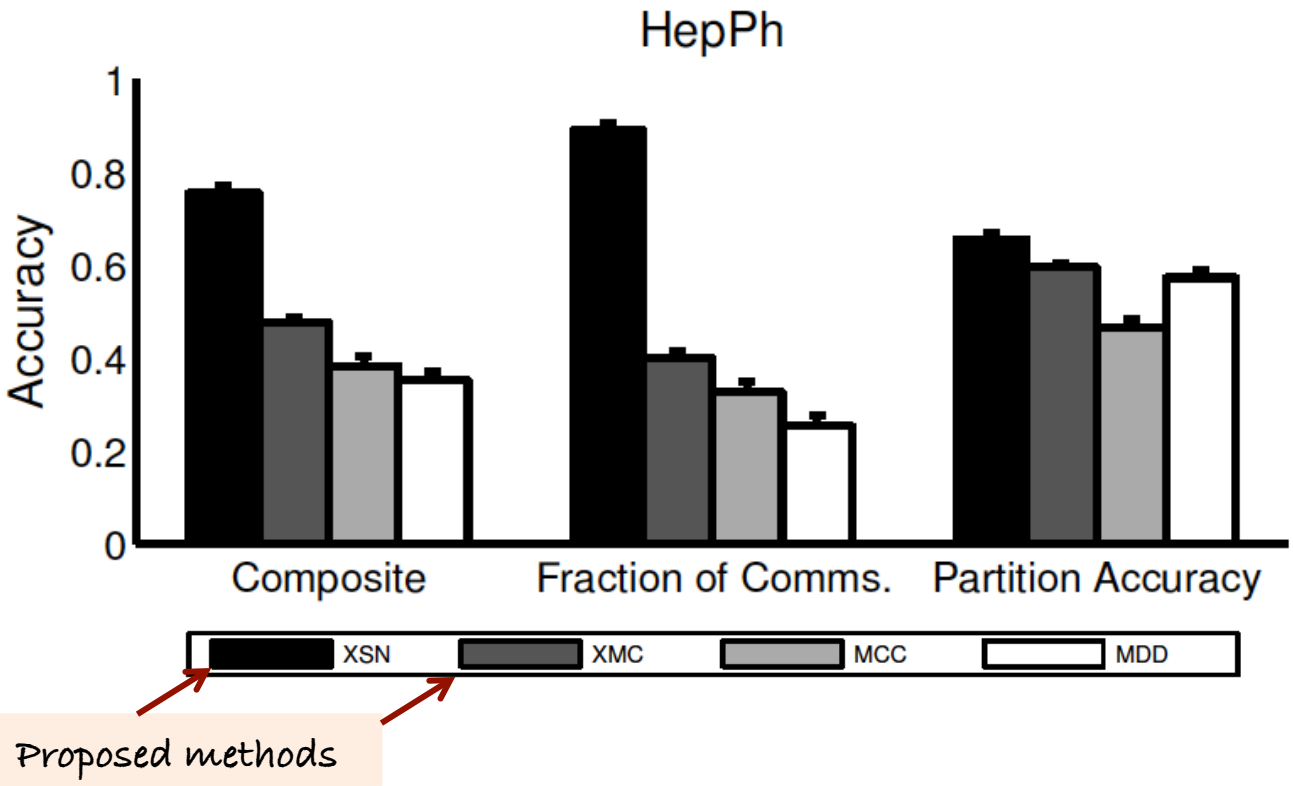
Expansion sampling



(Self avoiding) Random walk sampling

The expansion sampling strategy explores a wider portion of the graph and more clusters

Experiments – Community Representativeness



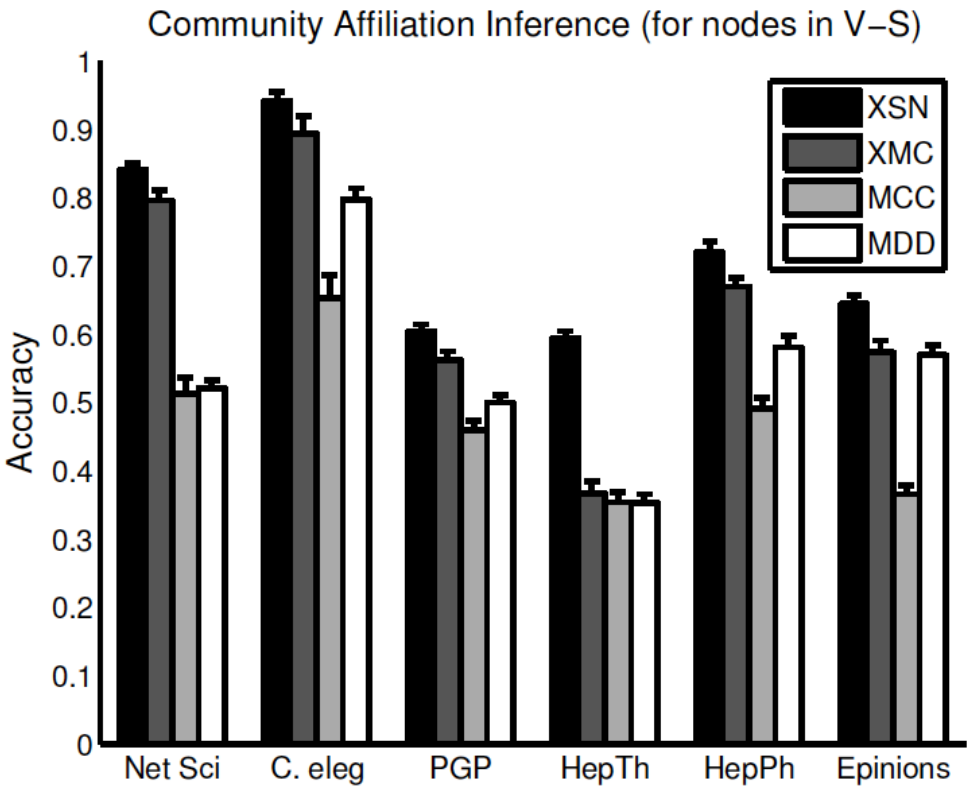
Baselines:

- MCC: Metropolis Using Clustering Coefficient
- MDD: Metropolis Using Degree Distribution

- Examine if nodes grouped together in the sample's community structure are also grouped together in the community structure of original network

[Maiya and Berger-Wolf '10]

Experiments – Community Affiliation Inference



- Use the samples to infer the community affiliation of nodes not present in the sample
  - Use collective inference techniques

[Maiya and Berger-Wolf '10]

Graph sparsification

for community detection

Graph Sparsification

- Graph sampling: produce a graph with both fewer nodes and edges than the original one
  - Preserve some properties of interest (e.g., degree distribution)
- Sampling representative subgraphs
  - Representative of the community structure of the graph

- Graph sparsification
  - Make the graph sparser by reducing the number of edges
  - Same number of nodes
  - Preserve a set of properties (e.g., spectral properties)

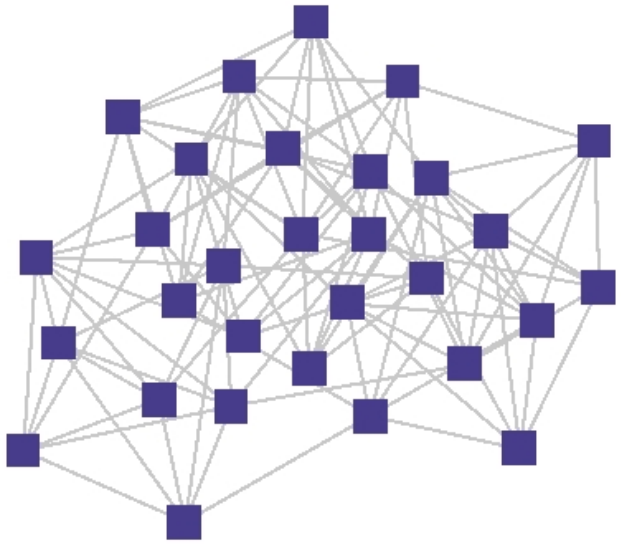
Graph Sparsification for Clustering

Is there a simple pre-processing of the graph to reduce the edge set that can "clarify" or "simplify" its clustering structure?

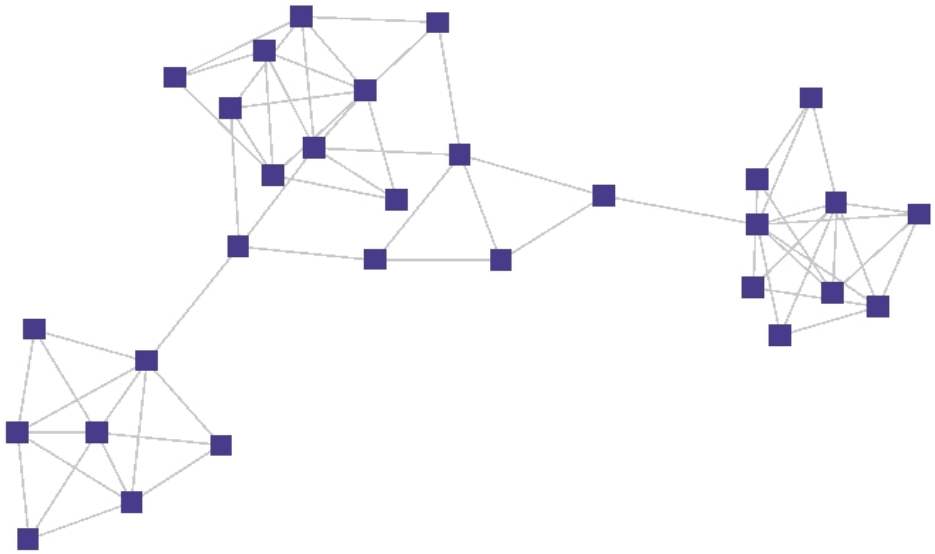
Sparsification for clustering:

1. Clustering the sparsified graph should be much faster than clustering the original graph
2. The quality of the clusters of the sparsified graph is close to the one from the original graph
3. The sparsification method itself should be fast

Original vs. Sparsified Graph



Original



Sparsified

[Automatically visualized using Prefuse]

Run any community detection algorithm on the sparsified graph

[Satuluri et al '11]

Similarity Sparsification

- Preferentially retain edges that are likely to be part of the same cluster
  - How to identify those edges?

- An edge (i, j) is likely to (not) lie within a cluster if nodes i and j have many (few) common neighbors
- Use the Jaccard measure to quantify the overlap between neighbors

$$\text{Sim}(i, j) = \frac{|N(i) \cap N(j)|}{|N(i) \cup N(j)|}$$

Set of neighbors of node i

[Satuluri et al '11]

Global Graph Sparsification Algorithm

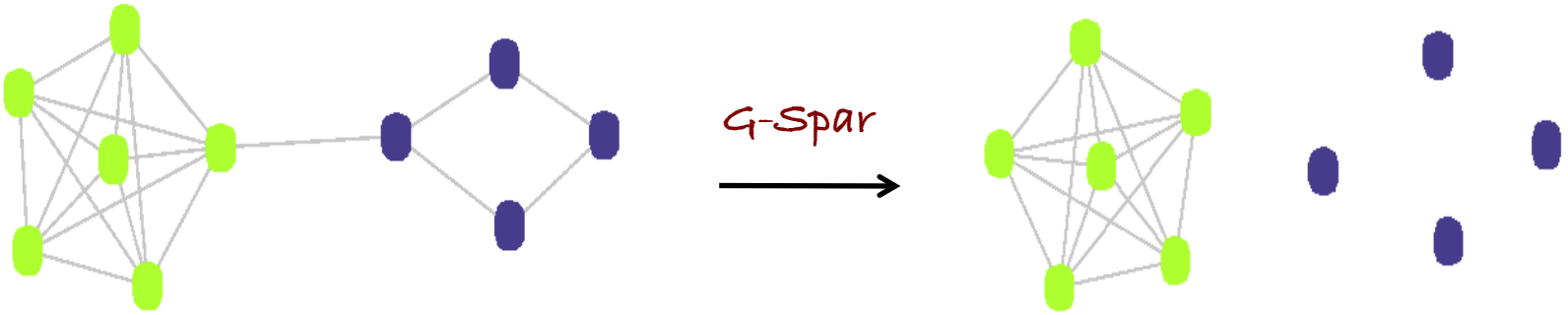
Algorithm G-Spar

- Input: graph G = (V, E)
  sparsification ratio s
  Output: the sparsified graph G_{sparse}

1. G_{sparse} <- { }
2. for each edge e = (i, j) in E do
3. Compute sim(i, j) based on the Jaccard measure
4. end for
5. Sort all edges in E by their similarity
6. Add the top s% of those edges to G_{sparse}

[Satuluri et al '11]

Small Clusters are Underrepresented



Original graph

Sparsified graph

- Dense clusters are overrepresented, sparse clusters underrepresented
- Works great when the goal is to just find the top communities

Local Graph Sparsification Algorithm

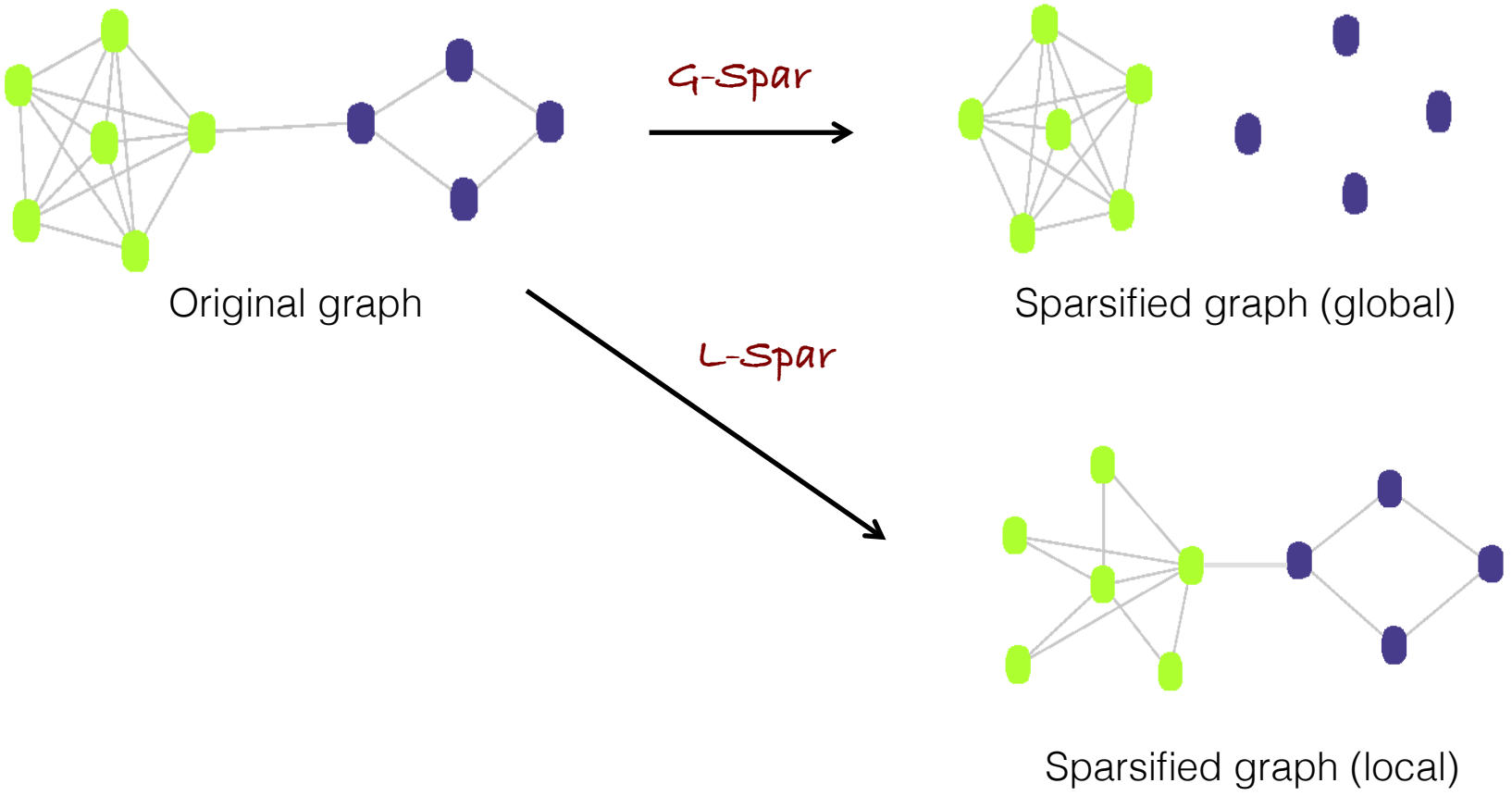
Algorithm L-Spar

- Input: graph $G = (V, E)$
  local sparsification exponent e
  Output: the sparsified graph G_{sparse}

1. G_{sparse} <- { }
2. for each node i in V do
3. Let k_i be the degree of i and E_i be the set of edges incident to i
4. for each edge (i, j) in E_i do
5. Compute $sim(i, j)$ based on the Jaccard measure
6. end for
7. Sort all edges in E_i by their similarity
8. Add the top k_i^e of those edges to G_{sparse}
9. end for

[Satuluri et al '11]

Global vs. Local Graph Sparsification



[Satuluri et al '11]

Experiments

- Datasets
  - 3 PPI networks (BioGrid, DIP, Human)
  - 2 Information (Wiki, Flickr) & 2 Social (Orkut , Twitter) networks
  - Largest network (Orkut), roughly a Billion edges
  - Ground truth available for PPI networks and Wiki
- Clustering algorithms
  - Metis [Karypis & Kumar '98], MLR-MCL [Satuluri & Parthasarathy, '09], Metis+MQI [Lang & Rao '04], Graclus [Dhillon et. al. '07], Spectral methods [Shi '00], Edge-based agglomerative/divisive methods [Newman '04]
- Compared sparsifications
  - L-Spar, G-Spar and RandomEdge

Code: https://sites.google.com/site/stochasticflowclustering/graphpreprocessing

Results using Metis

Dataset (n, m)	Spars. Ratio	Random		G-Spar		L-Spar	
		Speed	Quality	Speed	Quality	Speed	Quality
Yeast_Noisy (6k, 200k)	17%	11x	-10%	30x	-15%	25x	+11%
Wiki (1.1M, 53M)	15%	8x	-26%	104x	-24%	52x	+50%
Orkut (3M, 117M)	17%	13x	+20%	30x	+60%	36x	+60%

[Satuluri et al '11]

[Hardware: Quad-core Intel i5 CPU, 3.2 GHz, with 16GB RAM]

Results using Metis

Dataset (n, m)	Spars. Ratio	Random		G-Spar		L-Spar	
		Speed	Quality	Speed	Quality	Speed	Quality
Yeast_Noisy (6k, 200k)	17%	11x	-10%	30x	-15%	25x	+11%
Wiki (1.1M, 53M)	15%	8x	-26%	104x	-24%	52x	+50%
Orkut (3M, 117M)	17%	13x	+20%	30x	+60%	36x	+60%

Same sparsification ratio for all 3 methods

Results using Metis

Dataset (n, m)	Spars. Ratio	Random		G-Spar		L-Spar	
		Speed	Quality	Speed	Quality	Speed	Quality
Yeast_Noisy (6k, 200k)	17%	11x	-10%	30x	-15%	25x	+11%
Wiki (1.1M, 53M)	15%	8x	-26%	104x	-24%	52x	+50%
Orkut (3M, 117M)	17%	13x	+20%	30x	+60%	36x	+60%

Good speedups, but typically loss in quality

Results using Metis

Dataset (n, m)	Spars. Ratio	Random		G-Spar		L-Spar	
		Speed	Quality	Speed	Quality	Speed	Quality
Yeast_Noisy (6k, 200k)	17%	11x	-10%	30x	-15%	25x	+11%
Wiki (1.1M, 53M)	15%	8x	-26%	104x	-24%	52x	+50%
Orkut (3M, 117M)	17%	13x	+20%	30x	+60%	36x	+60%

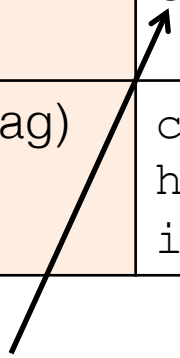
Great speedups and quality

L-Spar: Results Using MLR-MCL

Dataset (n, m)	Spars. Ratio	L-Spar	
		Speed	Quality
Yeast_Noisy (6k, 200k)	17%	17x	+4%
Wiki (1.1M, 53M)	15%	23x	-4.5%
Orkut (3M, 117M)	17%	22x	0%

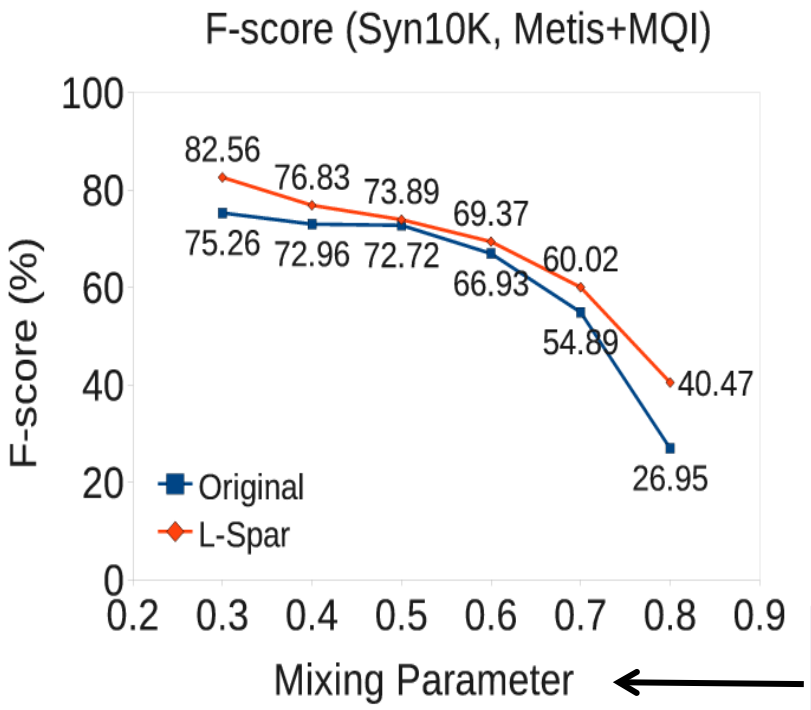
L-Spar: Qualitative Examples

Node	Retained neighbors	Discarded neighbors
`Graph` (Wiki article)	`Graph Theory, Adjacency list, Adjacency matrix, Model theory`	`Tessellation,Roman letters used in Mathematics, Morphism`
`Jack Dorsey` (Twitter user, and co-founder)	`Biz Stone, Evan Williams, Jason Goldman, Sarah Lacy`	`Alyssa Milano, JetBlue Airways, WholeFoods, Parul Sharma`
`Gladiator` (Flickr tag)	`colosseum, world-heritage, site, italy`	`europe, travel, canon, sky, summer`



Twitter executives, Silicon Valley figures

Impact of Sparsification on Noisy Data

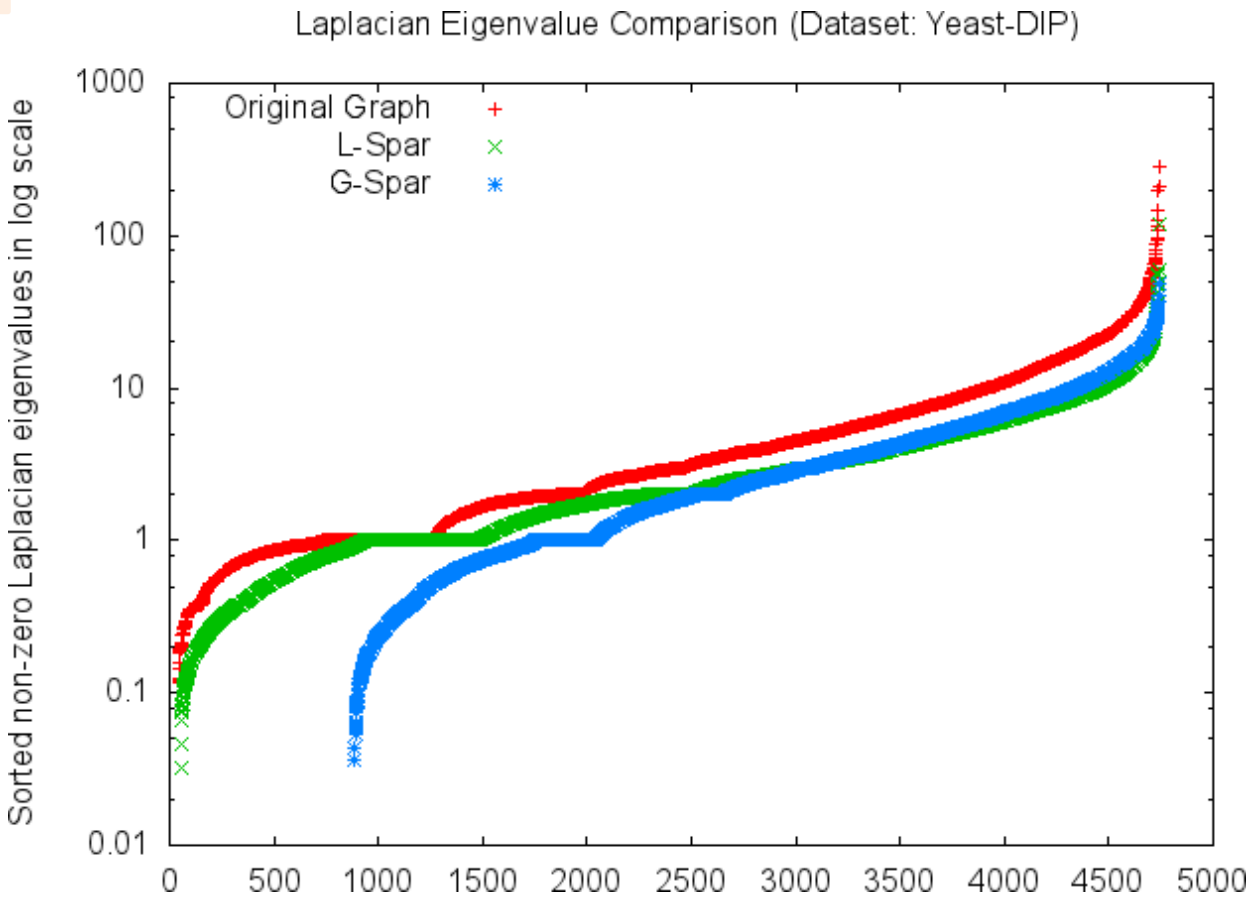


Add edges between different clusters

As the graphs get noisier, L-Spar is increasingly beneficial

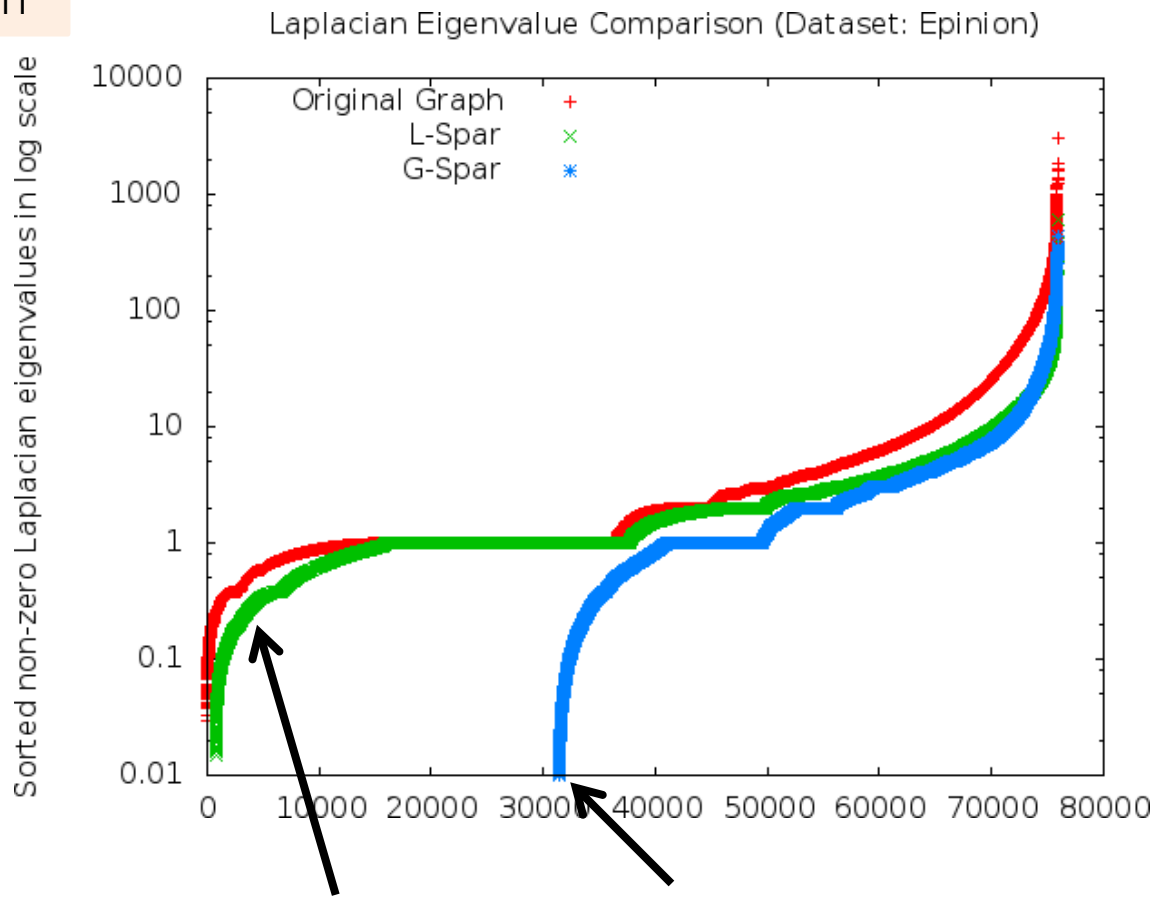
Impact of Sparsification on the Spectrum

Yeast graph



Impact of Sparsification on the Spectrum

Epinions graph



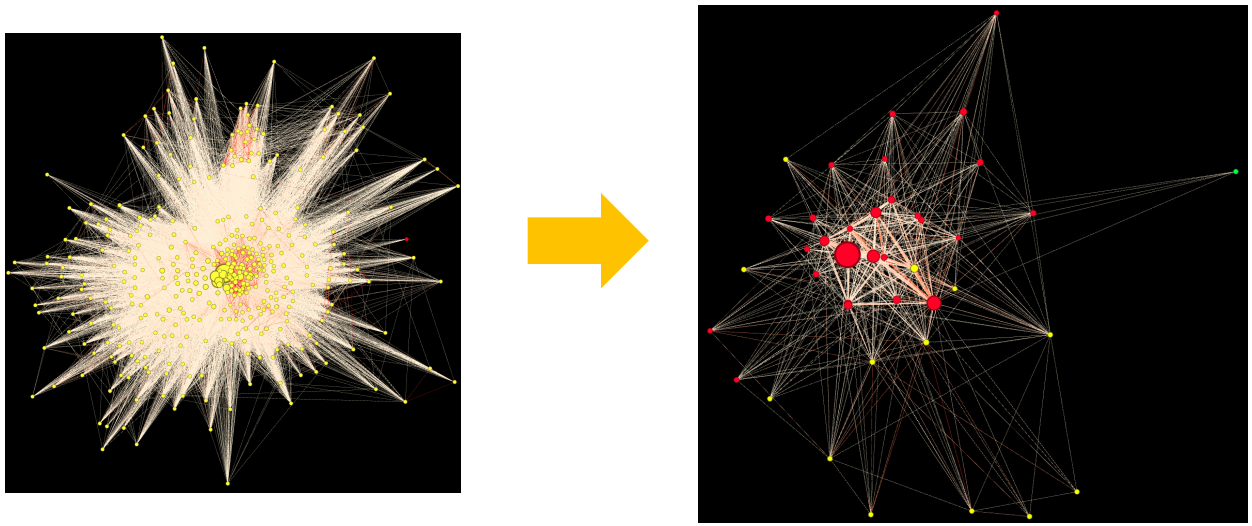
Local sparsification seems to match trends of original graph

Global Sparsification results in multiple components ($\lambda = 0$)

Graph summarization

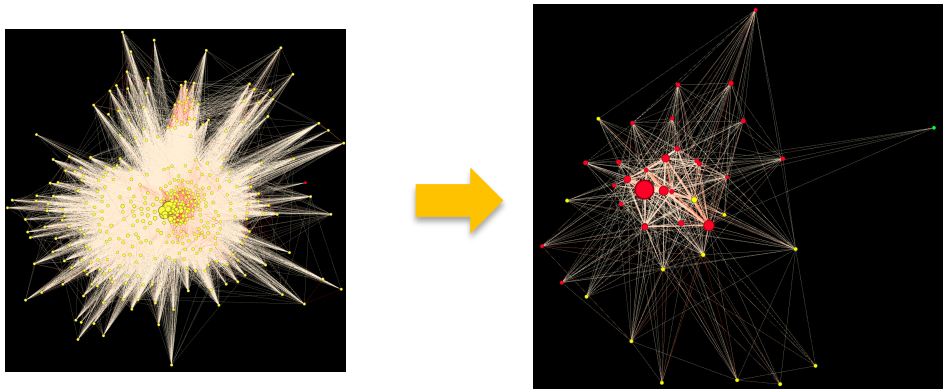
What is Graph Summarization?

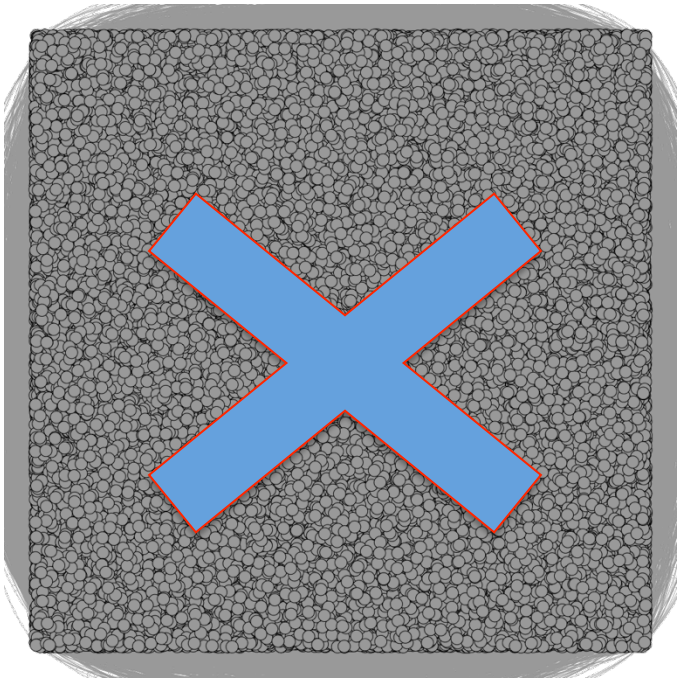
- It seeks to find
  - A short representation of the input graph
  - Often in the form of a summary or sparsified graph
    - Which reveals patterns in the original data and preserves specific structural or other properties, depending on the application domain



Why Graph Summarization?

- Reduction of data volume + storage
  - E.g., fewer I/O operations
- Speedup of algorithms + queries
- Interactive analysis
- Noise elimination
  - Reveals patterns

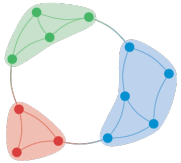




How to Evaluate a Summary?

- No metrics that apply to all summarization methods

- Compression-based
  - Minimize number of bits without losing much information, reduce # nodes / edges
- Query-oriented (e.g., reachability)
  - Accuracy vs. runtime
- Clustering-oriented
  - Maintain community structure
- Quality-based measures
  - “Interestingness”, reconstruction error



Plain Graph Summarization: Definition

- Input: static graph G or its adjacency matrix A
- Output:
  - Summary graph or
  - A set of structures or
  - A compressed data structure
- to concisely describe the given graph

Grouping-based Summarization

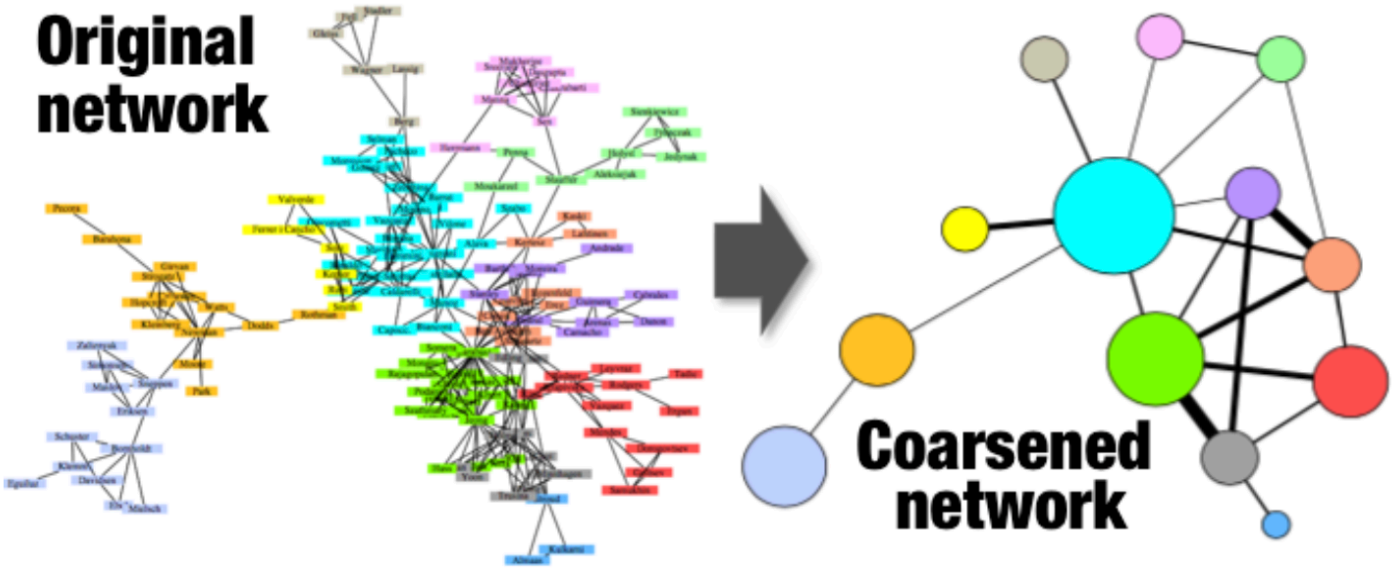
- The most popular technique for summarization
  - Aggregate nodes into supernodes
  - Connect supernodes with superedges

- Two main approaches
  - Leverage existing clustering methods

  or

  - Recursively aggregate nodes based on an optimization function

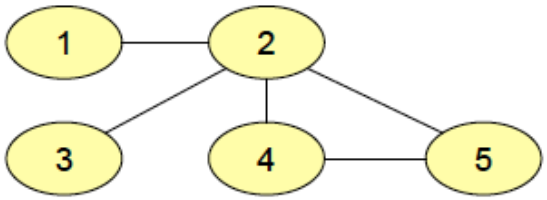
Clustering as a Summarization Technique



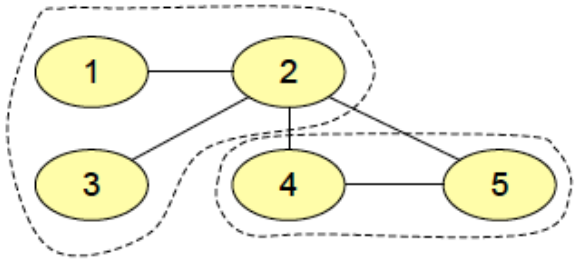
- Apply modularity clustering
- Create supernodes that correspond to communities
  - Condensed representation of the graph

[Newman and Girvan, '04]

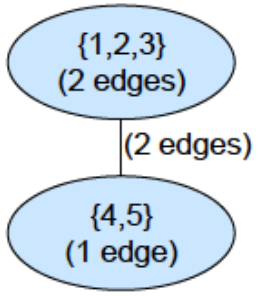
Grouping-based Approaches (1/2)



Input graph



Partitioned graph



Graph summary S

- Suppose that we are given a summary S
- There are typically several graphs that could have produced the same summary S

Let S be a summary graph. We define the expected adjacency matrix $\bar{A}$

1. If $u, v \in V$ are distinct nodes in the same supernode V_i, then $\bar{A}(u,v) = \frac{2E_i}{|V_i|(|V_i|-1)}$
2. If $u, v \in V$ are distinct nodes in different supernodes V_i, V_j, then $\bar{A}(u,v) = \frac{2E_{ij}}{|V_i| \cdot |V_j|}$
3. Otherwise (if u = v) $\bar{A}(u,v) = 0$

[Le Fevre and Terzi '10]

Grouping-based Approaches (2/2)

- Main Idea: Greedily group nodes…
  - … s.t. the reconstruction error is minimized

    without guarantees [Le Fevre and Terzi '10]

$$\text{Reconstr\_Error} = \frac{1}{|V|^2} \sum_{i=1}^{|V|} \sum_{j=1}^{|V|} |\bar{A}(i,j) - A(i,j)|$$

expected adjacency

    or with guarantees on reconstruction error [Riondato '14]

$$\text{Reconstr\_Error} = \left(\sum_{i=1}^{|V|} \sum_{j=1}^{|V|} |\bar{A}(i,j) - A(i,j)|^p \right)^{1/p}$$

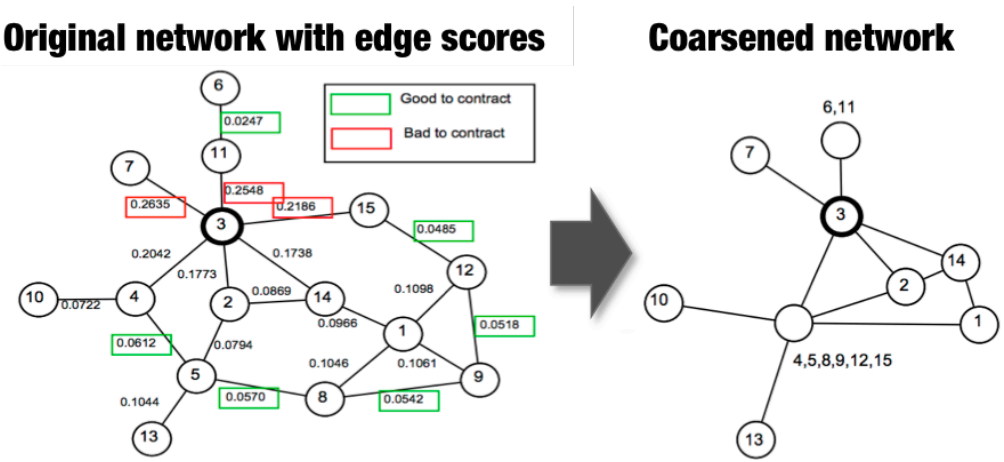
ℓ_p- rec error

Influence-based Coarsening (1/2)

- Main Idea: Greedily group nodes…
  - … that maintain the diffusive properties of the graph
  - Related to λ_1 (first eigenvalue of A)

$$\text{argmin}_{E' \subset E} \, | \lambda_1(G) - \lambda_1(G') |$$

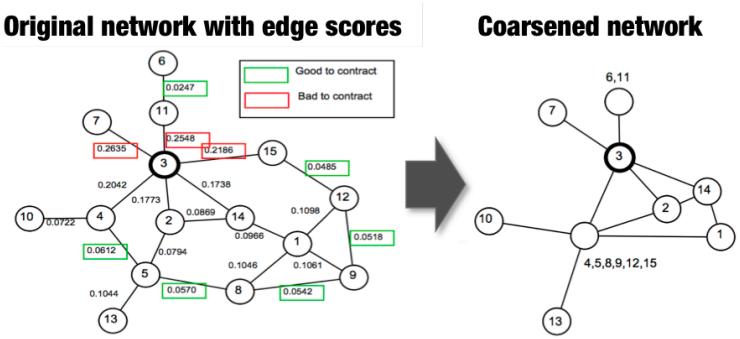
  - Ideally: Iteratively merge the (adjacent) node pair that minimizes the change in λ_1



[Purohit et al., '14]

Influence-based Coarsening (2/2)

- Algorithm: CoarseNet (Greedy approach)
  1. For each (adjacent) node pair compute the change in λ_1
     - Approximation based on perturbation theory
  2. Sort the node pairs in increasing order
  3. Until the desired contraction (# of nodes) is achieved, merge the node pairs in order
  4. Reweigh the edges to maintain λ_1



[Purohit et al., '14]

Next Lecture

- Epidemic models
- Influence maximization in networks

Thank You!

DiscoverGreece.com