Network Sampling: Methods and Applications

Mohammad Al Hasan
Assistant Professor, Computer Science
Indiana University Purdue University, Indianapolis, IN

Nesreen K. Ahmed
Final year PhD Student, Computer Science
Purdue University, West Lafayette, IN

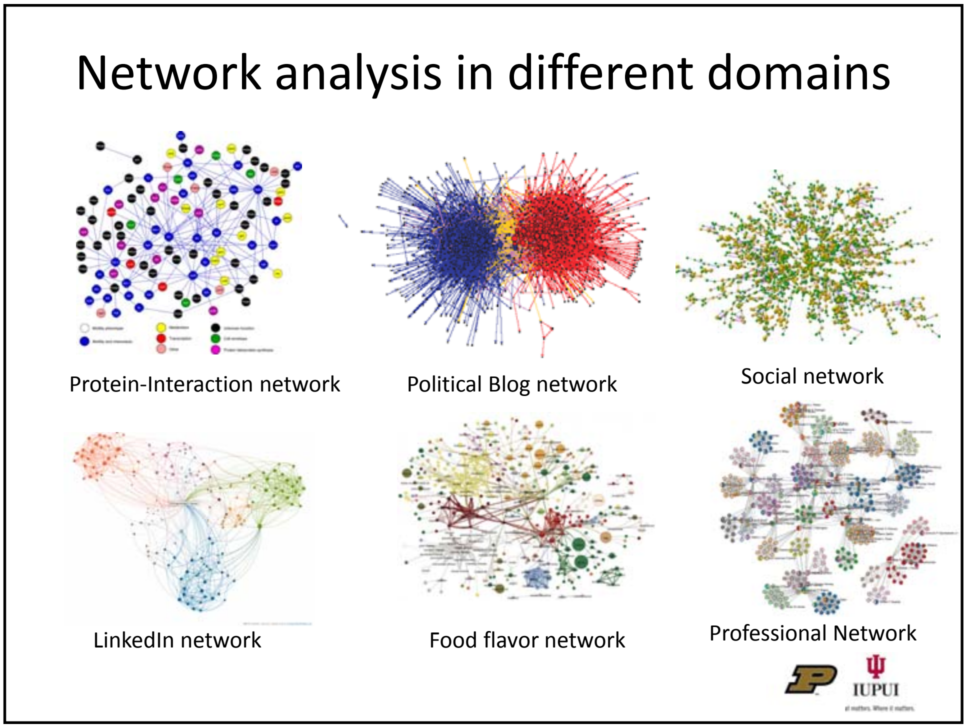
Jennifer Neville
Associate Professor, Computer Science
Purdue University, West Lafayette, IN

Tutorial Outline

- Introduction (15 minutes)
  - Motivation, Challenges
- Sampling Background (15 minutes)
  - Assumption, Definition, Objectives
- Network Sampling methods (full access, restricted access, streaming access)
  - Estimating nodal or edge characteristics (45 minutes)
  - Sampling representative sub-networks (30 minutes)
  - Sampling and counting of sub-structure of networks (30 minutes)
- Conclusion and Q/A (15 minutes)

Introduction

motivation and challenges



Network analysis in different domains

Protein-Interaction network

Political Blog network

Social network

LinkedIn network

Food flavor network

Professional Network

Network characteristics (1)

- Assume *G(V, E)* is a graph
  - $|V| = n, |E| = m$
- Average degree
  - For any vertex *v*, *d(v)* represents the degree of *v*
  - Average degree, $\bar{d} = \frac{1}{n}\sum_{v \in V} d(v)$
- Average clustering co-efficient
  - *C(v),* is the fraction of the $\langle u, w \rangle$ (ordered) pairs such that , $u, w \in adj(v), \text{and } (u, w) \in E$
- Diameter of the network $(\rho(G))$
  - The maximum value for the length of a shortest path between a pair of vertices
  - For disconnected network, we compute diameters only over the giant component
- Max *k*-core: Maximum *k*-value such that an induced subgraph exist in which every vertices in that subgraph has a minimum degree of *k*

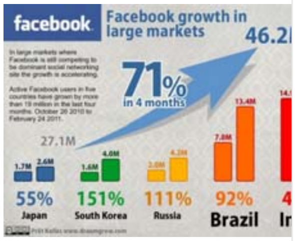
Network characteristics (2)

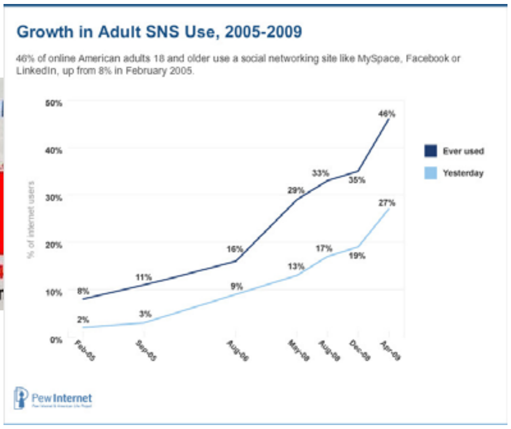
- Degree distribution
  - $[p_k]_{1 \le k \le n-1}$, for a degree value k, p_k is the fraction of vertices with that degree, $\sum_{k>0} p_k = 1$
  - For directed graph, we can consider both in-degree distribution, and out-degree distribution
- Hop-plot distribution
  - $[h_d]_{1 \le d \le \rho(G)}$, for a integer d, h_d denotes the fraction of ordered pairs of vertices (u, v) that are within a distance of d or less
  - This is a cumulative distribution
- Clustering coefficient distribution
  - $[C_k]_{1 \le k \le n-1}$, for a degree value k, C_k is the average clustering coefficient over all the vertices with degree k
- Distribution of between-ness centrality, and close-ness centrality of vertices
- Other network parameters are distribution of singular values of the graph adjacency matrix

Network analysis tasks

- Study the node/edge properties in networks
  - E.g., Investigate the correlation between attributes and local structure
  - E.g., Estimate node activity to model network evolution
  - E.g., Predict future links and identify hidden links

Network evolution

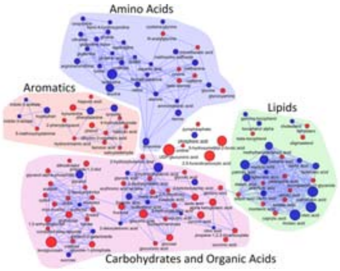


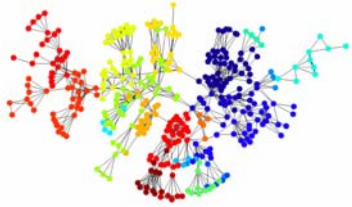

Growth in Adult SNS Use, 2005-2009

46% of online American adults 18 and older use a social networking site like MySpace, Facebook or LinkedIn, up from 8% in February 2005.

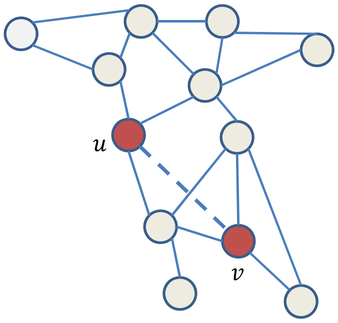
- 46% of online American adults 18 and older use a social networking site like MySpace, Facebook or LinkedIn
- 65% of teens 12-17 use online social networks as of Feb 2008

Link and communities!





Communities of physicists that work on social networks

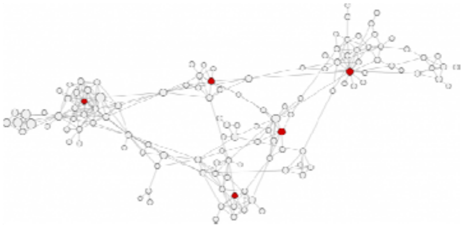


Link Prediction:
What is the probability that u and v will be connected in future?

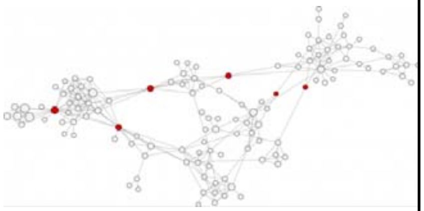
Network analysis tasks

- Study the node/edge properties in networks
  - E.g., Investigate the correlation between attributes and local structure
  - E.g., Estimate node activity to model network evolution
  - E.g., Predict future links and identify hidden links
- Study the connectivity structure of networks and investigate the behavior of processes overlaid on the networks
  - E.g., Estimate centrality and distance measures in communication and citation networks
  - E.g., Identify communities in social networks
  - E.g., Study robustness of physical networks to attack

Various centrality metrics



Red nodes has high closeness centrality



Red nodes has high between-ness centrality

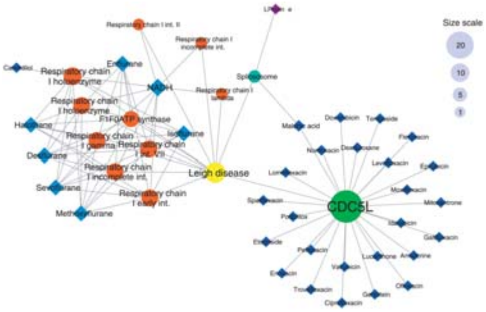
Other centralities:

Degree centrality

Eigenvector Centrality

Pagerank centrality

Centrality analysis is used to identify new pharmacological strategies


Respiratory chain I int. II
Respiratory chain I incomplete int.
Respiratory chain I holoenzyme
NADH
Respiratory chain I lambda
Respiratory chain I holoenzyme
F1F0ATP synthase
Respiratory chain I gamma
Respiratory chain I int. VII
Respiratory chain I incomplete int.
Respiratory chain I early int.
Leigh disease
CDC5L
Size scale
20
10
5
1

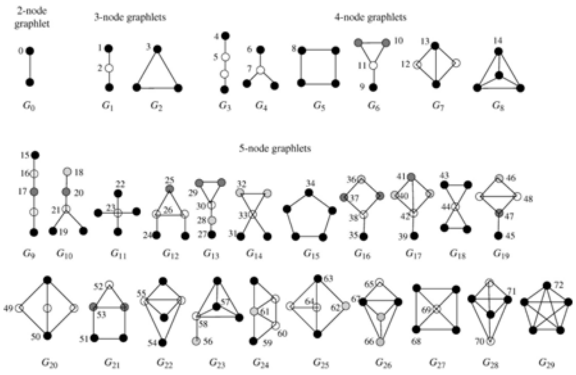
Source: Modularity in Protein Complex and Drug Interactions Reveals New Polypharmacological Properties, Jose C. Nacher mail, and Jean-Marc Schwartz, *PloS One, 2012*

- Yellow node is the disease nodes, protein complexes are circles, and diamond nodes are drugs.
- Links between the disease node and protein complexes represent associations between genes involved in these complexes and the named disease, as specified by the Disease Ontology.
- A drug is connected to a protein complex if at least one protein target of the drug is also a subunit of the protein complex.

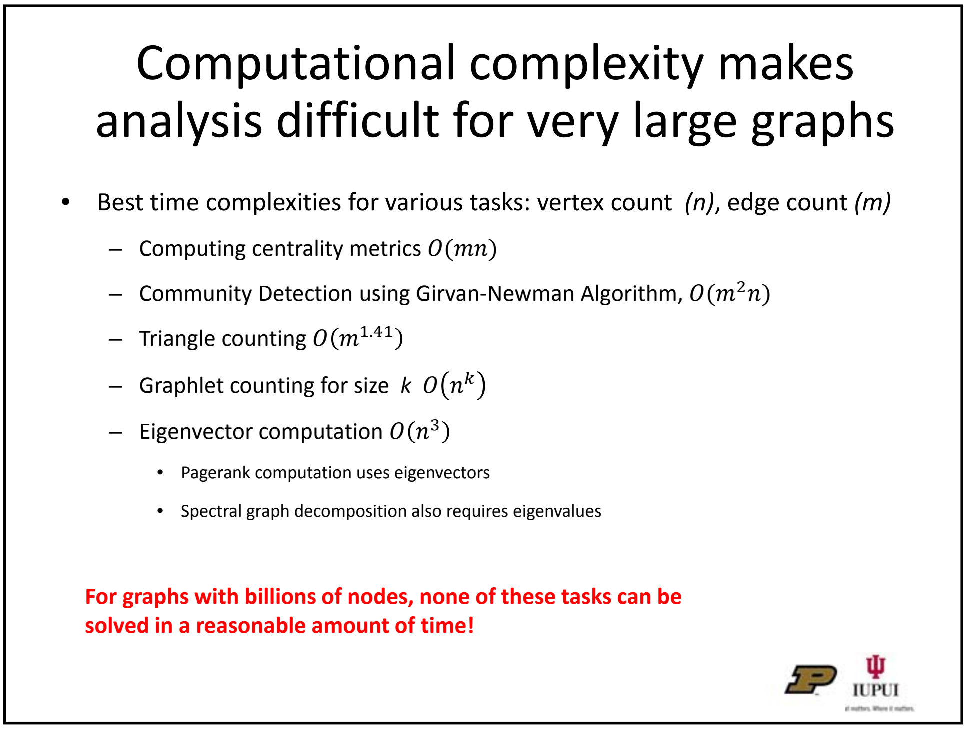
Network analysis tasks

- Study the node/edge properties in networks
  - E.g., Investigate the correlation between attributes and local structure
  - E.g., Estimate node activity to model network evolution
  - E.g., Predict future links and identify hidden links
- Study the connectivity structure of networks and investigate the behavior of processes overlaid on the networks
  - E.g., Estimate centrality and distance measures in communication and citation networks
  - E.g., Identify communities in social networks
  - E.g., Study robustness of physical networks to attack
- Study local topologies and their distributions to understand local phenomenon
  - E.g., Discovering network motifs in biological networks
  - E.g., Counting graphlets to derive network “fingerprints”
  - E.g., Counting triangles to detect Web (i.e., hyperlink) spam

Graphlet histogram is used for building network fingerprints



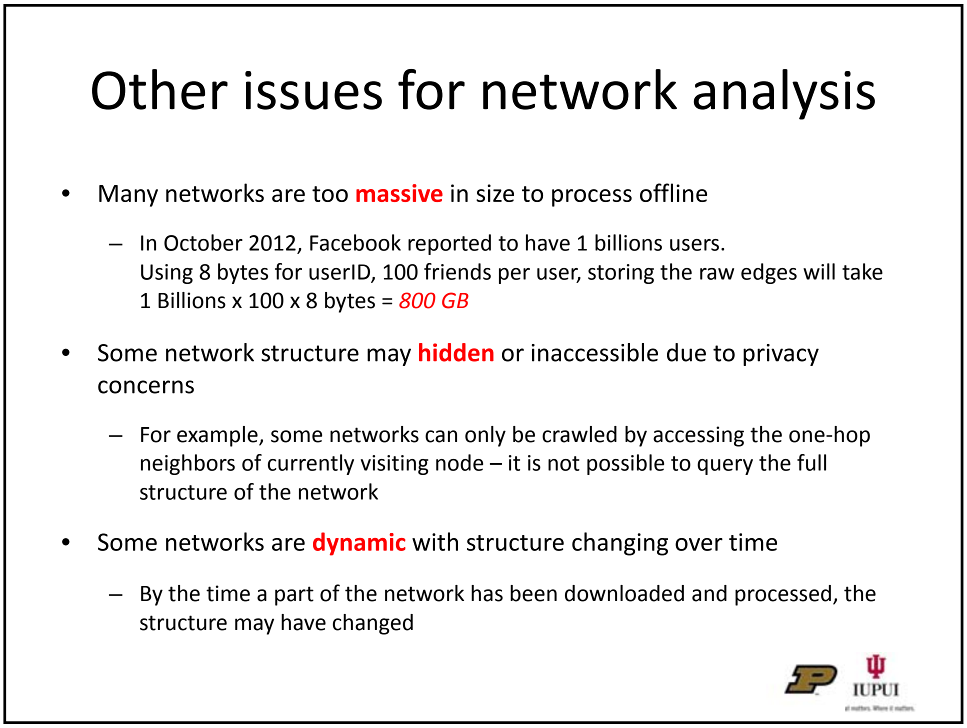
- Build fingerprints for large networks through frequency counts of graphlets
- Useful in anomaly detection (e.g., security applications) and differentiate network from different domains (e.g., biology applications)



Computational complexity makes analysis difficult for very large graphs

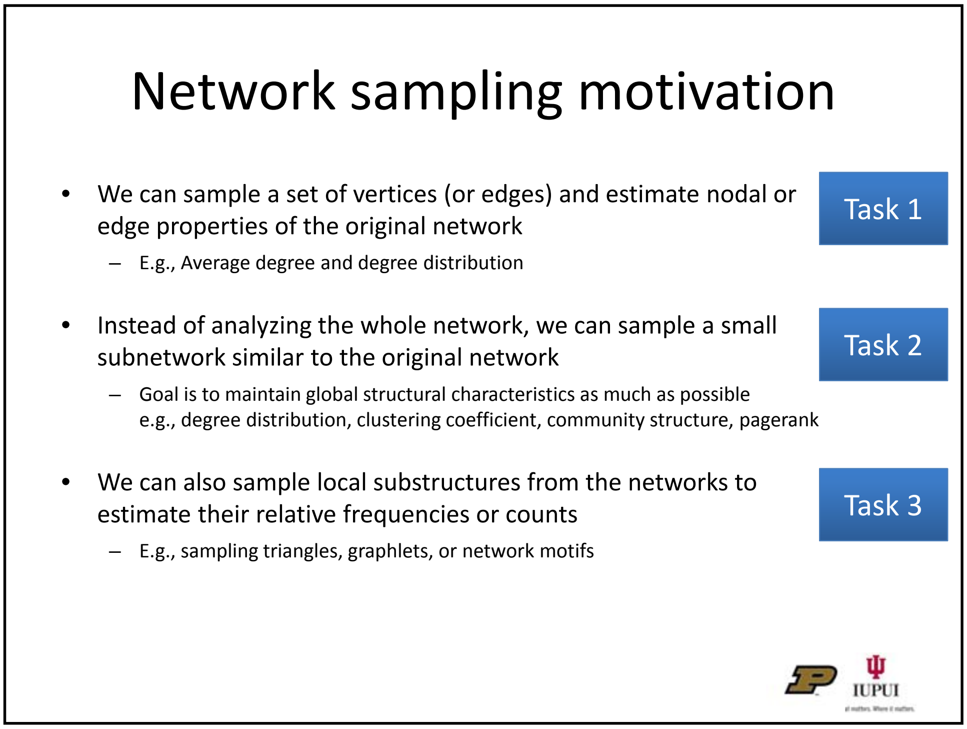
- Best time complexities for various tasks: vertex count *(n)*, edge count *(m)*
  - Computing centrality metrics $O(mn)$
  - Community Detection using Girvan-Newman Algorithm, $O(m^2n)$
  - Triangle counting $O(m^{1.41})$
  - Graphlet counting for size k $O(n^k)$
  - Eigenvector computation $O(n^3)$
    - Pagerank computation uses eigenvectors
    - Spectral graph decomposition also requires eigenvalues

For graphs with billions of nodes, none of these tasks can be solved in a reasonable amount of time!



Other issues for network analysis

- Many networks are too **massive** in size to process offline
  - In October 2012, Facebook reported to have 1 billions users.
    Using 8 bytes for userID, 100 friends per user, storing the raw edges will take
    1 Billions x 100 x 8 bytes = *800 GB*
- Some network structure may **hidden** or inaccessible due to privacy concerns
  - For example, some networks can only be crawled by accessing the one-hop neighbors of currently visiting node – it is not possible to query the full structure of the network
- Some networks are **dynamic** with structure changing over time
  - By the time a part of the network has been downloaded and processed, the structure may have changed



Network sampling motivation

- We can sample a set of vertices (or edges) and estimate nodal or edge properties of the original network
  - E.g., Average degree and degree distribution

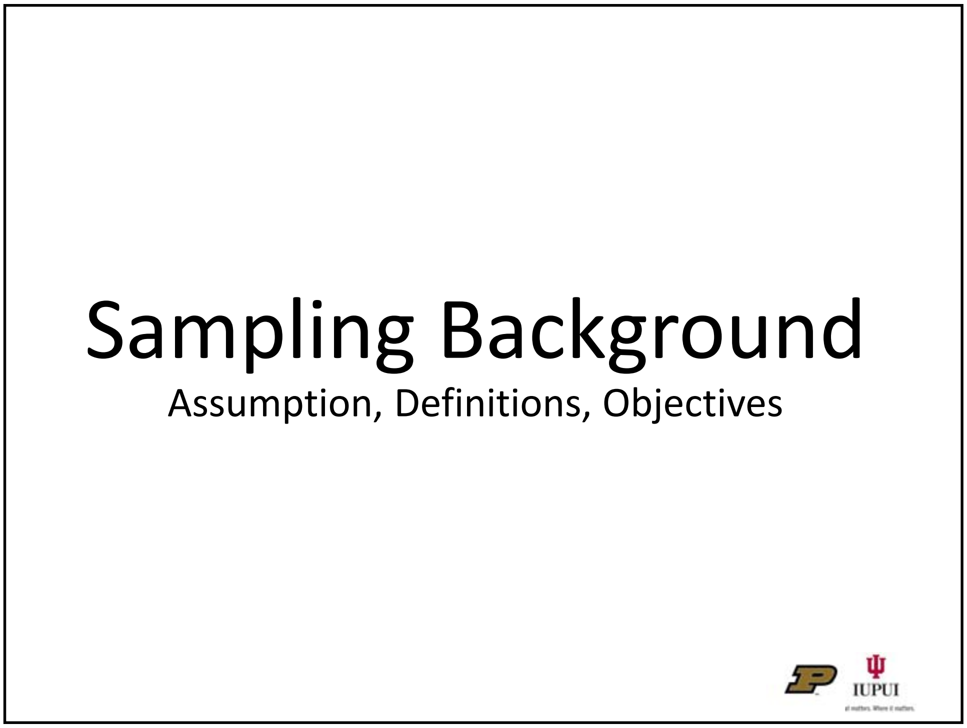
Task 1

- Instead of analyzing the whole network, we can sample a small subnetwork similar to the original network
  - Goal is to maintain global structural characteristics as much as possible e.g., degree distribution, clustering coefficient, community structure, pagerank

Task 2

- We can also sample local substructures from the networks to estimate their relative frequencies or counts
  - E.g., sampling triangles, graphlets, or network motifs

Task 3



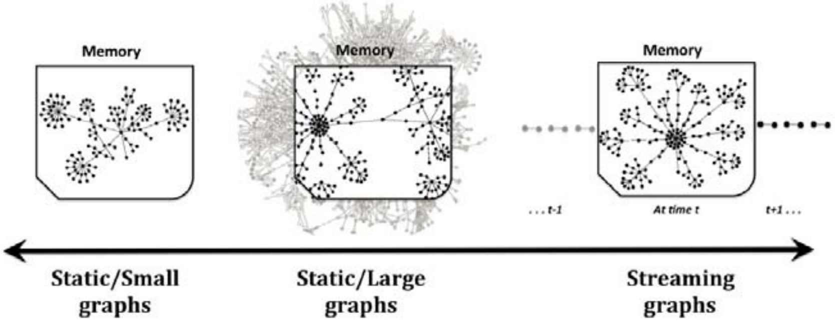
Sampling Background

Assumption, Definitions, Objectives

Sampling scenarios

- Full access assumption
  - The entire network is visible
  - A random node or a random edge in the network can be selected
- Restricted access assumption
  - The network is hidden, however it supports crawling, i.e. it allows to explore the neighbors of a given node.
  - Access to one seed node or a collection of seed nodes are given.
- Streaming access assumption (limited memory and fast moving data)
  - In the data stream, edges arrives in an arbitrary order (arbitrary edge order)
  - In the data stream, edges incident to a vertex arrives together (incident edge order)
  - Stream assumption is particularly suitable for dynamic networks

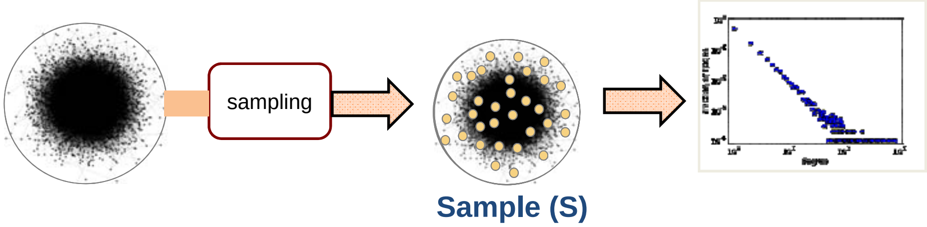
Sampling scenarios (cont.)



- For static and/or small network, computation model can assume that the entire network is in the memory
- For large but static network, part of the network can be loaded in the memory, but the entire network remains in disk or graph database
- Streaming scenario works well for both dynamic and large networks

Sampling objectives (Task 1)

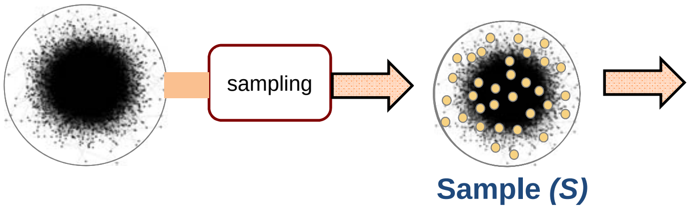
- Estimate network characteristics by sampling vertices (or edges) from the original networks
- Population is the entire vertex set (for vertex sampling) and the entire edge set (for edge sampling)
- Sampling is usually with replacement



Estimating degree distribution of the original network

Task 1 applications *(estimate node/edge attributes)*

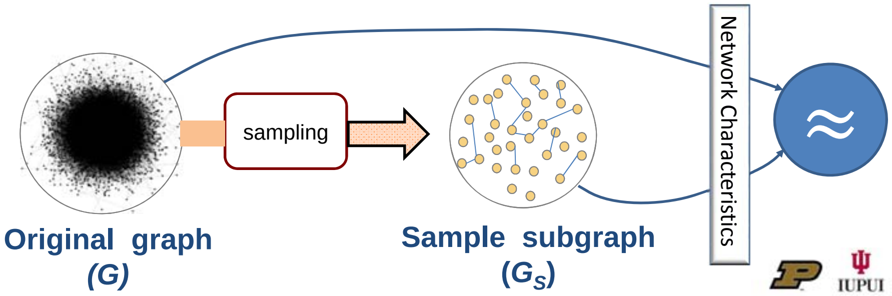
- Full Network: $G(V,E)$
  - Node-set: $\{v_1, v_2, \cdots, v_n\}$
  - Node-attributes: $[a_1(v_i), a_2(v_i), \cdots, a_k(v_i)] \text{ for } v_i \in V$
- Sample: $S \subset V$
- Goal: Compute $f_a(S) \approx f_a(V)$ for some function f_a of node attribute *a*



Compute $f_a(S)$ using *S*

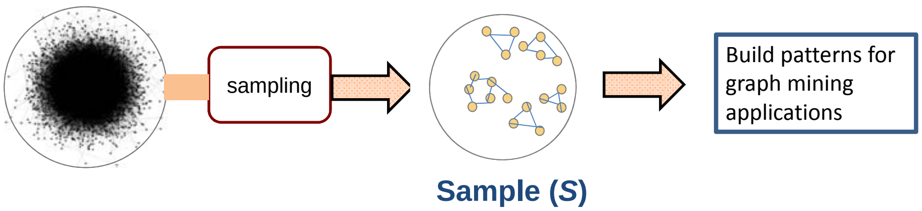
Sampling objectives (Task 2)

- Goal: From *G*, sample a subgraph with *k* nodes which preserves the value of key network characteristics of *G,* such as:
  - Clustering coefficient, degree Distribution, diameter, centrality, and community structure
  - Note that, the sampled network is smaller, so there is a scaling effect on some of the statistics; for instance, average degree of the sampled network is smaller
- Population: All subgraph of size *k*



Sampling objective (Task 3)

- Sample sub-structure of interest
  - Find frequent induced subgraph (network motif)
  - Sample sub-structure for solving other tasks, such as counting, modeling, and making inferences



Evaluation of sample quality

- For evaluating scalar measurement, such as average degree, average clustering coefficients, effective diameter
  - Analytically prove that the sampler provides unbiased estimate
  - Empirically estimate the accuracy of sample mean and variance
- For comparing two distributions
  - Kolmogorov-Smirnov (KS) *D*-Statistics can be used, which is the maximum difference between two cdfs

    $$D(F_1, F_2) = max_x \mid F_1(x) - F_2(x) \mid$$

  - Another option to use *K-L* divergence between two distribution smooth version of it:

    $$KL(\alpha f_1 + (1-\alpha) f_2 \parallel \alpha f_2 + (1-\alpha) f_1)$$

  - Neither of the above evaluation metrics address the issue of scaling, however, while comparing between two distributions that have scale mismatch (Task 2), *D*-statistics should be used

Sampling methods

methodologies, comparison, analysis

TASK 1

Estimating node/edge properties of the original network

Estimate node/edge properties in original network

Sample representative subgraph/subnetwork

Estimate frequency of subgraph patterns

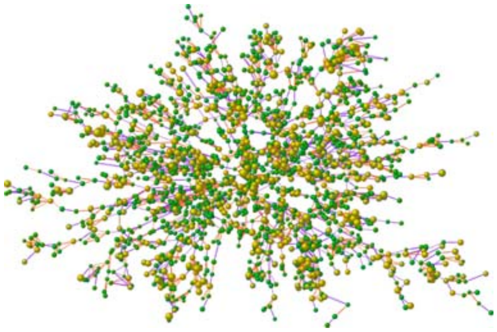
	Task 1	Task 2	Task 3
Full Access	✓		
Restricted Access			
Data Stream Access			

Data Access Assumption

Sampling Objective

Sampling methodology

Real-life Example : Birds of same feather flocks together!



- Red borders are women, blue borders are men
- Size of a node represents BMI
- Orange nodes are obese, and green nodes are normal
- Purple links are close genetic connection
- Gray node denotes non-genetic ties (friends, spouse, co-workers)

The New England Journal of Medicine. The study's authors suggest that obesity isn't just spreading; rather, it may be contagious between people, like a common cold.

Read more: http://www.time.com/time/health/article/0,8599,1646997,00.html#ixzz2XAtfg02V

How to conduct the same analysis on Facebook data where *n=1 billion*?

facebook
December 2010

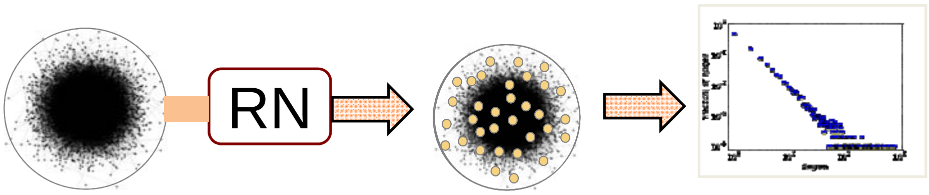
https://www.facebook.com/note.php?note_id=469716398919

Sampling methods
Task 1, Full Access assumption

- Uniform node sampling
  - Random node selection (RN)
- Non-uniform node sampling
  - Random degree node sampling (RDN)
  - Random pagerank node sampling (RPN)
- Uniform edge sampling
  - Random Edge selection (RE)
- Non-uniform edge sampling
  - Random node-edge (RNE)
  - RNE-RE Hybrid (HYB)

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 nodal attributes:
  - Average degree and degree distribution
  - Average of any nodal attribute
  - *f(u)* where *f* is a function that is defined over the node attributes



Sample (*S*)

Random Degree node selection (RDN)

- In this sampling, selection of a node is proportional to its degree
  - If $\pi(u)$ is the probability of selecting a node, $\pi(u)=\dfrac{d(u)}{2m}$
  - Node can be sampled using inverse-transform method, For π , we simply need to construct its cmf, say Π, then the sampled node, *x* is obtained by, $x=\Pi^{-1}(U), \text{where}, U \sim Uni(0,1)$
  - A second method is to first choose one edge uniformly, then choose one of its end-point with equal probability, clearly,

$$\pi(x)=\sum_{e\in inc(x)}\frac{1}{2m}=\frac{d(u)}{2m}$$

- High degree nodes have higher chances to be selected
  - Average degree estimation is higher than actual, and degree distribution is biased towards high-degree nodes.
  - Any nodal estimate is biased towards high-degree nodes.

Random pagerank node sampling (RPN) [Leskovec '06]

- Pagerank in the stationary distribution vector of a specially constructed Markov process
  - Visit a neighbor following an outgoing link with probability *c* (typically kept at 0.85), and jump to a random node (uniformly) with probability *1-c*
  - Pagerank vector satisfies following eigenvector equations: $\pi=\left(cA^T D^{-1}+(1-c)U\right)\pi=M\pi$, where π is the eigenvector of *M* with the eigenvalue 1
  - Pagerank can be computed efficiently by power method

Text

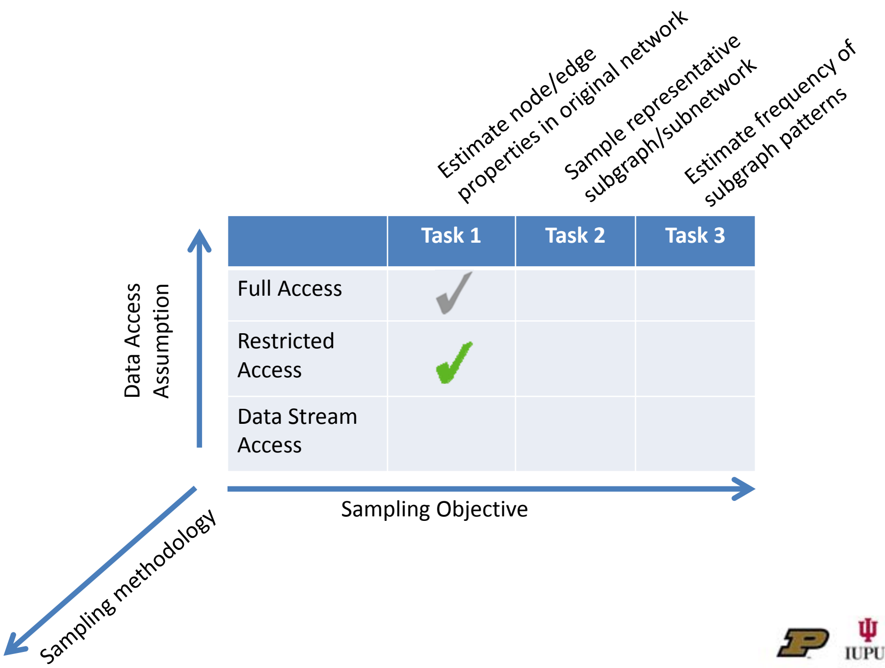
- Node *u* is sampled with a probability which is proportional to $c\cdot\frac{d_{in}(u)}{m}+\frac{1-c}{n}$
  - When *c=1*, the sampling is similar to RDN for the directed graph,
  - When *c=0* the sampling is similar to RN
- Nodes with high in-degree has higher chances to be selected
  - Due to the uniform random jump, high degree bias of RDN is somewhat reduced. So, it provides better estimation accuracy than RDN for average degree and degree distribution.

Random Edge Selection (RE)

- In a random edge selection, we uniformly select a set of edges and the sampled network is assumed to comprise of those edges.
- A vertex is selected in proportion to its degree
  - If $\rho = \frac{|E_S|}{|E|}$, the probability of a vertex *u* to be selected is: $1-(1-\rho)^{d(u)}$
  - when $\rho \to 0$ the probability is: $\rho \cdot d(u)$
  - With more sample degree bias is reduced
  - The selection of vertices are not independent as both endpoint of an edge are selected
- Nodal statistics will be biased to high-degree vertices
- Edge statistics is unbiased due to the uniform edge selection.

Random node-edge selection (RNE) [Leskovec '06]

- Select a vertex uniformly, and then pick an edge incident to the selected vertex (uniformly)
- Probability of selecting a vertex, is proportional to $\frac{1}{|V|}\left(1+\sum_{x\in adj(u)}\frac{1}{adj(x)}\right)$
- Node sampling is biased towards high degree vertices that are adjacent to many low degree vertices.
  - If the graph is assortative (social networks exhibits this property), the probability is almost uniform for all the vertices, so nodal estimates are better than the case of RE.
- Edge sampling is also non-uniform, edges that are incident to high degree nodes are under-sampled, and those the are incident to low degree nodes, are oversampled
  - An edge is sampled with the probability: $\frac{1}{|V|}\sum_{x\in inc(e)}\frac{1}{adj(x)}$


Estimate node/edge properties in original network
Sample representative subgraph/subnetwork
Estimate frequency of subgraph patterns
Data Access Assumption
Sampling Objective
Sampling methodology

	Task 1	Task 2	Task 3
Full Access	✓		
Restricted Access	✓		
Data Stream Access			

Public Facebook data can only be accessed by generating user IDs and crawling

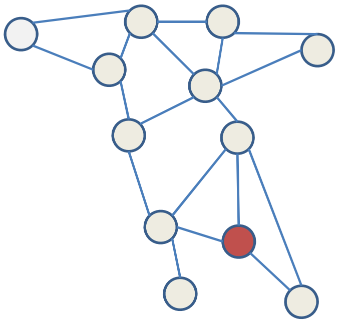
http://www.wired.com/wiredscience/2012/04/facebook-disease-friends/

Sampling under restricted (or full) access Assumption

- Assumptions
  - The network is connected, (if not) we can ignore the isolated nodes
  - The network is hidden, however it supports crawling, i.e. it allows to explore the neighbors of a given node. Access to one seed node or a collection of seed nodes is given.
- Methods
  - Graph traversal techniques (exploration without replacement)
    - Breadth-First Search (BFS)
    - Depth-First Search (DFS)
    - Forest Fire (FF)
    - Snowball Sampling (SBS)
    - Respondent Driven Sampling (RDS)
  - Random walk techniques (exploration with replacement)
    - Classic Random Walk
    - Markov Chain Monte Carlo (MCMC) using Metropolis-Hastings algorithm
    - Random walk with restart (RWR)
    - Random walk with random jump (RWJ)
- During the traversal or walk, the visited nodes are collected in a sample set and those are used for estimating network parameters

Breadth-first Sampling

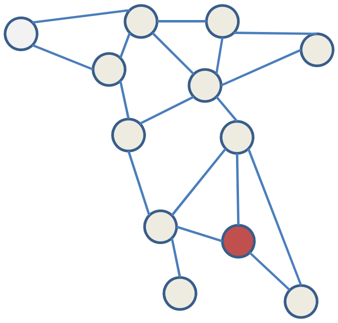
- At each iteration, earliest discovered but not yet visited node is selected
- For a selected node, the node is visited and all its neighbors are discovered
- It samples nodes from a specific region of the network
- It discovers all nodes within some distance from the seed node
- Nodal statistics are taken over the selected nodes.



- This sampling is biased as high-degree nodes have higher change to be sampled
- Nodal estimations are biased towards nodes with higher degree.

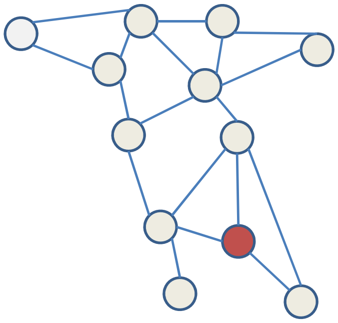
Depth-first Sampling

- At each iteration, we select the latest explored but, not-yet-visited node
- DFS explores first the nodes that are faraway from the seed
- The sampling is biased as high degree nodes has higher chance to be selected
- Same as BFS walk, estimation is biased towards nodes with higher degree.



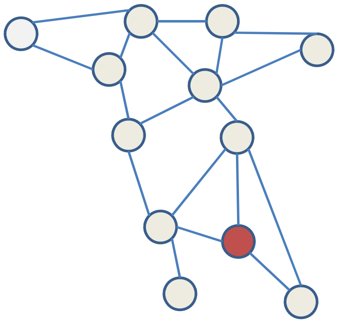
Forest Fire (FF) Sampling [Leskovec ’06]

- FF is a randomized version of BFS
- Every neighbor of current node is visited with a probability *p*. For *p=1* FF becomes BFS.
- FF has a chance to die before it covers all nodes.
- It is inspired by a graph evolution model and is used as a graph sampling technique
- its performance is similar as BFS sampling



Snowball Sampling

- Similar to BFS
- *n*-name snowball sampling is similar to BFS
- at every node *v*, not all of its neighbors but exactly *n* neighbors are chosen randomly to be scheduled
- A neighbor is chosen only if it has not been visited before.
- Performance of snowball sampling is also similar to BFS sampling



Snowball: n=2

Classic Random Walk Sampling (RWS)

- At each iteration, one of the neighbors of currently visiting node is selected to visit
- For a selected node, the node and all its neighbors are discovered
- The sampling follows depth-first search pattern

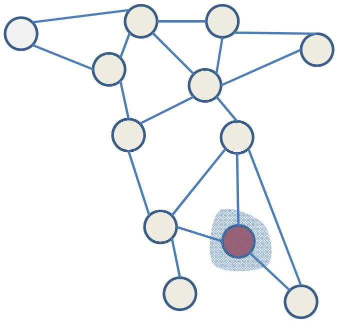
$$p_{u,v} = \begin{cases} 1/d(u) & \text{if } v \in adj(u) \\ 0 & \text{otherwise} \end{cases}$$

- This sampling is biased as high-degree nodes have higher change to be sampled, probability that node u is sampled is

$$\pi(u) = \frac{d(u)}{2m}$$

- Note that this method sample each edge uniformly, as it satisfy the detailed balanced equation,

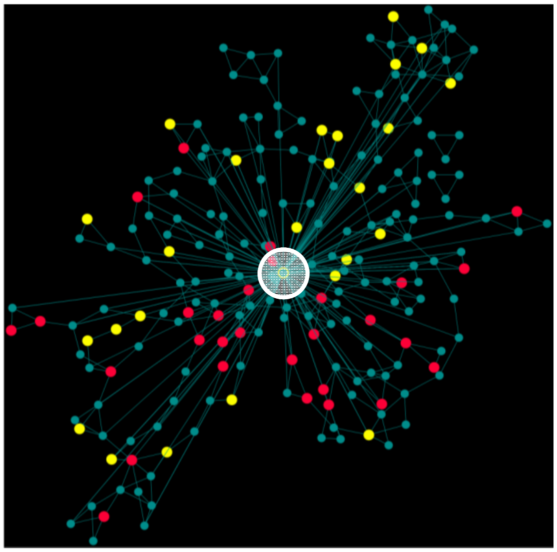
$$\pi(u) \cdot P_{u,v} = \pi(v) \cdot P_{v,u} = \frac{1}{2m}$$



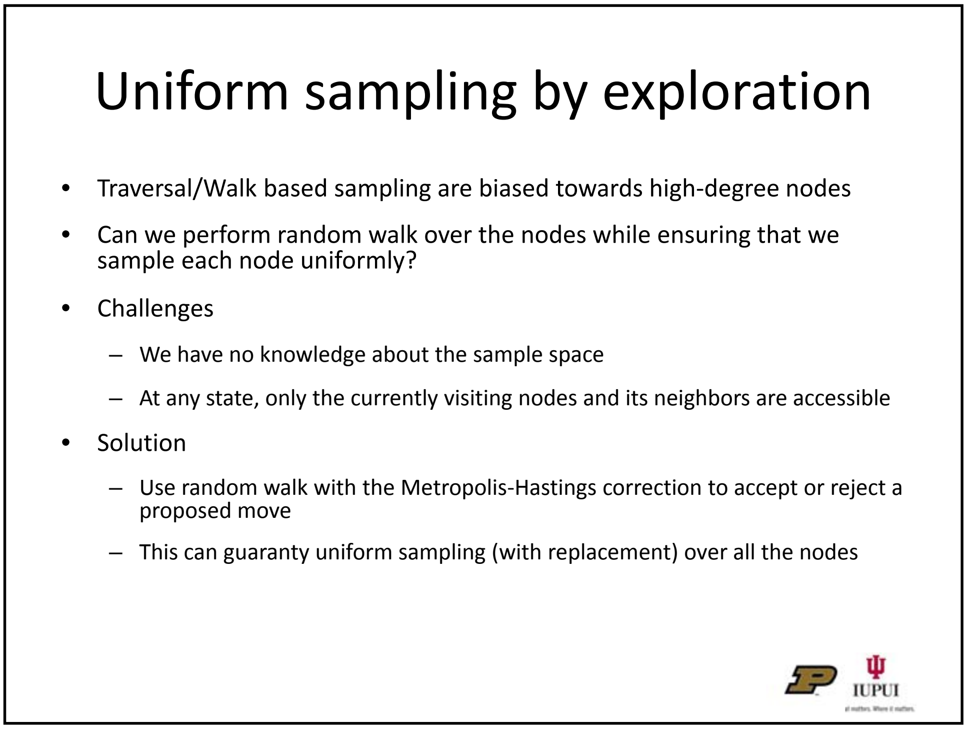
Other variants of random walk

- Random walk with restart (RWR)
  - Behaves like RWS, but with some probability *(1-c)* the walk restarts from a fixed node, *w*
  - The sampling distribution over the nodes models a non-trivial distance function from the fixed node, *w*
- Random walk with random jump (RWJ)
  - Access to arbitrary node is required
  - RWJ is motivated from the desire of simulating RWS on a directed network, as on directed network RWS can get stuck in a sink node, and thus no stationary distribution can be achieved
  - Behaves like RWS, but with some probability *(1-c)*, the walk jumps to an arbitrary node with uniform probability.
  - The stationary distribution is proportional to the pagerank score of a node, so the analysis is similar to RPN sampling

Exploration based sampling will be biased toward high degree nodes

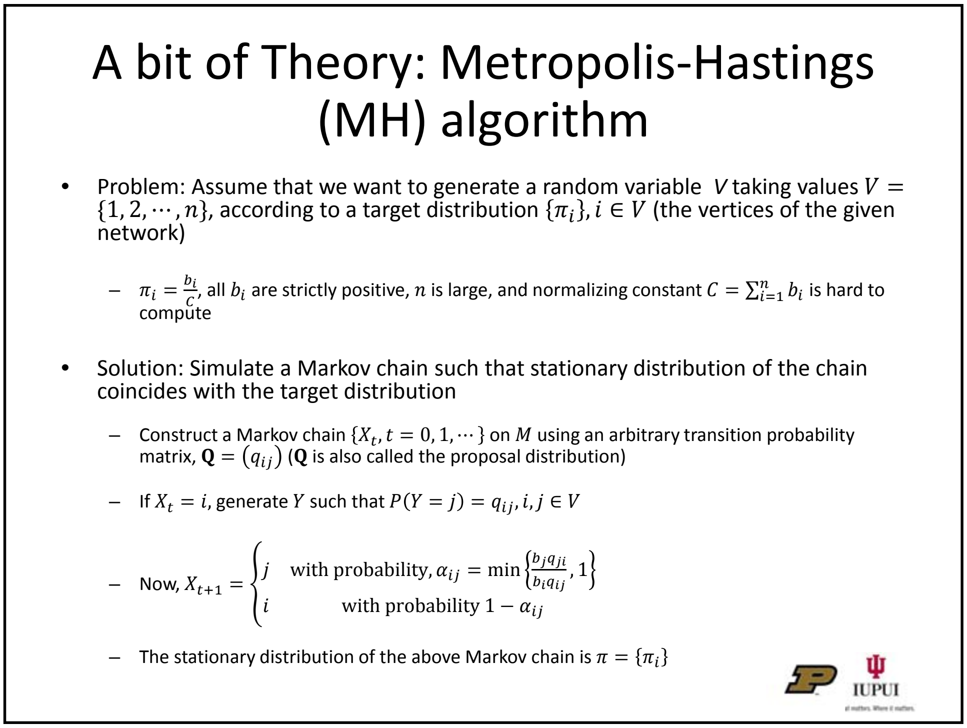


How can we modify the algorithms to ensure nodes are sampled uniformly at random?



Uniform sampling by exploration

- Traversal/Walk based sampling are biased towards high-degree nodes
- 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 proposed move
  - This can guaranty uniform sampling (with replacement) over all the nodes



A bit of Theory: Metropolis-Hastings (MH) algorithm

- Problem: Assume that we want to generate a random variable V taking values $V = \{1, 2, \cdots, n\}$, according to a target distribution $\{\pi_i\}, i \in V$ (the vertices of the given network)
  - $\pi_i = \frac{b_i}{C}$, all b_i are strictly positive, n is large, and normalizing constant $C = \sum_{i=1}^{n} b_i$ is hard to compute
- Solution: Simulate a Markov chain such that stationary distribution of the chain coincides with the target distribution
  - Construct a Markov chain $\{X_t, t = 0, 1, \cdots\}$ on M using an arbitrary transition probability matrix, $\mathbf{Q} = (q_{ij})$ ($\mathbf{Q}$ is also called the proposal distribution)
  - If $X_t = i$, generate Y such that $P(Y = j) = q_{ij}, i, j \in V$
  - Now, $X_{t+1} = \begin{cases} j & \text{with probability}, \alpha_{ij} = \min\left\{\frac{b_j q_{ji}}{b_i q_{ij}}, 1\right\} \\ i & \text{with probability } 1 - \alpha_{ij} \end{cases}$
  - The stationary distribution of the above Markov chain is $\pi = \{\pi_i\}$

Uniform node sampling with Metropolis-Hastings method [Henzinger '00]

- It works like random walk sampling (RWS), but it applies a correction so that high-degree bias of RWS is eliminated systematically
  - The proposal distribution, $\mathbf{Q}$ chooses one of the neighbor (say, j) of the current node (say, i) uniformly, thus proposal distribution works as RWS, $q_{ij} = 1/d(i)$
- Now, $X_{t+1} = \begin{cases} j & \text{with probability}, \alpha_{ij} = \min\left\{\frac{b_j q_{ji}}{b_i q_{ij}}, 1\right\} \\ i & \text{with probability } 1 - \alpha_{ij} \end{cases}$
  - For uniform target $b_i = b_j$, for RWS like proposal distribution, $q_{ij} = \frac{1}{d_i}$ and $q_j = \frac{1}{d_j}$
  - Thus, $\alpha_{ij} = \min\left\{\frac{d_i}{d_j}, 1\right\}$ if $d(j) \leq d(i)$, the choice is accepted only with probability 1, otherwise with probability $\frac{d(i)}{d(j)}$
- If a graph have n vertices, using the above MH variant, every node is sampled with $\frac{1}{n}$ probability

ANALYSIS

Task 1 performance summary over all the sampling method

Sampling	Direct sampling of Nodes or edges	Exploration (walk or traversal)
Uniform node sampling	RN	MH (uniform target distribution)
Almost Uniform node sampling	RNE	
Exactly Degree proportional	RDN	RWS
Apparently degree proportional (when sample size is small)	RE	BFS, FF, Snowball
PageRank proportional Sampling	RPN	RWJ

Comparison

- Node property estimation
  - Uniform node selection (RN) is the best as it selects each node uniformly
  - Average degree and degree distribution is unbiased
- Edge property estimation
  - Uniform edge selection (RE) is the best as it select each edge uniformly
  - For example, we can obtain an unbiased estimate of assortativity by RE method

Method	Vertex Selection Probability, $\pi(u)$ $\lvert V\rvert = n, \lvert E\rvert = m,$
RN, MH-uniform target	$\frac{1}{n}$
RDN, RWS	$\frac{d(u)}{2m}$
RPN, RWJ	$c\cdot\frac{d_{in}(u)}{m}+(1-c)\cdot\frac{1}{n}$ (undirected) $c\cdot\frac{d(u)}{2m}+(1-c)\cdot\frac{1}{n}$ (directed)
RE	$\sim\frac{d(u)}{2m}$
RNE	$\frac{1}{n}\left(1+\sum_{x\in adj(u)}\frac{1}{adj(x)}\right)$

Expected average degree and degree distribution

- For a degree value k assume p_k is the fraction of vertices with that degree $\sum_{k>0} p_k = 1$
- Uniform Sampling, node sampling probability, $\pi(v) = \frac{1}{|V|} = \frac{1}{n}$
  - Expected value of p_k: $q_k = \sum_{v \in V} \pi(v) \cdot 1_{d(v)=k} == \frac{1}{|V|} \cdot p_k \cdot |V| = p_k$
  - Expected observed node degree: $\sum_{k>0} k \cdot q_k = \sum_{k>0} k \cdot p_k$

Overestimate high-degree vertices, underestimate low-degree vertices

- Biased Sampling (degree proportional), $\pi(v) = \frac{d(v)}{2 \cdot |E|} = \frac{d(v)}{2m}$
  - Expected value of p_k: $q_k = \sum_{v \in V} \pi(v) \cdot 1_{d(v)=k} == \frac{k}{2 \cdot |E|} \cdot p_k \cdot |V| = \frac{k \cdot p_k}{\overline{d}}$
  - Expected observed node degree: $\sum_{k>0} k \cdot q_k = \frac{\sum_{k>0} k^2 \cdot p_k}{\overline{d}} = \frac{\overline{d^2}}{\overline{d}}$

Overestimate average degree

Example

- Actual degree distribution

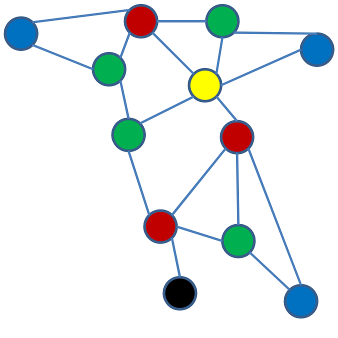
$$q_k = \left\{ \frac{1}{12}, \frac{3}{12}, \frac{4}{12}, \frac{3}{12}, \frac{1}{12} \right\}$$

- Average degree = 3
- RWS degree distribution

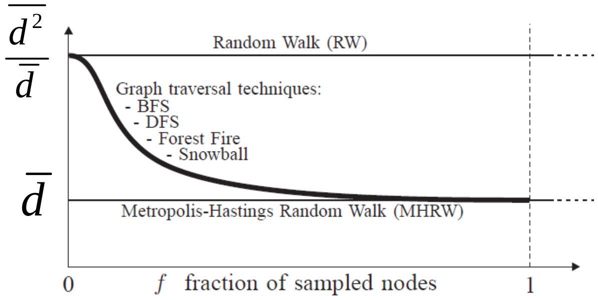
$$q_k = \left\{ \frac{1}{36}, \frac{6}{36}, \frac{12}{36}, \frac{12}{36}, \frac{5}{36} \right\}$$

$$\downarrow \quad \downarrow \quad - \quad \uparrow \quad \uparrow$$

- RWS average degree
  - 3.39

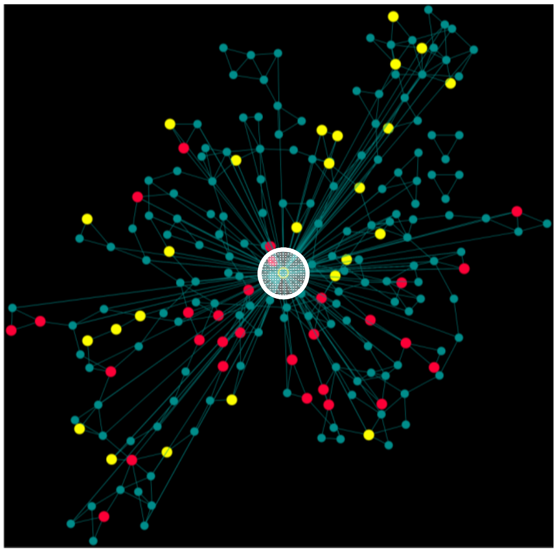


Expected average degree for traversal based methods [Kurant '10]



- Note that, for walk based method, the sampling is with replacement, so their analysis does not chance with the fraction of sampling
- Traversal based method behaves like RWS when the sample size is small, but as the sample size is increases its estimation quickly converges towards the true estimation
- Behavior of all the traversal method is almost identical. For traversal methods, it is better to have one sample with a large fraction than many samples of short fractions

When sampling algorithm selects nodes non-uniformly…



Node weights can be adjusted to remove the bias

Correction for bias in biased sampling [eg. Kurant '10]

- From $S \subset V$, we can compute estimated degree distribution, $\hat{q}_k$ (which is biased)
- We can derive unbiased estimated degree distribution $\hat{p}_k$ from $\hat{q}_k$
- For RWS, if $u \in S$, the probability of sampling is proportional to *u*'s degree, $d(u)$ and we have, $q_k = k \cdot \frac{p_k}{\bar{d}}$
- So, $\hat{p}_k \propto \frac{\hat{q}_k}{k}$, which implies $\hat{p}_k = C \cdot \frac{\hat{q}_k}{k}$
- *C* is a normalizing constant so that $\sum_{k>0} \hat{p}_k = 1$, thus, $C = \left(\sum_{l>0} \frac{\hat{q}_l}{l}\right)^{-1} = |S| / \sum_{u \in S} \frac{1}{d(u)}$
- Corrected average degree,

$$\hat{\bar{d}} = \sum_{k>0} k \cdot \bar{p}_k = \sum_{k>0} k \cdot C \cdot \frac{\hat{q}_k}{k} = C \cdot 1 = \frac{|S|}{\sum_{u \in S} 1/d(u)}$$

Correction of arbitrary nodal attributes for biased sampling (cont.)

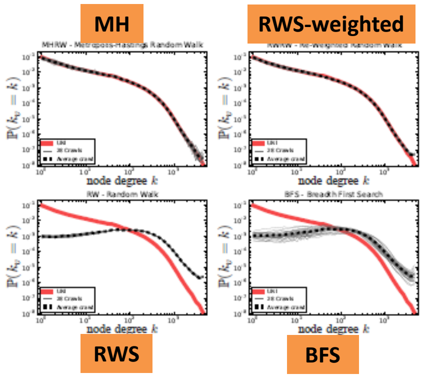
- Assume, $\{u_s\}_{\{s=1\}}^{t}$ are the nodes sampled to compute expectation of nodal attribute, $f(u)$
- π is a distribution that we achieve by a biased sampling, and U is a distribution which is unbiased
- Let's define a weight function $w: V \rightarrow R$ such that $w(i) = \frac{1/n}{\pi(i)}$
- For degree proportional sampling, $w(i) = \frac{1/n}{\pi(i)} = \frac{2m}{n} \cdot \frac{1}{d(i)}, i \in V$
- Unbiased expectation of $f(u)$ is: $\hat{\mu}_t(wf) = \frac{1}{t}\sum_{s=1}^{t} w(u_s) f(u_s) = E_\pi(wf) = E_u(f)$
- Correction of nodal attribute if *n* and *m* are unknown
  - So, we use weight which is correct up to a multiplicative constant: $w(i) = \frac{1}{d(i)}$
  - Then the un-biasing works as below:

$$E_u(f) = \frac{\hat{\mu}_t(wf)}{\hat{\mu}(w)} = \frac{\sum_{i \in [1..t]} f / d(i)}{\sum_{i \in [1..t]} 1/d(i)}$$

EMPIRICAL RESULTS

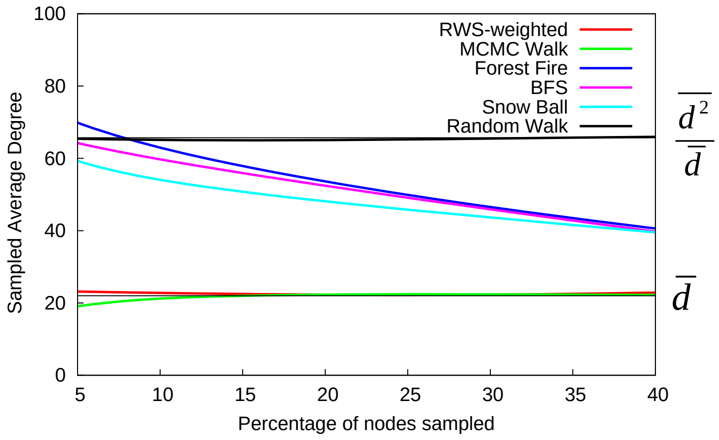
Comparison between different sampling strategies [Gjoka ’10, Gjoka ‘11]

- Gjoka et al. sampled Facebook network and compared the above sampling methods
- They confirmed that MH sampling and reweighted RWS can estimate degree distribution almost perfectly.

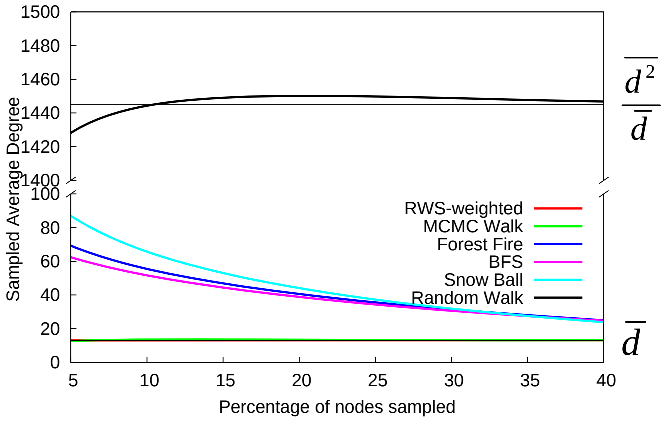


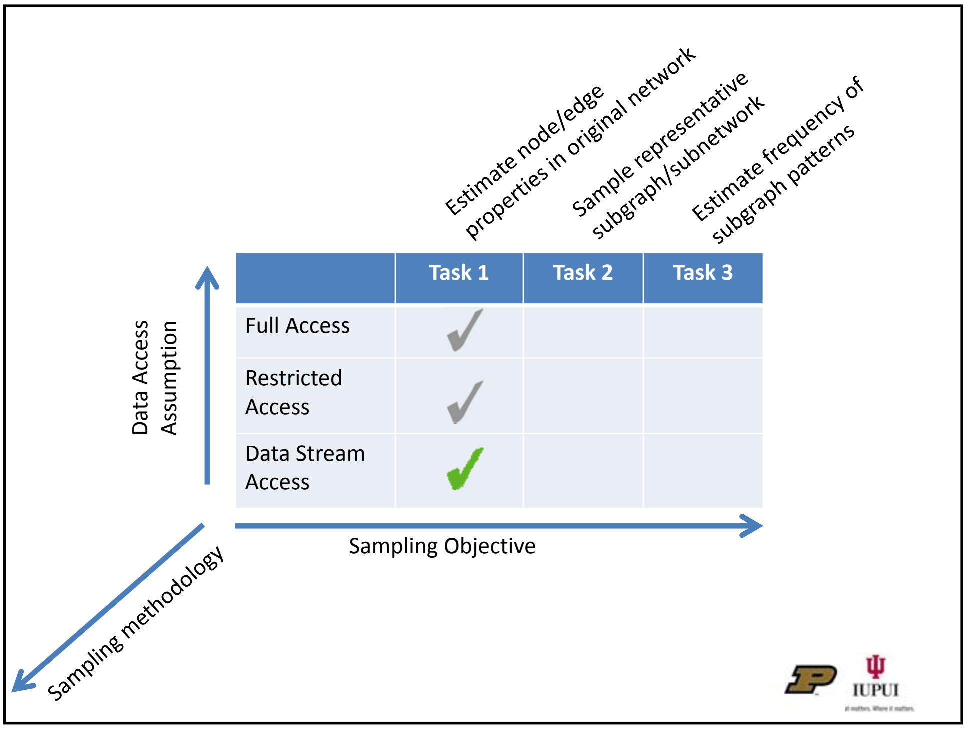
Average Degree Estimation	
BFS	285.9
RWS	338
MCMC	**95.2**
Actual	94.1
RWS (corrected)	93.9

Average degree estimation (astro-phy network)



Average degree estimation (skitter network)





Estimate node/edge properties in original network

Sample representative subgraph/subnetwork

Estimate frequency of subgraph patterns

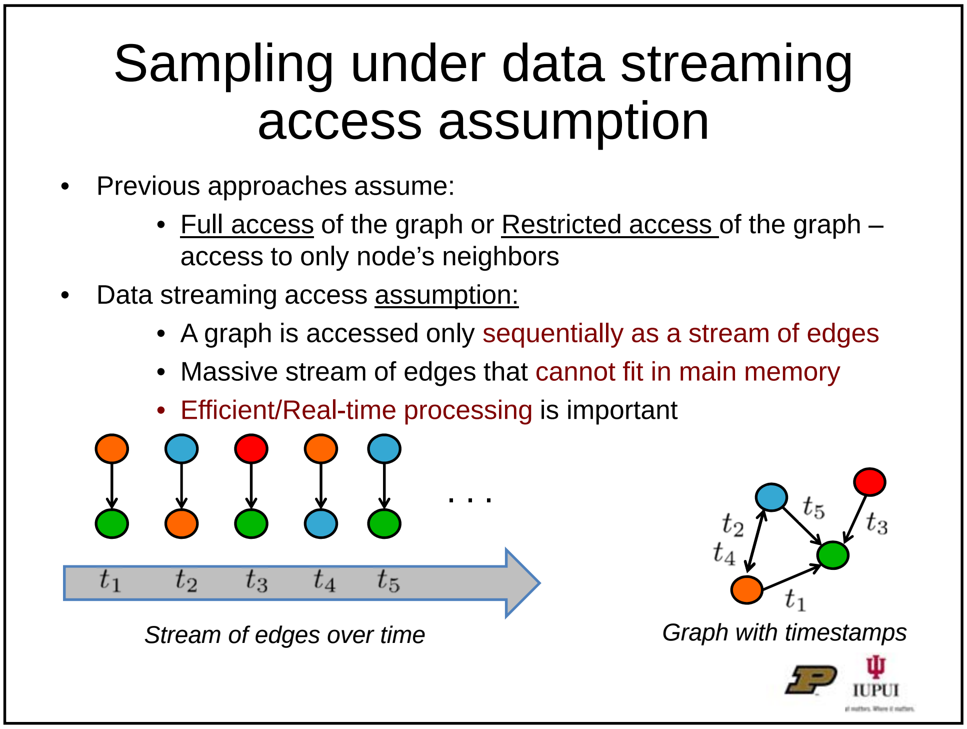
	Task 1	Task 2	Task 3
Full Access	✓		
Restricted Access	✓		
Data Stream Access	✓		

Data Access Assumption

Sampling Objective

Sampling methodology

Interaction networks can be extracted from dynamic communications



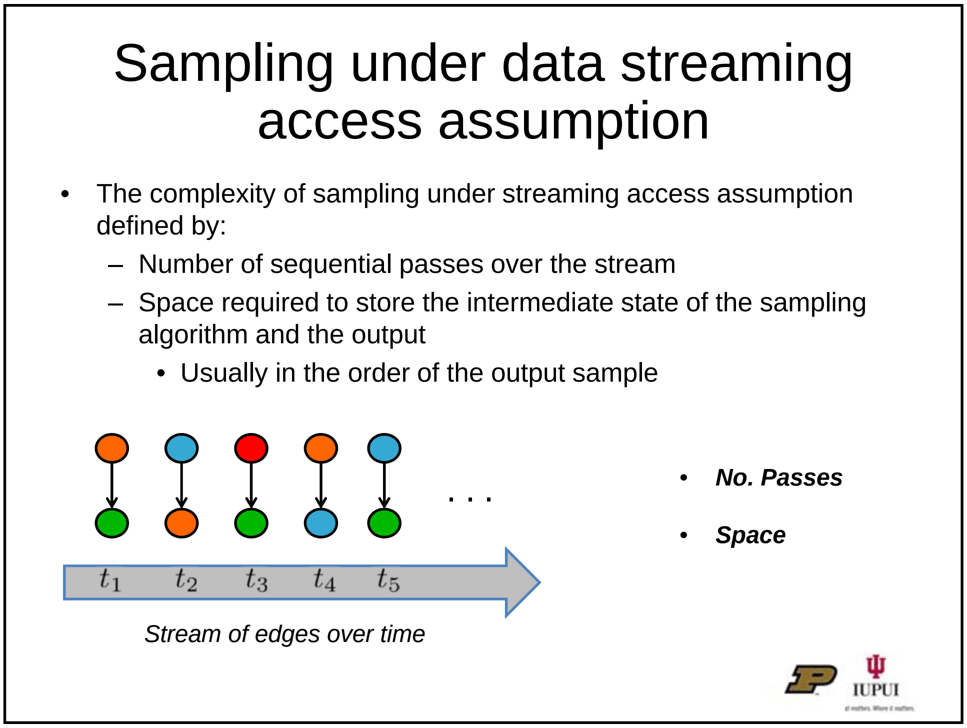
Sampling under data streaming access assumption

- Previous approaches assume:
  - Full access of the graph or Restricted access of the graph – access to only node’s neighbors
- Data streaming access assumption:
  - A graph is accessed only sequentially as a stream of edges
  - Massive stream of edges that cannot fit in main memory
  - Efficient/Real-time processing is important

. . .
t_1
t_2
t_3
t_4
t_5
t_2
t_4
t_5
t_3
t_1

Stream of edges over time

Graph with timestamps



Sampling under data streaming access assumption

- The complexity of sampling under streaming access assumption defined by:
  - – Number of sequential passes over the stream
  - – Space required to store the intermediate state of the sampling algorithm and the output
    - Usually in the order of the output sample

. . .
t_1
t_2
t_3
t_4
t_5

Stream of edges over time

- ***No. Passes***
- ***Space***

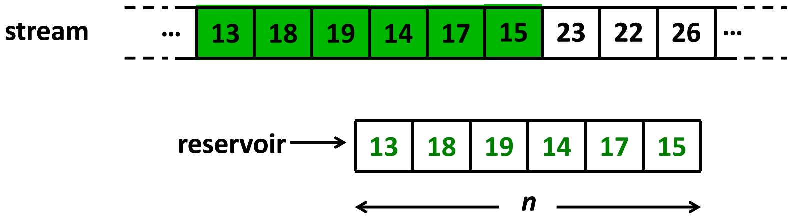
Sampling methods under data streaming access assumption

- Most stream sampling algorithms are based on *random reservoir sampling*

- Random Reservoir Sampling [Vitter'85]
  - A family of random algorithms for sampling from data streams
  - Choosing a set of *n* records from a large stream of *N* records
    - *n* << *N*
  - *N* is too large to fit in main memory and usually unknown
  - One pass, O(N) algorithm

Reservoir Sampling Algorithm

Step 1: Add first *n* records to reservoir

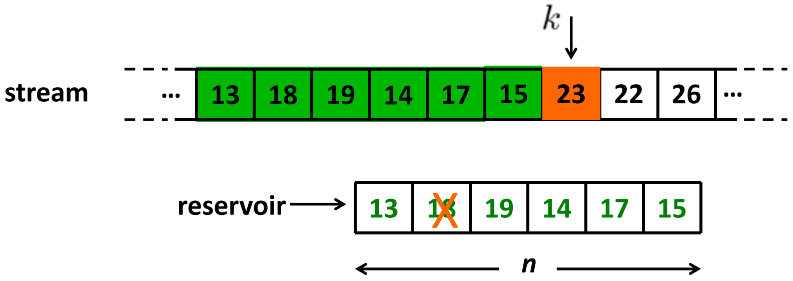


stream … 13 18 19 14 17 15 23 22 26 …

reservoir → 13 18 19 14 17 15

Reservoir Sampling Algorithm

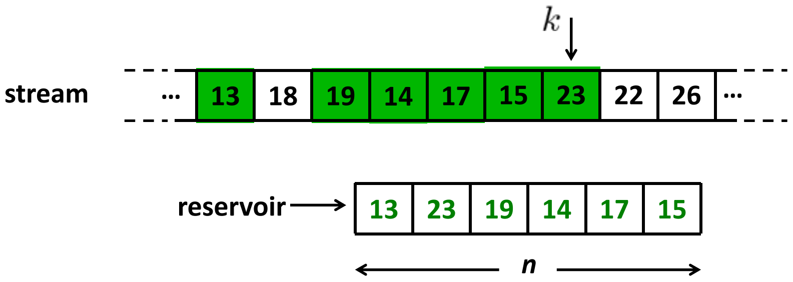
Step 2: Select the next record k with probability $P = n/k$



- When a record is chosen for the reservoir, it becomes a candidate and replaces one of the former candidates

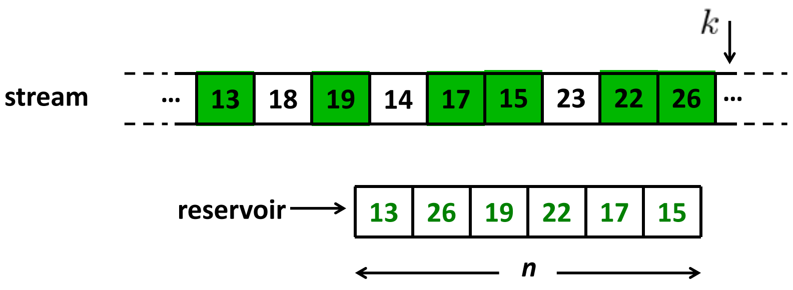
Reservoir Sampling Algorithm

Step 2: Select the next record k with probability $P = n/k$



Reservoir Sampling Algorithm

After repeating Step 2 …



k

stream … 13 18 19 14 17 15 23 22 26 …

reservoir → 13 26 19 22 17 15

← *n* →

- at the end of the sequential pass, the current set of *n* candidates is output as the final sample

Sampling methods under data streaming access assumption

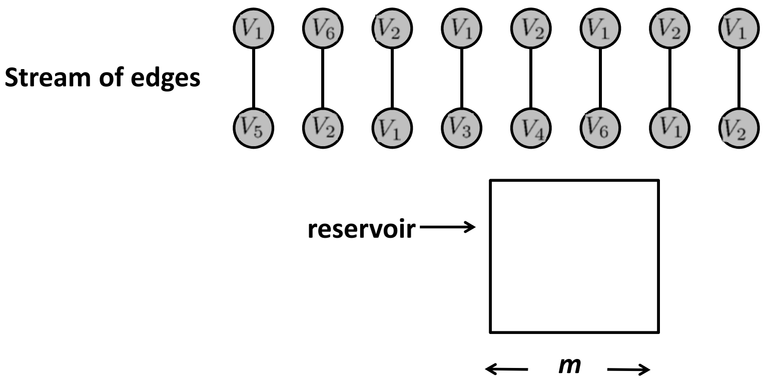
- Streaming Uniform Edge Sampling
  - Extends traditional random edge sampling (RE) to streaming access
- Streaming Uniform Node Sampling
  - Extends traditional random node sampling (RN) to streaming access

Sampling methods under data streaming access assumption

- Streaming Uniform Edge Sampling
  - Extends traditional random edge sampling (RE) to streaming access
- Streaming Uniform Node Sampling
  - Extends traditional random node sampling (RN) to streaming access

Streaming Edge Sampling (RE)

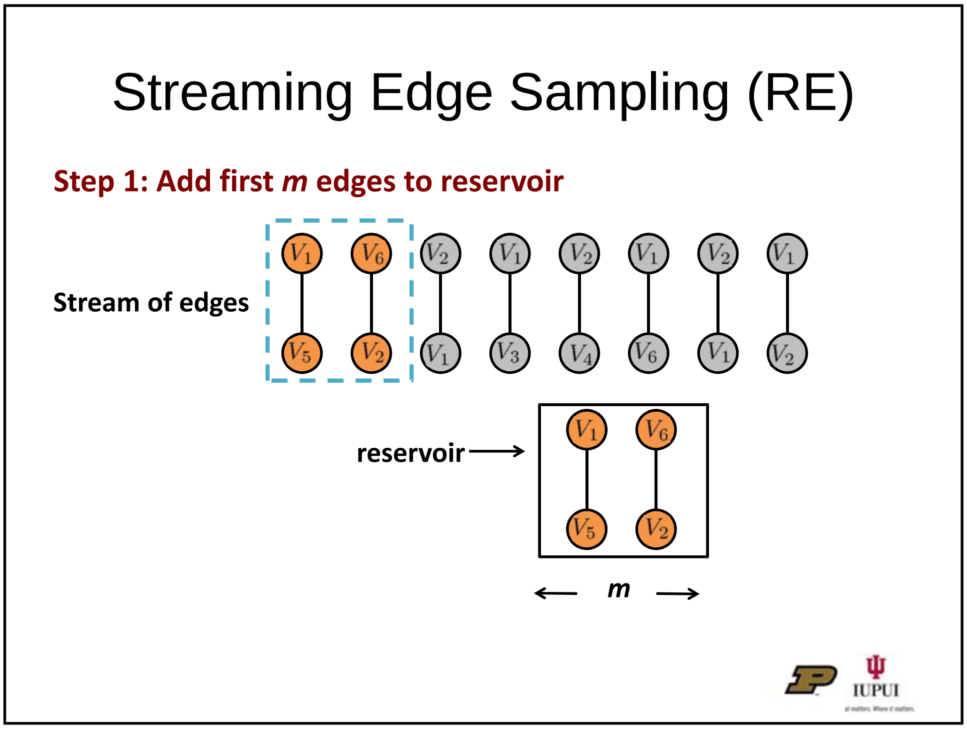
Step 0: Start with an empty reservoir of size m = 2



Stream of edges

reservoir

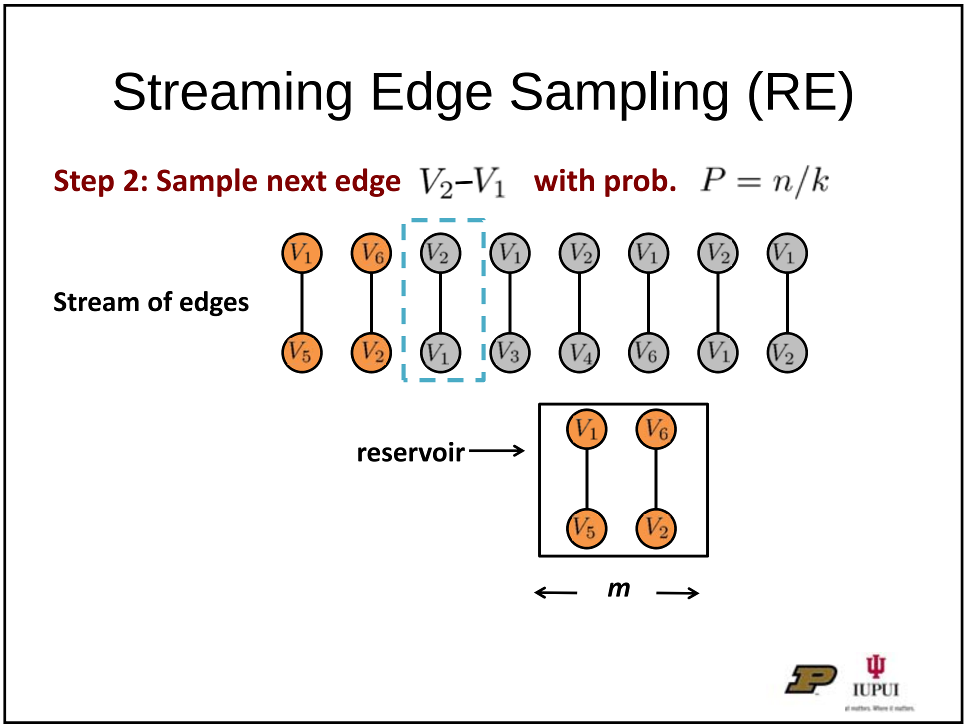
m



Streaming Edge Sampling (RE)

Step 1: Add first *m* edges to reservoir

Stream of edges
V_1 V_6 V_2 V_1 V_2 V_1 V_2 V_1
V_5 V_2 V_1 V_3 V_4 V_6 V_1 V_2
reservoir
V_1 V_6
V_5 V_2
m
IUPUI



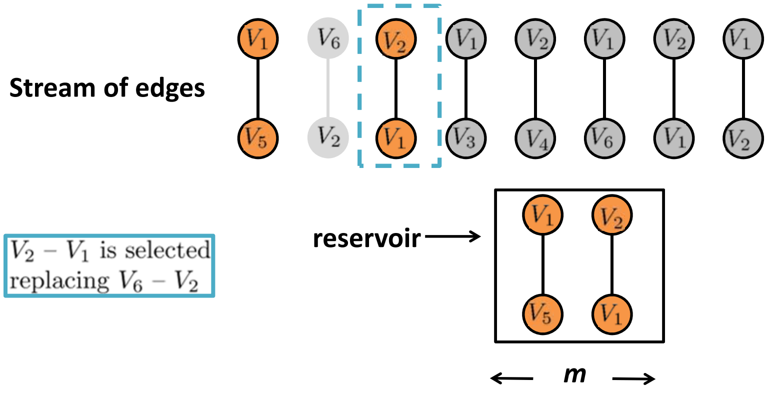
Streaming Edge Sampling (RE)

Step 2: Sample next edge V_2–V_1 **with prob.** $P = n/k$

Stream of edges
V_1 V_6 V_2 V_1 V_2 V_1 V_2 V_1
V_5 V_2 V_1 V_3 V_4 V_6 V_1 V_2
reservoir
V_1 V_6
V_5 V_2
m
IUPUI

Streaming Edge Sampling (RE)

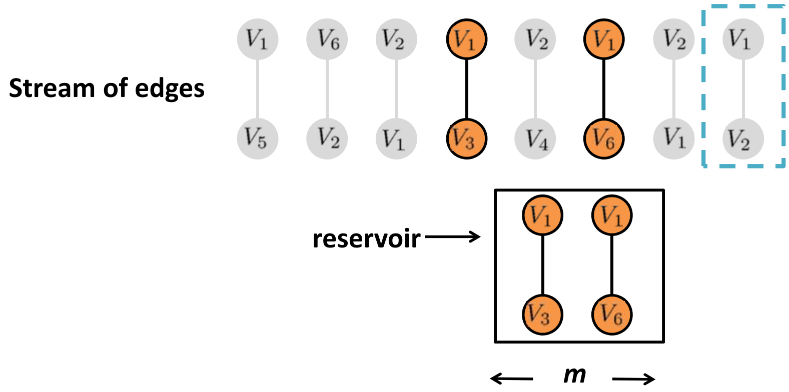
Edge V_2–V_1 is selected



$V_2 - V_1$ is selected replacing $V_6 - V_2$

Streaming Edge Sampling (RE)

At the end of the sequential pass . . .



- at the end of the sequential pass, the current set of *m edges* is output as the final sample

Min-Wise Sampling briefly …

- A random “tag" drawn independently from the Uniform(0, 1) distribution
- This “tag” is called the “hash value” associated with each arriving item
- The sample consists of the items with the *n* smallest tags/hash values seen thus far
- Uniformity follows by symmetry: every size-n subset of the stream has the same chance of having the smallest hash values

For any arriving item i

$h(i) \sim Uniform(0, 1)$

Sampling methods under data streaming access assumption

- Streaming Uniform Edge Sampling
  - Extends traditional random edge sampling (RE) to streaming access
- Streaming Uniform Node Sampling
  - Extends traditional random node sampling (RN) to streaming access

Sampling from the stream directly *selects nodes proportional to their degree*

Not suitable for sampling nodes *uniformly*

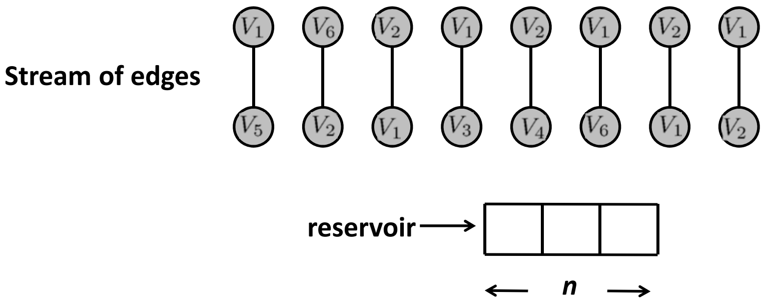
Use Min-Wise Sampling

Streaming Uniform Node Sampling (RN)

- Assumption: nodes arrive into the system only when an edge that contains the new node is streaming into the system
- It is difficult to identify which *n* nodes to select a priori with uniform probability
- Use min-wise sampling
- Maintain a reservoir of nodes with top-n minimum hash values
- a node with smaller hash value may arrive late in the stream and replace a node that was sampled earlier

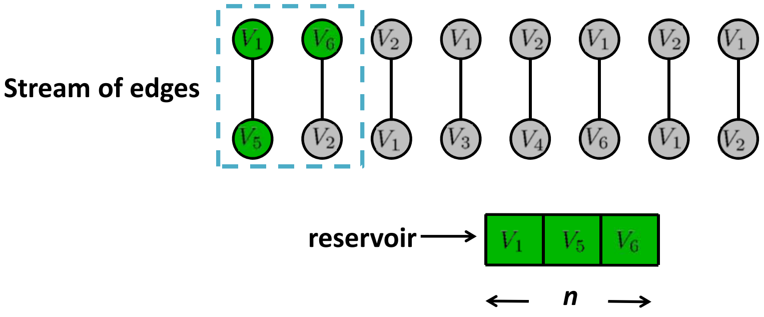
Streaming Uniform Node Sampling (RN) – Example

Step 0: Start with an empty reservoir

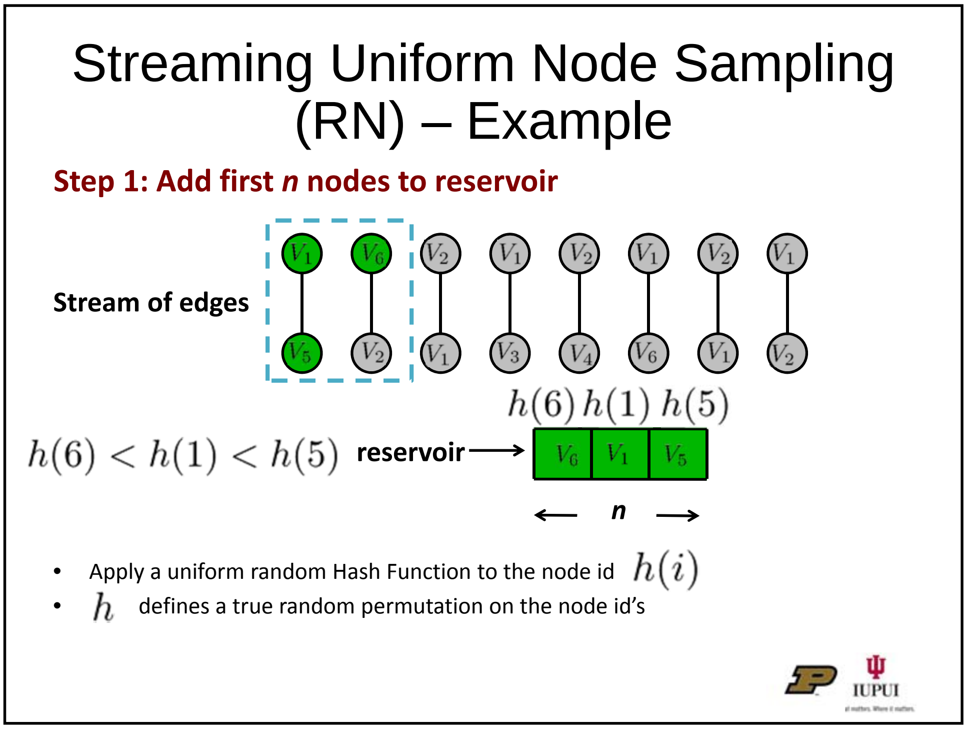


Streaming Uniform Node Sampling (RN) – Example

Step 1: Add first *n* nodes to reservoir



- Apply a uniform random Hash Function to the node id $h(i)$
- h defines a true random permutation on the node id's



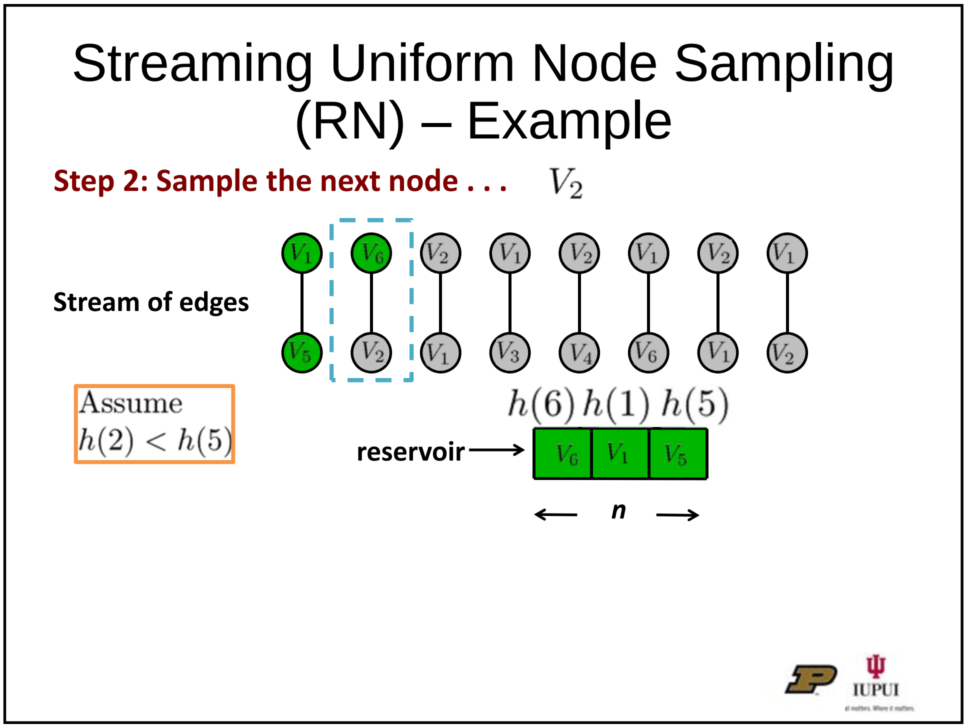
Streaming Uniform Node Sampling (RN) – Example

Step 1: Add first *n* nodes to reservoir

Stream of edges

$h(6) < h(1) < h(5)$ **reservoir** →

- Apply a uniform random Hash Function to the node id $h(i)$
- h defines a true random permutation on the node id's



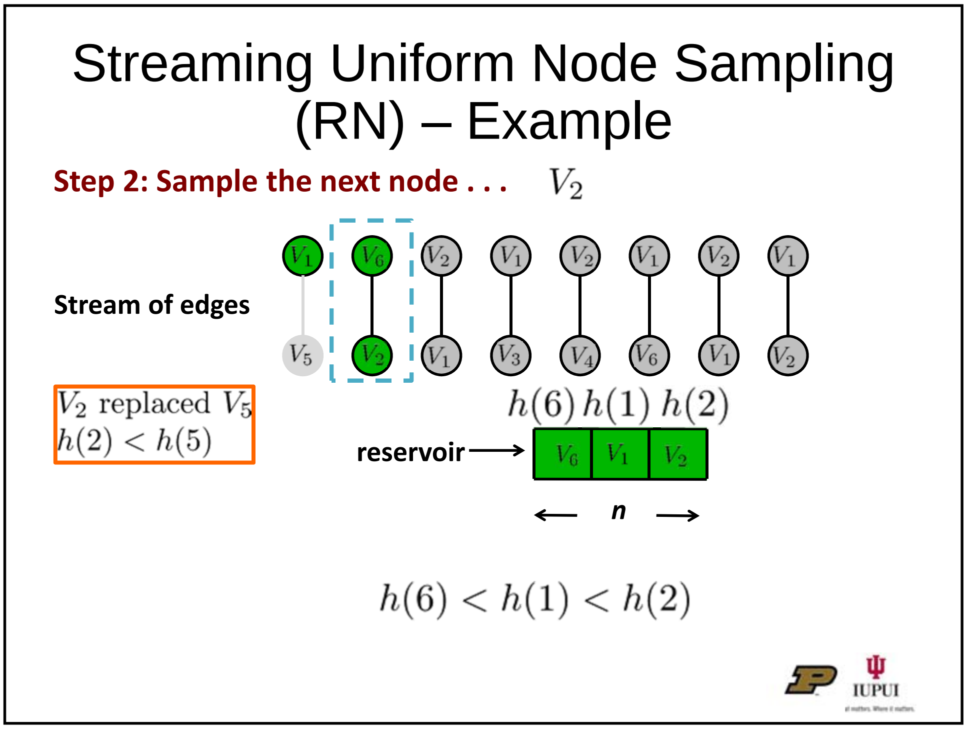
Streaming Uniform Node Sampling (RN) – Example

Step 2: Sample the next node . . . V_2

Stream of edges

Assume
$h(2) < h(5)$

reservoir →



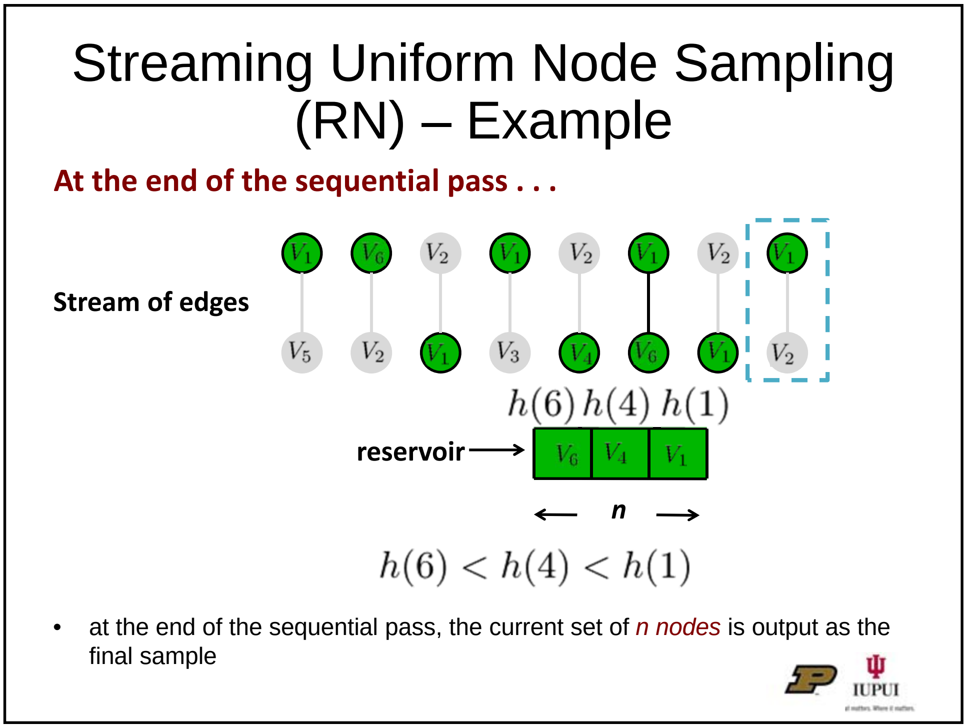
Streaming Uniform Node Sampling (RN) – Example

Step 2: Sample the next node . . . V_2

Stream of edges
V_1 V_6 V_2 V_1 V_2 V_1 V_2 V_1
V_5 V_2 V_1 V_3 V_4 V_6 V_1 V_2
$h(6)\,h(1)\,h(2)$
reservoir
V_6 V_1 V_2
n

V_2 replaced V_5
$h(2) < h(5)$

$$h(6) < h(1) < h(2)$$



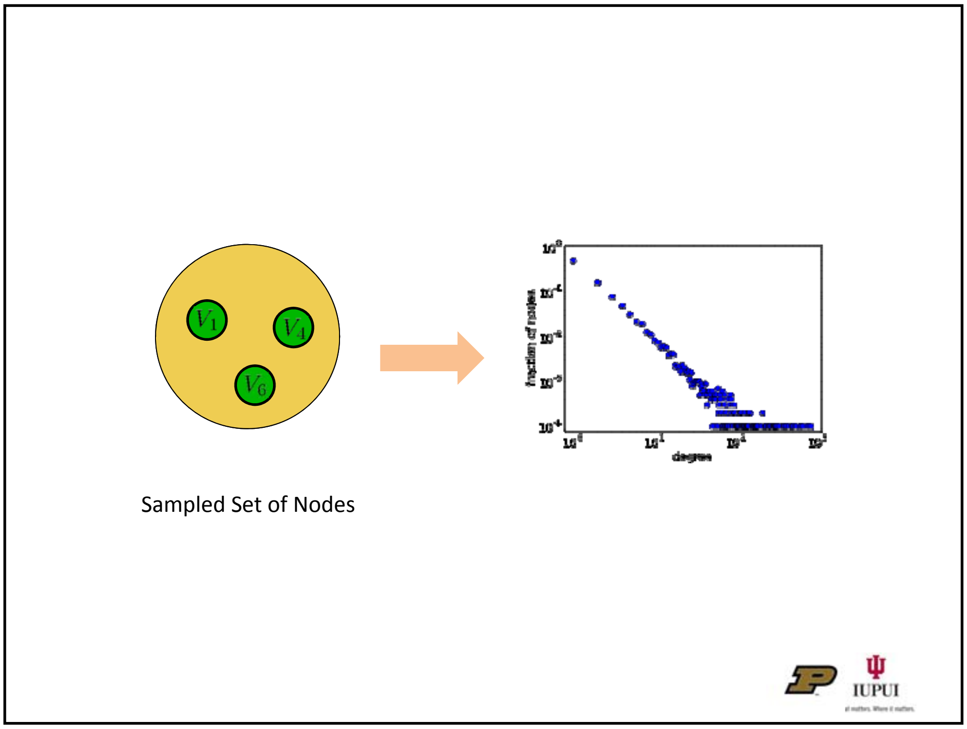
Streaming Uniform Node Sampling (RN) – Example

At the end of the sequential pass . . .

Stream of edges
V_1 V_6 V_2 V_1 V_2 V_1 V_2 V_1
V_5 V_2 V_1 V_3 V_4 V_6 V_1 V_2
$h(6)\,h(4)\,h(1)$
reservoir
V_6 V_4 V_1
n

$$h(6) < h(4) < h(1)$$

- at the end of the sequential pass, the current set of *n nodes* is output as the final sample

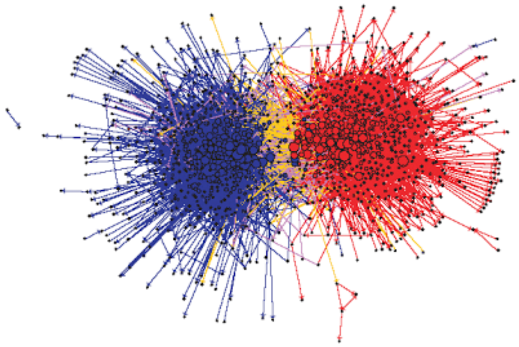


Sampled Set of Nodes

TASK 2

Representative Sub-network Sampling

Real-life example: People believe what they want to believe!

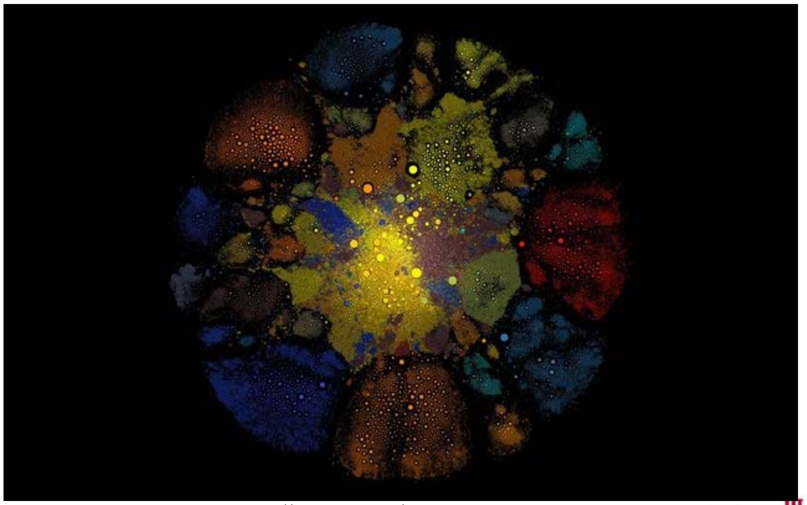


Researchers conclusion:

- people seek out information consistent with their world views.
- In doing so, they are less exposed to new ideas that do not comport with their ideology.
- The tight clustering suggests that "many of these writers are simply echoing each other"

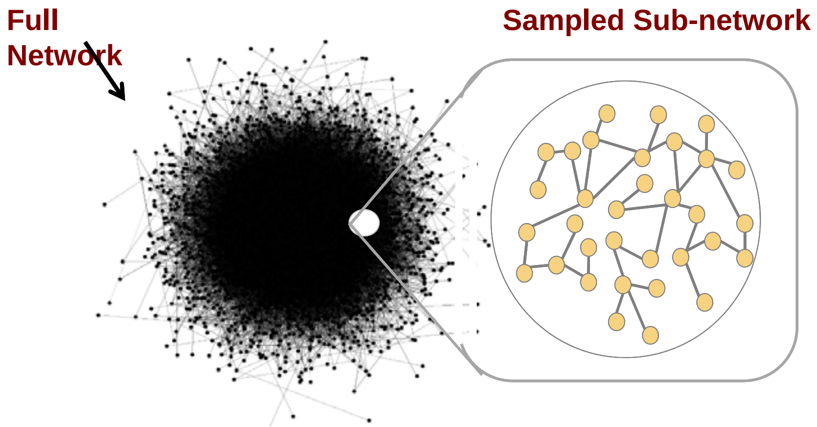
Linking pattern among conservative (red) and liberal (blue) blogs (from Lada Adamic and Natalie Glance, LinkKDD 2005)

How to sample a representative set of communities from MMOG networks?



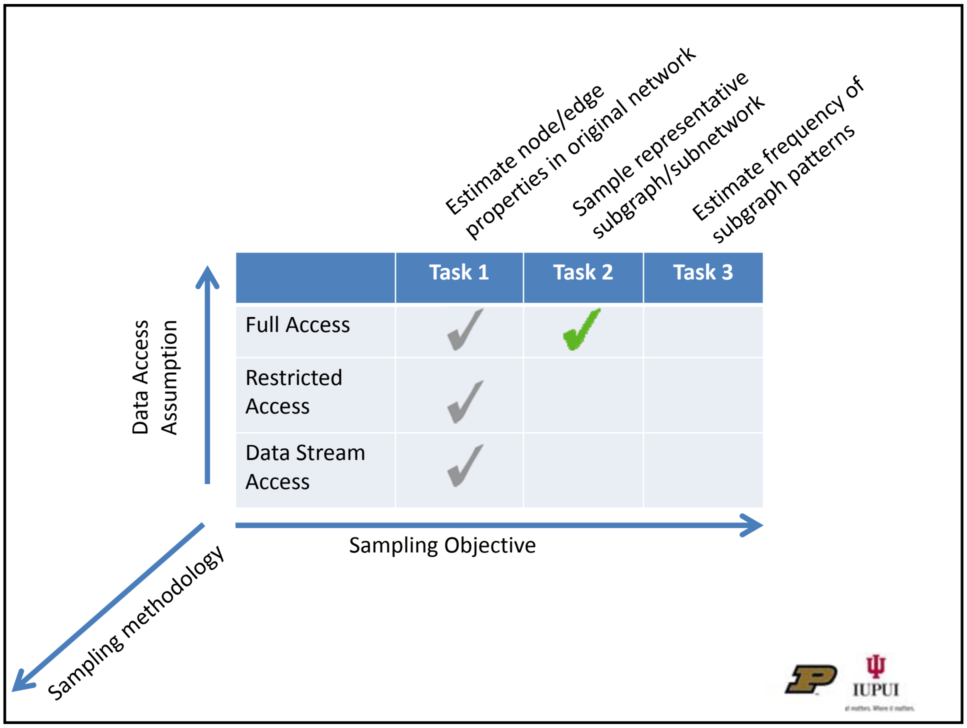
http://forum.gephi.org/viewtopic.php?t=2314

Representative Sub-network Sampling



- Select a representative sampled sub-network

- The sample is representative if its structural properties are similar to the full network



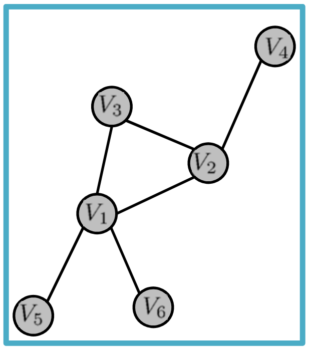
	Estimate node/edge properties in original network	Sample representative subgraph/subnetwork	Estimate frequency of subgraph patterns
	Task 1	Task 2	Task 3
Full Access	✓	✓	
Restricted Access	✓		
Data Stream Access	✓		

Sampling methods under full access assumptions

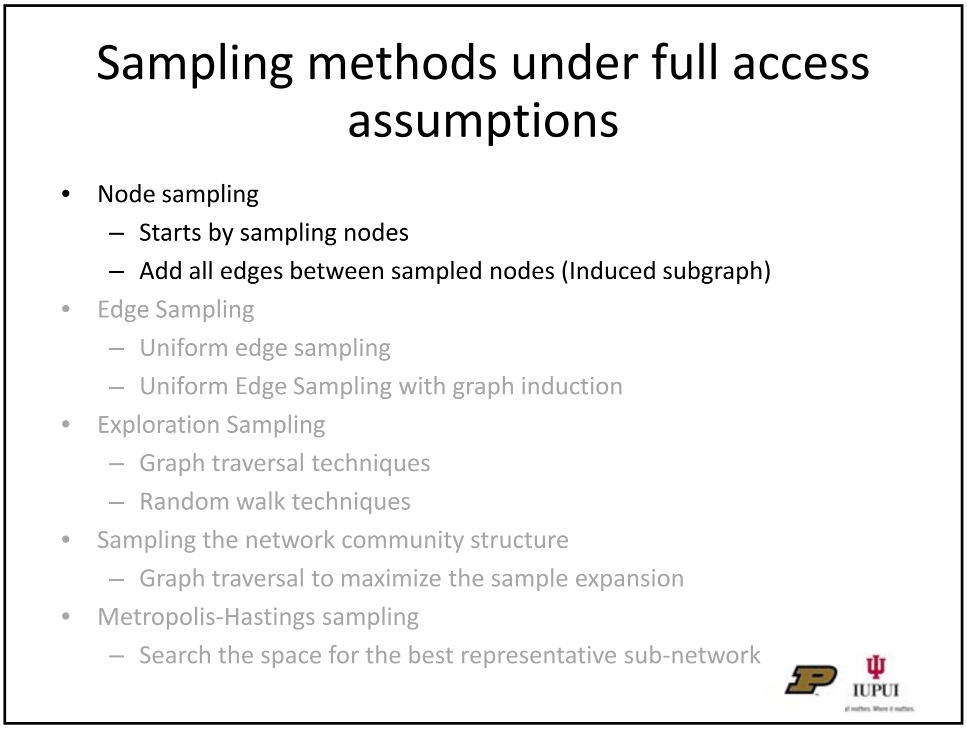
- Node sampling
  - Starts by sampling nodes
  - Add all edges between sampled nodes (Induced subgraph)
- Edge Sampling
  - Uniform edge sampling
  - Uniform Edge Sampling with graph induction
- Exploration Sampling
  - Graph traversal techniques
  - Random walk techniques
- Sampling the network community structure
  - Graph traversal to maximize the sample expansion
- Metropolis-Hastings sampling
  - Search the space for the best representative sub-network

- Assume we have the following network

- **Goal:** sample a representative sub-network of size = 4 nodes

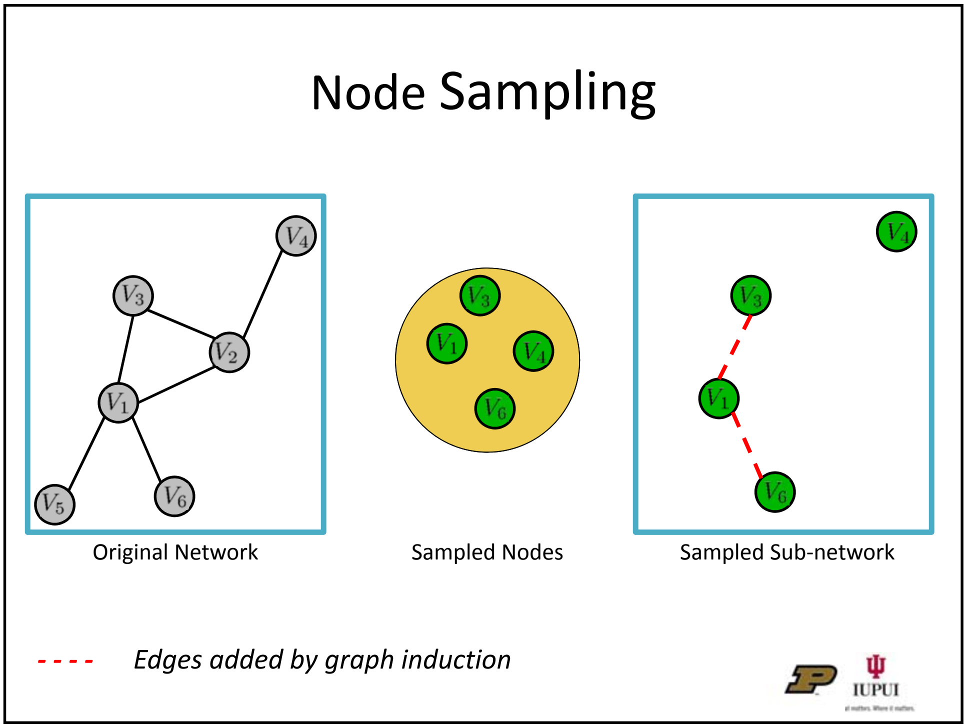


Original Network



Sampling methods under full access assumptions

- Node sampling
  - Starts by sampling nodes
  - Add all edges between sampled nodes (Induced subgraph)
- Edge Sampling
  - Uniform edge sampling
  - Uniform Edge Sampling with graph induction
- Exploration Sampling
  - Graph traversal techniques
  - Random walk techniques
- Sampling the network community structure
  - Graph traversal to maximize the sample expansion
- Metropolis-Hastings sampling
  - Search the space for the best representative sub-network



Node Sampling

Original Network

Sampled Nodes

Sampled Sub-network

- - - - *Edges added by graph induction*

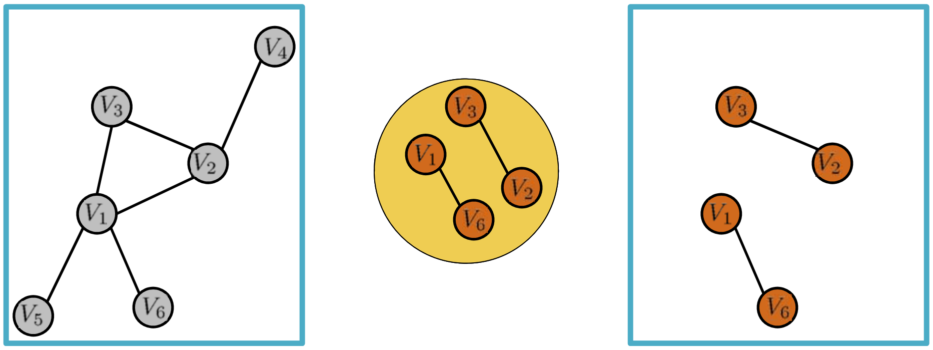
Node Sampling (NS)

- Node sampling samples nodes uniformly
- Generally biased to low degree nodes
- Sampled sub-network may include isolated nodes (zero degree)
- The sampled sub-network typically fails to preserve many properties of the original network
  - diameter, hop-plot, clustering co-efficient, and centrality values of vertices.

Sampling methods under full access assumptions

- Node sampling
  - Starts by sampling nodes
  - Add all edges between sampled nodes (Induced subgraph)
- Edge Sampling
  - Uniform edge sampling
  - Uniform Edge Sampling with graph induction
- Exploration Sampling
  - Graph traversal techniques
  - Random walk techniques
- Sampling the network community structure
  - Graph traversal to maximize the sample expansion
- Metropolis-Hastings sampling
  - Search the space for the best representative sub-network

Uniform Edge Sampling (ES)



Original Network

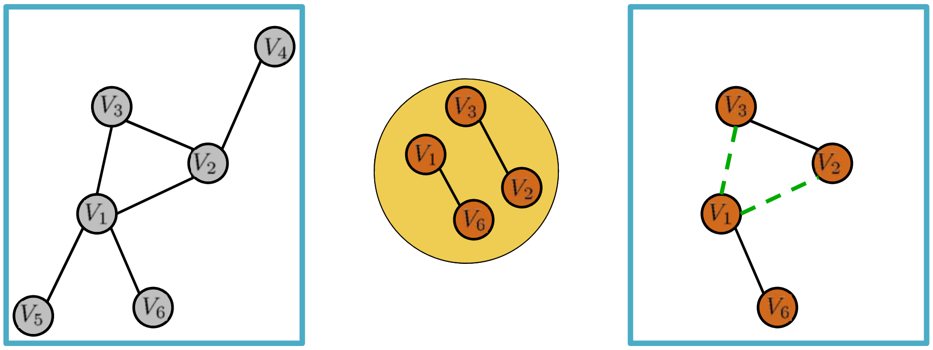
Sampled Edges

Sampled Sub-Network

Edge Sampling (ES)

- Edge sampling samples nodes proportional to their degree
- Generally biased to high degree nodes (hub nodes)
- The sampled sub-network typically fails to preserve many properties of the original network
  - diameter, hop-plot, clustering co-efficient, and centrality values of vertices.

Uniform Edge Sampling with graph induction (ES-i) [Ahmed '13]



Original Network

Sampled Edges

Sampled Sub-Network

- - - - *Edges added by graph induction*

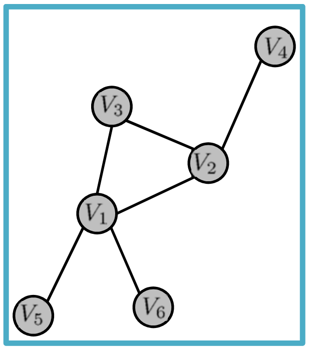
Edge Sampling with graph induction (ES-i)

- Similar to the edge sampling (ES)
  - ES-i samples nodes proportional to their degree
  - Generally biased to high degree nodes (hub nodes)
- Due to the additional graph induction step
  - The sampled sub-network preserves many properties of the original network
    - degree, diameter, hop-plot, clustering co-efficient, and centrality values of vertices.

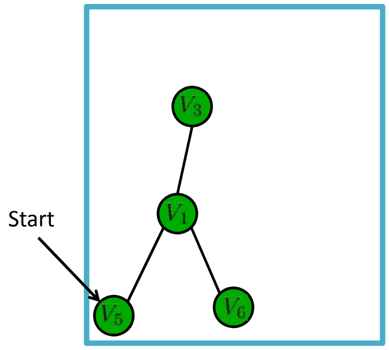
Sampling methods under full access assumptions

- Node sampling
  - Starts by sampling nodes
  - Add all edges between sampled nodes (Induced subgraph)
- Edge Sampling
  - Uniform edge sampling
  - Uniform Edge Sampling with graph induction
- Exploration Sampling
  - Graph traversal techniques
  - Random walk techniques
- Sampling the network community structure
  - Graph traversal to maximize the sample expansion
- Metropolis-Hastings sampling
  - Search the space for the best representative sub-network

Exploration Sampling – Breadth First



Original Network



Sampled Sub-Network

Sampled sub-network consists of all nodes/edges visited

Exploration Sampling

- Generally biased to high degree nodes (hub nodes)
- The sampled sub-network can preserve the diameter of the network if enough nodes are visited
- The sampled sub-network usually fails to preserve clustering co-efficient
- One solution: induce the sampled sub-network

Sampling methods under full access assumptions

- Node sampling
  - Starts by sampling nodes
  - Add all edges between sampled nodes (Induced subgraph)
- Edge Sampling
  - Uniform edge sampling
  - Uniform Edge Sampling with graph induction
- Exploration Sampling
  - Graph traversal techniques
  - Random walk techniques
- Sampling the network community structure
  - Graph traversal to maximize the sample expansion
- Metropolis-Hastings sampling
  - Search the space for the best representative sub-network

Sampling Community Structure [Maiya'10]

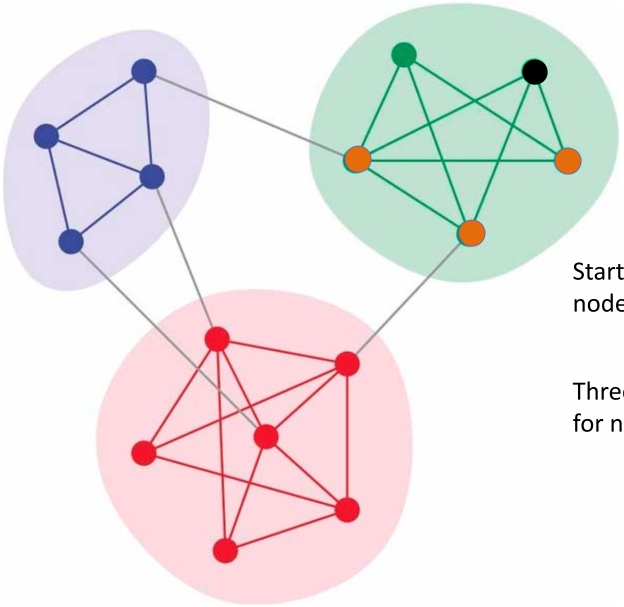
- Sampling a sub-network representative of the original network community structure
  - Expansion Sampling (XS)

- Expansion Sampling Algorithm:
  - Step1: Starts with a random vertex *u*, add *u* to the sample set *S*

    S = S U {u}
  - Step2: Choose *v* from *S's* neighborhood, s.t. *v* maximizes

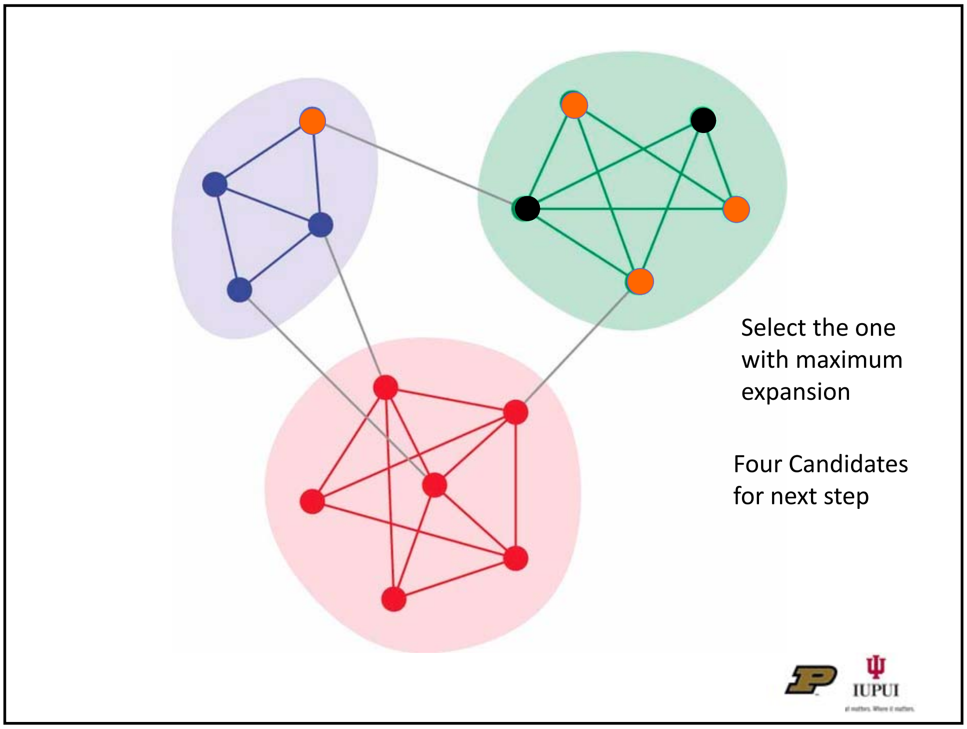
    |Neigh(v) – {Neigh(S) U S}|
  - Step3: Add *v* to *S*

    S = S U {v}
  - Repeat Step 2 and 3 until **|S| = k**



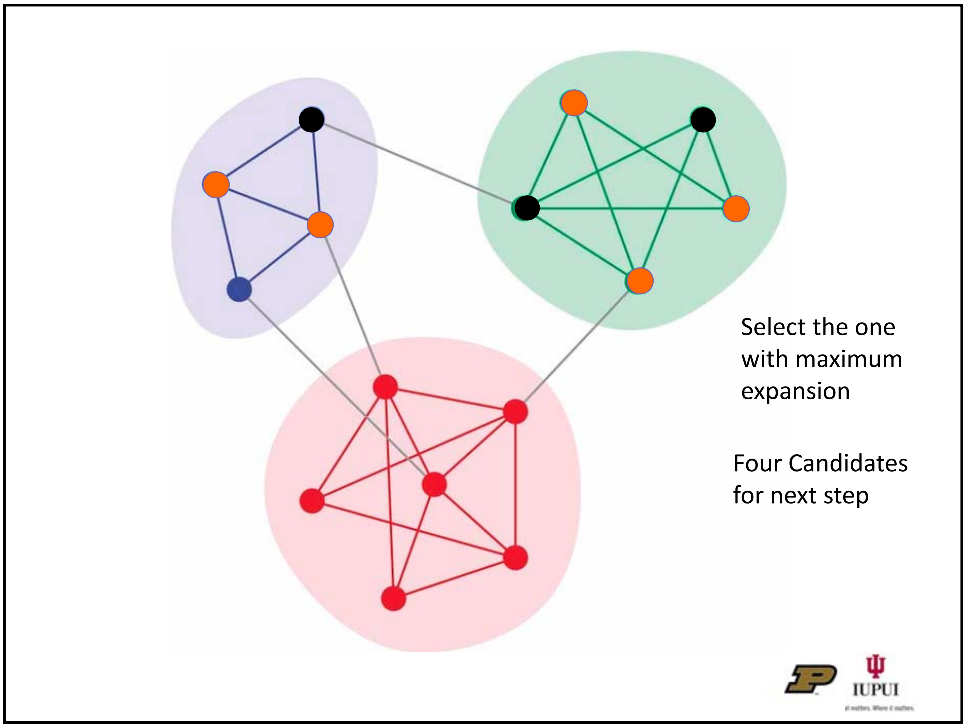
Start with random node

Three Candidates for next step



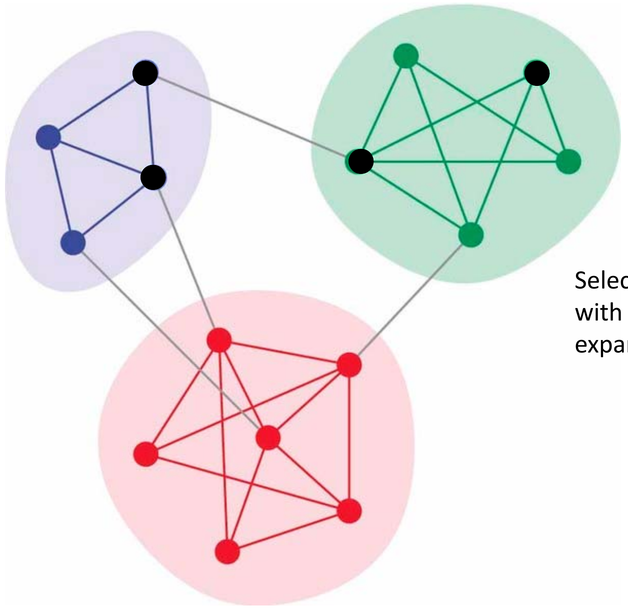
Select the one with maximum expansion

Four Candidates for next step



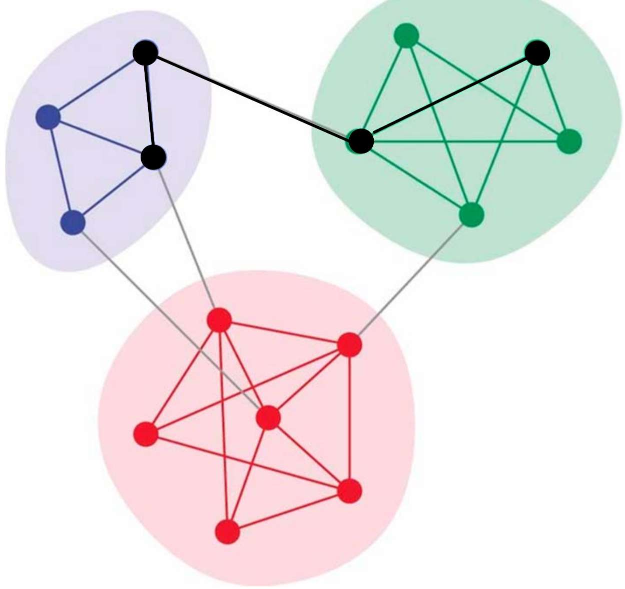
Select the one with maximum expansion

Four Candidates for next step



Select the one with maximum expansion

Sampled Sub-network



Sampling methods under full access assumptions

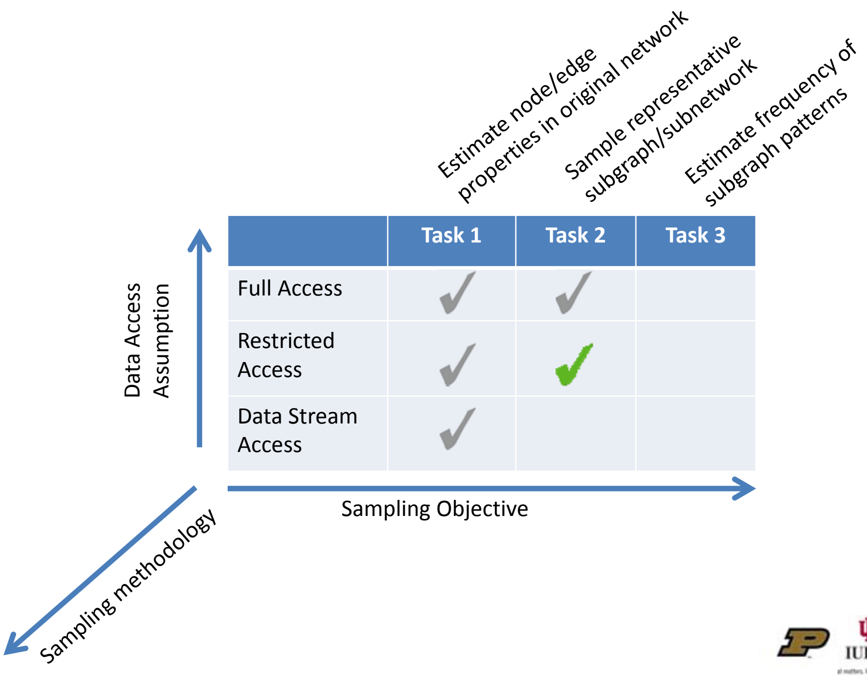
- Node sampling
  - Starts by sampling nodes
  - Add all edges between sampled nodes (Induced subgraph)
- Edge Sampling
  - Uniform edge sampling
  - Uniform Edge Sampling with graph induction
- Exploration Sampling
  - Graph traversal techniques
  - Random walk techniques
- Sampling the network community structure
  - Graph traversal to maximize the sample expansion
- Metropolis-Hastings sampling
  - Search the space for the best representative sub-network

Representative subgraph sampling by Metropolis Algorithm [Hubler'08]

- Given a graph, we want to generate a smaller graph which is representative to the given graph
  - We assume that the given graph is entirely visible
  - We can compute some characteristics of the given network that we want to compute
  - If $\sigma(G)$ defines a value (singular or multi-dimension) defining a topological property of G, and $S \subset G$ is an induced subgraph of G, our objective is to find a subgraph S that satisfies the following optimization problem

$$\underset{S \subset G : |S| = k}{\operatorname{argmin}} \ \Delta\big(\sigma(S), \sigma(G)\big)$$

- Characteristics that are generally considered
  - Degree distribution, clustering co-efficient, hop-plot distribution, small subgraph concentration



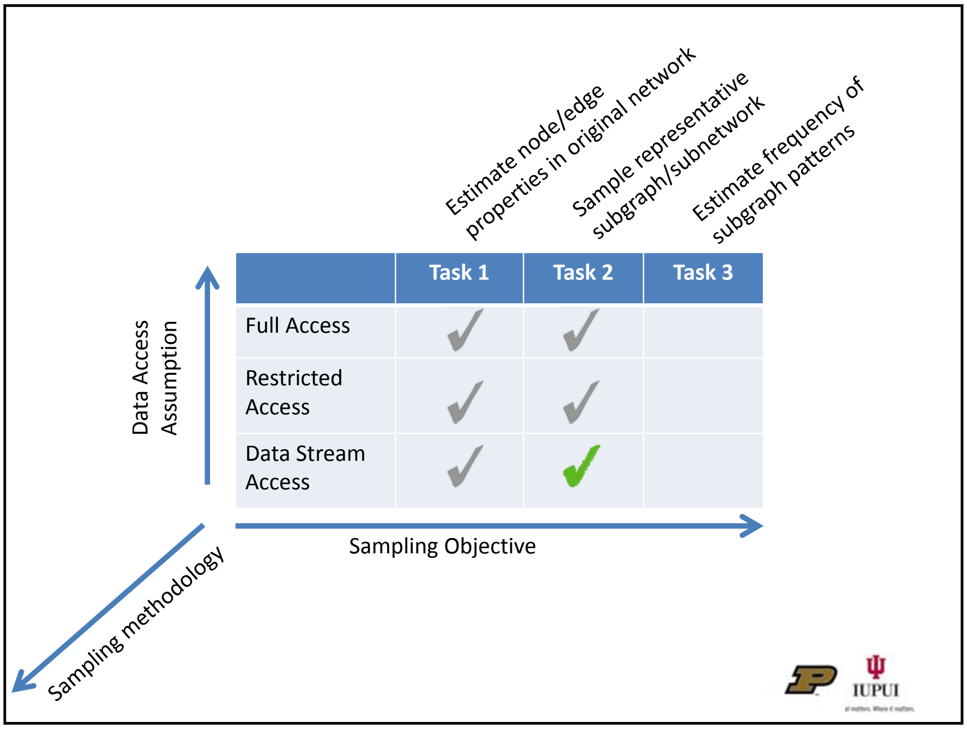
Estimate node/edge properties in original network
Sample representative subgraph/subnetwork
Estimate frequency of subgraph patterns
Data Access Assumption
Sampling Objective
Sampling methodology

	Task 1	Task 2	Task 3
Full Access	✓	✓	
Restricted Access	✓	✓	
Data Stream Access	✓		

Sampling methods under restricted access assumptions

- Exploration Sampling
  - Graph traversal techniques
  - Random walk techniques

- Sampling nodes or edges is not feasible under restricted access assumption



Estimate node/edge properties in original network

Sample representative subgraph/subnetwork

Estimate frequency of subgraph patterns

	Task 1	Task 2	Task 3
Full Access	✓	✓	
Restricted Access	✓	✓	
Data Stream Access	✓	✓	

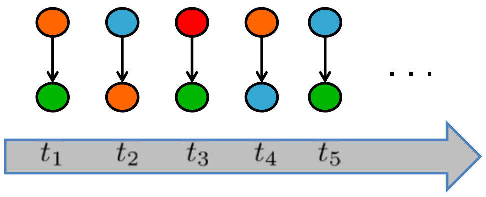
Data Access Assumption

Sampling Objective

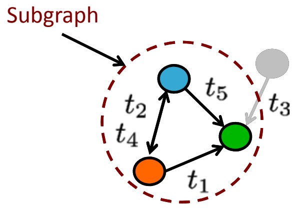
Sampling methodology

Sampling under data streaming access assumption

- Sampling a subgraph (sub-network) while the graph is streaming as a sequence of edges of any arbitrary order



Stream of edges over time



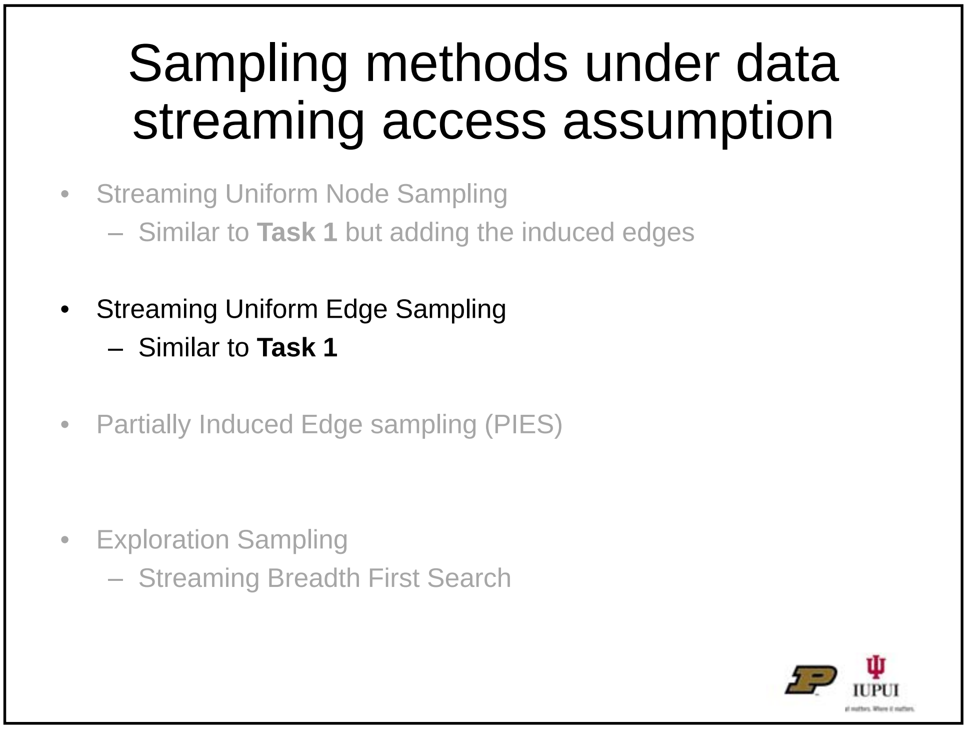
Graph with timestamps

Sampling methods under data streaming access assumption

- Streaming Uniform Node Sampling
- Streaming Uniform Edge Sampling
- Partially Induced Edge sampling (PIES)
- Exploration Sampling
  - Streaming Breadth First Search

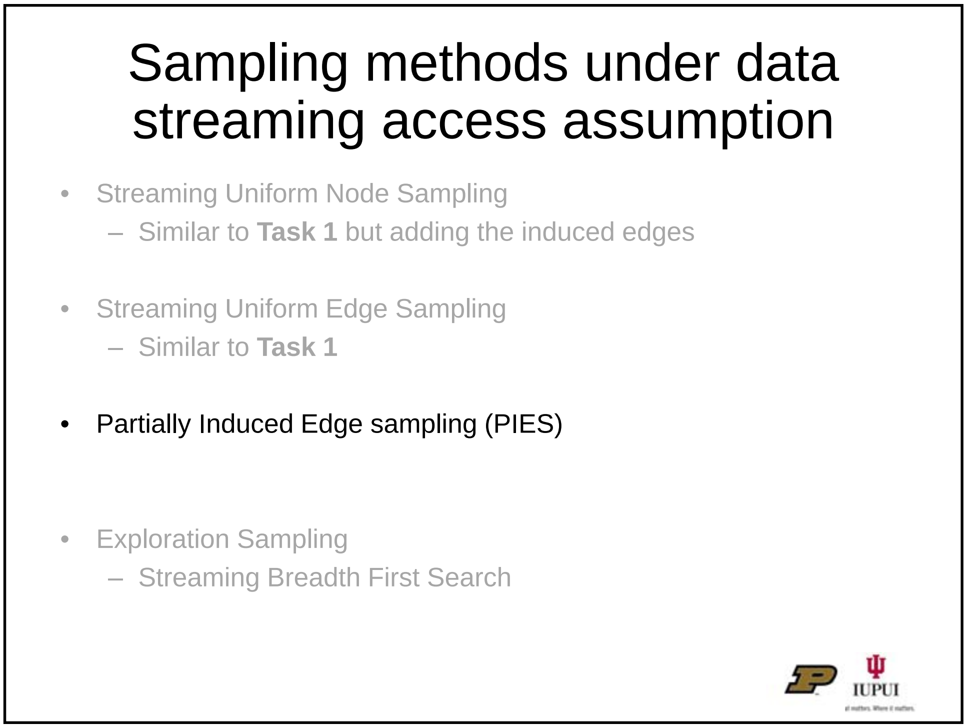
Sampling methods under data streaming access assumption

- Streaming Uniform Node Sampling
  - Similar to **Task 1** but adding the edges from the graph induction
- Streaming Uniform Edge Sampling
- Partially Induced Edge sampling (PIES)
- Exploration Sampling
  - Streaming Breadth First Search



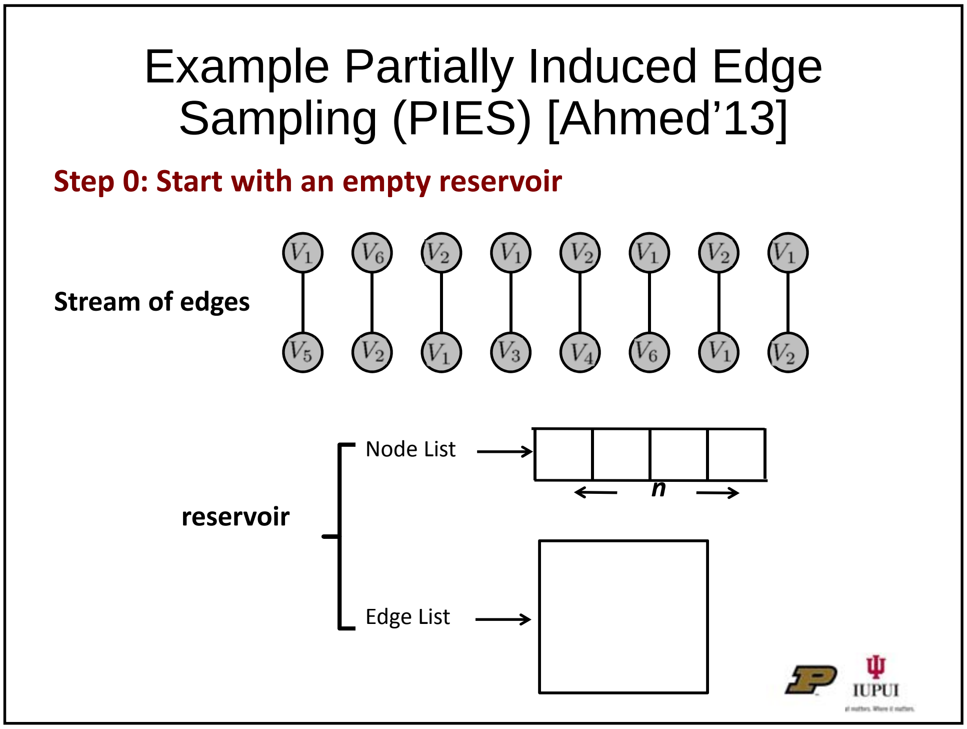
Sampling methods under data streaming access assumption

- Streaming Uniform Node Sampling
  - Similar to **Task 1** but adding the induced edges
- Streaming Uniform Edge Sampling
  - Similar to **Task 1**
- Partially Induced Edge sampling (PIES)
- Exploration Sampling
  - Streaming Breadth First Search



Sampling methods under data streaming access assumption

- Streaming Uniform Node Sampling
  - Similar to **Task 1** but adding the induced edges
- Streaming Uniform Edge Sampling
  - Similar to **Task 1**
- Partially Induced Edge sampling (PIES)
- Exploration Sampling
  - Streaming Breadth First Search



Example Partially Induced Edge Sampling (PIES) [Ahmed'13]

Step 0: Start with an empty reservoir

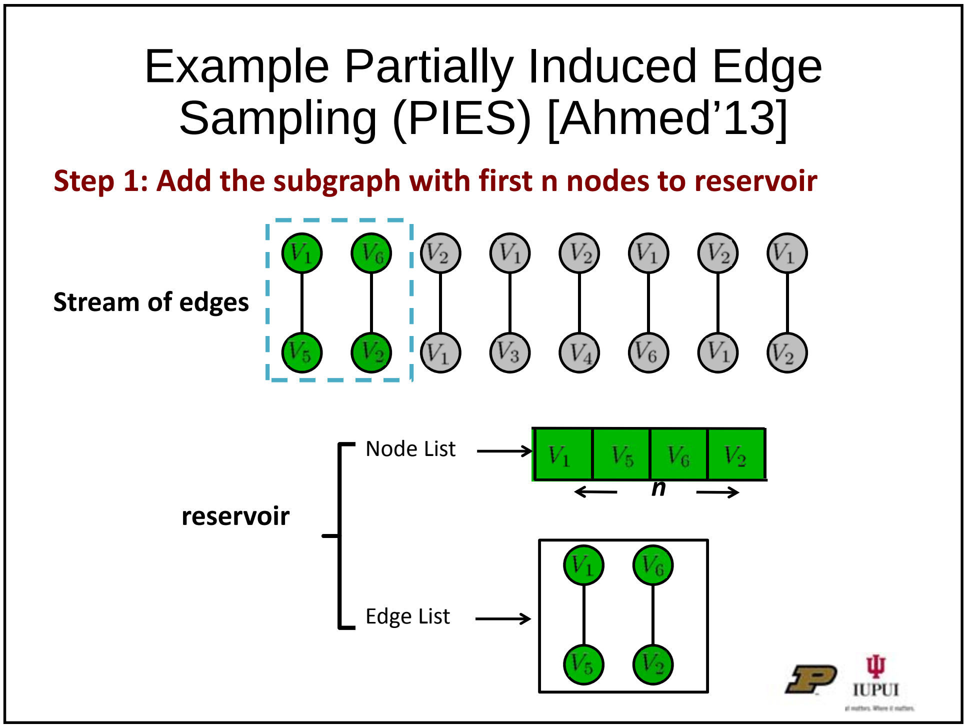
Stream of edges

reservoir

Node List

n

Edge List



Example Partially Induced Edge Sampling (PIES) [Ahmed'13]

Step 1: Add the subgraph with first n nodes to reservoir

Stream of edges

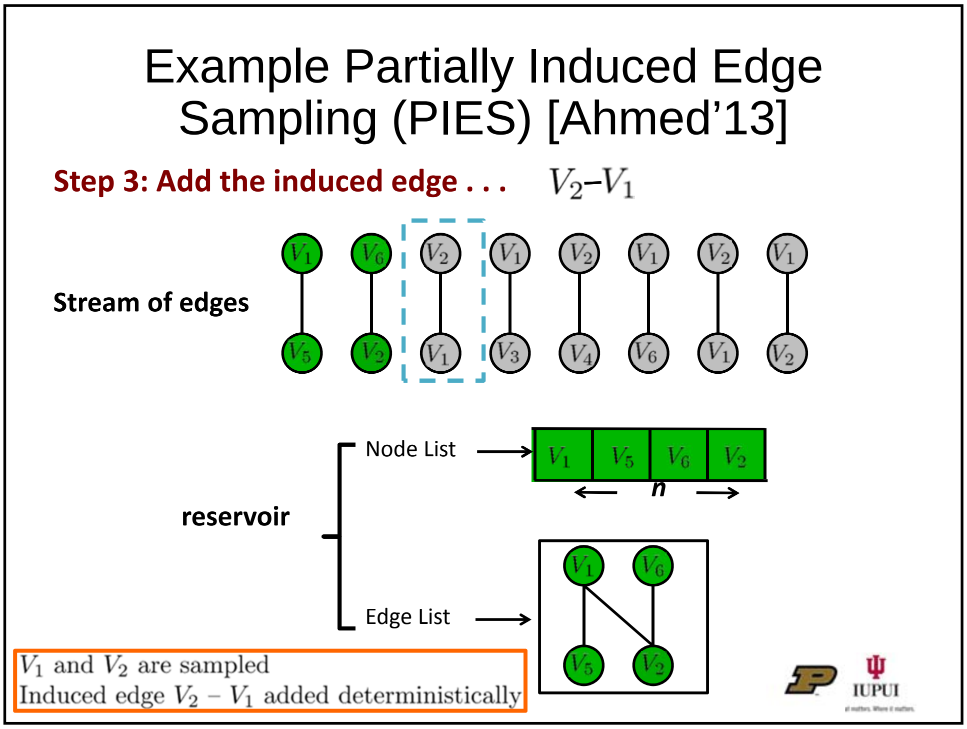
reservoir

Node List

V_1	V_5	V_6	V_2

n

Edge List



Example Partially Induced Edge Sampling (PIES) [Ahmed'13]

Step 3: Add the induced edge . . . V_2–V_1

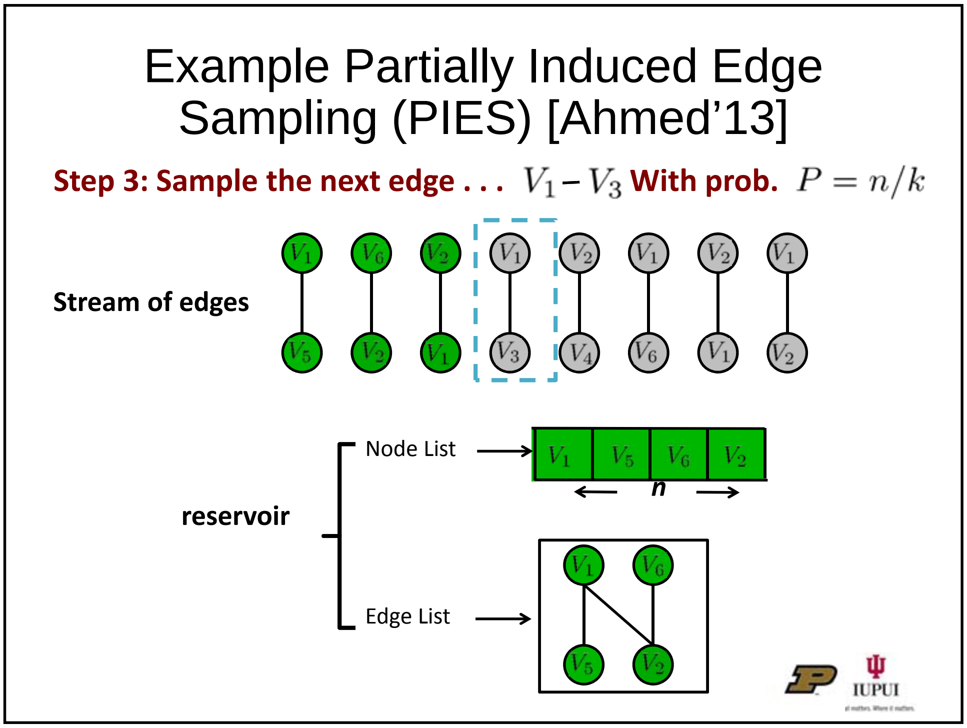
Stream of edges

reservoir

Node List

Edge List

V_1 and V_2 are sampled
Induced edge $V_2 - V_1$ added deterministically



Example Partially Induced Edge Sampling (PIES) [Ahmed'13]

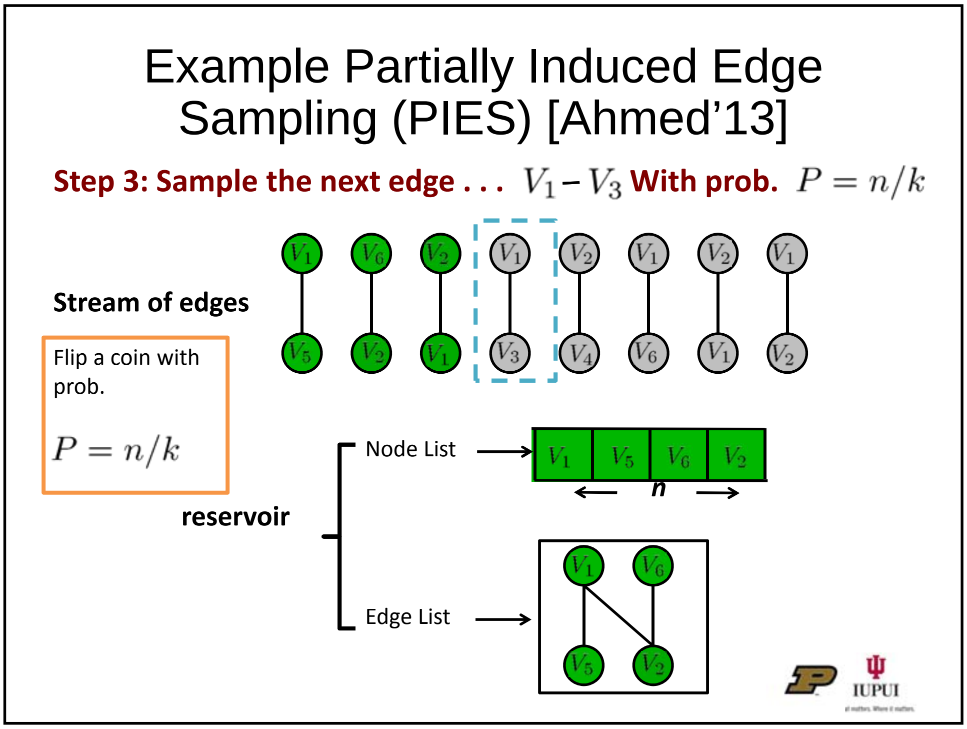
Step 3: Sample the next edge . . . $V_1 - V_3$ **With prob.** $P = n/k$

Stream of edges

reservoir

Node List

Edge List



Example Partially Induced Edge Sampling (PIES) [Ahmed'13]

Step 3: Sample the next edge . . . $V_1 - V_3$ **With prob.** $P = n/k$

Stream of edges

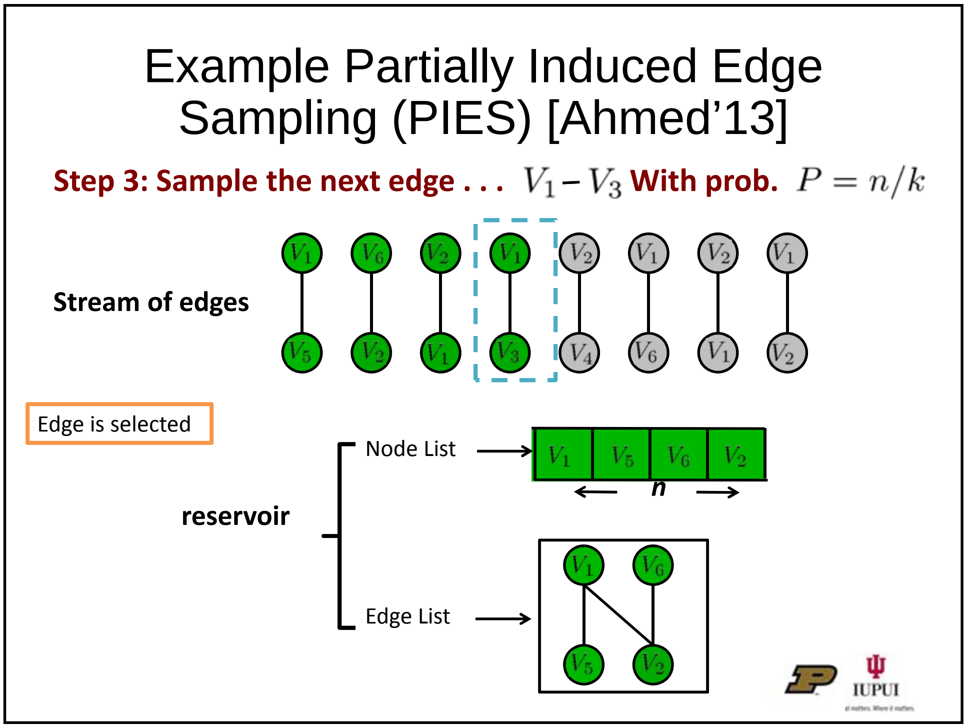
Flip a coin with prob.

$P = n/k$

reservoir

Node List

Edge List



Example Partially Induced Edge Sampling (PIES) [Ahmed'13]

Step 3: Sample the next edge . . . $V_1 - V_3$ **With prob.** $P = n/k$

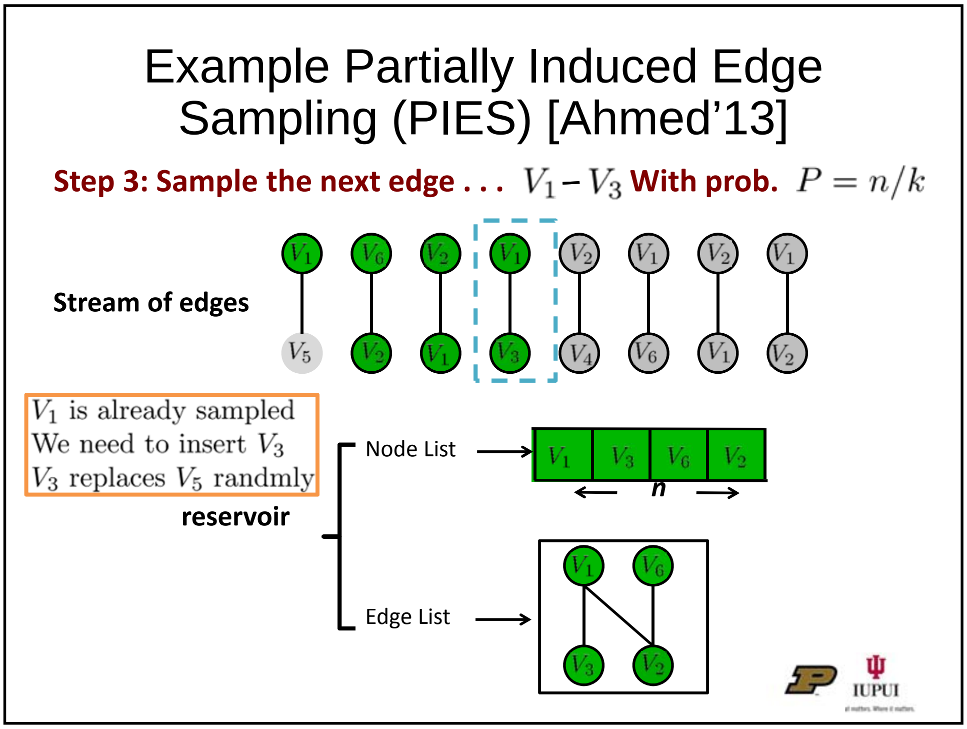
Stream of edges

Edge is selected

reservoir

Node List

Edge List



Example Partially Induced Edge Sampling (PIES) [Ahmed'13]

Step 3: Sample the next edge . . . $V_1 – V_3$ **With prob.** $P = n/k$

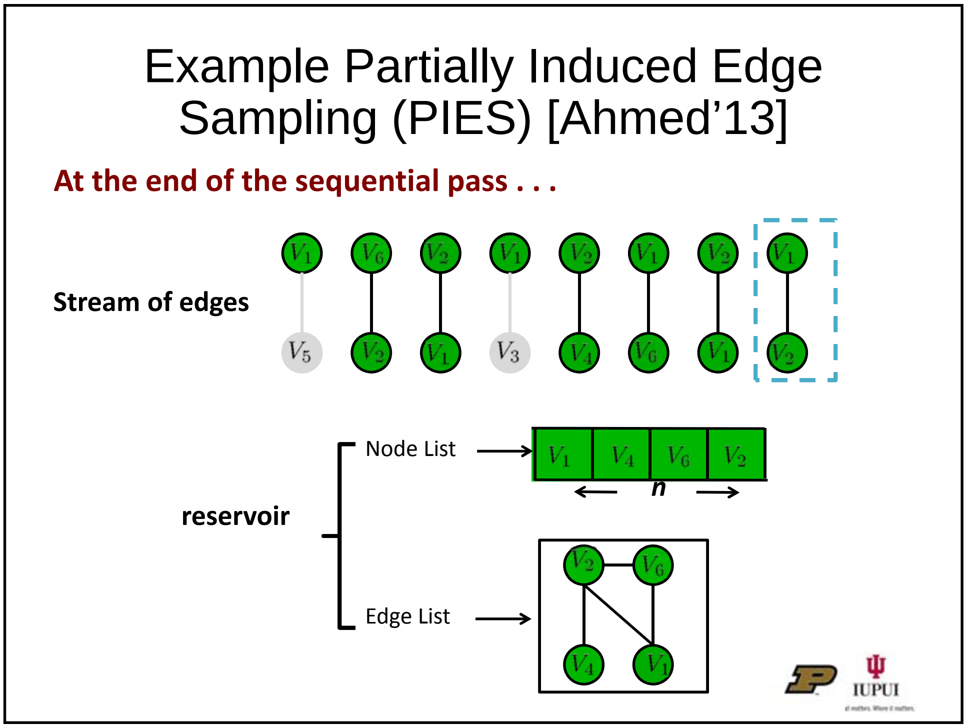
Stream of edges

V_1 is already sampled
We need to insert V_3
V_3 replaces V_5 randmly

reservoir

Node List

Edge List



Example Partially Induced Edge Sampling (PIES) [Ahmed'13]

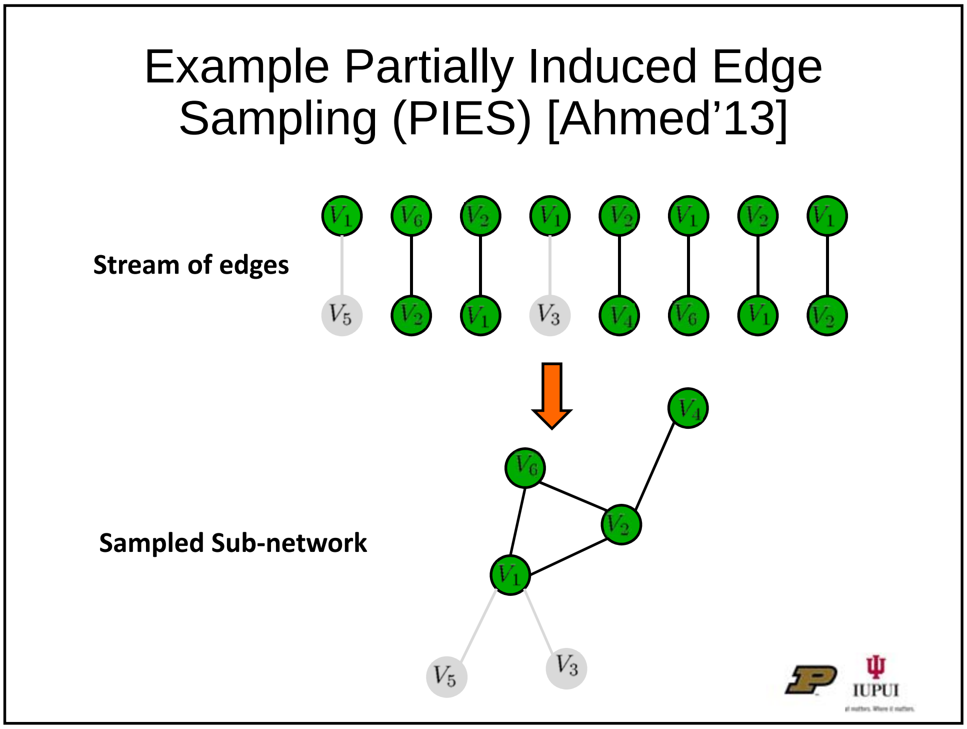
At the end of the sequential pass . . .

Stream of edges

reservoir

Node List

Edge List

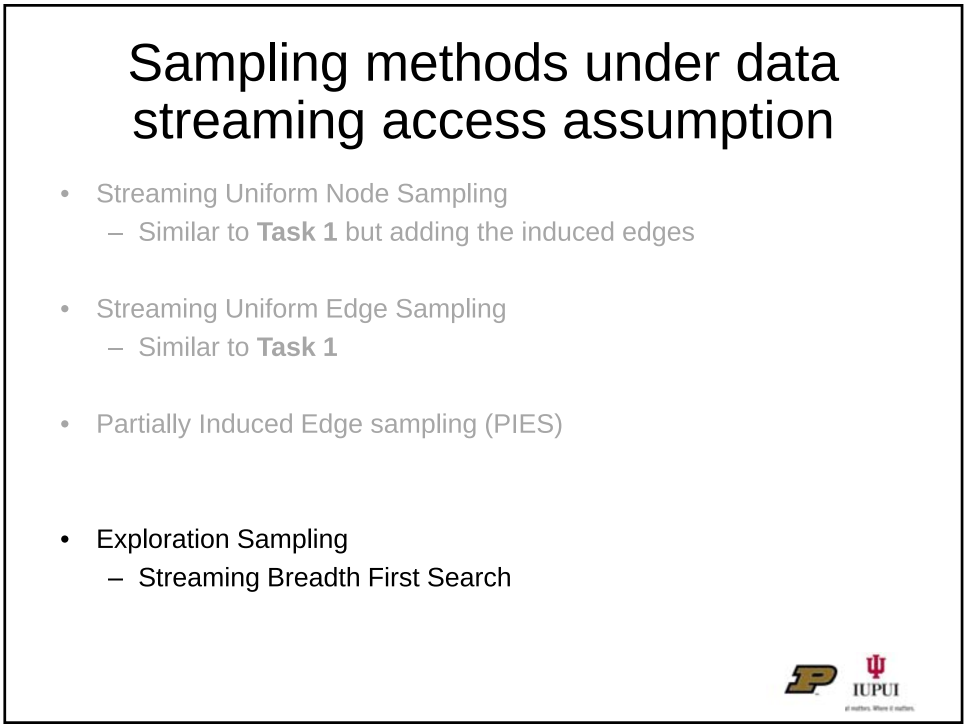


Example Partially Induced Edge Sampling (PIES) [Ahmed'13]

Stream of edges

Sampled Sub-network

V_1 V_6 V_2 V_1 V_2 V_1 V_2 V_1
V_5 V_2 V_1 V_3 V_4 V_6 V_1 V_2
V_4
V_6
V_2
V_1
V_5
V_3



Sampling methods under data streaming access assumption

- Streaming Uniform Node Sampling
  - Similar to **Task 1** but adding the induced edges
- Streaming Uniform Edge Sampling
  - Similar to **Task 1**
- Partially Induced Edge sampling (PIES)
- Exploration Sampling
  - Streaming Breadth First Search

Streaming Breadth First Search

- Streaming Breadth First Search in one pass
- Implements breadth first search in a window of consecutive edges
- Slide the window each time an edge is streaming in

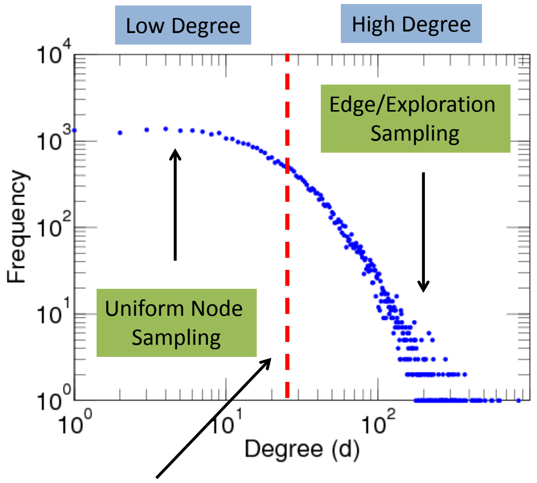
ANALYSIS & RESULTS

How different sampling methods sample from the degree distribution?

ArXiv Hep-PH

34K nodes

421K Edges



Edge/Exploration Sampling

Uniform Node Sampling

Ahmed'13

Avg. Degree = 24.4

More Analysis

- Expected value of degree k in a sampled sub-network using uniform node sampling

$$\begin{aligned} E[f_D(k)] &= f_D(k) \cdot n \cdot p \\ &= f_D(k) \cdot \frac{n}{N} \end{aligned}$$

- Expected value of degree k in a sampled sub-network using edge sampling with graph induction

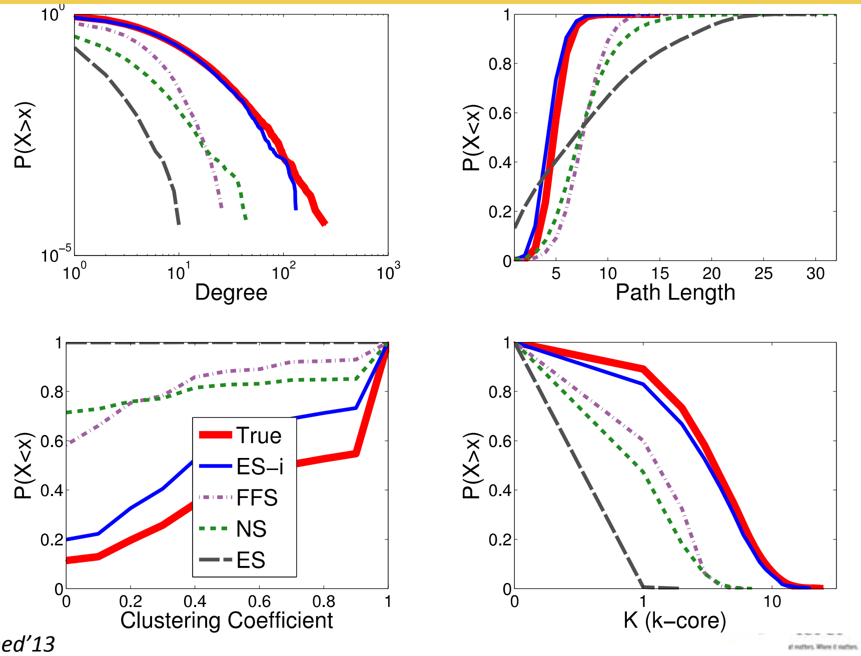
$$\begin{aligned} E'[f_D(k)] &= f_D(k) \cdot n \cdot p' \\ &= f_D(k) \cdot n \cdot \frac{k}{2 \cdot |E|} \end{aligned}$$

$$f_D(k) \cdot n \cdot \frac{k}{2 \cdot |E|} - f_D(k) \cdot \frac{n}{N} \geq 0$$

$$f_D(k) \cdot \frac{n}{N} \cdot \frac{k}{k_{avg}} - f_D(k) \cdot \frac{n}{N} \geq 0$$

$$k \geq k_{avg}$$

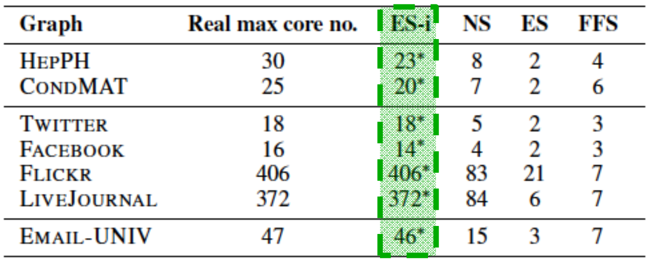
Edge sampling with graph induction (Es-i) performs better than other methods



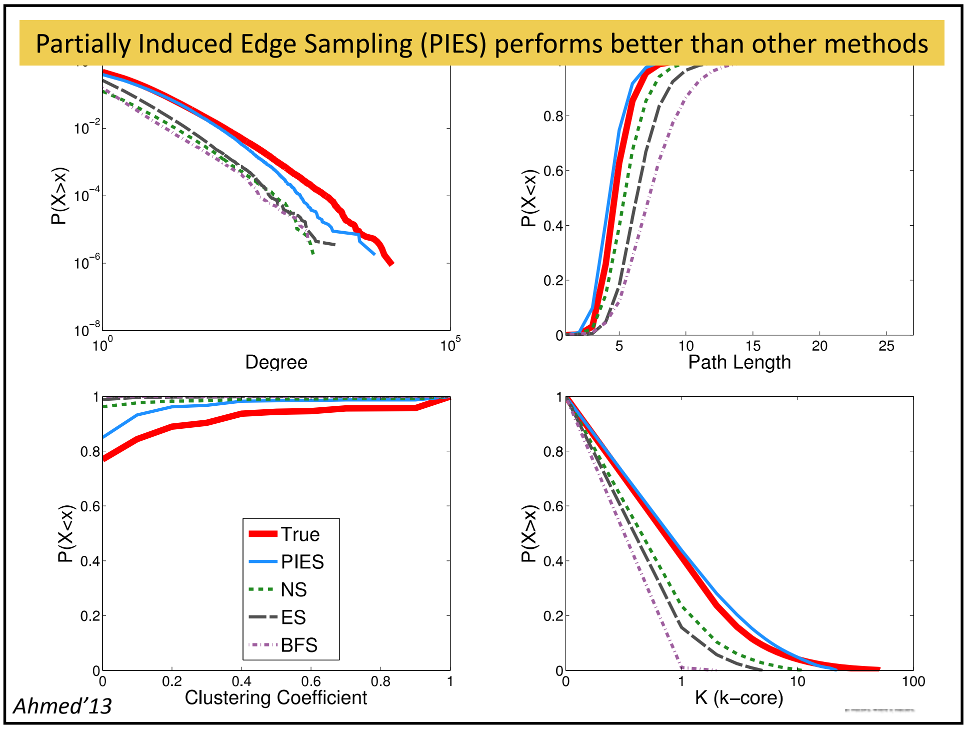
Ahmed'13

Preserving Max K-core number k_{max}

- Max K-core number k_{max}
  - Largest subgraph in the network with min degree k_{max}
- *Sample size = 20% /Nodes*



Graph	Real max core no.	ES-i	NS	ES	FFS
HepPH	30	23*	8	2	4
CondMAT	25	20*	7	2	6
Twitter	18	18*	5	2	3
Facebook	16	14*	4	2	3
Flickr	406	406*	83	21	7
LiveJournal	372	372*	84	6	7
Email-UNIV	47	46*	15	3	7



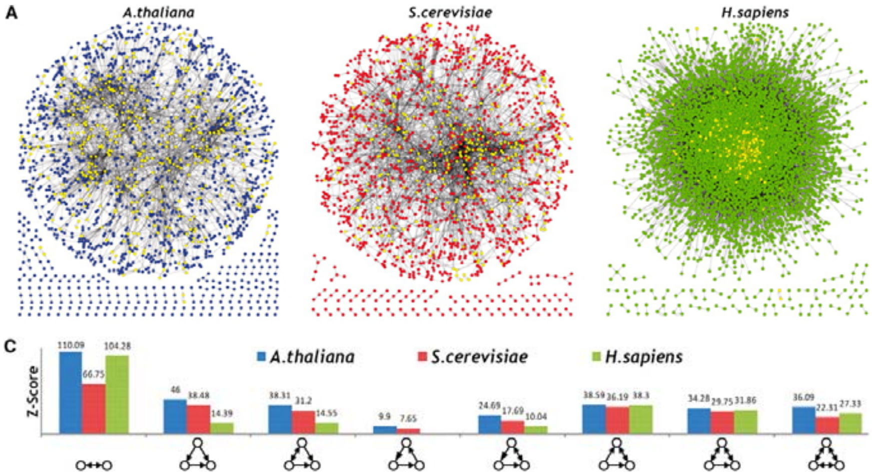
Partially Induced Edge Sampling (PIES) performs better than other methods

Ahmed'13

More conclusions …

- Sampling from the stream in one pass is a hard task
  - Exploration methods (Breadth-First) generally don't perform well
    - Solution: Perform more than one sequential pass over the stream
  - Edge-based methods generally perform well
    - **But sometimes** fail to preserve clustering coefficient
- Stream sampling perform better for sparse graphs

Differentiating species via network analysis of protein-protein interactions



http://bioinformatics.oxfordjournals.org/content/27/2/245/F1.expansion.html

Task 3 (Sampling sub-structures from networks)

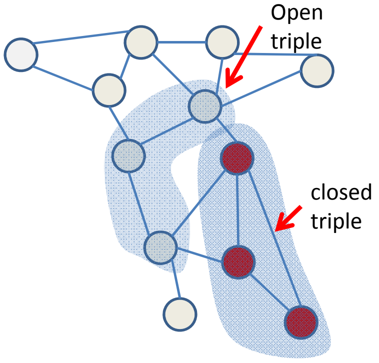
- In this case, our objective is to sample small substructure of interest
  - Sampling graphlets (Bhuiyan et al., 2012)
    - Is used for build graphlet degree distribution which characterize biological networks
  - Sampling triangles and triples (Rahman et al., 2013, Buriol et al., 2006)
    - Is used for estimating triangle count
  - Sampling network motifs (Kashtan et al., 2004, Wernicke 2006)
  - Sampling frequent patterns (Hasan et al. 2009)

Triple sampling from a network: Motivation

- A triple, (u, v, w) at a vertex , v is a path of length two for which v is the center vertex.
- A Triple can be closed (triangle), or open (wedge)

$$triple\text{-}cnt = \sum_{v \in V} \binom{d(v)}{2}$$

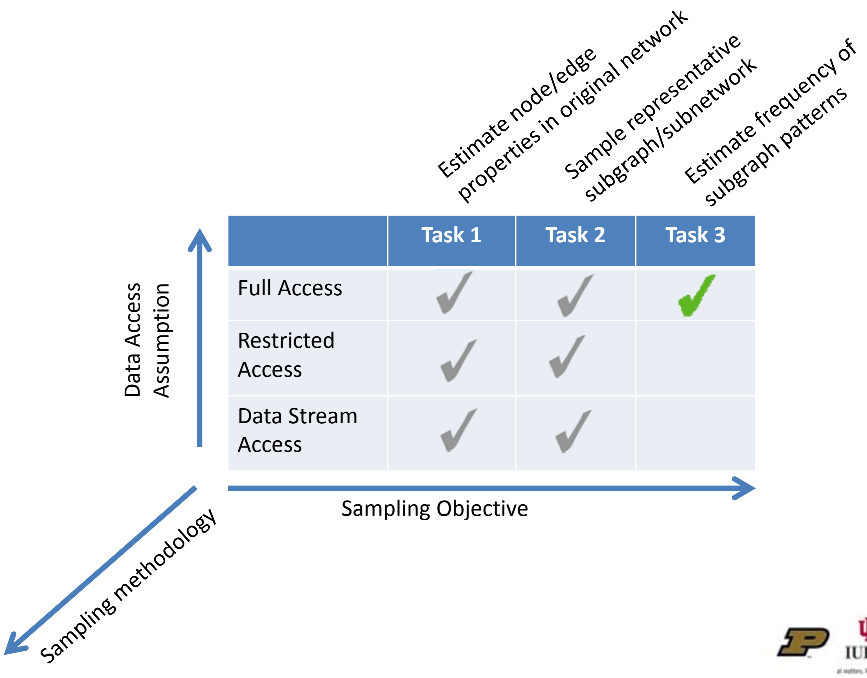
- It is useful for approximating triangle counting and transitivity



- T is a set of **uniformly** sampled triples, and $C \subseteq T$, is the set of closed triples
- Approximate transitivity is $\sigma = |C|/|T|$
- Approximate triangle count in a network is: $\frac{\sigma}{3} \cdot triple\text{-}count$

Induced k-Subgraph Sampling: motivation

- Network motifs are induced subgraphs that occur in the network far more often than in randomized networks with identical degree distribution
- To identify network motif, we first construct a set of random networks (typically few hundreds) with identical degree distributions.
- If the concentration of some subgraph in the given network is *significantly* higher than the concentration of that subgraph in the random networks, those subgraphs are called *network motif*
- Concentration of a size-k subgraph $S_i^{(k)}$ is defined as: $C_i^{(k)}(G) = \frac{f_i^{(k)}(G)}{\sum_{j:\,\text{size}(j)=k} f_j^{(k)}(G)}$
- Exact counting of k-subgraphs are costly, so sampling methods are used to approximate k-subgraph concentrations



Estimate node/edge properties in original network

Sample representative subgraph/subnetwork

Estimate frequency of subgraph patterns

Data Access Assumption

	Task 1	Task 2	Task 3
Full Access	✓	✓	✓
Restricted Access	✓	✓	
Data Stream Access	✓	✓	

Sampling Objective

Sampling methodology

Triple sampling with full access [Seshadhri ’13]

- Take a vertex v uniformly, choose two of its neighbors u , and w; return the triple (u,v,w)
- This **does not** yield uniform sampling, as the number of triples centered to a vertex is non-uniform,

$$\mathrm{P}\{\text{triple } (u,v,w) \text{ is sampled}\} = \frac{1}{n\cdot\binom{d(v)}{2}}$$

- For triangle counting from a set of sampled triples (say, T) , we can apply the same un-biasing trick that we applied for nodal characteristics estimation,

$$TC = \frac{\textit{triple-cnt}}{3\cdot W}\times\sum_{t\in T} w_t\cdot 1_{\{t \text{ is closed}\}}$$

here w_t is the probability of selecting a triple, and $W=\sum_{t\in T} w_t$

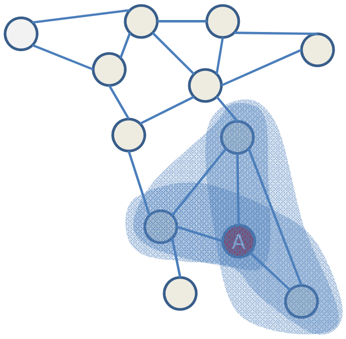
Uniform Triple sampling with full access

- Method
  - Select a node *v* in proportion to the number of triples that are centered at *v*, which is equal to

$$\frac{d(u)(d(u)-1)}{2}$$

  - Return one of the triple centered at *v* uniformly
  - For the first step, we need to know the degree of all the vertices
- Triangle count estimate is easy, we just need to use the formula

$$TC = \frac{triple\text{-}cnt}{3\cdot|T|}\times\sum_{t\in T}1_{\{t\text{ is closed}\}}$$



Node A has degree 3, it has 3 triples centered around it

Sampling of subgraphs of size k [Kashtan '04

- Method
  1. Select an edge uniformly
  2. Populate Neighborhood
  3. Choose one neighbor uniformly
  4. If size ≠ *k* , goto 2 else return the pattern

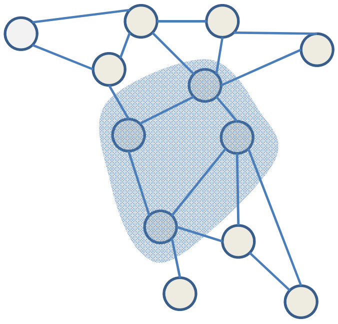
Probability of generating this pattern (in this edge order) :

$$= \frac{1}{m}\times$$

$$= \frac{1}{m}\times\frac{1}{7}$$

$$= \frac{1}{m}\times\frac{1}{7}\times\frac{1}{7}$$

$$= \frac{1}{49m}$$

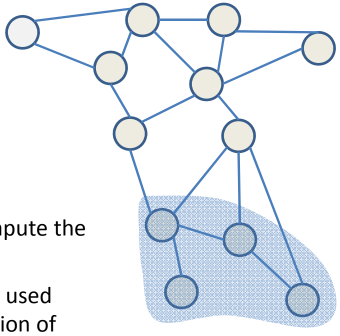


k=4

Sampling of subgraphs of size k Kash

Probability of generating this pattern (in this edge order) :

$$= \frac{1}{m} \times$$
$$= \frac{1}{m} \times \frac{1}{3}$$
$$= \frac{1}{m} \times \frac{1}{3} \times \frac{1}{3}$$
$$= \frac{1}{9\,m}$$

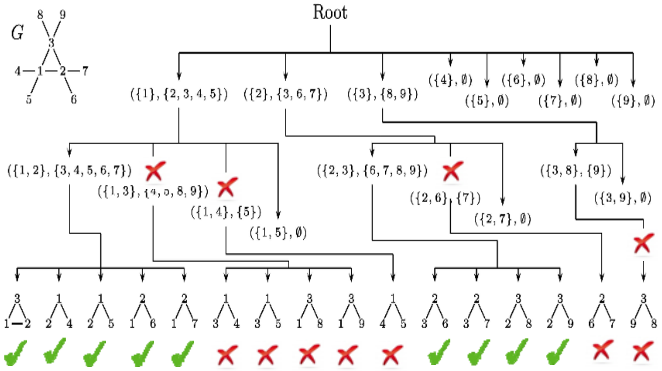


- Sampling probability is not uniform, but we can compute the probability
- We can apply the same unbiasing trick that we have used before to obtain uniform estimate of the concentration of some subgraph of size k
- Assume, we take B samples of size-k subgraph, Then the concentration of some subgraph type s_i is:,

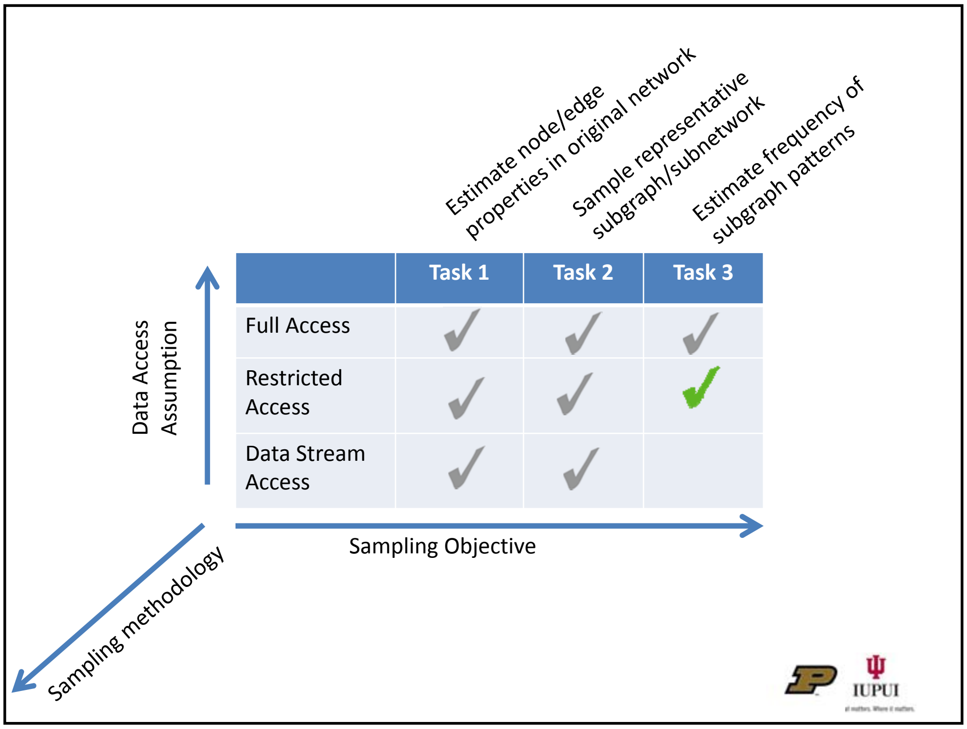
k=4

$$\frac{\sum_{j=1}^{B} w_j \cdot 1_{\{j\text{-}type=s_i\}}}{W} \quad \text{where} \quad w_j = \frac{1}{P_j}, \text{ and } W = \sum_{j=1}^{B} w_j$$

Uniform Sampling of k-subgraph by probabilistic enumeration [Wernicke '06]



- Complete enumeration generates each *k*-subgraph exactly once.
- For sampling, for every child of a node at level-*i*, enumerate the child with probability p_i.
- Each pattern is generated with probability $\prod_{i=1}^{k} p_i$ (uniform)



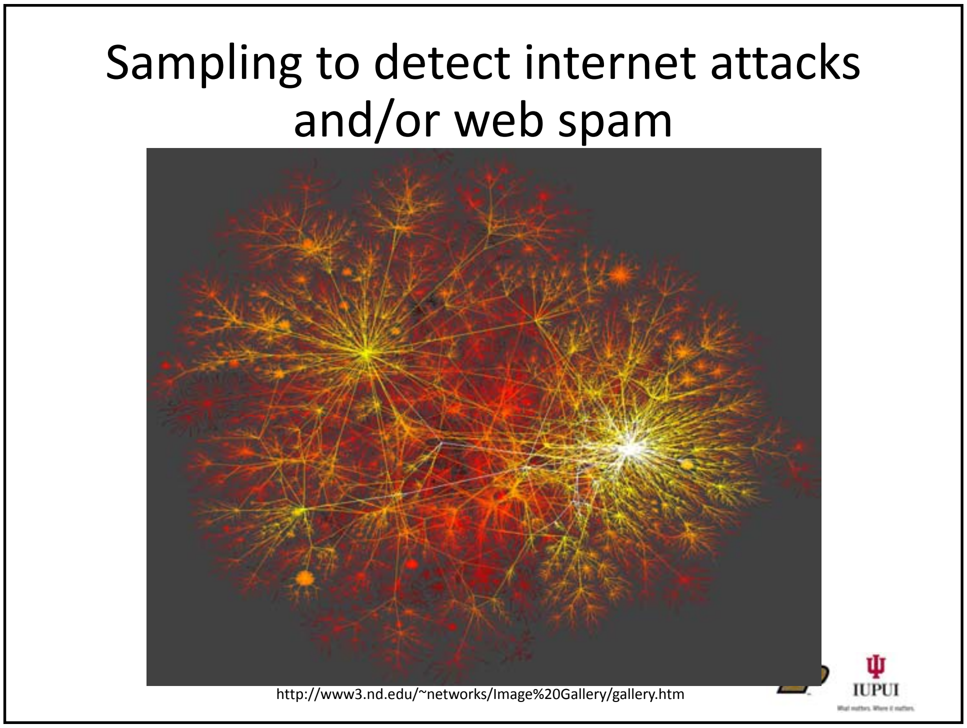
Estimate node/edge properties in original network

Sample representative subgraph/subnetwork

Estimate frequency of subgraph patterns

	Task 1	Task 2	Task 3
Full Access	✓	✓	✓
Restricted Access	✓	✓	✓
Data Stream Access	✓	✓	

Data Access Assumption
Sampling Objective
Sampling methodology



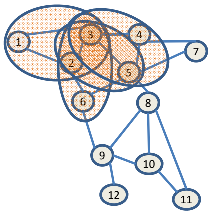
Sampling to detect internet attacks and/or web spam

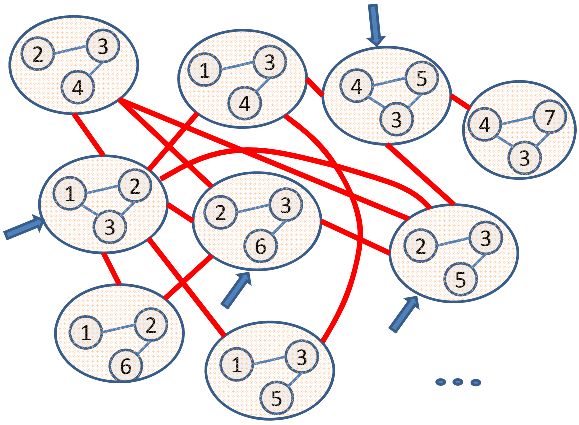
http://www3.nd.edu/~networks/Image%20Gallery/gallery.htm

Triple sampling by exploration

- Perform a random walk over the triple space
  - Given a seed node, form a triple from its neighborhood information
  - Define a neighborhood in the triple space that can be instantiated online using local information
  - Choose the next triple from the neighborhood
- The stationary distribution of the walk is proportional to the number of neighbors of a triple in the sampling space

Neighborhood





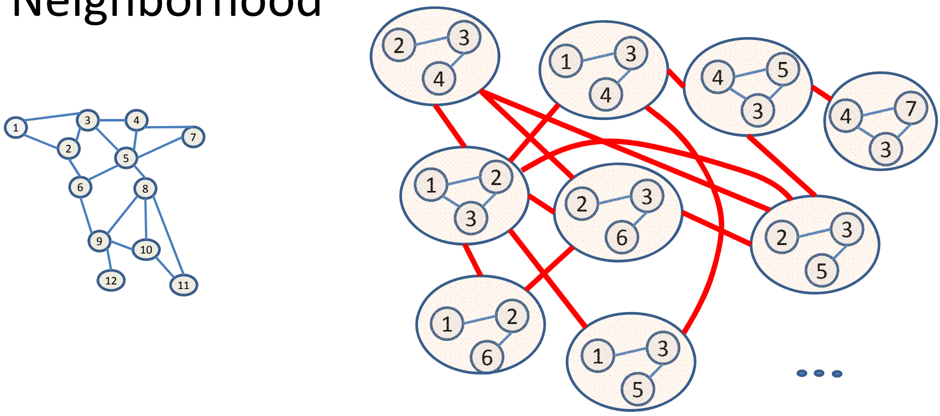
Two neighboring triples share two vertices among them, entire neighborhood graph is not shown

Triple sampling by exploration based solution for restricted access

- If we want to obtain unbiased estimate of the concentration of triangle (transitivity), we can use the same un-biasing strategy as before:
  - The concentration of triangle is: ,

$$\frac{\sum_{j=1}^{B} w_j \cdot 1_{\{j \cdot type = closed\}}}{W}, \text{ here, W=} \sum_{j=1}^{B} w_j \text{ and } w_j = 1/d(j)$$

Neighborhood



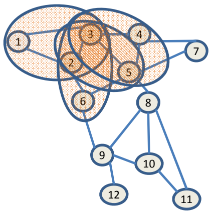
Degree of the triple (1, 2, 3) is 6 as can be seen in the above graph, so while unbiasing,

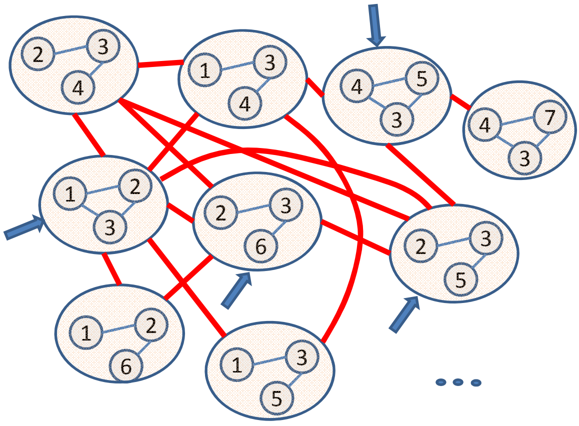
$$w_{(1,2,3)} = \frac{1}{6}$$

Another exploration-based triple sampling that uses MH algorithm

- MH algorithm uses simple random walk as proposal distribution
- Then is uses the MH correction (accept/reject) so that the target distribution is uniform over all the triples
- Like before, if triple T_i is currently visiting state, and the triple T_j is chosen by the proposal move, then this move is accepted with probability $\min\left\{1, \frac{d(i)}{d(j)}\right\}$
- MH algorithm thus confirm uniform sampling, so the transitivity can be easily computed as $\frac{1}{|T|}\sum_{k=1}^{|T|} 1_{\{T_k \text{ is closed}\}}$

Neighborhood

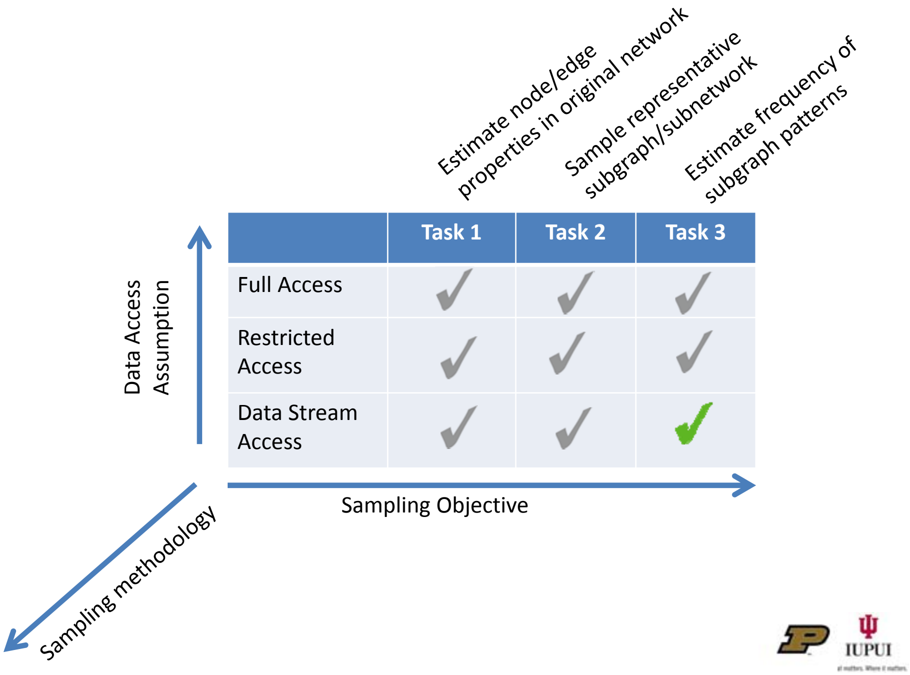




- For uniform sampling, Every move from triple T_i to triple T_j is accepted with probability $d(T_i)/d(T_j)$
- If $T_i = (1, 2, 3)$ and $T_j = (1,3,4)$, move from T_i to T_j is accepted with probability max$\left\{1, \frac{6}{4}\right\} = 1$, but a move from T_j to T_i is accepted with probability, max$\left\{1, \frac{4}{6}\right\} = 4/6$

Sampling subgraphs of size k using exploration

- Both random walk and M-H algorithm can also be used for sampling subgraph of size k
- As we did in triangle sampling, we simply perform a random walk on the space of all subgraphs of size-k
- We also define an appropriate neighborhood
  - For example, we can assume that two k-subgraphs are neighbors, if they share $k-1$ vertices
- This idea has recently been used in GUISE algorithm, which uniformly sampled all subgraphs of size 3, 4, and 5. It used M-H methods
  - Motivation of GUISE was to build a fingerprint for graphs that can be used for characterizing networks from different domains.

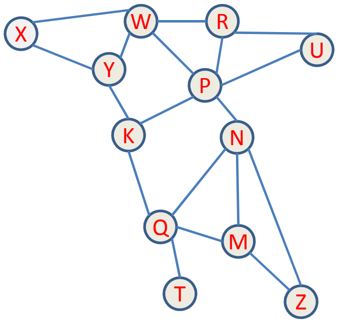

Estimate node/edge properties in original network
Sample representative subgraph/subnetwork
Estimate frequency of subgraph patterns
Data Access Assumption
Sampling Objective
Sampling methodology

	Task 1	Task 2	Task 3
Full Access	✓	✓	✓
Restricted Access	✓	✓	✓
Data Stream Access	✓	✓	✓

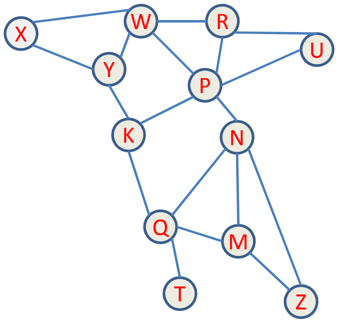
Triple Sampling from edge stream

- Sampling subgraphs using streaming method is not easy, existing works only covered subgraphs with three and four vertices
- Large number of works exist that aim to sample triples for approximating triangle counting from stream data
- For graph steam, there are two scenarios:
  - Arbitrary edge stream: Stream of the edges appear in arbitrary order
  - Incident edge stream: Stream of the edges incident to some vertices come together, whereas the ordering of vertices is arbitrary
- There is also another stream scenario for dynamic graph, which is known as turns as “turnstile model”
  - Each edge appears as a pair where the second term is either + (addition) or – (deletion) like below:

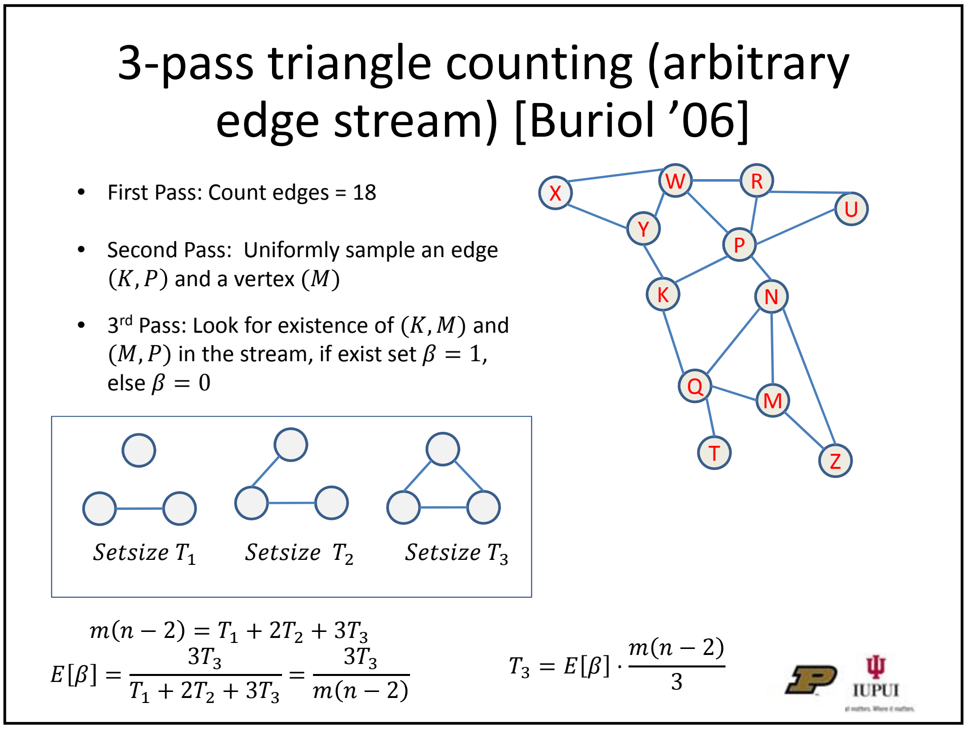
$$\{(e_1,+),(e_2,+),(e_3,-),(e_2,-),\cdots\}$$



Arbitrary edge stream



Incident edge stream



3-pass triangle counting (arbitrary edge stream) [Buriol '06]

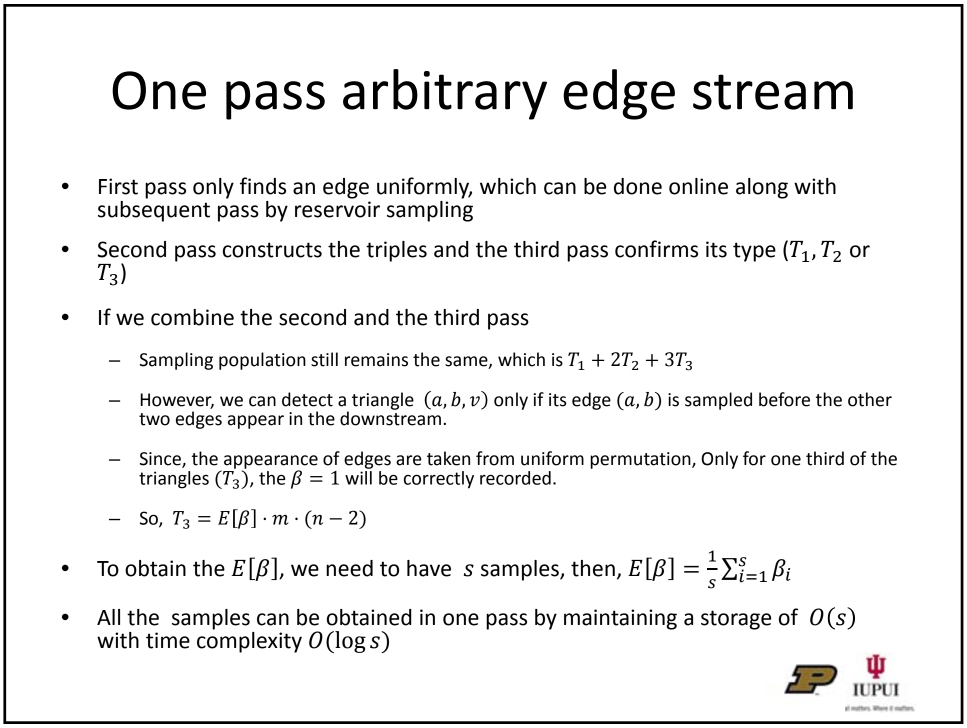
- First Pass: Count edges = 18
- Second Pass: Uniformly sample an edge (K, P) and a vertex (M)
- 3rd Pass: Look for existence of (K, M) and (M, P) in the stream, if exist set $\beta = 1$, else $\beta = 0$

$Setsize\ T_1$ $\quad$ $Setsize\ T_2$ $\quad$ $Setsize\ T_3$

$$m(n-2) = T_1 + 2T_2 + 3T_3$$

$$E[\beta] = \frac{3T_3}{T_1 + 2T_2 + 3T_3} = \frac{3T_3}{m(n-2)}$$

$$T_3 = E[\beta] \cdot \frac{m(n-2)}{3}$$

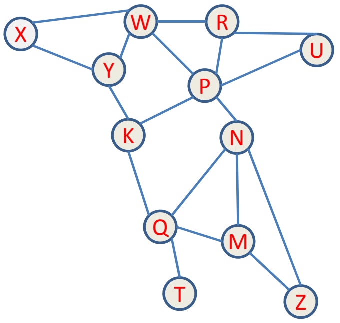


One pass arbitrary edge stream

- First pass only finds an edge uniformly, which can be done online along with subsequent pass by reservoir sampling
- Second pass constructs the triples and the third pass confirms its type (T_1, T_2 or T_3)
- If we combine the second and the third pass
  - Sampling population still remains the same, which is $T_1 + 2T_2 + 3T_3$
  - However, we can detect a triangle (a, b, v) only if its edge (a, b) is sampled before the other two edges appear in the downstream.
  - Since, the appearance of edges are taken from uniform permutation, Only for one third of the triangles (T_3), the $\beta = 1$ will be correctly recorded.
  - So, $T_3 = E[\beta] \cdot m \cdot (n-2)$
- To obtain the $E[\beta]$, we need to have s samples, then, $E[\beta] = \frac{1}{s}\sum_{i=1}^{s} \beta_i$
- All the samples can be obtained in one pass by maintaining a storage of $O(s)$ with time complexity $O(\log s)$

3-pass triangle counting (incident edge stream) [Buriol ’06]

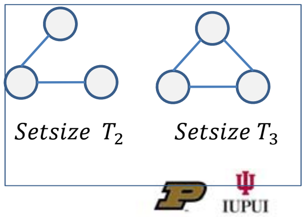
- First Pass: Count number of triples, $p = \Sigma_v \binom{d(v)}{2}$
- Second Pass: Uniformly sample a triple (K, P, N) centered at P
- 3rd Pass: Look for existence of (K, N) edge in the stream, if exist set $\beta = 1$, else $\beta = 0$



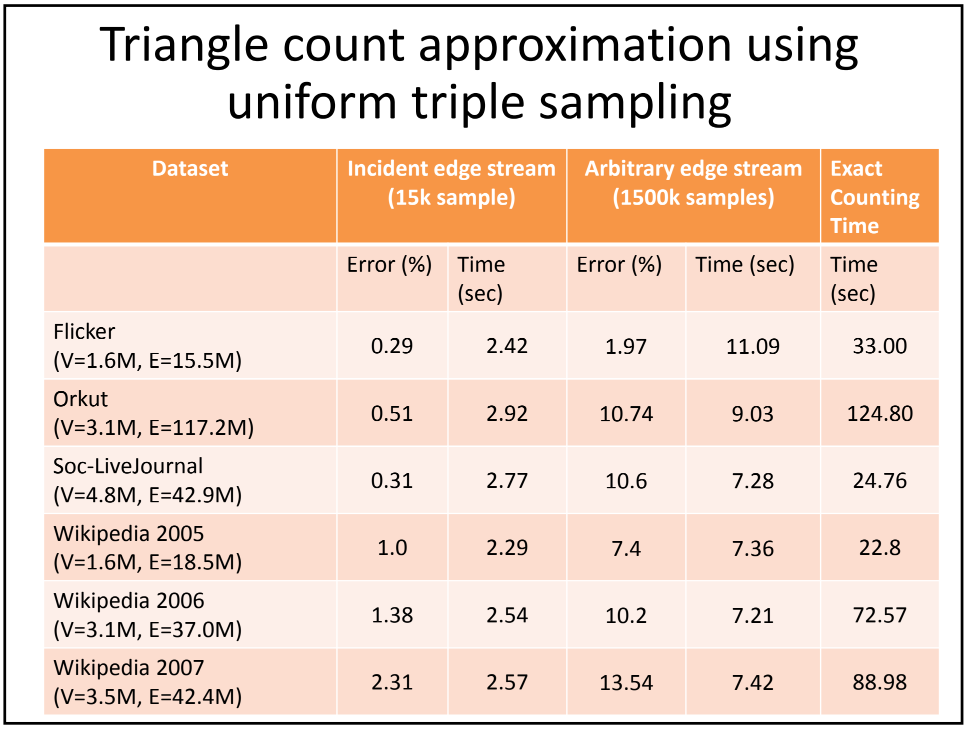
$$p = \sum_v \binom{d(v)}{2} = T_2 + 3T_3$$

$$E[\beta] = \frac{3T_3}{T_2 + 3T_3} = \frac{3T_3}{p}$$

$$T_3 = E[\beta] \times \frac{p}{3}$$

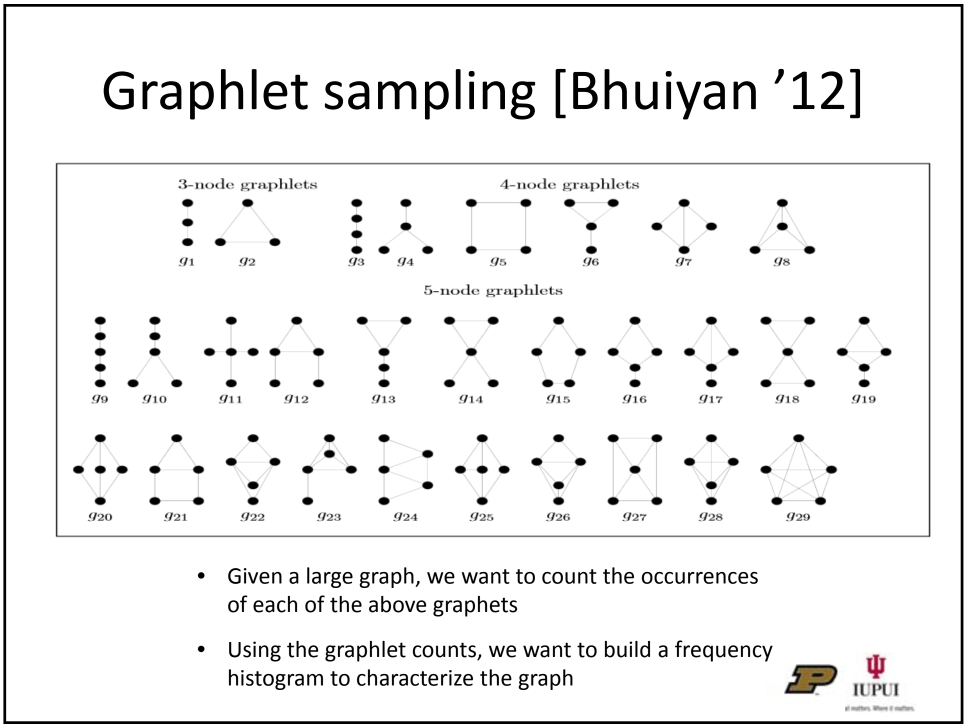


RESULTS



Triangle count approximation using uniform triple sampling

Dataset	Incident edge stream (15k sample)		Arbitrary edge stream (1500k samples)		Exact Counting Time
	Error (%)	Time (sec)	Error (%)	Time (sec)	Time (sec)
Flicker (V=1.6M, E=15.5M)	0.29	2.42	1.97	11.09	33.00
Orkut (V=3.1M, E=117.2M)	0.51	2.92	10.74	9.03	124.80
Soc-LiveJournal (V=4.8M, E=42.9M)	0.31	2.77	10.6	7.28	24.76
Wikipedia 2005 (V=1.6M, E=18.5M)	1.0	2.29	7.4	7.36	22.8
Wikipedia 2006 (V=3.1M, E=37.0M)	1.38	2.54	10.2	7.21	72.57
Wikipedia 2007 (V=3.5M, E=42.4M)	2.31	2.57	13.54	7.42	88.98

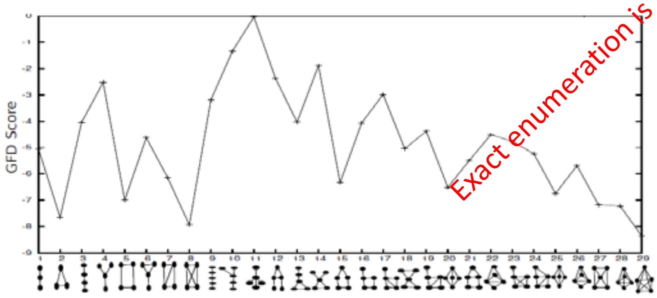


Graphlet sampling [Bhuiyan ’12]

- Given a large graph, we want to count the occurrences of each of the above graphets
- Using the graphlet counts, we want to build a frequency histogram to characterize the graph

Graphlet Frequency Distribution (GFD)

- Graphlet Frequency Distribution (GFD) is a histogram using the concentration of various graphlets in the graph G.
- is the frequency of each graphlet. Then the concentration of a graphlet is

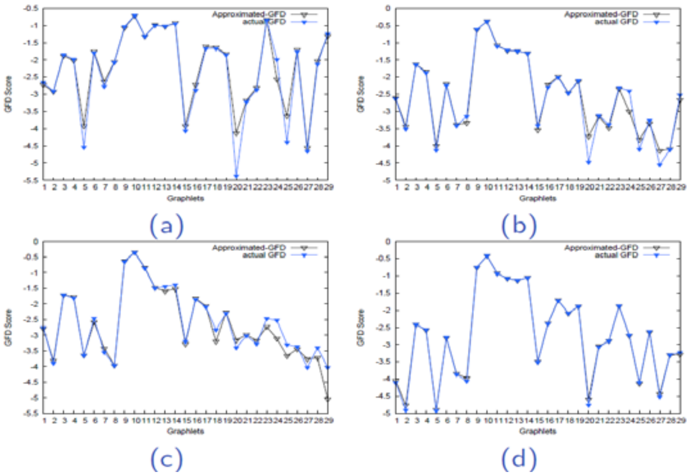


Exact enumeration is too costly!

Sampling graphlets

- GFD only needs relative frequencies of each of the graphlets
- Using uniform sampling of graphlets, we can approximate GFD
- A method similar to triple sampling will work
  - The sampling space is all the graphlets
  - The random walk moves from one graphlet to another graphlet
  - It accepts of reject a move based on the degree of a graphlet state.

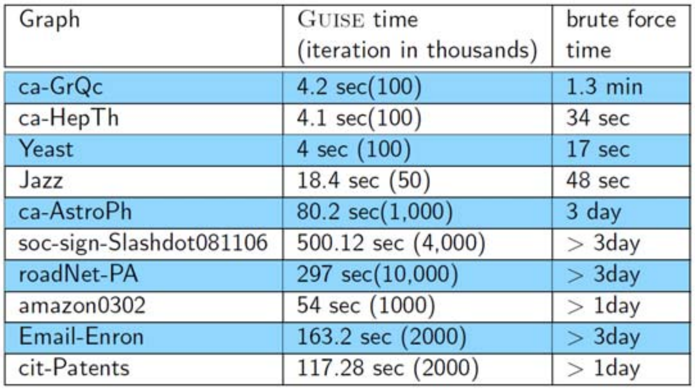
Exact Vs App GFD



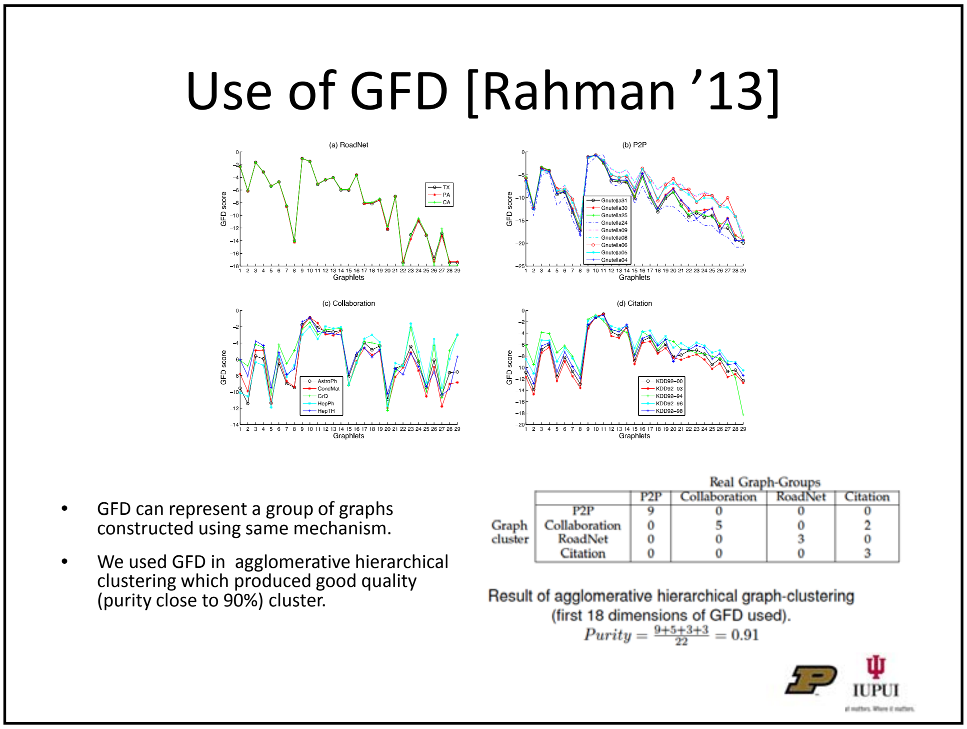
(a) (b) (c) (d)

Comparison with Actual and Approximate GFD for
(a) ca-GrQc (b) ca-Hepth (c) yeast (d) Jazz

Exact counting vs Sampling to constricut GFD



Graph	GUISE time (iteration in thousands)	brute force time
ca-GrQc	4.2 sec(100)	1.3 min
ca-HepTh	4.1 sec(100)	34 sec
Yeast	4 sec (100)	17 sec
Jazz	18.4 sec (50)	48 sec
ca-AstroPh	80.2 sec(1,000)	3 day
soc-sign-Slashdot081106	500.12 sec (4,000)	> 3day
roadNet-PA	297 sec(10,000)	> 3day
amazon0302	54 sec (1000)	> 1day
Email-Enron	163.2 sec (2000)	> 3day
cit-Patents	117.28 sec (2000)	> 1day



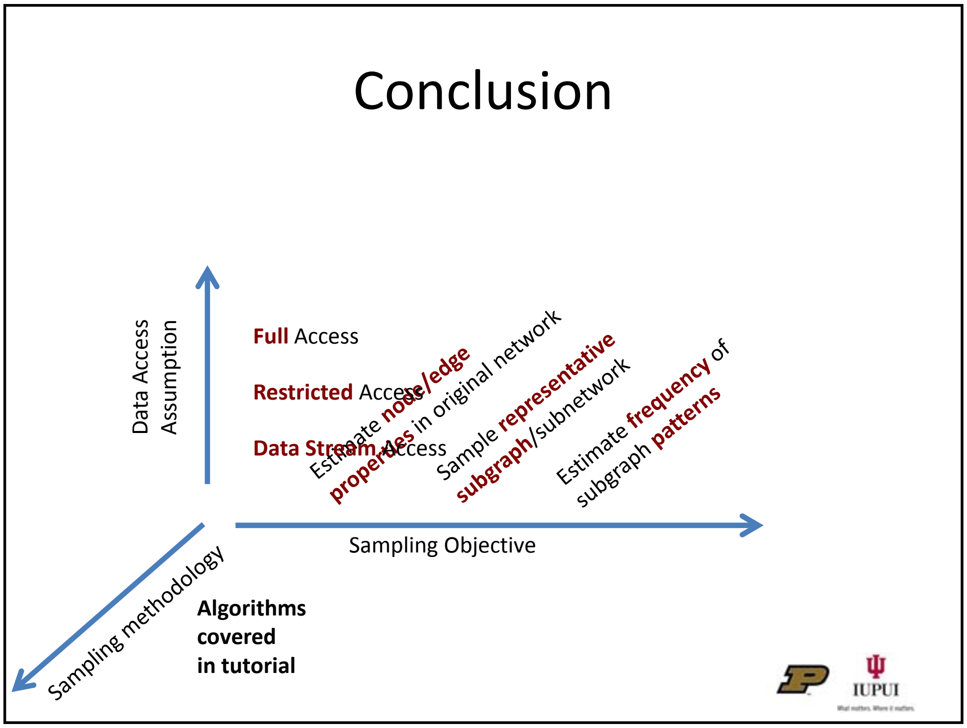
Use of GFD [Rahman ’13]

- GFD can represent a group of graphs constructed using same mechanism.
- We used GFD in agglomerative hierarchical clustering which produced good quality (purity close to 90%) cluster.

		Real Graph-Groups			
		P2P	Collaboration	RoadNet	Citation
Graph cluster	P2P	9	0	0	0
	Collaboration	0	5	0	2
	RoadNet	0	0	3	0
	Citation	0	0	0	3

Result of agglomerative hierarchical graph-clustering (first 18 dimensions of GFD used).

$$Purity = \frac{9+5+3+3}{22} = 0.91$$



Conclusion

Data Access Assumption

Full Access

Restricted Access

Data Stream Access

Estimate **node/edge properties** in original network

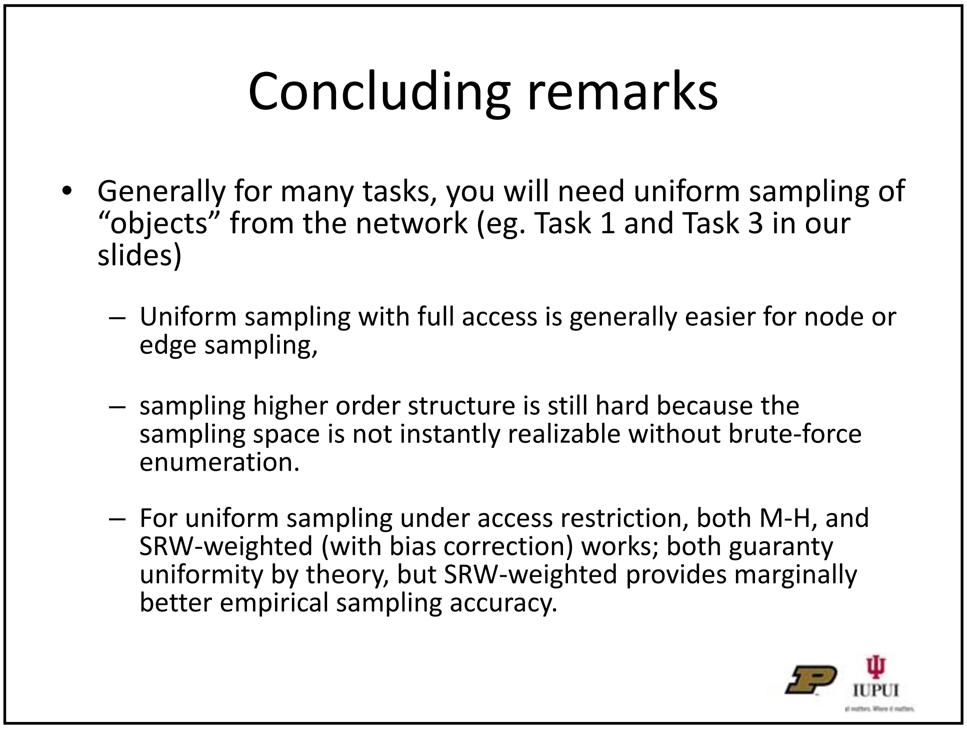
Sample **representative subgraph**/subnetwork

Estimate **frequency** of subgraph **patterns**

Sampling Objective

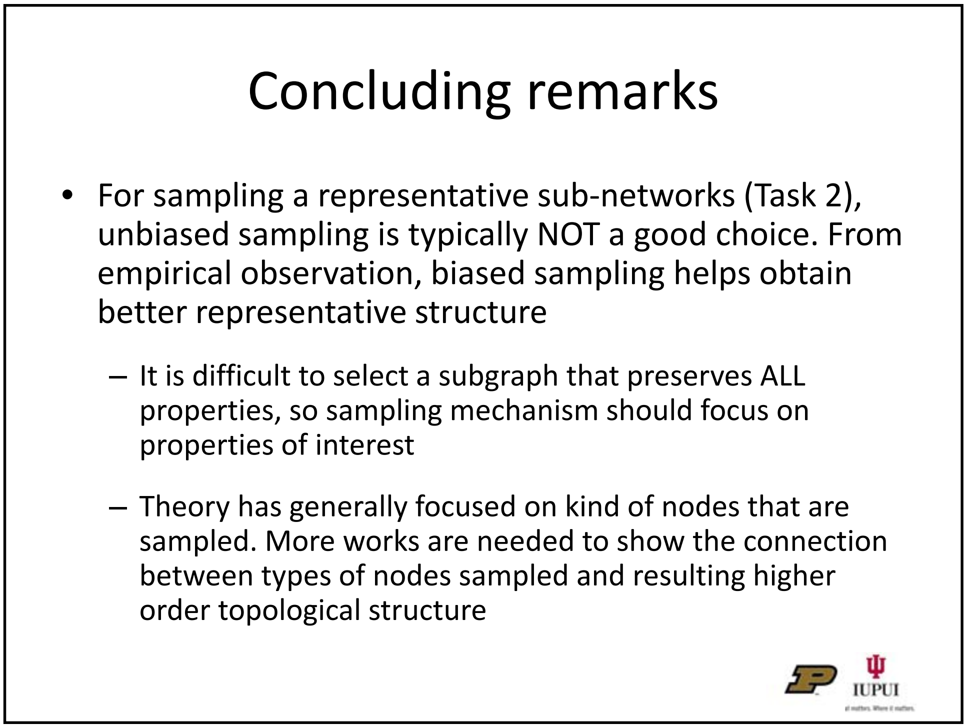
Sampling methodology

Algorithms covered in tutorial



Concluding remarks

- Generally for many tasks, you will need uniform sampling of “objects” from the network (eg. Task 1 and Task 3 in our slides)
  - Uniform sampling with full access is generally easier for node or edge sampling,
  - sampling higher order structure is still hard because the sampling space is not instantly realizable without brute-force enumeration.
  - For uniform sampling under access restriction, both M-H, and SRW-weighted (with bias correction) works; both guaranty uniformity by theory, but SRW-weighted provides marginally better empirical sampling accuracy.

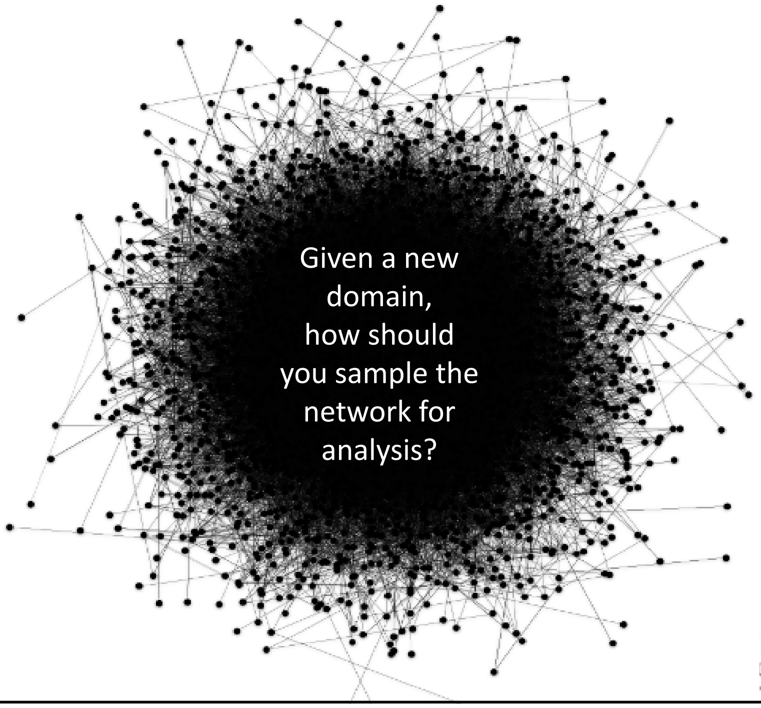


Concluding remarks

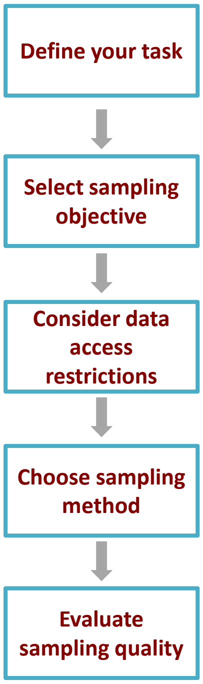
- For sampling a representative sub-networks (Task 2), unbiased sampling is typically NOT a good choice. From empirical observation, biased sampling helps obtain better representative structure
  - It is difficult to select a subgraph that preserves ALL properties, so sampling mechanism should focus on properties of interest
  - Theory has generally focused on kind of nodes that are sampled. More works are needed to show the connection between types of nodes sampled and resulting higher order topological structure

Concluding remarks (cont.)

- In present days, streaming is becoming more natural data access mechanism for graphs. So, currently it is an active area of research
  - It presents unique challenges to sample topology in an unbiased way, because local views of stream do not necessarily reflect connectivity.
  - Existing works have solved some isolated problems exactly (eg. triangle sampling, pagerank computation), and some other empirically (eg. representative sub-network)



Given a new domain, how should you sample the network for analysis?



Define your task

Select sampling objective

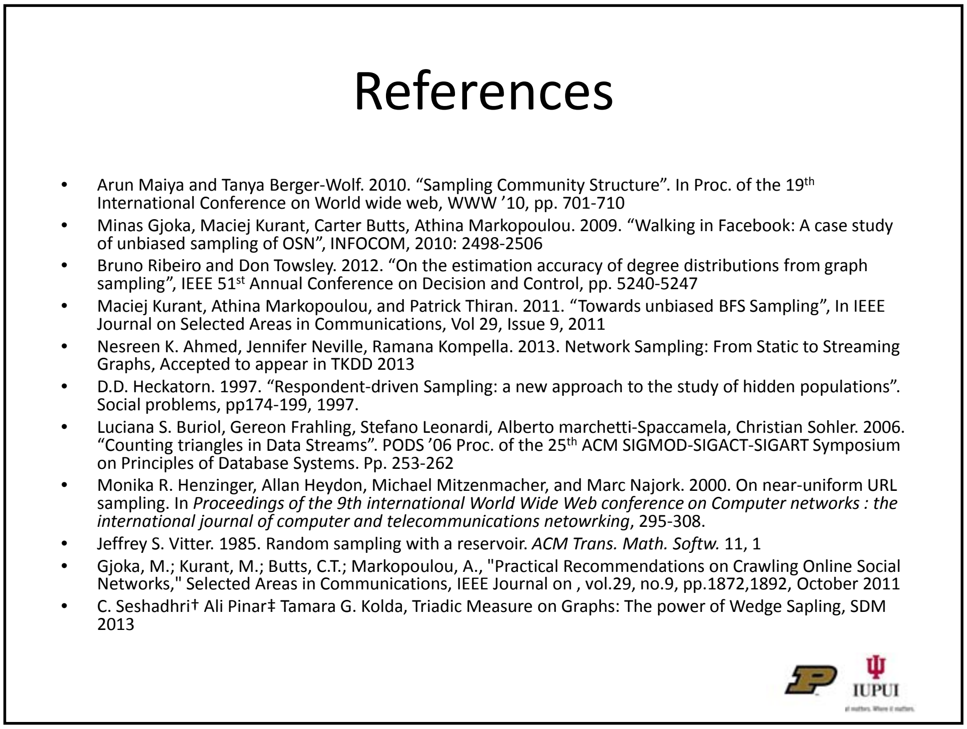
Consider data access restrictions

Choose sampling method

Evaluate sampling quality

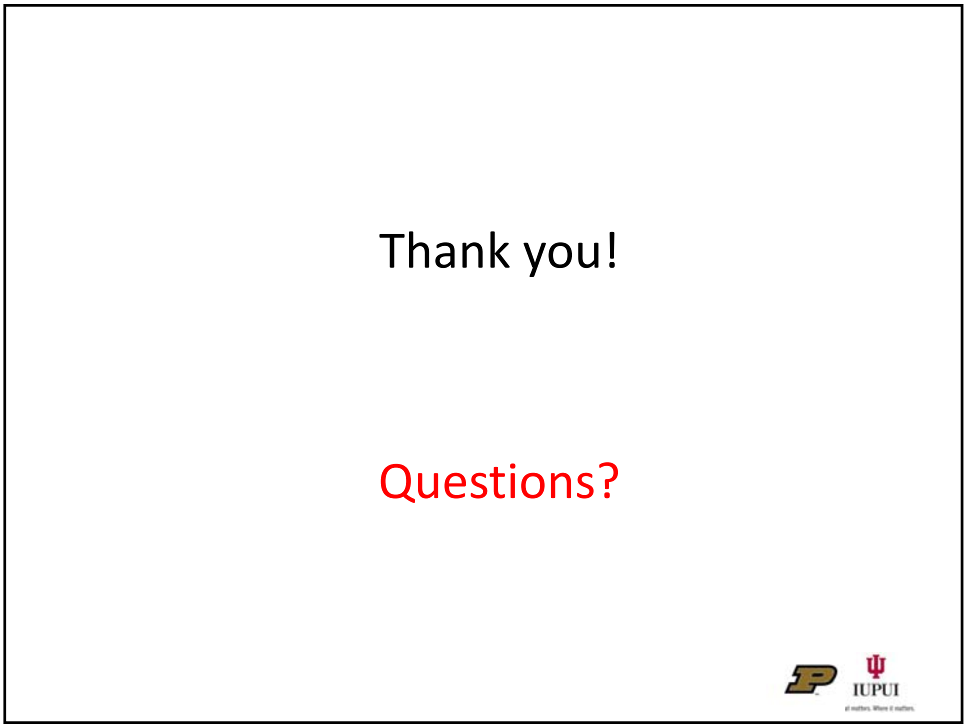
References

- Jure Leskovec and Christos Faloutsos. 2006. “Sampling from Large Graphs”. In proc. of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Miing, KDD ’06, pp. 631-636
- Bhuiyan, M.A.; Rahman, M.; Rahman, M.; Al Hasan, M., "GUISE: Uniform Sampling of Graphlets for Large Graph Analysis," *Data Mining (ICDM), 2012 IEEE 12th International Conference on* , pp.91-100
- N. Kashtan, S. Itzkovitz, R. Milo, and U. Alon. 2004. “Efficient sampling algorithm for estimating subgraph concentrations and detecting network motifs”. *Bioinformatics* 20, 11 (July 2004), 1746-1758.
- Sebastian Wernicke and Florian Rasche. 2006. “FANMOD: a tool for fast network motif detection”. *Bioinformatics* 22, 9 (May 2006), 1152-1153.
- Mohammad Al Hasan and Mohammed J. Zaki. 2009. “Output space sampling for graph patterns”. *Proc. VLDB Endow.* 2, 1 (August 2009), 730-741.
- Christian Hubler, hans-Peter Kriegel, Karsten Borgwardt, and Zoubin Ghahramani. 2008. “Metropolis algorithms for representative subgraph mining”. *Data Mining (ICDM), 2008 IEEE 8th International Conference on* , pp.283-292



References

- Arun Maiya and Tanya Berger-Wolf. 2010. “Sampling Community Structure”. In Proc. of the 19th International Conference on World wide web, WWW ’10, pp. 701-710
- Minas Gjoka, Maciej Kurant, Carter Butts, Athina Markopoulou. 2009. “Walking in Facebook: A case study of unbiased sampling of OSN”, INFOCOM, 2010: 2498-2506
- Bruno Ribeiro and Don Towsley. 2012. “On the estimation accuracy of degree distributions from graph sampling”, IEEE 51st Annual Conference on Decision and Control, pp. 5240-5247
- Maciej Kurant, Athina Markopoulou, and Patrick Thiran. 2011. “Towards unbiased BFS Sampling”, In IEEE Journal on Selected Areas in Communications, Vol 29, Issue 9, 2011
- Nesreen K. Ahmed, Jennifer Neville, Ramana Kompella. 2013. Network Sampling: From Static to Streaming Graphs, Accepted to appear in TKDD 2013
- D.D. Heckatorn. 1997. “Respondent-driven Sampling: a new approach to the study of hidden populations”. Social problems, pp174-199, 1997.
- Luciana S. Buriol, Gereon Frahling, Stefano Leonardi, Alberto marchetti-Spaccamela, Christian Sohler. 2006. “Counting triangles in Data Streams”. PODS ’06 Proc. of the 25th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems. Pp. 253-262
- Monika R. Henzinger, Allan Heydon, Michael Mitzenmacher, and Marc Najork. 2000. On near-uniform URL sampling. In *Proceedings of the 9th international World Wide Web conference on Computer networks : the international journal of computer and telecommunications netowrking*, 295-308.
- Jeffrey S. Vitter. 1985. Random sampling with a reservoir. *ACM Trans. Math. Softw.* 11, 1
- Gjoka, M.; Kurant, M.; Butts, C.T.; Markopoulou, A., "Practical Recommendations on Crawling Online Social Networks," Selected Areas in Communications, IEEE Journal on , vol.29, no.9, pp.1872,1892, October 2011
- C. Seshadhri† Ali Pinar‡ Tamara G. Kolda, Triadic Measure on Graphs: The power of Wedge Sapling, SDM 2013



Thank you!

Questions?