# Sampling Large Graphs: Algorithms and Applications

Don Towsley

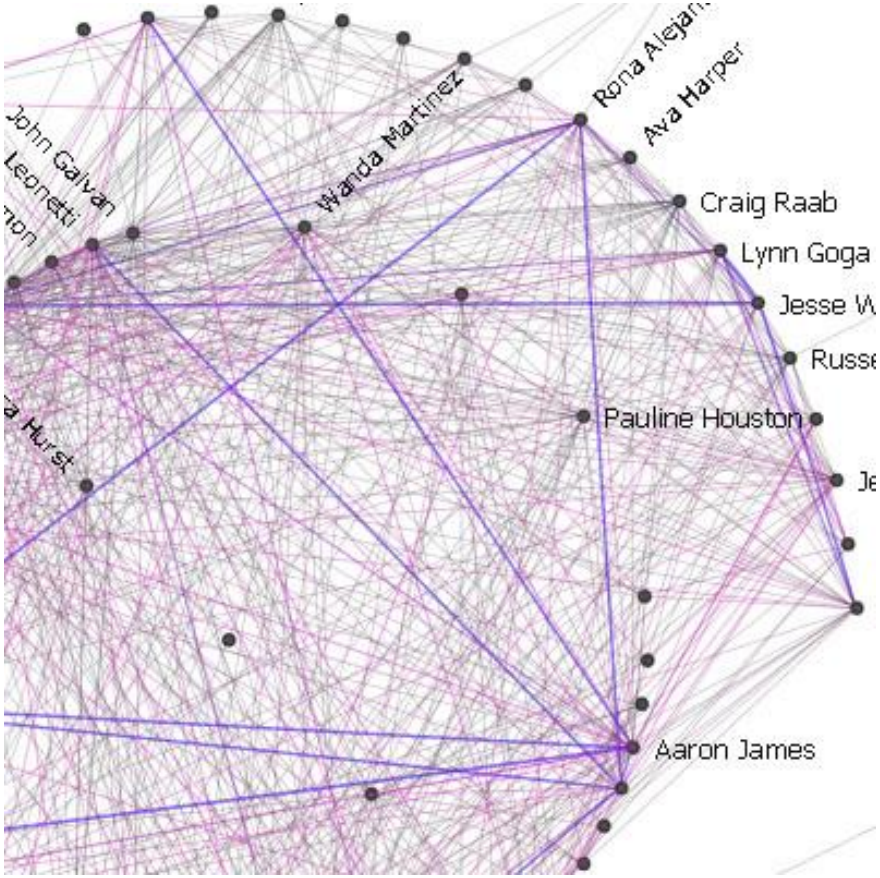
College of Information & Computer Science

Umass - Amherst

Collaborators: P.H. Wang, J.C.S. Lui,
J.Z. Zhou, X. Guan

# Measuring, analyzing large networks

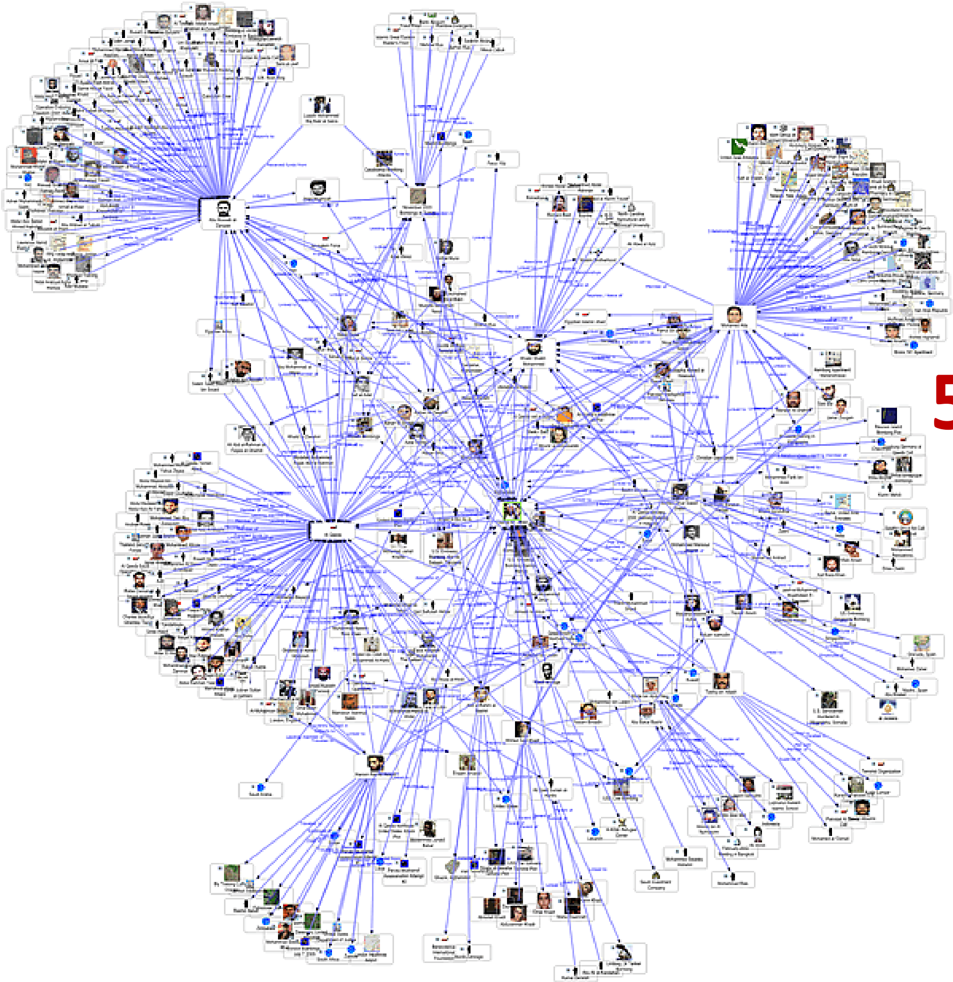
- large networks can be represented by graphs
- Facebook




**1+ Billion**

# Measuring, analyzing large networks

- large networks can be represented by graphs
- Facebook
- WWW

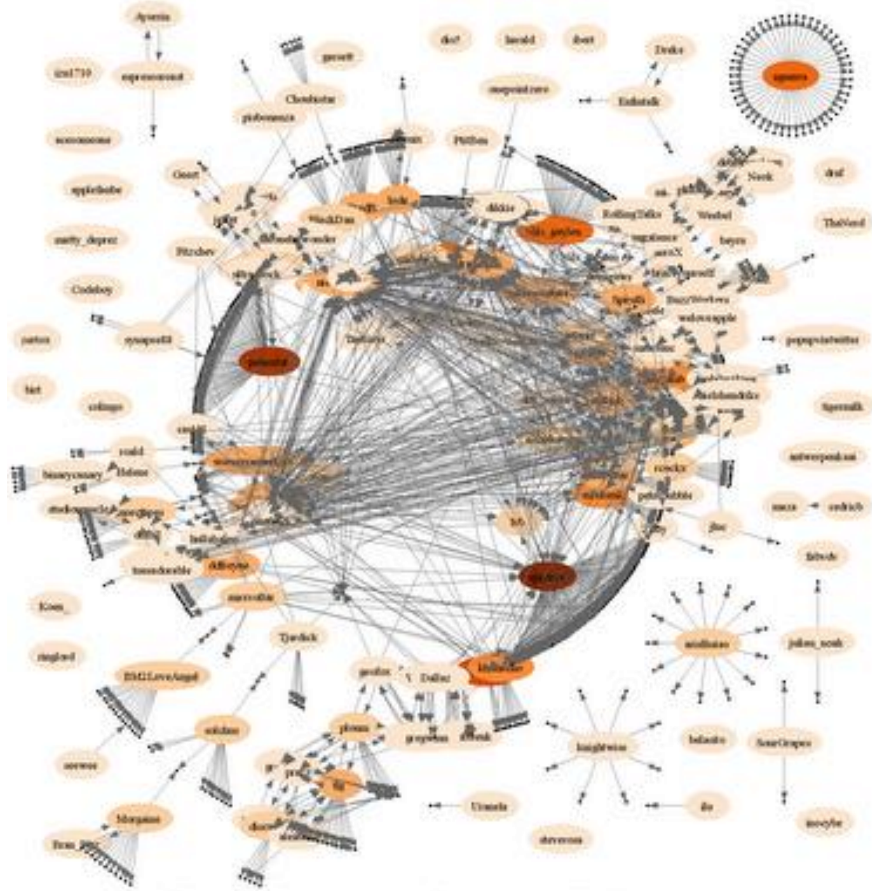


50 Billion

# Measuring, analyzing large networks

- large networks can be represented by graphs
- Facebook
- WWW
- Twitter

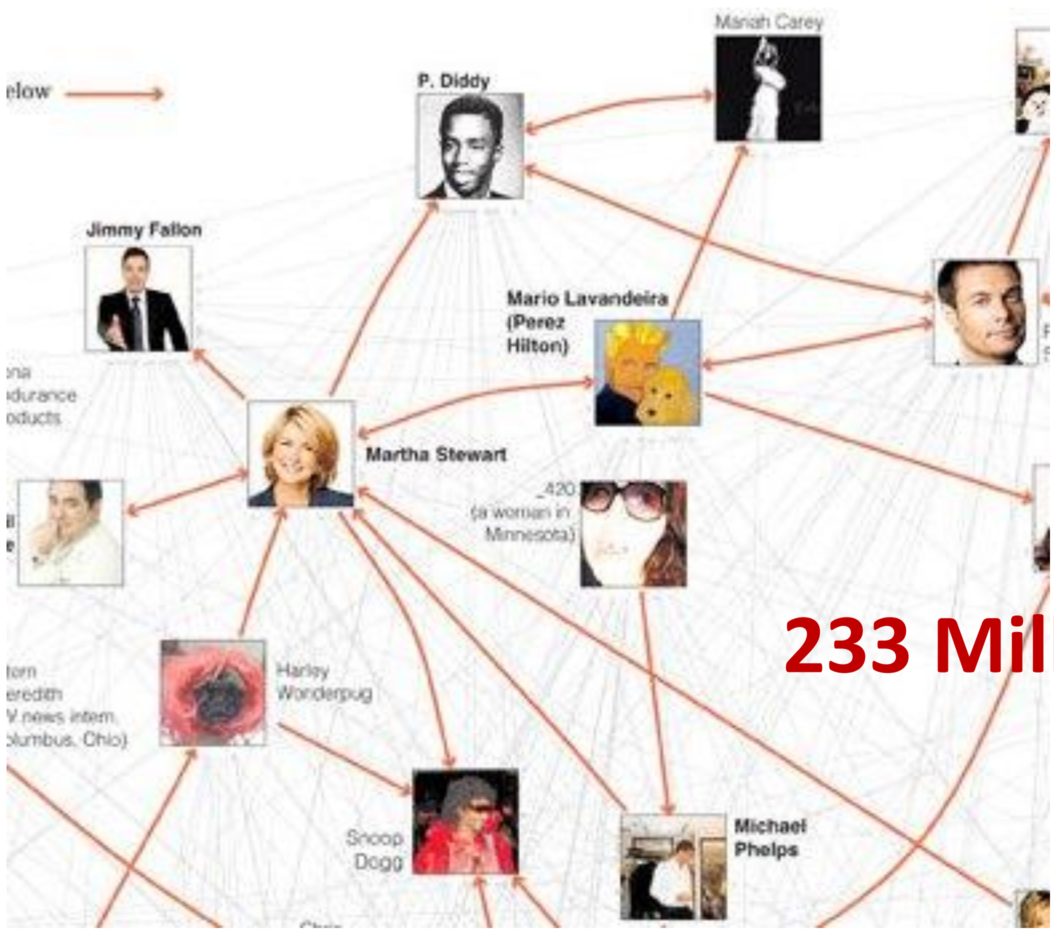
**300 million**



Twitter Friends van Belgische Twitteraars

# Measuring, analyzing large networks

- large networks can be represented by graphs
- Facebook
- WWW
- Twitter
- Ebay




**233 Million**

# Measuring, analyzing large networks

- large networks can be represented by graphs
- Facebook
- WWW
- Twitter
- Ebay

Curse of data dimensionality !!!

# Challenges in measurement: Information distortion

**"World Map" in 1459**

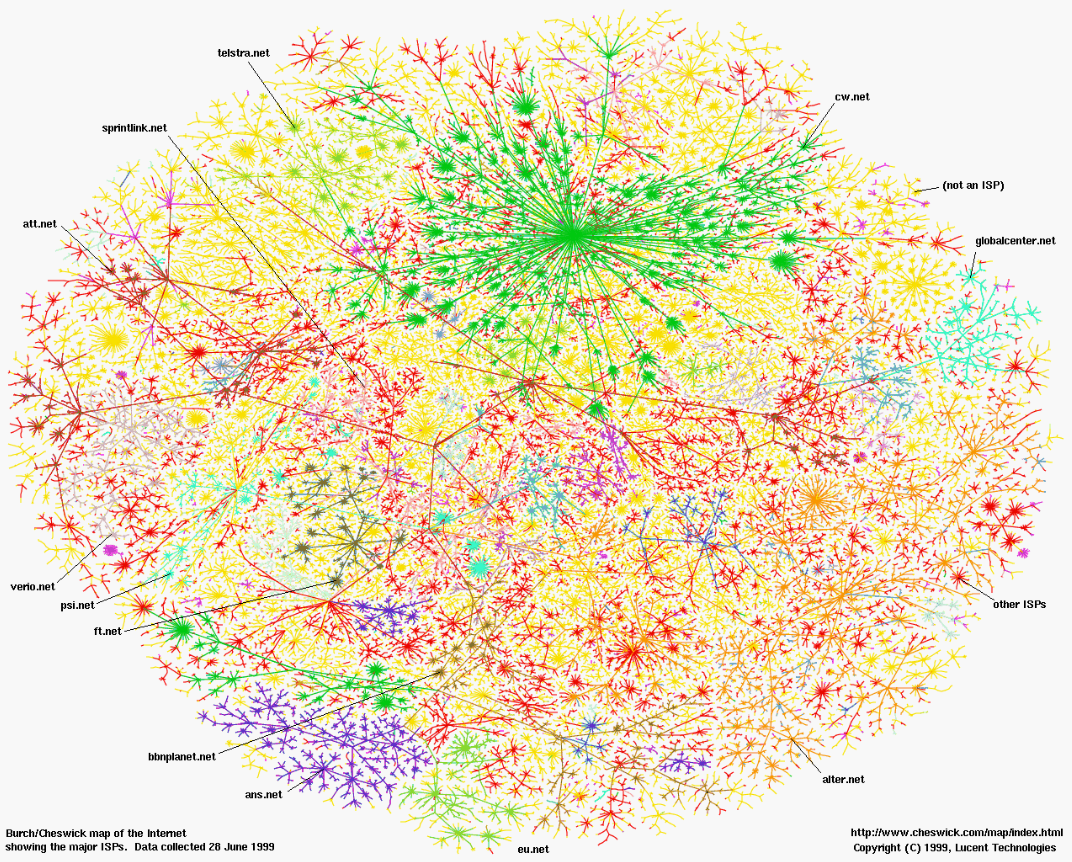
- incomplete (Columbus et al. 1492) (Australia 17$^{th}$ century)
- wrong proportions (Africa & Asia)

www.flickr.com/

# Why do we want to understand these networks?

Want to understand or find out

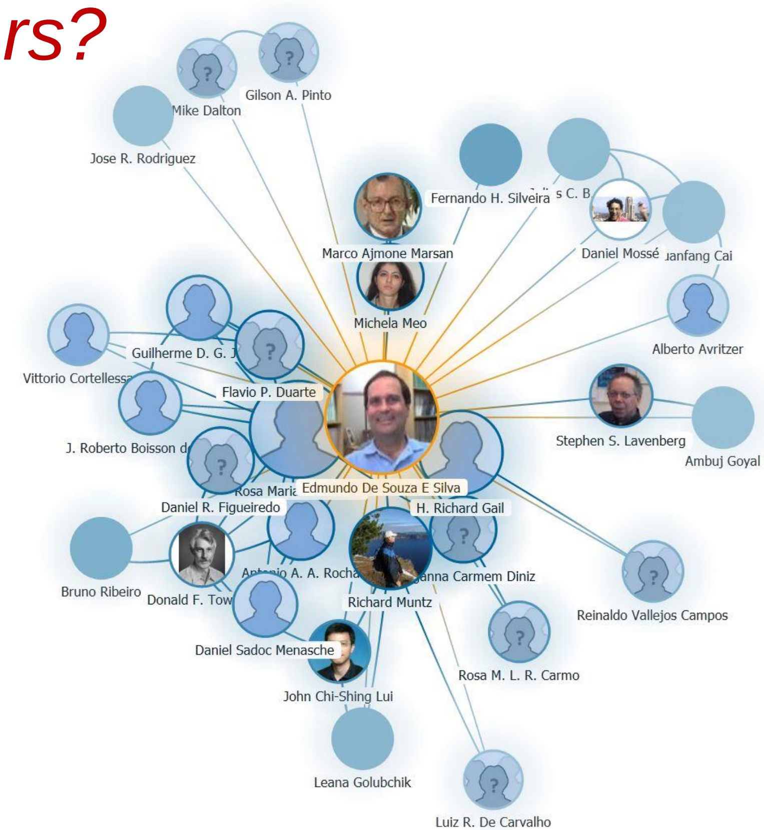
- *how did these networks evolve?*



# Why do we want to understand these networks?

Want to understand or find out

- *how did these networks evolve?*
- *who are the influential users?*

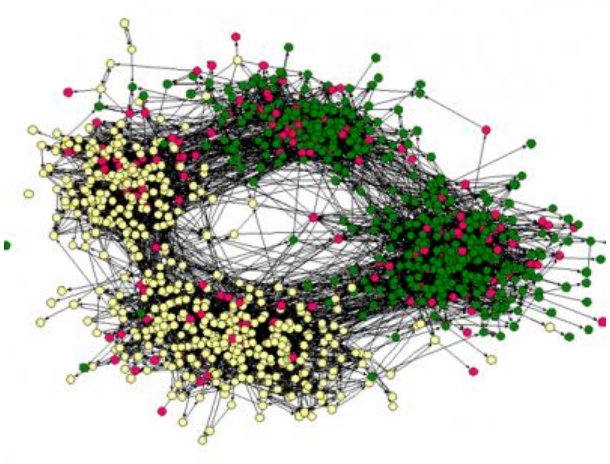


# Why do we want to understand these networks?

Want to understand or find out

- *how did these networks evolve?*
- *who are the influential users?*
- *how does influence propagate?*
- *communities in these networks?*
- ….etc.

High school friendship network



# Goals and challenges

## Goals

- generate statistically valid characterization of network structure
  - node pairs in this work

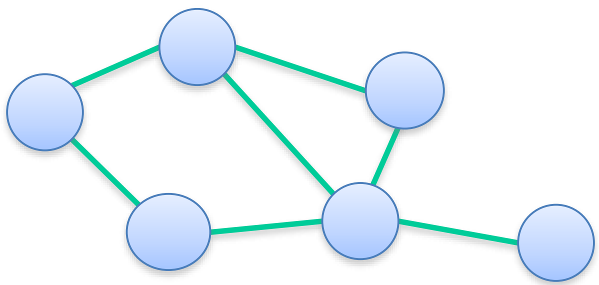
## Challenges

- large networks
- correcting for biases

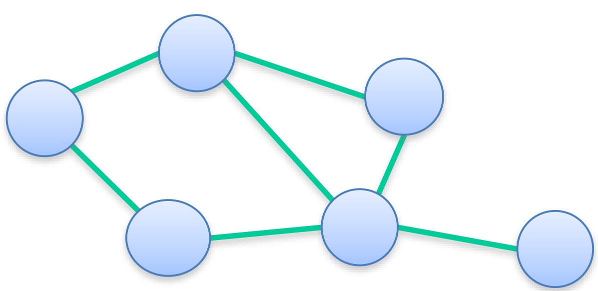
# How to measure: *sampling*

| **Random sampling** (uniform & independent) | **Crawling** |
| --- | --- |

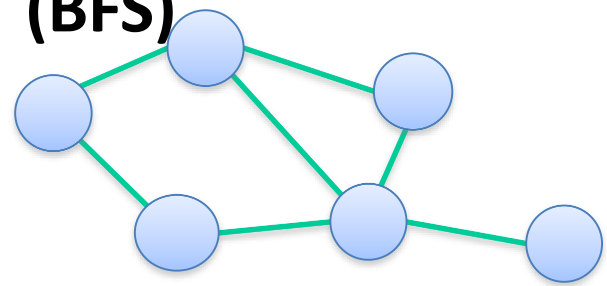
**Node sampling**



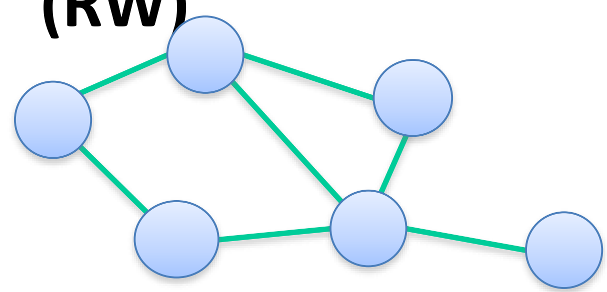
**Edge sampling**



**Breadth First sampling (BFS)**



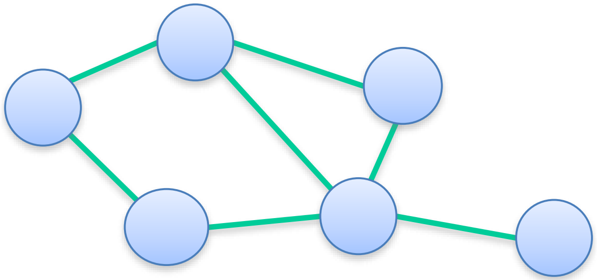
**Random walk sampling (RW)**



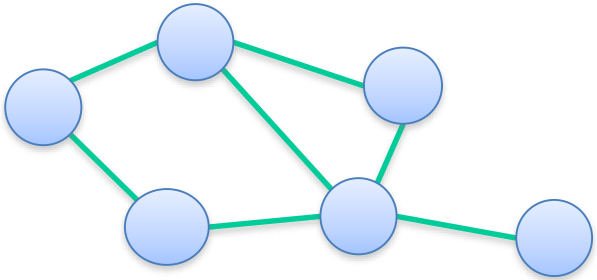
# How to measure: *sampling*

| **Random sampling** (uniform & independent) | **Crawling** |
| --- | --- |

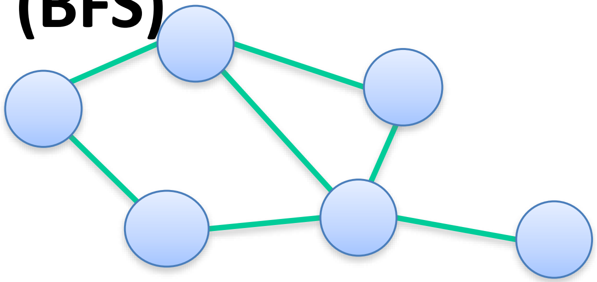
**Node sampling**



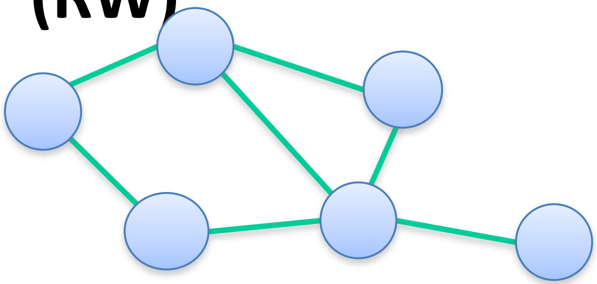
**Edge sampling**



**Breadth First sampling (BFS)**



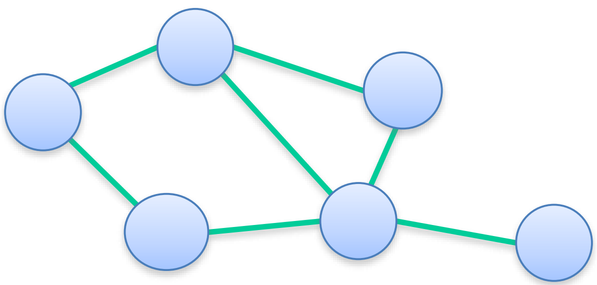
**Random walk sampling (RW)**



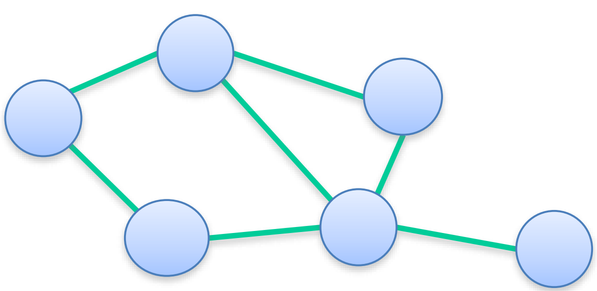
# How to measure: *sampling*

## Random sampling
(uniform & independent)

**Node sampling**

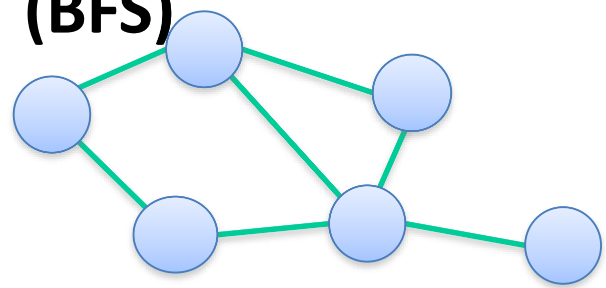


**Edge sampling**

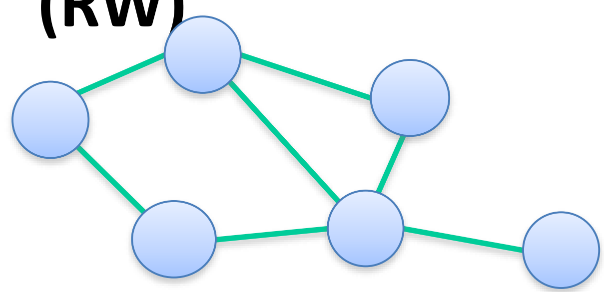


## Crawling

**Breadth First sampling (BFS)**



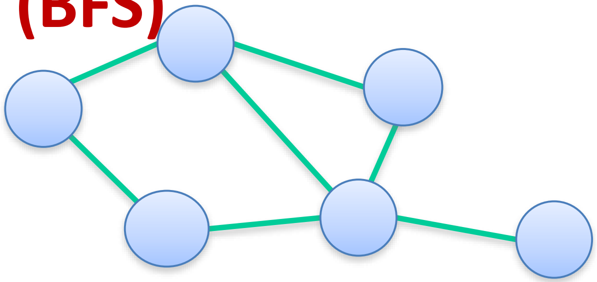
**Random walk sampling (RW)**



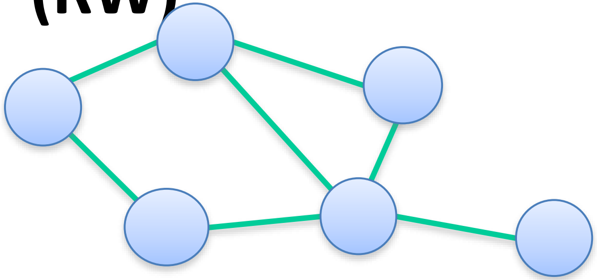
# How to measure: *sampling algorithms*

**Crawling**

**Breadth First sampling (BFS)**



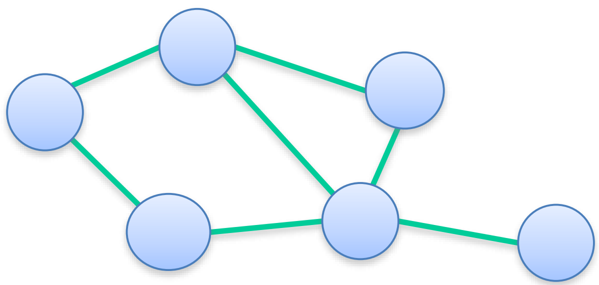
**Random walk sampling (RW)**



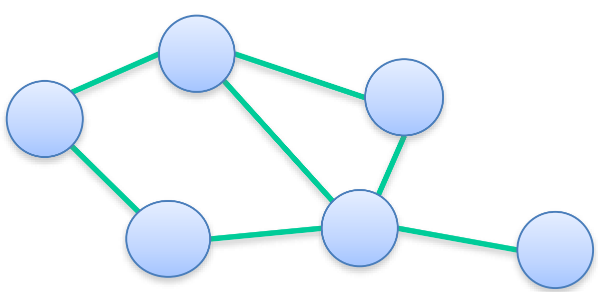
# How to measure: *sampling algorithms*

**Random sampling**
(uniform & independent)

**Node sampling**

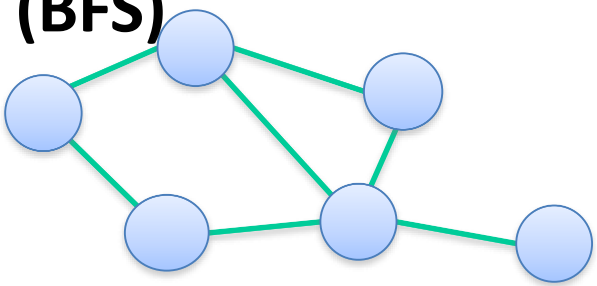


**Edge sampling**

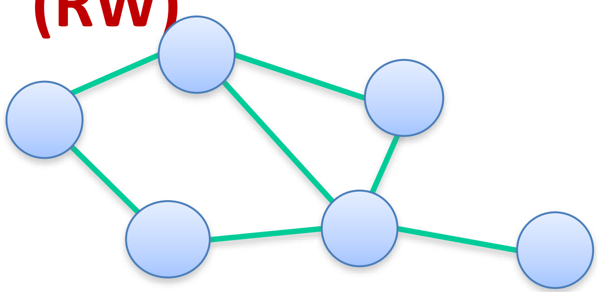


**Crawling**

**Breadth First sampling (BFS)**

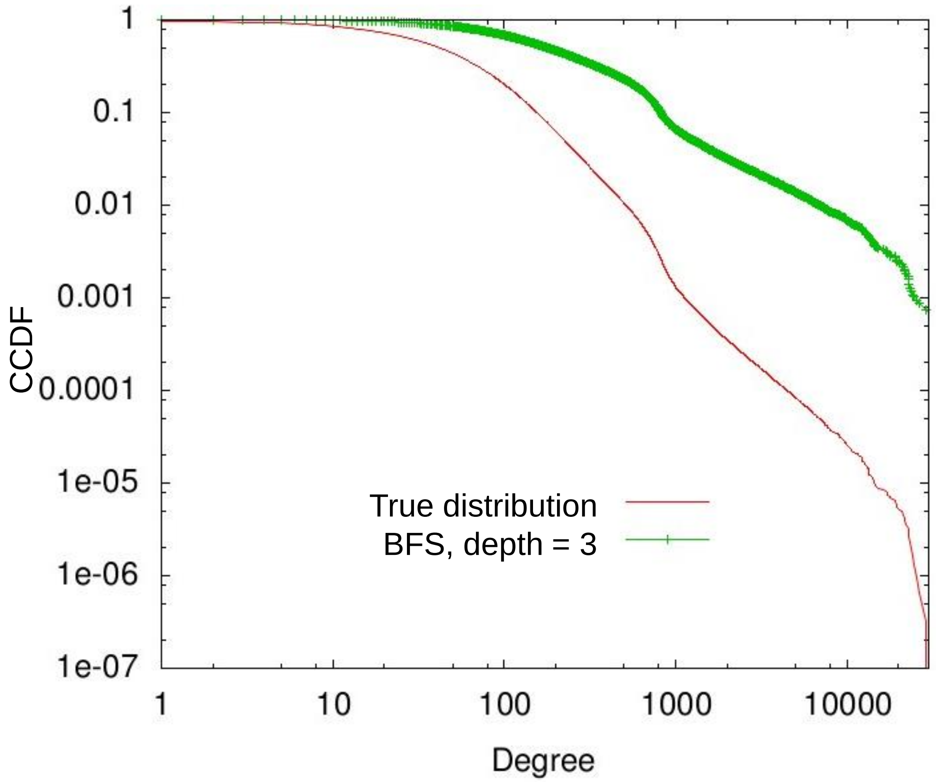


**Random walk sampling (RW)**



# Breadth first search sampling

- Orkut data set (Mislove 2007), 3M vertices, 200M edges




- BFS sampling highly biased
- difficult to remove bias

# Random walk sampling

## Bias removal?

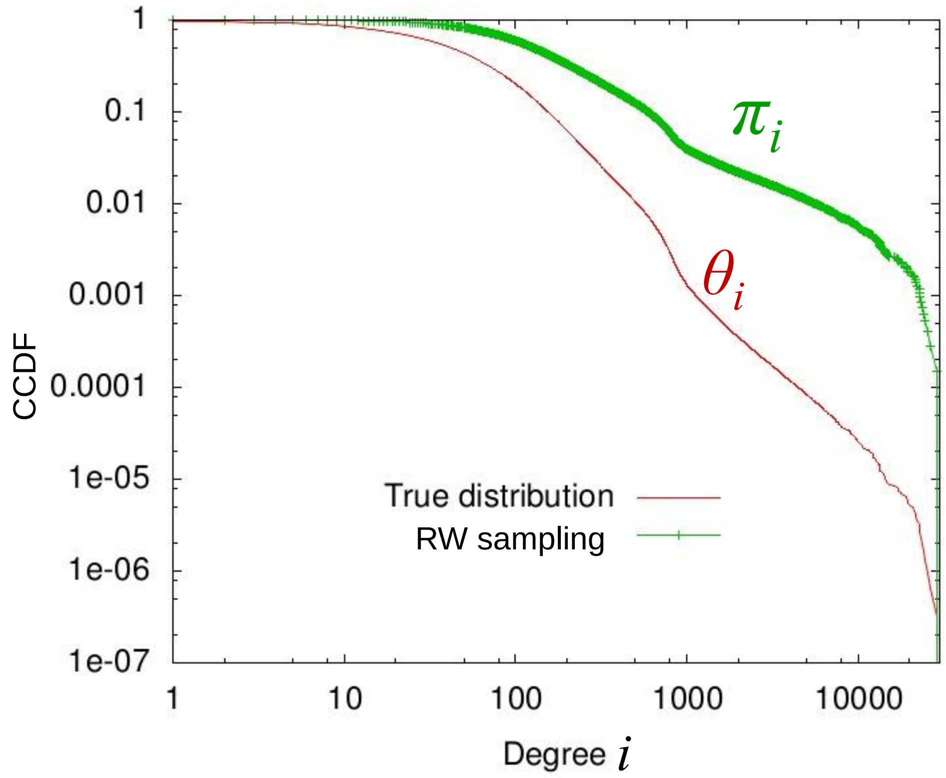
- Markov model
- at steady state visits edges uniformly at random (edge sampling)

## Model:

$\theta_i$ - P[node degree = $i$]

$\pi_i$ - P[visited degree = $i$]

$$\pi_i \propto \theta_i \times i$$



# Random walk sampling

## Bias removal?

- Markov model
- at steady state visits edges uniformly at random (edge sampling)

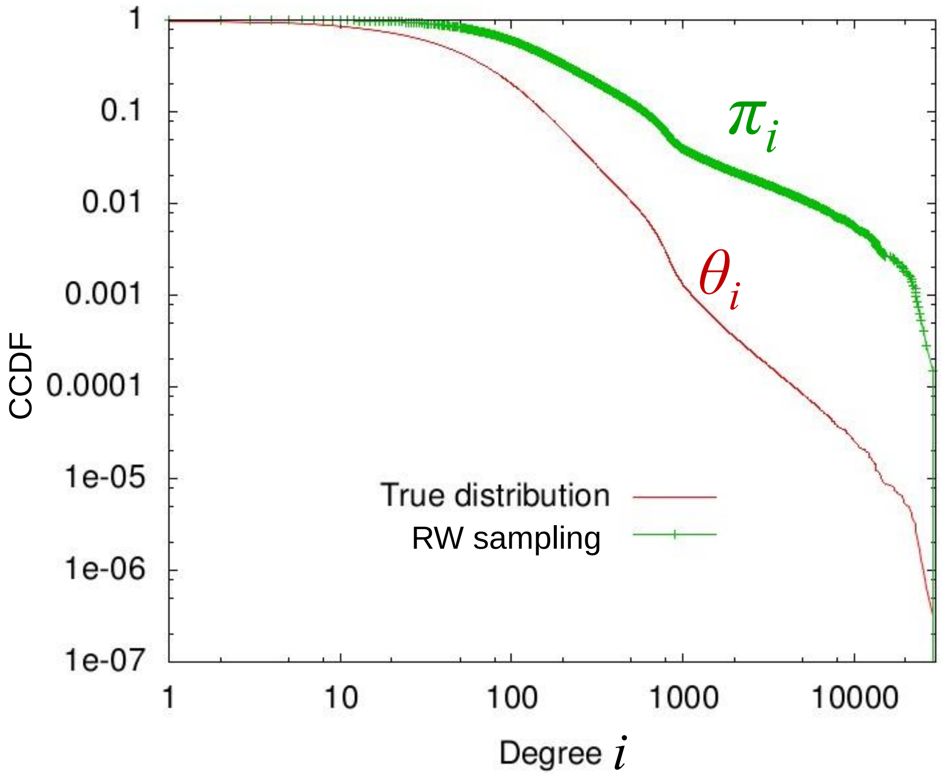
## Model:

$\theta_i$ - P[node degree = $i$]

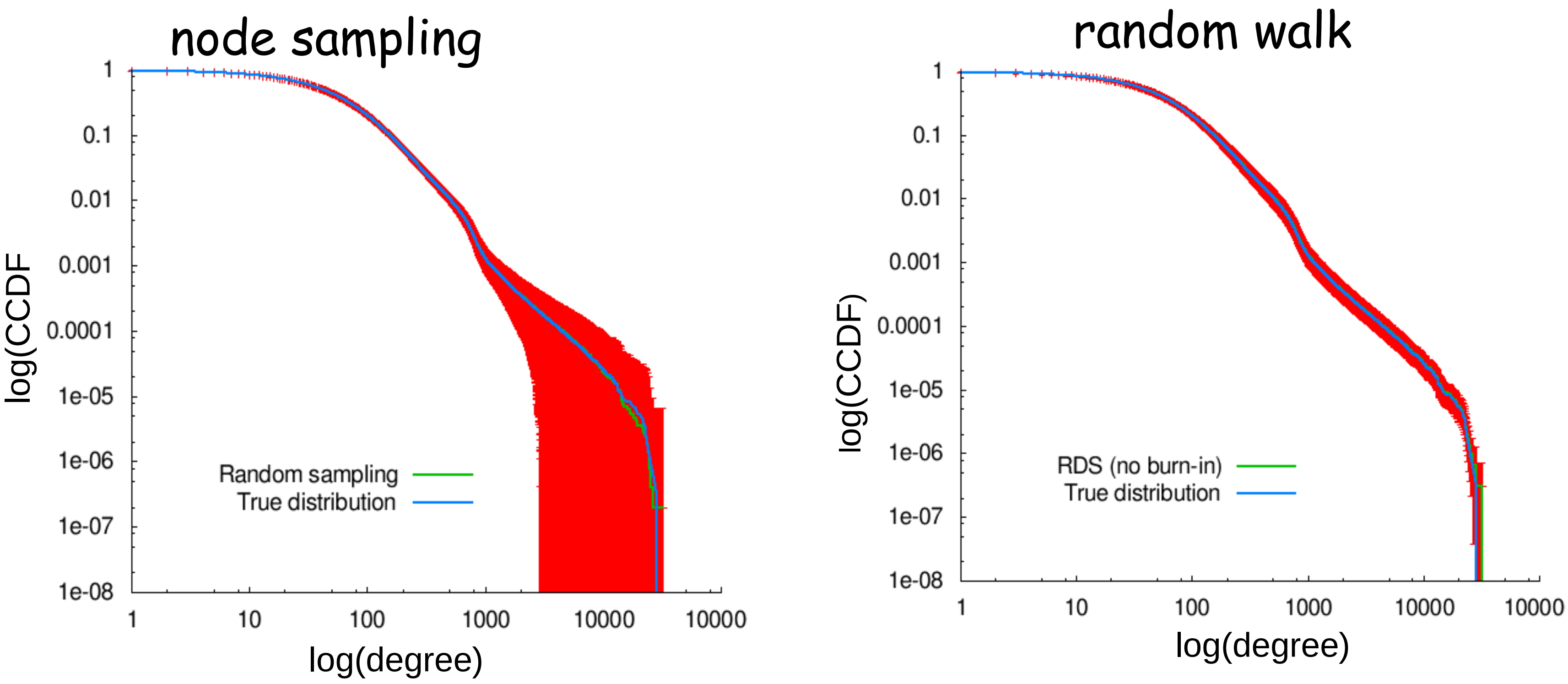
$\pi_i$ - P[visited degree = $i$]

$$\pi_i = \theta_i \times i \,/\, \text{avg degree}$$

or $$\theta_i = \text{Norm} \times \pi_i / i$$



# Node sampling vs. RW: Orkut



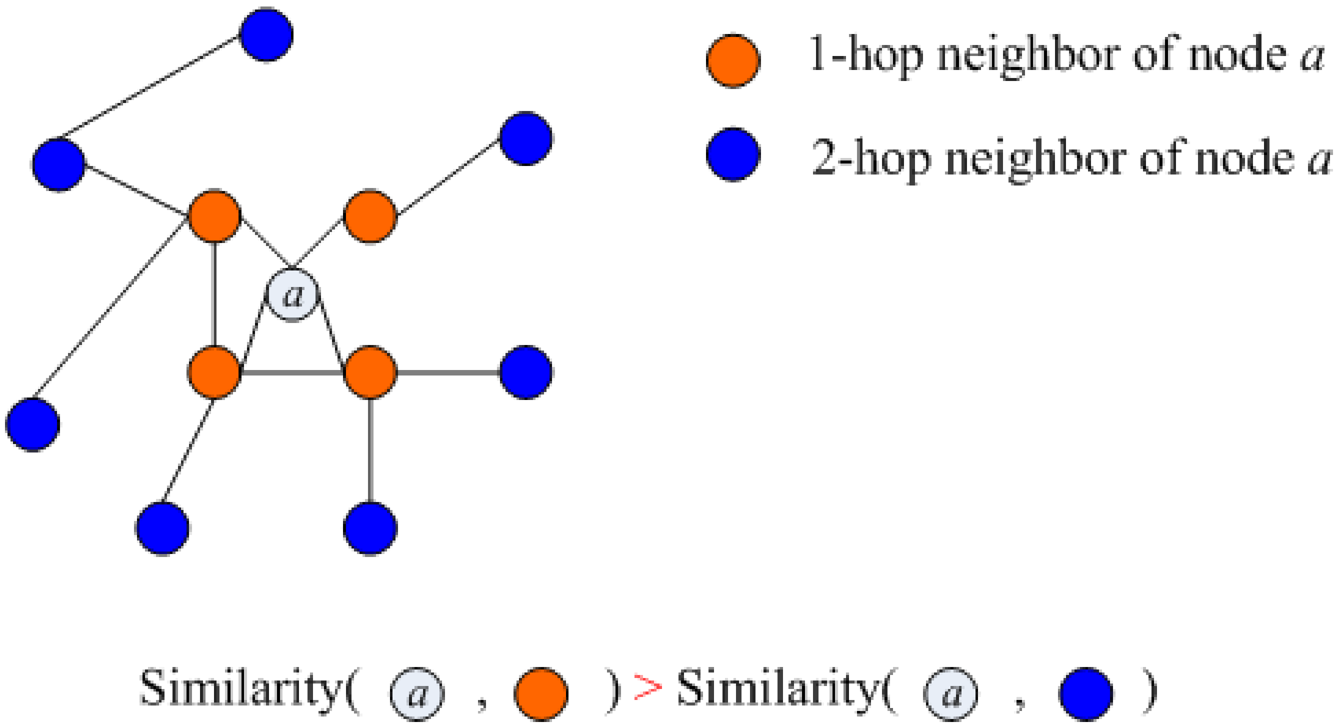

- RW – estimates tail well
- node sampling – estimates small degrees well

# Focus of talk

***Measure node pair statistics:***

***important for many applications!***

# Classification of node pairs




Classify node pair $[u, v]$ *using shortest path*

- **1-hop** node pair class if distance(*u*,*v*) = 1
- **2-hop** node pair class if distance(*u*,*v*) = 2
- …

# Homophily

Homophily: tendency of users to connect to others with common interests.

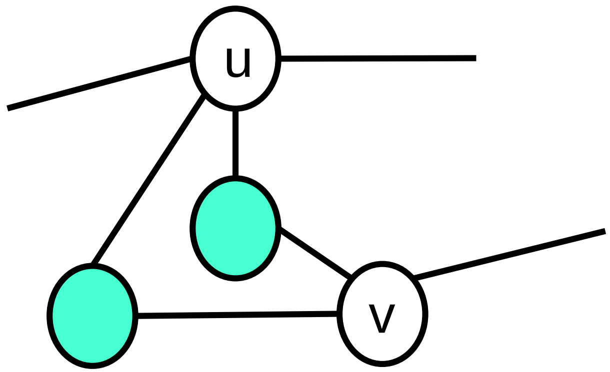
*P. Singla and M. Richardson. Yes, there is a correlation: from social networks to personal behavior on the web. In WWW 2008 (MSN)*

*Can infer characteristics and make recommendations*

Compare homophily($u$,$v$) between different node pair classes

# Pair similarity: Proximity

**Proximity(*u*,*v*)**: number of common neighbors of *u* and *v;* closeness of *u* and *v*




- knowing proximity distribution of node pairs important for
  - friendship prediction
  - interest recommendation
  - …

# Pair similarity: distance

- **Distance(*u*,*v*)**: length of shortest path between *u* and *v* in graph
- measure distance distribution of all node pairs to calculate
  - average distance
    - Twitter: 4.1
    - MSN: 6.6
  - effective diameter (the 90th percentile of all distances)
  - small world

# Problem formulation

- undirected graph $G = (V, E)$
- measure node pair characteristics in following sets:
  - all pairs - $S = \{[u, v]: u, v \in V, u \neq v\}$
  - one-hop pairs - pairs of connected nodes

$$S^{(1)} = \{[u, v]: (u, v) \in E\}$$

  - two-hop pairs - pairs of nodes with at least one common neighbor

$$S^{(2)} = \{[u, v]: u, v \in V, u \neq v; \exists x \in V \; st \; (x, u), (x, v) \in E\}$$

# Problem formulation

- $\boldsymbol{F(u,v)}$ – similarity of node pair under study, e.g., # of common neighbors of $u, v$
- $\{a_1, \ldots, a_K\}$ - range of $F(u,v)$
- distribution of $F(u,v)$
  - $S$: $(\omega_1, \ldots, \omega_K)$
  - $S^{(1)}$: $(\omega_1^{(1)}, \ldots, \omega_K^{(1)})$
  - $S^{(2)}$: $(\omega_1^{(2)}, \ldots, \omega_K^{(2)})$

  $\omega_k, \omega_k^{(1)}, \omega_k^{(2)}$- fractions of node pairs in $S, S^{(1)}, S^{(2)}$ with property $F(u,v) = a_k$

# Challenges

- OSNs large
  - Facebook, Google+, Twitter, Facebook, LinkedIn, …, $|V| > 500$ million users
- huge number of node pairs, $|V|^2 > 10^{16}$
- topology not available

  ⇒ sampling required
  - UVS (Uniform Vertex Sampling):
    - unbiased for $\boldsymbol{S}$
    - sampling bias for $\boldsymbol{S}^{(1)}, \boldsymbol{S}^{(2)}$.
    - sometimes UVS not allowed
  - crawling - RW: sampling bias
- need to construct unbiased estimates

# Node pair sampling based on UVS

Basic sampling techniques

- **UVS**: sample nodes from $V$ uniformly
- **weighted vertex sampling (WVS)**: sample nodes from $V$ with desired probability distribution $(\pi_x : x \in V)$
  - independent WVS (IWVS) (if we have topology)
  - **Metropolis-Hastings WVS (MHWVS)** (if not):

    at each step, MHWVS selects a node $v$ using UVS and then accepts the sample with probability $\min(\pi_v/\pi_u, 1)$ , where $u$ is previous sample; otherwise tries again

# Node pair sampling based on UVS

All pairs ***S***

**Sampling method:** select two different nodes $u$ and $v$ uniformly at random

**Estimator**: given sampled pairs $[u_i, v_i], i = 1, \dots, n$

$$\widehat{\omega}_k = \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}(F(u_i, v_i) = a_k), \qquad k = 1, \dots, K$$

**Accuracy** ***(unbiased)***

$$E[\widehat{\omega}_k] = \widehat{\omega}_k, k = 1, \dots, K$$

# Node pair sampling based on UVS

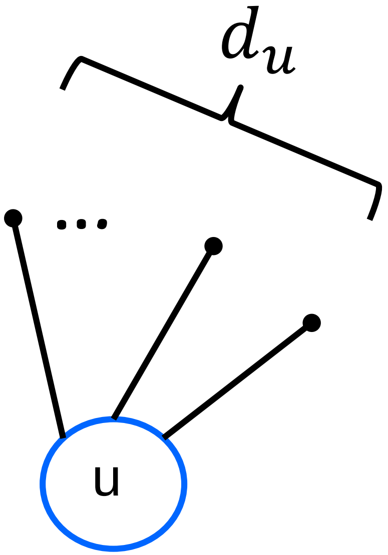
## One hop pairs $\boldsymbol{S^{(1)}}$

Sampling node pair $[u, v]$

1) sample node $u$ according to probability distribution $(\pi_u^{(1)}: u \in V)$, where

$$\pi_u^{(1)} = \frac{d_u}{2|E|}$$

$d_u$ - degree of node $u$



# Node pair sampling based on UVS

## One hop pairs $\boldsymbol{S}^{(\mathbf{1})}$

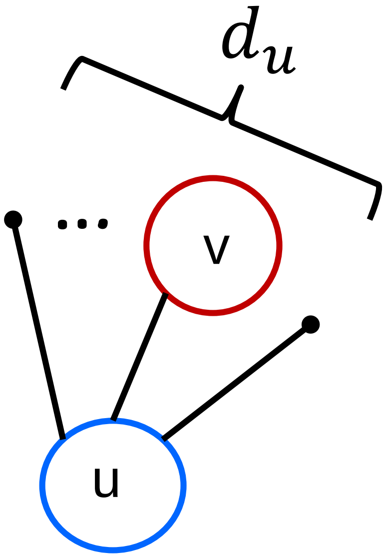
Sampling node pair $[u, v]$

1) sample node $u$ according to probability distribution $(\pi_u^{(1)}: u \in V)$, where

$$\pi_u^{(1)} = \frac{d_u}{2|E|}$$

$d_u$ - degree of node $u$

2) select neighbor $v$ at random




Each $[u, v]$ sampled uniformly from $\boldsymbol{S}^{(\mathbf{1})}$

# Node pair sampling based on UVS

One hop pairs $\boldsymbol{S^{(1)}}$

**Estimator**: given sampled pairs $[u_i, v_i], i = 1, \dots, n$

$$\hat{\omega}_k^{(1)} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}(F(u_i, v_i) = a_k), \qquad k = 1, \dots, K$$

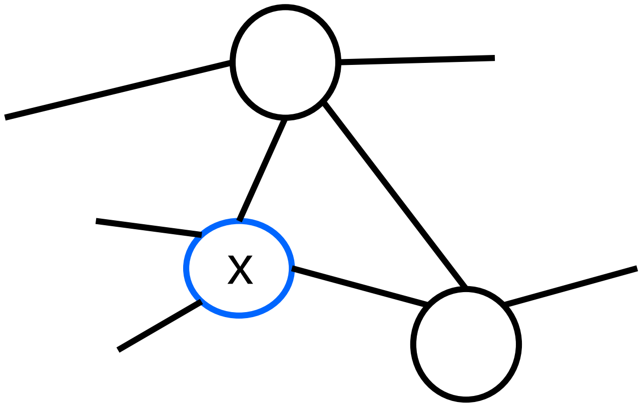
**Accuracy** *(unbiased)*

$$E\left[\hat{\omega}_k^{(1)}\right] = \omega_k^{(1)}, k = 1, \dots, K$$

# Node pair sampling based on UVS

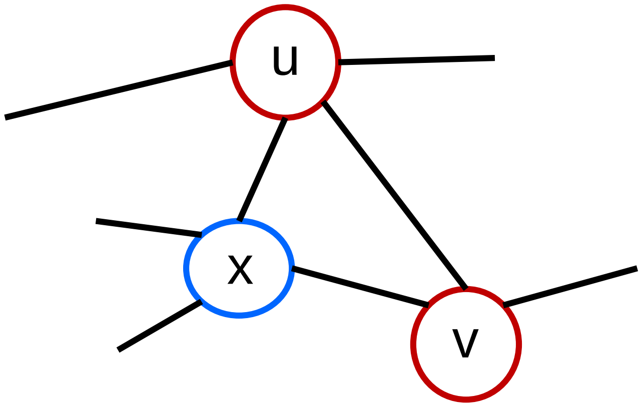
## Two hop pairs $\boldsymbol{S^{(2)}}$

- sampling node pair $[u, v]$
  1) sample node $x$
  2) select two neighbors $u$ and $v$ of node $x$ at random



# Node pair sampling based on UVS

## Two hop pairs $\boldsymbol{S^{(2)}}$




- sampling node pair $[u, v]$
  1) sample node *x*
  2) select two neighbors *u* and *v* of node $x$ at random
- produces asymptotically unbiased estimate of $\omega_k^{(2)}, k = 1, \ldots, K$
  - tight convergence rate

# Node pair sampling based on RW

**Why?**

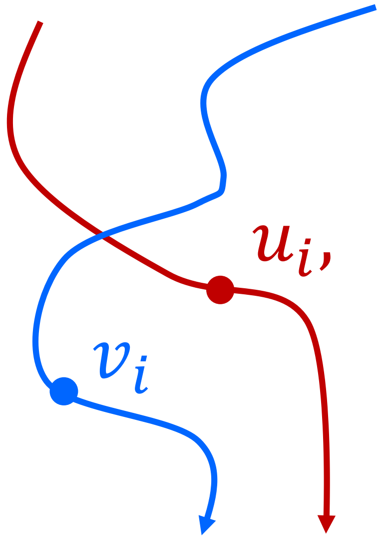
- **UVS not available, too costly**
  - API not provided
  - user IDs sparsely distributed
- only crawling techniques can be used
  - **random walk**: walker moves to random neighbor, samples its information
  - we saw for connected non-bipartite graph

$$\pi_v = \frac{d_v}{2|E|}, \qquad v \in V$$

# Node pair sampling based on RW

**All pairs *S***

- sample node pair $[u_i, v_i]$ by two independent RWs, where $u_i$, $v_i$ are nodes sampled by two RWs at step *i*
- node pair $[u,v]$ sampled according to stationary distribution




$$\pi_{[u,v]} = \frac{d_u d_v}{4|E|^2}, \qquad u, v \in V$$

# Node pair sampling based on RW

All pairs $\boldsymbol{S}$

**Estimator**: given sampled node pairs $[u_i, v_i]$, $i = 1, \ldots, n$

$$\widehat{\omega}_k^* = \frac{1}{J} \sum_{i=1}^{n} \frac{\mathbf{1}(F(u_i, v_i) = a_k)\mathbf{1}(u_i \neq v_i)}{d_{u_i} d_{v_i}}, k = 1, \ldots, K$$

$J$ – normalization constant

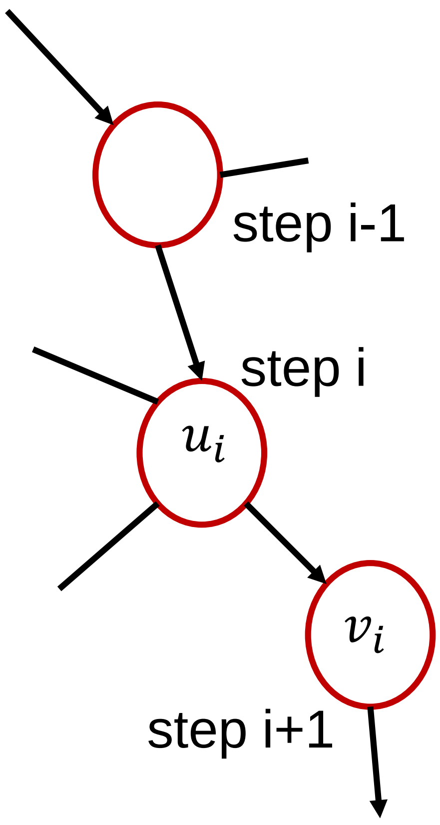
**Accuracy**: $\widehat{\omega}_k^*$ - asymptotically unbiased estimate of $\omega_k$, $k = 1, \ldots, K$

# Node pair sampling based on RW

**One hop pairs $S^{(1)}$**

**Sampling method**:

- random node pair $[u_i, v_i]$ sampled by RW
- $u_i$, $v_i$ - nodes sampled at steps $i$ and $i+1$
- produces asymptotically unbiased estimate of $\omega_k^{(1)}$, $k = 1, \ldots, K$

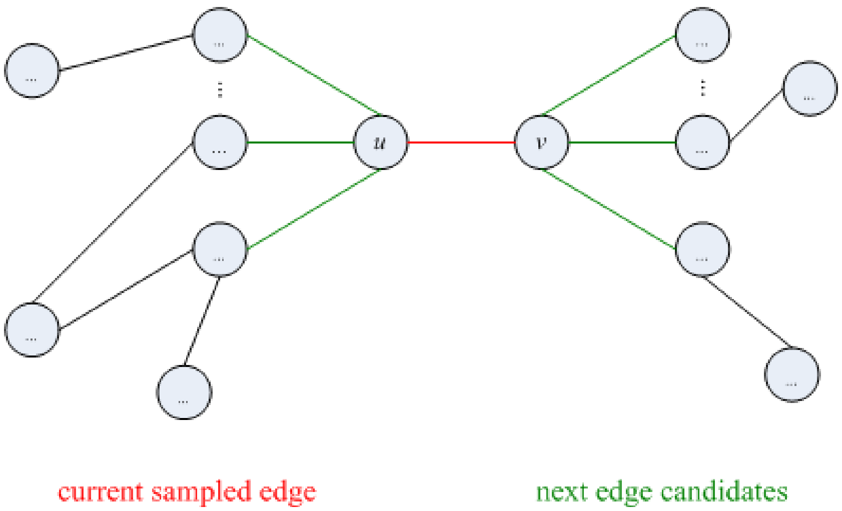


# Node pair sampling based on RW

## Two hop pairs $S^{(2)}$

Neighborhood RW (NRW)

- current edge $(u, v)$
- next edge: select randomly from edges connected to $u$ or $v$, except edge $(u, v)$




- RW on graph with edges as nodes

# Node pair sampling based on RW

**Two hop pairs $\boldsymbol{S^{(2)}}$**

Probability NRW samples node pair $[u,v]$ in $\boldsymbol{S^{(2)}}$ converges to

$$\pi_{[u,v]}^{(2)} = m(u,v)/M$$

$m(u,v)$ - number neighbors common to $u$, $v$

# Node pair sampling based on RW

## Two hop pairs $S^{(2)}$

**Estimator**: given sampled node pairs $[u_i, v_i], i = 1, \dots, n$

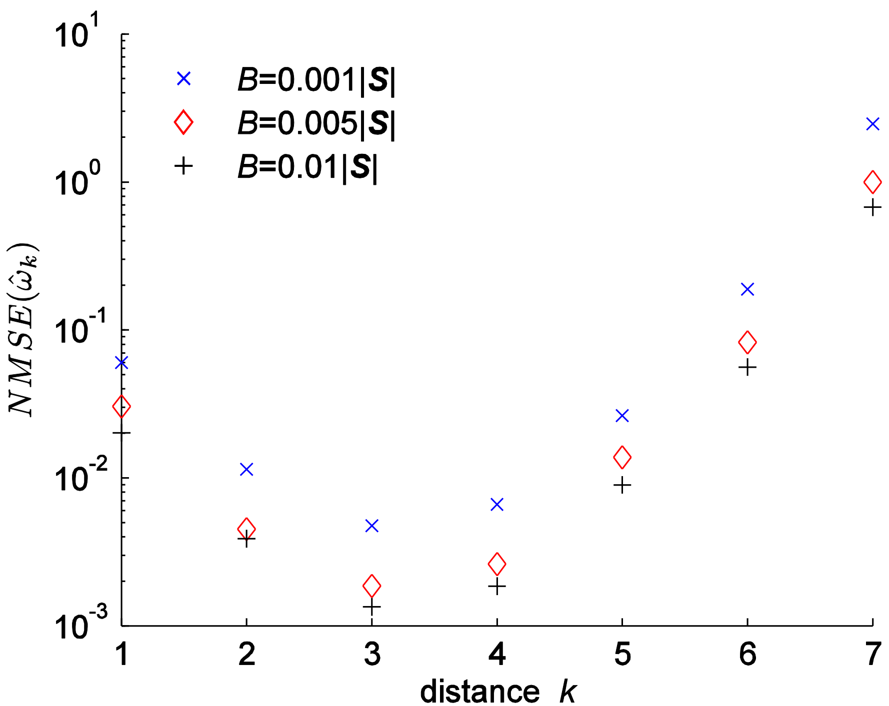
$$\hat{\omega}_k^{(2^*)} = \frac{1}{H} \sum_{i=1}^{n} \frac{\mathbf{1}(F(u_i, v_i) = a_k)}{m(u_i, v_i)}$$

$H$ – normalization constant

**Accuracy**: asymptotically unbiased estimate of $\omega_k^{(2)}, 1 \le k \le K$

# Simulations: Distance distribution estimation

- *B* - number of sampled node pairs
- |S| - total number of node pairs
- error metric - $NMSE(\widehat{\omega}_k) = \frac{\sqrt{E[(\widehat{\omega}_k - \omega_k)^2]}}{\omega_k}, \ k = 1, \ldots, K$

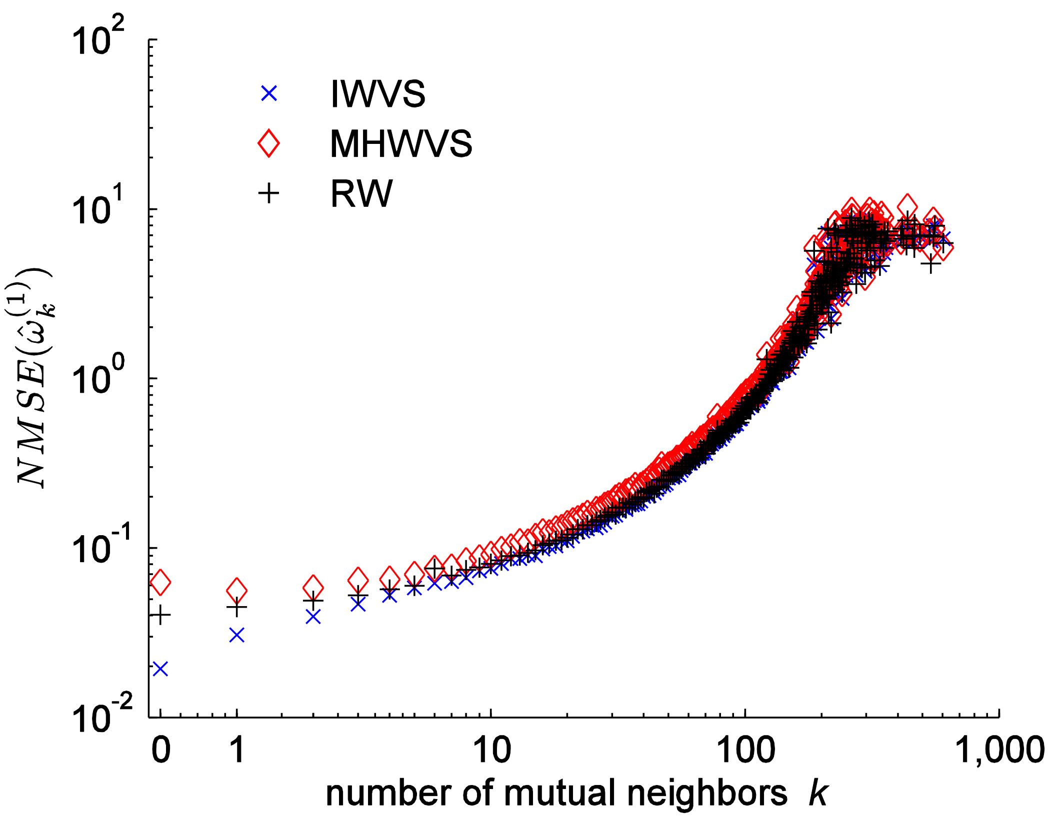



Gnutella - $|V| \approx 6300$

- B >.005|S|, NMSE < 1

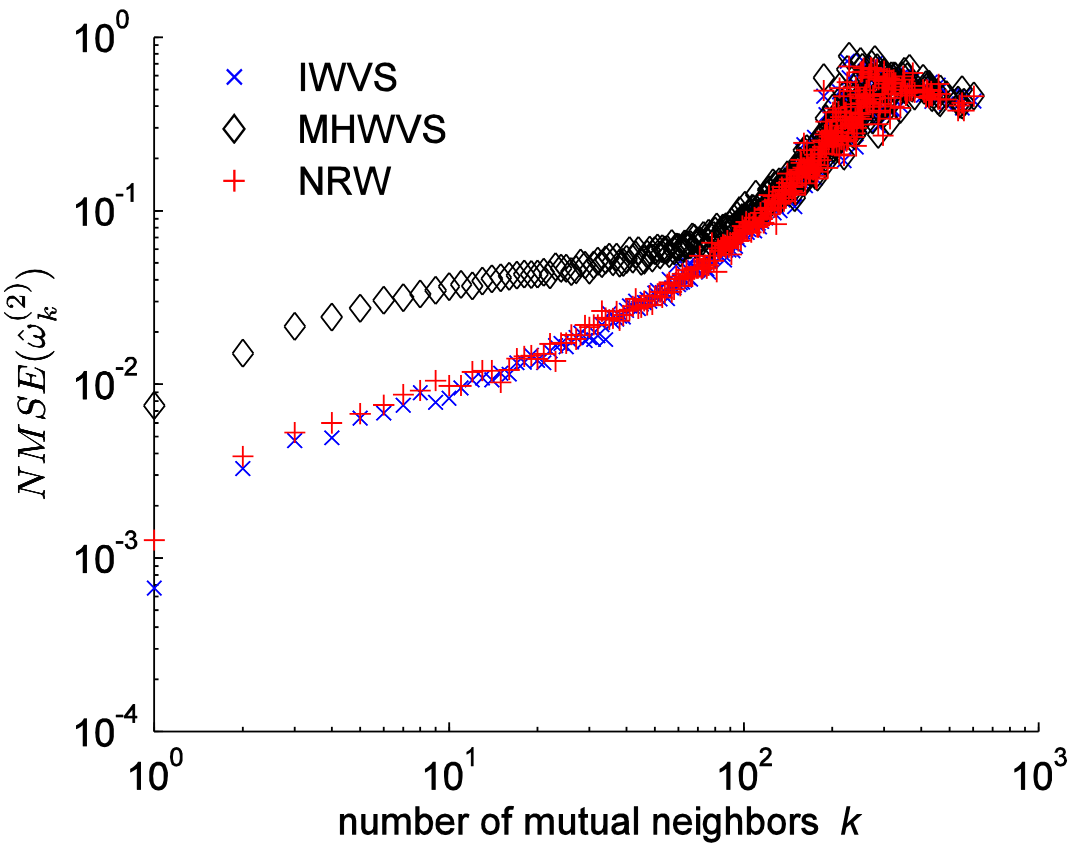
# Simulations: Mutual neighbor count distribution in $S^{(1)}$

$B = 0.01|S^{(1)}|$ node pairs sampled from $S^{(1)}$ of soc-Epinions (76,000 nodes)



# Simulations: Mutual neighbor count distribution in $S^{(2)}$

$B = 0.01|S^{(2)}|$ node pairs sampled from $S^{(2)}$ of soc-Epinions



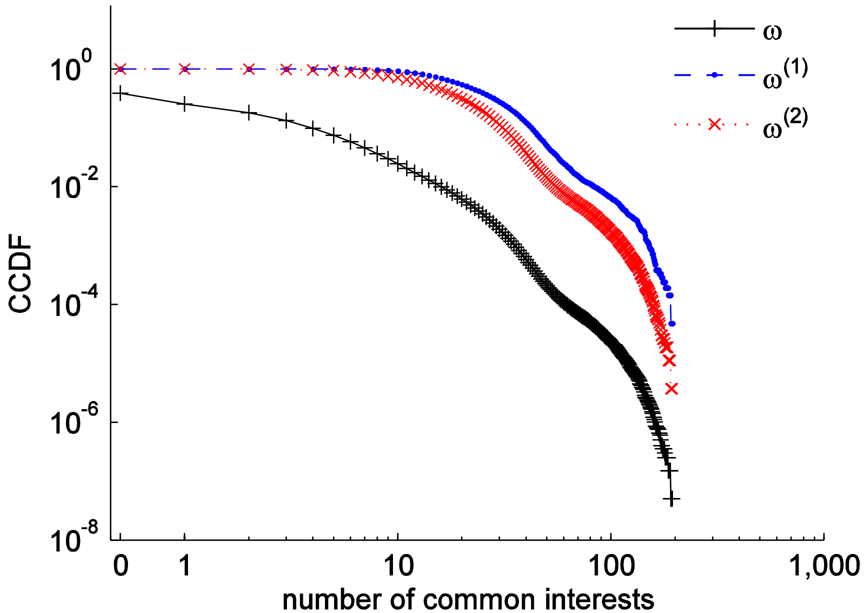
# Simulations: Interest distribution scheme

Distribute interests over graph:

- $10^5$ distinct interests: number nodes per interest ~ truncated Pareto distribution over $\{1, \ldots, 10^3\}$
- to distribute interest possessed by *k* different nodes
  1. select random node *v* that *reaches at least k-1 different nodes*
  2. distribute interest to node *v* and closest *k*-1 nodes connected to *v*

# Simulations: Common interest count distribution for generated content

- CCDF of common interest count distribution for S, $S^{(1)}$, and $S^{(2)}$
- # common interests smallest for S, largest for $S^{(1)}$
- consequence of construction

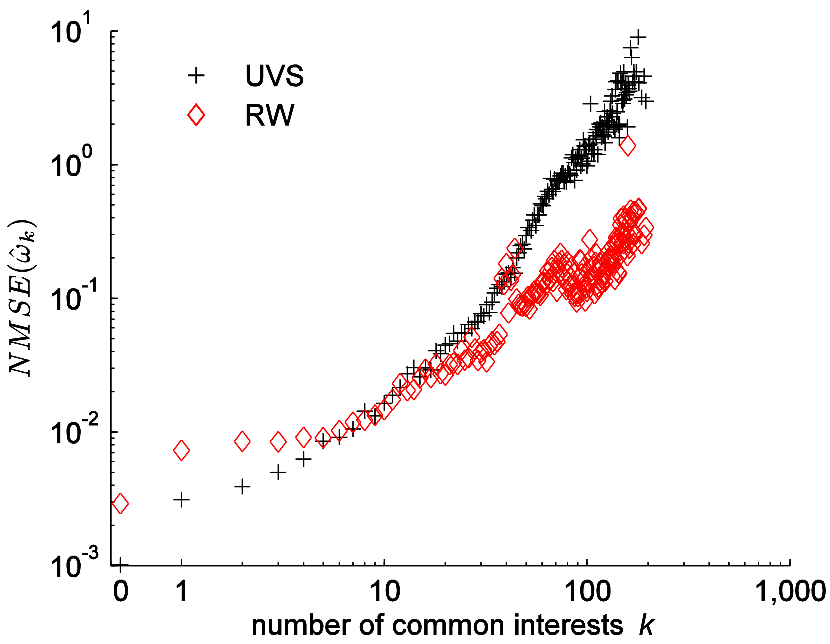



P2P-Gnutella

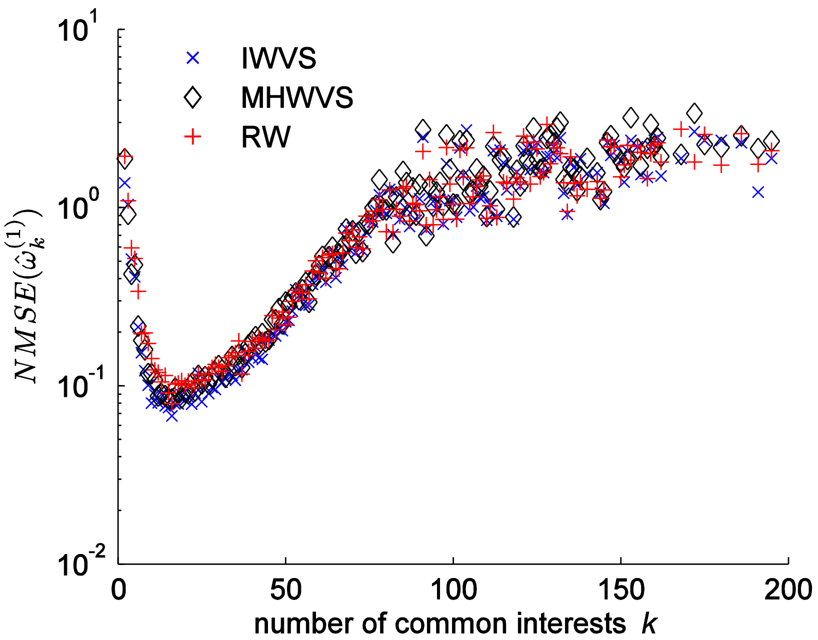
# Simulations: Common interest count distribution in S, $S^{(1)}$ (Gnutella)

$B = 0.01|S|$ node pairs sampled from $S, S^{(1)}$

S




$S^{(1)}$

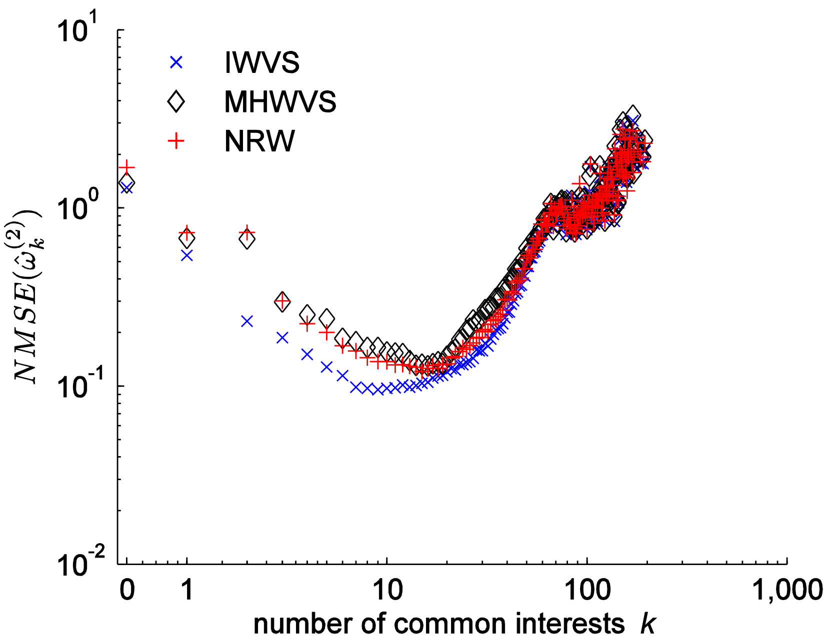



- RW better than UVS for all pairs
- little difference for neighbors

# Simulations: Common interest count distribution in $S^{(2)}$ (Gnutella)

$B = 0.01|S^{(2)}|$ node pairs sampled from $S^{(2)}$

- IWVS better for small numbers of interests
  - requires knowledge of topology



# Conclusions

- use sampling to estimate pair characteristics in sets S, $S^{(1)}$, and $S^{(2)}$.
- sampling methods based on independent vertex sampling and random walk
  - produce asymptotically unbiased estimates
- good illustration of power of random walk
- validated approaches on wide range of graphs

# Conclusions

- Markov Chain Mixing Times
- other more “powerful” & “elegant” sampling methods: Frontier Sampling (Ribeiro)
- Efficiently Estimating Motif Statistics of Large Networks in the Dark. TKDD 2014
- Design of Efficient Sampling Methods on Hybrid Social-Affiliation Networks. IEEE ICDE’15
- measuring, maximizing group closeness centrality over disk-resident graphs. WWW’14

# Thanks!

Slides (will be) at

http://www-net.cs.umass.edu/networks/towsley/UFRJ-sampling.pdf