Inference in OSNs via Lightweight Partial Crawls

Jithin K. Sreedharan
Inria, France

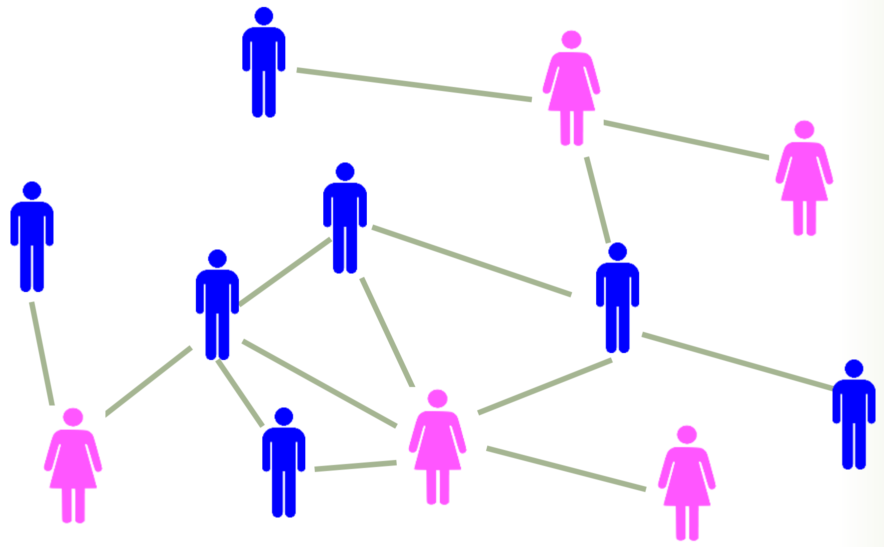
Konstantin Avrachenkov
Inria, France

Bruno Ribeiro
Purdue University, USA

Sigmetrics 2016, June 16

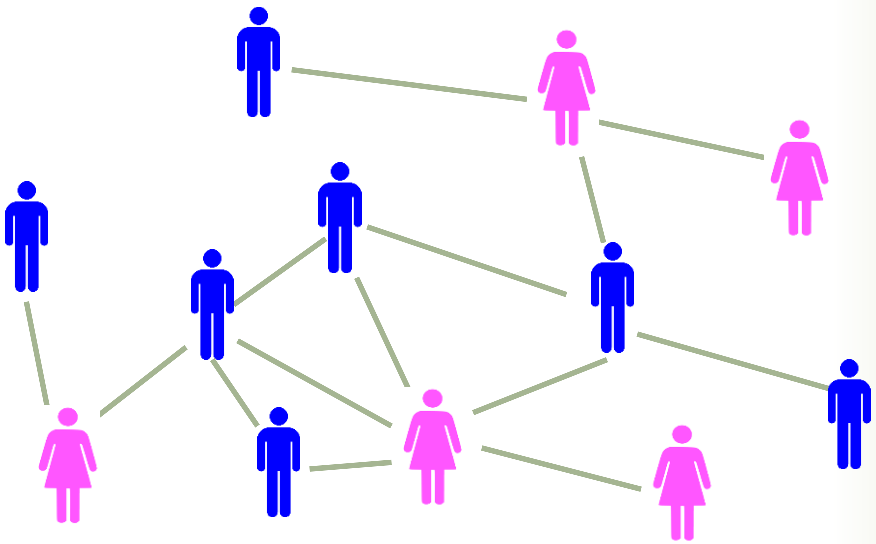
Motivation

- Estimation and inference in Online Social Network (OSN)
- Example:
  OSN users more likely to form edges with those with similar attributes ?



Motivation

- Estimation and inference in Online Social Network (OSN)
- Example:
  OSN users more likely to form edges with those with similar attributes ?



Easy to answer if the graph is fully known beforehand

What if the network is not known?

- Can only crawl network
- Few queries

Problem definition

Problem definition

Let $G = (V, E)$

Problem definition

Let $G = (V, E)$

- Undirected graph

Problem definition

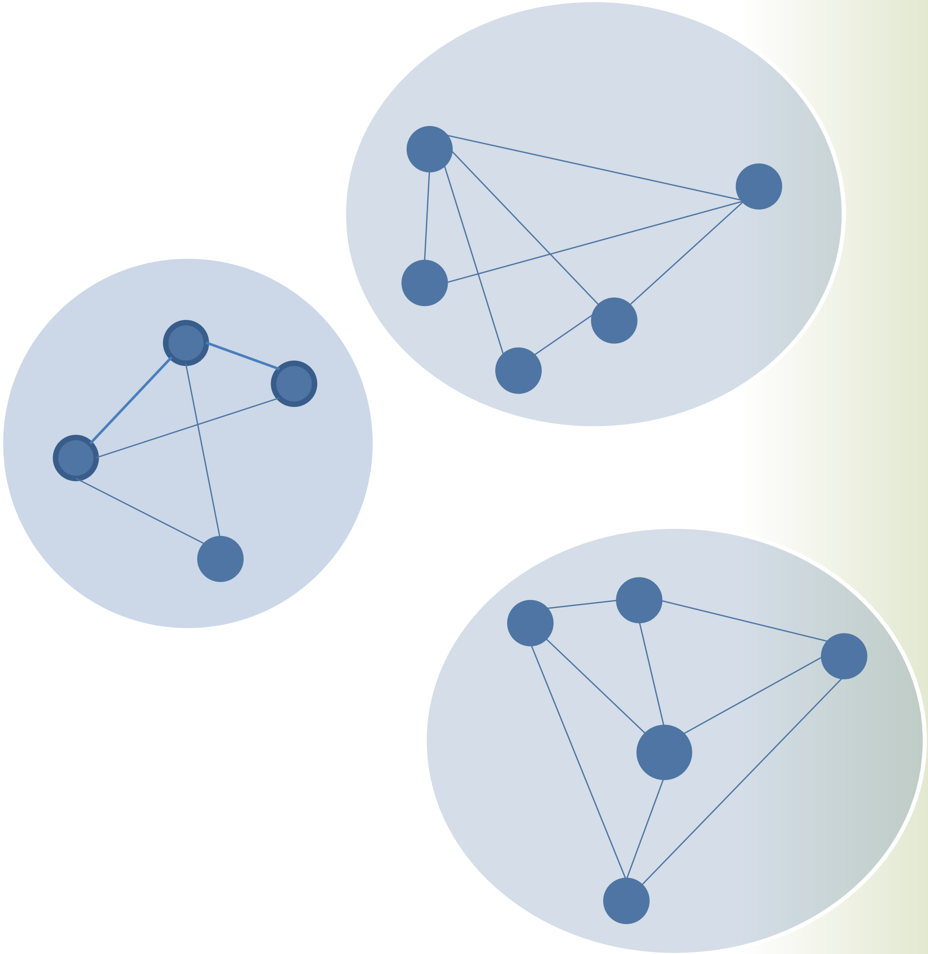
Let $G = (V, E)$

- Undirected graph
- Node and edge have labels

Problem definition

Let $G = (V, E)$

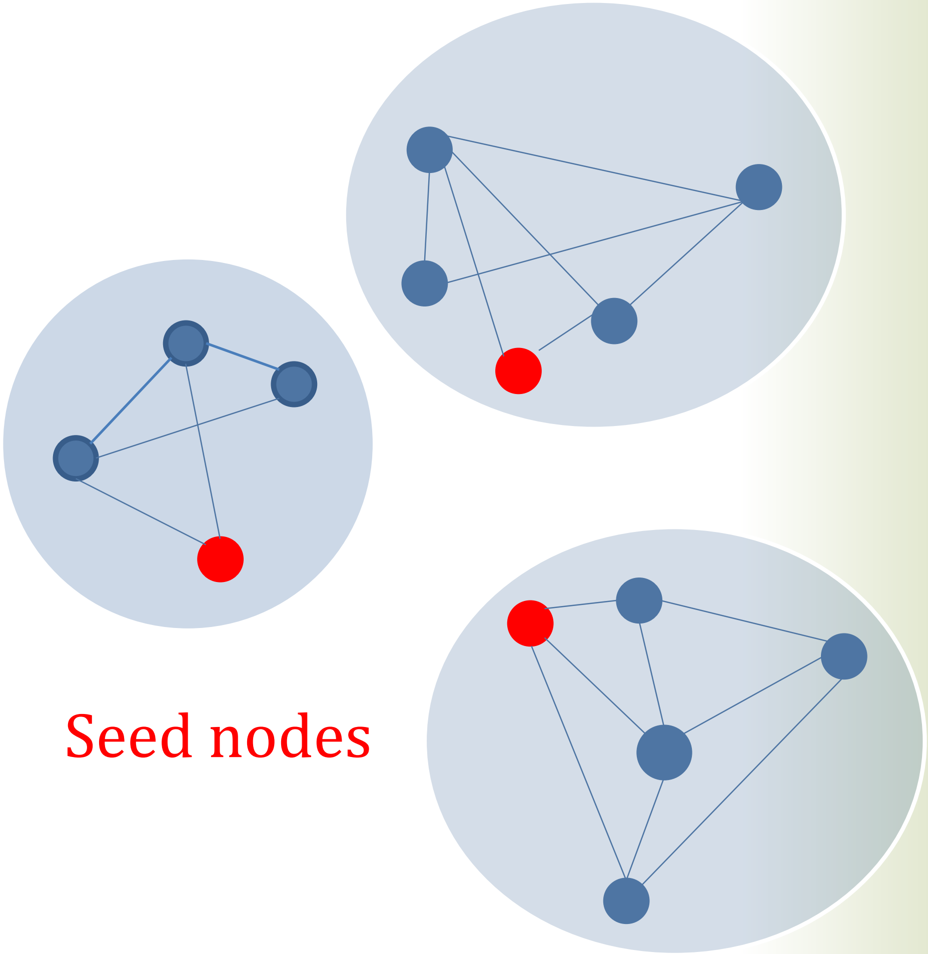
- Undirected graph
- Node and edge have labels
- Not necessarily connected or has included connected components of interest



Problem definition

Let $G = (V, E)$

- Undirected graph
- Node and edge have labels
- Not necessarily connected or has included connected components of interest
- Few seed nodes



Problem definition

Let $G = (V, E)$

- Undirected graph
- Node and edge have labels
- Not necessarily connected or has included connected components of interest
- Few seed nodes
- Large graph

Problem definition (contd.)

Problem definition (contd.)

Estimate $\mu(G) = \sum_{(u,v)\in E} g(u,v)$

Problem definition (contd.)

Estimate $\mu(G) = \sum_{(u,v)\in E} g(u,v)$

- Graph is unknown

Problem definition (contd.)

Estimate $\mu(G) = \sum_{(u,v)\in E} g(u,v)$

- Graph is unknown
- Only local information available

$\left\{ \begin{array}{l} \text{Seed nodes and their neighbor IDs} \\ \text{Query (visit) a neighbor} \\ \text{Visited nodes and their neighbor IDs} \end{array} \right.$

Problem definition (contd.)

Estimate $\mu(G) = \sum_{(u,v)\in E} g(u,v)$

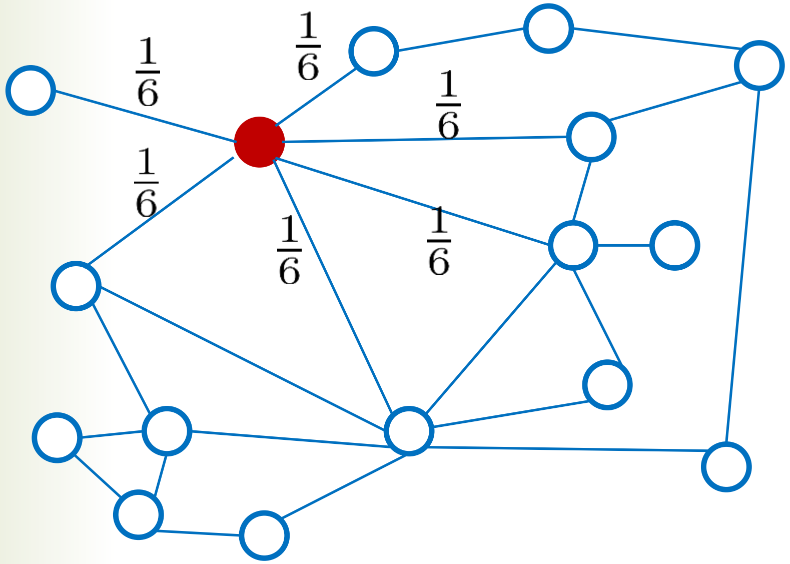
- Graph is unknown
- Only local information available

$\left\{ \begin{array}{l} \text{Seed nodes and their neighbor IDs} \\ \text{Query (visit) a neighbor} \\ \text{Visited nodes and their neighbor IDs} \end{array} \right.$

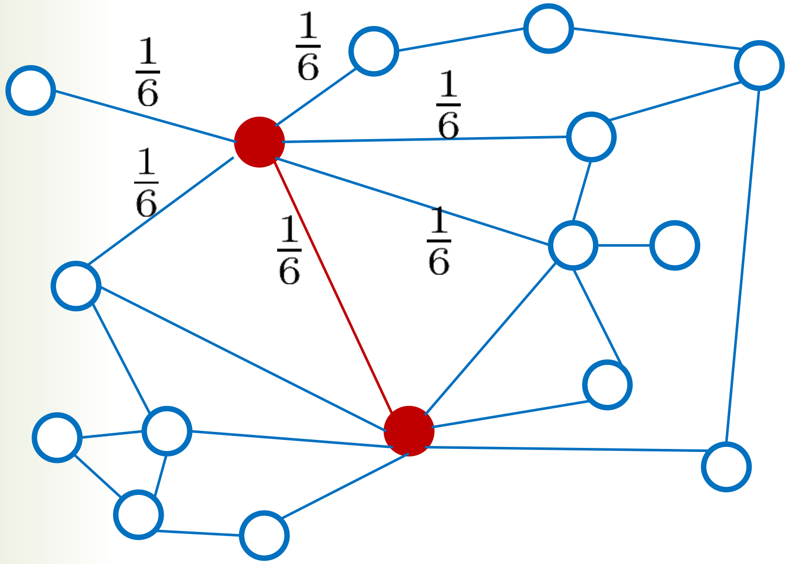
How do we know in real time if our estimates are accurate?

Random walk based estimation

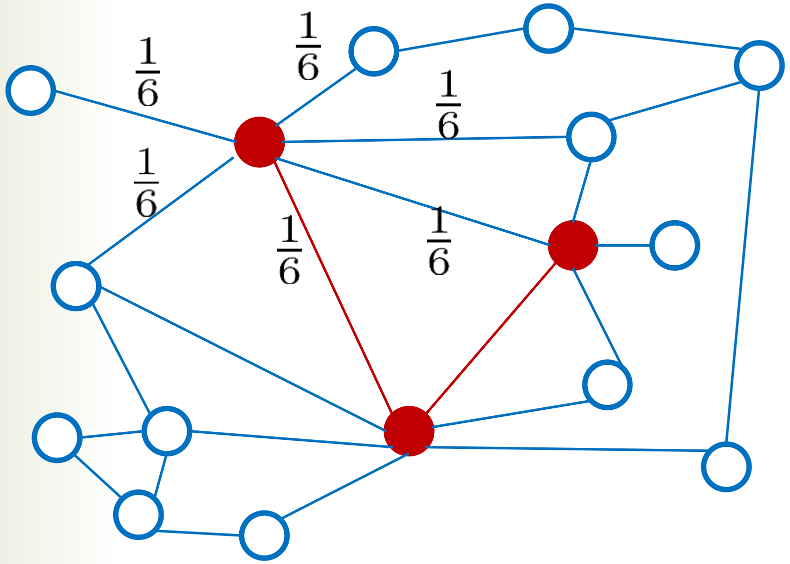
Random walk based estimation



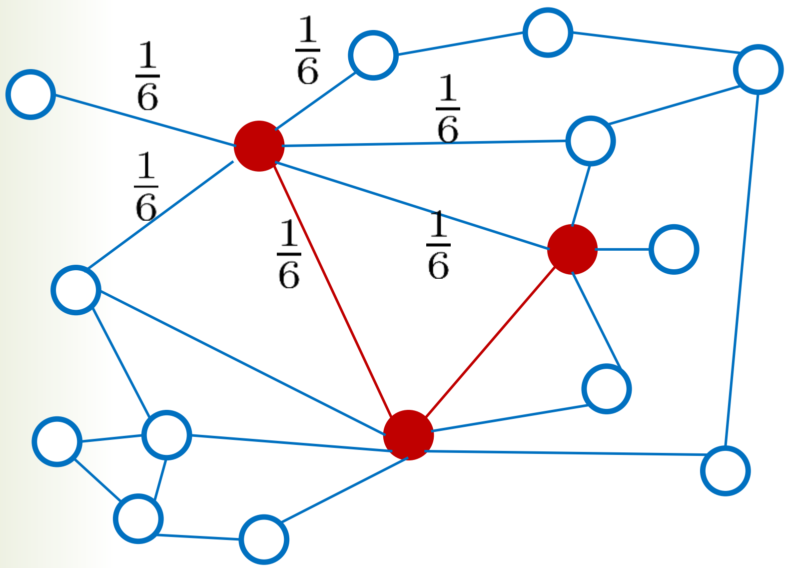
Random walk based estimation



Random walk based estimation



Random walk based estimation

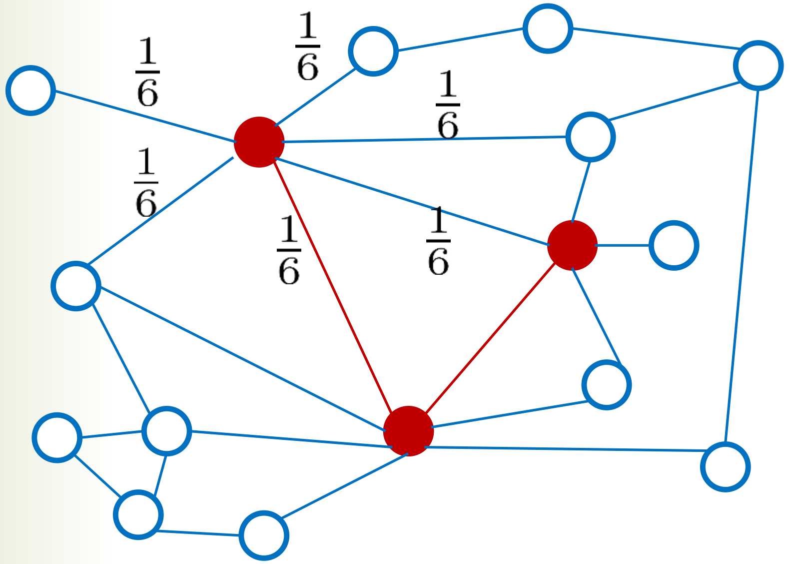


Random walk $\{X_k\}_{k\geq 1}$ has unique stationary distribution $\{\pi_i\}_{i=1}^n$ if graph G is connected and non-bipartite

- Goal:

  Estimate $\mu(G) = \sum_{(u,v)\in E} g(u,v)$

Random walk based estimation



Random walk $\{X_k\}_{k\geq 1}$ has unique stationary distribution $\{\pi_i\}_{i=1}^{n}$ if graph G is connected and non-bipartite

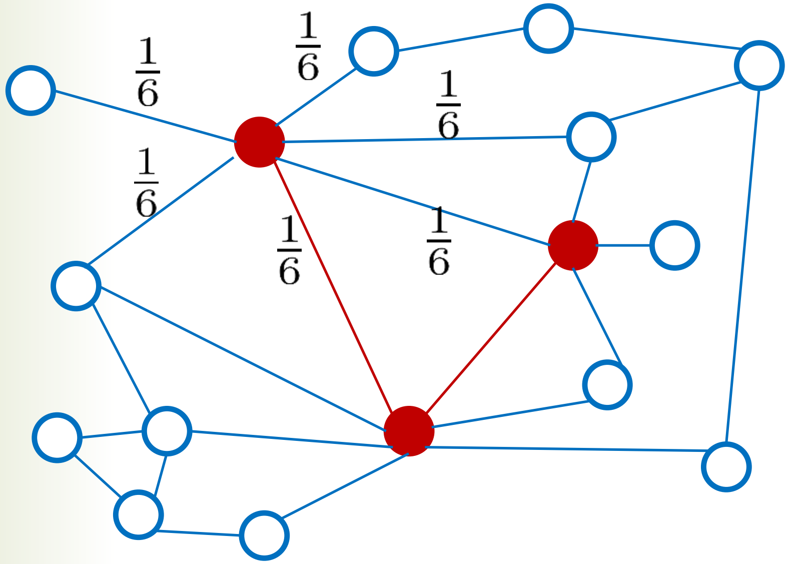
- Goal:

  Estimate $\mu(G) = \sum_{(u,v)\in E} g(u,v)$

- How [Ribeiro and Towsley `10]:

  Estimator for $\sum_{(u,v)\in E} g(u,v)$: $\frac{2|E|}{k}\sum_{i=1}^{k-1} g(X_i, X_{i+1})$

Random walk based estimation



Random walk $\{X_k\}_{k\geq 1}$ has unique stationary distribution $\{\pi_i\}_{i=1}^n$ if graph *G* is connected and non-bipartite

- Goal:

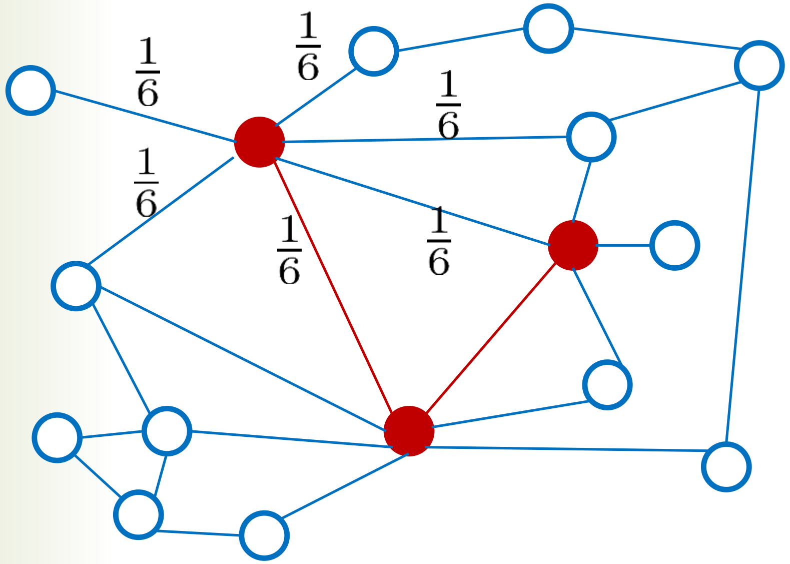
  Estimate $\mu(G) = \sum_{(u,v)\in E} g(u,v)$

- How [Ribeiro and Towsley `10]:

  Estimator for $\sum_{(u,v)\in E} g(u,v) : \frac{2|E|}{k}\sum_{i=1}^{k-1} g(X_i, X_{i+1})$

Asymptotically converges

Random walk based estimation



Random walk $\{X_k\}_{k\geq 1}$ has unique stationary distribution $\{\pi_i\}_{i=1}^n$ if graph G is connected and non-bipartite

- Goal:

  Estimate $\mu(G) = \sum_{(u,v)\in E} g(u,v)$

- How [Ribeiro and Towsley `10]:

  Estimator for $\sum_{(u,v)\in E} g(u,v)$: $\frac{2|E|}{k}\sum_{i=1}^{k-1} g(X_i, X_{i+1})$

Asymptotically converges

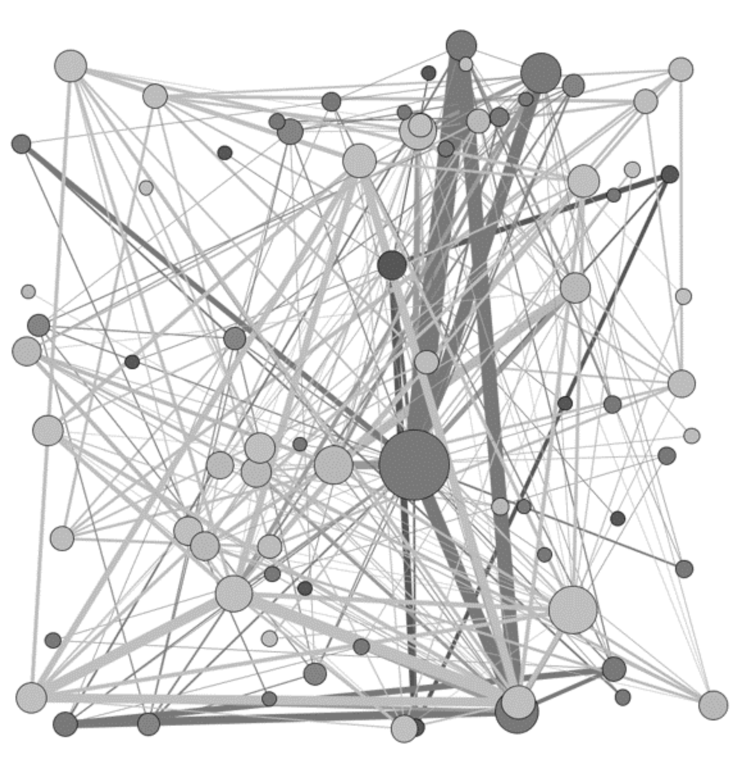
Extensions: [Lee et al. `12], [Gjoka et al. `11] [Ribeiro et al. `12]

We get an estimate of $\mu(G)$ but how accurate is it ?

Existing asymptotic techniques and issues

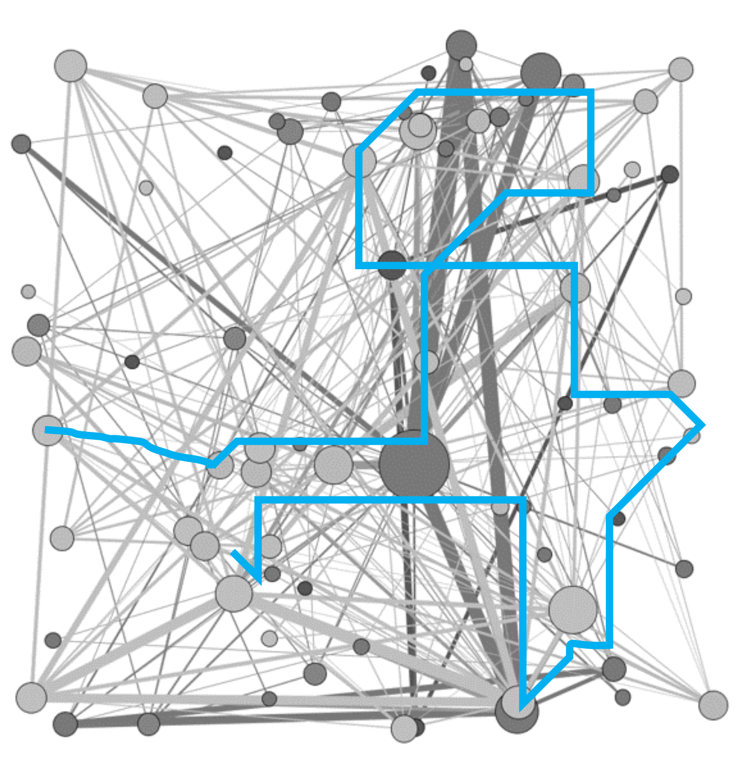
Existing asymptotic techniques and issues

- Asymptotic convergence: Ergodic theorem
  - Crawling the graph multiple times



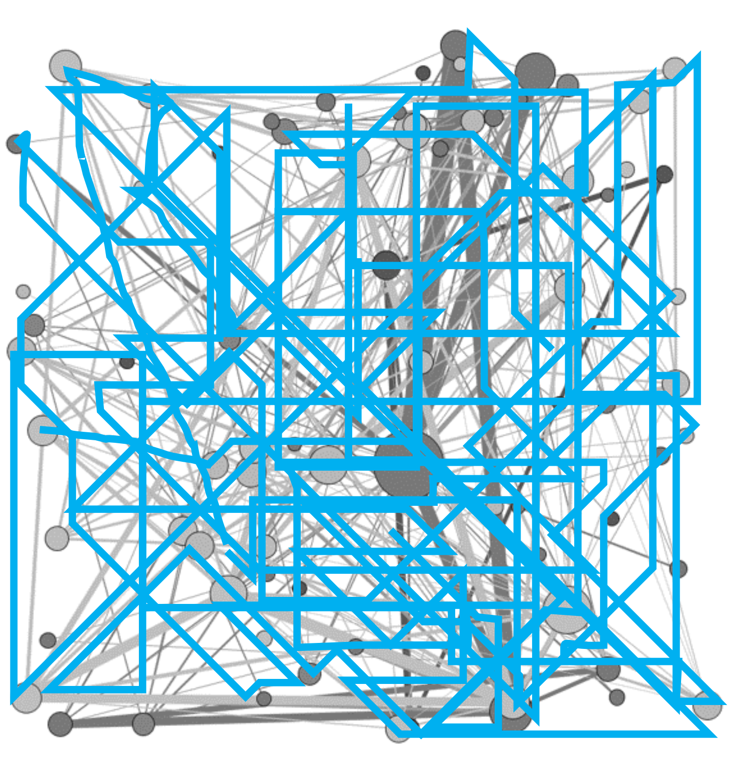
Existing asymptotic techniques and issues

- Asymptotic convergence: Ergodic theorem
  - Crawling the graph multiple times



Existing asymptotic techniques and issues

- Asymptotic convergence: Ergodic theorem
  - Crawling the graph multiple times



Existing asymptotic techniques and issues

- Asymptotic convergence: Ergodic theorem
  - Crawling the graph multiple times
- Variety of convergence diagnostics for MCMCs

Existing asymptotic techniques and issues

- Asymptotic convergence: Ergodic theorem
  - Crawling the graph multiple times
- Variety of convergence diagnostics for MCMCs

Roughly divided into:

Existing asymptotic techniques and issues

- Asymptotic convergence: Ergodic theorem
  - Crawling the graph multiple times
- Variety of convergence diagnostics for MCMCs

Roughly divided into:

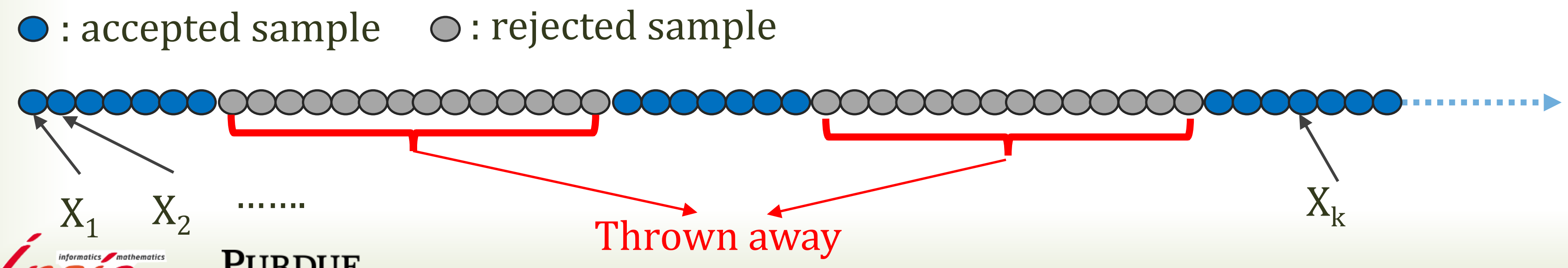
- Multiple walks to check convergence
  - Walks not independent (start at same seeds)
  - No guarantees

Existing asymptotic techniques and issues

- Asymptotic convergence: Ergodic theorem
  - Crawling the graph multiple times
- Variety of convergence diagnostics for MCMCs

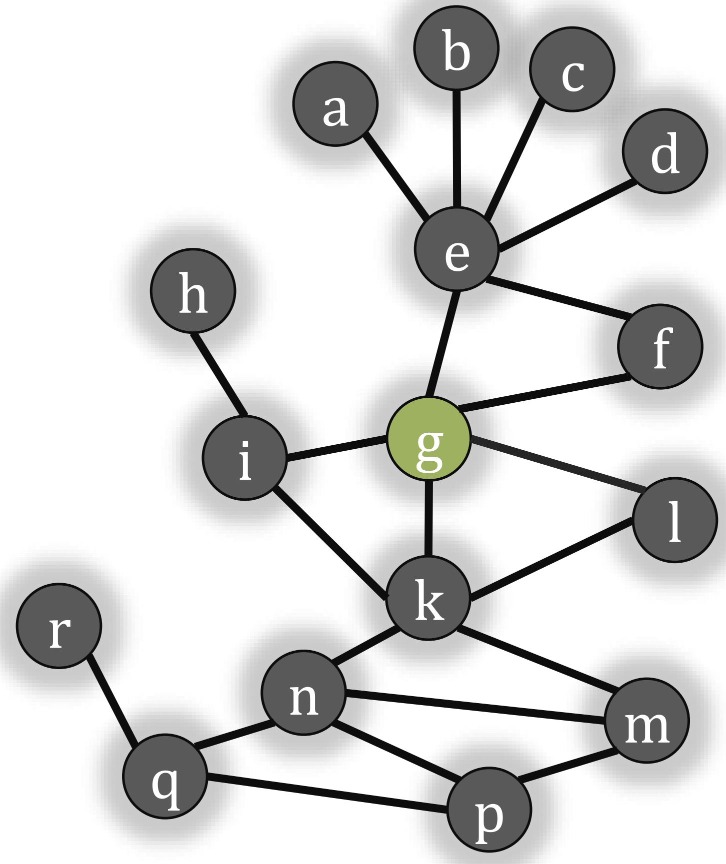
Roughly divided into:

- Multiple walks to check convergence
  - Walks not independent (start at same seeds)
  - No guarantees
- Break a long walk into “nearly” independent segments
  - Asymptotic & throws away most observations

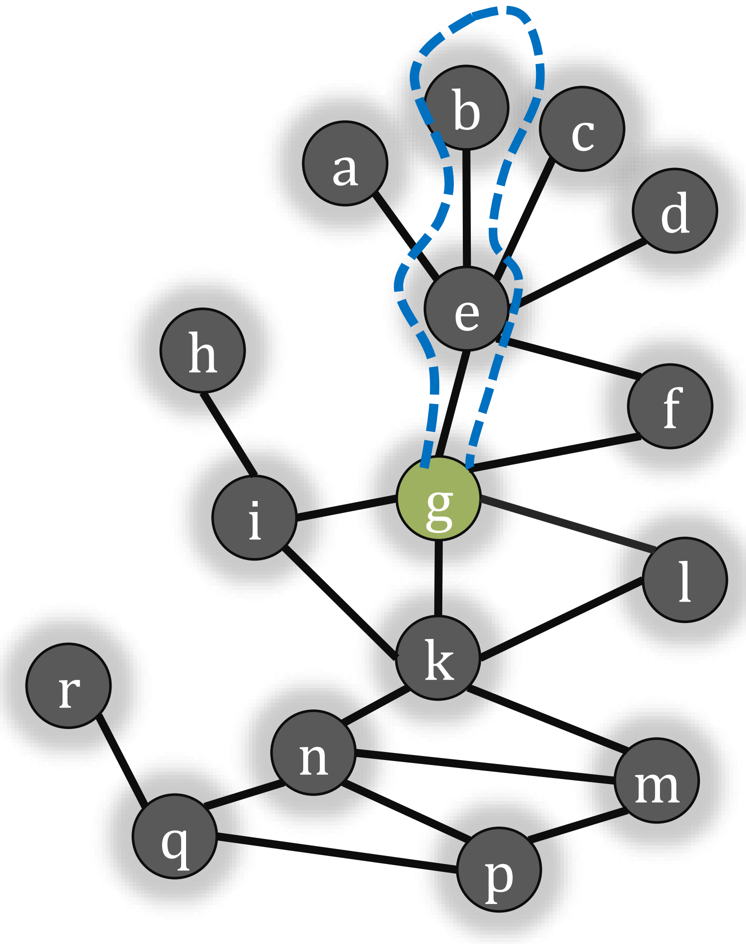


Idea of tours

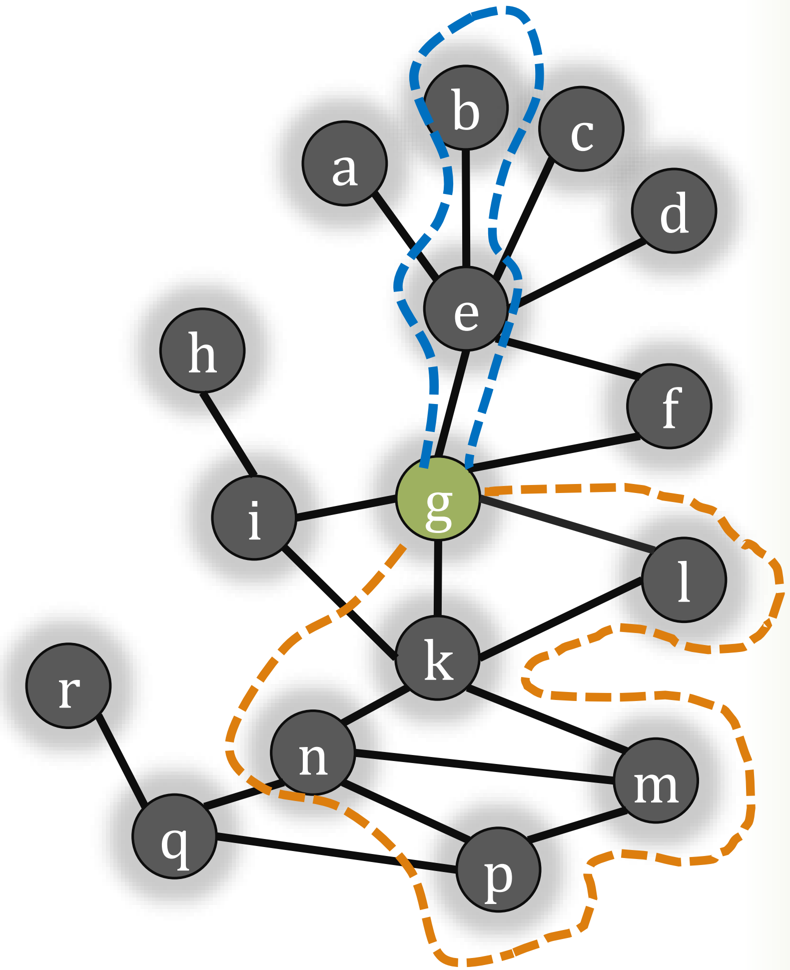
Idea of tours



Idea of tours

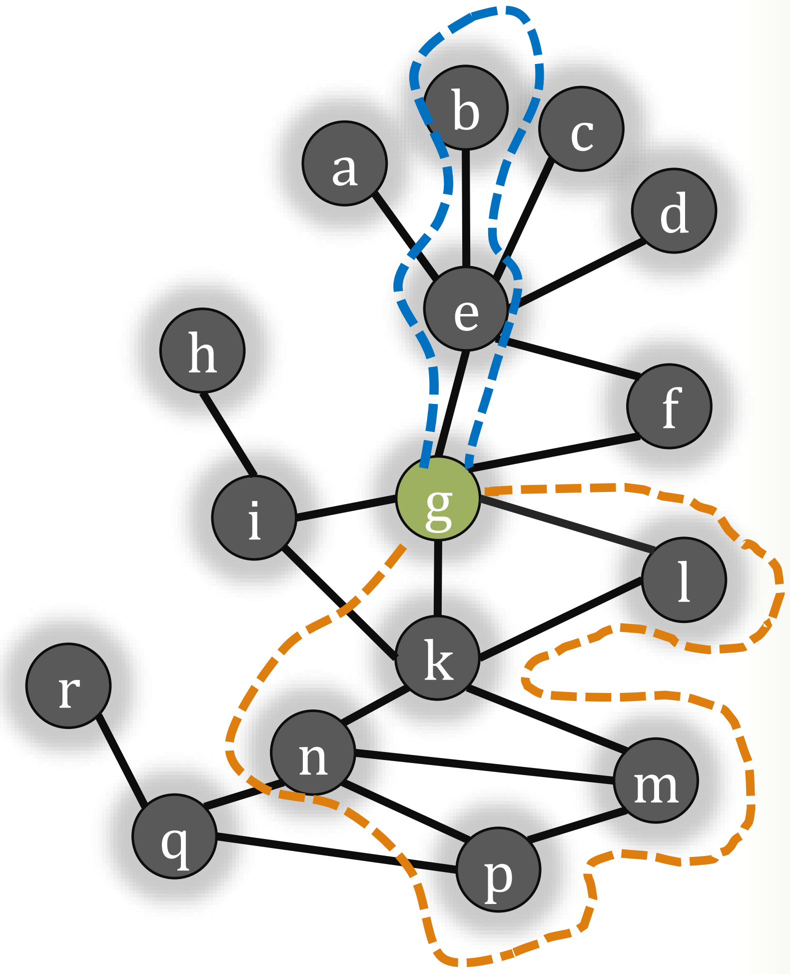


Idea of tours



Idea of tours

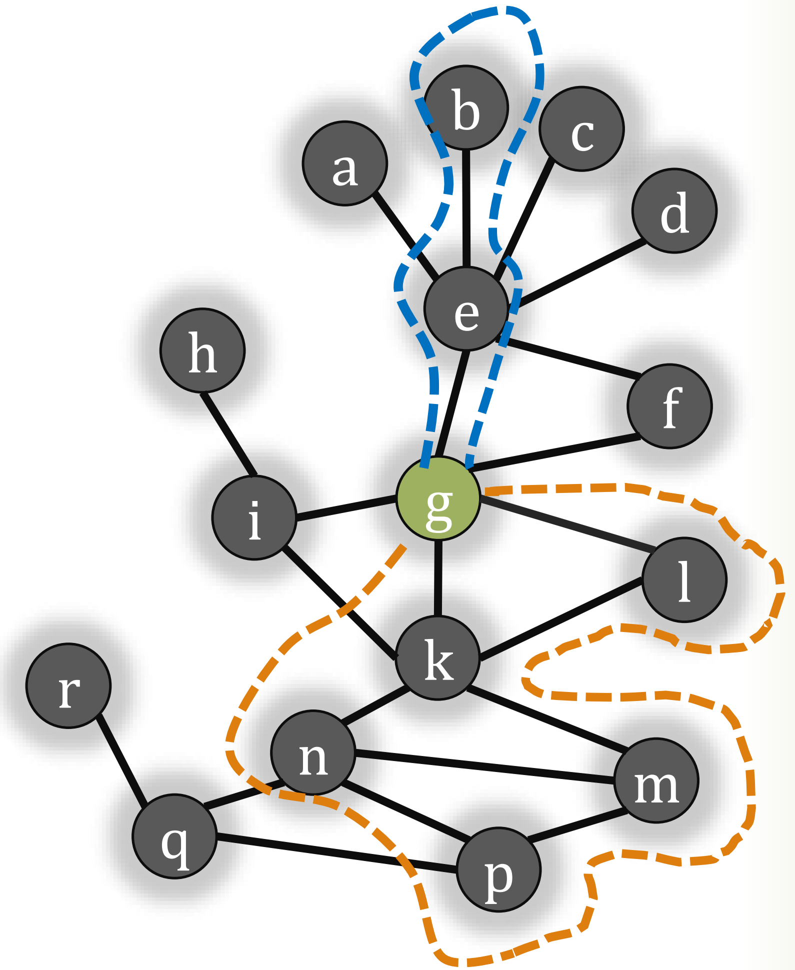
Properties of tours:



Idea of tours

Properties of tours:

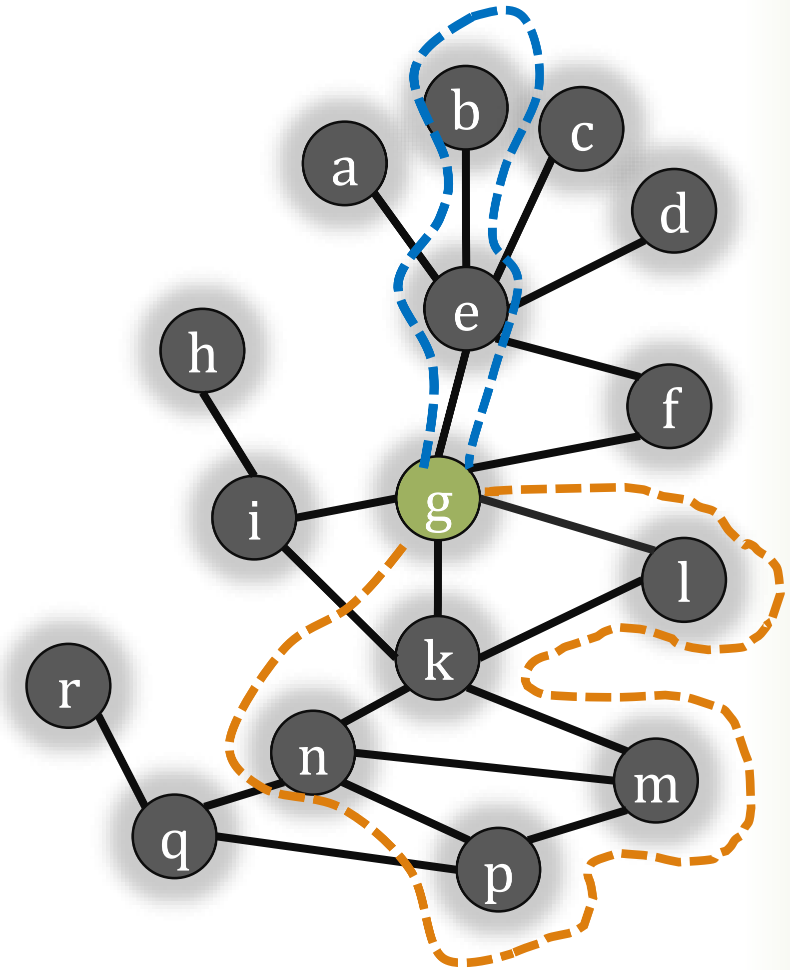
- Tours are independent



Idea of tours

Properties of tours:

- Tours are independent
- Fully distributed crawler implementation

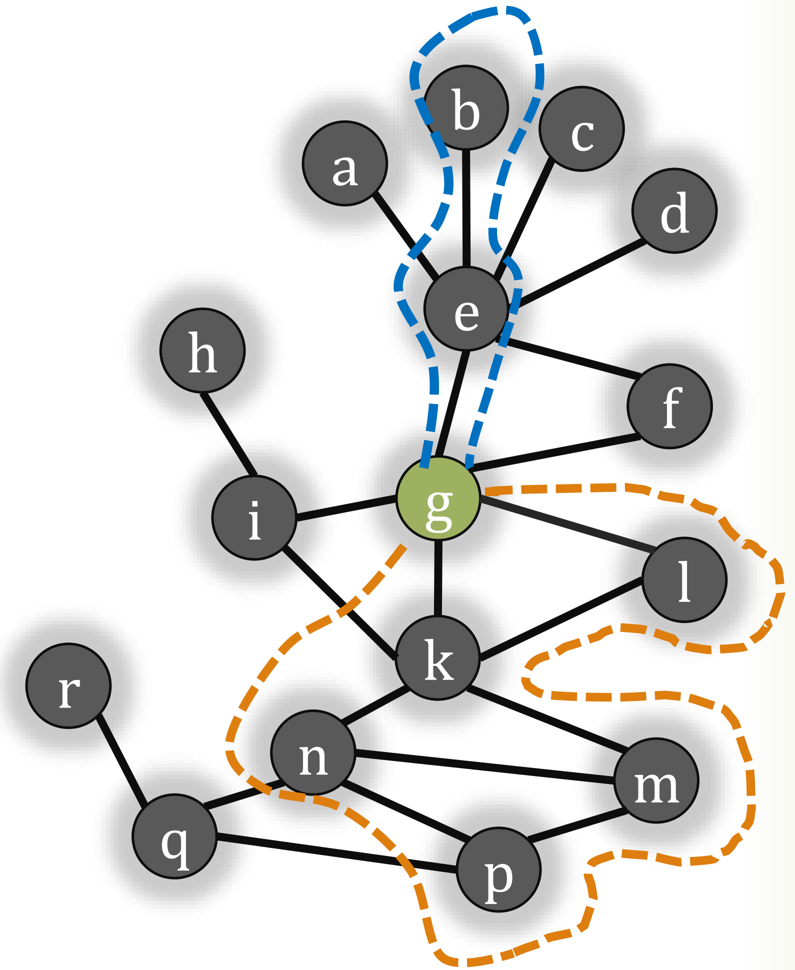


Idea of tours

Properties of tours:

- Tours are independent
- Fully distributed crawler implementation

Issues with tours:



Idea of tours

Properties of tours:

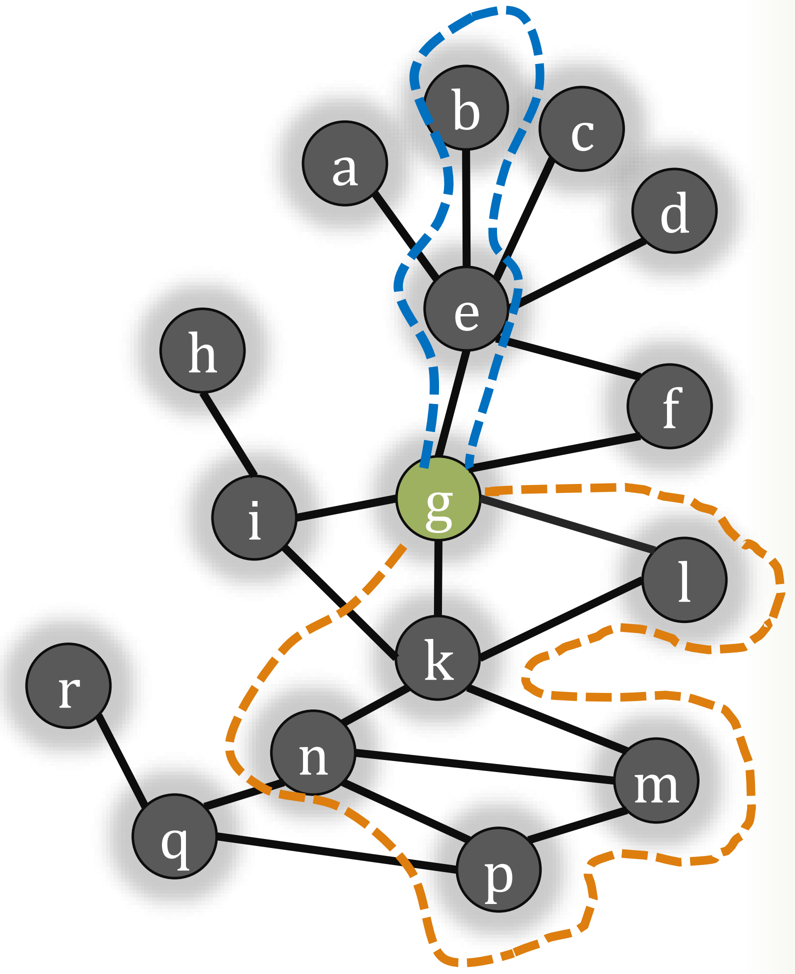
- Tours are independent
- Fully distributed crawler implementation

Issues with tours:

- Returning to same node will take “forever” in a large network [Massoulié et al’06]

$$\mathbb{E}[\text{Tour length}] = \frac{\text{vol}(G)}{\text{degree(g)}}$$

2|E|



Idea of tours

Properties of tours:

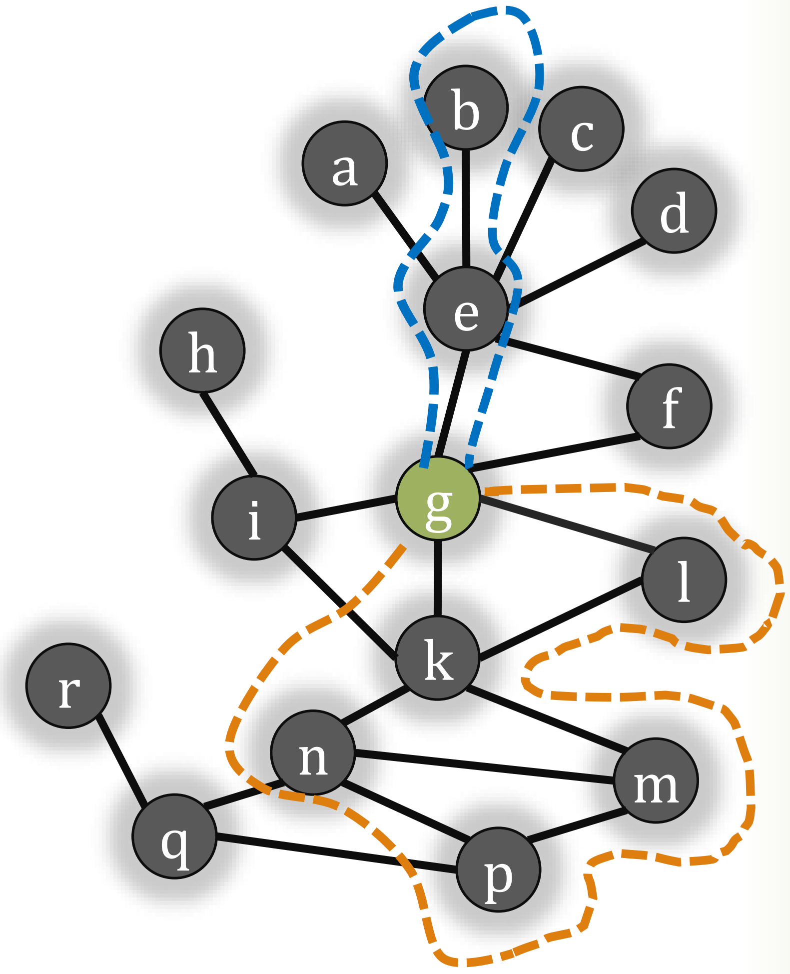
- Tours are independent
- Fully distributed crawler implementation

Issues with tours:

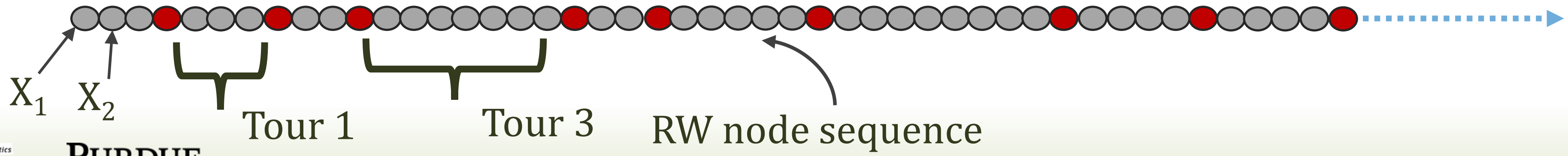
- Returning to same node will take "forever" in a large network [Massoulié et al'06]

$$\mathbb{E}[\text{Tour length}] = \frac{\overbrace{\mathrm{vol}(G)}^{2|E|}}{\mathrm{degree(g)}}$$

- Solution? Renewal from the most frequent node.



: most frequent node in sequence



Inria PURDUE UNIVERSITY

Idea of tours

Properties of tours:

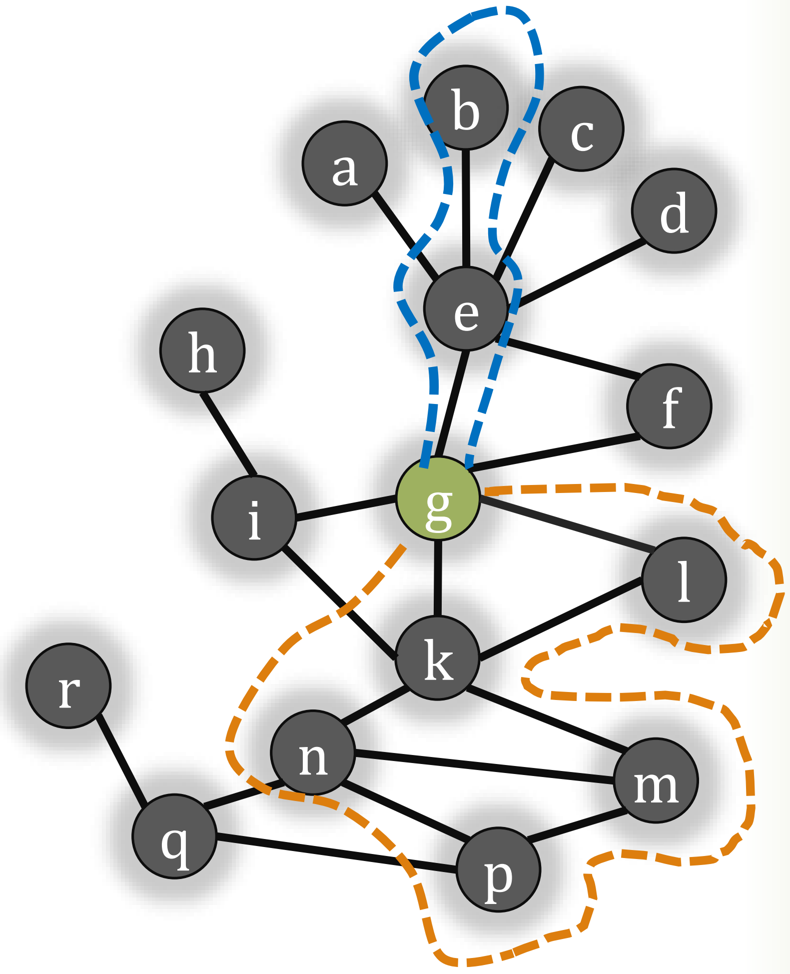
- Tours are independent
- Fully distributed crawler implementation

Issues with tours:

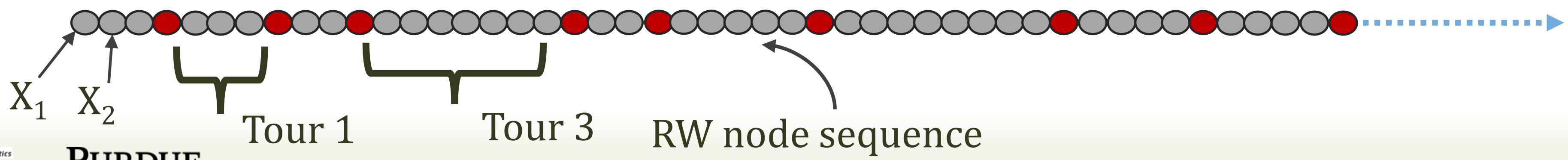
- Returning to same node will take "forever" in a large network [Massoulié et al'06]

$$\mathbb{E}[\text{Tour length}] = \frac{\overbrace{\mathrm{vol}(G)}^{2|E|}}{\text{degree(g)}}$$

- Solution? Renewal from the most frequent node.
  - No, tours will be interdependent

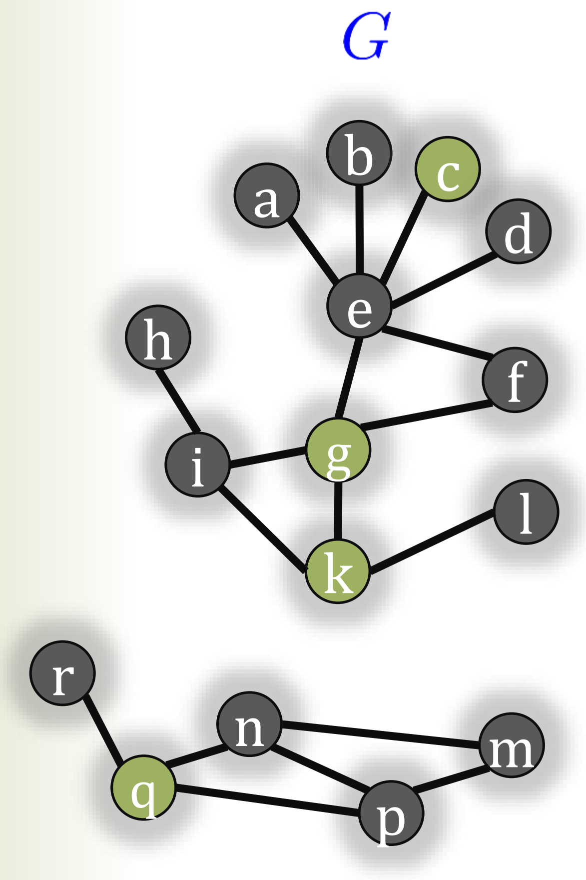


● : most frequent node in sequence

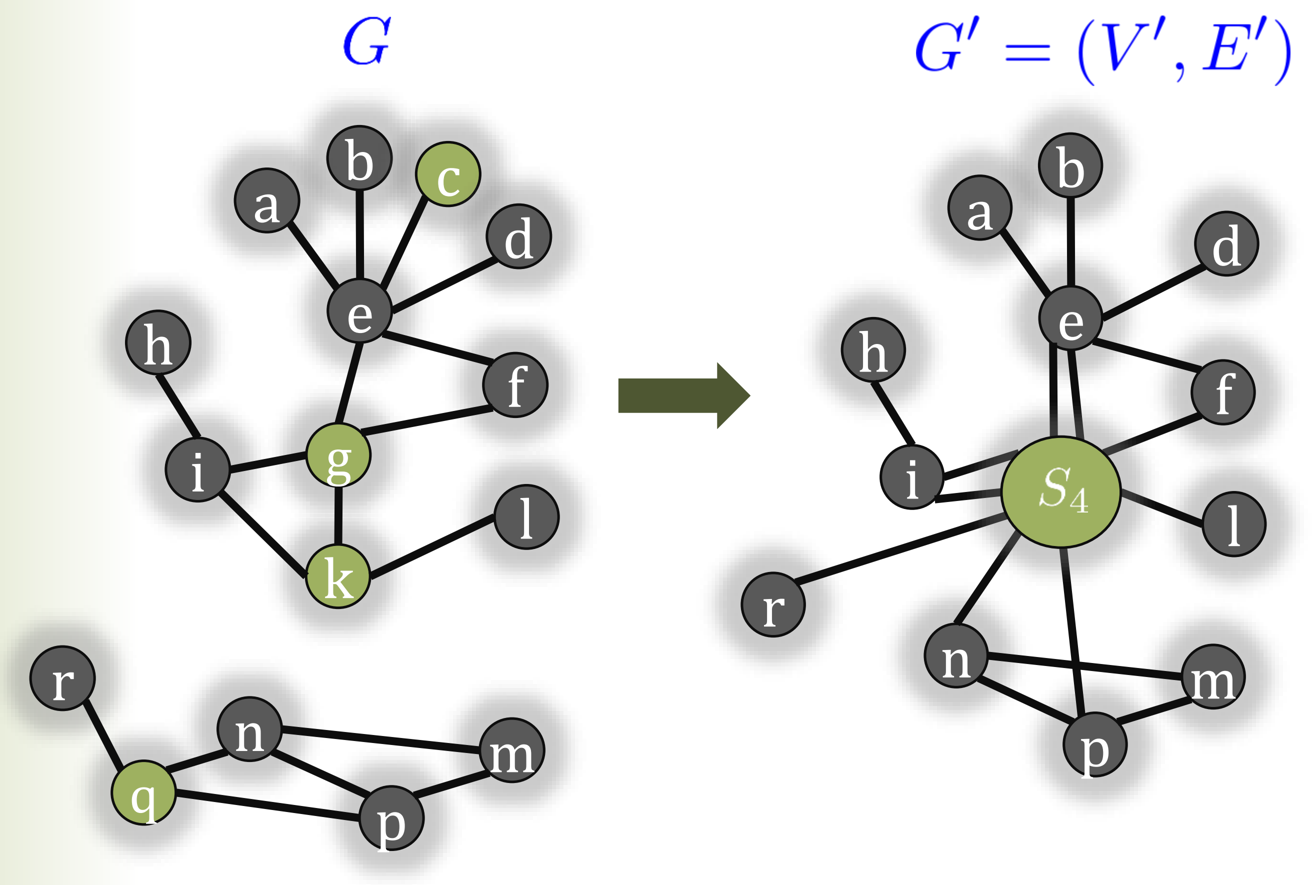


The idea of Super-node

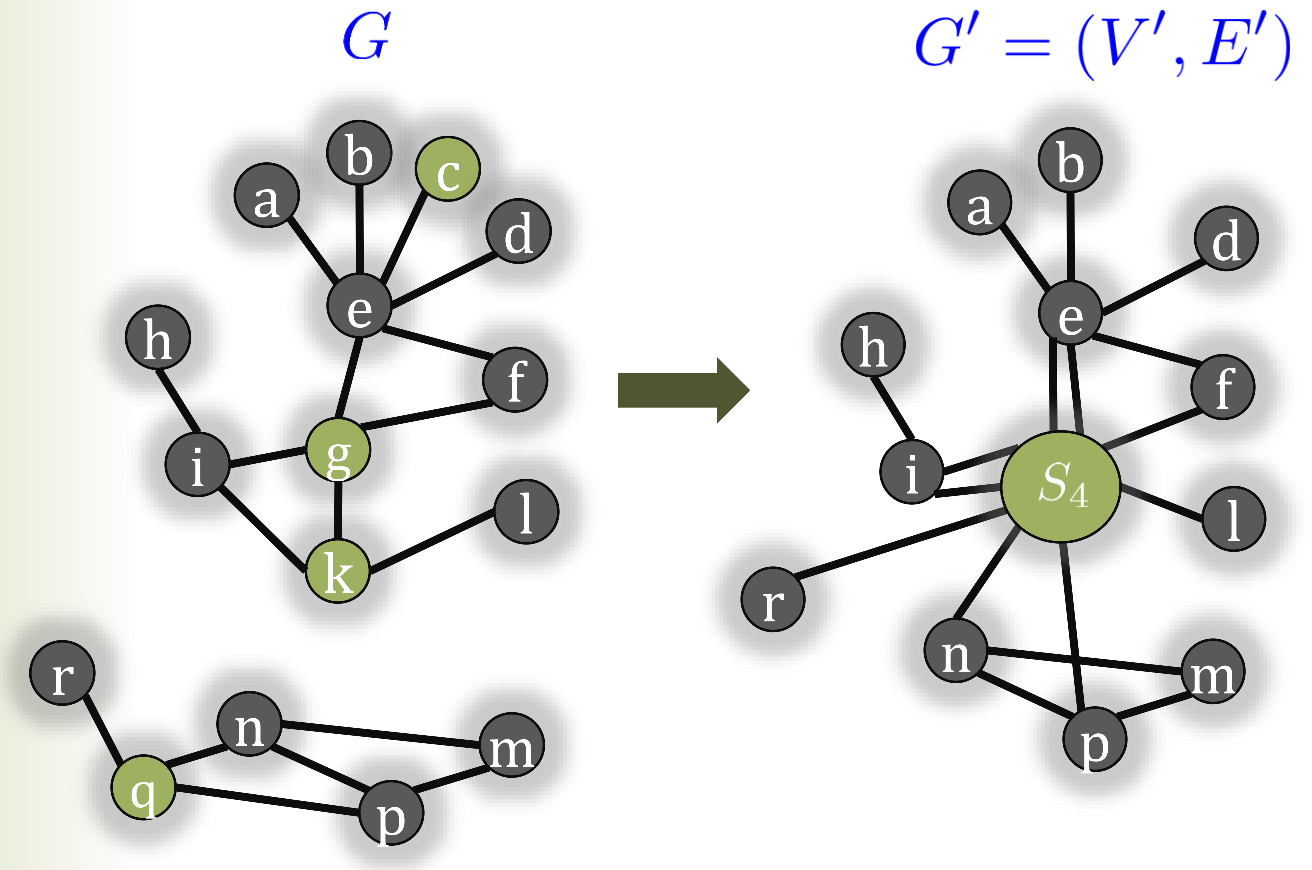
The idea of Super-node



The idea of Super-node

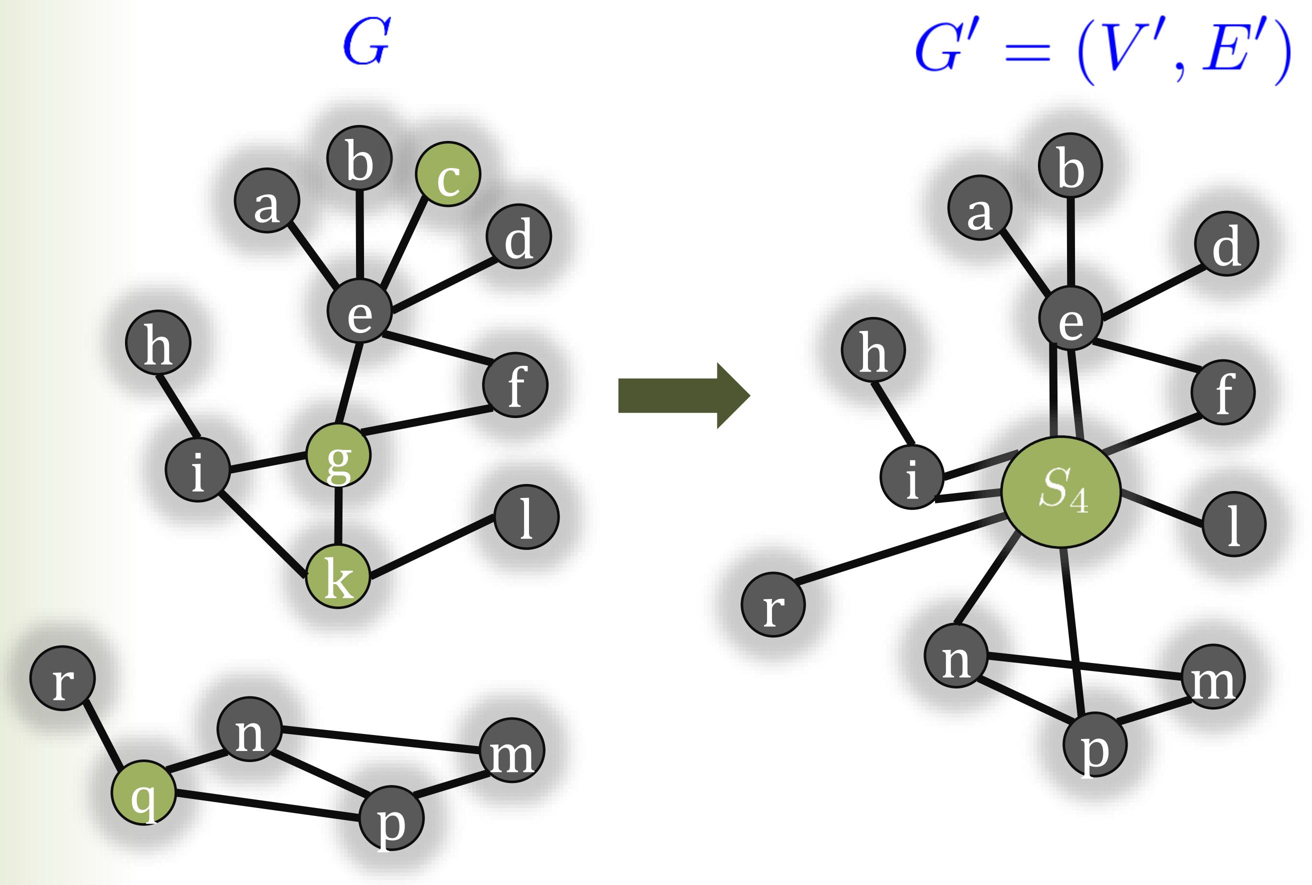


The idea of Super-node



- Tackling disconnected graph

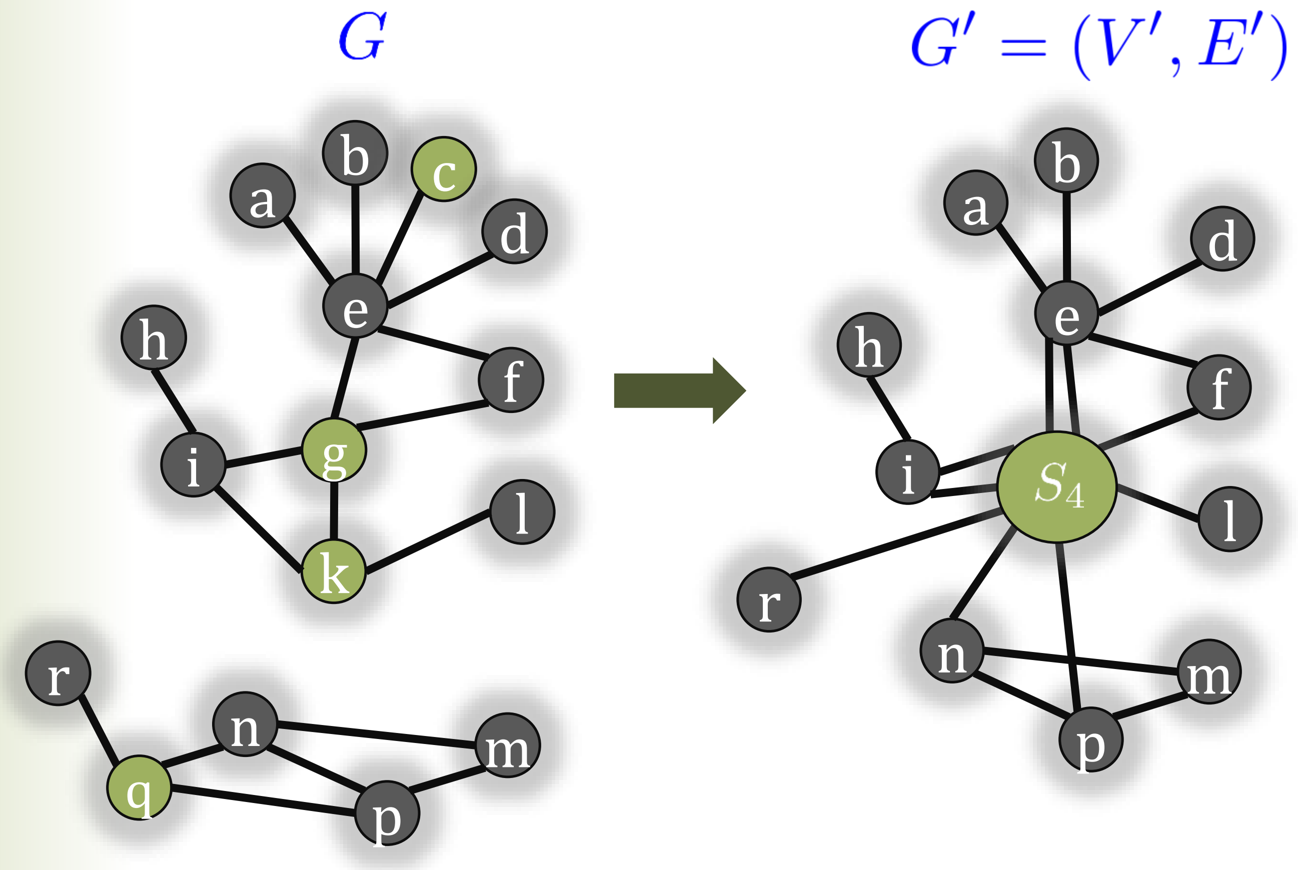
The idea of Super-node



- Tackling disconnected graph
- Faster estimate with shorter crawls

$$\mathbb{E}[\text{Tour length}] = \frac{\text{vol}(G)}{\text{degree}(S_4)}$$

The idea of Super-node

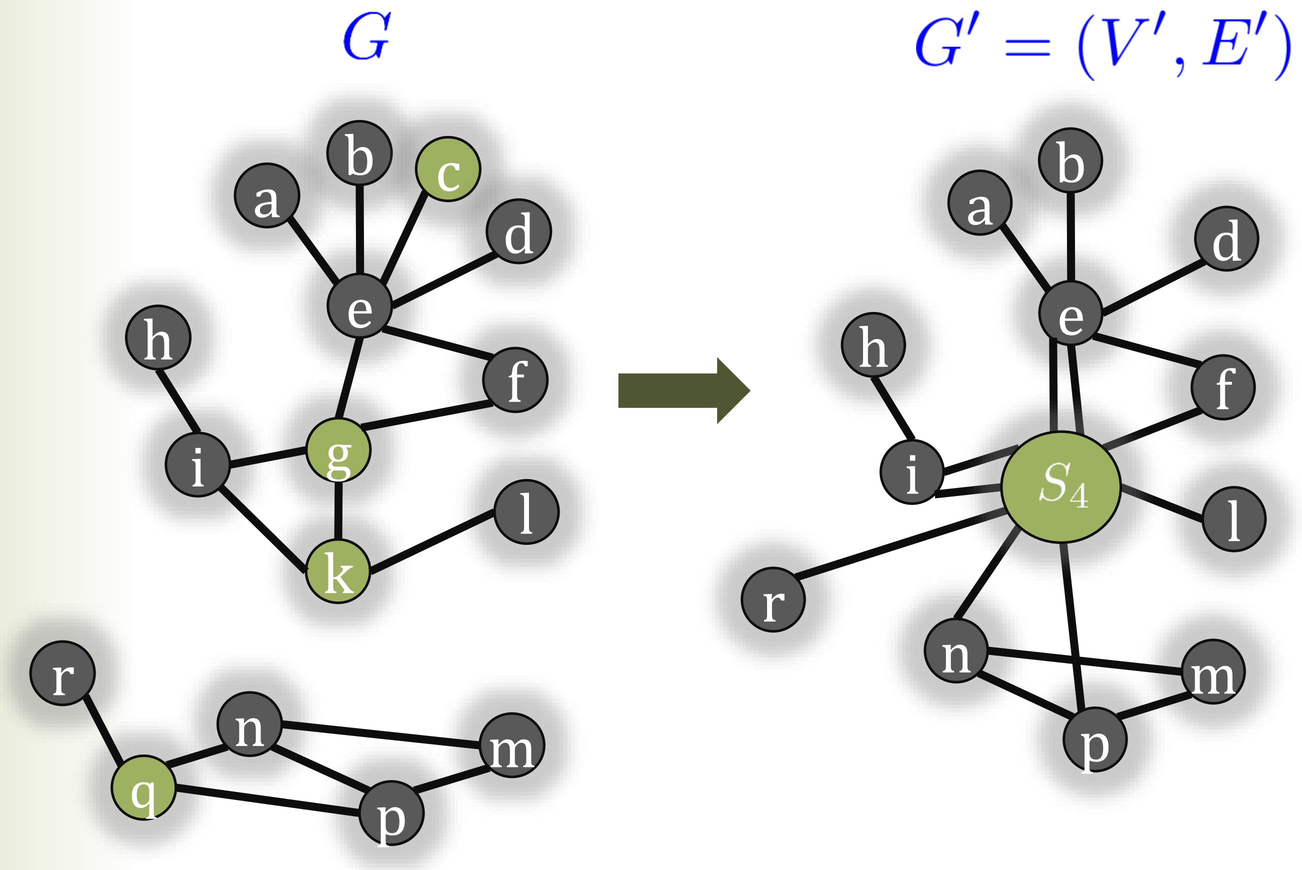


- Tackling disconnected graph
- Faster estimate with shorter crawls

$$\mathbb{E}[\text{Tour length}] = \frac{\text{vol}(G)}{\text{degree}(S_4)}$$

- Not related to *lumpability*

The idea of Super-node



- Tackling disconnected graph
- Faster estimate with shorter crawls

$$\mathbb{E}[\text{Tour length}] = \frac{\text{vol}(G)}{\text{degree}(S_4)}$$

- Not related to *lumpability*

Super-node formation:

- static and dynamic (will see later)

Estimator

Estimator

Key property of tours:

Estimator

Length of kth tour

True value of the contracted graph

Key property of tours: $$\mathbb{E}\Big[\sum_{t=2}^{\xi_k} f(X_{t-1}^{(k)}, X_t^{(k)})\Big] = \frac{2}{d_{S_n}}\mu(G')$$

$f(u,v) := g(u,v)$
except when u or v is S_n

Samples in kth tour

Degree of super-node

Estimator

Key property of tours: $\mathbb{E}\Big[\sum_{t=2}^{\xi_k} f(X_{t-1}^{(k)}, X_t^{(k)})\Big] = \frac{2}{d_{S_n}}\mu(G')$

Length of kth tour

True value of the contracted graph

$f(u,v) := g(u,v)$
except when u or v is S_n

Samples in kth tour

Degree of super-node

$$\hat{\mu}(G) = \overbrace{\frac{d_{S_n}}{2m}\sum_{k=1}^{m}\sum_{t=2}^{\xi_k} f(X_{t-1}^{(k)}, X_t^{(k)})}^{\text{Estimate from crawls}}$$

Estimator

Length of kth tour

True value of the contracted graph

Key property of tours: $\mathbb{E}\left[\sum_{t=2}^{\xi_k} f(X_{t-1}^{(k)}, X_t^{(k)})\right] = \frac{2}{d_{S_n}}\mu(G')$

$f(u,v) := g(u,v)$
except when u or v is S_n

Samples in kth tour

Degree of super-node

$$\hat{\mu}(G) = \underbrace{\frac{d_{S_n}}{2m}\sum_{k=1}^{m}\sum_{t=2}^{\xi_k} f(X_{t-1}^{(k)}, X_t^{(k)})}_{\text{Estimate from crawls}} + \underbrace{\sum_{(u,v)\in H} g(u,v)}_{\text{Given knowledge from nodes in super-node}}$$

Estimator

Estimator

- Unbiased (unlike asymptotic in [Ribeiro and Towsley '10])

$$\mathbb{E}[\hat{\mu}(G)] = \mu(G)$$

Estimator

- Unbiased (unlike asymptotic in [Ribeiro and Towsley '10])

$$\mathbb{E}[\hat{\mu}(G)] = \mu(G)$$

- Strongly consistent

$$\hat{\mu}(G) \to \mu(G) \text{ a.s.}$$

Estimator

- Unbiased (unlike asymptotic in [Ribeiro and Towsley '10])

$$\mathbb{E}[\hat{\mu}(G)] = \mu(G)$$

- Strongly consistent

$$\hat{\mu}(G) \to \mu(G) \text{ a.s.}$$

Confidence interval

$$P\left(|\mu(G) - \hat{\mu}(G)| \leq \varepsilon\right) \approx 1 - 2\Phi\left(\frac{\varepsilon\sqrt{m}}{\hat{\sigma}_m}\right)$$

Sampled variance

Estimator

- Unbiased (unlike asymptotic in [Ribeiro and Towsley '10])

$$\mathbb{E}[\hat{\mu}(G)] = \mu(G)$$

- Strongly consistent

$$\hat{\mu}(G) \to \mu(G) \text{ a.s.}$$

Confidence interval

$$P\left(|\mu(G) - \hat{\mu}(G)| \leq \varepsilon\right) \approx 1 - 2\Phi\left(\frac{\varepsilon\sqrt{m}}{\hat{\sigma}_m}\right)$$

Sampled variance

$$\mathrm{Var}\left[\sum_{t=2}^{\xi_k} f(X_{t-1}^{(k)}, X_t^{(k)})\right] \leq B^2\left(\frac{2\mathrm{vol}(G)}{d_{S_n}^2 \delta'} + 1\right)$$

$$\delta' := \text{spectral gap of new graph}$$

$$\max_{(i,j)\in E'} f(i,j) \leq B < \infty$$

Bayesian formulation

Find a posterior probability distribution

$$\mathbb{P}(\mu(G) < x | \{m \text{ tours}\})$$

with suitable prior distribution

Details in the paper

Bayesian formulation (contd.)

Details in the paper

Bayesian formulation (contd.)

$$\hat{F}_h = \frac{d_{S_n}}{2\lfloor\sqrt{m}\rfloor} \sum_{k=((h-1)\lfloor\sqrt{m}\rfloor+1)}^{h\lfloor\sqrt{m}\rfloor} \sum_{t=2}^{\xi_h} f(X_{t-1}^{(k)}, X_t^{(k)}) + \sum_{(u,v)\in H} g(u,v)\,, \quad \sigma^2 \triangleq \mathrm{Var}(\hat{F}_h)$$

Details in the paper

Bayesian formulation (contd.)

$$\hat{F}_h = \frac{d_{S_n}}{2\lfloor\sqrt{m}\rfloor} \sum_{k=((h-1)\lfloor\sqrt{m}\rfloor+1)}^{h\lfloor\sqrt{m}\rfloor} \sum_{t=2}^{\xi_h} f(X_{t-1}^{(k)}, X_t^{(k)}) + \sum_{(u,v)\in H} g(u,v)\,, \quad \sigma^2 \triangleq \mathrm{Var}(\hat{F}_h)$$

Assumption: $\hat{F}_h \sim \mathrm{Normal}(\mu(G), \sigma^2)$

(also justifiable via exponentially bounded tour lengths [Aldous anf Fill '02])

Details in the paper

Bayesian formulation (contd.)

$$\hat{F}_h = \frac{d_{S_n}}{2\lfloor\sqrt{m}\rfloor} \sum_{k=((h-1)\lfloor\sqrt{m}\rfloor+1)}^{h\lfloor\sqrt{m}\rfloor} \sum_{t=2}^{\xi_h} f(X_{t-1}^{(k)}, X_t^{(k)}) + \sum_{(u,v)\in H} g(u,v)\,, \quad \sigma^2 \triangleq \mathrm{Var}(\hat{F}_h)$$

Assumption: $\hat{F}_h \sim \text{Normal}(\mu(G), \sigma^2)$
(also justifiable via exponentially bounded tour lengths [Aldous anf Fill '02])

For $m \geq 2$ tours and assuming priors $\mu(G)|\sigma^2 \sim \text{Normal}(\mu_0, \sigma^2/m_0)$, $\sigma^2 \sim$ Inverse-gamma$(\nu_0/2, \nu_0\sigma_0^2/2)$, then for large values of m,

$$\mathbb{P}(\mu(G) \leq x | \{m \text{ tours}\}) \approx \phi_{\substack{\text{student-t} \\ (\nu,\tilde{\mu},\tilde{\sigma})}}(x)$$

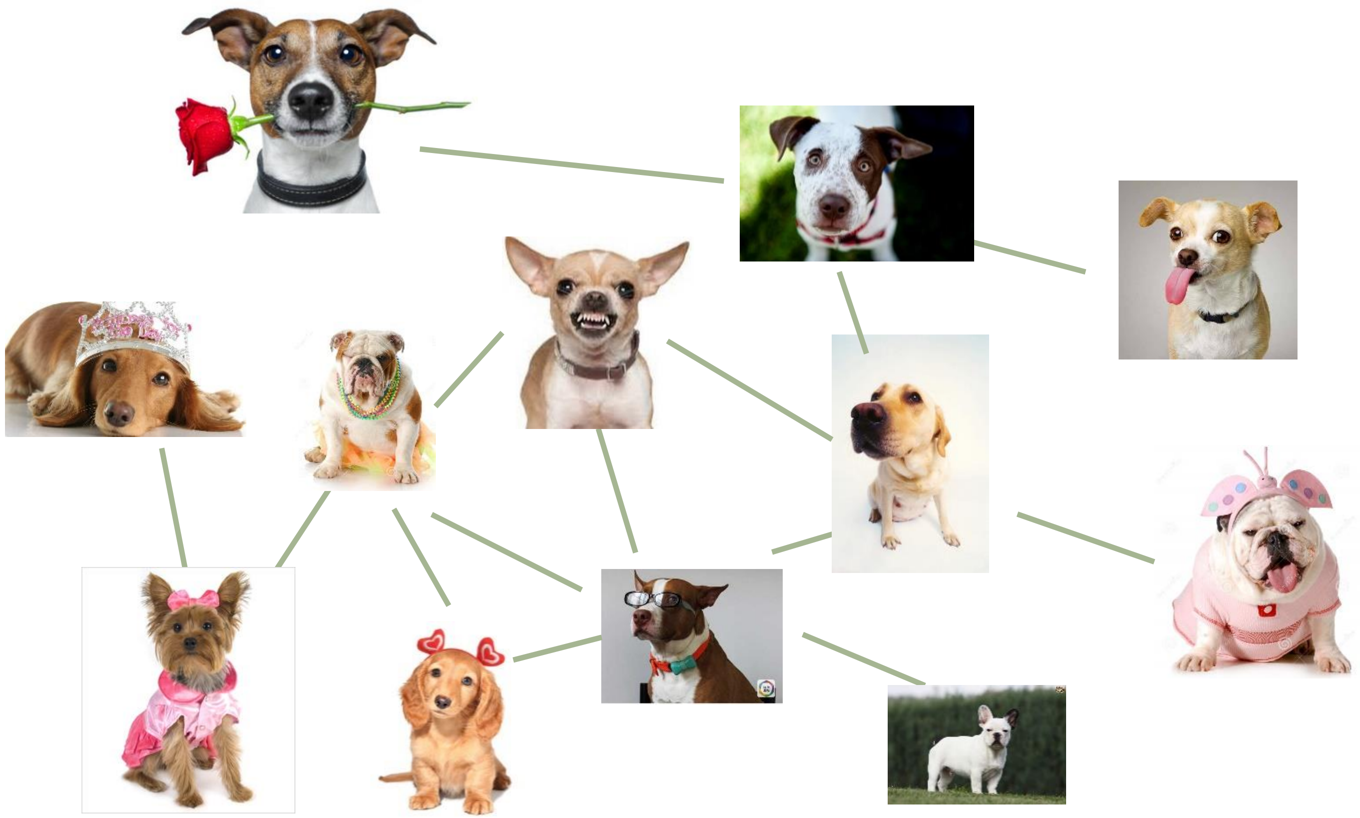
$$\nu = \nu_0 + \lfloor\sqrt{m}\rfloor,$$

$$\tilde{\mu} = \frac{m_0\mu_0 + \lfloor\sqrt{m}\rfloor\hat{\mu}(G))}{m_0 + \lfloor\sqrt{m}\rfloor}, \tilde{\sigma}^2 = \frac{\nu_0\sigma_0^2 + \sum_{k=1}^{\lfloor\sqrt{m}\rfloor}(\hat{F}_k - \hat{\mu}(G))^2 + \frac{m_0\lfloor\sqrt{m}\rfloor(\hat{\mu}(G)) - \mu_0)^2}{m_0 + \lfloor\sqrt{m}\rfloor}}{(\nu_0 + \lfloor\sqrt{m}\rfloor)(m_0 + \lfloor\sqrt{m}\rfloor)}$$

Simulations on real-world networks

Simulations on real-world networks

Dogster network: Online social network for dogs ?



Simulations on real-world networks: Dogster network

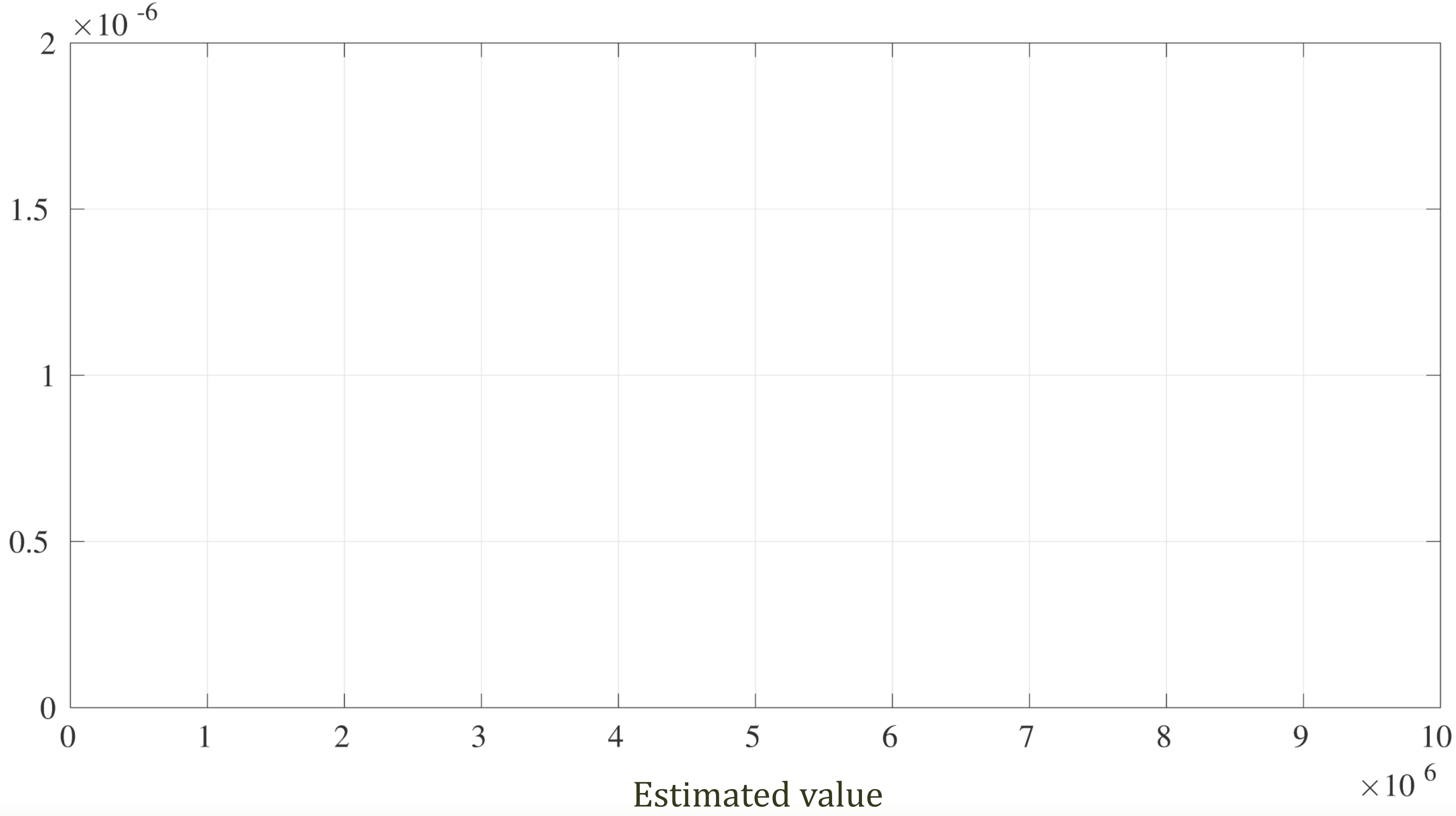
415K nodes, 8.27M edges

Percentage of graph covered: 2.72% (edges), 14.86% (nodes)

Simulations on real-world networks: Dogster network

415K nodes, 8.27M edges

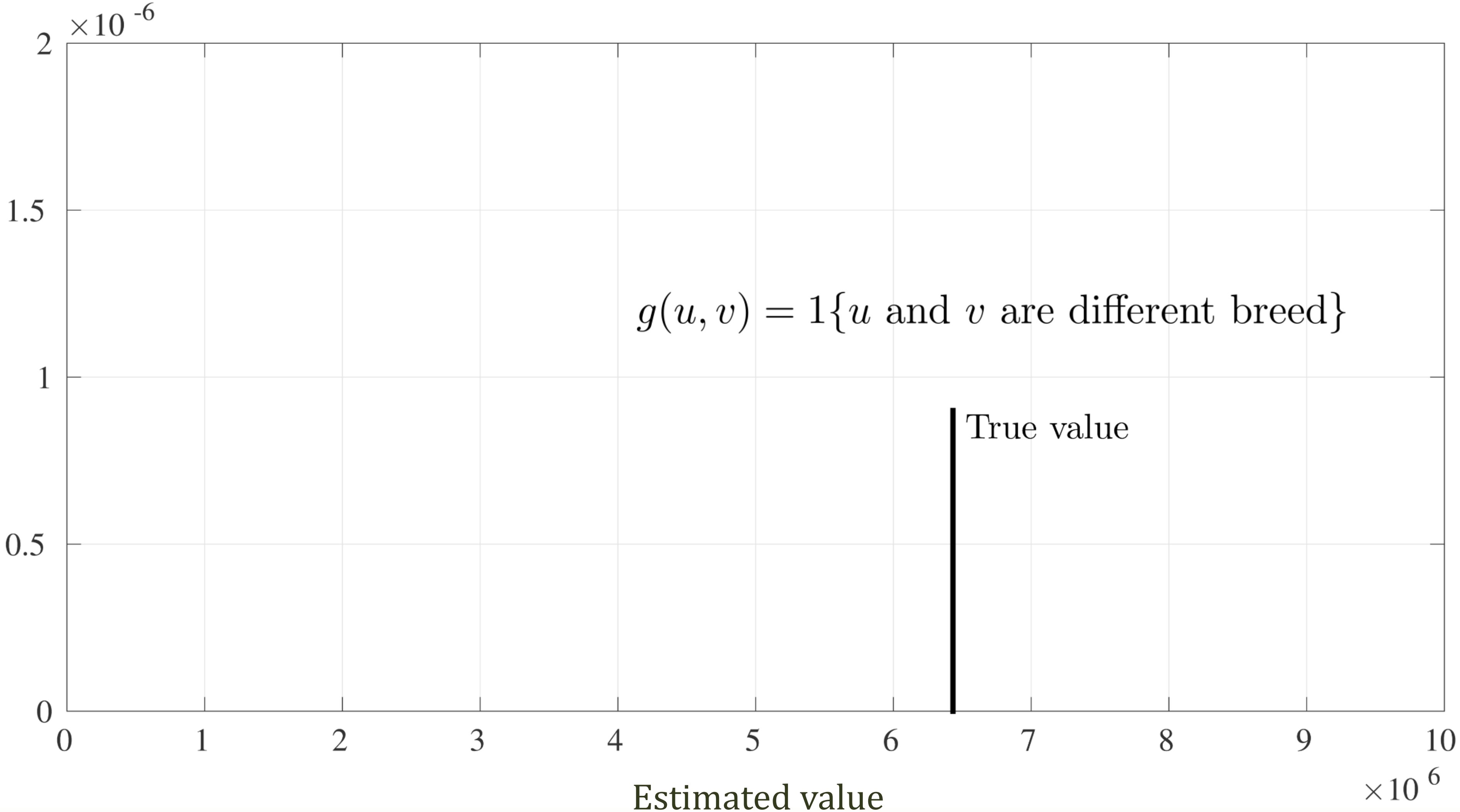
Percentage of graph covered: 2.72% (edges), 14.86% (nodes)



Simulations on real-world networks: Dogster network

415K nodes, 8.27M edges

Percentage of graph covered: 2.72% (edges), 14.86% (nodes)

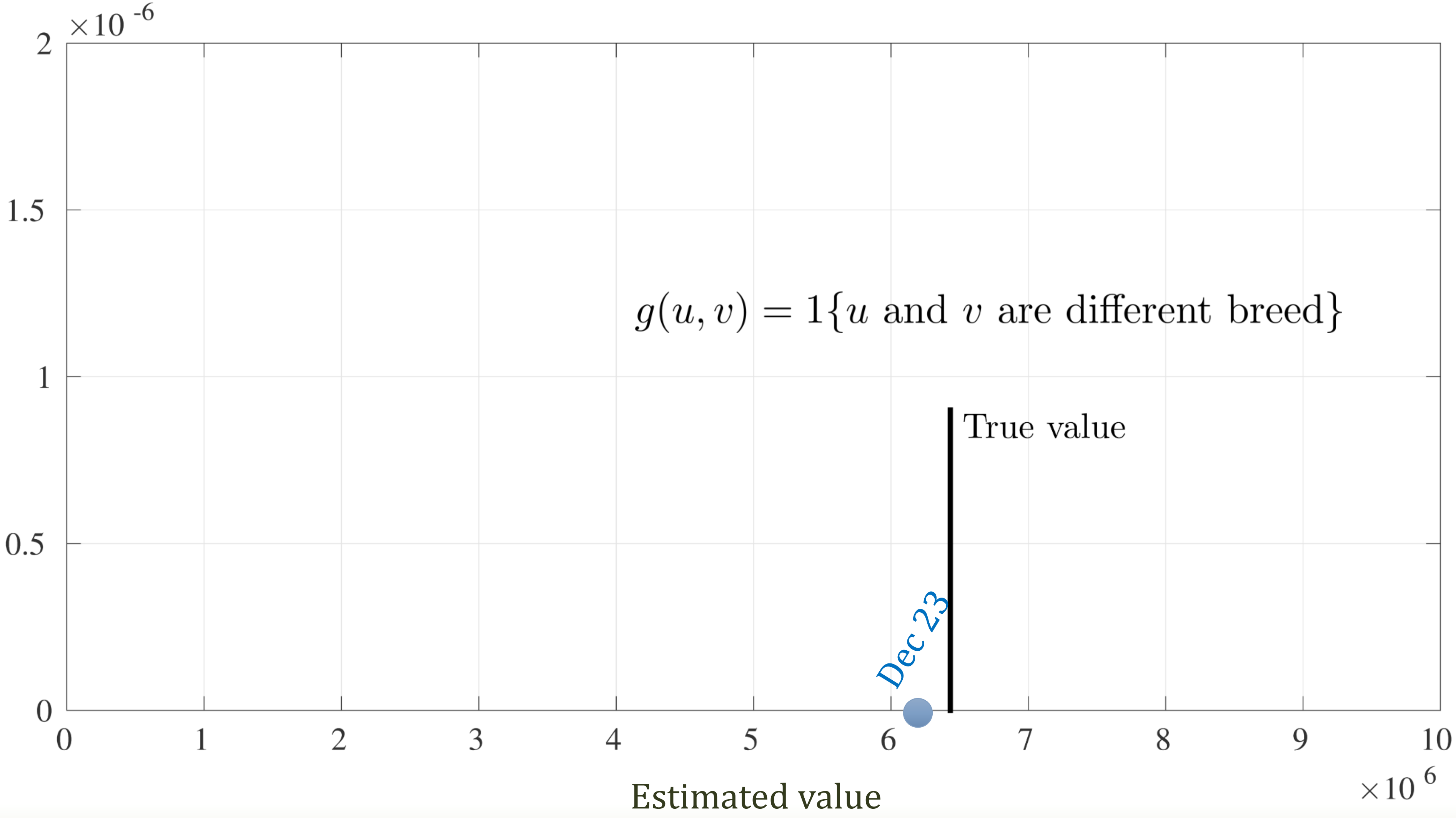


Estimated value

Simulations on real-world networks: Dogster network

415K nodes, 8.27M edges

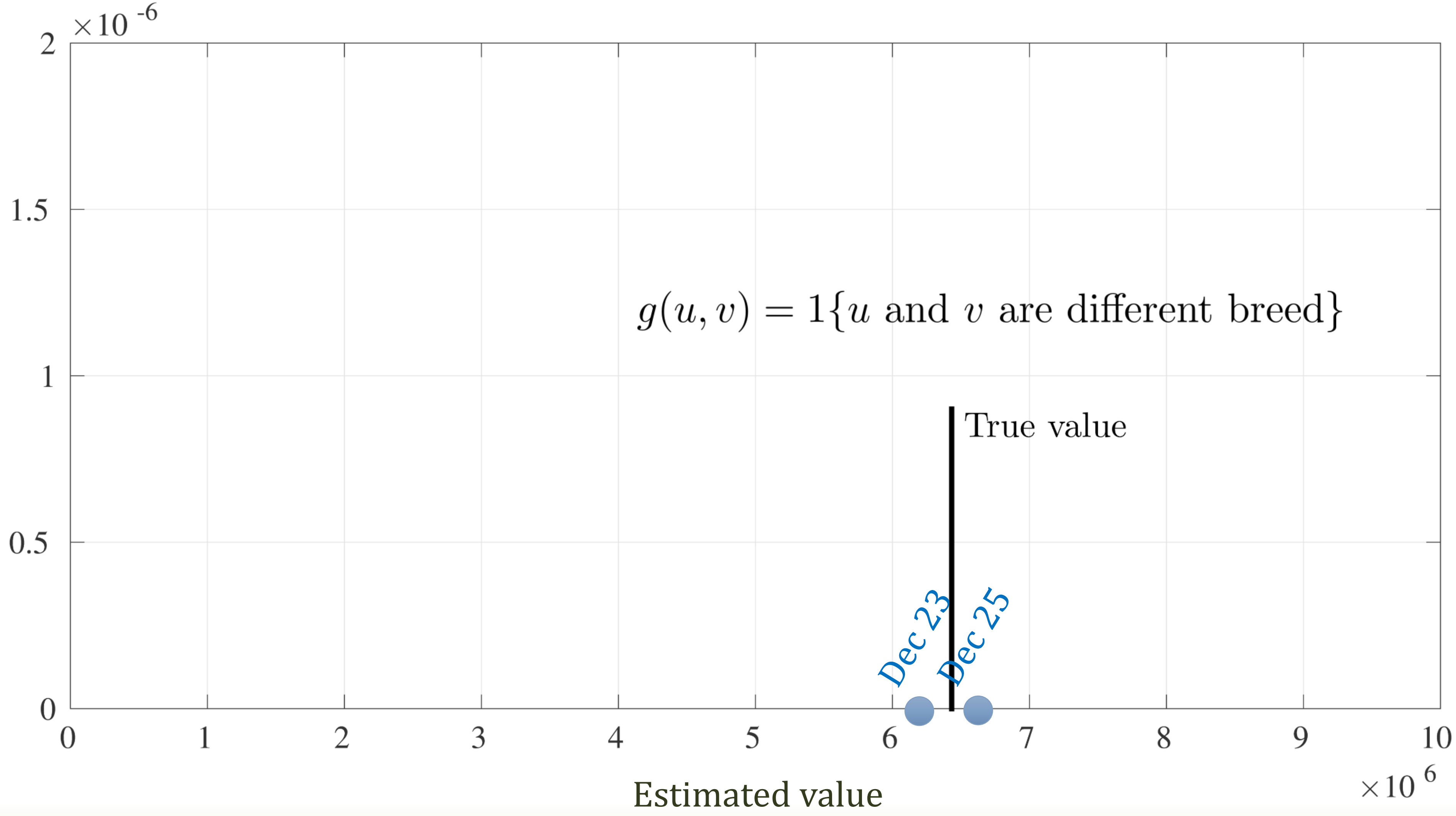
Percentage of graph covered: 2.72% (edges), 14.86% (nodes)



Simulations on real-world networks: Dogster network

415K nodes, 8.27M edges

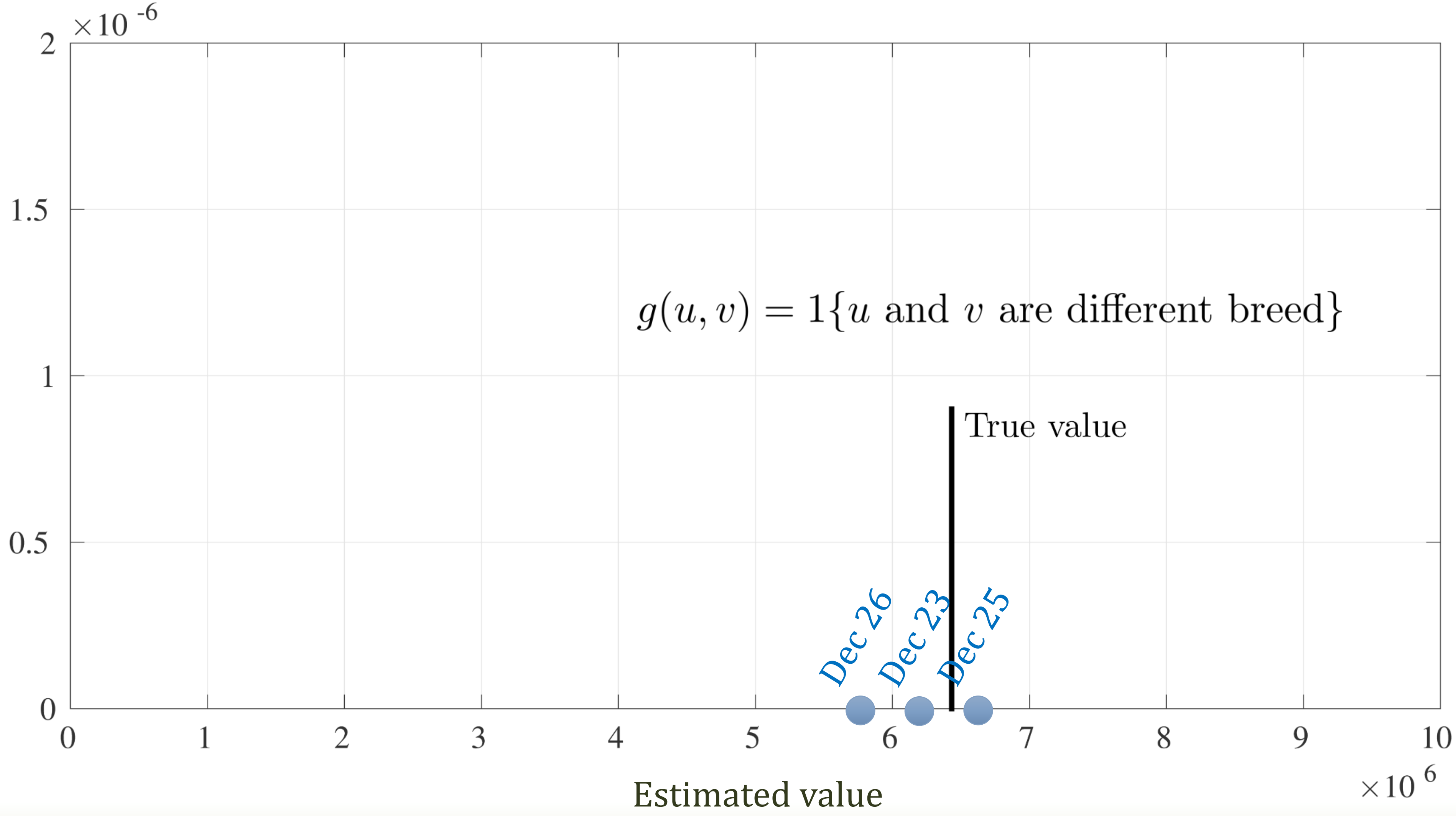
Percentage of graph covered: 2.72% (edges), 14.86% (nodes)



Simulations on real-world networks: Dogster network

415K nodes, 8.27M edges

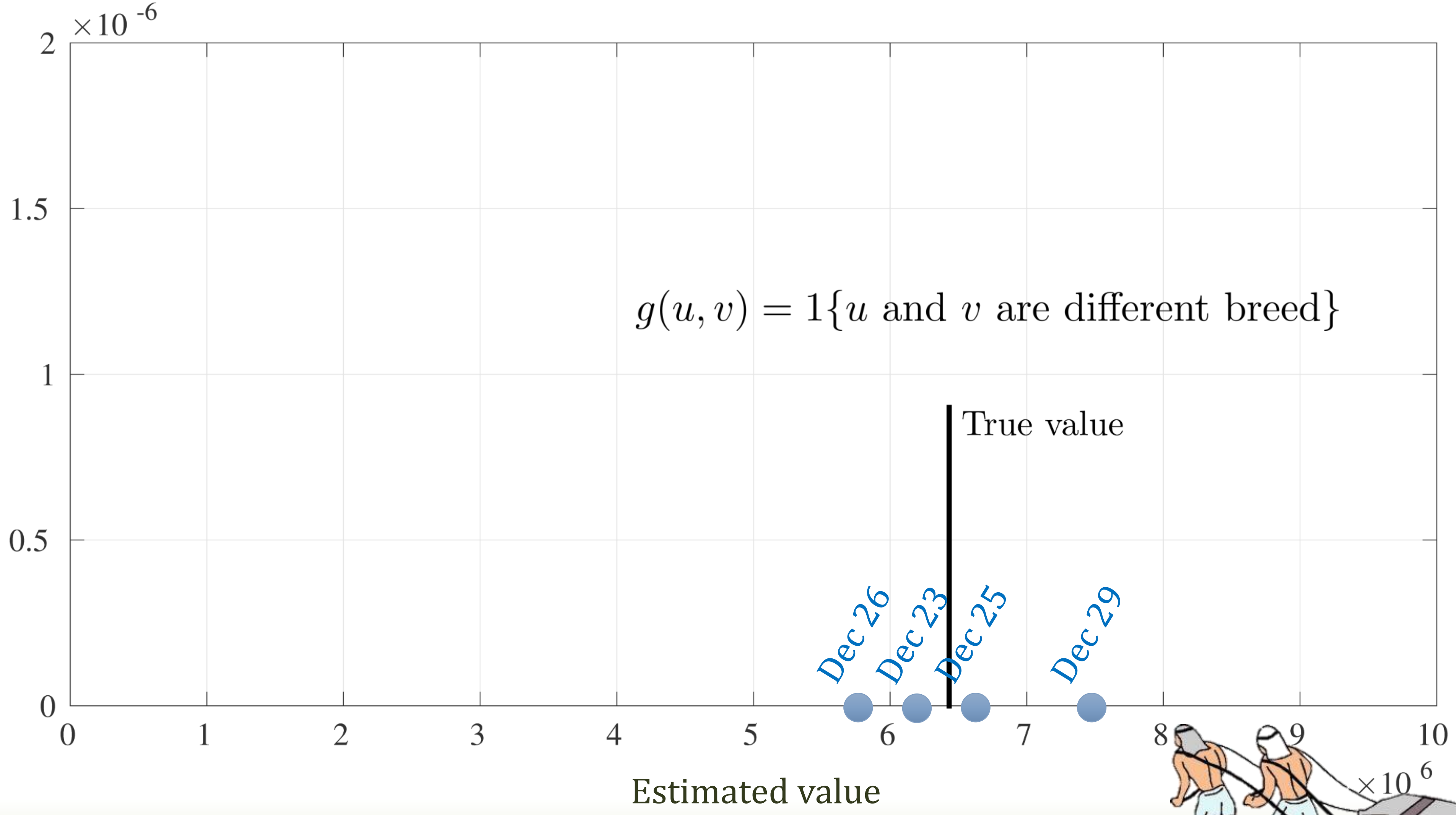
Percentage of graph covered: 2.72% (edges), 14.86% (nodes)

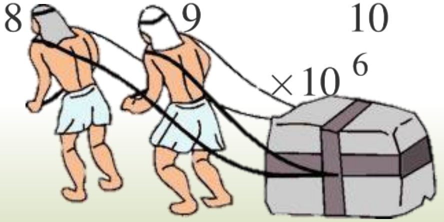


Simulations on real-world networks: Dogster network

415K nodes, 8.27M edges

Percentage of graph covered: 2.72% (edges), 14.86% (nodes)

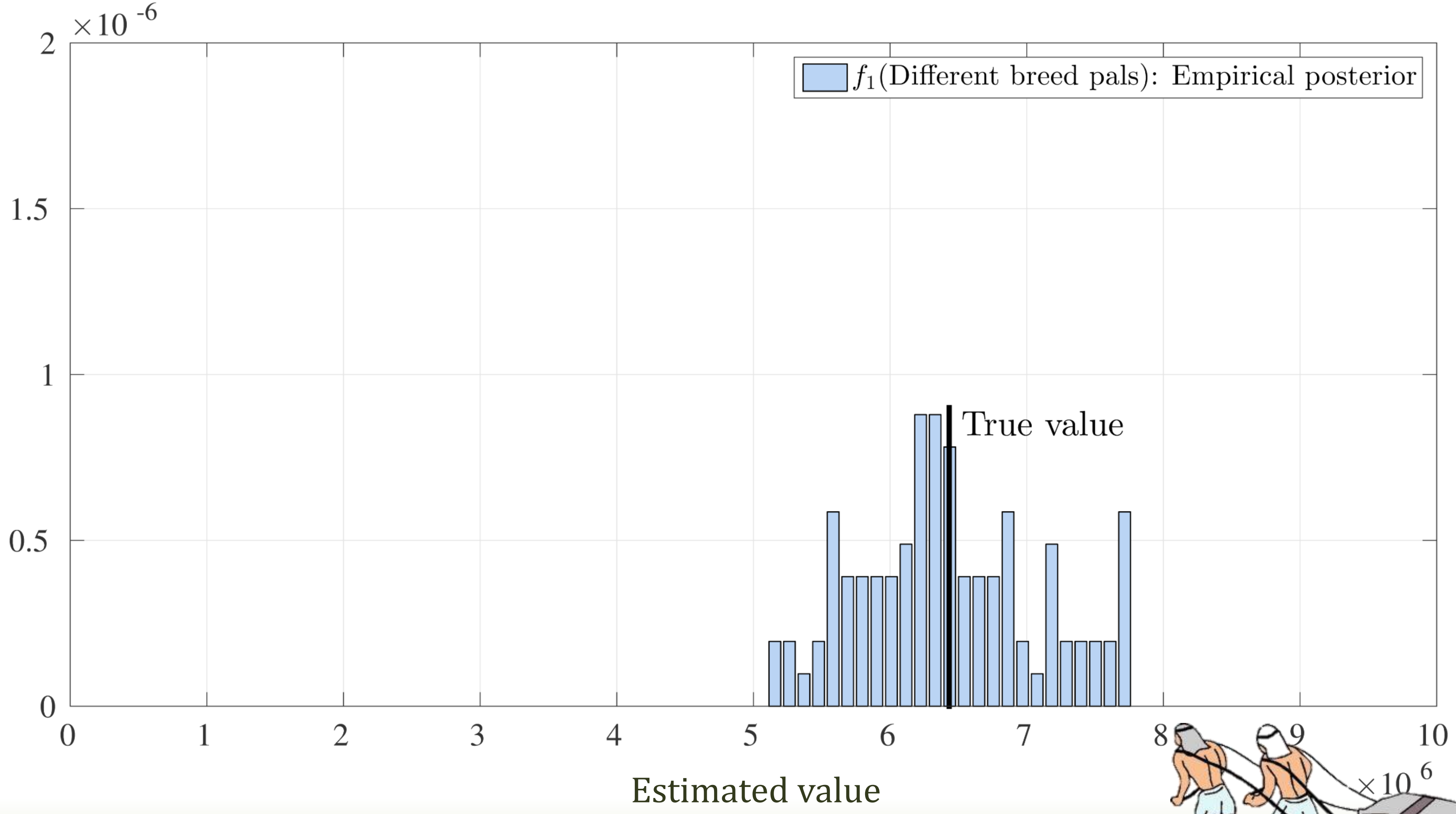


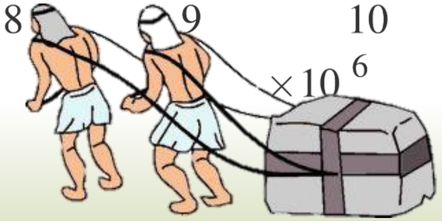


Simulations on real-world networks: Dogster network

415K nodes, 8.27M edges

Percentage of graph covered: 2.72% (edges), 14.86% (nodes)

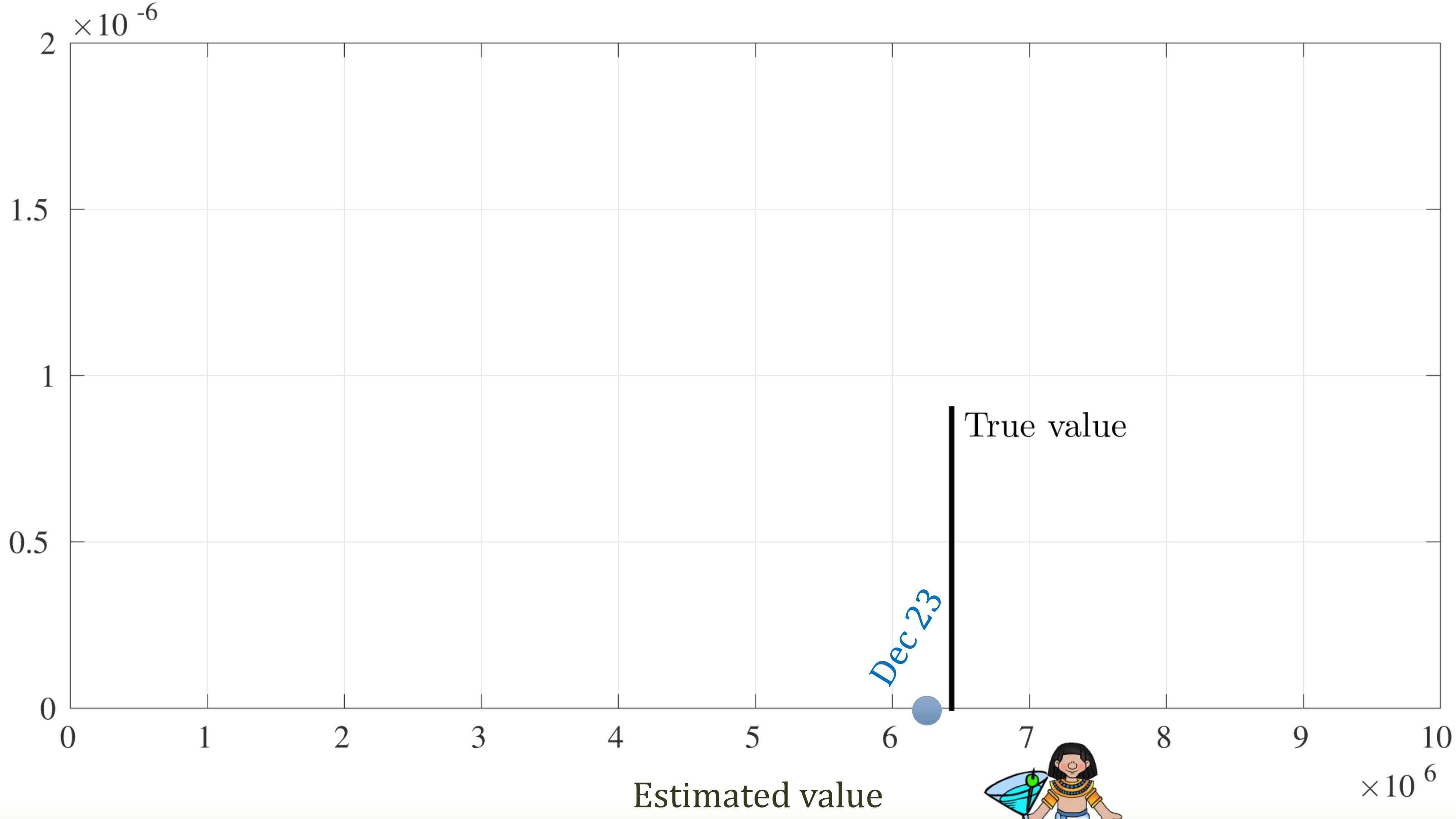




Simulations on real-world networks: Dogster network

415K nodes, 8.27M edges

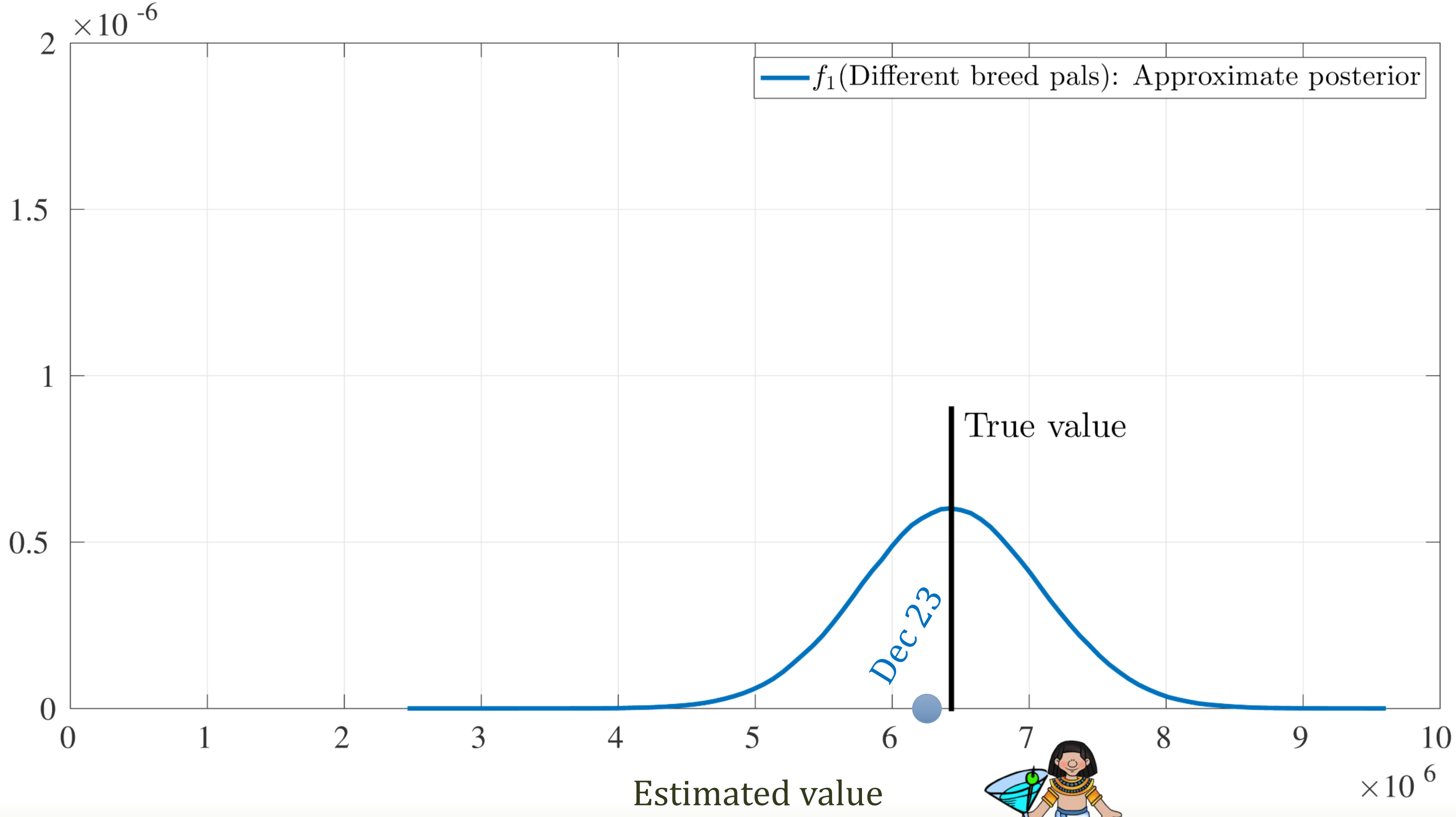
Percentage of graph covered: 2.72% (edges), 14.86% (nodes)



Simulations on real-world networks: Dogster network

415K nodes, 8.27M edges

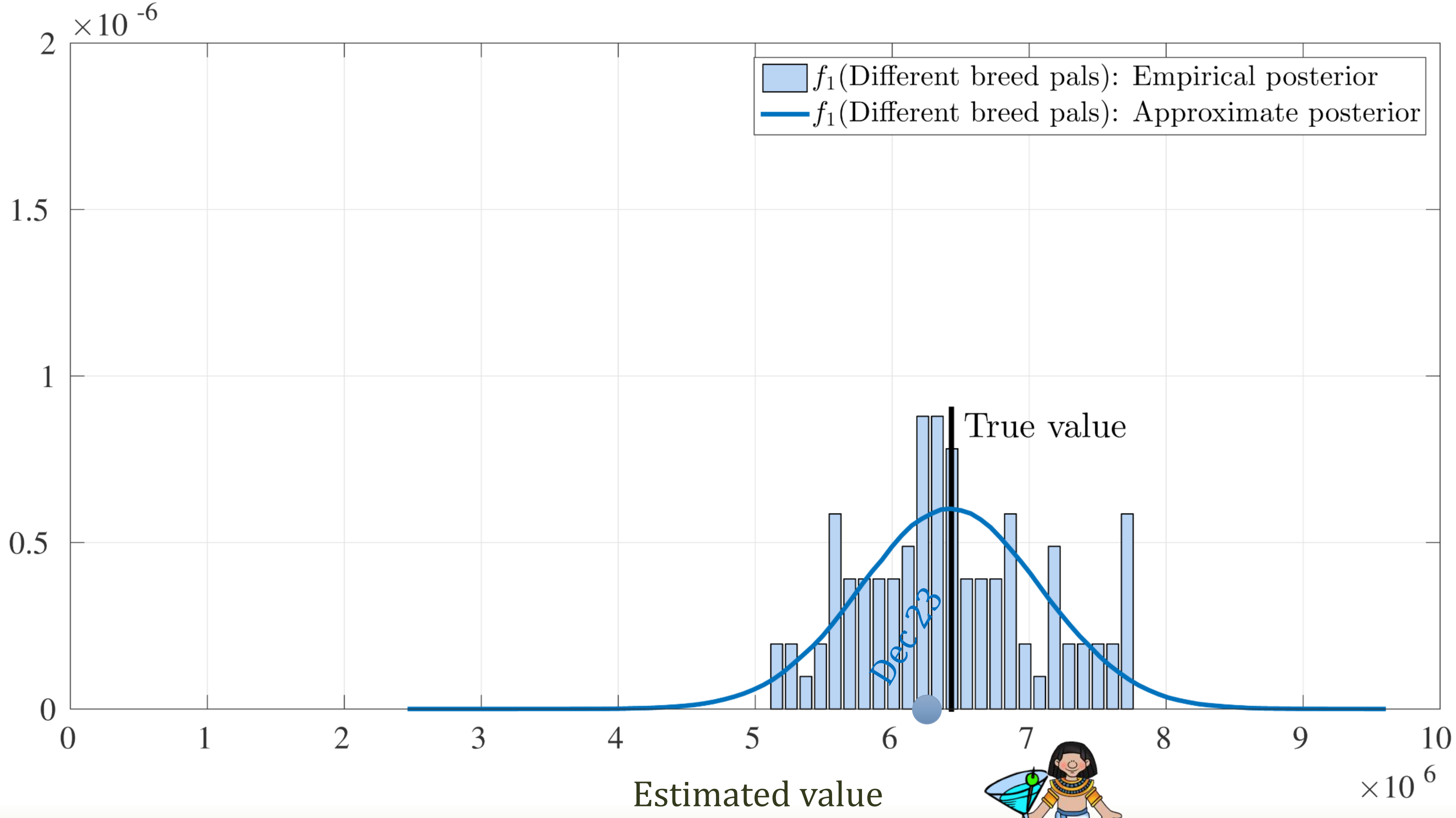
Percentage of graph covered: 2.72% (edges), 14.86% (nodes)



Simulations on real-world networks: Dogster network

415K nodes, 8.27M edges

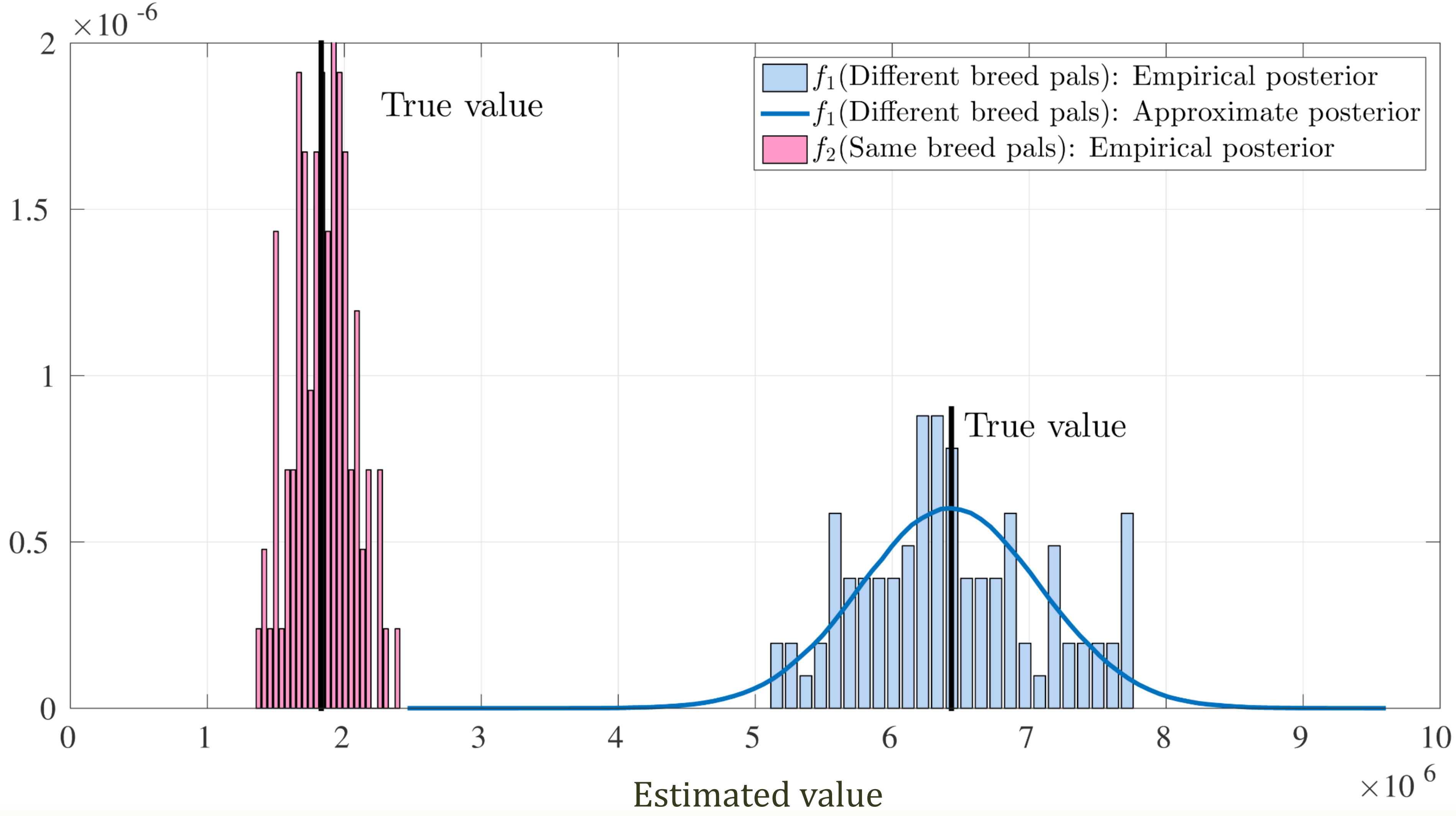
Percentage of graph covered: 2.72% (edges), 14.86% (nodes)



Simulations on real-world networks: Dogster network

415K nodes, 8.27M edges

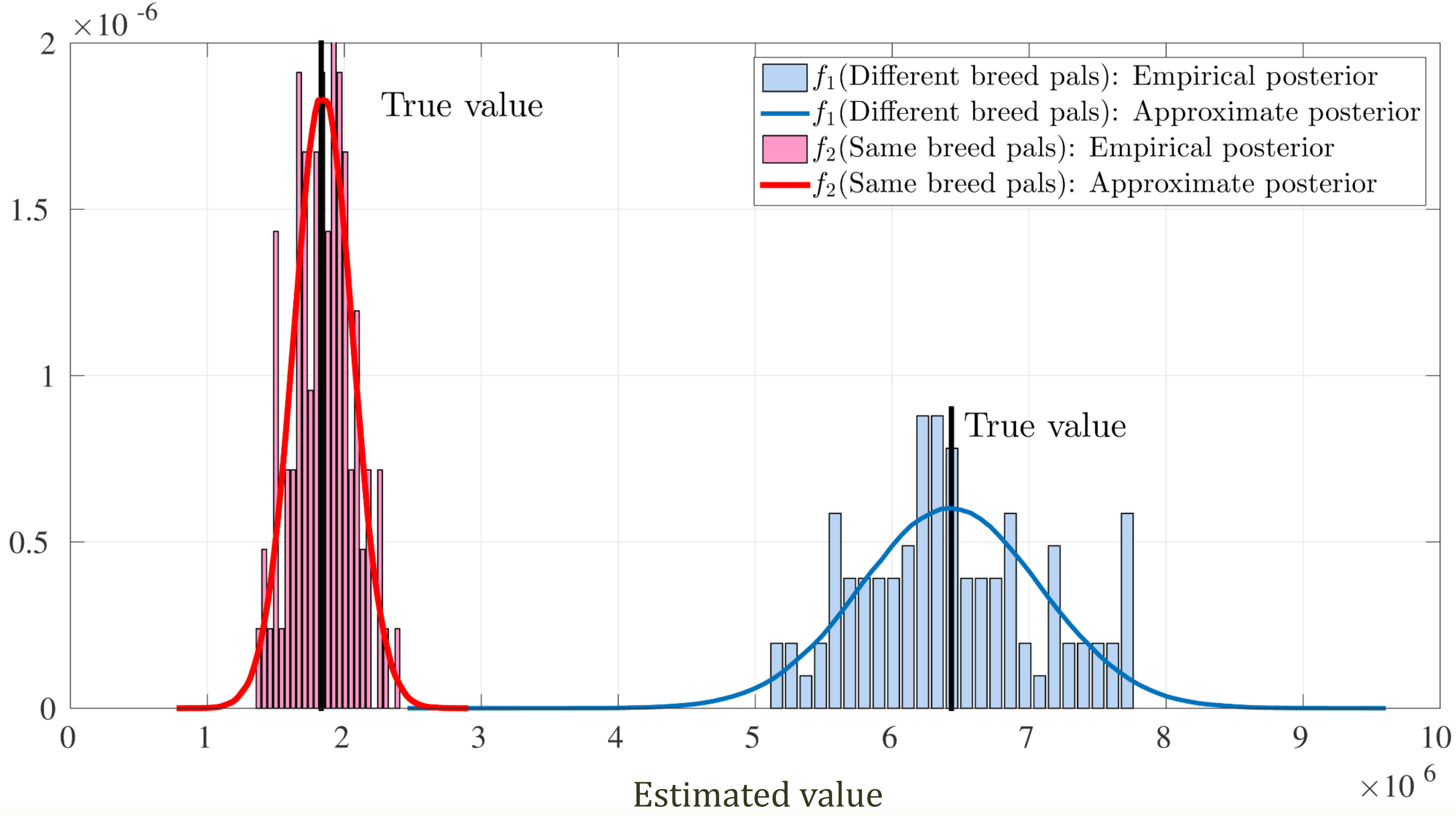
Percentage of graph covered: 2.72% (edges), 14.86% (nodes)



Simulations on real-world networks: Dogster network

415K nodes, 8.27M edges

Percentage of graph covered: 2.72% (edges), 14.86% (nodes)



Simulations on real-world networks: Friendster network

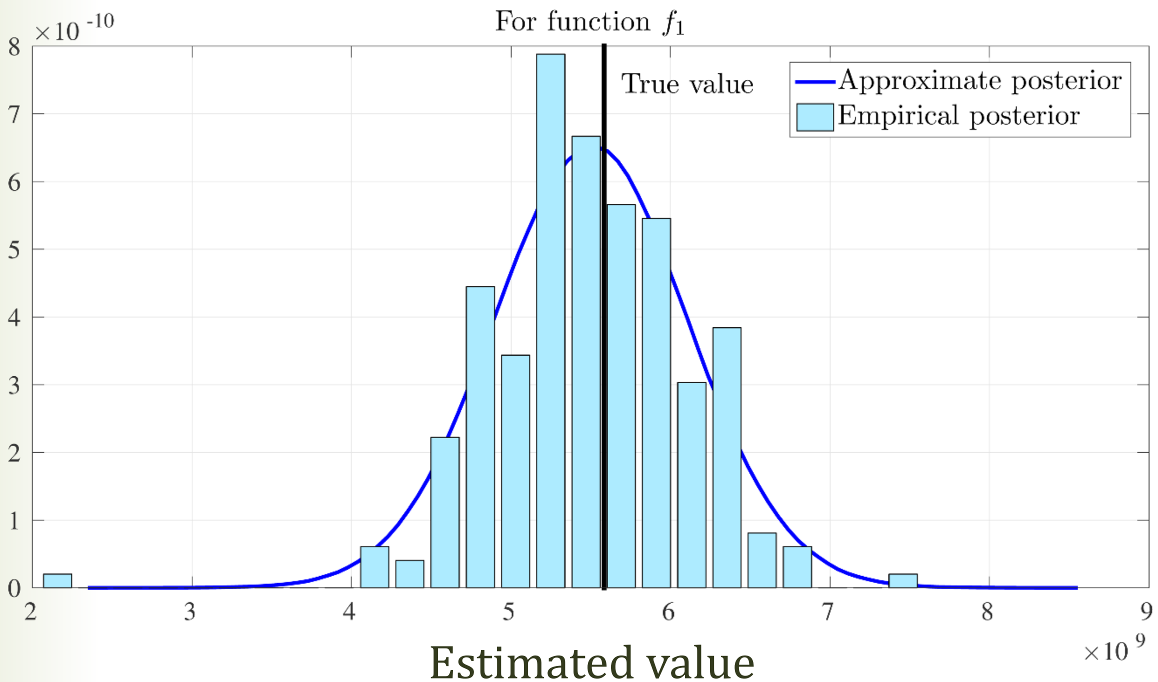
64K nodes, 1.25M edges

Percentage of graph covered: 7.43% (edges), 18.52% (nodes)

Simulations on real-world networks: Friendster network

64K nodes, 1.25M edges

Percentage of graph covered: 7.43% (edges), 18.52% (nodes)



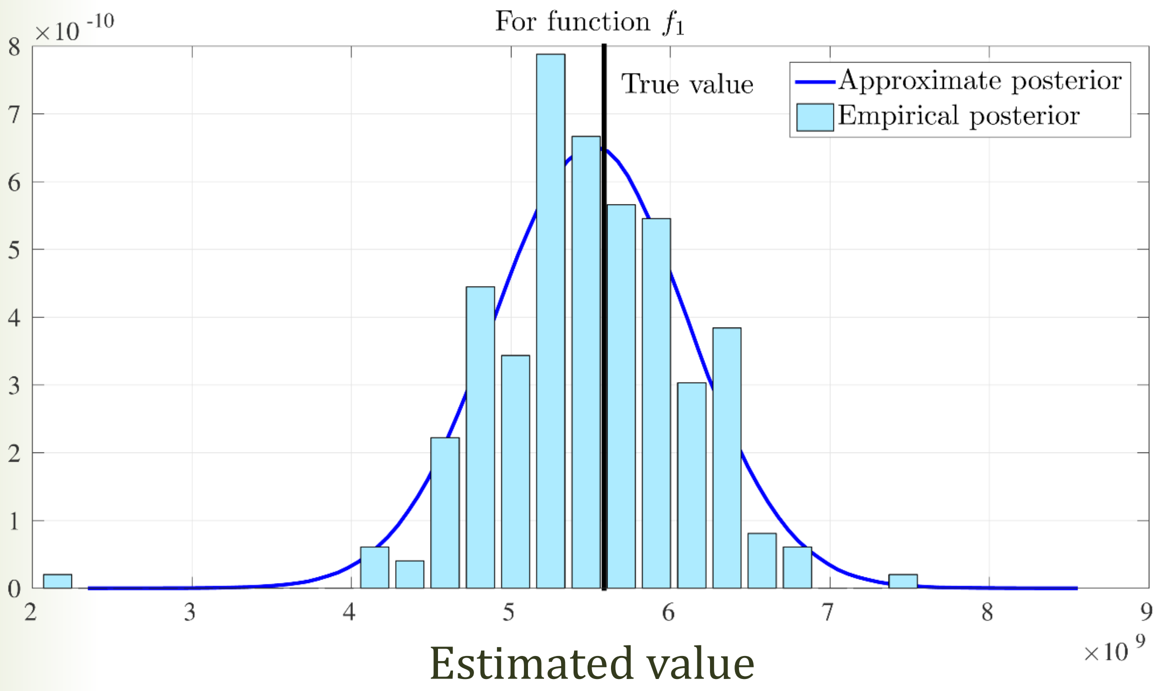
Estimated value

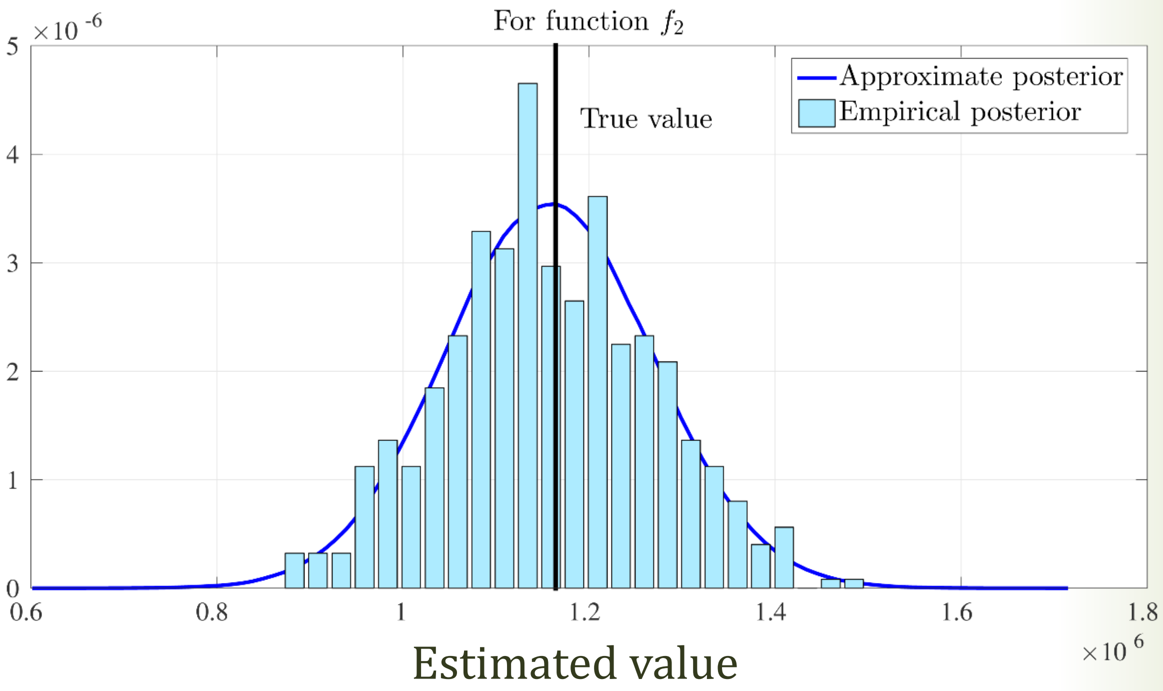
$$f_1 = d_{X_t}.d_{X_{t+1}}$$

Simulations on real-world networks: Friendster network

64K nodes, 1.25M edges

Percentage of graph covered: 7.43% (edges), 18.52% (nodes)





$$f_1 = d_{X_t}.d_{X_{t+1}}$$

$$f_2 = \begin{cases} 1 \text{ if } d_{X_t} + d_{X_{t+1}} > 50 \\ 0 \text{ otherwise} \end{cases}$$

Simulations on real-world networks: ADD Health data

A friendship network among high school students in USA

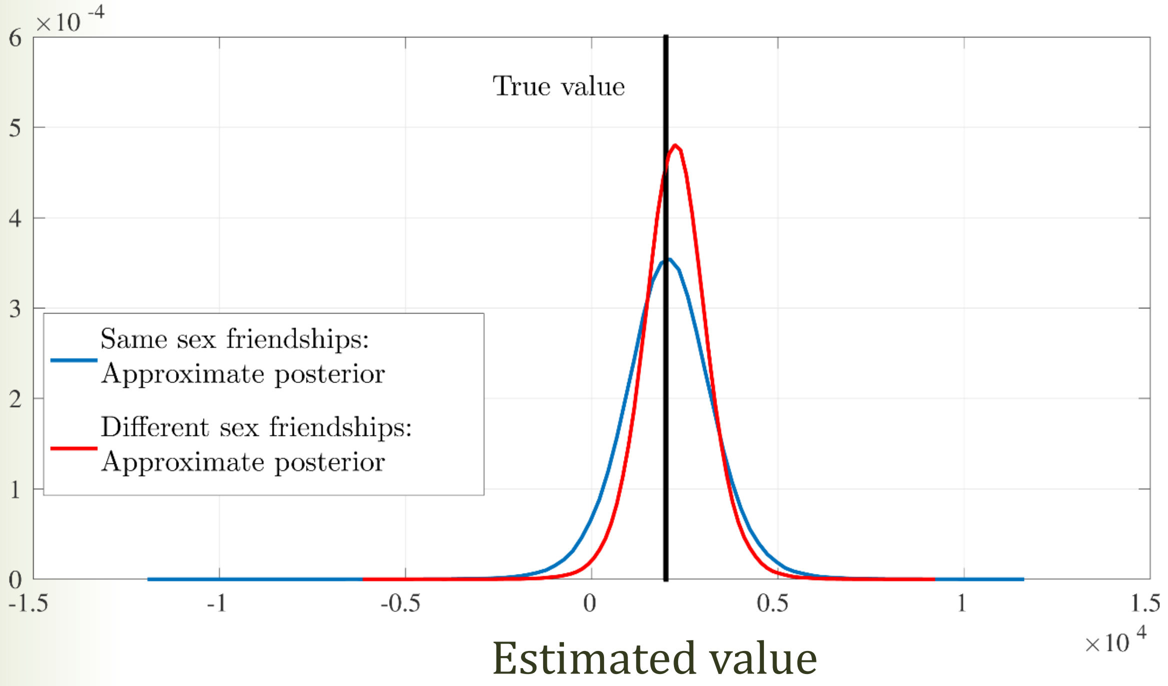
1545 nodes, 4003 edges
Percentage of graph covered: 10.87% (edges), 19.76% (nodes)

Simulations on real-world networks: ADD Health data

A friendship network among high school students in USA

1545 nodes, 4003 edges

Percentage of graph covered: 10.87% (edges), 19.76% (nodes)

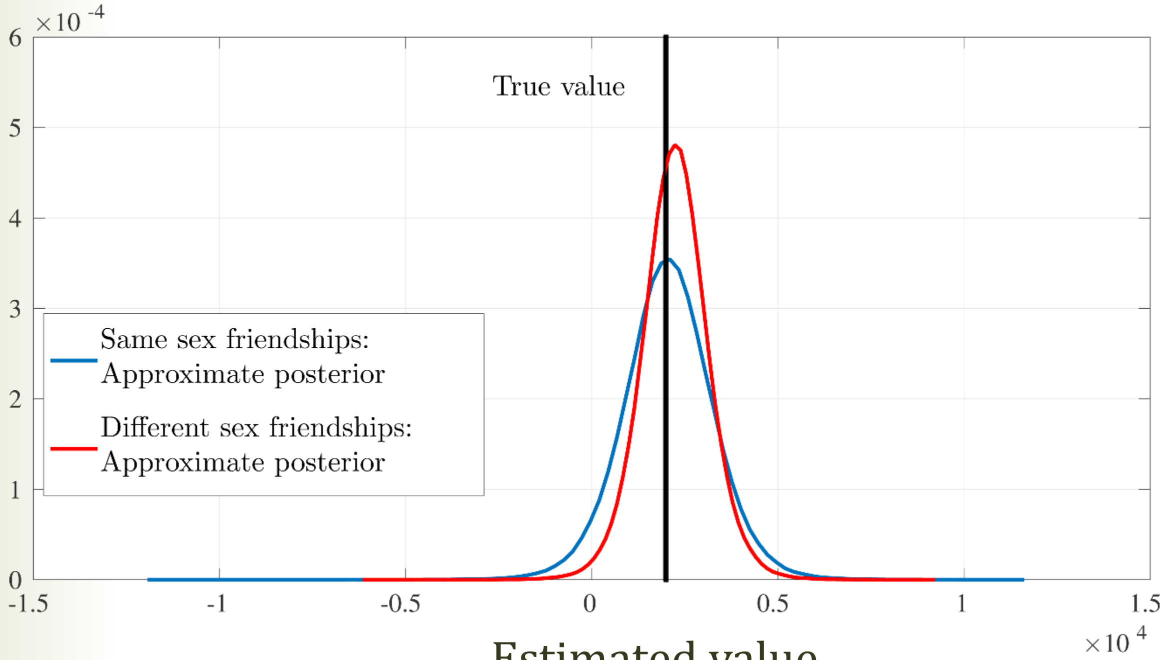


Simulations on real-world networks: ADD Health data

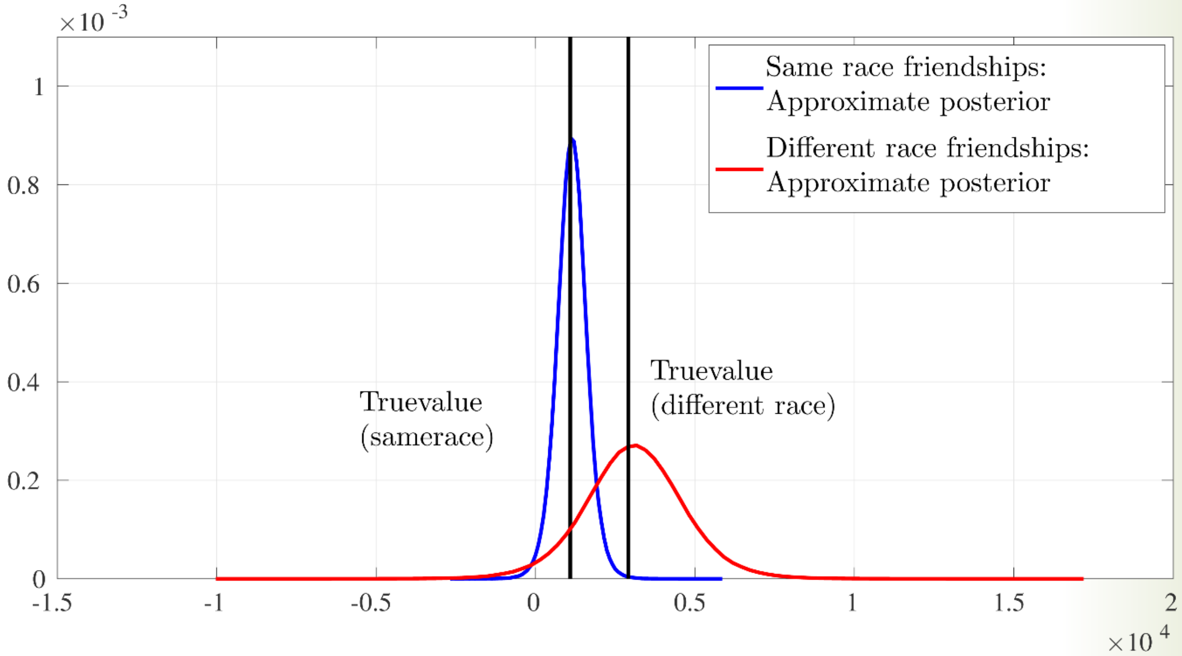
A friendship network among high school students in USA

1545 nodes, 4003 edges

Percentage of graph covered: 10.87% (edges), 19.76% (nodes)



Estimated value



Estimated value

What if the super-node is not that "super"?

What if the super-node is not that "super"?

Adaptive crawler: super-node gets bigger as crawling progresses

What if the super-node is not that "super"?

Adaptive crawler: super-node gets bigger as crawling progresses

How to add nodes to super-node:

What if the super-node is not that “super”?

Adaptive crawler: super-node gets bigger as crawling progresses

How to add nodes to super-node:

- via **any** method as long as independent of already observed tours

What if the super-node is not that “super”?

Adaptive crawler: super-node gets bigger as crawling progresses

How to add nodes to super-node:

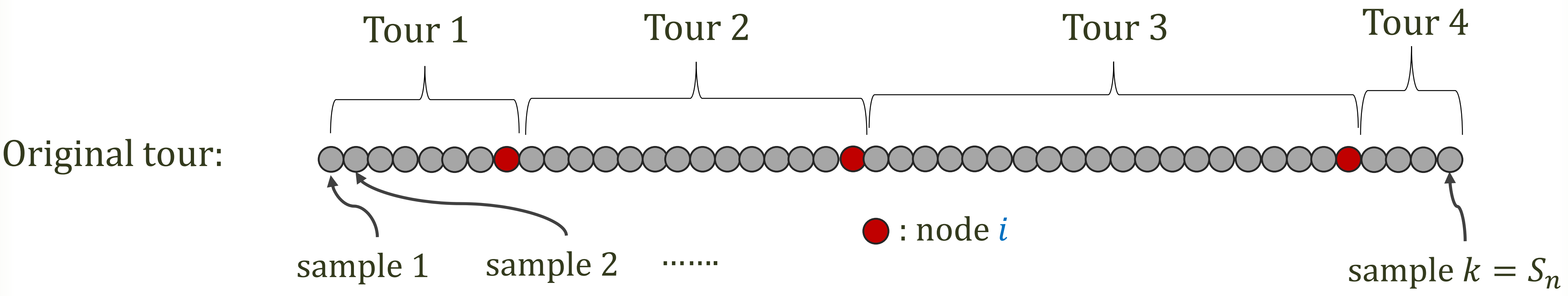
- via **any** method as long as independent of already observed tours
- Emulates retrospectively adding new node *i* into super-node S_n from the start

What if the super-node is not that “super”?

Adaptive crawler: super-node gets bigger as crawling progresses

How to add nodes to super-node:

- via **any** method as long as independent of already observed tours
- Emulates retrospectively adding new node i into super-node S_n from the start
- Checks previous tours. Breaks them when i is found.



What if the super-node is not that "super"?

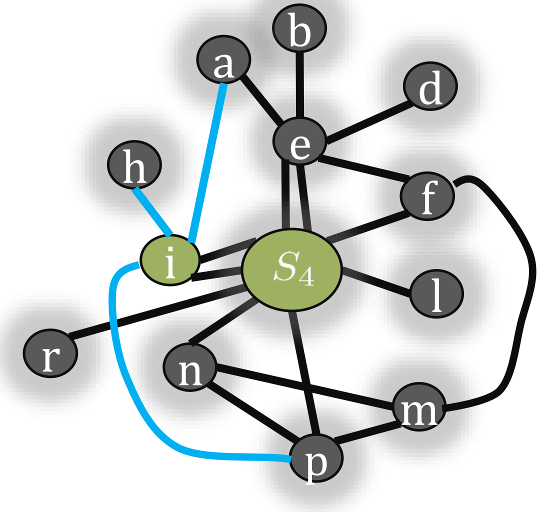
Adaptive crawler: super-node gets bigger as crawling progresses

How to add nodes to super-node:

- via **any** method as long as independent of already observed tours
- Emulates retrospectively adding new node i into super-node S_n from the start
- Checks previous tours. Breaks them when i is found.
- Start k new tours from newly added node i;
  $k \sim$ negative Binomial distribution (function of degrees of i, S_n and no of tours)

"Correction" tours from i:

Start at i, end in i or S_4



What if the super-node is not that "super"?

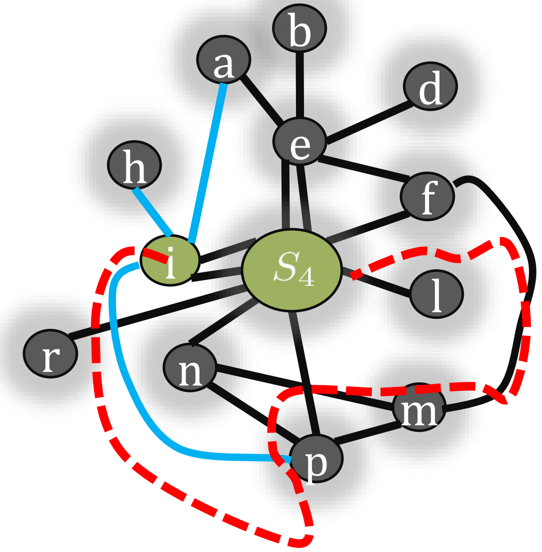
Adaptive crawler: super-node gets bigger as crawling progresses

How to add nodes to super-node:

- via **any** method as long as independent of already observed tours
- Emulates retrospectively adding new node i into super-node S_n from the start
- Checks previous tours. Breaks them when i is found.
- Start k new tours from newly added node i;
  $k \sim$ negative Binomial distribution (function of degrees of i, S_n and no of tours)

"Correction" tours from *i*:

Start at i, end in i or S_4



What if the super-node is not that "super"?

Adaptive crawler: super-node gets bigger as crawling progresses

How to add nodes to super-node:

- via **any** method as long as independent of already observed tours
- Emulates retrospectively adding new node i into super-node S_n from the start
- Checks previous tours. Breaks them when i is found.
- Start k new tours from newly added node i;
  $k \sim$ negative Binomial distribution (function of degrees of i, S_n and no of tours)

Theorem
Dynamic and static super-node sample paths are equivalent in distribution

From metric $\mu(G)$ does network look random ?

Estimation and hypothesis testing in Chung-Lu or configuration model

Estimation and hypothesis testing in Chung-Lu or configuration model

Assumption: edges labels can be written as a function of node labels

Estimation and hypothesis testing in Chung-Lu or configuration model

Assumption: edges labels can be written as a function of node labels

- Does the true value of the given graph $\mu(G) = \sum_{(u,v) \in E} g(u,v)$ belongs to the class of values when the edges are formed purely at random?

Estimation and hypothesis testing in Chung-Lu or configuration model

Assumption: edges labels can be written as a function of node labels

- Does the true value of the given graph $\mu(G) = \sum_{(u,v)\in E} g(u,v)$ belongs to the class of values when the edges are formed purely at random?

$$\mu(G) \sim \text{ Distribution}(\mathbb{E}[\mu(G_{\text{random}})],\ \text{Var}[\mu(G_{\text{random}})])$$

Estimation and hypothesis testing in Chung-Lu or configuration model

Assumption: edges labels can be written as a function of node labels

- Does the true value of the given graph $\mu(G) = \sum_{(u,v)\in E} g(u,v)$ belongs to the class of values when the edges are formed purely at random?

$$\mu(G) \sim \text{ Distribution}(\mathbb{E}[\mu(G_{\text{random}})], \text{ Var}[\mu(G_{\text{random}})])$$

- Does the true value belongs to the class when the connections are formed based on degrees alone with no other influence ?

Estimation and hypothesis testing in Chung-Lu or configuration model

Assumption: edges labels can be written as a function of node labels

- Does the true value of the given graph $\mu(G) = \sum_{(u,v)\in E} g(u,v)$ belongs to the class of values when the edges are formed purely at random?

$$\mu(G) \sim \text{ Distribution}(\mathbb{E}[\mu(G_{\text{random}})], \text{ Var}[\mu(G_{\text{random}})])$$

- Does the true value belongs to the class when the connections are formed based on degrees alone with no other influence ?

Configuration model:

Estimation and hypothesis testing in Chung-Lu or configuration model

Assumption: edges labels can be written as a function of node labels

- Does the true value of the given graph $\mu(G) = \sum_{(u,v)\in E} g(u,v)$ belongs to the class of values when the edges are formed purely at random?

$$\mu(G) \sim \text{ Distribution}(\mathbb{E}[\mu(G_{\text{random}})], \text{ Var}[\mu(G_{\text{random}})])$$

- Does the true value belongs to the class when the connections are formed based on degrees alone with no other influence ?

  Configuration model:

  - Assume the degree sequence same as that of G.

Estimation and hypothesis testing in Chung-Lu or configuration model

Assumption: edges labels can be written as a function of node labels

- Does the true value of the given graph $\mu(G) = \sum_{(u,v)\in E} g(u,v)$ belongs to the class of values when the edges are formed purely at random?

$$\mu(G) \sim \text{ Distribution}(\mathbb{E}[\mu(G_{\text{random}})], \text{ Var}[\mu(G_{\text{random}})])$$

- Does the true value belongs to the class when the connections are formed based on degrees alone with no other influence ?

  Configuration model:

  - Assume the degree sequence same as that of G.
  - Edges formed by uniformly selecting the half edges of each node

Details in the paper

Estimation in Chung-Lu or configuration model

Details in the paper

Estimation in Chung-Lu or configuration model

Estimate $\mathbb{E}[\mu(G_{\text{conf}})]$ & $\text{Var}[\mu(G_{\text{conf}})]$

- The entire degree sequence unknown; only the degrees of sampled nodes known

Details in the paper

Estimation in Chung-Lu or configuration model

Estimate $\mathbb{E}[\mu(G_{\text{conf}})]$ & $\text{Var}[\mu(G_{\text{conf}})]$

- The entire degree sequence unknown; only the degrees of sampled nodes known

$$\mathbb{E}[\mu(G_{\text{conf}})] = \sum_{\substack{(u,v)\in E\cup E^c \\ u\neq v}} g(u,v)\,\frac{d_u d_v}{2M} + \sum_{\substack{(u,v)\in E\cup E^c \\ u=v}} g(u,v)\,\frac{\binom{d_u}{2}}{2M}\,.$$

Details in the paper

Estimation in Chung-Lu or configuration model

Random walk with jumps to estimate $g(u,v)$, for $(u,v) \notin E$

Estimate $\mathbb{E}[\mu(G_{\text{conf}})]$ & $\text{Var}[\mu(G_{\text{conf}})]$

- The entire degree sequence unknown; only the degrees of sampled nodes known

$$\mathbb{E}[\mu(G_{\text{conf}})] = \sum_{\substack{(u,v)\in E\cup E^c \\ u\neq v}} g(u,v)\,\frac{d_u d_v}{2M} + \sum_{\substack{(u,v)\in E\cup E^c \\ u=v}} g(u,v)\,\frac{\binom{d_u}{2}}{2M}\,.$$

Details in the paper

Estimation in Chung-Lu or configuration model

Estimate $\mathbb{E}[\mu(G_{\text{conf}})]$ & $\text{Var}[\mu(G_{\text{conf}})]$

- The entire degree sequence unknown; only the degrees of sampled nodes known

$$\mathbb{E}[\mu(G_{\text{conf}})] = \sum_{\substack{(u,v)\in E\cup E^c \\ u\neq v}} g(u,v)\,\frac{d_u d_v}{2M} + \sum_{\substack{(u,v)\in E\cup E^c \\ u=v}} g(u,v)\,\frac{\binom{d_u}{2}}{2M}.$$

Random walk with jumps to estimate $g(u,v)$, for $(u,v) \notin E$

$$\Pr(\text{head}) := p = \frac{d_t}{d_t+\alpha}$$

Details in the paper

Estimation in Chung-Lu or configuration model

Estimate $\mathbb{E}[\mu(G_{\text{conf}})]$ & $\text{Var}[\mu(G_{\text{conf}})]$

- The entire degree sequence unknown; only the degrees of sampled nodes known

$$\mathbb{E}[\mu(G_{\text{conf}})] = \sum_{\substack{(u,v)\in E\cup E^c \\ u\neq v}} g(u,v)\,\frac{d_u d_v}{2M} + \sum_{\substack{(u,v)\in E\cup E^c \\ u=v}} g(u,v)\,\frac{\binom{d_u}{2}}{2M}\,.$$

Random walk with jumps to estimate $g(u,v)$, for $(u,v) \notin E$

$$\Pr(\text{head}) := p = \frac{d_t}{d_t+\alpha}$$

with p, follow RW

with $1-p$, uniform node sampling

Hypothesis testing with the Chung-Lu model

Hypothesis testing with the Chung-Lu model

$$\sum_{(u,v)\in E_{\text{C-L}}} g(u,v) \sim \text{Normal}\left(\mathbb{E}[\mu(G_{\text{C-L}})], \text{Var}(G_{\text{C-L}})\right)$$

(Lindeberg central limit theorem)

Hypothesis testing with the Chung-Lu model

$$\sum_{(u,v)\in E_{\text{C-L}}} g(u,v) \sim \text{Normal}\left(\mathbb{E}[\mu(G_{\text{C-L}})], \text{Var}(G_{\text{C-L}})\right)$$ (Lindeberg central limit theorem)

Look for the value of a the following satisfies

$$|\hat{\mu}(G) - \mathbb{E}[\mu(G_{\text{C-L}})]| \leq a\sqrt{\text{Var}(G_{\text{C-L}})}$$

Estimate value of given graph

Mean and variance of Chung-Lu graph

More results in the paper

Hypothesis testing with the Chung-Lu model

$$\sum_{(u,v)\in E_{\text{C-L}}} g(u,v) \sim \text{Normal}\left(\mathbb{E}[\mu(G_{\text{C-L}})], \text{Var}(G_{\text{C-L}})\right)$$ (Lindeberg central limit theorem)

Look for the value of a the following satisfies

$$|\hat{\mu}(G) - \mathbb{E}[\mu(G_{\text{C-L}})]| \leq a\sqrt{\text{Var}(G_{\text{C-L}})}$$

Estimate value of given graph

Mean and variance of Chung-Lu graph

Dogster network: Estimator for $\mathbb{E}[\mu(G_{\text{C-L}})]$

Hypothesis testing with the Chung-Lu model

$$\sum_{(u,v)\in E_{\text{C-L}}} g(u,v) \sim \text{Normal}\left(\mathbb{E}[\mu(G_{\text{C-L}})], \text{Var}(G_{\text{C-L}})\right)$$ (Lindeberg central limit theorem)

Look for the value of a the following satisfies

$$|\hat{\mu}(G) - \mathbb{E}[\mu(G_{\text{C-L}})]| \leq a\sqrt{\text{Var}(G_{\text{C-L}})}$$

Estimate value of given graph

Mean and variance of Chung-Lu graph

Dogster network: Estimator for $\mathbb{E}[\mu(G_{\text{C-L}})]$

Percentage of graph crawled: 8.9% (edges), 18.51% (nodes)

More results in the paper

Hypothesis testing with the Chung-Lu model

$$\sum_{(u,v)\in E_{\text{C-L}}} g(u,v) \sim \text{Normal}\left(\mathbb{E}[\mu(G_{\text{C-L}})], \text{Var}(G_{\text{C-L}})\right)$$ (Lindeberg central limit theorem)

Look for the value of a the following satisfies

$$|\hat{\mu}(G) - \mathbb{E}[\mu(G_{\text{C-L}})]| \le a\sqrt{\text{Var}(G_{\text{C-L}})}$$

Estimate value of given graph

Mean and variance of Chung-Lu graph

Dogster network: Estimator for $\mathbb{E}[\mu(G_{\text{C-L}})]$

Percentage of graph crawled: 8.9% (edges), 18.51% (nodes)

Edge function	True value	Estimated value
1{same breed nodes}	8.12×10^6	8.066×10^6
1{different breed nodes}	2.17×10^5	1.995×10^5

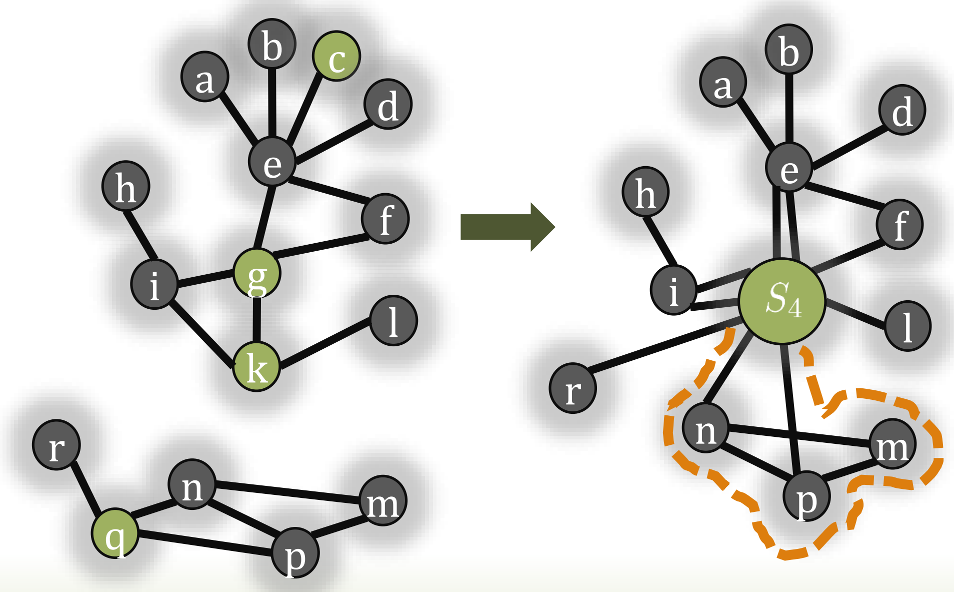
Conclusions

Conclusions

- Unbiased estimator of $\mu(G) = \sum_{(u,v)\in E} g(u,v)$

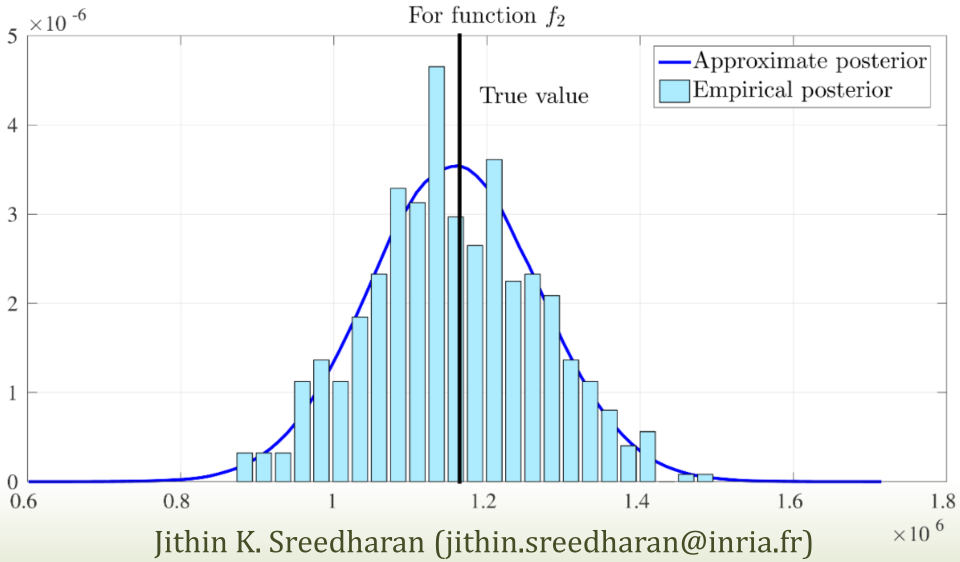
Conclusions

- Unbiased estimator of $\mu(G) = \sum_{(u,v)\in E} g(u,v)$
- Propose dynamic super-node:
  - ✓ Short parallel random walk crawls
  - ✓ Parameter-free crawling



Conclusions

- Unbiased estimator of $\mu(G) = \sum_{(u,v) \in E} g(u,v)$
- Propose dynamic super-node:
  - ✓ Short parallel random walk crawls
  - ✓ Parameter-free crawling
- Provides real-time assessment of estimation accuracy:
  - ✓ Bayesian formulation: posterior distribution, matches well true histogram



Conclusions

- Unbiased estimator of $\mu(G) = \sum_{(u,v)\in E} g(u,v)$
- Propose dynamic super-node:
  - ✓ Short parallel random walk crawls
  - ✓ Parameter-free crawling
- Provides real-time assessment of estimation accuracy:
  - ✓ Bayesian formulation: posterior distribution, matches well true histogram
- If the given network forms connections randomly:
  - ✓ Estimation of expected value and variance of $\mu(G_{\text{conf}})$
  - ✓ Check whether original network value samples from distribution of $\mu(G_{\text{conf}})$

Thank you!

Software and paper available at http://bit.do/Jithin