Random Walks & Graph Properties

Ravi Kumar

Google

Credits

- Joint work with Flavio Chierichetti, Anirban Dasgupta, Silvio Lattanzi, Tamas Sarlos

Setting

Given a graph, estimate its basic parameters

- Number of nodes
- Number of edges
- Fraction of nodes/edges of certain type
- Largest/average degree
- Local/global clustering coefficient
- Number of triangles

Applications

- Business intelligence
  - How many art lovers are in social network X?
  - Is X's social network in Paris as well connected as that of Y?
- Algorithmic reasons
  - Is the triangle density unusually small in certain portions of the graph?
  - How does the average degree vary over time?

Sampling

- Critical tool to understand and analyze large graphs
  - Study graph properties using samples
- Only realistic option in many situations
  - Graph constantly changing
  - Entire graph not accessible
- Important to have provably good algorithms
  - Sample quality $\Rightarrow$ quality of the output

Estimation by sampling

- German tank problem
  - Frequentist, Bayesian estimates
- Mark and recapture
  - Peterson-Lincoln-Chapman indices
  - Used in ecology
- Fraction of subpopulation
  - Population with a specific property

Estimation by sampling

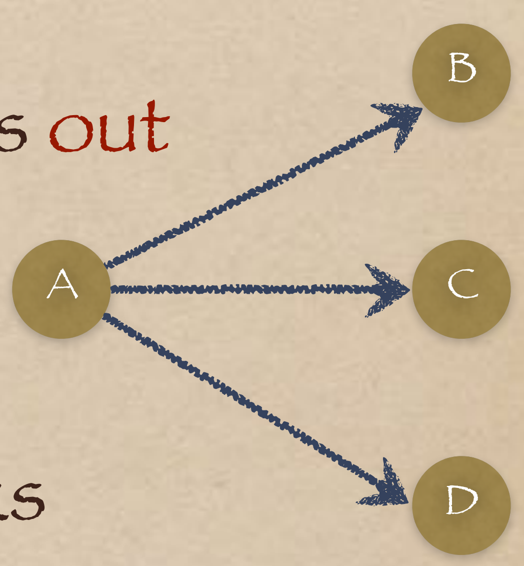
- Important when population is too large to obtain information from everyone
- Broad uses in statistics, computer science, sociology, economics, …
- Eg, polling to estimate
  - Political preferences
  - Average income, education level, …

Sampling in graphs

Graph access model

How to access the graph and what information is available to the algorithm?

- Can query any node by its name and get its out neighborhood
  - Subscribes to standard crawling model
  - Applies to both Web and social networks



- A small number of (truly random) nodes are available
  - Truly random nodes are expensive
- This access model supports random walks on the graph
- Querying is an expensive operation
  - Algorithms should minimize number of queries

Sampling according to a distribution

- $G = (V, E)$ be an undirected, connected graph
  - n = #nodes, m = #edges
- D = a distribution on V
- ε = error parameter

Problem. Using the graph access model, output a node in G according to D (to within ε additive error)

$$\Pr[\text{algorithm outputs } v] \approx D(v) \pm \varepsilon$$

- Measure #steps, #queries

An easy case

- Degree-proportional case (ie, uniform edge)
  - $D_1(v) \propto d(v)$
- Solution: do a uniform random walk on the graph

Fact. Limiting distribution of the walk is D_1

Fact. Expected number of steps is the mixing time (t_{mix}) of the graph

Uniform distribution

- Output a node uniform at random
  - $D_0(v) = 1/n$

Idea#1: Rejection sampling

Generate and reject

- Uniform random walk for t_{mix} steps
- Reached a node u
- With probability proportional to 1/d(u), output u and stop
- Otherwise, go to first step starting from u

Analysis

- Assume minimum degree is 1

Claim. E[#queries] = E[#steps] = $O(t_{mix} \cdot d_{avg})$

Proof. Generates u according to D_1 and outputs u wp 1/d(u). Probability of outputting some node

$$\Sigma_u \Pr[U = u] \times 1/d(u) = \Sigma_u d(u)/(2m) \times 1/d(u)$$

$$= \Sigma_u 1/(2m) = n / 2m = 1/d_{avg}$$

Repeat this d_{avg} times to successfully get a sample

Idea#2: Max-degree (MD) walk

- Make the graph uniform degree by spending more time at low degree nodes
  - Uniform random walk on modified graph generates D_0
- Use max degree (d_{max}) to define transitions
- #queries could be « #steps

MD Analysis

Claim. The steady-state of MD is D_0

Claim. E[#steps] spent at node u is $d_{max}/d(u)$

Claim. For any real-valued function f

$$\frac{\Sigma_{uv} (f(u) - f(v))^2 d(u)\, d(v)}{\Sigma_{uv} (f(u) - f(v))^2} \geq d_{avg}/2$$

MD Analysis (contd)

- Use the variational characterization

$$1 - \lambda_2 = \inf_f \frac{\Sigma_{uv} (f(u) - f(v))^2 \pi(u) P(u, v)}{\Sigma_{uv} (f(u) - f(v))^2 \pi(u) \pi(v)}$$

- Relate λ_2 of MD and original walk using this

Fact. $t_{mix} \leq 1/(1 - \lambda_2) \log n$

Claim. $E[\#steps] = \tilde{O}(t_{mix} \cdot d_{avg})$

Idea#3: Metropolis-Hastings (MH)

- A way to sample from any target distribution D starting from an arbitrary transition matrix Q
    - Current state = u
    - Generate $v \sim Q(u, \cdot)$
    - Move to v wp $\min(1, (Q(v, u)\, D(u)) / (Q(u, v)\, D(v)))$
- Fact. Steady-state of MH walk is D
- If $D = D_0$ and Q is given by the graph

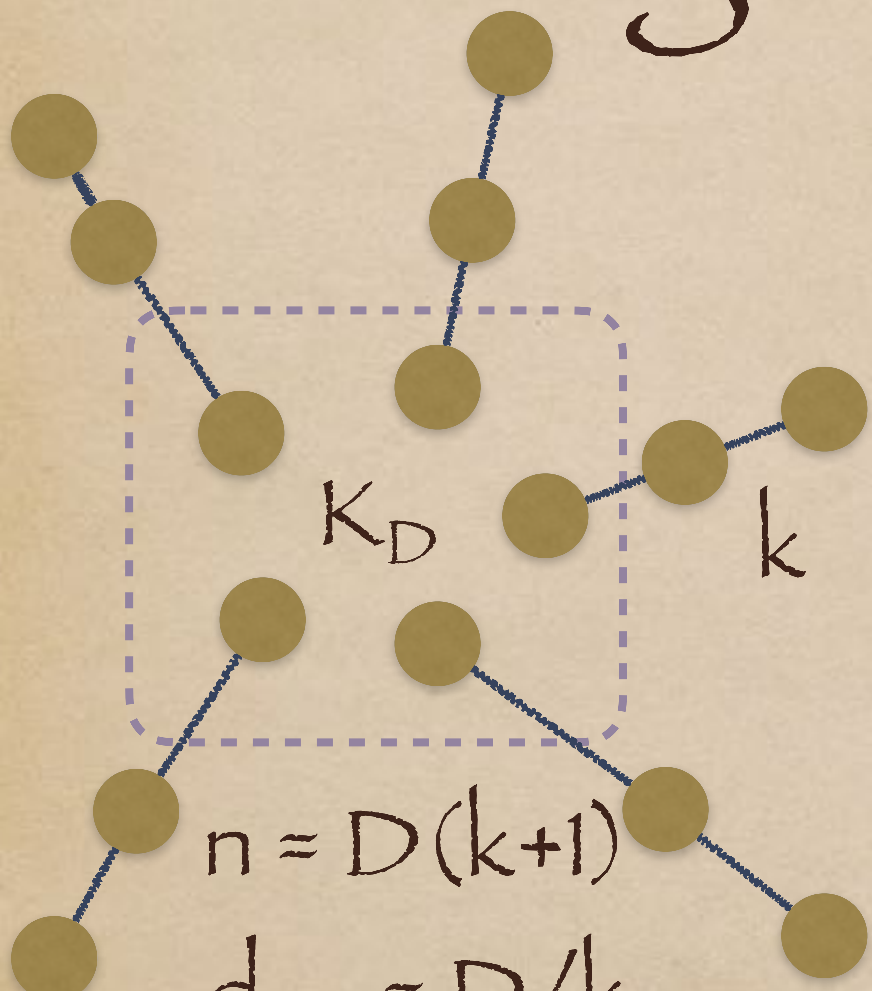
$\Pr[u \to v] = 1/d(u) \cdot \min(1, d(u)/d(v)) = 1/\max(d(u), d(v))$

MH Analysis

Claim. $E[\#steps] = \tilde{O}(t_{mix} \cdot d_{max})$

Proof. Use the variational characterization and steps as before

Tightness of MH



$n = D(k+1)$

$d_{avg} = D/k$

$d_{max} = D$

$t_{mix} = \Theta(k^2)$

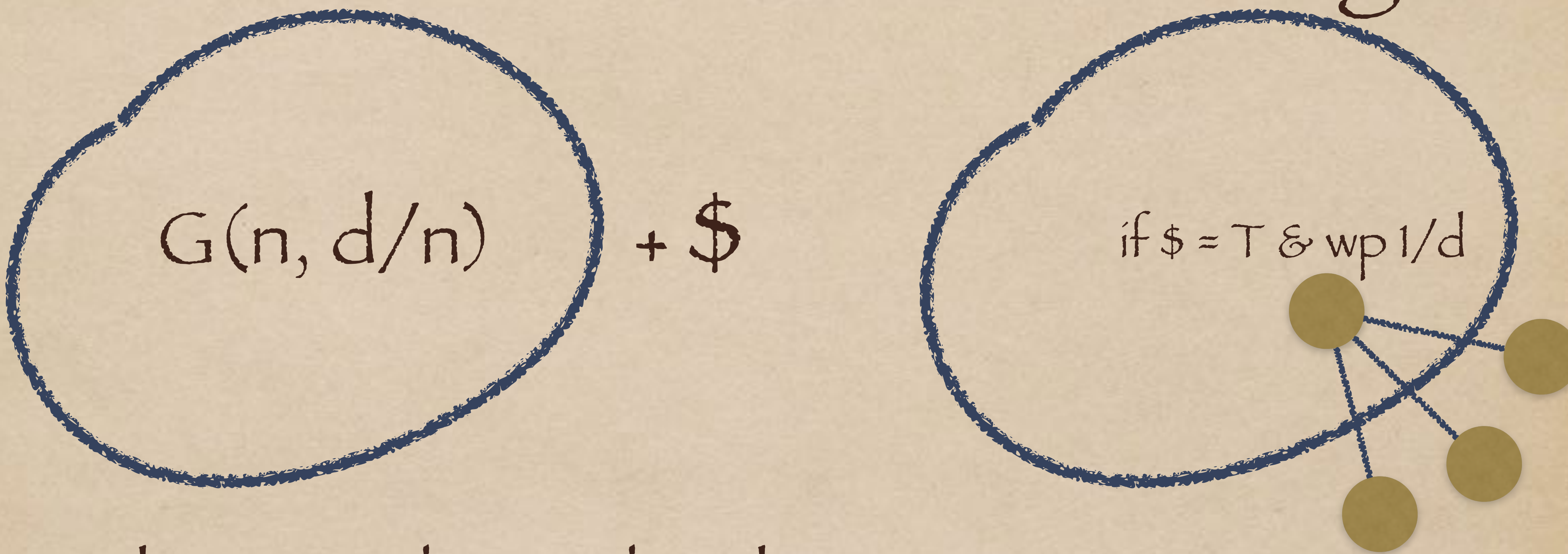
Claim. $E[\text{steps}] \geq \boldsymbol{\Omega}(t_{mix}\, d_{max})$

Proof. $o(k^2)$ non-self loop steps will miss constant fraction of path nodes

To be close to D_0 we need $\boldsymbol{\Omega}(k^2)$ steps

Self-loop steps on path nodes is $\boldsymbol{\Omega}(D)$

Lower bounds: $\Omega(d_{avg})$



- $d_{avg} = d$, $t_{mix} = O(\log n / \log d)$
- Distance between D_0 for c = H and c = T is 1/2 - o(1)
- #queries = o(d) ⇒ query only unchanged nodes wp 1 - o(1)

Lower bounds: $\Omega(t_{mix})$

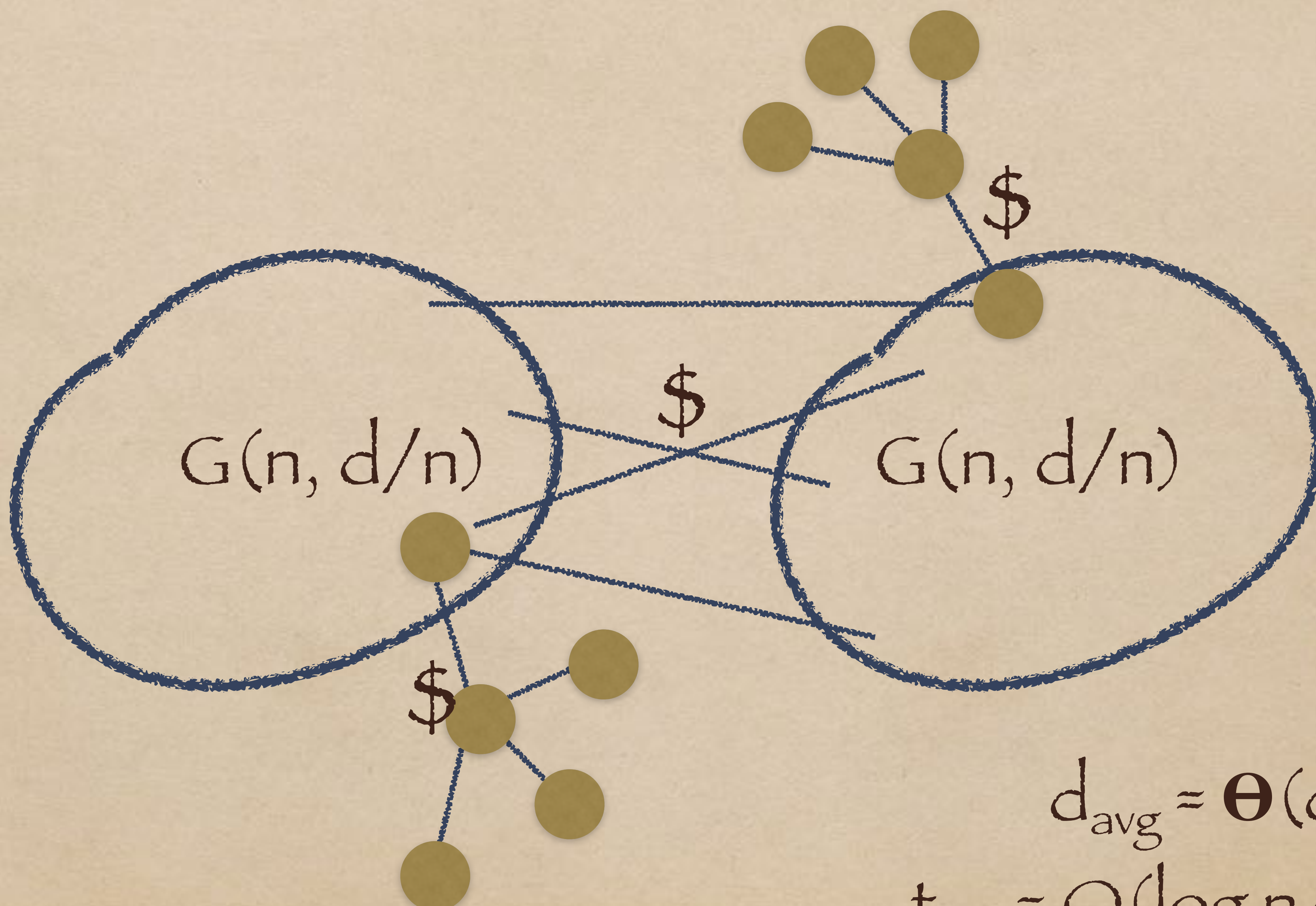
Claim. Any algorithm for D_0 must issue $\Omega(t_{mix})$ queries

Lower bounds: $\Omega(d_{avg}t_{mix})$

- (Chierichetti, Haddadan 2018)

Claim. Any algorithm to obtain, with probability at least $1-\delta$, an ϵ-additive approximation of the average of a bounded function on the nodes of a graph, must issue $\Omega(d_{avg}t_{mix})$ queries

Construction



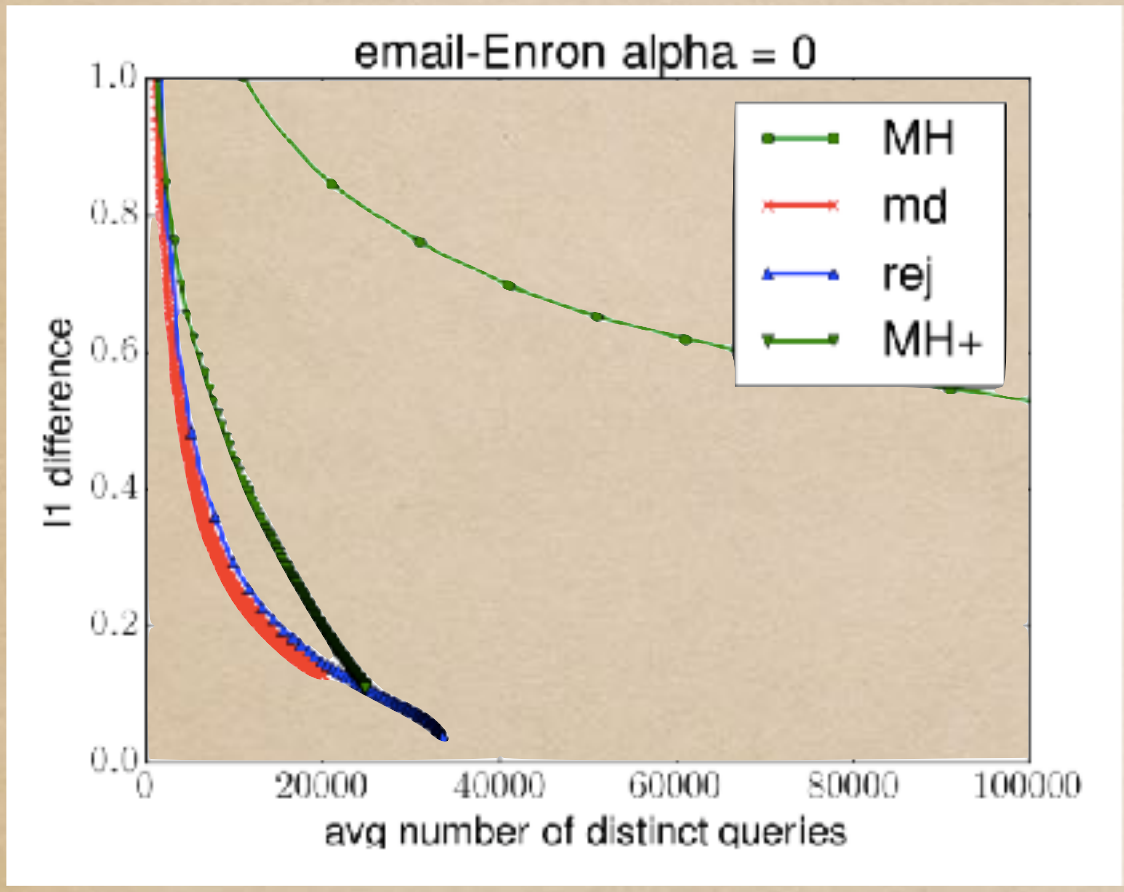
$d_{avg} = \Theta(d)$

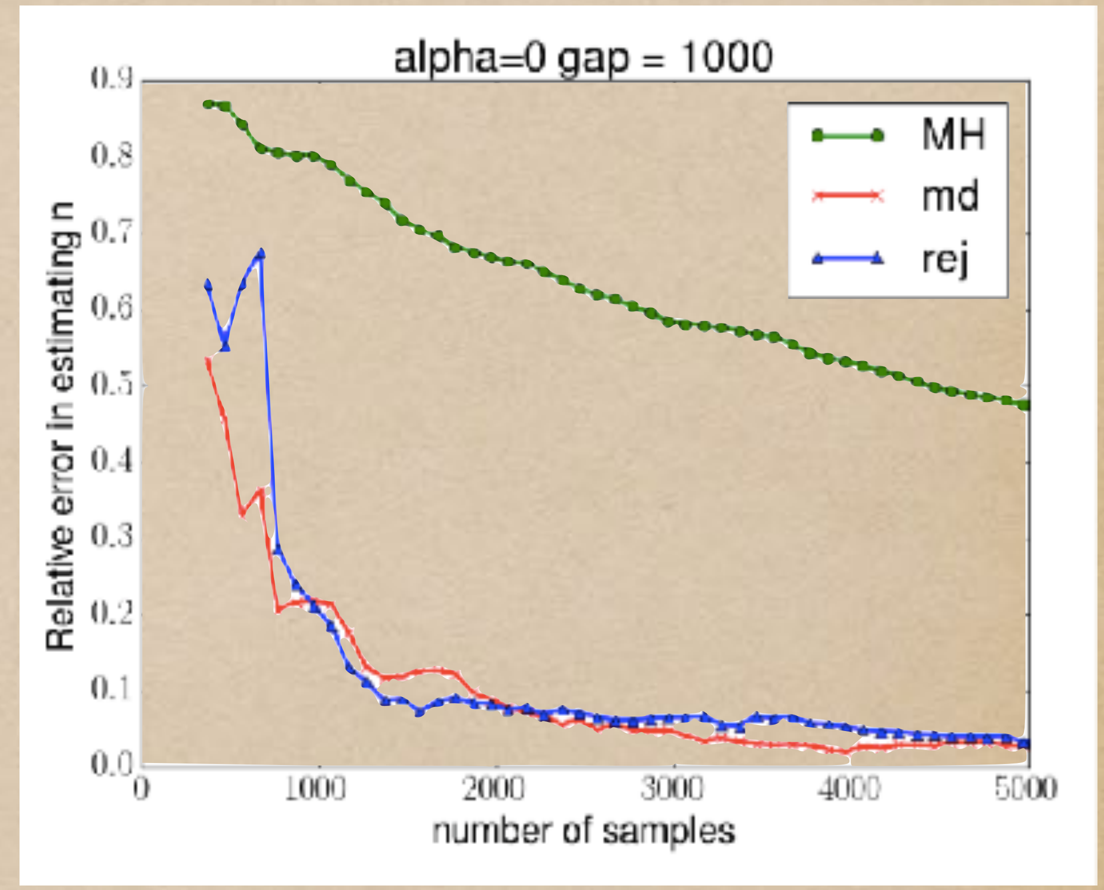
$t_{mix} = O(\log n / \log d)$

Experiments

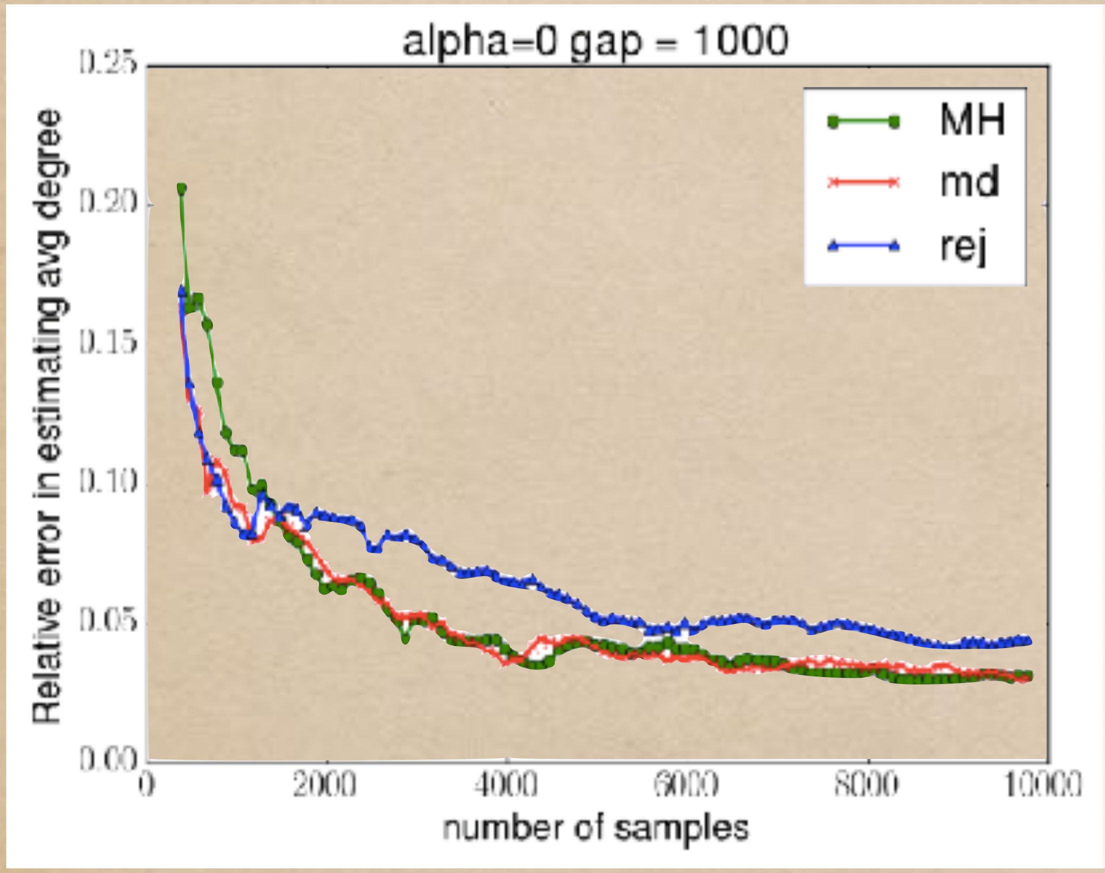
- Uniformity of the samples
  - Strict criterion
- Quality of estimators based on samples
  - Size of the network
  - Average degree
  - Clustering coefficient

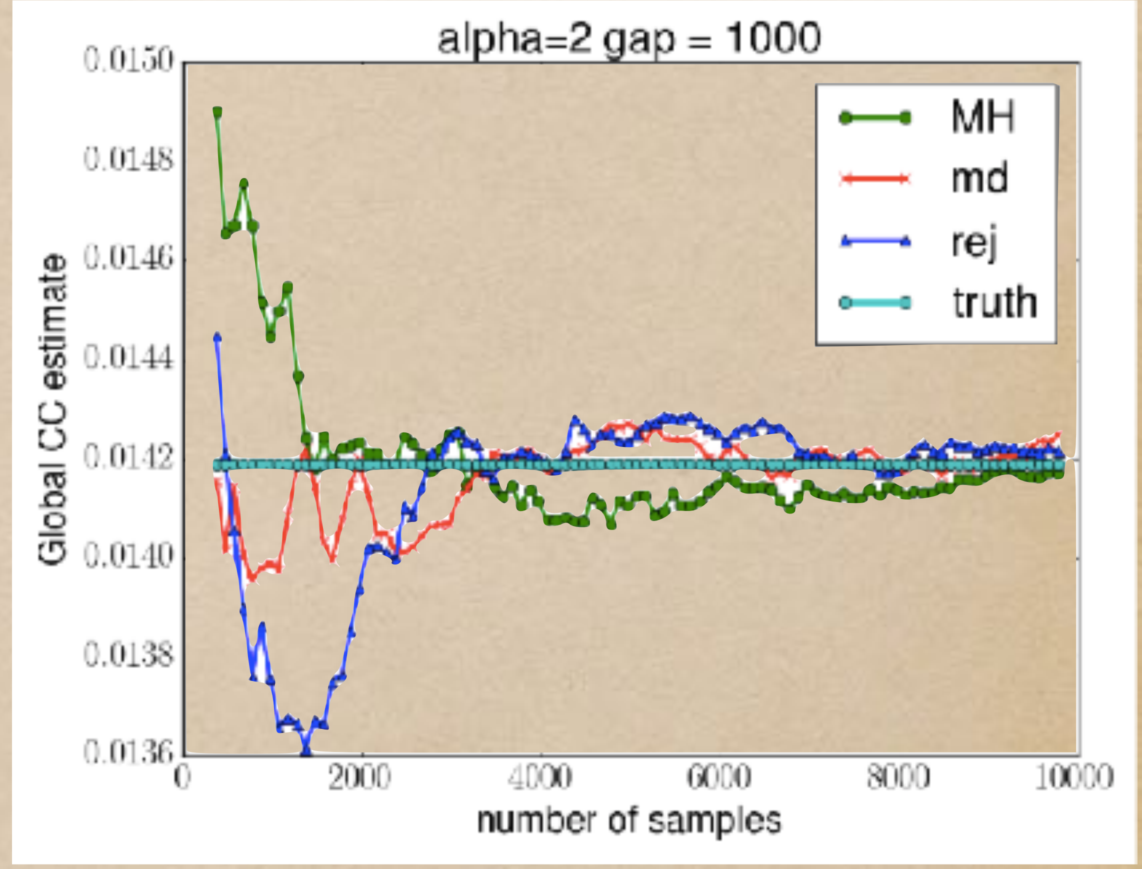
Results


email-Enron alpha = 0
MH
md
rej
MH+
l1 difference
avg number of distinct queries
0.0
0.2
0.4
0.6
0.8
1.0
0
20000
40000
60000
80000
100000

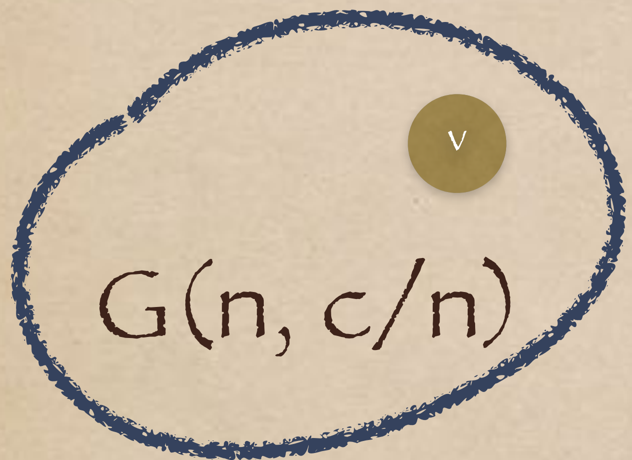

alpha=0 gap = 1000
MH
md
rej
Relative error in estimating n
number of samples
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
0
1000
2000
3000
4000
5000

Results (contd)





Other distributions



$d(v) = n^{1/(1+\varepsilon) + \delta}$

constant conductance

Claim. For $D = D_{1+\varepsilon}$ and for MH, $E[\text{steps}] \geq \boldsymbol{\Omega}(\text{poly}(n))$

Proof. A random walk will take time $n^{1 - 1/(1+\varepsilon) - \delta}$ to even visit the high degree node, so the MH algorithm will take this much time

Estimating parameters

Estimating n = #nodes

- Birthday paradox: expected #collisions in k uniform random samples is roughly $k^2/(2n)$
- Collision-counting (Katzir, Liberty, Somekh)
  - Sample nodes proportional to degree
  - Let $x_1,\ldots,x_k$ be the samples and let $d_i = \deg(x_i)$
  - Output $(\sum d_i)\ (\sum 1/d_i)$ / #collisions

Collision counting

E[#collisions] = ${}_kC_2 \cdot \sum (d_i/2m)^2$

Theorem. To get a relative estimate, #samples can be written as a function of (certain norms of) the degree distribution

- If graph is regular, then $O(\sqrt{n})$ samples suffice
- If graph has Zipfian degrees with parameter 2, then $O(n^{1/4})$ samples suffice

Can use return times (Cooper, Radzik, Siantos)

Estimating average degree

How to estimate average degree $d_{avg} = m/n$?

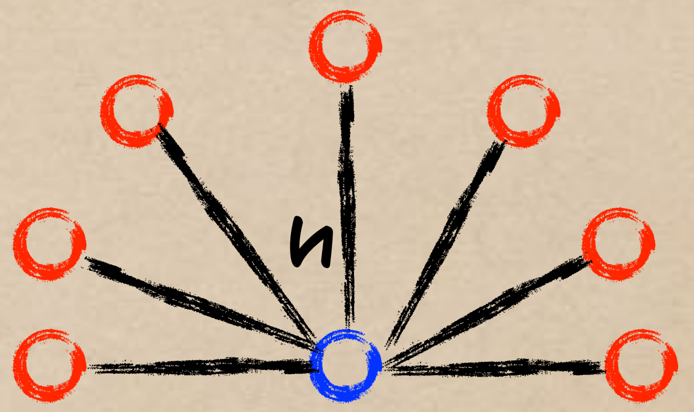
- Estimate n and m using collision-counting
  - Uses $O(\sqrt{m} + \sqrt{n})$ samples
- Estimate using just node collisions
  - Output $k^2 / 2n(\sum Collision_{ij}^u / deg(u))$
  - Uses $O(\sqrt{(n\, d_{avg}/d_{min})})$ samples
- Similarly can use just edge collisions

A natural algorithm

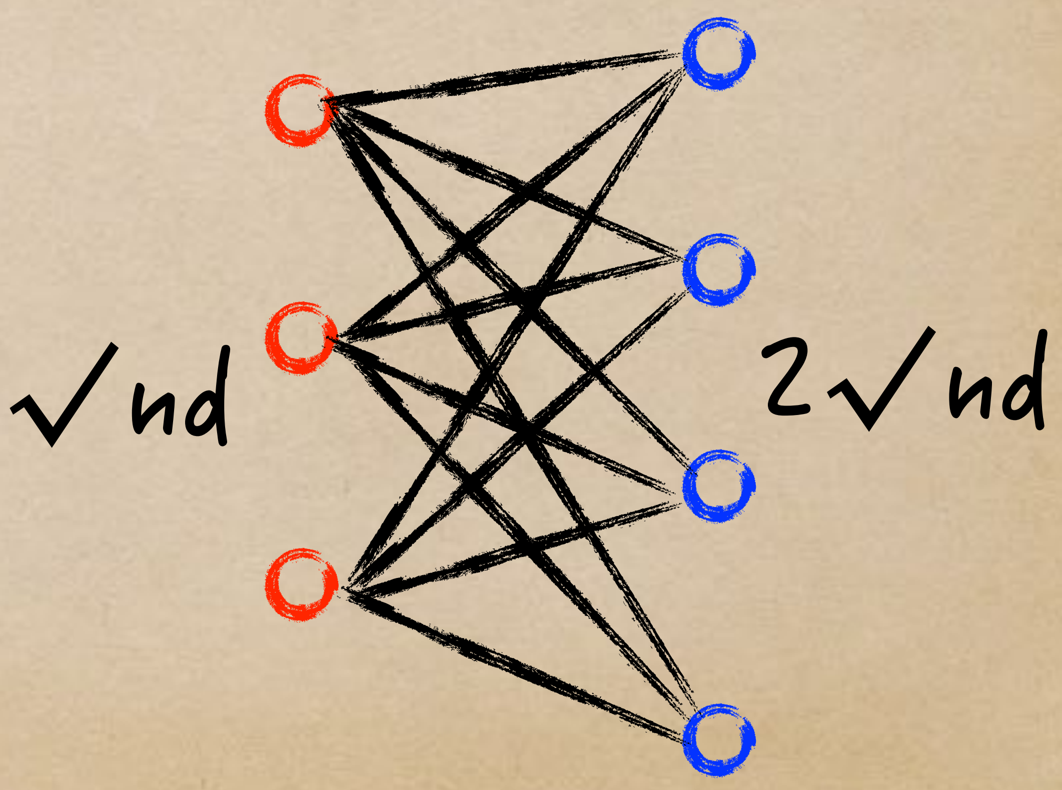
- Algorithm:
    - Sample nodes uniformly at random
    - Output the average of their degrees
- Theorem (Feige). If #samples is $O(\sqrt{n}/L)$, where $L < d_{avg}$, then it is a $(2+\epsilon)$-estimate

Limitations

- Naive bound will involve maximum degree



- Cannot get better than a 2-approximation
- This bound is tight



A different estimator

Goldreich, Ron

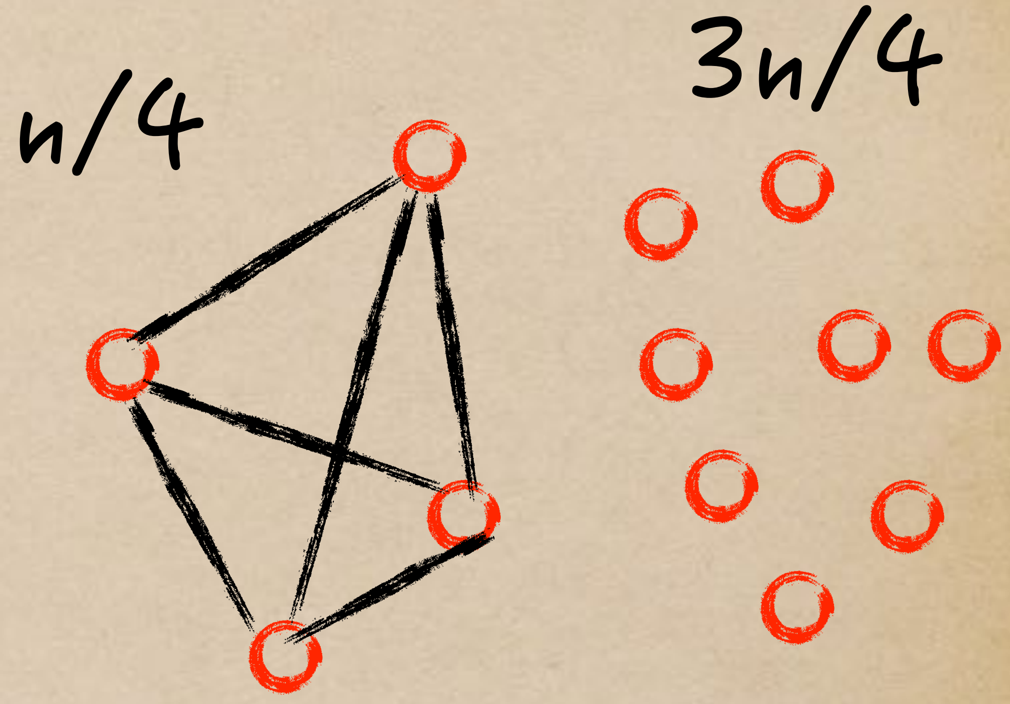
- Bucket uniformly sampled nodes by degrees
- Discard small buckets (high variance)
    - Estimator is not unbiased

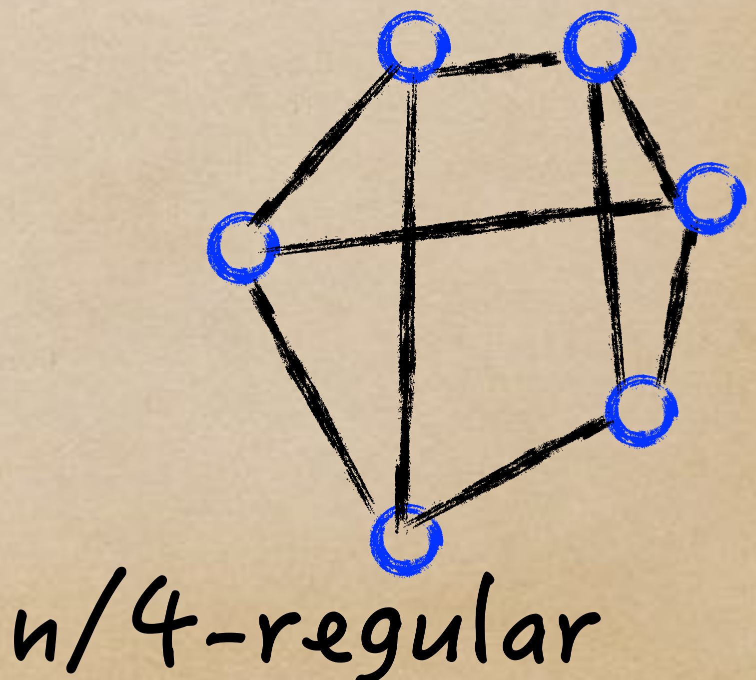
If a random neighbor is available for a node

Theorem. If #samples is $O(\sqrt{n/L})$, where $L < d_{avg}$, then it is a $(1+\epsilon)$-estimate

Can we do better?

- Sample lower bound of $\Omega(\sqrt{n})$
  - Uniform sampling
- What about non-uniform sampling?
  - Eg, degree-biased





Boosting low degrees

- Uniform: harsh for high-degrees
- Degree-biased: harsh for low-degrees
  - How to boost the degrees?
- Sample nodes with probability proportional to degree + smoothing constant
  - Sampling still random-walk friendly
  - How to choose the smoothing constant?

Algorithm: Three steps

- Coarse estimator: Gets constant approximation
- Refined estimator: Gets arbitrary approximation
- Combined estimator:
  - Run the coarse estimator
  - Use coarse estimate as the smoothing constant and run the refined estimator

Refined estimator

Given a coarse estimate c, sample k nodes $x_1,\ldots,x_k$ with probability proportional to degree + c, and output

$$\frac{\sum d_i/(d_i + c)}{\sum 1/(d_i + c)}$$

A

B

$E[A] / E[B] = d_{avg}$

Key property

Theorem. If $c = \alpha d_{avg}$ and $k = (1+\alpha)/\epsilon^2$, then Refined Estimator outputs a $(1+\epsilon)$-estimate

Proof sketch:

Show A and B are concentrated

- Analyze second moment and use Bernstein inequality
- B needs the coarse estimate:

$|B - E[B]| < 2/(d_{min} + c)$

Other properties

- Bias and variance are bounded
  - Bias at most $(\alpha d_{avg} + d_{avg}/\alpha)/k + o(1/k)$
  - Small if α is small
- Random walk version
  - Sample complexity in terms of eigenvalue gap

Coarse estimator

Guess and verify

For c in {1, 2, 4, 8, ... }

- Sample nodes with probability proportional to degree + c
- If the fraction of low-degree nodes (ie, degree below c) is more than 5/12, return c as a coarse approximation

Why does this work?

If $c = \alpha d_{avg}$, then

$(\alpha-1)/(\alpha+1) < \Pr[d_i \leq c] < 2\alpha/(\alpha+1)$

Using this, can show that

- $c < d_{avg}/3 \Rightarrow$ fraction of low-degree nodes is $< 5/12$
- $c > 3d_{avg} \Rightarrow$ fraction of low-degree nodes is $> 5/12$

Final bound

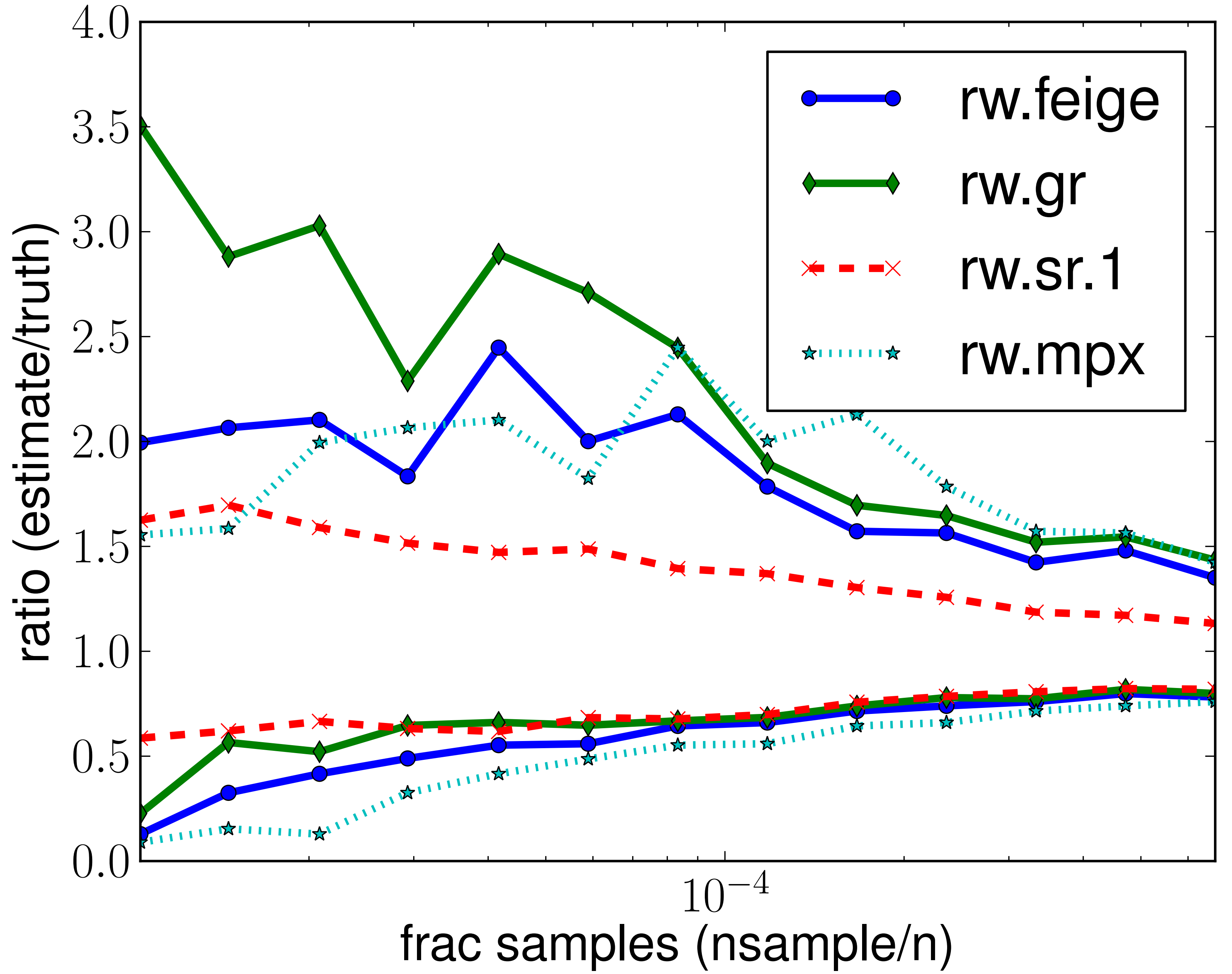
Theorem. Can $(1+\epsilon)$-estimate the average degree, wp $1-\delta$, by using

$$(\log U \log\log U + 1/\epsilon^2) \log 1/\delta$$

degree-biased node samples, where U ($< n$) is an upper bound on the maximum degree

Experiments

- SNAP (Skitter, DBLP, LiveJournal, Orkut)



Summary

- Random walks are powerful
- Bounds on generating a uniform node
  - Can extend to other distributions on V
- A better notion of mixing time for social graphs
  - Average-case notion?
- Power of non-uniform sampling
  - Other estimation problems

Thank you!

Questions/Comments: ravi.k53 @ gmail