Sampling a Uniform Node

Ravi Kumar

Google

Acknowledgments

- Joint work with Flavio Chierichetti, Anirban Dasgupta, Silvio Lattanzi, Tamas Sarlos
- To appear in WWW 2016

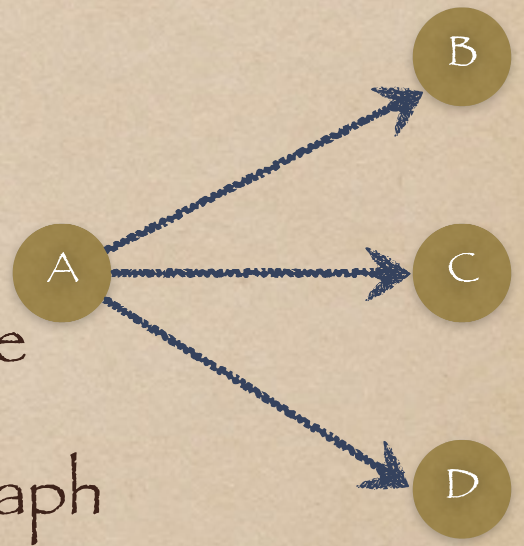
Sampling

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

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
- This access model supports random walks on the graph
- Querying is an expensive operation
  - Algorithms should minimize number of queries



Problem definition

- $G = (V, E)$ 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
  - $D_0(v) = 1/n$

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

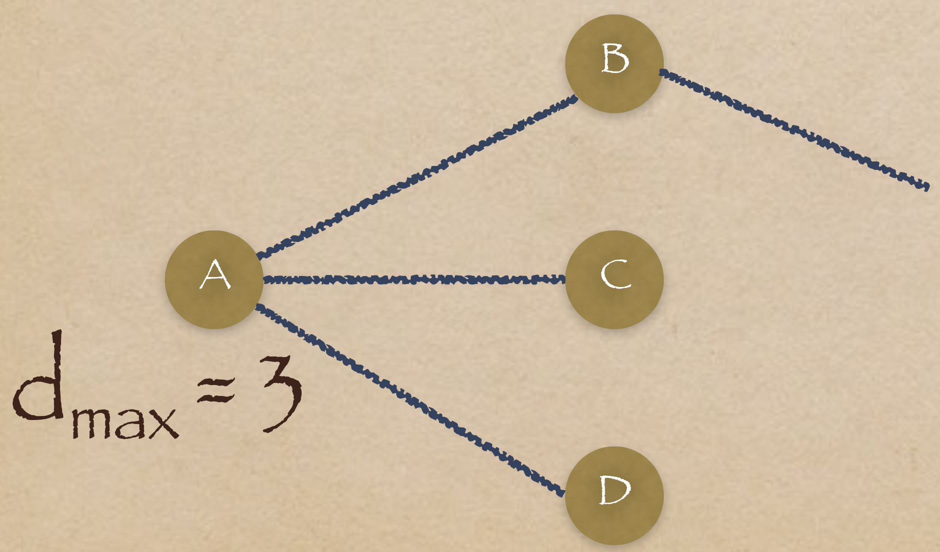
$$\Sigma_u \Pr[U = u] \times 1/d(u) = \Sigma_u d(u)/(2m) \times 1/d(u)$$

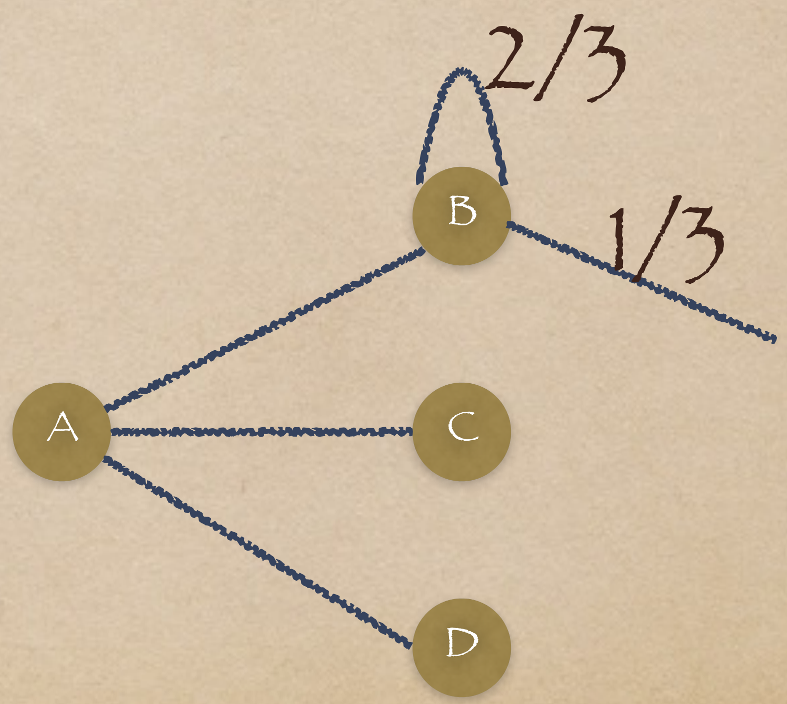
$$= \Sigma_u 1/(2m) = n / 2m = 1/d_{avg}$$

Repeat this d_{avg} times to obtain a sample

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

$$\frac{\Sigma_{uv} (f(u) - f(v))^2 d(u)\, d(v)}{\Sigma_{uv} (f(u) - f(v))^2} \geq (1/2)\, d_{avg}$$

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

Metropolis-Hastings (MH) walk

- 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

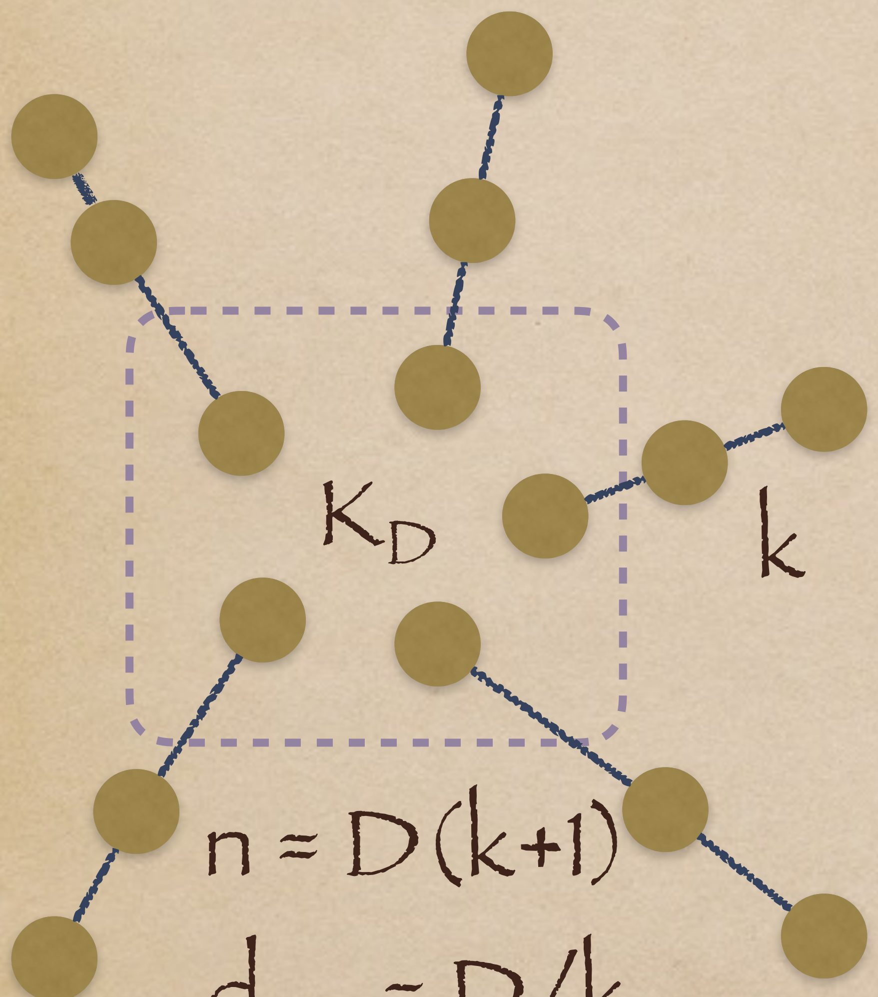
$$\Pr[u \to v] = 1/d(u) \cdot \min(1, d(u)/d(v)) = 1/\max(d(u), d(v))$$

MH Analysis

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

Proof. Use the variational characterization and steps as before

Tightness



$n = D(k+1)$

$d_{avg} = D/k$

$d_{max} = D$

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

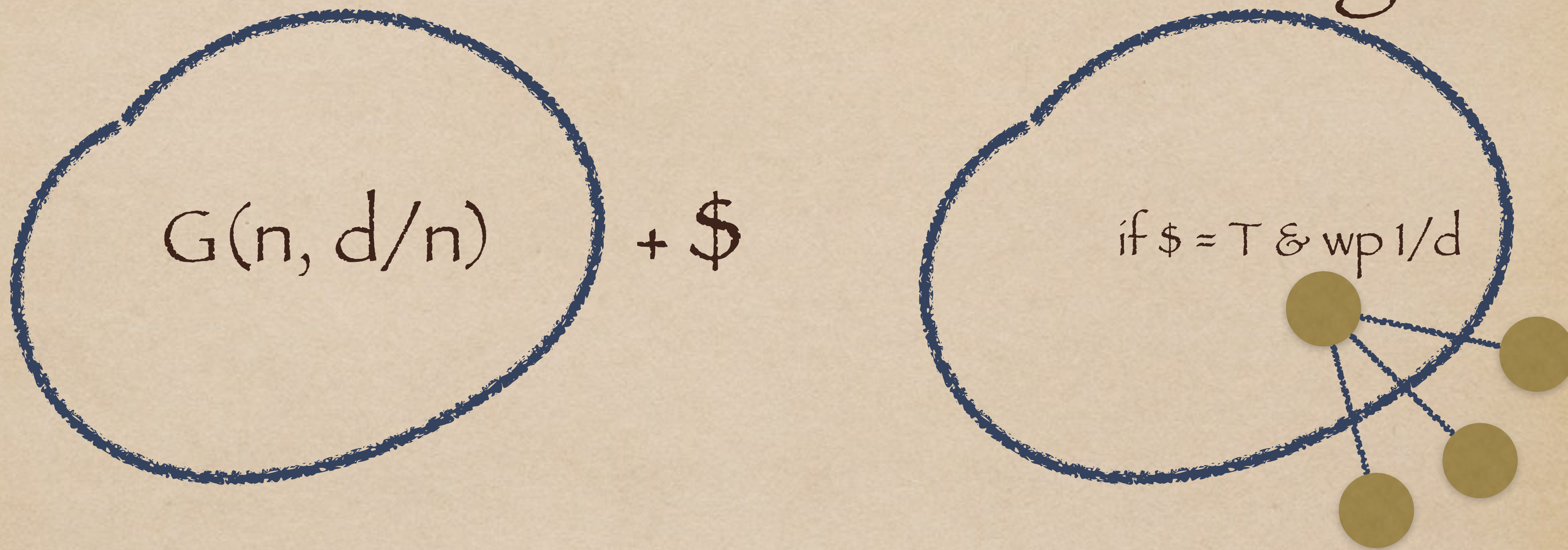
Claim. For MD, $E[\text{steps}] \geq \Omega(t_{mix} d_{max})$

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

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

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

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 $= o(d) \Rightarrow$ query only unchanged nodes wp $1 - o(1)$

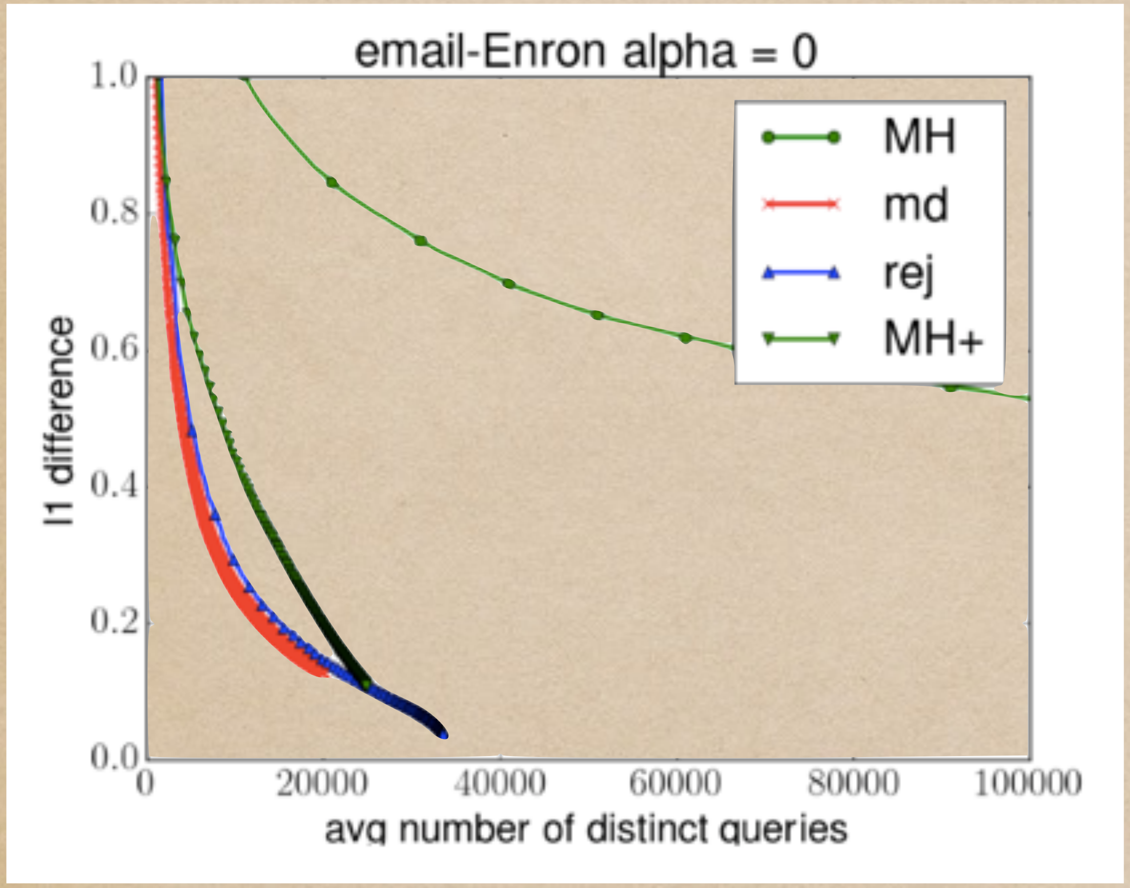
Lower bounds: $\boldsymbol{\Omega}(t_{mix})$

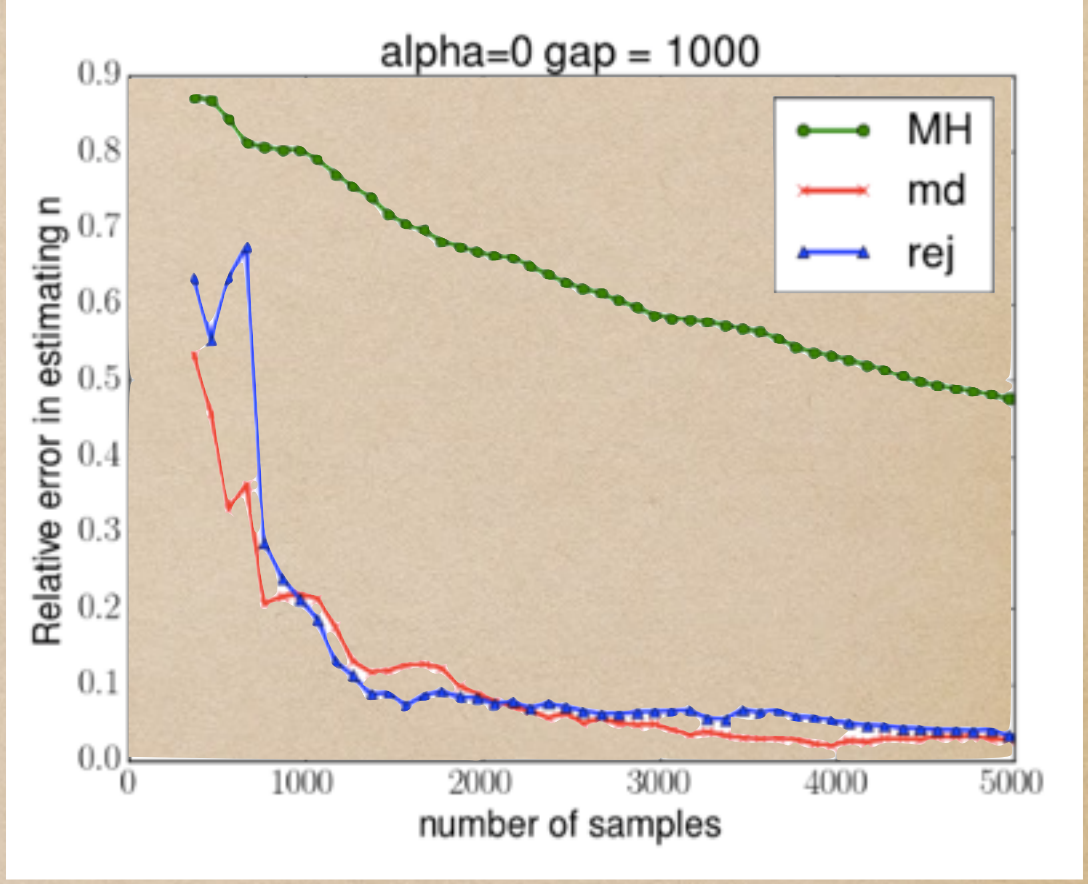
Claim. Any algorithm for D_0 must issue $\boldsymbol{\Omega}(t_{mix})$ queries

Experiments

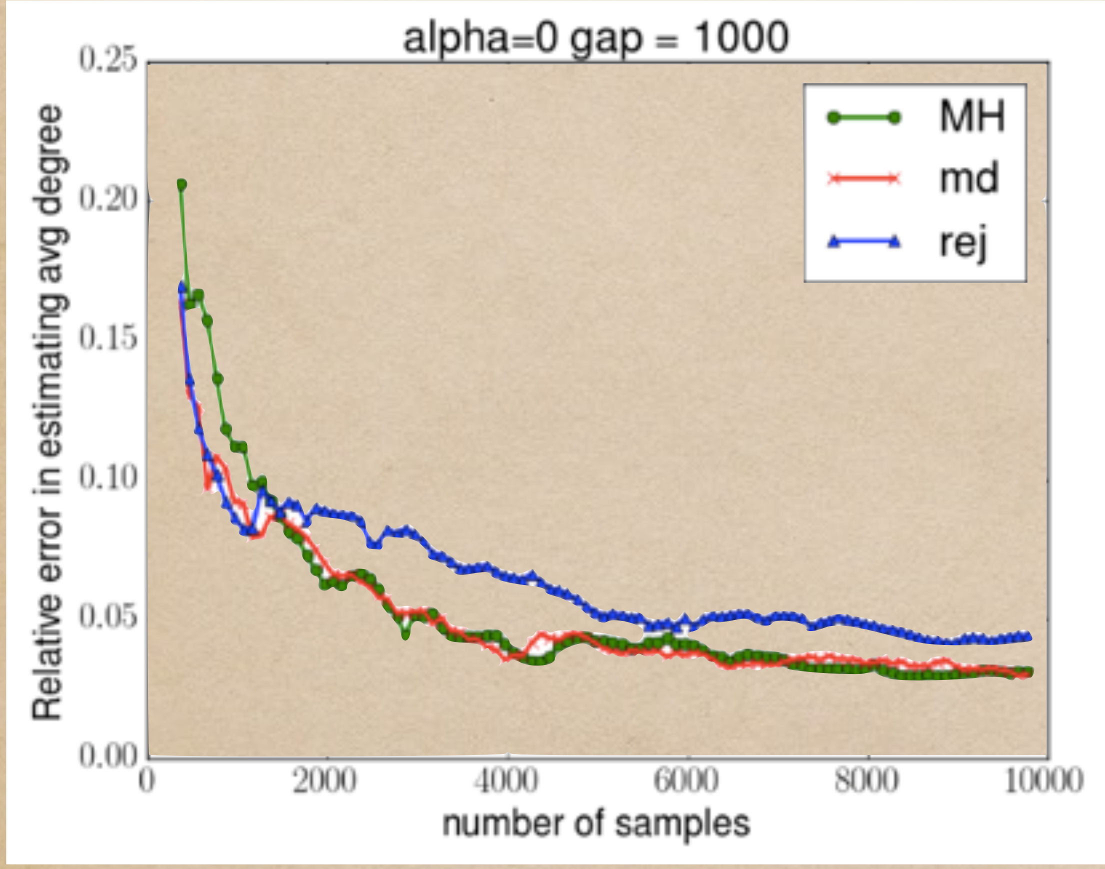
- Uniformity of the samples
  - Strict criterion
- Quality of estimators based on samples
  - Size of the network
  - Average degree
  - Clustering coefficient

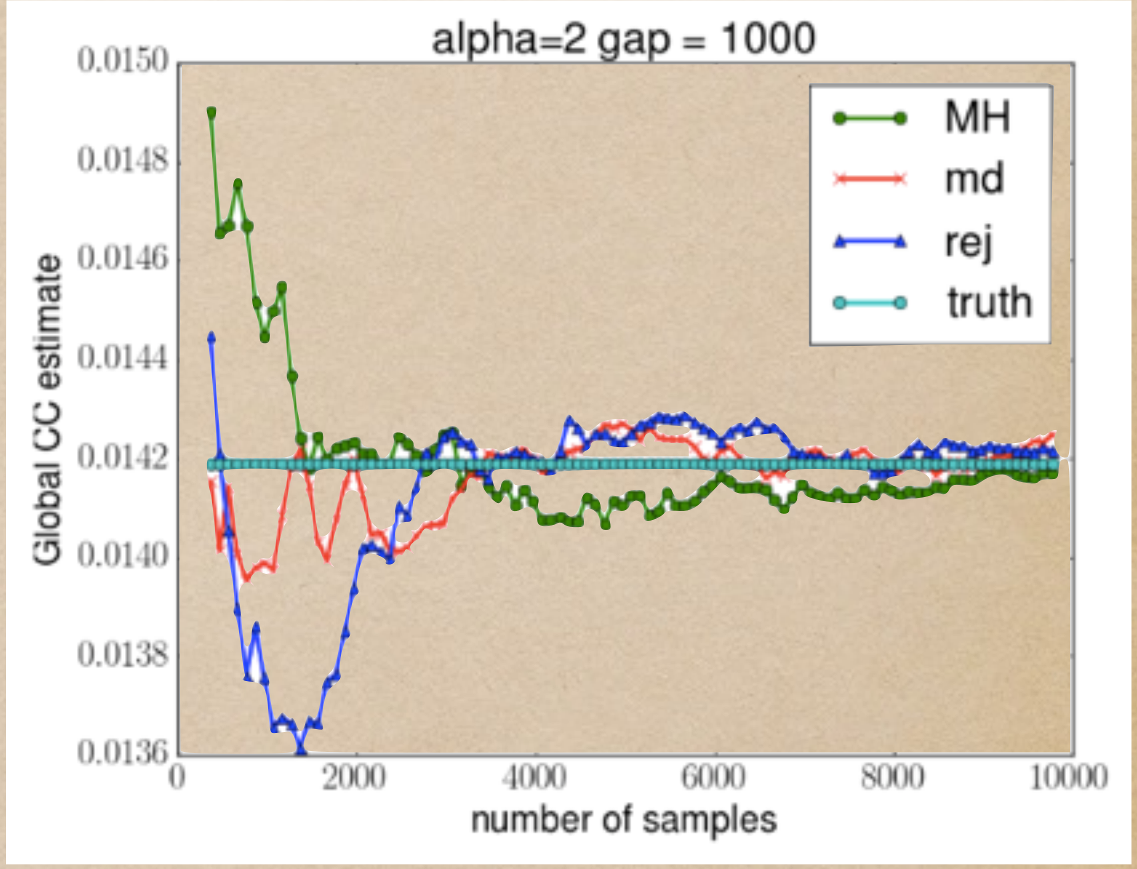
Results


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


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

Results (contd)





Summary

- Bounds on generating a uniform node
  - Can extend to other distributions on V
- Lower bound is not tight
  - Conjecture: #queries ≥ $\boldsymbol{\Omega}(d_{avg} \cdot t_{mix})$
- A better notion of mixing time for social graphs
  - Average-case notion?

Thank you!

Questions/Comments: ravi.k53 @ gmail