Private Analysis of Graph Structure

Grigory Yaroslavtsev
http://grigory.us

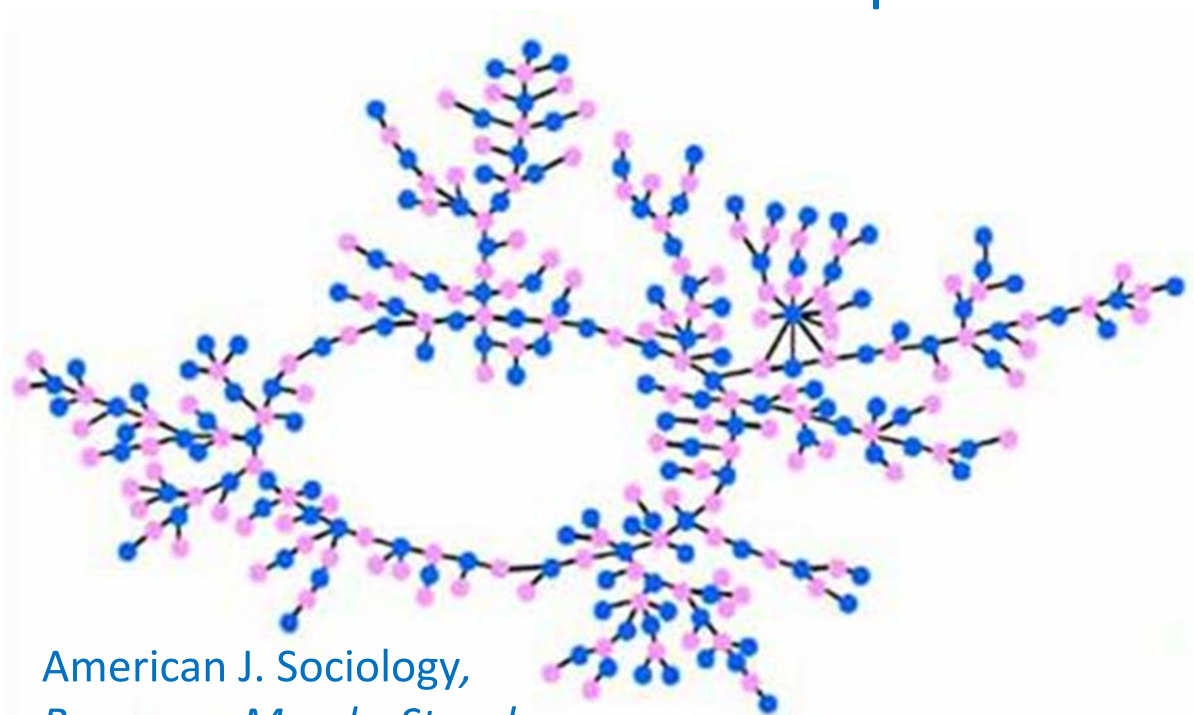
With Vishesh Karwa, Sofya Raskhodnikova and Adam Smith

Pennsylvania State University

Publishing network data

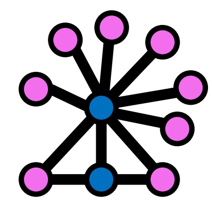
Many data sets can be represented as a graph:

- Friendship in online social network
- Financial transactions
- Romantic relationships



American J. Sociology, *Bearman, Moody, Stovel*

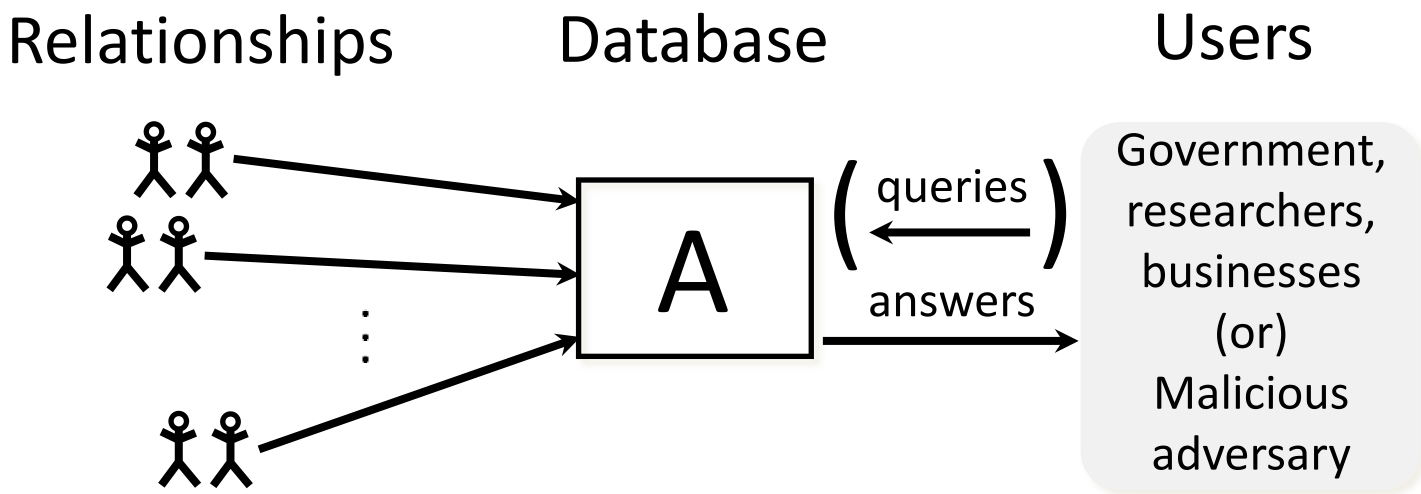
- Publish information about a graph
- Preserve privacy of relationships



Naïve approach: anonymization

Publishing network data

Goal: Publish structural information about a graph

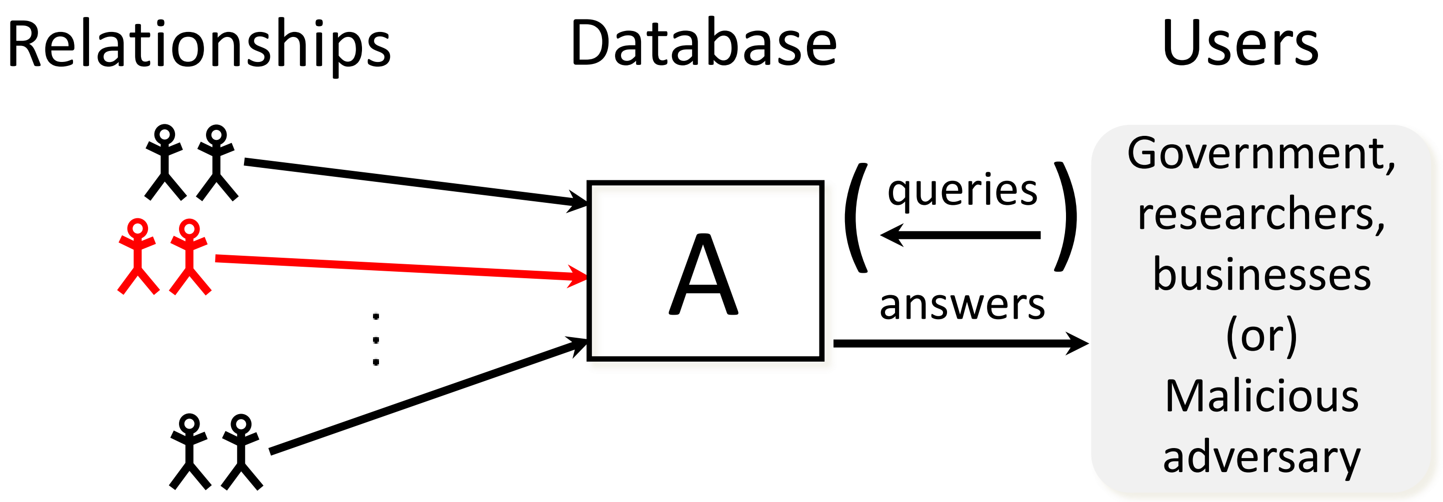


- Anonymization not sufficient [Backström, Dwork, Kleinberg '07, Narayanan, Shmatikov '09, Narayanan, Shi, Rubinstein '11]
- **Ideal:** Algorithms with rigorous privacy guarantee, no assumptions about attacker's prior information/algorithm

Differential privacy

[Dwork, McSherry, Nissim, Smith '06]

- Limits **incremental** information by hiding presence/absence of an individual relationship



- **Neighbors**: Graphs G and G' that differ in one edge
- Answers on neighboring graphs should be similar

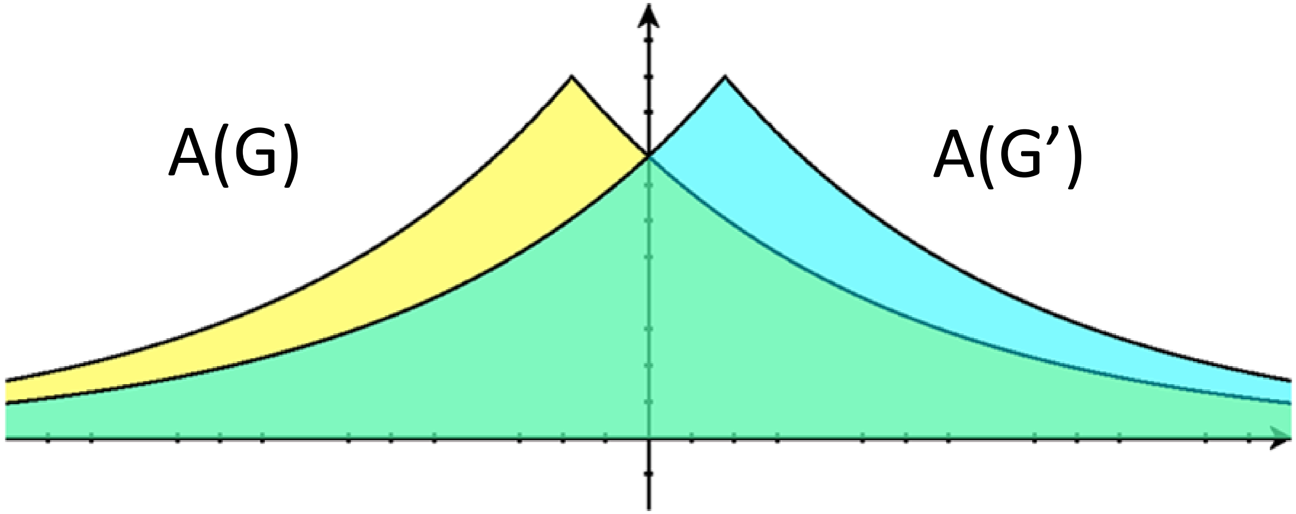
Differential privacy for relationships

<u>ϵ-differential privacy (edge privacy)</u>

For all pairs of neighbors G, G' and all events $\boldsymbol{S}$:

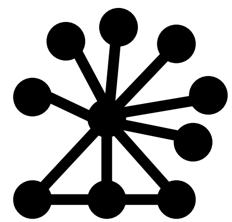
$$Pr[A(G) \in \boldsymbol{S}] \leq e^{\epsilon} \Pr[A(G') \in \boldsymbol{S}]$$

- Probability is over the randomness of A
- Definition requires that the distributions are close:

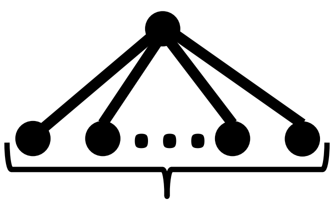


Subgraph counts

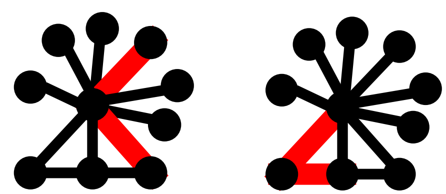
For graphs G and H: # of occurrences of H in G

Example: 

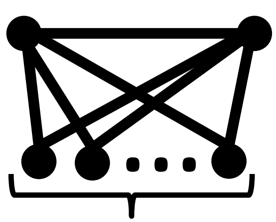
k-star



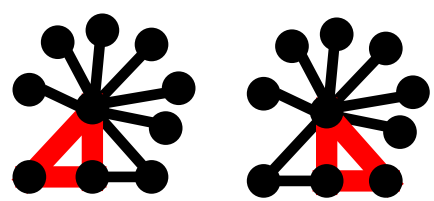
k

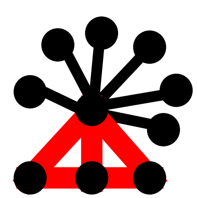
2-star:  Total: 40

k-triangle



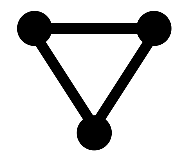
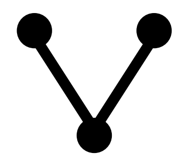
k

Triangle:  Total: 2

2-triangle:  Total: 1

Subgraph counts

- Subgraph counts are used in:
  - Exponential random graph models
  - Descriptive graph statistics, e.g.:

Clustering coefficient = #  / # 

- **Our focus:** efficient differentially private algorithms for releasing subgraph counts

Previous work

- Smooth Sensitivity [Nissim, Raskhodnikova, Smith '07]
  - Differentially private algorithm for triangles
  - **Open:** private algorithms for other subgraphs?
- Private queries with joins [Rastogi, Hay, Miklau, Suciu '09]
  - Works for a wide range of subgraphs
  - Weaker privacy guarantee, applies only for specific class of adversaries
- Private degree sequence [Hay, Li, Miklau, Jensen '09]
  - Guarantees differential privacy
  - Works for k-stars, but not for other subgraphs

Laplace Mechanism and Sensitivity

[Dwork, McSherry, Nissim, Smith '06]

- Add noise: mean = 0, standard deviation $\sim(\boldsymbol{S_f}/\boldsymbol{\epsilon})$, where $\boldsymbol{S_f}$ is **sensitivity** => $\boldsymbol{\epsilon}$-differential privacy:

$$f'(G) = f(G) + Lap(\boldsymbol{S_f}/\boldsymbol{\epsilon})$$

- **Local sensitivity** ([NRS'07], not differentially private!):

$$LS_f(G) = \max_{G': \textbf{Neighbor } of\ G} |f(G) - f(G')|$$

- Previous work (mostly): **Global sensitivity**

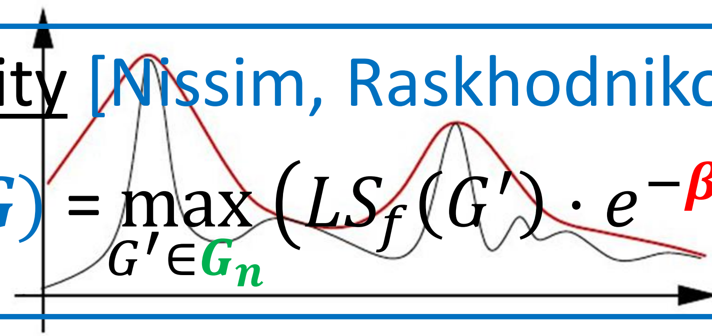
$\boldsymbol{S_f} = \boldsymbol{GS_f} = \max_{G} LS_f(G) \Rightarrow$ differentially private!

Instance-Specific Noise

$\boldsymbol{G_n}$ = set of all graphs on $\boldsymbol{n}$ vertices. **d(G,G')** = # edges in which G and G' differ.

<u>Smooth Sensitivity</u> [Nissim, Raskhodnikova, Smith '07]:

$$\boldsymbol{S^*_{f,\beta}(G)} = \max_{G'\in \boldsymbol{G_n}} \left(LS_f(G') \cdot e^{-\boldsymbol{\beta} d(\boldsymbol{G,G'})} \right)$$



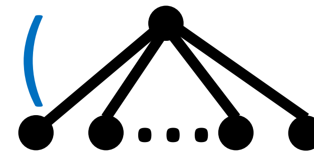
- Add Cauchy noise: median = 0, median absolute value $\propto S^*_{f,\beta}$ (G)/$\boldsymbol{\beta}$ (where $\boldsymbol{\beta} = c \cdot \boldsymbol{\epsilon}$) => $\boldsymbol{\epsilon}$-differential privacy:

$$f'(G) = f(G) + Cauchy(\boldsymbol{S^*_{f,\beta}}/\boldsymbol{\beta})$$

- Naïve computation requires exponential time
- [NRS'07]: Compute smooth sensitivity for triangles

Our contributions

- Differentially private algorithms for k-stars and k-triangles
  - Efficiently compute smooth sensitivity for k-stars
  - NP-hardness for k-triangles and k-cycles
  - Different approach for k-triangles
- Average-case analysis in G(n,p)
- Theoretical comparison with previous work
- Experimental evaluation

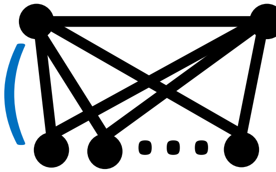
Smooth Sensitivity for k-stars ()

This paper: near-linear time algorithm for smooth sensitivity

- Algorithm also reveals structural results, e.g.:
  - **Proposition:**

    If ($\epsilon < 1$) and (maximum degree > $const \cdot k/\epsilon$)

    then (smooth sensitivity) = (local sensitivity)
- Algorithm optimal for large class of graphs
  - **Proposition**: error $>$ const ·(local sensitivity)
- Compared to [HLMJ'09] (private degree sequence):
  - Our error never worse by more than a constant factor
  - For 2-stars, our error can be better by $\Omega(\sqrt{\mathrm{n}/\epsilon})$ factor

Private Approximation to Local Sensitivity: k-triangles ()

Approximate differential privacy, (ϵ, δ)-privacy
[Dwork, Kenthapadi, McSherry, Mironov, Naor '06]:

$$\Pr[A(G) \in \boldsymbol{S}] \leq e^{\epsilon} \Pr[A(G') \in \boldsymbol{S}] + \delta$$

Idea: Private upper bound on local sensitivity ($\widetilde{LS}$).
Release: $A(G) = (\widetilde{LS}, f(G) + Lap(\widetilde{LS}/\epsilon))$.

If

- $\widetilde{LS}$ is ϵ-differentially private and
- $\Pr\left[\widetilde{LS} \geq LS\right] \geq 1 - \delta$

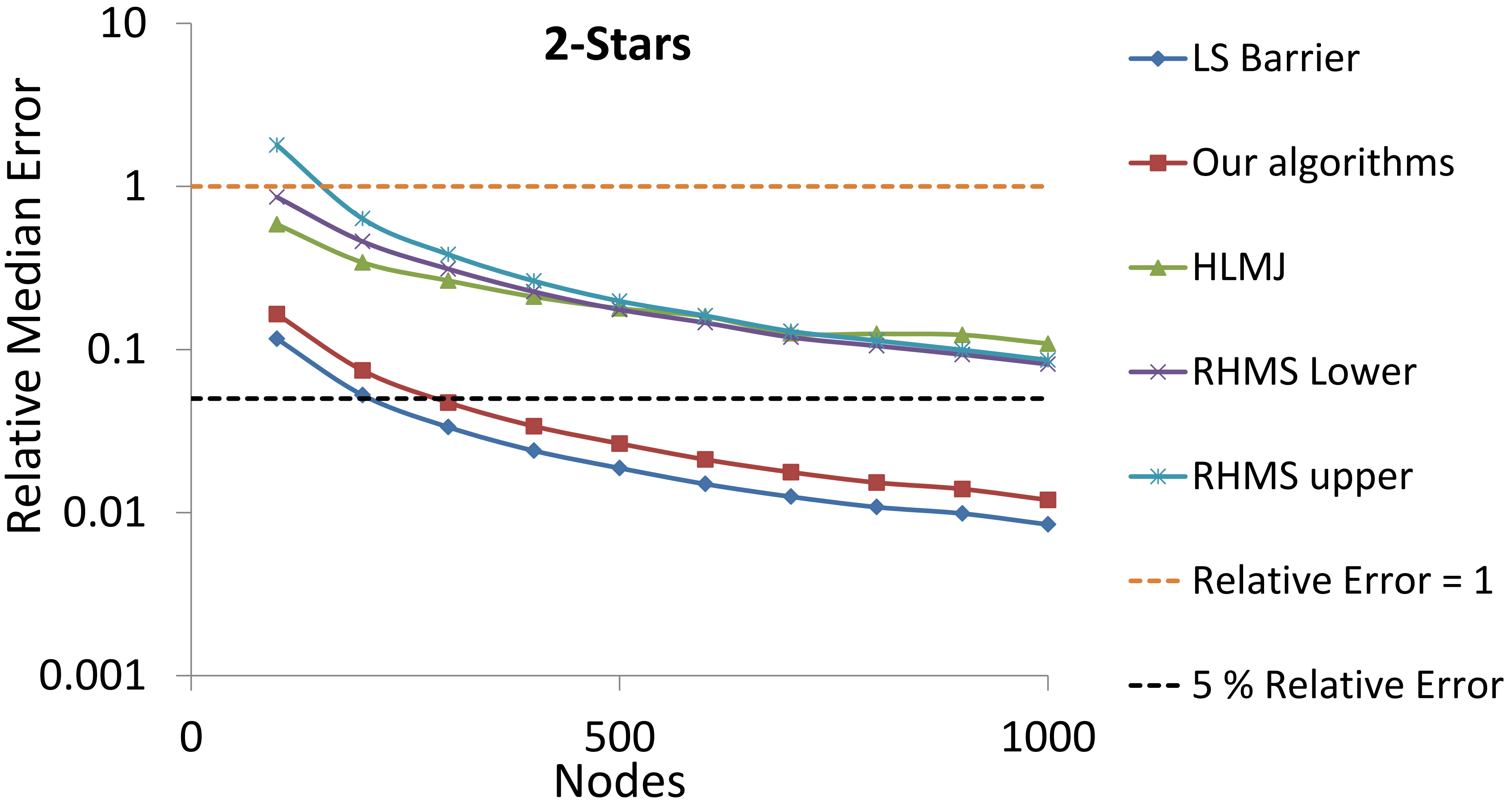
Then A is $(2\epsilon, e^{\epsilon}\delta)$-differentially private.

Evaluating our algorithms

- Theoretical evaluation in G(n,p) model
  - All of our algorithms have relative error -> 0 when the **average degree = np** grows
- Empirical evaluation
  - Synthetic graphs from G(n,p) model
  - Variety of real data sets

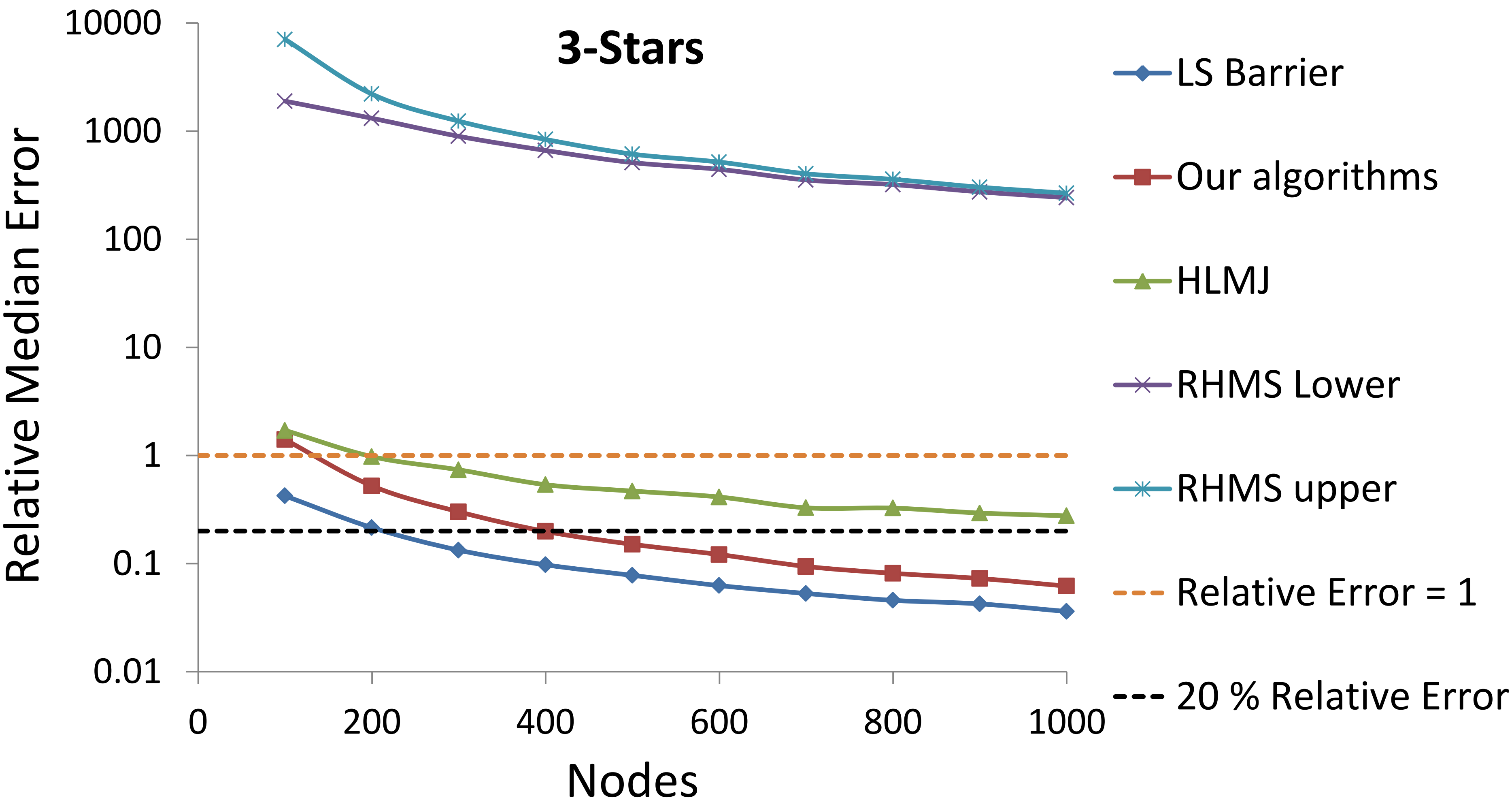
Experimental results for G(n,p)

- Comparison with previous work for $\mathbf{p} = \frac{\log n}{n}$



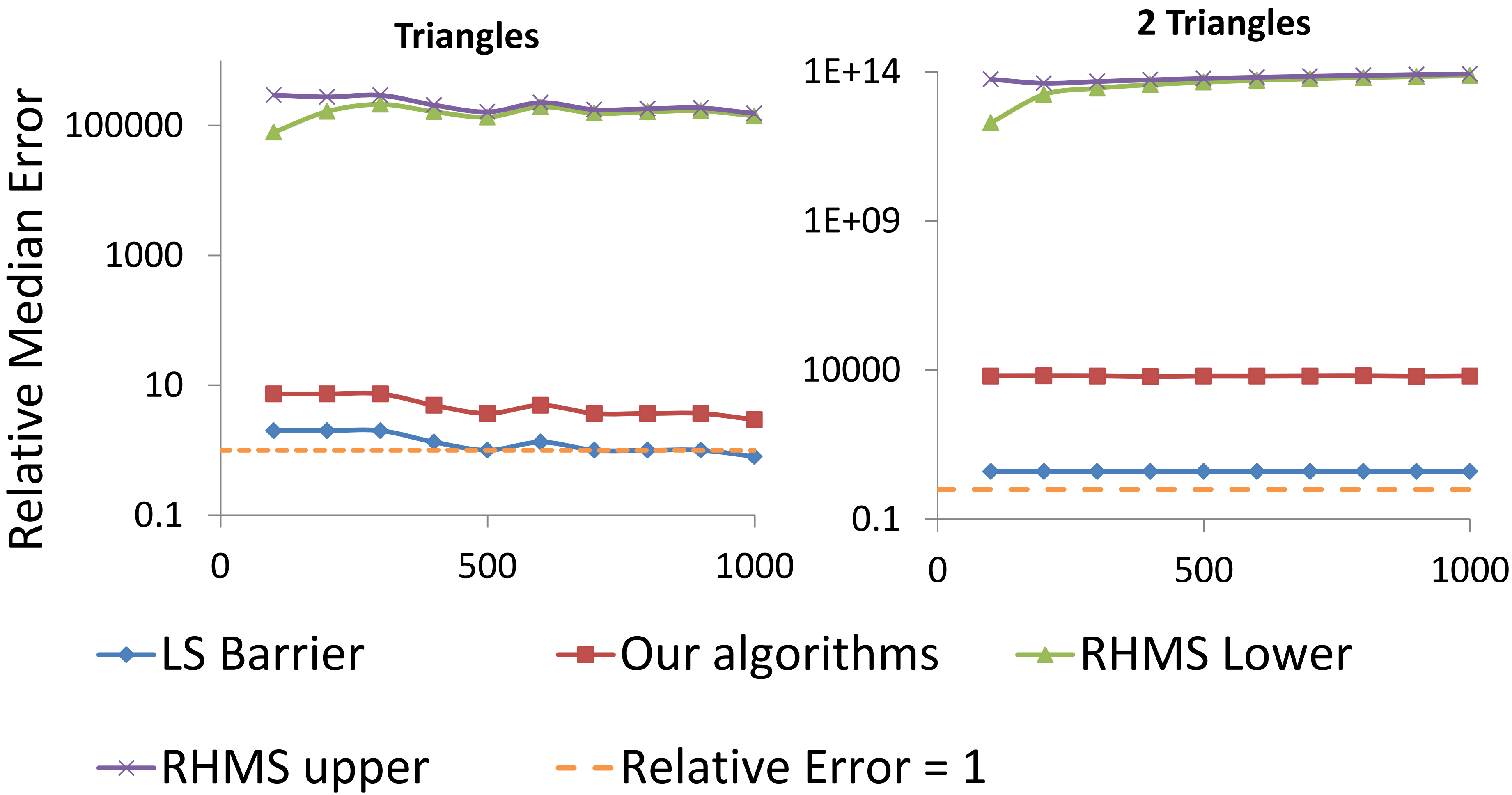
Experimental results for G(n,p)

- Comparison with previous work for $\mathbf{p} = \frac{\log n}{n}$

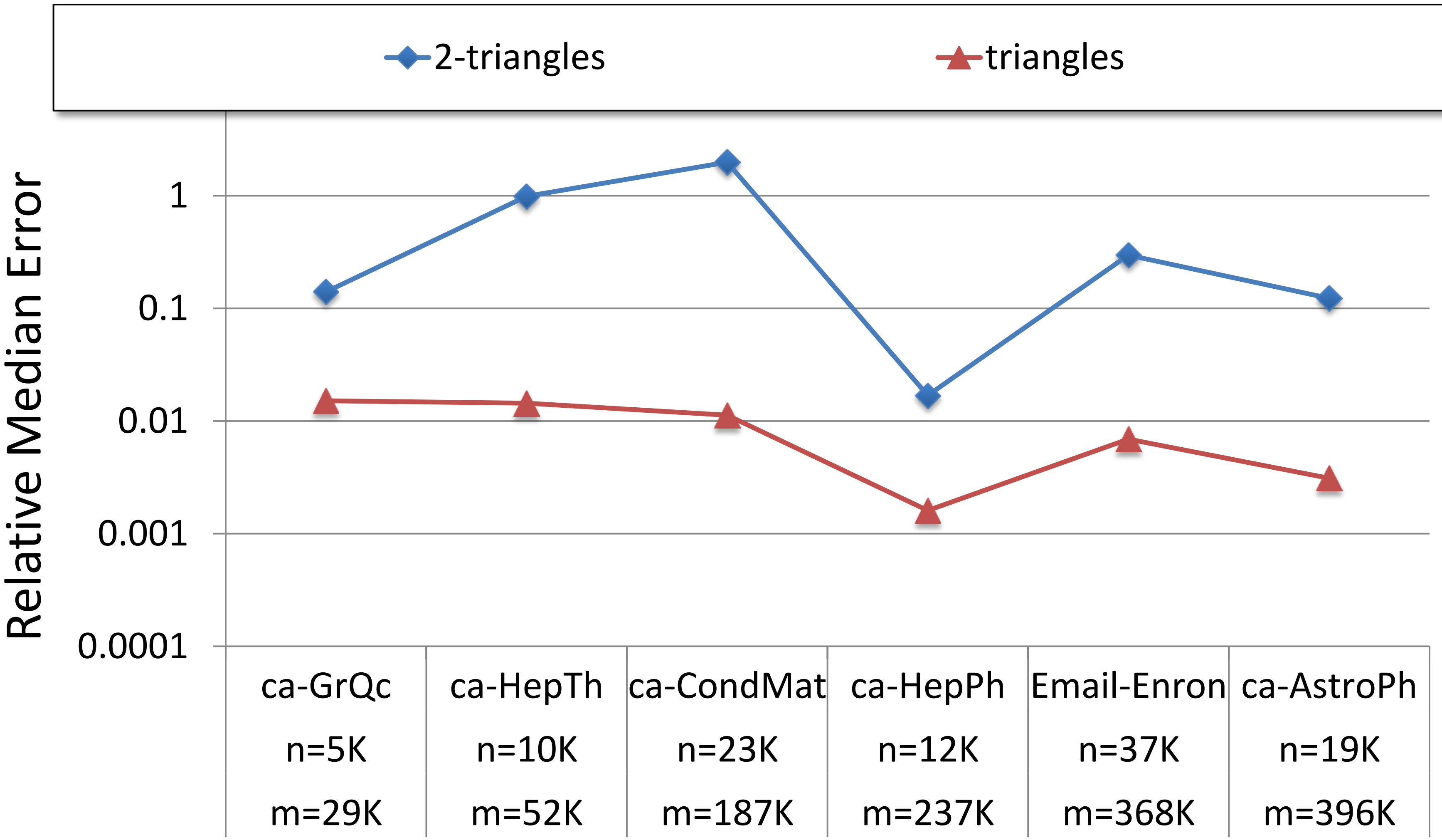


Experimental results for G(n,p)

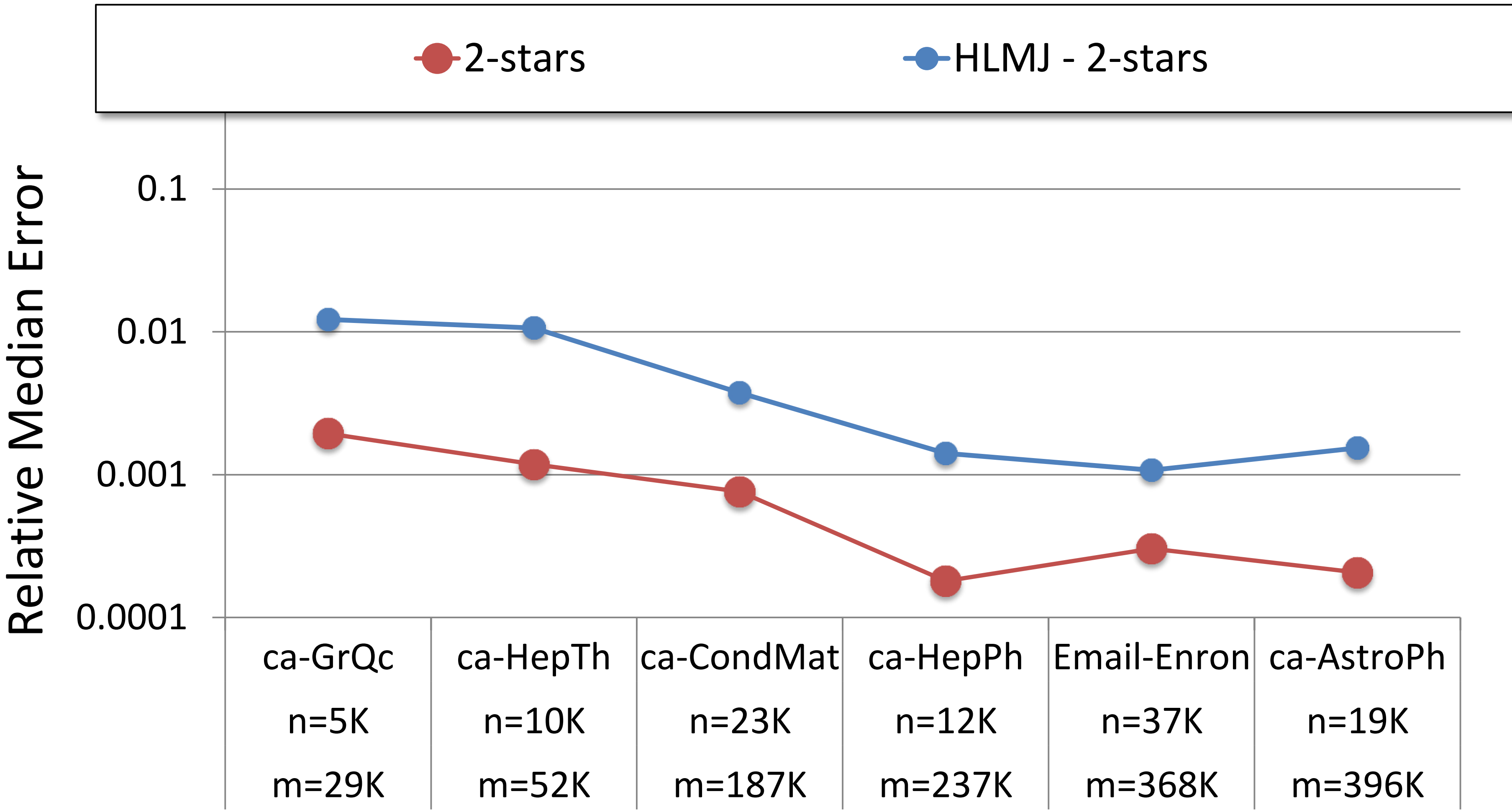
- Comparison with [RHMS'09] for $\mathbf{p} = \frac{\log n}{n}$



Experimental results (SNAP)



Experimental results (SNAP)



Summary

- Private algorithms for subgraph counts
  - Rigorous privacy guarantee (differential privacy)
  - Running time close to best algorithms for computing the subgraph counts
- Improvement in accuracy and (for some graph counts) in privacy over previous work
- Techniques:
  - Fast computation of smooth sensitivity
  - Differentially private upper bound on local sensitivity