Retrieving Top Weighted Triangles in Graphs

Paul Liu[1], Moses Charikar[2]
Stanford University
Department of Computer Science

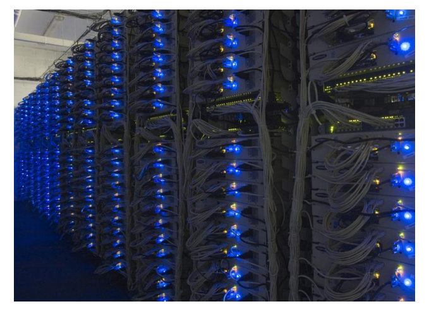
Raunak Kumar[1], Austin Benson[2]
Cornell University
Department of Computer Science

*superscript indicates author order

1. MOTIVATION

Nearly all methods for triangle mining are for *unweighted* graphs, but many networks have a natural notion of *weight*.

Payment systems
cc transactions, crypto, Venmo
weights = $ transferred



Technical infrastructure
router traffic, message passing loads
weights = traffic load

The *weight* of a triangle is the mean (or more generally p-mean) of its three edge weights. Heavy-weighted triangles have applications to *link-prediction* and *community detection*.

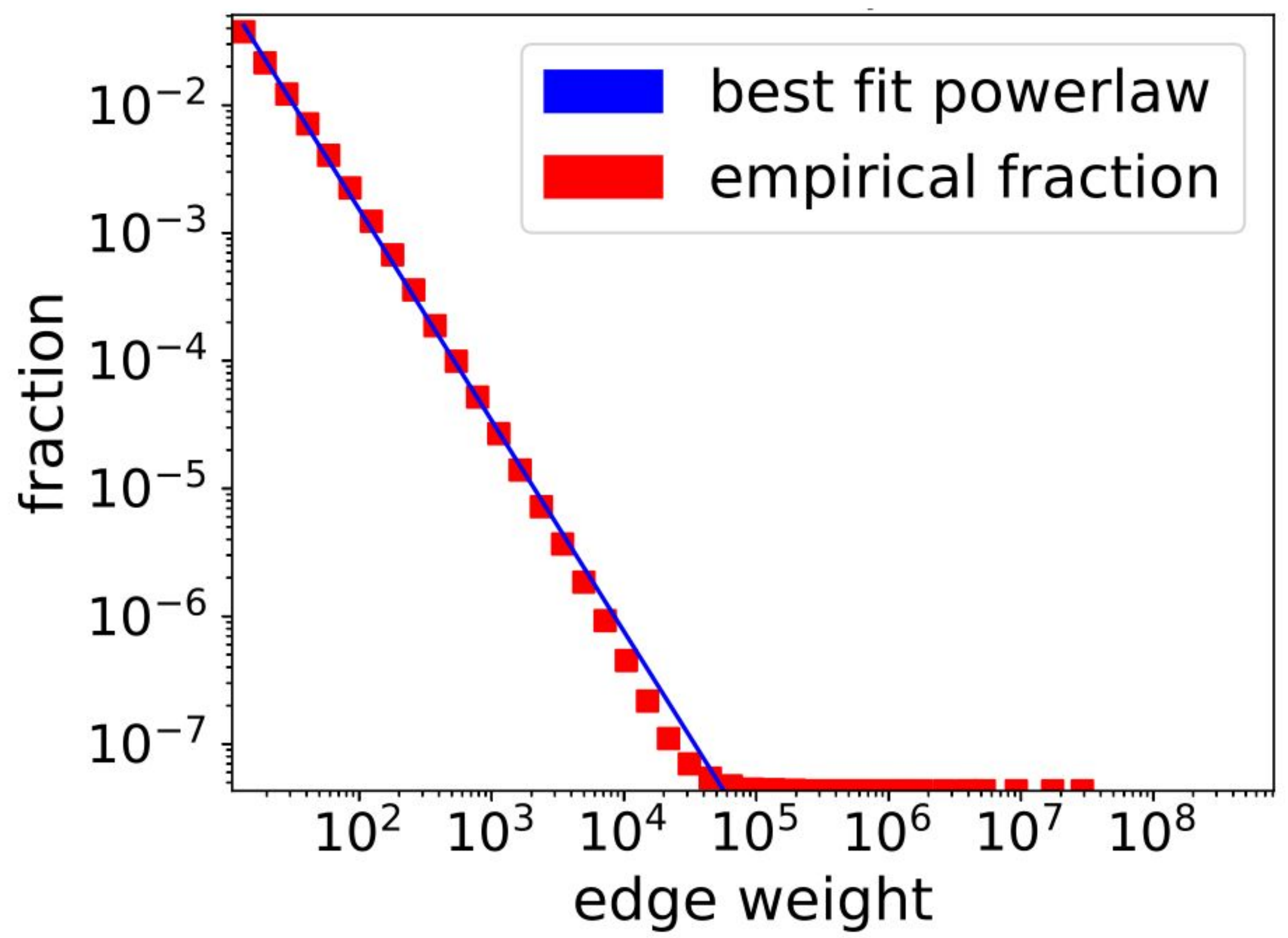
We establish a suite of algorithms for mining heavy-weighted triangles from graphs.

3. ALGORITHMIC SUMMARY

Our suite of algorithms satisfies two different regimes:

1. Exact top-k triangles for small k (<100k, sequential).
2. Approx. top-k triangles for large k (parallel sampling).

Key observation: exploit the **power-law properties** of graphs occuring in practice (Wikipedia dataset below).



2. RESULTS

Orders of magnitude faster than "fast" enumeration: top-100k triangles in a billion-edge graph in <30 s.
Results below: computing top-1000 triangles (HL = heavy-light, ES = edge samp. (64-cores, 95% acc.), BF = brute-force)

dataset	# nodes	# edges	edge weight	
			mean	max
Ethereum	38M	103M	2.8	1.9M
AMiner	93M	324M	1.3	13K
reddit-reply	8.4M	435M	1.5	165K
MAG	173M	545M	1.7	38K
Spotify	3.6M	1.9B	8.6	2.8M

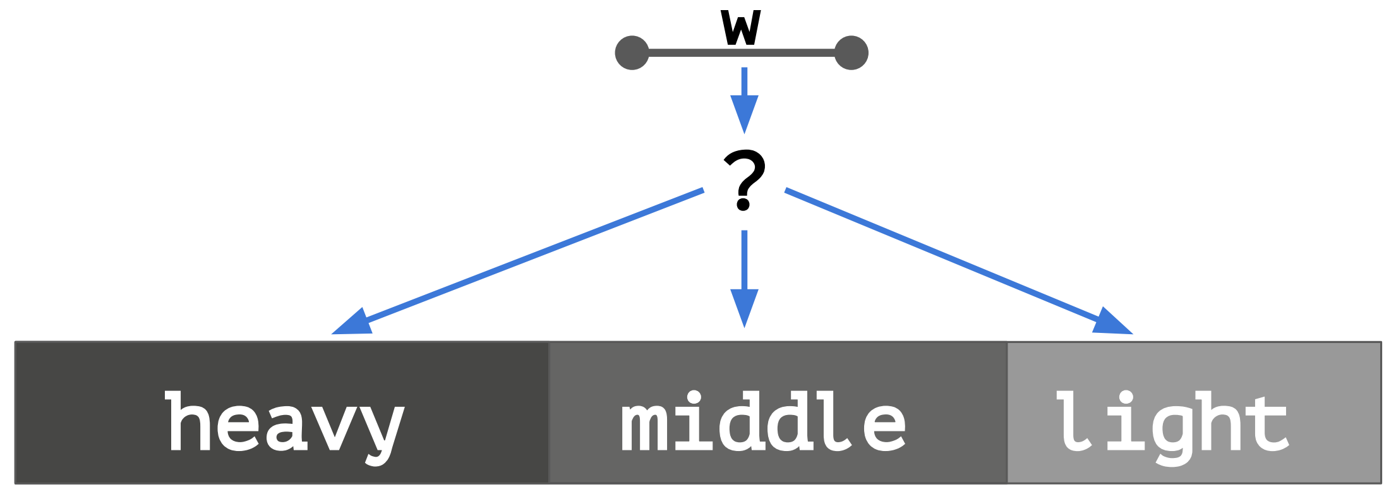
k	dataset	BF	ES	Auto-HL
1000	Ethereum	52.91	9.03	**6.94**
	Aminer	243.75	**3.72**	12.36
	reddit-reply	4047.62	5.19	**4.74**
	MAG	512.24	**4.92**	20.89
	Spotify	>86400	60.33	**30.79**

* further results for k = 100000 in paper.

4. DYNAMIC HEAVY-LIGHT

A simple algorithm: choose a threshold, partition edges into those above the threshold, and those below. Find all triangles in the subgraph induced by the heavy edges.

Problem: not correct, how to get all triangles?



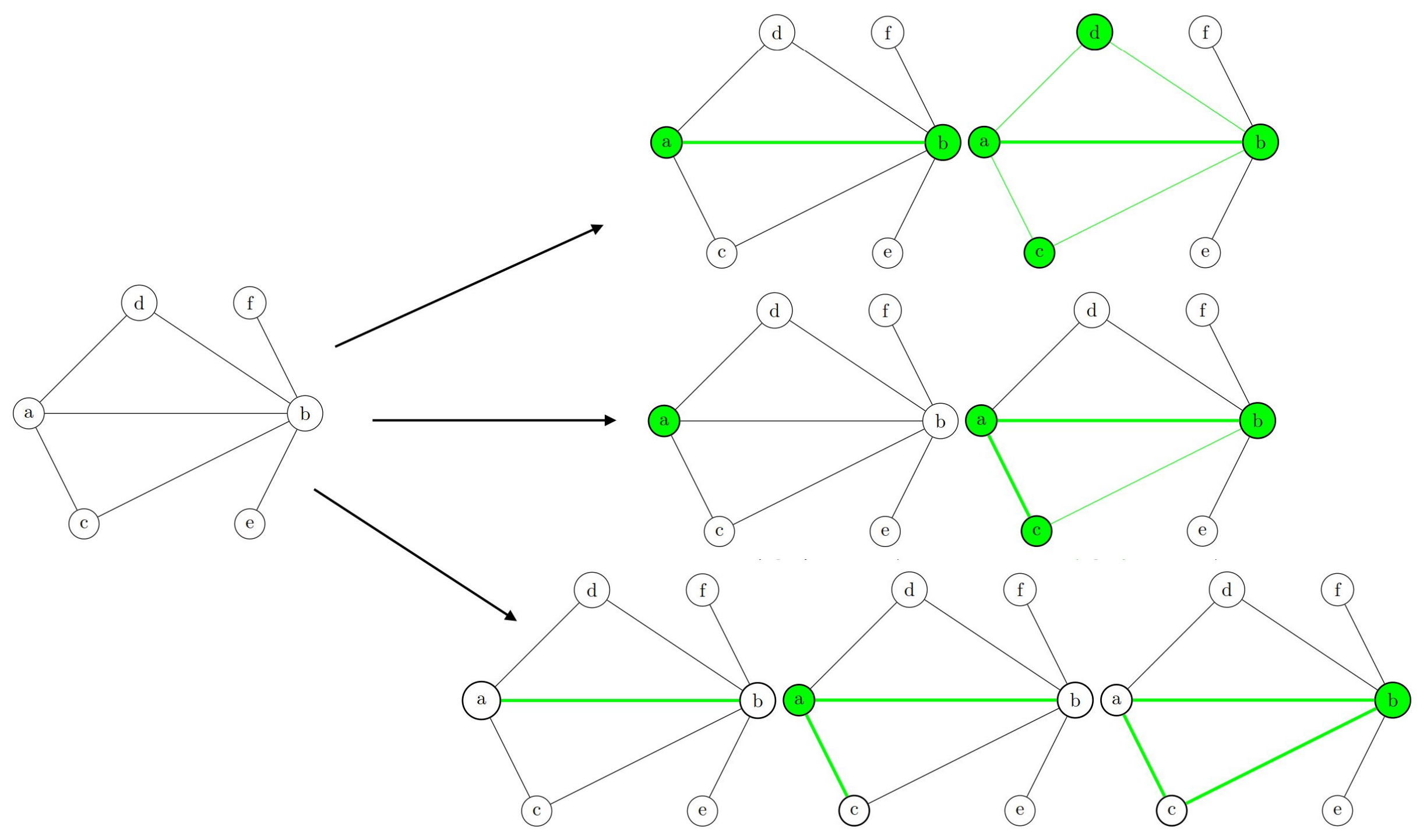
A better algorithm: use multiple thresholds that are dynamically moving. When an edge comes in, choose a threshold to put it in. **Different enumeration rules** for triangles with edges across different thresholds.

Theorem: If the input graph satisfies some common **power-law properties**, we can work out **optimal threshold values** (exact formulas in terms of the power law parameter).

5. THREE KINDS OF SAMPLING

We develop a family of sampling algorithms.

Algorithms sample an **edge**, two edges **(wedge)**, or three edges **(path)** and checks for triangles (inspired from well-known algorithms for *triangle counting*).



6. INTERESTED IN MORE?

Paper. *arXiv:1910.00692*
Code. `tinyurl.com/wsdm20-code`
✉ `paul.liu@stanford.edu`