1. MOTIVATION

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

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 occurring in practice (Wikipedia dataset below).

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?

Code. tinyurl.com/wsdm20-code
✉️ paul.liu@stanford.edu