March 29, 2018

Reax papers due tonight

Counting triangles (in graphs)

1. exact
2. approximate

Why triangles?

Simplest non-trivial small subgraph pattern (motifs)

Social Network Analysis
Clustering coefficient

\[ C_v = \frac{\text{number of triangles containing } v}{\binom{\text{degree of } v}{2}} \]

fraction of pairs of friends that are friends themselves

How do we count triangles (fast)?
Model: Undirected graph $G$

$O(1)$ time to check if $(i,j) \in E$

$O(d_v)$ time to get all of the neighbors of node $v$

First approach (brute force):
look at all triplets of nodes for $u, v, w \in V$

check if $\begin{array}{c} w \\
\end{array}$ is in the graph

$O(n^3)$  
$n = |V|$
$m = |E|
on every graph

Second approach (neighbor pairs)

Idea:

![Diagram: A graph with nodes and edges]

- Can access neighbor list efficiently

for \( v \in V 

- for neighbors \( u, w \) of \( v 

- if \( u \rightarrow w \)

- increment counter

\[ \Theta \left( \leq d_v^2 \right) \]
heavy-tailed degree distributions

\[ \implies \text{some high degree nodes in } G \]

\[ \sim n^2 \]

\[ n-1 \text{ neighbors} \]

only count once in a "smart" way
Third approach (low degree centers)

1. sort the nodes by degree
   \[ \Rightarrow \text{ordering } \sigma \]
   \[ \Rightarrow \sigma(u) \leq \sigma(v) \Rightarrow d_u \leq d_v \]

2. for each \( v \in V \)
   for each neighbor pair \( u, w \) with \( \sigma(v) \leq \sigma(u), \sigma(w) \)
   check if \( u \rightarrow w \) exists

\[ \Theta(m) \text{ time} \]
Claim: $\Theta(m^{3/2})$ worst-case for every graph

Complete graph: $m = n^2 \Rightarrow n^3 \checkmark$

Proof:

Split vertices into big or small

$\checkmark$ big if $d_v > \sqrt{m}$

$\checkmark$ small if $d_v \leq \sqrt{m}$

How much work do we do when processing small nodes?

$\forall v : d_v \leq m$

Work:

$O(\sum_{v \in V} d_v^2)$
Worst-case:
\[ d_v = \sqrt{m} \quad \forall \in \mathbb{S} \]
At most 2\(\sqrt{m} \) of these
\[ \exists d_v = 3\sqrt{m} \quad \forall \in \mathbb{S} \]
if more 2\(\sqrt{m} \) of these
\[ > \sqrt{m} \cdot 2\sqrt{m} = 2m \quad \forall \in \mathbb{S} \]
could have more nodes with smaller degree

\[
\begin{align*}
\text{d} & \quad \text{d}^2 \\
\text{m} & \quad \text{m}^2
\end{align*}
\]
\( \sum_{v \in S} d^2_v = \sum_{v \in S} m \)

\( \leq O(\sqrt{m \cdot m}) \)

\( = O(m^{3/2}) \)

What about big nodes?

At most 2 \( \sqrt{m} \) big nodes

(same reasoning as before)

When the algorithm processes a big node as a center

neighbor pairs that are
processed are also big

Bound big node work by

\[ O\left( \left| B^1 \right|^3 \right) \]

\[ \left| B^1 \right| \leq 2\sqrt{m} \]

\[ O\left( m^{3/2} \right) \]

Only really needed a big split

Degree ordering works well in practice

Can prove additional guarantees
if you assume something about the degree dist (power law) see Latapy
Approximate counting

Global clustering coeff.

\[ C = \frac{3T}{\sum_{v}^{} \left( \begin{array}{c} d_v \end{array} \right)} \]

(Seshadhri et al. 2013)

"Wedge sampling"

\[ \left( \begin{array}{c} d_v \end{array} \right) \] such wedges
\[ P(\nu) = \frac{1}{2} \frac{1}{\sum_{w} \frac{\nu}{2}} \]

(fraction of wedges centered at \( \nu \))

**Uniform WedgeSampler**

1. choose \( \nu \sim \{p_{\nu}\}_w \)

2. choose two neighbors of \( \nu \) uniformly at random

\[ \Pr(\text{select } \bigcirc) \]

\[ = P(\nu) \cdot \frac{1}{\nu} = \frac{1}{\sum_{w} \frac{\nu}{2}} \frac{1}{\nu} \]
Approx GCCF (k)

1. Sample k wedges unit at random

2. Output fraction of the wedges that induce a triangle

Why does this work well?

1. Unbiased estimator

\[ X_i = \left\{ \begin{array}{ll} 1 & \text{if wedge forms } \Delta \\ 0 & \text{o/w} \end{array} \right. \]

\[ E(X_i) = \Pr(X_i = 1) \]
\[ \Pr(\text{UAR wedge forms a D}) = \frac{3T}{\# \text{wedges}} = C \]

2 Concentration of the estimator

Hoeffding's inequality

$x_1, \ldots, x_k$ independent random vars

with $0 \leq x_j \leq 1$

\[ \overline{X} = \frac{1}{k} \sum_{j=1}^{k} x_j \]

then for any $\varepsilon \in (0, 1)$
\[
\Pr \left( | \bar{X} - \mathbb{E}(X) | \geq \varepsilon \right) \\
\geq 2 \exp(-2k\varepsilon^2)
\]

In our case, indicator RVs

\[X_i = \begin{cases} \\
1 & \text{ith wedge forms } \Sigma \\
0 & \text{o/w}
\end{cases} \]

k samples, output

\[\bar{X} = \frac{1}{k} \sum X_i \]

\(= \) fraction of sampled wedges forming a triangle

\[\mathbb{E}(\bar{X}) = C \text{ (global clustering coeff)}\]
Choose # of samples
\[ k = \frac{\log \left( \frac{2L}{\delta} \right)}{2\epsilon^2} \]
then \( |\bar{x} - c| \leq \epsilon \)
with prob. at least \( 1 - \delta \)

Can also use to get estimate on # of triangles \( T \)
\[ |\bar{x} \cdot \sum (d_w) - T| \leq \epsilon \frac{\sum (d_w)}{3} \]

Other things to notice

1. Can use uniform wedge samplers as uniform \( \Delta \) samplers
(just reject)

2. Can also get avg. ccf

\[ C = \frac{1}{n^2} \sum_{u,v} \frac{T_w}{(d_u, d_v)} \]

\[ T_w = \# D \text{ containing } w \]

(i) Pick k nodes uniformly at random

(ii) Choose uniform wedge for each node

(iii) Output fraction of closed wedges