Counting Stars and Other Small Subgraphs in Sublinear Time

Mira Goen, Dana Ron, Yuval Shavitt

presented by Marek Tesař

Let μ be a meassure defined over graphs and let G be an unknown graph over n vertices. An algorithm for estimating $\mu(G)$ is given an approximation parameter ϵ, the number of vertices n, and query access to the graph G (degree queries and neighbor queries). Algorithm output an estimate $\hat{\mu}$ of $\mu(G)$ such that with hight constant probability, $\hat{\mu} = (1 \pm \epsilon).\mu(G)$, where for $\gamma \in (0,1)$ we use the notation $a = (1 \pm \gamma).b$ to mean that $(1-\gamma).b \leq a \leq (1+\gamma).b$.

We denote $l(G)$ the number of length-2 paths in G. Set $\beta = \frac{\epsilon}{c}$ for some constant $c > 1$ and $t = \lceil \log_{(1+\beta)} n \rceil$ (so that $t = O(\frac{\log n}{\epsilon})$).

Definition For $i = 0, \ldots, t$ define

$$B_i = \{v : deg(v) \in ((1+\beta)^{i-1}, (1+\beta)^i]\}$$

Algorithm 1(Estimating the number of length-2 paths for $G = (V, E)$)

Input: ϵ and $\tilde{l}$

- 1. Let $\beta = \frac{\epsilon}{32}, t = \lceil \log_{(1+\beta)} n \rceil$, and

$$\theta_1 = \frac{\epsilon^{\frac{2}{3}} \tilde{l}^{\frac{1}{3}}}{32 t^{\frac{4}{3}}}$$

- 2. Uniformly and independently select $\Theta(\frac{n}{\theta_1} \cdot \frac{\log t}{\epsilon^2})$ vertices from V, and let S denote the multiset of selected vertices (we allow repetitions).
- 3. For $i = 0, \ldots, t$ determine $S_i = S \cap B_i$ by performing a degree query on every vertex in S.
- 4. Let $L = \{i : \frac{|S_i|}{|S|} \geq 2\frac{\theta_1}{n}\}$.

  If $\max_{i \in L}\{2\binom{(1+\beta)^{i-1}}{2} \cdot \theta_1\} > 4\tilde{l}$ then terminate.
- 5. For each $i \in L$ run Algorithm 2 to get estimates $\{\hat{e}_{i,j}\}_{j \notin L}$ for $\{|E_{i,j}|\}_{j \notin L}$.
- 6. Output

$$\hat{l} = \sum_{i \in L} n \cdot \frac{|S_i|}{|S|} \cdot \binom{(1+\beta)^i}{2} + \sum_{j \notin L} \frac{1}{2} \sum_{i \in L} \hat{e}_{i,j} \cdot ((1+\beta)^j - 1)$$

$\tilde{l}$	**Query and Time Complexity**
$\tilde{l} \le n^{\frac{3}{2}}$	$O(\frac{n}{\tilde{l}^{\frac{1}{3}}}).poly(\log n, \frac{1}{\epsilon})$
$n^{\frac{3}{2}} \le \tilde{l} \le n^2$	$O(n^{\frac{1}{2}}).poly(\log n, \frac{1}{\epsilon})$
$n^2 \le \tilde{l}$	$O(\frac{n^{\frac{3}{2}}}{\tilde{l}^{\frac{1}{2}}}).poly(\log n, \frac{1}{\epsilon})$

Theorem 0.1 *If $\frac{1}{2}l(G) \le \tilde{l} \le 2l(G)$ then with probability at least $\frac{2}{3}$, the output, $\hat{l}$, of Algorithm 1 satisfies $\hat{l} = (1 \pm \epsilon) \cdot l(G)$. The query complexity and running time of the algorithm are*

$$O(\frac{n}{\tilde{l}^{\frac{1}{3}}} + \min\{n^{\frac{1}{2}}, \frac{n^{\frac{3}{2}}}{\tilde{l}^{\frac{1}{2}}}\}) \cdot poly(\log n, \frac{1}{\epsilon})$$

.

Theorem 0.2 *Any constant-factor multiplication algorithm for the number of length-2 paths:*
1) must perform $\Omega(\frac{n}{l^{\frac{1}{3}}(G)})$ queries
2) must perform $\Omega(\sqrt{n})$ queries when the number of length-2 paths is $O(n^2)$
3) must perform $\Omega(\frac{n^{\frac{3}{2}}}{l^{\frac{1}{2}}(G)})$ queries when the number of length-2 paths is $\Omega(n^2)$

Theorem 0.3 *For $m = O(n)$ it is necessary to perform $\Omega(m)$ queries in order to distinguish with high constant probability between the case that a graph contains $\Theta(n)$ triangles and the case that it contains no triangles. This bound holds when neighbor and degree queries are allowed.*

Theorem 0.4 *For $m = O(n)$ it is necessary to perform $\Omega(m)$ queries in order to distinguish with high constant probability between the case that a graph contains $\Theta(n^2)$ length-3 paths and the case that it contains no length-3 path. This bound holds when neighbor and degree queries are allowed.*