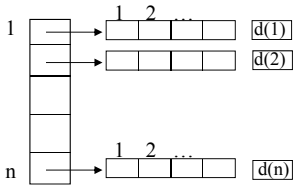
Query Algorithms

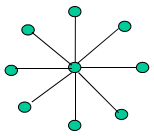
- Assume a very large graph
- We can query the graph
  - Minimize the query number
  - Query complexity vs. time complexity
- Sublinear (ε,δ)-approximation

Counting Stars: Model

- We assume graphs are represented by the incidence lists of the vertices, where each list is accompanied by its length.
- Allowed queries:
  - what is the degree, d(v), of any vertex v?
  - who is the i'th neighbor of v, for any vertex v and index $1 \leq i \leq d(v)$?



[Gonen, Ron, Shavitt, SIDMA 2011]



Approximating Stars

Upper Bound:

- Given an approximation parameter $0 < \varepsilon < 1$ and query access to a graph G, the algorithm outputs an estimate ν'_s such that, with high constant probability,

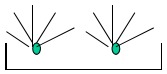
  $$(1-\varepsilon)\nu_s(G) \leq \nu'_s \leq (1+\varepsilon)\,\nu_s(G),$$

  where $\nu_s(G)$ denotes the number of stars of size $s+1$ in the graph.
- The expected query complexity and running time of the algorithm are

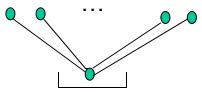
$$O\left(\frac{n}{\nu_s(G)^{\frac{1}{s+1}}} + \min\left\{n^{1-\frac{1}{s}}, \frac{n^{s-\frac{1}{s}}}{\nu_s(G)^{1-\frac{1}{s}}}\right\} \cdot poly(\log n, 1/\varepsilon)\right)$$

Upper Bound-Main Idea

- Consider a partition of the graph vertices into $O(\log n/\varepsilon)$ buckets where in each bucket all vertices have the same degree (with respect to the entire graph) up to a multiplicative factor of $(1\pm O(\varepsilon))$. The degree in bucket B_i is $\sim(1+\beta)^i$, $\beta=O(\varepsilon)$.
- If we could get a good estimate of the size of each bucket by sampling, then we would have a good estimate of the number of s-stars (since the vertices in each bucket are the centers of approximately the same number of stars).



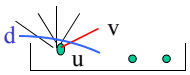
- The difficulty is that some buckets may be very small and we might not even hit them when sampling vertices. However, these buckets can significantly contribute to the number of stars in the graph.



Upper Bound

Solution: if we have an estimate ê on the number of edges incident to vertices in a certain bucket, and all vertices in that bucket have degree roughly d, then the number of stars whose center belongs to this bucket is approximately $\frac{1}{s}\hat{e}\binom{d-1}{s-1}$:

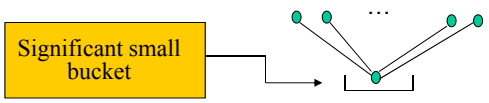
- Consider an edge (u,v) that is incident to a vertex u that has degree (roughly) d. Then the number of stars that include this edge and are centered at u is (roughly) $\binom{d-1}{s-1}$.
- If we sum this expression over all ê edges that are incident to vertices in the bucket of u, then each star (that is centered at a vertex in the bucket) is counted s time, and hence we divide the expression $\hat{e}\binom{d-1}{s-1}$ by s.



Upper Bound

→ we need to find such estimate ê:

- We can easily estimate #edges between large buckets.
- The difficulty is to estimate #edges between a large bucket and a small bucket.
- We first define the notion of significant small buckets. Such buckets have a non-negligible contribution to the total number of s-stars (where each vertex accounts for the number of stars that it is a center of).



Upper Bound

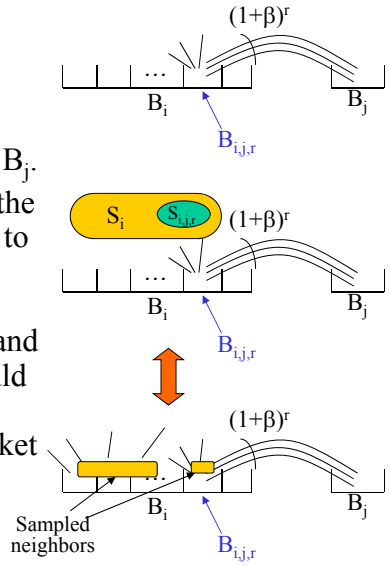
- For each large bucket B_i and (significant) small bucket B_j we further consider partitioning the vertices in B_i according to the number of neighbors they have in B_j.
- We encounter a tradeoff between the number of vertices in B_i that need to be sampled in order to get sufficiently many vertices that belong to a particular sub-bucket and the number of neighbors that should be sampled so as to detect (approximately) to which sub-bucket

large $r \Rightarrow |B_{i,j,r}|$ is relatively small

⇓

Need large sample of v to hit $B_{i,j,r}$

BUT small sample of neighbors of $v \in B_i$



Upper Bound – Case of length-2 paths

Notations:

- $\ell(G)$ = the number of length-2 paths (2-stars) in a graph G.
- $\Gamma(v)$ = the set of neighbors of a vertex v.
- $d(v)$ = the degree of a vertex v.
- For two (not necessarily disjoint) subsets of vertices V_1, V_2 of V let $E(V_1,V_2) = \{(v_1,v_2) \in E : v_1 \in V_1, v_2 \in V_2\}$.
- $E_{i,j} = E(B_i,B_j)$.
- $\beta = \varepsilon/c$ where $c > 1$ is a constant.
- $t = \log_{1+\beta} n$ ($t = O(\log n/\varepsilon)$).
- For $i = 0,\ldots,t$, the bucket $B_i = \{v : d(v) \in ((1+\beta)^{i-1}, (1+\beta)^i]\}$.

Upper Bound

Estimating the number of paths whose mid-point belongs to "large" buckets:

- Obtain an estimate, b'_i, such that $(1-\beta)|B_i| \le b'_i \le (1+\beta)|B_i|$.
- Our estimate for the number of length-2 paths whose midpoint is in a large bucket is

$$\sum_{i=0}^{t} b'_i \binom{(1+\beta)^i}{2}$$

Upper Bound

Estimating the number of length-2 paths whose mid-point is in significant "small" buckets:

- For each "large" bucket B_i and significant "small" bucket B_j, we obtain an estimate $\hat{e}_{i,j}$ to the number, $|E_{i,j}|$, of edges between the two buckets.
- The estimate is such that if $|E_{i,j}|$ is above some threshold, then $\hat{e}_{i,j} = (1\pm\beta)|E_{i,j}|$, and otherwise, $\hat{e}_{i,j}$ is small.
- Our estimate for the number of length-2 paths whose midpoint is in a significant "small" bucket is $\frac{1}{2}\sum_{i\in L}\sum_{j\notin L}\hat{e}_{i,j}((1+\beta)^j - 1)$ where L denotes the set of indices of the "large" buckets.
- We set our threshold of "largeness" so that the number of length-2 paths in which all vertices on the path do not belong to L is negligible.

Upper Bound

Estimating $\hat{e}_{i,j}$ for $i \in L$ and $j \notin L$:

- For each $i \in L$ and $j \notin L$ we consider partitioning the vertices in B_i that have neighbors in B_j into sub-buckets:
  for $r = 0,\ldots,i$, $B_{i,j,r} = \{v \in B_i : (1+\beta)^{r-1} < \Gamma(v) \cap B_j \leq (1+\beta)^r\}$.
- By the definition of $B_{i,j,r}$ we have that
  $\sum_r |B_{i,j,r}| (1+\beta)^r = (1 \pm \beta)|E_{i,j}|$.
- $\rightarrow$ good estimate of $|B_{i,j,r}|$ gives good estimate of $|E_{i,j}|$.
- r is large $\rightarrow$ $|B_{i,j,r}|$ may be relatively small $\rightarrow$ we need to take a relatively large sample of vertices in order to "hit" $B_{i,j,r}$. However, in order to determine whether a vertex (in B_i) belongs to $B_{i,j,r}$ for large r, it suffices to take a small sample of its neighbors.
- r is relatively small $\rightarrow$ $B_{i,j,r}$ must be relatively big (if $|E(B_{i,j,r},B_j)|$ is non-negligible) $\rightarrow$ it suffices to take a relatively small sample so as to "hit" $B_{i,j,r}$ and then we can afford performing many neighbor queries from the selected vertices.

The Algorithm

this assumption can be removed

- We assume first that we have a rough estimate ℓ' such that $\frac{1}{2}\ell(G) \leq \ell' \leq 2\ell(G)$.
- Estimating the number of length-2 paths for $G = (V,E)$:
  - Input: ε, ℓ'
  - Let $\beta = \varepsilon/32$, $t = \lceil \log_{1+\beta} n \rceil$, $\theta_1 = \varepsilon^{2/3}\ell'^{1/3}/(32t^{4/3})$.
  - Uniformly and independently select $\Theta((n/\theta_1)(\log t/\varepsilon^2))$ vertices from V, and let S denote the multi-set of the selected vertices (that is, we allow repetitions).
  - For $i = 0,\ldots,t$ determine $S_i = S \cap B_i$ by performing a degree query on every vertex in S.
  - Let $L = \{i : |S_i|/|S| \geq 2\theta_1/n\}$. If $\max_{i \in L}\left\{\binom{(1+\beta)^{i-1}}{2} \cdot \theta_1\right\} > 4\ell'$ then terminate.
  - For each $i \in L$ run Algorithm 2 to get estimates $\{\hat{e}_{i,j}\}_{j \notin L}$ for $\{|E_{i,j}|\}_{j \notin L}$.
  - Output $\ell'' = \sum_{i \in L} n \cdot \frac{|S_i|}{|S|} \cdot \binom{(1+\beta)^i}{2} + \sum_{j \notin L} \frac{1}{2} \sum_{i \in L} \hat{e}_{i,j}((1+\beta)^j - 1)$

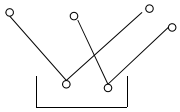
The Algorithm

Paths with mid-point in large buckets

Paths with mid-point in significant small buckets

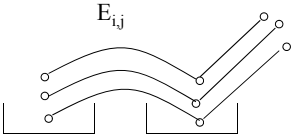
$$\ell'' = \sum_{i \in L} n \cdot \frac{|S_i|}{|S|} \cdot \binom{(1+\beta)^i}{2} + \sum_{j \notin L} \frac{1}{2} \sum_{i \in L} \hat{e}_{i,j} ((1+\beta)^j - 1)$$

Estimation of $|B_i|$



B_i

$E_{i,j}$



B_i B_j

Algorithm 2

Estimating $\{|E_{i,j}|\}$ for a given $i \in L$ and all $j \notin L$:

- Input: L, $i \in L$, ε, ℓ'
- For each $0 \leq p \leq i$ let $\theta_2(p) = \varepsilon^{3/2}\ell'^{1/2}/(c_2 t^{5/2}(1+\beta)^{p/2})$, where c_2 is a constant that will is set in the analysis. Let p_0 be the smallest value of p satisfying $(1/4)\theta_2(p+1) \leq n$.
- For $p = i$ down to p_0 initialize $\hat{S}^{(p)}_{i,j,p} = \phi$.
- For $p = i$ down to p_0 do:
  - Let $s^{(p)} = \Theta((n/\theta_2(p))(t/\beta)^2 \log t)$, and let $g^{(p)} = \Theta((1+\beta)^{i-p} \log(tn)/\beta^2)$.
  - Uniformly, independently at random select $s^{(p)}$ vertices from $S^{(p+1)}$ (where $S^{(i+1)} = V$) and let $S^{(p)}$ be the multiset of vertices selected.

Algorithm 2

- Determine $S_i^{(p)} = S^{(p)} \cap B_i$ by performing a degree query on every vertex in $S^{(p)}$. If $|S_i^{(p)}| < (s^{(p)}/n)\,\theta_2(p)/(4(1+\beta))$, then go to . Else, if $|S_i^{(p)}| > \frac{s^{(p)}}{n} \cdot \frac{4\ell'}{\binom{(1+\beta)^{i-1}}{2}}$ then terminate.
- For each $v \in S_i^{(p)}$ select (uniformly, independently at random) $g^{(p)}$ neighbors of v, and for each $j \notin L$ let $\gamma_j^{(p)}(v)$ be the number of these neighbors that belong to B_j . (If $g^{(p)} \geq d(v)$ then consider all neighbors of v.)
- For each $j \notin L$ and for each $v \in S_i^{(p)} \setminus \cup_{p'>p} \hat{S}^{(p')}_{i,j,p'}$, if $(1+\beta)^{p-1}/d(v) < \gamma_j^{(p)}(v)/g^{(p)}(v) \leq (1+\beta)^p/d(v)$ then add v to $\hat{S}^{(p)}_{i,j,p}$.

– For each $j \notin L$ let $\hat{e}_{i,j} = \sum_{p=p_0}^{i} \frac{n}{s^{(p)}} |\hat{S}^{(p)}_{i,j,p}| \cdot (1+\beta)^p$.

– Return $\{\hat{e}_{i,j}\}_{j \notin L}$.

Estimation of $B_{i,j,p}$

Main Theorem

With probability at least 2/3, the output, ℓ'', of Algorithm satisfies $\ell'' = (1 \pm \varepsilon)\ell(G)$

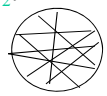
The query complexity and running time of the algorithm are

$$O\left(\frac{n}{\ell(G)^{1/3}} + \min\left\{n^{1/2}, \frac{n^{3/2}}{\ell(G)^{1/2}}\right\}\right) \cdot poly(\log n, 1/\varepsilon)$$

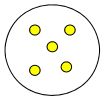
Lower Bounds for Approximating the Number of Length-2 Paths

Theorem: any multiplicative approximation algorithm for the number of length-2 paths must perform $\Omega\left(\frac{n}{\ell(G)^{1/3}}\right)$ queries.

Proof sketch:
Graph G_1: empty graph $\ell(G_1)=0$.
Graph G_2:



Clique of size $\lceil \ell^{1/3} \rceil$



Independent set of size n-$\lceil \ell^{1/3} \rceil$

$$\ell(G_2) = \lceil \ell^{1/3} \rceil \cdot \binom{\lceil \ell^{1/3} \rceil - 1}{2} = \theta(\ell)$$

In order to distinguish between G_1 and G_2 it is necessary to perform a query on a vertex in the clique. The probability of hitting such a vertex in $o\left(\frac{n}{\ell(G)^{1/3}}\right)$ queries is o(1).

Lower Bounds for Approximating the Number of Length-2 Paths

Theorem: any constant-factor approximation algorithm for the number of length-2 paths must perform $\Omega(\sqrt{n})$ queries when the number of length-2 paths is $O(n^2)$.
Proof sketch: by previous theorem we may consider the case that $\ell(G) > n^{3/2} > n$. We show that every n, every constant c and every $n < \ell < (n/2c)^2$ there exist two families of n vertex graphs for which the following holds. In both families the number of length-2 paths is $\theta(\ell)$, but in one family this number is a factor c larger than in the other family. However, it is not possible to distinguish with high constant probability between a graph selected randomly in one family and a graph selected randomly in the other family using $o(\sqrt{n})$ queries.

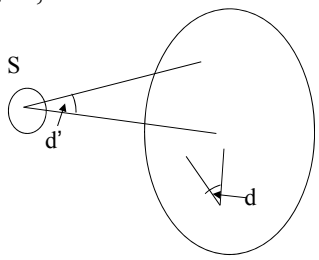
Lower Bounds for Approximating the Number of Length-2 Paths

Graph G_1: determined by $d=\lfloor 2\ell/n \rfloor$ matchings. $\rightarrow$ each vertex has degree d, and

$$\ell(G_1) = n \cdot \binom{d}{2} < \ell$$

Graph G_2: There is a small subset, S, of c vertices, where each vertex in S has degree $d' = \lceil \sqrt{(2\ell)} \rceil + 1$, and each vertex in V\ S has degree $d = \lfloor 2\ell/n \rfloor$.

$$\ell(G_2) > c \cdot \binom{d'}{2} = c \cdot \binom{\sqrt{2\ell}+1}{2} > c\ell$$



Lower Bounds for Approximating the Number of Length-2 Paths

Theorem: any constant-factor approximation algorithm for the number of length-2 paths must perform $\Omega(n^{3/2}/\ell^{1/2})$ queries when the number of length-2 paths is $\Omega(n^2)$.

Proof sketch: we show that every n, every constant c and every $\ell = \Omega(n^2)$, $\ell < n^3/(16c^2)$, there exist two families of n-vertex graphs for which the following holds. In both families the number of length-2 paths is $\theta(\ell)$, but in one family this number is a factor c larger than in the other family. However, it is not possible to distinguish with high constant probability between a graph selected randomly in one family and a graph selected randomly in the other family using $o(n^{3/2}/\ell^{1/2})$ queries.

Lower Bounds for Approximating the Number of Length-2 Paths

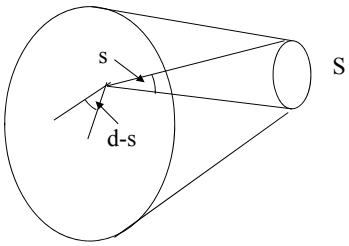
Graph G_1: determined by $d=\lfloor 2\ell/n \rfloor$ matchings. $\rightarrow$ each vertex has degree d, and $\ell(G_1) = n \cdot \binom{d}{2} < \ell$

Graph G_2: There is subset, S, of $s = \lceil 4c\ell/n^2 \rceil$ vertices, and a complete bipartite graph between S and V\S. In addition, there are d-s perfect matchings between vertices in V\S. $\rightarrow$ each vertex in V\S has degree d, and

$$\ell(G_2) \geq s \cdot \binom{n-s}{2} > s \cdot \binom{3n/4}{2}$$

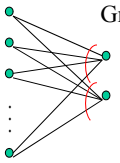
$$= \left\lceil \frac{4c\ell}{n^2} \right\rceil \cdot \binom{3n/4}{2} > c\ell$$

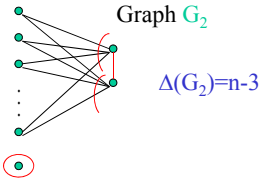
$\ell < n^3/(16c^2) \rightarrow s < n/4$



Lower Bounds for other small motifs:

- For triangles a lower bound that is linear in n when the number of edges is $\Theta(n)$:





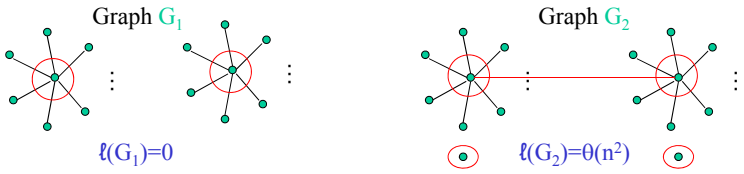
To distinguish between G_1 and G_2 it is necessary:

- to either hit the isolated vertex, OR
- to hit the edge between the two high-degree vertices, OR
- to observe all neighbors of one of the high-degree vertices.

For all cases $\Omega(n)$ queries are necessary.

Lower Bounds for other small motifs:

- For length-3 paths we show a lower bound that is linear in the number of edges when the number of edges is $\Theta(n)$:



In order to distinguish between G_1 and G_2 it is necessary

- to hit one of the isolated vertices, OR
- to hit the edge between the two centers, OR
- to observe all neighbors of one of the centers.

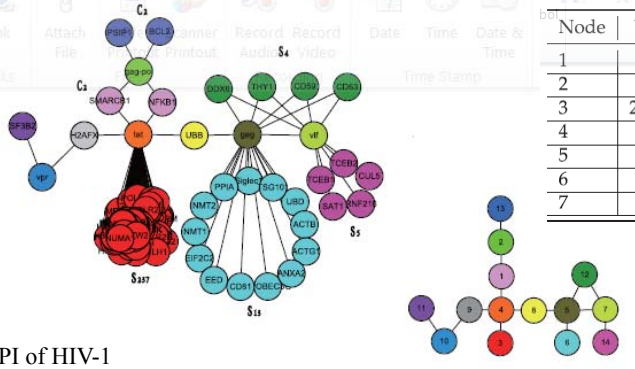
For all cases $\Omega(m)$ queries are necessary.

Related Problems

- For bounded degree graphs:
  If the graph is $\Omega(1)$-far from cycle-free graph an algorithm can find a cycle of polylog length is $\tilde{O}(\sqrt{n})$
- An $\tilde{O}(poly(\frac{1}{\varepsilon})\sqrt{n})$ tester for cycle freeness

[Czumaj *et al.*, RSA 2012]

Symmetry Compression Method



PPI of HIV-1

(a) (b)

Node	W	Sy	Node	W	Sy
1	2	0	8	1	0
2	1	0	9	1	0
3	237	0	10	1	0
4	1	0	11	1	0
5	1	0	12	4	0
6	13	0	13	2	0
7	1	0	14	5	0

[Wang *et al*., IEEE/ACM Trans on Comp. Bio. 2012]

Network Graphlets

Future challenges

- Efficient 5-node graphlets algorithms
  - There are 20 graphlets
  - … and 57 orbits
- Efficient algs for large important motifs
  - Cycles, bi-partites, cliques, almost cliques
- Following motifs in evolving networks