Sublinear Time Estimation of Degree Distribution Moments: The Degeneracy Connection*

Talya Eden†[1], **Dana Ron**‡[2], **and C. Seshadhri**[3]

1 School of Electrical Engineering, Tel Aviv University, Tel Aviv, Israel
talyaa01@gmail.com

2 School of Electrical Engineering, Tel Aviv University, Tel Aviv, Israel
danaron@tau.ac.il

3 Department of Computer Science, University of California – Santa Cruz, Santa Cruz, CA, USA
sesh@ucsc.edu

Abstract

We revisit the classic problem of estimating the degree distribution moments of an undirected graph. Consider an undirected graph $G = (V, E)$ with n (non-isolated) vertices, and define (for $s > 0$) $\mu_s = \frac{1}{n} \cdot \sum_{v \in V} d_v^s$. Our aim is to estimate μ_s within a multiplicative error of $(1+\varepsilon)$ (for a given approximation parameter $\varepsilon > 0$) in sublinear time. We consider the sparse graph model that allows access to: uniform random vertices, queries for the degree of any vertex, and queries for a neighbor of any vertex. For the case of $s = 1$ (the average degree), $\widetilde{O}(\sqrt{n})$ queries suffice for any constant ε (Feige, SICOMP 06 and Goldreich-Ron, RSA 08). Gonen-Ron-Shavitt (SIDMA 11) extended this result to all integral $s > 0$, by designing an algorithms that performs $\widetilde{O}(n^{1-1/(s+1)})$ queries. (Strictly speaking, their algorithm approximates the number of star-subgraphs of a given size, but a slight modification gives an algorithm for moments.)

We design a new, significantly simpler algorithm for this problem. In the worst-case, it exactly matches the bounds of Gonen-Ron-Shavitt, and has a much simpler proof. More importantly, the running time of this algorithm is connected to the *degeneracy* of G. This is (essentially) the maximum density of an induced subgraph. For the family of graphs with degeneracy at most α, it has a query complexity of $\widetilde{O}\left(\frac{n^{1-1/s}}{\mu_s^{1/s}}\left(\alpha^{1/s} + \min\{\alpha, \mu_s^{1/s}\}\right)\right) = \widetilde{O}(n^{1-1/s}\alpha/\mu_s^{1/s})$. Thus, for the class of bounded degeneracy graphs (which includes all minor closed families and preferential attachment graphs), we can estimate the average degree in $\widetilde{O}(1)$ queries, and can estimate the variance of the degree distribution in $\widetilde{O}(\sqrt{n})$ queries. This is a major improvement over the previous worst-case bounds. Our key insight is in designing an estimator for μ_s that has low variance when G does not have large dense subgraphs.

1998 ACM Subject Classification F.2 Analysis of Algorithms and Problem Complexity, F.2.2 Nonnumerical Algorithms and Problems, G.2.2 Graph Theory

Keywords and phrases Sublinear algorithms, Degree distribution, Graph moments

Digital Object Identifier 10.4230/LIPIcs.ICALP.2017.7

* The full version of this extended abstract is available at https://arxiv.org/abs/1604.03661.

† This research was partially supported by a grant from the Blavatnik fund. The author is grateful to the Azrieli Foundation for the award of an Azrieli Fellowship.

‡ This research was partially supported by the Israel Science Foundation grant No. 671/13 and by a grant from the Blavatnik fund.

44th International Colloquium on Automata, Languages, and Programming (ICALP 2017).
Editors: Ioannis Chatzigiannakis, Piotr Indyk, Fabian Kuhn, and Anca Muscholl;
Article No. 7; pp. 7:1–7:13
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

Estimating the mean and moments of a sequence of n integers $d_1, d_2, \ldots, d_n$ is a classic problem in statistics that requires little introduction. In the absence of any knowledge of the moments of the sequence, it is not possible to prove anything non-trivial. But suppose these integers formed the degree sequence of a graph. Formally, let $G = (V, E)$ be an undirected graph over n vertices, and let d_v denote the degree of vertex $v \in V$, where we assume that $d_v \geq 1$ for every v.[1] Feige proved that $O^*(\sqrt{n})$ uniform random vertex degrees (in expectation) suffice to provide a $(2+\varepsilon)$-approximation to the average degree [23]. (We use $O^*(\cdot)$ to suppress $\text{poly}(\log n, 1/\varepsilon)$ factors.) The variance can be as large as n for graphs of constant average degree (simply consider a star), but the constraints of a degree distribution allow for non-trivial approximations. Classic theorems of Erdős-Gallai and Havel-Hakimi characterize such sequences [29, 21, 27].

Again, the star graph shows that the $(2+\varepsilon)$-approximation cannot be beaten in sublinear time through pure vertex sampling. Suppose we could also access random neighbors of a given vertex. In this setting, Goldreich and Ron showed it is possible to obtain a $(1+\varepsilon)$-approximation to the average degree in $O^*(\sqrt{n})$ expected time [24].

In a substantial (and complex) generalization, Gonen, Ron, and Shavitt (henceforth, GRS) gave a sublinear-time algorithm that estimates the higher moments of the degree distribution [25]. Technically, GRS gave an algorithm for approximating the number of stars in a graph, but a simple modification yields an algorithm for moments estimation. For precision, let us formally define this problem. The *degree distribution* is the distribution over the degree of a uniform random vertex. The *s-th moment* of the degree distribution is $\mu_s \triangleq \frac{1}{n} \cdot \sum_{v \in V} d_v^s$.

The Degree Distribution Moment Estimation (DDME) Problem. *Let $G = (V, E)$ be a graph over n vertices, where n is known. Access to G is provided through the following queries. We can (i) get the id (label) of a uniform random vertex, (ii) query the degree d_v of any vertex v, (iii) query a uniform random neighbor of any vertex v. Given $\varepsilon > 0$ and $s \geq 1$, output a $(1+\varepsilon)$-multiplicative approximation to μ_s with probability*[2] $> 2/3$.

The DDME problem has important connections to network science, which is the study of properties of real-world graphs. There have been numerous results on the significance of heavy-tailed/power-law degree distributions in such graphs, since the seminal results of Barabási-Albert [5, 10, 22]. The degree distribution and its moments are commonly used to characterize and model graphs appearing in varied applications [7, 36, 14, 37, 8]. On the theoretical side, recent results provide faster algorithms for graphs where the degree distribution has some specified form [6, 9]. Practical algorithms for specific cases of DDME have been studied by Dasgupta et al and Chierichetti et al. [17, 13]. (These results requires bounds on the mixing time of the random walk on G.)

1.1 Results

Let m denote the number of edges in the graph (where m is not provided to the algorithm). For the sake of simplicity, we restrict the discussion in the introduction to case when $\mu_s \leq n^{s-1}$.

---

[1] The assumption on there being no isolated vertices is made here only for the sake of simplicity of the presentation, as it ensures a basic lower bound on the moments.

[2] The constant $2/3$ is a matter of convenience. It can be increased to at least $1-\delta$ by taking the median value of $\log(1/\delta)$ independent invocations.

As observed by GRS, the complexity of the DDME problem is smaller when μ_s is significantly larger. GRS designed an (expected) $O^*\left(n^{1-1/(s+1)}/\mu_s^{1/(s+1)} + n^{1-1/s}\right)$-query algorithm for DDME and proved this expression was optimal up to $\mathrm{poly}(\log n, 1/\varepsilon)$ dependencies. (Here $O^*(\cdot)$ also suppresses additional factors that depend only on s). Note that for a graph without isolated vertices, $\mu_s \geq 1$ for every $s > 0$, so this yields a worst-case $O^*(n^{1-1/(s+1)})$ bound. The $s = 1$ case is estimating the average degree, so this recovers the $O^*(\sqrt{n})$ bounds of Goldreich-Ron. We mention a recent result by Aliakbarpour et al. [1] for DDME, in a stronger model that assumes additional access to uniform random edges. They get a better bound of $O^*(m/(n\mu_s)^{1/s})$ in this stronger model, for $s > 1$ (and $\mu_s \leq n^{s-1}$). Note that the main challenge of DDME is in measuring the contribution of high-degree vertices, which becomes substantially easier when random edges are provided. In the DDME problem without such samples, it is quite non-trivial to even detect high degree vertices.

All the bounds given above are known to be optimal, up to $\mathrm{poly}(\log n, 1/\varepsilon)$ dependencies, and at first blush, this problem appears to be solved. We unearth a connection between DDME and the *degeneracy* of G. The degeneracy of G is (up to a factor 2) the maximum density over all subgraphs of G. We design an algorithm that has a nuanced query complexity, depending on the degeneracy of G. Our result subsumes all existing results, and provides substantial improvements in many interesting cases. Furthermore, our algorithm and its analysis are significantly simpler and more concise than in the GRS result.

We begin with a convenient corollary of our main theorem. A tighter, more precise bound appears as Theorem 3.

▶ **Theorem 1.** *Consider the family of graphs with degeneracy at most α. The DDME problem can be solved on this family using $O^*\left(\frac{n^{1-1/s}}{\mu_s^{1/s}}\left(\alpha^{1/s} + \min\{\alpha, \mu_s^{1/s}\}\right)\right)$ queries in expectation. The running time is linear in the number of queries.*

Consider the case of *bounded degeneracy* graphs, where $\alpha = O(1)$. This is a rich class of graphs. *Every* minor-closed family of graphs has bounded degeneracy, as do graphs generated by the Barabási-Albert preferential attachment process [5]. There is a rich theory of *bounded expansion graphs*, which spans logic, graph minor theory, and fixed-parameter tractability [32]. All these graph classes have bounded degeneracy. For every such class of graphs, we get a $(1+\varepsilon)$-estimate of μ_s in $O^*(n^{1-1/s}/\mu_s^{1/s})$ time. We stress that bounded degeneracy does not imply any bounds on the maximum degree or the moments. The star graph has degeneracy 1, but has extremely large moments due to the central vertex.

Consider any bounded degeneracy graph without isolated vertices. We can accurately estimate the average degree ($s = 1$) in $\mathrm{poly}(\log n)$ queries, and estimate the variance of the degree distribution ($s = 2$) in $\sqrt{n} \cdot \mathrm{poly}(\log n)$ queries. Contrast this with the (worst-case optimal) $\sqrt{n}$ bounds of Feige and Goldreich-Ron for average degree, and the $O^*(n^{2/3})$ bound of GRS for variance estimation. For general s, our bound is a significant improvement over the $O^*(n^{1-1/(s+1)}/\mu_s^{1/(s+1)})$ bound of GRS.

The algorithm attaining Theorem 1 requires an upper bound on the degeneracy of the graph. When an degeneracy bound is not given, the algorithm recovers the bounds of GRS, with an improvement on the extra $\mathrm{poly}(\log n)/\varepsilon$ factors. More details are in Theorem 3. We note that the degeneracy-dependent bound in Theorem 1 cannot be attained by an algorithm that is only given n as a parameter. In particular, if an algorithm is only provided with n and must work on all graphs with n vertices, then it must perform $\Omega(\sqrt{n})$ queries in order to approximate the average degree even for graphs of constant degeneracy (and constant average degree). Details are given in Subsection 7.1 in the full version of the paper.

The bound of Theorem 1 may appear artificial, but we prove that it is optimal when $\mu_s \leq n^{s-1}$. (For the general case, we also have optimal upper and lower bounds.) This construction is an extension of the lower bound proof of GRS.

▶ **Theorem 2.** *Consider the family of graphs with degeneracy α and where $\mu_s \leq n^{s-1}$. Any algorithm for the DDME problem on this family requires $\Omega\Big(\frac{n^{1-1/s}}{\mu_s^{1/s}} \cdot \Big(\alpha^{1/s} + \min\{\alpha, \mu_s^{1/s}\}\Big)\Big)$ queries.*

1.2 From degeneracy to moment estimation

We begin with a closer look at the lower bound examples of Feige, Goldreich-Ron, and GRS. The core idea is quite simple: DDME is hard when the overall graph is sparse, but there are small dense subgraphs. Consider the case of a clique of size $100\sqrt{n}$ connected to a tree of size n. The small clique dominates the average degree, but any sublinear algorithm with access only to random vertices pays $\Omega(\sqrt{n})$ for a non-trivial approximation. GRS use more complex constructions to get an $\Omega(n^{1-1/(s+1)})$ lower bound for general s. This also involves embedding small dense subgraphs that dominate the moments.

Can we prove a converse to these lower bound constructions? In other words, prove that the *non-existence* of dense subgraphs must imply that DDME is easier? A convenient parameter for this non-existence is the *degeneracy*.

But the degeneracy is a global parameter, and it is not clear how a sublinear algorithm can exploit it. Furthermore, DDME algorithms are typically very local; they sample random vertices, query the degrees of these vertices and maybe also query the degrees of some of their neighbors. We need a local property that sublinear algorithms can exploit, but can also be linked to the degeneracy. We achieve this connection via the *degree ordering* of G. Consider the DAG obtained by directing all edges from lower to higher degree vertices. Chiba-Nishizeki related the properties of the *out-degree* distribution to the degeneracy, and exploited this for clique counting [12]. Nonetheless, there is no clear link to DDME. (Nor do we use any of their techniques; we state this result merely to show what led us to use the degree ordering).

Our main insight is the construction of an estimator for DDME whose variance depends on the degeneracy of G. This estimator critically uses the degree ordering. Our proof relates the variance of this estimator to the density of subgraphs in G, which can be bounded by the degeneracy. We stress that our algorithm is quite simple, and the technicalities are in the analysis and setting of certain parameters.

1.3 Designing the algorithm

Designate the *weight* of an edge (u, v) to be $d_u^{s-1} + d_v^{s-1}$. A simple calculation yields that the sum of the weights of all edges is exactly $M_s \triangleq \sum_v d_v^s = n \cdot \mu_s$. Suppose we could sample uniform random *edges* (and knew the total number of edges). Then we could hope to estimate M_s through uniform edge sampling. The variance of the edge weights can be bounded, and this yields an $O^*(m/(n\mu_s)^{1/s}) = O^*(n^{1-1/s})$ algorithm (when no vertex is isolated). Indeed, this is very similar to the approach of Aliakbarpour et al. [1]. Such variance calculations were also used in the classic Alon-Matias-Szegedy result of frequency moment estimation [3].

Our approach is to simulate uniform edge samples using uniform vertex samples. Suppose we sampled a set R of uniform random vertices. By querying the degrees of all these vertices, we can select vertices in R with probability proportional to their degrees, which allows us to uniformly sample edges that are incident to vertices in R. Now, we simply run the uniform edge sampling algorithm on these edges. This algorithmic structure was recently used for sublinear triangle counting algorithms by Eden et al. [19].

Here lies the core technical challenge. How to bound the number of random vertices that is sufficient for effectively simulating the random edge algorithm? This boils down to the behavior of the variance of the "vertex weight" distribution. Let the weight of a vertex be the sum of weights of its incident edges. The weight distribution over vertices can be extremely skewed, and this approach would require a forbiddingly large R.

A standard technique from triangle counting (first introduced by Chiba-Nishizeki [12]) helps reduce the variance. Direct all edges from lower degree to higher degree vertices, breaking ties consistently. Now, set the weight of a vertex to be the sum of weights on incident *out-edges*. Thus, a high-degree vertex with lower degree neighbors will have a significantly reduced weight, reducing overall variance. In the general case (ignoring degeneracy), a relatively simple argument bounds the maximum weight of a vertex, which enables us to bound the variance of the weight distribution. This yields a much simpler algorithm and proof of the GRS bound.

In the case of graphs with bounded degeneracy, we need a more refined approach. Our key insight is an intimate connection between the variance and the existence of dense subgraphs in G. We basically show that the main structure that leads to high variance is the existence of dense subgraphs. Formally, we can translate a small upper bound on the density of any subgraph to a bound on the variance of the vertex weights. This establishes the connection to the graph degeneracy.

1.4 Simplicity of our algorithm

Our viewpoint on DDME is quite different from GRS and its precursor [24], which proceed by bucketing the vertices based on their degree. This leads to a complicated algorithm, which essentially samples to estimate the size of the buckets, and also the number of edges between various buckets (and "sub-buckets"). We make use of buckets in out analysis, in order to obtain the upper bound that depends on the degeneracy α (in order to achieve the GRS upper bound, our analysis does not use bucketing).

As explained above, our main DDME procedure, **Moment-estimator** is simple enough to present in a few lines of pseudocode (see Figure 1). We feel that the structural simplicity of **Moment-estimator** is an important contribution of our work.

Moment-estimator takes two sampling parameters r and q. The main result Theorem 3 follows from running **Moment-estimator** with a standard geometric search for the right setting of r and q. In **Moment-estimator** we use $id(v)$ to denote the label of a vertex v, where vertices have unique ids and there is a complete order over the ids.

1.5 Other related work

As mentioned at the beginning of this section, Aliakbarpour et al. [1] consider the problem of approximating the number of s-stars for $s \geq 2$ when given access to uniformly selected edges. Given the ability to uniformly select edges, they can select vertices with probability proportional to their degree (rather than uniformly). This can be used to get an unbiased estimator of μ_s (or the s-star count) with low variance. This leads to an $O(m/(n\mu_s)^{1/s})$ bound, which is optimal (for $\mu_s \leq n^{s-1}$).

Dasgupta, Kumar, and Sarlos give practical algorithms for average degree estimation, though they assume bounds on the mixing time of the random walk on the graph [17]. A recent paper of Chierichetti et al. build on these methods to sample nodes according to

Moment-estimator$_s(r,q)$

1. Select r vertices, uniformly, independently, at random and let the resulting multi-set be denoted by R. Query the degree of each vertex in R, and let $d_R = \sum_{v\in R} d_v$.
2. For $i = 1,\ldots,q$ do:
   a. Select a vertex v_i with probability proportional to its degree (i.e., with probability d_{v_i}/d_R), and query for a random neighbor u_i of v_i.
   b. If $d_{v_i} < d_{u_i}$ or $d_{v_i} = d_{u_i}$ and $id(v_i) < id(u_i)$, set $X_i = (d_{v_i}^{s-1} + d_{u_i}^{s-1})$. Else, set $X_i = 0$.
3. Return $X = \frac{1}{r}\cdot\frac{d_R}{q}\cdot\sum\limits_{i=1}^{q} X_i$.

Figure 1 Algorithm **Moment-estimator**$_s$ for approximating μ_s.

powers of their degree (which is closely related to DDME) [13]. Simpson, Seshadhri, and McGregor give practical algorithms to estimate the entire cumulative degree distribution in the streaming setting [38]. This is different from the sublinear query model we consider, and the results are mostly empirical.

In [19], Eden et al. present an algorithm for approximating the number of triangles in a graph. Although this is a very different problem than DDME, there are similar challenges regarding high-degree vertices. Indeed, as mentioned earlier, the approach of sampling random edges through a set of random vertices was used in [19].

The degeneracy is closely related to other “density" notions, such as the arboricity, thickness, and strength of a graph [4]. There is a rich history of algorithmic results where run time depends on the degeneracy [31, 12, 2, 20].

Other sublinear algorithms for estimating various graph parameters include: approximating the size of the minimum-weight spanning tree [11, 16, 15], maximum matching [33, 39] and of the minimum vertex cover [35, 33, 30, 39, 28, 34].

A Comment regarding this extended abstract

We defer some of the details of the analysis of the algorithm, as well as the lower bound proof, to the accompanying full version of the paper.

2 The main theorem

▶ **Theorem 3.** *For every graph G, there exists an algorithm that returns a value Z such that $Z \in [(1-\varepsilon)\mu_s(G), (1+\varepsilon)\mu_s(G)]$ with probability at least $2/3$. Assume that algorithm is given α, an upper bound on the degeneracy of G. (If no such bound is provided, the algorithm assumes a trivial bound of $\alpha = \infty$.) The expected running time is the minimum of the following two expressions.*

$$O\left(2^s \cdot n^{1-1/s}\cdot \log^2 n \cdot \left(\frac{\alpha}{\mu_s}\right)^{1/s} + \min\left\{\frac{n^{1-1/s}\cdot\alpha}{\mu_s^{1/s}}, \frac{n^{s-1}\cdot\alpha}{\mu_s}\right\}\right)\cdot\frac{s\log n\cdot\log(s\log n)}{\varepsilon^2} \tag{1}$$

$$O\left(\frac{n^{1-1/(s+1)}}{\mu_s^{1/(s+1)}} + \min\left\{n^{1-1/s}, \frac{n^{s-1-1/s}}{\mu_s^{1-1/s}}\right\}\right)\cdot\frac{s\log n\cdot\log(s\log n)}{\varepsilon^2} \tag{2}$$

Equation (2) is essentially the query complexity of GRS (albeit with a better dependence on $s, \log n$, and $1/\varepsilon$). Thus, our algorithm is guaranteed to be at least as good as that. If α is exactly the degeneracy of G, then we can prove that Equation (1) is less than Equation (2). Within each expression, there is a min of two terms. The first term is smaller iff $\mu_s \le n^{s-1}$.

The mechanism of deriving this rather cumbersome running time is the following. The algorithm of Theorem 3 runs **Moment-estimator** for geometrically increasing values of r and q, which is in turn derived from a geometrically decreasing guess of μ_s. It uses this guess to set r and q. There is a setting of values depending on α, and a setting independent of it. The algorithm simply picks the minimum of these settings to achieve the smaller running time.

3 Sufficient conditions for r and q in Moment-estimator

In this section we provide sufficient conditions on the parameters r and q that are used by **Moment-estimator** (Figure 1), in order for the algorithm to return a $(1+\varepsilon)$ estimate of μ_s. First we introduce some notations. For a graph $G = (V, E)$ and a vertex $v \in V$, let $\Gamma(v)$ denote the set of neighbors of v in G (so that $d_v = |\Gamma(v)|$). For any (multi-)set R of vertices, let E_R be the (multi-)set of edges incident to the vertices in R. We will think of the edges in E_R as ordered pairs; thus (v, u) is distinct from (u, v), and so $E_R \triangleq \{(v, u) : v \in R, u \in \Gamma(v)\}$. Observe that d_R, as defined in Step 1 of **Moment-estimator** equals $|E_R|$. Let $M_s = M_s(G) \triangleq \sum_{v\in V} d_v^s$, so that $\mu_s = M_s/n$. In the analysis of the algorithm, it is convenient to work with M_s instead of μ_s.

A critical aspect of our algorithm (and proof) is the *degree ordering on vertices*. Formally, we set $u \prec v$ if $d_u < d_v$ or, $d_u = d_v$ and $id(u) < id(v)$. Given the degree ordering, we let $\Gamma^+(v) \triangleq \{u \in \Gamma(v) : v \prec u\}$, $d_v^+ \triangleq |\Gamma^+(v)|$, and $E^+ \triangleq \{(v, u) : v \in V, u \in \Gamma^+(v)\}$. Here and elsewhere, we use $\sum_v$ as a shorthand for $\sum_{v\in V}$.

▶ **Definition 4.** We define the weight of an edge $e = (v, u)$ as follows: if $v \prec u$ define $\mathrm{wt}(e) \triangleq (d_v^{s-1} + d_u^{s-1})$. Otherwise, $\mathrm{wt}(e) \triangleq 0$.
For a vertex $v \in V$, $\mathrm{wt}(v) \triangleq \sum\limits_{u\in\Gamma(v)} \mathrm{wt}((v, u)) = \sum\limits_{u\in\Gamma^+(v)} \mathrm{wt}((v, u))$, and for a (multi-)set of vertices R, $\mathrm{wt}(R) \triangleq \sum\limits_{v\in R} \mathrm{wt}(v)$.

Observe that given the above notations and definition, **Moment-estimator** selects uniform edges from E_R and sets each X_i (in Step 2b) to $\mathrm{wt}((v_i, u_i))$. The next two claims readily follow from Definition 4 (and the description of the algorithm).

▶ **Claim 5.** $\sum_v \mathrm{wt}(v) = M_s$.

▶ **Claim 6.** $\mathrm{Ex}[X] = \mu_s$*, where X is as defined in Step 3 of the algorithm.*

3.1 Conditions on the parameters r and q

We next state two conditions on the parameters r and q, which are used in the algorithm, and then establish several claims, based on the conditions holding. The conditions are stated in terms of properties of the graph as well as the approximation parameter ε and a confidence parameter δ.

1. **The vertex condition:** $r \ge (120 \cdot n \cdot \sum_v \mathrm{wt}(v)^2)/(\varepsilon^2 \cdot \delta \cdot M_s^2)$,
2. **The edge condition:** $q \ge 2000 \cdot m \cdot M_{2s-1}/(\varepsilon^2 \cdot \delta^3 \cdot M_s^2)$.

▶ **Lemma 7.** *If Condition 1 holds, then with probability at least $1-\delta/2$, all the following hold.*

1. $\mathrm{wt}(R) \in \left[\left(1-\frac{\varepsilon}{2}\right)\cdot\frac{r}{n}\cdot M_s, \left(1+\frac{\varepsilon}{2}\right)\cdot\frac{r}{n}\cdot M_s\right]$.
2. $|E_R| \leq \frac{12}{\delta}\cdot\frac{r}{n}\cdot m$.
3. $\sum\limits_{(v,u)\in E_R^+} \mathrm{wt}\left((v,u)\right)^2 \leq \frac{18}{\delta}\cdot\frac{r}{n}\cdot M_{2s-1}$.

The proof of the first item in Lemma 7 follows from Chebyshev's inequality (using $\mathrm{Var}[\mathrm{wt}(R)] \leq \frac{r}{n}\cdot\sum_v \mathrm{wt}(v)^2$), and the proofs of the other two items follow from Markov's inequality (as well as the definition of M_{2s-1}).

▶ **Theorem 8.** *If Conditions 1 and 2 hold, then $X \in [(1-\varepsilon)\mu_s, (1+\varepsilon)\mu_s]$ with probability at least $1-\delta$.*

Proof. Condition on any choice of R. We have $\mathrm{Ex}[X|R] = (1/r)\mathrm{wt}(R)$. Turning to the variance, since the edges (v_i, u_i) are chosen from E_R uniformly at random, it is not hard to verify that

$$\mathrm{Var}[X|R] = \left(\frac{1}{r}\right)^2\cdot\left(\frac{|E_R|}{q}\right)^2\cdot\mathrm{Var}\left[\sum_{i=1}^{q} X_i \;\middle|\; R\right] = \frac{1}{q}\cdot\frac{|E_R|}{r}\cdot\frac{\sum_{(v,u)\in E_R^+}\mathrm{wt}\left((v,u)\right)^2}{r}\,.$$

Let us now condition on R such that the bounds of Lemma 7 hold. Note that such an R is chosen with probability at least $1-\delta/2$. We get $\mathrm{Var}[X|R] \leq \frac{250}{\delta^2}\cdot\frac{1}{q}\cdot\frac{m}{n}\cdot\frac{M_{2s-1}}{n}$. We apply Chebyshev's inequality and invoke Condition 2:

$$\Pr\left[\left|(X|R)-\mathrm{Ex}[X|R]\right| \leq \frac{\varepsilon}{2}\cdot\mu_s\right] \quad\leq\quad \frac{4\cdot\mathrm{Var}[X|R]}{\varepsilon^2\cdot\mu_s^2} \leq \frac{1}{q}\cdot\frac{4\cdot(250/\delta^2)\cdot m\cdot M_{2s-1}}{\varepsilon^2\cdot M_s^2} \leq \frac{\delta}{2}\,.$$

By Lemma 7, $\mathrm{Ex}[X|R] = (1/r)\mathrm{wt}(R) \in [(1-\varepsilon/2)\mu_s, (1+\varepsilon/2)\mu_s]$. The theorem follows by applying the union bound. ◀

4 Satisfying Conditions 1 and 2 in general graphs

We show how to set r and q to satisfy Conditions 1 and 2 in general graphs. Our setting of r and q will give us the same query complexity as [25] (up to the dependence on $1/\varepsilon$ and $\log n$, on which we improve, and the exponential dependence on s in [25], which we do not incur). In the next section we show how the setting of r and q can be improved using a degeneracy bound.

For c_r and c_q that are sufficiently large constants, we set

$$r = \frac{c_r}{\varepsilon^2\cdot\delta}\cdot\frac{n}{M_s^{1/(s+1)}}\;,\qquad q = \frac{c_q}{\varepsilon^2\cdot\delta^3}\cdot\min\left\{n^{1-1/s}, \frac{n^{s-1/s}}{M_s^{1-1/s}}\right\}\,. \tag{3}$$

This setting of parameters requires the knowledge of M_s, which is exactly what we are trying to approximate (up to the normalization factor of n). A simple geometric search argument alleviates the need to know M_s. For details see Section 6.

In order to assert that r as set in Equation (3) satisfies Condition 1, it suffices to establish the next lemma.

▶ **Lemma 9** (Condition 1 holds). $\sum_v \mathrm{wt}(v)^2 \leq 4M_s^{2-\frac{1}{s+1}}$.

Proof. Let $\theta = M_s^{1/(s+1)}$ be a degree threshold. We define $H \triangleq \{v : d_v > \theta\}$, $L \triangleq V \setminus H$. This partition into "high-degree" vertices (H) and "low-degree" vertices (L) will be useful in upper bounding the maximum weight $\mathrm{wt}(v)$ of a vertex v, and hence upper bounding $\sum_v \mathrm{wt}(v)^2$. Details follow.

We first observe that $|H| \le M_s^{1/(s+1)}$. This is true since otherwise, $\sum_{v\in H} d_v^s > M_s^{1/(s+1)} \cdot M_s^{\frac{s}{s+1}} = M_s$, which is a contradiction. We claim that this upper bound on $|H|$ implies that

$$\max_v d_v^+ \le M_s^{1/(s+1)} \; . \tag{4}$$

To verify this, assume, contrary of the claim, that for some v, $d_v^+ > M_s^{1/(s+1)}$. But then there are at least $M_s^{1/(s+1)}$ vertices u such that $d_u \ge d_v \ge d_v^+ > M_s^{1/(s+1)}$. This contradicts the bound on $|H|$.

It will also be useful to bound $\sum_{u\in H} d_u^{s-1}$. By Hölder's inequality with conjugates s and $s/(s-1)$ (a statement of Hölder's inequality can be found in the full version of the paper) and the bound on $|H|$,

$$\sum_{u\in H} d_u^{s-1} = \sum_{u\in H} 1\cdot d_u^{s-1} \le |H|^{1/s}\left(\sum_{u\in H} d_u^s\right)^{\frac{s-1}{s}} \le M_s^{\frac{1}{s(s+1)}}\cdot M_s^{\frac{s-1}{s}} \le M_s^{\frac{s}{s+1}} \; . \tag{5}$$

We now turn to bounding $\max_v\{\mathrm{wt}(v)\}$. By the definition of $\mathrm{wt}(v)$ and the degree ordering,

$$\mathrm{wt}(v) = \sum_{u\in\Gamma^+(v)} (d_v^{s-1} + d_u^{s-1}) \le 2\sum_{u\in\Gamma^+(v)} d_u^{s-1} = 2\sum_{u\in\Gamma^+(v)\cap L} d_u^{s-1} + 2\sum_{u\in\Gamma^+(v)\cap H} d_u^{s-1} \; . \tag{6}$$

For the first term on the right-hand-side of Equation (6), recall that $d_u \le M_s^{1/(s+1)}$ for $u \in L$. Thus, by Equation (4),

$$\sum_{u\in\Gamma^+(v)\cap L} d_u^{s-1} \le d_v^+ \cdot M_s^{\frac{s-1}{s+1}} \le M_s^{\frac{s}{s+1}} \; . \tag{7}$$

For the second term, using $\Gamma^+(v)\cap H \subseteq H$ and applying Equation (5),

$$\sum_{u\in\Gamma^+(v)\cap H} d_u^{s-1} \le \sum_{u\in H} d_u^{s-1} \le M_s^{\frac{s}{s+1}} \; . \tag{8}$$

Finally,

$$\sum_v \mathrm{wt}(v)^2 \le \max_v\{\mathrm{wt}(v)\}\cdot\sum_v \mathrm{wt}(v) \le M_s^{2-1/(s+1)} \; ,$$

where the second inequality follows by combining Equations (6)–(8) to get an upper bound on $\max_v\{\mathrm{wt}(v)\}$ and applying Claim 5. ◀

The next lemma implies that Condition 2 holds for q as set in Equation (3).

▶ **Lemma 10** (Condition 2 holds). $\min\left\{n^{1-1/s}, \frac{n^{s-1/s}}{M_s^{1-1/s}}\right\} \ge 2m\cdot\frac{M_{2s-1}}{M_s^2}$.

Proof. We can bound M_{2s-1} in two ways. First, by a standard norm inequality, since $s \ge 1$,

$$M_{2s-1} = \sum_v d_v^{2s-1} \le \left(\sum_v d_v^s\right)^{(2s-1)/s} = M_s^{2-1/s} \; . \tag{9}$$

We can also use the trivial bound $d_v \le n$ and get $M_{2s-1} \le n^{s-1} \cdot M_s$. Thus, $M_{2s-1} \le \min\{M_s^{2-1/s}, n^{s-1} \cdot M_s\}$. By applying Hölder's inequality with conjugates $s/(s-1)$ and s we get that

$$2m = \sum_v 1 \cdot d_v \le n^{(s-1)/s} \cdot \left(\sum_v d_v^s\right)^{1/s} = n^{1-1/s} \cdot M_s^{1/s} \,. \tag{10}$$

We multiply the bound by M_{2s-1} to complete the proof. ◀

5 The Degeneracy Connection

The degeneracy, or the coloring number, of a graph $G = (V, E)$ is the maximum value, over all subgraphs G' of G, of the minimum degree in G'. In this definition, we can replace "minimum" by "average" to get a 2-factor approximation to the degeneracy (refer to [26]; Theorem 2.4.4 and Corollary 5.2.3 of [18]). Abusing notation, it will be convenient for us to define $\alpha(G) = \max_{S \subseteq V}\left\{\frac{|E(S)|}{|S|}\right\}$.

We also make the following observation regarding the relation between $\alpha(G)$ and $M_s(G)$.

▶ **Claim 11.** *For every graph G, $\alpha(G) \le M_s(G)^{\frac{1}{s+1}}$.*

In this section, we show that the following setting of parameters for **Moment-estimator**$_s$ satisfies Conditions 1 and 2, for every graph G with degeneracy at most α (i.e., $\alpha(G) \le \alpha$), and for appropriate constants c_r and c_q.

$$r = \frac{c_r}{\varepsilon^2 \cdot \delta} \cdot \min\left\{\frac{n}{M_s^{1/(s+1)}}\ ,\ 2^s \cdot n \cdot \log^2 n \cdot \left(\frac{\alpha}{M_s}\right)^{1/s}\right\}, \tag{11}$$

$$q = \frac{c_q}{\varepsilon^2 \cdot \delta^3} \cdot \min\left\{\frac{n \cdot \alpha}{M_s^{1/s}}, \frac{n^s \cdot \alpha}{M_s}, n^{1-1/s}, \frac{n^{s-1/s}}{M_s^{1-1/s}}\right\} \,. \tag{12}$$

Clearly the setting of r and q in Equation (11) and Equation (12) respectively, can only improve on the setting of r and q for the general case in Equation (3) (Section 4).

Our main challenge is in proving that Condition 1 holds for r as set in Equation (11) (when the graph has degeneracy at most α). Here too, the goal is to upper bound $\sum_v \mathrm{wt}(v)^2$. However, as opposed to the proof of Lemma 9 in Section 4, where we simply obtained an upper bound on $\max_v\{\mathrm{wt}(v)\}$ (and bounded $\sum_v \mathrm{wt}(v)^2$ by $\max_v\{\mathrm{wt}(v)\} \cdot M_s$), here the analysis is more refined, and uses the degeneracy bound. For details see the proof of our main lemma, stated next.

▶ **Lemma 12** (Condition 1 holds). *For a sufficiently large constant c, $\sum_v \mathrm{wt}(v)^2 \le c \cdot 2^s \cdot \alpha^{1/s} \cdot M_s^{2-1/s} \cdot \log^2 n$.*

Proof Sketch. In this extended abstract we only provide the high-level structure of the proof. By the definition of $\mathrm{wt}(v)$, and since $d_v \le d_u$ for every v and $u \in \Gamma^+(v)$,

$$\sum_v \mathrm{wt}(v)^2 = \sum_v \Big(\sum_{u \in \Gamma^+(v)} \left(d_v^{s-1} + d_u^{s-1}\right)\Big)^2 \le 4 \cdot \sum_v \Big(\sum_{u \in \Gamma^+(v)} d_u^{s-1}\Big)^2 \,. \tag{13}$$

In order to bound the expression on the right-hand-side of Equation (13) we partition the vertices (with degree at least 1) according to their degree. Let $U_i \triangleq \{u \in V : d_u \in$

$(2^{i-1}, 2^i]\}$ for $0 \le i \le \lceil \log n \rceil$, and let $\Gamma_i^+(v)$ be a shorthand for $\Gamma^+(v) \cap U_i$. By considering each U_i separately and applying Hölder's inequality we get the following bound for every v.

$$\sum_{u \in \Gamma_i^+(v)} 1 \cdot d_u^{s-1} \le |\Gamma_i^+(v)|^{1/s} \cdot \Big(\sum_{u \in \Gamma_i^+(v)} d_u^s \Big)^{(s-1)/s} \le |\Gamma_i^+(v)|^{1/s} \cdot M_s^{(s-1)/s} . \tag{14}$$

For each i, we also partition the vertices in V according to the number of outgoing edges that they have to U_i. Specifically, for $1 \le j \le \lceil \log(n/\alpha) \rceil$, define $V_{i,j} \triangleq \Big\{ v \in V : |\Gamma_i^+(v)| \in (2^{j-1}\alpha, 2^j\alpha] \Big\}$. Also define $V_{i,0} \triangleq \Big\{ v \in V : |\Gamma_i^+(v)| \le \alpha \Big\}$. Hence, $\{V_{i,j}\}_{j=0}^{\lceil \log(n/\alpha) \rceil}$ is a partition of V for each i.

For a vertex u, let $\Gamma^-(u) \triangleq \{v : u \in \Gamma^+(v)\}$. For two sets of vertices S and T (which are not necessarily disjoint), let $E^+(S,T) \triangleq \{(u,v) : (u,v) \in E^+, u \in S, v \in T\}$. By applying Equation (14) (to one term of the square $\Big(\sum_{u \in \Gamma_i^+(v)} d_u^{s-1}\Big)^2$), and by the definition of $V_{i,j}$, it can be shown that

$$\sum_v \Big(\sum_{u \in \Gamma_i^+(v)} d_u^{s-1} \Big)^2 \le M_s^{(s-1)/s} \cdot \sum_{j=0}^{\lceil \log n \rceil} \Big(\sum_{u \in U_i} d_u^{s-1} \cdot \sum_{v \in \Gamma^-(u) \cap V_{i,j}} |\Gamma_i^+(v)|^{1/s} \Big) . \tag{15}$$

For $j < 2$ we can show that $\sum_{u \in U_i} d_u^{s-1} \sum_{v \in \Gamma^-(u) \cap V_{i,j}} |\Gamma_i^+(v)|^{1/s} \le 2 \cdot \alpha^{1/s} \cdot M_s$. Turning to $j \ge 2$, since all vertices in U_i have degree at most 2^i, we get:

$$\sum_{u \in U_i} d_u^{s-1} \cdot \sum_{v \in \Gamma^-(u) \cap V_{i,j}} |\Gamma_i^+(v)|^{1/s} \le 2^{j/s} \cdot \alpha^{1/s} \cdot 2^{i(s-1)} \cdot |E^+(V_{i,j}, U_i)| . \tag{16}$$

Since G has degeneracy at most α and by the definition of $V_{i,j}$, it can be shown that $|E^+(V_{i,j}, U_i)| \le 2\alpha \cdot |U_i|$, where $U_i = U_i \cap \Big(\bigcup_{v \in V_{i,j}} \Gamma^+(v) \Big)$. Furthermore, the definition of U_i (together with the degeneracy bound and the definition of M_s) implies that $|U_i| \le M_s \cdot 2^{-((i-1)(s-1)+j)} \cdot \alpha^{-1}$. The lemma follows by combining Equation (13) with Equation (15) and the above bounds for $j < 2$ and $j \ge 2$. ◀

The next lemma, which establishes Condition 2, can be proved similarly to Lemma 10.

▶ **Lemma 13** (Condition 2 holds)**.**

$$\min \left\{ \frac{n \cdot \alpha}{M_s^{1/s}}, \frac{n^s \cdot \alpha}{M_s}, n^{1-1/s}, \frac{n^{s-1/s}}{M_s^{1-1/s}} \right\} \ge m \cdot \frac{M_{2s-1}}{M_s^2} .$$

6 Wrapping things up

The proof of our final result, Theorem 3, follows by combining Theorem 8, Lemma 9, Lemma 12 and Lemma 13, with a geometric search for a factor-2 estimate of M_s (which determines the correct setting of r and q in the algorithm).

References

1 A. S. Aliakbarpour, M.and Biswas, T. Gouleakis, J. Peebles, R. Rubinfeld, and A. Yodpinyanee. Sublinear-time algorithms for counting star subgraphs via edge sampling. *Algorithmica*, pages 1–30, 2017. doi:10.1007/s00453-017-0287-3.

2 N. Alon and S. Gutner. Linear time algorithms for finding a dominating set of fixed size in degenerated graphs. In *Proceedings of the Annual International Conference Computing and Combinatorics (COCOON)*, pages 394–405, 2008.

3 N. Alon, Y. Matias, and M. Szegedy. The space complexity of approximating the frequency moments. *Journal of Computer and System Sciences*, 58(1):137–147, 1999.

4 Arboricity. Wikipedia. `https://en.wikipedia.org/wiki/Arboricity`.

5 A.-L. Barabasi and R. Albert. Emergence of scaling in random networks. *Science*, 286:509–512, October 1999.

6 J. W. Berry, L. A. Fostvedt, D. J. Nordman, C. A. Phillips, C. Seshadhri, and A. G. Wilson. Why do simple algorithms for triangle enumeration work in the real world? *Internet Mathematics*, 11(6):555–571, 2015.

7 Z. Bi, C. Faloutsos, and F. Korn. The dgx distribution for mining massive, skewed data. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pages 17–26. ACM, 2001.

8 P. Bickel, A. Chen, and E. Levina. The method of moments and degree distributions for network models. *Annals of Statistics*, 39(5):2280–2301, 2011.

9 P. Brach, M. Cygan, J. Lacki, and P. Sankowski. Algorithmic complexity of power law networks. In *Proceedings of the Annual Symposium on Discrete Algorithms (SODA)*, pages 1306–1325, 2016.

10 A. Broder, R. Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins, and J. Wiener. Graph structure in the web. *Computer Networks*, 33:309–320, 2000.

11 B. Chazelle, R. Rubinfeld, and L. Trevisan. Approximating the minimum spanning tree weight in sublinear time. *SIAM Journal on Computing*, 34(6):1370–1379, 2005.

12 N. Chiba and T. Nishizeki. Arboricity and subgraph listing algorithms. *SIAM J. Comput.*, 14:210–223, 1985.

13 F. Chierichetti, A. Dasgupta, R. Kumar, S. Lattanzi, and T. Sarlos. On sampling nodes in a network. In *Proceedings of the International Conference on World Wide Web (WWW)*, pages 471–481, 2016.

14 A. Clauset, C. R. Shalizi, and M. E. J. Newman. Power-law distributions in empirical data. *SIAM Review*, 51(4):661–703, 2009.

15 A. Czumaj, F. Ergun, L. Fortnow, A. Magen, I. Newman, R. Rubinfeld, and C. Sohler. Approximating the weight of the euclidean minimum spanning tree in sublinear time. *SIAM Journal on Computing*, 35(1):91–109, 2005.

16 A. Czumaj and C. Sohler. Estimating the weight of metric minimum spanning trees in sublinear time. *SIAM Journal on Computing*, 39(3):904–922, 2009.

17 A. Dasgupta, R. Kumar, and T. Sarlos. On estimating the average degree. In *Proceedings of the International Conference on World Wide Web (WWW)*, pages 795–806, 2014.

18 R. Diestel. *Graph Theory.* Springer, fourth edition edition, 2010.

19 T. Eden, A. Levi, D. Ron, and C. Seshadhri. Approximately counting triangles in sublinear time. In *Proceedings of the Annual Symposium on Foundations of Computer Science (FOCS)*, pages 614–633, 2015.

20 D. Eppstein, M. Loffler, and D. Strash. Listing all maximal cliques in sparse graphs in near-optimal time. In *International Symposium on Algorithms and Computation (ISAAC)*, pages 403–413, 2010.

21 P. Erdos and T. Gallai. Graphs with prescribed degree of vertices (hungarian). *Mat. Lapok*, 11:264–274, 1960.

22 M. Faloutsos, P. Faloutsos, and C. Faloutsos. On power-law relationships of the internet topology. In *Proceedings of Computer Communication Review (SIGCOMM)*, pages 251–262. ACM, 1999.

23 U. Feige. On sums of independent random variables with unbounded variance and estimating the average degree in a graph. *SIAM Journal on Computing*, 35(4):964–984, 2006.

24 O. Goldreich and D. Ron. Approximating average parameters of graphs. *Random Structures and Algorithms*, 32(4):473–493, 2008.

25 M. Gonen, D. Ron, and Y. Shavitt. Counting stars and other small subgraphs in sublinear-time. *SIAM Journal on Discrete Math*, 25(3):1365–1411, 2011.

26 Graph degeneracy. Wikipedia. `https://en.wikipedia.org/wiki/Degeneracy_(graph_theory)`.

27 S. L. Hakimi. On the realizability of a set of integers as degrees of the vertices of a graph. *SIAM Journal Applied Mathematics*, 10:496–506, 1962.

28 A. Hassidim, J. A. Kelner, H. N. Nguyen, and K. Onak. Local graph partitions for approximation and testing. In *Proceedings of the Annual Symposium on Foundations of Computer Science (FOCS)*, pages 22–31. IEEE, 2009.

29 V. Havel. A remark on the existence of finite graphs (czech). *Casopis Pest. Mat.*, 80:477–480, 1955.

30 S. Marko and D. Ron. Approximating the distance to properties in bounded-degree and general sparse graphs. *ACM Transactions on Algorithms*, 5(2), 2009.

31 D. Matula and L. Beck. Smallest-last ordering and clustering and graph coloring algorithms. *Journal of the ACM (JACM)*, 30(3):417–427, 1983.

32 J. Nešetřil and P. Ossana de Mendez. *Sparsity.* Springer, 2010.

33 H. N. Nguyen and K. Onak. Constant-time approximation algorithms via local improvements. In *Proceedings of the Annual Symposium on Foundations of Computer Science (FOCS)*, pages 327–336. IEEE, 2008.

34 K. Onak, D. Ron, M. Rosen, and R. Rubinfeld. A near-optimal sublinear-time algorithm for approximating the minimum vertex cover size. In *Proceedings of the Annual Symposium on Discrete Algorithms (SODA)*, pages 1123–1131. SIAM, 2012.

35 M. Parnas and D. Ron. Approximating the minimum vertex cover in sublinear time and a connection to distributed algorithms. *Theoretical Computer Science*, 381(1-3):183–196, 2007.

36 D. Pennock, G. Flake, S. Lawrence, E. Glover, and C. L. Giles. Winners don't take all: Characterizing the competition for links on the web. *Proceedings of the national academy of sciences (PNAS)*, 99(8):5207–5211, 2002.

37 A. Sala, L. Cao, C. Wilson, R. Zablit, H. Zheng, and B. Y. Zhao. Measurement-calibrated graph models for social network experiments. In *Proceedings of the International Conference on World Wide Web (WWW)*, pages 861–870. ACM, 2010.

38 O. Simpson, C. Seshadhri, and A. McGregor. Catching the head, tail, and everything in between: A streaming algorithm for the degree distribution. In *Proceedings on the International Conference on Data Mining (ICDM)*, pages 979–984, 2015.

39 Y. Yoshida, M. Yamamoto, and H. Ito. An improved constant-time approximation algorithm for maximum-matchings. In *Proceedings of the Annual Symposium on the Theory of Computing (STOC)*, pages 225–234. ACM, 2009.