On Sampling Edges Almost Uniformly

Talya Eden[1] and Will Rosenbaum[2]

1 **Tel Aviv University, Tel Aviv, Israel**
`talyaa01@gmail.com`
2 **Tel Aviv University, Tel Aviv, Israel**
`will.rosenbaum@gmail.com`

Abstract

We consider the problem of sampling an edge almost uniformly from an unknown graph, $G = (V, E)$. Access to the graph is provided via queries of the following types: (1) uniform vertex queries, (2) degree queries, and (3) neighbor queries. We describe a new simple algorithm that returns a random edge $e \in E$ using $\tilde{O}(n/\sqrt{\varepsilon m})$ queries in expectation, such that each edge e is sampled with probability $(1 \pm \varepsilon)/m$. Here, $n = |V|$ is the number of vertices, and $m = |E|$ is the number of edges. Our algorithm is optimal in the sense that any algorithm that samples an edge from an almost-uniform distribution must perform $\Omega(n/\sqrt{m})$ queries.

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, Graph Algorithms, Sampling Edges, Query Complexity

Digital Object Identifier 10.4230/OASIcs.SOSA.2018.7

1 Introduction

Suppose $G = (V, E)$ is a very large graph—too large to store in our local memory. Access to G is granted in accordance with the standard bounded degree graph query model of Goldreich and Ron [5] via queries of the following types: (1) sample a uniformly random vertex (*vertex queries*), (2) query for the i^{th} neighbor of a vertex v (*neighbor queries*), and (3) query the degree of a given vertex[1] (*degree queries*).[2] The query access model readily gives access to uniformly random vertices, but what if we are interested in sampling a uniformly random *edge* from E? How many queries are necessary and sufficient?

We describe an $\tilde{O}(n/\sqrt{\varepsilon m})$ time algorithm for sampling an edge in a graph such that each edge is sampled with almost equal probability. Our main result shows that it is possible to sample edges from a distribution which has *bias at most* ε, a notion formalized in the following definition.

▶ **Definition 1.** Let Q be a fixed probability distribution on a finite set Ω. We say that a probability distribution P is *pointwise ε-close to Q* if for all $x \in \Omega$,

$$|P(x) - Q(x)| \leq \varepsilon Q(x)\,, \quad \text{or equivalently} \quad 1 - \varepsilon \leq \frac{P(x)}{Q(x)} \leq 1 + \varepsilon\,.$$

If $Q = U$, the uniform distribution on Ω, then we say that P has *bias at most* ε.

[1] We note that degree queries can be implemented by performing $O(\log n)$ neighbor queries per degree query.
[2] One may also consider the more powerful "general graph model" of Parnas and Ron [8] that additionally allows *pair queries*. Indeed, our lower bound holds in the general graph model. Interestingly, our tight upper bound does not require the additional computational power afforded by pair queries.

▶ **Theorem 2.** *Let $G = (V, E)$ be an arbitrary graph with n vertices and m edges. There exists an algorithm that given n, $0 < \varepsilon < \frac{1}{2}$ and access to vertex, degree, and neighbor queries returns an edge with probability at least $2/3$, such that each returned edge is sampled according to a distribution P that has bias at most ε. The expected query complexity and running time of the algorithm are $\tilde{O}(n/\sqrt{\varepsilon m})$.*

The strength of the approximation guarantee of the algorithm in Theorem 2 allows us to obtain the following corollary for sampling weighted edges in graphs.

▶ **Corollary 3.** *Let $G = (V, E)$ and ε be as in Theorem 2, and let $w : E \to \mathbf{R}$ be an arbitrary function. Let P be the distributions on edges induced by the algorithm of Theorem 2. Then*

$$\left| \mathop{\mathbf{E}}_{e \sim P}(w(e)) - \mathop{\mathbf{E}}_{e \sim U}(w(e)) \right| \leq \varepsilon \left| \mathop{\mathbf{E}}_{e \sim U}(w(e)) \right| .$$

In Section 5, we show that the algorithm of Theorem 2 is essentially optimal, in the sense that any algorithm which returns an edge from E almost uniformly must use $\Omega(n/\sqrt{m})$ queries. This lower bound applies in the strictly more powerful general graph query model of Parnas and Ron [8], and even if the algorithm is only required to sample edges from a distribution that is close to uniform in total variational distance.

1.1 Related work

An algorithm for sampling an edge in a graph almost uniformly was first suggested by [7]. In [7], Kaufman et al. use random edge samples in order to test if a graph is bipartite. In particular, they devise a subroutine that guarantees that all but a small fraction of the edges are each sampled with probability $\Omega(1/m)$. More recently, in [1], Eden et al. use random edge sampling as a subroutine in their algorithm for approximating the number of triangles in a graph. A similar subroutine for random edge sampling is employed in [2], where the authors use edge sampling to approximate moments of the degree distribution of a graph. In all of the works [7, 1, 2], the authors avoid the "difficult task of selecting random edges from the entire graph," [1] by instead sampling edges from a smaller subgraph.

Our algorithm improves upon and simplifies the edge sampling procedures in the works cited above. Our approximation guarantee in terms of *pointwise* distance to uniformity is strictly stronger than the guarantees of any of these papers. In particular, the previous subroutines do not return edges with two high degree endpoints in the case of highly irregular graphs. However, such edges may be of significant interest in practice. Further, the subgraph sampling strategy used in these subroutines is memory intensive—a fairly large set of vertices must be sampled (and stored), and a random edge incident to the set is sampled. In contrast, each query made by our algorithm depends only on a constant number of previous queries. Thus our algorithm can be implemented using poly-logarithmic space, and can easily be parallelized.

1.2 Overview of the algorithm

We treat each undirected edge $\{u, v\}$ as a pair of directed edges, (u, v) and (v, u). For each vertex u, let $d(u)$ denote its (undirected) degree, and let $m = \sum_{u \in V} d(u) = 2|E|$ be the number of directed edges. Consider the following two process for sampling edges:

Process 1 Choose a vertex u uniformly at random, and choose v uniformly from u's neighbors. Return (u, v).

Process 2 Choose a vertex u uniformly at random and i uniformly from $\{1, \ldots, n - 1\}$. If $i \leq d(u)$, return (u, v) where v is u's i^{th} neighbor. Otherwise, fail.

In Process 1, each (directed) edge (u,v) is sampled with probability $1/(n\,d(u))$, thus biasing the sample towards edges originating from vertices with low degrees. Process 2 eliminates this bias, as each edge is sampled with probability $1/n(n-1)$. However, Process 2 only succeeds with probability $m/n(n-1)$. We can improve the success probability of Process 2 by sampling $i \in [\theta]$ for $\theta < n-1$, with the caveat that some edges incident with high-degree nodes will never be sampled. The idea of our algorithm is to choose $\theta = \sqrt{m/\varepsilon}$ and sample edges emanating from high and low degree vertices separately.

We call a vertex v *light* if $d(v) \leq \sqrt{m/\varepsilon}$, and *heavy* otherwise. Similarly a directed edge (u,v) is light (resp. heavy) if u is light (resp. heavy). We attempt to sample a light edge by using the modified Process 2 above with $\theta = \sqrt{m/\varepsilon}$, and failing if the sampled vertex u is heavy. Thus, each light edge (u,v) is sampled with probability $1/(n\sqrt{m/\varepsilon})$.

In order to sample a heavy edge we use the following procedure. We first sample a light edge (u,v) as described above. If v is heavy, we then query for a random neighbor w of v. The probability of hitting some specific heavy edge (v,w) is $\frac{d^{\mathcal{L}}(v)}{n\sqrt{m/\varepsilon}} \cdot \frac{1}{d(v)}$, where $d^{\mathcal{L}}(v)$ denotes the number of light neighbors of v. The threshold $\sqrt{m/\varepsilon}$ is set as to ensure that for every heavy vertex, at most an ε-fraction of its neighbors are heavy. It follows that $d^{\mathcal{L}}(v) \approx d(v)$, and thus each heavy edge (v,w) is chosen with probability roughly $1/(n\sqrt{m/\varepsilon})$.

Our algorithm invokes the procedures for sampling light and heavy edges, each with equal probability, sufficiently many times to ensure that an edge is returned with large constant probability. In our analysis, we show that the induced distribution on edges has bias at most ε.

1.3 Overview of the lower bound

The construction of the lower bound is similar to the lower bound construction of [4]. For an arbitrary graph G' on n' vertices with m' edges, let G be the graph obtained by adding a clique on $\Theta(\sqrt{m'})$ vertices to G'. Since the clique contains a constant fraction of G's edges, any almost-uniform edge sampler must return a clique edge with constant probability. The probability of sampling a clique vertex is $O(\sqrt{m}/n)$, so any algorithm that returns an edge according to an almost uniform distribution must perform $\Omega(n/\sqrt{m})$ queries.

2 Preliminaries

Let $G = (V,E)$ be an undirected graph with $n = |V|$ vertices. We treat each undirected edge $\{u,v\}$ in E as pair of directed edges (u,v), (v,u) from u to v and from v to u, respectively. We denote the (undirected) degree of a vertex $v \in V$ by $d(v)$. Thus, the number of (directed) edges in G is $m = \sum_{v\in V} d(v) = 2|E|$. For $v \in V$, we denote the set of neighbors of v by $\Gamma(v)$. We assume that each v has an arbitrary but fixed order on $\Gamma(v)$ so that we may refer unambiguously to v's i^{th} neighbor for $i = 1, 2, \ldots, d(v)$.

We partition V and E into sets of light and heavy elements depending on their degrees.

▶ **Definition 4.** We say that a vertex u is a *light vertex* if $d(u) \leq \sqrt{m/\varepsilon}$ and otherwise we say that it is a *heavy vertex.* We say that an edge is a *light edge* if it originates from a light vertex. A *heavy edge* is defined analogously. Finally, we denote the sets of light and heavy vertices by $\mathcal{L}$ and $\mathcal{H}$ respectively, and the sets of light and heavy edges by $E_{\mathcal{L}}$ and $E_{\mathcal{H}}$ respectively.

Note that for a fixed *un*directed edge $\{u, v\}$, it may be the case that one of the corresponding directed edges, say (u, v), is light while the other, (v, u), is heavy. Specifically, this will occur if u is a light vertex and v is a heavy vertex.

The algorithms we consider access a graph G via queries. Our algorithm uses the following queries:

1. Vertex query: returns a uniformly random vertex $v \in V$.
2. Degree query: given a vertex $v \in V$, returns $d(v)$.
3. Neighbor query: given $v \in V$ and $i \in \mathbf{N}$, return v's i^{th} neighbor; if $i > d(v)$, this operation returns *fail*.

Our lower bounds apply additionally to a computational model that allows pair queries.

4. Pair query: given $v, w \in V$, returns *true* if $(v, w) \in E$; otherwise returns *false*.

The (expected) query cost of an algorithm $\mathcal{A}$ is the (expected) number of queries that $\mathcal{A}$ makes before terminating. We make no restrictions on the computational power of $\mathcal{A}$ except for the number of queries $\mathcal{A}$ makes to G. The query complexity of a task is the minimum query cost of any algorithm which performs the task.

We will require the following lemma, which gives sufficient conditions for a probability distribution P to have bias at most ε (recall Definition 1).

▶ **Lemma 5.** *Let P be a probability distribution over a finite set Ω which satisfies*

$$1 - \varepsilon \leq \frac{P(x)}{P(y)} \leq 1 + \varepsilon \quad \text{for all } x, y \in \Omega\,.$$

Then P has bias at most ε.

Proof. Suppose P satisfies the hypothesis of the lemma. Then

$$1 - \varepsilon \leq \frac{P(x)}{P(y)} \leq 1 + \varepsilon \implies (1 - \varepsilon)\frac{P(y)}{U(x)} \leq \frac{P(x)}{U(x)} \leq (1 + \varepsilon)\frac{P(y)}{U(x)}.$$

Summing the second expression over all $y \in \Omega$ gives

$$(1 - \varepsilon)\frac{1}{U(x)} \leq \frac{P(x)}{U(x)} \cdot \frac{1}{U(x)} \leq (1 + \varepsilon)\frac{1}{U(x)}\,.$$

The factor of $1/U(x)$ appears in the middle term because we sum over $|\Omega| = 1/U(x)$ terms. Hence P has bias at most ε. ◀

3 The Basic Algorithm

We start by presenting our main algorithm – Sample-edge-almost-uniformly– that samples a random edge in E almost uniformly with high probability. The algorithm is given query access to G and takes n, m and ε as parameters. For simplicity of presentation, we assume that m is known precisely. In Section 4, we show that this assumption is unnecessary.

We defer the statement of the lemma and proof regarding the correctness of the algorithm to the end of the section, and first present the two subroutines used in the algorithm Sample-light-edge and Sample-heavy-edge, for sampling a uniform light edge and an almost uniform heavy edge, respectively.

▶ **Lemma 6.** *The procedure* Sample-light-edge *performs a constant number of queries and succeeds with probability $|E_{\mathcal{L}}| / (n\sqrt{m/\varepsilon})$. In the case where* Sample-light-edge *succeeds, the edge returned is uniformly distributed in $E_{\mathcal{L}}$.*

Algorithm 1 Sample-edge-almost-uniformly(n, m, ε)

```
1. For $i = 1$ to $q = \frac{10n}{(1-\varepsilon)\sqrt{\varepsilon m}}$ do:
   a. With probability 1/2 invoke Sample-light-edge$(m)$ and with probability 1/2
      invoke Sample-heavy-edge$(m)$.
   b. If an edge $(u, v)$ was returned, then **return** $(u, v)$.
2. **Return** *fail*.
```

Algorithm 2 Sample-light-edge(m)

```
1. Sample a vertex $u \in V$ uniformly at random and query for its degree.
2. If $d(u) > \sqrt{m/\varepsilon}$ **return** *fail*.
3. Choose a number $j \in \left[\sqrt{m/\varepsilon}\right]$ uniformly at random.
4. Query for the $j^{\text{th}}$ neighbor of $u$.
5. If no vertex was returned then **return** *fail*. Otherwise, let $v$ be the returned
   vertex.
6. **Return** $(u, v)$.
```

Algorithm 3 Sample-heavy-edge(m)

```
1. Sample a vertex $u \in V$ uniformly at random and query for its degree.
2. If $d(u) > \sqrt{m/\varepsilon}$ **return** *fail*.
3. Choose a number $j \in \left[\sqrt{m/\varepsilon}\right]$ uniformly at random.
4. Query for the $j^{\text{th}}$ neighbor of $u$.
5. If no vertex was returned or if the returned vertex is light then **return** *fail*.
   Otherwise, let $v$ be the returned vertex.
6. Sample $w$ a random neighbor of $v$.
7. **Return** $(v, w)$.
```

Proof. Suppose a light vertex u is sampled in Step 1 of the procedure. Then the probability that we obtain a neighbor of u in Step 4 is $d(u)/\sqrt{m/\varepsilon}$. Hence,

$$\Pr[\text{ Success }] = \sum_{u \in \mathcal{L}} \frac{1}{n} \cdot \frac{d(u)}{\sqrt{m/\varepsilon}} = \frac{|E_{\mathcal{L}}|}{n\sqrt{m/\varepsilon}} .$$

In any invocation of the algorithm, the probability that a particular (directed) edge e is returned is $1/(n\sqrt{m/\varepsilon})$ if e is light, and 0 otherwise. Thus, each light edge is returned with equal probability. ◀

▶ **Lemma 7.** *The procedure* Sample-heavy-edge *performs a constant number of queries and succeeds with probability in* $\left[(1-\varepsilon)\frac{|E_{\mathcal{H}}|}{n\sqrt{m/\varepsilon}}, \frac{|E_{\mathcal{H}}|}{n\sqrt{m/\varepsilon}}\right]$. *In the case where* Sample-heavy-edge *succeeds, the edge returned is distributed according to a distribution P that has bias at most ε.*

Proof. Since each $v \in \mathcal{H}$ satisfies $d(v) > \sqrt{m/\varepsilon}$, we have $|\mathcal{H}| < m/\sqrt{m/\varepsilon} = \sqrt{\varepsilon m}$. For every vertex $v \in \mathcal{H}$, let $d^{\mathcal{L}}(v)$ denote the cardinality of $\Gamma(v) \cap \mathcal{L}$, the set of light neighbors of v. Similarly, let $d^{\mathcal{H}}(v)$ denote $|\Gamma(v) \cap \mathcal{H}|$. Thus, $d^{\mathcal{H}}(v) \leq |\mathcal{H}| < \sqrt{\varepsilon m} < \varepsilon d(v)$.

Since $d^{\mathcal{L}}(v) + d^{\mathcal{H}}(v) = d(v)$ for every v, we have

$$d^{\mathcal{L}}(v) > (1-\varepsilon)\, d(v) . \tag{1}$$

Each vertex $v \in \mathcal{H}$ is chosen in Step 5 with probability $d^{\mathcal{L}}(v)/(n\sqrt{m/\varepsilon})$. Therefore, the probability that $e = (v, w)$ is chosen in Step 6 satisfies

$$\Pr[e \text{ is returned}] = \frac{d^{\mathcal{L}}(v)}{n\sqrt{m/\varepsilon}} \cdot \frac{1}{d(v)} > \frac{(1-\varepsilon)}{n\sqrt{m/\varepsilon}}\,, \tag{2}$$

where the inequality follows from Equation (1). On the other hand, $d^{\mathcal{L}}(v) \le d(v)$ implies that $\Pr[e \text{ is returned}] \le 1/\left(n\sqrt{m/\varepsilon}\right)$. Finally, we bound the success probability by

$$\Pr[\text{ Success }] = \Pr\left[\bigcup_{e \in E_{\mathcal{H}}} \{e \text{ is returned}\}\right] = \sum_{e \in E_{\mathcal{H}}} \Pr[e \text{ is returned}] > \frac{(1-\varepsilon)\,|E_{\mathcal{H}}|}{n\sqrt{m/\varepsilon}}\,.$$

The second equality holds because the events $\{e \text{ is returned}\}$ are disjoint, while the inequality holds by Equation (2). ◀

Assuming that m is known to the algorithm Sample-edge-almost-uniformly, Theorem 2 is an immediate consequence of the following lemma. In Section 4, we consider the case where m is not initially known to the algorithm and prove Theorem 2 in its full generality.

▶ **Lemma 8.** *For any ε satisfying $0 < \varepsilon < 1/2$,* Sample-edge-almost-uniformly *returns an edge with probability at least $2/3$. If an edge is returned, then it is distributed according to a distribution P that has bias at most 2ε.*

Proof. We first prove that the induced distribution on edges has bias at most 2ε. By Lemmas 6 and 7, the probability of successfully returning an edge in Step 1a satisfies

$$\begin{aligned}\Pr[\text{ Success }] &= \frac{1}{2}\Pr[\text{ Sample-light-edge succeeds }] + \frac{1}{2}\Pr[\text{ Sample-heavy-edge succeeds }] \\ &> \frac{1}{2}\cdot\frac{|E_{\mathcal{L}}|}{n\sqrt{m/\varepsilon}} + \frac{1}{2}\cdot\frac{(1-\varepsilon)\,|E_{\mathcal{H}}|}{n\sqrt{m/\varepsilon}} \ge (1-\varepsilon)\frac{m}{2n\sqrt{m/\varepsilon}}\,.\end{aligned}$$

The second inequality holds because $|E_{\mathcal{L}}| + |E_{\mathcal{H}}| = m$. Also by Lemmas 6 and 7, the probability, p_e, that a specific edge e is returned satisfies

$$\frac{1-\varepsilon}{2n\sqrt{m/\varepsilon}} \le p_e \le \frac{1}{2n\sqrt{m/\varepsilon}}.$$

Thus, the distribution on sampled edges satisfies

$$1-\varepsilon \le \frac{P(e)}{P(e')} \le \frac{1}{1-\varepsilon} \le 1+2\varepsilon\,, \quad \text{for all } e, e' \in E.$$

Therefore, P has bias at most 2ε by Lemma 5.

We now prove that the algorithm returns an edge with probability at least $2/3$. Let χ_i be the indicator variable for the event that an edge (u, v) was returned in the i^{th} step of the for loop of the algorithm. By Lemmas 6 and 7,

$$\Pr[\chi_i = 0 \text{ for all } i] \le \left(1 - \frac{(1-\varepsilon)\sqrt{\varepsilon m}}{2n}\right)^{\frac{10n}{(1-\varepsilon)\sqrt{\varepsilon m}}} < 1/3.$$

Finally, since every invocation of Sample-light-edge and Sample-heavy-edge takes a constant number of queries, the query complexity and running time of the algorithm are $O(n/\sqrt{\varepsilon m})$. ◀

4 Sampling Edges with Unknown m

In the previous section, we assumed that the value of m (or more specifically, $\sqrt{m/\varepsilon}$) was known to the algorithm. In this section, we argue that such an assumption is unnecessary. In particular, it is sufficient to have any estimate $\widehat{m} \in [m, cm]$ for a fixed constant c. Such an estimate can be obtained with high probability using $\tilde{O}(n/\sqrt{m})$ expected queries by employing an algorithm of Goldreich and Ron [6].[3]

▶ **Theorem 9** (Goldreich & Ron [6]). *Let $G = (V, E)$ be a graph with n vertices and m edges. There exists an algorithm that uses $\tilde{O}(n/\sqrt{m})$ vertex, degree, and neighbor queries in expectation and outputs an estimate $\widehat{m}$ of m that with probability at least $2/3$ satisfies $m \le \widehat{m} \le 2m$.*

Analogues of Lemmas 6 and 7 hold for any estimate $\widehat{m}$ of m, with the threshold between light and heavy vertices redefined to be $\sqrt{\widehat{m}/\varepsilon}$. In particular, if $\widehat{m}$ is an overestimate—i.e., $\widehat{m} > m$—then the approximation guarantees of both lemmas are still satisfied. However, an overestimate results in a smaller success probability (by a factor of $\sqrt{m/\widehat{m}}$). It is straightforward to verify that so long as $m \le \widehat{m} \le 2m$, the conclusion of Lemma 8 still holds (for complete details please see the full version of the paper [3]). Using Lemma 8 and Theorem 9, we can prove Theorem 2 in its full generality.

Proof of Theorem 2. Let $\widehat{m}$ be an estimate of m. We call $\widehat{m}$ a *good* estimate if it satisfies $m \le \widehat{m} \le 2m$. By repeating the algorithm of Goldreich & Ron (Theorem 9) $O(\log(n/\varepsilon))$ times, then taking $\widehat{m}$ to be the median value reported, a straightforward application of Chernoff bounds guarantees that $\widehat{m}$ is good with probability at least $1 - \varepsilon/2n^2$. If $\widehat{m}$ is good, by Lemma 8, calling Sample-edge-almost-uniformly$(n, \widehat{m}, \varepsilon/4)$ will successfully return an edge e with probability at least $2/3$, and the returned edge is distributed according to a distribution P which has bias at most $\varepsilon/2$.

Let Q be the distribution of returned edges. If $\widehat{m}$ is not good, we have no guarantee of the success probability of returning an edge, nor of the distribution from which the edge is drawn. However, since $\widehat{m}$ is bad with probability at most $\varepsilon/2n^2$, for each e we can bound

$$|Q(e) - U(e)| \le |Q(e) - P(e)| + |P(e) - U(e)| \le \frac{\varepsilon}{2n^2} + \frac{\varepsilon}{2m} \le \frac{\varepsilon}{m} = \varepsilon U(e).$$

Thus Q has bias at most ε.

We now turn to analyze the expected query cost of the algorithm. By Theorem 9, the expected query cost of Goldreich and Ron's algorithm is $\tilde{O}(n/\sqrt{m})$. By Lemmas 6 and 7, the procedures Sample-light-edge and Sample-heavy-edge each perform a constant number of queries per invocation. Hence, the query cost of the algorithm is $O(q) = \tilde{O}\left(n/\sqrt{\varepsilon \widehat{m}}\right)$. If $\widehat{m}$ is good then q is at most $\tilde{O}\left(n/\sqrt{\varepsilon m}\right)$, and otherwise it is at most $O(n)$. Since m is good with probability at least $1 - \varepsilon/2n^2$, it follows that the expected query cost is $\tilde{O}(n/\sqrt{\varepsilon m})$. ◀

5 A Lower Bound

In this section we prove a lower bound on the number of queries necessary to sample an edge from an almost-uniform distribution over E. Specifically we show that any algorithm

[3] An earlier result of Feige [4] would also suffice, but we found the result of Goldreich and Ron simpler to apply.

$\mathcal{A}$ that samples an edge almost uniformly must perform $\Omega\left(n/\sqrt{m}\right)$ queries, even if $\mathcal{A}$ is given m. Thus, the algorithm we present is asymptotically optimal (up to poly-logarithmic factors). Further our lower bound applies to (1) strictly weaker approximation to the uniform distribution (by total variational distance), and (2) to algorithms which are additionally allowed "pair queries" at unit cost.

We first recall the definition of the total variational distance between two distributions.

▶ **Definition 10.** Let P and Q be probability distributions over a finite set Ω. We denote the *total variational distance* or *statistical distance* between P and Q by

$$\mathrm{dist}_{\mathrm{TV}}(P,Q) = \frac{1}{2}\left\|P-Q\right\|_1 = \frac{1}{2}\sum_{x\in\Omega}\left|P(x)-Q(x)\right|.$$

Observe that if P and Q are pointwise ε-close, then $\mathrm{dist}_{\mathrm{TV}}(P,Q)\le\varepsilon$, but the converse is not true in general.

▶ **Theorem 11.** *Let $\varepsilon < 1/2$ be fixed and suppose $\mathcal{A}$ is an algorithm that performs $q = q(n,m)$ vertex, degree, neighbor, or pair queries and with probability at least $2/3$ returns an edge $e \in E$ sampled according to a distribution P over E. If for all $G = (V,E)$, P satisfies $\mathrm{dist}_{\mathrm{TV}}(P,U) < \varepsilon$, then $q = \Omega(n/\sqrt{m})$.*

Proof. The result is trivial if $m = \Omega(n^2)$, so we assume that $m = o(n^2)$. Suppose there exists an algorithm $\mathcal{A}$ that performs t queries and with probability at least $2/3$ returns an edge sampled from a distribution P satisfying $\mathrm{dist}_{\mathrm{TV}}(P,U)\le\varepsilon$. Let G' be an arbitrary graph, and let n' and m' denote the number of vertices and edges, respectively, in G'. Let K be a clique on $k=\sqrt{2m'}$ nodes. Let $G = G'\cup K$ be the disjoint union of G' and K, and let V_K and E_K denote the vertices and edges of K in G. Thus G has $n = n'+k$ nodes, $m > 2m'$ edges, and E_K contains at least $m/2$ edges. The remainder of the proof formalizes the intuition that since $\mathcal{A}$ makes relatively few queries, it is unlikely to sample vertices in V_K. Thus $\mathcal{A}$ must sample edges from E_K with probability significantly less than $1/2$.

Assume that the vertices are assigned distinct labels from $[n]$ uniformly at random, independently of any decisions made by the algorithm $\mathcal{A}$. Let $q_1,\ldots,q_t$ denote the set of queries that the algorithm performs, and let $a_1,\ldots,a_t$ denote the corresponding answers. We say that a query-answer pair (q_i,a_i) is a *witness pair* if (1) q_i is a degree query of $v\in V_K$, or (2) q_i is a neighbor query for some $v\in V_k$, or (3) q_i is a pair query for some $(v,w)\in E_K$. For $i\in[t]$ let NW_i denote that event that $(q_1,a_1),\ldots,(q_i,a_i)$ are not witness pairs, and let $\mathsf{NW}=\mathsf{NW}_t$. Let A_K be the event that $\mathcal{A}$ returns some edge $e\in E_K$. Since $\mathrm{dist}_{\mathrm{TV}}(P,U)<\varepsilon$, we must have $\left|\Pr_P[A_K]-\Pr_U[A_K]\right|\le\varepsilon$. Since $|E_K|\ge m/2$, we have $\Pr_U[A_K]\ge 1/2$, hence

$$\Pr_P[A_k]\ge\frac{1}{2}-\varepsilon. \tag{3}$$

The law of total probability gives

$$\Pr_P[A_K]=\Pr_P[\mathsf{NW}]\cdot\Pr_P[A_K\mid\mathsf{NW}]+\Pr_P[\mathsf{NW}^c]\cdot\Pr_P[A_K\mid\mathsf{NW}^c], \tag{4}$$

where NW^c denotes the complement of the event NW.

Claim. $\Pr_P[A_K\mid\mathsf{NW}]=o(1)$.

Proof of Claim. Suppose the event NW occurs, i.e., $\mathcal{A}$ does not observe a witness pair after t queries. Thus, if $\mathcal{A}$ returns an edge $e=(u,v)\in E_K$, it cannot have made queries involving u or v. We can therefore bound

$$\Pr_P[A_K\mid\mathsf{NW}]\le|E_K|\,\frac{1}{(n-2t)^2}\le\frac{m}{2(n-2t)^2}\le\frac{2m}{n^2}.$$

The first inequality holds because the identities of any $u, v \in V_K$ are uniformly distributed among the (at least) $n - 2t$ vertices not queried by $\mathcal{A}$. The second inequality holds assuming $t < n/4$. The claim follows from the assumption that $m = o(n^2)$.

Combining the result of the claim with equations (3) and (4) gives

$$\frac{1}{2} - \varepsilon \leq \Pr_P[A_k] = \Pr_P[\mathsf{NW}] \cdot \Pr_P[A_K \mid \mathsf{NW}] + \Pr_P[\mathsf{NW}^c] \cdot \Pr_P[A_K \mid \mathsf{NW}^c] \leq \Pr_P[\mathsf{NW}^c] + o(1)\,. \quad (5)$$

We bound $\Pr_P[\mathsf{NW}^c]$ by

$$\begin{aligned}\Pr[\mathsf{NW}^c] &= \Pr\left[\bigcup_{i \leq t} \{(q_i, a_i) \text{ is the first witness pair}\}\right] \\ &= \sum_{i \leq t} \Pr[(q_i, a_i) \text{ is a witness pair} \mid \mathsf{NW}_{i-1}] \\ &\leq \sum_{i \leq t} \frac{2k}{n - 2i} \leq \frac{4kt}{n} \leq t\frac{4\sqrt{2m}}{n}.\end{aligned}$$

Combining this bound with (5) and solving for t gives $\frac{2}{3}\left(\frac{1}{2} - \varepsilon - o(1)\right)\frac{n}{4\sqrt{2m}} < t$. The factor of $2/3$ is because $\mathcal{A}$ returns an edge with probability at least $2/3$. Thus, $t = \Omega(n/\sqrt{m})$, as desired. ◀

Acknowledgments. We are very grateful to Dana Ron and Oded Goldreich for valuable discussions and comments.

References

1 Talya Eden, Amit Levi, Dana Ron, and C. Seshadhri. Approximately counting triangles in sublinear time. *SIAM J. Comput.*, 46(5):1603–1646, 2017. `doi:10.1137/15M1054389`.

2 Talya Eden, Dana Ron, and C. Seshadhri. Sublinear time estimation of degree distribution moments: The degeneracy connection. In Ioannis Chatzigiannakis, Piotr Indyk, Fabian Kuhn, and Anca Muscholl, editors, *44th International Colloquium on Automata, Languages, and Programming, ICALP 2017, July 10-14, 2017, Warsaw, Poland*, volume 80 of *LIPIcs*, pages 7:1–7:13. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017. `doi:10.4230/LIPIcs.ICALP.2017.7`.

3 Talya Eden and Will Rosenbaum. On sampling edges almost uniformly. *CoRR*, abs/1706.09748, 2017. `arXiv:1706.09748`.

4 Uriel Feige. On sums of independent random variables with unbounded variance and estimating the average degree in a graph. *SIAM J. Comput.*, 35(4):964–984, 2006. `doi:10.1137/S0097539704447304`.

5 Oded Goldreich and Dana Ron. Property testing in bounded degree graphs. In Frank Thomson Leighton and Peter W. Shor, editors, *Proceedings of the Twenty-Ninth Annual ACM Symposium on the Theory of Computing, El Paso, Texas, USA, May 4-6, 1997*, pages 406–415. ACM, 1997. `doi:10.1145/258533.258627`.

6 Oded Goldreich and Dana Ron. Approximating average parameters of graphs. *Random Struct. Algorithms*, 32(4):473–493, 2008. `doi:10.1002/rsa.20203`.

7 Tali Kaufman, Michael Krivelevich, and Dana Ron. Tight bounds for testing bipartiteness in general graphs. *SIAM J. Comput.*, 33(6):1441–1483, 2004. `doi:10.1137/S0097539703436424`.

8 Michal Parnas and Dana Ron. Testing the diameter of graphs. *Random Struct. Algorithms*, 20(2):165–183, 2002. `doi:10.1002/rsa.10013`.