Counting Arbitrary Subgraphs in Data Streams*

Daniel M. Kane[1], Kurt Mehlhorn[2], Thomas Sauerwald[2], and He Sun[2,3]

[1] Department of Mathematics, Stanford University, USA
[2] Max Planck Institute for Informatics, Germany
[3] Institute for Modern Mathematics and Physics, Fudan University, China

Abstract. We study the subgraph counting problem in data streams. We provide the first non-trivial estimator for approximately counting the number of occurrences of an *arbitrary* subgraph H of constant size in a (large) graph G. Our estimator works in the turnstile model, i.e., can handle both edge-insertions and edge-deletions, and is applicable in a distributed setting. Prior to this work, only for a few non-regular graphs estimators were known in case of edge-insertions, leaving the problem of counting general subgraphs in the turnstile model wide open. We further demonstrate the applicability of our estimator by analyzing its concentration for several graphs H and the case where G is a power law graph.

Keywords: data streams, subgraph counting, network analysis.

1 Introduction

Counting (small) subgraphs in massive graphs is one of the fundamental tasks in algorithm design and has various applications, including analyzing the connectivity of networks, uncovering the structural information of large graphs, and indexing graph databases. The current best known algorithm for the simplest non-trivial version of the problem, counting the number of triangles, is based on matrix multiplication, and is infeasible for massive graphs. To overcome this, we consider the problem in the data streaming setting, where the edges come sequentially and the algorithm is required to approximate the number of subgraphs without storing the whole graph.

Formally in this problem, we are given a set of items $s_1, s_2, \ldots$ in a data stream. These items arrive sequentially and represent edges of an underlying graph $G = (V, E)$. Two standard models [14] in this context are the *Cash Register Model* and the *Turnstile Model*. In the cash register model, each item s_i represents one edge and these arrived items form a graph G with edge set $E := \bigcup\{s_i\}$, where $E = \emptyset$ initially. The turnstile model generalizes the cash register model and is applicable to dynamic situations. Specifically, each item s_i in the turnstile model is of the form (e_i, sign_i), where e_i is an edge of G and $\mathrm{sign}_i \in \{+, -\}$ indicates that e_i is inserted to or deleted from G. That is, after reading the ith item, $E \leftarrow E \cup \{e_i\}$ if $\mathrm{sign}_i = +$, and $E \leftarrow E \setminus \{e_i\}$ otherwise.

* This material is based upon work supported by the National Science Foundation under Award No. 1103688.

A. Czumaj et al. (Eds.): ICALP 2012, Part II, LNCS 7392, pp. 598–609, 2012.

In a more general distributed setting, there are k distributed sites, each receiving a stream S_i of elements over time, and every S_i is processed by a local host. When the number of subgraphs is asked for, these k hosts cooperate to give an approximation for the underlying graph formed by $\bigcup_{i=1}^{k} S_i$.

Our Results & Techniques. We present the first sketch for counting *arbitrary* subgraphs of constant size in data streams. While most of the previous algorithms are based on sampling techniques and cannot be extended to count subgraphs with complex structures, our algorithm can approximately count arbitrary (possibly directed) subgraphs. Moreover, our algorithm runs in the turnstile model and is applicable in the distributed setting.

More formally, for any fixed subgraph H of constant size, we present an algorithm that $(1 \pm \varepsilon)$-approximates the number of occurrences of H in G, denoted by $\#H$. That is, for any constant $0 < \varepsilon < 1$, with probability at least $2/3$ the output Z of our algorithm satisfies $Z \in [(1-\varepsilon) \cdot \#H, (1+\varepsilon) \cdot \#H]$. For several families of graphs G and H, our algorithm achieves a $(1 \pm \varepsilon)$-approximation for the number of subgraphs H in G within sublinear space. Our result generalizes previous work which can only count cycles in the turnstile model [10, 11], and answers the 11th open problem in the 2006 IITK Workshop on Algorithms for Data Steams [12].

We further consider counting stars in power law graphs, which include many practical networks. We show that $O\left(\frac{1}{\varepsilon^2} \cdot \log n\right)$ bits suffice to get a $(1 \pm \varepsilon)$-approximation for counting stars S_k, while the exact counting needs $n \cdot \log n$ bits of space. Our main results are summarized in Table 1.

Our sketch relies on a novel approach of designing random vectors that are based on different combinations of complex numbers. By using different roots of unity and random mappings from vertices in G to complex numbers, we obtain an unbiased estimator for $\#H$. This partially answers Problem 4 of the survey by Muthukrishnan [14], which asks for suitable applications of complex-valued hash functions in data streaming algorithms. Apart from counting subgraphs in streams, we believe that our new approach will have more applications.

Discussion. To demonstrate that for a large family of graphs G our algorithm achieves a $(1 \pm \varepsilon)$-approximation within sublinear space, we consider Erdös-Rényi random graphs $G = G(n,p)$, where each edge is placed independently with a fixed probability $p \geqslant (1+\varepsilon) \cdot \ln(n)/n$. Random graphs are of interest for the performance of our algorithm, as the independent appearance of the edges in $G = G(n,p)$ reduces the number of particular patterns. In other words, if our algorithm has low space complexity for counting a subgraph H in $G(n,p)$, then the space complexity is even lower for counting a more frequently occurring subgraph in a real-world graph G which has the same density as $G(n,p)$.

Regarding the space complexity of our algorithm on random graphs, assume for instance that the subgraph H is a P_3 or S_3 (i.e., a path or a star with three edges). The expected number of occurrences of such a graph is of order $n^4 p^3 \gg 1$. It can be shown by standard techniques (cf. [1, Section 4.4]) that the number of occurrences is also of this order with probability $1 - o(1)$ as $n \to \infty$. Assuming

Table 1. Space requirement for $(1 \pm \varepsilon)$-approximately counting an undirected and connected graph H with $k = O(1)$ edges. Here δ and Δ denote the minimum and maximum degree, respectively. Space complexity is measured in terms of bits.

Conditions	Space Complexity	Reference
any graph G any graph H	$O\left(\frac{1}{\varepsilon^2} \cdot \frac{m^k \cdot \Delta(G)^k}{(\#H)^2} \cdot \log n\right)$	Theorem 7
any graph G H with $\delta(H) \geqslant 2$	$O\left(\frac{1}{\varepsilon^2} \cdot \frac{m^k}{(\#H)^2} \cdot \log n\right)$	Theorem 7
any graph G stars S_k	$O\left(\frac{n^{1-1/(2k)}}{\varepsilon^2} \cdot \left(\frac{n^{3/2-1/(2k)} \cdot \Delta(G)^{2k}}{(\#S_k)^2} + 1\right) \cdot \log n\right)$	Theorem 8
Power law graph G stars S_k	$O\left(\frac{1}{\varepsilon^2} \cdot \log n\right)$	Theorem 9

that this event occurs, Theorem 7 along with the facts that $m = \Theta(n^2 p)$ and $\Delta(G) = \Theta(np)$ implies a $(1 \pm \varepsilon)$-approximation algorithm for P_3 (or S_3) with space complexity $O(\frac{1}{\epsilon^2} \cdot n \cdot \log n)$. For stars S_k with any constant k, the result from Theorem 8 yields a $(1 \pm \varepsilon)$-approximation algorithm in space $O\left(\frac{1}{\epsilon^2} \cdot \sqrt{n} \cdot \log n\right)$. Finally, for any cycle with $k = O(1)$ edges, Theorem 7 gives an algorithm with space complexity $O(\frac{1}{\epsilon^2} \cdot p^{-k} \cdot \log n)$, which is sublinear for sufficiently large values of p, e.g., $p = \omega(n^{-1/k})$.

Related Work. Bar-Yossef, Kumar and Sivakumar were the first to study the subgraph counting problem in data streams and presented an algorithm for counting triangles [3]. After that, the problem of counting triangles in data streams was studied extensively [4, 6, 10, 16]. The problem of counting other subgraphs was also addressed in the literature. Buriol et al. [7] considered the problem of estimating clustering indexes in data streams. Bordino et al. [5] extended the technique of counting triangles [6] to all subgraphs on three and four vertices. Manjunath et al. [11] presented an algorithm for counting cycles of constant size in data streams. Among these results, only two algorithms [10, 11] work in the turnstile model and these only hold for cycles.

Apart from designing algorithms in the streaming model, the subgraph counting problem has been studied extensively. Alon et al. [2] presented an algorithm for counting given-length cycles. Gonen et al. [9] showed how to count stars and other small subgraphs in sublinear time. In particular, several small subgraphs in a network, named network motifs, have been identified as the simple building blocks of complex biological networks and the distribution of their occurrences could reveal answers to many important biological questions [13, 17].

Notation. Let $G = (V, E)$ be an undirected graph without self-loops and multiple edges. The set of vertices and edges are represented by $V[G]$ and $E[G]$, respectively. We will assume that $V[G] = \{1, \ldots, n\}$ and n is known in advance.

For any vertex $u \in V[G]$, the degree of u is denoted by $\deg(u)$. The maximum and minimum degree of G are denoted by $\Delta(G)$ and $\delta(G)$, respectively.

Given two directed graphs H_1 and H_2, we say that H_1 is *homomorphic* to H_2 if there is a mapping $\varphi : V[H_1] \to V[H_2]$ such that $(u, v) \in E[H_1]$ implies $(\varphi(u), \varphi(v)) \in E[H_2]$. Graphs H_1 and H_2 are said to be *isomorphic* if there is a bijection $\varphi : V[H_1] \to V[H_2]$ such that $(u, v) \in E[H_1]$ iff $(\varphi(u), \varphi(v)) \in E[H_2]$. Let $\mathrm{auto}(H)$ be the number of automorphisms of graph H.

For any graph H, we call a subgraph H_1 of G that is not necessarily induced an *occurrence* of H, if H_1 is isomorphic to H. Let $\#(H, G)$ be the number of occurrences of H in G. When reference to G is clear, we may also write $\#H$. A kth root of unity is any number of the form $\mathrm{e}^{2\pi \mathrm{i} \cdot j/k}$, where $0 \leqslant j < k$. For $p, q \in \mathbb{N}$ define $\left(\mathrm{e}^{2\pi \mathrm{i} \cdot j/k}\right)^{p/q}$ as $\mathrm{e}^{2\pi \mathrm{i} \cdot (jp)/(kq)}$.

2 An Unbiased Estimator for Counting Subgraphs

We present a framework for counting general subgraphs. Suppose that H is a fixed graph with t vertices and k edges, and we want to count the number of occurrences of H in G. For the notation, we denote vertices of H by a, b and c, and vertices of G by u, v and w, respectively. Let the degree of vertex a in H be $\deg_H(a)$. We equip the edges of H with an arbitrary orientation, as this is necessary for the further analysis. Therefore, each edge in H together with its orientation can be expressed as $\overrightarrow{ab}$ for some $a, b \in V[H]$. For simplicity and with slight abuse of notation we will use H to denote such an oriented graph.

At a high level, our estimator maintains k complex-valued variables $Z_{\overrightarrow{ab}}(G)$, where $\overrightarrow{ab} \in E[H]$, and these variables are set to be zero initially. For every arriving edge $\{u, v\} \in E[G]$ we update each $Z_{\overrightarrow{ab}}(G)$ according to

$$Z_{\overrightarrow{ab}}(G) \leftarrow Z_{\overrightarrow{ab}}(G) + \mathcal{M}_{\overrightarrow{ab}}(u, v) + \mathcal{M}_{\overrightarrow{ab}}(v, u) \ ,$$

where $\mathcal{M}_{\overrightarrow{ab}} : V[G] \times V[G] \to \mathbb{C}$ is defined with respect to edge $\overrightarrow{ab} \in E[H]$ and can be computed in constant time. Hence

$$Z_{\overrightarrow{ab}}(G) = \sum_{\{u,v\} \in E[G]} \mathcal{M}_{\overrightarrow{ab}}(u, v) + \mathcal{M}_{\overrightarrow{ab}}(v, u) \ .$$

Intuitively $\mathcal{M}_{\overrightarrow{ab}}(u, v)$ gives $\{u, v\}$ the orientation $\overrightarrow{uv}$ and maps $\overrightarrow{uv}$ to $\overrightarrow{ab}$, and $\mathcal{M}_{\overrightarrow{ab}}(u, v) + \mathcal{M}_{\overrightarrow{ab}}(v, u)$ is used to express two different orientations of edge $\{u, v\}$. For every query for $\#(H, G)$, the estimator simply outputs the real part of $\alpha \cdot \prod_{\overrightarrow{ab} \in E[H]} Z_{\overrightarrow{ab}}(G)$, where $\alpha \in \mathbb{R}^+$ is a scaling factor. For any k edges $(u_1, v_1), \ldots, (u_k, v_k)$ in G and k edges $\overrightarrow{a_1 b_1}, \ldots, \overrightarrow{a_k, b_k}$ in H, we want $\alpha \cdot \prod_{i=1}^{k} \mathcal{M}_{\overrightarrow{a_i b_i}}(u_i, v_i)$ to be one if these edges $(u_1, v_1), \ldots, (u_k, v_k)$ form an occurrence of H, and zero otherwise.

More formally, each $\mathcal{M}_{\overrightarrow{ab}}(u, v)$ is defined according to the degree of vertices a, b in graph H and consists of the product of three types of random variables $Q, X_c(w)$ and $Y(w)$, where $c \in V[H]$ and $w \in V[G]$:

- Variable Q is a random τth root of unity, where $\tau := 2^t - 1$.
- For vertex $c \in V[H]$ and $w \in V[G]$, function $X_c(w)$ is a random $\deg_H(c)$th root of unity, and for each vertex $c \in V[H]$, $X_c : V[G] \to \mathbb{C}$ is chosen independently and uniformly at random from a family of $4k$-wise independent hash functions. Variables Q and $X_c(\cdot)$ for $c \in V[H]$ are chosen independently.
- For every $w \in V[G]$, $Y(w)$ is a random element from $S := \{1, 2, 4, 8, \ldots, 2^{t-1}\}$ as part of a $4k$-wise independent hash function. Variables $X_c(\cdot)$ for $c \in V[H]$, $Y(\cdot)$ and Q are chosen independently.

Given the notations above, we define each function $\mathcal{M}_{\overrightarrow{ab}}$ as

$$\mathcal{M}_{\overrightarrow{ab}}(u, v) := X_a(u)\ X_b(v)\ Q^{\frac{Y(u)}{\deg_H(a)}}\ Q^{\frac{Y(v)}{\deg_H(b)}}\ .$$

Estimator 1 gives the formal description of the update and query procedures.

Estimator 1. Counting $\#(H, G)$

Step 1 (Update): When an edge $e = \{u, v\} \in E[G]$ arrives, update each $Z_{\overrightarrow{ab}}$ w.r.t.

$$Z_{\overrightarrow{ab}}(G) \leftarrow Z_{\overrightarrow{ab}}(G) + \mathcal{M}_{\overrightarrow{ab}}(u, v) + \mathcal{M}_{\overrightarrow{ab}}(v, u). \tag{1}$$

Step 2 (Query): When $\#(H, G)$ is required, output the real part of

$$\frac{t^t}{t! \cdot \mathrm{auto}(H)} \cdot Z_H(G)\ , \tag{2}$$

where Z_H is defined by

$$Z_H(G) := \prod_{\overrightarrow{ab} \in E[H]} Z_{\overrightarrow{ab}}(G)\ . \tag{3}$$

Estimator 1 is applicable in a quite general setting: First, the estimator runs in the turnstile model. For simplicity the update procedure above is only described for the edge-insertion case. For every item of the stream that represents an edge-deletion, we replace "+" by "−" in (1). Second, our estimator also works in the distributed setting, where every local host maintains variables $Z_{\overrightarrow{ab}}$ for $\overrightarrow{ab} \in E[H]$, and does the update for every arriving item in the local stream. When the output is required, these variables located at different hosts are summed up and we return the estimated value according to (3). Third, the estimator above can be revised easily to count the number of directed subgraphs in a directed graphs. Since in this case we need to change the constant of (2) accordingly, in the rest of our paper we only focus on the case of counting undirected graphs.

3 Analysis of the Estimator

Let us first explain the intuition behind our estimator. By definition we have

$$Z_H(G) = \prod_{\overrightarrow{ab} \in E[H]} Z_{\overrightarrow{ab}}(G) = \prod_{\overrightarrow{ab} \in E[H]} \sum_{\{u,v\} \in E[G]} \left(\mathcal{M}_{\overrightarrow{ab}}(u, v) + \mathcal{M}_{\overrightarrow{ab}}(v, u) \right)\ .$$

Since H has k edges, $Z_H(G)$ is a product of k terms and each term is a sum over all edges of G each with two possible orientations. Hence, in the expansion of $Z_H(G)$ any k-tuple $(e_1, \ldots, e_k) \in E^k[G]$ contributes 2^k different terms to $Z_H(G)$ and each term corresponds to a certain orientation of $(e_1, \ldots, e_k)$. Let $\overrightarrow{T} = (\overrightarrow{e_1}, \ldots, \overrightarrow{e_k})$ be an arbitrary orientation of $(e_1, \ldots, e_k)$ and $G_{\overrightarrow{T}}$ be the directed graph induced from $\overrightarrow{T}$.

At a high level, we use three types of variables to test if $G_{\overrightarrow{T}}$ is isomorphic to H. These variables play different roles, as described below. (i) For $c \in V[H]$ and $w \in V[G]$, we have $\mathbb{E}\big[X_c^i(w)\big] \neq 0$ $(1 \leqslant i \leqslant \deg_H(c))$ iff $i = \deg_H(c)$. Random variables $X_c(w)$ guarantee that $G_{\overrightarrow{T}}$ contributes to $\mathbb{E}[Z_H(G)]$ only if $G_{\overrightarrow{T}}$ is homomorphic to H. (ii) Through function $Y : V[G] \to S$ every vertex $u \in V_{\overrightarrow{T}}$ maps to one element $Y(u)$ in S randomly. If $|V_{\overrightarrow{T}}| = |S| = t$, then with constant probability, vertices in $V_{\overrightarrow{T}}$ map to different t numbers in S. Otherwise, $|V_{\overrightarrow{T}}| < t$ and vertices in $V_{\overrightarrow{T}}$ cannot map to different t elements. Since Q is a random τth root of unity, $\mathbb{E}\big[Q^i\big] \neq 0$ $(1 \leqslant i \leqslant \tau)$ iff $i = \tau$, where $\tau = \sum_{\ell \in S} \ell$. The combination of Q and Y guarantees that $G_{\overrightarrow{T}}$ contributes to $\mathbb{E}[Z_H(G)]$ only if graph H and $G_{\overrightarrow{T}}$ have the same number of vertices. Combining (i) and (ii), only subgraphs isomorphic to H contribute to $\mathbb{E}[Z_H(G)]$.

Lemma 1 ([8]). *For any $c \in V[H]$ let X_c be a randomly chosen $\deg_H(c)$th root of unity. Then for any $1 \leqslant i \leqslant \deg_H(c)$, it holds that*

$$\mathbb{E}[X_c^i] = \begin{cases} 1, & i = \deg_H(c) \ , \\ 0, & 1 \leqslant i < \deg_H(c) \ . \end{cases}$$

In particular, $\mathbb{E}[X_c] = 1$ if $\deg_H(c) = 1$.

Lemma 2. *Let R be a primitive τth root of unity and $k \in \mathbb{N}$. Then*

$$\sum_{\ell=0}^{\tau-1} (R^k)^\ell = \begin{cases} \tau, & \tau \mid k \ , \\ 0, & \tau \nmid k \ . \end{cases}$$

Lemma 3. *Let $x_i \in \mathbb{Z}_{\geqslant 0}$ and $\sum_{i=0}^{t-1} x_i = t$. Then $2^t - 1 \mid \sum_{i=0}^{t-1} 2^i \cdot x_i$ if and only if $x_0 = \cdots = x_{t-1} = 1$.*

Based on the three lemmas above, we prove that $Z_H(G)$ is an unbiased estimator for $\#(H, G)$.

Theorem 4. *Let H be a graph with t vertices and k edges. Assume that variables $X_c(w), Y(w)$ for $c \in V[H], w \in V[G]$ and Q are as defined above. Then*

$$\mathbb{E}[Z_H(G)] = \frac{t! \cdot \mathrm{auto}(H)}{t^t} \cdot \#(H, G) \ .$$

Proof. Let $(e_1, \ldots, e_k) \in E^k(G)$ and $\overrightarrow{T} = (\overrightarrow{e_1}, \ldots, \overrightarrow{e_k})$ be an arbitrary orientation of $(e_1, \ldots, e_k)$, where $\overrightarrow{e_i} = \overrightarrow{u_i v_i}$. Consider the expansion of $Z_H(G)$ below:

$$Z_H(G) = \prod_{\overrightarrow{ab} \in E[H]} Z_{\overrightarrow{ab}}(G) = \prod_{\overrightarrow{ab} \in E[H]} \sum_{\{u,v\} \in E[G]} \left(\mathcal{M}_{\overrightarrow{ab}}(u, v) + \mathcal{M}_{\overrightarrow{ab}}(v, u) \right) \ .$$

The term corresponding to $(\overrightarrow{e_1}, \ldots, \overrightarrow{e_k})$ in the expansion of $Z_H(G)$ is

$$\prod_{i=1}^{k} \mathcal{M}_{\overrightarrow{a_i b_i}}(u_i, v_i) = \prod_{i=1}^{k} X_{a_i}(u_i)\ X_{b_i}(v_i)\ Q^{\frac{Y(u_i)}{\deg_H(a_i)}}\ Q^{\frac{Y(v_i)}{\deg_H(b_i)}}\ , \tag{4}$$

where $\overrightarrow{a_i b_i}$ is the ith edge of H (where we assume any order) and $\overrightarrow{u_i v_i}$ is the ith edge in $\overrightarrow{T}$. We show that the expectation of (4) is non-zero if and only if the graph induced by $\overrightarrow{T}$ is an occurrence of H in G. Moreover, if the expectation of (4) is non-zero, then its value is a constant.

For any vertex w of G and any vertex c of H, let

$$\theta_{\overrightarrow{T}}(c, w) := \left|\left\{i : (u_i = w \text{ and } a_i = c) \text{ or } (v_i = w \text{ and } b_i = c)\right\}\right|$$

be the number of edges in $\overrightarrow{T}$ with head (or tail) w mapping to the edges in H with head (or tail) c. Since every vertex c of H is incident to $\deg_H(c)$ edges, for any $c \in V[H]$ it holds that $\sum_{w \in V_{\overrightarrow{T}}} \theta_{\overrightarrow{T}}(c, w) = \deg_H(c)$. By the definition of $\theta_{\overrightarrow{T}}$, we can rewrite (4) as

$$\left(\prod_{c \in V[H]} \prod_{w \in V_{\overrightarrow{T}}} X_c^{\theta_{\overrightarrow{T}}(c,w)}(w)\right) \cdot \left(\prod_{c \in V[H]} \prod_{w \in V_{\overrightarrow{T}}} Q^{\frac{\theta_{\overrightarrow{T}}(c,w) Y(w)}{\deg_H(c)}}\right).$$

Therefore $Z_H(G)$ is equal to

$$\sum_{\substack{e_1,\ldots,e_k \\ e_i \in E[G]}} \sum_{\overrightarrow{T} = (\overrightarrow{e_1}, \ldots, \overrightarrow{e_k})} \left(\prod_{c \in V[H]} \prod_{w \in V_{\overrightarrow{T}}} X_c^{\theta_{\overrightarrow{T}}(c,w)}(w)\right) \cdot \left(\prod_{c \in V[H]} \prod_{w \in V_{\overrightarrow{T}}} Q^{\frac{\theta_{\overrightarrow{T}}(c,w) Y(w)}{\deg_H(c)}}\right),$$

where the first summation is over all k-tuples of edges in $E[G]$ and the second summation is over all their possible orientations. By linearity of expectations of these random variables and the assumption that $X_c(\cdot)$ for $c \in V[H]$, $Y(\cdot)$, and Q have sufficient independence, we have

$$\mathbb{E}[Z_H(G)]$$

$$= \sum_{\substack{e_1,\ldots,e_k \\ e_i \in E[G]}} \sum_{\overrightarrow{T} = (\overrightarrow{e_1}, \ldots, \overrightarrow{e_k})} \left(\prod_{c \in V[H]} \mathbb{E}\left[\prod_{w \in V_{\overrightarrow{T}}} X_c^{\theta_{\overrightarrow{T}}(c,w)}(w)\right]\right) \cdot \mathbb{E}\left[\prod_{\substack{c \in V[H] \\ w \in V_{\overrightarrow{T}}}} Q^{\frac{\theta_{\overrightarrow{T}}(c,w) Y(w)}{\deg_H(c)}}\right].$$

Let

$$\alpha_{\overrightarrow{T}} := \underbrace{\left(\prod_{c \in V[H]} \mathbb{E}\left[\prod_{w \in V_{\overrightarrow{T}}} X_c^{\theta_{\overrightarrow{T}}(c,w)}(w)\right]\right)}_{A} \cdot \underbrace{\mathbb{E}\left[\prod_{c \in V[H]} \prod_{w \in V_{\overrightarrow{T}}} Q^{\frac{\theta_{\overrightarrow{T}}(c,w) Y(w)}{\deg_H(c)}}\right]}_{B}.$$

We will next show that $\alpha_{\overrightarrow{T}}$ is either zero or a nonzero constant independent of $\overrightarrow{T}$. The latter is the case if and only if G_T, the undirected graph induced from the edge set $\overrightarrow{T}$, is an occurrence of H in G.

We consider the product A at first. Assume that $A \neq 0$. Using the same technique as [11], we construct a homomorphism from H to $G_{\overrightarrow{T}}$. Remember that: (i) For any $c \in V[H]$ and $w \in V_{\overrightarrow{T}}$, we have $\theta_{\overrightarrow{T}}(c,w) \leqslant \deg_H(c)$, and (ii) $\mathbb{E}\left[X_c^i(w)\right] \neq 0$ iff $i \in \{0, \deg_H(c)\}$. Therefore for any fixed $\overrightarrow{T}$ and $c \in V[H]$, it holds that $\mathbb{E}\left[\prod_{w \in V_{\overrightarrow{T}}} X_c^{\theta_{\overrightarrow{T}}(c,w)}(w)\right] \neq 0$ iff $\theta_{\overrightarrow{T}}(c,w) \in \{0, \deg_H(c)\}$ for all w. Now assume that $\mathbb{E}\left[\prod_{w \in V_{\overrightarrow{T}}} X_c^{\theta_{\overrightarrow{T}}(c,w)}(w)\right] \neq 0$ for every $c \in V[H]$. Then $\theta_{\overrightarrow{T}}(c,w) \in \{0, \deg_H(c)\}$ for all $c \in V[H]$ and $w \in V[G]$. Since $\sum_w \theta_{\overrightarrow{T}}(c,w) = \deg_H(c)$ for any $c \in V[H]$, there is a unique vertex $w \in V_{\overrightarrow{T}}$ such that $\theta_{\overrightarrow{T}}(c,w) = \deg_H(c)$. Define $\varphi : V[H] \to V_{\overrightarrow{T}}$ as $\varphi(c) = w$ for the vertex w satisfying $\theta_{\overrightarrow{T}}(c,w) = \deg_H(c)$. Then φ is a homomorphism, i.e. $(a,b) \in E[H]$ implies $(\varphi(a), \varphi(b)) \in E[G_{\overrightarrow{T}}]$. Hence $A \neq 0$ implies H is homomorphic to $G_{\overrightarrow{T}}$, and

$$\prod_{c \in V[H]} \mathbb{E}\left[\prod_{w \in V_{\overrightarrow{T}}} X_c^{\theta_{\overrightarrow{T}}(c,w)}(w)\right] = \prod_{c \in V[H]} \mathbb{E}\left[X_c^{\deg_H(c)}(\varphi(c))\right] = 1 \ . \tag{5}$$

Second we consider the product B. Our task is to show that, under the condition $A \neq 0$, $G_{\overrightarrow{T}}$ is an occurrence of H if and only if $B \neq 0$. Observe that

$$\mathbb{E}\left[\prod_{c \in V[H]} \prod_{w \in V_{\overrightarrow{T}}} Q^{\frac{\theta_{\overrightarrow{T}}(c,w) Y(w)}{\deg_H(c)}}\right] = \mathbb{E}\left[Q^{\sum_{c \in V[H]} \sum_{w \in V_{\overrightarrow{T}}} \frac{\theta_{\overrightarrow{T}}(c,w) Y(w)}{\deg_H(c)}}\right] \ .$$

Case 1: Assume that $G_{\overrightarrow{T}}$ is an occurrence of H in G. Then $|V_{\overrightarrow{T}}| = |V[H]|$ and the function φ constructed above is a bijection, which implies that

$$\sum_{c \in V[H]} \sum_{w \in V_{\overrightarrow{T}}} \frac{\theta_{\overrightarrow{T}}(c,w) Y(w)}{\deg_H(c)} = \sum_{c \in V[H]} Y(\varphi(c)) = \sum_{w \in V_{\overrightarrow{T}}} Y(w) \ .$$

Without loss of generality, let $V_{\overrightarrow{T}} = \{w_1, \ldots, w_t\}$. By considering all possible choices for $Y(w_1), \ldots, Y(w_t)$, denoted by $y(w_1), \ldots, y(w_t) \in S$, and independence between Q and $Y(w)$, where $w \in V[G]$, we have

$$\begin{aligned} B =& \sum_{j=0}^{\tau-1} \sum_{y(w_1),\ldots,y(w_t) \in S} \frac{1}{\tau} \left(\prod_{i=1}^{t} \mathbf{Pr}\left[\, Y(w_i) = y(w_i) \,\right]\right) \cdot \exp\left(\frac{2\pi \mathrm{i} j}{\tau} \sum_{\ell=1}^{t} y(w_\ell)\right) \\ =& \sum_{j=0}^{\tau-1} \sum_{\substack{y(w_1),\ldots,y(w_t) \in S \\ \vartheta := y(w_1)+\cdots+y(w_t),\, \tau \mid \vartheta}} \frac{1}{\tau} \left(\frac{1}{t}\right)^t \exp\left(\frac{2\pi \mathrm{i}}{\tau} \cdot \vartheta \cdot j\right) + \\ & \sum_{j=0}^{\tau-1} \sum_{\substack{y(w_1),\ldots,y(w_t) \in S \\ \vartheta := y(w_1)+\cdots+y(w_t),\, \tau \nmid \vartheta}} \frac{1}{\tau} \left(\frac{1}{t}\right)^t \exp\left(\frac{2\pi \mathrm{i}}{\tau} \cdot \vartheta \cdot j\right) \ . \end{aligned}$$

Applying Lemma 2 with $R = \exp\left(\frac{2\pi \mathrm{i}}{\tau}\right)$, the second summation is zero. Hence by Lemma 3 we have

$$B = \sum_{\substack{y(w_1),\ldots,y(w_t)\in S \\ \tau \mid y(w_1)+\cdots+y(w_t)}} \left(\frac{1}{t}\right)^t = \sum_{\substack{y(w_1),\ldots,y(w_t)\in S \\ y(w_1)+\cdots+y(w_t)=\tau}} \left(\frac{1}{t}\right)^t = \left(\frac{1}{t}\right)^t \cdot t! = \frac{t!}{t^t} \ .$$

Case 2: Assume that $G_{\overrightarrow{T}}$ is not an occurrence of H in G and let $V_{\overrightarrow{T}} = \{w_1, \ldots, w_{t'}\}$, where $t' < t$. Then there is a vertex $w \in V_{\overrightarrow{T}}$ and different $b, c \in V[H]$, such that $\varphi(b) = \varphi(c) = w$. As before we have

$$\sum_{c\in V[H]} \sum_{w\in V_{\overrightarrow{T}}} \frac{\theta_{\overrightarrow{T}}(c,w) Y(w)}{\deg_H(c)} = \sum_{c\in V[H]} Y(\varphi(c)) \ .$$

By Lemma 3, $\tau \nmid \sum_{c\in V[H]} Y(\varphi(c))$. Hence

$$B = \sum_{j=0}^{\tau-1} \sum_{\substack{y(w_1),\ldots,y(w_{t'})\in S \\ \vartheta := \sum_{c\in V[H]} y(\varphi(c))}} \frac{1}{\tau} \left(\frac{1}{t}\right)^{t'} \exp\left(\frac{2\pi \mathrm{i}}{\tau} \cdot \vartheta \cdot j\right) = 0 \ ,$$

where the last equality follows from Lemma 2 with $R = \exp\left(\frac{2\pi \mathrm{i}}{\tau}\right)$.

Let $\mathbf{1}_{G_{\overrightarrow{T}} \equiv H}$ be the indicator variable that is one if $G_{\overrightarrow{T}}$ and H are isomorphic and zero otherwise. By the definition of graph automorphism and (5),

$$\mathbb{E}[Z_H(G)] = \sum_{\substack{e_1,\ldots,e_k \\ e_i \in E[G]}} \sum_{\overrightarrow{T} = (\overrightarrow{e_1},\ldots,\overrightarrow{e_k})} \frac{t!}{t^t} \cdot \left(\mathbf{1}_{G_{\overrightarrow{T}} \equiv H}\right) = \frac{t! \cdot \mathrm{auto}(H)}{t^t} \cdot \#(H, G) \ . \qquad \square$$

We can use a similar technique to analyze the variance of $Z_H(G)$ and apply Chebyshev's inequality on complex-valued random variables to upper bound the number of trials required for a $(1 \pm \varepsilon)$-approximation. Since $Z_H(G)$ is complex-valued, we need to upper bound $Z_H(G) \cdot \overline{Z_H(G)}$, which relies on the number of subgraphs of $2k$ edges in G with certain properties.

Lemma 5. *Let G be a graph with m edges and H be any graph with k edges (possibly with multiple edges), where k is a constant. The following statements hold: (i) If $\delta(H) \geqslant 2$, then $\#(H, G) = O\left(m^{k/2}\right)$; (ii) If every connected component of H contains at least two edges, then $\#(H, G) = O\big(m^{k/2} \cdot (\Delta(G))^{k/2}\big)$.*

Lemma 6. *Let G be any graph with m edges, H be any graph with k edges for a constant k. Random variables $X_c(w)$ $(c \in V[H], w \in V[G])$ and Q are defined as above. Then the following statements hold:*

1. *If $\delta(H) \geqslant 2$, then $\mathbb{E}\left[Z_H(G) \cdot \overline{Z_H(G)}\right] = O\left(m^k\right)$.*
2. *Let H be a connected graph with $k \geqslant 2$ edges and $\mathcal{H}$ be the set of all subgraphs H' in G with the following properties: (i) H' has $2k$ edges, and (ii) every connected component of H' contains at least two edges. Then $\mathbb{E}\left[Z_H(G) \cdot \overline{Z_H(G)}\right] = O\left(|\mathcal{H}|\right)$.*

By using Chebyshev's inequality, we can get a $(1 \pm \varepsilon)$-approximation by running independent copies of our estimator in parallel and returning the average of the output of these copies. This leads to our main result for counting the number of occurrences of H.

Theorem 7. *Let G be any graph with m edges and H be any graph with $k = O(1)$ edges. For any constant $0 < \varepsilon < 1$, there is an algorithm to $(1 \pm \varepsilon)$-approximate $\#(H, G)$ using (i) $O\big(\frac{1}{\varepsilon^2} \cdot \frac{m^k}{(\#H)^2} \cdot \log n\big)$ bits if $\delta(H) \geqslant 2$, or (ii) using $O\big(\frac{1}{\varepsilon^2} \cdot \frac{m^k \cdot (\Delta(G))^k}{(\#H)^2} \cdot \log n\big)$ bits for any H.*

Discussion. Statement (i) of Theorem 7 extends the main result of [11, Theorem 1] which requires H to be a cycle. Note that a naïve sampling-based approach would choose a random k-tuple of edges and require $m^k/(\#H)$ space. Theorem 7 improves upon this approach, in particular if the graph G is sparse and the number of occurrences of H is a growing function in n.

4 Extensions

We have developed a general framework for counting arbitrary subgraphs of constant size. For several typical applications we can further improve the space complexity by grouping the sketches or using certain properties of the underlying graph G. For the ease of the discussion we only focus on counting stars.

Grouping Sketches. The space complexity in Theorem 7 relies on the number of edges that the sketch reads. To reduce the variance, a natural way is to use multiple copies of the sketches, and every sketch is only responsible for the updates of the edges from a certain subgraph.

To formulate this intuition, we partition $V = \{1, \ldots, n\}$ into $g := n^{1-1/(2k)}$ subsets $\mathcal{V}_1, \ldots, \mathcal{V}_g$, and $\mathcal{V}_i := \big\{j \colon (i-1) \cdot n^{1/(2k)} + 1 \leqslant j \leqslant i \cdot n^{1/(2k)}\big\}$. Without loss of generality we assume that $n^{1/(2k)} \in \mathbb{N}$. Associated with every $\mathcal{V}_i$, we maintain a sketch $\mathcal{C}_i$, whose description is shown in Estimator 2. For every arriving edge $e = \{u, v\}$ in the stream, we update sketch $\mathcal{C}_i$ if $u \in \mathcal{V}_i$ or $v \in \mathcal{V}_i$. Since (i) the central vertex of every occurrence of S_k is in exactly one subset $\mathcal{V}_i$, and (ii) every edge adjacent to one vertex in $\mathcal{V}_i$ is taken into account by sketch $\mathcal{C}_i$, every occurrence of S_k in G is only counted by one sketch $\mathcal{C}_i$.

Estimator 2. Counting $\#(S_k, G|_{\mathcal{V}_i})$, update procedure

Step 1 (Update): When an edge $e = \{u, v\} \in E[G]$ arrives, update each variable $Z_{\overrightarrow{ab}}$:
(a) If $u \in \mathcal{V}_i$ and $v \in \mathcal{V}_i$, then

$$Z_{\overrightarrow{ab}}(G) \leftarrow Z_{\overrightarrow{ab}}(G) + \mathcal{M}_{\overrightarrow{ab}}(u, v) + \mathcal{M}_{\overrightarrow{ab}}(v, u).$$

(b) If $u \in \mathcal{V}_i$ and $v \in \partial\mathcal{V}_i$, then $Z_{\overrightarrow{ab}}(G) \leftarrow Z_{\overrightarrow{ab}}(G) + \mathcal{M}_{\overrightarrow{ab}}(u, v)$.
(c) If $u \in \partial\mathcal{V}_i$ and $v \in \mathcal{V}_i$, then $Z_{\overrightarrow{ab}}(G) \leftarrow Z_{\overrightarrow{ab}}(G) + \mathcal{M}_{\overrightarrow{ab}}(v, u)$.

More formally, let $\widetilde{\#}(S_k, G|_{\mathcal{V}_i})$ be the number of S_k whose central vertex is in $\mathcal{V}_i$. It holds that $\#(S_k, G) = \sum_{i=1}^{g} \widetilde{\#}(S_k, G|_{\mathcal{V}_i})$. This indicates that if every $\mathcal{C}_i$ is unbiased for $\widetilde{\#}(S_k, G|_{\mathcal{V}_i})$, then we can use the sum of returned values from different $\mathcal{C}_i$'s to approximate $\#(S_k, G)$.

Theorem 8. *Let G be a graph with n vertices. For any constants $0 < \varepsilon < 1$ and k, there is an algorithm to $(1 \pm \varepsilon)$-approximate $\#(S_k, G)$ with space complexity*

$$O\left(\frac{n^{1-1/(2k)}}{\varepsilon^2} \cdot \left(\frac{n^{3/2-1/(2k)} \cdot \Delta(G)^{2k}}{(\#S_k)^2} + 1\right) \cdot \log n\right).$$

Let us consider graphs G with $\Delta(G)/\delta(G) = o(n^{1/(4k)})$ and $\delta(G) \geqslant k$. Since $\#(S_k, G) = \Omega\left(n \cdot \delta(G)^k\right)$, Theorem 8 implies that $o\left(\frac{1}{\varepsilon^2} \cdot n \cdot \log n\right)$ bits suffice to give a $(1 \pm \varepsilon)$-approximation.

Counting on Power Law Graphs. Besides organizing the sketches into groups, the space complexity can be also reduced by using the structural information of the underlying graph G. One important property shared by many biological, social or technological networks is the so-called *Power Law* degree distribution, i.e., the number of vertices with degree d, denoted by $f(d) := |\{v \in V : \deg(v) = d\}|$, satisfies $f(d) \sim d^{-\beta}$, where $\beta > 0$ is the power law exponent. For many networks, experimental studies indicate that β is between 2 and 3, see [15].

Formally, we use the following model based on the cumulative degree distribution. For given constants $\sigma \geqslant 1$ and $d_{\min} \in \mathbb{N}$, we say that G has an approximate power law degree distribution with exponent $\beta \in (2, 3)$, if $\sum_{d=k}^{n-1} f(d) \in \left[\lfloor \sigma^{-1} \cdot n \cdot k^{-\beta+1} \rfloor, \sigma \cdot n \cdot k^{-\beta+1}\right]$ for any $k \geqslant d_{\min}$. Our result on counting stars on power law graphs is as follows.

Theorem 9. *Assume that G has an approximate power law degree distribution with exponent $\beta \in (2, 3)$. Then, for any two constants $0 < \varepsilon < 1$ and k, we can $(1 \pm \varepsilon)$-approximate $\#(S_k, G)$ using $O\left(\frac{1}{\varepsilon^2} \cdot \log n\right)$ bits.*

References

[1] Alon, N., Spencer, J.: The Probabilistic Method, 3rd edn. Wiley-Interscience Series in Discrete Mathematics and Optimization. John Wiley & Sons (2008)
[2] Alon, N., Yuster, R., Zwick, U.: Finding and counting given length cycles. Algorithmica 17(3), 209–223 (1997)
[3] Bar-Yossef, Z., Kumar, R., Sivakumar, D.: Reductions in streaming algorithms, with an application to counting triangles in graphs. In: Proc. 13th Symp. on Discrete Algorithms (SODA), pp. 623–632 (2002)
[4] Becchetti, L., Boldi, P., Castillo, C., Gionis, A.: Efficient semi-streaming algorithms for local triangle counting in massive graphs. In: Proc. 14th Intl. Conf. Knowledge Discovery and Data Mining (KDD), pp. 16–24 (2008)
[5] Bordino, I., Donato, D., Gionis, A., Leonardi, S.: Mining large networks with subgraph counting. In: Proc. 8th Intl. Conf. on Data Mining (ICDM), pp. 737–742 (2008)

[6] Buriol, L.S., Frahling, G., Leonardi, S., Marchetti-Spaccamela, A., Sohler, C.: Counting triangles in data streams. In: Proc. 25th Symp. Principles of Database Systems (PODS), pp. 253–262 (2006)
[7] Buriol, L.S., Frahling, G., Leonardi, S., Sohler, C.: Estimating Clustering Indexes in Data Streams. In: Arge, L., Hoffmann, M., Welzl, E. (eds.) ESA 2007. LNCS, vol. 4698, pp. 618–632. Springer, Heidelberg (2007)
[8] Ganguly, S.: Estimating Frequency Moments of Data Streams Using Random Linear Combinations. In: Jansen, K., Khanna, S., Rolim, J.D.P., Ron, D. (eds.) RANDOM 2004 and APPROX 2004. LNCS, vol. 3122, pp. 369–380. Springer, Heidelberg (2004)
[9] Gonen, M., Ron, D., Shavitt, Y.: Counting stars and other small subgraphs in sublinear-time. SIAM J. Disc. Math. 25(3), 1365–1411 (2011)
[10] Jowhari, H., Ghodsi, M.: New Streaming Algorithms for Counting Triangles in Graphs. In: Wang, L. (ed.) COCOON 2005. LNCS, vol. 3595, pp. 710–716. Springer, Heidelberg (2005)
[11] Manjunath, M., Mehlhorn, K., Panagiotou, K., Sun, H.: Approximate Counting of Cycles in Streams. In: Demetrescu, C., Halldórsson, M.M. (eds.) ESA 2011. LNCS, vol. 6942, pp. 677–688. Springer, Heidelberg (2011)
[12] McGregor, A.: Open Problems in Data Streams and Related Topics, IITK Workshop on Algorithms For Data Sreams (2006), `http://www.cse.iitk.ac.in/users/sganguly/data-stream-probs.pdf`
[13] Milo, R., Shen-Orr, S., Itzkovitz, S., Kashtan, N., Chklovskii, D., Alon, U.: Network motifs: Simple building blocks of complex networks. Science 298(5594), 824–827 (2002)
[14] Muthukrishnan, S.: Data Streams: Algorithms and Applications. Foundations and Trends in Theoretical Computer Science 1(2) (2005)
[15] Newman, M.E.J.: The structure and function of complex networks. SIAM Review 45, 167–256 (2003)
[16] Pagh, R., Tsourakakis, C.E.: Colorful triangle counting and a mapreduce implementation. Inf. Process. Lett. 112(7), 277–281 (2012)
[17] Wong, E., Baur, B., Quader, S., Huang, C.: Biological network motif detection: principles and practice. Briefings in Bioinformatics, 1–14 (June 2011)