Theoretical Computer Science 552 (2014) 44–51

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

# A second look at counting triangles in graph streams

Graham Cormode*, Hossein Jowhari

ARTICLE INFO

*Article history:*
Received 4 January 2014
Received in revised form 21 July 2014
Accepted 31 July 2014
Available online 19 August 2014
Communicated by G. Ausiello

*Keywords:*
Data streams
Randomized algorithms
Graphs

ABSTRACT

In this paper we present improved results on the problem of counting triangles in edge streamed graphs. For graphs with $m$ edges and at least $T$ triangles, we show that an extra look over the stream yields a two-pass streaming algorithm that uses $O(\frac{m}{\epsilon^{4.5}\sqrt{T}})$ space and outputs a $(1+\epsilon)$ approximation of the number of triangles in the graph. This improves upon the two-pass streaming tester of Braverman et al. [2], which distinguishes between triangle-free graphs and graphs with at least $T$ triangles using $O(\frac{m}{T^{1/3}})$ space. Also, in terms of dependence on $T$, we show that more passes would not lead to a better space bound. In other words, we prove there is no constant pass streaming algorithm that distinguishes between triangle-free graphs from graphs with at least $T$ triangles using $O(\frac{m}{T^{1/2+\rho}})$ space for any constant $\rho \geq 0$.

## 1. Introduction

Many applications produce output in form of graphs, presented an edge at a time. These include social networks that produce edges corresponding to new friendships or other connections between entities in the network; communication networks, where each edge represents a communication (phone call, email, text message) between a pair of participants; and web graphs, where each edge represents a link between pages. Over such graphs, we wish to answer questions about the induced graph, relating to the structure and properties.

One of the most basic structures that can be present in a graph is a triangle: an embedded clique on three nodes. Questions around counting the number of triangles in a graph have been widely studied, due to the inherent interest in the problem, and because it is a necessary stepping stone to answering questions about more complex structures in graphs. Triangles are of interest within social networks, as they indicate common friendships: two friends of an individual are themselves friends. Counting the number of friendships within a graph is therefore a measure of the closeness of friendship activities. Another use of the number of triangles is as a parameter for evaluation of large graph models [10].

For these reasons, and for the fundamental nature of the problem, there have been numerous studies of the problem of counting or enumerating triangles in various models of data access: external memory [11,6], map-reduce [15,13,17], and RAM model [16,18]. Indeed, it seems that triangle counting and enumeration is becoming a *de facto* benchmark for testing “big data” systems and their ability to process complex queries. The reason is that the problem captures an essentially hard problem within big data: accurately measuring the degree of correlation. In this paper, we study the problem of triangle counting over (massive) streams of edges. In this case, lower bounds from communication complexity can be applied to show that exactly counting the number of triangles essentially requires storing the full input, so instead we look for methods which can approximate the number of triangles. In this direction, there has been series of works that have attempted to capture the right space complexity for algorithms that approximate the number of triangles. However, most of these works

* Corresponding author.
*E-mail addresses:* G.Cormode@warwick.ac.uk (G. Cormode), hjowhari@sfu.ca (H. Jowhari).

http://dx.doi.org/10.1016/j.tcs.2014.07.025
0304-3975/© 2014 Elsevier B.V. All rights reserved.

have focused on one-pass algorithms and thus, due to the hard nature of the problem, their space bounds have become complicated, suffering from dependencies on multiple graph parameters such as maximum degree, number of paths of length 2, number of cycles of length 4, etc.

In a recent work by Braverman et al. [2], it has been shown that at the expense of an extra pass over stream, a straightforward sampling strategy gives a sublinear bound that depends only on $m$ (number of edges) and $T$ (a lower bound on the number of triangles[1]). More precisely, Braverman et al. [2] have shown that one extra pass yields an algorithm that distinguishes between triangle-free graphs from graphs with at least $T$ triangles using $O(\frac{m}{T^{1/3}})$ words of space. Although their algorithm does not give an estimate of the number of triangles and more important is not clearly superior to the $O(\frac{m\Delta}{T})$ one-pass algorithm by [13,14] (especially for graphs with small maximum degree $\Delta$), it creates some hope that perhaps with the expense of extra passes one could get improved and cleaner space complexities that beat the one-pass bound for a wider range of graphs. In particular one might ask is there an $O(\frac{m}{T})$ space multi-pass algorithm? In this paper, while we refute such a possibility, we show that a more modest bound is possible. Specifically here we show modifications to the sampling strategy of [2] along with a different analysis results in a 2-pass $(1+\epsilon)$ approximation algorithm that uses only $O(\frac{m}{\epsilon^{4.5}\sqrt{T}})$ space. We also observe that this bound is attainable in one-pass – if we make the strong assumption that the order of edge arrivals is random. Additionally, via a reduction to a hard communication complexity problem, we demonstrate that this bound is optimal in terms of its dependence on $T$. In other words there is no constant pass algorithm that distinguishes between triangle-free graphs from graphs with at least $T$ triangles using $O(\frac{m}{T^{1/2+\rho}})$ for any constant $\rho > 0$. We also give a similar two-pass algorithm that has better dependence on $\epsilon$ but sacrifices the optimal dependence on $T$. Our results are summarized in Fig. 2 in terms of the problem addressed, bound provided, and number of passes.

In line with prior work, we assume a simple graph – that is, each edge of the graph is presented exactly once in the stream. Note that our lower bounds immediately hold for the case when edges are repeated.

### 1.1. *Algorithms for triangle counting in graph streams*

The triangle counting problem has attracted particular attention in the model of graph streams: there is now a substantial body of study in this setting. Algorithms are evaluated on the amount of space that they require, the number of passes over the input stream that they take, and the time taken to process each update. Different variations arise depending on whether deletions of edges are permitted, or the stream is 'insert-only'; and whether arrivals are ordered in a particular way, so that all edges incident on one node arrive together, or the edges are randomly ordered or adversarially ordered.

The work of Jowhari and Ghodsi [7] first studied the most popular of these combinations: insert-only, adversarial ordering. The general approach, common to many streaming algorithms, is to build a randomized estimator for the desired quantity, and then repeat this sufficiently many times to provide a guaranteed accuracy. Their approach begins by sampling an edge uniformly from the stream of $m$ arriving edges on $n$ vertices. Their estimator then counts the number of triangles incident on a sampled edge. Since the ordering is adversarial, the estimator has to keep track of all edges incident on the sampled edge, which in the worst case is bounded by $\Delta$, the maximum degree. The sampling process is repeated $O(\frac{1}{\epsilon^2}\frac{m\Delta}{T})$ times (using the assumed bound on the number of triangles, $T$), leading to a total space requirement proportional to $O(\frac{1}{\epsilon^2}\frac{m\Delta^2}{T})$ to give an $\epsilon$ relative error estimation of $t$, the number of triangles in the graph. The parameter $\varepsilon$ ensures that the error in the count is at most $\varepsilon t$ (with constant probability, since the algorithm is randomized). The process can be completed with a single pass over the input. Jowhari and Ghodsi also consider the case where edges may be deleted, in which case a randomized estimator using "sketch" techniques is introduced, improving over a previous sketch algorithm due to Bar-Yossef et al. [4] in some cases.

The work of Buriol et al. [1] also adopted a sampling approach, and built a one-pass estimator with smaller working space. An algorithm is proposed which samples uniformly an edge from the stream, then picks a third node, and scans the remainder of the stream to see if the triangle on these three nodes is present. Recall that $n$ is the number of nodes in the graph, $m$ is number of edges, and $T \leq t$ is lower bound on the (true) number of triangles. To obtain an accurate estimate of number of triangles in the graph, this procedure is repeated independently $O(\frac{mn}{\varepsilon^2 T})$ times to achieve $\epsilon$ relative error.

Recent work by Pavan et al. [14] extends the sampling approach of Buriol et al.: instead of picking a random node to complete the triangle with a sampled edge, their estimator samples a second edge that is incident on the first sampled edge. This estimator is repeated $O(\frac{m\Delta}{\epsilon^2 T})$ times, where $\Delta$ represents the maximum degree of any node. That is, this improves the bound of Buriol et al. by a factor of $n/\Delta$. In the worst case, $\Delta = n$, but in general we expect $\Delta$ to be substantially smaller than $n$.

Braverman et al. [2] take a different approach to sampling. Instead of building a single estimator and repeating, their algorithms sample a set of edges, and then look for triangles induced by the sampled edges. Specifically, an algorithm which takes two passes over the input stream distinguishes triangle-free graphs from those with $T$ triangles in space $O(m/T^{1/3})$.

For graphs with $W \geq m$ where $W$ is the number of wedges (paths of length 2), Jha et al. [8] have shown a single pass $O(\frac{1}{\epsilon^2}m/\sqrt{T})$ space algorithm that returns an additive error estimation of the number of triangles.

[1] In this and prior works, some assumption on the number of triangles is required. This is due in part to the fact that distinguishing triangle-free graphs from those with one or more triangles requires space proportional to the number of edges. Other works have required even stronger assumptions, such as a bound on $T_2$, the number of paths of length 2, or the maximum degree of the graph.