Future Generation Computer Systems 108 (2020) 244–255

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

Reservoir-based sampling over large graph streams to estimate triangle counts and node degrees

Lingling Zhang [a], Hong Jiang [b], Fang Wang [a,*], Dan Feng [a], Yanwen Xie [a]

[a] School of Computer Science and Technology, Huazhong University of Science and Technology, China
[b] Department of Computer Science and Engineering, University of Texas at Arlington, USA

ARTICLE INFO

Article history:
Received 26 November 2019
Received in revised form 15 January 2020
Accepted 26 February 2020
Available online 2 March 2020

Keywords:
Reservoir sampling
Graph streams
Triangle counts
Node degrees

ABSTRACT

Reservoir sampling is widely employed to characterize large graph streams by producing edge samples. However, existing reservoir-based sampling methods mainly focus on counting triangles but perform poorly in analyzing topological characteristics reflected by node degrees. This paper proposes a new method, called triangle-induced reservoir sampling, or T-Sample, to count triangles and estimate node degrees simultaneously and efficiently. While every edge in a graph stream is processed only once by T-Sample, a dual sampling mechanism performing both uniform sampling and non-uniform sampling is carefully designed. Specifically, T-Sample's uniform sampling is used to count triangles by a newly proposed method with smaller estimation variances than existing reservoir-based sampling methods; whereas, its non-uniform sampling ensures that edge samples are connected. Experimental results driven by real datasets show that T-Sample can count triangles with smaller estimation errors and variances than the state-of-the-art reservoir-based sampling methods while obtaining much more accurate information about node degrees at smaller time and memory costs.

1. Introduction

The rapid growth in real-world application scenarios, e.g., bioinformatics, social media, computer network traffic, etc., necessitates the storage, processing and analysis of the data content of large graph streams. In a graph stream, each edge carries the information about interaction between one node (entity) and another node (entity) [1–4]. Importantly, these edges are not isolated in representing pair-wise node interactions but are usually connected in some ways to convey valuable information about a graph stream [5–7]. Given the sheer size of data, it is much more cost effective to use samples to substitute for the entire original dataset to analyze the graph stream [4,8].

Many recent studies focus on *one-pass stream sampling methods* [8–13], in which each edge is processed for only one time to determine whether it is a candidate sample by either of two analyses. The first is an analysis of counting triangles (triangle-count) of a graph stream, which has attracted considerable attentions [10,12–14]. The second is to show connectivity by an analysis of the number of different node-degrees and the numbers of nodes with specific node-degrees in a graph stream, referred to respectively as *node-degree types* and *node-degree counts* in this paper.

However, existing one-pass sampling methods only focus on characterizing the connectivity either from triangle-count analysis or node-degree analysis but not both. To produce edge samples conducive to obtaining the two analyses simultaneously, a new one-pass stream sampling method, called triangle-induced reservoir sampling or T-Sample is proposed in this paper by employing a dual sampling mechanism, namely, combining a uniform sampling with a non-uniform sampling to produce edge samples. The uniform and non-uniform samplings in T-Sample cooperate to estimate both triangle counts and node degrees over large graph streams.

With the design and implementation of T-Sample, this paper makes the following contributions.

1. To the best of our knowledge, T-Sample, as a one-pass stream sampling method, is a first attempt at characterizing the connectivity of large graph streams by counting triangles while presenting an approximate description of the actual connectivity for a graph stream by proposing a dual sampling mechanism. (Section 3)
2. T-Sample's uniform sampling is used to estimate the total triangle counts by a newly proposed method with higher accuracy and smaller estimation variance than the existing one-pass reservoir-based sampling methods. (Section 4)

* Corresponding author.
E-mail addresses: llzh@hust.edu.cn (L. Zhang), hong.jiang@uta.edu (H. Jiang), wangfang@hust.edu.cn (F. Wang), dfeng@hust.edu.cn (D. Feng), ywxie@hust.edu.cn (Y. Xie).

https://doi.org/10.1016/j.future.2020.02.077
0167-739X/

3. T-Sample uses a limited memory capacity to produce edge samples which can be proven to be connected theoretically and empirically. These edge samples thus can be employed to estimate information about the node degrees of a graph stream. (Section 4)
4. Experimental results driven by different real datasets show that T-Sample can obtain more accurate information about node degrees than the existing reservoir-based sampling methods at smaller time and memory costs over the graph streams with more than one-billion edges. At the same time, the experimental results show that T-Sample can estimate the triangle counts with smaller errors and variances than the existing one-pass sampling methods. (Section 5)

The rest of the paper is organized as follows. Section 2 describes the related work and motivation of T-Sample. Section 3 elaborates on T-Sample's design while Section 4 describes how to count triangles and why T-Sample can be used to estimate node degrees for a large graph stream. Section 5 presents the evaluations of T-Sample while Section 6 concludes our work. Based on T-Sample, Section 7 illustrates the future work.

2. Related work and motivation

Although there are many sampling techniques for estimating node degrees [15–18], they are not applicable in graphs in the form of edge streams because these sampling techniques focus on producing samples in the form of nodes [19–21]. The existing reservoir-based sampling methods can be classified into two categories, i.e., uniform reservoir-based sampling, which is capable of learning the probability of an edge entering a reservoir in a graph stream prior to sampling, and non-uniform reservoir-based sampling for which the probability of an edge entering a reservoir is not known before the sampling process. In what follows we introduce these two categories, along with the shortcomings of their representative methods, to motivate the T-Sample research.

2.1. Uniform reservoir-based sampling

In this approach, the probability of an edge entering a reservoir is known by setting either a static reservoir capacity or a static probability.

• **Static reservoir capacity.** Suppose that c is the static capacity of a reservoir used by a reservoir-based sampling method that first preserves the front c edges of a graph stream directly in the reservoir. With each subsequent arriving edge, there are two probabilities that must be set to determine the outcomes of two corresponding events, *entrance* — whether the new edge will enter the reservoir or not — and *replacement* — which edge currently in the reservoir will be replaced given the entrance of the new edge. Existing uniform reservoir-based sampling methods set these probabilities in either a uniform way or a weighted way, as follows.

(a). Uniform setting. Let p_i^{in} denote the probability of the ith arriving edge entering the reservoir and p_{out} the probability of an edge already preserved in the reservoir being replaced by the newly sampled edge. p_i^{in} and p_{out} are given as,

$$p_i^{in} = min\{1, \frac{c}{i}\},\ p_{out} = \frac{1}{c}, \tag{1}$$

Triest and Triest-IMPR, proposed in [14], employ the idea of uniform setting to count the triangles. Triest and Triest-IMPR do not consider the actual connectivity among the edges, rendering the produced edge samples useless in estimating the node degrees. Furthermore, in Triest and Triest-IMPR, the probability of forming triangles by the edge samples, which is required for estimating the triangle counts, is inferred without consideration about the specific sampling probability of each edge in a graph stream and thus increase the estimation variance as analyzed in detail in Section 4.

(b). Weighted setting. The probability of the currently processed edge entering a reservoir can be set according to a prescribed rule (i.e., a randomly generated number). Specifically speaking, if the newly arrived edge has a higher weight than the smallest weight, labeled as z^*, of the edges preserved in the reservoir thus far, it is selected to be stored in the reservoir replacing the edge with the smallest weight.

GPS Post stream (GPS-Post) and GPS In stream (GPS-In), proposed in [8], use the idea of weighted setting to sample large graph streams. Intuitively, the weighted setting can help characterize the actual connectivity of a graph stream when the weight of the newly arrived edge is set according to its connectivity with the edges preserved in the reservoir, i.e., the number of triangles formed by the newly arrived edge and the edges preserved in the reservoir thus far. However, there are three problems associated with this sampling process. First, it is costly to sort the weights of the edges in the reservoir to select the edge with the minimum weight while deciding whether the newly arrived edge is to be sampled or not. The total number of sort operations is equal to the number of edges in the graph stream. Second, GPS-Post and GPS-In cause major estimation variances and errors of the total number of triangles that is highly dependent on how the weights are set. Third, the edge samples produced by GPS-Post and GPS-In, which are currently connected with one another, may be replaced in the subsequent sampling process given that the capacity of the reservoir is limited and static.

• **Static probability.** It is a straightforward way for reservoir-based sampling methods, such as ***Mascot*** and ***Graph Sample and Hold (GSH)***, proposed in [9,10] respectively to produce edge samples by setting a uniformly static value as the probability of each edge entering a reservoir in a graph stream. However, the probability is arbitrarily set, which may cause large estimation errors on triangle counts or occupy large memory because of no strategies used to limit the reservoir capacity. Furthermore, similar to Triest and Triest-IMPR that do not consider the specific connectivity among edges during the sampling process, the edge samples produced by Mascot and GSH are rarely connected, making them inadequate in reflecting the actual connectivity of a graph stream.

2.2. Non-uniform reservoir-based sampling

In contrast to the uniform reservoir-based sampling methods, their non-uniform counterparts consider the specific connectivity among the edges when sampling, although the probability of each edge entering a reservoir by the latter cannot be learned before the sampling process. Existing non-uniform sampling methods can be divided into two groups. Methods in the first are designed to produce edge samples, such as ***NeiSampling*** [22] and ***StreamSampling*** [23] while those in the second produce node samples, such as ***PIES*** [24]. Their drawbacks are analyzed below.

The processing times of both ***NeiSampling*** and ***StreamSampling*** are long. In NeiSampling, a newly arrived edge is compared with edges preserved in a base reservoir c times, where c is the capacity of the base reservoir, to determine whether it can be sampled or form triangles with the edges already preserved in the base reservoir. In StreamSampling, each newly arrived edge is compared with $2 \times c$ times to determine whether it can form triangles with the edges preserved in an auxiliary reservoir whose capacity is equal to the base reservoir, while determining whether it can be sampled and further form wedges (i.e., any two of the three edges of a triangle) with the edges preserved in the base

reservoir. Besides, in StreamSampling, the expected number of times of updating the edges preserved in the base reservoir is $c \times log(n)$ with n being the total number of edges in a graph stream. Furthermore, the edges preserved in the base reservoir or the auxiliary reservoir are always updated. Thus, even if the edges preserved in them are connected with one another at some time, they may be replaced by edges arriving in the subsequent sampling process.

PIES, as a representative non-uniform reservoir-based node sampling method proposed in [24], is highly cost-effective by using a reservoir of static capacity c to preserve the node samples. The front c distinct nodes are preserved in the fixed-capacity reservoir while edges relevant to these nodes are preserved in an auxiliary reservoir with a dynamic capacity of m. The entrance and replacement probabilities in PIES are set as follows. If an edge $e's$ two nodes are already in the base reservoir, e is preserved in the auxiliary reservoir directly. Otherwise, it is necessary to replace the nodes preserved in the base reservoir by the nodes of the ith edge with the probability $\frac{m}{i}$. Meanwhile, the edges preserved in the auxiliary reservoir, which are relevant to the replaced nodes, will be eliminated from the auxiliary reservoir.

However, it is costly by using the node samples to estimate the node degrees in PIES — the c nodes in the base reservoir need $c \times n$ comparisons with the edges in the graph stream whose total size is n. Worse still, PIES cannot be used to estimate the total number of triangles and there is almost no connectivity among the edge samples.

2.3. Motivation

In many classes of applications, the analyses of both triangle counts and node degrees of a large graph stream are required simultaneously. For example, when to evaluate the propagation of a product, rumor and so on over a large graph stream, the more triangles mean that users can be infected from multiple channels while the properties of the node degrees are used to reflect the users can be infected from a single channel [25–27]. Suppose that α and β are the propagation coefficients of one channel and multiple channel respectively. We use inf to denote the average influence (propagation ability) of a large graph stream as described in Eq. (2), where $AverageND$ is the average degree of a node and $AverageTC$ is the average triangle counts of a node.

$$Inf = AverageND \times \alpha + AverageTC \times \beta, \tag{2}$$

Suppose that $3\alpha = \beta$. According to Eq. (2), the average influence of the graph stream shown in Fig. 1(a) is $4\alpha + 0.196\beta = 4.588\alpha$ while that of another stream shown in Fig. 1(b) is $3.36\alpha + 0.419\beta = 4.617\alpha$. Obviously, the average influence of a graph stream is decided by both the triangle counts and node degrees rather than a single factor. From the above analysis, both the existing uniform and non-uniform reservoir-based sampling methods produce edge samples that no longer contain or convey sufficient actual connectivity of a graph stream. As shown in Fig. 2, the existing reservoir-based sampling methods provide very limited information about the node-degree types as the percentage of the nodes with degrees more than 10 is almost equal to zero. Furthermore, Fig. 2 also shows that the degrees of most of the nodes (92% at most and 70% at least) are equal to one, based on the edge samples obtained by the existing reservoir-based methods implemented in the same platform (described in Section 5). Since the degree of any node is at least equal to one because each edge consists of exactly two nodes, the results in Fig. 2 clearly imply that the edge samples produced by the existing reservoir based sampling methods are mostly isolated, unconnected edges.

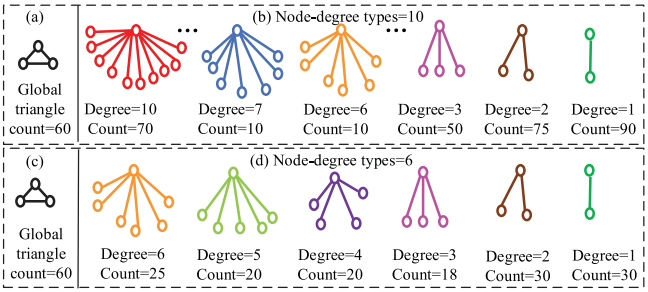


Fig. 1. An example of connectivity of two graph streams in terms of triangle-count and node-degree analyses. (a) and (b) describe one graph stream while (c) and (d) depict another. Although the two graph streams have the same triangle count (60), they have very different node-degree types (10 vs. 6) and counts (skewed vs. less skewed) for specific node-degrees labeled in different colors.

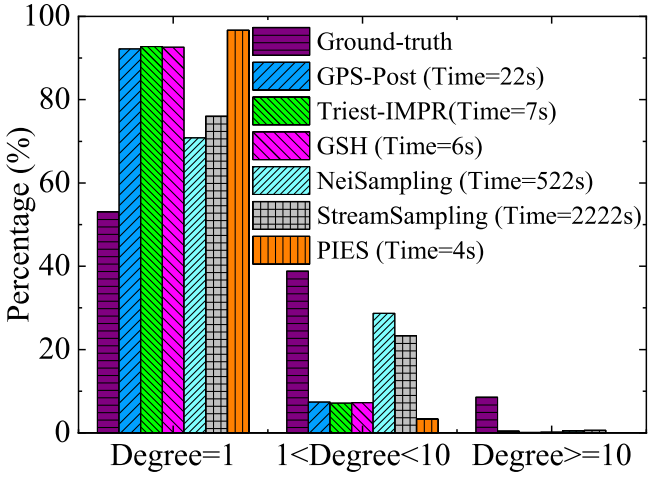


Fig. 2. The distribution of node-degree counts (and processing times) of edge samples generated by the existing reservoir-based sampling methods over the Youtube graph stream (Section 5) when the capacity of the reservoir is set to 5 K.

Furthermore, except for PIES which does not estimate the triangle counts, Fig. 2 shows that the non-uniform sampling methods (NeiSampling and StreamSampling) have slightly better results than the uniform sampling ones in estimating the information of node degrees while they spend much more time than the latter. Fig. 2 implies that it is more cost-efficient to estimate the triangle counts using the uniform reservoir-based sampling while it is more effective in estimating the specific topological characteristics by non-uniform reservoir-based sampling. Therefore, to meet the requirement of the applications to analyze both the node degrees and the triangle counts, a new reservoir-based sampling method should inherit the advantages of both categories of uniform and non-uniform sampling while alleviating their disadvantages.

Specifically, in designing such a new one-pass stream sampling method, three factors should be taken into consideration. First, the specific connectivity among the edges should be considered, similar to the non-uniform reservoir-based sampling methods, while reducing the processing times. Second, the constraint of the static capacity of the reservoir should be alleviated during the non-uniform sampling process so that the edge samples produced by the new method are no longer replaced by the newly sampled edges to avoid the loss of the connectivity already preserved in the current edge samples. Third, to estimate the

Table 1
The comparisons of the cost and accuracy among the reservoir-based sampling methods.

Methods		Cost	Accuracy	
			Triangle count	Node degree
Uniform reservoir-based sampling	Triest	Low	High	Low
	Triest-IMPR	High	High	Low
	GPS-Post	Medium	High	Medium
	GPS-In	High	High	Medium
	Mascot	Low	High	Low
	GSH	Low	Medium	Low
Non-uniform Reservoir-based sampling	PIES	Low	Impossible	Medium
	NeiSampling	High	High	Medium
	StreanSampling	High	High	Medium
T-Sample		Low	High	High

Table 2
The symbols used in this paper.

$G = (V, E)$	Graph stream G with node set V and edge set E
c	Capacity of the base reservoir
R_{base}	Base reservoir
R_{incre}	Incremental reservoir
num_i	Total number of edges, among the front $i-1$ edges satisfying the prerequisite for entering R_{incre}
p_i^{in}	Probability of the ith edge being preserved in R_{base} in T-Sample
p_i	Sampling probability of each of the front i edge at the arrival of ith edge in T-Sample
p_i^{incre}	Probability of the ith edge entering R_{incre} in T-Sample
$T_{capacity}$	Total capacity of T-Sample
T_{time}	Total processing time of T-Sample
$p(\triangle_{mean})$	Mean probability of the edges in a graph stream satisfies the prerequisite of entering R_{incre}
$p(\triangle)_i$	Probability of triangles formed by the front $i-1$ edges and the ith edge
$p(\triangle)_i^{T-Sample}$	Probability of triangles formed by the front $i-1$ edges and the ith edge via T-Sample

triangle counts in a very short time efficiently, the idea of the uniform reservoir-based sampling can be leveraged while designing a method for counting the triangles with smaller estimation errors and variances. Motivated by these insights, we propose in this paper a new reservoir-based sampling method, called T-Sample that employs a dual-sampling mechanism and is capable of producing connected edge samples for analyzing both the triangle counts and node degrees of a large graph stream. Table 1 shows the cost and accuracy of the reservoir-based sampling methods. Compared with the existing uniform and non-uniform reservoir-based sampling methods, T-Sample can estimate the triangle counts and node degrees accurately at low costs.

3. Design and analysis of T-Sample

In this section, we first elaborate on the design of T-Sample. Then, we analyze T-Sample in detail to learn the probabilities of an edge being sampled by uniform reservoir-based sampling and non-uniform reservoir-based sampling respectively, which is followed by the description of T-Sample's memory usage and processing time. The notations frequently used in this paper and their definitions are presented in Table 2.

3.1. Dual sampling

T-Sample, as a one-pass sampling method for which each edge of a graph stream is processed only one time, employs a dual sampling mechanism (uniform and non-uniform) with the help of triangles in the graph stream. The key idea behind the dual sampling mechanism is that the newest edge sample produced by the non-uniform sampling can form at least one triangle with the edge samples that the uniform sampling has produced thus far. Specifically, such a sampling process relies on two types of reservoirs, base reservoir and incremental reservoir, to estimate the triangle counts while simultaneously obtaining the actual connectivity information of a graph stream based on node degrees (their types and counts).

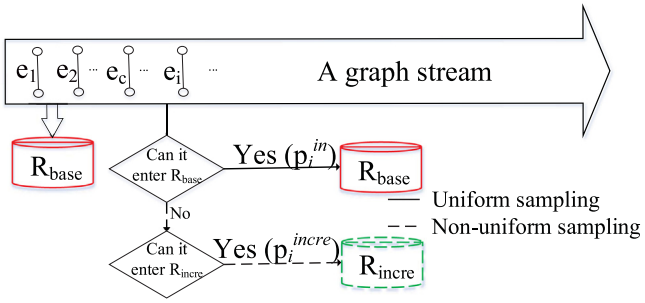


Fig. 3. T-Sample's dual sampling mechanism, with uniform sampling in solid black lines and non-uniform sampling in dashed black lines while p_i^* denotes the probability of entering a reservoir respectively for the ith edge.

Base reservoir. T-Sample's uniform sampling employs a reservoir with a static capacity, namely, a base reservoir R_{base}, to learn the sampling probabilities of the edges in a graph stream by employing the uniform setting (Section 2). These probabilities are used to infer the probabilities of the triangles formed by the edge samples, which in turn are used to estimate the triangle counts as described in Section 4. An important characteristic for the edge samples preserved in the base reservoir is that these edges are updated frequently with the arrival of each new edge in a graph stream while the number of the edges preserved in it is static.

Incremental reservoir. T-Sample's non-uniform sampling employs a reservoir with a dynamic capacity, namely, an incremental reservoir R_{incre}, to produce the connected edge samples. The ***prerequisite*** for an edge to enter the incremental reservoir is that the edge can form triangles with the edges currently preserved in the base reservoir. An edge once sampled by the non-uniform sampling cannot be removed from the incremental reservoir. Therefore, the volume of the edges preserved in the incremental reservoir, is always non-decreasing. To limit the memory space used by the incremental reservoir, we design a parameter to control the probability of an edge entering into the incremental reservoir by exploiting the density/sparsity of the connectivity of a graph stream. Before we derive the sampling probabilities, we first present the work flow of T-Sample with its dual-sampling mechanism.

As illustrated in Fig. 3, at the very beginning of T-Sample's process, the front c edges of a graph stream are directly preserved in the base reservoir of capacity c. From this point on, *whether a newly arrived edge is sampled or not by T-Sample depends on if the edge has a chance to be preserved in either the base reservoir or the incremental reservoir.* Notice that any edge of a graph stream can only be preserved in at most one of the two reservoirs. Fig. 3 depicts the T-Sample's dual sampling process: the ith ($i > c$) edge will first try to enter the base reservoir and, when this effort fails, it then tries to enter the incremental reservoir.

Generally speaking, each edge in a graph stream has a chance to be preserved in the base reservoir and thus the probabilities of the triangle formations based on the whole graph stream can be inferred accurately. On the other hand, the edges that fail to

enter the base reservoir have chances of entering the incremental reservoir by leveraging the important structures of triangles that express the basic and cohesive connectivity among the edges in a graph stream. Thus, the edge samples, preserved in both the base and incremental reservoirs, are able to largely preserve the connectivity.

3.2. Sampling probabilities

Due to the dual sampling mechanism, the sampling probability of an edge in T-Sample is analyzed from two cases as follows: that in uniform sampling and that in non-uniform sampling. The former is used to estimate the total triangle counts while the latter helps determine the total memory usage and infer the actual connectivity among the edge samples.

Uniform sampling. Recall from Section 2 that $p_i^{in} = min\{1, \frac{c}{i}\}$ is the probability of the ith edge entering the base reservoir with a capacity of (c) and $p_{out} = \frac{1}{c}$ is the probability of an edge already preserved in the base reservoir being removed from it later due to the entrance of a newly sampled edge. Thus, the sampling probability of any of the front i edges at the time when the ith edge $(i > c)$ is being processed is given in Lemma 1.

Lemma 1. *Suppose that the ith edge $(i > c)$ is being processed in a graph stream, then the sampling probability (p_i) of any of the front i edges is equal to $p_i = \frac{c}{i}$.*

Proof. Suppose that the kth edge is currently preserved in the base reservoir, meaning that it has entered into the base reservoir without being replaced in the remainder of $(i-k)$ sampling steps. Thus, the sampling probability of the kth edge at the time of processing the ith edge is given as,

$$p_i = p_k^{in} \times \prod_{j=k+1}^{i} (1 - p_j^{in} + p_j^{in} \times (1 - p_{out})) \tag{3}$$

Though $p_k^{in} = 1$ if $k \leq c$ and $p_k^{in} = \frac{c}{k}$ if $k > c$, the processes of computing p_i for the two cases are similar. To simplify the proof, suppose $k > c$, then p_i is computed as,

$$\begin{aligned} p_i = &\frac{c}{k} \times [(\frac{j-c}{k+1} + \frac{c}{k+1} \times \frac{c-1}{c}) \times \cdots \\ &\times (\frac{i-c}{i} + \frac{c}{i} \times \frac{c-1}{c})] = \frac{c}{i}. \end{aligned} \tag{4}$$

Non-uniform sampling. Since the size of the incremental reservoir is non-decreasing with i, the sampling probability for edges to be preserved in it must be properly controlled to limit its memory usage while preserving the topological structures approximately in terms of node-degree types and counts.

Intuitively, a more densely connected graph stream tends to have a correspondingly higher triangle counts, implying that a newly arrived edge is more likely to form at least one triangle with edges preserved in the base reservoir and thus meet the perquisite for entering the incremental reservoir. On the other hand, the opposite is true for a sparsely connected graph stream, i.e., a newly arrived edge is less likely to meet the prerequisite. Based on this intuition, the parameter $\frac{c}{c+num_i}$ helps indicate whether a graph stream being sampled is densely or sparsely connected, where num_i is the total number of edges satisfying the prerequisite for entering the incremental reservoir among the front $i-1$ edges and can be calculated during the process of counting the triangles (Section 4). That is, the lower the value of this parameter, the more densely connected a graph stream is. Thus, we use this parameter $\frac{c}{c+num_i}$ to control the probability of an edge entering the incremental reservoir and further limit T-Sample's memory usage.

Specifically, in face of a densely connected graph stream, the value of $\frac{c}{c+num_i}$ decreases rapidly as i increases, meaning that the probability of an edge entering the incremental reservoir will diminish rapidly. This helps limit the number of edges added to the incremental reservoir when sampling a densely connected graph stream for which there are indeed many edges already preserved in the incremental reservoir. On the other hand, for a sparely connected graph stream, the value of $\frac{c}{c+num_i}$ decreases very slowly as i increases, meaning that the probability of an edge entering the incremental reservoir will remain relatively steady. This helps obtain as many connected edge samples as possible for uncovering the original connectivity of a sparsely connected graph stream. Therefore, the probability p_i^{incre} of the ith edge entering the incremental reservoir is given as,

$$p_i^{incre} = 1_i^{meetPre} \times (1 - p_i^{in}) \times \frac{c}{c + num_i}, \tag{5}$$

where $1_i^{meetPre}$ signifies whether an edge meets the prerequisite to enter the incremental reservoir. In other words, in an actual sampling process, $1_i^{meetPre} = 1$ means the ith edge meets the prerequisite, or $1_i^{meetPre} = 0$ otherwise. The sampling process of T-Sample is described in Algorithm 1. Lines 4–9 of the pseudocode decide whether an edge can enter R_{base} or not, according to $p_i^{in} = \frac{c}{i}$. If an edge is denied of its entrance to R_{base}, p_i^{incre} is evaluated by Lines 10–15 to decide whether it can enter R_{incre} or not.

Algorithm 1: T-Sample

```
Input: E = {e_1, e_2, ..., e_n}: a graph stream and c: the capacity of R_base;
Output: S: set of edge samples;
1  R_base ← {e_1, ..., e_c} and R_incre ← ∅ ;
2  for i ← c + 1 to n do
3      num_i ← num_{i−1};
4      p_i^in = c/i;
5      Generate r_1 randomly from (0,1];
6      if r_1 < p_i^in then
7          Remove an edge from R_base randomly;
8          R_base ← R_base ∪ {e_i};
9      else
10         if e_i can form triangles with the edges in R_base then
11             num_i ← num_i + 1;
12             Generate r_2 randomly from (0,1];
13             if r_2 < c/(c+num_i) then
14                 R_incre ← R_incre ∪ {e_i};
15 S = R_base ∪ R_incre;
```

3.3. Total reservoir capacity and processing time

Total reservoir capacity. From the above description, the total capacity of the reservoirs used by T-Sample is dynamic because of the unknown capacity of the incremental reservoir. Nevertheless, it can be described quantitatively as follows to estimate T-Sample's memory cost.

As described above, with the increase in the number of the edges preserved in the incremental reservoir, the probability of the edges entering this reservoir decreases. Suppose that the set $TE = \{te_1, te_2, \ldots, te_f\}$ contains the edges which can form triangles with the edges currently preserved in the base reservoir during the whole sampling process, where $TE \subseteq E$. Combined with the probability of entering the incremental reservoir described in Eq. (5), T-Sample's total reservoir capacity $T_{capacity}$ is

described quantitatively as follows.

$$\begin{aligned} T_{capacity} &= c + \sum_{e_i \in TE} p_i^{incre} \\ &= c + \sum_{e_i \in TE} (1 - p_i^{in}) \times \frac{c}{c + num_i} \times p(\triangle_{mean}) \\ &< c + \sum_{i=c+1}^{|E|} (1 - \frac{c}{i}) \times \frac{c}{i} \times p(\triangle_{mean}) \\ &< c + p(\triangle_{mean}) \times \sum_{i=c+1}^{|E|} \frac{c}{i}, \end{aligned} \tag{6}$$

where let $p(\triangle_{mean})$ be the mean probability of an edge in a graph stream satisfying the prerequisite of entering the incremental reservoir and is determined by the specific connectivity of a large graph stream. Although the specific value of $p(\triangle_{mean})$ is unknown without prior knowledge of the connectivity of a graph stream being sampled, it can be used to show that the more connected edges in the original graph stream, the more capacity is required by the incremental reservoir.

According to the Equation 2.1 in [28], we have the following equation:

$$\sum_{i=c+1}^{|E|} \frac{c}{i} = \frac{c}{c+1} + \cdots + \frac{c}{|E|} \simeq c \times ln(\frac{|E|}{c}). \tag{7}$$

Thus, we have:

$$T_{capacity} < c(1 + ln(\frac{|E|}{c}) p(\triangle_{mean})) \ll c(1 + ln(\frac{|E|}{c})). \tag{8}$$

Therefore, the design of the capacity of the incremental reservoir overcomes the disadvantage of the static reservoir capacity used by the existing non-uniform reservoir-based sampling methods to produce connected edge samples. On the other hand, the total capacity of T-Sample's reservoirs is much smaller than the total volumes of a graph stream and is subject to the capacity of the base reservoir and the connectivity of a graph stream itself, as expressed in Eq. (8)).

Processing time. As described in Section 1, T-Sample is designed to carry out the two tasks of counting triangles and estimating node degrees simultaneously by using the dual sampling mechanism. Therefore, T-Sample's processing time (T_{time}) can be given as,

$$T_{time} = T_{un} + T_{non-un} + T_{tc(pre)}, \tag{9}$$

where T_{un} and T_{non-un} are the times of producing edge samples to estimate node degrees in T-Sample's uniform and non-uniform sampling processes respectively and $T_{tc(pre)}$ is the processing time of counting triangles which overlaps with the processing time of determining whether an edge meets the prerequisite for entering the incremental reservoir. In other words, the operation of counting the triangles formed by any two edges in the base reservoir and the ith edge can also be used to confirm whether the ith edge satisfies the prerequisite for entering the incremental reservoir. For an existing uniform reservoir-based sampling method to finish the two tasks, i.e., Triest-IMPR, the processing time ($Uniform_{time}$) is given as,

$$Uniform_{time} = Uniform_{un} + Uniform_{tc}, \tag{10}$$

where $Uniform_{un}$ and $Uniform_{tc}$ are the processing times of producing edge samples and counting triangles in the uniform reservoir-based sampling method respectively. Thus, in contrast to the existing method, T-Sample spends slightly more time in non-uniform reservoir-based sampling to produce the same number of edge samples in the base reservoir as those by the existing method. However, T_{non-un} is relatively small because T-Sample's non-uniform sampling only determines whether an edge can enter a reservoir based on the given sampling probability. The experimental results of PIES in Section 2 confirm that such non-uniform sampling spends very little time even based on the whole graph stream. Therefore, T-Sample inherits the advantage of the short processing time of the existing uniform reservoir-based sampling methods while preserving the connectivity approximately of a large graph stream with a limited memory usage.

4. Estimations for triangle counts and node degrees

In this section, we first propose a new, improved method to count the total number of triangles based on T-Sample's uniform reservoir-based sampling process. Then, we prove that T-Sample is able to produce connected edge samples, which is the basis for uncovering information of the node degrees.

4.1. Triangle counts

To reduce the estimation errors and variances caused by existing triangle-counting approaches, we propose a new method to count triangles, referred to as TS-Triangle. To better understand TS-triangle and its comparisons with the state-of-the-art methods, the ideas of obtaining the ground-truth of triangle counts and the estimation of the triangle counts via sampling are described as follows.

The ground-truth of triangle counts. The key idea behind TS-Triangle is to distinguish two ways triangles ($\triangle^i$) are formed by the front i edges in a graph stream, i.e., those ($\triangle^{i-1}$) formed exclusively by the front $i-1$ edges and those ($\triangle^{\{i-1,i\}}$) formed by any two edges of the front $i-1$ edges and the ith edge. The first group of triangles can in turn be divided into two groups, those formed exclusively by the front $i-2$ edges and those formed by any two edges of the front $i-2$ edges and the $(i-1)$th edge, and this recursive process can continue for the front $i-3$ edges, $i-4$ edges, …, etc. Based in thus recursive way of triangle formations, an iterative equation can be used to obtain the ground-truth of triangle counts in a graph stream.

$$\triangle^i = \triangle^{i-1} + \triangle^{\{i-1,i\}} = \triangle^c + \sum_{j=c+1}^{i} \triangle^{\{j-1,j\}}, \tag{11}$$

where $\triangle^{i-1} = \triangle^{i-2} + \triangle^{\{i-2,i-1\}}$ and $\triangle^{i-2}$, $\triangle^{i-3}$, …, $\triangle^{c+1}$ can be expressed in the same iterative way while $\triangle^c$ is the number of triangles formed by the front c edges.

The estimation of triangle counts via sampling. As the order of computing $\triangle^{\{c,c+1\}}$, …, $\triangle^{\{i-1,i\}}$ $(i > c)$ is exactly the same as that of a typical reservoir-based sampling method processing the front i edges, the total number of triangles ($\triangle^i$) formed by the front i edges can be obtained by the sum of triangles from the $(c+1)$th sampling step through the ith step. Thus, based on a sampling method, the estimated value (labeled as $\triangle^i_{estimated}$) of $\triangle^i$ is given as,

$$\triangle^i_{estimated} = \sum_{i=3}^{n} \sum_{\alpha \in \triangle_i^\tau} \frac{E(\lambda_\alpha)}{p(\triangle_\alpha)}, \tag{12}$$

where $\triangle_i^\tau$ is the set of all triangles formed by any two sampled edges from the front $i-1$ edges and the ith edge and $p(\triangle_\alpha)$ is the probability of the triangle α being formed by any two of the front $i-1$ edges and the ith edge. Furthermore, $\lambda_\alpha = 1$ denotes the triangle α can be obtained by two sampled edges of the front $i-1$ edges and the ith edge, $\lambda_\alpha = 0$ otherwise. Based

on LEMMA 1 in [10], it is obvious that $E(\lambda_\alpha) = p(\triangle_\alpha)$ and thus $E(\triangle^i_{estimated}) = \triangle^i$.

The estimation of triangle counts via T-Sample. To estimate the triangle counts by T-Sample, it is necessary to obtain the probability ($p(\triangle)_i^{T-Sample}$) of the triangles formed by any two of the front $i-1$ edges and the ith edge during the T-Sample's process. As described in Lemma 1 in Section 3, each edge of the front $i-1$ edges in T-Sample's uniform sampling process has the same chance $\frac{c}{i-1}$ to be preserved in the base reservoir at the time of the ith edge's arrival. Thus, $p(\triangle)_i^{T-Sample} = min\{1, (\frac{c}{i-1})^2\}$ and the estimation of the total number of triangles, denoted by $\triangle_{TS-Triangle}$, is given as,

$$\triangle_{TS-Triangle} = \sum_{i=3}^{n} \frac{m_i^{T-sample}}{p(\triangle)_i^{T-Sample}}, \tag{13}$$

where $m_i^{T-Sample}$ denotes the number of triangles formed by the edge samples obtained from the front $i-1$ edges and the ith edge in T-Sample's uniform sampling.

Although in this paper we focus on leveraging the total number of triangles to support a description of the overall connectivity of a graph stream, the idea of counting the local triangles ($\triangle(\mu)^i$) that a specific node (μ) participates in among the front i edges is similar to that of counting the global triangles ($\triangle^i$) (the total number of triangles) in a graph stream as described in Eq. (11). Once the probability of any triangle obtained from the front i edges is estimated, TS-Triangle can be easily extended to estimate $\triangle(\mu)^i$ using Eq. (13) by replacing $m_i^{T-Sample}$ with $m(\mu)_i^{T-sample}$ which denotes the number of triangles that a specific node (μ) participates in and can be obtained from the edge samples.

Comparison with Triest-IMPR. As one of the most representative triangle-counting algorithms for large graph streams using sampling methods, Triest-IMPR [13] has theoretically smaller estimation variance than other methods, such as Triest [13], Mascot [10] and Neighbor-Sampling [22]. Since Triest-IMPR uses a similar idea of iterative counting of triangles (Eq. (12)), it is necessary for Triest-IMPR to compute the probability $p(\triangle_\alpha)$ ($\alpha \in \triangle_i^\tau$). In Triest-IMPR, the probability ($p(\triangle)_i^{Triest-IMPR}$) of each triangle being formed by two sampled edges from the front $i-1$ edges and the ith edge is based on Lemma A.1 in [13] and the total number of triangles ($\triangle^i_{Triest-IMPR}$) is expressed as,

$$\triangle^i_{Triest-IMPR} = \sum_{i=3}^{n} \frac{m_i^{Triest-IMPR}}{p(\triangle)_i^{Triest-IMPR}}, \tag{14}$$

where $p(\triangle)_i^{Triest-IMPR} = min\{1, \frac{c\times(c-1)}{(i-1)\times(i-2)}\}$ and $m_i^{Triest-IMPR}$ is the number of triangles formed by any two of the edge samples among the $(i-1)$ edges and the ith edge in Triest-IMPR. Based on how probability $p(\triangle)_i^{Triest-IMPR}$ is inferred, Triest-IMPR does not consider the specific sampling probabilities of the edges forming the triangles, which increase the estimation errors and variances as explained below.

Based on Eq. (12), the estimation variance of the total number of triangles $\triangle^\tau$ can be given as,

$$\begin{aligned} Var(\triangle^i_{estimated}) &= Var(\frac{1}{p(\triangle)_i}\sum_{\alpha\in\triangle_i^\tau}\alpha) \\ &= \frac{1}{p(\triangle)_i{}^2}\sum_{\alpha\in\triangle_i^\tau}\sum_{\beta\in\triangle_i^\tau} Cov(\alpha,\beta) \\ &= \frac{1}{p(\triangle)_i{}^2}[\sum_{\alpha\in\triangle_i^\tau} Var(\alpha) + \sum_{\alpha\in\triangle_i^\tau}\sum_{(\beta\in\triangle_i^\tau\ \alpha\neq\beta)} Cov(\alpha,\beta)] \end{aligned} \tag{15}$$

According to Eq. (15), $Var(\alpha) = p(\triangle)_i - p(\triangle)_i^2$. If the two triangles ($\alpha$ and β) do not share any edge, then $Cov(\alpha,\beta) = 0$. Otherwise, $Cov(\alpha,\beta) < p(\alpha) - p(\alpha)\times p(\beta) = p(\triangle)_i - p(\triangle)_i^2$. Therefore, the estimation variance of the total number of triangles is decided by the probability $p(\triangle)_i$. The larger the $p(\triangle)_i$ value, the smaller the estimation variance is. According to the above description of T-Sample and Triest-IMPR, when $i > c+1$, the relationship between $p(\triangle)_i^{T-Sample}$ and $p(\triangle)_i^{Triest-IMPR}$ is given as,

$$\frac{p(\triangle)_i^{T-Sample}}{p(\triangle)_i^{Triest-IMPR}} > \frac{c\times(i-2)}{(c-1)\times c} = \frac{i-2}{c-1} > 1. \tag{16}$$

Thus, the following relationship is established,

$$Var(\triangle^i_{TS-Triangle}) < Var(\triangle^i_{Triest-IMPR}). \tag{17}$$

4.2. Node degrees

The premise for estimating node degrees of a large graph stream is that the obtained edge samples have connectivity. In T-Sample, since the edge samples produced by uniform sampling are rarely connected as discussed in Section 2, the actual connectivity among the edge samples produced by T-Sample mainly stems from the following two types of connectivity.

Connectivity between the edges preserved in the base reservoir and those in the incremental reservoir. Since the prerequisite for the edges entering R_{incre} is their connectivity with edges in R_{base} to form triangles, the edges being preserved in the incremental reservoir must be connected with the edges currently preserved in the base reservoir.

Connectivity among the edges preserved in the incremental reservoir. If the incremental reservoir has zero or small capacity after T-Sample's process, meaning that the graph stream is very sparsely connected, it is sufficient for the edges in the base and incremental reservoirs to characterize the connectivity of a very sparse graph stream. Otherwise, the edges in R_{incre} are indeed connected as explained below.

Definition of the triangles that the edges in R_{incre} participating in. Since the prerequisite for the ith edge being preserved in R_{incre}, the ith edge must form triangles with the wedges in R_{base} at the time of the arrival of the ith edge. Note that a wedge is a triangle with one of the latter's edges removed. Let R_{wedge} be the set of wedges in the base reservoir that can form triangles with edges in the incremental reservoir with f edge samples, i.e., $R_{wedge} = R^{t_1}_{wedge} \cup \cdots \cup R^{t_f}_{wedge}$, where $R^{t_j}_{wedge}$ ($0 \le j \le f$) is the set of the wedges that can form triangles ($R^{t_j}_{triangle}$) with the jth edge entering R_{incre} at the time it enters, t_j. Then $R_{triangle} = R^{t_1}_{triangle} \cup \cdots \cup R^{t_f}_{triangle}$ ($|R_{triangle}| \ge f$) is the set of all triangles formed by wedges in R_{wedge} with edges in R_{incre}.

Three relationships based that the edges in R_{incre} are disconnected. We observe that for any pair of disconnected edges in R_{incre}, there are exactly three possible relationships between their corresponding wedges in R_{wedge}, as shown in Fig. 4, namely, sharing no common nodes (Fig. 4(a)) (no repetitive nodes among the nodes forming the triangles in $R_{triangle}$), sharing one common node (Fig. 4(b)) (no repetitive edges among the edges forming the triangles in $R_{triangle}$), and sharing one common edge (Fig. 4(c)) (one edge shared by two triangles in $R_{triangle}$ given the way that triangles form).

Zero-probability that the edges in R_{incre} are disconnected. However, the probability of any triangle pair in $R_{triangle}$ satisfying any of the three relationships, or equivalently, any edge pair in R_{incre} being disconnected, is almost zero (zero-probability). The analysis of the probability of the triangles in $R_{triangle}$ from the perspective of edges to satisfy the relationships of (b) and (c) in Fig. 4 is similar to that from the perspective of nodes to satisfy the relationship in 4(a). Therefore, for simplicity, we take Fig. 4(a) for example to explain the zero-probability of the edges in R_{incre} being disconnected.

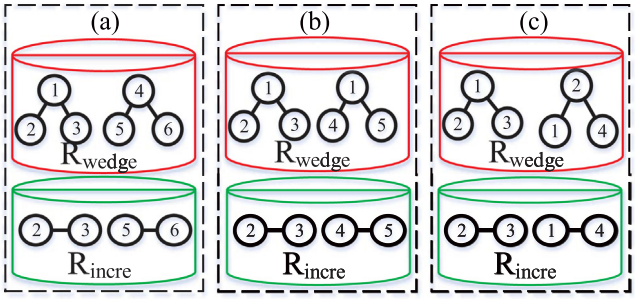


Fig. 4. Three distinct edge-wedge relationships for triangles formed by any pair of disconnected edges in R_{incre} and their corresponding wedges in R_{wedge}. If all the triangles in $R_{triangle}$ formed between R_{incre} and R_{wedge} satisfy any of these three edge-wedge relationships, the edges in the incremental reservoir are disconnected.

Specifically, suppose there are M triangles that node μ participates in the original graph stream, referred to as $\mu's$ triangles. Given the definition of the relationship shown in Fig. 4(a), only one of $\mu's$ triangle satisfying this particular relationship (sharing no nodes between any triangle pair) can be in $R_{triangle}$. However, for $\mu's$ M triangles, there are $C_M^0 + \cdots + C_M^M = 2^M$ groups of triangles that can be in $R_{triangle}$. Due to the arbitrary order of edges in a graph stream, the probability p_{each} of any triangle being in $R_{triangle}$ can be considered the same. The probability $p_{(one)}^{\mu}$ of only one of $\mu's$ triangles in $R_{triangle}$ to satisfy the relationship of Fig. 4(a) is given as,

$$p_{(one)}^{\mu} = \frac{C_M^1}{2^M} \times p_{each} \times (1 - p_{each})^{M-1}. \tag{18}$$

From Eq. (18), the larger M means the smaller $p_{(one)}^{\mu}$. Suppose μ_i has the minimum number (M_i) of triangles in the original graph stream among the three nodes μ_i, ν_i and α_i forming the ith triangle $\triangle(\mu_i, \nu_i, \alpha_i)$ $(0 < i \leq |R_{triangle}|)$ in $R_{triangle}$. Thus, the probability $P(\triangle)_{all}$ of all the triangles in $R_{triangle}$ whose formations satisfy the relationship of Fig. 4(a), is expressed as,

$$P(\triangle)_{all} < \prod_{i=1}^{i=|R_{triangle}|} (p_{(one)}^{\mu_i}), \tag{19}$$

where the meaning of $p_{(one)}^{\mu_i}$ is similar to that of $p_{(one)}^{\mu}$. As shown in Eq. (18), $p_{(one)}^{\mu_i}$ itself is a small value. For example, suppose $M_i = 1$ for each i $(0 < i \leq |R_{triangle}|)$, $p_{each} = 1$ and $|R_{incre}| = 10$, then $|R_{triangle}| \geq 10$, $p_{(one)}^{\mu_i} = 0.5$ and $P(\triangle)_{all} < 0.5^{10} \simeq 0$. Furthermore, in an actual sampling process with T-Sample over a graph stream representing a real-world application scenario as described in Section 1 and evaluated in Section 5, $M_i \geq 1$, $p_{each} \ll 1$ and $|R_{incre}| \gg 10$ that means $P(\triangle)_{all}$ is very close to zero. Therefore, the edge samples produced by T-Sample based on the design of Section 3 are certain to be connected.

5. Evaluation

This section presents the evaluation of T-Sample based on extensive data-driven simulations. The objective of this evaluation is to assess T-Sample's efficacy on its connectivity estimations in terms of triangle counts and node degrees (types and counts), in comparison to baseline methods that represent the state of the art.

Platform and Workload. The simulations are conducted on a computer with Xeon(R) E5-2620 CPU which is running at 2.10 GHz and has 8 cores, 4 G memory space and 64-bit Ubuntu Linux OS. Each experiment, which employs a single core with at most 4 GB of RAM, entails 20 runs of the simulation so that the results reported are statistically stable and meaningful. T-Sample can be applied in many applications represented by graph streams and we choose the most frequently used graphs to evaluate the sampling method. The workload traces, summarized in Table 3 and downloadable from [29] and [30] include six public real-world graph datasets (graph streams) of which two each contain more than a billion edges.

Table 3
Summary of graph datasets, where $|V|$, $|E|$ and $\triangle_{total}$ denote the total numbers of nodes, edges and triangles in a graph stream $G = (V, E)$, respectively.

Graph	$\|V\|$	$\|E\|$	$\triangle_{total}$
DBLP	317,080	1,049,866	2,224,385
Youtube	1,134,890	2,987,624	3,056,386
Live-journal	4,847,571	68,475,391	285,730,264
Orkut	3,072,441	117,184,899	627,577,371
Twitter	41,652,230	1,468,365,182	34,824,916,864
Friendster	65,608,366	1,806,067,135	4,173,724,142

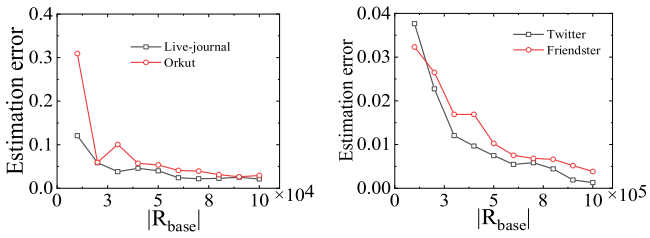
Baseline methods. From consideration of the low cost and high accuracy of the existing sampling methods as described in Table 1, the following state-of-the-art reservoir-based sampling methods, GPS Post-Stream (GPS-Post) [8], Triest-IMPR [13], GSH [9] and PIES [24], are chose as the evaluation baselines for T-Sample. Although In-Stream (GPS-In), proposed in [8], shows smaller estimation errors and variances than GPS-Post, it consumes much more time than GPS-Post and thus is not selected as a baseline. In GPS-Post, the weight of a newly arrived edge is set as the number of the triangles formed by it and the edges preserved in the reservoir thus far. This setting causes relatively small estimation variances and errors for triangle counting, as shown in [8]. All these sampling schemes are implemented in C++.

Capacity of the reservoir. As described in Section 3, for T-Sample, the edge samples preserved in the base reservoir are used to estimate triangle counts while those preserved in both the base and incremental reservoirs are used to obtain information on node degrees. However, for all the baseline methods, only one reservoir is used to preserve edge samples. Therefore, when estimating the triangle counts, the baseline methods set the capacity of the reservoir equal to that of T-Sample's base reservoir, i.e., $|R_{baseline}| = |R_{base}|$. On the other hand, when the baseline schemes are used to estimate the node degrees, there are two cases for comparison with T-Sample in terms of estimation accuracy, time and memory costs. In the first case, the capacity of the reservoir for the baseline sampling schemes is set to be the same as that of T-Sample's base reservoir, which means that T-Sample will use more total memory capacity to obtain information about node degrees. The second case sets the reservoir capacity for the baseline schemes to be the sum of those for the base and incremental reservoirs of T-Sample, $|R_{baseline}| = |R_{total}| = |R_{base}| + |R_{incre}|$.

5.1. Estimations on triangle counts

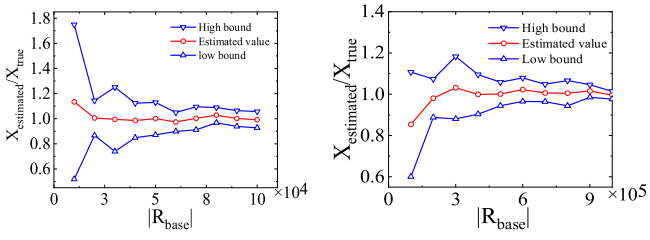
Simulation results on triangle counts are evaluated in two metrics, estimation error and estimation variance.

1. ***Estimation error.*** Let X_{true} denote the ground truth of a property X, X_t be the tth estimated value of X and X_T the mean estimated value of X after T simulation runs. The estimation error of the property X after T simulation runs is measured as: $Err(X_T) = \frac{1}{T}\sum_{t=1}^{T}(\frac{|X_{true} - X_t|}{X_{true}})$. T is set at 20 in this paper. The ground-truth values (X_{true}) of the triangle counts over the five graph streams used in the evaluation are given in Table 3 in the $\triangle_{total}$ column and can also be obtained from Eq. (11) of Section 4.



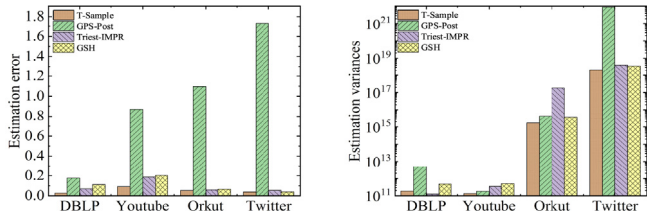
(a) Live-journal & Orkut (b) Twitter & Friendster

Fig. 5. T-Sample's estimation errors, $Err(X_{20})$, over four different graph streams as a function of base reservoir capacity (in terms of the number of edge samples).



(a) Orkut (b) Friendster

Fig. 6. T-Sample's confidence bounds over Orkut and Friendster as a function of base reservoir capacity (in terms of the number of edge samples) with $X_{estimated} = X_{20}(\pm 1.96 \times \sqrt{Var(X_{20})})$.

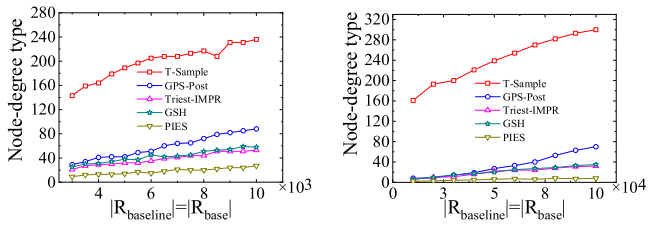


(a) Estimation errors (b) Estimation variances

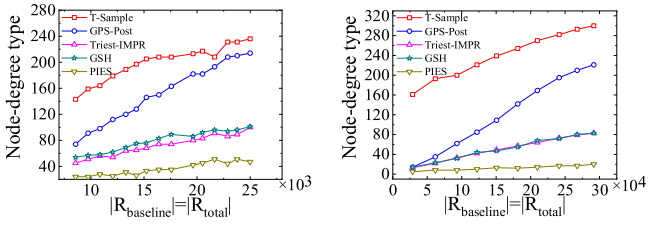
Fig. 7. The mean estimation errors and variances when these sampling methods are used to estimate the triangle counts with $|R_{baseline}| = |R_{base}| = \{3 \times 10^3, 5 \times 10^3, 3 \times 10^4, 10^5\}$ respectively over the four datasets.

2. ***Estimation variance*** is used to evaluate the confidence of the estimation results. The estimation variance of X_T (labeled as $Var(X_T)$) is $Var(X_T) = \sum_{t=1}^{T}(\frac{(X_t - X_T)^2}{T})$. As described in [31], the high confidence bound of the estimation value is computed as $X_T + 1.96 \times \sqrt{Var(X_T)}$ while the low confidence bound is $X_T - 1.96 \times \sqrt{Var(X_T)}$.

For ease of describing the evaluation results among different sampling methods, we simply use the name of a sampling method to stand for a specific algorithm of counting the triangles. For example, T-Sample uses the algorithm of TS-Triangle to count the triangles, and thus we use the notation of T-Sample to describe the experimental results of TS-Triangle. Figs. 5(a) and 5(b) show that T-Sample estimates the total number of triangles with small estimation errors ranging from 0.40 to 0.029, which decrease with an increasing base reservoir capacity, over the four different graph streams. Furthermore, Figs. 6(a) and 6(b) show that the confidence intervals are small. For example, Fig. 6(b) shows that the high confidence ranges from 1.01 to 1.98 while the low confidence ranges from 0.6 to 0.97 over Friendster. Importantly, both the high and the low confidence intervals are decreasing with an increasing base reservoir capacity over Orkut and Friendster.



(a) Youtube (b) Orkut



(c) Youtube (d) Orkut

Fig. 8. The node-degree types inferred by edge samples as a function of the based reservoir capacity (in terms of the number of edge samples), over Youtube and Orkut with $|R_{baseline}| = |R_{base}|$ and $|R_{baseline}| = |R_{total}|$ respectively.

Comparisons with the baselines. As explained in Section 2, since PIES cannot be used to estimate the triangle counts, the baseline methods for T-Sample include GPS-Post, Triest-IMPR and GSH. Fig. 7(a) shows that T-Sample can estimate the counts of triangles with 71.4%, 51%, 9.6% and 2% more accuracies than the best performance of the baseline methods over DBLP, Youtube, Orkut and Twitter. Due to the small size of DBLP and the limited number of sampling times, Fig. 7(b) shows T-Sample just exhibits 56.7% larger estimation variance than Triest-IMPR. Furthermore, Fig. 7(b) shows that T-Sample has 27.6%, 56.4% and 40.6% respectively smaller variances than the method that performs the best among GPS-Post, Triest-IMPR and GSH over the other three datasets while T-Sample has 2.8, 105 and 4110 times respectively more accuracies than the worst performance of the baseline methods.

5.2. Estimations on node degrees

In this paper, we evaluate node degrees in terms of node-degree types and node-degree counts described as follows.

Node-degree types refer to the different node degrees (i.e., type i means node degree of i) of a graph stream inferred by the sampled edges and its inference accuracy is measured by the *node coverage*, i.e., the total number of nodes of all degree types inferred by the edge samples in the original graph stream (G) divided by the total number of the nodes in G. Note the number in the nominator of this division can be determined from the graph dataset given all the inferred node-degree types, and the denominator is directly given in the dataset. These measures are collectively used to indicate how closely can the edge samples infer specific and local connectivity. For example, with $|R_{baseline}| = |R_{base}| = 7000$, Figs. 8(a) and 8(b) show that T-Sample obtains at least 2.45 and 6 times more node types than the baseline methods over Youtube and Orkut. Furthermore, given a fixed capacity of the base reservoir, Figs. 8(a) and 8(b) show that T-Sample obtains on average 12 times more node-degree types than the four baseline methods over Youtube and Orkut when $|R_{baseline}| = |R_{base}|$. Figs. 8(c) and 8(d) further show that T-Sample still produces on average 5.6 times more node-degree types than the four baseline methods over Youtube and Orkut when these methods consume

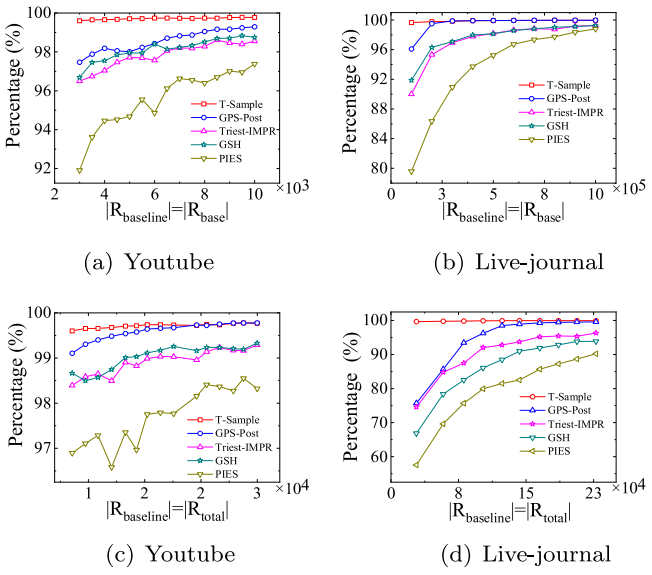


Fig. 9. The percentage coverage of node-degree types inferred by edge samples as a function of the base reservoir capacity (in terms of the number of edge samples), over Youtube and Live-journal with $|R_{baseline}| = |R_{base}|$ and $|R_{baseline}| = |R_{total}|$ respectively.

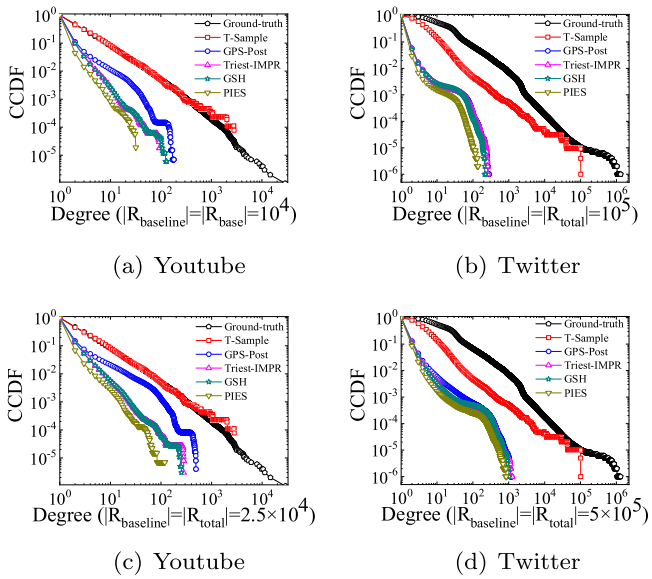


Fig. 10. The distributions of node-degree counts over Youtube and Twitter with $|R_{baseline}| = |R_{base}|$ and $|R_{baseline}| = |R_{total}|$ respectively. Note that each data point (x, y) in the figures indicates that $100 \times y\%$ of nodes are of degree equal to or smaller than x.

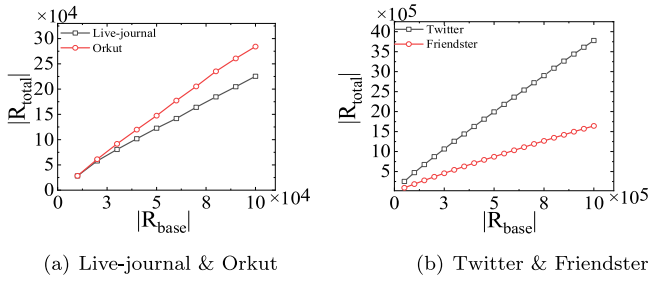


Fig. 11. The total reservoir capacities (in terms of the number of edge samples) of T-Sample as a function of the base reservoir capacity for the four graph streams.

the same total memory as T-Sample ($|R_{baseline}| = |R_{total}|$). For example, $|R_{baseline}| = |R_{total}| = 16400$, T-Sample can obtain 0.41 times more node types than GPS-Post, Triest-IMPR, GSH and PIES over Youtube.

Furthermore, Fig. 9 shows that T-Sample can cover a much higher percentage of nodes whose connectivity can be inferred by edge samples from the perspective of node-degree types over Youtube and Orkut, whether $|R_{baseline}| = |R_{base}|$ or $|R_{baseline}| = |R_{total}|$. For example, Figs. 9(c) and 9(d) show that the percentage of nodes covered by all node-degree types obtained by T-Sample is more than 98% of all nodes of the original graph stream by using a base reservoir whose capacity is only a tiny fraction of the volume of the original graph stream (i.e., 0.1% of the total volumes over Youtube and 0.03% over Live-journal). With the increase of the sample size over the two datasets, the four baseline methods can produce higher node coverages of node-degree types but still at levels lower than T-Sample, as shown in 9(c) and Fig. 9(d). Besides, GPS-Post shows the best node coverages of node-degree types among the four baseline methods because it considers the connectivity during the sampling process as described in Section 2.

Node-degree counts refer to the numbers of nodes with different node degrees and are measured by the distributions of node-degree types inferred by edge samples (or the ground truth from the dataset) among the nodes in a graph stream. This measure indicates how closely the node-degree counts inferred by edge samples reflect the ground truth with a very small sample set. Fig. 10 shows that, whether $|R_{baseline}| = |R_{base}|$ (10^4 in Youtube and 10^5 in twitter) or $|R_{baseline}| = |R_{total}|$ (2.5×10^4 in Youtube and 5×10^5 in twitter), T-Sample obtains the node-degree counts that are much closer to the ground-truth values than the four baseline methods, as measured in the complementary cumulative distribution function (CCDF), over YouTube and Twitter. For example, Fig. 10(a) shows that the respective distributions with *degree* > 20, the ground-truth, T-Sample, GPS-Post, Triest-IMPR, GSH and PIES are 0.03853, 0.0378, 0.00581, 0.00051, 0.000568 and 0.00027. Figs. 10(b) and 10(d) also show T-Sample can obtain the degree distribution closer to the ground-truth value than the baseline methods. Furthermore, the node-degree counts distributions obtained by the baseline methods do not change significantly with the increase of the sample size, as shown in Fig. 10, because these methods are not able to produce connected edge samples.

5.3. Memory costs

Fig. 11 shows the total reservoir capacities used by T-Sample for the four different graph streams as a function of the base reservoir capacity, suggesting that the higher the capacity of the base reservoir, the more memory usage T-Sample consumes. On the other hand, the total capacity is also affected by the specific connectivity of a graph stream. For example, Fig. 11(b) shows that the Twitter graph stream consumes more memory than the Friendster graph stream while the total volumes of the former is smaller than that of the latter, because the former is much more densely connected than the latter as reflected by their triangle counts.

5.4. Processing time

Fig. 12 shows that GPS-Post consistently consumes much more time than the other methods over Orkut because it employs the time-consuming weighted setting (Section 2). Furthermore, GPS-Post consumes more time than that reported in [8] because

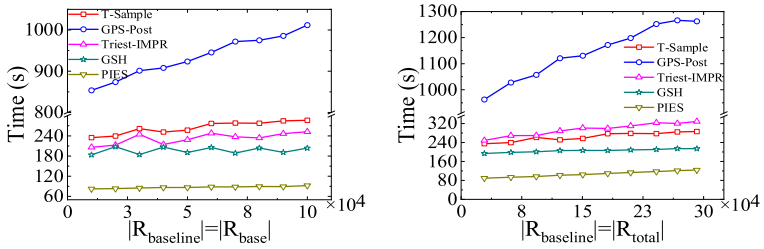


Fig. 12. The processing time over Orkut with $|R_{baseline}| = |R_{base}|$ and $|R_{baseline}| = |R_{total}|$ respectively.

the experiments in [8] used much more CPUs (i.e., 16 cores of two Intel Xeon E5-2687 W 3.1 GHz CPUs) than those in this paper (i.e., one core of one Intel(R) Xeon(R) CPU). Besides, GSH's processing time is slightly less than T-Sample and Triest-IMPR, while it causes larger estimation errors and variances on triangle counts as shown in Fig. 7. Because PIES only preserves the edge samples without estimating triangle counts, it thus consumes the least amount of time among all methods. For the same capacity of base reservoir, Fig. 12(a) shows that T-Sample consumes slightly more time than Triest-IMPR, Fig. 12(b) shows the exact opposite. Considering the superior estimation results of T-Sample on both node degrees and triangle counts, T-Sample is much more cost-effective than the baseline methods. For example, T-Sample obtains at least 7 times more node-degree types with $|R_{base}| = 10^4$ and $|R_{total}| = 3 \times 10^4$ than PIES with $|R_{baseline}| = \{10^5, 2.9 \times 10^5\}$ respectively while it costs at most 1.5X more time than PIES.

According to the energy expression $Q = P \times T$, where Q refers to the consumed energy, P refers to the power rating of the used machines and T is the time cost of the evaluated algorithm, the more processing time means the more energy consumed by the algorithm. Therefore, from Fig. 12, we can infer that GPS-Post consistently consumes much more energy than the other methods while PIES consumes the least energy among the five methods and the remaining three methods consumes similar energy. From consideration the estimation accuracy, T-Sample performs better than the baseline methods as T-Sample obtains 7 times more node-degree types at the cost of 1.5X more energy than PIES.

6. Conclusions

In this paper, we make the same assumption as that by many existing reservoir-based sampling methods [8–10,13,22], that each edge in a graph stream is stored for only one time and the deleted edges are no longer stored to reduce the storage overhead. This is a reasonable assumption, particularly for graph streams used to express the current relationships among the users of social networks, the molecules in bioinformatics and the nodes in large computer networks where the volumes of the edges are always increasing while the relatively insignificant number of deleted edges will not likely change the connectivity of the graph streams fundamentally. Thus, when the data in applications (i.e., online social networks, attributed networks, computer networks and biological networks) is preserved in the form of edge streams [32], the reservoir-based sampling methods can be employed to characterize the applications. On the other hand, when the graphs are organized in the form of nodes and their respective neighbors, other sampling methods oriented towards nodes will be more effective to estimate the properties of graphs.

Furthermore, we propose a new reservoir-based sampling method, called triangle-induced sampling or T-Sample. Significantly different from existing reservoir-based sampling methods that produce rarely connected edge samples, T-Sample is a first attempt at counting the triangles accurately by inferring the probability of the triangles formed by the edge samples of the graph stream precisely while simultaneously inferring information about the node degrees with a limited memory usage by producing connected edge samples. Extensive dataset-driven experimental results show that T-Sample can estimate the triangle counts even 50% more accurate than the baseline methods and meanwhile can obtain more than 90% node-degree types at smaller time and memory costs.

7. Future work

Graph sampling is a broad concept and one specific sampling technique cannot characterize all the properties of graphs in any forms. Therefore, many researches in the field of graph sampling can be further proceeded. From the perspective of graph storages, this paper aims at graphs in the form of streams to count triangles and estimate node degrees. In the future, we would pay attention to developing sampling techniques to characterize other forms of graphs, such as online social network and attributed networks are preserved in adjacency lists. From the perspective of estimation goal, this paper is to characterize edge connectivity in terms of triangle counts and node degrees, other forms of connectivity, such as motif estimation, link prediction and so on would be studied. Furthermore, from the perspective of real applications, T-Sample can perform well in the graph streams which may add connectivity during the sampling process. However, the data in a large graph can be added or deleted, for example, the friendship in social networks can be added or removed, and we would design sampling techniques to deal with fully dynamic graphs.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

We would like to thank the anonymous reviewers of this paper. This work is supported by National Defense Preliminary Research, China Project (31511010202) and NSFC, China No. 61832020, 61772216 and No. 61821003. This work is also funded from Science, Technology and Innovation Commission of Shenzhen Municipality, China (JCYJ20170307172248636) and Hubei province technical innovation special project, China (2017AAA129) Fundamental Research Funds for the Central Universities, China.

References

[1] A. McGregor, Graph stream algorithms: a survey, ACM SIGMOD.
[2] P. Zhao, C. Aggarwal, G. He, Link prediction in graph streams, in: IEEE ICDE, 2016, pp. 553–564.
[3] L.S. Buriol, G. Frahling, S. Leonardi, A. Marchetti-Spaccamela, C. Sohler, Counting triangles in data streams, in: ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, 2006, pp. 253–262.
[4] P. Wang, J.C. Lui, D. Towsley, J. Zhao, Minfer: A method of inferring motif statistics from sampled edges, in: IEEE ICDE, 2016.
[5] U. Kang, B. Meeder, E.E. Papalexakis, C. Faloutsos, Heigen: Spectral analysis for billion-scale graphs, IEEE Trans. Knowl. Data Eng. (TKDE) 26 (2) (2014) 350–362.
[6] J. Leskovec, D. Huttenlocher, J. Kleinberg, Predicting positive and negative links in online social networks, in: ACM WWW, 2010.
[7] X. Lu, S. Bressan, Sampling connected induced subgraphs uniformly at random, in: International Conference on Scientific and Statistical Database Management, Springer, 2012, pp. 195–212.
[8] N.K. Ahmed, N. Duffield, T.L. Willke, R.A. Rossi, On sampling from massive graph streams, VLDB 10 (11).
[9] N.K. Ahmed, N. Duffield, J. Neville, R. Kompella, Graph sample and hold: A framework for big-graph analytics, in: ACM KDD, 2014.
[10] Y. Lim, U. Kang, Mascot: Memory-efficient and accurate sampling for counting local triangles in graph streams, in: ACM KDD, 2015, pp. 685–694.
[11] B. Wu, K. Yi, Z. Li, Counting triangles in large graphs by random sampling, IEEE Trans. Knowl. Data Eng. (TKDE) 28 (8) (2016) 2013–2026.
[12] P. Wang, Y. Qi, Y. Sun, X. Zhang, J. Tao, X. Guan, Approximately counting triangles in large graph streams including edge duplicates with a fixed memory usage, VLDB 11 (2).
[13] L.D. Stefani, A. Epasto, M. Riondato, E. Upfal, Triest: Counting local and global triangles in fully dynamic streams with fixed memory size, ACM Trans. Knowl. Discov. Data (TKDD) 11 (4) (2017) 43.
[14] L. De Stefani, A. Epasto, M. Riondato, E. Upfal, Trièst: Counting local and global triangles in fully-dynamic streams with fixed memory size, in: ACM KDD, 2016, pp. 825–834.
[15] R.-H. Li, J.X. Yu, X. Huang, H. Cheng, Random-walk domination in large graphs, in: ICDE, IEEE, 2014, pp. 736–747.
[16] F. Chiericetti, A. Dasgupta, R. Kumar, S. Lattanzi, T. Sarlós, On sampling nodes in a network, in: Proceedings of the 25th International Conference on World Wide Web, International World Wide Web Conferences Steering Committee, 2016, pp. 471–481.
[17] B. Ribeiro, D. Towsley, Estimating and sampling graphs with multidimensional random walks, in: SIGCOMM, ACM, 2010, pp. 390–403.
[18] B. Ribeiro, P. Wang, F. Murai, D. Towsley, Sampling directed graphs with random walks, in: INFOCOM, IEEE, 2012, pp. 1692–1700.
[19] Y. Murase, H.-H. Jo, J. Török, J. Kertész, K. Kaski, Sampling networks by nodal attributes, Phys. Rev. E 99 (2019) 052304, http://dx.doi.org/10.1103/PhysRevE.99.052304, URL https://link.aps.org/doi/10.1103/PhysRevE.99.052304.
[20] A. Rezvanian, B. Moradabadi, M. Ghavipour, M.M.D. Khomami, M.R. Meybodi, Social network sampling, in: Learning Automata Approach for Social Networks, Springer, 2019, pp. 91–149.
[21] Y. Xie, S. Chang, Z. Zhang, M. Zhang, L. Yang, Efficient sampling of complex network with modified random walk strategies, Physica A 492 (2018) 57–64.
[22] A. Pavan, K. Tangwongsan, S. Tirthapura, K.-L. Wu, Counting and sampling triangles from a graph stream, VLDB 6 (14).
[23] M. Jha, C. Seshadhri, A. Pinar, A space-efficient streaming algorithm for estimating transitivity and triangle counts using the birthday paradox, ACM Trans. Knowl. Discov. Data (TKDD) 9 (3) (2015) 15.
[24] N.K. Ahmed, J. Neville, R. Kompella, Network sampling: From static to streaming graphs, ACM Trans. Knowl. Discov. Data (TKDD) 8 (2) (2014) 7.
[25] Y. Han, J. Tang, Probabilistic community and role model for social networks, in: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2015, pp. 407–416.
[26] J. Zhang, J. Tang, Y. Zhong, Y. Mo, J. Li, G. Song, W. Hall, J. Sun, Structinf: Mining structural influence from social streams, in: Thirty-First AAAI Conference on Artificial Intelligence, 2017.
[27] Y. Li, J. Fan, Y. Wang, K.-L. Tan, Influence maximization on social graphs: A survey, IEEE Trans. Knowl. Data Eng. 30 (10) (2018) 1852–1872.
[28] J.S. Vitter, Random sampling with a reservoir, ACM Trans. Math. Softw. (TOMS) 11 (1) (1985) 37–57.
[29] Snap datasets, http://snap.stanford.edu/.
[30] Konect datasets, http://konect.uni-koblenz.de/.
[31] Y. Tillé, Sampling algorithms, in: International Encyclopedia of Statistical Science, Springer, 2011, pp. 1273–1274.
[32] M. Ghavipour, M.R. Meybodi, A streaming sampling algorithm for social activity networks using fixed structure learning automata, Appl. Intell. 48 (4) (2018) 1054–1081.

Lingling Zhang is currently a Ph.D. student majoring in Computer Architecture in Huazhong University of Science and Technology (HUST), Wuhan, China. Her current research interests include processing of big data and computer networks.

Hong Jiang received the B.Sc. degree in Computer Engineering in 1982 from Huazhong University of Science and Technology, China; the M.A.Sc. degree in Computer Engineering in 1987 from the University of Toronto, Canada; and the PhD degree in Computer Science in 1991 from the Texas A&M University, USA. He is currently Chair and Wendell H. Nedderman Endowed Professor of Computer Science and Engineering Department at the University of Texas at Arlington. Prior to joining UTA, he served as a Program Director at National Science Foundation (2013.12015.8) and he was at University of Nebraska-Lincoln since 1991, where he was Willa Cather Professor of Computer Science and Engineering. His present research interests include computer architecture, computer storage systems and parallel I/O, high performance computing, big data computing, cloud computing, performance evaluation. He recently served as an Associate Editor of the IEEE Transactions on Parallel and Distributed Systems. He has over 200 publications in major journals and international Conferences in these areas, including IEEE-TPDS, IEEE-TC, Proceedings of the IEEE, ACM-TACO, JPDC, ISCA, MICRO, USENIX ATC, FAST, EUROSYS, LISA, SIGMETRICS, ICDCS, IPDPS, MIDDLEWARE, OOPLAS, ECOOP, SC, ICS, HPDC, INFOCOM, ICPP, etc., and his research has been supported by NSF, DOD, the State of Texas and the State of Nebraska. Dr. Jiang is a Fellow of IEEE, Member of ACM and USENIX.

Fang Wang received her BE degree and Master degree in computer science in 1994, 1997, and Ph.D. degree in computer architecture in 2001 from Huazhong University of Science and Technology (HUST), China. She is a professor of computer science and engineering at HUST. Her interests include distribute file systems, parallel I/O storage systems and graph processing systems. She has more than 50 publications in major journals and international conferences, including FGCS, ACM TACO, SCIENCE CHINA Information Sciences, Chinese Journal of Computers and HiPC, ICDCS, HPDC, ICPP.

Dan Feng received the BE, ME, and Ph.D. degrees in Computer Science and Technology in 1991, 1994, and 1997, respectively, from Huazhong University of Science and Technology (HUST), China. She is a professor and vice dean of the School of Computer Science and Technology, HUST. Her research interests include computer architecture, massive storage systems, and parallel file systems. She has more than 100 publications in major journals and international conferences, including IEEE-TC, IEEE-TPDS, ACM-TOS, JCST, FAST, USENIX ATC, ICDCS, HPDC, SC, ICS, IPDPS, and ICPP. She serves on the program committees of multiple international conferences, including SC 2011, 2013 and MSST 2012. She is a member of IEEE and a member of ACM.

Yanwen Xie is currently a Ph.D. student majoring in Computer Architecture in Wuhan National Laboratory for Optoelectronics, Huazhong University of Science and Technology. His current research interests include erasure coding, distributed storage systems and disk failure prediction.