Knowledge and Information Systems (2021) 63:939–969
https://doi.org/10.1007/s10115-021-01543-x

REGULAR PAPER

Tracking triadic cardinality distributions for burst detection in high-speed graph streams

Junzhou Zhao[1] · Pinghui Wang[1,2] · Zhouguo Chen[3] · Jianwei Ding[3] · John C. S. Lui[4] · Don Towsley[5] · Xiaohong Guan[1,2]

Received: 19 August 2017 / Revised: 30 December 2020 / Accepted: 30 December 2020 / Published online: 18 January 2021

Abstract
In everyday life, we often observe unusually frequent interactions among people before or during important events, e.g., people send/receive more greetings to/from their friends on holidays than regular days. We also observe that some videos or hashtags suddenly go viral through people's sharing on online social networks (OSNs). Do these seemingly different phenomena share a common structure? All these phenomena are associated with the sudden surges of node interactions in networks, which we call "*bursts*" in this work. We uncover that, in many scenarios, the emergence of a burst is accompanied with the formation of triangles in networks. This finding motivates us to propose a new and robust method for burst detection on an OSN. We first introduce a new measure, i.e., "*triadic cardinality distribution,*" corresponding to the fractions of nodes with different numbers of triangles, i.e., triadic cardinalities, in a network. We show that this distribution not only changes when a burst occurs, but it also has a robustness property that it is immunized against common spamming social-bot attacks. Hence, by tracking triadic cardinality distributions, we can more reliably detect bursts than simply counting node interactions on an OSN. To avoid handling massive activity data generated by OSN users during the triadic tracking, we design an efficient "*sample-estimate*" framework to provide maximum likelihood estimate of the triadic cardinality distribution. We propose several sampling methods and provide insights into their performance difference through both theoretical analysis and empirical experiments on real-world networks.

✉ Pinghui Wang
phwang@mail.xjtu.edu.cn

[1] MOE Key Lab for Intelligent Networks and Network Security, Xi'an Jiaotong University, Xi'an 710049, People's Republic of China

[2] Shenzhen Research Institute of Xi'an Jiaotong University, Shenzhen 518057, People's Republic of China

[3] Science and Technology on Communication Security Laboratory, 30th Research Institute of China Electronics Technology Group Corporation, Chengdu 610041, People's Republic of China

[4] Department of Computer Science and Engineering, The Chinese University of Hong Kong, Shatin, Hong Kong, People's Republic of China

[5] School of Computer Science, University of Massachusetts at Amherst, Amherst, MA 01003, USA

Keywords Burst detection · Graph sampling · Streaming algorithms · Online social network · Estimation theory

1 Introduction

Online social networks (OSNs) have become ubiquitous platforms that provide various ways for people to interact over the internet such as (re-)tweeting tweets, sharing links, messaging friends, commenting on posts, and mentioning/replying to other users (i.e., @someone). When intense user interactions take place in a short time period, there will be a surge in the volume of user activities in an OSN. Such a surge of user activity, which we call a "*burst*" in this work, usually relates to emergent events that are occurring or about to occur in the real world. For example, Michael Jackson's death on June 25, 2009, triggered a global outpouring of grief on Twitter [22], and the event even crashed Twitter for several minutes [45]. In addition to bursts caused by real-world events, some bursts arising from OSNs can also cause enormous social impact in the real world. For example, the 2011 England riots, in which people used OSNs to organize themselves, resulted in 3443 crimes across London due to this disorder [31]. Hence, detecting bursts in OSNs is an important task, both for OSN managers to monitor the operation status of an OSN, as well as for government agencies to anticipate any emergent social disorder.

Typically, there are two types of user interactions in OSNs. First is the interaction between users (we refer to this as *user–user interaction*), e.g., a user sends a message to another user, while the second is the interaction between a user and a media content piece (we refer to this as *user–content interaction*), e.g., a user (re-)posts a video link. Examples of bursts caused by these two types of interactions include, many greetings being sent/received among people on Christmas Day, and videos suddenly becoming viral after one day of sharing in an OSN. At first sight, detecting such bursts in an OSN is not difficult. For example, a naive way to detect bursts caused by user–user interactions is to *count* the number of pairwise user interactions within a time window, and report a burst if the volume lies above a given threshold. However, this method is vulnerable to spamming social-bot attacks [7,8,12,21,48,51], which can suddenly generate a huge amount of spamming interactions in the OSN. Hence, this method may result in many *false alarms* due to the existence of social bots. Similar problem also exists when detecting bursts caused by user–content interactions. Many previous works on burst detection are based on idealized assumptions [17,24,37,62] and simply ignore the existence of social bots.

The primary goal of this work is to leverage a special *triangle structure*, which is a feature of human interaction and behavior, to design a robust burst detection method that is immune against common social-bot attacks. We first describe the triangle structure shared by both types of user interactions.

Interaction triangles in user–user interactions Humans form social networks with larger clustering coefficients than those in random networks [60] because social networks exhibit many *triadic closures* [26]. This is due to the social phenomenon "*friends of my friends are also my friends.*" Since user–user interactions usually take place along social links, this property implies that user–user interactions should also exhibit many triadic closures (which we will verify in later experiments). In other words, when a group of users suddenly become active, or we say an *interaction burst* occurs, in addition to observing the rise of volume of pairwise interactions, we expect to also observe many interactions among three neighboring users, i.e., many *interaction triangles* form if we consider an edge of an interaction triangle

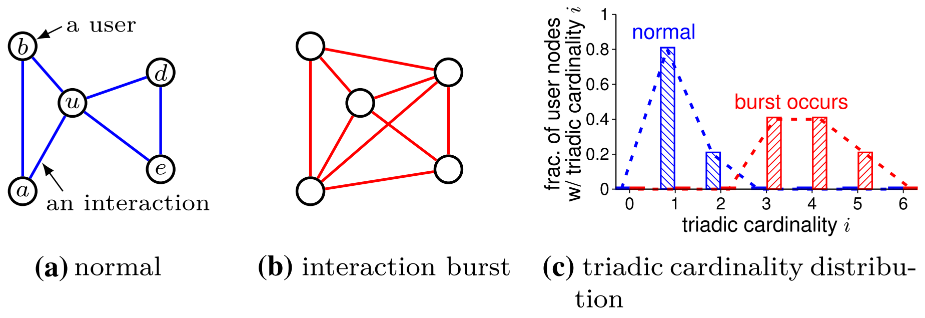


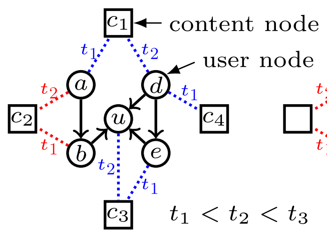
Fig. 1 Interaction triangles and interaction burst. Edges in **a** and **b** represent interactions among users

to be a user–user interaction. This is illustrated in Fig. 1a when no interaction burst occurs, and in Fig. 1b, an interaction burst occurs. In contrast, activities generated by social bots do not possess many triangles since social bots typically select their targets randomly from an OSN [8,51].

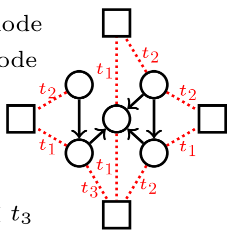
Influence triangles in user–content interactions Similar triangle structure can also be observed in bursts caused by user–content interactions. We say that a media content piece becomes *bursty* if many users interact with it in a short time period. There are many reasons why a user interacts with a piece of media content. Here, we are particularly interested in the case where one user *influences* another user to interact with the content, aka the cascading diffusion [28] or word-of-mouth spreading [40]. It is known that many emerging news stories arising from OSNs are related to this mechanism such as the story about the killing of Osama bin Laden [53]. We find that a bursty media content piece formed by this mechanism is associated with triangle formations in a network. To illustrate this, consider Fig. 2a, in which there are five user nodes $\{a, b, d, e, u\}$ and four content nodes $\{c_1, c_2, c_3, c_4\}$. A directed edge between two users means that one follows another, and an undirected edge labeled with a timestamp between a user node and a content node represents an interaction between the user and the content at the labeled time. We say content node c has an *influence triangle* if there exist two users a, b such that a follows b and a interacts with c *later* than b does. In other words, the reason that a interacts with c is due to the influence of b on a. In Fig. 2a, only c_2 has an influence triangle, the others have no influence triangle, meaning that the majority of user–content interactions are not due to influence. In Fig. 2b, every content node is part of at least one influence triangle, meaning that many content pieces are spreading in a cascading manner in the OSN. From the perspective of an OSN manager who wants to know the operation status of the OSN, if the OSN suddenly switches to a state similar to Fig. 2a (from a previous state similar to Fig. 2a), he knows that a *cascading burst* is present on the network.

Characterizing bursts So far, we find a common structure shared by different types of bursts: the emergence of *interaction bursts* (caused by user–user interactions) and *cascading bursts* (caused by user–content interactions) are both accompanied with the formation of triangles, i.e., interaction or influence triangles, in appropriately defined networks. This finding motivates us to characterize patterns of bursts in an OSN by characterizing the triangle statistics of a network, which we called the *triadic cardinality distribution*.

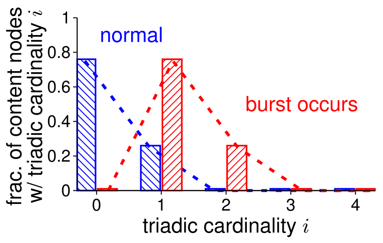
Triadic cardinality of a node in a network, e.g., a user node in Fig. 1a or a content node in Fig. 2a, is the number of triangles that it belongs to. The triadic cardinality distribution then characterizes the fractions of nodes with certain triadic cardinalities. When a burst occurs,



(a) normal



(b) cascading burst



(c) triadic cardinality distribution

Fig. 2 Influence triangles and cascading burst

because many new interaction/influence triangles are formed, we will observe that some nodes' triadic cardinalities increase, and this results in the distribution "shifting" to right, as illustrated in Figs. 1c and 2c . The triadic cardinality distribution provides succinct summary information to characterize burst patterns of a large scale OSN. Hence, by tracking triadic cardinality distributions, we can detect the presence of bursts.

In this paper, we assume that user interactions are aggregated chronologically to form a *social activity stream*, which could represent Twitter's tweets timeline stream or Facebook's news feed stream. We aim to calculate triadic cardinality distributions from this stream. The challenge is that when a network is large or users are active, the social activity stream may be of high speed. For example, the speed of the Twitter's tweets stream can be as high as 5700 tweets per second on average, 143,199 tweets per second during the peak time, and about 500 million to 12 billion tweets are aggregated per day [27]. To handle such a high-speed social activity stream, we design a sample-estimate framework, which can provide *maximum likelihood estimates* of the triadic cardinality distributions using sampled data. Our framework works in a near-real-time fashion, and is demonstrated to be accurate and efficient.

In this work, we make the following contributions:

- We discover a useful and robust measure, i.e., triadic cardinality distribution, that can be used to characterize burst patterns formed by both user–user interactions and user–content interactions in a large OSN. It has a natural robustness property and it is immunized against common spamming social-bot attacks.
- We design a unified sample-estimate framework that is able to provide maximum likelihood estimates of the triadic cardinality distribution. Under this framework, we study two types of stream sampling methods and provide insights into their performance difference through calculating the Fisher information matrices and empirical evaluations.
- We conduct extensive experiments using real-world data to demonstrate the usefulness of triadic cardinality distribution, and prove the effectiveness of our sample-estimate framework.

The remainder of this work will proceed as follows. In Sect. 2, we summarize some related work. In Sect. 3, we formally define the triadic cardinality distribution, and give an overview about our sample-estimate framework. In Sect. 4, we design two types of stream sampling methods to reduce storage complexity and improve computational efficiency. We then elaborate a maximum likelihood estimation method in Sect. 5, and obtain its Cramér–Rao lower bound in Sect. 6. We provide detailed validations of our methods in Sect. 7, including a real-world application on detecting bursts during the 2014 Hong Kong occupy central movement. Finally, Sect. 8 concludes.

2 Related work

The problem of burst detection from data streams was first studied in [24], where the author used a multistate automaton to model a stream consisting of messages such as an email stream. The occurrence of a burst is modeled by an underlying state transiting into a bursty state that emits messages at a higher rate than at the non-bursty state. Based on this model, many variant models have been proposed for detecting bursts from document streams [33,62], e-commerce query stream [37], time series [67], and social networks [17]. Although these models are theoretically interesting, some assumptions made by them are idealism such as the Poisson process of message arrivals (refer to [4] on arguing against this assumption) and nonexistence of spams/bots, which may limit their practical usage.

The topic of (anomaly) event detection is also related to our work. Chierichetti et al. [11] found that Twitter user tweeting and retweeting count information can be used to detect sub-events during some large event such as the soccer World Cup of 2010. Takahashi et al. [49] proposed a probabilistic model to detect emerging topics in Twitter by assigning an anomaly score for each user. Sakaki et al. [41] proposed a spatiotemporal model to detect earthquakes using tweets. Manzoor et al. [32] studied anomaly event detection from a graph stream based on graph similarity metrics. Different from theirs, we exploit the triangle structure existing in user interactions which is robust against common spams and can be efficiently estimated via maximum likelihood estimation.

Anomaly event detection on graphs is also extensively studied on dynamic graphs [10,30, 50,63], streaming graphs [18], and social media platforms [13]. In [63], the authors proposed AnomRank to assign node scores for detecting two types of anomalies caused by (i) addition of edges between previously unrelated nodes and (ii) increase in the number of edges between connected nodes, respectively. Note that AnomRank does not leverage the subgraph structure information when calculating node scores. In [50], the authors proposed DeepSphere that leverages deep learning methods to detect anomalies from time series data generated from edges in a dynamic graph. Their problem setting is significantly different from ours. In [10], the authors designed algorithms to solve the heaviest dynamic subgraph problem for anomalous detection. In [18], the authors define anomaly as the sudden (dis)appearance of a dense subgraph in a graph and proposed the SpotLight sketch to calculate the anomalousness of the original graph. In [30], the authors proposed a stochastic approach to find dense lasting subgraphs in dynamic graphs. Note that the goals in [10,18,30] are similar to each other, and their key problem is to count the number of edges (or sum of edge weights) in a subgraph (in addition, [10] and [30] require the subgraph to last for a period of time), while our key problem is to count the number of triangles corresponding to each node in a graph rather than counting edges. In [13], the authors aim to detect anomalies caused by social bots, and their method depends on modeling the inter-arrival times (IATs) of human communications in social media such as posting photograph, comments, and videos. Note that our goal is to detect bursts caused by humans not spamming social bots.

The triangle structure can be considered as a type of network motif. Network motif is studied in the seminal work [34] for the purpose of characterizing the structural difference of different types of networks. Turkett et al. [56] used motifs to analyze computer network usage, and [59] proposed sampling methods to efficiently estimate motif statistics in a large graph. Paranjape et al. [36] introduced δ temporal motifs and designed algorithms to count the temporal motif patterns in dynamic graphs. However, both the motivation in [36,56] and subgraph statistics defined in [59] are different from ours. Recently, a special network motif called butterfly motif is also studied on bipartite graphs [42,43] such as the graph formed by the user-product purchasing relation.

Recently, there are many works on estimating the number of global and local triangles [1,2,9,23,29,38,46,47,54,61,65], or clustering coefficient [44] in a large graph. If we apply these existing algorithms to estimate triadic cardinality distribution, we need to estimate the number of triangles for each node in the graph, i.e., run the algorithm on a subgraph centered at the node. As a result, this approach is rather inefficient. Becchetti et al. [5] used a min-wise hashing method to approximately count triangles for each individual node in an undirected simple graph, and the algorithm needs to traverse the graph data many times. Our method does not rely on counting triangles for each individual node. Rather, we use a carefully designed estimator to estimate the statistics from a sampled graph in a streaming fashion.

3 Problem formulation

We first formally define the notions of social activity stream and triadic cardinality distribution mentioned in Introduction. Then, we give an overview of our sample-estimate framework.

3.1 Social activity stream

We represent an OSN by a simple graph $G = (V, E, C)$, where V is a set of users, E is a set of relations among users, and C is a set of media content such as hashtags and video links. Here, a relation between two users can be undirected like the friend relationship in Facebook, or directed like the follower relationship in Twitter.

Users in the OSN generate *social activities*, e.g., interact with other users in V, or interact with content in C. We denote a social activity by $a \in V \times (V \cup C) \times [0, \infty)$. A *user–user interaction*, $a = (u, v, t)$, corresponds to user u interacting with user v at time t. A *user–content interaction*, $a = (u, c, t)$, corresponds to user u interacting with content c at time t. These social activities are then aggregated chronologically to form a *social activity stream*, denoted by $S \triangleq \{a_1, a_2, \dots\}$ where a_l denotes the l-th social activity in the stream.

3.2 Triadic cardinality distribution

We introduce two *interaction multigraphs* formed by the two types of user interactions, respectively. Triadic cardinality distribution is then defined on these two interaction multigraphs.

Interaction multigraphs. Within a time window (e.g., an hour, a day, or a week), user–user interactions in stream S form a multigraph $\mathcal{G}_{\text{uu}} = (V, \mathcal{E}_{\text{uu}})$, where V is the set of users, and $\mathcal{E}_{\text{uu}}$ is a multiset consisting of user–user interactions in the window. The *triadic cardinality* of a user $u \in V$ is the number of interaction triangles to which u belongs in $\mathcal{G}_{\text{uu}}$. For example, user u in Fig. 1a has triadic cardinality two, and all other users have triadic cardinality one.

Likewise, user–content interactions also form a multigraph $\mathcal{G}_{\text{uc}} = (V \cup C, E \cup \mathcal{E}_{\text{uc}})$ in a time window. Unlike $\mathcal{G}_{\text{uu}}$, the node set in $\mathcal{G}_{\text{uc}}$ includes both user nodes V and content nodes C, and the edge set includes user relations E and a multiset $\mathcal{E}_{\text{uc}}$ denoting user–content

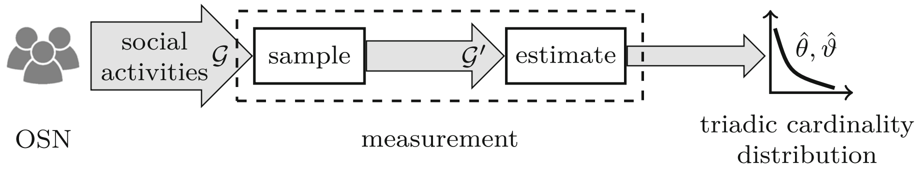


Fig. 3 A sample-estimate framework

interactions in the window. Note that in $\mathcal{G}_{\text{uc}}$, triadic cardinality is only defined for content nodes, and the triadic cardinality of a content node $c \in C$ is the number of influence triangles to which c belongs in $\mathcal{G}_{\text{uc}}$. For example, in Fig. 2a, content c_2 has triadic cardinality one, and all other content nodes have triadic cardinality zero.

Triadic cardinality distribution Let $\theta \triangleq (\theta_0, \dots, \theta_W)$ and $\vartheta \triangleq (\vartheta_0, \dots, \vartheta_{W'})$ denote the triadic cardinality distributions on $\mathcal{G}_{\text{uu}}$ and $\mathcal{G}_{\text{uc}}$, respectively. Here, θ_i (or ϑ_i) is the fraction of user (or content) nodes with triadic cardinality i in $\mathcal{G}_{\text{uu}}$ (or $\mathcal{G}_{\text{uc}}$), and W (or W') denotes the maximum triadic cardinality.

The importance of the triadic cardinality distribution lies in its capability of providing succinct summary information to characterize burst patterns in a large scale OSN as we mentioned previously. By tracking triadic cardinality distributions, we will discover burst occurrences in an OSN, as illustrated in Figs. 1c and 2c.

3.3 Overview of our sample-estimate framework

We propose an online solution capable of tracking the triadic cardinality distribution from a high-speed social activity stream in a near-real-time fashion. Our framework is illustrated in Fig. 3.

Our framework consists of two stages. In the first stage, we sample a social activity stream in a time window maintaining only summary statistics. In the second stage, we construct an estimate of the triadic cardinality distribution from the summary statistics at the end of a time window. The advantage of this approach is that it reduces the amount of data needed to be stored and processed, and enables detecting bursts in a near-real-time fashion.

4 Stream sampling methods

In this section, we elaborate the sampling module in our framework, and design two types of stream sampling methods. The purpose of sampling is to reduce the computational cost in handling the massive amount of data in a high-speed stream.

4.1 Identical triangle sampling (ITS)

The simplest stream sampling method works as follows. We toss a biased coin for each coming social activity $a \in S$. We keep a with probability p, and ignore it with probability $1 - p$. Hence, each social activity is independently sampled, and at the end of the time window, only a fraction p of the stream is kept. When social activities are sampled, triangles

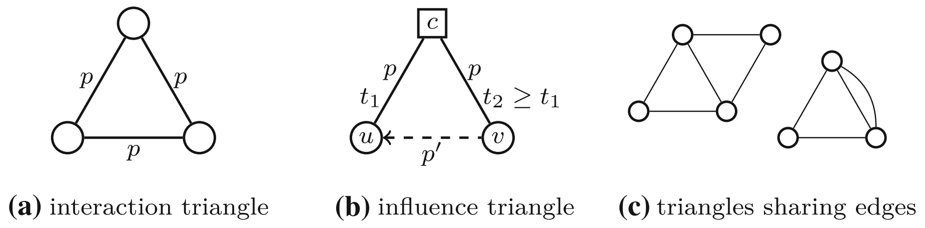


Fig. 4 Sampling triangles. A solid edge represents an interaction in $\mathcal{E}_{\text{uu}} \cup \mathcal{E}_{\text{uc}}$. A dashed edge represents a social edge in E

in graphs $\mathcal{G}_{\text{uu}}$ and $\mathcal{G}_{\text{uc}}$ are sampled accordingly. Obviously, an interaction triangle is sampled with probability p^3, as illustrated in Fig. 4a.

For influence triangles, we need a few more considerations. First, an influence triangle consists of two user–content interaction edges in $\mathcal{E}_{\text{uc}}$ and one social edge in E. Second, stream sampling only applies to edges in $\mathcal{E}_{\text{uc}} \cup \mathcal{E}_{\text{uu}}$, and edges in E are not sampled because they do not appear in the stream. In Fig. 4b, suppose we have sampled two user–content interactions (u, c, t_1) and (v, c, t_2), and assume $t_1 \leq t_2$, i.e., user u interacts with c earlier than v does. To determine whether content c has an influence triangle formed by users u and v, we need to check whether (directed) edge (v, u) exists in E. This can be done by querying neighbors of one of the two users in the OSN. For example, in Twitter, we query *followees* of v and check whether u is in these followees; or in Facebook, we query friends of v and check whether u is a friend of v. Sometimes this query cost is expensive if we do not own G and need to call the APIs provided by the OSN. To reduce this query cost, we check a user pair with probability p'. This is equivalent to sampling a social edge in E with probability p', conditioned on the two associated user–content interactions being sampled. Thus, an influence triangle is sampled with identical probability $p^2 p'$. In summary, we have the following result.

Lemma 1 *If we independently sample each social activity in stream S with probability p, and check a user relation in E with probability p′, then each interaction (influence) triangle in $\mathcal{G}_{uu}$ ($\mathcal{G}_{uc}$) is sampled with probability*

$$p_{\triangle}^{ITS} \triangleq \begin{cases} p^3 & \textit{for an interaction triangle}, \\ p^2 p' & \textit{for an influence triangle}. \end{cases} \tag{1}$$

ITS-color: A more efficient identical triangle sampling method An obvious drawback of the previous sampling method is that an interaction triangle is sampled with probability cubic to the edge sampling probability. This means that an interaction triangle is hardly sampled if p is small. In fact, we can increase triangle sampling probability to p^2 and still keep each edge being sampled with probability p by leveraging a clever *colorful triangle sampling* method [35]. Let $N = 1/p$ be an integer, and $[N]$ represents a set of N colors. Define a hash function $h\colon V \mapsto [N]$ that maps a node to one of these N colors uniformly at random. During sampling, for a coming social activity $a = (u, v, t)$, we keep a if $h(u) = h(v)$, and drop a otherwise. We can see that a user–user interaction is still sampled with probability p, but an interaction triangle is now sampled with probability p^2, and hence it is more efficient in collecting triangles from edge samples. For influence triangle, we can let $p' = 1$, i.e., we check every sampled user pair (similar to gSH [1]), and an influence triangle is also sampled with probability p^2. We will refer to ITS method with colorful triangle sampling as ITS-color in the following discussion.

Remark 1 In both ITS and ITS-color, although triangles are sampled identically, they may *not* be sampled *independently*, such as the cases two triangles share edges in Fig. 4c. We will consider this issue in detail later.

4.2 Harvesting triangles by subgraph sampling (SGS)

The ITS-based methods are easy to implement, and they are already used for counting the triangles in a large network [35,54]. However, ITS-based methods have drawbacks when they are used for estimating the triadic cardinality distribution. One main drawback is that, because ITS samples each triangle with identical probability, the sampling will be biased toward nodes belonging to many triangles. That is, nodes with larger triadic cardinalities are more likely to be sampled, and for nodes with small triadic cardinalities, the triangles these nodes belonging to will be seldom sampled. This hence may incur large estimation error for nodes with small triadic cardinalities. To address this weakness, we propose another triangle sampling method that leverages interaction multigraphs and social graph in a different way, which we call the subgraph sampling (SGS) method.

For sampling interaction triangles, assume that we are only interested in user–user interactions along social edges. Then, SGS works as follows. At the beginning of a time window, we first sample a set of user samples that each user is sampled with probability p_n (and this step can be implemented on social graph G using well-studied graph sampling techniques [19]). For each sampled user, a subgraph induced by the user and the user's neighbors in G is maintained, i.e., each edge in the induced subgraph is a social edge in G. During stream sampling, for each social activity $a = (u, v, t)$, if (u, v) is an edge in one of these subgraphs, we keep a; otherwise a is dropped. In this way, interaction triangles related to the user sample are kept completely.

Similar procedure is also applied to sampling influence triangles, and here we aim to keep complete influence triangles related to each sampled content node. To sample a content node with probability p_n, we need to store content nodes seen so far in a Bloom filter for ease of testing whether a coming content node is new or not. For a coming social activity $a = (u, c, t)$, if c is already marked as a sampled content node, we keep a; if c is new (i.e., c is not found in the Bloom filter), we mark c as a content node sample with probability p_n and save a, otherwise we drop a. No matter a is saved or discarded, if c is new, we store c in the Bloom filter. So, we can see that social activities related to sampled content nodes are all kept. If we also check the existence of social edges with probability $p' = 1$, then influence triangles related to sampled content nodes are kept completely.

Comparing with ITS methods, SGS method has a feature that it keeps every triangle a node sample belonging to, whatever the node sample has a large or small triadic cardinality. We will later develop rigorous method to compare the performance of ITS methods and SGS method.

4.3 Statistics of sampled data

ITS, ITS-color, and SGS can be thought of as sampling edges in multigraphs $\mathcal{G}_{\text{uu}}$, $\mathcal{G}_{\text{uc}}$, and social graph G, in different manners. In ITS, interaction edges $e \in \mathcal{E}_{\text{uu}} \cup \mathcal{E}_{\text{uc}}$ are independently sampled with probability p, and social edges $e' \in E$ are sampled with conditional probability p' conditioned on the two user–content interaction edges being sampled. In SGS, we first sample a collection of user/content nodes with probability p_n, and then only keep triangle subgraphs related to these sampled user/content nodes.

At the end of the time window, we obtain two sampled multigraphs, denoted by $\mathcal{G}'_{\text{uu}}$ and $\mathcal{G}'_{\text{uc}}$, respectively. We calculate statistics on these sampled graphs, and will show that these statistics are sufficient to estimate the triadic cardinality distributions. Specifically, for the sampled graph $\mathcal{G}'_{\text{uu}}$, we calculate triadic cardinality for each (sampled) user node in ITS (SGS), and obtain statistics $g \triangleq (g_0, \ldots, g_W)$, where g_j, $0 \le j \le W$, denotes the number of user nodes belonging to j triangles in graph $\mathcal{G}'_{\text{uu}}$. Similar statistics are also obtained from $\mathcal{G}'_{\text{uc}}$, denoted by $f \triangleq (f_0, \ldots, f_{W'})$, where f_j is the number of content nodes belonging to j influence triangles in graph $\mathcal{G}'_{\text{uc}}$. We only need to store g and f in main memory and use them to estimate θ and ϑ in the next section.

5 Maximum likelihood estimate

In this section, we elaborate the estimation module in our framework, and derive a maximum likelihood estimate (MLE) of the triadic cardinality distribution using statistics obtained in the sampling step. The estimation in this section can be viewed as an analog of the network flow size distribution estimation [15,39,55,57,58], in which a packet in a flow is viewed to be a triangle a node belonging to. However, in our case, triangle samples are not independent, and a node may have no triangles at all. These issues complicate the estimation algorithm, and we will describe how to solve these issues in this section.

Note that we only discuss how to estimate θ using g, as the MLE of ϑ using f is easily obtained using a similar approach. To estimate θ, we first consider the easier case where graph size $|V| = n$ is known. Later, we extend our analysis to the case where $|V|$ is unknown.

5.1 MLE when graph size is known

Recall that g_j, $0 \le j \le W$, is the number of nodes having j sampled triangles in graph $\mathcal{G}'_{\text{uu}}$. First, note that observing a node with j sampled triangles in graph $\mathcal{G}'_{\text{uu}}$ implies that the node has at least j triangles in graph $\mathcal{G}_{\text{uu}}$. Second, we also need to pay special attention to g_0, which is the number of nodes with no triangle observed in graph $\mathcal{G}'_{\text{uu}}$. Due to sampling, some nodes may be unobserved (e.g., no edge attached to the node is sampled in ITS, or the node is not sampled in SGS), and these "evaporated" nodes are actually "observed" to have zero triangle because graph $\mathcal{G}_{\text{uu}}$ has n nodes. Hence, we need to include these evaporated nodes in g_0, i.e.,

$$g_0 := g_0 + n - \sum_{j=0}^{W} g_j = n - \sum_{j=1}^{W} g_j.$$

To derive a MLE of θ, we use a conditional probability to model the sampling process. For a randomly chosen node, let X denote the number of triangles to which it belongs in $\mathcal{G}_{\text{uu}}$, and let Y denote the number of triangles observed during sampling. Let $b_{ji} \triangleq P(Y = j|X = i)$ for $0 \le j \le i$ be the conditional probability that a node has j sampled triangles in $\mathcal{G}'_{\text{uu}}$ given that it has i triangles in the original graph $\mathcal{G}_{\text{uu}}$. Then, the probability of observing a node to have j sampled triangles is

$$P(Y = j) = \sum_{i=j}^{W} P(Y = j|X = i)P(X = i) = \sum_{i=j}^{W} b_{ji}\theta_i. \tag{2}$$

Then, the log-likelihood of observations $\{Y_l = y_l\}_{k=1}^n$, where $Y_l = y_l$ denotes the k-th node having y_l sampled triangles, yields

$$\mathcal{L}(\theta) \triangleq \log P(\{Y_l = y_l\}_{k=1}^n) = \sum_{j=0}^{W} g_j \log \sum_{i=j}^{W} b_{ji}\theta_i. \tag{3}$$

The MLE of θ can then be obtained by maximizing $\mathcal{L}(\theta)$ with respect to θ under the constraint that $\sum_{i=0}^{W}\theta_i = 1$. To solve the likelihood maximization problem, we face two challenges: (1) What are the specific formulas of b_{ji} for the sampling methods we have proposed previously? (2) Note that it is impossible to obtain a closed form solution maximizing Eq. (3), so how should we design an algorithm to maximize Eq. (3) conveniently?

5.1.1 Sampling model specification

We specify the sampling models b_{ji} for ITS and SGS, respectively. We start with SGS for its simplicity.

SGS Remember that SGS keeps all the triangles of each sampled node, and a node is sampled with probability p_n. Hence, if we observe a sampled node belonging to $j > 0$ triangles in $\mathcal{G}'_{\text{uu}}$, the node must have $i = j$ triangles in $\mathcal{G}_{\text{uu}}$, and this occurs with probability p_n. If we observe that a node has no triangle in $\mathcal{G}'_{\text{uu}}$, i.e., $j = 0$, there are two possibilities, i.e., the node indeed has no triangle in $\mathcal{G}_{\text{uu}}$, $i = 0$, or the node is not sampled. Therefore,

$$b_{ji} = \begin{cases} 1, & j = i = 0, \\ p_n, & j = i > 0, \\ 1 - p_n, & j = 0 \wedge i > 0, \\ 0, & \text{otherwise.} \end{cases}$$

ITS and ITS-color In ITS or ITS-color, each triangle is sampled with an identical probability, denoted by $p_\triangle$. Sampling a triangle can be thought of as a Bernoulli trial with success probability $p_\triangle$. If Bernoulli trials are independent with each other, i.e., triangles are independently sampled, then b_{ji} should follow the standard binomial distribution parameterized by i and $p_\triangle$, i.e., $b_{ji} = \text{Bin}(j|i, p_\triangle) \triangleq \binom{i}{j} p_\triangle^j (1 - p_\triangle)^{i-j}$. Unfortunately, independence does not hold for triangles sharing edges, as illustrated in Fig. 4c. As a result, it is non-trivial to derive an accurate sampling model b_{ji} for ITS and ITS-color due to the dependence among triangle samples. To deal with this issue, we propose to use the *beta-binomial distribution* [6], which is a suggested model to approximate the sums of dependent Bernoulli random variables [64]. In our case, we denote the beta-binomial distribution by

$$\text{BetaBin}(j|i, p_\triangle, \alpha) \triangleq \binom{i}{j} \frac{\prod_{s=0}^{j-1}(s\alpha + p_\triangle)\prod_{s=0}^{i-j-1}(s\alpha + 1 - p_\triangle)}{\prod_{s=0}^{i-1}(s\alpha + 1)}$$

and $\prod_0^{-1} \triangleq 1$. The above distribution parameterized by $\alpha \geq 0$ allows pairwise identically distributed Bernoulli random variables to have a covariance $\alpha p_\triangle(1 - p_\triangle)/(1 + \alpha)$. It reduces to the standard binomial distribution when $\alpha = 0$. Hence, for ITS and ITS-color sampling, we propose to approximate the sampling model by

$$b_{ji}(\alpha) = \text{BetaBin}(j|i, p_\triangle, \alpha), \quad 0 \leq j \leq i.$$

Intuitively, if there are many triangles sharing edges in the graph, then covariance between two triangle samples will turn to be large, and we expect to have a large α in the model; otherwise the graph may be sparse and only few triangles share edges, then covariance will be small and we expect a small α in the model. Therefore, the covariance parameter α allows us to model dependent triangle samples in our problem.

5.1.2 MLE via EM algorithm

To solve the second challenge, we propose to use the expectation-maximization (EM) algorithm to obtain the MLE in a more convenient way. For general consideration, we use $b_{ji}(\alpha)$ to denote the sampling model.

If we already know that the k-th node has x_l triangles in $\mathcal{G}_{\text{uu}}$, i.e., $X_l = x_l$, then the complete likelihood of observations $\{(Y_l, X_l)\}_{l=1}^n$ is

$$\begin{aligned} P(\{(Y_l, X_l)\}_{l=1}^n) &= \prod_{l=1}^{n} P(Y_l = y_l, X_l = x_l) \\ &= \prod_{j=0}^{W} \prod_{i=j}^{W} P(Y = j, X = i)^{z_{ij}} \\ &= \prod_{j=0}^{W} \prod_{i=j}^{W} \left[b_{ji}(\alpha)\theta_i\right]^{z_{ij}} \end{aligned}$$

where $z_{ij} = \sum_{l=1}^n \mathbf{1}(x_l = i \wedge y_l = j)$ is the number of nodes with i triangles and j of them being sampled; $\mathbf{1}(\cdot)$ is the indicator function. The complete log-likelihood is

$$\mathcal{L}_c(\theta, \alpha) \triangleq \sum_{j=0}^{W} \sum_{i=j}^{W} z_{ij} \log\left[b_{ji}(\alpha)\theta_i\right]. \tag{4}$$

Here, we can treat $\{X_l\}_{l=1}^n$ as hidden variables, and apply the EM algorithm to calculate the MLE.

E-step We calculate the expectation of the complete log-likelihood in Eq. (4) with respect to hidden variables $\{X_l\}_l$, conditioned on data $\{Y_l\}_l$ and previous estimates $\theta^{(t)}$ and $\alpha^{(t)}$. That is

$$Q(\theta, \alpha; \theta^{(t)}, \alpha^{(t)}) \triangleq \sum_{j=0}^{W} \sum_{i=j}^{W} \mathbb{E}[z_{ij}|\theta^{(t)}, \alpha^{(t)}] \log\left[b_{ji}(\alpha)\theta_i\right].$$

Here, $\mathbb{E}[z_{ij}|\theta^{(t)}, \alpha^{(t)}]$ can be viewed as the average number of nodes that have i triangles in $\mathcal{G}_{\text{uu}}$, of which j are sampled. Because

$$\begin{aligned} P(X = i|Y = j, \theta^{(t)}, \alpha^{(t)}) &= \frac{P(Y = j|X = i, \alpha^{(t)})P(X = i|\theta^{(t)})}{\sum_{i'} P(Y = j|X = i', \alpha^{(t)})P(X = i'|\theta^{(t)})} \\ &= \frac{b_{ji}(\alpha^{(t)})\theta_i^{(t)}}{\sum_{i'} b_{ji'}(\alpha^{(t)})\theta_{i'}^{(t)}} \triangleq p_{i|j} \end{aligned}$$

and we have observed g_j nodes belonging to j sampled triangles, then $\mathbb{E}[z_{ij}|\theta^{(t)}, \alpha^{(t)}] = g_j p_{i|j}$.

M-step We now maximize $Q(\theta, \alpha; \theta^{(t)}, \alpha^{(t)})$ with respect to θ and α subject to the constraint $\sum_{i=0}^{W} \theta_i = 1$. After the log operation, θ and α are well separated. Hence, we obtain

$$\begin{aligned}\theta_i^{(t+1)} &= \arg\max_{\theta} Q(\theta, \alpha; \theta^{(t)}, \alpha^{(t)}) \\ &= \frac{\sum_{j=0}^{i} \mathbb{E}[z_{ij}|\theta^{(t)}, \alpha^{(t)}]}{\sum_{j=0}^{W}\sum_{i'=j}^{W} \mathbb{E}[z_{i'j}|\theta^{(t)}, \alpha^{(t)}]}, \quad 0 \le i \le W,\end{aligned}$$

and $\alpha^{(t+1)} = \arg\max_{\alpha} Q(\theta, \alpha; \theta^{(t)}, \alpha^{(t)})$ can be solved using classic gradient ascent methods.

Alternating iterations of the E-step and M-step, EM algorithm converges to a solution, which is a local maximum of $\mathcal{L}(\theta, \alpha)$. We denote this solution by $\hat{\theta}$ and $\hat{\alpha}$.

5.2 MLE when graph size is unknown

When the graph size is unknown, one can use probabilistic counting methods such as loglog counting [16] to obtain an estimate of graph size from the stream, and then apply our previously developed method to obtain estimate $\hat{\theta}$. Note that this introduces additional statistical errors to $\hat{\theta}$ due to the inaccurate estimate of the graph size. In what follows, we slightly reformulate the problem and develop a method that can simultaneously estimate both the graph size and the triadic cardinality distribution from the sampled data.

When the graph size is unknown, we cannot calibrate g_0 because "evaporated" nodes are indeed unobservable in this case. There is no clear relationship between an unsampled node and its triadic cardinality. As a result, we cannot easily model the absence of nodes by θ. If we observe a node having no triangle after sampling, we cannot reason out which way caused the observation, the node has no triangle in the original graph, or the triangles the node belonging to are not sampled. This difficulty hence complicates the estimation design.

To solve this issue, we need to slightly reformulate our problem: (1) Instead of estimating the total number of nodes in $\mathcal{G}_{\text{uu}}$, we estimate the number of nodes belonging to at least one triangle in $\mathcal{G}_{\text{uu}}$, denoted by n_+; (2) We estimate the triadic cardinality distribution $\theta^+ = (\theta_1^+, \dots, \theta_W^+)$, where θ_i^+ is the fraction of nodes with i triangles over the nodes having at least one triangle in $\mathcal{G}_{\text{uu}}$.

Estimating n_+ Under the beta-binomial model, the probability that a node has i triangles in $\mathcal{G}_{\text{uu}}$, of which none are sampled, is

$$q_i(\alpha) \triangleq P(Y = 0|X = i) = b_{0i}(\alpha).$$

Then, the probability that a node has triangles in $\mathcal{G}_{\text{uu}}$, of which none are sampled, is

$$q(\theta^+, \alpha) \triangleq P(Y = 0|X \ge 1) = \sum_{i=1}^{W} q_i(\alpha)\theta_i^+.$$

Because there are $\sum_{j=1}^{W} g_j$ nodes having been observed to have at least one sampled triangle, n_+ can be estimated by

$$\hat{n}_+ = \frac{\sum_{j=1}^{W} g_j}{1 - q(\theta^+, \alpha)}. \tag{5}$$

Note that estimator (5) relies on θ^+ and α, and we can estimate them using the following procedures.

Estimating θ^+ and α We discard g_0 and only use $g^+ \triangleq (g_1, \ldots, g_W)$ to estimate θ^+ and α. The basic idea is to derive the likelihood for nodes that are observed to have at least one sampled triangle, i.e., $\{Y_l = y_l : y_l \geq 1\}$. In this case, the probability that a node has $X = i$ triangles, and $Y = j$ of them are sampled, conditioned on $Y \geq 1$, is

$$\begin{aligned} P(Y = j | X = i, Y \geq 1) &= \frac{P(Y = j | X = i)}{P(Y \geq 1 | X = i)} \\ &= \frac{b_{ji}(\alpha)}{1 - q_i(\alpha)} \triangleq a_{ji}(\alpha), \quad j \geq 1. \end{aligned}$$

Then, the probability that a node is observed to have j sampled triangles, conditioned on $Y \geq 1$, is

$$\begin{aligned} P(Y = j | Y \geq 1) &= \sum_{i=j}^{W} P(Y = j | X = i, Y \geq 1) P(X = i | Y \geq 1) \\ &= \sum_{i=j}^{W} a_{ji}(\alpha) \phi_i, \end{aligned}$$

where

$$\phi_i \triangleq P(X = i | Y \geq 1) = \frac{\theta_i^+ [1 - q_i(\alpha)]}{\sum_{i''=1}^{W} \theta_{i'}^+ [1 - q_{i'}(\alpha)]}, \tag{6}$$

is the distribution of observed node triadic cardinalities. Now it is straightforward to obtain the previously mentioned likelihood. Furthermore, we can leverage our previously developed EM algorithm by replacing θ_i by ϕ_i, b_{ji} by a_{ji}, to obtain MLEs for ϕ and α. We omit these details, and directly provide the final EM iterations:

$$\phi_i^{(t+1)} = \frac{\sum_{j=1}^{i} \mathbb{E}[z_{ij} | \phi^{(t)}, \alpha^{(t)}]}{\sum_{j=1}^{W} \sum_{i'=j}^{W} \mathbb{E}[z_{i'j} | \phi^{(t)}, \alpha^{(t)}]}, \quad i \geq 1,$$

where

$$\mathbb{E}[z_{ij} | \phi^{(t)}, \alpha^{(t)}] = \frac{g_j a_{ji}(\alpha^{(t)}) \phi_i^{(t)}}{\sum_{i'=j}^{W} a_{ji'}(\alpha^{(t)}) \phi_{i'}^{(t)}}, \quad i \geq j \geq 1,$$

and $\alpha^{(t+1)} = \arg\max_\alpha Q(\phi, \alpha; \phi^{(t)}, \alpha^{(t)})$ is solved using gradient ascent methods.

Once EM converges, we obtain estimates $\hat{\phi}$ and $\hat{\alpha}$. The estimate for θ^+ is then obtained by Eq. (6), i.e.,

$$\hat{\theta}_i^+ = \frac{\hat{\phi}_i / [1 - q_i(\hat{\alpha})]}{\sum_{i'=1}^{W} \hat{\phi}_{i'} / [1 - q_{i'}(\hat{\alpha})]}, \quad 1 \leq i \leq W. \tag{7}$$

Finally, $\hat{n}_+$ is obtained by the estimator in Eq. (5).

5.3 Logarithmic binning simplification

In our previous study [66], we have observed that triadic cardinality distributions in many real-world networks exhibit heavy tails. Thus, it is better to characterize them in the logarithmic scale. That is, instead of estimating the fraction of nodes having exact triadic cardinality i, we may want to estimate the fraction of nodes with triadic cardinality in $\log_2$ scaled bins. We aim to estimate the fraction of nodes having triadic cardinality in the k-th bin $[2^k, 2^{k+1})$, denoted by θ_k^+, for $k = 0, 1, \ldots, K$ where $K \triangleq \lfloor \log_2 W \rfloor$. If we allow $i = 0$ as in the case where graph size is known, we define the first bin to be $\{0\}$ assigning to bin $k = -1$, and use $\theta_k, k = -1, 0, \ldots, K$, to represent the binned triadic cardinality distribution.

In the logarithmic binning simplification, for each i in the k-th bin, we assume that θ_i has the same value, and $\theta_i = 2^{-k}\theta_k$ for $k \geq 0$. We further define

$$\begin{aligned} b_{jk} &\triangleq P(Y = j | X \in \text{bin}(k)) && (8)\\ &= \sum_{i=2^k}^{2^{k+1}-1} P(Y = j | X = i) P(X = i | X \in \text{bin}(k)) && (9)\\ &= 2^{-k} \sum_{i=2^k}^{2^{k+1}-1} b_{ji} && (10) \end{aligned}$$

for $k \geq 0$. For $k = -1$, we define $b_{jk} = 1$ if $j = 0$ and 0 otherwise. Similar to Eq. (2), the probability of observing a node having j triangles after sampling becomes $P(Y = j) = \sum_{k=-1}^{K} b_{jk}\theta_k$. Thus, it is straightforward to obtain a MLE of θ_k using previously developed methods. Similar analysis can also be applied to estimate θ_k^+, and hence is omitted. In the logarithmic binning method, parameters $\{\theta_k\}$ and $\{\theta_k^+\}$ are actually smoothed versions of the original parameters $\{\theta_i\}$ and $\{\theta_i^+\}$, respectively, which we will observe in experiments.

The logarithmic binning simplification reduces the number of parameters from $O(W)$ to $O(\log_2 W)$. This allows us to consider large triadic cardinality bound W in large networks. Because the computational complexity of the EM algorithm is proportional to the number of parameters to be estimated, hence the logarithmic binning simplification reduces the computational complexity by a factor of $O(W/\log_2 W)$.

6 Asymptotic estimation error analysis

To evaluate the performance of MLEs using different sampling methods, this section devotes to analyze the asymptotic estimation error of the MLEs by calculating the Cramér–Rao lower bound (CRLB) of $\hat{\theta}$ and $\hat{\theta}^+$. It is well known that MLE is asymptotically Gaussian centered at the true value with variance the CRLB, and the Cramér–Rao theorem further states that the mean squared error of any unbiased estimator is lower bounded by the CRLB, which is the inverse of the Fisher information (see [52, Chapter 2] for more details).

Intuitively, the Fisher information can be thought of as the amount of information that observations $\{Y_l\}$ carry about unobservable parameters θ (or θ^+) upon which the probability distribution of the observations depends. When graph size is known, the Fisher information of observations $\{Y_l\}$ is a $(W+1) \times (W+1)$ square matrix $J(\theta)$ whose ir-th element is given by

$$J_{ir}(\theta) \triangleq \mathbb{E}_Y \left[\frac{\partial \log P(\{Y_l\}_l | \theta)}{\partial \theta_i} \frac{\partial \log P(\{Y_l\}_l | \theta)}{\partial \theta_r} \right].$$

In our problem, $J_{ir}(\theta)$ can be further simplified to

$$J_{ir}(\theta) = n \sum_{j=0}^{W} \frac{\partial \log P(Y|\theta)}{\partial \theta_i} \frac{\partial \log P(Y|\theta)}{\partial \theta_r} P(Y=j)$$
$$= n \sum_{j=0}^{W} \frac{b_{ji}(\alpha) b_{jr}(\alpha)}{P(Y=j)}.$$

When graph size is unknown, the Fisher information matrix $J(\theta^+)$ is a $W \times W$ matrix. To obtain $J(\theta^+)$, we can first obtain the Fisher information matrix regarding to ϕ, denoted by $J(\phi)$, using an approach similar to above (by replacing n by n^+). Then, $J(\phi)$ and $J(\theta^+)$ are known to have the following relationship [14]

$$J(\theta^+)^{-1} = \nabla H J(\phi)^{-1} \nabla H^T,$$

where ∇H is the Jacobian matrix, and its ir-th element is given by $\partial \theta_i^+(\phi)/\partial \phi_r$ (and $\theta_i^+(\phi)$ is given by Eq. (7)).

The inverse constrained Fisher information of θ with constraint $\sum_i \theta_i = 1$ is then obtained by

$$I(\theta) = J(\theta)^{-1} - \theta\theta^T,$$

where the term $\theta\theta^T$ corresponds to the accuracy gain due to constraint $\sum_i \theta_i = 1$ (see [20,55] for more details). Then, mean squared error of an estimator $\hat{\theta}$ is lower bounded by the diagonal elements of $I(\theta)$, i.e., $\mathbb{E}[(\hat{\theta}_i - \theta_i)^2] \geq I_{ii}(\theta)$. Similar relation also holds for $\hat{\theta}^+$.

MLE is asymptotically efficient, and CRLB is its asymptotic variance. We are thus able to leverage CRLB to compare the asymptotic estimation accuracy of MLEs using different sampling methods.

7 Experiments and validations

In this section, we first empirically verify the claims we have made previously. Then, we validate the proposed estimation methods on several real-world networks. Finally, we illustratively show the usefulness of triadic cardinality distribution in detecting bursts during the Hong Kong occupy central movement in Twitter.

7.1 Analyzing bursts in enron dataset

In the first experiment, we use a public email communication dataset to empirically show how bursts in networks can change the triadic cardinality distribution, and verify our claims previously made.

7.1.1 Enron email dataset

The Enron email dataset [25] includes the entire email communications (e.g., who sent an email to whom at what time) of the Enron corporation from its startup to bankruptcy. The used dataset is carefully cleaned by removing spamming accounts/emails and emails with incorrect timestamps. The cleaned dataset contains 22, 477 email accounts and 164, 081

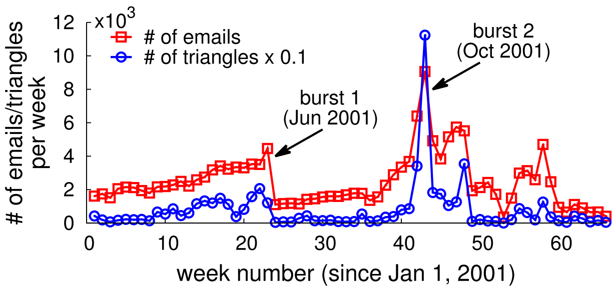


Fig. 5 Email and triangle volumes per week

email communications between Jan 2001 and Apr 2002. We use this dataset to study patterns of bursts caused by email communications among people, i.e., by user–user interactions.

7.1.2 Observations from data

Because the data has been cleaned, the number of user–user interactions, i.e., number of sent emails per time window, reliably indicates burst occurrences. We show the number of emails sent per week in Fig. 5, and observe at least two bursts that occurred in Jun and Oct 2001, respectively. We also show the number of interaction triangles formed during each week. The Pearson correlation coefficient (PCC) between the email and triangle volume series is 0.8, which reflects a very strong correlation. The sudden increase (or decrease) of email volumes during the two bursts is accompanied with the sudden increase (or decrease) of the number of triangles. Thus, this observation verifies our claim that the emergence of a burst is accompanied with the formation of triangles in networks.

How bursts change triadic cardinality distributions Our burst detection method relies on a claim that, when a burst occurs, the triadic cardinality distribution changes. To see this, we show the triadic cardinality distributions before and during the bursts in Fig. 6. For the first burst, due to the sudden decrease in email communications from week 23 to week 24, we observe in Fig. 6a that the distribution "shifts" to the left. While for the second burst, due to the gradual increase in email communications, we observe in Fig. 6b that the distribution in week 43 shifts to the right in comparison to previous weeks. Again, the observation verifies our claim that triadic cardinality distribution changes when a burst occurs.

Impacts of spam As we mentioned earlier, if spam exists, simply using the volume of user interactions to detect bursts will result in false alarms, while the triadic cardinality distribution is a good indicator immune to spam. To demonstrate this claim, suppose a spammer suddenly becomes active in week 23, and generates email spams to distort the original triadic cardinality distribution of week 23. We consider the following two spamming strategies:

- *Random* The spammer randomly chooses many target users to send spam.
- *Random-Friend* At each step, the spammer randomly chooses a user and a random friend of the user[1], as two targets; and sends spams to each of these two targets. The spammer repeats this step a number of times.

In order to measure the extent that spams can distort the original triadic cardinality distribution of week 23, we use Kullback–Leibler (KL) divergence to measure the difference

[1] We assume two Enron users are friends if they have at least one email communication in the dataset.

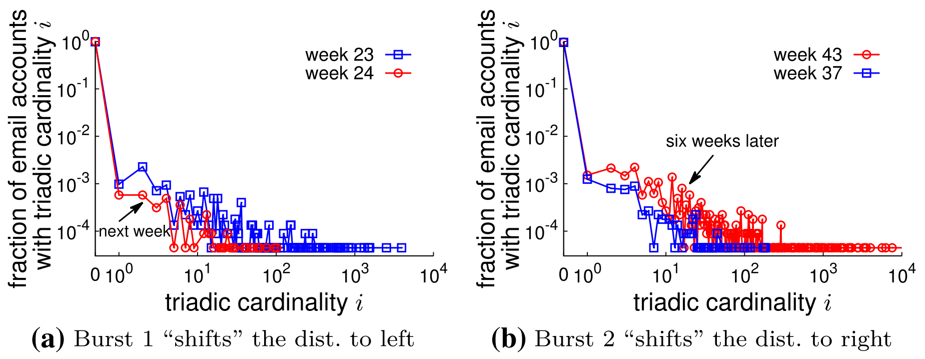


Fig. 6 Bursts change distribution curves. For burst 1 (burst 2), the probability mass at $i = 0$ slightly increases (decreases) actually

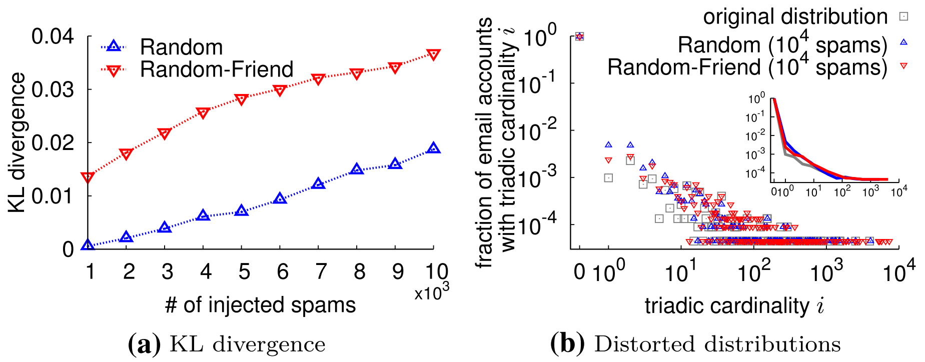


Fig. 7 Impacts of spam. In (**b**), the inset shows fitted curves of the three distributions

between the original and distorted distributions. The relationship between KL divergence and the number of injected spams is shown in Fig. 7a. For both strategies, KL divergences both increase as more spams are injected into the interaction network, which is expected. The Random-Friend strategy can cause larger divergences than the Random strategy, as Random-Friend strategy is easier to introduce new triangles to the interaction network of week 23 for the reason that two friends are more likely to communicate in a week. However, even when 10^4 spams are injected, the spams incur an increasing KL divergence of less than 0.04. From Fig. 7b, we can see that the divergence is indeed small. (This may be explained by the "*center of attention*" phenomenon [3], i.e., a person may have hundreds of friends but he usually only interacts with a small fraction of them in a time window. Hence, Random-Friend strategy does not form many triangles.) Therefore, these observations verify that triadic cardinality distribution is robust against common spamming attacks.

7.2 Validating estimation performance

In the second experiment, we evaluate the MLE performance using different sampling methods and demonstrate the computational efficiency.

Table 1 Network statistics

Network	Type	Nodes	Edges
HepTh	Directed, citation	27,770	352,807
DBLP	Undirected, coauthor	317,080	1,049,866
YouTube	Undirected, OSN	1,134,890	2,987,624
Pokec	Directed, OSN	1,632,803	30,622,564

7.2.1 Datasets

Because the input of our estimation methods is actually a sampled graph, we use public available graphs of different types and scales from the SNAP graph repository (http://snap.stanford.edu/data) as our testbeds. Note that these graphs are treated as the aggregated graphs in a time window, and we apply our estimation algorithms on these graphs to evaluate how accurate our algorithms can estimate the true triadic cardinality distributions. We summarize the statistics of these graphs in Table 1.

For each graph, we first shuffle the edges to form a stream, then we apply stream sampling methods on the stream, and obtain a sampled graph. We calculate the triadic cardinality for nodes in the sampled graph, and obtain statistics g. Note that the estimator uses g to obtain an estimate of the triadic cardinality distribution for each graph, which is then compared with the ground truth distribution, i.e., the triadic cardinality distribution of the original unsampled graph, to evaluate the performance of the estimation method.

7.2.2 CRLB analysis

Our goal is to compare the amount of information contained in edge samples collected using different sampling methods, in terms of CRLB. A small CRLB indicates small asymptotic variance of a MLE, and hence implies that the corresponding sampling method is efficient in gathering information from data. We will mainly use the HepTh and DBLP networks in this study, and for ease of conducting matrix algebra, we truncate the stream with $W = 20$ by discarding edges that may increase a node's triadic cardinality to larger than 20.

We depict the results when graph size is known in Fig. 8. In (a) and (b), we show the rooted CRLB of ITS and ITS-color with different triangle sampling rates $p_\triangle$ on the two networks, respectively. As expected, when $p_\triangle$ increases, CRLB decreases, indicating that we can obtain more accurate estimates by increasing edge sampling rate. However, we find that ITS and ITS-color are not efficient in gathering information from data. As we can see, to decrease CRLB to less than 0.1, we need to increase $p_\triangle$ to about 0.6, which corresponds to a very large edge sampling rate! We then study the performance of SGS in (c) and (d). We observe that CRLB decreases when p_n increases, i.e., when more nodes (or subgraphs) are sampled. We also observe that CRLB of SGS is much smaller than ITS, even with small p_n. It seems that SGS is more efficient in gathering information from data than ITS. However, people may argue that SGS may sample more edges than ITS even with small p_n. To compare them fairly, we need to fix the number of edge samples used by different methods. In ITS or ITS-color, if the graph contains m edges, then ITS or ITS-color samples mp edges on average. In SGS, because a randomly chosen node has $\sum_i i\theta_i$ triangles, then approximately, SGS samples $\Theta(np_n \sum_i i\theta_i)$ edges on average (if we assume #edges $= \Theta$(#triangles)). In the experiment, we turn $p_\triangle$ and p_n to make sure that different methods indeed use same amounts of edge samples approximately, and show the results in (e) and (f). We can see

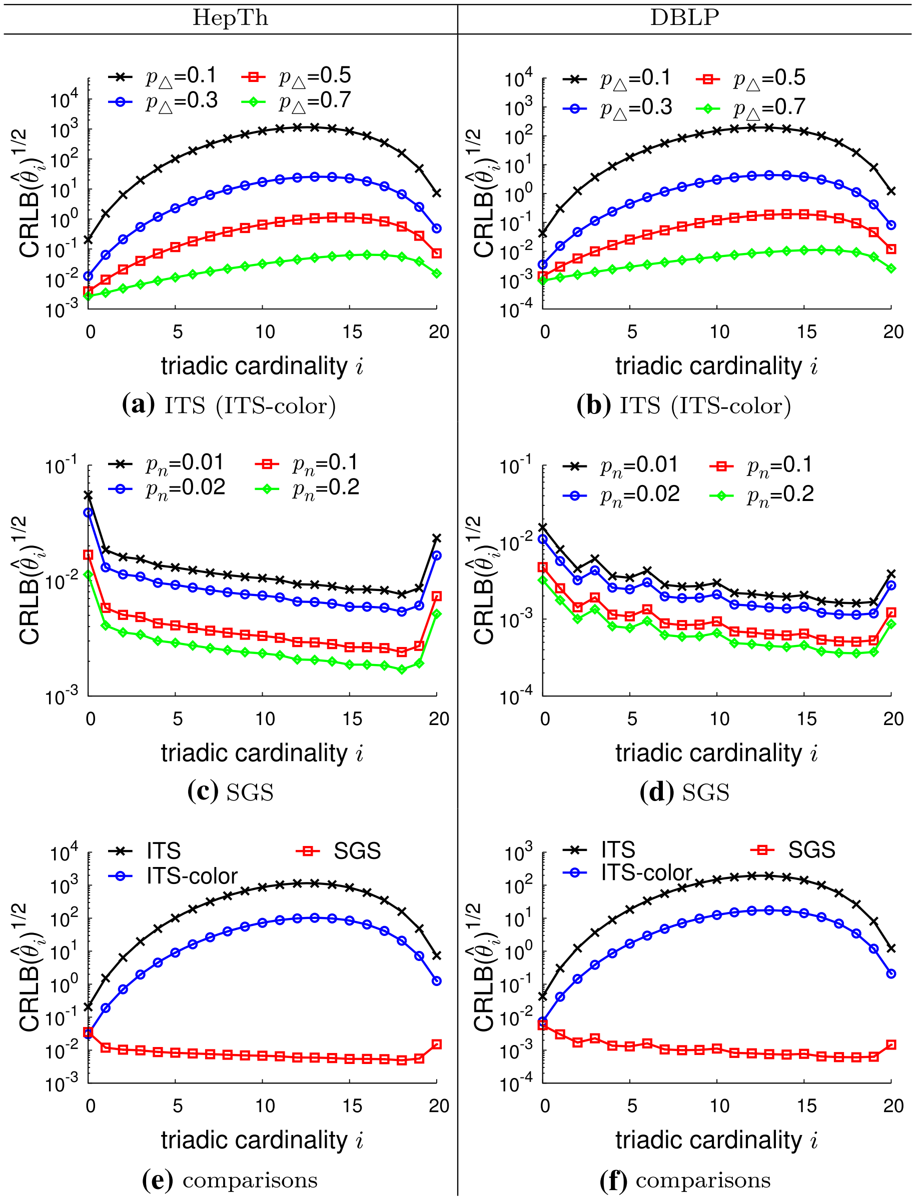


Fig. 8 CRLB analysis when graph size is known. $\alpha = 0.1$. In **e** and **f**, $p_\triangle^{\text{ITS}} = 0.1$ and $p_\triangle^{\text{ITS-color}} = 0.22$. In (**e**), $p_n = 0.024$. In (**f**), $p_n = 0.07$

clearly that SGS is indeed more efficient than ITS and ITS-color. ITS-color is also more efficient than ITS since ITS-color samples a triangle with larger probability than ITS using same edge sampling rate.

We also conduct same experiments when graph size is unknown, and the results are depicted in Fig. 9. The observations are consistent with the results when graph size is known.

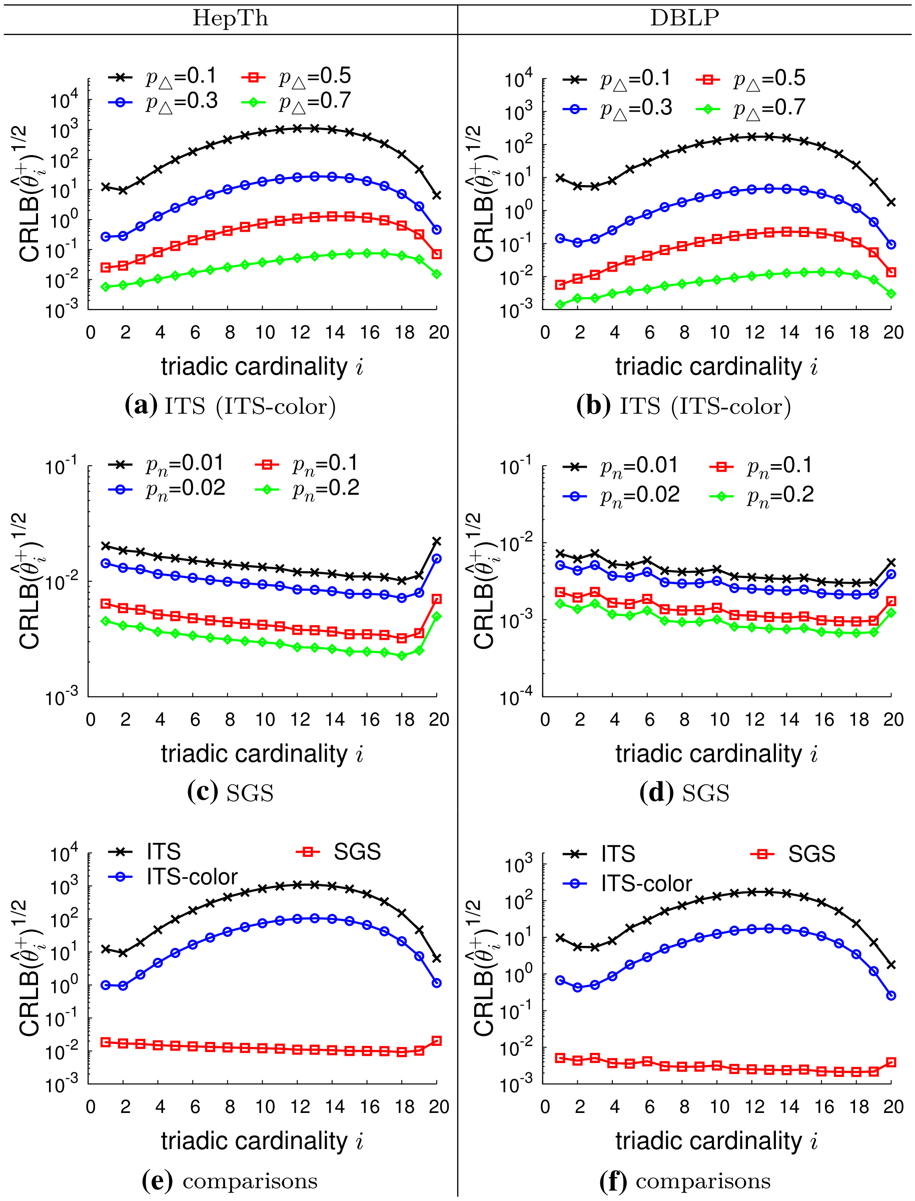


Fig. 9 CRLB analysis when graph size is unknown. In ITS and ITS-color, $\alpha = 0.1$. Parameters in (**e**) and (**f**) are the same as in Fig. 8

7.2.3 NRMSE analysis

CRLB reflects the asymptotic variance of a MLE, i.e., the variance when sample size approaches infinity. However, in practice, we cannot collect infinite many samples because the number of edges in a stream is finite, or afford to use large sample rate. When sample rate is small, or collected edge samples are not large, MLE is usually biased and we cannot leverage CRLB to analyze its performance (see [52, p. 147] for details). Instead, we propose to use the normalized rooted mean squared error (NRMSE) of an estimator, which is defined by $\text{NRMSE}(\hat{\theta}_i) = \sqrt{\mathbb{E}(\hat{\theta}_i - \theta_i)^2}/\theta_i$. The smaller the NRMSE, the more accurate an estimator is. In the following experiment, we mainly use the HepTh network, and compare different sampling methods using approximately the same amount of edge samples.

We depict the results when graph size is known in Fig. 10. In (a) and (b), we compare the estimates and NRMSE of different methods. In general, SGS is better than ITS-color, and ITS-color is better than ITS. This observation is consistent with our previous CRLB analysis. The NRMSE plots in (b) provide more valuable observations. We observe that SGS can provide more accurate estimates for nodes with small triadic cardinalities than ITS and ITS-color; however, SGS performs much worse than ITS and ITS-color for nodes with large triadic cardinalities. This observation can be explained from the different nature between SGS and ITS-based methods. The ITS-based methods sample each triangle with identical probability, and if dependence between triangles is negligible, the sampling will be strongly biased toward nodes with many triangles. That is, nodes with larger triadic cardinalities are more likely to be sampled, and for nodes with small triadic cardinalities, the triangles these nodes belonging to will be seldom sampled. This results in that triangles of small triadic cardinality nodes are difficult to be sampled, and hence incurs large estimation error for these nodes. SGS is completely different, and it reserves all the triangles of each node sample. Because nodes are sampled with same probability, node samples will be dominated by nodes with small triadic cardinalities. Hence, SGS can provide more accurate estimates for the head of triadic cardinality distribution. However, SGS is inefficient in sampling nodes with large triadic cardinalities, resulting in large NRMSE at the tail of the triadic cardinality distribution. To address their weaknesses, one way is to increase sampling rates, as depicted in (c) and (d). We observe that after increasing sampling rates, estimation accuracy increases more or less for each method. An alternative way is to design a mixture estimator, which combines the advantage (and the disadvantage) of each method. For example, we define

$$\hat{\theta}_i^{\text{mix}} \triangleq c\hat{\theta}_i^{\text{ITS-color}} + (1-c)\hat{\theta}_i^{SGS},$$

where $c \in [0, 1]$ is a constant. $\hat{\theta}_i^{\text{mix}}$ has the property that, its variance is smaller than $\hat{\theta}^{\text{SGS}}$ for nodes with large triadic cardinalities, with the loss of accuracy for nodes with small triadic cardinalities, and the variance of the mixture estimator achieves minimal at $c^* = var(\hat{\theta}_i^{SGS})/[var(\hat{\theta}_i^{\text{ITS-color}}) + var(\hat{\theta}_i^{SGS})]$. Figure 10e, f shows the results of a mixture estimator with $c = 0.5$. We indeed observe improvements for nodes with large triadic cardinalities.

We also conduct experiments when graph size is unknown, and show the results in Fig. 11. The observations are consistent with the results when graph size is known in general, and we observe that SGS has smaller NRMSE than ITS-based methods even for nodes with large triadic cardinalities.

7.2.4 Computational efficiency

Finally, we conduct experiments to analyze the computational efficiency of the proposed sample-estimate framework.

As mentioned previously (see Sect. 2), there are many researches on efficiently counting triangles [1,2,9,23,29,38,46,47,54,61,65] or more general subgraphs [36,42,43] in large

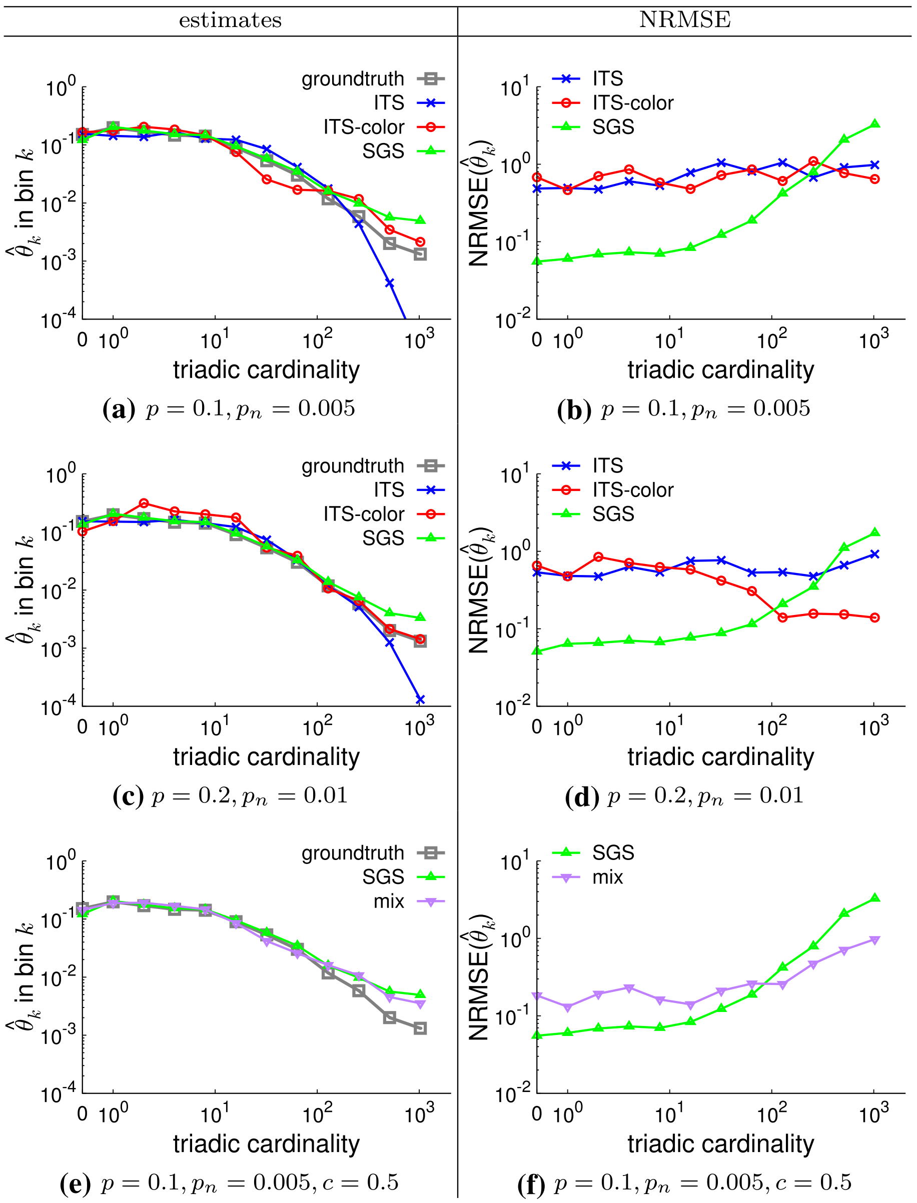
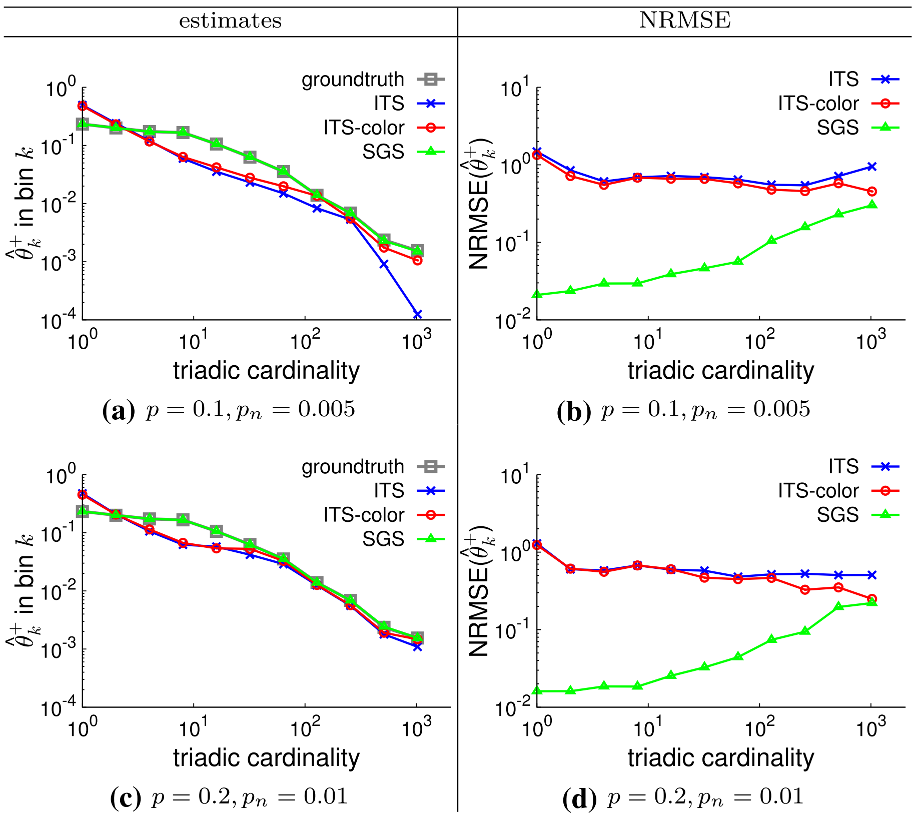


Fig. 10 Estimation accuracy analysis on HepTh when graph size is known

graphs. These methods mainly focus on counting the number of triangles or subgraphs existing in the graph. Note that our goal is to obtain the triadic cardinality distribution of the graph, which requires stating the triangle counts in a finer granularity than simply counting triangles globally. A straightforward way to apply existing methods on our problem is that: (i) we first obtain the ego-network—the graph representing the connections among the neighbors of a node—for each node; (ii) then we apply existing methods on these ego-networks to obtain



(a) $p = 0.1, p_n = 0.005$ **(b)** $p = 0.1, p_n = 0.005$

(c) $p = 0.2, p_n = 0.01$ **(d)** $p = 0.2, p_n = 0.01$

Fig. 11 Estimation accuracy analysis on HepTh when graph size is unknown

the triadic cardinality of each node; (iii) stating these triadic cardinalities, we finally obtain the triadic cardinality distribution of the graph. Obviously, this approach is significantly inefficiently because there may be millions of nodes in a graph (e.g., YouTube and Pokec), and counting triangles for each node will be of high cost. Our sample-estimate approach actually avoids brutally counting triangles for each node but just for a subset of these nodes, and the estimates based on node/edge samples are guaranteed to be asymptotically unbiased (due to the property of maximum likelihood estimate).

We have conducted experiments on two larger networks, i.e., YouTube and Pokec, and we compare the computational efficiency of our sampling approach against a naive method that uses all of the original graph to calculate θ in an exact fashion. The results are shown in Fig. 12. In general, for ITS, using edge sampling rates between 0.1 and 0.3, or using node sampling rates between 0.01 and 0.1, we can achieve a significant speed-up between 10 and 200.

In addition, we also show the throughput of the proposed sample-estimate approach on YouTube and Pokec in Fig. 13. The throughput of an algorithm is defined as the number of edges that the algorithm can process per second. The results show that the throughput decreases as the sample rate increases. ITS with $p = 0.2$ and SGS with $p_n = 0.01$ use approximately the same memory space, but SGS has a higher throughput than ITS.

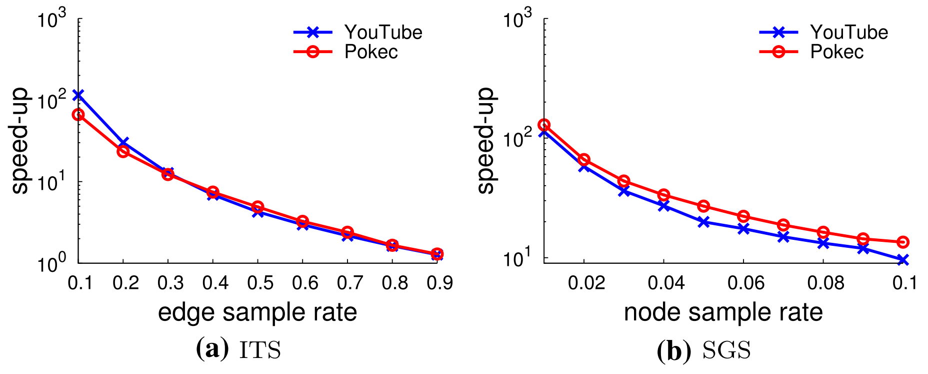


Fig. 12 Efficiency comparison

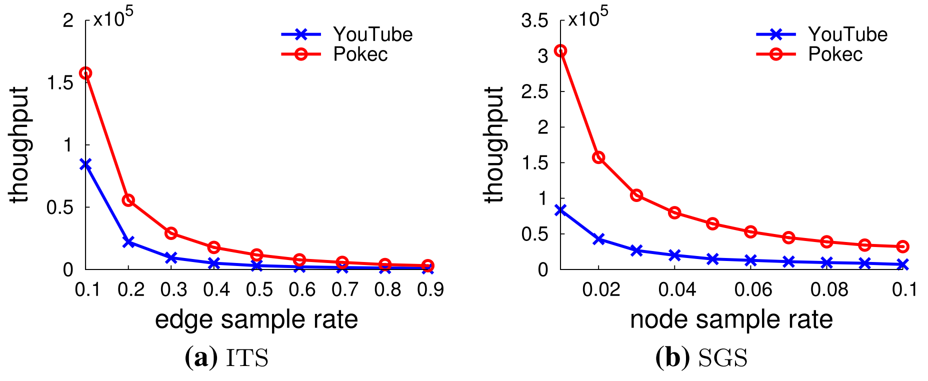


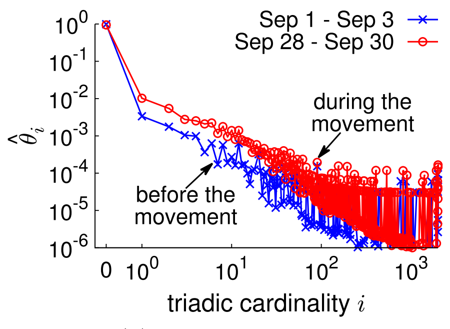
Fig. 13 Throughput analysis

7.3 Application: tracking triadic cardinality distributions during the 2014 Hong Kong occupy central movement

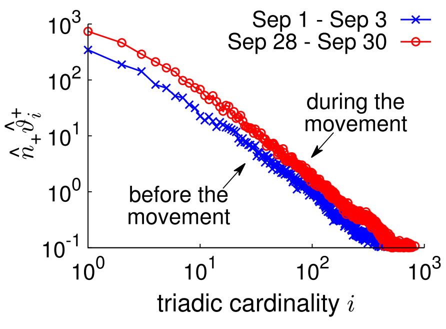
Last, we conduct an application to illustratively show the usefulness of tracking triadic cardinality distributions during the 2014 Hong Kong Occupy Central movement in Twitter.

Hong Kong Occupy Central movement a.k.a. the Umbrella Revolution, began in Sept 2014 when activists in Hong Kong protested against the government and occupied several major streets of Hong Kong to go against a decision made by China's Standing Committee of the National People's Congress on the proposed electoral reform. Protesters began gathering from Sept 28 on and we collected the data between Sept 1 and Nov 30 in 2014.

Building a Twitter social activity stream. The input of our solution is a social activity stream from Twitter. For Twitter itself, this stream is easily obtained by directly aggregating tweets of users. While for third parties who do not own user's tweets, the stream can be obtained by following a set of users, and aggregating tweets from these users to form a social activity stream. Since the movement had already begun prior to our starting this work, we rebuilt the social activity stream by searching tweets containing at least one of the following hashtags: #OccupyCentral, #OccupyHK, #UmbrellaRevolution, #UmbrellaMovement and #UMHK,



(a) Interaction burst



(b) Cascading burst

Fig. 14 Triadic cardinality distributions before and during the movement. Estimated using a mixture estimator with $c = 0.5$, $p = 0.2$, $p_n = 0.002$

between Sept 1 and Nov 30 using Twitter search APIs. This produced 66, 589 Twitter users, and these users form the detectors from whom we want to detect bursts. Next, we collect each user's tweets between Sept 1 and Nov 30, and extract user mentions (i.e., user–user interactions) and user hashtags (i.e., user–content interactions) from tweets to form a social activity stream, with a time span of 91 days.

Settings We set the length of a time window to be one day. For interaction bursts caused by user–user interactions, because we know the user population, i.e., $n = 66,589$, we apply the first estimation method to obtain $\hat{\theta} = (\hat{\theta}_0, \ldots, \hat{\theta}_W)$ for each window. For cascading bursts caused by user–content interactions, as we do not know the number of hashtags in advance, we apply the second method to obtain estimates $\hat{n}_+$, i.e., the number of hashtags with at least one influence triangle, and $\hat{\vartheta}^+ = (\hat{\vartheta}_1^+, \ldots, \hat{\vartheta}_W^+)$ for each window. Combining $\hat{n}_+$ with $\hat{\vartheta}^+$, we use $\hat{n}_+\hat{\vartheta}^+$, i.e., frequencies, to characterize patterns of user–content interactions in each window.

Results We first answer the question: are there significant differences for the two distributions before and during the movement? In Fig. 14, we compare the distributions before (Sept 1 to Sept 3) and during (Sept 28 to Sept 30) the movement. We can find that when the movement began on Sept 28, the distributions of the two kinds of interactions shift to the right, indicating that many interaction and influence triangles form when the movement starts. Therefore, these observations confirm our motivation for detecting bursts by tracking triadic cardinality distributions.

Next, we track the daily triadic cardinality distributions to look up the distribution change during the movement. To characterize the sudden change in the distributions, we use KL divergence to calculate the difference between $\hat{\theta}$ and a base distribution θ_{base}. The base distribution θ_{base} represents a distribution when the network is dormant, i.e., no bursts are occurring. Here we omit the technique details, and simply average the triadic cardinality distributions from Sept 1 to Sept 7 to obtain an approximate base distribution $\hat{\theta}_{\text{base}}$, and show the KL divergence $D_{\text{KL}}(\hat{\theta}_{\text{base}} \parallel \hat{\theta})$ in Fig. 15.

We find that the KL divergence exhibits a sudden increase on Sept 28 when the movement broke out. The movement keeps going on and reaches a peak on Oct 19 when repeated clashes happened in Mong Kok at that time. The movement temporally returned to peace between Oct 22 and Oct 25, and restarted again after Oct 26. In Fig. 15, we also show the estimated

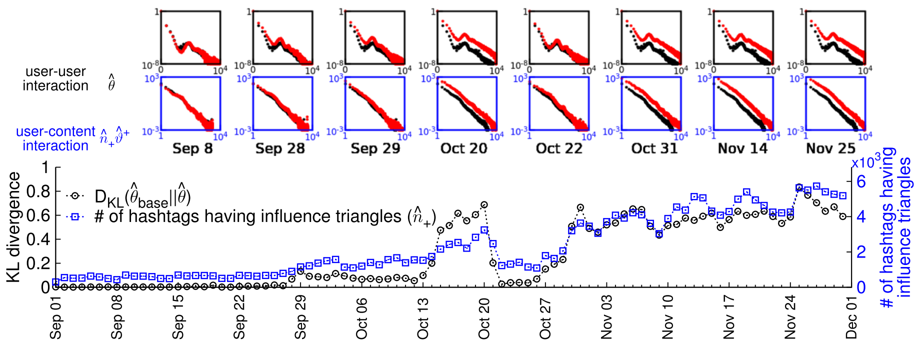


Fig. 15 Triadic cardinality distributions change during the Hong Kong occupy central movement in Twitter

number of hashtags having at least one influence triangle. Its trend is similar to the trend of KL divergence which indicates that the movement is accompanied with rumors spreading in a word-of-mouth manner.

In conclusion, the application in this section demonstrates that the using of the triadic cardinality distribution can track bursts from a social activity stream and the result is consistent with real-world events.

8 Conclusion

Online social networks provide various ways for users to interact with other users or media content over the Internet, which bridge the online and offline worlds tightly. This provides an opportunity to researchers to leverage online users' interactions to detect bursts that may cause negative impacts to the offline world. This work studied the burst detection problem from a high-speed social activity stream generated by user's interactions in an OSN. We show that the emergence of bursts caused by either user–user or user–content interaction is accompanied with the formation of triangles in users' interaction networks. This finding prompts us to devise a novel method for burst detection in OSNs by introducing the triadic cardinality distribution. Triadic cardinality distribution is found to be robust against common spamming attacks which makes it a more suitable indicator for detecting bursts than the volume of user activities. We design a sample-estimate solution that aims to estimate triadic cardinality distribution from a sampled social activity stream. We show that, in general, SGS is more efficient in gathering information from data than ITS-based methods. However, SGS incurs larger NRMSE than ITS for nodes with large triadic cardinalities. We can combine ITS and SGS and use a mixture estimator to further reduce the NRMSE of SGS at the tail estimates. We believe our work sets the foundation for robust burst detection, and it is an open problem for finding and designing better or optimal sampling and estimation methods.

There are several limitations in this work that offer opportunities for future research. First, in Sect. 5.1.1, we used a beta-binomial distribution to model sums of dependent Bernoulli trials as triangle samples are dependent in our problem. Note that beta-binomial distribution is just an approximate model, and it is worthy to find more accurate models. Second, the SGS method may sample many nodes with large degrees. Large degree nodes usually have large ego-networks (i.e., the graph representing the connections among the neighbors of a node), and counting triangles in a large ego-network will be time-consuming. As a result, SGS may be inefficient if many large degree nodes are sampled. There may be other ways to address this issue such as parallelly processing each ego-network and using sampling methods to estimate the number of triangles in a large ego-network [61]. We postpone the solution to this issue for future work. Third, in Sect. 7.1, we studied the impacts of spam through synthetically adding spam interactions to the interaction network, and concluded that triadic cardinality distribution is robust against the Random and random-friend spamming strategies. It is also important to verify this conclusion in real-world datasets, and we leave this for future work.

Acknowledgements This work was supported in part by the National Key R&D Program of China (2016YFE0206700), Shenzhen Basic Research Grant (JCYJ20170816100819428), Sichuan Science and Technology Program (2019YFG0407), and the National Natural Science Foundation of China (61922067, U1736205, 61902305).

References

1. Ahmed NK, Duffield N, Neville J, Kompella R (2014) Graph sample and hold: a framework for big-graph analytics. In: Proceedings of the 20th ACM SIGKDD international conference on knowledge discovery and data mining
2. Arifuzzaman S, Khan M, Vishnu M (2019) Fast parallel algorithms for counting and listing triangles in big graphs. ACM Trans Knowl Discov Data 14(1):1–34
3. Backstrom L, Bakshy E, Kleinberg J, Lento TM, Rosenn I (2011) Center of attention: How Facebook users allocate attention across friends. In: Proceedings of the 5th international AAAI conference on weblogs and social media
4. Barabasi AL (2005) The origin of bursts and heavy tails in human dynamics. Nature 435:207–211
5. Becchetti L, Boldi P, Castillo C, Gionis A (2008) Efficient semi-streaming algorithms for local triangle counting in massive graphs. In: Proceedings of the 14th ACM SIGKDD international conference on knowledge discovery and data mining
6. Beta-binomial distribution. https://en.wikipedia.org/wiki/Beta-binomial_distribution (Retrieved June 2020)
7. Beutel A, Xu W, Guruswami V, Palow C, Faloutsos C (2013) CopyCatch: stopping group attacks by spotting lockstep behavior in social networks. In: Proceedings of the 22nd international world wide web conference
8. Boshmaf Y, Muslukhov I, Beznosov K, Ripeanu M (2011) The socialbot network: When bots socialize for fame and money. In: Proceedings of the 27th annual computer security applications conference
9. Budak C, Agrawal D, Abbadi AE (2011) Structural trend analysis for online social networks. In: Proceedings of the VLDB endowment
10. Cadena J, Vullikanti A (2018) Mining heavy temporal subgraphs: Fast algorithms and applications. In: Proceedings of the 32nd AAAI conference on artificial intelligence
11. Chierichetti F, Kleinberg J, Kumar R, Mahdian M, Pandey S (2014) Event detection via communication pattern analysis. In: Proceedings of the 8th international AAAI conference on weblogs and social media
12. Chu Z, Gianvecchio S, Wang H, Jajodia S (2010) Who is tweeting on Twitter: Human, bot, or cyborg? In: Proceedings of the 26th annual computer security applications conference
13. Costa AF, Yamaguchi Y, Traina AJM Jr, Faloutsos CT (2017) Modeling temporal activity to detect anomalous behavior in social media. ACM Trans Knowl Discov Data 11(4):1–23
14. Cramér-Rao bound. https://en.wikipedia.org/wiki/Cram%C3%A9r%E2%80%93Rao_bound (Retrived June 2020)
15. Duffield N, Lund C, Thorup M (2003) Estimating flow distributions from sampled flow statistics. In: Proceedings of the ACM special interest group on data communication
16. Durand M, Flajolet P (2003) Loglog counting of large cardinalities. In: Proceedings of the 11th annual European symposium on algorithms
17. Eftekhar M, Koudas N, Ganjali Y (2013) Bursty subgraphs in social networks. In: Proceedings of the 6th international ACM conference on web search and data mining
18. Eswaran D, Faloutsos C, Guha S, Mishra N (2018) SpotLight: Detecting anomalies in streaming graphs. In: Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery and data mining

19. Gjoka M, Kurant M, Butts CT, Markopoulou A (2011) Practical recommendations on crawling online social networks. IEEE J Sel Areas Commun 29(9):1872–1892
20. Gorman JD, Hero AO (1990) Lower bounds for parametric estimation with constraints. IEEE Transit Inf Theory 26(6):1285–1301
21. Grier C, Thomas K, Paxson V, Zhang M (2010) @spam: The underground on 140 characters or less. In: Proceedings of the ACM SIGSAC conference on computer and communications security
22. Harvey M (2020) Fans mourn artist for whom it didn't matter if you were black or white. http://wayback.archive.org/web/20100531165925/http://www.timesonline.co.uk/tol/news/world/us_and_americas/article6580897.ece (Retrieved June 2020)
23. Jha M, Seshadhri C, Pinar A (2013) A space efficient streaming algorithm for triangle counting using the birthday paradox. In: Proceedings of the 19th ACM SIGKDD international conference on knowledge discovery and data mining
24. Kleinberg J (2002) Bursty and hierarchical structure in streams. In: Proceedings of the 8th ACM SIGKDD international conference on knowledge discovery and data mining
25. Klimt B, Yang Y (2004) The Enron corpus: A new dataset for email classification research. In: Proceeding of the European conference on machine learning and principles and practice of knowledge discovery in databases
26. Kossinets G, Watts DJ (2006) Empirical analysis of an evolving social network. Science 311(5757):88–90
27. Krikorian R (2020) New tweets per second record, and how! https://blog.twitter.com/engineering/en_us/a/2013/new-tweets-per-second-record-and-how.html (Retrived June 2020)
28. Leskovec J, McGlohon M, Faloutsos C, Glance N, Hurst M (2007) Cascading behavior in large blog graphs. In: Proceedings of the 7th SIAM international conference on data mining
29. Lim Y, Kang U (2015) MASCOT: Memory-efficient and accurate sampling for counting local triangles in graph streams. In: Proceedings of the 21st ACM SIGKDD international conference on knowledge discovery and data mining. Sydney, Australia
30. Liu X, Ge T, Wu Y (2019) Finding densest lasting subgraphs in dynamic graphs: a stochastic approach. In: Proceedings of the 35th IEEE international conference on data engineering
31. London riots: more than 2,000 people arrested over disorder. http://www.mirror.co.uk/news/uk-news/london-riots-more-than-2000-people-185548 (Retrived June 2020)
32. Manzoor E, Milajerdi, SM, Akoglu L (2016) Fast memory-efficient anomaly detection in streaming heterogeneous graphs. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining. San Francisco, California, USA
33. Mathioudakis M, Bansal N, Koudas N (2010) Identifying, attributing and describing spatial bursts. In: Proceedings of the VLDB endowment
34. Milo R, Shen-Orr S, Itzkovitz S, Kashtan N, Chklovskii D, Alon U (2002) Network motifs: simple building blocks of complex networks. Science 298(5594):824–827
35. Pagh R, Tsourakakis CE (2012) Colorful triangle counting and a MapReduce implementation. J Inf Process Lett 112(7):277–281
36. Paranjape A, Benson AR, Leskovec J (2017)Motifs in temporal networks. In: Proceedings of the 10th ACM international conference on web search and data mining
37. Parikh N, Sundaresan N (2008)Scalable and near real-time burst detection from ecommerce queries. In: Proceedings of the 14th ACM SIGKDD international conference on knowledge discovery and data mining
38. Pavan A, Tangwongsan K, Tirthapura S, Wu KL (2013)Counting and sampling triangles from a graph stream. In: Proceedings of the VLDB endowment
39. Ribeiro B, Towsley D, Ye T, Bolot JC (2006) Fisher information of sampled packets: An application to flow size estimation. In: Proceedings of the 6th ACM SIGCOMM conference on internet measurement
40. Rodrigues T, Benevenuto F, Cha M, Gummadi KP, Almeida V (2011) On word-of-mouth based discovery of the web. In: Proceedings of the 11th ACM SIGCOMM conference on internet measurement
41. Sakaki T, Okazaki M, Matsuo Y (2010) Earthquake shakes Twitter users: Real-time event detection by social sensors. In: Proceedings of the 19th international world wide web conference
42. Sanei-Mehri SV, Sarüyüce AE, Tirthapura S (2018) Butterfly counting in bipartite networks. In: Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery and data mining
43. Sarüyüce AE, Pinar A (2018) Peeling bipartite networks for dense subgraph discovery. In: Proceedings of the 11th international ACM conference on web search and data mining, pp. 504–512
44. Seshadhri C, Pinar A, Kolda TG (2013) Triadic measures on graphs: The power of wedge sampling. In: Proceedings of the 13th SIAM international conference on data mining
45. Shiels M (2020) Web slows after Jackson's death. http://news.bbc.co.uk/2/hi/technology/8120324.stm (Retrieved June 2020)
46. Shin K, Oh S, Kim J, Hooi B, Faloutsos C (2019) Fast, accurate and provable triangle counting in fully dynamic graph streams. ACM Trans Knowl Discov Data 14(2):1–39

47. Stefani LD, Epasto A, Riondato M, Upfal E (2016) TRIEST: Counting local and global triangles in fully-dynamic streams with fixed memory size. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining
48. Stringhini G, Kruegel C, Vigna G (2010) Detecting spammers on social networks. In: Proceedings of the 26th annual computer security applications conference
49. Takahashi T, Tomioka R, Yamanishi K (2011) Discovering emerging topics in social streams via link anomaly detection. In: Proceedings of the IEEE international conference on data mining
50. Teng X, Yan M, Ertugrul AM, Lin YR (2018) Deep into hypersphere: Robust and unsupervised anomaly discovery in dynamic networks. In: Proceedings of the 27th international joint conference on artificial intelligence
51. Thomas K, Grier C, Paxson V, Song D (2011) Suspended accounts in retrospect: an analysis of Twitter spam. In: Proceedings of the 11th ACM SIGCOMM conference on internet measurement
52. Trees HLV (2001) Detection, estimation, and modulation theory. Part I. Wiley, Hoboken
53. Tsotsis A (2020) First credible reports of Bin Laden's death spread like wildfire on Twitter. https://techcrunch.com/2011/05/01/news-of-osama-bin-ladens-death-spreads-like-wildfire-on-twitter (Retrived June 2020)
54. Tsourakakis CE, Kang U, Miller GL, Faloutsos C (2009) DOULION: Counting triangles in massive graphs with a coin. In: Proceedings of the 15th ACM SIGKDD international conference on knowledge discovery and data mining
55. Tune P, Veitch D (2011) Fisher information in flow size distribution estimation. IEEE Trans Inf Theory 57(10):7011–7035
56. Turkett W, Fulp E, Lever C, Edward Allan J (2011) Graph mining of motif profiles for computer network activity inference. In: Proceedings of the 7th workshop on mining and learning with graphs
57. Veitch D, Tune P (2015) Optimal sampling for the flow size distribution. IEEE Trans Inf Theory 61(6):3075–3099
58. Wang P, Guan X, Zhao J, Tao J, Qin T (2014) A new sketch method for measuring host connection degree distribution. IEEE Trans Inf Forensics Secur 9(6):948–960
59. Wang P, Lui JC, Ribeiro B, Towsley D, Zhao J, Guan X (2014) Efficiently estimating motif statistics of large networks. ACM Trans Knowl Discov Data 9(2):1–27
60. Watts DJ, Strogatz SH (1998) Collective dynamics of 'small-world' networks. Nature 393:440–442
61. Wu B, Yi K, Li Z (2016) Counting triangles in large graphs by random sampling. IEEE Trans Knowl Data Eng 28(8):2013–2026
62. Yi J (2005) Detecting buzz from time-sequenced document streams. In: Proceedings of the IEEE international conference on e-technology, e-commerce and e-service
63. Yoon M, Hooi B, Shin K, Faloutsos C (2019) Fast and accurate anomaly detection in dynamic graphs with a two-pronged approach. In: Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery and data mining
64. Yu C, Zelterman D (2002) Sums of dependent Bernoulli random variables and disease clustering. Stat Probab Lett 57(1):363–373
65. Yu M, Song C, Gu J, Liu M (2019) Distributed triangle counting algorithms in simple graph stream. In: Proceedings of the 25th IEEE international conference on parallel and distributed system
66. Zhao J, Lui JC, Towsley D, Wang P, Guan X (2015) Tracking triadic cardinality distributions for burst detection in social activity streams. In: ACM conference on online social networks
67. Zhu Y, Shasha D (2003) Efficient elastic burst detection in data streams. In: Proceedings of the 9th ACM SIGKDD international conference on knowledge discovery and data mining

Junzhou Zhao is currently an associate professor at the School of Cyber Science and Engineering, Xi'an Jiaotong University, Xi'an 710049, P. R. China. He holds a B.S. degree in Automation Science and Technology, and a Ph.D. degree in Control Science and Engineering, both from Xi'an Jiaotong University. His research interests mainly focus on data management and mining, high performance computation, machine learning, and data security.

Pinghui Wang is currently a professor with MOE KLINNS Lab, Xi'an Jiaotong University, Xi'an 710049, P. R. China. He received the B.S. degree in Information Engineering and the Ph.D. degree in Control Science and Engineering from Xi'an Jiaotong University, Xi'an, China, in 2006 and 2012, respectively. From 2012 to 2013, he was a Postdoctoral Fellow with the School of Computer Science, McGill University. From 2014 to 2015, he was a Researcher with HUAWEI Noah's Ark Lab, Hong Kong. His research interests include traffic measurement and abnormal detection.

Zhouguo Chen is currently a network security Senior Engineer at Science and Technology on Communication Security Laboratory, Chengdu, China. He received a Graduate degree in signal and information processing, University of Electronic Science and Technology of China (UESTC), in 2006. His research interests include Network Security, Big Data, and Network Forensics.

Jianwei Ding is a senior engineer in the Science and Technology on Communication Security Laboratory at 30th Research Institute of China Electronics Technology Group Corporation. He finished his Ph.D degree in computer science and technology department at Tsinghua University, China in 2016. His major research interests include dark web threat intelligence, malware analysis, intelligence security analysis and so on. He has totally published more than 10 papers, 3 patents for invention, and participated in more than 10 projects as core staff including national key research and development project, National Natural Science Foundation of China and so on.

John C.S. Lui received the PhD degree in computer science from UCLA. He is currently a professor in the Department of Computer Science and Engineering at The Chinese University of Hong Kong. His current research interests include communication networks, network system security (e.g., cloud security, mobile security, etc.), network economics, network sciences (e.g., online social networks, information spreading, etc.), cloud computing, large-scale distributed systems and performance evaluation theory. He serves in the editorial board of IEEE/ACM Transactions on Networking, IEEE Transactions on Computers, IEEE Transactions on Parallel and Distributed Systems, Journal of Performance Evaluation and International Journal of Network Security. He was the chairman of the CSE Department from 2005–2011. He received various departmental teaching awards and the CUHK Vice-Chancellor's Exemplary Teaching Award. He is also a co-recipient of the IFIP WG 7.3 Performance 2005 and IEEE–IFIP NOMS 2006 Best–Student Paper Awards. He is an elected member of the IFIP WG 7.3, fellow of the ACM, fellow of the IEEE, and croucher senior research fellow. His personal interests include films and general reading.

Don Towsley holds a B.A. in Physics (1971) and a Ph.D. in Computer Science (1975) from University of Texas. From 1976 to 1985 he was a member of the faculty of the Department of Electrical and Computer Engineering at the University of Massachusetts, Amherst. He is currently a Distinguished Professor at the University of Massachusetts in the Department of Computer Science. He has held visiting positions at IBM T.J. Watson Research Center, Yorktown Heights, NY; Laboratoire MASI, Paris, France; INRIA, Sophia-Antipolis, France; AT&T Labs-Research, Florham Park, NJ; and Microsoft Research Lab, Cambridge, UK. His research interests include networks and performance evaluation. He currently serves as Editor-in-Chief of IEEE/ACM Transactions on Networking and on the editorial boards of Journal of the ACM, and IEEE Journal on Selected Areas in Communications, and has previously served on numerous other editorial boards. He was Program Co-chair of the joint ACM SIGMETRICS and PERFORMANCE 92 conference and the Performance 2002 conference. He is a member of ACM and ORSA, and Chair of IFIP Working Group 7.3. He has received the 2007 IEEE Koji Kobayashi Award, the 2007 ACM SIGMETRICS Achievement Award, the 1998 IEEE Communications Society William Bennett Best Paper Award, and numerous best conference/workshop paper awards. Last, he has been elected Fellow of both the ACM and IEEE.

Xiaohong Guan received the B.S. and M.S. degrees in automatic control from Tsinghua University, Beijing, China, in 1982 and 1985, respectively, and the Ph.D. degree in electrical engineering from the University of Connecticut, Storrs, US, in 1993. From 1993 to 1995, he was a consulting engineer at PG&E. From 1985 to 1988, he was with the Systems Engineering Institute, Xi'an Jiaotong University, Xi'an, China. From January 1999 to February 2000, he was with the Division of Engineering and Applied Science, Harvard University, Cambridge, MA. Since 1995, he has been with the Systems Engineering Institute, Xi'an Jiaotong University, and was appointed Cheung Kong Professor of Systems Engineering in 1999, and dean of the School of Electronic and Information Engineering in 2008. Since 2001 he has been the director of the Center for Intelligent and Networked Systems, Tsinghua University, and served as head of the Department of Automation, 2003–2008. He is an Editor of IEEE Transactions on Power Systems and an Associate Editor of Automata. His research interests include allocation and scheduling of complex networked resources, network security, and sensor networks. He has been elected Fellow of IEEE.