Tracking Triadic Cardinality Distributions for Burst Detection in Social Activity Streams

Junzhou Zhao* John C.S. Lui† Don Towsley‡ Pinghui Wang§ Xiaohong Guan*

* Xi'an Jiaotong University, China
† The Chinese University of Hong Kong, Hong Kong
‡ University of Massachusetts at Amherst, US
§ Huawei Noah's Ark Lab, Hong Kong
{jzzhao, xhguan}@sei.xjtu.edu.cn, cslui@cse.cuhk.edu.hk, towsley@cs.umass.edu, wang.pinghui@huawei.com

ABSTRACT

In everyday life, we often observe unusually frequent interactions among people before or during important events, e.g., we receive/send more greetings from/to our friends on Christmas Day, than usual. We also observe that some videos suddenly go viral through people's sharing in online social networks (OSNs). Do these seemingly different phenomena share a common structure?

All these phenomena are associated with sudden surges of user activities in networks, which we call "*bursts*" in this work. We find that the emergence of a burst is accompanied with the formation of triangles in networks. This finding motivates us to propose a new method to detect bursts in OSNs. We first introduce a new measure, "*triadic cardinality distribution*", corresponding to the fractions of nodes with different numbers of triangles, i.e., triadic cardinalities, within a network. We demonstrate that this distribution changes when a burst occurs, and is naturally immunized against spamming social-bot attacks. Hence, by tracking triadic cardinality distributions, we can reliably detect bursts in OSNs. To avoid handling massive activity data generated by OSN users, we design an efficient sample-estimate solution to estimate the triadic cardinality distribution from sampled data. Extensive experiments conducted on real data demonstrate the usefulness of this triadic cardinality distribution and the effectiveness of our sample-estimate solution.

Categories and Subject Descriptors: J.4 [**Computer Applications**]: Social and Behavioral Sciences

General Terms: Design, Measurement

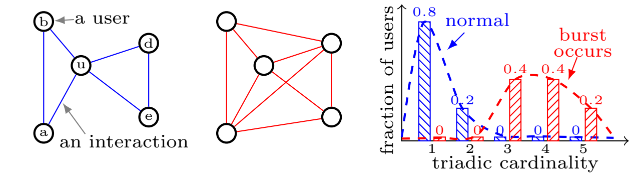
Keywords: Social Activity Streams, Burst Detection, Sampling Methods, Data Stream Algorithms

1. INTRODUCTION

Online social networks (OSNs) have become ubiquitous platforms that provide various ways for users to interact over the Internet, such as 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 [15], and the event even crashed Twitter for several minutes [30]. 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, resulted in 3,443 crimes across London due to this disorder [1]. Hence, detecting bursts in OSNs is an important task, both for OSN managers to monitor the operation status of an OSN, and 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 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 straightforward 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 [10, 14, 31, 7, 33, 6], which can suddenly generate a huge amount of spamming interactions in the OSN. Hence, this method can result in many *false alarms* due to the existence of social bots. Similar problem also exist when detecting bursts caused by user-content interactions. Many previous works on burst detection are based on idealistic assumptions [17, 39, 24, 13] and simply ignore the existence of social bots.

Present work. The primary goal of this work is to leverage a special *triangle structure*, which is a feature of humans, to design a robust burst detection method that is immunized against common social-bot attacks. We first describe the



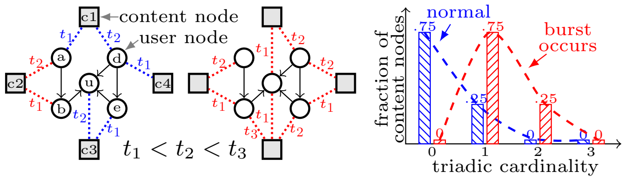
(a) normal (b) burst occurs (c) distribution changes

Figure 1: Interaction burst and interaction triangle

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 [38] because social networks exhibit many *triadic closures* [19]. This is due to the social phenomenon of "*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 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 to be a user-user interaction. This is illustrated in Fig. 1(a) when no interaction burst occurs, while in Fig. 1(b), 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 [7, 33].

Influence triangles in 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, a.k.a. the cascading diffusion [21] or word-of-mouth spreading [27]. 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 [34]. 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. 2(a), 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 a interacts with c is due to the influence of b on a. In Fig. 2(a), 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; while in Fig. 2(b), 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. 2(b) (from a previous state similar to Fig. 2(a)), he knows that a *cascading burst* is present in the OSN.



(a) normal (b) burst occurs (c) distribution changes

Figure 2: Cascading burst and influence triangle

Characterizing bursts. So far, we find a common structure shared by different types of bursts: the emergence of *interaction bursts* (caused by user-user interaction) and *cascading bursts* (caused by user-content interaction) 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. 1(a) or a content node in Fig. 2(a), 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, 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. 1(c) and 2(c). 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 can be considered as an abstraction of a *tweet stream* in Twitter, or a *news feed* in Facebook. We aim to calculate triadic cardinality distributions from this stream. However, when a network is large or users are very active, the social activity stream will be high speed. For example, the speed of the Twitter's tweets stream can be as high as $5,700$ 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 [20]. To handle such a high-speed social activity stream, we design a sample-estimate solution, which provides a *maximum likelihood estimate* of the triadic cardinality distribution using sampled data. Our method works in a near-real-time fashion, and is demonstrated to be accurate and efficient.

Overall, we make three contributions in this work.

- We propose a useful measure, triadic cardinality distribution, which provides succinct summary information to characterize burst patterns of user interactions in a large scale OSN (Section 2).
- We design a sample-estimate method that is able to accurately and efficiently estimate triadic cardinality distributions from high-speed social activity streams in near-real-time (Sections 3 and 4).
- Extensive experiments conducted on real data demonstrate the usefulness of the proposed triadic cardinality distribution and effectiveness of our sample-estimate solution. We also show how to apply our method to detect bursts in Twitter during the 2014 Hong Kong Occupy Central movement (Section 5).

2. PROBLEM FORMULATION

We first formally define the notion of social activity stream as mentioned in previous section. Then we define triadic cardinality distribution and describe our proposed solution.

2.1 Social Activity Stream

We represent an OSN by $G(V, E, C)$, where V is a set of users, E is a set of relationships among users, and C is a set of media content such as hashtags and video links. Here, a relationship 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 content in C. We denote a social activity by $a \in V \times (V \cup C) \times [0, \infty)$. Here user-user interaction, $a = (u, v, t)$, corresponds to user u interacting with user v at time t; and user-content interaction, $a = (u, c, t)$, corresponds to user u interacting with content c at time t.

These social activities are aggregated chronologically to form a *social activity stream*, denoted by $S = \{a_1, a_2, \ldots\}$, where a_k denotes the k-th social activity in the stream.

2.2 Triadic Cardinality Distribution

Triadic cardinality distributions are defined on two *interaction multi-graphs* which are formed by user-user and user-content interactions, respectively.

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

Similarly, user-content interactions also form a multi-graph $\mathcal{G}_{\text{uc}}(V \cup C, E \cup \mathcal{E}_{\text{uc}})$ in a time window. Unlike $\mathcal{G}_{\text{uu}}$, the node set includes both user nodes V and content nodes C, and the edge set includes user relations E and a multi-set $\mathcal{E}_{\text{uc}}$ denoting user-content 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 related to c in $\mathcal{G}_{\text{uc}}$. For example, in Fig. 2(a), content c_2 has triadic cardinality one, and all other content nodes have triadic cardinality zero.

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

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. By tracking triadic cardinality distributions, we will discover burst occurrences in an OSN.

2.3 Overview of Our Solution

We propose an on-line solution capable of tracking the triadic cardinality distribution from a high-speed social activity stream, as illustrated in Fig. 3.

Our solution consists of sampling a social activity stream in a time window maintaining only summary statistics, and constructing 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 is unnecessary to store all of the samples in the stream, and enables us to detect bursts in a near-real-time fashion.

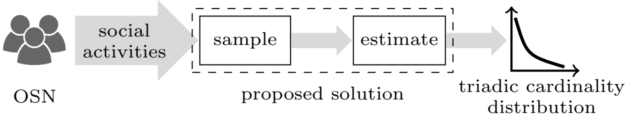


Figure 3: Overview of our sample-estimate solution

3. STREAM SAMPLING METHOD

In this section, we introduce the sampling method in our solution. The purpose of sampling is to reduce the computational cost in handling the massive amount of data in a high-speed social activity stream.

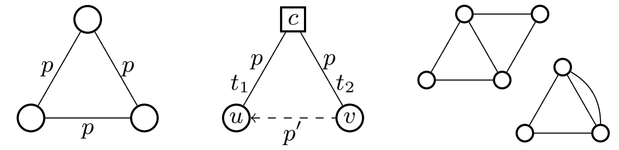
3.1 Sampling Stream with a Coin

The stream sampling method works as follows. We toss a biased coin for each 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. We use these samples to obtain a summary statistics of the stream in the current window, which we describe later.

3.2 Probability of Sampling a Triangle

When social activities in the stream are sampled, triangles in $\mathcal{G}_{\text{uu}}$ and $\mathcal{G}_{\text{uc}}$ are sampled accordingly. Obviously, the probability of sampling a triangle depends on p. In what follows, we analyze the relationship between triangle sampling probability and p, for an interaction triangle and an influence triangle, respectively.

Probability of sampling an interaction triangle. Sampling an interaction triangle, which consists of three user-user interaction edges in $\mathcal{E}_{\text{uu}}$, is equivalent to all its three edges being sampled. Because each interaction edge is independently sampled with probability p, then an interaction triangle is sampled with probability p^3, as illustrated in Fig. 4(a).



(a) interaction triangle (b) influence triangle ($t_1 < t_2$) (c) triangles having shared edges

Figure 4: Sampling triangles. A solid edge represents an interaction, and a dashed edge represents a user relation in E (i.e., a social edge). Figure (c) illustrates two cases of two interaction triangles having shared edges.

Probability of sampling an influence triangle. Calculating the probability of sampling an influence triangle is more complicated. First, we know that an influence triangle consists of two user-content interaction edges in $\mathcal{E}_{\text{uc}}$ and one social edge in E. Second, we note that stream sampling only applies to edges in $\mathcal{E}_{\text{uc}} \cup \mathcal{E}_{\text{uu}}$; edges in E are not sampled as they do not appear in the social activity stream.

In Fig. 4(b), suppose we have sampled two user-content interaction edges uc and vc, and assume user u interacted with content c earlier than user v. To determine whether content c has an influence triangle formed by 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 v follows u; or in Facebook, we query friends of v and check whether u is a friend of v.

Suppose we observe n_c sampled users that all interact with c during the current time window, denoted by $V_c = \{u_1, \ldots, u_{n_c}\}$ where u_i interacted with c earlier than u_j, $i < j$. To verify every sampled triangle related to c, we need to query the OSN $n_c(n_c - 1)/2$ times. This query cost is obviously expensive when n_c is large. To reduce this query cost, instead of checking every possible user pair, 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 having been sampled. Then, it is easy to see that an influence triangle is sampled with probability p^2p'.

We summarize the above discussions in Theorem 1.

THEOREM 1. *If we independently sample each social activity in stream S with probability p, and check the existence of a user relation in the OSN with probability p', then each interaction (influence) triangle in graph $\mathcal{G}_{uu}$ ($\mathcal{G}_{uc}$) is sampled with identical probability*

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

Remark. Although triangles of the same type are sampled identically, they may *not* be sampled *independently*, such as the cases two triangles have shared edges in Fig. 4(c). We will consider this issue in detail in Section 4.

3.3 Statistics of Sampled Data

The above sampling process is equivalent to sampling edges in multi-graphs $\mathcal{G}_{\mathrm{uu}}$ and $\mathcal{G}_{\mathrm{uc}}$: an activity edge $e \in \mathcal{E}_{\mathrm{uu}} \cup \mathcal{E}_{\mathrm{uc}}$ is independently sampled with probability p; a social edge $e' \in E$ is sampled with conditional probability p'.

At the end of the time window, we obtain two *sampled multi-graphs* $\mathcal{G}'_{\mathrm{uu}}$ and $\mathcal{G}'_{\mathrm{uc}}$[1]. Calculating the triadic cardinalities for nodes in these reduced graphs is much easier than on the original unsampled graphs. For $\mathcal{G}'_{\mathrm{uu}}$, we calculate triadic cardinality for each user node, and obtain statistics $g = (g_0, \ldots, g_M)$, where g_j, $0 \le j \le M$, denotes the number of nodes with j triangles in $\mathcal{G}'_{\mathrm{uu}}$. Similar statistics are also obtained from $\mathcal{G}'_{\mathrm{uc}}$, denoted by $f = (f_0, \ldots, f_{M'})$ (where f_j is the number of content nodes with j influence triangles in $\mathcal{G}'_{\mathrm{uc}}$). We only need to store g and f in computer memory and use them to estimate θ and ϑ in the next section.

4. ESTIMATION METHODS

We are now ready to 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 network flow size distribution estimation [11, 26] in which a packet in a flow is viewed to be a triangle of a node. However, in our case, triangle samples are not independent, and a node may have no triangles. These issues complicate estimation, and we will describe how to solve these issues in this section.

Note that we only discuss how to obtain the MLE of 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.

4.1 MLE when Graph Size is Known

Recall that g_j, $0 \le j \le M$, is the number of nodes with j sampled triangles in $\mathcal{G}'_{\mathrm{uu}}$. First, note that observing a node with j sampled triangles in $\mathcal{G}'_{\mathrm{uu}}$ implies that the node has at least j triangles in $\mathcal{G}_{\mathrm{uu}}$.

We also need to pay special attention to g_0, which is the number of nodes with no triangle in $\mathcal{G}'_{\mathrm{uu}}$. Due to sampling, some nodes may be unobserved (e.g., no edge attached to the node is sampled), and these unobserved nodes also have no sampled triangle. We include these in g_0; the advantage of this inclusion will be seen later. Since we have assumed a total of n nodes in $\mathcal{G}_{\mathrm{uu}}$, the number of unobserved nodes is $n - \sum_{j=0}^{M} g_j$. Therefore, we calibrate g_0 by

$$g_0 \triangleq n - \sum_{j=1}^{M} g_j.$$

Our goal is to derive an MLE of θ. To this end, we need to model the sampling process. For a randomly chosen node, let X denote the number of triangles to which it belongs in $\mathcal{G}_{\mathrm{uu}}$, and let Y denote the number of triangles observed during sampling. Then $P(Y = j|X = i), 0 \le j \le i$, is the conditional probability that a node has j sampled triangles in $\mathcal{G}'_{\mathrm{uu}}$ given that it has i triangles in $\mathcal{G}_{\mathrm{uu}}$. The sampling of a triangle can be viewed as a Bernoulli trial with a success probability of p_δ, according to Theorem 1. If Bernoulli trials are independent, which means triangles are independently sampled, then $P(Y = j|X = i)$ follows a binomial distribution. However, independence does not hold for triangles having shared edges, as illustrated in Fig. 4(c). As a result, it is non-trivial to derive $P(Y = j|X = i)$ with the existence of dependence. To deal with this dependence, we approximate sums of dependent Bernoulli random variables by a Beta-binomial distribution [40], which yields

$$\begin{aligned} P(Y = j|X = i) &= BetaBin(j|i, p_\delta/\alpha, (1 - p_\delta)/\alpha) \\ &= \binom{i}{j} \frac{\prod_{s=0}^{j-1}(s\alpha + p_\delta) \prod_{s=0}^{i-j-1}(s\alpha + 1 - p_\delta)}{\prod_{s=0}^{i-1}(s\alpha + 1)} \triangleq b_{ji}(\alpha) \end{aligned}$$

where $\prod_0^{-1} \triangleq 1$. The above Beta-binomial distribution parameterized by α allows pairwise identically distributed Bernoulli trials to have covariance $\alpha p_\delta(1 - p_\delta)/(1 + \alpha)$. It reduces to a binomial distribution when $\alpha = 0$. We have carried out χ^2 goodness-of-fit tests and the results demonstrate that the above model indeed fits well the observed data on many graphs (and is always better than the binomial model, of course).

Using this model, we easily obtain the likelihood of observing a node to have j sampled triangles, i.e.,

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

Then, the log-likelihood of all observations $\{Y_k = y_k\}_{k=1}^{n}$,

[1] In $\mathcal{G}'_{\mathrm{uc}}$, each sampled social edge e' needs to be marked with the influence triangle which e' belongs to, corresponding to the two user-content interactions that e' is checked for.

where $Y_k = y_k$ denotes the k-th node having y_k sampled triangles, yields

$$\mathcal{L}(\theta, \alpha) \triangleq \log P(\{Y_k = y_k\}_{k=1}^{n}) = \sum_{j=0}^{M} g_j \log \sum_{i=j}^{W} b_{ji}(\alpha)\theta_i. \tag{2}$$

The MLE of θ can then be obtained by maximizing (2) with respect to θ and α under the constraint that $\sum_{i=0}^{W} \theta_i = 1$. Note that this is non-trivial due to the summation inside the log operation. In the next subsection, we use the expectation-maximization (EM) algorithm to obtain the MLE in a more convenient way.

4.2 EM Algorithm when Graph Size is Known

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

$$\begin{aligned} P(\{(Y_k, X_k)\}_{k=1}^{n}) &= \prod_{k=1}^{n} P(Y_k = y_k, X_k = x_k) \\ &= \prod_{j=0}^{M} \prod_{i=j}^{W} P(Y = j, X = i)^{z_{ij}} = \prod_{j=0}^{M} \prod_{i=j}^{W} [b_{ji}(\alpha)\theta_i]^{z_{ij}} \end{aligned}$$

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

$$\mathcal{L}_c(\theta, \alpha) \triangleq \sum_{j=0}^{M} \sum_{i=j}^{W} z_{ij} \log [b_{ji}(\alpha)\theta_i]. \tag{3}$$

Here, we can treat $\{X_k\}_{k=1}^{n}$ 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. (3) with respect to hidden variables $\{X_k\}_k$, conditioned on data $\{Y_k\}_k$ and previous estimates $\theta^{(t)}$ and $\alpha^{(t)}$. That is

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

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

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

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

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

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

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

Multiple iterations of the E-step and the M-step, EM algorithm converges to a solution, which is a local maximum of (2). We denote this solution by $\hat{\theta}$ and $\hat{\alpha}$.

4.3 MLE when Graph Size is Unknown

When the graph size is unknown, one can use probabilistic counting methods such as loglog counting [12] 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 we do not know the number of unsampled nodes. A node of degree d is not sampled with probability $(1-p)^d$. 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, and this complicates estimation design.

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

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

$$q_i(\alpha) \triangleq P(Y = 0 | X = i) = \prod_{s=0}^{i-1} \left(1 - \frac{p_\delta}{s\alpha + 1}\right).$$

Then, the probability that a node has triangles in $\mathcal{G}_{\mathrm{uu}}$, 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 $\sum_{j=1}^{M} g_j$ nodes having been observed to have at least one sampled triangle. Hence, n_+ can be estimated by

$$\hat{n}_+ = \frac{\sum_{j=1}^{M} g_j}{1 - q(\theta^+, \alpha)}. \tag{4}$$

Note that estimator (4) relies on θ^+ and α, and we can obtain them using the following procedure.

Estimating θ^+ and α. We discard g_0 and only use $g^+ \triangleq (g_1, \ldots, g_M)$ 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., $\{Y_k = y_k \colon y_k \ge 1\}$. In this case, the probability that a node has $X = i$ triangles, and $Y = j$ of them are sampled, conditioned on $Y \ge 1$, is

$$\begin{aligned} &P(Y = j | X = i, Y \ge 1) \\ &\quad = \frac{\mathit{BetaBin}(j | i, p_\delta/\alpha, (1-p_\delta)/\alpha)}{1 - \mathit{BetaBin}(0 | i, p_\delta/\alpha, (1-p_\delta)/\alpha)} \triangleq a_{ji}(\alpha), \; j \ge 1. \end{aligned}$$

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

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

where

$$\phi_i \triangleq P(X=i|Y \geq 1) = \frac{\theta_i^+[1-q_i(\alpha)]}{\sum_{i'=1}^{W} \theta_{i'}^+[1-q_{i'}(\alpha)]}, \quad i \geq 1. \quad (5)$$

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:

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

where

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

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

Once EM converges, we obtain estimates $\hat{\phi}$ and $\hat{\alpha}$. The estimate for θ^+ is then obtained by Eq. (5), i.e.,

$$\hat{\theta}_i^+ = \frac{\hat{\phi}_i/[1-q_i(\hat{\alpha})]}{\sum_{i'=1}^{W} \hat{\phi}_{i'}/[1-q_{i'}(\hat{\alpha})]}, \quad 1 \leq i \leq W. \quad (6)$$

Finally, $\hat{n}_+$ is obtained by the estimator in Eq. (4).

5. EXPERIMENTS

In this section, we first empirically verify the claims we have made. Then, we validate the proposed estimation methods on several real-world networks. Finally, we illustrate our method to detect bursts in Twitter during the 2014 Hong Kong Occupy Central movement.

5.1 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.

Enron email dataset. The Enron email dataset [18] 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 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.

Observations from data. Because the data has been cleaned, the number of user-user interactions (i.e., number of sent emails[2]) 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 volum 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.

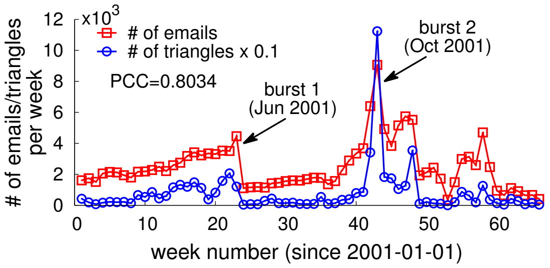
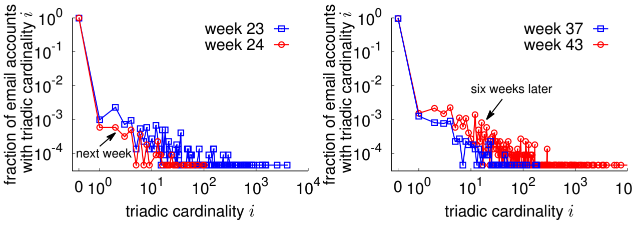


Figure 5: Email and triangle volumes per week

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 of email communications from week 23 to week 24, we observe in Fig. 6(a) that the distribution shifts to the left. While for the second burst, due to the gradual increase of email communications, we observe in Fig. 6(b) 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.



(a) Burst 1 shifts the distribution to left.

(b) Burst 2 shifts the distribution to right.

Figure 6: Bursts change distribution curves.

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:

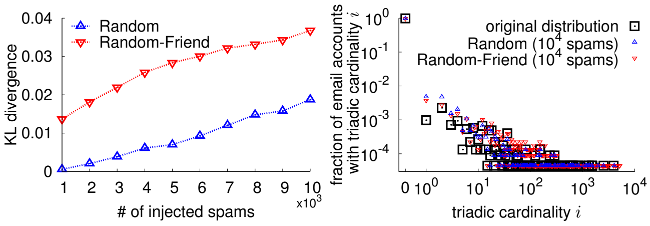
- *Random*: The spammer randomly chooses many target users to send spam.
- *Random-Friend*: At each step, the spammer randomly chooses a user and a random friend of the user[3], as two

[2] If an email has x recipients, we count it x times.

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

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 between the original and distorted distributions. The relationship between KL divergence and the number of injected spams is shown in Fig. 7(a). 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. 7(b), 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.



(a) KL divergence (b) Distorted distributions

Figure 7: Impacts of spam.

5.2 Validating Estimation Methods

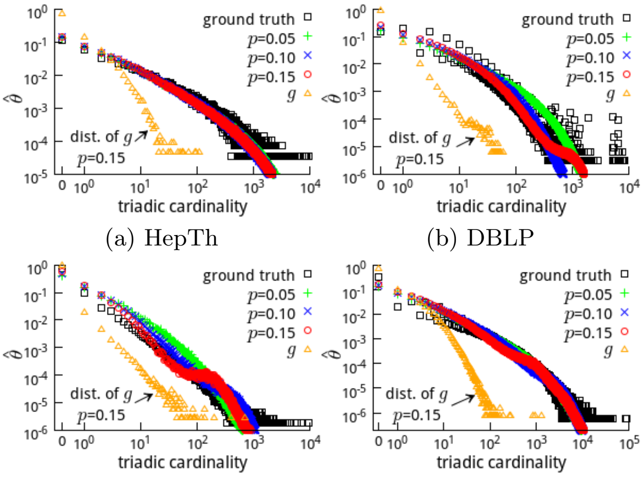
In the second experiment, we demonstrate that our proposed estimation methods produce good estimates of triadic cardinality distributions using sampled data while reducing computational cost.

Datasets. Because the input of our estimation methods is in fact a sampled graph, we use several public available graphs of different types and scales from the SNAP graph repository (`snap.stanford.edu/data`) as our testbeds. We summarize statistics of these graphs in Table 1.

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

For each graph, we sample an edge with probability p, and obtain a sampled graph. We then calculate the triadic cardinality for each node 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.



(a) HepTh (b) DBLP

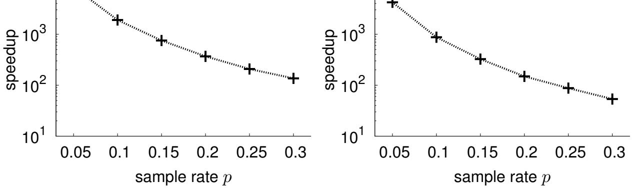
(c) YouTube (d) Pokec

Figure 8: Estimates of θ when graph size is known. $\hat{\alpha}$ corresponding to each graph and p is typically small, ranging from 0.00015 to 0.028. $W = 10^4$ and each result is averaged over 100 runs.

Validation when graph size is known. We first evaluate the estimation method when the graph size is known in advance, as is the assumption of our first method. The first method outputs estimate $\hat{\theta} = (\hat{\theta}_0, \ldots, \hat{\theta}_W)$.

The estimates on the four graphs and comparisons with ground truth distributions are depicted in Fig. 8. For each graph, we set $p = 0.05, 0.1$ and 0.15, respectively. From these results, we show that when more data is sampled the estimate generally improves, but even when $p = 0.05$ is sufficient to obtain a good estimate. The sampled triadic cardinality distribution of g for $p = 0.15$ is also shown for each graph. It is clear to see that the estimator has the ability to "correct" this distribution to approach the ground truth distribution.

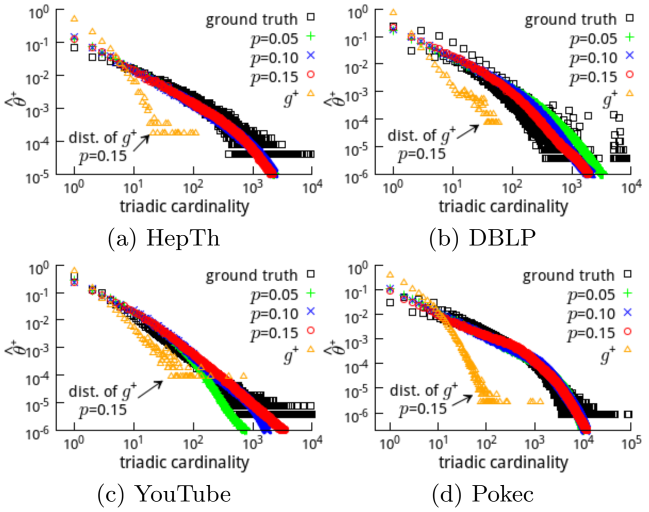
We also compare the computational efficiency of our sampling approach against a naive method that uses all of the original graph to calculate θ in an exact fashion. The results are depicted in Fig. 9. Obviously, the naive method is very inefficient and our sample-estimate solution is at least about 50 times faster with $p = 0.3$ on all of the four graphs.



(a) HepTh (b) DBLP

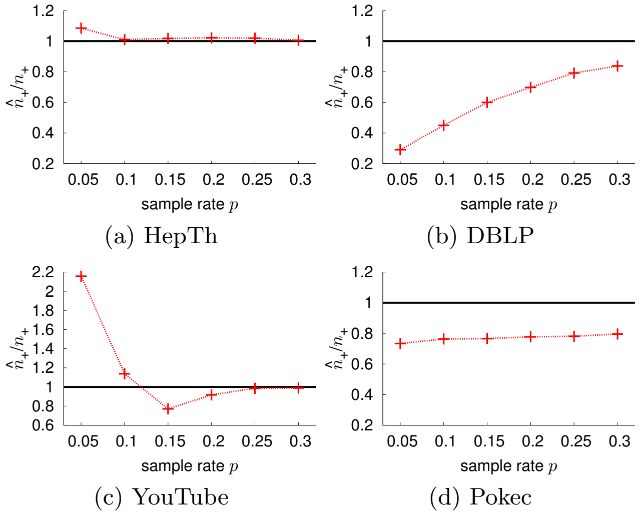
(c) YouTube (d) Pokec

Figure 9: Computational efficiency comparison



(a) HepTh (b) DBLP (c) YouTube (d) Pokec

Figure 10: Estimates of θ^+ when $|V|$ is unknown. $\hat{\alpha}$ for each graph and p ranges from 0.0001 to 0.01. $W = 10^4$ and each result is averaged over 100 runs.



(a) HepTh (b) DBLP (c) YouTube (d) Pokec

Figure 11: Estimates of n_+. $W = 10^4$ and each result is averaged over 100 runs.

Validation when graph size is unknown. When the graph size is unknown, the second method in Subsection 4.3 provides estimates for the number of nodes with at least one triangle in the graph $\hat{n}_+$ and triadic cardinality distribution $\hat{\theta}^+ = (\hat{\theta}_1^+, \ldots, \hat{\theta}_W^+)$ for the nodes with at least one triangle.

The results are shown in Fig. 10, using three sample rates $p = 0.05, 0.1$ and 0.15, respectively. It is clear that the second method also provides good estimates. Using a fraction of 5% of the data is sufficient to obtain good estimates. The computational efficiency is similar to results depicted in Fig. 9.

The estimate of n_+ for each graph is shown in Fig. 11. Because the majority of the nodes have small triadic cardinalities, good estimates of θ_i^+ for small values of i are critical for a good estimate of n_+ using estimator (4). For the HepTh graph, estimate $\hat{n}_+$ is very accurate even with small p. While for the other three graphs, accurate estimates of n_+ require relatively large sample rates, and $\hat{n}_+$ is usually an underestimate of n_+ on DBLP and Pokec due to a slight underestimate of θ_i^+ for small values of i on the two graphs. Nevertheless, using a sample rate $p = 0.3$, the relative estimation error for $\hat{n}_+$ is less than 20% for all four graphs. The design of a better estimator for n_+ is left for future work.

5.3 Application: Burst Detection in 2014 Hong Kong Occupy Central Movement

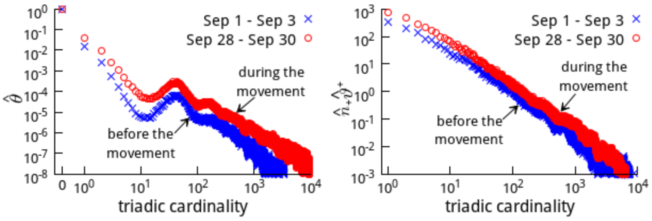
In the third experiment, we apply our solution to detect bursts in Twitter during the 2014 Hong Kong Occupy Central movement.

2014 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 the movement was still ongoing while we were collecting the data.

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 users using a set of Twitter accounts, called *detectors*, and aggregating tweets received by detectors (i.e., detectors' timelines) 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, 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. In a time window, we sample each social activity with probability $p = 0.3$ and check a social relation with probability $p' = 0.3$. 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. For both $\hat{\theta}$ and $\hat{\vartheta}^+$, W is set to be 10^4.

Results. We first answer the question: are there significant differences for the two distributions before and during the movement? In Fig. 12, 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.



(a) Interaction burst

(b) Cascading burst

Figure 12: Triadic cardinality distributions before and during the movement.

Next, we track the daily triadic cardinality distributions for the purpose of burst detection. 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. For simplicity, we 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. 13.

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. 13, we also show the estimated 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 is very useful for detecting bursts from social activity streams.

6. RELATED WORK

Kleinberg first studied this topic in [17], where he used a multi-state automaton to model a stream consisting of messages. 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 are proposed for detecting bursts from document streams [39, 22], e-commerce queries [24], time series [41], and social networks [13]. Although these models are theoretically interesting, some assumptions made by them are inappropriate, such as the Poisson process of message arrivals (see [4]) and nonexistence of spams/bots, which may limit their practical usage.

The topic of event detection is also related to our work. Recently, Chierichetti et al. [9] 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. [32] proposed a probabilistic model to detect emerging topics in Twitter by assigning an anomaly score for each user. Sakaki et al. [28] proposed a spatiotemporal model to detect earthquakes using tweets. Different from theirs, we exploit the triangle structure existing in user interactions which is robust against common spams and can be efficiently estimated using our method.

The triangle structure can be considered as a type of network motif, which is introduced in [23] when the authors were studying how to characterize structures of different types of networks. Turkett et al. [36] used motifs to analyze computer network usage, and [37] proposed sampling methods to efficiently estimate motif statistics in a large graph. However, both the motivation in [36] and subgraph statistics defined in [37] are different from ours.

Recently, there are many works on estimating the number of triangles [35, 8, 25, 16, 2] or clustering coefficient [29] in a large graph. However, these methods cannot be used to estimate the triadic cardinality distribution. Becchetti et al. [5] used a min-wise hashing method to approximately count triangles for each individual node in an undirected simple graph. 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, which is demonstrated to be efficient and accurate.

7. 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 user interactions so as to detect bursts that may cause impact to the offline world. We find that the emergence of bursts caused by either user-user interaction or user-content interaction are accompanied with the formation of triangles in users' interaction networks. This finding prompts us to devise a new method for burst detection in OSNs by introducing the triadic cardinality distribution. Triadic cardinality distribution is demonstrated to be robust against common spams which makes it a more suitable indicator for detecting bursts than the volume of user activities. We design a sample-estimate solution that can efficiently and accurately estimate triadic cardinality distribution from high-speed social activity streams in a near-real-time fashion, which makes it applicable in practice.

8. REFERENCES

[1] London riots: More than 2,000 people arrested over disorder. `http://goo.gl/4iBU3t`, Aug. 2011.

[2] N. K. Ahmed, N. Duffield, J. Neville, and R. Kompella. Graph sample and hold: A framework for big-graph analytics. In *KDD*, 2014.

[3] L. Backstrom, E. Bakshy, J. Kleinberg, T. M. Lento, and I. Rosenn. Center of attention: How Facebook users allocate attention across friends. In *ICWSM*, 2011.

[4] A.-L. Barabasi. The origin of bursts and heavy tails in human dynamics. *Nature*, 435:207–211, 2005.

[5] L. Becchetti, P. Boldi, C. Castillo, and A. Gionis. Efficient semi-streaming algorithms for local triangle counting in massive graphs. In *KDD*, 2008.

[6] A. Beutel, W. Xu, V. Guruswami, C. Palow, and C. Faloutsos. CopyCatch: Stopping group attacks by spotting lockstep behavior in social networks. In *WWW*, 2013.

[7] Y. Boshmaf, I. Muslukhov, K. Beznosov, and M. Ripeanu. The socialbot network: When bots socialize for fame and money. In *ACSAC*, 2011.

[8] C. Budak, D. Agrawal, and A. E. Abbadi. Structural trend analysis for online social networks. In *VLDB*, 2011.

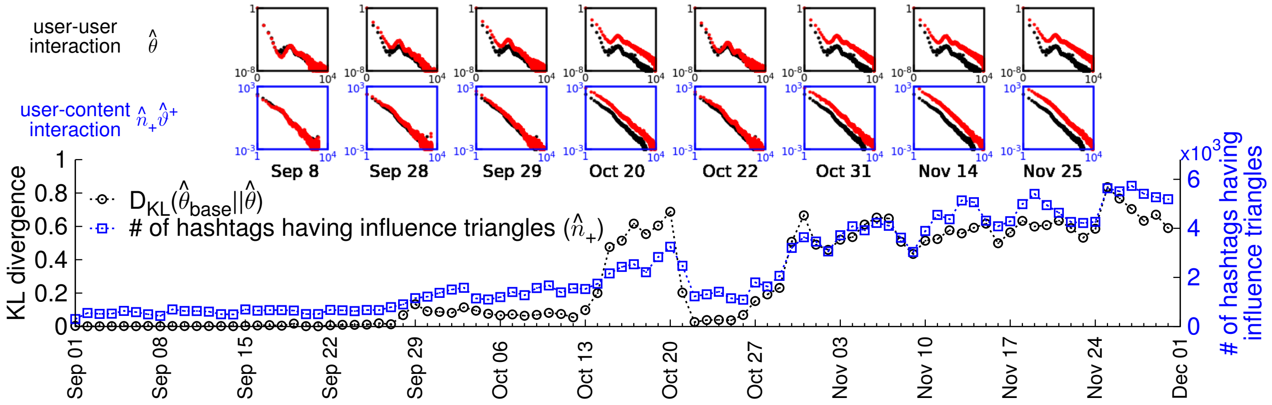


Figure 13: Burst detection during the 2014 Hong Kong Occupy Central movement in Twitter

[9] F. Chierichetti, J. Kleinberg, R. Kumar, M. Mahdian, and S. Pandey. Event detection via communication pattern analysis. In *ICWSM*, 2014.

[10] Z. Chu, S. Gianvecchio, H. Wang, and S. Jajodia. Who is tweeting on Twitter: Human, bot, or cyborg? In *ASCAC*, 2010.

[11] N. Duffield, C. Lund, and M. Thorup. Estimating flow distributions from sampled flow statistics. In *SIGCOMM*, 2003.

[12] M. Durand and P. Flajolet. Loglog counting of large cardinalities. In *ESA*, 2003.

[13] M. Eftekhar, N. Koudas, and Y. Ganjali. Bursty subgraphs in social networks. In *WSDM*, 2013.

[14] C. Grier, K. Thomas, V. Paxson, and M. Zhang. @spam: The underground on 140 characters or less. In *CCS*, 2010.

[15] M. Harvey. Fans mourn artist for whom it didn't matter if you were black or white. `http://goo.gl/5tCnu6`, June 2009.

[16] M. Jha, C. Seshadhri, and A. Pinar. A space efficient streaming algorithm for triangle counting using the birthday paradox. In *KDD*, 2013.

[17] J. Kleinberg. Bursty and hierarchical structure in streams. In *KDD*, 2002.

[18] B. Klimt and Y. Yang. The Enron corpus: A new dataset for email classification research. In *ECML*, 2004.

[19] G. Kossinets and D. J. Watts. Empirical analysis of an evolving social network. *Science*, 311(5757):88–90, 2006.

[20] R. Krikorian. New tweets per second record, and how! `http://goo.gl/I2m29h`, Aug. 2013.

[21] J. Leskovec, M. McGlohon, C. Faloutsos, N. Glance, and M. Hurst. Cascading behavior in large blog graphs. In *SDM*, 2007.

[22] M. Mathioudakis, N. Bansal, and N. Koudas. Identifying, attributing and describing spatial bursts. In *VLDB*, 2010.

[23] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon. Network motifs: Simple building blocks of complex networks. *Science*, 298(5594):824–827, 2002.

[24] N. Parikh and N. Sundaresan. Scalable and near real-time burst detection from ecommerce queries. In *KDD*, 2008.

[25] A. Pavan, K. Tangwongsan, S. Tirthapura, and K.-L. Wu. Counting and sampling triangles from a graph stream. In *VLDB*, 2013.

[26] B. Ribeiro, D. Towsley, T. Ye, and J. C. Bolot. Fisher information of sampled packets: An application to flow size estimation. In *IMC*, 2006.

[27] T. Rodrigues, F. Benevenuto, M. Cha, K. P. Gummadi, and V. Almeida. On word-of-mouth based discovery of the web. In *IMC*, 2011.

[28] T. Sakaki, M. Okazaki, and Y. Matsuo. Earthquake shakes Twitter users: Real-time event detection by social sensors. In *WWW*, 2010.

[29] C. Seshadhri, A. Pinar, and T. G. Kolda. Triadic measures on graphs: The power of wedge sampling. In *SDM*, 2013.

[30] M. Shiels. Web slows after Jackson's death. `http://goo.gl/HvXwdF`, June 2009.

[31] G. Stringhini, C. Kruegel, and G. Vigna. Detecting spammers on social networks. In *ACSAC*, 2010.

[32] T. Takahashi, R. Tomioka, and K. Yamanishi. Discovering emerging topics in social streams via link anomaly detection. In *ICDM*, 2011.

[33] K. Thomas, C. Grier, V. Paxson, and D. Song. Suspended accounts in retrospect: An analysis of Twitter spam. In *IMC*, 2011.

[34] A. Tsotsis. First credible reports of Bin Laden's death spread like wildfire on Twitter. `http://goo.gl/0Xlnqs`, May 2011.

[35] C. E. Tsourakakis, U. Kang, G. L. Miller, and C. Faloutsos. DOULION: Counting triangles in massive graphs with a coin. In *KDD*, 2009.

[36] W. Turkett, E. Fulp, C. Lever, and J. Edward Allan. Graph mining of motif profiles for computer network activity inference. In *MLG*, 2011.

[37] P. Wang, J. C. Lui, B. Ribeiro, D. Towsley, J. Zhao, and X. Guan. Efficiently estimating motif statistics of large networks. *TKDD*, 9(2):1–27, 2014.

[38] D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393:440–442, 1998.

[39] J. Yi. Detecting buzz from time-sequenced document streams. In *EEE*, 2005.

[40] C. Yu and D. Zelterman. Sums of dependent Bernoulli random variables and disease clustering. *Statistics and Probability Letters*, 57(1):363–373, 2002.

[41] Y. Zhu and D. Shasha. Efficient elastic burst detection in data streams. In *KDD*, 2003.