

Minfer: A Method of Inferring Motif Statistics From Sampled Edges

Pinghui Wang

MOE Key Laboratory for Intelligent Networks and Network Security
Xi'an Jiaotong University, Xi'an, China
Email: phwang@mail.xjtu.edu.cn

John C.S. Lui

Department of Computer Science and Engineering
The Chinese University of Hong Kong, Hong Kong
Email: cslui@cse.cuhk.edu.hk

Don Towsley

Department of Computer Science
University of Massachusetts Amherst, MA, USA
Email: towsley@cs.umass.edu

Junzhou Zhao

MOE Key Laboratory for Intelligent Networks and Network Security
Xi'an Jiaotong University, Xi'an, China
Email: jzzhao@sei.xjtu.edu.cn

Abstract—Characterizing motif (i.e., locally connected sub-graph patterns) statistics is important for understanding complex networks such as online social networks and communication networks. Previous work made the strong assumption that the graph topology of interest is known in advance. In practice, sometimes researchers have to deal with the situation where the graph topology is unknown because it is expensive to collect and store all topological and meta information. Hence, typically what is available to researchers is only a snapshot of the graph, i.e., a subgraph of the graph. Crawling methods such as breadth first sampling can be used to generate the snapshot. However, these methods fail to sample a streaming graph represented as a high speed stream of edges. Therefore, graph mining applications such as network traffic monitoring use random edge sampling (i.e., sample each edge with a fixed probability) to collect edges and generate a sampled graph, which we called a “RESampled graph”. Clearly, a RESampled graph’s motif statistics may be quite different from those of the underlying original graph. To resolve this, we propose a framework and implement a system called Minfer, which takes the given RESampled graph and accurately infers the underlying graph’s motif statistics. We also apply Fisher information to bound the errors of our estimates. Experiments using large scale datasets show the accuracy and efficiency of our method.

I. INTRODUCTION

Complex networks are widely studied across many fields of science and technology, from physics to biology, and from nature to society. Networks which have similar global topological features such as degree distribution and graph diameter can exhibit significant differences in their local structures. There is a growing interest to explore these local structures (also known as “*motifs*”), which are small connected and *induced* subgraph (or CIS) patterns that form during the growth of a network. For a set of nodes in the graph G of interest, its induced subgraph is defined as a graph that consists of all of the edges that connect them in G . Motifs have many applications, for example, they are used to characterize communication and evolution patterns in online social networks (OSNs) [1]–[4], pattern recognition in gene expression profiling [5], protein-protein interaction prediction [6], and coarse-grained topology generation of networks [7]. For instance, 3-node motifs can reveal relationships like “*the friend of my friend is my friend*”

and “*the enemy of my enemy is my friend*”, which are well known evolution patterns in signed (i.e., friend/foe) social networks. Kunegis et al. [2] considered the significance of motifs in Slashdot Zoo¹ and how they impact the stability of signed networks. Other more complex examples include 4-node motifs such as bi-fans and bi-parallels defined in [8].

Although motifs are important in helping researchers understand the underlying network, one major technical hurdle is that it is computationally expensive to enumerate and count all CISes in a large network. Note that there exist a large number of CISes even for a medium size network with less than one million edges. For example, the graphs Slashdot [9] and Epinions [10], which contain approximately 1.0×10^5 nodes and 1.0×10^6 edges, contain more than 2.0×10^{10} 4-node connected and induced subgraphs [11]. To address this problem, several sampling methods have been proposed to estimate the frequency distribution of motifs [11]–[14]. However, all previous methods focus on designing *computationally efficient* methods to characterize motifs when the *entire graph* of interest either fits into memory, or an I/O efficient neighbor query API exists to allow one to explore the graph topology, when it is stored on disk. In summary, *these methods assume that the entire graph topology is known*.

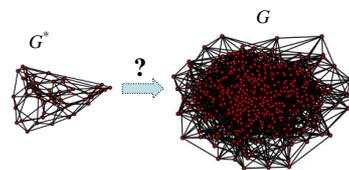


Figure 1. An example of the available G^* and the underlying graph G .

In practice the graph of interest may not be known, and the available dataset is just a subgraph sampled from the original graph, because it is expensive to collect and store all topological and meta information. A simple example is given in Fig. 1, where the sampled graph G^* is derived from the dataset representing G . G^* can be generated by crawling

¹www Slashdot.org

methods such as breadth first sampling. However, these methods fail to sample a streaming graph represented as a high speed stream of edges. In this work, we assume the available graph G^* is an RESampled graph that is obtained through random edge sampling, i.e, each edge in G is independently sampled with the same probability $0 \leq p \leq 1$. In practice, this sampling method is popular and easy to implement for streaming graphs. Obtaining a RESampled graph is easy and cheap (the computational and space complexities are both $O(1)$). A RESampled graph can also be used to estimate many other graph statistics such as average node degree, node label distribution, and edge label distribution, which have been studied in previous work [15]. These properties make the random edge sampling technique suitable for the following applications.

- **Network Traffic Analysis.** Network traffic on network devices such as routers can be represented as a sequence of network packets/flows. Sampling is inevitable for collecting network traffic on backbone routers in order to study the network graph, where a node in the graph represents a host and an edge (u, v) represents a connection from host u to host v , because packets go through routers at too high a rate to gather information from all packets. Therefore, current network devices support simple sampling techniques such as random packets/flows sampling, where random flow sampling can be viewed as random edge sampling over the network graph.
- **Network Data Publishing.** It is common for service providers to release a small sampled dataset (e.g., a RESampled graph) for a third-party research.

Formally, we denote the graph G^* as a RESampled graph of G . One easily observes that a RESampled graph's motif statistics differ from those of the original graph due to uncertainties introduced by sampling. For example, Fig. 2 shows that s^* is a 4-node induced subgraph in the RESampled graph G^* , but we do not know which original induced subgraph s in G it derives from. In fact, s could be any one of the five subgraphs depicted in Fig. 2.

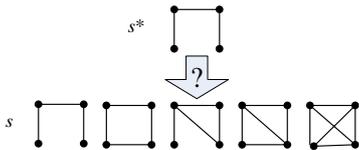


Figure 2. s^* is a 4-node induced subgraph in the RESampled graph G^* , and s is the original induced subgraph of s^* in the original graph G .

Unlike previous methods [11]–[14], in this paper we assume that it is impossible or computationally expensive to apply graph traversal algorithms over G and we aim to design an *accurate method* to infer the motif statistics of the original graph G from an available RESampled graph G^* . Note that previous methods focus on designing computationally efficient sampling/crawling methods based on sampling *induced* subgraphs in G to avoid the problem shown in Fig. 2. Hence they fail to infer the underlying graph's motif statistics from the given RESampled graph. The gSH method in [16] can be used to estimate the number of connected subgraphs from sampled

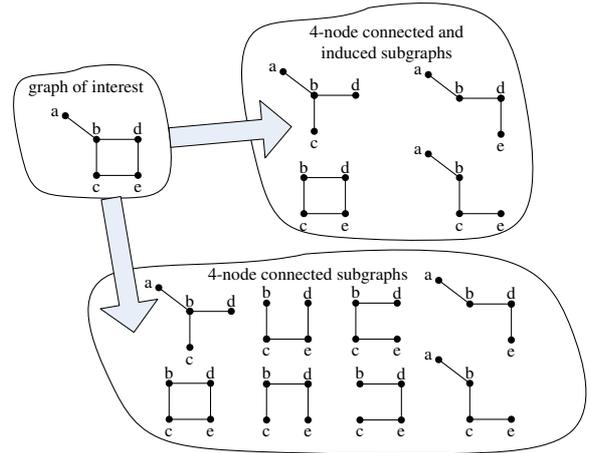


Figure 3. 4-node CISes vs. 4-node connected subgraphs.

edges. However it cannot be applied to characterize motifs, because motif statistics can differ from connected subgraph statistics. For example, Fig. 3 shows that 75% of a graph's 4-node connected subgraphs are isomorphic to a 4-node line (i.e., the 1st motif in Fig. 4(b)), while 50% of its 4-node CISes are isomorphic to a 4-node line.

Contribution: To the best of our knowledge, we are the “first” to study and provide an accurate and efficient solution to estimate motif statistics from a given RESampled graph. Hence, we do away with the previous assumption requiring the entire topology of the graph to be available. We introduce a probabilistic model to study the relationship between motifs in the RESampled graph and in the underlying graph. Based on this model, we propose an accurate method, Minfer, to infer the underlying graph's motif statistics from the RESampled graph. We also provide a Fisher information based method to bound the error of our estimates. Experiments on real world datasets show that our method can accurately estimate the motif statistics of a graph based on a small RESampled graph.

This paper is organized as follows: The problem formulation is presented in Section II. Section III presents our method (i.e. Minfer) for inferring subgraph class concentrations of the graph under study from a given RESampled graph. Section IV presents methods for computing the given RESampled graph's motif statistics. The performance evaluation and testing results are presented in Section V. Section VI summarizes related work. Concluding remarks then follow.

II. PROBLEM FORMULATION

In this section, we first introduce the concept of motif concentration and then discuss the challenges of computing motif concentrations in practice.

Denote the underlying graph of interest as a labeled undirected graph $G = (V, E, L)$, where V is a set of nodes, E is a set of *undirected* edges, $E \in V \times V$, and L is a set of labels $l_{u,v}$ associated with edges $(u, v) \in E$. For example, we attach a label $l_{u,v} \in \{\rightarrow, \leftarrow, \leftrightarrow\}$ to indicate the direction of the edge $(u, v) \in E$ for a directed network. Edges may have other labels too, for instance, in a signed network, edges have positive or negative labels to represent *friend* or *foe* relationship. If L

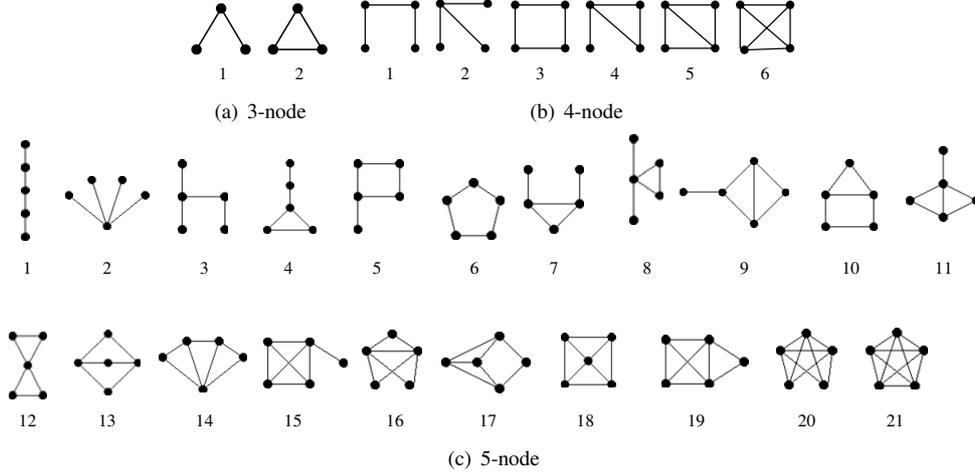


Figure 4. All classes of three-node, four-node, and five-node undirected and connected motifs (The numbers are the motif IDs).

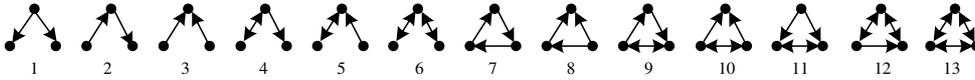


Figure 5. All classes of three-node directed and connected motifs (The numbers are the motif IDs).

is empty, then G is an unlabeled undirected graph, which is equivalent to an undirected graph.

Motif concentration is a metric that represents the distribution of various subgraph patterns that appear in G . To illustrate, we show the 3-, 4- and 5-node subgraph patterns in Figs. 4, 5, and 6 respectively. To define motif concentration formally, we first introduce some notation. For clarity of presentation, Table I depicts the notation used in this paper.

Table I. TABLE OF NOTATION.

G	$G = (V, E, L)$ is the graph under study
G^*	$G^* = (V^*, E^*, L^*)$ is a RESampled graph
$V(s), s \in C^{(k)}$	set of nodes for k -node CIS s
$E(s), s \in C^{(k)}$	set of edges for k -node CIS s
$M(s)$	associated motif of CIS s
T_k	number of k -node motif classes
$M_i^{(k)}$	i^{th} k -node motif
$C^{(k)}$	set of k -node CISes in G
$C_i^{(k)}$	set of CISes in G isomorphic to $M_i^{(k)}$
$n^{(k)} = C^{(k)} $	number of k -node CISes in G
$n_i^{(k)} = C_i^{(k)} $	number of CISes in G isomorphic to $M_i^{(k)}$
$m_i^{(k)}$	number of CISes in G^* isomorphic to $M_i^{(k)}$
$\omega_i^{(k)} = \frac{n_i^{(k)}}{n^{(k)}}$	concentration of motif $M_i^{(k)}$ in G
P	matrix $P = [P_{ij}]_{1 \leq i, j \leq T_k}$
$P_{i,j}$	probability that a k -node CIS s^* in G^* isomorphic to $M_i^{(k)}$ given its original CIS s in G isomorphic to $M_j^{(k)}$
$\phi_{i,j}$	number of subgraphs of $M_j^{(k)}$ isomorphic to $M_i^{(k)}$
$\mathbf{n}^{(k)}$	$\mathbf{n}^{(k)} = (n_1^{(k)}, \dots, n_{T_k}^{(k)})^T$
$\mathbf{m}^{(k)}$	$\mathbf{m}^{(k)} = (m_1^{(k)}, \dots, m_{T_k}^{(k)})^T$
$m^{(k)}$	$m^{(k)} = \sum_{i=1}^{T_k} m_i^{(k)}$
$\rho_i^{(k)} = \frac{m_i^{(k)}}{m^{(k)}}$	concentration of motif $M_i^{(k)}$ in G^*
p	probability of sampling an edge
q	$q = 1 - p$

An induced subgraph of G , $G' = (V', E', L')$, $V' \subset V$,

$E' \subset E$ and $L' \subset L$, is a subgraph whose edges and associated labels are all in G , i.e. $E' = \{(u, v) : u, v \in V', (u, v) \in E\}$, $L' = \{l_{u,v} : u, v \in V', (u, v) \in E\}$. We define $C^{(k)}$ as the set of all CISes with k nodes in G , and denote $n^{(k)} = |C^{(k)}|$. For example, Fig. 3 depicts all possible 4-node CISes. Let T_k denote the number of k -node motifs and $M_i^{(k)}$ denote the i^{th} k -node motif. For example, $T_4 = 6$ and $M_1^{(4)}, \dots, M_6^{(4)}$ are the six 4-node undirected motifs depicted in Fig. 4(b). Then we partition $C^{(k)}$ into T_k equivalence classes, or $C_1^{(k)}, \dots, C_{T_k}^{(k)}$, where CISes within $C_i^{(k)}$ are isomorphic to $M_i^{(k)}$. Note that in this paper node labels are not taken into account when checking the isomorphism. When there exist multiple isomorphisms between a CIS and a motif, we only count one of them.

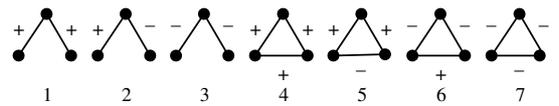


Figure 6. All classes of three-node signed and undirected motifs (The numbers are the motif IDs).

Let $n_i^{(k)}$ denote the frequency of motif $M_i^{(k)}$, i.e., the number of CISes in G isomorphic to $M_i^{(k)}$. Formally, we have $n_i^{(k)} = |C_i^{(k)}|$, which is the number of CISes in $C_i^{(k)}$. Then the concentration of $M_i^{(k)}$ is defined as

$$\omega_i^{(k)} = \frac{n_i^{(k)}}{n^{(k)}}, \quad 1 \leq i \leq T_k.$$

Thus, $\omega_i^{(k)}$ is the fraction of k -node CISes isomorphic to motif $M_i^{(k)}$ among all k -node CISes. In this paper, we make the follow assumptions:

- **Assumption 1:** The complete G is not available to us, but a RESampled graph $G^* = (V^*, E^*, L^*)$ of G

is given, where $V^* \in V$, $E^* \in E$, and L^* are node, edge, and edge label sets of G^* respectively. G^* is obtained by random edge sampling, i.e., each edge in E is independently sampled with the same probability $0 < p < 1$, where p is known in advance.

- **Assumption 2:** The label of a sampled edge $(u, v) \in G^*$ is the same as that of (u, v) in G , i.e., $l_{u,v}^* = l_{u,v}$.

These two assumptions are satisfied by many applications' data collection procedures. For instance, the data generated by an application such as network traffic monitoring is given as a stream of directed edges. The following simple method is computationally and memory efficient for collecting edges and generating a small RESampled graph that can be sent to remote network traffic analysis center: Each incoming directed edge $u \rightarrow v$ is sampled when $\tau(u, v) \leq \rho p$, where ρ is an integer (e.g., 10,000) and $\tau(u, v)$ is a hash function satisfying $\tau(u, v) = \tau(v, u)$ and mapping edges into integers $0, 1, \dots, \rho - 1$ uniformly. The property $\tau(u, v) = \tau(v, u)$ guarantees that edges $u \rightarrow v$ and $u \leftarrow v$ are either both sampled or discarded. Hence the label of a sampled edge $(u, v) \in E^*$ is the same as that of (u, v) in G . Using universal hashing [17], a simple instance of $\tau(u, v)$ is given as the following function when each $v \in V$ is an integer smaller than Δ

$$\tau(u, v) = (a(\min\{u, v\}\Delta + \max\{u, v\}) + b) \bmod \gamma \bmod \rho,$$

where γ is a prime larger than Δ^2 , a and b are any integers with $a \in \{1, \dots, \rho - 1\}$ and $b \in \{0, \dots, \rho - 1\}$. We can easily find that $\tau(u, v) = \tau(v, u)$ and $\tau(u, v)$ maps edges into integers $0, 1, \dots, \rho - 1$ uniformly. The computational and space complexities of the above sampling method are both $O(1)$, which make it practical for data collection. *As alluded before, in this paper, we aim to accurately infer the motif concentrations of G based on the given RESampled graph G^* .*

III. MOTIF STATISTICAL INFERENCE

The motif statistics of RESampled graph G^* and original graph G can be quite different. In this section, we introduce a probabilistic model to bridge the gap between the motif statistics of G^* and G . Using this model, we will establish a simple and concise relationship between the motif statistics of G and G^* . We then propose an efficient method to infer the motif concentration of G from G^* . Finally, we also give a method to construct confidence intervals of our estimates of motif concentrations.

A. Probabilistic Model of Motifs in G^* and G

To estimate the motif statistics of G based on G^* , we develop a probabilistic method to model the relationship between the motifs in G^* and G . Define $P = [P_{i,j}]$ where $P_{i,j}$ is the probability that s^* is isomorphic to motif $M_i^{(k)}$ given that s is isomorphic to motif $M_j^{(k)}$, i.e., $P_{i,j} = P(M(s^*) = M_i^{(k)} | M(s) = M_j^{(k)})$.

To obtain $P_{i,j}$, we first compute $\phi_{i,j}$, which is the number of subgraphs of $M_j^{(k)}$ isomorphic to $M_i^{(k)}$. For example, $M_2^{(3)}$ (i.e., the triangle) includes three subgraphs isomorphic to $M_1^{(3)}$ (i.e., the wedge) for the undirected graph shown in Fig. 4(a).

Thus, we have $\phi_{1,2} = 3$ for 3-node undirected motifs. When $i = j$, $\phi_{i,j} = 1$. Note that it is not easy to compute $\phi_{i,j}$ manually for 4- and 5-node motifs. Hence we provide a simple method to compute $\phi_{i,j}$ in Algorithm 1. The computational complexity is $O(k^2 k!)$. We want to emphasize that the cost of computing $\phi_{i,j}$ is not a big concern, because these values are static and independent of the input graph, and they can be computed once and for all and stored in a static table. Denote by $V(s)$ and $E(s)$ the sets of nodes and edges in subgraph s respectively. We have the following equation

$$P_{i,j} = \phi_{i,j} p^{|E(M_i^{(k)})|} q^{(|E(M_j^{(k)})| - |E(M_i^{(k)})|)},$$

where $q = 1 - p$. The above model implies that in expectation, the fraction of these CISes that appear as CISes isomorphic to $M_i^{(k)}$ in G^* is $P_{i,j}$.

Algorithm 1: Pseudo-code of computing $\phi_{i,j}$, i.e., the number of subgraphs of $M_j^{(k)}$ that are isomorphic to $M_i^{(k)}$.

- 1: **Step 1:** Generate two graphs $\hat{G} = (\{v_1, \dots, v_k\}, \hat{E}, \hat{L})$ and $\tilde{G} = (\{u_1, \dots, u_k\}, \tilde{E}, \tilde{L})$, isomorphic to motifs $M_i^{(k)}$ and $M_j^{(k)}$ respectively, where \hat{E} and \hat{L} are the edges and edge labels of \hat{G} with nodes v_1, \dots, v_k , and \tilde{E} and \tilde{L} are the edges and edge labels of \tilde{G} with nodes u_1, \dots, u_k .
 - 2: **Step 2:** Initialize a counter $y_{i,j} = 0$. For each permutation (x_1, \dots, x_k) of integers $1, \dots, k$, $y_{i,j}$ stays unchanged when there exists an edge $(v_a, v_b) \in \hat{E}$ satisfying $(u_{x_a}, u_{x_b}) \notin \tilde{E}$ or $\hat{l}_{v_a, v_b} \neq \tilde{l}_{u_{x_a}, u_{x_b}}$, and $y_{i,j} = y_{i,j} + 1$ otherwise.
 - 3: **Step 3:** Initialize a counter $z_j = 0$. For each permutation (x_1, \dots, x_k) of integers $1, \dots, k$, z_j stays unchanged when there exists an edge $(v_a, v_b) \in \hat{E}$ satisfying $(v_{x_a}, v_{x_b}) \notin \hat{E}$ or $\hat{l}_{v_a, v_b} \neq \hat{l}_{v_{x_a}, v_{x_b}}$, and $z_j = z_j + 1$ otherwise.
 - 4: **Step 4:** Finally, $\phi_{i,j} = y_{i,j} / z_j$.
-

B. Motif Concentration Estimation

Using the above probabilistic model, we propose a method Minfer to estimate motif statistics of G from G^* . Denote by $m_i^{(k)}$, $1 \leq i \leq T_k$, $k = 3, 4, \dots$, the number of CISes in G^* isomorphic to motif $M_i^{(k)}$. The method to compute $m_i^{(k)}$ is presented in the next section. Then, the expectation of $m_i^{(k)}$ is computed as

$$\mathbb{E}[m_i^{(k)}] = \sum_{1 \leq j \leq T_k} n_j^{(k)} P_{i,j}. \quad (1)$$

In matrix notation, Equation (1) can be expressed as

$$\mathbb{E}[\mathbf{m}^{(k)}] = P \mathbf{n}^{(k)},$$

where $P = [P_{i,j}]_{1 \leq i, j \leq T_k}$, $\mathbf{n}^{(k)} = (n_1^{(k)}, \dots, n_{T_k}^{(k)})^\top$, and $\mathbf{m}^{(k)} = (m_1^{(k)}, \dots, m_{T_k}^{(k)})^\top$. Then, we have

$$\mathbf{n}^{(k)} = P^{-1} \mathbb{E}[\mathbf{m}^{(k)}].$$

Thus, we estimate $\mathbf{n}^{(k)}$ as

$$\hat{\mathbf{n}}^{(k)} = P^{-1} \mathbf{m}^{(k)},$$

where $\hat{\mathbf{n}}^{(k)} = (\hat{n}_1^{(k)}, \dots, \hat{n}_{T_k}^{(k)})^\top$. We easily have

$$\mathbb{E}[\hat{\mathbf{n}}^{(k)}] = \mathbb{E}[P^{-1} \mathbf{m}^{(k)}] = P^{-1} \mathbb{E}[\mathbf{m}^{(k)}] = \mathbf{n}^{(k)},$$

therefore $\hat{\mathbf{n}}^{(k)}$ is an unbiased estimator of $\mathbf{n}^{(k)}$. Finally, we estimate $\omega_i^{(k)}$ as follows

$$\hat{\omega}_i^{(k)} = \frac{\hat{n}_i^{(k)}}{\sum_{j=1}^{T_k} \hat{n}_j^{(k)}}, \quad 1 \leq i \leq T_k. \quad (2)$$

Denote by $\rho_i^{(k)}$ the concentration of motif $M_i^{(k)}$ in G^* , i.e., $\rho_i^{(k)} = \frac{m_i^{(k)}}{m^{(k)}}$. We observe that (2) is equivalent to the following equation, which directly describes the relationship between motif concentrations of G and G^* . Let $\hat{\omega} = [\hat{\omega}_1^{(k)}, \dots, \hat{\omega}_{T_k}^{(k)}]^\top$ and $\rho = [\rho_1^{(k)}, \dots, \rho_{T_k}^{(k)}]^\top$, then we have

$$\hat{\omega} = \frac{P^{-1} \rho}{W}, \quad (3)$$

where $W = [1, \dots, 1] P^{-1} \rho$ is a normalizer. For 3-node undirected motifs, $P = \begin{pmatrix} p^2 & 3qp^2 \\ 0 & p^3 \end{pmatrix}$, and the inverse of P is $P^{-1} = \begin{pmatrix} p^{-2} & -3qp^{-3} \\ 0 & p^{-3} \end{pmatrix}$. Due to limited space, we present the expressions for P and P^{-1} for 3-node signed undirected motifs, 3-node directed motifs, 4-node undirected motifs, and 5-node undirected motifs in [18].

C. Lower Bound on Estimation Errors

It is difficult to directly analyze the errors of our estimate $\hat{\omega}$, because it is complex to model the dependence of sampled CISes due to their shared edges and nodes. Instead, we derive a lower bound on the mean squared error (MSE) of $\hat{\omega}$ using the Cramér-Rao lower bound (CRLB) of $\hat{\omega}$, which gives the smallest MSE that any unbiased estimator of ω can achieve. For a k -node CIS s selected from k -node CISes of G at random, the probability that s is isomorphic to the j^{th} k -node motif is $P(M(s) = M_j^{(k)}) = \omega_j^{(k)}$. Let s^* be the induced subgraph of the node set $V(s)$ in RESampled graph G^* . Clearly, s^* may not be connected. Furthermore, there may exist nodes in $V(s)$ that are not present in G^* . We say s^* is *evaporated* in G^* for these two scenarios. Let $P_{0,j}$ denote the probability that s^* is evaporated given that its original CIS s is isomorphic to the j^{th} k -node motif. Then, we have

$$P_{0,j} = 1 - \sum_{l=1}^{T_k} P_{l,j}.$$

For a random k -node CIS s of G , the probability that its associated s^* in G^* is isomorphic to the i^{th} k -node motif is

$$\xi_i = P(M(s^*) = M_i^{(k)}) = \sum_{j=1}^{T_k} P_{i,j} \omega_j^{(k)}, \quad 1 \leq i \leq T_k,$$

and the probability that s^* is evaporated is $\xi_0 = \sum_{j=1}^{T_k} P_{0,j} \omega_j^{(k)}$. When s^* is evaporated, we denote $M(s^*) = 0$.

Then, the likelihood function of $M(s^*)$ with respect to $\omega^{(k)}$ is

$$f(i|\omega^{(k)}) = \xi_i, \quad 0 \leq i \leq T_k.$$

The Fisher information of $M(s^*)$ with respect to $\omega^{(k)}$ is defined as a matrix $J = [J_{i,j}]_{1 \leq i,j \leq T_k}$, where

$$\begin{aligned} J_{i,j} &= \mathbb{E} \left[\frac{\partial \ln f(l|\omega^{(k)})}{\partial \omega_i} \frac{\partial \ln f(l|\omega^{(k)})}{\partial \omega_j} \right] \\ &= \sum_{l=0}^{T_k} \frac{\partial \ln f(l|\omega^{(k)})}{\partial \omega_i} \frac{\partial \ln f(l|\omega^{(k)})}{\partial \omega_j} \xi_l = \sum_{l=0}^{T_k} \frac{P_{l,i} P_{l,j}}{\xi_l}. \end{aligned}$$

For simplicity, we assume that the CISes of G^* are independent (i.e., no overlapping edges). Then the Fisher information matrix of all k -node CISes is $n^{(k)} J$. The Cramér-Rao Theorem states that the MSE of any unbiased estimator is lower bounded by the inverse of the Fisher information matrix, i.e.,

$$\text{MSE}(\hat{\omega}_i^{(k)}) = \mathbb{E}[(\hat{\omega}_i^{(k)} - \omega_i^{(k)})^2] \geq \frac{(J^{-1})_{i,i} - \omega^{(k)}(\omega^{(k)})^\top}{n^{(k)}}$$

provided some weak regularity conditions hold [19]. Here the term $\omega^{(k)}(\omega^{(k)})^\top$ corresponds to the accuracy gain obtained by accounting for the constraint $\sum_{i=1}^{T_k} \omega_i^{(k)} = 1$. The CRLB method provides us a way to set p properly, i.e., we can perform a pilot study to estimate/guess the original graph's statistics, and then use the CRLB method to evaluate the estimation errors for different p .

IV. ENUMERATE 3-, 4-, AND 5-NODE CISES

The existing generalized graph enumeration method [13] can be used for enumerating all k -node CISes in RESampled graph G^* , while it is not easy to apply and is (computationally and memory) inefficient for small values of $k = 3, 4, 5$. In this section, we first present a method (an extension of the NodeIterator++ method in [20]) to enumerate and count 3-node CISes in G^* . Then, we propose new methods to enumerate and count 4 and 5-node CISes in G^* respectively. In what follows, we denote $N^*(u)$ as the neighbors of u in G^* . Note that in this section G^* is the default graph when we define a function. For example, the CIS with nodes u, v , and w refers to the CIS with nodes u, v , and w in G^* . We would like to point out in this paper we focus on obtaining an accurate estimate from a given RESampled graph, and our aim is not to reduce the computational cost, so the method in this section can be replaced when there exist other more efficient methods of counting motifs.

A. Enumerate 3-node CIS

Algorithm 2 shows our 3-node CISes enumeration method. Similar to the NodeIterator++ method in [20], we ‘‘pivot’’ (the associated operation is discussed later) each node $u \in V^*$ to enumerate CISes including u . For any two neighbors v and w of u , we can easily find that the induced graph s with nodes u, v and w is a 3-node CIS. Thus, we enumerate all pairs of two nodes in $N^*(u)$, and update their associated 3-node CIS for u . We call this process ‘‘pivoting’’ u for 3-node CISes.

Clearly, a 3-node CIS s is counted three times when the associated undirected graph of s by discarding edge labels is isomorphic to a triangle, once by pivoting each node u, v , and

Algorithm 2: 3-node CIS enumeration via pivoting.

```

input :  $G^* = (V^*, E^*, L^*)$ 
/*  $m_i^{(3)}$  records the number of CISes in  $G^*$ 
   isomorphic to motif  $M_i^{(3)}$ ,  $1 \leq i \leq T_3$ . */
output:  $\mathbf{m}^{(3)} = (m_1^{(3)}, \dots, m_{T_3}^{(3)})^\top$ 

for  $u \in V^*$  do
  for  $v \in N^*(u)$  do
    for  $w \in N^*(u)$  and  $w \succ v$  do
      /* induced( $G^*, \Gamma$ ) returns the CIS with
         the node set  $\Gamma$  of  $G^*$ . */
       $s \leftarrow \text{induced}(G^*, \{u, v, w\})$ ;
      if  $(w, v) \in E^*$  and  $u \succ v$  then
        | continue();
      end
      /*  $M(s)$  is the motif class ID
         of  $s$ . */
       $i \leftarrow M(s)$ ;
       $m_i^{(3)} \leftarrow m_i^{(3)} + 1$ ;
    end
  end
end

```

w . Let \succ be an arbitrary total order on all of the nodes, which can be easily defined and obtained, e.g. from array position or pointer addresses. To ensure each CIS is enumerated once and only once, we let one and only one node in a CIS be the *leader* of the CIS, which is “responsible” for making sure the CIS gets counted. When we “pivot” u and enumerate a CIS s , s is counted if u is the leader of s . Otherwise, s is discarded and not counted. We use the same method in [20], [21], i.e., let the node with lowest order in a CIS whose associated undirected graph is isomorphic to a triangle be the leader. For the other classes of CISes, their associated undirected graphs are isomorphic to an unclosed wedge, i.e., the 1st motif in Fig. 4(a). For each of these CISes, we let the node in the center of its associated undirected graph (e.g., the node with degree 2 in the unclosed wedge) be the leader.

B. Enumerate 4-node CISes

Algorithm 3 shows our 4-node CISes enumeration method. To enumerate 4-node CISes, we “pivot” each node u as follows: For each pair of u ’s neighbors v and w where $w \succ v$, we compute the neighborhood of u, v , and w , defined as $\Gamma = N^*(u) \cup N^*(v) \cup N^*(w) - \{u, v, w\}$. For any node $x \in \Gamma$, we observe that the induced graph s consisting of nodes u, v, w , and x is a 4-node CIS. Thus, we enumerate each node x in Γ , and update the 4-node CIS consisting of u, v, w , and x . We repeat this process until all pairs of u ’s neighbors v and w are enumerated and processed.

Similar to 3-node CISes, some 4-node CISes may be enumerated and counted more than once when we “pivot” each node u as above. To solve this problem, we propose the following methods for making sure each 4-node CIS s is enumerated and gets counted once and only once: When $(u, x) \in E^*$ and $w \succ x$, we discard x . Otherwise, denote by \hat{s} the associated undirected graph of s by discarding edge labels. When \hat{s} includes one and only one node u having at least 2 neighbors in $V(\hat{s})$, we let u be the *leader* of s . For example,

the node 4 is the leader of the 1st subgraph in Fig. 7. When \hat{s} includes more than one node having at least 2 neighbors in $V(\hat{s})$, we let the node with lowest order among the nodes having at least 2 neighbors in $V(\hat{s})$ be the leader of s . For example, the nodes 6 and 3 are the leaders of the 2nd and 3rd subgraphs in Fig. 7.

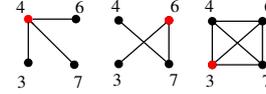


Figure 7. Examples of the leaders of 4-node CISes. Graphs shown are CISes’ associated undirected graphs, and the number near to a node represents the node order. Red nodes are the leaders.

Algorithm 3: 4-node CIS enumeration via pivoting.

```

input :  $G^* = (V^*, E^*, L^*)$ 
/*  $m_i^{(4)}$  records the number of CISes in  $G^*$ 
   isomorphic to motif  $M_i^{(4)}$ ,  $1 \leq i \leq T_4$ . */
output:  $\mathbf{m}^{(4)} = (m_1^{(4)}, \dots, m_{T_4}^{(4)})^\top$ 

for  $u \in V^*$  do
  for  $v \in N^*(u)$  do
    for  $w \in N^*(u)$  and  $w \succ v$  do
       $\Gamma = N^*(u) \cup N^*(v) \cup N^*(w) - \{u, v, w\}$ ;
      for  $x \in \Gamma$  do
        if  $(u, x) \in E^*$  and  $w \succ x$  then
          | continue();
        end
        /* induced( $G^*, \{u, v, w, x\}$ ) is defined
           same as Alg. 2. */
         $s \leftarrow \text{induced}(G^*, \{u, v, w, x\})$ ;
        /* undirected( $s$ ) returns the
           associated undirected graph of
            $s$  by discarding edge labels. */
         $\hat{s} \leftarrow \text{undirected}(s)$ ;
        /* findNodes( $\hat{s}, t$ ) returns the set
           of nodes in  $V(\hat{s})$  having at
           least  $t$  neighbors in  $V(\hat{s})$ . */
         $\Lambda \leftarrow \text{findNodes}(\hat{s}, 2)$ ;
        if  $|\Lambda| \geq 2$  then
          /* minNodes( $\Lambda$ ) returns the node
             with the lowest order in
              $V(\hat{s})$ . */
          if  $u \succ \text{minNodes}(\Lambda)$  then
            | continue();
          end
        end
         $i \leftarrow M(s)$ ;
         $m_i^{(4)} \leftarrow m_i^{(4)} + 1$ ;
      end
    end
  end
end

```

C. Enumerate 5-node CISes

Algorithm 4 describes our 5-node CISes enumeration method. For a 5-node CIS s , we classify it into two types according to its associated undirected graph \hat{s} :

- **Type 1 5-node CIS** s : \hat{s} includes at least one node with more than two neighbors in $V(\hat{s})$;
- **Type 2 5-node CIS** s : All nodes in \hat{s} have no more than two neighbors in $V(\hat{s})$, i.e., \hat{s} is isomorphic to a 5-node line or a circle, i.e., the 1st or 6th motifs in Fig. 4(c).

We propose two different methods to enumerate these two types of 5-node CISes respectively.

To enumerate type 1 5-node CISes, we “pivot” each node u as follows: When u has at least three neighbors, we enumerate each combination of three nodes $v, w, x \in N^*(u)$ where $x \succ w \succ v$, and then compute the neighborhood of u, v, w , and x , defined as $\Gamma \leftarrow N^*(u) \cup N^*(v) \cup N^*(w) \cup N^*(x) - \{u, v, w, x\}$. For any node $y \in \Gamma$, we observe that the induced graph s consisting of nodes u, v, w, x , and y is a 5-node CIS. Thus, we enumerate each node y in Γ , and update the associated 5-node CIS consisting of u, v, w, x , and y . We repeat this process until all combinations of three nodes $v, w, x \in N^*(u)$ are enumerated and processed. Similar to 4-node CISes, we propose the following method to make sure each 5-node s is enumerated and gets counted once and only once: When $(y, u) \in E^*$ and $y \succ x$, we discard y . Otherwise, let \hat{s} be the associated undirected graph of s , and we then pick the node with lowest order among the nodes having more than two neighbors in $V(\hat{s})$ be the leader. The 3rd and 4th subgraphs in Fig. 8 are two corresponding examples.

To enumerate type 2 5-node CISes, we “pivot” each node u as follows: When u has at least two neighbors, we first enumerate each pair of u 's neighbors v and w where $(v, w) \notin E^*$. Then, we compute Γ_v defined as the set of v 's neighbors not including u and w and not connected to u and w , that is, $\Gamma_v \leftarrow N^*(v) - \{u, w\} - N^*(u) - N^*(w)$. Similarly, we compute Γ_w defined as the set of w 's neighbors not including u and v and not connected to u and v , i.e., $\Gamma_w \leftarrow N^*(w) - \{u, v\} - N^*(u) - N^*(v)$. Clearly, $\Gamma_v \cap \Gamma_w = \emptyset$. For any $x \in \Gamma_v$ and $y \in \Gamma_w$, we observe that the induced graph s consisting of nodes u, v, w, x , and y is a type 2 5-node CIS. Thus, we enumerate each pair $(x, y) \in \Gamma_v \times \Gamma_w$, and update the 5-node CIS consisting of u, v, w, x , and y . We repeat this process until all pairs of u 's neighbors v and w are enumerated and processed. To make sure each CIS s is enumerated and gets counted once and only once, we let the node with lowest order be the leader when the associated undirected graph \hat{s} of s is isomorphic to a 5-node circle. When \hat{s} is isomorphic to a 5-node line, we let the node in the center of the line be the leader. The 1st and 2nd subgraphs in Fig. 8 are two examples.

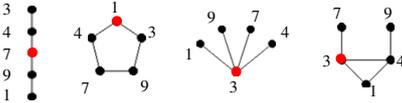


Figure 8. Examples of the leaders of 5-node CISs. Graphs shown are CISes' associated undirected graphs, and the number near to a node represents the node order. Red nodes are the leaders.

V. EVALUATION

In this section, we first introduce our experimental datasets and then present results of experiments used to evaluate the

Algorithm 4: 5-node CIS enumeration via pivoting.

```

input :  $G^* = (V^*, E^*, L^*)$ 
/*  $m_i^{(5)}$  records the number of CISes in  $G^*$ 
   isomorphic to motif  $M_i^{(5)}$ ,  $1 \leq i \leq T_5$ . */
output:  $\mathbf{m}^{(5)} = (m_1^{(5)}, \dots, m_{T_5}^{(5)})^T$ 

/* The functions findNodes, minNodes, induced,
   and undirected are defined in Algorithms 2
   and 3. */
for  $u \in V^*$  do
  for  $v \in N^*(u)$  do
    for  $w \in N^*(u)$  and  $w \succ v$  do
      /* Enumerate and update CIS  $s$  with
         undirected( $s$ ) not isomorphic to a
         5-node line and circle. */
      for  $x \in N^*(u)$  and  $x \succ w$  do
         $\Gamma \leftarrow N^*(u) \cup N^*(v) \cup N^*(w) \cup$ 
         $N^*(x) - \{u, v, w, x\}$ ;
        for  $y \in \Gamma$  do
          if  $(y, u) \in E^*$  and  $x \succ y$  then
            | continue();
          end
           $s \leftarrow \text{induced}(G^*, \{u, v, w, x, y\})$ ;
           $\hat{s} \leftarrow \text{undirected}(s)$ ;
           $\Lambda \leftarrow \text{findNodes}(\hat{s}, 3)$ ;
          if  $|\Lambda| \geq 2$  then
            | if  $u \succ \text{minNodes}(\Lambda)$  then
              | | continue();
            | end
          end
           $i \leftarrow M(s)$ ;
           $m_i^{(5)} \leftarrow m_i^{(5)} + 1$ ;
        end
      end
      /* Enumerate and update  $s$  with
         undirected( $s$ ) isomorphic to a
         5-node line or circle. */
      if  $(v, w) \notin E^*$  then
         $\Gamma_v \leftarrow N^*(v) - \{u, w\} - N^*(u) - N^*(w)$ ;
        for  $x \in \Gamma_v$  do
          /*  $s$  with undirected( $s$ )
             isomorphic to a 5-node
             circle. */
           $\Gamma_w \leftarrow$ 
           $N^*(w) - \{u, v\} - N^*(u) - N^*(v)$ ;
          for  $y \in \Gamma_w$  do
            if  $(x, y) \in E^*$  and
             $u \succ \text{minNodes}(\{u, v, w, x, y\})$ 
            then
              | continue();
            end
             $s \leftarrow$ 
             $\text{induced}(G^*, \{u, v, w, x, y\})$ ;
             $i \leftarrow M(s)$ ;
             $m_i^{(5)} \leftarrow m_i^{(5)} + 1$ ;
          end
        end
      end
    end
  end
end

```

Table II. GRAPH DATASETS USED IN OUR SIMULATIONS, “EDGES” REFERS TO THE NUMBER OF EDGES IN THE UNDIRECTED GRAPH GENERATED BY DISCARDING EDGE LABELS, “MAX-DEGREE” REPRESENTS THE MAXIMUM NUMBER OF EDGES INCIDENT TO A NODE IN THE UNDIRECTED GRAPH.

Graph	nodes	edges	max-degree
Flickr [22]	1,715,255	15,555,041	27,236
Pokec [23]	1,632,803	22,301,964	14,854
LiveJournal [22]	5,189,809	48,688,097	15,017
YouTube [22]	1,138,499	2,990,443	28,754
Wiki-Talk [24]	2,394,385	4,659,565	100,029
Web-Google [25]	875,713	4,322,051	6,332
soc-Epinions1 [10]	75,897	405,740	3,044
soc-Slashdot08 [9]	77,360	469,180	2,539
soc-Slashdot09 [9]	82,168	504,230	2,552
sign-Epinions [26]	119,130	704,267	3,558
sign-Slashdot08 [26]	77,350	416,695	2,537
sign-Slashdot09 [26]	82,144	504,230	2,552
com-DBLP [27]	317,080	1,049,866	343
com-Amazon [27]	334,863	925,872	549
p2p-Gnutella08 [28]	6,301	20,777	97
ca-GrQc [29]	5,241	14,484	81
ca-CondMat [29]	23,133	93,439	279
ca-HepTh [29]	9,875	25,937	65

Table III. VALUES OF $\omega_i^{(3)}$, THE CONCENTRATIONS OF 3-NODE UNDIRECTED AND DIRECTED MOTIFS. (i IS THE MOTIF ID.)

i	Flickr	Pokec	LiveLive-Journal	Wiki-Talk	Web-Google
undirected 3-node motifs					
1	9.60e-01	9.84e-01	9.55e-01	9.99e-01	9.81e-01
2	4.04e-02	1.60e-02	4.50e-02	7.18e-04	1.91e-02
directed 3-node motifs					
1	2.17e-01	1.77e-01	7.62e-02	8.91e-01	1.27e-02
2	6.04e-02	1.11e-01	4.83e-02	4.04e-02	1.60e-02
3	1.28e-01	1.60e-01	3.28e-01	3.91e-03	9.28e-01
4	2.44e-01	1.74e-01	1.14e-01	5.43e-02	3.09e-03
5	1.31e-01	1.91e-01	1.73e-01	5.48e-03	1.92e-02
6	1.80e-01	1.71e-01	2.15e-01	3.88e-03	1.92e-03
7	5.69e-05	7.06e-05	2.74e-05	1.37e-05	4.91e-05
8	6.52e-03	2.49e-03	8.66e-03	1.81e-04	6.82e-03
9	1.58e-03	1.03e-03	1.06e-03	8.42e-05	2.84e-04
10	5.19e-03	1.91e-03	6.63e-03	1.28e-04	2.77e-03
11	6.46e-03	2.03e-03	6.27e-03	8.03e-05	5.98e-03
12	1.07e-02	5.13e-03	9.82e-03	1.78e-04	1.21e-03
13	9.86e-03	3.45e-03	1.26e-02	6.65e-05	2.00e-03

performance of our method, Minfer, for characterizing CIS classes of size $k = 3, 4, 5$.

A. Datasets

We evaluate the performance of our methods on publicly available datasets taken from the Stanford Network Analysis Platform (SNAP)(www.snap.stanford.edu), which are summarized in Table II. We start by evaluating the performance of our methods in characterizing 3-node CISes over million-node graphs: Flickr, Pokec, LiveJournal, YouTube, Web-Google,

Table IV. NRMSEs OF $\hat{\omega}_i^{(3)}$, THE CONCENTRATION ESTIMATES OF 3-NODE UNDIRECTED MOTIFS FOR $p = 0.01$ AND $p = 0.05$ RESPECTIVELY. (i IS THE MOTIF ID.)

i	Flickr	Pokec	LiveLive-Journal	Wiki-Talk	Web-Google
$p = 0.01$					
1	1.92e-03	3.26e-03	2.69e-03	5.21e-03	2.93e-04
2	4.56e-02	6.92e-02	1.64e-01	2.67e-01	4.00e-01
$p = 0.05$					
1	2.90e-04	4.10e-04	2.64e-04	6.06e-04	2.92e-05
2	6.90e-03	8.68e-03	1.61e-02	3.11e-02	3.99e-02

Table V. VALUES OF $\omega_i^{(3)}$, THE CONCENTRATIONS OF 3-NODE SIGNED AND UNDIRECTED MOTIFS. (i IS THE MOTIF ID.)

i	sign-Epinions	sign-Slashdot08	sign-Slashdot09
1	6.69e-01	6.58e-01	6.68e-01
2	2.12e-01	2.32e-01	2.25e-01
3	9.09e-02	1.02e-01	9.96e-02
4	2.29e-02	5.86e-03	5.75e-03
5	2.76e-03	9.74e-04	9.34e-04
6	2.49e-03	1.14e-03	1.13e-03
7	3.81e-04	1.80e-04	1.76e-04

Table VI. VALUES OF $\omega_i^{(4)}$, THE CONCENTRATIONS OF 4-NODE UNDIRECTED MOTIFS. (i IS THE MOTIF ID.)

i	soc-Epinions1	soc-Slashdot08	soc-Slashdot09	com-Amazon
1	3.24e-01	2.93e-01	2.90e-01	2.10e-01
2	6.15e-01	6.86e-01	6.89e-01	6.99e-01
3	2.78e-03	1.25e-03	1.30e-03	2.37e-03
4	5.45e-02	1.86e-02	1.84e-02	7.69e-02
5	3.01e-03	7.77e-04	8.48e-04	1.05e-02
6	2.25e-04	9.19e-05	9.36e-05	1.55e-03

and Wiki-talk, contrasting our results with the ground truth computed through an exhaustive method. It is computationally intensive to calculate the ground-truth of 4-node and 5-node CIS classes in large graphs. For example, we easily observe that a node with degree $d > 4$ is included in at least $\frac{1}{6}d(d-1)(d-2)$ 4-node CISes and $\frac{1}{24}d(d-1)(d-2)(d-3)$ 5-node CISes, therefore it requires more than $O(10^{15})$ and $O(10^{19})$ operations to enumerate the 4-node and 5-node CISes within the Wiki-talk graph, which contains one node with 100,029 neighbors. Even for a relatively small graph such as soc-Slashdot08, it takes almost 20 hours to compute all of its 4-node CISes. To solve this problem, the experiments for 4-node CISes are performed on four medium-sized graphs soc-Epinions1, soc-Slashdot08, soc-Slashdot09, com-DBLP, and com-Amazon, and the experiments for 5-node CISes are performed on four relatively small graphs ca-GR-QC, ca-HEP-TH, ca-CondMat, and p2p-Gnutella08, where computing the ground-truth is feasible. We also evaluate the performance of our methods for characterizing signed CIS classes in graphs

Table VII. VALUES OF $\omega_i^{(5)}$, CONCENTRATIONS OF 5-NODE UNDIRECTED MOTIFS. (i IS THE MOTIF ID.)

i	com-Amazon	com-DBLP	p2p-Gnutella08	ca-GrQc	ca-CondMat	ca-HepTh
1	2.9e-2	1.4e-1	2.6e-1	9.8e-2	1.4e-1	2.6e-1
2	7.5e-1	1.8e-1	1.8e-1	5.2e-2	2.2e-1	8.2e-2
3	1.6e-1	4.4e-1	4.6e-1	2.1e-1	4.3e-1	4.4e-1
4	6.0e-3	4.8e-2	1.1e-2	1.0e-1	4.9e-2	6.0e-2
5	2.3e-3	1.1e-3	2.7e-2	1.4e-3	2.1e-3	5.4e-3
6	3.6e-5	5.0e-5	1.4e-3	9.2e-5	1.1e-4	4.1e-4
7	1.5e-2	5.6e-2	2.7e-2	1.1e-1	5.5e-2	6.4e-2
8	3.5e-2	7.9e-2	2.2e-2	1.2e-1	8.0e-2	5.2e-2
9	1.4e-3	4.2e-3	1.4e-3	1.5e-2	7.0e-3	8.4e-3
10	1.7e-4	1.4e-4	1.0e-3	6.5e-4	3.0e-4	8.0e-4
11	7.3e-3	8.1e-3	4.3e-3	2.3e-2	9.9e-3	1.0e-2
12	5.3e-4	6.4e-3	2.8e-4	2.3e-2	4.5e-3	3.6e-3
13	8.2e-5	3.5e-6	7.4e-4	4.5e-6	6.4e-6	3.5e-5
14	3.9e-4	5.2e-4	1.7e-4	2.8e-3	6.6e-4	1.0e-3
15	6.7e-4	2.6e-2	7.6e-5	1.5e-1	5.9e-3	5.3e-3
16	7.1e-4	3.4e-4	1.4e-4	1.4e-3	9.2e-4	4.4e-4
17	3.9e-5	1.1e-5	8.0e-5	4.3e-5	2.9e-5	8.4e-5
18	2.3e-5	4.9e-6	6.0e-6	2.3e-5	8.5e-6	3.0e-5
19	2.4e-4	2.8e-3	1.5e-5	1.9e-2	9.8e-4	5.8e-4
20	5.8e-5	4.2e-4	7.0e-7	8.0e-3	1.4e-4	8.2e-5
21	7.2e-6	7.9e-3	1.5e-8	6.1e-2	1.5e-4	3.2e-3

sign-Epinions, sign-Slashdot08, and sign-Slashdot09.

B. Error Metric

In our experiments, we focus on the normalized root mean square error (NRMSE) to measure the relative error of the estimator $\hat{\omega}_i$ of the subgraph class concentration ω_i , $i = 1, 2, \dots$. NRMSE($\hat{\omega}_i$) is defined as:

$$\text{NRMSE}(\hat{\omega}_i) = \frac{\sqrt{\text{MSE}(\hat{\omega}_i)}}{\omega_i}, \quad i = 1, 2, \dots,$$

where $\text{MSE}(\hat{\omega}_i)$ is defined as the mean square error (MSE) of an estimate $\hat{\omega}_i$ with respect to its true value $\omega_i > 0$, that is

$$\text{MSE}(\hat{\omega}_i) = \mathbb{E}[(\hat{\omega}_i - \omega_i)^2] = \text{Var}(\hat{\omega}_i) + (\mathbb{E}[\hat{\omega}_i] - \omega_i)^2.$$

We note that $\text{MSE}(\hat{\omega}_i)$ decomposes into a sum of the variance and bias of the estimator $\hat{\omega}_i$. Both quantities are important and need to be as small as possible to achieve good estimation performance. When $\hat{\omega}_i$ is an unbiased estimator of ω_i , then we have $\text{MSE}(\hat{\omega}_i) = \text{Var}(\hat{\omega}_i)$ and thus NRMSE($\hat{\omega}_i$) is equivalent to the normalized standard error of $\hat{\omega}_i$, i.e., $\text{NRMSE}(\hat{\omega}_i) = \sqrt{\text{Var}(\hat{\omega}_i)}/\omega_i$. Note that our metric uses the relative error. Thus, when ω_i is small, we consider values as large as $\text{NRMSE}(\hat{\omega}_i) = 1$ to be acceptable. In all our experiments, we average the estimates and calculate their NRMSEs over 1,000 runs.

C. Accuracy Results

1) Inferring 3-node motif concentrations: Table III shows the real values of the 3-node undirected and directed motifs' concentrations for the undirected graphs and directed graphs of Flickr, Pokec, LiveJournal, Wiki-Talk, and Web-Google. Among all 3-node directed motifs, the 7th motif exhibits the smallest concentration for all these five directed graphs. Here the undirected graphs are obtained by discarding the edge directions of directed graphs. Flickr, Pokec, LiveJournal, Wiki-Talk, and Web-Google have 1.35×10^{10} , 2.02×10^9 , 6.90×10^9 , 1.2×10^{10} , and 7.00×10^8 3-node CISEs respectively. Table IV shows the NRMSEs of our estimates of 3-node undirected motifs' concentrations for $p = 0.01$ and $p = 0.05$ respectively. We observe that the NRMSEs associated with the sampling probability $p = 0.05$ is about ten times smaller than the NRMSEs when $p = 0.01$. The NRMSEs are smaller than 0.04 when $p = 0.05$ for all five graphs. Fig. 9 shows the NRMSEs of our estimates of 3-node directed motifs' concentrations for $p = 0.01$ and $p = 0.05$ respectively. Similarly, we observe the NRMSEs when $p = 0.05$ are nearly ten times smaller than the NRMSEs when $p = 0.01$. The NRMSE of our estimates of $\omega_7^{(3)}$ (i.e., the 7th 3-node directed motif concentration) exhibits the largest error. Except for $\omega_7^{(3)}$, the NRMSEs of the other motif concentrations' estimates are smaller than 0.2 when $p = 0.05$. Web-Google exhibits larger errors than the other graphs, because it has less 3-node CISEs.

Table V shows the real values of 3-node signed motifs' concentrations for sign-Epinions, sign-Slashdot08, and sign-Slashdot09. Sign-Epinions, sign-Slashdot08, and sign-Slashdot09 have 1.72×10^8 , 6.72×10^7 , and 7.25×10^7 3-node CISEs respectively. Fig. 10 shows the NRMSEs of our estimates of 3-node signed and undirected motifs' concentrations for $p = 0.05$ and $p = 0.1$ respectively. For all these three

signed graphs, the NRMSEs are smaller than 0.9 and 0.2 when $p = 0.05$ and $p = 0.1$ respectively.

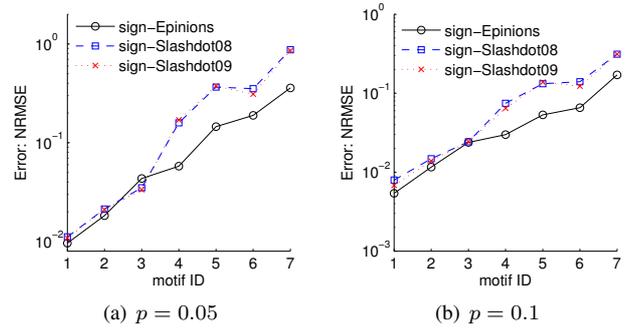


Figure 10. NRMSEs of $\omega_i^{(3)}$, the concentration estimates of 3-node signed and undirected motifs for $p = 0.05$ and $p = 0.1$ respectively.

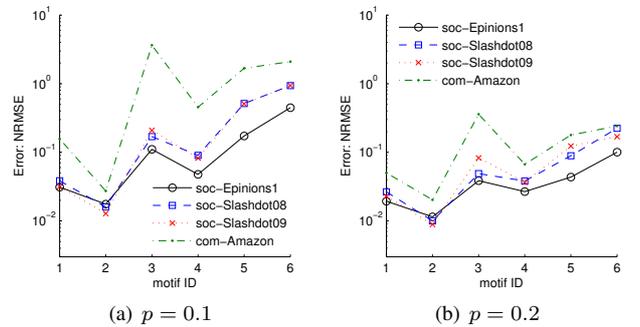


Figure 11. NRMSEs of $\hat{\omega}_i^{(4)}$, the concentration estimates of 4-node undirected motifs for $p = 0.1$, and $p = 0.2$ respectively.

2) Inferring 4-node motif concentrations: Table VI shows the real values of $\omega_i^{(4)}$, i.e., the concentrations of 4-node undirected motifs for soc-Epinions1, soc-Slashdot08, soc-Slashdot09, and com-Amazon. Soc-Epinions1, soc-Slashdot08, soc-Slashdot09, and com-Amazon have 2.58×10^{10} , 2.17×10^{10} , 2.42×10^{10} , and 1.78×10^8 4-node CISEs respectively. Fig. 11 shows the NRMSEs of $\hat{\omega}_i^{(4)}$, the concentration estimates of 4-node undirected motifs for $p = 0.05$, $p = 0.1$, and $p = 0.2$ respectively. We observe that motifs with smaller $\omega_i^{(4)}$ exhibit larger NRMSEs. Except for $\omega_3^{(4)}$, the NRMSEs of the motif concentration estimates are smaller than 0.2 for $p = 0.2$.

3) Inferring 5-node motif concentrations: Table VII shows the real values of $\omega_i^{(5)}$, i.e., the concentrations of 5-node undirected motifs for com-Amazon, com-DBLP, p2p-Gnutella08, ca-GrQc, ca-CondMat, and ca-HepTh. Com-Amazon, com-DBLP, p2p-Gnutella08, ca-GrQc, ca-CondMat, and ca-HepTh contains 8.50×10^9 , 3.34×10^{10} , 3.92×10^8 , 3.64×10^7 , 3.32×10^9 , and 8.73×10^7 5-node CISEs respectively. Fig. 12 shows the NRMSEs of $\hat{\omega}_i^{(5)}$, the concentration estimates of 5-node undirected motifs for $p = 0.1$, $p = 0.2$, and $p = 0.3$ respectively. We observe that NRMSE decreases as p increases, and the 6th, 10th, 13th, 17th, 18th 5-node motifs with small $\omega_i^{(5)}$ exhibit large NRMSEs. We generate a large graph G consisting of R soc-Amazon graphs, i.e., G has R components, and each component is an instance of the soc-Amazon graph. Clearly G has the same motif distributions

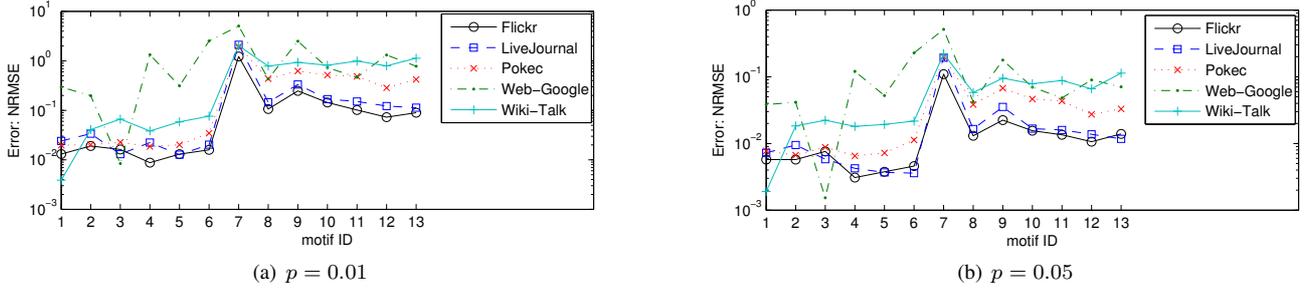


Figure 9. NRMSEs of $\hat{\omega}_i^{(3)}$, the concentration estimates of 3-node directed motifs for $p = 0.01$ and $p = 0.05$ respectively.

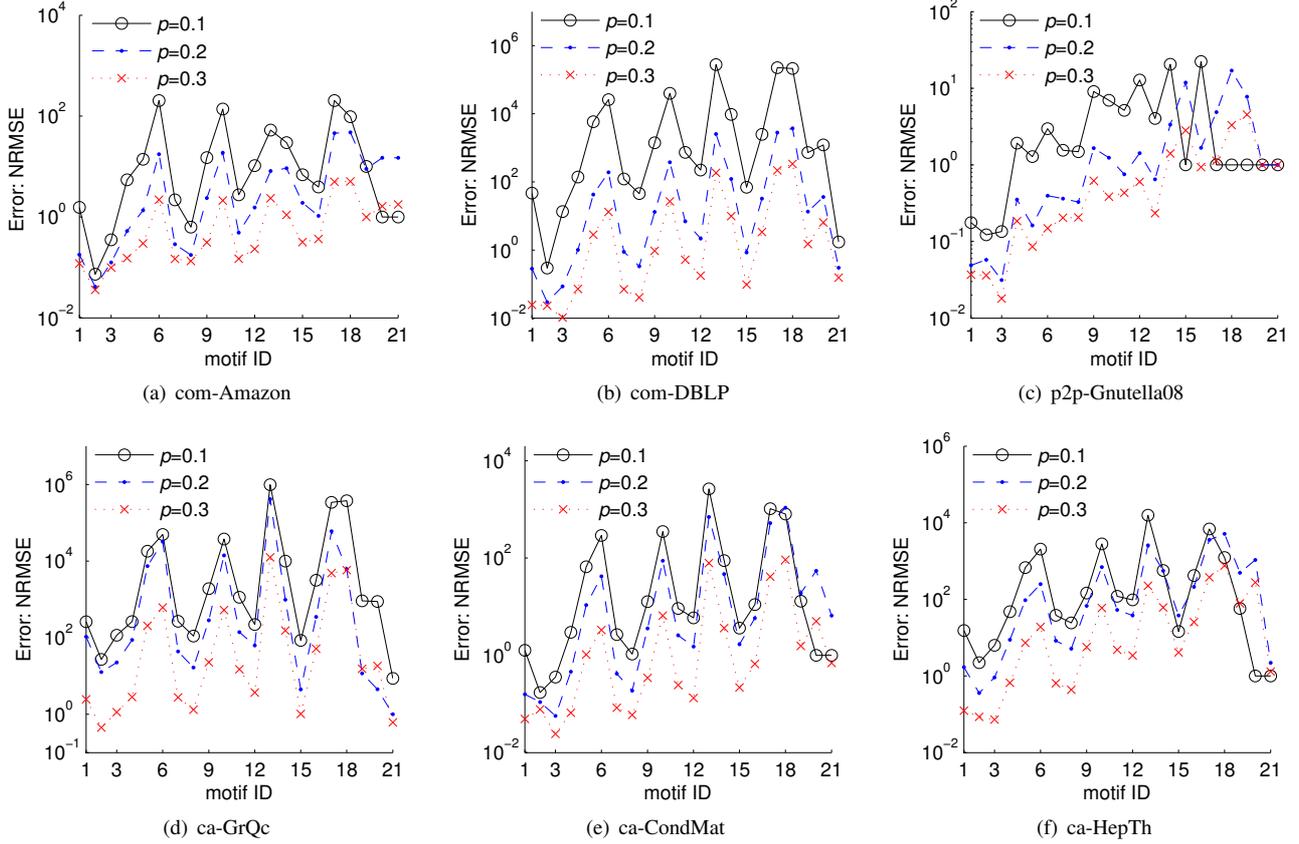


Figure 12. NRMSEs of $\hat{\omega}_i^{(5)}$, the concentration estimates of 5-node undirected motifs for $p = 0.1$, $p = 0.2$, and $p = 0.3$ respectively.

as the soc-Amazon graph. Fig. 13 shows that the NRMSEs decreases as R increases, which indicates that our methods may exhibit small errors for characterizing 5-node motif of large graphs.

D. Error Bounds

Figure 14 shows the root CRLBs (RCRLBs) and the root MSEs (RMSEs) of our estimates of 3-node directed motifs' concentrations, 4-, and 5-node undirected motifs' concentrations, where graphs LiveJournal, soc-Epinions, and com-DBLP are used for studying 3-node directed motifs, 4-, and 5-node undirected motifs respectively. We observe that the RCRLBs are smaller than the RMSEs, and fairly close to the RMSEs. The RMSEs and RCRLBs are almost indistinguishable for 3-node directed motifs, where $p = 0.01$ and LiveJournal is used.

It indicates that the RCRLBs can efficiently bound the errors of our motif concentration estimations.

VI. RELATED WORK

There has been considerable interest in designing efficient sampling methods for counting specific subgraph patterns such as triangles [16], [30]–[33], cliques [34], [35], and cycles [30], [36], because it is computationally intensive to compute the number of the subgraph pattern's appearances in a large graph. Similar to the problem studied in [11]–[14], [37], in this work we focus on characterizing 3-, 4-, and 5-nodes CISEs in a *single large graph*, which differs from the problem of estimating the number of subgraph patterns appearing in a *large set of graphs* studied in [38]. OmidGenes et al. [37] proposed a subgraph enumeration and counting method using

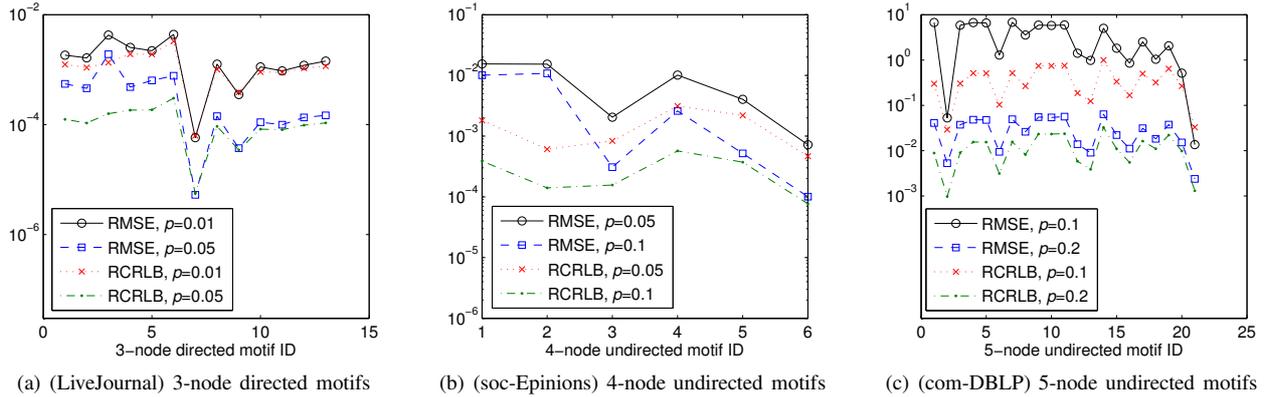


Figure 14. RCRLBs and RMSEs of concentration estimates of 3, 4, and 5-node directed motifs.

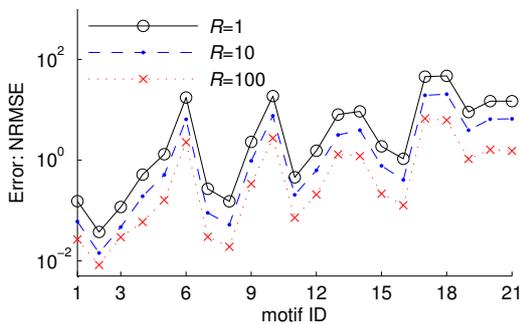


Figure 13. NRMSEs of $\hat{\omega}_i^{(5)}$, the concentration estimates of 5-node undirected motifs for a graph consisting of R soc-Amazon graphs, where $p = 0.2$.

sampling. However this method suffers from an unknown sampling bias. To estimate subgraph class concentrations, Kashtan et al. [12] proposed a subgraph sampling method, but their method is computationally expensive when calculating the weight of each sampled subgraph, which is needed to correct for the bias introduced by sampling. To address this drawback, Wernicke [13] proposed an algorithm, FANMOD, based on enumerating subgraph trees to detect network motifs. Bhuiyan et al. [14] proposed a Metropolis-Hastings based sampling method GUISE to estimate 3-node, 4-node, and 5-node subgraph frequency distribution. Wang et al. [11] proposed an efficient crawling method to estimate online social networks' motif concentrations, when the graph's topology is not available in advance and it is costly to crawl the entire topology. Work on graph sparsifiers such as [39] focuses on designing methods to obtain a sparse graph similar to the original graph with respect to a specific graph metric such as triangle count. We are interested in a different problem, to infer the original graph's motif statistics from a given RESampled graph and we assume the original graph is not available or cannot be explored. Triangle sparsifiers is used to estimate the original graph's triangle count. It cannot characterize 4- and 5-node motifs, and 3-node directed motifs, because it does not consider how to remove the uncertainty that a sampled 3-node CIS may differ from its original CISes. In summary, previous methods focus on designing efficient sampling methods and crawling methods for estimating motif statistics when the

graph is directly available or indirectly available (i.e., it is not expensive to query a node's neighbors [11]). They cannot be applied to solve the problem studied in this paper, i.e., we assume the graph is not available but a RESampled graph is given and we aim to infer the underlying graph's motif statistics from the RESampled graph. At last, we would like to point out our method of estimating motif statistics and its error bound computation method are inspired by methods of estimating flow size distribution for network traffic measurement and monitoring [40]–[43].

VII. CONCLUSIONS

In this paper, we study the problem of inferring the underlying graph's motif statistics when the entire graph topology is not available, and only a RESampled graph is given. We propose a model to bridge the gap between the underlying graph's and its RESampled graph's motif statistics. Based on this probabilistic model, we develop a method Minfer to infer the underlying graph's motif statistics, and give a Fisher information based method to bound the error of our estimates. and experimental results on a variety of known data sets validate the accuracy of our method.

ACKNOWLEDGMENT

We thank the anonymous reviewers as well as Dr. Wei Fan for helpful suggestions. This work was supported in part by Ministry of Education & China Mobile Joint Research Fund Program (MCM20150506) the National Natural Science Foundation of China (61103240, 61103241, 61221063, 91118005, 61221063, U1301254), 863 High Tech Development Plan (2012AA011003), 111 International Collaboration Program of China, the Application Foundation Research Program of SuZhou (SYG201311), and the Prospective Research Project on Future Networks of Jiangsu Future Networks Innovation Institute. This work was also supported by ARO under MURI W911NF-08-1-0233 and ARL under Cooperative Agreement W911NF-09-2-0053. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied of the ARL, or the U.S. Government. The work of John C.S. Lui was supported in part by the GRF 415013.

REFERENCES

- [1] H. Chun, Y. yeol Ahn, H. Kwak, S. Moon, Y. ho Eom, and H. Jeong, "Comparison of online social relations in terms of volume vs. interaction: A case study of cyworld," in *Proceedings of ACM SIGCOMM Internet Measurement Conference 2008*, November 2008, pp. 57–59.
- [2] J. Kunegis, A. Lommatzsch, and C. Bauckhage, "The slashdot zoo: mining a social network with negative edges," in *Proceedings of WWW 2009*, April 2009, pp. 741–750.
- [3] J. Zhao, J. C. S. Lui, D. Towsley, X. Guan, and Y. Zhou, "Empirical analysis of the evolution of follower network: A case study on douban," in *Proceedings of IEEE INFOCOM NetSciCom 2011*, April 2011, pp. 941–946.
- [4] J. Ugander, L. Backstrom, and J. Kleinberg, "Subgraph frequencies: mapping the empirical and extremal geography of large graph collections," in *Proceedings of the 22nd international conference on World Wide Web*, ser. WWW 2013, 2013, pp. 1307–1318.
- [5] S. S. Shen-Orr, R. Milo, S. Mangan, and U. Alon, "Network motifs in the transcriptional regulation network of escherichia coli," *Nature Genetics*, vol. 31, no. 1, pp. 64–68, May 2002.
- [6] I. Albert and R. Albert, "Conserved network motifs allow protein–protein interaction prediction," *Bioinformatics*, vol. 4863, no. 13, pp. 3346–3352, 2004.
- [7] S. Itzkovitz, R. Levitt, N. Kashtan, R. Milo, M. Itzkovitz, and U. Alon, "Coarse-graining and self-dissimilarity of complex networks," *Physica Rev.E*, vol. 71, p. 016127, 2005.
- [8] R. Milo, E. Al, and C. Biology, "Network motifs: Simple building blocks of complex networks," *Science*, vol. 298, no. 5549, pp. 824–827, October 2002.
- [9] J. Leskovec, K. J. Lang, A. Dasgupta, and M. W. Mahoney, "Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters," *Internet Mathematics*, vol. 6, no. 1, pp. 29–123, 2009.
- [10] M. Richardson, R. Agrawal, and P. Domingos, "Trust management for the semantic web," in *Proceedings of the 2nd International Semantic Web Conference*, October 2003, pp. 351–368.
- [11] P. Wang, J. C. Lui, J. Zhao, B. Ribeiro, D. Towsley, and X. Guan, "Efficiently estimating motif statistics of large networks," *ACM Transactions on Knowledge Discovery from Data*, 2014.
- [12] N. Kashtan, S. Itzkovitz, R. Milo, and U. Alon, "Efficient sampling algorithm for estimating subgraph concentrations and detecting network motifs," *Bioinformatics*, vol. 20, no. 11, pp. 1746–1758, 2004.
- [13] S. Wernicke, "Efficient detection of network motifs," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 3, no. 4, pp. 347–359, 2006.
- [14] M. A. Bhuiyan, M. Rahman, M. Rahman, and M. A. Hasan, "Guise: Uniform sampling of graphlets for large graph analysis," in *Proceedings of IEEE ICDM 2012*, December 2012, pp. 91–100.
- [15] B. Ribeiro and D. Towsley, "Estimating and sampling graphs with multidimensional random walks," in *Proceedings of ACM SIGCOMM Internet Measurement Conference 2010*, November 2010, pp. 390–403.
- [16] N. Ahmed, N. Duffield, J. Neville, and R. Kompella, "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*, 2014, pp. 589–597.
- [17] J. Chen, W. Hsu, M.-L. Lee, and S.-K. Ng, "Nemofinder: dissecting genome-wide protein-protein interactions with meso-scale network motifs," in *Proceedings of ACM SIGKDD 2006*, August 2006, pp. 106–115.
- [18] P. Wang, J. C. S. Lui, and D. Towsley, "Minfer: Inferring motif statistics from sampled edges," *CoRR*, vol. abs/1502.06671, 2015. [Online]. Available: <http://arxiv.org/abs/1502.06671>
- [19] H. L. van Trees, *Estimation and Modulation Theory, Part 1*. New York: Wiley, 2001.
- [20] S. Suri and S. Vassilvitskii, "Counting triangles and the curse of the last reducer," ser. WWW 2011, 2011, pp. 607–614.
- [21] T. Schank, "Algorithmic aspects of triangle-based network analysis," Ph.D. dissertation, 2007.
- [22] A. Mislove, M. Marcon, K. P. Gummadi, P. Druschel, and B. Bhattacharjee, "Measurement and analysis of online social networks," in *Proceedings of ACM SIGCOMM Internet Measurement Conference 2007*, October 2007, pp. 29–42.
- [23] L. Takac and M. Zabovsky, "Data analysis in public social networks," in *International Scientific Conference and International Workshop Present Day Trends of Innovations*, May 2012, pp. 1–6.
- [24] J. Leskovec, D. Huttenlocher, and J. Kleinberg, "Predicting positive and negative links in online social networks," in *Proceedings of WWW 2010*, April 2010, pp. 641–650.
- [25] "Google programming contest," <http://www.google.com/programming-contest/>, 2002.
- [26] J. Leskovec, D. Huttenlocher, and J. Kleinberg, "Signed networks in social media," in *Proceedings of the 28th ACM Conference on Human Factors in Computing Systems (CHI)*, April 2010, pp. 1361–1370.
- [27] J. Yang and J. Leskovec, "Defining and evaluating network communities based on ground-truth," in *12th IEEE International Conference on Data Mining (ICDM)*, 2012, pp. 745–754.
- [28] M. Ripeanu, I. T. Foster, and A. Iamnitchi, "Mapping the gnutella network: Properties of large-scale peer-to-peer systems and implications for system design," *IEEE Internet Computing Journal*, vol. 6, no. 1, pp. 50–57, 2002.
- [29] J. Leskovec, J. Kleinberg, and C. Faloutsos, "Graph evolution: Densification and shrinking diameters," *Transactions on Knowledge Discovery from Data (TKDD)*, vol. 1, no. 1, Mar. 2007.
- [30] N. Alon, R. Yuster, and U. Zwick, "Color-coding," *J. ACM*, vol. 42, no. 4, pp. 844–856, Jul. 1995.
- [31] C. E. Tsourakakis, U. Kang, G. L. Miller, and C. Faloutsos, "Doulion: Counting triangles in massive graphs with a coin," in *PROCEEDINGS OF ACM KDD 2009*, 2009.
- [32] A. Pavany, K. T. S. Tirthapuraz, and K.-L. Wu, "Counting and sampling triangles from a graph stream," in *Proceedings of VLDB*, 2013, pp. 1870–1881.
- [33] M. Jha, C. Seshadhri, and A. Pinar, "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*, 2013, pp. 589–597.
- [34] J. Cheng, Y. Ke, A. W.-C. Fu, J. X. Yu, and L. Zhu, "Finding maximal cliques in massive networks," *ACM Transactions on Database Systems*, vol. 36, no. 4, pp. 21:1–21:34, dec 2011.
- [35] M. Gjoka, E. Smith, and C. T. Butts, "Estimating Clique Composition and Size Distributions from Sampled Network Data," *ArXiv e-prints*, Aug. 2013.
- [36] M. Manjunath, K. Mehlhorn, K. Panagiotou, and H. Sun, "Approximate counting of cycles in streams," in *Proceedings of the 19th European Conference on Algorithms*, ser. ESA'11. Berlin, Heidelberg: Springer-Verlag, 2011, pp. 677–688.
- [37] S. Omid, F. Schreiber, and A. Masoudi-nejad, "Moda: An efficient algorithm for network motif discovery in biological networks," *Genes and Genet systems*, vol. 84, no. 5, pp. 385–395, 2009.
- [38] M. A. Hasan and M. J. Zaki, "Output space sampling for graph patterns," in *Proceedings of the VLDB Endowment 2009*, August 2009, pp. 730–741.
- [39] C. E. Tsourakakis, M. N. Kolountzakis, and G. L. Miller, "Triangle sparsifiers," *J. Graph Algorithms Appl.*, vol. 15, no. 6, pp. 703–726, 2011.
- [40] N. Duffield, C. Lund, and M. Thorup, "Estimating flow distributions from sampled flow statistics," in *Proceedings of ACM SIGCOMM 2003*, August 2003, pp. 325–336.
- [41] B. Ribeiro, D. Towsley, T. Ye, and J. Bolot, "Fisher information of sampled packets: an application to flow size estimation," in *Proceedings of ACM SIGCOMM IMC 2006*, October 2006, pp. 15–26.
- [42] P. Tune and D. Veitch, "Towards optimal sampling for flow size estimation," in *Proc. of the IMC*, 2008, pp. 243–256.
- [43] P. Wang, X. Guan, J. Zhao, J. Tao, and T. Qin, "A new sketch method for measuring host connection degree distribution," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 6, pp. 948–960, 2014.