Knowledge-Based Systems 198 (2020) 105891

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

DRaWS: A dual random-walk based sampling method to efficiently estimate distributions of degree and clique size over social networks

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

[a] Wuhan National Laboratory for Optoelectronics, Key Laboratory of Information Storage System, Engineering Research Center of data storage systems and technology (School of Computer Science and Technology, Huazhong University of Science and Technology), Ministry of Education of China, China

[b] Department of Computer Science and Engineering, University of Texas at Arlington, USA

ARTICLE INFO

Article history:
Received 18 November 2019
Received in revised form 5 April 2020
Accepted 6 April 2020
Available online 13 April 2020

Keywords:
Random-walk based graph sampling
Estimator
Distributions of degree and clique structures
Social networks

ABSTRACT

Random-walk based sampling methods have been widely employed to characterize social networks. However, existing random-walk based sampling methods cause inaccuracies in estimating the degree structure and high sampling costs in estimating clique structures. In this paper, we propose a dual random-walk based sampling method, called DRaWS by designing a dual residence of the random walker, to estimate both the distributions of degree and clique size with low costs. The key idea behind DRaWS is that it leverages the many-to-one formation between many nodes and one clique in a large graph to shorten the sampling paths and thus reduce the sampling costs greatly while reflecting the different sampling probabilities of the two types of node structures. Meanwhile, DRaWS employs the one-to-many representativeness between one node and many nodes in a clique to improve the quality of samples. Furthermore, two re-weighted estimators for DRaWS's process are proposed to estimate the two different node structures. Experimental evaluation driven by real graph datasets shows that DRaWS drastically cuts down the sampling costs of the state-of-the-art methods while increasing the accuracy when estimating both the degree and clique structural properties.

1. Introduction

In social networks that can be represented by large graphs, degree is referred to as the number of neighbors of a user while clique size is the number of maximum completely connected users related to the user. Both degree and clique size are important structures in social networks. In this paper, the statistical properties of degree in a large graph are called the *degree structure* and those of clique size are called the *clique structure*. The former is used to describe the coarse-grained property of social networks while the latter is to describe the fine-grained properties of relationship among the users and can be applied in the applications of data mining, classifications and visualizations [1–5]. Furthermore, the union of the two structures is important to reflect the connectivity and can be applied in many applications [6]. For example, according to the propagation model described in [7], the average degree is to reflect the number of infected users just through one way (user) while the average number of the maximum clique that a user participates in is to reflect the number of infected users from multiple ways (users). Thus, the union of the average degree and size of the maximum clique of the social network can reflect the average influence of social networks more accurately than a single factor does.

However, given the sheer volumes of data in online social networks, combined with the inaccessibility of the datasets in their entireties under many circumstances, the distributions of the degree and the clique size that are the basis for their average values respectively, cannot be learned from the whole datasets of social networks. Thus, random-walk sampling methods [8,9] are conveniently employed to obtain a small part of users (samples) to obtain their degrees and clique sizes given the impractical uses of random sampling on nodes or edges and the serious biases of the traversal-based sampling [10,11]. The existing random-walk based sampled methods are referred to as RaWS in this paper. These RaWS methods [11–13] share the key step of selecting the next sampling node from the neighbors of the latest-sampled one [9,14,15] or the previously sampled one. During the sampling processes, they obtain the degree and clique structures from the sampled nodes and then output the structural properties of the whole graph by employing estimation algorithms [9]. From these sampling processes, we can infer that the existing RaWS methods mainly focus on designing the sampling processes based on the degree structures of the sampling nodes while they do not develop the already-obtained clique structures. Therefore, the existing RaWS methods cannot differentiate the two structures during sampling processes, resulting in inaccurate estimations.

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

https://doi.org/10.1016/j.knosys.2020.105891
0950-7051/© 2020 Elsevier B.V. All rights reserved.

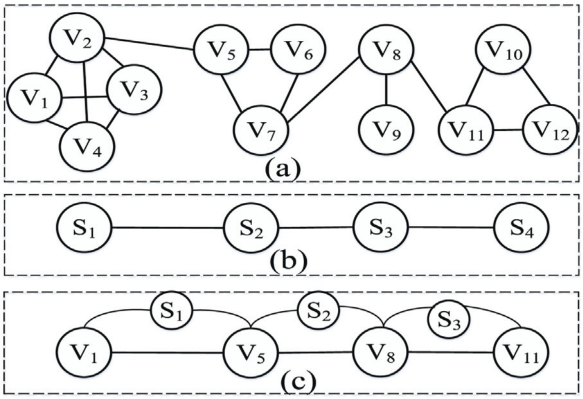


Fig. 1. An example comparing the length and number of paths of the random walks between RaWS and DRaWS. Assume superstructures $S1 = \{V_1, V_2, V_3, V_4\}$, $S_2 = \{V_5, V_6, V_7\}$, $S_3 = \{V_8, V_9\}$ and $S4 = \{V_{10}, V_{11}, V_{12}\}$, RaWS walks over at least 4 nodes, among 6 possible paths, from V_1 to V_{11} with 5 sampling steps while DRaWS walks over 2 superstructures S_2, S_3 along a single path, with 3 sampling steps from S_1 to S_4 or 2 nodes V_5, V_8.

Furthermore, each of the existing RaWS methods can be described as a Markov-chain based sampling process [16,17]. To obtain accurate estimations about node structures, Markov-chain based sampling methods are required to reach a stationary state. Due to the large volumes of data in online social networks, the number of sampling steps required for a typical RaWS [18–20] method to reach a stationary state, defined as mixing time of a corresponding Markov chain process, is extremely large.

In this paper, we propose a new random-walk based sampling method, called a *D*ual *Ra*ndom-*W*alk based *S*ampling (DRaWS) method to estimate a large graph. In DRaWS, the maximum clique of a sampling node is used as a superstructure. The residence of DRaWS's random walker, which is referred to as the stepping point through which the random walker traverses a large graph, is designed as a dual form of a node and its corresponding superstructure in any given sampling step. Specifically, when the random walker is staying on a node, it changes its residence to the superstructure after the latter is found for the node. Once on the superstructure, the random walker chooses its next sampling node from the neighboring nodes of the superstructure (defined in Section 3). Then the new residence of the walker is the newly chosen sampling node which will trigger a new round of changes of residence as described above. DRaWS moves forward with the changes of the residence of the random walker until the sampling budget is met (i.e., a given number of samples required to estimate node structures).

Because finding the degree and clique structures are the prerequisite for their estimations, DRaWS, as a new sampling strategy, does not incur additional costs. Furthermore, DRaWS employs the superstructures to simplify the sampling paths and then it can reduce sampling costs because the edges among the nodes within any superstructure are 'eliminated' when traversing among superstructures, leaving only edges between nodes of neighboring superstructures, or *bridges* between superstructures. As described in Fig. 1, without considering backtrackings, the length of the paths of the random walker for a typical RaWS method is at least 5 (i.e., $V_1 \rightarrow V_2 \rightarrow V_5 \rightarrow V_7 \rightarrow V_8 \rightarrow V_{11}$) while the number of possible paths for its walker is 6 ($C_3^1 \times C_2^1$ where C_3^1 is the number of the paths from V_1 to V_5 and C_2^1 the number of paths from V_5 to V_8) as shown in Fig. 1(a). Whereas, from the perspective of superstructures, the paths from V_1 to V_{11} in a typical RaWS method are replaced by the paths from S_1 to S_4 in DRaWS's process whose length and number are 3 ($S_1 \rightarrow S_2 \rightarrow S_3 \rightarrow S_4$) and 1 respectively, as shown in Fig. 1(b). On the other hand, from the perspective of nodes, those paths in a typical RaWS are replaced by a single path of length 3 (i.e., $V_1 \rightarrow V_5 \rightarrow V_8 \rightarrow V_{11}$) by leveraging the dual residence of the random walker, as shown in Fig. 1(c).

With the design and evaluation of DRaWS, this paper makes the following contributions.

1. We analyze the key factors affecting the mixing time of a Markov-chain based sampling process and reveal the reasons why the existing sampling methods have large sampling costs and errors especially in estimating the clique structures (Section 2). To address these problems, we propose a dual random-walk based sampling process, named DRaWS, the first of its kind, with high accuracy and low cost to estimate the degree and clique structures simultaneously. (Section 3).
2. The mixing time of DRaWS' sampling process is decreased substantially by reducing the length and the number of the paths of the random walks and the probabilities of backtracking to the already sampled nodes while improving the representativeness of the samples. This enables DRaWS to drastically reduce its sampling costs while significantly reducing sampling errors (Section 3).
3. To accurately estimate the degree and clique structures in large graphs with samples obtained by DRaWS, we propose different re-weighted estimators by theoretical analysis on the dual random-walk based process of DRaWS, combined with the Horvitz–Thompson estimator [21] and the unordered estimator [22] (Section 4).
4. To evaluate the effectiveness of DRaWS, extensive experiments driven by real-world graph datasets based on a DRaWS prototype and the existing state-of-the-art random-walk based methods are conducted. They show that DRaWS' sampling costs, such as memory consumption, network overhead, and computation time, are drastically reduced while its sampling errors are consistently smaller than those of the existing random-walk based sampling methods (Section 5).

The rest of the paper is organized as follows. Section 2 describes the necessary background and motivates the DRaWS research. Section 3 introduces the DRaWS method while Section 4 proposes re-weighted estimators for DRaWS. Evaluation results from experiments are presented and discussed in Section 5. Section 6 summarizes the related works while Section 7 concludes our work.

2. Background and motivation

In this section, we first introduce the mixing time of a Markov-chain based process and then analyze the factors that affect the mixing time and thus the sampling costs. Next, we analyze and examine the processes of SRW (simple random walk) and MHRW (Metropolis–Hasting random walk), which lay foundations for most of the variations of existing RaWS methods, to identify their shortcomings of long mixing time and repetitive samples. The insights obtained from these analyses motivate us to propose the DRaWS approach.

2.1. Mixing time and hitting time

Suppose M is an irreducible and aperiodic Markov chain process. The state space of M is V and $P = \{p(\mu, \nu)\}, \mu, \nu \in V$, is the transition matrix of M. After t steps, the relative distance ($\triangle(t, \mu)$) between $\mu's$ current distribution and its stationary distribution ($\pi(\mu)$) is described as:

$$\triangle(t, \mu) = min_{\nu \in V}\{t, |\frac{\pi(\mu) - p^{(t)}(\mu, \nu)}{\pi(\mu)}|\}, \quad (1)$$

where $p^{(t)}(\mu, \nu)$ denotes $\mu's$ distribution after t steps for the Markov chain process. The *mixing time* is the minimum value of $\triangle(t, \mu)$ with respect to a parameter ϵ for all the items in the state space. A typical RaWS process is equivalent to a Markov chain process where the state space is the node set and the transition matrix is formed by the probabilities of the random walker traversing from one node to another in the node set. However, it is impractical for a large state space (i.e., billions of nodes in a large graph) to obtain the specific number of steps for all the nodes to reach their respective stationary distributions.

Nevertheless, it is useful to reduce the mixing time of a Markov chain process by means of reducing the *hitting time* $H_{(\mu,\nu)}, (\mu, \nu \in V)$, which is defined as the mean number of steps consumed by the random walker from μ to ν based on the transition matrix P. As inferred in [23,24], in a connected graph, less hitting time results in less mixing time.

Suppose that $\{p\_1, \ldots, p\_k\}$ are k paths from μ to ν and $Pr(path(\mu, \nu) = p\_i), 1 \le i \le k$ is the probability of the random walker reaching ν from μ along p_i. $Pr(path(\mu, \nu) = p\_i)$ is given as follows.

$$Pr(path(\mu, \nu) = p\_i) = p(\mu, \alpha_1) \times p(\alpha_1, \alpha_2) \times \cdots \times p(\alpha_n, \nu), \tag{2}$$

where $\alpha_1, \ldots, \alpha_n$ are constituent nodes of path p_i connecting nodes μ and ν. Thus, the mean number of steps (labeled as $H(\mu, \nu)_i$) required by the random walker from μ to ν along the $i - th$ path p_i is given as below.

$$H(\mu, \nu)_i = \frac{1}{Pr(path(\mu, \nu) = p\_i)}. \tag{3}$$

Therefore, the hitting time from μ to ν is given as below.

$$H(\mu, \nu) = \frac{1}{k} \sum_{i=1}^{i=k} H(\mu, \nu)_i. \tag{4}$$

Based on Eqs. (2)–(4), there are at least the following three factors affecting the hitting time $H(\mu, \nu)$ and thus the mixing time of a random-walk based sampling process.

- *Length of the paths*. Suppose that μ and ν are two nodes in a large graph, a path from μ to ν is defined as $shortL(\mu, \nu)$. For simplicity, we only consider the length of a path from μ to ν without self-loops. $shortL(\mu, \nu)$ can be described as follows.

  $$shortL(\mu, \nu) = (\mu, \alpha_1) + (\alpha_1, \alpha_2) + \cdots + (\alpha_n, \nu), \tag{5}$$

  where $\alpha_1, \ldots, \alpha_n \in V$. The longer the path between μ and ν $(\mu, \nu \in V)$, the more intermediate nodes are traversed by the random-walker. More intermediate nodes mean more multiplicative terms in Eq. (2) that are each less than or at most equal to one, resulting in a smaller $Pr(path(\mu, \nu) = p\_i)$ and thus a larger mean number of steps $H(\mu, \nu)_i$ (Eq. (3)).
- *Number of the paths*. Suppose there are two pairs: (μ, ν) and (α, β), the number of paths from μ to ν is k while that from α to β is $k + 1$, and $H(\mu, \nu)_i = H(\alpha, \beta)_i$ $(i \le k)$. Thus, as described in Eq. (4), the distance between the hitting time from μ to ν and that from α to β is described as follows:

  $$\begin{aligned} H(\alpha, \beta) - H(\mu, \nu) &= \frac{1}{k+1} \sum_{i=1}^{i=k+1} H(\alpha, \beta)_i - \frac{1}{k} \sum_{i=1}^{i=k} H(\mu, \nu)_i \\ &= \frac{k \times H(\alpha, \beta)_{(i+1)} - (\sum_{i=1}^{i=k} H(\mu, \nu)_i)}{k \times (k+1)} \end{aligned} \tag{6}$$

  If $H(\alpha, \beta)_{(i+1)}$ is larger than the average value in the set $\{H(\mu, \nu)_i, 1 \le i \le k\}$, we have $H(\alpha, \beta) > H(\mu, \nu)$. According to Eq. (5), the longer length means the larger hitting time. Therefore, a larger number of long paths can further increase the hitting time.
- *The reversibility of the random walker*. Backtracking, where the random walker traverses to previously sampled nodes, further lengthens the walker's paths from μ to ν while also increasing the number of the paths traversed. As described in the studies in [25–27], a non-reversible Markov chain sampling process shows shorter mixing time than a reversible one.

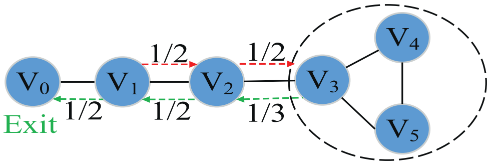


Fig. 2. The process of SRW being trapped in the superstructure consisting of nodes V_4, V_5, and V_6. Given the current sample node V_2, the probability of entering into the superstructure from V_2 is $\frac{1}{2} \times \frac{1}{2} = \frac{1}{4}$ while the probability of jumping out of it from V_4, once in it, is $\frac{1}{3} \times \frac{1}{2} \times \frac{1}{2} = \frac{1}{12}$ without considering other backtracking possibilities (i.e., $V_4 \rightarrow V_3 \rightarrow V_2 \rightarrow V_3 \rightarrow V_4$). Then the probability of the sampling process being trapped in the superstructure is about $0.23(\simeq \frac{1}{4} \times \frac{11}{12})$.

2.2. SRW and MHRW

SRW and MHRW are two representative Markov-chain based sampling processes widely used by many applications.

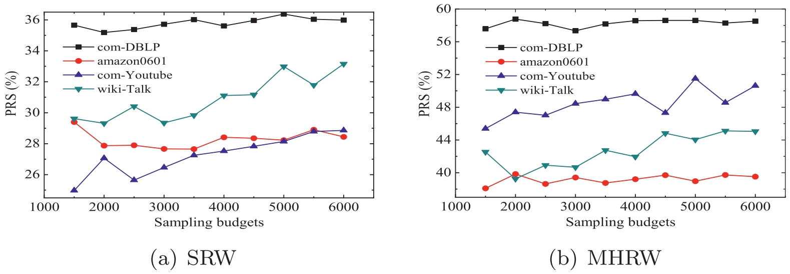
Simple Random Walk (SRW). SRW [9,14,28] first initializes a sample node randomly, and then continues to select the next sample randomly from the neighbors of the current sample node until the sampling budget is met. The transition probability from the current sample node μ to the next sample node ν is defined as follows.

$$P^{SRW}_{(\mu,\nu)} = \begin{cases} \frac{1}{deg(\mu)} & \text{if } \nu \text{ is the neighbor of } \mu, \\ 0 & \text{otherwise,} \end{cases}$$

where $deg(\mu)$ is the degree of μ. When SRW is required to reach the stationary distribution, the sampling probability of the node μ converges to a fixed value, $\pi^{SRW}_{\mu} = \frac{deg(\mu)}{2*|E|}$ [29]. SRW's process can be seen as Markov-chain based. When evaluating the mixing time of SRW, we observe the following two serious problems.

First, the existence of cliques or superstructures increases the mixing time by increasing the length and the number of the random walker's paths from one node to another. Specifically, as illustrated in Fig. 2, once the sampling procedure enters the superstructure indicated by the black dashed circle, the probability of the SRW process backtracking to the nodes of that superstructure is about $\frac{11}{12}$. Thus, the number of steps for the process required to 'jump out' from such superstructures is large, thereby increasing the mixing time.

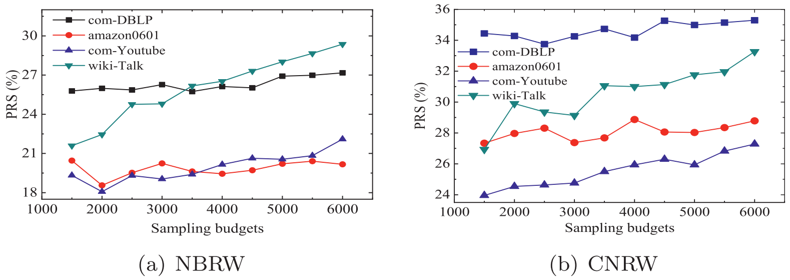
Second, a great number of repetitive samples are produced because of the sampling process backtracking to the sampled nodes. In this paper, we measure repetitive samples by *percentage of repetitive samples (PRS)*, defined as $PRS = 100 \times \frac{(B-U)}{B}$ with B and U being the total number of samples and the number of unique samples among B respectively. Clearly, the higher the PRS value, the less useful information can be derived from the samples obtained with a given sampling budget. Fig. 3(a) shows that the average PRS values obtained by a single-run simulation of SRW are significant, between 24% and 36% over the four datasets



(a) SRW

(b) MHRW

Fig. 3. Percentages of repetitive node samples (PRS) generated averagely by a single-run simulation of the SRW and MHRW methods over the four datasets detailed in Section 5 as a function of the sampling budget.



(a) NBRW

(b) CNRW

Fig. 4. Percentages of repetitive node samples (PRS) over the four datasets with NBRW and CNRW as a function of sampling budget. Despite of their ability to keep track of a subset of sampled node, both methods still generate high percentages of repetitive samples because they do not change the sampling paths of SRW fundamentally.

described in Section 5. The high PRS values imply that SRW walks over the same small connected subgraphs repeatedly during its sampling process.

Metropolis–Hasting random walk (MHRW). MHRW [16,17] was proposed to address SRW's problem of being biased to nodes with high degrees based on the converged sampling probability [30]. In MHRW, the sampling probability of the node μ converging to the stationary distribution is $\pi = \frac{1}{|V|}$ [16], where V is the node set of a graph. MHRW first selects an initial node randomly, and then continues to select the next sample randomly from the neighbors of either a previously sampled node or the current sample one according to a transition probability that is different from that of SRW, until the termination conditions are met. The probability of sampling node ν given the previously sampled node μ in MHRW is defined as follows.

$$P^{MHRW}_{(\mu,\nu)} = \begin{cases} \frac{1}{deg(\mu)} \cdot min(1, \frac{deg(\mu)}{deg(\nu)}) & \text{if } \nu \neq \mu, \\ 1 - \sum_{\theta \neq \mu} P^{MHRW}_{(\mu,\theta)} & \text{if } \nu = \mu. \end{cases}$$

Compared with SRW, MHRW worsens SRW's problems of long mixing time and high percentage of repetitive samples. This is because in MHRW nodes with low degrees will have higher probability of being sampled than those in SRW [16], which leads to higher probability of backtracking to superstructures and already sampled nodes. As showed in Fig. 3(b), the percentage of repetitive samples in MHRW ranges from 38% to 58% across the different datasets.

Existing RaWS methods, such as non-backtracking random walk (NBRW) [17] and circulated Neighbors random walk (CNRW) [31], address the problem of long mixing time by making a very small part of the nodes or paths irreversible but ignoring the long paths among the nodes. NBRW avoids backtracking to the previously sampled node while CNRW prevents backtracking to any two consecutively sampled paths. However, they cannot change the random walker's paths fundamentally because there are still many superstructures in the paths. Thus, the mixing time is still large, resulting in huge sampling costs of acquiring useful samples. Furthermore, NBRW and CNRW still produce many repetitive samples. Figs. 4(a) and 4(b) show that neither NBRW or CNRW can significantly cut down the repetitive samples because of their limited sampling spaces during their respective sampling processes. Note that CNRW's PRS is similar to that of SRW because many nodes are still sampled repeatedly by all unblocked paths.

Motivation. In addition to the problems of long mixing time and repetitive samples analyzed above, existing RaWS methods do not differentiate the degree and clique structures during their sampling processes, resulting in inaccurate estimations when characterizing the two different node structures. For example, the sampling probabilities of superstructures, which are necessary measures used to estimate the clique structures, are different from the sampling probabilities of the nodes. In fact, the probability of the former is larger than that of the latter given that a superstructure can be found through other member nodes in the superstructure (detailed in Section 3). Thus, the existing RaWS methods are unable to estimate the degree and clique attributes simultaneously and accurately. These problems motivate us to propose a dual random-walk based sampling method, DRaWS, which can cut down the mixing time and the repetitive samples significantly while differentiating the sampling probabilities of the two structures.

3. Design and analysis of DRaWS

In this section, we first introduce the definition of a hybrid superstructure-based graph including nodes and superstructures to facilitate the design of the DRaWS method. Then, we analyze DRaWS in a more formal manner to explain why it can be regarded as a dual random-walk based model from the perspectives of the node and superstructure respectively. Finally, DRaWS is analyzed to reveal its advantages over the existing RaWS methods in terms of the quality of the samples and the mixing time.

3.1. A hybrid superstructure-based graph

Superstructures, formed by the maximum cliques of the nodes, can be used to simplify the representation of a large graph without self-loops, $G = (V, E)$, where V and E are the sets of nodes

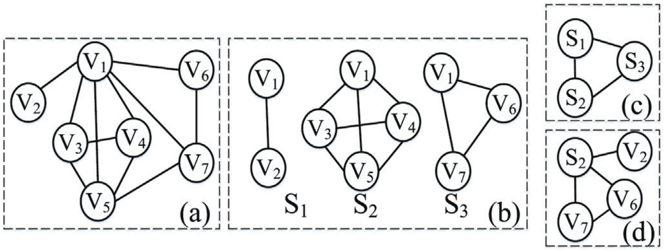


Fig. 5. (a) an example graph, (b) three superstructures formed by three corresponding cliques, (c) a superstructure-based graph, (d) a hybrid graph consisting of both nodes and superstructures when the random walker is residing in V_1 (S_2).

Table 1
The definitions used in this paper.

$G = (V, E)$	graph G
$\|V\|$	Number of nodes in G
$\|E\|$	Number of edges in G
$nei(\mu)$	Set of neighbors of the node μ
$deg(\mu)$	Degree of the node μ
$subG(\mu)$	μ's neighbors and the edges among them
$S(\mu)$	Superstructure formed by $\mu's$ subgraph
$neiS(S(\mu))$	Set of neighbor superstructures of $S(\mu)$
$NeiNode(S(\mu))$	Set of neighbor nodes of $S(\mu)$
$UNeiNode(S(\mu))$	Sampling set of DRaWS
B	Sampling budget

and edges respectively. For example, the original graph in Fig. 5(a) is simplified into Fig. 5(c) by the three superstructures from the corresponding cliques shown in Fig. 5(b). Thus, 7 nodes and 11 edges are simplified to 3 superstructures and 3 edges. Although a superstructure-based graph can help simplify the original graph greatly, it is time-consuming to construct a pure superstructure-based graph before the specific sampling process. Then, a hybrid superstructure-based graph based on the sampled nodes not all the nodes in a large graph during DRaWS's process is constructed to reduce the costs. Furthermore, the existence of both the nodes and the cliques in the hybrid superstructure-based graph is to support the dual residence of the random walker as described in Section 1.

In this paper, a large graph is simplified by its corresponding hybrid superstructure-based counterpart in two steps during the sampling process, identifying superstructures and neighboring nodes of each superstructure.

Identifying the superstructures in a large graph. To simplify a large graph while retaining its key structural properties for the purpose of sampling, we are interested in superstructures that are conducive to learning the clique structures. For each node being sampled (i.e., current sample node), we identify its maximum clique as a superstructure to simplify the graph.

Identifying the neighboring nodes of a superstructure. Once a superstructure is constructed, the set of the unique neighbors of the nodes in the superstructure are the neighboring nodes of the superstructure. Notice that only one edge is preserved between a superstructure and its neighboring node even if there are more than one edge between the node and the nodes in the superstructure. For the example of Fig. 5(a), suppose that the random walker is residing in node V_1. Since S_2 is $V_1's$ maximum clique and hence its superstructure, $S_2's$ unique neighboring nodes are V_2, V_6 and V_7. Thus, from the viewpoint of the currently sample node V_1, the origin graph in the form of Fig. 5(a) is simplified into that in Fig. 5(d), a hybrid superstructure-based graph, which in fact is equivalent to the pure superstructure-based graph in Fig. 5(c) if V_2 and V_6 are sampled in the future and thus S_1 and S_3 are constructed accordingly. Therefore, a hybrid superstructure-based graph includes both nodes and superstructures and is conducive to estimating both the degree and clique structures accurately and simultaneously.

Furthermore, in a hybrid superstructure-based graph, there is a dual many-to-one relationship among nodes inside a superstructure.

The first many-to-one relationship, referred to in this paper as ***many-to-one formation***, stems from the fact that the same superstructure can be identified from at least one but often more than one nodes since the same clique can be the maximum clique of more than one node. As shown in Fig. 5, the superstructure S_2 formed by the four nodes V_1, V_3, V_4 and V_5 can be identified through any of the four nodes since all the four nodes share the same maximum clique.

The second many-to-one relationship, referred to in this paper as ***many-to-one representativeness***, reflects the equal or similar representativeness of the nodes within a superstructure as these nodes are completely connected. In other words, while the superstructure is identified via only one node (μ), other nodes in the same superstructure ($S(\mu)$) exhibit same or similar structures or attributes as μ. Thus, other nodes in S(μ) can be adequately represented by node (μ), making the latter an efficient and accurate representation of the former for the purpose of sampling.

3.2. Design

To avoid backtracking in DRaWS, the neighboring nodes of each superstructure $S(\mu)$ are divided into two groups. The first contains the unique neighboring nodes of the superstructures that have not been sampled and is used as the sampling set, denoted as $UNeiNode(S(\mu))$, while the neighbors that have already been sampled form the second group so as to avoid repetitive samples. The key to the DRaWS process is thus to select the next sample node randomly from the nodes in the first group. As shown in Fig. 5(d), the next sample node is randomly selected from its unvisited neighbors $\{V_2, V_6, V_7\}$ when the random walker is residing in V_1 (S_2). The key notations and their definitions used in this paper are given in Table 1.

To obtain the specific clique size when obtaining the sample set, it is necessary to find the maximum clique of the node μ being sampled (current residence of the random walker) during a typical sampling process. In this paper, we use the algorithm *FindMaxClique* proposed in [32] to find $\mu's$ maximum clique because it can set the upper bound of the clique size dynamically and has no restrictions on the types of the graphs. This is extremely valuable for limiting the processing time of finding the maximum cliques of nodes especially for very large graphs. Algorithm 1 shows DRaWS's sampling process, where the function '$randomSelect(UneiNode(S(\mu_i)))$' is to select one node randomly from $UneiNode(S(\mu_i))$ and the function *FindMaxClique* detailed in Line 3–8 in Algorithm 1 is used to find $\mu_i's$ maximum clique. According to the study in [32], the time complexity of *FindMaxClique* is less than $O(deg(\mu_i) \times deg(\mu_i))$. Note that if there are many completely connected subgraphs consisting of the same number of nodes related to a certain node, the first discovered by the function *FindMaxClique* is labeled as the its maximum clique in practice. The other algorithms [33–35] of finding cliques can be easily employed with minor modifications to obtain clique structures during a sampling process.

Time complexity. Assume that the cost for acquiring one neighbor of a node is O(1). During a typical RaWS sampling process, the time complexity for obtaining the degree of a node (i.e., μ) can be described as $O(deg(\mu))$ while that for acquiring its degree structure can be described as $O(deg(\mu) \times deg(\mu))$. In each

Algorithm 1: DRaWS

```
Input: Sampling budget $B$
Output: Samples : $\mu_1, \mu_2, ..., \mu_B$;
$\mu_0 \leftarrow$ Initialize a node randomly from the graph;
for $i \leftarrow 0$ to $B$ do
    $N \leftarrow$ the number of the nodes in $nei(\mu_i)$;
    $nei[N] \leftarrow$ the neighbors of $\mu_i$;
    $conn[N][N] \leftarrow$ the edge set containing node connection relationship of $nei(\mu_i)$, e.g., there is an edge between $\alpha_i$ and $\alpha_j$ ($\alpha_i, \alpha_j \in nei(\mu_i)$), $conn[i][j] = 1$;
    $Tlimits \leftarrow 0.025$ ;
    /* $Tlimits$ is set as the limit of the fraction of the current steps of calculating and sorting the degrees of the nodes to the total steps which are required to find the clique $S(\mu)$*/;
    $S(\mu_i) \leftarrow FindMaxClique(conn, N, Tlimits)$;
    $\mu_i \leftarrow hasVisited$;
    $\mu_{i+1} \leftarrow randomSelect(UneiNode(S(\mu_i)))$;
```

Table 2
Time complexities of RaWS and DRaWS for estimating the degree and clique structures respectively.

	Degree structures	Clique structures
RaWS	$\sum_{i=1}^{i=T_R} O(deg(\mu_i))$	$\sum_{i=1}^{i=T_R} O(deg(\mu_i) \times deg(\mu_i))$
DRaWS	$\sum_{i=1}^{i=T_D} O(deg(\mu_i) \times deg(\mu_i))$	$\sum_{i=1}^{i=T_D} O(deg(\mu_i) \times deg(\mu_i))$

sampling step, the cost for existing RaWS methods to prepare the sample set for the next sample can be considered as $O(deg(\mu))$, also covering the cost of obtaining the degree structure. Whereas, the cost for the DRaWS method to select the next sample is covered by that of obtaining the clique structure of the currently sampled node (i.e., μ) by Function *FindMaxClique* in Algorithm 1, which is at most $deg(\mu) \times deg(\mu)$. Table 2 describes the time complexities of RaWS and DRaWS where T_R is the total number sampling steps for existing RaWS while T_D is that for DRaWS.

As elaborated later in this section about the high quality of samples and the significantly reduced hitting time from one node to another in DRaWS, the sampling steps required by DRaWS is much fewer than those by RaWS to estimate a large graph accurately and hence $T_D \lll T_R$. Consequently, when both the degree and clique structures are required to be estimated, DRaWS will incur much lower costs than RaWS. Even if DRaWS is only required to estimate the degree structure of a large graph, it still incurs no more costs than RaWS while increasing the accuracy.

3.3. Analysis

Formally speaking, DRaWS uses superstructures to construct a higher-order and irreversible Markov Chain by remembering the already sampled nodes. With DRaWS, although the sampled superstructures can be backtracked by not-yet-sampled member nodes, the sampled nodes will not be backtracked. This helps explain why the order of the Markov chain process based on DRaWS is no more than the sampling budget (B) from the angle of superstructures. For example, for a given sampling budget B, the process of DRaWS can be described as an $m - th(m \leq B)$ order Markov model as follows,

$$\begin{aligned} &P(X_n = s_n | X_{n-1} = s_{n-1}, \ldots, X_1 = s_1) \\ &= P(X_n = s_n | X_{n-1} = s_{n-1}, \ldots, X_{n-m} = s_{n-m}), \end{aligned} \tag{7}$$

where i $(1 \leq i \leq n)$ is the order of the samples, X_i is the state of the Markov Chain and s_i denotes the sampled node along with the formed superstructure. Based on the sampling process of DRaWS described above, the transition probability of DRaWS's Markov chain process from one superstructure to a node in a hybrid graph is given as follows, where the requirement for ① in Eq. (8) is $\nu \in UNeiNode(S(\mu))$.

$$P^{DRaWS}_{(S(\mu),\nu)} = \begin{cases} \frac{1}{|UNeiNode(S(\mu))|} & \text{if ①,} \\ 0 & \text{Otherwise.} \end{cases} \tag{8}$$

When all the neighboring nodes of the superstructures have been sampled, the sampling process will be re-initialized. Usually, the probability of re-initialization decreases with the increase in the graph size. In our experiments, the maximum probability of reinitialization is 1% and the minimum probability is 0.1% with different sampling budgets across the four tested datasets. Such a small probability of re-initialization will arguably not impact the effectiveness of DRaWS noticeably.

Although the stationary distribution of the higher-order Markov chain exists as explained in [26], we do not discuss its specific value because it is non-trivial to compute and it is not used in DRaWS to re-weight samples for estimations in a large graph. Whereas, we analyze DRaWS's process based on the hypothesis that DRaWS can backtrack to the sampled nodes (called a reversible DRaWS) to weight the samples with a theoretical analysis presented in Section 4. Since DRaWS walks from one superstructure to one of its neighboring nodes, the unit of the residence of DRaWS's random walker can be considered as a dual random-walk based model. We analyze the dual random-walk based model from the perspective of the reversible Markov chain as follows.

First, a Markov-chain based sampling process, which traverses a large graph from one node to another node, is hidden in DRaWS's reversible process. The hidden reversible sampling process stays on the node μ with the probability $p_n = \frac{1}{|S(\mu)|}$, and then transits to one of μ's neighbors labeled as ν, except the nodes in the superstructure of μ with the probability $p'_n = \frac{1}{deg(\mu)+1-|S(\mu)|}$. On the other hand, the hidden reversible sampling process stays on the nodes in $S(\mu)$ other than μ with the probability $p_{nn} = \frac{|S(\mu)|-1}{|S(\mu)|}$ and then transfers to another node except $\mu's$ neighbors in $NeiNode(S(\mu))$ with the probability $p'_{nn} = \frac{1}{|NeiNode(S(\mu))|-(deg(\mu)+1-|S(\mu)|)}$. Thus, the transition probability of the hidden sampling process based on DRaWS's reversible sampling process, is given as follows.

$$P^{node}_{(\mu,\nu)} = \begin{cases} p_n \times p'_n & \text{if } \nu \in nei(\mu), \\ p_{nn} \times p'_{nn} & \text{if } \nu \notin nei(\mu) \& \ \nu \in NeiNode(S(\mu)), \\ 0 & \text{otherwise.} \end{cases} \tag{9}$$

Second, DRaWS's reversible process can be considered as traversing a large graph from one superstructure to one of its neighboring superstructures. The transition probability of such a reversible Markov chain process from one superstructure to one of its neighboring superstructures, labeled as $P^{revesible}_{(S(\mu),S(\nu))}$, from $S(\mu)$ to $S(\nu)$ is given as follows and the requirement for ① in Eq. (10) is $\nu \in NeiNode(S(\mu))$.

$$P^{revesible}_{(S(\mu),S(\nu))} = \begin{cases} \frac{m(\nu)}{|NeiNode(S(\mu))|} & \text{if ①,} \\ 0 & \text{otherwise,} \end{cases} \tag{10}$$

where $m(\nu)$ is the number of the nodes in $NeiNode(S(\mu))$ that share the same superstructure of $S(\nu)$.

Given the many-to-one formation between the nodes and its superstructure, the sampling probabilities of the nodes and the superstructures are different. Eqs. (9) and (10) reflect the two different transition probabilities, which can reflect the different sampling probabilities in both the degree and the clique structures respectively. Based on the dual random-walk sampling process of DRaWS, we design two estimators to estimate the different node structures as described in Section 4.

3.4. Improvements over RaWS methods

DRaWS improves the quality of samples by removing repetitive samples and increasing the chances of obtaining representative samples. Based on the DRaWS's sampling process, each node in a large graph has a chance of being the basis for constructing at least one superstructure that in turn has a chance to be sampled. However, not all the nodes have the same, or any chance to be sampled. For example, only the node through which a sampled superstructure is constructed/identified will be sampled, while all other nodes sharing the same superstructure will be hidden and not directly sampled. However, the one-to-many representativeness relationship between the sampled node and the nodes in its superstructure imply that the these hidden nodes in the clique are adequately represented by the sampled node, thus effectively increasing the number of nodes accurately represented by the obtained samples in a large graph for a given sampling budget.

Besides for the high quality of samples it produces, DRaWS employs three important strategies to reduce the mixing time of the Markov-chain based sampling process. Since SRW is widely used in many studies and lays foundations for many variations of random-walk based methods, we take SRW as a representative of RaWS methods to show DRaWS' improvements over RaWS methods.

First, DRaWS shortens the length of the paths of the random walker by using the superstructures. As stated in Section 2, the length of the paths along which the random walker traverses a large graph affects the mixing time of the Markov-chain based sampling process.

Let $L(\mu,\nu)_{SRW}$ be a path from node μ to node ν without self-loops in SRW's process.

$$L(\mu,\nu)_{SRW} = (\mu,\alpha_1) + (\alpha_1,\alpha_2) + \cdots + (\alpha_n,\nu), \tag{11}$$

where $\alpha_1 \in nei(\mu)$, $\alpha_{n-1} \in nei(\alpha_{n-2})$ and $\nu \in nei(\alpha_n)$ and n is the total number of the intermediate nodes along $L(\mu,\nu)_{SRW}$. Based on the dual residence of the random walker described in Section 1, the path from μ to ν in SRW expressed in Eq. (11) can be simplified in DRaWS into one from one superstructure to another superstructure, $L(S(\mu),S(\nu))_{DRaWS}$, which can be described as follows.

$$\begin{aligned} &L(S(\mu),S(\nu))_{DRaWS} \\ &= (S(\mu),S(\beta_1)) + (S(\beta_1),S(\beta_2)) + \cdots + (S(\beta_m),S(\nu)), \end{aligned} \tag{12}$$

where $S(\beta_1) \in neiS(S(\mu))$, $S(\beta_{m-1}) \in neiS(S(\beta_m))$ and $S(\nu) \in neiS(S(\beta_m))$ and m is the total number of the intermediate superstructures in $L(S(\mu),S(\nu))_{DRaWS}$. The relationship between $L(\mu,\nu)_{SRW}$ and $L(S(\mu),S(\nu))_{DRaWS}$ is discussed in the following two cases.

(a.) Suppose $\alpha_1 \in S(\mu)$, then $\alpha_2 \in nei(\mu)$, $\alpha_2 \in neiS(S(\mu))$ and $\alpha_2 \notin S(\mu)$ which means that the path in DRaWS leverages the bridge of superstructure of $S(\mu)$ to jump to the neighbor of μ directly. Thus the part of the path labeled as $(\mu,\alpha_1)+(\alpha_1,\alpha_2)$ in SRW can be replaced by $(S(\mu),S(\beta_1))$ in DRaWS.

$$(\mu,\alpha_1) + (\alpha_1,\alpha_2) \Longrightarrow (S(\mu),S(\beta_1)) \tag{13}$$

(b.) Suppose $\alpha_1 \notin S(\mu)$, then $\alpha_1 \in nei(\mu)$ and $\alpha_1 \in neiS(S(\mu))$ which means that SRW does not choose the nodes in the superstructure as the next residence of its random walker when the latter resides on μ. However, only if all the other intermediate nodes in path $L(\mu,\nu)_{SRW}$ choose the next residence of the random walker in the same way as μ does, will $L(\mu,\nu)_{SRW} = L(S(\mu),S(\nu))_{DRaWS}$. Otherwise, $L(\mu,\nu)_{SRW} > L(S(\mu),S(\nu))_{DRaWS}$ according to Case (a). The probability that $L(\mu,\nu)_{SRW} = L(S(\mu),S(\nu))_{DRaWS}$ is given as,

$$\begin{aligned} &Pr(L(\mu,\nu)_{SRW} = L(S(\mu),S(\nu))_{DRaWS}) \\ &= \frac{deg(\mu)+1-|S(\mu)|}{deg(\mu)} \times \cdots \times \frac{deg(\alpha_n)+1-|S(\mu)|}{deg(\alpha_n)} \end{aligned}$$

$Pr(L(\mu,\nu)_{SRW} = L(S(\mu),S(\nu))_{DRaWS})$ approaches to zero with the increase in the length of $L(\mu,\nu)_{SRW}$. Thus, DRaWS has the ability to reduce the length of the path traversed by the random walker, in contrast to the existing RaWS methods.

Second, DRaWS reduces the number of the paths of the random walker. Suppose that there are r_1 nodes forming a superstructure $S(\mu)$. Then, there are r_1 types of different paths corresponding to the superstructure for the random walker in SRW, which can be replaced by one path with the help of the superstructure in DRaWS. Suppose that SRW's random walker needs to traverse a large graph from node μ to node α in superstructure $S(\beta)$ through the nodes in $S(\mu)$, where $S(\beta)$ and $S(\mu)$ are neighboring superstructures, which implies multiple possible paths. In DRaWS, however, the random walker traverses from $S(\mu)$ to superstructure $S(\beta)$ in exactly one path. According to Eq. (3) in Section 2, the hitting time ($H^{SRW}_{(\mu,\alpha)}$) from μ to α in SRW is given as:

$$\begin{aligned} H^{SRW}_{(\mu,\alpha)} &= \frac{1}{r_1}\sum_{i=1}^{i=r}\frac{1}{P^{SRW}_{(\mu,\mu_i)} \times P^{SRW}_{(\mu_i,\alpha)}} \\ &= \frac{1}{r_1}\sum_{i=1}^{i=r}(deg(\mu)\times deg(\mu_i)) \end{aligned} \tag{14}$$

where $\mu_i \in S(\mu)$. Similarly, the hitting time ($H^{DRaWS}_{(\mu,\alpha)}$) from $S(\mu)$ to $S(\beta)$ in DRaWS is given as,

$$H^{DRaWS}_{(S(\mu),S(\beta))} = \frac{1}{P^{reversible}_{(S(\mu),S(\beta))}} = |NeiNode(S(\mu))|. \tag{15}$$

Therefore, we have:

$$H^{SRW}_{(\mu,\alpha)} - H^{DRaWS}_{(S(\mu),S(\beta))} > deg(\mu) \times |S(\mu)|. \tag{16}$$

Suppose that a path in DRaWS from $S(\mu)$ to $S(\nu)$ has k 'bridge' superstructures labeled as $S(\beta_1),\ldots,S(\beta_k)$. Then, this path can replace $r_1 \times \cdots \times r_k$ paths from μ to ν in SRW. Therefore, we have:

$$H^{SRW}_{(\mu,\nu)} - H^{DRaWS}_{(S(\mu),S(\nu))} > \prod_{i=1}^{i=k} deg(\beta_i) \times |S(\beta_i)|. \tag{17}$$

Therefore, DRaWS is able to cut the hitting time from one node (superstructure) to another node (superstructure) and thus the mixing time by cutting the number of the paths of the random walker.

Third, DRaWS reduces the likelihood of reversibility to further reduce the mixing time by avoiding backtracking to the already sampled nodes. The non-backtracking strategy reduces the length of a path and the number of paths from one node to another by avoiding self-loops and thus cuts down the mixing time of a Markov-chain based sampling process.

4. Estimator

To quantitatively analyze the structural properties of a large graph through samples obtained by a typical RaWS method, it requires an estimator that takes the samples as the input and outputs the estimations of the graph's structural properties. Since DRaWS has obviously different transition processes from the existing RaWS methods, the estimators proposed by the RaWS methods could not be directly used in the DRaWS's process.

To propose a proper estimator corresponding to DRaWS, we first introduce and analyze the most frequently used estimator, referred to as the default estimator for RaWS, and discuss why it is unsuitable for analyzing samples obtained by DRaWS. To facilitate the analysis and development of appropriate, suitable estimators for DRaWS, we then introduce two other important estimators, the Horvitz–Thompson estimator and an unordered estimator, that are usually used to analyze samples obtained with unequal sampling probabilities. Integrating these two estimators with appropriate weights assigned to samples obtained by DRaWS, we introduce re-weighted estimators for DRaWS.

The default estimator works as follows. If a property of the sampled element is defined as $pro(\mu)$ with a value of k, then the function $1(pro(u) = k) = 1$ is established. Otherwise, $1(pro(u) = k) = 0$. The estimated distribution $\tilde{\omega_k}$ of the property $pro(\mu)$ can be calculated as follows.

$$\tilde{\omega_k} = \frac{1}{B}\sum_{\mu=1}^{|B|} 1(pro(\mu) = k). \tag{18}$$

The default estimator treats all obtained samples equally without considering the sampling biases in terms of unequal sampling probabilities of most of random-walk based sampling processes including RaWS and DRaWS. In DRaWS, for example, a superstructure with more member nodes generally has a larger sampling probability than one with fewer member nodes. The unequal sampling probabilities make the default estimator inaccurate in its estimation of the graph properties.

The unequal sampling probabilities can be analyzed from two angles. The first is to consider the entire sampling process of DRaWS as a higher-order Markov chain process as described in Eq. (7). The second is to focus on a version of the sampling process in which units already sampled will no longer participate in the subsequent sampling process, a process we refer to as *sampling without replacement*. The Horvitz–Thompson estimator can be used to analyze samples from the higher-order Markov chain [21] and the unordered estimator is used to analyze the samples in sampling without replacement [22].

Horvitz–Thompson estimator, proposed by Daniel G. Horvitz and Donovan J. Thompson [21], is an unbiased estimator highly efficient for random-walk based graph sampling [36]. Suppose a sample μ in a large graph is selected with the converged sampling probability $\pi(\mu)$ and the range of the property $pro(\mu)$ is $\{\alpha_1, \dots, \alpha_k\}$. Then, by using the Horvitz–Thompson estimator, the distribution $\tilde{\omega_k}$ of α_k with the sampling budget B can be obtained by acquiring the expectation $E(\tilde{\omega_k})$ as follows.

$$E(\tilde{\omega_k}) = \frac{1}{B}\sum_{\mu=1}^{|B|} \frac{1(pro(\mu) = \alpha_k)}{\pi(\mu)} \tag{19}$$

Furthermore, when the specific value of $\pi(s)$ is complicated to obtain, the Horvitz–Thompson estimator, which can be transformed to another form [36], is used to weight the samples with the transition probability $p(s)$ of each sampling step instead of $\pi(s)$ as follows.

$$\tilde{\omega_k} = \frac{1}{W}\sum_{\mu=1}^{|B|} 1(pro(\mu) = \alpha_k)\cdot p(\mu), \tag{20}$$

where $W = \sum_{\mu=1}^{|B|} p(\mu)$, $s \in G$.

Unordered estimator, proposed in [22], is designed to estimate the properties of samples obtained by sampling processes without re-placement by ignoring the arrival orders of samples in its estimation. Suppose there are n items sampled without replacement from a certain set with a total number of N items. If the samples are seen as unordered, there are C_N^n types of unordered samples labeled as $x_s\{s = 1, \dots, C_N^n\}$. For any given type of unordered samples, there are $n!$ types of ordered samples. Let $g_{si}\{s = 1, \dots, C_N^n, i = 1, \dots, n!\}$ be the set of ordered samples. Based on the sampled items, the ordered estimator θ_{Ok} and the unordered estimator θ_{Uk} are used to estimate a certain property of the sampling set (labeled as pro) by computing their expectations $E(\theta_{Ok})$ and $E(\theta_{Uk})$ respectively as follows.

$$\begin{aligned} E(\theta_{Ok}) &= \sum_{s=1}^{C_N^n}\sum_{i=1}^{n!} 1(pro(g_{si}) = k)\cdot p_{si}, \\ E(\theta_{Uk}) &= \sum_{s=1}^{C_N^n} 1(pro(x_s) = k)\cdot p(s) \end{aligned} \tag{21}$$

where p_{si} and $p(s)$ are defined as the sampling probability of the ordered samples and the unordered samples respectively and k is one of the value of the sample's property. In sampling without replacement, the unordered samples are obtained, requiring that all the related types of the ordered samples are obtained. The relationship of the sampling probability p_{si} and $p(s)$ can be described as follows.

$$p(s) = \sum_{i=1}^{n!} p_{si} \tag{22}$$

The following relationship holds and the proof can be found in [22].

$$E(\theta_{Ok}) = E(\theta_{Uk}) \tag{23}$$

When using the Horvitz–Thompson estimator, it is required to compute the stable transition probability of a sample (μ) by considering its arrival orders, which is complicated for DRaWS's process especially for a large graph. When using the unordered estimator, it is required to learn the specific values of $S(\mu)'s$ and $\mu's$ sampling probabilities. However, it is time-consuming to compute these specific values. For example, based on the description in [29], $\pi(S(\mu)) = \frac{|NeiNode(S(\mu))|}{m(\mu)\sum_{\mu\in V}|NeiNode(S(\mu))|}$ is $S(\mu)'s$ sampling probability whose denominator is the sum of the numbers of the neighboring nodes of all nodes in a large graph. The case for $\mu's$ sampling probability is the same as that for $S(\mu)$. Thus, neither Horvitz–Thompson estimator nor unordered estimator can be used to weight the samples directly obtained by DRaWS with low complexity. Thus, we propose a new re-weight estimator based on the ideas of these two estimators as follows.

Re-weighted estimator. Since DRaWS is designed to sample without replacement, the sampling probability of a sample μ obtained based on the higher-order Markov chain process converges to a fixed value $\pi(\mu)$ by considering all of μ's possible arrival orders. Based on the analysis of the unordered estimator, $\pi(\mu)$ can be replaced by the converged sampling probability $p(\mu)$ without considering μ's arrival orders. Then the Horvitz–Thompson estimator can be employed as follows.

$$E(\tilde{\omega_k}) = \frac{1}{B}\sum_{\mu=1}^{|B|} \frac{1(pro(\mu) = \alpha_k)}{p(\mu)} \tag{24}$$

Then, we use Eq. (20) to convert the original Horvitz–Thompson estimator by replacing the sampling probability $p(\mu)$ with $\mu's$ transition probability $p(us)$, resulting in a new estimator referred to in this paper as the *re-weighted estimator*

$$\tilde{\omega_k} = \frac{1}{W}\sum_{\mu=1}^{|B|} 1(pro(\mu) = \alpha_k)\cdot p(us), \tag{25}$$

where $W = \sum_{\mu=1}^{|B|} p(us)$, $\mu \in G$.

For DRaWS, both the degree and clique size are required to be estimated. However, since the transition probabilities of the two types of node structures are different, they require different re-weighted estimators, as described below.

4.1. Weights for the degree structure

When the properties of the degree structure are analyzed, the samples are studied from the perspective of nodes. Thus, the transition probability of the reversible DRaWS's process from one node to another is used to re-weight the sampled nodes. The algorithm for estimating the properties of degree structures through analyzing the samples obtained by DRaWS is similar to Algorithm 2 by replacing the weights $P^{revesible}_{(S(\mu),S(\nu))}$ of $P^{node}_{(\mu,\nu)}$ which is described in Eq. (9) in Section 3.

$$\tilde{\omega_k} = \frac{1}{SB}\sum_{\nu=1}^{B} 1(pro(\mu) = k) \cdot P^{node}_{(\mu,\nu)}, \tag{26}$$

where $SB = \sum_{\mu=1}^{B} P^{node}_{(\mu,\nu)}$.

4.2. Weights for the clique structure

When the properties of the cliques are estimated, the samples are studied from the perspective of superstructures, requiring the transition probability of each sampled superstructure in DRaWS. Therefore, the samples are weighted according to the transition probability $P^{revesible}_{(S(\mu),S(\nu))}$ (Eq. (10)) of the 1st order Markov chain. Furthermore, the sampled superstructure ($S(\mu)$) is obtained from only one of its member nodes in $S(\mu)$ by obtaining its maximum clique. Thus, the probability of one node in $S(\mu)$ being selected to obtain the superstructure is $\frac{1}{|S(\mu)|}$. Based on Eq. (25), the re-weighted estimator for high-order node attributes is described as follows and detailed in Algorithm 2 analyze the properties of the clique structures.

$$\tilde{\omega_k} = \frac{1}{SUM}\sum_{\mu=1}^{B} 1(pro(S(\mu) = k) \times \frac{1}{|S(\mu)|}) \cdot P^{revesible}_{(S(\mu),S(\nu))}, \tag{27}$$

where $SUM = \sum_{\mu=1}^{B} P^{revesible}_{(S(\mu),S(\nu))}$.

Algorithm 2: DRaWS's estimator for clique structures

```
Input: Samples: $\mu_1, \mu_2, ..., \mu_B$;
Output: The property of the high-order node structure $pro_j$;
if $|S(\mu)| == j$ then
    $\omega_j \leftarrow \omega_j + \frac{1}{|S(\mu)|} \times P^{revesible}_{(S(\mu),S(\nu))}$;
$totalW \leftarrow totalW + \frac{1}{|S(\mu)|} \times P^{revesible}_{(S(\mu),S(\nu))}$;
$pro_j \leftarrow \frac{\omega_j}{totalW}$;
```

5. Evaluation

This section presents the evaluation of DRaWS through simulation experiments conducted on a computer with Intel Xeon E5620 processors and 64bit Ubuntu Linux OS. For simplicity, a single core is used to evaluate the costs of the sampling processes, although our algorithms can be easily implemented in a parallel and distributed environment. We choose four real-world datasets, summarized in Table 3, that have been frequently used in evaluating sampling algorithms in recently published studies.

Baseline methods. We select four existing RaWS methods as the baseline. These include two fundamental methods based on the 1st order Markov chain: Simple Random Walk (SRW) and Metropolis–Hastings Random Walk (MHRW), and two other state-of-the-art methods based on higher-order Markov chains: non-back-tracking random walk (NBRW) and Circulated Neighbor Random Walk (CNRW).

Sampling steps. The total number of sampling steps for a sampling method is defined to be $T \times B$ where T and B are respectively the number of simulation runs and the sampling budget in a single-run simulation. In this paper, we compare sampling steps of DRaWS and the baseline methods in three different ways. First, for a given number of sampling steps in a single-run simulation, SRW, MHRW, NBRW and CNRW are simulated for 1000 runs over the com-DBLP and amazon0601 datasets. Second, for the com-Youtube and wiki-Talk datasets, the numbers of simulations are 100 and 10 respectively. Third, due to the high efficiency of DRaWS which is able to reduce the number of sampling steps greatly as explained in Section 3.4, only 10 simulations of DRaWS are implemented over the four datasets with a given sampling budget in a single-run simulation.

Table 3
Summary of Graph Datasets.

Graph	$\|V\|$	$\|E\|$
com-dblp [37]	317,080	1049,866
amazon0601 [38]	403,394	3387,388
com-Youtube [37]	1134,890	2987,624
wiki-Talk [39][40]	2394,385	5021,410

Estimator. To keep the consistency of the original studies, we use their respective estimators to weight the samples obtained by the existing RaWS (including SRW [9], NBRW [17] and CNRW [31]). The samples obtained by DRaWS are weighted according to Section 4. Furthermore, MHRW, which is an unbiased sampling method where the samples are obtained with equal sampling probabilities (Section 2), employs the default estimator.

5.1. Estimation error and effectiveness

To quantitatively present the estimation errors of different methods, we adopt the measure of normalized mean square error (NMSE), defined below.

$$NMSE(\tilde{\omega_k}) = \frac{\sqrt{E[(\tilde{\omega_k} - \omega_k)^2]}}{\omega_k}$$

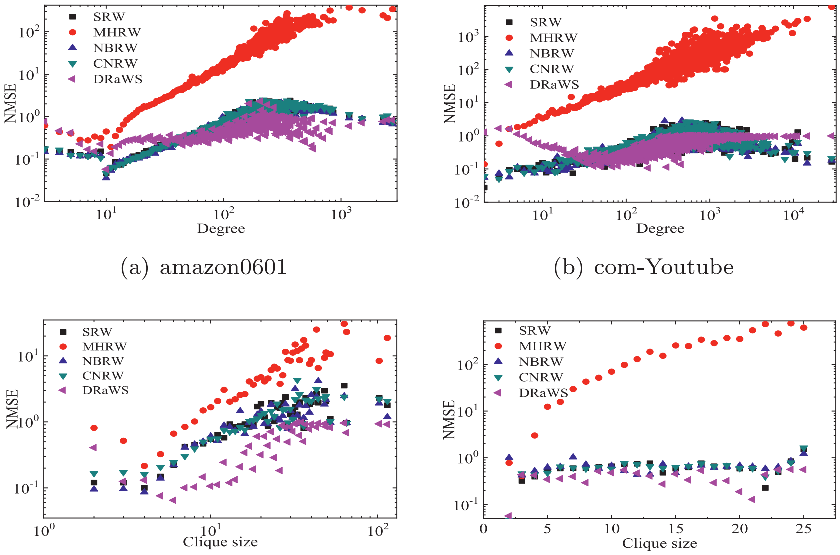
where ω_k and $\tilde{\omega_k}$ are respectively the true and estimated distributions about the graph property labeled as k. Furthermore, $NMSE(\tilde{\omega_k}) \leq 1$ means that property k is effectively estimated as the estimation error is sufficiently small and acceptable [10]. Otherwise, it means that property k cannot be accurately estimated. Assume that there are total N different values of a property among which M values are effectively estimated. Then, the effectiveness of estimating this property is $\frac{M}{N}$ (with 100% being the most effective estimation), which is an important measure for estimation accuracy in addition to estimation errors.

Table 4 shows that DRaWS is consistently the most effective in estimation, with the lowest average estimation errors among the five methods evaluated over the four datasets when estimating the degree and clique structures. Table 4 further confirms that samples obtained by DRaWS are highly representative and can reflect more than 90% original values in a large graph when used to estimate the two node structures. The specific distributions of the two structures over the four datasets are described in Fig. 6 in which most of the values about the two node structures are estimated more accurately than the other four methods.

Furthermore, the five methods are evaluated in terms of the average estimation errors of the distributions of the degree and clique size on com-DBLP and amazon0601 as a function of the sampling budget, shown in Fig. 7. DRaWS is shown to consistently exhibit higher estimation accuracy than the four baseline RaWS methods; whereas, MHRW stands out as a much worse performer than all the other because it produces much more repetitive samples than others (Section 2). This also confirms that the default estimator is not adequate for a random-walk based sampling method because the transition probabilities of nodes in a large graph fluctuate greatly in practice.

Table 4
Estimation accuracy in terms of effectiveness and average estimation error (average NMSE) on the distributions of clique size and node degree, where *Total_C* is the number of different sizes of the clique that the nodes participate in and *Total_D* is the number of different degrees in a large graph. The sampling budgets for com-DBLP, amazon0610, com-Youtube and wiki-Talk are 3500, 4500, 6000 and 10000 respectively.

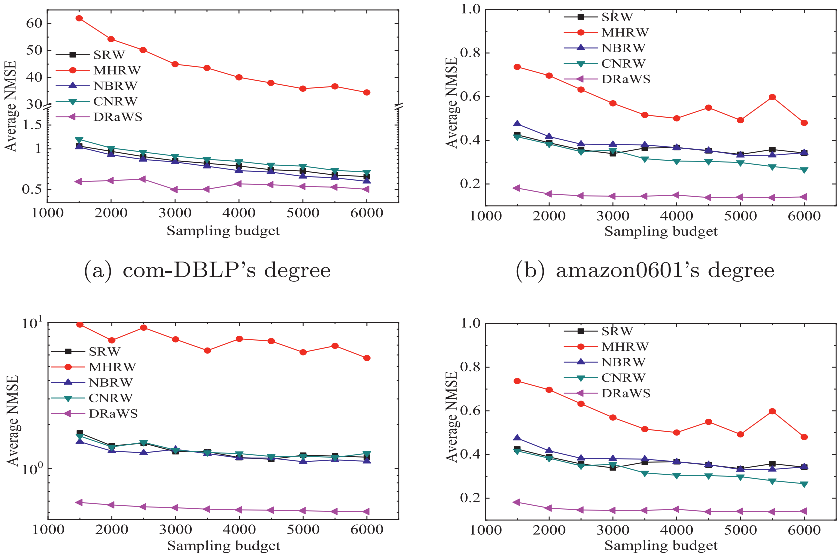
Methods		Clique size		Degree	
		com-DBLP *Total_C* = 47	wiki-Talk *Total_C* = 25	amazon0601 *Total_D* = 346	com-Youtube *Total_D* = 978
SRW	Effectiveness (%) Average NMSE	45 1.31	96 0.61	47 1.05	80 0.74
MHRW	Effectiveness (%) Average NMSE	13 7.56	8 237.8	5 68.15	1 294.3
NBRW	Effectiveness (%) Average NMSE	47 1.23	92 0.67	51 0.99	85 0.69
CNRW	Effectiveness (%) Average NMSE	42.55 1.32	96 0.63	47 1.08	73 0.80
DRaWS	Effectiveness (%) Average NMSE	100 0.53	100 0.39	92 0.57	95 0.64



(a) amazon0601 (b) com-Youtube

(c) com-DBLP (d) wiki-Talk

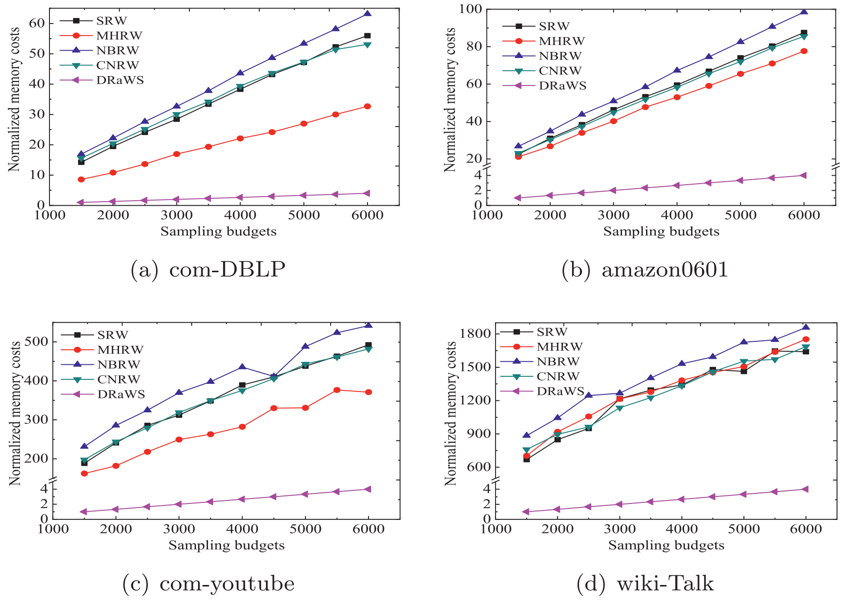
Fig. 6. The estimation errors of the distributions of the degree and clique size.



(a) com-DBLP's degree (b) amazon0601's degree

(c) com-DBLP's clique size (d) amazon0601's clique size

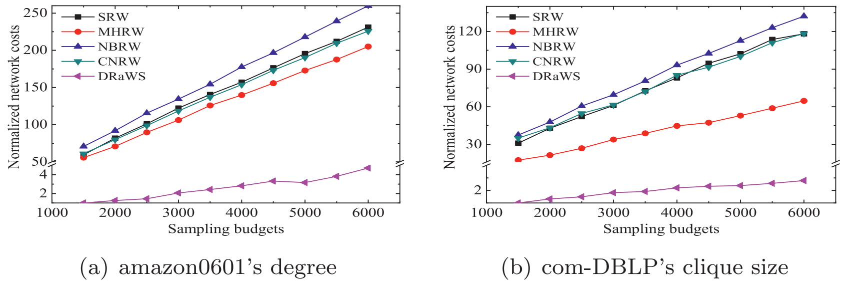
Fig. 7. The average estimation errors of the distributions of the degree and clique size as a function of the sampling budget.



(a) com-DBLP (b) amazon0601

(c) com-youtube (d) wiki-Talk

Fig. 8. The normalized memory costs of different methods as a function of the sampling budget where the memory costs are normalized to the minimum value of DRaWS's memory costs.



(a) amazon0601's degree (b) com-DBLP's clique size

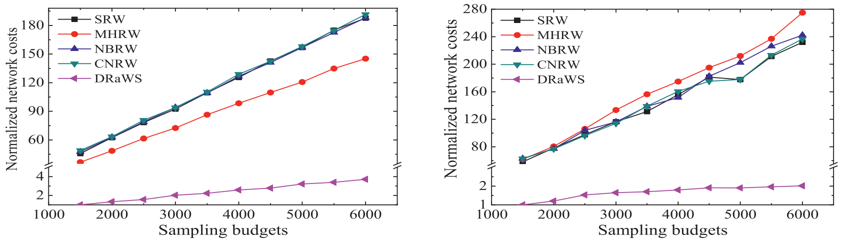
Fig. 9. The normalized network costs with the sampled data saved in memory where the network costs are normalized to the minimum value of DRaWS's network costs.

5.2. Sampling costs

To obtain both the degree and the clique structures, a typical RaWS process in each sample step acquires a node set *nei*(μ) containing potential samples to choose from in the next step based on the previously sampled node μ and a set *Connection*(μ) of edges connecting nodes in the node set. Note that, for the baseline methods, the node set *nei*(μ) consists of the neighbors of μ; whereas for DRaWS, the node set *UneiNode*($S(\mu)$) contains the unvisited neighbors of the superstructure associated with μ. The edge set *Connection*(μ), which is to reflect the relationship among the $\mu's$ neighbors, is used for both existing RaWS and DRaWS to find the maximum clique of the currently sampled node. The two types of sets are used to evaluate the costs in terms of memory and network bandwidth.

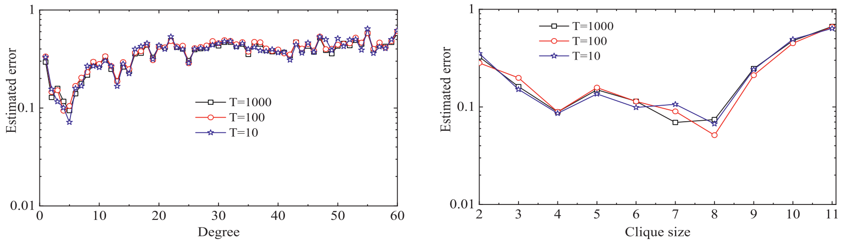
Memory costs. For SRW, MHRW, NBRW and CNRW, the sampling sets of each step can be saved in memory so that it is unnecessary to occupy the network bandwidth to collect them again when the sampled nodes are visited again. In doing so the processing time for dealing with the repetitive samples can also be saved. For DRaWS, it is necessary to record the sampled nodes to avoid backtracking and the repetitively sampled node. Fig. 8 shows that DRaWS is able to significantly reduce the memory usage across all the datasets, by a factor ranging from 13× to 561×, with an average of 186×. It is clear that the amount of reduction in memory usage by DRaWS increases with the graph size. NBRW occupies more memory than SRW, MHRW and CNRW because it increases the chance of selecting the nodes with high degree by avoiding backtracking to the previously sampled nodes. The sampled nodes with a higher degree means that more neighboring nodes should be saved in memory than those with a lower degree.

Network costs. When the sampling methods are used to obtain the distributions of clique size and degree of online networks, the two sampling node sets described above along each sampling step are acquired by the network. Therefore, the cumulative size of the node sets along all sampling steps is used to measure the network costs of obtaining the distribution of node degree while that of edge sets along all sampling steps, which is usually much larger than that of the node sets, is used to estimate the network costs of obtaining the distribution of clique size. Since the network costs are influenced by memory costs as stated above, the former are evaluated, respectively, with or without recording the history information in memory about the previously sampled nodes. Fig. 9 shows that DRaWS cuts down the network costs by a factor from 16× to 48×, with an average of 34×, to obtain the distribution of clique size on com-DBLP and from 43× to 70×, with an average 58×, to obtain that of node degree on amazon0601 when the historical information is saved in memory. Moreover, Fig. 10 shows that DRaWS reduces the network costs from 58× to 136×, with an average of 87×, to obtain the distribution of clique size on amazon0601 and from 36× to 51×, with an average 46×, to obtain that of node degree on com-DBLP when the memory is not used by the comparative methods to store the historical information.



(a) com-DBLP's degree (b) amazon0601's clique size

Fig. 10. The normalized network costs without the sampled data saved in memory where the network costs are normalized to the minimum value of DRaWS's network costs.



(a) com-DBLP (degree) (b) amazon0601 (clique size)

Fig. 11. Estimated errors of DRaWS with different simulation runs.

Time costs. As described in Section 3.4, DRaWS cuts down the mixing time which means that DRaWS can estimate the structures accurately at a small number of simulation runs (T). Fig. 11 shows that even if DRaWS cuts down the number of simulation runs from $T = 1000$ to $T = 10$, it still estimates the properties with no significant degradation of accuracies. Fig. 12 shows that DRaWS consumes the least time costs when these methods are used to estimate both the degree and the clique structures simultaneously. To evaluate the processing time of estimate the clique size and degree respectively and clearly, we evaluate the time costs of the five methods that are normalized to the minimum value of DRaWS's processing time. Fig. 13(b) shows that DRaWS cuts down the processing time of the baseline sampling methods in estimating the distributions by a factor of about 10× on average over com-Youtube. When these five methods are used to estimate the distribution of node degree, Fig. 13(a) shows that DRaWS spends less time than CNRW since the latter needs to avoid backtracking to two consecutive sampling steps (Section 2). However, Fig. 13(a) shows that DRaWS spends lightly more time than other three base-line methods. This is because, for each sampling step, DRaWS must find the superstructure of the sampled node and then collect the neighboring nodes of the sampled superstructure while other methods only collect the neighbors of the most recently sampled node to form the sampling set for the next step. However, such a tiny cost can be more than compensated by the high-quality samples and accurate estimations with DRaWS. Furthermore, Fig. 13 shows that the times that DRaWS cuts down in estimating the clique structure are much more than that in estimating the degree structure. Therefore, when estimating the degree and clique structures simultaneously, DRaWS can cut down the time costs significantly because of its reduced number of sampling steps.

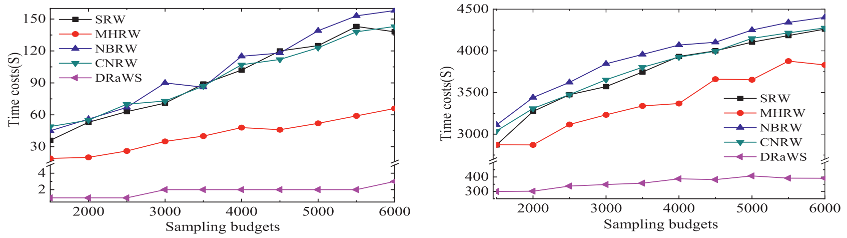
6. Related work

Besides the graph sampling methods described in Section 2, there are other random-walk based sampling methods. Frontier sampling (FS) [10] is proposed to leverage the advantage of multiple uniform walkers to increase the probability of selecting the nodes in the disconnected subgraphs. Researchers in [14] propose the generalized maximum-degree random walk (GMD) to address the problem of SRW biasing to the nodes with higher degrees. Rejection-controlled Metropolis–Hastings (RCMH) is also proposed by [14] to address the problem of MHRW that causes high ratio of repetitive samples by controlling the probability of staying on the previously sampled nodes. Moreover, researchers in [11] propose to skip some nodes without sampling and researchers in [19] design different sampling probabilities according to the connectivity of nodes. However, the existing random-walk based methods cannot essentially change the key steps in SRW and MHRW in which the walker traverses from one node to one of its neighboring nodes and result in many repetitive samples and these methods do not differentiate the two structures. Besides, the existing random-walk based sampling methods [41–43] require a large number of steps, resulting in huge sampling costs especially in estimating the distributions of clique size. Researchers in [44] considered that the samples were produced with different rates to effectively evaluate the imbalanced enterprise credit. However, DRaWS improves the quality of samples greatly by producing representative samples to reflect the influence of social network.

Furthermore, as a remedy to sampling biases, more effective estimators are proposed to obtain the structural properties of large graphs by existing sampling methods (i.e., FS [10] and NBRW [17]). However, these existing estimators cannot be directly employed by other sampling methods, such as DRaWS, whose sampling process is quite different from the existing random-walk based methods. Therefore, we propose two reweighted estimators to better analyze the degree and the clique structures.

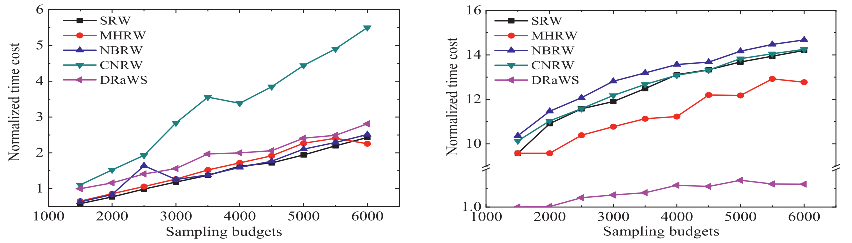
7. Conclusions

This paper proposes a new dual random-walk based sampling method called DRaWS to estimate large graphs fast and accurately. It leverages many-to-one relationships between nodes and superstructures reflected in the formation of the latter in a large



(a) com-DBLP (b) com-Youtube

Fig. 12. The time costs consumed by the five methods when estimating both the degree and clique size simultaneously.



(a) com-DBLP's degree (b) com-Youtube's clique size

Fig. 13. The normalized time costs consumed by the five methods when estimating the degree and clique size respectively.

graph to construct a dual Markov chain so that the sampling process can more accurately reflect different transition probabilities in the degree and clique structures simultaneously, while using the many-to-one representativeness to produce high quality of samples. Moreover, new re-weighted estimators from the perspective of superstructures and nodes are proposed by leveraging the knowledge of the Horvitz–Thompson estimator and the unordered estimator to obtain the structural properties of large graphs accurately. Extensive experiments driven by real-world graph datasets show that DRaWS cuts down the sampling costs dramatically while estimating graph characteristics more accurately than the state-of-the-art sampling methods. In the future, we will try to design different sampling methods, similar to DRaWS, to estimate several other structures (i.e., motif estimations [36,45,46]) accurately and simultaneously with low costs.

Declaration of competing interest

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

CRediT authorship contribution statement

Lingling Zhang: Conceptualization, Methodology, Software, Writing - original draft. **Hong Jiang:** Conceptualization, Writing - original draft. **Fang Wang:** Supervision, Writing - review & editing. **Dan Feng:** Writing - review & editing.

Acknowledgments

We would like to thank the anonymous reviewers of this paper. This work was supported by National Defense Preliminary Research, China Project No. 31511010202, NSFC No. 61832020, No. 61772216, No. 61821003, Wuhan application basic research, China project 2017010201010103.

References

[1] L. Backstrom, D. Huttenlocher, J. Kleinberg, X. Lan, Group formation in large social networks: membership, growth, and evolution, in: SIGKDD, ACM, 2006, pp. 44–54.

[2] M. Gjoka, E. Smith, C. Butts, Estimating clique composition and size distributions from sampled network data, in: INFOCOM WKSHPS, IEEE, 2014, pp. 837–842.

[3] D. Krackhardt, M. Kilduff, Structure, culture and simmelian ties in entrepreneurial firms, Soc. Netw, 24 (3) (2002) 279–290.

[4] S. Wasserman, K. Faust, Social Network Analysis: Methods and Applications, Vol. 8, Cambridge university press, 1994.

[5] B. Yan, S. Gregory, Detecting communities in networks by merging cliques, in: ICIS, Vol. 1, IEEE, 2009, pp. 832–836.

[6] N. Ohsaka, T. Akiba, Y. Yoshida, K.-i. Kawarabayashi, Dynamic influence analysis in evolving networks, Proc. VLDB Endow. 9 (12) (2016) 1077–1088.

[7] W. Chen, Y. Wang, S. Yang, Efficient influence maximization in social networks, in: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2009, pp. 199–208.

[8] X. Chen, Y. Li, P. Wang, J. Lui, A general framework for estimating graphlet statistics via random walk, Proc. VLDB Endow. 10 (3) (2016) 253–264.

[9] R.-H. Li, J.X. Yu, X. Huang, H. Cheng, Random-walk domination in large graphs, in: ICDE, IEEE, 2014, pp. 736–747.

[10] B. Ribeiro, D. Towsley, Estimating and sampling graphs with multidimensional random walks, in: SIGCOMM, ACM, 2010, pp. 390–403.

[11] X. Xu, C.-H. Lee, et al., Challenging the limits: Sampling online social networks with cost constraints, in: INFOCOM, 2017.

[12] F. Chiericetti, A. Dasgupta, R. Kumar, S. Lattanzi, T. Sarlós, On sampling nodes in a network, in: WWW, International World Wide Web Conferences Steering Committee, 2016, pp. 471–481.

[13] A. Nazi, Z. Zhou, S. Thirumuruganathan, N. Zhang, G. Das, Walk, not wait: Faster sampling over online social networks, Proc. VLDB Endow. 8 (6) (2015) 678–689.

[14] R.-H. Li, J.X. Yu, L. Qin, R. Mao, T. Jin, On random walk based graph sampling, in: ICDE, IEEE, 2015, pp. 927–938.

[15] B. Ribeiro, P. Wang, F. Murai, D. Towsley, Sampling directed graphs with random walks, in: INFOCOM, IEEE, 2012, pp. 1692–1700.

[16] M. Gjoka, M. Kurant, C.T. Butts, A. Markopoulou, Walking in facebook: A case study of unbiased sampling of osns, in: INFOCOM, IEEE, 2010, pp. 1–9.

[17] C.-H. Lee, X. Xu, D.Y. Eun, Beyond random walk and metropolis-hastings samplers: why you should not backtrack for unbiased graph sampling, in: SIGMETRICS, Vol. 40, (1) ACM, 2012, pp. 319–330.

[18] Y. Li, Z. Wu, S. Lin, H. Xie, M. Lv, Y. Xu, J.C. Lui, Walking with perception: Efficient random walk sampling via common neighbor awareness, in: 2019 IEEE 35th International Conference on Data Engineering (ICDE), IEEE, 2019, pp. 962–973.

[19] J. Zhao, P. Wang, J.C. Lui, D. Towsley, X. Guan, Sampling online social networks by random walk with indirect jumps, Data Min. Knowl. Discov. 33 (1) (2019) 24–57.

[20] Z. Zhou, N. Zhang, Z. Gong, G. Das, Faster random walks by rewiring online social networks on-the-fly, ACM Trans. Database Syst. 40 (4) (2016) 26.

[21] D.G. Horvitz, D.J. Thompson, A generalization of sampling without replacement from a finite universe, J. Amer. Statist. Assoc. 47 (260) (1952) 663–685.

[22] M. Murthy, Ordered and unordered estimators in sampling without replacement, Sankhyā 18 (3/4) (1957) 379–390.

[23] R. Oliveira, et al., Mixing and hitting times for finite Markov chains, Electron. J. Probab. 17 (2012).

[24] Y. Peres, P. Sousi, Mixing times are hitting times of large sets, J. Theoret. Probab. 28 (2) (2015) 488–519.

[25] F. Chen, L. Lovász, I. Pak, Lifting Markov chains to speed up mixing, in: Proceedings of the Thirty-First Annual ACM Symposium on Theory of Computing, ACM, 1999, pp. 275–281.

[26] P. Diaconis, S. Holmes, R.M. Neal, Analysis of a nonreversible Markov chain sampler, Ann. Appl. Probab. (2000) 726–752.

[27] R.M. Neal, Improving asymptotic variance of MCMC estimators: Non-reversible chains are better, 2004, arXiv preprint math/0407281.

[28] J. Leskovec, C. Faloutsos, Sampling from large graphs, in: SIGKDD, ACM, 2006, pp. 631–636.

[29] L. Lovász, Random walks on graphs: A survey, Combinatorics, Paul erdos is eighty 2 (1) (1993) 1–46.

[30] M. Gjoka, M. Kurant, C.T. Butts, A. Markopoulou, Practical recommendations on crawling online social networks, IEEE J. Sel. Areas Commun. 29 (9) (2011) 1872–1892.

[31] Z. Zhou, N. Zhang, G. Das, Leveraging history for faster sampling of online social networks, VLDB 8 (10) (2015) 1034–1045.

[32] J. Konc, D. Janezic, An improved branch and bound algorithm for the maximum clique problem, Proteins 4 (2007) 5.

[33] L. Chang, Efficient maximum clique computation over large sparse graphs, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD'19, ACM, New York, NY, USA, 2019, pp. 529–538.

[34] C. Lu, J.X. Yu, H. Wei, Y. Zhang, Finding the maximum clique in massive graphs, Proc. VLDB Endow. 10 (11) (2017) 1538–1549.

[35] R.A. Rossi, D.F. Gleich, A.H. Gebremedhin, M.M.A. Patwary, Fast maximum clique algorithms for large graphs, in: Proceedings of the 23rd International Conference on World Wide Web, WWW'14 Companion, ACM, 2014, pp. 365–366.

[36] P. Wang, J. Lui, B. Ribeiro, D. Towsley, J. Zhao, X. Guan, Efficiently estimating motif statistics of large networks, ACM Trans. Knowl. Discov. Data 9 (2) (2014) 8.

[37] J. Yang, J. Leskovec, Defining and evaluating network communities based on ground-truth, Knowl. Inf. Syst. 42 (1) (2015) 181–213.

[38] J. Leskovec, L.A. Adamic, B.A. Huberman, The dynamics of viral marketing, ACM Trans. Web 1 (1) (2007) 5.

[39] J. Leskovec, D. Huttenlocher, J. Kleinberg, Signed networks in social media, in: SIGCHI, ACM, 2010, pp. 1361–1370.

[40] J. Leskovec, D. Huttenlocher, J. Kleinberg, Predicting positive and negative links in online social networks, in: WWW, ACM, 2010, pp. 641–650.

[41] X. Xu, C.-H. Lee, et al., A general framework of hybrid graph sampling for complex network analysis, in: IEEE INFOCOM 2014-IEEE Conference on Computer Communications, IEEE, 2014, pp. 2795–2803.

[42] J. Lu, D. Li, Sampling online social networks by random walk, in: Proceedings of the First ACM International Workshop on Hot Topics on Interdisciplinary Social Networks Research, ACM, 2012, pp. 33–40.

[43] X. Lei, X. Yang, H. Fujita, Random walk based method to identify essential proteins by integrating network topology and biological characteristics, Knowl.-Based Syst. 167 (2019) 53–67.

[44] J. Sun, J. Lang, H. Fujita, H. Li, Imbalanced enterprise credit evaluation with DTE-sbd: Decision tree ensemble based on SMOTE and bagging with differentiated sampling rates, Inform. Sci. 425 (2018) 76–91.

[45] M. Bressan, F. Chierichetti, R. Kumar, S. Leucci, A. Panconesi, Counting graphlets: Space vs time, in: Proceedings of the Tenth ACM International Conference on Web Search and Data Mining, ACM, 2017, pp. 557–566.

[46] M. Bressan, F. Chierichetti, R. Kumar, S. Leucci, A. Panconesi, Motif counting beyond five nodes, ACM Trans. Knowl. Discov. Data 12 (4) (2018) 48.