Data Mining and Knowledge Discovery (2019) 33:24–57
https://doi.org/10.1007/s10618-018-0587-5

Sampling online social networks by random walk with indirect jumps

Junzhou Zhao[1] · Pinghui Wang[2] · John C. S. Lui[1] · Don Towsley[3] · Xiaohong Guan[2]

Received: 20 August 2017 / Accepted: 16 August 2018 / Published online: 30 August 2018

Abstract
Random walk-based sampling methods are gaining popularity and importance in characterizing large networks. While powerful, they suffer from the slow mixing problem when the graph is loosely connected, which results in poor estimation accuracy. Random walk with jumps (RWwJ) can address the slow mixing problem but it is inapplicable if the graph does not support uniform vertex sampling (UNI). In this work, we develop methods that can efficiently sample a graph without the necessity of UNI but still enjoy the similar benefits as RWwJ. We observe that many graphs under study, called target graphs, do not exist in isolation. In many situations, a target graph is related to an auxiliary graph and a bipartite graph, and they together form a better connected *two-layered network structure*. This new viewpoint brings extra benefits to graph sampling: if directly sampling a target graph is difficult, we can sample it indirectly with the assistance of the other two graphs. We propose a series of new graph sampling techniques by exploiting such a two-layered network structure to estimate target graph characteristics. Experiments conducted on both synthetic and real-world networks demonstrate the effectiveness and usefulness of these new techniques.

Keywords Graph sampling · Random walk · Markov chain · Estimation theory

1 Introduction

Online social networks (OSNs) such as Facebook and Twitter have attracted much attention in recent years because of their ever-increasing popularity and importance in our daily lives. An OSN not only provides a platform for people to connect with their friends, but also offers an opportunity to study various user characteristics, which are valuable in many applications such as understanding human behaviors (Leskovec et al. 2010; Zhang et al. 2013; Backstrom and Kleinberg 2014) and inferring user

Responsible editor: Hanghang Tong.

Extended author information available on the last page of the article

preferences (Han et al. 2014; Li et al. 2016). Exactly measuring user characteristics requires the complete OSN data. For third parties who do not possess the data, they can only rely on public APIs to crawl the OSN. However, commercial OSNs are typically unwilling to grant third parties full permission to access the data due to user privacy and business secrecy. They often impose barriers to limit third parties' large-scale crawling (Mondal et al. 2012), e.g., by limiting the API requesting rate.[1] As a result, collecting the complete data of a large-scale OSN is practically impossible.

To address this challenge, sampling methods have been developed, i.e., a small fraction of OSN users are sampled and used to estimate the whole OSN user characteristics. In the literature, random walk based sampling methods have gained popularity (Massoulié et al. 2006; Avrachenkov et al. 2010; Ribeiro and Towsley 2010; Gjoka et al. 2011b; Ribeiro et al. 2012; Lee et al. 2012; Xu et al. 2014). In a typical random walk sampling, a walker is launched over a graph, which continuously moves from a node to one of its neighbors selected uniformly at random, to obtain a collection of node samples. These samples can be used to obtain unbiased estimates of nodal or topological properties of the graph. Because a random walk only explores neighborhood of a node during sampling, it is suitable for crawling and sampling large-scale OSNs.

1.1 Problems in random walk based sampling

While random walk sampling is powerful, if a graph is loosely connected, e.g., consists of communities, it will suffer from *slow mixing* (Sinclair and Jerrum 1989), i.e., requires a long "burn-in" period to reach steady state, which results in the need of a large number of samples in order to achieve desired estimation accuracy. Previous studies have found that mixing times in many real-world networks are larger than expected (Mohaisen et al. 2010).

To overcome the slow mixing problem, an effective approach is to incorporate uniform node sampling (UNI) into random walk sampling, and enable the walker to jump to other parts of the graph while walking, aka the *random walk with jumps* (RWwJ; Avrachenkov et al. 2010; Ribeiro et al. 2012; Xu et al. 2014). In UNI, a node is independently sampled uniformly at random from the graph, and in practice, if users in an OSN have unique numerical IDs, then UNI is conducted by generating random numbers in the ID space and including those valid IDs as UNI samples. RWwJ then leverages UNI to perform jumps on a graph. Specifically, at each step of RWwJ, the walker jumps with a probability determined by the node where it currently resides, to a node sampled by UNI. By incorporating UNI into random walk sampling, the walker can jump out of a community or disconnected component of a graph, and avoid being trapped, thereby reducing the mixing time (Avrachenkov et al. 2010).

The main problem of using RWwJ to sample an OSN is that, *some OSNs may not support UNI at all* because user IDs are not numerical, or *UNI is resource intensive* because the valid IDs are sparsely populated in the ID space. For example, in Pinterest,[2] a user's ID is an arbitrary length string, which hence makes UNI practically impossible. In MySpace and Flickr, although the user IDs are numerical, the fractions of valid user

[1] Twitter API rate limiting. https://dev.twitter.com/rest/public/rate-limiting.

[2] http://www.pinterest.com.

IDs are only about 10% and 1.3%, respectively (Ribeiro et al. 2012); in other words, one has to generate about 10 (or 77) random numbers (and verify them by querying OSN APIs) to obtain *one* valid user ID in MySpace (or Flickr). In some situations, the valid ID space could become extremely sparse.

Example 1 (Sampling Weibo users in a city) Suppose we want to measure user characteristics in Sina Weibo,[3] which is a popular OSN in China. Rather than measuring all the Weibo users, we are only interested in users who checked in[4] venues in a specified city. For example, users who shared check-in information at tourist spots, hotels, and restaurants of a city could be used to evaluate the city's internationality, economic index, etc. Suppose the users who checked in the city account for about 0.1% of all Weibo users. We also know that each Weibo user has a unique 10-digit numerical ID, and the fraction of valid IDs is about 10%.[5]

In the above example, when conducting UNI, we expect that a randomly generated number is a valid user ID, and the corresponding user checked in the city of interest. This happens with probability 10^{-4}, and as a result, we have to try 10^4 times on average to obtain one valid UNI sample. Without the efficiency of conducting UNI on a graph, we cannot perform jumps, and hence RWwJ is inapplicable. This raises the following problem we want to solve in this work:

> If we cannot perform jumps on a graph, can we conduct random walk sampling that still has the similar benefits as RWwJ?

1.2 Overview of our approach

In this work, we design a series of graph sampling techniques that can efficiently sample a network without the necessity of UNI, but still enjoy the similar benefits as RWwJ. The main idea behind our method is to leverage a "*two-layered network structure*" to perform "*indirect jumps*" on the graph under study, and indirect jumps can bring similar benefits as the direct jumps in RWwJ. We first use Example 1 to briefly explain what we mean by two-layered network structure, and then this discovery immediately motivates us to design an indirect sampling method, which enables us to perform indirect jumps on a graph.

In Example 1, directly applying UNI on the user network is inefficient because of the sparsity of user ID space, i.e., a randomly generated number is very likely to be an invalid user ID, or the user just lies outside of the city of interest. Since directly sampling users by UNI is difficult, we propose to sample users in an indirect manner. We notice that besides the user network, we are actually also provided with a space consisting of venues on a map, as illustrated in Fig. 1a. If we can sample venues in the city by UNI (or its variants), then we can sample users indirectly because venues

[3] http://weibo.com.

[4] Sina Weibo provides a check-in service (http://place.weibo.com) that allows users to share location information with their friends, e.g., the restaurants they took lunch, the hotels they lived during travel. The service is similar to the function in Foursquare and other location-based OSNs.

[5] A Weibo user ID is in the range [1,000,000,000, 6,200,000,000], as of May 2017. About 10% of the IDs in this range represent valid users.

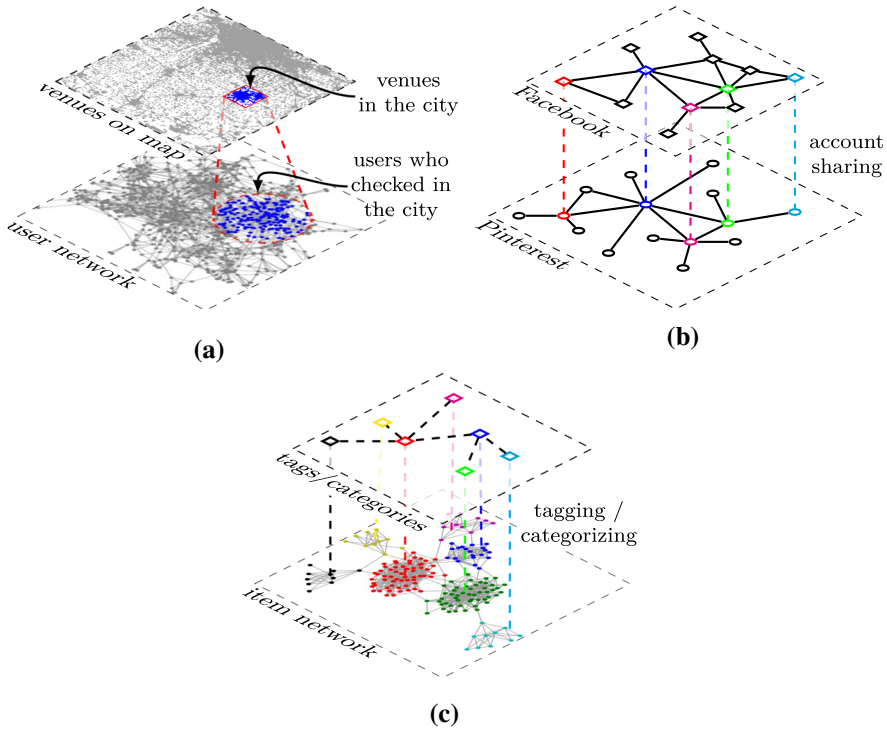

Fig. 1 Examples of two-layered network structures. **a** User network and venues on a map, **b** accounts sharing between two OSNs, **c** item network and tag/category network

and users are related by their check-in relationships. The check-ins tell us which user checked in which place, and for a given venue, we can query the users who checked in this venue, and hence easily *obtain a user sample from a venue sample*. Sampling venues in an area is indeed possible by leveraging the APIs provided by many location-based OSNs (LBSNs). Many LBSNs provide APIs for querying venues within an area specified by a rectangle region with southwest and northeast corners latitude-longitude coordinates given.[6] This function can be used to design efficient sampling methods for sampling venues in an area on a map (Li et al. 2012, 2014; Wang et al. 2014a). For example, we can efficiently sample a venue in the city specified by a rectangle region, and the probability of obtaining this venue sample is calculable. Note that a user sample obtained from a venue sample is no longer uniformly distributed. Because if a user checked in many venues in the city, the user is likely to be oversampled. But such bias can be easily removed by a reweighting strategy, which we will elaborate in Sect. 4.

An important lesson learned from solving the problem in Example 1 is that, the *two-layered network structure*, consisting of the user network layer and the venues layer, can help us to obtain samples of one layer when sampling another layer is easy. Hence, this enables us to conduct "indirect jumps" on the user net-

[6] Weibo search API. http://open.weibo.com/wiki/2/location/pois/search/by_area.

work with the help of venue sampling. We further find that the two-layered network structure is not unique to the problem in Example 1, but is pervasive in a wide range of graph sampling problems, and more examples will be presented in Sect. 3. Hence, it is necessary to develop some unified graph sampling techniques that can leverage the two-layered network structure to address these graph sampling problems.

In general, there are *three graphs* related to the two-layered network structure: (1) a *target graph*, whose characteristics are of interest to us and need to be estimated, e.g., the user sub-network in Example 1; (2) an *auxiliary graph*, which is easier to be sampled than the target graph, e.g., the venues can be viewed as nodes in an auxiliary graph, and Example 1 is a special case where the auxiliary graph has an empty edge set; and (3) a *bipartite graph* that connects nodes in the target and auxiliary graphs. When directly sampling the target graph is difficult, we can turn to sample the auxiliary graph, and the bipartite graph bridges the two sample spaces and allows us to sample the target graph in an indirect manner. This thus enables us to perform indirect jumps on the target graph, and allows us to develop random walk sampling methods with indirect jumps that have the similar benefits as RWwJ.

1.3 Our contributions

This work extends our preliminary research in Zhao et al. (2015), and mainly makes three contributions:

- *Two-layered network structure.* We discover the usefulness of a "two-layered network structure" that exists in many real-world applications. This structure can be exploited to efficiently sample a graph in an indirect manner if directly sampling the graph is difficult. We provide motivating examples to show evidence of the applicability of our discovery.
- *Random walk sampling with indirect jumps.* We extend the classical RWwJ sampling method that relays on UNI sampling on a graph to random walk with indirect jumps methods that remove the necessity of conducting UNI on target graph. We design three new sampling techniques by leveraging the two-layered network structure. These new techniques enable us to conduct random walk sampling that has the similar benefits as RWwJ.
- *Experiments on various networks.* We conduct extensive experiments on both synthetic and real-world networks to validate our proposed techniques. The experimental results demonstrate the effectiveness of our designed sampling techniques.

1.4 Outline

The reminder of this paper will proceed as follows. In Sect. 2, we provide some preliminaries about graph sampling. In Sect. 3, we formally define the two-layered network structure along with more examples. In Sect. 4, we elaborate three new sampling methods. In Sect. 5, we conduct experiments to validate our methods. Section 6 reviews some related literature, and Sect. 8 concludes.

2 Preliminaries

In this section, we provide some preliminaries about the graph sampling problem, and review a random walk based sampling method named random walk with jumps (RWwJ).

2.1 Graph sampling

An OSN can be modeled as an undirected graph[7] $G = (U, E)$, where U is a finite set of nodes representing users, and $E \subseteq U \times U$ is a set of edges representing relations among users. We assume that the graph G has no self-loops and no multiple edges connecting two nodes. Also, the graph size $|U| = n$ may be not known in advance.

Let $f : U \mapsto \mathbb{R}$ be any desired *characteristic function* that maps a node in the graph to a real number. The goal of measuring the characteristic of graph G is to estimate

$$\theta \triangleq \frac{1}{n} \sum_{u \in U} f(u),$$

which is the aggregated nodal characteristic of the graph. For example, in an OSN, we let $f(u) = 1$ if user u is female, and otherwise $f(u) = 0$, then θ represents the fraction of female users in the OSN.

The goal of graph sampling is to design an algorithm for collecting node samples S from graph G, constrained by a budget $|S| \leq B \ll n$, and for providing unbiased estimate of θ with low statistical error.

2.2 Random walk with jumps

Random walk with jumps (RWwJ; Avrachenkov et al. 2010) is a popular graph sampling method that can address the slow mixing issue of a simple random walk when the graph has community structures. RWwJ generally works as follows: A walker starts from a node in the graph, and at each step, it moves to a neighbor selected uniformly at random, or jumps to a node uniformly sampled from the graph, and the probability of jumping is determined by the node where the walker currently resides; this process continues until enough samples are collected.

An easier way to think about RWwJ is that, we modify the structure of the original graph by connecting every node in the graph to a virtual *jumper node*, with edge weight $\alpha \geq 0$; then a simple random walk on this modified graph is equivalent to RWwJ. Figure 2 illustrates RWwJ on a loosely connected graph. Comparing the modified graph with the original graph, we can find that the modified graph always has larger *graph conductance* than the original graph, and because larger graph conductance usually implies faster mixing of a random walk (Sinclair and Jerrum 1989), hence,

[7] For Facebook, the friendship network is an undirected graph; for Twitter, because the followees and followers of a user are known once the user is collected, hence we can build an undirected graph of the Twitter follower network on-the-fly.

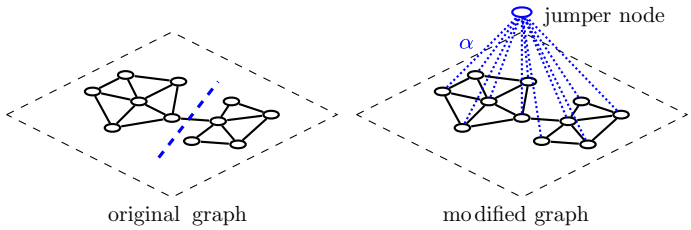

Fig. 2 RWwJ is viewed as a simple random walk on the modified graph

RWwJ has the advantage of faster mixing than a simple random walk on poorly connected graphs (Avrachenkov et al. 2010).

In RWwJ, the probability transition matrix of the underlying Markov chain is given by

$$P_{ij}^{\mathrm{RWwJ}} = \begin{cases} \frac{\alpha/n+1}{d_i+\alpha}, & (i,j) \in E, \\ \frac{\alpha/n}{d_i+\alpha}, & (i,j) \notin E, \end{cases}$$

where d_i denotes the degree of node i in the original graph. That is, if $(i,j) \in E$, the walker starting from i could walk to j (in one step) through the edge (i,j) with probability $\frac{1}{d_i+\alpha}$; or jump to j through UNI with probability $\frac{\alpha}{d_i+\alpha} \cdot \frac{1}{n} = \frac{\alpha/n}{d_i+\alpha}$; thus the transition probability on edge (i,j) is $\frac{\alpha/n+1}{d_i+\alpha}$. If $(i,j) \notin E$, the walk starting from i can only walk to j (in one step) by jumping with probability $\frac{\alpha/n}{d_i+\alpha}$.

When RWwJ reaches the steady state, a node $u \in U$ is sampled with probability proportional to $d_u + \alpha$. If we let S denote the samples collected by RWwJ, an asymptotically unbiased estimator of θ is given by

$$\hat{\theta}^{\mathrm{RWwJ}} = \frac{1}{Z^{\mathrm{RWwJ}}} \sum_{s \in S} \frac{f(s)}{d_s + \alpha}, \tag{1}$$

where $Z^{\mathrm{RWwJ}} \triangleq \sum_{s \in S} 1/(d_s + \alpha)$. We can understand the unbiasedness of Estimator (1) by leveraging the ratio form of the *Law of Large Numbers* of Markov chains (Meyn and Tweedie 2009, pp. 427–428).

Lemma 1 (Law of large numbers) *Let S be a sample path obtained by a Markov chain defined on state space U with stationary distribution π. For any function $f, g: U \mapsto \mathbb{R}$, and let $F_S(f) \triangleq \sum_{s \in S} f(s)$, $\mathbb{E}_\pi[f] \triangleq \sum_{u \in U} \pi_u f(u)$. It holds that*

$$\lim_{|S| \to \infty} \frac{1}{|S|} F_S(f) = \mathbb{E}_\pi[f] \quad a.s., \tag{2}$$

$$\lim_{|S| \to \infty} \frac{F_S(f)}{F_S(g)} = \frac{\mathbb{E}_\pi[f]}{\mathbb{E}_\pi[g]} \quad a.s. \tag{3}$$

Here, "a.s." denotes "almost sure" convergence, i.e., the event of interest happens with probability one.

Therefore, in Estimator (1), replacing $f(s)/(d_s + \alpha)$ by $f(s)$, and $1/(d_s + \alpha)$ by $g(s)$, we obtain that $\hat{\theta}^{\text{RWwJ}}$ converges to $\mathbb{E}_\pi[f]/\mathbb{E}_\pi[g] = \theta$, almost surely.

Although RWwJ can address the slow mixing problem, it requires UNI to perform jumps on a graph. If the OSN does not support UNI, or UNI is inefficient, RWwJ becomes inapplicable. In this work, we introduce a two-layered network structure that exists in many real-world applications, and we will show that such a structure can be leveraged to design random walk sampling methods having the similar benefits as RWwJ even though we cannot conduct UNI on the graph.

3 Two-layered network structure

In this section, we first formally describe the two-layered network structure we discovered in Example 1. Then we provide more examples to demonstrate the pervasiveness of such a structure.

3.1 Definition

We use three undirected graphs to describe a two-layered network structure: $G(U, E)$, $G'(V, E')$, and $G_b(U, V, E_b)$, where U, V are two sets of nodes, and $E \subseteq U \times U$, $E' \subseteq V \times V$, $E_b \subseteq U \times V$ are three sets of edges. More specifically,

- $G(U, E)$ is the *target graph*, whose characteristic θ is of interest to us and needs to be measured. For example, the user social network in Example 1 can be treated as the target graph.
- $G'(V, E')$ is an *auxiliary graph*, which can be more efficiently sampled than the target graph. In Example 1, we can construct an auxiliary graph where the nodes represent the venues in the city, and the edge set is left empty (i.e., $E' = \emptyset$).
- $G_b(U, V, E_b)$ is a *bipartite graph* that connects nodes in the target and auxiliary graphs. In Example 1, the bipartite graph is formed by users, venues and their check-in relationships.

An example of such a two-layered network structure is illustrated in Fig. 3. The target graph consists of two disconnected components. If we can find a well connected auxiliary graph and a proper bipartite graph, then the two graphs bridge the two disconnected components and thus make target graph better connected than target graph itself. Hence, it is possible to sample the target graph efficiently with the help of the other two graphs. With this intuition in mind, we will see in next section that we indeed can design efficient sampling methods by leveraging this two-layered network structure.

3.2 More examples

The two-layered network structure is not unique to Example 1, but exists in a wide range of real-world applications. In what follows, we provide more examples.

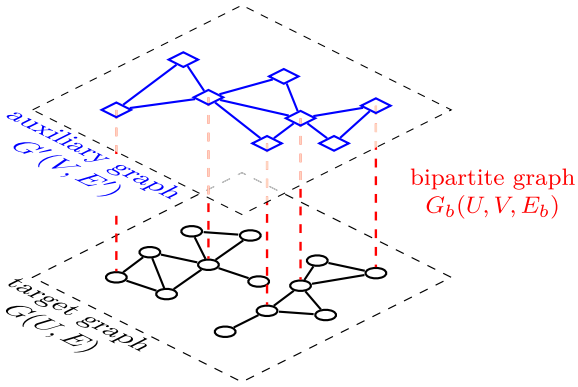

Fig. 3 Illustration of the two-layered network structure

Example 2 (Accounts sharing between two OSNs) Many OSNs now support using an existing OSN's accounts to login another OSN. For example, Facebook users can login Pinterest using their Facebook accounts. This naturally forms a two-layered network structure consisting of Facebook and Pinterest. Suppose we want to measure Pinterest, then we can let target graph represent Pinterest, auxiliary graph represent Facebook, and bipartite graph represent their account sharing relations.

Figure 1b illustrates Example 2. Note that Pinterest does not support UNI, hence RWwJ is inapplicable. Instead, using the techniques developed in this work, we will be able to leverage Facebook to sample Pinterest.

Example 3 (Amazon item network and categories) Items in Amazon are related with each other to form an item network. Each item also belongs to one or more categories. Meanwhile, Amazon provides a complete category list to facilitate customers to quickly navigate to the items they are looking for. This forms a two-layered network structure consisting of items and categories. Suppose we want to measure the item network, then we can let target graph represent the item network, auxiliary graph represent the category list, and bipartite graph represent the affiliation relations between items and categories.

Figure 1c illustrates Example 3. Note that categories could also be tags and they may also form a tag network. Items are very likely to form clusters, and hence easily trap a random walker. If we can leverage the category information, and help a random walker to jump out of clusters, we can sample the item network in a more efficient way.

4 Sampling design

In this section, we leverage the two-layered network structure and design three new sampling techniques to sample and characterize the target graph.

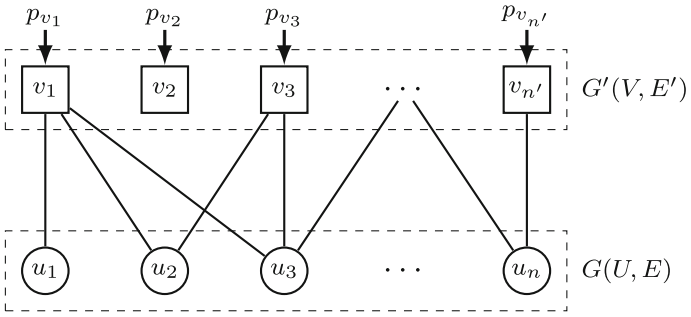

Fig. 4 Illustration of VS^{A}. Edges in G and G' are omitted

4.1 Indirectly sampling target graph by vertex sampling on auxiliary graph (VS^{A})

The first method assumes that vertex sampling is easier to conduct on the auxiliary graph than on the target graph, as is the case in Example 1, and each node in the target graph is connected to at least one node in the auxiliary graph. We present a sampling method VS^{A} (and its two implementations VS^{A}-I and VS^{A}-II) to indirectly sample the target graph under this setting. The basic idea of VS^{A} is illustrated in Fig. 4.

VS^{A}-I. Assume that a node $v \in V$ is sampled with probability $p_v \propto a_v > 0$ in auxiliary graph G'. Here we do not need to know the exact value of p_v, and knowing p_v is proportional to some value a_v is enough to design the sampling. For example, if auxiliary graph G' supports UNI, then $p_v = 1/n$; however, graph size n is usually unknown, and we will see that knowing $p_v \propto a_v \equiv 1, \forall v \in V$ is enough for the sampling design. The simplest way to implement VS^{A} is as follows: We first sample a node $v \in V$ in G', and then sample a neighbor of v in G_b uniformly at random, denoted by u. Obviously, $u \in U$, and we collect u as a sample. We refer to this simple sampling method as VS^{A}-I, and will show that samples collected by VS^{A}-I can indeed yield unbiased estimate of θ. The detailed design of VS^{A}-I is described as follows.

Sampling design. VS^{A}-I repeats the following two steps until sample collection S reaches budget B.

- Sample a node v from auxiliary graph G';
- If v has neighbors in bipartite graph G_b, sample a neighbor u uniformly at random, and put u into samples S.

Estimator. In VS^{A}-I, we can see that a node $u \in U$ is sampled with probability

$$p_u \propto b_u \triangleq \sum_{v \in V_u} \frac{a_v}{d_v^{(b)}}, \tag{4}$$

where $V_u \subseteq V$ is the set of neighbors of u in G_b, and $d_v^{(b)}$ is the degree of v in G_b. Because we have assumed that each node in the target graph is connected to at least one node in the auxiliary graph, so $V_u \neq \emptyset$ and $b_u > 0$. Then, we propose to use the following estimator to estimate θ:

$$\hat{\theta}^{\text{VS}^{\text{A}}\text{-I}} = \frac{1}{Z^{\text{VS}^{\text{A}}\text{-I}}} \sum_{u \in S} \frac{f(u)}{b_u}, \tag{5}$$

where $Z^{\text{VS}^{\text{A}}\text{-I}} \triangleq \sum_{u \in S} 1/b_u$. The following theorem guarantees its unbiasedness.

Theorem 1 *Assume each node in G is connected to at least one node in G'. Then using the sampling design of VS^{A}-I, Estimator (5) provides an asymptotically unbiased estimate of θ.*

Proof VS$^{\text{A}}$-I can be viewed as sampling U with replacement according to distribution $\{p_u\}_{u \in U}$. This can be further viewed as generating samples according to a Markov chain which has a probability transition matrix with all rows the same vector $[p_u]_{u \in U}$, and $\pi_u = p_u, \forall u \in U$. This allows us to leverage Lemma 1, and obtain that

$$\begin{aligned} \lim_{B \to \infty} \hat{\theta}^{\text{VS}^{\text{A}}\text{-I}} &= \frac{\mathbb{E}[f(u)/b_u]}{\mathbb{E}[1/b_u]} = \frac{\sum_{u \in U} p_u f(u)/b_u}{\sum_{u \in U} p_u/b_u} \\ &= \frac{1}{n} \sum_{u \in U} f(u) = \theta \qquad a.s. \end{aligned}$$

This thus completes the proof. □

VS$^{\text{A}}$-I has one drawback. To correct the bias of each sample $u \in S$, we require b_u, which further requires a_v for each neighbor of u in G_b by Eq. (4). This is not an issue if we are conducting UNI on the auxiliary graph, as we have known $a_v, \forall v \in V$ before conducting UNI (i.e., $a_v = 1, \forall v \in V$). But in some applications where more complex vertex sampling methods are applied on auxiliary graph, a_v is not known in a prior—we know a_v only if v is sampled, otherwise a_v is not known in advance. This is actually the case we met in Example 1: we know the probability of obtaining a venue sample only if the venue is sampled.[8] To address this problem, we propose another sampling method VS$^{\text{A}}$-II.

VS$^{\text{A}}$-II. When a node $v \in V$ is sampled in auxiliary graph, we collect all of its neighbors in the bipartite graph as samples; we repeat this process until enough samples are collected. We use these samples to estimate θ. The detailed design of VS$^{\text{A}}$-II is described as follows.

Sampling design. VS$^{\text{A}}$-II repeats the following steps to obtain two sample collections S and S' from G and G' respectively. Samples in S are used to estimate θ.

- Sample a node v from auxiliary graph G';
- If v has neighbors in bipartite graph G_b, put v into samples S', and put all the neighbors of v in G_b into samples S.

Estimator design for VS$^{\text{A}}$-II. We propose to estimate θ using the following estimator:

$$\hat{\theta}^{\text{VS}^{\text{A}}\text{-II}} = \frac{1}{Z^{\text{VS}^{\text{A}}\text{-II}}} \sum_{v \in S'} \frac{1}{a_v} \sum_{u \in U_v} \frac{f(u)}{d_u^{(b)}}, \tag{6}$$

[8] This should become clear when we use the random region zoom-in (RRZI; Wang et al. 2014a) method to conduct venue sampling in Sect. 5.

where $U_v \subseteq U$ is the set of neighbors of v in G_b, $d_u^{(b)}$ is the degree of node u in G_b, and $Z^{\text{VS}^\text{A}\text{-II}} \triangleq \sum_{v\in S'} 1/a_v \sum_{u\in U_v} 1/d_u^{(b)}$. Because we have assumed that each node in the target graph is connected to at least one node in the auxiliary graph, so $d_u^{(b)} > 0$. The following theorem guarantees its unbiasedness.

Theorem 2 *Assume each node in G is connected to at least one node in G'. Then using the sampling design of VS^A-II, Estimator (6) provides an asymptotically unbiased estimate of θ.*

Proof Using the similar idea as we proved Theorem 1, we have

$$\mathbb{E}\left[\frac{1}{a_v}\sum_{u\in U_v}\frac{f(u)}{d_u^{(b)}}\right] = \sum_{v\in V}\frac{p_v}{a_v}\sum_{u\in U_v}\frac{f(u)}{d_u^{(b)}} = c\sum_{v\in V}\sum_{u\in U_v}\frac{f(u)}{d_u^{(b)}}$$
$$= c\sum_{u\in U} d_u^{(b)}\frac{f(u)}{d_u^{(b)}} = c\sum_{u\in U} f(u) = cn\theta$$

where $c \triangleq p_v/a_v$ is a constant. The third equation holds because each inside item is added exactly $d_u^{(b)}$ times before we merge the two sums into one sum. Similarly,

$$\mathbb{E}\left[\frac{1}{a_v}\sum_{u\in U_v}\frac{1}{d_u^{(b)}}\right] = \sum_{v\in V}\frac{p_v}{a_v}\sum_{u\in U_v}\frac{1}{d_u^{(b)}} = c\sum_{v\in V}\sum_{u\in U_v}\frac{1}{d_u^{(b)}} = cn.$$

By Lemma (1), we thus obtain

$$\lim_{B\to\infty}\hat{\theta}^{\text{VS}^\text{A}\text{-II}} = \frac{\mathbb{E}\left[1/a_v\sum_{u\in U_v} f(u)/d_u^{(b)}\right]}{\mathbb{E}\left[1/a_v\sum_{u\in U_v} 1/d_u^{(b)}\right]} = \theta \quad a.s. \qquad \square$$

Remarks

- *Apply condition* It is important to know that VS$^\text{A}$ (either VS$^\text{A}$-I or VS$^\text{A}$-II) can provide an unbiased estimate of target graph characteristic under the condition that *every node in the target graph is connected to nodes in the auxiliary graph.* If a node u is not connected to any node in G', u cannot be indirectly sampled by VS$^\text{A}$. This will result in biased estimates, and it is difficult to correct the bias. In Example 1, since we are only interested in users who share their check-ins in Weibo, therefore Example 1 satisfies this condition.
- *Sampling cost* The main advantage of VS$^\text{A}$ is that it leverages auxiliary graph to sample target graph efficiently when directly sampling target graph is inefficient. We give an example often met by statisticians to help readers understand the reason intuitively. Suppose we want to collect patient samples about some rare disease in a population. One straightforward method is that we conduct surveys and collect samples on the street, i.e., we randomly pick a person on the street, and ask whether the person is a patient of the disease; if yes, we obtain a sample. Since it is a rare

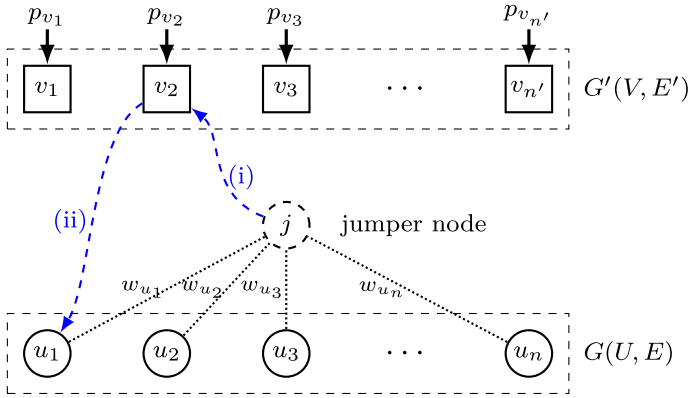

Fig. 5 Illustration of RW^TVS^A and indirect jump. Each node u in G is virtually connected to a jumper node j with weight w_u. An indirect jump is performed by: (**i**) randomly sampling a node v in G', and (**ii**) randomly choosing a neighbor of v in G_b as the target node to jump to

disease, we will fail to collect a sample very frequently. A more efficient method is that we first randomly pick a hospital, and then collect patient samples recorded by that hospital. Let the target graph represent the population, and auxiliary graph represent hospitals, then VS$^{\text{A}}$ actually uses the idea of the second better approach in essence, and thus significantly reduces sampling cost.

4.2 Random walk on target graph incorporating with vertex sampling on auxiliary graph (RW$^{\text{T}}$VS$^{\text{A}}$)

In some situations, $d_u^{(b)} = 0$ for some $u \in U$, such as the case in Example 2, where some Pinterest users may not have Facebook accounts at all, and these users cannot be sampled by VS$^{\text{A}}$ (and as a result, VS$^{\text{A}}$ cannot provide unbiased estimates of Pinterest user characteristics). To address this issue, we propose a second sampling method RW$^{\text{T}}$VS$^{\text{A}}$, which combines random walk sampling on the target graph with vertex sampling on the auxiliary graph.

The basic idea of RW$^{\text{T}}$VS$^{\text{A}}$ is that, we launch a random walk on the target graph, and at each step allow the walker to jump with a probability dependent on the node where the walker currently resides. This is similar to RWwJ on the target graph G, but with the major difference that in RW$^{\text{T}}$VS$^{\text{A}}$ the walker jumps to a node in G by jumping first to a node in G', and then randomly selecting one of its neighbors in G_b (similar to VS$^{\text{A}}$-I). We refer to this as an *indirect jump*, and show in experiments that indirect jumps in RW$^{\text{T}}$VS$^{\text{A}}$ bring similar benefits as the direct jumps in RWwJ. An additional advantage of using random walk on the target graph is that it better characterizes highly connected nodes than uniform node sampling as random walks are biased towards high degree nodes in G. We depict RW$^{\text{T}}$VS$^{\text{A}}$ in Fig. 5, where each node in G is virtually connected to a virtual jumper node to conduct indirect jumps, through doing vertex sampling over auxiliary graph G'.

Similar to VS$^{\text{A}}$, we assume that a node v in G' can be sampled with probability $p_v \propto a_v > 0$. Similar to the discussion of RWwJ in Sect. 2, in RW$^{\text{T}}$VS$^{\text{A}}$, we virtually

connect each node $u \in U$ to a jumper node j with edge (u, j), and assign a weight w_u for edge (u, j). The main challenge in designing $\mathrm{RW^TVS^A}$ is to determine the edge weights $\{w_u\}_{u\in U}$. With proper edge weights assignment, we can guarantee the *time reversibility*[9] of random walks, which can facilitate us to determine the stationary probability of a random walk visiting a node on target graph, and also simplify the estimator design. The following theorem states our main result on edge weights assignment.

Theorem 3 *If we assign the edge weights* $\{w_u\}_{u\in U}$ *by*

$$w_u = \alpha \sum_{v\in V_u} \frac{a_v}{d_v^{(b)}}, \quad u \in U \tag{7}$$

for any constant $\alpha \geq 0$*, then the random walk in* RW^TVS^A *is time reversible, and the stationary probability of the random walk visiting node* $u \in U$ *satisfies* $\pi_u \propto d_u + w_u$*, where* d_u *is the degree of* u *in target graph* G*.*

Proof If the random walk is time reversible, the stationary probabilities of visiting u and j are

$$\pi_u = \frac{d_u + w_u}{2|E| + 2\sum_u w_u} \quad \text{and} \quad \pi_j = \frac{\sum_u w_u}{2|E| + 2\sum_u w_u}.$$

Because for any $w_u \geq 0$, it always holds that

$$\pi_u p_{uu'} = \pi_{u'} p_{u'u} = \frac{1}{2|E| + 2\sum_u w_u}, \quad \forall (u, u') \in E.$$

That is, the random walk is always time reversible along the transitions in E. We only need to prove that with the w_u given by Theorem 3, the random walk is also time reversible along the transitions (u, j) and (j, u), i.e., $\pi_u p_{uj} = \pi_j p_{ju}$.

The walker residing at node u moves to j to perform an indirect jump with probability $p_{uj} = w_u/(d_u + w_u)$. Because an indirect jump is performed by first sampling a node v in G', and then choosing a neighbor u of v uniformly at random. Thus, the walker jumps from j to u with probability

$$p_{ju} = c \sum_{v\in V_u} \frac{a_v}{d_v^{(b)}} \triangleq c b_u \tag{8}$$

where c is a constant. When $w_u = \alpha b_u$, so $\sum_u w_u = \alpha/c$, it indeed holds that

$$\pi_u p_{uj} = \pi_j p_{ju} = \frac{w_u}{2|E| + 2\alpha/c}, \ \forall u \in U.$$

[9] A Markov chain is said to be time reversible with respect to stationary distribution π if it satisfies condition $\pi_i p_{ij} = \pi_j p_{ji}, \forall i, j$.

This demonstrates that when $w_u = \alpha b_u$, the random walk is time reversible, and the stationary probability of visiting u satisfies $\pi_u \propto d_u + w_u$. □

Note that if $d_u^{(b)} = 0$, then $p_{uj} = p_{ju} = 0$, i.e., the walker does not jump from/to u; the worker just moves from/to u to/from a neighbor of u. Hence, u can still be sampled by the random walk. α controls the probability of conducting a jump on a node. If $\alpha = 0$, RW$^{\mathrm{T}}$VS$^{\mathrm{A}}$ does not perform jumps, and it actually becomes a simple random walk on the target graph; if $\alpha \to \infty$, RW$^{\mathrm{T}}$VS$^{\mathrm{A}}$ is equivalent to VS$^{\mathrm{A}}$-I. Thus, RW$^{\mathrm{T}}$VS$^{\mathrm{A}}$ behaves similarly as RWwJ.

Sampling design. Suppose the random walk starts at node $x_1 \in U$, and at step i the random walk is at node x_i. At step i, we calculate the probability of jumping w_{x_i} by Eq. (7), then the walker jumps with probability $w_{x_i}/(d_{x_i} + w_{x_i})$; otherwise, the walker moves to a neighbor u of x_i chosen uniformly at random and $x_{i+1} = u$. An indirect jump is performed as follows:

- repeatedly sample a node v in auxiliary graph until v has neighbors in G_b;
- sample a neighbor u of v in G_b uniformly at random, and $x_{i+1} = u$.

Estimator. Using the collected samples, denoted by $S = (x_i, \ldots, x_B)$, we propose to estimate θ by

$$\hat{\theta}^{\mathrm{RW^TVS^A}} = \frac{1}{Z^{\mathrm{RW^TVS^A}}} \sum_{u \in S} \frac{f(u)}{d_u + w_u}, \tag{9}$$

where $Z^{\mathrm{RW^TVS^A}} \triangleq \sum_{u \in S} 1/(d_u + w_u)$.

Theorem 4 *If the target graph is connected, or could become connected by adding nodes in auxiliary graph and edges in bipartite graph, then using samples collected by RW^{T}VS^A, Estimator* (9) *provides an asymptotically unbiased estimate of* θ.

Proof Since $\pi_u \propto d_u + w_u$, then

$$\mathbb{E}_\pi \left[\frac{f(u)}{d_u + w_u} \right] = \sum_{u \in U} \pi_u \frac{f(u)}{d_u + w_u} = cn\theta.$$

Similarly,

$$\mathbb{E}_\pi \left[\frac{1}{d_u + w_u} \right] = \sum_{u \in U} \pi_u \frac{1}{d_u + w_u} = cn.$$

By Lemma (1), we obtain

$$\lim_{B \to \infty} \hat{\theta}^{\mathrm{RW^TVS^A}} = \frac{\mathbb{E}_\pi \left[f(u)/(d_u + w_u) \right]}{\mathbb{E}_\pi \left[1/(d_u + w_u) \right]} = \theta \quad a.s.$$

This completes the proof. □

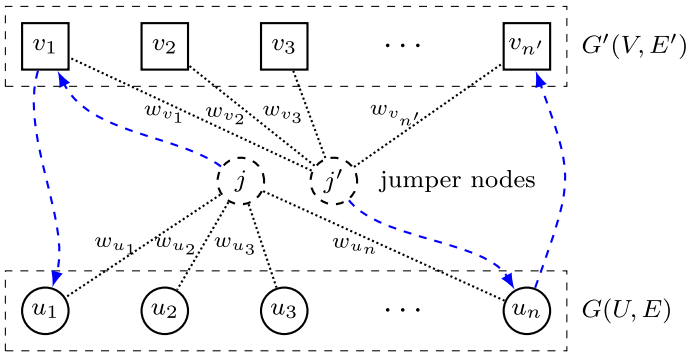

Fig. 6 Illustration of $\mathrm{RW^TRW^A}$ and indirect jumps

Remark Note that $\mathrm{RW^TVS^A}$ requires vertex sampling (e.g., UNI) on the auxiliary graph G'. If vertex sampling is also not allowed on G', $\mathrm{RW^TVS^A}$ is inapplicable. However, one can replace the vertex sampling on G' by a random walk on G'. Unfortunately, this naive approach can perform very poorly when the auxiliary graph G' is not well connected, because a poorly connected graph can easily trap a simple random walk in a community. In what follows, we design a third method to address this challenge.

4.3 Random walk on target graph incorporating with random walk on auxiliary graph ($\mathrm{RW^TRW^A}$)

When both the target and auxiliary graphs do not support vertex sampling, neither $\mathrm{VS^A}$ nor $\mathrm{RW^TVS^A}$ is applicable. Therefore, we design the $\mathrm{RW^TRW^A}$ method to address this challenge. $\mathrm{RW^TRW^A}$ consists of two parallel random walks on G and G' respectively. The two random walks cooperate with each other, and can be viewed as two RWwJs, as illustrated in Fig. 6. Unlike $\mathrm{RW^TVS^A}$ where only nodes in G are virtually connected to a jumper node, in $\mathrm{RW^TRW^A}$, nodes in both G and G' are virtually connected to two jumper nodes j and j' with weights $\{w_u\}_{u\in U}$ and $\{w_v\}_{v\in V}$ to perform indirect jumps on G and G' respectively.

The basic idea behind $\mathrm{RW^TRW^A}$ is as follows. Suppose the two random walks are RW on G and RW' on G', and at step i, they reside at $x_i \in U$ and $y_i \in V$, respectively. If one random walk needs to jump at step i, say RW on G, then it jumps to a uniformly at random chosen neighbor of y_i in the bipartite graph, which is assigned to x_{i+1}. Similar jumping procedure also applies to RW' on G'. Hence, they are analogous to two RWwJs, and both can avoid being trapped on G and G'.

Similar to $\mathrm{RW^TVS^A}$, the main challenge in designing $\mathrm{RW^TRW^A}$ is to determine edge weights $\{w_u\}_{u\in U}$ and $\{w_v\}_{v\in V}$, which control the probability of jumping of the two random walks. Obviously, the stationary distributions $\{\pi_u\}_{u\in U}$ and $\{\pi_v\}_{v\in V}$ of the two random walks are also related to these weights. Here we leverage our previous analysis of $\mathrm{RW^TVS^A}$, and derive that, when parameters w_u and w_v satisfy the following conditions

$$w_u = \alpha \sum_{v \in V_u} \frac{\pi_v}{d_v^{(b)}}, u \in U, \quad w_v = \beta \sum_{u \in U_v} \frac{\pi_u}{d_u^{(b)}}, v \in V, \tag{10}$$

for any $\alpha, \beta > 0$, the stationary distributions of the two random walks on G and G' (discarding states j and j') are

$$\pi_u = \frac{d_u + w_u}{2|E| + \alpha}, u \in U, \quad \pi_v = \frac{d_v + w_v}{2|E'| + \beta}, v \in V. \tag{11}$$

The matrix forms of Eqs. (10) and (11) yield

$$w_U = \alpha A D_V^{-1} \pi_V, \qquad w_V = \beta A^T D_U^{-1} \pi_U, \tag{12}$$

$$\pi_U = \frac{d_U + w_U}{2|E| + \alpha}, \qquad \pi_V = \frac{d_V + w_V}{2|E'| + \beta}, \tag{13}$$

where $A_{n \times n'}$ is the adjacency matrix of G_b, $w_U = [w_u]_{u \in U}^T$, $w_V = [w_v]_{v \in V}^T$, $\pi_U = [\pi_u]_{u \in U}^T$, $\pi_V = [\pi_v]_{v \in V}^T$, $d_U = [d_u]_{u \in U}^T$ and $d_V = [d_v]_{v \in V}^T$ are vectors, $D_U = diag(d_{u_1}^{(b)}, \ldots, d_{u_n}^{(b)})$ and $D_V = diag(d_{v_1}^{(b)}, \ldots, d_{v_{n'}}^{(b)})$ are diagonal matrices.

Equations (12) and (13) uniquely determine w_U and w_V, i.e.,

$$w_U^* = c \left(I - cc' A D_V^{-1} A^T D_U^{-1} \right)^{-1} A D_V^{-1} \left(d_V + c' A^T D_U^{-1} d_U \right)$$

$$w_V^* = c' \left(I - cc' A^T D_U^{-1} A D_V^{-1} \right)^{-1} A^T D_U^{-1} \left(d_U + c A D_V^{-1} d_V \right)$$

where $c = \alpha / \left(2|E'| + \beta\right)$ and $c' = \beta / (2|E| + \alpha)$ are constants.

The above results illustrate that, when α and β are given, w_U and w_V are uniquely determined. However, one needs complete knowledge of G, G' and G_b to determine their values. In graph sampling, we are interested in methods without having to know the complete graph structure in advance. In what follows, we design $\mathrm{RW^TRW^A}$ in a way that only makes use of *local knowledge* of these graphs.

In general, if $w_U \neq w_U^*$ (or $w_V \neq w_V^*$), the random walks on the two modified graphs are no longer timer reversible, and Eqs. (12) and (13) do not hold. There is another way to understand why they do not hold, and this understanding could motivate us to propose a solution. Variables in Eqs. (12) and (13) form dependent relations, as illustrated in Fig. 7. Given w_U, we can obtain π_U [from the first Eq. of (13)], and then obtain w_V [from the second Eq. of (12)], and finally obtain w'_U [from the first Eq. of (12)]. If $w_U = w_U^*$, then $w'_U = w_U^*$; otherwise, $w'_U \neq w_U \neq w_U^*$, and this forms a contradiction.

We find that this contradiction has a physical meaning, and it is fixable. The normalized weights w_U can be viewed as a distribution that describes the probability a walker jumping to a node in G. When we specify some particular weights w_U, it means that we expect the walker to jump to a node in G following a distribution specified by w_U. If $w_U \neq w_U^*$, we will derive a different w'_U using Eqs. (12) and (13). It means that the walker actually jumps to a node in G following a different distribution

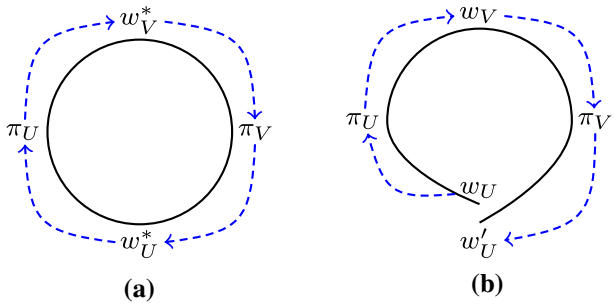

Fig. 7 Dependent relations among variables. The variable at the head of an arrow depends on the variable at the tail of the arrow. **a** Perfect weights, **b** imperfect weights

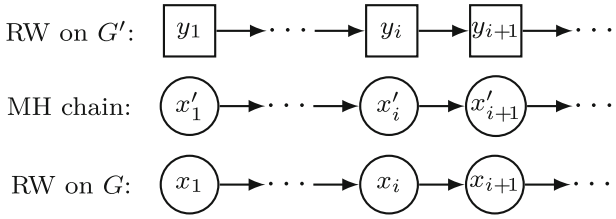

Fig. 8 Three parallel Markov chains in $\mathrm{RW^TRW^A}$

specified by w'_U. This is the reason why the random walk is not time reversible. Fortunately, with this understanding, the contradiction becomes fixable by applying the famous Metropolis–Hastings (MH) sampler (Robert and Casella 2004). We can treat (normalized) w_U as the *desired distribution*, and (normalized) w'_U as the *proposal distribution*, and we use a MH sampler to build a Markov chain (referred as the MH chain) that generates samples with the desired distribution. Each time when the walker requires jumping, it jumps to a node generated by the MH chain. This guarantees that the walker jumps to nodes in G following the desired distribution, and ensures that π_U and π_V are still the stationary distributions of the random walks.

Sampling design. The complete design of $\mathrm{RW^TRW^A}$ comprises three parallel Markov chains as illustrated in Fig. 8, and we need to specify desired weights w_U in advance, e.g., from a uniform distribution.

• *Random walk on auxiliary graph* G': Suppose the random walk resides at node $y_i \in V$ at step i. Then we can calculate w_{y_i} according to Eq. (10). At step $i+1$, the random walk executes one of the following two steps.

Jump: With probability $w_{y_i}/(d_{y_i} + w_{y_i})$, the walker jumps to a random neighbor $v \in V$ of node x_i in G_b, and $y_{i+1} = v$;

Walk: Otherwise, the walker moves to a random neighbor $v \in V$ of y_i in G', and $y_{i+1} = v$.

• *MH chain:* Suppose the MH chain resides at node x'_i at step i. At step $i+1$, we randomly choose a neighbor $u \in U$ of y_i in G_b. This is equivalent to sample a node $u \in U$ with probability proportional to w'_u.

Acceptance: With probability r_i, we accept u and $x'_{i+1} = u$, where $r_i = \min\left\{1, \left(w_u w'_{x'_i}\right) / \left(w_{x'_i} w'_u\right)\right\}$;
Rejection: Otherwise, we reject u and $x'_{i+1} = x'_i$.

• *Random walk on target graph G*: Suppose the random walk resides at node $x_i \in U$ at step i. At step $i+1$, the walker executes one of the following two steps.

Jump: With probability $w_{x_i}/(d_{x_i} + w_{x_i})$, the walker jumps to x'_{i+1}, and $x_{i+1} = x'_{i+1}$;
Walk: Otherwise, the walker moves to a random neighbor $u \in U$ of x_i in G, and $x_{i+1} = u$.

This sampling design ensures that we use only local knowledge of the three graphs to obtain a sample path $S = (x_1, \ldots, x_B)$, which can yield unbiased estimate of θ.

Estimator. Given the sample path $S = (x_1, \ldots, x_B)$, we propose to use the following estimator to estimate θ.

$$\hat{\theta}^{\mathrm{RW^TRW^A}} = \frac{1}{Z^{\mathrm{RW^TRW^A}}} \sum_{u \in S} \frac{f(u)}{d_u + w_u}, \tag{14}$$

where $Z^{\mathrm{RW^TRW^A}} \triangleq \sum_{u \in S} 1/(d_u + w_u)$.

Theorem 5 *If the target graph is connected, or could become connected by adding nodes in auxiliary graph and edges in bipartite graph, then using samples collected by RW^TRW^A, Estimator (14) provides an asymptotically unbiased estimate of θ.*

Proof Since we have constructed the Markov chain on G with stationary distribution $\pi_u \propto d_u + w_u$, the proof is exactly the same as Theorem 4. □

5 Experiments

In this section, we conduct experiments on both synthetic and real datasets to validate our sampling designs. Our goal is to demonstrate the unbiasedness of proposed estimators [(5), (6), (9), (14)] and study their estimation errors with respect to different factors such as sampling budget B and parameter settings α and β.

We consider to estimate the PDF and CCDF of degree distribution of a graph. For PDF, the characteristic function is defined as $f_d(u) \triangleq \mathbf{1}(d_u = d)$, where $\mathbf{1}(\cdot)$ is the indicator function, and the graph characteristic is the distribution $\{\theta_d\}_{d \geq 0}$ where $\theta_d = \sum_u f_d(u)/n$ is the fraction of nodes with degree d in graph G. For CCDF, the characteristic function is defined as $f_d(u) \triangleq \mathbf{1}(d_u > d)$, and the graph characteristic is the distribution $\{\theta_d\}_{d \geq 0}$ where $\theta_d = \sum_u f_d(u)/n$ is the fraction of nodes with degree larger than d in graph G. In some experiments, we will only show the results of estimating CCDF due to space limitation.

5.1 Experiments on synthetic data

In the first experiment, we validate the sampling methods using synthetic data. The purpose is to show how significant advantage can be achieved using our methods

than those methods not leveraging the auxiliary graph and bipartite graph. We will mainly consider the simple random walk (RW) and Metropolis–Hastings random walk (MHRW) as two baselines.

Synthetic data. We generate a two-layered network structure by connecting three Barabási-Albert (BA) graphs (Barabási and Albert 1999) G_1, G_2 and G_3. Each BA graph contains 100,000 nodes, and the three BA graphs have average degree 4, 10 and 20, respectively. G_1 and G_3 are connected by one edge to form the target graph G, which thus has a barbell structure. G_2 is the auxiliary graph G', and the bipartite graph G_b is formed by connecting nodes in G and G' according to the following two steps:

- connect every node in G to a randomly selected node in G';
- randomly connect 200,000 pairs of nodes, and each pair has one node in G and the other node in G'.

The first step ensures that every node in U satisfies $d_u^{(b)} > 0$ so that we can apply VS^{A} on this dataset.

Results and analysis. First we demonstrate that the proposed estimators $\hat{\theta}_d^{\text{VS}^{\text{A}}\text{-I}}$, $\hat{\theta}_d^{\text{VS}^{\text{A}}\text{-II}}$, $\hat{\theta}_d^{\text{RW}^{\text{T}}\text{VS}^{\text{A}}}$, and $\hat{\theta}_d^{\text{RW}^{\text{T}}\text{RW}^{\text{A}}}$ are asymptotically unbiased. To show this, we apply these sampling methods to estimate the fraction of nodes with degree 2 and 12 in the target graph, denoted by θ_2 and θ_{12}. We compare their estimates to the ground truth for different sampling budgets B. We also show the estimates using RW[10] and MHRW[11] on the target graph. Because the target graph has a barbell structure, both RW and MHRW are easily to be trapped into one component and fail to explore the other component. We expect to see that the RW estimator and MHRW estimator do not perform well. The results are depicted in Figs. 9 and 10. Indeed, the two baseline random walk methods incur large biases, and cannot converge to ground truth within $B = 0.01n$ steps. In comparison, our proposed estimators can obtain more accurate estimates, and it is clear to see that when sampling budget B increases, all our proposed estimators can converge to the ground truth. Hence, these results demonstrate that our proposed estimators are asymptotically unbiased and converge faster than baseline methods.

Next, we study the estimation error of each estimator for estimating the PDF and CCDF of degree distribution. We choose the *normalized rooted mean squared error* (NRMSE) as a metric to evaluate the estimation error of an estimator, which is defined as follows

$$\text{NRMSE}\left(\hat{\theta}\right) = \frac{\sqrt{\mathbb{E}\left[\left(\hat{\theta} - \theta\right)^2\right]}}{\theta}.$$

NRMSE measures the relative difference between an estimated value $\hat{\theta}$ and a real value θ. The smaller the NRMSE, the more accurate the estimator $\hat{\theta}$ is. To compare

[10] Let $\hat{\theta}^{\text{RW}}$ denote the RW estimate. Then $\hat{\theta}^{\text{RW}} = 1/Z^{\text{RW}} \sum_{u \in S} f(u)/d_u$ where samples in multiset S are collected using a simple RW in G, and $Z^{\text{RW}} = \sum_{u \in S} 1/d_u$.

[11] Let $\hat{\theta}^{\text{MHRW}}$ denote the MHRW estimate. Because MHRW obtains samples uniformly at random, thus the estimator is simply $\hat{\theta}^{\text{MHRW}} = 1/|S| \sum_{u \in S} f(u)$ where samples in multiset S are collected using a MHRW in G.

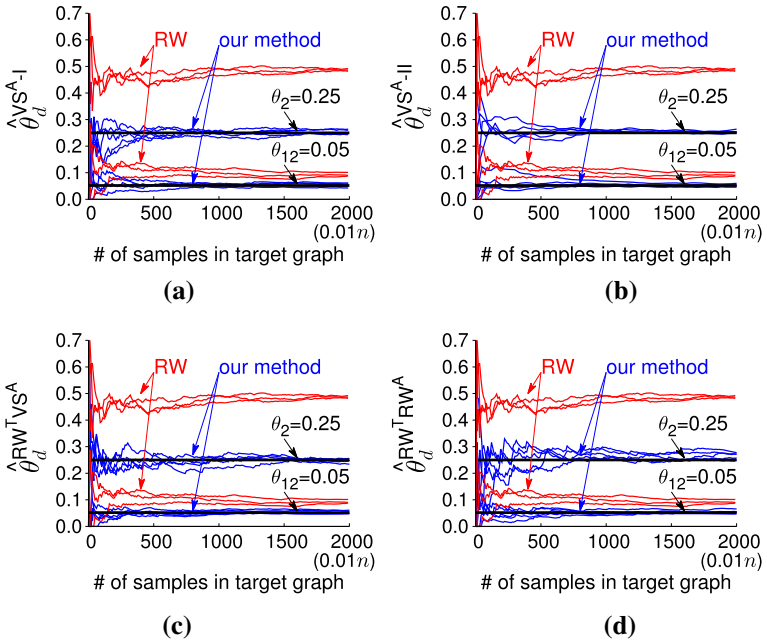

Fig. 9 Asymptotic unbiasedness of estimators (comparing with RW). **a** $\hat{\theta}_d^{\text{VS}^\text{A}\text{-I}}$, **b** $\hat{\theta}_d^{\text{VS}^\text{A}\text{-II}}$, **c** $\hat{\theta}_d^{\text{RW}^\text{T}\text{VS}^\text{A}}$ ($\alpha = 10$), **d** $\hat{\theta}_d^{\text{RW}^\text{T}\text{RW}^\text{A}}$ ($\alpha = \beta = 10$)

the NRMSE of different estimators, we fix the sampling budget B to be 1% of the target graph size, and calculate the averaged empirical NRMSE over 1000 runs. The results are depicted in Figs. 11 and 12.

To clearly see the performance difference, we also show the NRMSE of estimators based on RW and MHRW as baselines. Because RW and MHRW can hardly converge over a barbell graph within $B = 0.01n$ steps, we observe that NRMSEs of RW and MHRW are almost the largest among all estimators for low degrees. Comparing VS^A-I and VS^A-II with RW and MHRW, we find that the two VS^A estimators provide smaller PDF/CCDF NRMSE for low degree nodes than RW and MHRW. However, VS^A estimators produce larger NRMSE for high degree nodes than RW and MHRW. Therefore, VS^A can better estimate low degree nodes than high degree nodes in a graph.

The weakness of VS^A can be overcome by $\text{RW}^\text{T}\text{VS}^\text{A}$ and $\text{RW}^\text{T}\text{RW}^\text{A}$. From Fig. 12, it is clearer to see that when indirect jumps are incorporated into random walks in $\text{RW}^\text{T}\text{VS}^\text{A}$ and $\text{RW}^\text{T}\text{RW}^\text{A}$, NRMSE for high degree nodes decreases, and NRMSE for low degree nodes remains smaller than RW and MHRW. If we increase the probability of jumping at each step of random walk by increasing α and β, we observe that NRMSE for low degree nodes decreases, but NRMSE for high degree nodes increases. This behavior is similar to RWwJ (Avrachenkov et al. 2010; Ribeiro et al. 2012) and demonstrates that the indirect jumps in $\text{RW}^\text{T}\text{VS}^\text{A}$ and $\text{RW}^\text{T}\text{RW}^\text{A}$ indeed behave similarly as the direct jumps in RWwJs.

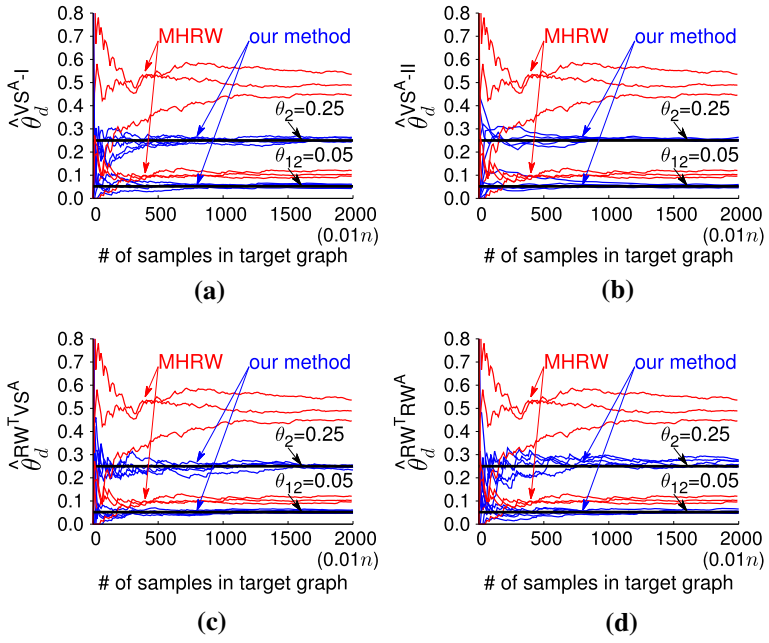

Fig. 10 Asymptotic unbiasedness of estimators (comparing with MHRW). **a** $\hat{\theta}_d^{\mathrm{VS^A\text{-}I}}$, **b** $\hat{\theta}_d^{\mathrm{VS^A\text{-}II}}$, **c** $\hat{\theta}_d^{\mathrm{RW^TVS^A}}$ ($\alpha = 10$), **d** $\hat{\theta}_d^{\mathrm{RW^TRW^A}}$ ($\alpha = \beta = 10$)

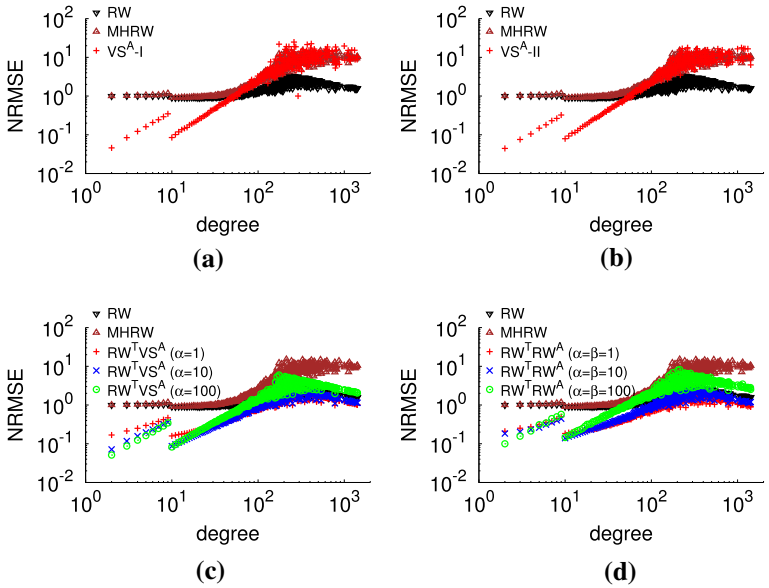

Fig. 11 PDF NRMSE of different estimators. **a** $\mathrm{VS^A}$-I, **b** $\mathrm{VS^A}$-II, **c** $\mathrm{RW^TVS^A}$, **d** $\mathrm{RW^TRW^A}$

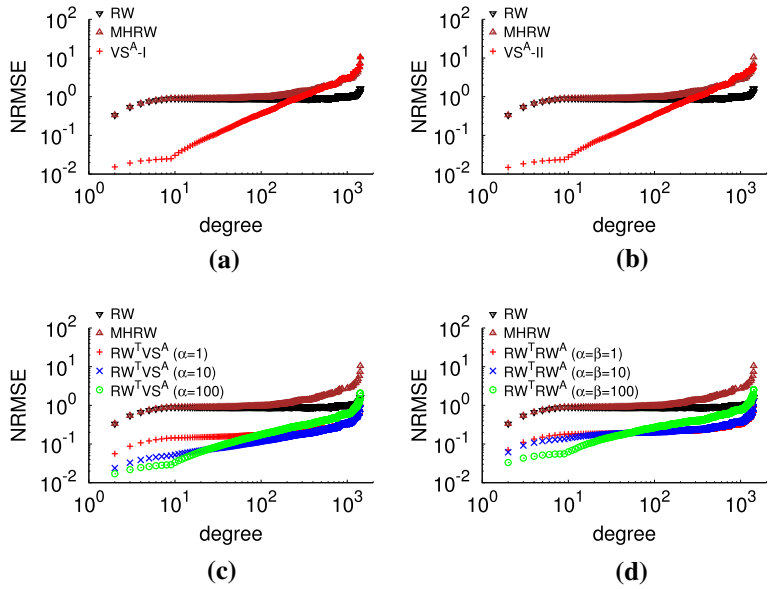

Fig. 12 CCDF NRMSE of different estimators. **a** VS^A-I, **b** VS^A-II, **c** RW^TVS^A, **d** RW^TRW^A

Last, we empirically study the sampling cost of our proposed methods. The cost is mainly caused by conducting indirect jumps in our methods. However, strictly quantifying the cost of an indirect jump is nontrivial because many factors are related to cost, e.g., the sparsity of ID space in the graph, what information can be leveraged for sampling from a node, etc. To simplify our analysis, we consider a simple cost model inspired by Avrachenkov et al. (2010). Assume each indirect jump incurs $c \geq 1$ penalty from the budget, and a regular walk step incurs one penalty from the budget. (In other words, a jump step costs c units in the budget and a walk step costs just one unit.) Thus, a jump step is c times more expensive than a walk step. Then, we can compare the estimation error of our methods with RW (or MHRW) under different c's, and know how sampling cost is related to estimation performance. Using RW^TVS^A as an example, we fix $\alpha = 1$, budget $B = 0.01n$, and set $c = 1, 10, 100$ and 1000 respectively. We show the NRMSE of our methods and RW in Fig. 13. We observe that, when c increases, the estimation accuracy indeed decreases. From the CCDF NRMSE plots, it is clearer to see that when c increases from 1 to 100, RW^TVS^A is still better than RW. When $c = 1000$, RW^TVS^A becomes worse than RW. Therefore, we conclude that, even when the cost of an indirect jump step is 100 times more expensive than a walk step, our method still improves estimation

5.2 Experiments on LBSN datasets

In the second experiment, we apply the VS^A-II method on two real-world LBSN datasets to solve the problem in Example 1, i.e., measure user characteristics in an area of interest on the map.

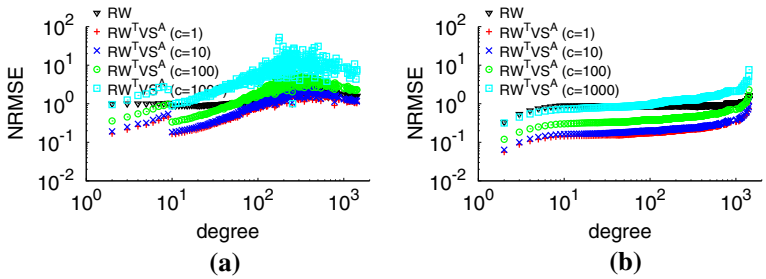

Fig. 13 Sampling cost of $\mathrm{RW^TVS^A}$ ($\alpha = 1$, $B = 0.01n$). **a** PDF, **b** CCDF

Table 1 Summary of two LBSN datasets

Dataset		Brightkite	Gowalla
G	Network type	Undirected	Undirected
	Users	58,228	196,591
	Friendship edges	214,078	950,327
	Users in LCC[1]	56,739	196,591
	Edges in LCC	212,945	950,327
G' and G_b	Venues	772,966	1,280,969
	Users having check-ins	51,406	107,092
	Check-ins	4,491,143	6,442,890
G' and G_b for NYC	Venues in NYC[2]	23,484	26,448
	Users checking in NYC	4,257	7,399
	Check-ins in NYC	33,656	113,423

[1]The largest connected component
[2]The New York City (Fig. 14)

LBSN datasets. We use two public LBSN datasets from Brightkite and Gowalla (Cho et al. 2011). Brightkite and Gowalla are two popular LBSNs where users shared their locations by checking-in. Users in the two LBSNs are also connected by undirected friendship relations, which form two user social networks. The statistics of these two datasets are summarized in Table 1.

Because we are only interested in users that have check-ins, i.e., each node in the target graph connects to at least one node in the auxiliary graph, $\mathrm{VS^A}$ is applicable on these two datasets. Suppose that we want to measure characteristics of users located around New York City (NYC), which is specified by a rectangle region on a map: latitude range 40.4° to 41.4°, longitude range $-74.3°$ to $-73.3°$ (see Fig. 14). The goal is to estimate degree distribution of the users who checked in this region. As we explained in introduction, directly sampling users is inefficient. Here, we apply the $\mathrm{VS^A}$-II along with a venue sampling method—random region zoom-in (RRZI; Wang et al. 2014a) to illustrate how to sample users in NYC more efficiently.

Venue sampling. RRZI utilizes a venue query API provided by LBSNs to sample venues on a map. The API requires a user to specify a rectangle region by providing

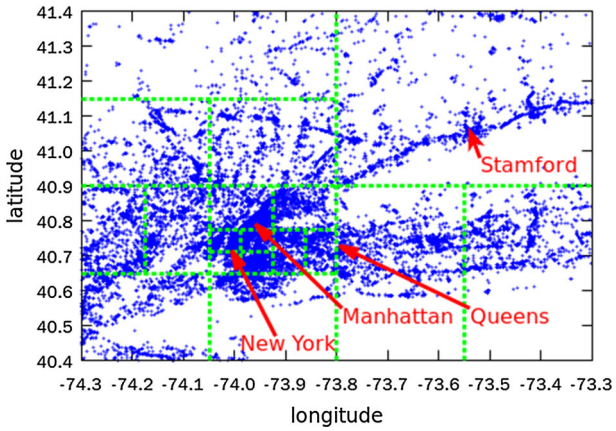

Fig. 14 Venue distribution in New York City and illustration of accessible subregions used by RRZI. Each subregion contains less than K venues

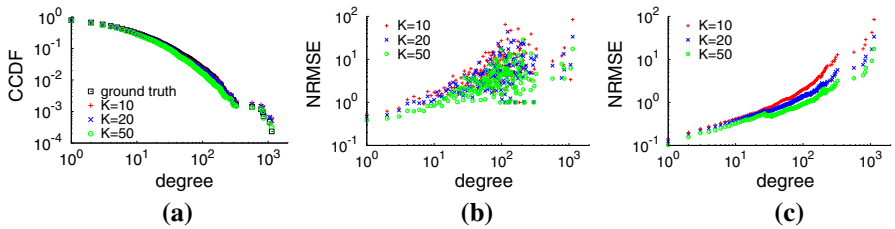

Fig. 15 Performance of RRZI-VS^{A} on Brightkite. **a** Estimates (Brightkite), **b** PDF NRMSE (Brightkite), **c** CCDF NRMSE (Brightkite)

the south-west and north-east corners latitude-longitude coordinates, and then the API returns a set of venues in this region. Usually, the API only returns at most K venues in a queried region. RRZI regularly zooms in the region until the subregion is fully *accessible*, i.e., the API returns strictly less than K venues in the subregion. The zooming-in process is equivalent to dividing the region into many non-overlapping accessible subregions, as illustrated in Fig. 14, and each subregion is associated with a fixed probability related to the zooming-in strategy. When the final query subregion becomes accessible, we random pick a venue v uniformly at random from the returned venues, and its sampling probability p_v can be calculated in RRZI.

Results. Combining VS^{A}-II with RRZI, denoted by RRZI-VS^{A}, we conduct experiments on Brightkite and Gowalla to indirectly sample users in NYC. We totally sample 5% of venues in NYC and calculate the degree distribution of users in NYC. The results are depicted in Figs. 15 and 16.

Figures 15a and 16a depict the estimates of CCDF with different query capacity K. We observe that our RRZI-VS^{A} method can provide good estimates of user characteristics in NYC on both datasets. Specifically, the estimates for low degree users are better than high degree users, and this is clear to see from the PDF/CCDF NRMSE plots. This feature coincides with our previous analysis using synthetic data. From the

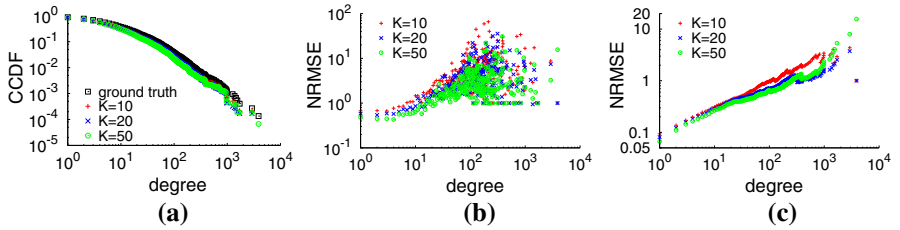

Fig. 16 Performance of RRZI-VS^{A} on Gowalla. **a** Estimates, **b** PDF NRMSE, **c** CCDF NRMSE

Table 2 Amazon product co-purchasing network statistics

G	Product co-purchasing network	Undirected
	# of products	4,015,942
	# of co-purchases	78,792,050
G'	# of categories	10,164
G_b	# of product-category associations	15,829,046
	Avg. # of categories a product belongs to	4
	Avg. # of products in a category	1557

NRMSE plots, we can also find an approximate law that a larger query capacity K, i.e., the maximum number of venues the API can return, reduces the estimation error of RRZI-VS^{A}. However, it is not true for estimating high degree users on Gowalla in Fig. 16c. In fact, a better way to reduce estimation error is to combine VS^{A}-II with other better venue sampling methods discussed in Li et al. (2012, 2014) and Wang et al. (2014a).

5.3 Experiments on amazon product co-purchasing network

In the third experiment, we compare the performance of VS^{A}-I and RW^{T}VS^{A} sampling methods on the Amazon product co-purchasing network.

Amazon product co-purchasing network. We build an Amazon product co-purchasing network from the Amazon dataset provided by McAuley et al. (2015). The network is created based on "customers who bought this item also bought" feature of the Amazon website. That is, if a product i is co-purchased with product j, the network contains an undirected edge between i and j. In addition, each product belongs to at least one category on Amazon, and Amazon provides a complete category list on its homepage to facilitate customers to conveniently browse the products. Thus, we can leverage this category list to perform indirect sampling of the co-purchasing network. The detailed statistics of the Amazon dataset are provided in Table 2.

This dataset is suitable for us to study the performance of VS^{A}-I and RW^{T}VS^{A}, where the availability of the complete category list allows us to conduct uniform vertex sampling on the auxiliary graph. Here we sample 1% of the nodes from target graph, and compare the accuracy of estimating PDF/CCDF degree distribution using different methods. The results are averaged over 1000 runs and are depicted in Fig. 17.

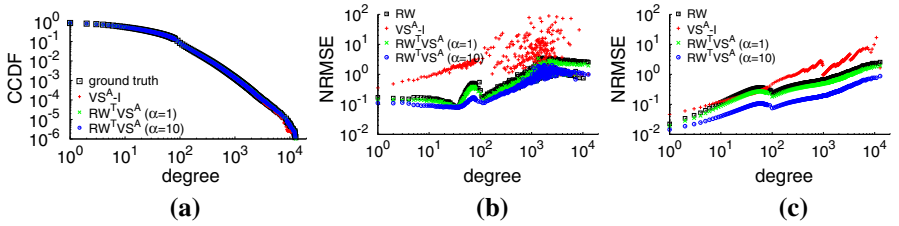

Fig. 17 Amazon product co-purchasing network characterizing. **a** Unbiasedness of CCDF estimates, **b** PDF NRMSE, **c** CCDF NRMSE

Results. From Fig. 17a, we observe that the two methods can indeed provide unbiased estimates of the CCDF. From Fig. 17b, c, we also observe that different methods have different estimation accuracy. In general, VS^A-I has relatively large estimation error, then comes the random walk estimator, and RW^TVS^A has the lowest estimation error among these three estimators. RW^TVS^A leverages the category list to perform indirect jumps on the target graph, and this approach can significantly improve the estimation accuracy. If we slightly increase α to increase the jumping probability, we observe that the estimation error further decreases.

5.4 Experiments on MTime dataset

In the fourth experiment, we apply RW^TVS^A and RW^TRW^A on MTime to measure the MTime user characteristics.

MTime dataset. MTime[12] is a popular online movie database in China, which comprises two types of accounts: MTime users and movie actors. MTime users can follow each other to form a social network, and movie actors can form connections with each other if they cooperated in the same movies. A MTime user can follow movie actors if she is a fan of the actor. Suppose we want to measure MTime user characteristics, then the relations between MTime users and movie actors naturally form a two-layered network structure, where

- the target graph consists of MTime users and their following relations;
- the auxiliary graph consists of movie actors and their cooperation relations;
- and the bipartite graph consists of MTime users, movie actors and the fan relations between them.

To build a groundtruth dataset, we have collected the complete MTime network by traversing MTime user and movie actor ID spaces.[13] For each MTime user, we collect the set of users she follows and users who follow her. This builds up a directed follower network among MTime users. Each MTime user maintains a list including a subset of movie actors she is interested in. This information is used to build up the fan-relations between MTime users and movie actors. For each movie actor, we collect the movies she participated in, and if two actors participated in a same movie, we connect them.

[12] http://www.mtime.com.

[13] The user ID space ranges from 100,000 to 10,000,000, and actor ID space ranges from 892,000 to 2,100,000.

Table 3 Summary of the MTime dataset

G	User follower network type	Directed
	Total users (isolated and non-isolated)	1,878,127
	Non-isolated users in follower network	1,035,164
	Following relations	14,861,383
	Users in LCC	987,055
	Following relations in LCC	14,791,482
G'	Actor cooperative network type	undirected
	Total actors (isolated and non-isolated)	1,123,340
	Non-isolated actors in cooperative network	1,122,166
	Cooperative relations	10,344,364
	Actors in LCC	1,114,065
	Cooperative relations in LCC	10,328,904
G_b	Fan relations	225,558,343
	Users following actors	1,419,339
	Isolated users following actors	842,963
	Actors having fans	441,413
	Isolated actors having fans	1174
	Isolated actors having only isolated fans	225
	Isolated users following only isolated actors	393

This builds up a cooperative network among actors. The complete MTime dataset is summarized in Table 3.

Analysis of the dataset. First we provide some analysis about the MTime dataset. In Table 3, comparing the first block with second block, which are related to target graph G and auxiliary graph G' respectively, we find that about 19% of the user IDs and 93% of the actor IDs are valid. This indicates that conducting UNI on the auxiliary graph is more efficient than conducting UNI on the target graph. Moreover, we find that more than 47% of the MTime users are not in LCC, but the number for actors is less than 0.1%. This indicates that the auxiliary graph is better connected than the target graph. Although a large fraction of users are isolated nodes in the target graph, from the last block, we find that almost all the isolated users are connected to non-isolated actors (except a few hundreds of them). So the majority of isolated users are indirectly connected to other users through actors. This is illustrated in Fig. 18. The advantage of introducing the two-layered network structure is now clear for MTime dataset, i.e., we can study a larger user space than simply the LCC of target graph.

Results. Using the MTime dataset as a testbed, we demonstrate that $\mathrm{RW^TVS^A}$ and $\mathrm{RW^TRW^A}$ methods can provide good estimates of user characteristics. Although the user follower network is directed, we can build an undirected version of the target graph on-the-fly while sampling because a user's in-coming and out-going neighbors are known once the user is queried (Ribeiro and Towsley 2010; Ribeiro et al. 2012). Slightly different from previous experiments, here we will estimate both the in- and out-degree distributions.

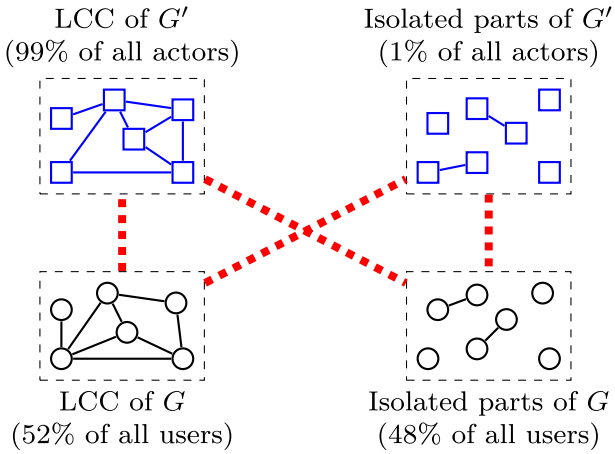

Fig. 18 The MTime network components. Dashed red lines denote fan relations between actors and users

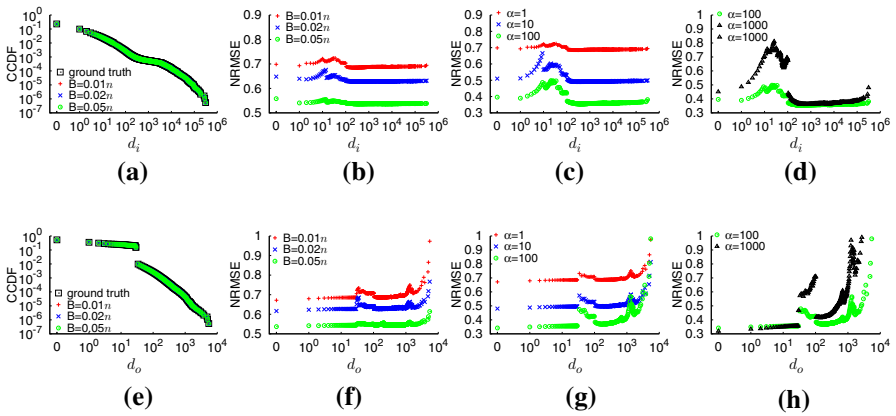

Fig. 19 $\mathrm{RW^TVS^A}$ degree distribution estimation and NRMSE analysis. **a** In-degree estimates ($\alpha = 1$), **b** CCDF NRMSE ($\alpha = 1$), **c** CCDF NRMSE ($B = 0.01n$), **d** too frequent jumping ($B = 0.01n$), **e** out-degree estimates ($\alpha = 1$), **f** CCDF NRMSE ($\alpha = 1$), **g** CCDF NRMSE ($B = 0.01n$), **h** too frequent jumping ($B = 0.01n$)

Figure 19 depicts the results of $\mathrm{RW^TVS^A}$. In Fig. 19a, e, we show the in-degree and out-degree CCDF estimates. We can see that $\mathrm{RW^TVS^A}$ can provide unbiased estimates. From Fig. 19b, f, we observe that when sampling budget increases, the NRMSE decreases for both in-degree and out-degree estimations. From Fig. 19c, g we observe that when more jumps are allowed by increasing α from 1 to 100, estimation accuracy also increases.

Figure 20 depicts the results of $\mathrm{RW^TRW^A}$, and they are similar to the results of $\mathrm{RW^TVS^A}$. First, from Fig. 20a, e, we observe that $\mathrm{RW^TRW^A}$ can also provide unbiased estimates of the in- and out-degree distributions. Second, from Fig. 20b, f, we can find that as sampling budget increases, the estimation error decreases accordingly for both in- and out-degree estimations. Last, from Fig. 20c, g, we find that when jumping probability increases (by increasing α and β), the NRMSE also decreases.

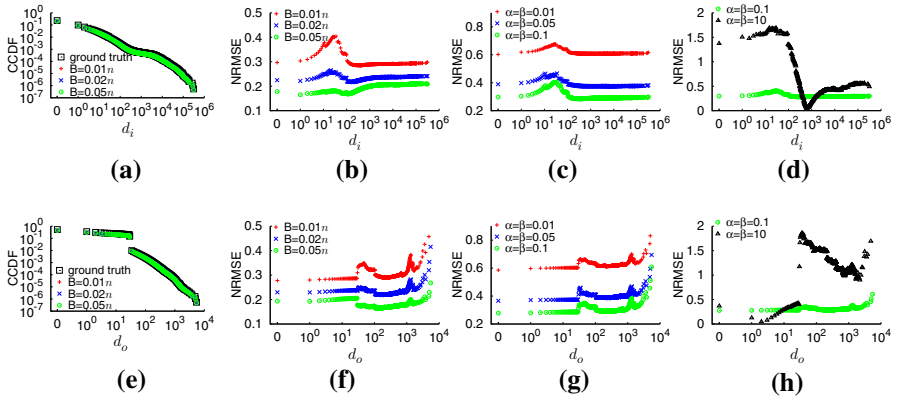

Fig. 20 $\mathrm{RW^TRW^A}$ degree distribution estimation and NRMSE analysis, **a** in-degree est. ($\alpha = \beta = 0.1$), **b** CCDF NRMSE ($\alpha = \beta = 0.1$), **c** CCDF NRMSE ($B = 0.01n$), **d** too frequent jumping ($B = 0.01n$), **e** out-degree est. ($\alpha = \beta = 0.1$), **f** CCDF NRMSE ($\alpha = \beta = 0.1$), **g** CCDF NRMSE ($B = 0.01n$), **h** too frequent jumping ($B = 0.01n$)

However, it is worth noting that α and β should not be too large for both $\mathrm{RW^TVS^A}$ and $\mathrm{RW^TRW^A}$. Because we know that when $\alpha \to \infty$, $\mathrm{RW^TVS^A}$ becomes $\mathrm{VS^A}$, which is biased on the MTime dataset, and hence causes large NRMSE. Similar behavior happens to $\mathrm{RW^TRW^A}$.

6 Related work

Graph sampling methods, especially random walk based graph sampling methods, have been widely used to characterize large-scale complex networks. These applications include, but are not limited to, estimating peer statistics in peer-to-peer networks (Gkantsidis et al. 2006; Massoulié et al. 2006), uniformly sampling users from OSNs (Gjoka et al. 2010, 2011b; Lee et al. 2012; Xu et al. 2014), characterizing structure properties of large-scale networks (Katzir et al. 2011; Hardiman and Katzir 2013; Seshadhri et al. 2013; Wang et al. 2014b), and measuring statistics of point-of-interests on maps (Wang et al. 2014a). The above literature is mostly concerned with sampling methods that seek to *directly* sample nodes (or samples) in target graphs (or some sample spaces). However, direct sampling is not always efficient as we argued in this work.

When the target graph (or sample space) cannot be directly sampled or direct sampling is inefficient, several methods based on graph manipulation have been proposed to improve sampling efficiency. For example, Gjoka et al. (2011a) study an approach to improve sampling efficiency through building a *multigraph* using different kinds of relations (i.e., different types of edges) that exist on an OSN. A multigraph is better connected than any individual graph formed by only one kind of relations. Therefore, the random walk can converge fast on this multigraph. Zhou et al. (2013) exploit several criteria to rewire the target graph on-the-fly to increase the graph conductance (Sinclair and Jerrum 1989) and reduce mixing time of a random walk. Our method differs from theirs in that we do not manipulate target graph structure. We study a new approach that utilizes a widely existed two-layered network structure to assist sampling on target graph indirectly. In parallel to our methods by leveraging

two-layered network structures, there have been recent progresses on speeding up random walk on graphs by leveraging historical sampling information (Zhou et al. 2015) and Rao–Blackwellization property of Monte Carlo methods (Lee et al. 2017).

Birnbaum and Sirken (1965) designed a survey method for estimating the number of diagnosed cases of a rare disease in a population. Directly sampling patients of a rare disease from the huge human population is obviously inefficient, so they studied how to sample hospitals so as to sample patients indirectly. Their method motivates us to design the VS^{A} method. However, as we pointed out, VS^{A} method cannot sample nodes that are not connected to auxiliary graph, and we overcome this problem by designing $\text{RW}^{\text{T}}\text{VS}^{\text{A}}$ and $\text{RW}^{\text{T}}\text{RW}^{\text{A}}$ methods. Our work also complements existing sampling methods related to random walk with jumps (Avrachenkov et al. 2010; Ribeiro et al. 2012; Xu et al. 2014) by removing the necessity of uniform node sampling on target graphs.

7 Limitions and future work

This paper developed several new sampling methods for measuring network nodal characteristics using random walk with indirect jumps by leveraging the two-layered network structure. There are several limitions/weaknesses that we did not study or not discussed in detail, and hence they offer opportunities for future work.

- *Sampling cost analysis is shallow.* We only qualitatively discussed the sampling cost of our methods, and empirically studied a very simple cost model. Hence, a more comprehensive sampling cost analysis is required for better understanding the performance and use conditions of proposed methods. Sampling cost analysis can also help us to choose the optimal parameters in the proposed methods.
- *Extension to the MHRW framework.* Our methods are based on the simple random walk framework, and frequently use the re-weighting strategy to remove estimation bias. In the literature, MHRW is also a popular random walk sampling method that has the ability of collecting uniform samples from a network, and hence it has no necessity for re-weighting. Thus, there is an opportunity to extend our framework to the MHRW case.
- *From two-layered network structure to multi-layered network structure.* It is also possible to consider more-than-two layered network structure, and such multi-layered network structures do exist in reality. However, sampling cost needs to be carefully considered and justified, and we leave this interesting extension to future work.

8 Conclusion

When graphs become large in scale, sampling methods become necessary tools in the study of characterizing their properties. Among these sampling methods, random walk-based crawling methods are effective and are gaining popularity. However, if the graph under study is not well connected, random walk-based graph sampling methods suffer from the slow mixing problem. In this work, we observe that a graph usually does not exist in isolation. In many applications, the target graph is accompanied with

an auxiliary graph and a bipartite graph, and they together form a better connected two-layered network structure. This new viewpoint brings extra benefits to the graph sampling framework. We design three sampling methods to measure the target graph from this new viewpoint, and these methods are demonstrated to be effective on both synthetic and real datasets. Therefore, our method complements existing methods in the literature of graph sampling.

Acknowledgements The authors wish to thank the anonymous reviewers for their helpful feedback. The research presented in this paper is supported in part by National Key R&D Program of China (2018YFC0830500), National Natural Science Foundation of China (U1301254, 61603290, 61602371, 61772412), the Ministry of Education&China Mobile Research Fund (MCM20160311), the Natural Science Foundation of Jiangsu Province (SBK2014021758), 111 International Collaboration Program of China, the Prospective Joint Research of Industry-Academia-Research Joint Innovation Funding of Jiangsu Province (BY2014074), Shenzhen Basic Research Grant (JCYJ20160229195940462, JCYJ20170816100819428), China Postdoctoral Science Foundation (2015M582663), Natural Science Basic Research Plan in Shaanxi Province of China (2016JQ6034). The work by John C. S. Lui was supported in part by GRF 14208816.

Appendix

See Table 4.

Table 4 Notations

UNI	Uniform vertex sampling
RW	Random walk
MHRW	Metropolis–Hastings random walk
RWwJ	Random walk with jumps
PDF	Probability distribution function
CCDF	Complementary cumulative distribution function
NRMSE	Normalized rooted mean squared error
LCC	Largest connected component
$G = (U, E)$	Target graph with node set U and edge set E
$G' = \left(V, E'\right)$	Auxiliary graph with node set V and edge set E'
$G_b = (U, V, E_b)$	Bipartite graph with node sets U, V and edge set E_b
$U_v \subseteq U, V_u \subseteq V$	Subsets of nodes in G or G'
n, n'	$n = \lvert U\rvert$ and $n' = \lvert V\rvert$
$\theta, \hat{\theta}$	Target graph characteristic ground truth/estimate
$f: U \mapsto \mathbb{R}$	Characteristic function
$u \in U, v \in V$	Nodes in target/auxiliary graph
$d_\cdot, d_\cdot^{(b)}$	Degree of a node in G, G' or G_b
α, β	Parameters controlling jumping probability
$p_v \propto a_v$	Probability of sampling node v
p_{uj}	Markov chain transition probability from node u to node j
w_u, ω_v	Edge weights
$\pi_\cdot$	Stationary probability of a Markov chain
B	Sampling budget

References

Avrachenkov K, Ribeiro B, Towsley D (2010) Improving random walk estimation accuracy with uniform restarts. In: Proceedings of the 7th workshop on algorithms and models for the web graph

Backstrom L, Kleinberg J (2014) Romantic partnerships and the dispersion of social ties: a network analysis of relationship status on Facebook. In: Proceedings of the 17th ACM conference on computer supported cooperative work and social computing

Barabási AL, Albert R (1999) Emergence of scaling in random networks. Science 286(5439):509–512

Birnbaum ZW, Sirken MG (1965) Design of sample surveys to estimate the prevalence of rare diseases: three unbiased estimates. Vital Health Stat 2(11):1–8

Cho E, Myers SA, Leskovec J (2011) Friendship and mobility: user movement in location-based social networks. In: Proceedings of the 17th ACM SIGKDD international conference on knowledge discovery and data mining

Gjoka M, Kurant M, Butts CT, Markopoulou A (2010) Walking in Facebook: a case study of unbiased sampling of OSNs. In: Proceedings of the 29th annual IEEE international conference on computer communications

Gjoka M, Butts CT, Kurant M, Markopoulou A (2011a) Multigraph sampling of online social networks. IEEE J Sel Areas Commun 29(9):1893–1905

Gjoka M, Kurant M, Butts CT, Markopoulou A (2011b) Practical recommendations on crawling online social networks. IEEE J Sel Areas Commun 29(9):1872–1892

Gkantsidis C, Mihail M, Saberi A (2006) Random walks in peer-to-peer networks: algorithms and evaluation. Perform Eval 63(3):241–263

Han J, Choi D, Chun BG, Kwon TT, Chul Kim H, Choi Y (2014) Collecting, organizing, and sharing pins in Pinterest: interest-driven or social-driven? In: Proceedings of the ACM special interest group (SIG) for the computer systems performance evaluation community

Hardiman SJ, Katzir L (2013) Estimating clustering coefficients and size of social networks via random walk. In: Proceeding of the 22nd international world wide web conference

Katzir L, Liberty E, Somekh O (2011) Estimating sizes of social networks via biased sampling. In: Proceedings of the 19th international world wide web conference

Lee CH, Xu X, Eun DY (2012) Beyond random walk and Metropolis–Hastings samplers: why you should not backtrack for unbiased graph sampling. In: Proceedings of the ACM special interest group (SIG) for the computer systems performance evaluation community

Lee CH, Xu X, Eun DY (2017) On the Rao–Blackwellization and its application for graph sampling via neighborhood exploration. In: Proceedings of the 36th annual IEEE international conference on computer communications

Leskovec J, Huttenlocher D, Kleinberg J (2010) Signed networks in social media. In: Proceedings of the SIGCHI conference on human factors in computing systems

Li Y, Steiner M, Wang L, Zhang ZL, Bao J (2012) Dissecting foursquare venue popularity via random region sampling. In: Proceedings of the 8th international conference on emerging networking experiments and technologies

Li Y, Wang L, Steiner M, Bao J, Zhu T (2014) Region sampling and estimation of geosocial data with dynamic range calibration. In: Proceedings of the 30th IEEE international conference on data engineering

Li H, Ai W, Liu X, Tang J, Huang G, Feng F, Mei Q (2016) Voting with their feet: inferring user preferences from app management activities. In: Proceedings of the 25th international world wide web conference

Massoulié L, Merrer EL, Kermarrec AM, Ganesh A (2006) Peer counting and sampling in overlay networks: random walk methods. In: Proceedings of ACM symposium on principles of distributed computing

McAuley J, Pandey R, Leskovec J (2015) Inferring networks of substitutable and complementary products. In: Proceedings of the 21st ACM SIGKDD international conference on knowledge discovery and data mining

Meyn S, Tweedie RL (2009) Markov Chains and statistic stability, 2nd edn. Cambridge University Press, Cambridge

Mohaisen A, Yun A, Kim Y (2010) Measuring the mixing time of social graphs. In: Proceedings of the 10th ACM SIGCOMM conference on Internet measurement conference

Mondal M, Viswanath B, Druschel P, Gummadi KP, Clement A, Mislove A, Post A (2012) Defending against large-scale crawls in online social networks. In: Proceedings of the 8th international conference on emerging networking experiments and technologies

Ribeiro B, Towsley D (2010) Estimating and sampling graphs with multidimensional random walks. In: Proceedings of the 10th ACM SIGCOMM conference on Internet measurement conference
Ribeiro B, Wang P, Murai F, Towsley D (2012) Sampling directed graphs with random walks. In: Proceedings of the 31st annual IEEE international conference on computer communications
Robert CP, Casella G (2004) Monte Carlo statistic methods, 2nd edn. Springer, Berlin
Seshadhri C, Pinar A, Kolda TG (2013) Triadic measures on graphs: the power of wedge sampling. In: Proceedings of the 13th SIAM international conference on data mining
Sinclair A, Jerrum M (1989) Approximate counting, uniform generation and rapidly mixing Markov chains. Inf Comput 82(1):93–133
Wang P, He W, Liu X (2014a) An efficient sampling method for characterizing points of interests on maps. In: Proceedings of the 30th IEEE international conference on data engineering
Wang P, Lui JC, Ribeiro B, Towsley D, Zhao J, Guan X (2014b) Efficiently estimating motif statistics of large networks. ACM Trans Knowl Discov Data 9(2):1–27
Xu X, Lee CH, Eun DY (2014) A general framework of hybrid graph sampling for complex network analysis. In: Proceedings of the 33rd annual IEEE international conference on computer communications
Zhang B, Kreitz G, Isaksson M, Ubillos J, Urdaneta G, Pouwelse JA, Epema D (2013) Understanding user behavior in Spotify. In: Proceedings of the 32nd annual IEEE international conference on computer communications
Zhao J, Lui JC, Towsley D, Wang P, Guan X (2015) A tale of three graphs: sampling design on hybrid social-affiliation networks. In: Proceedings of the 31st IEEE international conference on data engineering
Zhou Z, Zhang N, Gong Z, Das G (2013) Faster random walks by rewiring online social networks on-the-fly. In: Proceedings of the 29th IEEE international conference on data engineering
Zhou Z, Zhang N, Das G (2015) Leveraging history for faster sampling of online social networks. In: Proceedings of the VLDB endowment

Affiliations

Junzhou Zhao[1] · Pinghui Wang[2] · John C. S. Lui[1] · Don Towsley[3] · Xiaohong Guan[2]

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

Junzhou Zhao
junzhouzhao@gmail.com

John C. S. Lui
cslui@cse.cuhk.edu.hk

Don Towsley
towsley@cs.umass.edu

Xiaohong Guan
xhguan@mail.xjtu.edu.cn

1 The Chinese University of Hong Kong, Hong Kong, China

2 Xi'an Jiaotong University, Xi'an, China

3 University of Massachusetts at Amherst, Amherst, USA