Journal of Statistical Mechanics: Theory and Experiment
An IOP and SISSA journal

PAPER: Interdisciplinary statistical mechanics

Sampling on bipartite networks: a comparative analysis of eight crawling methods

Saran Chen[1], Xin Lu[2,3,4], Zhong Liu[1] and Zhongwei Jia[5]

[1] College of Systems Engineering, National University of Defense Technology, Changsha, 410073, People's Republic of China
[2] School of Business, Central South University, Changsha, People's Republic of China
[3] School of Mathematics and Big Data, Foshan University, Foshan, 528000, People's Republic of China
[4] Department of Public Health Sciences, Karolinska Institutet, 17177 Stockholm, Sweden
[5] National Institute of Drug Dependence, Health Science Center, Peking University, Beijing, 100191, People's Republic of China
E-mail: xin.lu@flowminder.org

Received 5 March 2018
Accepted for publication 10 June 2018
Published 11 July 2018

Online at stacks.iop.org/JSTAT/2018/073403
https://doi.org/10.1088/1742-5468/aace0f

Abstract. Sampling networks via crawling has become a feasible and widely used approach when the global network information is difficult to obtain. But there is little focus on two-mode networks, i.e. bipartite networks in which nodes can be divided into two disjoint partitions. In this paper, we adopt eight popular crawling methods (BFS, DFS, FFS, RW, SNS, MHRW, MDRW and RDS) from studies of one-mode networks and evaluate their applicability and performance on bipartite networks. Simulation results show that Metropolis–Hastings random walk (MHRW), maximum-degree random walk (MDRW) and respondent-driven sampling (RDS) perform better than the other methods, and population estimates from them are minimally affected by the structures of degree distribution, number of nodes in two node layers, degree correlation and communities. In addition, we find that strategies used in the sampling design—selection approaches for seed nodes, the number of seed nodes, and the number of branches—have very little influence on the estimation bias. Finally, we list suggestions for the selection of crawling methods on bipartite networks under different situations.

1742-5468/18/073403+22$33.00

Keywords: information technology networks, random graphs, networks, socio-economic networks, statistical inference

Contents

1. Introduction

Network sampling has become a critical technique for studying large-scale complex systems. It provides feasible, cost-efficient ways to analyze a network's properties through samples. Many network sampling methods have been proposed, e.g. random node sampling [1–3] and random edge sampling [2, 4, 5]. When a sampling frame has been constructed with the network's global information, random node sampling and random edge sampling can easily obtain a uniform sample of nodes and edges. In practice, however, global information has remained difficult to obtain. The size and topological structure of real social networks remain incomplete [6, 7], while the space of user IDs is sparse in online social platforms [8, 9].

To overcome the difficulties in accessing the global information of networks, several crawling methods [2, 4, 10, 11] (also called 'graph-exploring methods') have been proposed. The crawling methods begin with one or more nodes and then explore nodes in the vicinity without requiring the sampling frame or global structure of the network [2]. The most popular methods include breadth first search (BFS) [12], depth first search (DFS) [12], forest-fire sampling (FFS) [2], random walk (RW) [13], snowball sampling (SNS) [14], respondent driven sampling (RDS) [15], Metropolis–Hastings random walk (MHRW) [16] and maximum-degree random walk (MDRW) [17]. Many studies have examined crawling method strategies, evaluating their differences in terms of efficiency and bias [2, 10, 11, 18, 19], how well they improve upon existing methods with certain prior knowledge [6, 20–24], and how they can be applied to empirically assess real-world networks, like online social platforms [11, 25], P2P networks [26, 27], and hidden populations [28–30]. These studies have focused on one-mode networks, which have only one type of nodes.

However, many real-world systems are naturally represented as bipartite networks in which nodes are divided into two disjoint partitions, which is to say as dual-layered networks. For example, many artistic collaborative networks are formed with nodes representing musicians and live musical performances and edges representing the participation of musical artists in shows [31, 32]. Scientific collaboration networks are formed with nodes representing scientists and research publications and edges representing scientists' contributions to the publications [33, 34]. Protein interaction networks are formed with nodes representing two types of proteins and edges representing the bonds between them [35, 36]. E-commerce platforms are represented as nodes standing for consumers and commodities and edges that show documented consumer purchases [37, 38]. Product recommendation systems are formed with nodes representing users and products and edges representing the online comments with which users recommend items to others [39, 40]. And online forums are modeled with nodes representing users and forums and edges representing how users' interactions aggregate to shape forums' characteristics [41, 42]. Many studies have analyzed bipartite networks. These have ranged from empirical analyses [37, 40], characterizations of networks [43, 44], projections from bipartite structure to monopartite structure [45, 46], generation modeling [47, 48], and community detection [49, 50].

However, little attention has been paid to the study of sampling methods for bipartite networks [51]. Many real-world bipartite networks are too large to afford researchers access to all relevant data. For example, acquiring complete data for online e-commerce

J. Stat. Mech. (2018) 073403

platforms or online rating systems is unfeasible. Although it is possible to project the bipartite network into a one-mode network and to conduct sampling methods on this projected network, the one-mode projection generally brings lots of drawbacks such as losing information about the original networks and that the global information of networks is needed for the projection operation.

In this paper, we focus on the above-mentioned crawling methods, which are regularly used in the study of one-mode networks. We evaluate the feasibility and effectiveness of applying them to bipartite networks. From samples, we aim to estimate the population mean of node variables in two layers. Basically, we choose two kinds of variables. One is a numerical variable, i.e. the degree of nodes; another is a categories variable, i.e. the a binary-valued node property. Specifically, our goal is to answer three questions: (1) can these crawling methods be used in sampling bipartite networks; (2) what factors in network structures and the parameters of sampling strategies affected the crawling method's performance; (3) last, which methods are more adequate under different scenarios. We finally provide references for the selection of sampling methods on different kinds of bipartite networks.

The tested topological structures include (1) degree distributions, (2) the number of nodes in two layers, (3) the degree correlation, and (4) community structure. For the aspect of sampling design, we evaluate the influence of: (1) the selection of initial nodes (the selection strategies for seeds and the number of seeds for SNS and RDS); and (2) the number of branches (the number of neighbors selected when implementing FFS, SNS and RDS). In simulations, we generate artificial networks and implement crawling methods to collect samples from each layer. We then analyze the effects of network structures and sampling design on the precision of estimation in two layers.

The rest of this paper is organized as follows. In section 2, we briefly detail the eight crawling methods evaluated in this study. In section 3, we introduce the bipartite network generation models which incorporate the structural parameters required for testing crawling methods. In section 4, we compare the simulation results for different topological structures and delineate the results derived by different sampling designs. In section 5, we conclude by reporting our findings and providing suggestions for selecting the appropriate crawling methods under different settings.

2. Crawling methods

Crawling methods are used when the global information is unknown, or it is more efficient to infer the network's global properties with a small subset of nodes and edges. In this section, we describe the crawling methods tested in this study. The basic sampling designs of these methods are shown in table 1. Their sampling processes are shown in figure 1.

2.1. Breadth first search (BFS)

BFS is a classic graph traversal algorithm in computer science. It starts with a single seed node and explores neighbors of visited nodes iteratively. At each iteration, the earliest explored but not-yet-visited node is selected [52]. When a node is visited,

J. Stat. Mech. (2018) 073403

information regarding the studied variables is collected. By the end of crawling, we calculate the sample mean as an indicator for the population estimates (see table 1).

2.2. Depth first search (DFS)

Similar to BFS, DFS also starts with a single seed node. However, it selects the latest explored but not-yet-visited node at each iteration [52]. And the sample mean is also used as the population estimates for the studied variables (see table 1).

2.3. Forest-fire sampling (FFS)

FFS is a probabilistic version of BFS [2]. A burning probability p decides whether a neighbor of the current node is explored. When $p = 1$, FFS is identical to BFS. In the sampling process, the studied variables of 'burned' node (sample) are selected. And the sample mean is used as the population estimates for the studied variables (see table 1).

2.4. Snowball sampling (SNS)

SNS is one of the best-known chain referral sampling methods in sociology and statistics research [14]. It starts with a set of seed nodes and selects a number of nodes randomly from all neighbors of each seed node. The selected nodes then become new seed nodes and iterate the same process of node selection. When all the neighbors are selected at each iteration, SNS works like BFS. Similarity, the sample mean is used as the population estimates for the studied variables in SNS (see table 1).

In summary, in the absence of global information, the above methods conduct the statistic inference by calculating the sample mean directly (see table 1), which introduces the bias toward high degrees.

2.5. Respondent driven sampling (RDS)

RDS has been widely used to study hidden populations [53]. It adopts the sampling process of SNS and the number of randomly selected neighbors is fixed (generally, three to five) due to the fact that the rejection rate of hidden populations is high in practical implementation of selecting neighbors [53]. In the sampling process, the studied variables and the degree of the sample node are collected. Compared with SNS, RDS can use the collected local information to conduct bias adjustment and statistical inference [54]. Differently from the other above-mentioned methods, RDS estimates the population mean though the re-weighted correction [28] instead of by calculating sample mean directly (see table 1).

2.6. Random walk (RW)

RW is a widely used method. It starts with one seed and selects the next node randomly from the current node's neighbors [13]. All nodes in the chain of RW are sampled. RW is typically implemented with replacement. As the inclusion probability of nodes is proportional to the node degree in RW, high degree nodes are overrepresented in samples. Consequently, RW samples are biased by the prevalence of high degree nodes.

J. Stat. Mech. (2018) 073403

Table 1. Basic sampling design of crawling methods.

Crawling methods	Seed number	Branching number	With/without replacement	Prior information	Population mean estimates of a variable y
BFS	1	1	WOR	—	$\sum_{i\in S} y_i/n_s$
DFS	1	1	WOR	—	$\sum_{i\in S} y_i/n_s$
FFS	1	Probability	WOR	—	$\sum_{i\in S} y_i/n_s$
RW	1	1	WR	—	$\sum_{i\in S} y_i/n_s$
MHRW	1	1	WR	Neighbors' degree	$\sum_{i\in S} y_i/n_s$
MDRW	1	1	WR	Max degree of nodes	$\sum_{i\in S} y_i/n_s$
SNS	Multiple	Multiple	WOR	—	$\sum_{i\in S} y_i/n_s$
RDS	Multiple	Multiple	WR	—	$\sum_{i\in S} k_i^{-1} y_i / \sum_{i\in S} k_i^{-1}$

y_i, the value of the variable y for node i; S, the sample set; n_s, the sample size; k_i, the degree of node i.

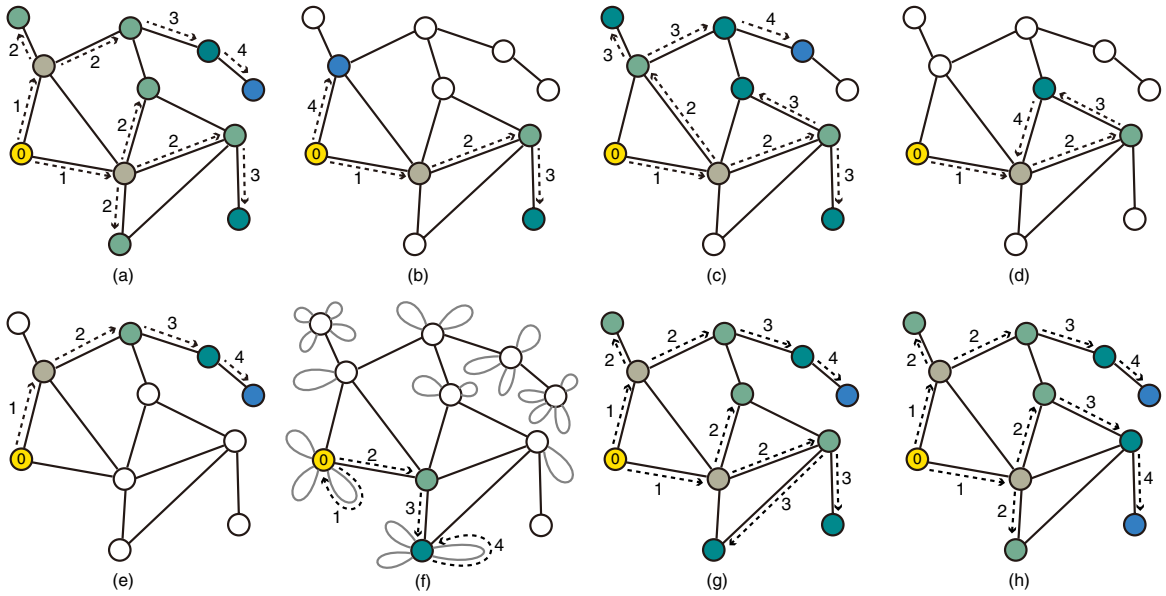

Figure 1. Sampling process of crawling methods. (a) BFS, (b) DFS, (c) FFS, (d) RW, (e) MHRW, (f) MDRW, (g) SNS and (h) RDS. The burning probability of FFS is 0.5. The number of initial seeds and branches is 1 and 2 for SNS and RDS. RDS is done without replacement.

For adjusting the bias of transitional random walk and using the sample mean to conduct the unbiased estimation, Metropolis–Hastings random walk and maximum-degree random walk have been provided.

2.7. Metropolis–Hastings random walk (MHRW)

MHRW is a random-walk based method which modifies the transition probabilities in the sampling process [11]. At each iteration, it randomly selects a neighbor j of the current node i and walks to the neighbor with probability $\min(1, k_i/k_j)$ where k_i and k_j are the degree of i and j. In this way, the walk towards a node with smaller degree is accepted, while some of the walks towards nodes with higher degree are rejected. Consequently, the method adjusts the bias towards high degree nodes during the sampling process. At each step, the degrees of the current node i and the randomly selected neighbor j are collected for calculating the transition probability. Then the studied variables of the sampled node is collected. MHRW also use the sample mean as an indicator for the population estimates (see table 1).

2.8. Maximum-degree random walk (MDRW)

MDRW assumes that a random walk has been performed on a modified regular network. This regular network is generated from the original network by adding a different number of self-loops on nodes so that the degree of each node ends up with the maximum degree of the original network [17]. Thus, at each iteration, it selects one neighbor of the current node i with the probability $k_i/k_{\max}$ and stays at the current node i with the probability $(k_{\max} - k_i)/k_{\max}$ where k_i is the degree of node i in the original network, and $k_{\max}$ is the maximum degree of the original network. In this way, the bias introduced by high degree nodes is adjusted during the sampling process. In the sampling process, the degree and the studied variables of current node are collected. And the sample mean is used as the population estimates for the studied variables (see table 1).

3. Generation of bipartite network

To investigate the effect of different structures of bipartite networks on estimations of node variables of two node layers, we generate artificial networks with different network structures. The basic statistics for a realization of each type of bipartite networks are shown in table 2.

3.1. Notation

Consider a bipartite network G in which nodes are divided in two disjoint partitions (layers), U and V; and E refers to the edges connecting the two layers, so that $G = (U, V, E)$ with $U \cap V$ and $E \subseteq U \otimes V$. The node degree sequences of U and V are denoted as $\mathrm{Seq}U = \{k_i^u | i = 1, 2, .., |U|\}$ and $\mathrm{Seq}V = \{k_j^v | j = 1, 2, ..., |V|\}$.

J. Stat. Mech. (2018) 073403

Table 2. Basic statistics for the realization of different types of bipartite networks.

Network	$\|U\|$	$\|V\|$	$\|E\|$	$\langle k_U \rangle$	$\langle k_V \rangle$	r	N_c
G_{poi}	10 000	10 000	120 000	12	12	−0.004	0
G_{pow}	10 000	10 000	120 000	12	12	−0.013	0
$G_{\text{diff\_size}}$	10 000	1000	120 000	12	120	−0.303	0
G_{assort}	10 000	10 000	120 000	12	12	0.2	0
$G_{\text{disassort}}$	10 000	10 000	120 000	12	12	−0.1	0
G_{commu}	10 000	10 000	120 000	12	12	−0.031	5

$|U|$ and $|V|$, the size of two layers; $|E|$, the number of all edges; $\langle k_U \rangle$ and $\langle k_V \rangle$, the average degree of U and V; r, the degree correlation between two layers; N_c, the number of communities.

A set of bipartite networks are then generated in accordance with the following structural parameters:

3.2. Degree distribution

We extend the configuration model [55, 56] to generate a bipartite network with two given degree distributions. Specifically, we first generate two layers of nodes, U and V. And each node is assigned a degree drawn from the given degree sequences Seq$U$ and Seq$V$, so that each node $i \subseteq U$ has k_i^u stubs and each node $j \subseteq V$ has k_j^v stubs. Lastly the stubs of U are randomly connected to the stubs of V. Note that $\sum_{i=1}^{|U|} k_i^u = \sum_{j=1}^{|V|} k_j^v$.

The degree distributions of two node layers typically follow power laws in real-world bipartite networks, such as those in scientific collaboration networks [57] and actor-movie networks [58]. So, we first generate two types of bipartite networks with different degree distributions for comparison: (1) G_{poi}: the degree distributions for U and V follow a Poisson distribution; (2) G_{pow}: the degree distribution for U and V follow a power-law distribution. The generation steps are as follows. We first generate a Barabási–Albert (BA) network [59] and a Erdős–Rényi (ER) network [60]. Both of them have 10 000 nodes and 120 000 edges. Then we extract their node degree sequences, defined as Seq$_{\text{BA}}$ and Seq$_{\text{ER}}$ respectively. Finally, we use two Seq$_{\text{BA}}$ sequences to generate G_{pow} with the aforementioned configuration model. In the same way, we use two Seq$_{\text{ER}}$ sequences to generate G_{poi}.

3.3. Number of nodes in each layer

The number of nodes for two layers in bipartite networks may be unequal, e.g. the number of user nodes is much larger than that of company nodes in labor-company networks [61] and the number of rater nodes is much larger than that of movie nodes in movie rating networks [62]. For comparison with G_{pow} in which two layers have the same number of nodes, we generate the bipartite network $G_{\text{diff\_size}}$ in which the size of U is ten times larger than V. The generation steps are as follows. We first generate a BA network which has 1000 nodes and 120 000 edges. Then we extract the node degree sequence Seq$'_{\text{BA}}$ from this BA network. After that, we use Seq$_{\text{BA}}$ and Seq$'_{\text{BA}}$ to generate $G_{\text{diff\_size}}$ by the aforementioned configuration model.

J. Stat. Mech. (2018) 073403

3.4. Degree correlations

Degree correlation describes the tendency of nodes to connect preferentially to other nodes with either similar or opposite degree values. Networks in which nodes tend to be connected with similar degree values show assortative mixing, otherwise they show disassortative mixing. The degree correlation structures have been observed in many real-world bipartite networks [44]. We use the Pearson correlation coefficient r to quantify the tendency [63, 64]:

$$r = \frac{|E|^{-1}\sum_i d_i k_i - [|E|^{-1}\sum_i \frac{1}{2}(d_i + k_i)]^2}{|E|^{-1}\sum_i \frac{1}{2}(d_i^2 + k_i^2) - [|E|^{-1}\sum_i \frac{1}{2}(d_i + k_i)]^2}, \tag{1}$$

where $|E|$ is the number of edges in the network, and d_i and k_i are the degrees of nodes at the end of the ith edge, $i = 1, 2, ..., |E|$. The correlation coefficient r lies between -1 and 1. When $r > 0$ the network shows assortative mixing patterns, when $r = 0$ the network shows no degree correlations, and when $r < 0$ the network is dissassortative.

To generate bipartite networks with varying degree correlations, we use an edge rewiring operation. Specifically, given a bipartite network $G = (U, V, E)$, we first randomly pick a pair of edges, $e_i = (u_i, v_i)$ and $e_j = (u_j, v_j)$ where $u_i, u_j \subseteq U$ and $v_i, v_j \subseteq V$. Then we rewire two edges as $e_i' = (u_i, v_j)$ and $e_j' = (u_j, v_i)$ and recalculate degree correlation (r') of the network. If the new edges do not exist, and r' is approaching the desired value, this rewiring operation will be kept; otherwise the operation is rolled back, and a pair of edges are reselected. The above processes are repeated until r' reaches the desired value. Note that r' is bound to an upper (assortative) and lower (disassortative) limit due to the fact that edges in bipartite networks only connect nodes across the two node layers. Here we find the upper and lower bound are about -0.150 and 0.295 in edge rewiring simulations.

According to the above approach, we take G_{pow}, whose degree correlation coefficient almost equals to 0, as the baseline and generate two types of bipartite networks with degree correlation structures: G_{assort} with $r = 0.2$ and $G_{\text{disassort}}$ with $r = -0.1$.

3.5. Community structures

A network has community structure if its nodes tend to gather into groups such that each group of nodes is densely connected internally [65]. Empirical studies show that community structures exist in many bipartite networks [49].

In this study, we extend Girvan and Neman's (GN) benchmark model [65, 66] to generate bipartite networks with community structure by using two given degree sequences. The main steps are shown in figure 2. In the first step, two layers of nodes, U and V, are generated, and each node is assigned a degree drawn from the given degree sequences Seq$U$ and Seq$V$. No nodes are assigned to any community at this stage. We assume that the number of communities is N_c and each community has the same size in the bipartite network to be generated, so that the size of each community s is $(|U| + |V|)/N_c$. In the next step, a mixing parameter α is introduced [66]: each node i with degree k_i shares a fraction $1 - \alpha$ of its edges with the other nodes of the same community, and a fraction α with the other nodes of the networks, i.e. the internal degree (the number of its edges inside the community) $k_i^{(\text{in})} = (1 - \alpha)k_i$ and external degree

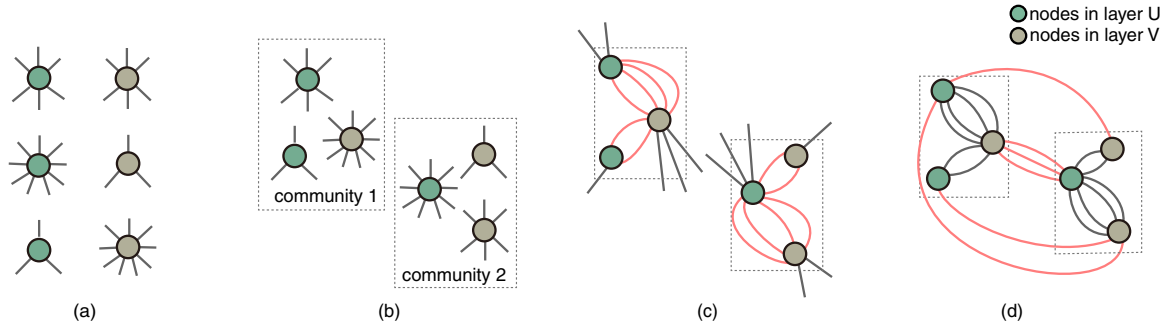

Figure 2. The basic steps for the generation of bipartite network with community structure. (a) Generate two layers of nodes. (b) Assign nodes to different communities. (c) Connect nodes with internal edges. (d) Connect communities with external edges. In this figure, the mixing parameter α is 0.3 and the number of communities is 2.

(the number of its edges outside the community) $k_i^{(\mathrm{out})} = \alpha k_i$. Therefore, a bipartite network with stronger community structures is generated by a small α.

Then we assign all nodes to the communities. In the first iteration, we randomly select a node i from $U \cup V$ and assign it to a community ξ; if the internal degree of node i does not exceed the community size, the node is assigned to ξ, otherwise it remains solitary. In successive iterations, we continue to assign solitary nodes to ξ; if the community is assigned to s nodes and the sum of the internal degree of nodes from U equals that of nodes from V in ξ, we stop the procedure. Otherwise, a node randomly selected from the community will be taken out and become solitary. The remaining solitary nodes will also be assigned to the rest of communities by the same procedure.

After each node has been assigned to a community, we connect all nodes to generate the whole network. Inside each community, we use the internal edges of nodes to connect them by the aforementioned configuration model. Then we consider each community as a component with stubs representing the external edges of the community and randomly connect its stubs to those of other communities under the condition that both ends of a connection are from different node layers.

According to the above steps, we take G_{pow}, which is of no community structures, as a baseline, and use its degree sequences of two layers to generate the bipartite network G_{commu} with 5 communities. The mixing parameter α is 0.1.

3.6. Node property assignment

For each generated network, we assign each node with a binary-valued property. The rule is as follows. In layer U, we select 60% of nodes with the probability proportional to their degree, i.e. the node with high degree has higher probability to be selected. Then the selected nodes are assigned with property A and the remaining nodes are assigned with property B. In such a property list, the property values of the nodes with property A are 1 and those with property B are set at 0. The proportion of nodes in layer U with property A is then 0.6. Similarity, we assign and set the proportion of nodes with property A in layer V to be 0.4.

4. Simulation settings

In each simulation, we first generate a bipartite network according to the specific purpose. Then we randomly select the initial seed nodes and implement a crawling method on the generated bipartite network to collect samples (less than 10%) from each layer. After collecting samples, we analyze the effect of network structure and sampling design on the estimation of the population means of two representative variables in two layers, one is numerical variable, i.e. the degree of nodes. Another is a categories variable, i.e. the binary-valued node property. The population means of these two variables characterize two important characteristics of networks: the average degree of network $\langle k\rangle$ and the proportion of nodes with one property (property A in this paper) $P(A)$.

The basic settings are as follows. The burning probability p of FFS is 0.5, the number of initial seeds and the number of branches for SNS and RDS are 5 and 3 respectively. All simulations are repeated 100 times, and results are averaged over 100 simulations.

In this paper, we use relative error (RE) to measure the performance of crawling methods on the estimate of variables of two layers. The relative error of the population mean of a variable y for U, RE_U, is

$$\mathrm{RE}_U = \frac{|\langle \widehat{y}_U\rangle - \langle y_U\rangle|}{\langle y_U\rangle}, \tag{2}$$

where $\langle \widehat{y}_U\rangle$ is the population mean of variable y for U calculated from the estimator of different methods, and $\langle y_U\rangle$ is the true population mean of y for U. The relative error of the population mean of y for V, RE_V, can also be calculated in the same way.

5. Results

5.1. Effect of network structure

In this section, we focus on whether the performance of a sampling method is affected by the network structures of bipartite networks, i.e. degree distribution, number of nodes in two layers, degree correlation and community structure.

5.1.1. Degree distribution. First, the effect of the degree distribution on the performance of a crawling method is investigated. Figure 3 shows the relative error RE_U and RE_V for the implementation of different methods on bipartite networks with different degree distributions of two layers. It reveals that, for MHRW, MDRW and RDS, both RE_U and RE_V vary little in these networks and are almost equal to 0, for estimating either the average degree ($\langle k\rangle$) or the proportion of nodes with property A ($P(A)$), which is to say, the performance of these methods is not affected by the degree distribution.

However, the performance of other crawling methods, including BFS, DFS, FFS, RW and SNS, is affected by the degree distribution. When the degree distribution follows a power-law, the relative error of the population mean of variables is much larger than that of Poisson distribution. Even the relative error decreases with sample size

J. Stat. Mech. (2018) 073403

in all these methods except RW, the difference remains larger than 1 for $\langle k \rangle$ and large than 0.16 for $P(A)$, when the sample proportion reaches 10%. The varying performance can be explained by the sample representativeness obtained by these different crawling methods: BFS, DFS, FFS, RW, and SNS all tend to oversample high degree nodes in the network with power-law degree distribution of layers. But for MHRW, MDRW and RDS, correction mechanisms effectively reduce the bias introduced by over-representativeness of high degree nodes.

5.1.2. Unequal number of nodes in two layers. Secondly, we focus on whether an unequal number of nodes in the two layers of bipartite networks affects the performance of crawling methods. Figure 4 shows the relative error RE_U and RE_V for the implementation of crawling methods on bipartite networks with an unequal number of nodes in two layers. We can see that both RE_U and RE_V vary little for MDRW and RDS. However, for MHRW, the RE_U for $\langle k \rangle$ and $P(A)$ varies significantly in these networks, although the variation of RE_V is little: the difference in RE_U between G_{pow} and $G_{\mathrm{diff\_size}}$ (the size of V is 1/10 of that in G_{pow}) for $\langle k \rangle$ is 1.02, and that for $P(A)$ is 0.20, when sampling ratio reaches 10%. The result indicates that its performance in U is strongly affected by the unequal number of nodes in two layers. The reason for the results is that the correction mechanism of MHRW becomes invalid in U: the walks from V to U cannot select nodes with a smaller degree in U when the average degree of V is much larger than U.

For other crawling methods, including BFS, DFS, FFS, RW and SNS, the difference in RE_U among these networks decreases to almost 0 when the sample proportion exceeds 5%, thus indicating that their performance in U is not influenced by the unequal number of nodes in layers. However, their performance in V is affected. Although the difference in RE_V among these networks decreases with sample size, it also remains large when the sample proportion reaches 10%. For $\langle k \rangle$ of the layer, the difference between G_{pow} and $G_{\mathrm{diff\_size}}$ is 0.75 for BFS, 0.69 for DFS, 0.66 for FFS, 1.4 for RW and 0.63 for SNS. For $P(A)$, the difference between G_{pow} and $G_{\mathrm{diff\_size}}$ is 0.21 for BFS, 0.13 for DFS, 0.17 for FFS, 0.24 for RW, and 0.19 for SNS.

5.1.3. Degree correlation. In addition, we discuss the influence of degree correlation on the performance of a sampling method. Figure 5 shows the relative error RE_U and RE_V for the implementation of crawling methods on bipartite networks with different degree correlations. First of all, we can see that the RE_U and RE_V for DFS, RW, MHRW and MDRW vary little with the change of degree correlation, for both $\langle k \rangle$ and $P(A)$, revealing that these methods are virtually not affected by the degree correlation.

Second, the RE_U and RE_V for SNS are large when the sampling proportion is small, and they decrease with the sample proportion. Both for $\langle k \rangle$ and $P(A)$, these relative errors vary slightly between the network with $r = 0$ and the network with $r = 0.2$. But such variations are not obvious between the network with $r = 0$ and the network with $r = -0.1$. Third, RDS has constant relative errors which do not change with the sample proportion. But the relative errors of the network with $r = 0.2$ are slightly larger than those of the network with $r = 0$. Similar to SNS, the relative errors show little difference between the network with $r = 0$ and the network with $r = -0.1$. These results indicate that SNS and RDS are slightly affected by degree correlation.

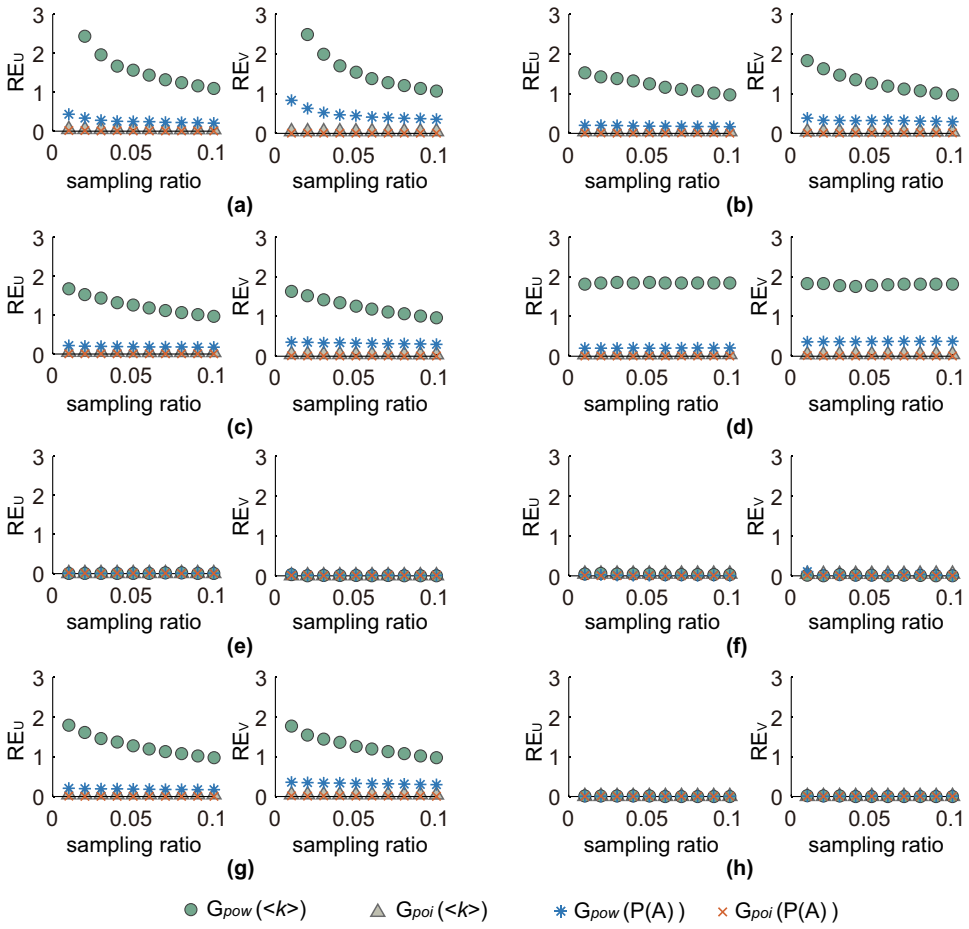

Figure 3. Relative error RE_U and RE_V obtained from (a) BFS, (b) DFS, (c) FFS, (d) RW, (e) MHRW, (f) MDRW, (g) SNS, (h) RDS in G_{poi} and G_{pow} (i.e. bipartite networks with different degree distributions of two layers).

Lastly, we can see that RE_U and RE_V for BFS and FFS both vary greatly in different networks. With the increasing of the degree correlation (from -0.1 to 0.2), the relative error of both layers for the two methods increases substantially: when sampling ratio is 10%, for BFS, the relative error of $\langle k\rangle$ increases from 0.30 to 2.23 in layer U and from 0.41 to 2.11 in layer V, and that of $P(A)$ increases from 0.02 to 0.30 in layer U and from 0.06 to 0.53 in layer V; for FFS, that of $\langle k\rangle$ increases from 0.61 to 1.75 in layer U and from 0.60 to 1.74 in layer V, and that of $P(A)$ increases from 0.07 to 0.60 in layer U and from 0.13 to 0.63 in layer V. That is to say, BFS and FFS are seriously influenced by degree correlation.

5.1.4. Community structures. Finally, we focus on whether the performance of a sampling method is affected by community structure. The results are shown in figure 6. We can see that the RE_U and RE_V for BFS, DFS, FFS and SNS are large when the sampling proportion is small, and they decrease with the sampling proportion. Among them, for BFS and FFS, the difference in RE_U and RE_V between networks with and without community structures does not decrease with the sampling proportion. When the sampling proportion reaches 10%, the difference remains larger than 0.25 for $\langle k\rangle$

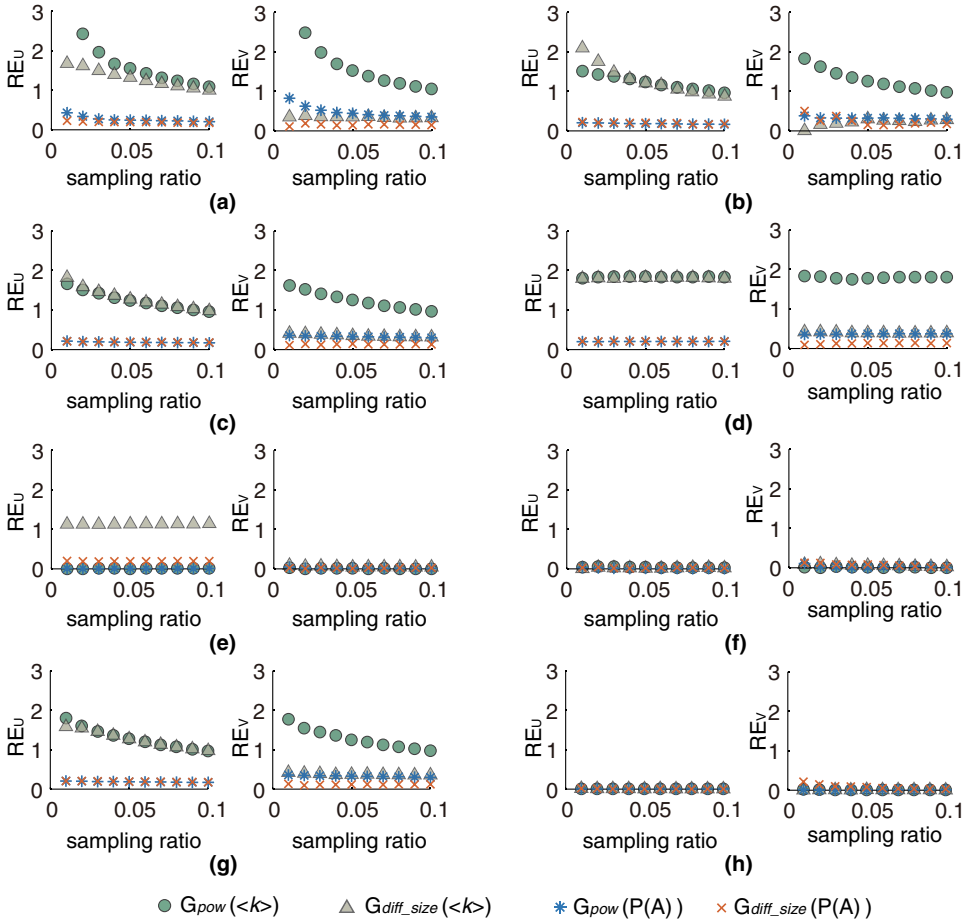

Figure 4. Relative error RE_U and RE_V obtained from (a) BFS, (b) DFS, (c) FFS, (d) RW, (e) MHRW, (f) MDRW, (g) SNS, (h) RDS in G_{pow} and $G_{\mathrm{diff\_size}}$ (i.e. bipartite networks with unequal number of nodes in two layers).

and larger than 0.10 for $P(A)$. Such differences for DFS and SNS also exist but are very small. Other methods, including RW, MDRW, MHRW and RDS, have constant relative errors which do not change with the sample proportion. For these methods, the difference in RE_U and RE_V between networks with and without community structures is small and almost equal to 0.

These results indicate that, among the eight crawling methods, only BFS and FFS are affected by the strong community structure. As the chains of samples (number of recruitment waves) can grow relatively long in MHRW, MDRW and RDS, the crawling can easily break the communities such that the representativeness of the sample can be retained.

5.2. Effect of the sampling design

In this section, we focus on whether the performance of a sampling method is affected by the different settings of sampling design on bipartite networks. Such settings include the selection approach for seeds, the number of seeds for SNS and RDS, and the branching number for FFS, SNS and RDS.

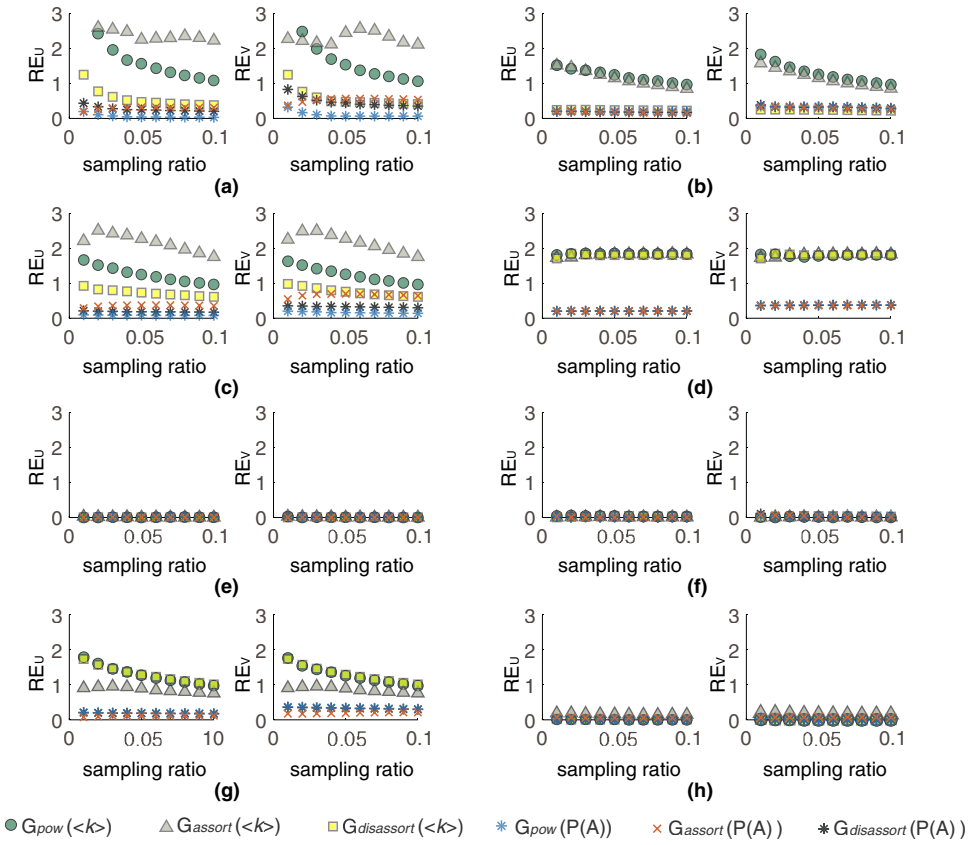

Figure 5. Relative error RE_U and RE_V obtained from (a) BFS, (b) DFS, (c) FFS, (d) RW, (e) MHRW, (f) MDRW, (g) SNS, (h) RDS in G_{pow}, G_{assort} and $G_{\mathrm{disassort}}$ (i.e. bipartite networks with different degree correlations).

5.2.1. Selection approach for seed nodes. First, we compare two kinds of selection approaches for seed nodes. The first approach selects seed nodes uniformly from the bipartite network. The second approach selects seed nodes with probability proportional to their degree, insofar as nodes with more connections are more likely to be selected as seeds. Figure 7 shows the RE_U and RE_V when sampling methods select node seeds by different approaches in G_{pow}. We can see that, for all these crawling methods, the difference in both RE_U and RE_V for $\langle k\rangle$ and $P(A)$ is little, regardless of the different selection approaches used. The result indicates that these crawling methods are not affected by the selection approaches for seed nodes.

These results are similar to what was found in the one-mode network [54, 67], i.e. the dependence of subsequent nodes on seed nodes will be weak with the growth of the crawling chain.

5.2.2. Number of seed nodes and branches. Subsequently, we evaluate the influence of the number of seed nodes on the performance of SNS and RDS by fixing the number of branches at 3 and selecting all seed nodes uniformly. Simulation results are presented in figures 8(a) and (b). We can see that when the number of seeds increases from 3 to 10, both RE_U and RE_V for $\langle k\rangle$ and $P(A)$ are almost identical, indicating that the performance of SNS and RDS is not affected by the number of seed nodes.

https://doi.org/10.1088/1742-5468/aace0f

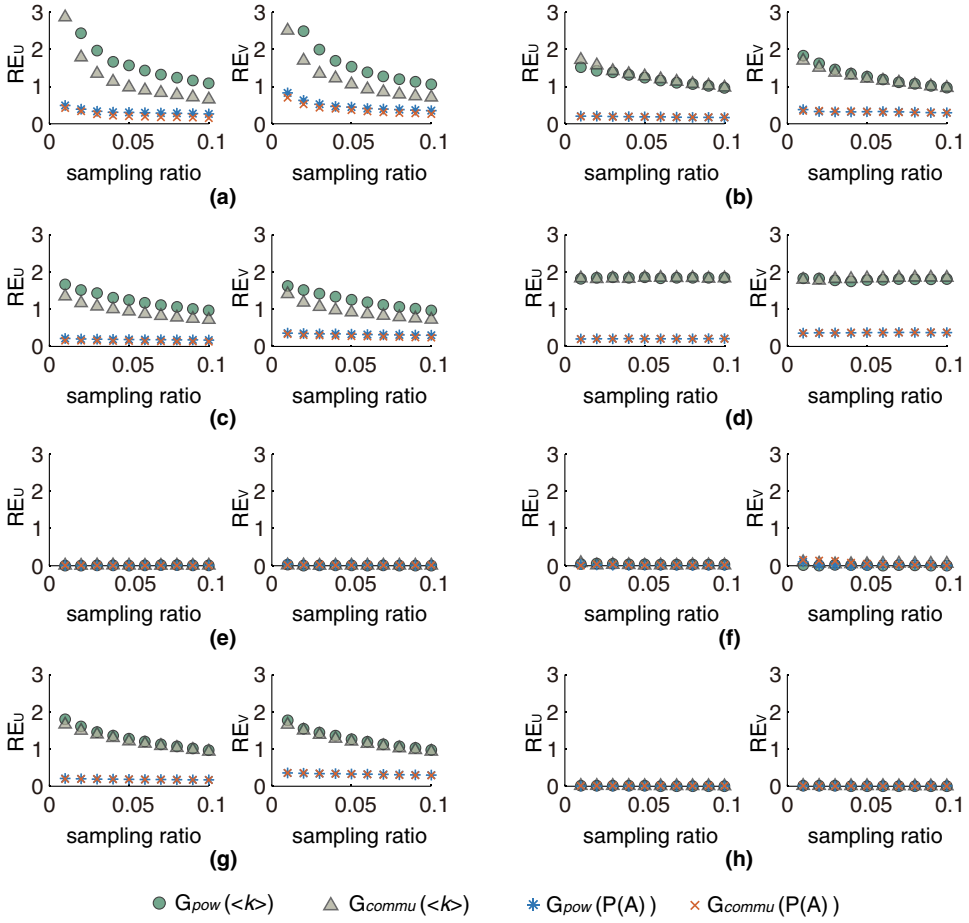

Figure 6. Relative error RE_U and RE_V obtained from (a) BFS, (b) DFS, (c) FFS, (d) RW, (e) MHRW, (f) MDRW, (g) SNS, (h) RDS in G_{pow} and G_{commu} (i.e. bipartite networks with community structures).

For the implementation of FFS, SNS and RDS, multiple branches are typically used, as the response rate in practice may be low. With this mind, we then explore the performance of FFS, SNS and RDS with different numbers of branches. In simulations, the seed nodes are selected uniformly and the number is set at 5. Results are shown in figures 8(c)–(e). Again, we can see that there is no visible difference in RE_U and RE_V for all these methods when the number of branches increases from 3 to 10 (SNS and RDS) and the burning probability increases from 0.3 from 0.9 (FFS).

The above results are similar to what has been found in analyses of one-mode networks [54, 67], i.e. these crawling methods are not sensitive to the number of seeds and branches due to the fact that the number of seeds or branches does not change the inclusion probability of nodes. It is worth noting that even if there is no significant influence from seed number and branching number, in practice the number of seeds and branches should be set adequately, so as to avoid sampling chains stopping growing as the response rate may be low.

J. Stat. Mech. (2018) 073403

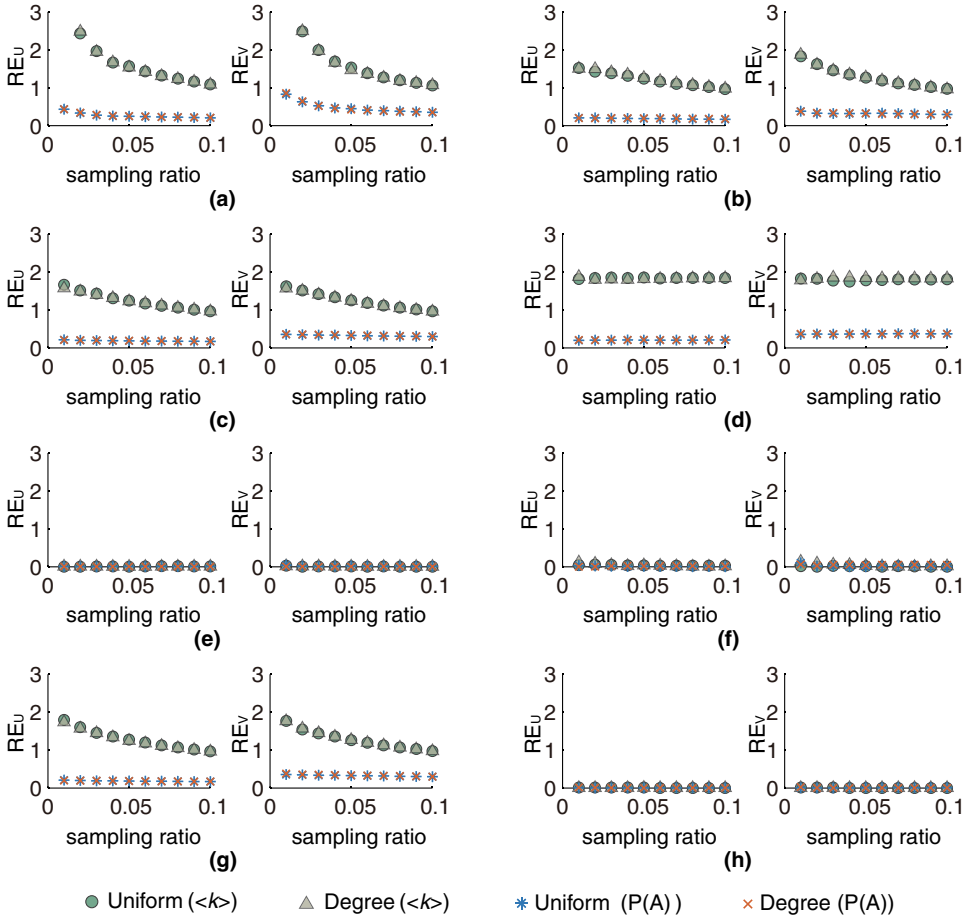

Figure 7. Relative error RE_U and RE_V obtained from (a) BFS, (b) DFS, (c) FFS, (d) RW, (e) MHRW, (f) MDRW, (g) SNS, (h) RDS in G_{pow} when seeds are selected uniformly or with probability proportional to degree.

6. Synthesis comparison

After investigating the effect of different factors on each crawling method, in this section, we conduct a synthesis comparison of eight crawling methods on four network structures. Base settings of simulations are adopted, and the sampling proportion is set at 10%.

In figure 9, we show the comparison of relative error obtained by these crawling methods on four types of networks: a network with power-law degree distributions (G_{pow}), a network with extremely unequal number of nodes in U and V ($G_{\text{diff\_size}}$), a network with positive degree correlation (G_{assort}) and a network with strong community structures (G_{commu}).

First of all, we discover that, due to the over-representativeness of high degree nodes, RW generates the large bias in these networks. The relative errors of both layers are about 2 for $\langle k\rangle$ and all over 0.2 for $P(A)$ in G_{pow}, G_{assort} and G_{commu}, and are larger than other methods in $G_{\text{diff\_size}}$.

Besides RW, crawling methods, including BFS, DFS, FFS and SNS, also show large bias: for $\langle k\rangle$, their relative errors in G_{pow} and G_{commu} are all near 1. And the relative

J. Stat. Mech. (2018) 073403

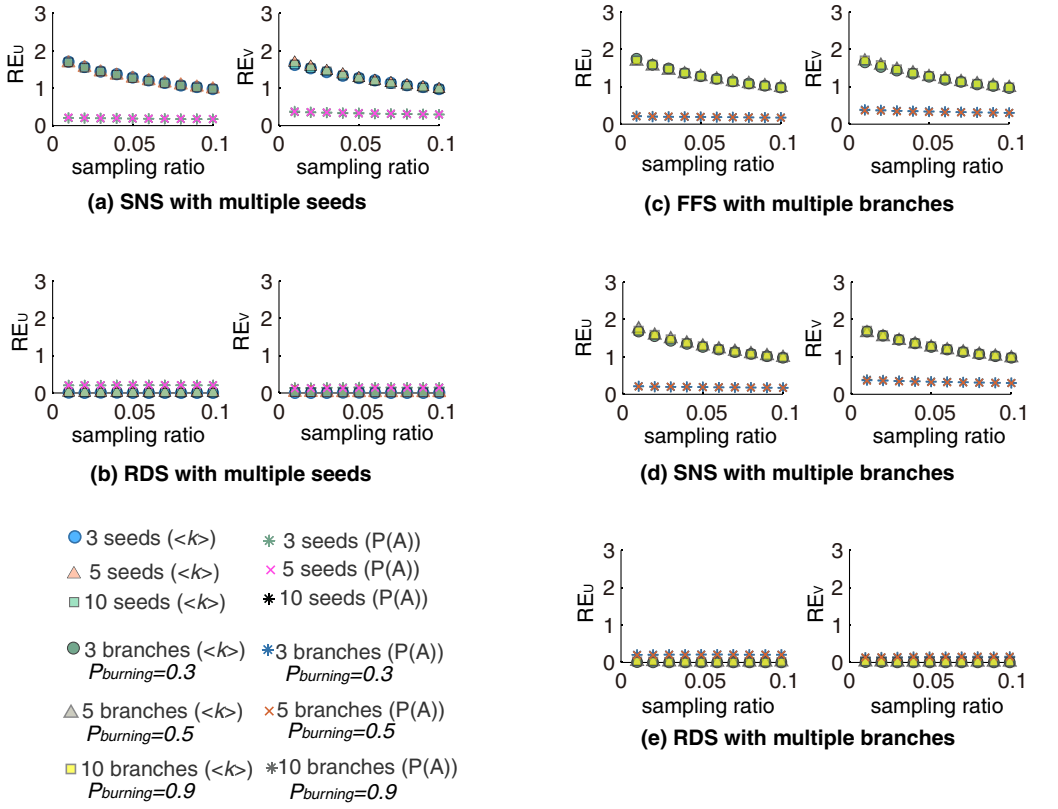

Figure 8. Relative error RE_U and RE_V obtained from (a) SNS, (b) RDS with multiple seeds and (c) FFS, (d) SNS, (e) RDS with multiple braches in G_{pow}.

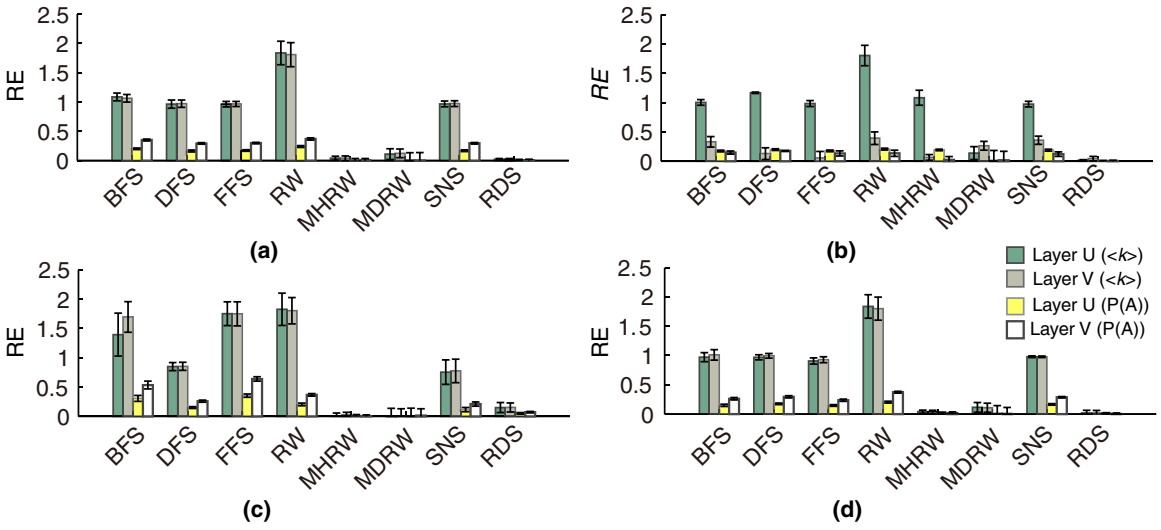

Figure 9. Relative errors in U and V obtained from BFS, DFS, FFS, RW, MHRW, MDRW, SNS, RDS in (a) G_{pow}, (b) $G_{\mathrm{diff\_size}}$, (c) G_{assort} and (d) G_{commu}. Sampling ratio for each layer is 0.1.

J. Stat. Mech. (2018) 073403

Table 3. Summary of the performance of eight crawling methods.

Factors	BFS	DFS	FFS	RW	MHRW	MDRW	SNS	RDS
Degree distribution	×	×	×	×	°	°	×	°
Unequal size	×	×	×	×	×	°	×	°
Assortativity	×	°	×	°	°	°	×	°
Community structure	×	×	×	°	°	°	°	°
Seed distribution	°	°	°	°	°	°	°	°
Seed number	—	—	—	—	—	—	°	°
Branching number	—	—	°	—	—	—	°	°

×, affected; °, not affected; —, not applicable.

Table 4. The selection of crawling methods under different settings for generating a population mean.

	BFS	DFS	FFS	RW	MHRW	MDRW	SNS	RDS
Power-law degree distribution					°	°		•
Unequal node size in layers						°		•
Strong assortativity					°	°		•
Strong community structure					°	°		•

°, recommended with prior information; •, recommended without prior information.

errors of BFS and FFS are larger than 1.7 in G_{assort}. Although these methods show relatively small bias in layer V of $G_{\text{diff\_size}}$, the bias in layer U is large: the relative errors all exceed 1. For $P(A)$, their relative errors in G_{pow}, $G_{\text{diff\_size}}$ and G_{commu} are all over 0.15. And the relative errors of BFS and FFS are larger than 0.3 in G_{assort}.

Third, the results reveal that MDRW, MHRW and RDS have the least bias in all types of network structures. Specifically, MDRW shows good performance in all these network structures; its relative errors for $\langle k\rangle$ and $P(A)$ are all less than 0.1 in each of the four types of networks. MHRW is almost unbiased in G_{assort} but generates large bias in layer U of $G_{\text{diff\_size}}$. RDS outperforms MHRW and MDRW in G_{pow}, $G_{\text{diff\_size}}$ and G_{commu}; its relative errors are all less than 0.05 in these networks. However, the bias of RDS for both $\langle k\rangle$ and $P(A)$ is slightly larger than MDRW and MHRW in G_{assort}. The better performance of the three methods is owing to their correction mechanisms for the inclusion probability of nodes. And it is worth noting that effective use of MDRW and MHRW requires some prior information, i.e. the maximum degree of a network and the degree of neighbors.

J. Stat. Mech. (2018) 073403

7. Conclusion and discussion

In this paper, we evaluate eight crawling methods on bipartite networks and compare their performance on the estimation of two network variables (the average degree of networks and the proportion of nodes with property A) with sample data under a variety of conditions.

Results reveal that network structures, including degree distribution, number of nodes in layers, degree correlation and communities, have varying effect on these crawling methods. We summarize these results in table 3. While BFS and FFS are significantly affected by all four network structures, DFS, RW and SNS are affected by degree distribution and the unequal number of nodes in layers. Community structure has a slight effect on DFS and assortativity has an effect on SNS. The methods, with the ability to adjust the inclusion probability of nodes, perform much better than the others. MHRW is only affected by the unequal number of nodes in layers, RDS is slightly affected by assortativity. All these variances in network structures have almost no effect on MDRW.

Compared to the effect of the network structures, the settings of the sampling design have almost no influence on the performance of crawling methods (see table 3), which is due to the fact that the long sampling chain of crawling methods reduces the influence of initial seeds and branching number. Therefore, when response rate is low, increasing the number of seeds and branches properly is generally the best approach to enhancing efficiency of sample collection.

To summarize, for sampling and inferring in bipartite networks, RDS outperforms other methods, and its estimates of network variables for U and V have the least bias in almost all scenarios, including networks with power-law degree distribution, extremely unequal number of nodes in layers and strong community structure. When the network is assortative, RDS is still among the best three sampling methods, together with MHRW and MDRW.

In all cases, methods with statistical approaches to adjust the inclusion probability of nodes have substantially improved performance. MHRW and MDRW adjust the inclusion probability of nodes during the sampling process by using some prior information, i.e. the degree of neighbors for MHRW and the maximum degree of the network for MDRW. After collecting samples, RDS adjusts the inclusion probability by means of the re-weighting strategy, which depends on the well-known Hanse–Hurwitz estimator. However, these adjustment approaches may not be effective and feasible. The adjustment mechanism of MHRW is invalid in the network with the unequal number of nodes in two layers, while the degree of neighbors and the maximum degree of networks are difficult to obtain in practical implementations [11, 68]. In addition, the adjustment approaches during the sampling process have drawbacks, like a high rejection rate for nodes [69] and the tendency to collect too many repeated samples [70]. By comparison, RDS has many advantages, e.g. the adjustment strategy after the sampling process which does not need prior information makes RDS more flexible in practice. In addition, as RDS does not reject nodes during the sampling process, its collection of samples is more efficient.

This paper provides a comprehensive investigation of implementing eight crawling methods on bipartite networks; this has not been examined in the literature. We endeavor to identify factors that critically affect the performance of crawling methods.

We concluded that particular crawling methods prove consistently more effective in different sampling scenarios. The methods for networks with different structures that we advocate, based on repeated testing, are summarized in table 4. In general, we recommend the robustness of RDS as an effective instrument in many different scenarios. If the prior information is available, MHRW and MDRW are also efficient, reliable methods for sampling bipartite systems.

Acknowledgment

XL acknowledges the Natural Science Foundation of China under Grant Nos. 71771213, 71522014 and 71790615. SC was partially supported by the Natural Science Foundation of China under Grant Nos. 91546203, 71731009 and 71725001.

References

[1] Chiang W C, Lin H H, Huang C S, Lo L J and Wan S Y 2005 *Proc. Natl Acad. Sci. USA* **102** 4221
[2] Leskovec J and Faloutsos C 2006 Sampling from large graphs *KDD '06 Proc. of the 12th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining* pp 631–6
[3] Rejaie R, Torkjazi M, Valafar M and Willinger W 2010 *Netw. IEEE* **24** 32–7
[4] Sang H L, Kim P J and Jeong H 2006 *Phys. Rev.* E **73** 016102
[5] Krishnamurthy V, Faloutsos M, Chrobak M, Lao L, Cui J H and Percus A G 2005 Reducing large internet topologies for faster simulations *IFIP-TC6 Int. Conf. on NETWORKING Technologies, Services, and Protocols; Performance of Computer and Communication Networks; Mobile and Wireless Communication Systems* pp 328–41
[6] Salehi M, Rabiee H R and Rajabi A 2012 *Chaos* **22** 2202–29
[7] Lu X 2013 *Respondent-Driven Sampling Theory, Limitations & Improvements* (Stockholm: Karolinska Institutet)
[8] Ribeiro B, Wang P, Murail F and Towsleyl D 2012 *Proc. IEEE INFOCOM* vol **131** pp 1692–700
[9] Ribeiro B, Gauvin W, Liu B and Towsley D 2010 On MySpace account spans and double pareto-like distribution of friends *INFOCOM IEEE Conf. on Computer Communications Workshops* pp 1–6
[10] Maiya A S and Berger-Wolf T Y 2011 Benefits of bias:towards better characterization of network sampling *KDD '11 Proc. of the 17th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining* (San Diego, CA: ACM) pp 105–13
[11] Gjoka M, Kurant M, Butts C T and Markopoulou A 2010 *Proc. IEEE INFOCOM* (San Diego, CA) pp 1–9
[12] Cormen T H, Leiserson C E, Rivest R L and Stein C 2012 1297–305
[13] Lovasz L 1993 *Lecture Notes Math.* **8** 285–303
[14] Biernacki P and Dan W 1981 *Sociol. Methods Res.* **10** 141–63
[15] Heckathorn D D 1997 *Soc. Problems* **44** 174–99
[16] Hastings W K 1970 *Biometrika* **57** 97–109
[17] Bar-Yossef Z, Berg A, Chien S, Fakcharoenphol J and Weitz D 2000 Approximating aggregate queries about web pages via random walks *Int. Conf. on Very Large Data Bases* pp 535–44
[18] Kurant M, Markopoulou A and Thiran P 2011 *IEEE J. Sel. Areas Commun.* **29** 1799–809
[19] Avrachenkov K, Borkar V S, Kadavankandy A and Sreedharan J K 2016 *Comparison of Random Walk Based Techniques for Estimating Network Averages* (Berlin: Springer)
[20] Ribeiro B and Towsley D 2010 Estimating and sampling graphs with multidimensional random walks *Proc. of the 10th ACM SIGCOMM Conf. on Internet Measurement* pp 390–403
[21] Lu X 2013 *Soc. Netw.* **35** 669–85
[22] Rezvanian A, Rahmati M and Meybodi M R 2014 *Physica* A **396** 224–34
[23] Rezvanian A and Meybodi M R 2015 *Physica* A **424** 254–68
[24] Jalali Z S, Rezvanian A and Meybodi M R 2016 *Int. J. Mod. Phys.* C **27**
[25] Salehi M, Rabiee H R, Nabavi N and Pooya S 2012 Characterizing Twitter with respondent-driven sampling *IEEE 9th Int. Conf. on Dependable, Autonomic and Secure Computing* pp 1211–7
[26] Stutzbach D, Rejaie R, Duffield N, Sen S and Willinger W 2009 *IEEE/ACM Trans. Netw.* **17** 377–90

J. Stat. Mech. (2018) 073403

[27] Stutzbach D, Rejaie R, Duffield N and Sen S 2006 Sampling techniques for large, dynamic graphs *IEEE INFOCOM IEEE International Conference on Computer Communications* pp 1–6
[28] Bell D C, Erbaugh E B, Serrano T, Daytonshotts C A and Montoya I D 2017 *Soc. Sci. Res.* **62** 350–61
[29] Salganik M J and Heckathorn D D 2017 *Sociol. Methodol.* **34** 193–239
[30] Chen S and Lu X 2017 *Sci. Rep.* **7** 3268
[31] Guimera R, Uzzi B, Spiro J and Amaral L A N 2005 *Science* **308** 697–702
[32] Schilling M A and Phelps C C 2007 *Manage. Sci.* **53** 1113–26
[33] Barabâsi A L, Jeong H, Néda Z, Ravasz E, Schubert A and Vicsek T 2002 *Physica* A **311** 590–614
[34] Börner K, Maru J T and Goldstone R L 2004 *Proc. Natl Acad. Sci.* **101** 5266–73
[35] Uetz P *et al* 2000 *Nature* **403** 623–7
[36] Li S *et al* 2004 *Science* **303** 540–3
[37] Medo M, Mariani M S, Zeng A and Zhang Y C 2016 *Sci. Rep.* **6**
[38] Ren Z M, Kong Y, Shang M S and Zhang Y C 2016 *Phys. Lett.* A **380** 2608–14
[39] Zhang P, Wang D and Xiao J 2017 *Physica* A **471** 147–53
[40] Hu X, Mai Z, Zhang H, Xue Y, Zhou W and Chen X 2016 A hybrid recommendation model based on weighted bipartite graph and collaborative filtering *IEEE/WIC/ACM Int. Conf. on Web Intelligence Workshops* (Piscataway, NJ: IEEE) pp 119–22
[41] Zhou T 2012 *Internet Res.* **21** 67–81
[42] Wu L, Zhang J and Zhao M 2014 *PLoS One* **9** e102646
[43] Estrada E and Rodríguez-Velázquez J A 2005 *Phys. Rev.* E **72** 046105
[44] Peltomäki M and Alava M 2006 *J. Stat. Mech.* P01010
[45] Zhou T, Ren J, Medo M and Zhang Y C 2007 *Phys. Rev.* E **76** 046115
[46] Qiao J, Meng Y Y, Chen H, Huang H Q and Li G Y 2016 *Physica* A **457** 270–9
[47] Ohkubo J, Tanaka K and Horiguchi T 2005 *Phys. Rev.* E **72** 036120
[48] Saavedra S, Reed-Tsochas F and Uzzi B 2009 *Nature* **457** 463–6
[49] Zhang P, Wang J, Li X, Li M, Di Z and Fan Y 2008 *Physica* A **387** 6869–75
[50] Kheirkhahzadeh M, Lancichinetti A and Rosvall M 2016 *Phys. Rev.* E **93** 032309
[51] Miklós I, Erdös P L and Soukup L 2010 *Electron. J. Comb.* **20** 229–40
[52] Cormen T H, Leiserson C E, Rivest R L and Stein C 2009 *Introduction to Algorithms* 3rd edn (Cambridge, MA: The MIT Press)
[53] Goel S and Salganik M J 2010 *Proc. Natl Acad. Sci.* **107** 6743–7
[54] Lu X, Bengtsson L, Britton T, Camitz M, Kim B J, Thorson A and Liljeros F 2012 *J. R. Stat. Soc.* **175** 191–216
[55] Molloy M and Reed B 1998 *Comb. Probab. Comput.* **7** 295–305
[56] Newman M E, Watts D J and Strogatz S H 2002 *Proc. Natl Acad. Sci.* **99** 2566–72
[57] Newman M E 2000 *Proc. Natl Acad. Sci. USA* **98** 404–9
[58] Latapy M, Magnien C and Vecchio N D 2008 *Soc. Netw.* **30** 31–48
[59] Albert R and Barabási A L 2002 *Rev. Mod. Phys.* **74** 47
[60] Erdős P and Rényi A 1959 *Publ. Math.* **6** 290–7
[61] Bonsón E and Bednárová M 2013 *Online Inf. Rev.* **37** 969–84
[62] Golbeck J and Hendler J 2006 Filmtrust: movie recommendations using trust in web-based social networks *Ccnc Consumer Communications and Networking Conf.* pp 282–6
[63] Newman M E 2002 *Phys. Rev. Lett.* **89** 208701
[64] Ramasco J J, Dorogovtsev S N and Pastorsatorras R 2004 *Phys. Rev.* E **70** 036106
[65] Girvan M and Newman M E 2002 *Proc. Natl Acad. Sci.* **99** 7821–6
[66] Lancichinetti A, Fortunato S and Radicchi F 2008 *Phys. Rev.* E **78** 046110
[67] Illenberger J and Flötteröd G 2012 *Soc. Netw.* **34** 701–11
[68] González-Bailón S, Wang N, Rivero A, Borge-Holthoefer J and Moreno Y 2014 *Soc. Netw.* **38** 16–27
[69] Li R H, Yu J X, Qin L, Mao R and Jin T 2015 On random walk based graph sampling *IEEE 31st Int. Conf. on Data Engineering* (Piscataway, NJ: IEEE) pp 927–38
[70] Lee C H, Xu X and Eun D Y 2012 Beyond random walk and Metropolis–Hastings samplers: why you should not backtrack for unbiased graph sampling *ACM SIGMETRICS Performance Evaluation Review* vol 40 (New York: ACM) pp 319–30

J. Stat. Mech. (2018) 073403