Unbiased Sampling of Social Media Networks for Well-connected Subgraphs

Dong Wang*, Zhenyu Li*, Gareth Tyson‡, Zhenhua Li†, Gaogang Xie*
*ICT-CAS, ‡ Queen Mary University of London, †Tsinghua Univ.
{wangdong01, zyli, xie}@ict.ac.cn, {g.tyson}@qmul.ac.uk, {lizhenhua1983}@gmail.com

***Abstract*—Sampling social graphs is critical for studying things like information diffusion. However, it is often necessary to laboriously obtain *unbiased* and *well-connected* datasets because existing survey algorithms are unable to generate well-connected samples, and current random-walk based unbiased sampling algorithms adopt rejection sampling, which heavily undermines performance when applied in social media networks that show local disassortative mixing pattern. This paper proposes a novel random-walk based algorithm which implements Unbiased Sampling using Dummy Edges (USDE). It injects dummy edges between nodes, on which the walkers would otherwise experience excessive rejections before moving out from such nodes. We propose a rejection probability estimation algorithm to facilitate the construction of dummy edges and the computation of moving probabilities. Theoretical analysis shows that adding dummy edges improves both the sampling efficiency and convergence. We apply USDE in two synthetic networks and in two real-life social media: Twitter and Sina Weibo. The results demonstrate that USDE generates well-connected samples, and outperforms existing approaches in terms of sampling efficiency and quality of samples.**

I. Introduction

Social media have gained tremendous popularity in recent years. In order to characterize, optimize and simulate information diffusion within such networks, it is often necessary to collect realistic datasets [1][2][3]. However, the huge size of these networks makes it very hard to gain a true snapshot of the complete graph. Hence, it becomes necessary to obtain an *unbiased* and *well-connected* subgraph of the network. Here, a subgraph sample is unbiased if every user in the social media is sampled with equal probability, as widely adopted in literature [4][5]; and a sample is said to be well-connected if it has only a few connected components. Well-connected subgraphs are mandatory in information diffusion studies, because otherwise the scope of diffusion would be limited to small isolated components, failing to capture the process in the original social media.

Sampling networks has been heavily studied in the literature (§II). Survey sampling approaches (like stratified sampling [6] or uniform sampling [7]) and those using random jumps during sampling [8][5][9], while being able to provide unbiased estimations of individual node attributes (like node degree), fail to generate well-connected samples. Another kind of sampling, called random-walk sampling, while being able to provide well-connected samples, generates biased towards high-degree nodes. Several algorithms, including Metropolis-Hastings Random Walk (MHRW) [4] and its variants [10][11][12][13], were proposed to adapt random-walk based sampling to generate unbiased samples using the *rejection sampling* procedure.

The rejection sampling procedure explicitly rejects moving to high-degree nodes by increasing the probability of (re)sampling low-degree nodes. This procedure incurs the cost and delay of sampling without gathering information in exchange, especially when applied to online social media with local disassortative mixing pattern, where users tend to follow power-users (like celebrities) with degrees that are orders of magnitude larger [14][15]. In such networks, the random walkers will be trapped in the low-degree nodes for a long time due to the rejection sampling of high-degree nodes. Low-degree nodes, on the other hand, will be repeatedly sampled wastefully. Unfortunately, if these repetitions are removed, MHRW and its variants using rejection sampling fail to provide unbiased samples [16][11].

The above facts motivate us to propose a new sampling algorithm (§III): Unbiased Sampling with Dummy Edges (*USDE*). We aim to provide a good quality of samples (measured by the ability to provide unbiased and well-connected subgraphs) with high sampling efficiency (measured by the sampling convergence and the speed of discovering new nodes). USDE is a random-walk based algorithm that exploits artificially injected *dummy edges* (§III-A). The intuition is that, instead of rejecting high-degree nodes, the walkers move through dummy edges to other nodes, on which the walkers would otherwise experience excessive rejections. Note that the dummy edges are not included in the final samples and are only used in the sampling process temporarily. We propose a rejection probability (also called *self-sampling probability*) estimation algorithm to facilitate the construction of dummy edges and the computation of traversal (moving) probabilities (§III-B). By carefully assigning moving probabilities between adjacent nodes that are connected by either original edges or dummy edges, we show that our algorithm is able to generate unbiased and well-connected samples (§III-E). Theoretical analysis shows that adding dummy edges improves both the sampling speed and convergence (§IV), while evaluation on synthetic networks, as well as Twitter and Sina Weibo, further validates the efficiency (§VI).

II. Background and Motivation

A. Background

To study social phenomena it is necessary to collect realistic social media data. However, collecting entire social graphs is

often impossible. Hence, most studies rely on social graph *sampling* to collect data for analysis. Due to a low level of reciprocity [15], these social media are usually abstracted as directed graphs. From the perspective of sampling, it has been shown that the random walk with "backward edge traversals" in directed graphs (which allows the crawler to take unidirectional edges as bidirectional ones) can achieve similar performance to the ones in the corresponding undirected graphs [16][12]. We use a similar idea here. Taking Twitter as an example, we treat the unidirectional edges representing follower and following relationships as bidirectional ones. Hence, the Twitter network can be viewed as an undirected graph: $G = (V, E)$, where V is the set of nodes representing users, and E is the set of bidirectional edges.

Another prominent feature of online social media is that nodes with a very high degree (*e.g.* celebrities) may be surrounded by many low-degree nodes [15][14], implying *high local disassortativity*. Here, dissassortativity captures the phenomenon that nodes tend to be connected with other nodes with different degrees. Formally, the local (dis)assortativity metric is defined by in [17], where for a node of degree $(j+1)$, its local (dis)assortativity coefficient ρ is:

$$\rho = \frac{j(j+1)(\bar{k} - \mu_q)}{2M\sigma_q^2} \tag{1}$$

where $\bar{k}$ is the average remaining degree of the node's neighbors, M is the number of links in the network, μ_q and σ_q are the mean and standard deviation of the remaining degree distribution of the network respectively. A positive (*resp.* negative) coefficient ρ indicates an effect of local assortativity (*resp.* disassortativity).

For a high-degree node i (*e.g.* a celebrity in Twitter) that is surrounded by a large number of low-degree nodes, $\bar{k}$ is likely to be less than the global average of remaining degree distribution μ_q. In this case, ρ of the high-degree node i is negative but with a large absolute value, leading to a high local disassortativity.

B. Motivation

The existence of local disassortative nodes makes unbiased sampling challenging. Considering a high-degree node u that it is surrounded by many low-degree nodes, crawlers in the native random-walk sampling will move from u to its low-degree neighbor v with a low probability. And once the crawlers reach low-degree nodes, the native sampling prefers high-degree nodes as the next hop. This leads to sampling bias towards high-degree nodes. To address this problem, two kinds of methods can be adopted. One is using random jumps [16], [5], *i.e.* allowing the crawler to jump to a randomly chosen node (instead of following edges). Random jumps essentially decrease the sampling probability of high-degree nodes, while increase the probability of low-degree ones. Random jumps, however, will yield many small and isolated components in the final samples.

The other solution is using rejection sampling, like MHRW [4], *i.e.* rejecting to move to high-degree nodes but sampling low-degree nodes many times (called *self-sampling* here). This kind of solution generates well-connected samples because crawlers follow edges to move to the current node's neighbors. Self-sampling therefore increases the sampling probability of low-degree nodes and implicitly reduces the probability of high-degree nodes. However, for a network with local disassortativity, the self-sampling probability of low-degree nodes will be extremely high, because crawlers must experience excessive rejections of moving towards high-degree neighbors in order to compensate the sampling probabilities of low-degree nodes. The excessive rejections greatly hurts the efficiency of discovering new nodes.

Further, as shown in [4], MHRW achieves unbiased estimation of node properties by counting m self-samplings on a node as m distinct nodes with the same attributes in the final samples. However, if we should include edges in the final samples (for studies of information diffusion and alike), a unique node, along with its edges, can only be counted once in the graph, because otherwise it will increase the degree of the node's neighbors by adding $m-1$ edges on each neighbor. In this case, MHRW fails to provide unbiased samples. In this paper, we aim at solving this shortcoming of rejection sampling, in order to improve sampling efficiency and generate unbiased samples while eliminating repeated samplings.

III. Unbiased Sampling with Dummy Edges

This section presents our proposed sampling algorithm, USDE. The idea is to keep the connectivity and unbiased nature of samples obtained by rejection sampling algorithms (like MHRW), while avoiding excessive rejections (or self-samplings). To this end, we add dummy edges between nodes that would otherwise experience high self-samplings, and amortize the self-sampling probabilities of individual nodes to moving probabilities on the dummy edges. Importantly, dummy edges are only used during the sampling process to allow the sampling crawler to move to another node, and they are not involved in the final sample. Table I lists the notations used throughout of this paper.

TABLE I: Notations in the description of USDE

Notation	Definition
V'	set of nodes having been visited by sampling walkers
k_i	degree of node i in the abstracted undirected network
$P_{i,j}$	moving probability from node i to j through one step
$P_{i,j}^{(n)}$	moving probability from node i to j through n steps
π_i	probability that node i can be sampled
L_i	self-sampling probability of node i, *i.e.* $P_{i,i}$
$U(i)$	set of nodes that connect with i through dummy edges
$S(i)$	set of nodes that connect with i through original edges
$DP_{i,j}$	moving probability from i to j via the dummy edge
LP_i	lower bound of node i's self-sampling probability
$E_r(k)$	the average node degree with sampling repetitions
$E_u(k)$	the average node degree without sampling repetitions

A. Dummy Edges

Dummy edges are built between nodes that experience extensive self-sampling (rejection) probabilities in random-walk

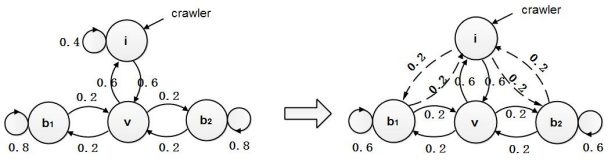


Fig. 1: Example of adding dummy edges on multiple nodes

based rejection samplings. Suppose the crawler is currently on node i; the candidate nodes for building dummy edges from i are the previously visited nodes' neighbors that have not been visited yet *and* have non-zero self-sampling probabilities. In this way, a node that connects with i through a dummy edge once being visited cannot form new connected components, since at least one of its neighbors has been visited. This avoids jumping to random nodes, and therefore avoids the generation of many small components.

Fig. 1 shows an example where multiple dummy edges are added. Note that when assigning a moving probability over edges, we keep the probability of moving out and moving in from every node to be 1, *i.e.*, the sum of moving probabilities over all edges of a node is 1. In the example, the sampling crawler is at node i with self-sampling probability of 0.4. Node v has already been visited. Two nodes, b_1 and b_2, have not been visited yet and their self-sampling probabilities are $L_{b_1} = L_{b_2} = 0.8$. If we add two dummy edges $\{i, b_1\}$ and $\{i, b_2\}$ and assign the moving probabilities on the two edges as $P_{i,b_1} = P_{b_1,i} = 0.2$, $P_{i,b_2} = P_{b_2,i} = 0.2$, then L_i, L_{b_1}, then L_{b_2} are reduced to 0, 0.6 and 0.6 respectively. As b_1 and b_2 have not yet been visited, moving the crawler to them improves the efficiency of identifying new nodes and user attributes. Besides, as their neighbor v has been visited, moving the crawler to them will not generate new connected component.

B. Moving Probability in USDE

At first, we describe the calculation of the moving probability from node i to node v in USDE using Eq. 2:

$$P_{i,v} = \begin{cases} min(\frac{1}{k_v}, \frac{1}{k_i}) & \text{if } v \in S(i) \\ DP_{i,v}, & \text{if } v \in U(i) \\ 1 - \sum_{x \in S(i)} P_{i,x} - \sum_{y \in U(i)} P_{i,y} & \text{if } v = i \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

where $U(i)$ is the set of nodes that have dummy edges with i, $S(i)$ is the set of original neighbors of i, and $DP_{i,j} = DP_{j,i}$.

We have showed in our previous work [10] that the sufficient and necessary condition for unbiased (nodal) sampling in a network G that has at least one node with non-zero clustering coefficient is: **(1)** if $P_{i,j} > 0$, then $P_{j,i} > 0$; **(2)** $\forall j \in V, \sum_{i=1}^{|V|} P_{i,j} = 1$, where $|V|$ is the number of nodes in network G. The above moving probability of USDE meets these two conditions, because following Eq. 2: $P_{i,j} = P_{j,i}$ and thus if $P_{i,j} > 0$, then $P_{j,i} > 0$; and $\forall i \in V, \sum_{j=1}^{|V|} P_{j,i} = \sum_{j=1}^{|V|} P_{i,j} = 1$. Hence, USDE generates unbiased (nodal) samples where each node is sampled with equal probability.

C. Dummy Edge Addition

The selection of nodes to build dummy edges is critical in USDE. The nodes to which dummy edges can be built from the currently visited node are the previously visited nodes' neighbors that have a non-zero self-sampling probability *and* have not yet been visited by the sampling crawler. However, finding such nodes is challenging because we are unable get the exact self-sampling probability of an unvisited node. Instead, we estimate the lower bound of the self-sampling probability for an unvisited neighbor based on the degree of the current visited node and the degree of this neighbor.

From Eq. 2, we can infer that if there is a neighbor u of node i with degree $k_u > k_i$, then the self-sampling probability of i without dummy edges is at least $(\frac{1}{k_i} - \frac{1}{k_u})$. Following this observation, the lower bound of an unvisited node i's self-sampling probability without dummy edges, LP_i, is:

$$LP_i = \sum_{v \in \{S(i) \cap V'\}} (\frac{1}{k_i} - \frac{1}{k_v}) \quad \text{where } k_v > k_i \quad (3)$$

where $S(i)$ is the neighbor set of i and V' is the set of nodes that have been visited.

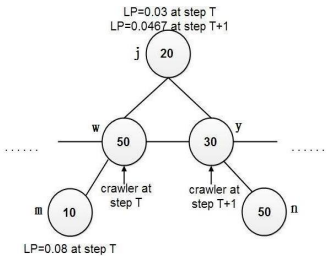


Fig. 2: Estimating self-sampling probability

Fig. 2 illustrates how the lower bounds are estimated during the sampling process. The node degrees are marked on the nodes and the node IDs are labeled beside the nodes. Let's assume that, before step T, the self-sampling probability for all nodes is 0. At step T, the crawler visits node w. Node j and node m are neighbors of w and $k_j < k_w$, $k_m < k_w$. Hence, we can estimate that $LP_j = \frac{1}{k_j} - \frac{1}{k_w} = \frac{1}{20} - \frac{1}{50} = 0.03$ and $LP_m = \frac{1}{k_m} - \frac{1}{k_w} = \frac{1}{10} - \frac{1}{50} = 0.08$ at step T. At the next step $T+1$, the crawler visits y. We update the estimation of LP_j because j is also a neighbor of y and $k_j < k_y$. The LP value of j is updated as $LP_j = 0.03 + \frac{1}{20} - \frac{1}{30} = 0.0467$. But we are unable to estimate LP_n at $T+1$ as $k_n > k_y$.

During the process of sampling, we use a queue Q to record the node ID and the estimated lower bound of self-sampling probability, LP_v, for each unvisited node v with $LP_v > 0$, *i.e.* a tuple (v, LP_v) for a node v. The nodes in this queue are candidate nodes for dummy edge addition from the currently visited node.

D. Computation of moving probabilities

When a node i is visited for the first time, we obtain its self-sampling probability as $1 - \sum_{x \in S(i)} P_{i,x} - \sum_{y \in U(i)} P_{i,y}$ according to Eq. 2. $U(i)$ is empty in the case that i has never been selected for building dummy edges before. If such a probability is larger than a threshold δ, we pop a tuple (v, LP_v)

from Q and add a new dummy edge between node i and v. The moving probability of the added dummy edge $DP_{i,v}$ ($DP_{v,i}$) is computed as Eq. 4.

$$DP_{i,v} = min(LP_v, 1 - \sum_{x \in S(i)} P_{i,x} - \sum_{y \in U(i)} P_{i,y}) \quad (4)$$

We then update LP_v with $LP_v^{'} = LP_v - DP_{v,i}$. If the updated $LP_v^{'} > 0$, we push the updated tuple (v, $LP_v^{'}$) back to Q. Such an estimation method (for the lower bounds of self-sampling probability) ensures that the sampled subgraphs are well-connected because all the nodes recorded in Q are neighbors of previously sampled ones.

The moving probability $DP_{i,v}$ ($DP_{v,i}$) of the dummy edge can be significant in the case of large LP_v and self-sampling probability. In this case, the sampling crawler will wander between i and v for a long time, which prevents the crawler from finding new nodes. To solve this problem, rather than pop only 1 tuple, we pop γ ($\gamma > 1$) tuples $(v_1, LP_{v_1}), (v_2, LP_{v_2}), (v_3, LP_{v_3}) \cdots (v_\gamma, LP_{v_\gamma})$ from Q at one time, and γ dummy edges are added $\{i,v_j\}$ ($j = 1, 2 \cdots \gamma$). The moving probability on dummy edge $\{i,\ v_j\}$ DP_{i,v_j} ($DP_{v_j,i}$) is computed as Eq. 5.

$$DP_{i,v_j} = min(LP_{v_j}, \frac{1 - \sum_{x \in S(i)} P_{i,x} - \sum_{y \in U(i)} P_{i,y}}{\gamma}) \quad (5)$$

LP_{v_j} is then updated accordingly and the tuples with non-zero LPs are pushed back to Q. The addition of dummy edges ceases when the self-sampling probability on i is reduced to 0, or dummy edges to all the γ nodes are added.

The queue Q applies FIFO (First In First Out), which avoids adding too many dummy edges on a single node. The stored nodes with fewer dummy edges and larger LP are more likely to be popped out, reducing self-sampling probabilities as much as possible. The queue Q is initialized in the first t iterations of sampling. During this time period, dummy edges are not added, and the LPs of the visited nodes' neighbors are estimated and pushed into Q as dummy edge candidates.

E. USDE Implementation

To sum up, the pseudo code of USDE is listed in Algorithm 1, where t (the number of iterations for the queue initialization), δ (the threshold on self-sampling probability) and γ (the number of records popped from Q at a time) are design parameters. The list R stores unique node ID sampled.

Our USDE implementation leverages multiple random-walk based crawlers that run in parallel for efficient sampling. At the beginning, a set of initial nodes (called seeds) are uniformly selected from the network at random. Each seed initializes a crawler following Algorithm 1. Crawlers share the common queue Q for candidates of end points for dummy edges.[1] In this way, each crawler is able to access the candidate end points for dummy edges generated by other crawlers, enabling the building of dummy edges between two nodes in different communities of the network. The shared queue Q essentially facilitates the formation of large connected components in the sampled subgraphs. Finally, nodes that are visited by crawlers, along with the edges between them, are exported to form the final samples. The dummy edges, however, are not included in the final result.

It is also noteworthy that since sampling on the graph with dummy edges still keeps the necessary and sufficient condition for unbiased sampling stated in [10], the sampling is still nodal unbiased.

[1]We associate the queue Q with a mutex lock.

Algorithm 1 Unbiased sampling with dummy edge (USDE)

procedure USDE(t, δ, γ)
  $R \leftarrow \varnothing$, $Q \leftarrow \varnothing$, $v \leftarrow$ initial seed
  while stopping criterion has not been met **do**
    if $v \notin R$ **then**
      insert v in R
      if $(v, LP_v) \in Q$ **then** ▷ visited nodes are not eligible for dummy edges
        pop (v, LP_v) from Q
      end if
      $L_v \leftarrow 1$
      for each neighbor i of v **do** ▷ estimating self-sampling prob. of neighbors
        $P_{i,v} = P_{v,i} \leftarrow min(\frac{1}{k_i}, \frac{1}{k_v})$
        $L_v \leftarrow L_v - P_{v,i}$
        EstimateLP(v, i, Q, R)
      end for
      for each tuple $(k, P_{v,k}) \in U(v)$ **do** ▷ if v has been chosen as the end point of dummy edges before it is visited
        $L_v \leftarrow L_v - P_{v,k}$
      end for
      if $Iteration > t$ and $L_v > \delta$ **then** ▷ adding dummy edges
        AddDummyEdges(v, Q, γ)
      end if
    end if
    Select a neighbor w according to Eq. 2
    $v \leftarrow w$ ▷ moving to a neighbor
  end while
end procedure

procedure ESTIMATELP(v, i, Q, R)
  if $k_v > k_i$ and $i \notin R$ and $i \in Q$ **then**
    $LP_i \leftarrow LP_i + (\frac{1}{k_i} - \frac{1}{k_v})$ in Q
  end if
  if $k_v > k_i$ and $i \notin R$ and $i \notin Q$ **then**
    $LP_i \leftarrow \frac{1}{k_i} - \frac{1}{k_v}$, push $(i,\ LP_i)$ into Q
  end if
end procedure

procedure ADDDUMMYEDGES(v, Q, γ)
  $N \leftarrow 0$
  while $N < \gamma$ and $L_v > 0$ **do**
    pop (b, LP_b) from Q
    $P_{v,b} = P_{b,v} \leftarrow min(LP_b, \frac{L_v}{\gamma})$
    push(b, $P_{v,b}$) into $U(v)$ ▷ adding dummy edges
    push $(v, P_{b,v})$ into $U(b)$
    $L_v \leftarrow L_v - P_{v,b}$ ▷ updating self-sampling probability
    $LP_b \leftarrow LP_b - P_{b,v}$
    if $LP_b > 0$ **then**
      push (b,LP_b) into Q
    end if
    $N \leftarrow N + 1$
  end while
end procedure

IV. SAMPLING EFFICIENCY ANALYSIS

We analyze the sampling efficiency of USDE from two perspectives: the speed in discovering new nodes during the sampling process and the convergence of the crawlers. We show that the addition of dummy edges via USDE can reduce

the sampling times per node and improve the convergence by increasing the "conductance" of the sampled network.

A. Sampling Times per Node

Given the sampling iterations, the speed of discovering new nodes is inversely proportional to the sampling times per node. Suppose the crawler starts from a node i. The average sampling times on node i within T steps is $\sum_{n=1}^{T} P_{i,i}^{(n)}$ [18]. Recall that USDE meets the sufficient and necessary condition for unbiased (nodal) sampling that is given in [10]. Thus, when $P_{j,k}^{(n)}$ reaches the stable state (when n is sufficiently large), it converges to the stationary distribution of k, π_k (*i.e.*$\frac{1}{|V|}$) for $\forall j, k \in V$. Since $|V|$ is sufficiently large (*e.g.* billions in Twitter and Sina Weibo), $\sum_{n=1}^{T} P_{i,i}^{(n)}$ is dominated by the first m iterations $\sum_{n=1}^{m} P_{i,i}^{(n)}$, where $m < T$ is a small number. In other words, the sampling time per node is largely determined by the self-sampling probability in the first m iterations.

Next, let's analyze the reduction of sampling time per node from the perspective of the transition matrix P. For a Markov chain, $P_{i,j}^{(n)}$ is the element in the ith row and the jth column (we use (i,i)-element to represent) of P^n. For a node i with high self-sampling probability in the rejection sampling algorithm (*e.g.* MHRW), we have $P_{i,i} \gg P_{i,j}$ and $P_{i,i} \gg P_{j,i}$, where $j \neq i$. As such, for a small number n, $P_{i,i}^{(n)} \gg P_{i,j}^{(n)}$ holds for any j that $j \neq i$ in P^n.

After the addition of dummy edges, $P_{i,i}$ is amortized to $P_{i,j}$ and $P_{j,i}$, where $j \neq i$, making $\hat{P}_{i,i}$ close to (or even smaller than) the others in the same row or the same column in the new transition matrix $\hat{P}$. In that case, the (i,i)-element in $\hat{P}^n$ gets close to (or even smaller than) the other elements in the ith row, compared with the one in P^n. As $\sum_{\forall j \in V} P_{i,j}^{(n)} = 1$ and $\sum_{\forall j \in V} \hat{P}_{i,j}^{(n)} = 1$, it is obvious that $P_{i,i}^{(n)}$ is likely to be higher than $\hat{P}_{i,i}^{(n)}$ for a small number n. Recall that $P_{i,i}^{(n)}$ largely determines the sampling times per node, we can conclude that USDE reduces average sampling times per node, and in turn improves the speed in discovering new nodes compared with traditional rejection samplings, like MHRW and its variations.

B. Convergence of USDE

Because of the addition of dummy edges, the structure of the graph is changing dynamically during the execution of USDE. It is thus difficult to calculate the exact convergence rate of the sampling probability. Fortunately, we show that the addition of dummy edges in each step is beneficial to the convergence.

The "conductance" of a graph $G(V, E)$ measures how "well-knit" the graph is. It also controls how fast a random-walk based crawler on G converges to a uniform distribution of sampling probabilities [19]. We thus study the convergence of USDE from the perspective of the conductance of sampled graphs. The conductance of a set A in graph G is defined as follows [20]:

$$\varphi(A) = \frac{F(A)}{\pi(A)(1-\pi(A))} \tag{6}$$

where $F(A)$ is called as the *ergodic flow* from A, which is defined as $F(A) = \sum_{i \in A} \sum_{j \notin A} \pi_i P_{i,j}$, and π is the stationary distribution vector of the sampling process. $F(A)$ represents the fraction of steps of a very long random walk which moves from A to its complement. The denominator denotes the fraction of steps of a very long sequence of independent samples from π that moves from A to its complement, where $\pi(A) = \sum_{i \in A} \pi_i$. The conductance of the whole graph is the minimum conductance over all non-empty proper subsets of V.

A small conductance indicates that the network consists of several communities, which have few links to others. In this context, USDE with a single random walk crawler may prevent the crawler in one community from discovering other communities as the dummy edges are always added in the community of the starting node.

The above issue is addressed by the multiple random walkers (or crawlers) running in parallel. Since a common queue Q of candidate nodes for dummy edges is shared among crawlers, crawlers in one community have access to the candidates in other communities, enabling the building of dummy edges between two nodes in different communities. Let $\varphi(A)$ be the minimum conductance of G without dummy edges and $F(A)$ be the ergodic flow from the community A. We further denote $K(A)$ as the set of dummy edges between nodes $i \in A$ and nodes $j \notin A$ added by USDE. After adding dummy edges, the updated conductance of A in USDE $\hat{\varphi}(A)$ is:

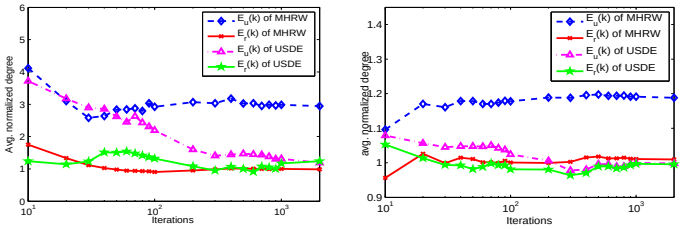
$$\begin{aligned} \hat{\varphi}(A) &= \frac{\hat{F}(A)}{\hat{\pi}(A)(1-\hat{\pi}(A))} \\ &= \frac{F(A) + \sum_{i \in A} \sum_{(i,j) \in K(A)} \hat{\pi}_i P_{i,j}}{\pi(A)(1-\pi(A))} \\ &= \varphi(A) + \frac{\sum_{i \in A} \sum_{(i,j) \in K(A)} \hat{\pi}_i P_{i,j}}{\pi(A)(1-\pi(A))} \end{aligned} \tag{7}$$

where $\hat{\pi}_i = \hat{\pi}_j = \frac{1}{|V|}, \forall i, j \in V$ since the sampling on the new graph is still an unbiased stationary process as already stated. Obviously, $\hat{\varphi}(A)$ is larger than $\varphi(A)$ as long as $K(A)$ is non-empty. In other words, dummy edges added by USDE between A and its complement increase the "conductance" of the graph and therefore improve the sampling convergence.

The condition for $K(A)$ being non-empty is that some crawlers start from A *and* some from the complement of A. Suppose the proportion of nodes in A is ρ_A. Given that the initial seeds are randomly selected, the probability of $K(A)$ is non-empty is $p_{K(A)} = 1 - \rho_A^n - (1-\rho_A)^n$, where n is the number of crawlers running in parallel. Therefore, $p_{K(A)}$ is very high with a limited number of crawlers. For instance, if $\rho_A = 5\%$, $p_{K(A)}$ is as high as 87% with only $n = 40$ crawlers (similar to MHRW in [11]).

V. Evaluation on Synthetic Networks

This section evaluates USDE on two synthetic networks, each with 20,000 nodes. The synthetic networks allow us to have more control over the experiments.



(a) G_p (b) G_b

Fig. 3: Average normalized node degree

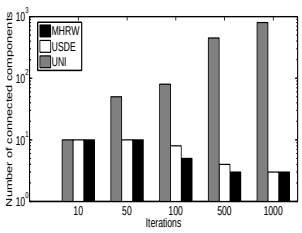


Fig. 4: # connected components in sampled subgraphs

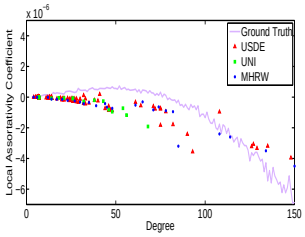


Fig. 5: Estimated local assortativity coefficients

A. Experiment Setup

Test Data: The first network is generated using *Exploration without replacement* [21] with a mean degree around 30 following a power-law distribution (referred to as G_p). This follows the degree distribution of real-life online social media [15]. The other one is a barbell graph (referred to as G_b), of which two communities with 10,000 nodes are generated with a Gaussian degree distribution with mean degrees of 20 and 40, respectively. We randomly choose from each community one node and then link them with a single edge. This is used to measure how the conductance of the graph controls the convergence rate of sampling.

We also associate nodes with user attributes. Each node is assigned an integer $a \in [1, 200]$ with two probability distributions: Gaussian and power-law distributions, which are denoted as G-attribute and P-attribute, respectively. Each integer represents a category of attributes. We run sampling experiments with 10 crawlers simultaneously starting from randomly selected seeds. The default design parameters of USDE are: $t = 100$; $\delta = 0.00001$; $\gamma = 100$.

Metrics: We evaluate the performance from two perspectives: *the quality of samples* and *the sampling efficiency*. The quality of samples is measured by the closeness of sampled node degrees and user attributes to the ground truth, as well as the connectivity of the sampled subgraphs; the sampling efficiency is measured by the speed of identifying new nodes and new user attributes, and the convergency of crawler.

B. Quality of samples

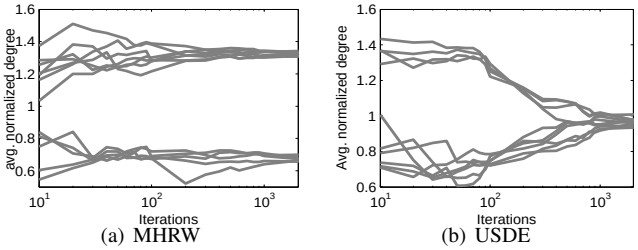
An important metric used in the literature to measure the unbias of sampling algorithms is the sampled average node degree. Due the existence of self-samplings, we measure two node degree metrics: $E_r(k)$ — the average node degree when m samplings on a node are counted as m nodes with identical properties; and $E_u(k)$ — the average node degree when a node is only counted once no matter how many times it is sampled. Fig. 3 plots these two metrics normalized by ground truth of average node degree. Both USDE and MHRW provide stable results after 300 iterations. However, MHRW converges to a higher $E_u(k)$ than the ground truth, although $E_r(k)$ is close to that. Our results reconfirm that if repetitions are removed, MHRW fails to provide unbiased samples [16][11]. In contrast, both $E_u(k)$ and $E_r(k)$ computed using samples generated by USDE converge to the ground truth.

Next, we measure the connectivity of samples by counting the number of connected components in samples, an important metric for studies of user interactions and information propagation [22][3]. Besides MHRW, we also compare USDE with uniform sampling (UNI), which samples nodes in a graph randomly with uniform probability and is always used to generate unbiased degree distribution [4]. Fig. 4 shows the results generated by 10 crawlers on G_b. The crawlers select distinct seeds at the beginning of the sampling process (which is evidenced by the 10 connected components after 10 iterations). The connectivity of samples of MHRW and USDE increases with the growth of iterations, while that of UNI decreases quickly.

The connectivity of sampled subgraphs is also of great importance for the computation of topological features such as local assortativity. As depicted in Eq. 1, the degree of the current node i and its neighbors are both necessary for the definition of i's local assortativity coefficient. However, in samples with lots of isolated small components, the information of most edges might be lost. To demonstrate this, we compute the local assortativity coefficient based on the sampled edges and plot the distribution in Fig. 5, where 1% nodes are sampled on G_p.

Although UNI is able to provide an unbiased estimation of degree distribution, the distribution of local assortativity coefficients on samples (consisting of only sampled nodes and sampled edges) is unable to reveal the dominant pattern of local disassortativity. This is because it generates many isolated small components. On the contrary, both USDE and MHRW yield much more accurate distributions of the coefficients.

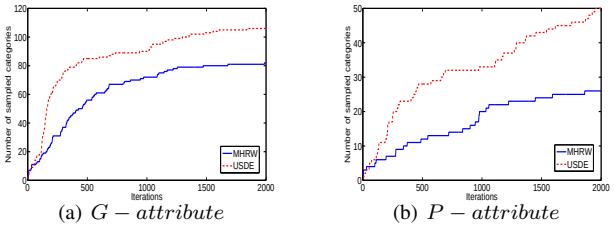
C. Sampling Efficiency



(a) MHRW (b) USDE

Fig. 6: Convergence of MHRW and USDE on G_b

We next measure the time (in terms of sampling iterations) required to obtain stable and accurate results on the barbell network G_b. Fig. 6 compares the sampling convergence speed of MHRW and USDE, where the y-axis is the average normalized degree of $E_r(k)$. It is notable that USDE has a much better performance in terms of convergence than MHRW, which confirms our analysis in the previous section. The effect of dummy edges is indeed very notable, because USDE becomes convergent after 100 iterations when we begin to add dummy edges ($t = 100$).



(a) $G-attribute$ (b) $P-attribute$

Fig. 7: Efficiency of identifying new attributes on G_b

We finally examine the speed of identifying new attributes in Fig. 7. USDE consistently outperforms MHRW for both attribute distributions. For example, while about 90 categories in Fig. 7(a) have been discovered by USDE within 1,000 sampling iterations, MHRW has discovered only 60 within the same period, implying USDE is 50% more efficient than MHRW. We also observe the effect of the user attribute distribution on the efficiency in identifying new attributes. It requires more sampling iterations to discover the same amount of categories for P-attribute than G-attribute. The reason is that P-attribute contains lots of "slight" attributes in its tail distribution, which are more difficult to be discovered.

VI. Sampling Twitter and Sina Weibo

To confirm the practicality of our technique, we now use USDE to sample Twitter and Sina Weibo.We focus on the quality of samples and the sampling efficiency as in the previous section.

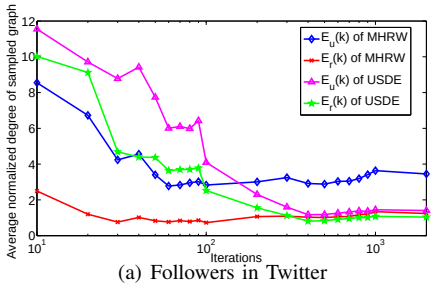
A. Experiment Setup

Both Twitter and Sina Weibo have a dense numerical ID space. We randomly generate a number of IDs, and choose 10 IDs that correspond to valid users as the initial seeds for crawlers. Since the relationship between users might be non-reciprocal in Twitter and Sina Weibo, we leverage the idea of "backward edge traversals" [12], where unidirectional edges are treated as bidirectional ones.

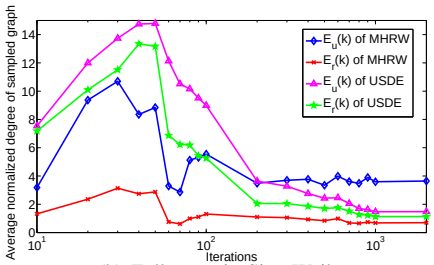
The evaluation of unbias requires the ground truth of user attributes in Twitter and Sina Weibo. Since we are aiming at unbias in terms of nodes (as opposed to edges), the ground truth of user attributes can be estimated using UNI[2] [4]. We thus uniformly generate a large number of user IDs at random and use these IDs to query Twitter and Sina Weibo APIs. During the sampling process, we rely on the APIs provided by Twitter and Sina Weibo to obtain the information of the currently visited node's neighbors.

It is worth briefly noting that for some social network platforms, like Google Plus, the user ID space is sparse and the random ID generation is difficult. Fortunately, popular social networks always provide public profile directories that can be used for random seed selection. For example, Google maintains a sitemap file that contains a link to every Google Plus profile public[3]; Twitter[4] and Facebook[5] also provide public user profile directories. The initial seeds can therefore be uniformly chosen from these profile directories [23].

B. Quality of samples



(a) Followers in Twitter



(b) Followers in Sina Weibo

Fig. 8: Normalized average number of followers

We first examine the accuracy of estimating the number of followers in Fig. 8, where the y-axis shows the average number of followers (normalized by the ground truth). We observe large variations of $E_r(k)$ and $E_u(k)$ for the initial sampling iterations due to the random choice of seed nodes. The normalized $E_r(k)$ gradually converges to 1 after taking several hundred iterations. Nevertheless, while the normalized $E_u(k)$ estimated by USDE in both networks converges close to 1, the $E_u(k)$ estimated by MHRW is about four times as high as the ground truth. Such a huge difference is due to excessive sampling repetitions in MHRW.

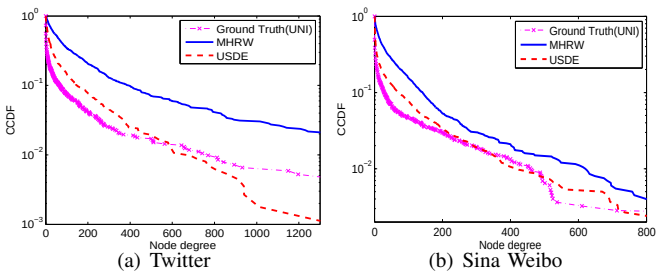
Fig. 9 further examines the distribution of followers obtained by sampling algorithms after 2,000 iterations, where the sampled users are counted uniquely. It is notable that USDE generates a much closer distribution to UNI (ground truth for nodal properties) than MHRW. For instance, samples

[2]It is worth noting that uniform sampling is unable to provide well-connected samples, and thus is not an alternative for USDE or MHRW.

[3]http://www.gstatic.com/s2/sitemaps/profiles-sitemap.xml

[4]https://twitter.com/i/directory/profiles

[5]http://www.facebook.com/directory



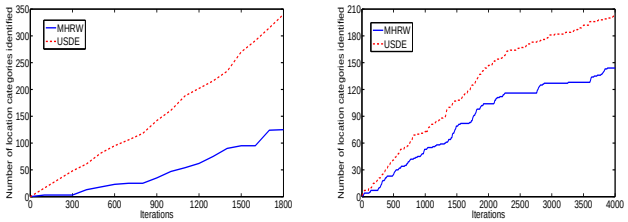
(a) Twitter (b) Sina Weibo

Fig. 9: Complementary CDF of followers in samples

generated by USDE reveal 93% users have fewer than 200 followers, close to the the corresponding proportion of ground truth 95%. Nevertheless, this percentage in samples generated by MHRW is only 80%.

C. Sampling Efficiency

Our results reveal that, for both Twitter and Sina Weibo, the average sampling times per node by USDE is close to 2, much lower than that by MHRW, which is 6-8 and 8-10 for Twitter and Sina Weibo, respectively.



(a) Twitter (b) Sina Weibo

Fig. 10: Efficiency of identifying location information

We finally investigate the efficiency in identifying new locations in Fig. 10. USDE consistently identifies many more new locations than MHRW. For example, USDE identifies more than 200 cities in Twitter within 1,200 iterations, while MHRW only identified about 50. The difference between the two curves in Twitter is more significant than that in Sina Weibo, possibly due to the categories of attributes in Twitter (world-wide), which are more diverse than the ones in Sina Weibo (country-wide).

VII. Conclusion

Studies on information diffusion in online social media require unbiased and well-connected samples of online social media networks. However, the local disassortative mixing patterns in online social media makes efficient sampling difficult. To this end, we propose a novel random-walk based sampling algorithm: USDE. It introduces dummy edges between nodes with high self-sampling probabilities to allow crawlers to move between nodes, while still keeping the connectivity of samples. We have detailed the way of building dummy edges and the computation of moving probabilities, and theoretically analyzed the performance of USDE. The evaluations on both synthetic networks and real-life social media have demonstrated that, in comparison with existing algorithms, USDE achieves a better quality of samples and higher sampling efficiency. Our future work will focus on further utilisation of USDE for collecting real data sets. We wish to then exploit this to further validate its ability to effectively sample a wide range of node attributes, and structural properties. We envisage this might lead to subsequent improvements.

References

[1] D. Wang, H. Park, G. Xie, S. Moon, M.-A. Kaafar, and K. Salamatian, "A genealogy of information spreading on microblogs: A galton-watson-based explicative model," in *Proceedings IEEE INFOCOM*, 2013.

[2] J. Yang and S. Counts, "Predicting the speed, scale, and range of information diffusion in Twitter," in *Proceedings of the ICWSM*, 2010.

[3] A. Sridharan, Y. Gao, K. Wu, and J. Nastos, "Statistical behavior of embeddedness and communities of overlapping cliques in online social networks," in *Proceedings IEEE INFOCOM*, 2011.

[4] M. Gjoka, M. Kurant, C. T. Butts, and A. Markopoulou, "Walking in facebook: A case study of unbiased sampling of osns," in *Proceedings of the IEEE INFOCOM*, 2010.

[5] B. Ribeiro, P. Wang, F. Murai, and D. Towsley, "Sampling directed graphs with random walks," in *Proceedings of the IEEE INFOCOM*, 2012.

[6] C.-E. Sarndal, "Stratified sampling," in *Model Assisted Survey Sampling*. Springer, 2003.

[7] M. R. Henzinger, A. Heydon, M. Mitzenmacher, and M. Najork, "On near-uniform url sampling," *Computer Networks*, vol. 33, no. 1, 2000.

[8] K. Avrachenkov, B. Ribeiro, and D. Towsley, "Improving random walk estimation accuracy with uniform restarts." in *Proceedings of the 7th Workshop on Algorithms and Models for the Web Graph*, 2010.

[9] E. Voudigari, N. Salamanos, T. Papageorgiou, and E. J. Yannakoudakis, "Rank degree: An efficient algorithm for graph sampling," in *Proceedings of ASONAM*, 2016.

[10] D. Wang, Z. Li, and G. Xie, "Towards unbiased sampling of online social networks," in *Proceedings of the IEEE ICC*, 2011.

[11] M. Gjoka, M. Kurant, C. T. Butts, and A. Markopoulou, "Practical recommendations on crawling online social networks," *IEEE JSAC*, vol. 29, no. 9, 2011.

[12] T. Wang, Y. Chen, Z. Zhang, P. Sun, B. Deng, and X. Li, "Unbiased sampling in directed social graph," *ACM SIGCOMM Computer Communication Review*, vol. 41, no. 4, 2011.

[13] F. Chiericetti, A. Dasgupta, R. Kumar, S. Lattanzi, and T. Sarlos, "On sampling nodes in a network," in *Proceedings of the WWW*, 2016.

[14] M. Cha, H. Haddadi, F. Benevenuto, and P. K. Gummadi, "Measuring user influence in twitter: The million follower fallacy." in *Proceedings of the ICWSM*, 2010.

[15] H. Kwak, C. Lee, H. Park, and S. Moon, "What is twitter, a social network or a news media?" in *Proceedings of WWW*, 2010.

[16] B. F. Ribeiro and D. Towsley, "On the estimation accuracy of degree distributions from graph sampling," in *Proceedings of IEEE Decision and Control (CDC)*, 2012.

[17] M. Piraveenan, M. Prokopenko, and A. Zomaya, "Local assortativeness in scale-free networks," *EPL (Europhysics Letters)*, vol. 89, no. 4, p. 49901, 2010.

[18] S. M. Ross *et al.*, *Stochastic processes*. John Wiley & Sons New York, 1996, vol. 2.

[19] B. Bollobás, *Modern graph theory*. Springer, 1998, vol. 184.

[20] R. Kannan, L. Lovász, and R. Montenegro, "Blocking conductance and mixing in random walks," *Combinatorics Probability & Computing*, vol. 15, no. 4, 2006.

[21] M. Kurant, A. Markopoulou, and P. Thiran, "On the bias of bfs (breadth first search)," in *Proceedings of the ITC*, 2010.

[22] Y. Singer, "How to win friends and influence people, truthfully: influence maximization mechanisms for social networks," in *Proceedings of ACM WSDM*, 2012.

[23] T. Chen, A. Chaabane, P. Tournoux, M.-A. Kaafar, and R. Boreli, "How much is too much? leveraging ads audience estimation to evaluate public profile uniqueness," in *Proceedings of Privacy Enhancing Technologies (PETs)*, 2013.