On Estimating Degree Distributions of Directed Graphs through Sampling

UMass Technical Report UM-CS-2010-046

Bruno Ribeiro[1], Pinghui Wang[2], and Don Towsley[1]

[1]Computer Science Department
University of Massachusetts
Amherst, MA, 01003
{ribeiro, towsley}@cs.umass.edu

[2]State Key Lab for Manufacturing Systems
Xi'an Jiaotong University
Xi'an P.R.China
phwang@sei.xjtu.edu.cn

***Abstract*—Despite the recent popularity of online social networks (OSN), little attention has been given to developing tools that can characterize directed OSN networks. An OSN is directed when the relationships (edges) between its vertices (users or profiles) may not be reciprocated (or the reciprocation is not made public). In networks where sampling vertices uniformly at random is feasible but expensive, crawling the OSN graph is often a cheaper sampling method. For undirected graphs and when the cost of independent sampling is high, random walks (a type of crawl) are known to be better alternatives to uniform (independent) vertex sampling. However, in the presence of hidden directed edges (e.g. the WWW and Flickr graphs only have visible outgoing edges), there may be no known path from a given vertex to all other vertices, which is a barrier to estimating graph characteristics.**

In this work we propose a sampling algorithm, *random walk with jumps*, that estimates out-degree distributions efficiently. Our random walker walks known edges backwards and also performs random jumps. We also show that, when the in-degree of a vertex is a latent variable (i.e., incoming edges are hidden, which is the case at the Flickr social network, and private profiles at Facebook and MySpace), any unbiased in-degree distribution estimator needs to sample most of the hidden edges in order to obtain accurate estimates of the in-degree distribution.

I. Introduction

Despite the recent efforts to characterize online social networks (OSNs), little attention has been given to developing tools that can characterize directed OSNs. An OSN is said to be directed when the relationships between its vertices (users or profiles) may not be reciprocated (or the reciprocation is not made public). For instance, if a Flickr user a (Flickr, [6] is a popular photo-sharing social network) subscribes to user's b photo updates it does not imply that user b also subscribes to user's a updates.

One of the main difficulties in sampling directed OSNs, such as Flickr, is the presence of hidden incoming edges. An edge $b \to a$ is hidden from user a if $b \to a$ can only be observed by querying user b. For instance, querying user a in the Flickr network returns only a's subscriptions (i.e., all a's outgoing edges) but not a's subscribers (i.e., none of a's incoming edges). Clearly, one can find a's subscribers by querying the subscriptions of all Flickr users, but this is not a practical approach for large OSNs. Crawling a graph with hidden edges is difficult. The existence of hidden incoming edges prevents crawling the graph as there may be no (known) path from a given vertex to all other vertices. Thus, even seemly simple tasks such as estimating the in- and out-degree distributions can be challenging when large directed graphs have hidden (incoming) edges.

The above navigability issue is not restricted to Flickr. This issue also arises in OSNs like Facebook [5] and MySpace [14]. The literature often portrays Facebook and MySpace as undirected graphs, where vertices are user profiles and edges are "friendships" relations among profiles [5], [13], [14]. In practice, however, user privacy settings can lead to asymmetric disclosure of edges (friendships). For instance, a Facebook profile a can publicly disclose its friendship with profile b while profile b does not publicly disclose its friendship with a [5]. The same is true with MySpace [14]. We denote a and b public and private vertices (profiles), respectively. Thus, Facebook and MySpace can be seen as directed graphs where a subset of its vertices (private profiles) have hidden incoming edges and no outgoing edges; and its remaining vertices (public profiles) have visible incoming and outgoing edges.

Contributions

Our work makes two contributions:

1) Random Walk with Jumps: To address the above navigability issue, we modify the random walk proposed by Bar-Yossef et al. [1] and add random jumps (a jump to a randomly chosen vertex) similar (but not equivalent) to the random jumps performed by the PageRank [3] algorithm. We call our algorithm *random walk with jumps* (RWwJ). Our algorithm, unlike the random walk in [1] or PageRank, can be used to estimate out-degree distributions efficiently and accurately. RWwJ can be used over networks such as Flickr, Facebook and MySpace, which admit random jumps. RWwJ is advantageous over just randomly sampling vertices when sampling a neighbor of a known vertex is much cheaper (resource-wise) than performing independent uniform sampling. Moreover, RWwJ has a jump probability parameter that allows one to tradeoff estimation accuracy gained by jumping with the cost of independently sampling vertices.

2) In-degree Distribution Estimation: We also present an unbiased in-degree distribution estimator for graphs that have hidden incoming edges. Our work answers the following question: How much of the graph needs to be sampled in order to obtain accurate in-degree distribution estimates? We analyze two OSNs (Flickr and Facebook) and observe that the answer to the above question is negative. We show that *any* unbiased in-degree distribution estimator needs to sample most of the OSN in order to obtain accurate estimates. We also see that side information, such as knowing the fraction of edges in the Flickr graph that are symmetric (an edge $a \rightarrow b$ is symmetric if the graph has an edge $b \rightarrow a$), has little impact on the accuracy of the estimates.

Outline

The rest of the paper is organized as follows. Section II presents the graph model and some definitions used throughout this work. Section III presents our *random walk with jumps* algorithm and an estimator for the out-degree distribution. Section III-D presents an out-degree distribution estimator using the samples obtained in the random walk. Section IV shows that, for OSN graphs with hidden incoming edges, it is necessary to sample most of the graph edges in order to accurately estimate the in-degree distribution. Section V reviews the related work. Finally, Section VI presents our conclusions and future work.

II. Definitions and Problem Formulation

Let $G_d = (V, E_d)$ be a directed graph, where V is the set of vertices and E_d is the set of edges. Let $\mathrm{o}(v)$ denote the number of edges to vertex $v \in V$ (out-degree) and $\mathrm{i}(v)$ denote the number of edges from vertex $v \in V$ (in-degree). We seek to obtain both the out-degree distribution $\phi = (\phi_0, \phi_1, ..., \phi_R)$ and the in-degree distribution $\theta = (\theta_0, \theta_1, ..., \theta_W)$, where ϕ_l is the fraction of vertices with out-degree l, θ_j is the fraction of vertices with in-degree j, R is the largest out-degree, and W is the largest in-degree.

The degree distribution of a large undirected graph can be estimated using random walks (RW) [8], [13], [15]. But these RW methods cannot be readily applied to directed graphs with hidden incoming edges, the case of a number of interesting directed networks, e.g., the WWW and Flickr.

To address these problems, we build a random walk with jumps under the assumption that vertices can be sampled uniformly at random from G_d (something not feasible for the WWW graph but possible for Flickr, Facebook, and MySpace). But why perform a random walk if we can sample vertices uniformly? This is useful for networks where uniform vertex sampling is costly (e.g., Flickr, Facebook, and MySpace). In networks such as Flickr, Facebook, and MySpace one can uniformly sample users (vertices) as users have numeric IDs between a minimum and a maximum ID values. The high cost of sampling comes from the fact that the ID space in these networks is sparsely populated [5], [6], [14] and most of the uniformly generated ID values are invalid. The cost of random vertex sampling, which we denote as c, is the average number of IDs queried until one valid ID is obtained. For instance, in the case of MySpace and Flickr, these costs are estimated to be $c = 10$ [14] and $c = 77$ (as seen in the Appendix), respectively.

III. Sampling Directed Graphs with RWs

Estimating characteristics of *undirected* graphs with random walks (RWs) is the subject of a number of recent works [13], [15], [17]. RW estimation methods presented in the literature require that $\forall u, v \in V$, the probability of eventually reaching u given that the walker is in v be non-zero. However, over a directed graph with incoming hidden edges this may not be true. For instance, consider a vertex $v \in V$ that has one outgoing edge but no incoming edges. If the random walker does not start at v then v is not visited by the walker (as the outgoing edge of v is a hidden incoming edge if some other vertex). On the other hand, a vertex $u \in V$ with no outgoing edges becomes a sink to the random walker.

To address this issue our algorithm borrows elements from the PageRank algorithm [3] and from the RW algorithm described by Bar-Yossef [1]. Our algorithm, however, differs from both algorithms, in that it can be used to estimate graph characteristics from a (possibly small) sample of the graph.

To guarantee that the RW can reach any vertex from any other vertex, we allow the random walker to jump to a randomly (uniformly) chosen vertex in the graph, similar to the PageRank algorithm [3]. In the PageRank algorithm a RW at vertex v jumps to a uniformly chosen vertex in the graph with probability w; and with probability $(1-w)$ the random walker follows an edge chosen uniformly at random from the set of outgoing edges of v. However, as stated before, uniform vertex sampling is a potentially "costly" form of sampling. Tuning w allows us to control this cost.

Unfortunately, PageRank does not allow us to accurately estimate graph characteristics, such as the out-degree distribution, from a sampled subset of the graph. Estimating these characteristics requires obtaining the steady state distribution of the RW without exploring the entire graph [15].

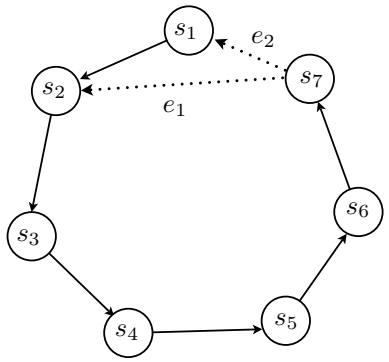

Figure 1. The steady state distribution of PageRank over this graph depends on the existence of edges e_1 and e_2.

In the following example we see that the steady state distribution of PageRank requires knowing the graph structure.

Consider the directed graph shown in Figure 1, which has 7 vertices. Assume that we do not observe incoming edges. Let s_1 be the starting vertex of PageRank. At i-th step the RW is at vertex s_i. Let $\pi(v)$ denote the steady state probability of PageRank visiting vertex v, $\forall v \in V$. Assume that at the k-th step PageRank has not visited vertex s_7. Without knowing the edges of s_7, say whether edges e_1 or e_2 exist, we cannot determine $\pi(s_1)$. If e_1 exists but e_2 does not, $\pi(s_1)$ is a function of w where $\lim_{w\to 0}\pi(s_1) = 0$. If e_2 exists but e_1 does not, $\pi(s_1) = 1/7$ (independently of the value of w). In this example we need to know the entire graph to find a reasonable approximation to $\pi(s_1)$. Thus, to avoid having to explore the entire graph, we perform:

- (**Backward edge traversals**) Similar to Bar-Yossef's algorithm, [1] we allow the random walker to traverse some known outgoing edges backwards. For instance, if at the i-th step the RW is at vertex s_i, we allow the random walker to traverse the edge $s_{i-1} \to s_i$ backwards. To do this our algorithm interactively constructs an undirected graph $G^{(\infty)}$ whose vertices are the vertices of G_d (though the edges of $G^{(\infty)}$ may not be the undirected equivalent of G_d).
- (**Degree-proportional jumps**) At PageRank, a random jump out of vertex v, $\forall v \in V$, is performed with probability w, independent of the degree of v. In our algorithm a random jump from vertex v, $\forall v \in V$, is performed with probability $w/(w + \deg(v))$, where $\deg(v)$ is the degree of v in $G^{(\infty)}$.

We denote our proposed random walk algorithm *random walk with jumps* (RWwJ). In what follows we detail our approach. The *backward edge traversal* is detailed in Section III-A and the *degree-proportional jump* is detailed in Section III-B.

A. Backward edge traversals

We allow the walker to traverse some outgoing edges backwards. In general, if we apply this "backward walking" principle to all outgoing edges in G_d, we can construct an undirected version of G_d. The undirected version of G_d allows us to apply the techniques described in [15] to estimate the characteristics of G_d such as the out-degree distribution. However, the degree of a vertex v, $\forall v \in V$, in the final undirected version of G_d is only known after exploring all edges of G_d. Thus, the steady state probability of sampling v also requires access to the complete underlying graph (as the probability is a function of v's degree [15]).

To avoid this problem, our RW interactively builds an undirected graph $G^{(\infty)}$. This building process is such that once a vertex is visited at the i-th step of the RW no more edges can be added to that vertex in subsequent steps. Such a restriction allows us to be certain of the degree of vertices visited by the RW, independent of the actual number of incoming edges to such vertices. Note that the final undirected graph $G^{(\infty)}$ depends on the sample path taken by the random walker. The undirected graph $G^{(\infty)} = (V, E^{(\infty)})$ is connected, undirected, and has the same vertices as G_d. Because $G^{(\infty)}$ is undirected and connected, we can estimate characteristics such as the degree distribution [15]. Based on the above design principle, we implement a "backward edge traversal" approach similar to the one described by Bar-Yossef [1]. The details of the algorithm are described in Section III-C.

The above solution addresses the problem of knowing the degree of a vertex as soon as the vertex is sampled. However, we still do not know the steady state distribution of the RW when we add random jumps. In what follows we present an algorithm that allows us to obtain a simple closed-form solution to the steady state distribution.

B. Degree-proportional jumps

Let $G = (V, E)$ be an undirected graph. In RWwJ, the probability of randomly jumping out of a vertex v, $\forall v \in V$, is $w/(w+\deg(v))$, $w > 0$. This modification is based on a simple observation: let G' be a weighted undirected graph formed by adding a vertex σ to G such that σ is connected to all vertices in V with edges having weight w. All remaining edges have unitary weight. In a weighted graph a random walk jumps walks over an edge with probability proportional to the edge weight. The steady state distribution of a vertex v, $\forall v \in V$, of a RW over G' is $(w + \deg(v))/(\text{vol}(V) + w|V|)$, where $\text{vol}(V) = \sum_{\forall u \in V} \deg(u)$. Thus, except for the unknown constant normalization term $(\text{vol}(V) + w|V|)$, the steady state distribution of v is known as we know the degree of v and the value of parameter w when v is visited by the random walker. By combining *backward edge traversal* (Section III-A) and *degree-proportional jumps* (Section III-B) we obtain the following algorithm, which we denote RWwJ.

C. The RWwJ algorithm

The RWwJ algorithm is a regular random walk over a weighted undirected connected graph $G^{(\infty)} = (V, E^{(\infty)})$, which is built on-the-fly. The algorithm works as follows. We build an undirected graph using the underlying directed graph G_d and the ability to perform random jumps. Let $G^{(i)} = (V^{(i)}, E^{(i)})$ be a tuple where $V^{(i)}$ is the vertex set and $E^{(i)}$ is the edge set at the i-th random walk step. The tuple $G^{(i)}$ is such that $\lim_{i\to\infty} G^{(i)} = G^{(\infty)}$, but $G^{(i)}$, $i < \infty$ is not necessarily a graph.

Let $v \in V$ be the initial vertex in the random walk. Let $\mathcal{N}(v)$ denote the outgoing edges of v. Let $G^{(1)} = (\{s_1\}, E^{(1)})$, where $E^{(1)} = \mathcal{N}(s_1) \cup \{(u, \sigma) : \forall u \in V\}$, where $\{(u, \sigma) : \forall u \in V\}$ is the set of all virtual edges to the virtual vertex σ (this construct is just to simplify our exposition, in practice we do not need to know all vertices or add the virtual edges to $G^{(i)}$). The random walker proceeds as follows.

We start with $i = 1$; at step i the random walker is at vertex s_i. Let

$$W(u, v) = \begin{cases} w & \text{if } u = \sigma \text{ or } v = \sigma \\ 1 & \text{otherwise} \end{cases}$$

denote the weight of edge (u, v), $\forall (u, v) \in E^{(i)}$, $i = 1, 2, \ldots$. The next vertex, s_{i+1}, is selected from $E^{(i)}$ with probability

$W(s_i, s_{i+1}) / \sum_{\forall (s_i,v) \in E^{(i)}} W(s_i, v)$. Upon selecting s_{i+1} we update $G^{(i+1)} = (V^{(i)} \cup \{s_{i+1}\}, E^{(i+1)})$, where

$$E^{(i+1)} = E^{(i)} \cup \mathcal{N}'(s_{i+1}), \quad (1)$$

and

$$\mathcal{N}'(s_{i+1}) = \{(s_{i+1}, v) : \forall (s_{i+1}, v) \in \mathcal{N}(s_{i+1}) \text{ s.t. } v \notin V^{(i)}\}$$

is the set of all vertices (u, v) in $\mathcal{N}(s_{i+1})$ where vertex v is not already in $V^{(i)}$. Note that $\mathcal{N}'(s_{i+1}) \subseteq \mathcal{N}(s_{i+1})$. By using $\mathcal{N}'(s_{i+1})$ instead of $\mathcal{N}(s_{i+1})$ in equation (1) we guarantee that no vertices in $V^{(i)}$ change their degrees, i.e., $\forall v \in V^{(i)}$ the degree of v in $G^{(i)}$ is also the degree of v in $G^{(\infty)}$. Thus, we comply with the requirement presented in Section III-A that once a vertex v, $\forall v \in V$, is visited by the RW no edges can be added to the graph with v as an endpoint.

The edges in $G^{(i)}, i = 1, 2, \ldots$, that connect all vertices to the virtual vertex σ can be easily emulated with uniform vertex sampling. The pseudo code of the RWwJ algorithm is shown in Algorithm 1, where c is the cost of randomly jumping (i.e., the average number of IDs queried until one valid ID is obtained), B is the sampling budget, w is the random jump weight (a quantity that influences the random jump probability), and s_1 is the starting vertex.

Space complexity: The space required to store $G^{(i)}$ is $O(K)$, where K is the number of distinct vertices observed by the random walker.

Algorithm 1: Random Walk with Jumps pseudo-code.

```
/* B is the sampling budget, s_1 ∈ V is the initial RW
   vertex, w is the random jump weight, and c is the
   cost of randomly jumping                              */
input : B, s_1 ∈ V, and w
output: m, s_1, s_2, ..., s_m
/* S is the set of sampled vertices                      */
S ← {};
/* E* is a set of undirected edges                       */
E* ← {};
i ← 1;
m ← 1;
while i < B − c do
    E* ← E* ∪
      {(s_m, u) : ∀u ∈ V s.t. u ∉ V* and (s_m, u) ∈ E};
    S ← S ∪ {s_m};
    /* U(0,1) is a uniform (0,1) random sample;
       deg(s_m, E*) returns the number of edges in E*
       with vertex s_m                                   */
    if U(0,1) < w/(w + deg(s_m, E*)) then
        s_{m+1} ← randomV(V);
        /* randomV(V) returns a vertex of V chosen
           uniformly at random                           */
        i ← i + c;
    else
        s_{m+1} ← randomNeighbor(s_m, E*);
        /* randomNeighbor(s_m, E*) returns a randomly
           chosen neighbor of vertex s_m among the
           (undirected) edges in E*                      */
        i ← i + 1;
    end
    m ← m + 1;
end
```

D. Out-degree Distribution Estimator

In this section we use the vertices visited (sampled) by our *random walk with jumps* algorithm to estimate the out-degree distribution [12]. The estimator presented in this section can be easily extended to obtain the distribution of vertex labels. For instance, if vertices can be labeled either red or blue, we can calculate the fraction of red and blue vertices in a graph if we can directly query if the vertex is red or blue. Out-degrees can be seen as a type of vertex label.

Let s_i denote the i-th edge visited by RWwJ, $i = 1, \ldots, B$. Let ϕ_j be the fraction of vertices with out-degree j. Let $\pi(v)$ be the steady state probability of sampling vertex v in $G^{(\infty)}$, $\forall v \in V$. The out-degree distribution can be estimated as

$$\hat{\phi}_j = \frac{1}{B} \sum_{i=1}^{B} \frac{h_j(s_i)}{\hat{\pi}(s_i)}, j = 0, 1, \ldots \quad (2)$$

where $h_j(v)$ is the indicator function

$$h_j(v) = \begin{cases} 1 & \text{if the out-degree of } v \text{ in } G_d \text{ is } j \\ 0 & \text{otherwise} \end{cases},$$

and $\hat{\pi}(s_i)$ is an estimate of $\pi(s_i)$: $\hat{\pi}(s_i) = (w + \deg(s_i))S$. Here $\deg(v)$ is the degree of v in $G^{(\infty)}$ and

$$S = \frac{1}{B} \sum_{i=1}^{B} \frac{1}{w + \deg(s_i)}.$$

To show that $\hat{\pi}(s_i)$ is an asymptotically unbiased estimate of $\pi(s_i)$ we invoke Theorem 4.1 of [15], which yields $\lim_{B\to\infty} S = |V|/(|E^{(\infty)}| + |V|w)$, almost surely, and thus $\lim_{B\to\infty} \hat{\pi}(s_i) = \pi(s_i)$, almost surely. Taking the expectation of equation (2) in the limit $B \to \infty$ yields $E[\lim_{B\to\infty} \hat{\phi}_j] = \phi_j$.

E. Experimental Results

This section compares the out-degree distribution estimates obtained with RWwJ against the ones obtained with independent uniform vertex sampling (UNI). Our experiments are performed over the Flickr graph. Our Flickr graph dataset [11] consists of 1.6 millions vertices and 22 millions of edges. Let $\Phi_j = \sum_{\forall i > j} \phi_i$ be the fraction of vertices with out-degree larger than j and $\hat{\Phi}_j$ be an estimate of Φ_j.

We begin with a comparison of the estimates $\hat{\Phi}_{10}$ and $\hat{\Phi}_{10000}$ (the choice of out-degrees 10 is 10000 is arbitrary) obtained using RWwJ and UNI. Figures 2 and 3 show $\hat{\Phi}_{10}$ and $\hat{\Phi}_{10000}$ as a function of the number of sampled vertices, respectively. Note that the transient of RWwJ can be long; we comment on this transient in Section III-E. Let c denote the cost of UNI which is also the cost of a random jump

(the average number of IDs queried until one valid ID is obtained). Let B denote the sampling budget (when $c = 1$, B is the number of sampled vertices). Recall that in RWwJ the probability of performing a random jump increases with w (such that in the limit $w \to \infty$ RWwJ is equivalent to UNI). We see that UNI and RWwJ with large w are better suited to estimate Φ_{10} than RWwJ is with w small. On the other hand, for $B < 0.1|V|$, RWwJ with small w is more accurate at estimating Φ_{10000} than UNI or RWwJ with large w.

Transient Bias: Unfortunately, the transient of the RWwJ algorithm can be quite long. Say vertex $v \in V$ has a large number of hidden incoming edges. The numerous incoming edges increase the probability that v is sampled in the beginning of the walk. However, once v is sampled the degree of v in $G^{(\infty)}$ may be small (as only a subset of the incoming edges belong to $G^{(\infty)}$). With a small degree in $G^{(\infty)}$, the probability that v is subsequently sampled is small which makes the first sample an outlier. Eventually, as RWwJ progresses, this initial outlier plays a diminishing role over the statistics being computed. However, in practice, outliers can significantly increase estimation errors even with moderately large number of RWwJ steps. Thus, our estimator throws away the first αB RWwJ samples, $\alpha < 1$. In the following results we use $\alpha = 0.1$ as we found it to be a good compromise between getting rid of outliers and keeping enough samples to accurately estimate the out-degree distribution.

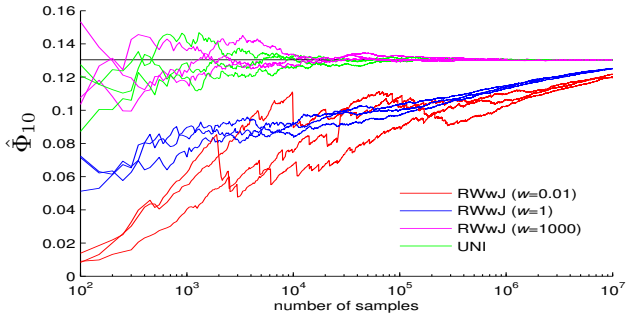

Figure 2. (Flickr) Three sample paths showing the estimates of Φ_{10} (true value is 0.13) using RWwJ with jump weights $w = 0.01, 1, 1000$ and independent vertex sampling. The cost of independent vertex sampling is $c = 1$.

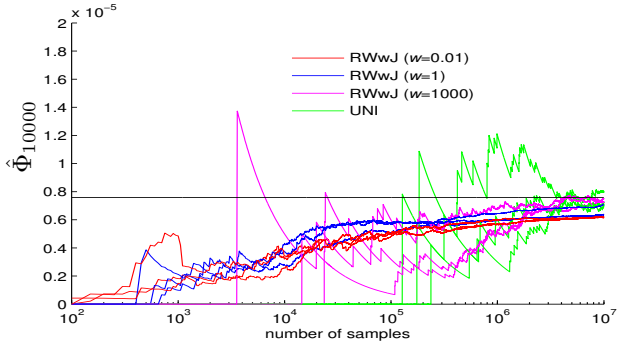

Figure 3. (Flickr) Three sample paths showing the estimates of Φ_{10000} (true value is 7.58×10^{-6}) using RWwJ with jump weights $w = 0.01, 1, 1000$ and independent vertex sampling. The cost of independent vertex sampling is $c = 1$.

Estimation Error: However, the comparison between RWwJ and UNI shown in Figures 2 and 3 is not fair as we assume that $c = 1$ and, in the real world, Flickr has a random vertex sampling cost of $c = 77$ (as observed in the experiments presented in this section). Let

$$\text{CNMSE}(\hat{\Phi}_j) = \frac{\sqrt{E[(\hat{\Phi}_j - \Phi_j)^2]}}{\Phi_j}, j = 1, 2, \dots,$$

be a metric that measures the relative error of the estimate $\hat{\Phi}_j$ in respect to its true value Φ_j. Figure 4 shows an estimate of the CNMSE (over 1000 runs) with $c = 77$ and sampling budget $B = 0.01|V|$. The advantage of small values of w in estimating large out-degrees is more noticeable with $c = 77$. Still, UNI and large values of w are clearly more accurate at estimating small out-degrees. But note that when $w = 0.01$, RWwJ performs slightly worse than RWwJ when $w = 1$ for all out-degrees. This means that when estimating large out-degrees we should keep w small without impairing the walker's ability to randomly jump. Our results show that RWwJ is more efficient in estimating the out-degree distribution when vertex sampling is expensive. Moreover, with RWwJ we can control the cost of vertex sampling by tuning w.

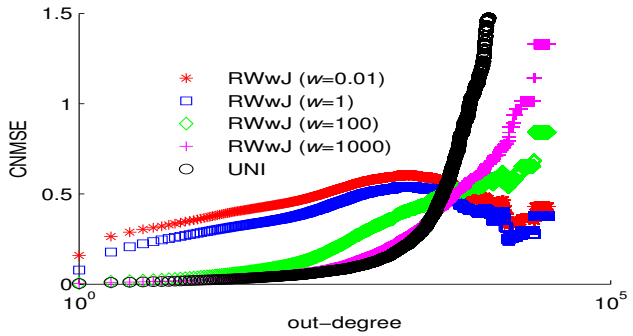

Figure 4. (Flickr) CNMSE of RWwJ with $w = 0.01, 1, 100, 1000$ and UNI, $c = 77$ and $B = 0.01|V|$.

Here we look at the impact of the sampling budget, B, on the estimation accuracy of the out-degree distribution. Figure 5 presents the CNMSE of Flickr for budgets $B \in \{0.01|V|, 0.1|V|, |V|\}$ with $c = 1$ and $w = 0.01$. We observed that one order of magnitude increase in B reduces the error roughly by half. We also study the impact of the cost of uniform vertex sampling over the accuracy of the estimates. Figure 6 shows the CNMSE with $c = 1, 10$, and 10, $B = 0.01|V|$ and $w = 100$. Unsurprisingly, we observe that the estimation error of the out-degree distribution tail increases with c.

We also estimated the out-degree distribution of the Livejournal graph. The Livejournal dataset [11] consists of 5 million vertices and 77 million edges of a blog social network. Figures 7 and 8 compare RWwJ (with $w \in \{0.01, 1, 1000\}$) against UNI using the estimates of Φ_{10} and Φ_{722} plotted as function of the number of sampled vertices (B), respectively. These experiments are unrealistic (and favorable to UNI) as

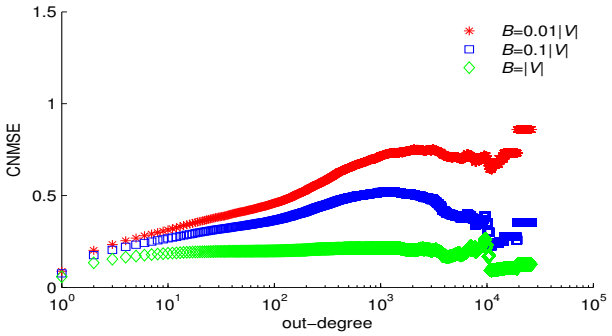

Figure 5. (Flickr) CNMSE for varying budget B with $c = 1$ and $w = 0.01$

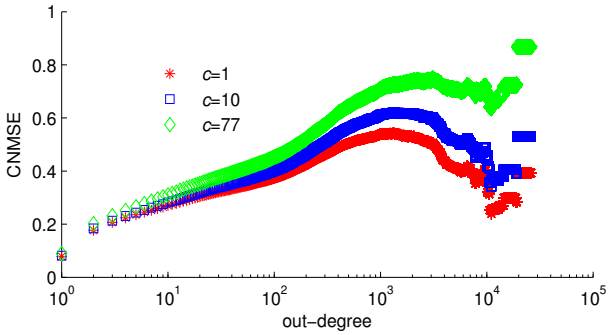

Figure 6. (Flickr) CNMSE for varying cost c with $B = 0.01|V|$ and $w = 100$

we assume $c = 1$. We observe that results are similar to the ones in Figures 2 and 3, i.e., we see that UNI and RWwJ with large w are better suited to estimate Φ_{10} than RWwJ is with w small. On the other hand, for $B < 0.1|V|$, RWwJ with small w is more accurate at estimating Φ_{10000} than UNI or RWwJ with large w.

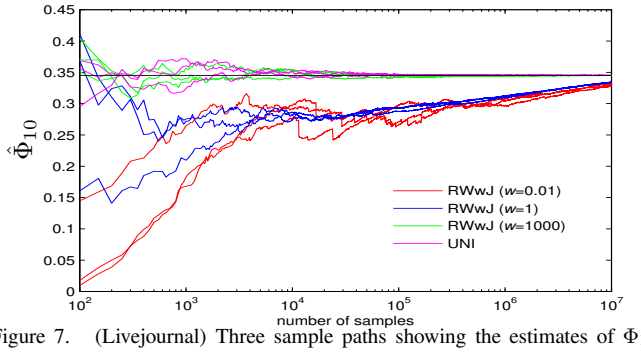

Figure 7. (Livejournal) Three sample paths showing the estimates of Φ_{10} (true value is 0.35) using RWwJ with jump weights $w = 0.01, 1, 1000$ and independent vertex sampling. The cost of independent vertex sampling is $c = 1$.

IV. Estimating Latent In-degree Distributions

The above approach, used to estimate the out-degree distribution, can also be used to estimate the in-degree distribution if in-degrees are visible to the random walker. However, in this section we consider a much harder problem: estimating the in-degree distribution when in-degrees are hidden. Unfortunately, our results are negative. We show that in the presence of hidden incoming edges one needs to sample most

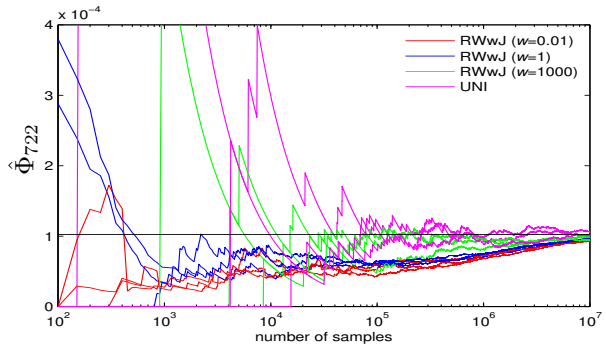

Figure 8. (Livejournal) Three sample paths showing the estimates of Φ_{722} (true value is 10^{-4}) using RWwJ with jump weights $w = 0.01, 1, 1000$ and independent vertex sampling. The cost of independent vertex sampling is $c = 1$.

of the edges of the graph in order to obtain an accurate in-degree distribution estimate. Here the in-degree distribution is an example of a latent graph characteristic. A latent graph characteristic is one that cannot be directly observed but is rather inferred (through a mathematical model) from other observable variables. The in-degree distribution can be inferred by independently sampling edges in the graph. For instance, in the Flickr graph if user b subscribes to user a's photo updates, then the graph has a directed edge (b, a). Now consider estimating the distribution of the number of subscribers *per account* (in-degree) in the Flickr photosharing network by randomly sampling edges. Let i be the in-degree of a given user a and let X be a random variable that denotes the number of sampled incoming edges of a if edges are sampled independently and with probability p. It is easy to see that

$$P[X = j] = b(j, i) = \binom{i}{j} p^j (1-p)^{i-j}, \; j = 1, 2, \ldots \quad (3)$$

with $b(j, i) = 0$, $\forall j > i$. Equation (3) provides a model from which the in-degree distribution can be inferred. Another similar example is estimating the distribution of the number of friends (neighbors) of private Facebook profiles by randomly sampling Facebook friendships (edges).

However, independently sampling edges is a difficult task. For instance, neither Flickr nor Facebook provide public interfaces to sample edges and rejection sampling can be quite inefficient. However, there are other ways to perform uniform edge sampling. For instance, a random walk over an undirected graph samples edges uniformly at random (but not independently) [15]. Another way to sample hidden incoming edges in a graph is to sample vertices and all of their outgoing edges (edge samples may not be independent). One can also monitor the traffic of an OSN in order to observe edges of the friendship graph [2]. Our model is not tied to any specific sampling method. Rather, we consider an optimistic model where we can sample edges independently with probability p. We say that this model is optimistic because, in practice, dependency often increases estimation errors (e.g., in a RW, the Mean Squared Error (MSE) of observed (not latent) variables is consistently larger than the MSE of independent edge sampling [15]). In this section we provide a tight lower bound

on the MSE of the in-degree distribution for independent edge sampling, which we call MSELB.

A. Model

We use the Flickr graph to exemplify our uniform edge sampling model. The following model is not intended to be an accurate depiction of a real world scenario. Rather, it is optimistic, assuming that edges are sampled independently. In what follows we see that, even under our unrealistic assumptions, no unbiased estimator can accurately compute the in-degree distribution without sampling most of the edges in the graph.

Consider the Flickr graph, a directed graph $G_d = (V, E_d)$ where vertices, V, are users and edges, E_d, represent user subscriptions to updates from other users. Recall that the in-degree is a latent variable in this graph. Assume that we observe an edge e, $\forall e \in E_d$, with probability p (i.e., with probability p we see a user subscription). Further, assume edges are sampled independently. In what follows we present an estimator for the in-degree distribution.

B. Observed in-degree distribution

In this section we find the relationship between the observed in-degree distribution (obtained from the sampled edges) and the true in-degree distribution. We say an edge is *observed* if it is sampled. In a sampled edge (u, v) we observe two vertices u and v. But if we were to estimate the in-degree distribution using u as a sample, we would need to know the dependence between the out-degree of u (which defines the probability by which u is sampled) and the in-degree of u. A much simpler estimator can be built considering just v (the "destination" vertex). As edges are sampled independently, the probability that a vertex v with i incoming edges has j of them sampled is $b(j, i)$ (equation (3)). Note that to be observed a vertex must have at least one sampled edge. In this case, the probability that j incoming edges of v are sampled given that v has at least one sampled incoming edge is

$$b'(j, i) = b(j, i)/(1 - b(0, i)) .$$

Let $\theta = (\theta_1, \theta_2, \ldots, \theta_W)$ be the in-degree distribution of G_d, and let $d = (d_1, d_2 \ldots, d_W)$ be the observed in-degree distribution, and W be the largest in-degree. The following equation relates the observed in-degree distribution d with the true in-degree distribution θ:

$$d = B\theta^{\mathsf{T}} , \tag{4}$$

where $B = [b'(j, i)]$ is a $W \times W$ matrix whose element (j, i) is $b'(j, i)$.

C. Mean squared error lower bound (MSELB)

This section presents a lower bound on the mean squared error of any such unbiased estimator. In Section IV-E we use this lower bound to show that an accurate estimate of the in-degree distribution of Flickr or Facebook requires that we sample edges with high probability (e.g., $p = 0.9$). In doing so we make extensive use of the Fisher information. The Fisher information can be thought of as the amount of information that a set of observable samples, d, carry about unobservable parameters, θ, upon which the probability distribution of the samples depends. The following closely follows the exposition in [16], which estimates the flow size distribution from sampled packets.

The likelihood function f of one sampled vertex associated with j sampled incoming links is

$$f(j \mid \theta) = d(j) , \tag{5}$$

where $d(j)$ is the j-th element of d in equation (4). The unconstrained Fisher information [16] is a matrix $J = [J_{i,k}]$ where

$$J_{i,k} \triangleq \sum_{\forall j} \frac{\partial \ln f(j \mid \theta)}{\partial \theta_i} \cdot \frac{\partial \ln f(j \mid \theta)}{\partial \theta_k} d(j).$$

Equations (4) and (5) yield $\partial \ln f(j \mid \theta)/\partial \theta_i = b'(j, i)/d(i)$. Thus,

$$J = BDB^{\mathsf{T}} , \tag{6}$$

where D is a diagonal matrix whose element (j, j) is $D_{j,j} = 1/d(j)$. With the likelihood function it is trivial to build a Maximum Likelihood Estimator (MLE).

Equation (6) gives the Fisher information of *one* sampled vertex. A vertex v is sampled if v has at least one sampled incoming edge. Let p be the edge sampling probability and S be the set of sampled vertices. Since each sampled edge is selected independently, vertices in S are also sampled independently. The Fisher information of S is then the sum of the Fisher information of each of the samples [4]. Let N be a random variable that denotes the number of distinct vertices sampled.

The most notable property of the Fisher information is a bound on the accuracy of estimators. Let T be an unbiased estimator of θ ($E[T(S)] = \theta$). The Cramér-Rao theorem states that the mean squared error of any unbiased estimator T is lower bounded by the inverse of the Fisher information, provided some weak regularity conditions [20], i.e.,

$$E[(T(S)_j - \theta_j)^2] \geq I_{j,j}, \tag{7}$$

where $I = J^{-1}/N$ and J^{-1} is the inverse of the Fisher information matrix. We refer to the lower bound in equation (7) as the *MSELB*. Thus the MSELB is

$$E[I] = J^{-1}E[1/N] = \\ J^{-1} \sum_{n_1=0}^{|V|\theta_1} \cdots \sum_{n_W=0}^{|V|\theta_W} \frac{\prod_{i=1}^{W} \binom{|V|\theta_i}{n_i} q_i^{n_i} (1 - q_i)^{|V|\theta_i - n_i}}{\sum_{i=1}^{W} n_i} ,$$

where $\sum_{i=1}^{W} n_i \neq 0$, $q_i = 1 - (1 - p)^i$, and the number of vertices with degree i is $|V|\theta_i$.

Constraints on the estimated parameters provide information to the estimator and can increase the Fisher information content of the samples. As θ is a distribution, we have the following constraints,

$$0 < \theta_i < 1, \ \forall i \in \{1, ..., W\} \quad \text{and} \tag{8}$$

$$\sum_{i=0}^{W} \theta_i = 1. \tag{9}$$

If the equality constraint in equation (9) is active [18]

$$I = (J^{-1} - \theta\theta^{\mathsf{T}})/N\,. \tag{10}$$

In our experiments we observe that the diagonal elements of $\theta\theta^{\mathsf{T}}/N$ are negligible in respect to the diagonal elements of J^{-1}/N when $p < 0.9$, and, thus, we conclude from equations (7) and (10) that the equality constraint in equation (9) does not significantly increase the estimation accuracy when $p < 0.9$. Our Fisher information calculations also ignore the inequality constraints in equations (8) (they are not crucial if we are dealing with unbiased estimators). In [16] the reader finds the necessary treatment to include equations (8) in our Fisher information calculations.

The inverse of J is a crucial step at calculating the MSELB. J^{-1} can be written $J^{-1} = B^{-1}D^{-1}(B^{-1})^{\mathsf{T}}$, where B^{-1} is given in the following lemma.

Lemma 4.1: $B^{-1} = [b^\star(j,i)]$, where

$$b^\star(j,i) = \binom{i}{j} p^{-i}(p-1)^{i-j}\left(1-(1-p^j)\right), i \geq j$$

and $b^\star(j,i) = 0$, $i < j$.

Proof: Let $B^{-1} = [b^\star(j,i)]$ be as defined in Lemma 4.1. We need to show that Y, $Y = BB^{-1}$, is an identity matrix. Consider element (j,i) of Y:

$$y_{j,i} = \sum_{l=1}^{W} b'(j,l)b^\star(l,i)\,. \tag{11}$$

Let's divide $y_{j,i}$ into three distinct cases: $j > i$, $j = i$, and $j < i$. Note that the definition of b yields $b'(h,k) = 0, \forall h > k$. If $j > i$ equation (11) yields $y_{j,i} = 0$ as $b'(j,l) = 0$, $\forall l \leq i$ and $b^\star(l,i) = 0$, $\forall l > i$. If $j = i$, then $b'(j,l)b^\star(l,j) = 0$, $\forall l \neq j$ and, thus, equation (11) yields

$$y_{jj} = \frac{p^j}{1-(1-p)^j} p^{-j}(1-(1-p)^j) = 1\,.$$

If $j < i$, equation (11) yields

$$\begin{aligned}
y_{ji} &= \sum_{l=j}^{i} (-1)^{i-l} p^{j-i}(1-p)^{i-j} \binom{l}{j}\binom{i}{l} \\
&= p^{j-i}(1-p)^{i-j} \sum_{l=j}^{i} (-1)^{i-l} \binom{i}{j}\binom{i-j}{l-j} \\
&= p^{j-i}(1-p)^{i-j} \binom{i}{j} \sum_{l=j}^{i} (-1)^{i-l} \binom{i-j}{l-j} \\
&= p^{j-i}(1-p)^{i-j} \binom{i}{j} (1-1)^{i-j} \\
&= 0
\end{aligned}$$

Thus, $y_{j,j} = 1$, $\forall j$ and $y_{j,i} = 0$, $\forall j \neq i$, which concludes our proof. ∎

In what follows we assume that the out-degree distribution is known. By assuming we know the out-degree distribution we are also assuming that we know the average in-degree, as the average in- and out-degrees are the same. Unfortunately, our results show no significant gain in accuracy when the average in-degree is used as side information.

In a graph where all edges are symmetric (i.e., every edge $(u,v) \in E_d$ has a corresponding edge $(v,u) \in E_d$) the in-degree and the out-degree distributions are the same. Thus, in a graph with a large fraction of symmetric edges, one expects to be able to shift the information regarding the out-degree distribution to the in-degree distribution. In what follows we consider graphs that are highly symmetric, i.e., most edges $(u,v) \in E_d$ have a corresponding edge $(v,u) \in E_d$. In Section IV-E we see that, unless almost all edges are symmetric, graph symmetry has a little impact on the accuracy of the estimator.

D. Symmetric Edge Information

Consider a directed graph $G_d = (V, E_d)$. An edge $(u,v) \in E_d$ is said to be symmetric if $(v,u) \in E_d$. Let s denote the fraction of symmetric edges in E_d, where $s = 1$ when all edges in G_d are symmetric. Edge symmetry can convey information about the in-degree distribution. For instance, if $s = 1$ the in-degree distribution equals to the out-degree distribution. To assess the increase in estimation accuracy that can come from the presence of symmetric edges, consider the following model.

Let v be a sampled vertex. Consider the following random variables of v:

- Z: in-degree of v.
- Z_s: number of symmetric incoming edges.
- Z_a: number of incoming asymmetric edges.
- Y: observed out-degree.
- X_s observed number of symmetric incoming edges.
- X_a observed number of asymmetric incoming edges.

Also, let $\rho(y,z) = P[Y = y, Z = z]$ be the joint in-degree and out-degree distribution of v, p be the sampling rate, and α be the fraction of symmetric edges. We assume that the number of outgoing edges of v that are symmetric is a Bernoulli random variable with parameter α and has distribution

$$P[Z_s = z_s | Y = y, Z = z] = \begin{cases} \binom{\min(y,z)}{z_s} \alpha^{z_s}(1-\alpha)^{\min(y,z)-z_s} & \text{if } z_s \leq \min(y,z), \\ 0 & \text{otherwise.} \end{cases} \tag{12}$$

We seek to find a likelihood function of the observed random variables Y, X_s, and X_a in respect to ρ, $P[Y = y, X_s = x_s, X_a = x_a|\rho]$. Note that

$$\begin{aligned}
&P[Y = y, X_s = x_s, X_a = x_a|\rho] \\
&= \sum_{\forall z} P[X_s = x_s, X_a = x_a | Y = y, Z = z]\rho_{y,z} \\
&= \sum_{\forall z} \rho_{y,z} \sum_{z_s=0}^{z} P[X_s = x_s, X_a = x_a | Z_s = z_s, Y = y, Z = z] \times \\
&P[Z_s = z_s | Y = y, Z = z],
\end{aligned}$$

where

$$
\begin{aligned}
&P[X_s = x_s, X_a = x_a | Z_s = z_s, Y = y, Z = z] \\
&= P[X_s = x_s, X_a = x_a | Z_s = z_s, Y = y, Z_a = z - z_s] \\
&= \binom{z_s}{x_s} p^{x_s} (1-p)^{z_s - x_s} \binom{z - z_s}{x_a} p^{x_a} (1-p)^{z - z_s - x_a} \\
&= \binom{z_s}{x_s} \binom{z - z_s}{x_a} p^{x_s + x_a} (1-p)^{z - x_s - x_a}
\end{aligned}
$$

with $P[Z_s = z_s | Y = y, Z = z]$ as defined in equation (12).

The in-degree distribution Fisher information associated with the symmetric edge information can be computed from the Fisher information of $P[Y = y, X_s = x_s, X_a = x_a | \rho]$ with respect to ρ by noting that θ, the in-degree distribution, can be defined as $\theta_z = \sum_{\forall y} \rho(y, z), \forall z$, or in matrix form $\theta = H\rho^{\mathsf{T}}$, where $\rho = (\rho(1,1), \rho(2,1), \dots)$ and

$$
H = \begin{bmatrix}
1 & \dots & 1 & & & & & & \\
 & & & 1 & \dots & 1 & & & \\
 & & & & & & \dots & & \\
 & & & & & & & 1 & \dots & 1
\end{bmatrix}.
$$

Let J_ρ denote the Fisher information with respect to the joint distribution ρ. Computing J_ρ from $P[Y = y, X_s = x_s, X_a = x_a | \rho]$ is an easy task. Let J_θ denote the Fisher information with respect to the in-degree distribution θ. Then [20, pages 83–84]

$$J_\theta = H J_\rho H^{\mathsf{T}}.$$

The above equations allow us to compute the information obtained from knowing that the graph is highly symmetric. In the results presented in Section IV-E we observe that adding symmetric information does not significantly improve the estimation error unless most edges in the graph are symmetric.

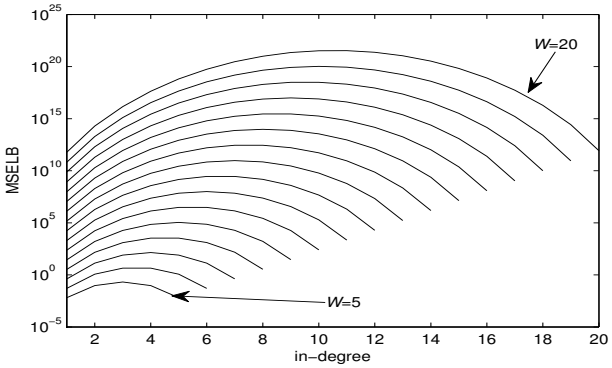

Figure 9. (Flickr) MSELB with $W = 5, \dots, 14$ and $p = 0.1$.

E. Experimental Results

This section studies the "hardness" of estimating the in-degree distribution of Flickr and Facebook using independently sampled edges. The Flickr dataset [11] consists of 1.5 millions vertices and 17 millions of edges and the Facebook dataset [7] has 1 million vertices and 92 millions of edges. Let W be the maximum in-degree in the graph. Both Facebook and Flickr have vertices with more than 10^4 incoming edges, i.e., $W > 10^4$ and recall that the Fisher information is a $W \times W$ matrix. In order to reduce the computational effort to compute the MSELB (and keep numerical errors under control) we restrict $W \in \{100, 50, 20\}$. Our experiments show that the MSELB increases with W, which means that the results for $W > 100$ should be worst than the (already negative) results we show in this section. In order to evaluate the impact of choosing a small value of W over the MSELB, Figure 9 uses the MSELB with $p = 0.1$ for different values of W over the Flickr graph. We observe that the MSELB roughly increases one order of magnitude when W is incremented by 1, which indicates that our MSELB (taken for small values of W) is optimistic when compared to the MSELB with the true value of W ($W > 10^4$).

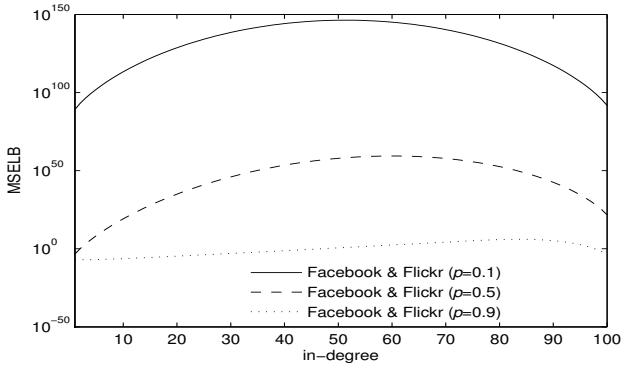

Figure 10. MSELB with $p = 0.1, 0.5,$ and 0.9.

We begin by showing the MSELB of Flickr and Facebook for $W = 100$ and $p = 0.1, 0.5,$ and 0.9 in Figure 10. The Flickr and Facebook curves overlap for each different value of p. In Figure 10 we observe that the MSELB is high for all out-degrees when $p = 0.1$ (and the MSELB is high for most out-degrees when $p = 0.5$). This means that unbiased estimates are inaccurate when, on average, 10% or even 50% of the graph edges are sampled. Also note that the MSELB is low when $p = 0.9$, which means that accurate unbiased estimates can be obtained when (on average) 90% of the graph edges are sampled.

In the following experiment we include symmetry information in the MSELB (Section IV-D). Figure 11 shows the results of Flickr for $W = 20$, with and without Flickr's symmetric information. In Flickr the fraction of edges that are symmetric is $\alpha = 0.62$. Observe that while symmetry reduces the MSELB, it is not enough to significantly increase the estimation accuracy.

In practice the MSELB defines a bound on what we can expect from our experiments. However, different than in the MSELB, the inequality constraints in equations (8) play a significant role at keeping the MSE bounded, but they also bias the estimator (and in practice offer little help at obtaining correct estimates). In what follows we estimate the in-degree distribution of Flickr by sampling edges of the Flickr graph (independently with probability p) and then counting the number of incoming edges (observed in-degree) of each sampled vertex. Using a MLE over the observed in-degrees we estimate the original in-degree distribution. According to the MSELB results (shown in Figure 10) no unbiased

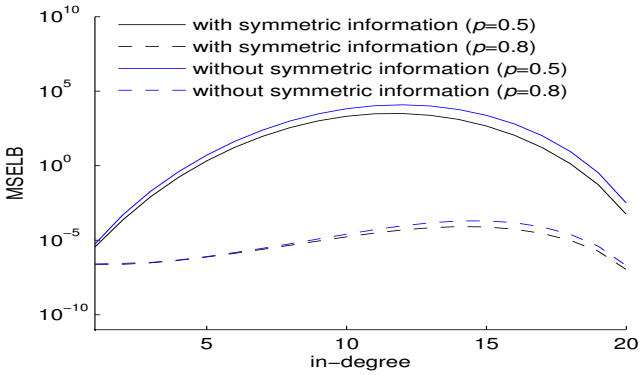

Figure 11. (Flickr) MSELB with and without symmetric edge information.

estimator can obtain an accurate estimate of the distribution using sampling probabilities $p = 0.1$ and $p = 0.5$.

The following results are optimistic as we limit our estimator to $W = 50$ (i.e., we remove vertices with more than 50 incoming edges from the graph). Figure 12 plots the estimates for $p = 0.1, 0.5,$ and 0.9. Observe that while the estimates with an average of 90% of the edges sampled ($p = 0.9$) are reasonable (but not accurate at the tail, though), the estimates with $p = 0.1$ and $p = 0.5$ are all over the place (as predicted by the MSELB). But how far are the MSE of the MLE estimates in respect to the MSELB? Figure 13 shows an extreme case where, on average 99% of the edges are sampled ($p = 0.99$) with $W = 50$. Observe that only the error of the tail estimates get close to the MSELB.

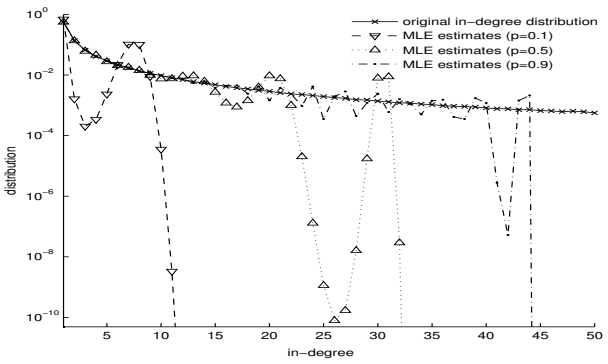

Figure 12. (Flickr) MLE in-degree distribution estimates for $p = 0.1, 0.5,$ and 0.9.

V. Related Work

To the best of our knowledge our work is the first to study and provide a sound theoretical analysis of the problem of estimating latent in-degree distributions. Regarding estimating observable characteristics, sampling a directed graph (in this case, the Web graph) has been the subject of [1] and [9], which transform the directed graph of web-links into an undirected graph by adding reverse links, and then use a Metropolis-Hastings RW to sample webpages uniformly. However, as the Web graph does not allow random jumps, these algorithms are unable to sample all vertices. On the other hand, our RWwJ algorithm samples the entire graph thanks to the ability

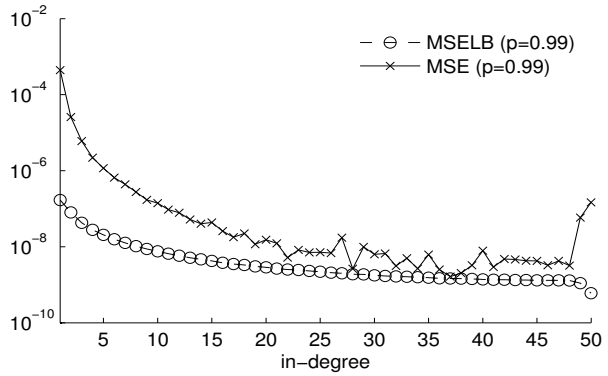

Figure 13. (Flickr) MSE of MLE in-degree distribution estimates with $p = 0.99$

to perform random jumps. Random walks with PageRank-style jumps are used in [10] to sample large graphs. In [10], however, the estimates presented in are highly biased and the authors do not present a technique to remove such bias. In contrast, our out-degree distribution estimates are asymptotically unbiased.

VI. Conclusions & Future Work

In this work we presented a random walk algorithm that can estimate the out-degree distribution of a directed graph when random jumps are allowed. Our algorithm is better suited to estimate the tail of the out-degree distribution than uniform vertex sampling. Because random vertex sampling can be expensive, our algorithm has a parameter w that controls the probability of performing a random jump. By tuning w we transition between pure uniform vertex sampling and a pure random walk with no jumps. We also study the problem of estimating latent in-degree distributions. We show that accurate unbiased in-degree distribution estimates require sampling almost all of the edges in the Flickr and in the Facebook graphs. We also show that the extra information obtained from the symmetric edges in the Flickr graph does not significantly increase estimation accuracy. Our future work includes reducing the transient of our RWwJ algorithm.

Acknowledgment

We would like to thank Alan Mislove and Minas Gjoka for kindly making available the datasets we used in our experiments. This research was sponsored by the ARO under MURI W911NF-08-1-0233 and the U.S. Army Research Laboratory and the U.K. Ministry of Defence and was accomplished under Agreement Number W911NF-06-3-0001. The views and conclusions contained in this document are those of the author(s) and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Army Research Laboratory, the U.S. Government, the U.K. Ministry of Defence, or the U.K. Government. The U.S. and U.K. Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

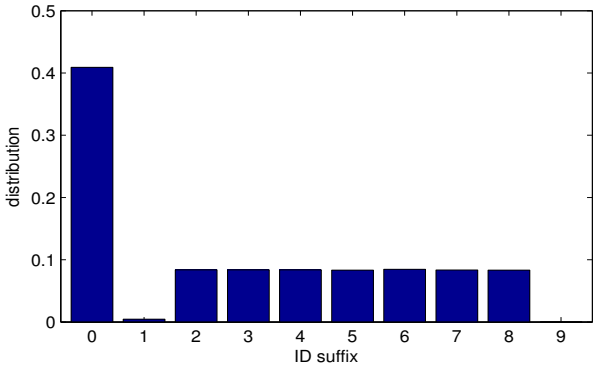

Figure 15. Flickr ID suffix distribution.

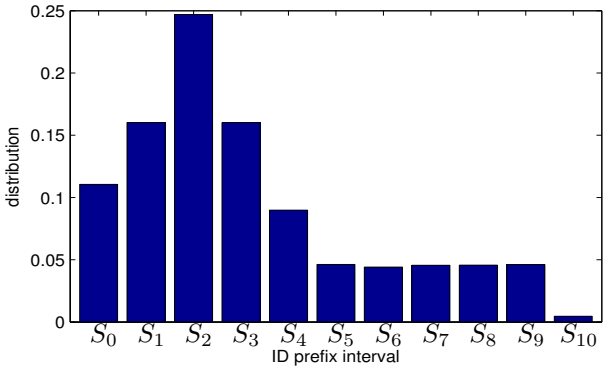

Figure 14. Flickr ID prefix distribution.

APPENDIX

The cost of uniformly sampling Flickr vertices

Flickr assigns a numeric ID to each user. The Flickr ID consists of a numeric eleven digit prefix number followed by "@N", followed by a two digit suffix number [19]. However, Filckr has fewer users than a 11+2 digit numeric ID suggests (10^{13} is the approximate theoretical maximum number of users of Flickr). Thus, we first assess how IDs are dispersed over the ID space. In what follows we say an ID is valid if it has at least one outgoing link. We start our sampling from 100 random (but valid) IDs and crawl the all their visible outgoing links until 85000 distinct IDs are collected. Note that all visited IDs (module the 100 seed IDs) have at least one incoming edge. The distribution of ID prefixes and suffixes observed in our experiment are shown in Figures 14 and 15. To plot Figure 14 we split ID prefixes into bins $S_i = [i \times 10^7, (i+1) \times 10^7)$ $(0 \leq i < 10)$ and $s_{10} = [10^8, \infty)$. Observed that nearly 90% ID prefixes are in the interval $[10^7, 10^8)$. We also observe that the suffix is a number in the interval $[0, 8]$ (Figure 15).

Our experiment consists querying 16000 numeric IDs with prefixes uniformly sampled from the interval $[10^7, 10^8)$ and with suffixes uniformly sampled in the interval $[0, 8]$. We restrict the prefixes to the interval $[10^7, 10^8)$ in order to increase our hit-to-miss ratio (thus, the actual cost of randomly sampling IDs is likely to be higher). Querying all 16000 IDs took twelve hours and only 206 valid IDs were found, i.e., in average only one valid ID is obtained from every 77 queries. Thus, we say that the cost of uniformly querying Flickr IDs is $c = 77$.

REFERENCES

[1] Ziv Bar-Yossef and Maxim Gurevich. Random sampling from a search engine's index. *J. ACM*, 55(5):1–74, 2008.

[2] Fabrício Benevenuto, Tiago Rodrigues, Meeyoung Cha, and Virgílio Almeida. Characterizing user behavior in online social networks. In *Proc. of the IMC*, pages 49–62, New York, NY, USA, 2009. ACM.

[3] S. Brin and L. Page. The anatomy of a large-scale hypertextual Web search engine. In *Proc. of the WWW*, 1998.

[4] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. John Wiley & sons, 1991.

[5] Facebook. http://www.facebook.com, 2010.

[6] Flickr. http://www.flickr.com, July 2010.

[7] Minas Gjoka, Maciej Kurant, Carter T. Butts, and Athina Markopoulou. A walk in Facebook: Uniform sampling of users in online social networks. In *Proc. of the IEEE Infocom*, March 2010.

[8] Douglas D. Heckathorn. Respondent-driven sampling: A new approach to the study of hidden populations. *Social Problems*, 1997.

[9] Monika R. Henzinger, Allan Heydon, Michael Mitzenmacher, and Marc Najork. On near-uniform url sampling. In *Proceedings of the WWW*, pages 295–308, 2000.

[10] Jure Leskovec and Christos Faloutsos. Sampling from large graphs. In *Proc. of the KDD*, pages 631–636, 2006.

[11] Alan Mislove, Massimiliano Marcon, Krishna P. Gummadi, Peter Druschel, and Bobby Bhattacharjee. Measurement and Analysis of Online Social Networks. In *Proc. of the IMC*, October 2007.

[12] M. E. J. Newman. Assortative mixing in networks. *Phys. Rev. Lett.*, 89(20):208701, Oct 2002.

[13] Amir H. Rasti, Mojtaba Torkjazi, Reza Rejaie, Nick Duffield, Walter Willinger, and Daniel Stutzbach. Respondent-driven sampling for characterizing unstructured overlays. In *Proc. of the IEEE Infocom*, pages 2701–2705, April 2009.

[14] Bruno Ribeiro, William Gauvin, Benyuan Liu, and Don Towsley. On MySpace account spans and double Pareto-like distribution of friends. In *Proceedings of the IEEE Infocom NetSciCom Workshop*, 2010.

[15] Bruno Ribeiro and Don Towsley. Estimating and sampling graphs with multidimensional random walks. In *Proc. of the IMC*, 2010.

[16] Bruno Ribeiro, Don Towsley, Tao Ye, and Jean Bolot. Fisher information of sampled packets: an application to flow size estimation. In *Proc. of the IMC*, pages 15–26, 2006.

[17] D. Stutzbach, R. Rejaie, N. Duffield, S. Sen, and W. Willinger. On unbiased sampling for unstructured peer-to-peer networks. *IEEE/ACM Trans. Netw.*, 17(2):377–390, 2009.

[18] Paul Tune and Darryl Veitch. Towards optimal sampling for flow size estimation. In *Proc. of the IMC*, pages 243–256, 2008.

[19] Masoud Valafar and Reza Rejaie. Beyond friendship graphs:a study of user interactions in flickr. In *Proc. of the ACM WOSN*, August 2009.

[20] Hary L. van Trees. *Estimation and Modulation Theory, Part 1*. Wiley, New York, 2001.