Estimating Distributions of Large Graphs from Incomplete Sampled Data

Shiju Li Xin Huang Chul-Ho Lee

***Abstract*—We study the problem of how to estimate the latent in-degree distribution of large directed graphs from random samples, when the samples only indicate the presence of partial incoming edges into nodes and thus their sampled distribution is far from the original one. While this problem can be cast as an inverse problem, it often appears to be ill-posed and leads to poor estimation performance. There have thus been few recent studies to overcome this problem, which include a constrained, penalized weighted least squares estimator and an asymptotic estimator. The recent estimators, however, are computationally expensive or only limited to estimating the tail distribution, and their performance may not be satisfactory. In this paper, we formulate the problem as a maximum-likelihood estimation problem. We then employ the expectation-maximization algorithm to solve this problem and derive a simple iterative estimator, which is easy to implement and computationally fast. Finally, we empirically demonstrate that our estimator is significantly more accurate than the state-of-the-art estimators and it can also be further improved with a proper choice of its parameter.**

I. Introduction

Sampling and estimating structural and topological properties and characteristics has been at the heart of understanding of large complex networks such as Web graphs and online social networks, which is prohibitively expensive without resorting to sampling due to their size and scale. In other words, since networks are often too large to observe in their entirety, the estimation and inference of their properties need to be made from sampled networks. Thus, there have been a plethora of research works in the literature [1] that develop and analyze network/graph sampling techniques and estimators to evaluate a wide range of target quantities, including degree distribution, density, diameter, assortativity coefficient, and clustering coefficient [2]–[8], as well as subgraph patterns, e.g., triples, motifs and graphlets [9]–[11].

Most of the sampling problems can be tackled by developing estimators in the form of sample averages or using sampled distributions. There is, however, still a non-trivial problem that is no longer solvable by the common framework and has not been well studied in the literature. It is to infer or estimate the latent in-degree distributions of directed graphs from random samples, when the samples represent only a few 'discovered' incoming edges into nodes and their sampled distribution is *far* from the original distribution. The latent nature of in-degrees arises in practice, because outgoing edges or links are only visible to users when querying and sampling nodes on a graph. The technical challenge is then how to recover the true yet unknown distribution from a skewed, sampled distribution. While this problem itself can find its importance for social network analysis and recommendation systems, it can also be generalized as recovering the distribution of set sizes on a graph when samples are available only in the form of some 'elements' of the sets, not the set sizes. Such a problem recently arises in the literature, e.g., for correcting the bias in classification tasks on a graph [12] and for inferring the entity frequency from Twitter data sampling [13].

The above problem can be formulated as an inverse problem and leads to a simple inversion estimator, as originally shown in [14], [15]. However, the inverse problem often appears to be ill-posed and thus results in poor estimation performance. For example, the resulting estimated distribution can exhibit oscillations to a great extent and even contain negative values. To avoid this problem, Zhang *et al.* [16] propose a novel estimator as the solution to a penalized, generalized least squares problem with non-negative constraints. While this estimator greatly improves the performance of the inversion estimator, it turns out to be computationally expensive. It is a 'numerical' solution of a complicated optimization problem, which first requires two key parameters to be determined by non-trivial algorithmic operations and then is numerically solved by an optimization toolbox. In addition, Antunes *et al.* [17] recently propose a simple asymptotic estimator that aims to estimate the tail distribution. Specifically, this estimator provides estimates on the probabilities of a few large, distant degrees. Despite its limited estimation capability, it is computationally much cheaper than the one in [16] and practically usable due to its simple design.

In this paper, we cast the problem as a constrained maximum likelihood estimation (MLE) problem under a random graph model. We first show that the solution to the MLE problem becomes identical to that of the inverse problem, when the non-negativity constraints of the MLE problem are ignored. We then resort to the expectation-maximization algorithm to solve the constrained MLE problem, which iteratively finds the maximum likelihood estimate of the *unknown* in-degree distribution. This leads to a simple iterative estimator in a closed form, where a prior distribution needs to be chosen. Here a uniform distribution can be simply used as the prior. We can summarize the benefits of our iterative estimator over the state-of-the-art estimators as follows.

- First, our estimator is easy to implement due to its closed-form expression and computationally fast, unlike the penal-

The authors are with Department of Computer Engineering and Sciences, Florida Institute of Technology, Melbourne, FL 32901, USA. Email: {sli2015, xhuang2016}@my.fit.edu; clee@fit.edu. This work was supported in part by the National Science Foundation under grants IIS-1908375 and CNS-2007828.

ISBN 978-3-903176-39-3© 2021 IFIP

ized weighted least squares estimator in [16]. The empirical evaluation under real-world network datasets shows that the runtime of our estimator is faster than the runtime of the latter by two orders of magnitude.

- Second, it estimates the *entire* distribution, while the asymptotic estimator in [17] is only limited to estimating its *tail* distribution.
- Third, our estimator is empirically shown to be substantially more accurate than these estimators under real-world network datasets. In particular, the reduction in the mean squared error (MSE) of our estimator compared to the MSE of the one in [16] can be up to over 90%.
- Finally, our estimator can also be further improved with a proper choice of the prior distribution.

II. Preliminaries

We explain the sampling problem of inferring or estimating the latent distributions of large graphs, whose representative example is to estimate the latent in-degree distributions of directed graphs. While this is our focus in this paper and its specific problem is described below in detail, it can be a more general problem. Suppose that we have n sets with their corresponding sizes $S_1, S_2, \ldots, S_n$. The problem here is how to recover the distribution of set sizes $\{S_i\}$ from a sample that is some 'elements' of the sets, which only provide *partial* information of the set sizes. If we are given samples directly from $\{S_i\}$, it would be straightforward to estimate the set-size distribution.[*] The problem at hand, however, is fundamentally different and becomes non-trivial, as shall be shown shortly.

A. Problem Setup

Consider a directed graph $G = (N, E)$, which represents a network of interest with nodes $N = \{1, 2, \ldots, n_G\}$ and edges E. Here n_G is assumed to be known a priori, as it has been the case in the literature [14]–[17]. Let d_i denote the *in-degree* of node $i \in N$, which is given by $d_i = |\{j : (j, i) \in E\}|$. Let w be the maximum in-degree, so $0 \le d_i \le w$ for all i. Hereafter we simply refer to 'in-degree' as 'degree' for brevity, unless otherwise stated. Define n_k, $k = 0, 1, \ldots, w$, to be the number of nodes of degree k in G, which is given by

$$n_k := \sum_{i=1}^{n_G} \mathbb{1}\{d_i = k\}, \tag{1}$$

and $n_G = \sum_{k=0}^{w} n_k$, where $\mathbb{1}\{A\}$ denotes an indicator function of an event A, having $\mathbb{1}\{A\} = 1$ if A occurs, and $\mathbb{1}\{A\} = 0$ otherwise. The fraction of nodes having degree k, denoted by f_k, can also be written as

$$f_k := n_k / n_G.$$

Let $\boldsymbol{n} := [n_0, n_1, \ldots, n_w]^T$ and $\boldsymbol{f} := [f_0, f_1, \ldots, f_w]^T$. The latter is the degree distribution of graph G or its probability mass function, with $\sum_{k=0}^{w} f_k = 1$. The former is an 'unnormalized' version of the degree distribution, or to indicate degree counts. Then, if we let D denote the degree of a node chosen *uniformly at random* in G, we have that the probability that it has degree k is simply $\mathbb{P}\{D = k\} = f_k$.

We consider the Bernoulli sampling methods with sampling rate $p > 0$, which are random node sampling and random edge sampling [1], [14]–[17]. Note that the solutions to the inference problem developed under these sampling methods have also been effectively used with samples obtained under other sampling methods such as random-walk sampling [16], [17]. While we expect that our solution can also be applied similarly, we here simply focus on the Bernoulli sampling methods to tackle the inference problem *itself*, which is still non-trivial. In the random node sampling, each node is selected or sampled with probability p. Then, all the edges between those selected nodes are included into the 'sampled' graph. For random edge sampling, we select (or sample) each edge with probability p and then include all the nodes that are incident to *at least* one selected edge into the sampled graph.

Observe that both sampling methods can then be characterized as follows. For a node with degree k in G, after sampling, it retains j neighbors (out of k) with probability

$$b_{jk} := \mathbb{P}\left\{\text{its sampled degree is } j \,|\, \text{original degree is } k\right\}, \tag{2}$$

where $b_{jk} = 0$ for all $j > k$. Specifically, for random node sampling, this probability becomes, for $0 \le j, k \le w$,

$$b_{jk} = \begin{cases} \binom{k}{j} p^{j+1} q^{k-j} + q\mathbb{1}\{j=0\} & \text{if } j \le k \text{ for } k = 0, \ldots, w, \\ 0 & \text{otherwise,} \end{cases} \tag{3}$$

where $q := 1 - p$. Similarly, under random edge sampling, it is given by

$$b_{jk} = \begin{cases} \binom{k}{j} p^{j} q^{k-j} & \text{if } j \le k \text{ for } k = 0, \ldots, w, \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

Note that in addition to sampled nodes (or the ones with sampled edges), we here also consider *unsampled/unselected* nodes as nodes with *zero* sampled degree, since their counts are still available. In other words, for each sampling method, we can write its $(w+1) \times (w+1)$ 'sampling' matrix $\mathbf{B} := [b_{jk}]$. Note that the sampling methods under consideration are agnostic to the underlying network structure [16], [17], so they are mainly characterized by their corresponding sampling matrices $\mathbf{B}$. One can easily see that $\mathbf{B}$ is a column stochastic matrix, i.e., $b_{jk} \ge 0$ and $\sum_{j=0}^{w} b_{jk} = 1$. Note that the forms of $\mathbf{B}$ in (3) and (4) are slightly different from the ones in [16], due to the inclusion of unsampled/unselected nodes. Nonetheless, the inference problem itself still remains the same.

Let $G' = (N', E')$ denote the resulting *sampled* graph by either random node sampling or edge sampling, where N' is a (random) permutation of N and $E' \subseteq E$ is the set of sampled edges. The nodes that are not sampled are included as zero-degree nodes in N', so its size remains the same as n_G. In

[*] Most of the previous studies in the 'graph sampling' literature [3]–[5], [7]–[11] fall into this category. Their common task is to estimate an expectation $\mathbb{E}_{\boldsymbol{\pi}}\{f\} = \sum_{i \in N} f(i)\pi(i)$ of a target function defined over the node set N, with a desired probability distribution $\boldsymbol{\pi} = [\pi(i), i \in N]$. The estimators are often in the form of the *sample average* of $f(X_1), f(X_2), \ldots, f(X_t)$, where $\{X_k\}$ are independently drawn from N according to $\boldsymbol{\pi}$ or form a Markov chain on N whose stationary distribution equals $\boldsymbol{\pi}$. The sample average over a large number of samples becomes a good approximation of $\mathbb{E}_{\boldsymbol{\pi}}\{f\}$ due to the ergodic theorem, i.e., $\hat{\mu}_t(f) := \sum_{s=1}^{t} f(X_s)/t \to \mathbb{E}_{\boldsymbol{\pi}}\{f\}$ as t grows [18].

other words, we have a set of the *sampled* degrees of all nodes as a sampling outcome. Let d'_j be the sampled degree of node j in G'. Then, if we let n'_j be the number of the nodes of (sampled) degree j in G', we have

$$n'_j := \sum_{i=1}^{n_G} \mathbb{1}\{d'_i = j\}, \quad j = 1, 2, \ldots, w, \tag{5}$$

and $n'_0 := n_G - \sum_{j=1}^{w} n'_j$. We can also obtain the fraction of nodes having (sampled) degree j in G', denoted by f'_j, as

$$f'_j := n'_j / n_G, \quad j = 0, 1, \ldots, w. \tag{6}$$

Let $\boldsymbol{n}' := [n'_0, n'_1, \ldots, n_w]^T$ and $\boldsymbol{f}' := [f'_0, f'_1, \ldots, f'_w]^T$. The latter indicates the *sampled* degree distribution of G' with $\sum_{j=0}^{w} f'_j = 1$, while the former is its unnormalized version. From a given $\boldsymbol{f}'$ (or $\boldsymbol{n}'$), which is considered collectively as 'a sample' throughout the rest of the paper, our problem is to recover the original degree *distribution* $\boldsymbol{f}$ (or $\boldsymbol{n}$). Note that this should be distinguished from the problem of estimating the *exact* degree of each node from samples [19]. Also, since it is to estimate the 'marginal' distribution of degrees, it does not require the estimation of any possible degree-degree correlation in the original graph.

B. Inversion Estimator and Its Drawbacks

For a randomly chosen node in G, if we let D' be its *sampled* degree in G', we can write

$$g_j := \mathbb{P}\{D' = j\} = \sum_{k=0}^{w} b_{jk} f_k, \quad j = 0, 1, \ldots, w, \tag{7}$$

which can also be written in a matrix form as

$$\boldsymbol{g} = \mathbf{B}\boldsymbol{f}, \tag{8}$$

where $\boldsymbol{g} := [g_0, g_1, \ldots, g_w]^T$. In addition, from (5) and (6), we have

$$\begin{aligned} \mathbb{E}[f'_j] &= \frac{\mathbb{E}\left[\sum_{i=1}^{n_G} \mathbb{1}\{d'_i = j\}\right]}{n_G} = \frac{\sum_{i=1}^{n_G} \mathbb{E}[\mathbb{1}\{d'_i = j\}]}{n_G} \\ &= \frac{\sum_{i=1}^{n_G} \mathbb{P}\{D' = j\}}{n_G} = \frac{n_G g_j}{n_G} = g_j, \end{aligned} \tag{9}$$

where the third equality follows since d'_i has the same distribution as D' for all i. Thus, from (8) and (9), we have

$$\mathbb{E}[\boldsymbol{f}'] = \mathbf{B}\boldsymbol{f}. \tag{10}$$

One can then naturally construct the following inversion estimator for a given sample $\boldsymbol{f}'$ [14]–[16]:

$$\hat{\boldsymbol{f}}_{\text{inv}} = \mathbf{B}^{-1}\boldsymbol{f}'. \tag{11}$$

It is straightforward to see that this estimator $\hat{\boldsymbol{f}}_{\text{inv}}$ is unbiased, since $\mathbb{E}[\hat{\boldsymbol{f}}_{\text{inv}}] = \mathbf{B}^{-1}\mathbf{B}\boldsymbol{f} = \boldsymbol{f}$. By leveraging the singular value decomposition of the sampling matrix $\mathbf{B}$, we can also rewrite (11) as

$$\hat{\boldsymbol{f}}_{\text{inv}} = \mathbf{V}\mathbf{D}^{-1}\mathbf{U}^T\boldsymbol{f}' = \sum_{k=0}^{w} \left[\frac{1}{\mu_k} \boldsymbol{u}_k^T \boldsymbol{f}'\right] \boldsymbol{v}_k, \tag{12}$$

from $\mathbf{B} = \mathbf{U}\mathbf{D}\mathbf{V}^T$, where $\mathbf{D} = \text{diag}(\mu_0, \mu_1, \ldots, \mu_w)$ is a diagonal matrix of singular values μ_i, and $\mathbf{U} = [\boldsymbol{u}_0, \boldsymbol{u}_1, \ldots, \boldsymbol{u}_w]$, $\mathbf{V} = [\boldsymbol{v}_0, \boldsymbol{v}_1, \ldots, \boldsymbol{v}_w]$ are the orthogonal matrices of the left- and right-singular vectors, respectively [20].

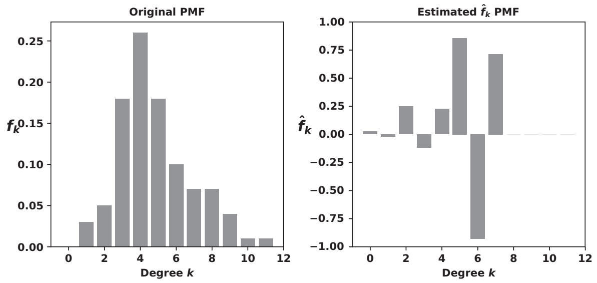


Fig. 1. Estimating the in-degree distribution of a directed Erdős-Rényi graph of 100 nodes by the estimator $\hat{\boldsymbol{f}}_{\text{inv}}$ from a sample $\boldsymbol{f}'$, which is drawn with $p = 0.6$.

Despite its simple form, this inversion estimator $\hat{\boldsymbol{f}}_{\text{inv}}$ has two drawbacks [16]. First, the matrix $\mathbf{B}$ may not be invertible *in practice*. Second, some elements of $\hat{\boldsymbol{f}}_{\text{inv}}$ may be negative, even when $\mathbf{B}$ is invertible. In other words, this estimator suffers from an ill-posed inversion problem. Specifically, the stability of the estimator can be characterized by the condition number of $\mathbf{B}$, which is the ratio of the maximum singular value to the minimum singular value. The larger the condition number is, the more unstable the estimator would be. Note that a non-invertible matrix has condition number equal to infinity. When the estimator is unstable, i.e., $\mathbf{B}$ is ill-conditioned, its estimated degree distribution can exhibit oscillations to a great extent. See Figure 1 for an example. Here the in-degree distribution of a directed Erdős-Rényi random graph with 100 nodes is estimated from a sample $\boldsymbol{f}'$, which is drawn by the random edge sampling with the sampling rate $p = 0.6$.[†]

III. State-of-The-Art Methods

We provide an overview of two recently proposed estimators for the problem of recovering the latent in-degree distribution of a directed graph from a sample $\boldsymbol{f}'$, or $\boldsymbol{n}'$, while overcoming the poor estimation performance of the inversion estimator.

A. Improved Inversion Estimator

We first explain an 'improved' inversion estimator proposed in [16]. It is based on a regularization method to solve the above inversion problem, especially when the matrix $\mathbf{B}$ is ill-conditioned. Specifically, the proposed estimator is a penalized, generalized least squares estimator with non-negative constraints, which is to estimate the degree counts $\boldsymbol{n}$ and is the solution $\hat{\boldsymbol{n}} := [\hat{n}_0, \hat{n}_1, \ldots, \hat{n}_w]$ to the following optimization problem for a given sample $\boldsymbol{n}'$:

$$\begin{aligned} &\operatorname*{arg\,min}_{\boldsymbol{n}} \; (\mathbf{B}\boldsymbol{n} - \boldsymbol{n}')^T \, \mathbf{C}^{-1} \, (\mathbf{B}\boldsymbol{n} - \boldsymbol{n}') + \lambda \cdot \|\boldsymbol{\mathcal{D}}\boldsymbol{n}\|_2^2 \\ &\text{subject to} \;\; n_k \geq 0, k = 0, 1, \ldots, w, \; \sum_{i=0}^{w} n_k = n_G, \end{aligned} \tag{13}$$

where $\mathbf{C} := \text{Cov}(\boldsymbol{n}')$ is the covariance matrix of $\boldsymbol{n}'$ and λ is a tuning parameter, to be determined separately. Note that the resulting 'estimated' degree distribution $\hat{\boldsymbol{f}} = [\hat{f}_0, \hat{f}_1, \ldots, \hat{f}_w]$

[†] A directed ER graph of 100 nodes is generated by placing a directed edge for each pair of nodes (in each direction) with probability $p_{gen} = 0.05$.

is given by $\hat{f}_i = \hat{n}_i/n_G$ for all i. The remaining second term in (13) is the regularization term and is a squared l_2-norm penalty function of $\boldsymbol{n}$, where $\boldsymbol{\mathcal{D}}$ is a $(w-1)\times(w+1)$ matrix representing the second-order differencing operator defined by

$$\boldsymbol{\mathcal{D}} = \begin{bmatrix} 1 & -2 & 1 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & 1 & -2 & 1 & \cdots & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 1 & -2 & 1 & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 1 & -2 & 1 \end{bmatrix}. \quad (14)$$

We can then rewrite the objective function in (13) so that the above problem is transformed to the following quadratic programing problem:

$$\begin{aligned} &\arg\min_{\boldsymbol{n}} \ \frac{1}{2}\boldsymbol{n}^T\mathbf{H}\boldsymbol{n} + \boldsymbol{\eta}^T\boldsymbol{n} \\ &\text{subject to } \ \boldsymbol{n} \succeq \mathbf{0}, \ \mathbf{1}^T\boldsymbol{n} = n_G, \end{aligned} \quad (15)$$

where

$$\mathbf{H} := 2\left[\mathbf{B}^T\mathbf{C}^{-1}\mathbf{B} + \lambda\boldsymbol{\mathcal{D}}^T\boldsymbol{\mathcal{D}}\right], \text{and } \ \boldsymbol{\eta} := -2\left[\boldsymbol{n}'^T\mathbf{C}^{-1}\mathbf{B}\right]^T.$$

Here $\succeq$ denotes the componentwise inequality and $\mathbf{0}$ is the w-dimensional column vector whose elements are all zeros. Similarly for $\mathbf{1}$ with all-one elements. Since this problem is a well-defined quadratic programing problem, we use the MATLAB optimization toolbox to solve this problem for numerical simulations of the improved inversion estimator in Section V.

While the improved inversion estimator in [16] is a solution to the problem in (15), or the original one in (13), there are two *non-trivial* algorithmic operations to determine the covariance matrix $\mathbf{C}$ and the penalty parameter λ before the problem is solved. First, since the matrix $\mathbf{C}$ needs to be estimated based only on the given sample $\boldsymbol{n}'$, by ignoring non-zero off-diagonal terms, they approximate the matrix $\mathbf{C}$ with a diagonal matrix of the following form:

$$\hat{\mathbf{C}} = \text{diag}(\boldsymbol{n}') + \delta\mathbf{I}, \quad (16)$$

where $\mathbf{I}$ is the $(w+1)\times(w+1)$ identity matrix and δ is a constant. Each diagonal element of $\mathbf{C}$, i.e., $\text{Var}[n'_i]$, is here approximated based on the Poisson approximation, which implies that $\text{Var}[n'_i] = \mathbb{E}[n'_i]$, and by replacing $\mathbb{E}[n'_i]$ with a sample value n'_i for all i. Since the errors between $\boldsymbol{n}'$ and $\mathbb{E}[\boldsymbol{n}']$ can be substantial, they employ a kernel-smoothing method to smooth out $\boldsymbol{n}'$ and obtain its smoothed version, say $\boldsymbol{n}'_{\text{smooth}}$, which is then used to compute $\hat{\mathbf{C}}$. In addition, the value of δ is chosen to adjust $\hat{\mathbf{C}}$ so that the resulting optimization problem remains stable [16].

Second, the parameter λ needs to be determined judiciously, since it controls the amount of the regularization or penalty term in (13), which can in turn significantly affect the accuracy of the resulting estimator. For a given $\boldsymbol{n}'$, the value of λ is chosen so as to minimize the 'estimate' of the weighted mean square error (WMSE), which is in the form of the first term in (13). To this end, they employ the so-called Stein's unbiased risk estimation (SURE) method [21] to obtain an estimate of the WMSE with a value of λ. Then, for a given $\boldsymbol{n}'$, by minimizing the estimated WMSE with respect to λ, they find the optimal value of λ that minimizes the estimated WMSE. Note that this entire process of choosing the value of λ is *computationally expensive*, since it needs to search through a grid of λ values, for which their corresponding estimated WMSEs are computed, and then to find the optimal value of λ. We refer to [16] for more details.

B. Asymptotic Estimator

We next present an asymptotic estimator proposed in [17] to mainly estimate the 'tail' distribution of in-degrees from a sample $\boldsymbol{n}'$. Recall that D is the original degree of a node chosen uniformly at random in G and D' is its *sampled* degree after sampling. Then observe that

$$D' = \sum_{k=1}^{D} Z_k,$$

where Z_k is an *i.i.d.* Bernoulli random variable with p. The following approximation can be made under mild assumptions [17], [22].

$$\mathbb{P}\{D' > i\} \approx \mathbb{P}\left\{\mathbb{E}[Z_1]D > i\right\} = \mathbb{P}\{pD > i\} \quad (17)$$

for sufficiently large values of i. Since $\mathbb{P}\{D' = i\} = \mathbb{P}\{D' > i-1\} - \mathbb{P}\{D' > i\}$, (17) can be written as $\mathbb{P}\{D' = i\} \approx (1/p)\cdot\mathbb{P}\{D = i/p\}$, which leads to

$$\mathbb{P}\{D = i\} \approx p\cdot\mathbb{P}\{D' = pi\} \quad (18)$$

for sufficiently large values of i. Thus, from (18), they develop the following asymptotic estimator to estimate the degree counts $\boldsymbol{n}$, which forms an 'unnormalized' version of the degree distribution $\boldsymbol{f}$, from a given sample $\boldsymbol{n}'$:

$$\hat{n}_i = \begin{cases} p\cdot n'_{pi} - 1, & i \in [1/(p\varepsilon^2), \tau/p], \\ n'_{pi}, & i \in (\tau/p, w'/p], \end{cases} \quad (19)$$

where ε is some predetermined small value, τ is a threshold value given by $\tau := \arg\min_i\{n'_i : n'_i > 0\}$, and w' is the largest sampled degree. Note that w'/p is an estimate of the maximum degree w. While the asymptotic estimator $\hat{n}_i$ in (19) mainly originates from (18), it also reflects their observation that the number of large-degree nodes near the maximum degree is one (more or less) for each large degree [17]. We choose the value of ε as in [17], and use $\hat{f}_i = \hat{n}_i/n_G$ to estimate the degree distribution $\boldsymbol{f}$ for numerical simulations of the asymptotic estimator in Section V.

It is worth noting that the asymptotic estimator $\hat{n}_i$ produces valid estimates only for a few distant values of i due to the discrete nature of the degree distribution. In other words, the nearest integer value of pi can remain the same for a wide range of (contiguous) values of i and so is n'_{pi}. Nonetheless, this estimator is practically usable thanks to its simple design, and it is computationally much cheaper than the improved inversion estimator.

IV. An Iterative EM Estimator

In this section, we propose a new estimator based on the expectation-maximization (EM) algorithm. We first formulate the problem of estimating the latent in-degree estimation as a constrained maximum likelihood estimation (MLE) problem under a random graph model. We show that the solution to its *unconstrained* MLE problem becomes identical to the inversion estimator in (11). We then resort to the EM algorithm to solve the original *constrained* MLE problem, which leads to a simple iterative estimator that is computationally inexpensive and readily usable in practice. It is also substantially more accurate than the state-of-the-art estimators, as shall be shown based on real-world network datasets in Section V.

Consider a random graph model in which the in-degree of each node in G is independently drawn from an arbitrary (yet unknown) distribution $\boldsymbol{f}$. Fix a sample $\boldsymbol{n}' = [n'_0, n'_1, \ldots, n'_w]$ that is obtained from G, where n'_j is the number of nodes having the sampled degree j in the sampled graph G'. Then, observe from (7) that the sampled degree d'_i of each node i in G' follows

$$\mathbb{P}\{d'_i = j\} = g_j = \sum_{k=0}^{w} b_{jk} f_k, \quad j = 0, 1, \ldots, w.$$

That is, each node i in G' contributes to one of the sampled-degree counts $n'_0, n'_1, \ldots, n'_w$ according to the probabilities g_j. Thus, we can write the following likelihood function, which is the probability that the sample $\boldsymbol{n'}$ is observed when the degree distribution is $\boldsymbol{f}$, and is in the form of a multinomial distribution:

$$p(\boldsymbol{n}'; \boldsymbol{f}) = \binom{n_G}{n'_0, n'_1, \ldots, n'_w} \prod_{j=0}^{w} \left(\sum_{k=0}^{w} b_{jk} f_k \right)^{n'_j}. \tag{20}$$

Taking the log function on both sides of (20) and ignoring the constant terms, we can write the following MLE problem that is to find the distribution $\boldsymbol{f}$ that maximizes the log-likelihood function from a given sample $\boldsymbol{n'}$.

$$\begin{aligned} &\arg\max_{\boldsymbol{f}} \ \mathcal{L}(\boldsymbol{f}; \boldsymbol{n}') := \sum_{j=0}^{w} n'_j \log \left(\sum_{k=0}^{w} b_{jk} f_k \right) \\ &\text{subject to } \boldsymbol{f} \succeq \boldsymbol{0}, \ \boldsymbol{1}^T \boldsymbol{f} = 1. \end{aligned} \tag{21}$$

Note that the problem here is to estimate the (unknown) degree distribution $\boldsymbol{f}$ from a sample $\boldsymbol{n'}$, but without knowing which nodes have how many (incoming) edges. In other words, it is not to estimate the *exact* degree of each node but to estimate the *distribution* $\boldsymbol{f}$.

We see that since the log function is concave and $\mathcal{L}$ is a linear combination of them, $\mathcal{L}$ has a unique stationary point $\boldsymbol{f}^*$ with $\boldsymbol{1}^T \boldsymbol{f}^* = 1$, which is the solution to an unconstrained problem of (21). Then we have the following.

Lemma 1: When the non-negativity constraints, i.e., $\boldsymbol{f} \succeq \boldsymbol{0}$, are ignored, the solution to (21) becomes identical to the inversion estimator in (11), provided that $\mathbf{B}$ is invertible.

Proof: See our technical report [23]. ■

From Lemma 1, we can see that the stationary point $\boldsymbol{f}^*$ may not fall within the feasible region of the problem in (21), since some elements of $\boldsymbol{f}^*$ may be negative, as seen from the inversion estimator in (11). In such a case, the solution to (21) lies on the boundary of the feasible region, which may be hard to solve analytically.

We employ the EM algorithm [24]–[26] to solve the MLE problem in (21), which leads to a simple iterative estimator. The main idea behind the EM algorithm is to find the maximum likelihood estimate of the (unknown) distribution $\boldsymbol{f}$ by maximizing the expectation of the likelihood function that involves 'unobserved' latent variables in addition to the observed sample $\boldsymbol{n}'$. The EM algorithm iteratively alternates between an expectation step and a maximization step to update the estimate of $\boldsymbol{f}$, as described below in detail. Let $\boldsymbol{f}^{(t)}$ be the estimate of $\boldsymbol{f}$ at iteration t.

1) Initialization: Pick a reasonable prior distribution $\boldsymbol{f}^{(0)}$. We use a uniform distribution as the prior, unless otherwise specified.

2) Expectation: We introduce unobserved latent variables, denoted by x_{jk}, $0 \le j \le k \le w$, to represent the number of nodes whose degrees are k in the original graph G and become j in the sampled graph G'. Note that $n'_j = \sum_{k=j}^{w} x_{jk}$. Let $\boldsymbol{x} := \{x_{jk} \,|\, 0 \le j \le k \le w\}$.

From (2), we can see that the probability that the 'sampled' degree of a node becomes j from its original degree k is $b_{jk} f_k$. In a manner similar to (21), we can write the following joint probability of having the latent variables $\boldsymbol{x}$ and the observed sample $\boldsymbol{n}'$ when the degree distribution is $\boldsymbol{f}$:

$$p(\boldsymbol{x}, \boldsymbol{n}'; \boldsymbol{f}) = \frac{n_G!}{\prod_{j=0}^{w} \prod_{k=j}^{w} x_{jk}!} \prod_{j=0}^{w} \prod_{k=j}^{w} \left(b_{jk} f_k \right)^{x_{jk}}. \tag{22}$$

We thus define the *complete* log-likelihood function of $\boldsymbol{f}$ as

$$\mathcal{L}_c(\boldsymbol{f}; \boldsymbol{x}, \boldsymbol{n}') := \sum_{j=0}^{w} \sum_{k=j}^{w} x_{jk} \log(b_{jk} f_k). \tag{23}$$

Also, we can obtain the conditional probability of $\boldsymbol{x}$ given $\boldsymbol{n}'$ as

$$p(\boldsymbol{x} | \boldsymbol{n}'; \boldsymbol{f}) = \prod_{j=0}^{w} \binom{n'_j}{x_{jj}, \ldots, x_{jw}} \prod_{k=j}^{w} \left(\frac{b_{jk} f_k}{\sum_{i=j}^{w} b_{ji} f_i} \right)^{x_{jk}}. \tag{24}$$

Letting

$$p_{jk} := \frac{b_{jk} f_k}{\sum_{i=j}^{w} b_{ji} f_i}, \quad 0 \le j \le k \le w, \tag{25}$$

we see that its marginal probability becomes, for $0 \le j \le k \le w$,

$$p(x_{jk} | \boldsymbol{n}'; \boldsymbol{f}) = \binom{n'_j}{x_{jk}} p_{jk}^{x_{jk}} \left(1 - p_{jk} \right)^{n'_j - x_{jk}}, \tag{26}$$

and $\mathbb{E}[x_{jk} | \boldsymbol{n}'; \boldsymbol{f}] = n'_j p_{jk}$. We refer to our technical report [23] for more details.

Having the estimate $\boldsymbol{f}^{(t)}$ at iteration t, we define the 'expectation' of the complete log-likelihood function as

$$Q(\boldsymbol{f} | \boldsymbol{f}^{(t)}) := \mathbb{E}_{\boldsymbol{x} \sim p(\boldsymbol{x} | \boldsymbol{n}'; \boldsymbol{f}^{(t)})} \left[\mathcal{L}_c(\boldsymbol{f}; \boldsymbol{x}, \boldsymbol{n}') \right],$$

where the expectation is with respect to $\boldsymbol{x}$ drawn according to $p(\boldsymbol{x} | \boldsymbol{n}'; \boldsymbol{f}^{(t)})$, which is given by (24) with $\boldsymbol{f}$ replaced by $\boldsymbol{f}^{(t)}$. From (23) and (26), we have

$$Q(\boldsymbol{f} | \boldsymbol{f}^{(t)}) = \sum_{j=0}^{w} \sum_{k=j}^{w} \mathbb{E}[x_{jk} | \boldsymbol{n}'; \boldsymbol{f}^{(t)}] \log(b_{jk} f_k)$$

$$= \sum_{j=0}^{w} \sum_{k=j}^{w} n'_j p_{jk}^{(t)} \log(b_{jk} f_k), \tag{27}$$

where $p_{jk}^{(t)}$ is defined as in (25) with $\boldsymbol{f}$ replaced by $\boldsymbol{f}^{(t)}$. That is, the expectation step at iteration t is to compute $Q(\boldsymbol{f}|\boldsymbol{f}^{(t)})$ in (27) based on the current estimate $\boldsymbol{f}^{(t)}$ and the given sample $\boldsymbol{n}'$. This expectation step, in fact, turns out to be *unnecessary* in finding the (maximum likelihood) estimate of $\boldsymbol{f}$, as will be shown below.

3) Maximization: Update the estimate $\boldsymbol{f}^{(t+1)}$ as the solution to the problem of maximizing $Q(\boldsymbol{f}|\boldsymbol{f}^{(t)})$, i.e.,

$$\boldsymbol{f}^{(t+1)} := \arg\max_{\boldsymbol{f}} Q(\boldsymbol{f}|\boldsymbol{f}^{(t)}). \tag{28}$$

Unlike the original MLE problem in (21), *we can ignore the non-negativity constraints* $\boldsymbol{f} \succeq \boldsymbol{0}$, due to the structure of $Q(\boldsymbol{f}|\boldsymbol{f}^{(t)})$ in (27), where f_k should be positive for all k. Thus, as was done in the proof of Lemma 1, we can simply use the Lagrange multiplier method to solve the problem in (28) with the equality constraint, i.e., $\boldsymbol{1}^T \boldsymbol{f}^{(t+1)} = 1$. Then, we obtain that $\boldsymbol{f}^{(t+1)}$ needs to be in the following form:

$$f_k^{(t+1)} = \frac{1}{C} \sum_{j=0}^{w} n'_j p_{jk}^{(t)}, \quad k = 0, 1, \ldots, w, \tag{29}$$

for some constant C, and $\boldsymbol{1}^T \boldsymbol{f}^{(t+1)} = 1$. In addition, from (25), we observe that $\sum_{k=0}^{w} p_{jk}^{(t)} = \sum_{k=j}^{w} p_{jk}^{(t)} = 1$, where $b_{jk} = 0$ for all $k < j$, as can be seen from (2). Thus, we have

$$\sum_{k=0}^{w} f_k^{(t+1)} = \frac{1}{C} \sum_{k=0}^{w} \sum_{j=0}^{w} p_{jk}^{(t)} n'_j = \frac{1}{C} \sum_{j=0}^{w} n'_j \sum_{k=0}^{w} p_{jk}^{(t)} = \frac{1}{C} \sum_{j=0}^{w} n'_j,$$

which leads to $C = n_G$, since $\boldsymbol{1}^T \boldsymbol{f}^{(t+1)} = 1$. Therefore, from (29), we finally have, for $k = 0, 1, \ldots, w$,

$$f_k^{(t+1)} = \sum_{j=0}^{w} p_{jk}^{(t)} f'_j = \sum_{j=0}^{w} \frac{b_{jk} f_k^{(t)}}{\sum_{i=0}^{w} b_{ji} f_i^{(t)}} f'_j, \tag{30}$$

where $f'_j = n'_j / n_G$ for all j. That is, the maximization step at iteration t is to update $\boldsymbol{f}^{(t+1)}$ in (30) based on the current estimate $\boldsymbol{f}^{(t)}$ and the given sample $\boldsymbol{n}'$. Note that this step does not require the computation of $Q(\boldsymbol{f}|\boldsymbol{f}^{(t)})$ in (27) at the expectation step.

To sum up, the EM algorithm leads to an iterative estimator in a closed form, which is simply to keep on updating the estimate $\boldsymbol{f}^{(t+1)}$ in (30) for a given sample $\boldsymbol{n}'$, or $\boldsymbol{f}'$. The iteration continues until the difference between two consecutive estimates becomes insignificant, i.e., $\|\boldsymbol{f}^{(t+1)} - \boldsymbol{f}^{(t)}\|_2 < \epsilon$ for a given value ϵ, where $\|\cdot\|_2$ indicates the l_2 norm. This iterative estimator is summarized in Algorithm 1, where the prior distribution is a uniform prior. Note that Algorithm 1 takes the maximum in-degree w as an input, but it also works with a rough estimate $\hat{w}$, as will be shown in Section V. Note also that ϵ is a tunable parameter, whose value can be chosen based on the graph size and the choice of the prior distribution.

In addition, we can characterize the time complexity of our EM estimator at each iteration. From lines 4 to 10 of Algorithm 1, we see that the time complexity at each iteration is $O(w|S|)$, where S denotes the set of nodes with non-zero sampled degrees. In addition, the number of required iterations until the stopping criterion is met (i.e., the speed of convergence) turns out to be *insignificant*, as shall be shown in the next section, e.g., Table III. We shall also empirically demonstrate that the overall runtime of our estimator is fast enough for various graphs.

Algorithm 1: Iterative EM estimator

```
   Input: $w$, $\mathbf{B}$, $\boldsymbol{f}'$
1  Define $\boldsymbol{c}$ as a $w$-dimensional vector
2  $\boldsymbol{f}^{(0)} \leftarrow \frac{1}{w}\boldsymbol{1}$; $S \leftarrow \{j : f'_j > 0\}$
3  for $t = 0, 1, 2, \ldots$ do
4      $\boldsymbol{c} \leftarrow \boldsymbol{0}$
5      for $j \in S$ do
6          for $i = 0, 1, \ldots, w$ do
7              $c_j \leftarrow c_j + b_{ji} f_i^{(t)}$
8      for $k = 0, 1, \ldots, w$ do
9          for $j \in S$ do
10             $f_k^{(t+1)} \leftarrow f_k^{(t+1)} + \frac{b_{jk} f_k^{(t)}}{c_j} f'_j$
11     if $\|\boldsymbol{f}^{(t+1)} - \boldsymbol{f}^{(t)}\|_2 < \epsilon$ then
12         break
13 return $\boldsymbol{f}^{(t+1)}$
```

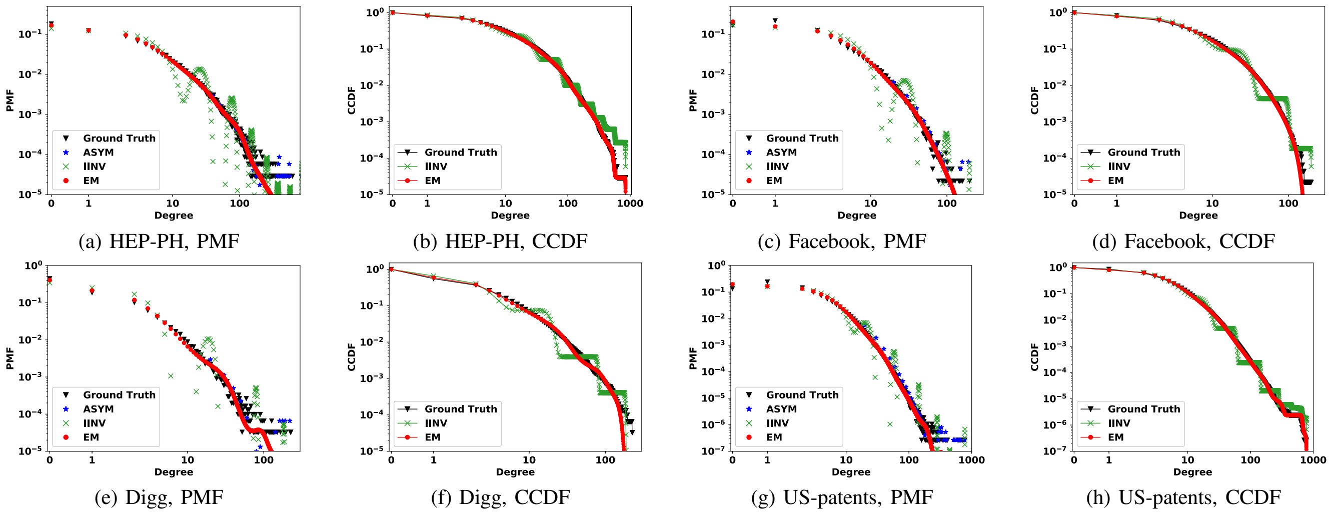
V. Simulation results

We present simulation results to demonstrate the efficacy of our EM estimator compared to the improved inversion estimator and the asymptotic estimator explained in Section III, which are referred to as 'IINV' and 'ASYM', respectively. We evaluate not only the accuracy of each estimator but also its runtime (time efficiency). All estimators are implemented and evaluated in MATLAB on a machine with 3.6-GHz Intel i7 CPU and 8-GB RAM. We consider four real-world *directed* network datasets from SNAP [27] and KONECT [28]. We preprocess each graph to remove self-loops and duplicate edges. Note that nodes that only have self-loops are removed, and the graphs may not be strongly connected. The statistics of the graphs after preprocessing are summarized in Table I.

TABLE I
Graph statistics

	HEP-PH	Facebook	Digg	US-patents
# Nodes	34,546	45,813	30,360	3,774,768
# Edges	421,578	264,004	85,247	16,518,947

For numerical simulations, we focus on the random edge sampling with varying sampling rate p for estimating or inferring the in-degree distributions from samples. Specifically, for a given graph G, and for a sample $\boldsymbol{f}'$, which is a set of the 'sampled' in-degrees of n nodes (as a sampling outcome with a given value of p), each estimator provides an estimate for the in-degree distribution. While the estimated in-degree distributions are in the form of probability mass function (PMF), $\hat{f}_d = \hat{\mathbb{P}}\{D = d\}$, we also report the results in the



(a) HEP-PH, PMF

(b) HEP-PH, CCDF

(c) Facebook, PMF

(d) Facebook, CCDF

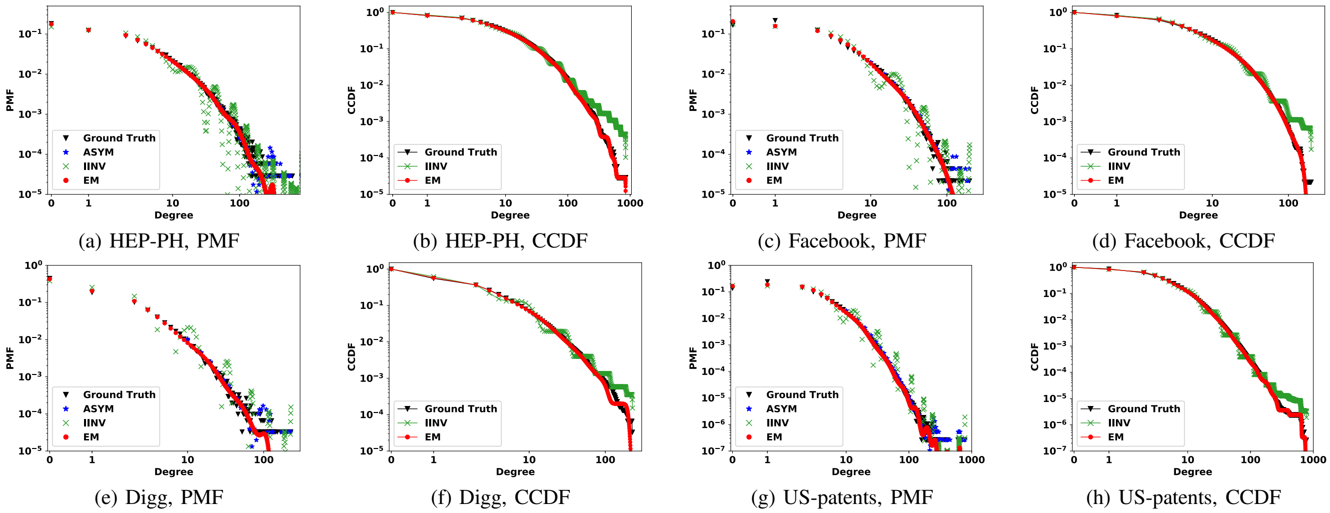
(e) Digg, PMF

(f) Digg, CCDF

(g) US-patents, PMF

(h) US-patents, CCDF

Fig. 2. Simulation results on a log-log scale for the degree-distribution estimation with $p = 0.1$.



(a) HEP-PH, PMF

(b) HEP-PH, CCDF

(c) Facebook, PMF

(d) Facebook, CCDF

(e) Digg, PMF

(f) Digg, CCDF

(g) US-patents, PMF

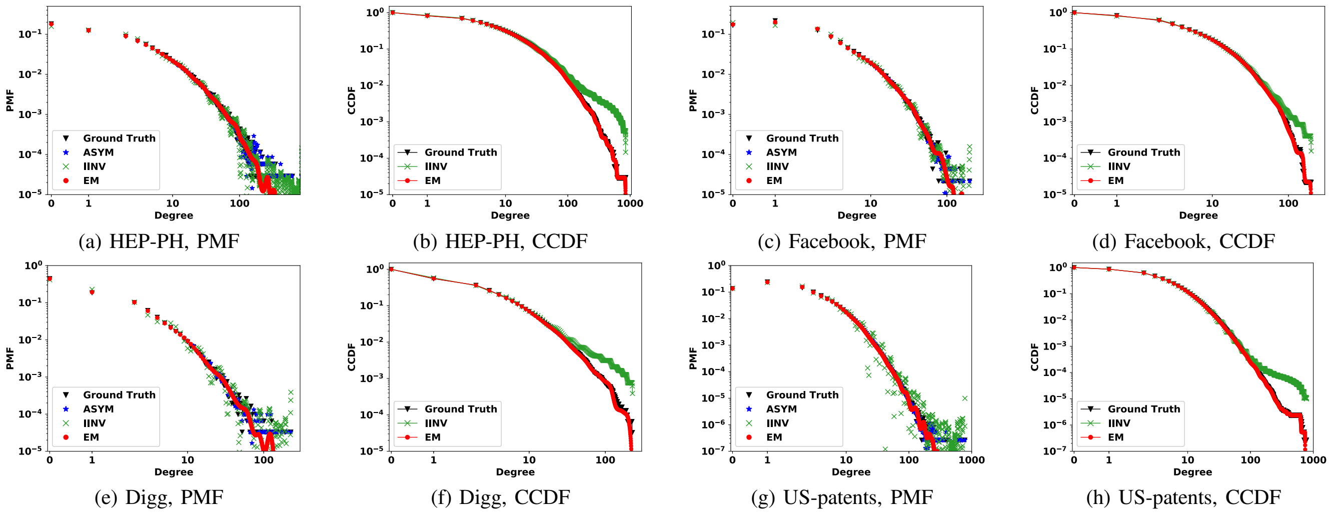
(h) US-patents, CCDF

Fig. 3. Simulation results on a log-log scale for inferring the degree distributions from samples with $p = 0.2$.

complementary cumulative distribution function (CCDF), say $\hat{\mathbb{P}}\{D > d\}$. Each estimated distribution shall be 'visually' compared to its ground-truth distribution. We also use the *empirical* mean squared error (MSE) to evaluate the accuracy of each estimator. It is to measure the average squared difference between an estimated probability and its original one, i.e., MSE $= \mathbb{E}[\hat{f}_d - f_d]^2$, for each degree d. We here only report the average of the MSEs over all d, each of which is obtained based on 10 different samples while they remain the same for all estimators.

We note that the value of the maximum degree w needs to be set for each estimator. We use the 'actual' value of w for the IINV estimator, as used in [16], since it requires the exact value of w to be known a priori. We employ an estimate $\hat{w} = w'/p$ for the ASYM estimator, as used in [17], where w' is the observed largest sampled degree and p is the sampling rate. For our EM estimator, it does not need the exact w, but it works with just a rough estimate $\hat{w}$. We empirically observed that our estimator is robust for a wide range of values of $\hat{w}$, as long as $\hat{w}$ is not severely underestimated. For example, there was not much difference between the cases of $\hat{w} = w'/p$ and $\hat{w} = 2w'/p$ for the performance of our estimator. Thus, we simply choose the latter as a bit more conservative estimate in this paper. In addition, we set the 'termination' threshold $\epsilon = 10^{-3}$ for the EM estimator for all graphs except US-patents, and use $\epsilon = 10^{-4}$ for US-patents due to its sheer size.

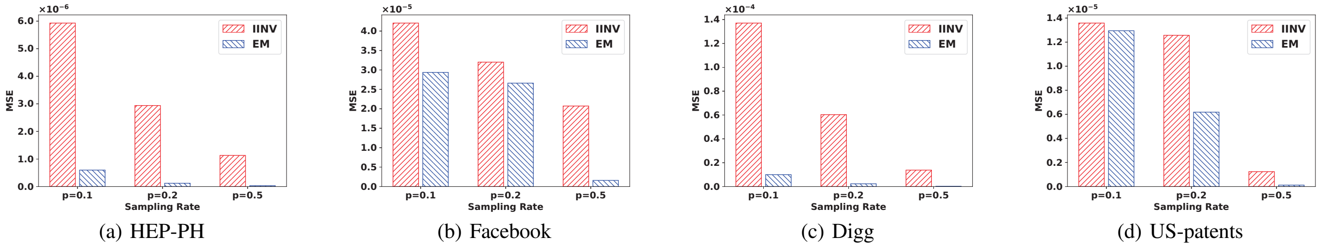
Figures 2–4 present the original in-degree distributions and their estimated distributions by the ASYM, IINV, and EM estimators. For each graph and for each sampling rate p, we use the same sample $\boldsymbol{f}'$ to obtain the estimated in-degree distributions. Note that for the ASYM estimator, the PMF results are only presented, since it is designed to provide an estimate $\hat{f}_d$ for *only a few large* degrees d that are also *not contiguous*, as explained in Section III. As can be seen from Figures 2–4, we make the following observations. First, the estimation of our EM estimator is *most accurate* for almost entire range of the values of d in both CCDF and PMF. Second, while the IINV estimator greatly improves the 'vanilla' inversion estimator in (11), it still exhibits non-negligible oscillations in its estimated degree distribution, which are mostly noticeable



(a) HEP-PH, PMF (b) HEP-PH, CCDF (c) Facebook, PMF (d) Facebook, CCDF

(e) Digg, PMF (f) Digg, CCDF (g) US-patents, PMF (h) US-patents, CCDF

Fig. 4. Simulation results on a log-log scale for estimating the degree distributions from samples with $p = 0.5$.



(a) HEP-PH (b) Facebook (c) Digg (d) US-patents

Fig. 5. MSEs of the IINV and EM estimators when estimating the degree distribution.

TABLE II
RUNTIMES OF THE IINV AND EM ESTIMATORS (IN SECONDS)

Graph	HEP-PH			Facebook			Digg			US-patents		
p	0.1	0.2	0.5	0.1	0.2	0.5	0.1	0.2	0.5	0.1	0.2	0.5
IINV	49.4270	48.1393	46.7614	1.8836	1.8556	1.8225	2.3836	2.3564	2.3119	41.8413	40.0046	38.8662
EM	0.0750	0.0829	0.1226	0.0187	0.0189	0.0399	0.0275	0.0308	0.0343	0.1790	0.4471	0.6022

for the PMFs and for small values of p, i.e., $p = 0.1, 0.2$. We also observe that this oscillating behavior of the IINV estimator is affected by the choice of its penalty parameter λ. We refer to our technical report [23] for more details. Third, the IINV estimator is accurate in capturing the 'head' of the in-degree distribution, while the accuracy of the ASYM estimator is reasonable for the 'tail' of the distribution, as intended by its design.

We show the (averaged) MSEs of the IINV and EM estimators in Figure 5. The results again confirm the superiority of our EM estimator over the IINV estimator. The improvement from the EM estimator compared to the IINV estimator turns out to be significant for most cases, regardless of the underlying graph and the sampling rate p. The reduction in MSE can be even more than 90% for several cases. In addition, while the comparison is here limited between the IINV and EM estimators due to the aforementioned issue with ASYM, we have also observed that the EM estimator generally outperforms the ASYM estimator, even when computing the MSEs for the values of d for which ASYM's estimates are available. We omit the results for brevity.

We next turn our attention to the time efficiency of our EM estimator. We measure the runtimes of the IINV and EM estimators and report them in Table II. Our EM estimator turns out to be faster than the IINV estimator by *two orders of magnitude*. This is in fact well expected from their algorithmic operations, which clearly exhibit the advantage of our estimator over the IINV estimator. In other words, as explained in Section III-A, the IINV estimator involves non-trivial operations and solving an optimization problem, while our estimator is just a simple iterative method, as seen from Algorithm 1. Note that we here do not report the runtime of the ASYM estimator, since it merely estimates the *tail* distribution, not the entire distribution. Nonetheless, we observed that the runtime of our estimator is still comparable to that of the ASYM estimator.

We finally show that the performance of our EM estimator can be further improved with a proper choice of the prior distribution. To this end, we newly consider an exponential distribution as the prior distribution. Note that all the results so far are obtained based on the uniform prior distribution. The parameter of the exponential distribution is set to be a reciprocal of the 'estimated' averaged degree, which is the average 'sampled' degree divided by p. We then report the

TABLE III
IMPACT OF THE PRIOR ON THE EM ESTIMATOR

Graph		HEP-PH			Facebook			Digg			US-patents		
p		0.1	0.2	0.5	0.1	0.2	0.5	0.1	0.2	0.5	0.1	0.2	0.5
# Iter.	Unif.	19	13	8	17	10	19	35	23	10	108	162	83
	Exp.	7	6	6	7	8	17	31	20	9	94	154	80
MSE ($\times 10^{-6}$)	Unif.	0.61	0.14	0.04	29.32	26.81	2.14	10.91	2.39	0.24	12.92	6.17	0.11
	Exp.	0.33	0.13	0.03	28.99	26.32	2.02	10.90	2.39	0.24	12.88	6.14	0.11

results for the convergence speed in terms of the number of iterations and the MSE in Table III. They all indicate that the exponential prior can reduce the convergence speed, while having the estimation accuracy to be comparable to or even better than that of the uniform prior. We have also observed that there is not much difference in the shapes of their resulting distributions, which are omitted for brevity.

In summary, our EM estimator turns out to be substantially better than the IINV and ASYM estimators, which are the state-of-the-art estimators. The IINV estimator still exhibits oscillations in the estimated distributions, especially in the tails, as seen from the *inversion* estimator. It is also sensitive to the choice of its parameter λ, not to mention the high complexity of its algorithmic operation. In addition, the ASYM estimator can only estimate the distributions for a few large degrees that are far between.

VI. CONCLUSION

We have studied the problem of inferring or estimating the latent in-degree distributions of directed graphs from random samples. The technical challenge behind this problem is that it often becomes an *ill-posed* inverse problem. In this work, we have formulated the problem as an MLE problem and resorted to the EM algorithm to solve the problem, which iteratively finds the maximum likelihood estimate of the unknown in-degree distribution $\boldsymbol{f}$ from an observed sample $\boldsymbol{f}'$. The resulting iterative estimator has shown to be significantly more accurate than the state-of-the-art estimators, while being easy to implement, computationally fast, and amenable to further improvement with a proper choice of the prior. Finally, we expect that our iterative estimator can also be readily adopted and used for other problems, e.g., classification tasks on a graph and inferring the entity frequency from Twitter, which involve recovering the distribution of set sizes when a sample is available only in the form of some elements of the sets.

REFERENCES

[1] N. K. Ahmed, J. Neville, and R. Kompella, "Network sampling: From static to streaming graphs," *ACM Trans. Knowl. Discov. Data*, vol. 8, no. 2, Jun. 2013.

[2] J. Leskovec and C. Faloutsos, "Sampling from large graphs," in *Proceedings of ACM SIGKDD*, 2006, pp. 631–636.

[3] M. Gjoka, M. Kurant, C. T. Butts, and A. Markopoulou, "Walking in Facebook: A case study of unbiased sampling of OSNs," in *Proceedings of IEEE INFOCOM*, Mar. 2010, pp. 1–9.

[4] B. Ribeiro and D. Towsley, "Estimating and sampling graphs with multidimensional random walks," in *Proceedings of ACM SIGCOMM Conference on Internet Measurement*, 2010, pp. 390–403.

[5] C.-H. Lee, X. Xu, and D. Y. Eun, "Beyond random walk and Metropolis-Hastings samplers: Why you should not backtrack for unbiased graph sampling," in *Proc. ACM SIGMETRICS*, Jun. 2012, pp. 319–330.

[6] L. Katzir and S. J. Hardiman, "Estimating clustering coefficients and size of social networks via random walk," *ACM Trans. Web*, vol. 9, no. 4, Sep. 2015.

[7] Z. Zhou, N. Zhang, and G. Das, "Leveraging history for faster sampling of online social networks," *Proceedings of the VLDB Endowment*, vol. 8, no. 10, pp. 1034–1045, Jun. 2015.

[8] C.-H. Lee, X. Xu, and D. Y. Eun, "On the Rao-Blackwellization and its application for graph sampling via neighborhood exploration," in *Proceedings of IEEE INFOCOM*, May 2017, pp. 1–9.

[9] M. Rahman and M. A. Hasan, "Sampling triples from restricted networks using MCMC strategy," in *Proceedings of ACM CIKM*, 2014, p. 1519–1528.

[10] P. Wang, J. Zhao, J. C. S. Lui, D. Towsley, and X. Guan, "Unbiased characterization of node pairs over large graphs," *ACM Trans. Knowl. Discov. Data*, vol. 9, no. 3, Apr. 2015.

[11] X. Chen, Y. Li, P. Wang, and J. C. S. Lui, "A general framework for estimating graphlet statistics via random walk," *Proceedings of the VLDB Endowment*, vol. 10, no. 3, pp. 253–264, Nov. 2016.

[12] G. Berry, A. Sirianni, N. High, A. Kellum, I. Weber, and M. Macy, "Estimating group properties in online social networks with a classifier," in *Social Informatics*, S. Staab, O. Koltsova, and D. I. Ignatov, Eds. Springer International Publishing, 2018, pp. 67–85.

[13] S. Wu, M.-A. Rizoiu, and L. Xie, "Variation across scales: Measurement fidelity under twitter data sampling," *Proceedings of the International AAAI Conference on Web and Social Media*, pp. 715–725, May 2020.

[14] O. Frank, "Estimation of the number of vertices of different degrees in a graph," *Journal of Statistical Planning and Inference*, vol. 4, no. 1, pp. 45–50, 1980.

[15] ——, "A survey of statistical methods for graph analysis," *Sociological Methodology*, vol. 12, pp. 110–155, 1981.

[16] Y. Zhang, E. D. Kolaczyk, and B. D. Spencer, "Estimating network degree distributions under sampling: An inverse problem, with applications to monitoring social media networks," *Annals of Applied Statistics*, vol. 9, no. 1, pp. 166–199, 2015.

[17] N. Antunes, S. Bhamidi, T. Guo, V. Pipiras, and B. Wang, "Sampling-based estimation of in-degree distribution with applications to directed complex networks," *arXiv preprint arXiv:1810.01300*, 2018.

[18] P. Brémaud, *Markov Chains: Gibbs Fields, Monte Carlo Simulation, and Queues*. Springer-Verlag, 1999.

[19] A. Ganguly and E. D. Kolaczyk, "Estimation of vertex degrees in a sampled network," in *Procceedings of the 51st Asilomar Conference on Signals, Systems, and Computers*, 2017, pp. 967–974.

[20] G. H. Golub and C. Reinsch, "Singular value decomposition and least squares solutions," in *Linear Algebra*. Springer, 1971, pp. 134–151.

[21] Y. C. Eldar, "Generalized sure for exponential families: Applications to regularization," *IEEE Transactions on Signal Processing*, vol. 57, no. 2, pp. 471–481, 2009.

[22] C. Y. Robert and J. Segers, "Tails of random sums of a heavy-tailed number of light-tailed terms," *Insurance: Mathematics and Economics*, vol. 43, no. 1, pp. 85–92, 2008.

[23] S. Li, X. Huang, and C.-H. Lee, "Estimating Distributions of Large Graphs from Incomplete Sampled Data," Technical Report, May 2021.

[24] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society: Series B*, vol. 39, no. 1, pp. 1–22, 1977.

[25] C. F. J. Wu, "On the convergence properties of the EM algorithm," *Annals of Statistics*, vol. 11, no. 1, pp. 95–103, 1983.

[26] T. M. Mitchell, *Machine Learning*. McGraw-Hill, 1997.

[27] J. Leskovec and A. Krevl, "SNAP Datasets: Stanford large network dataset collection," http://snap.stanford.edu/data, Jun. 2014.

[28] J. Kunegis, "KONECT – The Koblenz Network Collection," in *Proceedings of International Conference on World Wide Web*, 2013, pp. 1343–1350.