Networks cardinality estimation using order statistics

Riccardo Lucchese, Damiano Varagnolo

Abstract

We consider a network of collaborative peers that aim at distributedly estimating the network cardinality. We assume nodes to be endowed with unique identification numbers (IDs), and we study the performance of size estimators that are based on exchanging these IDs. Motivated by practical scenarios where the time-to-estimate is critical, we specifically address the case where the convergence time of the algorithm, i.e., the number of communications required to achieve the final estimate, is minimal. We thus construct estimators of the network size by exploiting statistical inference concepts on top of the distributed computation of order statistics of the IDs, i.e., of the M biggest IDs available in the network. We then characterize the statistical performance of these estimators from theoretical perspectives and show their effectiveness in practical estimation situations by means of numerical examples.

Index Terms

Distributed size estimation, distributed counting, order statistics consensus, peer-to-peer networks, cooperative systems, event detection.

I. Introduction

In distributed applications knowing the properties of the underlying communication networks may lead to better performing algorithms. E.g., knowing the number of nodes may lead to more precise distributed estimators [1]. It is thus meaningful to seek for estimators of the properties of the communication graphs that sense these properties with the smallest possible computational / communications overheads. Moreover, this sensing should be distributed, i.e., conform to the distributed computations paradigm where the network lacks of a centralized authority and the nodes are peers.

Consider then the following technology for solving the archetypal problem of estimating the size of a network, at first sight the most simple one in terms of computational and communications overheads: let every node i of the network be associated to an identification number (or ID) y_i, initially known only by itself. Then let nodes form, store and propagate lists of these y_is among them. When a node i has collected the complete list, the size of the network can be determined exactly by inspecting the size of the list itself[1].

This work is supported by the Framework Programme for Research and Innovation Horizon 2020 under grant agreement n. 636834 (DISIRE), and the Swedish research council Norrbottens Forskningsråd.

Riccardo Lucchese and Damiano Varagnolo are with the Department of Computer Science, Electrical and Space Engineering, Luleå University of Technology, Luleå, Sweden. Emails: { riccardo.lucchese | damiano.varagnolo }@ltu.se.

[1]In fact, one can build on top of this simple strategy to infer the entire topology of the network, see [2].

Without constraints on how many y_is one can send per packet, the network size estimation problem is thus trivial. If, instead, the number of transmittable y_is is limited, and therefore only a subset of the y_is can be sent per packet, then nodes must select which y_is should be communicated at each round. This degree of freedom makes the estimation problem more interesting.

In this manuscript we study which selection mechanisms should be implemented, and what are the consequences of these selection processes, considering the following two practical requirements: *i)* the strategy should minimize the convergence time, i.e., the number of communications among nodes to achieve the final estimate; *ii)* the strategy should lead nodes to share an identical final estimate.

Literature review: the problem of distributedly counting or inferring networks cardinalities has been extensively studied in the literature. There is a vast portfolio of techniques, each characterized by different properties and trade-offs: performing random walks [3], [4], [5], [6], [7], computing averages of the IDs [8], [9], [10], computing the eigenvalues of the Laplacian of the communication graph [11], exploiting Good-Turing estimators based on the number of occurrences of the IDs [12], scanning opportunely the binary representation of the IDs of the nodes [13], [14], borrowing concepts from identification of LTI systems over finite fields [15], performing opportune Gram-Schmidt orthogonalization of randomly generated IDs [16], and Bayesian schemes [17].

W.r.t. the estimation scheme proposed in this manuscript, all the strategies above perform more complex computational operations and require longer convergence times.

A strategy in the same playground of the one considered here (i.e., with the same computational complexity and minimal convergence time) is the basis of [9], [18], [19], [20], and works as follows: *i)* let each node locally generate M random IDs $y_{i,m}$, $m = 1, \dots, M$ instead of just one ID y_i; *ii)* make nodes distributedly compute over the network the M different maxima $y_m^{\max} = \max_i \{y_{i,m}\}$; *iii)* have each node estimate the network size using Maximum Likelihood (ML) concepts. As it will be clear later, the approach proposed in this manuscript has overall better statistical performance.

Other strategies that are also based on the computation of order statistics have been proposed in [21], [22], [23], [24]. Nonetheless, the results obtained in this work are distinguished in three fundamental ways (cf. also the following statement of contributions): *i)* the proposed point estimator is derived from approximated ML concepts rather than methods of moments; *ii)* the proposed interval estimator and the related hypothesis testing results are completely novel; *iii)* the strategy is tailored for the case of networks of peer nodes, and not for databases or other centralized applications.

Statement of contributions: the previously posed assumptions, i.e., that each node is associated to a scalar ID y_i, that nodes can form and propagate lists of IDs, but that the number of y_is that can be propagated per transmission round is limited, introduce the problem of which IDs should be selected for transmission. The question is then which is the (statistically) best performing strategy that satisfies to the practical requirements of *i)* minimizing the convergence time, i.e., the number of communications to achieve the final estimate; *ii)* leading nodes to all share an identical final estimate.

Since stochastic selection processes would lead to stochastic convergence times, we specifically consider deter-

ministic strategies that minimize the convergence time, namely the computation of order statistics of the y_is, i.e., the computation of their M biggest and/or smallest values [25].

The contributions of this manuscript are thus the following:

- motivate why it is meaningful to analyze just what can be obtained computing maximum values, and neglect minimum values or ranges (differences between maximum and minimum values);
- derive and statistically characterize approximated ML point estimators of the network size that follow from approximated score functions;
- derive and statistically characterize interval estimators in the form of statistical hypothesis tests on the network size.

Organization of the manuscript: in Section II we collect the notation used throughout this text and frame the cardinality estimation problem in a formal way. In Section III we discuss in detail an iterative algorithm to distributedly compute order statistics. In Section IV we derive an approximated ML estimator and characterize its statistical performance. In Section V we consider how the nodes can perform statistical hypothesis testing on the network size. In Section VI we show the effectiveness of the estimation strategies by means of simulated experiments. Finally, in Section VII we collect some concluding remarks and discuss future research directions.

II. Problem Formulation

Due to the iterative nature of the estimation algorithm, we describe the quantities of interest at discrete points in time. For this purpose, we partition time into an ordered set of equally lasting intervals indexed by the integer variable $t = 0, 1, 2, \ldots$. We informally refer to each of these time-intervals as to an "epoch".

We model the communication network as a directed, strongly connected and, w.l.o.g. for our findings, time-constant graph $G = (V, E)$ with $V = \{1, 2, \ldots, n\}$. In particular, the cardinality of the network is $n = |V|$. Communications are assumed to be perfect (i.e., no collisions, no delays, no information sharing errors). The exchange of information between nodes follows a broadcast communication protocol, i.e., when $i \in V$ transmits, it transmits to all its neighbors $\mathcal{N}_i := \{j : (i, j) \in E\}$ simultaneously, and these js are not required to acknowledge the transmission.

Nodes are assumed to be equipped with a local random number generator that, during initialization, draws an independent sample from a common absolutely continuous distribution $P_Y(\cdot)$. The random outcome is then stored in the local variable y_i. In the following, we informally refer to y_i as the ID of node i. The fictitious IDs $y_1, \ldots, y_n$ are thus viewed as an n-dimensional sample extracted from $P_Y(\cdot)$.

Importantly, we assume that when nodes communicate, they exchange packets containing at most M different IDs, with M fixed a priori. Different Ms thus trade off the amount of information that is locally available for statistical inference with the communication requirements.

The aim of the nodes is to reach consensus, in the smallest number of epochs possible, on an estimate of the network cardinality n starting from no a priori knowledge on the network topology or on n itself. The purpose is thus not only to estimate effectively n, but also to reach, as soon as possible, agreement on the same estimate $\widehat{n}$.

Remark 1 *Each ID y_i is assumed to be a real number, and we thus neglect in first approximation quantization issues. We nonetheless notice that if the set of all plausible IDs is finite, e.g., strings of b bits, then the probability of collisions is described by a generalized birthday paradox. Specifically, this probability is given by $1-\prod_{k=0}^{n-1}\frac{2^b-k}{2^b}$ and therefore is strictly decreasing in b.*

III. Order statistics consensus

Let $x_{(1)},\dots,x_{(n)}$ be the outcome of sorting the vector of initial IDs $y_1,\dots,y_n$ in ascending order. By construction the variable $x_{(m)}$ takes the m-th smallest value in $y_1,\dots,y_n$ and is called the m-th order statistic.

Assume that the maximum number of transmittable IDs per communication has been fixed through M. Then, the order statistic $x_{(n-M+1)}$ can be distributedly computed by the network through the following Algorithm 1.

Algorithm 1 Order statistics consensus

1: *(local storage requirements)* vectors ${}^ix, {}^iw \in \mathbb{R}^M$;

2: *(initialization of the local storage)* let ${}^ix = \left[{}^ix_1,\dots,{}^ix_M\right]$ by setting ${}^ix_m = 0$ for $m = 1,\dots,M-1$, and ${}^ix_M = y_i$; let ${}^iw = 0$;

3: **for** each epoch $t = 0,1,2,\dots$ **do**

4: *(on epoch start, save the current local state)* ${}^iw \leftarrow {}^ix$;

5: *(on transmission, that happens once per epoch, and uniformly i.i.d. during the epoch)* broadcast the current ${}^iw$ to the neighboring nodes $j \in \mathcal{N}_i$;

6: *(on reception, that happens $|\mathcal{N}_i|$ times per epoch)* upon reception of ${}^jw$ from neighbor j, update ${}^ix$ by selecting and sorting (in ascending order) the M biggest elements in ${}^ix \cup {}^jw$. I.e., letting $\xi \in \mathbb{R}^{2M}$ be a temporary vector, and describing this operation in Matlab-like pseudo-code, let

$$\xi \leftarrow \text{unique}\left(\text{sort}\left(\text{stack}\left({}^ix, {}^jw\right)\right)\right) ,$$

$${}^ix \leftarrow \xi_{(M+1):(2M)} .$$

7: **end for**

Notice that during each epoch each node performs two basic functions: *i)* it uses the received IDs to update its local information; *ii)* it broadcasts its information to its neighbors. Importantly, to compute $x_{(n-M+1)}$, the nodes are required to compute also $x_{(n-M+2)},\dots,x_{(n)}$.

We stress that the number of scalars that are broadcast by a node per epoch does not grow indefinitely, but instead stays bounded by the design parameter M. Moreover, nodes may skip the zero entries in the current ${}^iw$ when transmitting, so that the length of packets scales with the network cardinality up to $n = M$.

Consensus, i.e., the condition where all the nodes have computed correctly $x_{(n-M+1)}$, is achieved at most after d epochs, with d the diameter of the network. In fact, a sufficient condition for achieving consensus is that the information of any given node can eventually be propagated to the rest of nodes in the network. Therefore, given

our strong connectivity assumptions, the local states ${}^1x, \ldots, {}^nx$ converge, at most after d steps, to the consensus vector

$$x := \begin{bmatrix} x_{(n-M+1)} & x_{(n-M+2)} & \ldots & x_{(n)} \end{bmatrix} . \tag{1}$$

For notational brevity, in the following we let x_m, $1 \leq m \leq M$, indicate the m-th component of x, so that $x_1 := x_{(n-M+1)}$, $x_2 := x_{(n-M+2)}$ and so on.

Assuming that the consensus vector (1) has been computed, a node can distinguish between two cases:

1) x has some zero entries: this implies that $n < M$, and thus the cardinality of the network is given precisely by the number of non-zero entries of x;
2) x has no zero entries: this implies that $n \geq M$, and thus that x_1 is the searched $(n-M+1)$-th order statistic. This is the interesting case from our statistical perspectives and gives birth to the question of how to estimate n given $x_{(n-M+1)}$.

IV. Estimating cardinalities using order statistics

This section leverages Algorithm 1 for the estimation of the cardinality of a network, and is articulated in four parts: IV-A, showing that to compute maxima (i.e., statistics of the kind $x_{(n-m)}$, where n is the cardinality of the network), minima (i.e., statistics of the kind $x_{(m)}$), or ranges (i.e., combinations like $x_{(n-m)} - x_{(m)}$), is for our purposes equivalent; IV-B, deriving a closed-form for the Maximum Likelihood (ML) estimator of the network size n; IV-C, proposing and characterizing an estimator naturally approximating the ML one; and IV-D, characterizing the statistical properties of the approximated estimator.

A. Computing maxima, minima or ranges is equivalent for cardinality estimation purposes

Let the initial IDs $y_1, \ldots, y_n$ be n i.i.d. realizations of the same continuous r.v. Y, i.e., let Y be described by a generic absolutely continuous probability distribution $P_Y(y)$, so that Y admits its density $p_Y(y)$. Define the random variables $X_{(1)}, \ldots, X_{(n)}$ as the order statistics of an n-dimensional sample with i.i.d. components extracted from Y. We denote with $x_{(m)}$, $1 \leq m \leq M$, the realization of $X_{(m)}$, so that $x_{(m)}$ is the m-th smallest value in $y_1, \ldots, y_n$.

If $m_1, \ldots, m_M$ are M generic indexes s.t. $1 \leq m_1 < \ldots < m_M \leq n$, then the joint probability density of the order statistics $X_{(m_1)}, \ldots, X_{(m_M)}$ is [25, Eq. (2.2.2)]

$$\begin{aligned} p_{X_{(m_1)},\ldots,X_{(m_M)}}(x_1, \ldots, x_M \,;\, n) = \\ \frac{n!}{(m_1-1)!(m_2-m_1-1)!\ldots(n-m_k)!} \\ \cdot (P_Y(x_1))^{m_1-1} p_Y(x_1) \\ \cdot (P_Y(x_2) - P_Y(x_1))^{m_2-m_1-1} p_Y(x_2) \\ \cdot \quad \vdots \\ \cdot (1 - P_Y(x_M))^{n-m_M} p_Y(x_M) \end{aligned} \tag{2}$$

subject to

$$x_1 \leq x_2 \leq \cdots \leq x_M \ .$$

Notice then that, given the continuity assumptions on Y, we may restrict Y to be (0,1)-uniform, so that $p_Y(y) \sim \mathcal{U}[0,1]$. Indeed, it is always possible to transform any non-uniform continuous Y into $Y' = P_Y(Y) \sim \mathcal{U}[0,1]$ by means of the so-called probability integral transform. We can thus eventually consider the equivalent uniform r.v. since it retains the same information content (cf. also Proposition 7 in [9]).

As for the indexes $m_1, \ldots, m_M$, we notice that there are only 3 meaningful arrangements:

1) case $m_1 = 1, \ldots, m_M = M$, so that the considered order statistics are $X_{(1)}, \ldots, X_{(M)}$, i.e., the M smallest IDs. In this case we can observe that $m_1 = 1$, $m_2 - m_1 - 1 = 0$, $m_3 - m_2 - 1 = 0$, ..., $n - m_M = n - M$. Thus the density (2), given that the $y_1, \ldots, y_n$ are i.i.d. realizations from a uniform distribution, particularizes to

$$p\left(x_1, \ldots, x_M \ ; \ n\right) = \frac{n!}{(n-M)!}(1 - x_M)^{n-M}. \tag{3}$$

Since in this case x_M is the M-th smallest element of $y_1, \ldots, y_n$, it follows that $x_M \sim B(M, n - M + 1)$ with $B(\cdot, \cdot)$ the Beta distribution [25, Example 2.3]. Notice also that the structure of this joint density reflects the fact that, conditioned on x_M, the various order statistics x_m with $m < M$ are (0, x_M)-uniform r.v.s;

2) case $m_1 = n - M + 1, \ldots, m_M = n$, so that the considered order statistics are $X_{(n-M+1)}, \ldots, X_{(n)}$, i.e., the M biggest IDs. In this case we can observe that $m_1 = n - M + 1$, $m_2 - m_1 - 1 = 0$, $m_3 - m_2 - 1 = 0$, ..., $n - m_M = 0$. Thus (2) particularizes to

$$p\left(x_1, \ldots, x_M \ ; \ n\right) = \frac{n!}{(n-M)!}x_1^{n-M}. \tag{4}$$

x_1 is now the M-th biggest element of $y_1, \ldots, y_n$ and, similarly to the previous case, is distributed as $B(n - M + 1, M)$ [25, Example 2.3];

3) case $m_1 = 1, \ldots, m_k = k, m_{k+1} = n - M + k + 1, \ldots, m_M = n$, so that the considered order statistics are $X_{(1)}, \ldots, X_{(k)}$ and $X_{(n-M+k+1)}, \ldots, X_{(n)}$, i.e., the k smallest and the $M - k$ biggest IDs. Combining the observations made for the two cases above, we have that in this case (2) particularizes to

$$p\left(x_1, \ldots, x_M \ ; \ n\right) = \frac{n!}{(n-M)!}(x_{k+1} - x_k)^{n-M}. \tag{5}$$

Since x_k and x_{k+1} are respectively the k-th smallest and $(M-k)$-th biggest element of $y_1, \ldots, y_n$, it follows that (again) $(x_{k+1} - x_k) \sim B(n - M + 1, M)$ [25, Example 2.3].

Importantly, as suggested in Section III, the previous 3 cases are the only meaningful ones in our distributed computations setting. Indeed, to compute the M-th biggest element of a given set requires the computation also of the $M - 1$-th, $M - 2$-th, etc., biggest values, that can then be considered as available information when the computation is ended. The same conclusion applies also for the computation of the M-th smallest elements and of ranges.

Given that (3), (4) and (5) have exactly the same functional structure, *estimators derived from the 3 different cases will have the same statistical performance.* In the remainder of this manuscript we thus consider w.l.o.g. the

case where the order statistics correspond to computing maxima over the network. I.e., from now on we assume that x_M is the n-th order statistic of $y_1, \ldots, y_n$ or, equivalently, the biggest ID in the network; x_{M-1} is the $(n-1)$-th order statistic, i.e., the second biggest ID; $\ldots$; x_1 is the $(n-M+1)$-th order statistic, i.e., the M-th biggest ID.

B. The ML estimator of n

For $n \geq M$ the ML estimator of the size can be found by inspection of the likelihood ratio [26]

$$R = \frac{p\left(x_1 \ldots, x_M \,;\, \widetilde{n}\right)}{p\left(x_1 \ldots, x_M \,;\, \widetilde{n}-1\right)} \,.$$

Indeed, given (4) we obtain that

$$\begin{aligned} n_{ML} &:= \arg \max_{\widetilde{n} \in \{M, M+1, \ldots\}} p\left(x_1 \ldots, x_M \,;\, \widetilde{n}\right) \\ &= \max\left\{\widetilde{n} \,|\, R \geq 1\right\} = \text{floor}\left(\frac{M}{1-x_1}\right) \,. \end{aligned} \tag{6}$$

However, a difficulty with (6) is that the probability density $p(n_{ML} \,;\, n)$ (and thus all the statistical performance indexes like $\text{var}\,(n_{ML} - n)$) is not available in closed form and must be computed numerically. Moreover, numerical approaches prevent a full understanding of the properties of the estimator, and complicate design steps such as choosing that design parameter M that leads to specific performance requirements.

C. The ML estimator admits a natural approximation with performance indexes expressible in closed forms

We thus consider an alternative estimator for the cardinality of the network that has performance indexes that can be expressed in closed forms and that represents a natural approximation of the ML estimator (6).

Consider then that if $n \geq M$ then the joint log-pdf of $x_1, \ldots, x_M$ is

$$\log\left(p\left(x_1, \ldots, x_M; n\right)\right) = (n-M)\log\left(x_1\right) + \sum_{k=0}^{M-1} \log(n-k)$$

so that the score is

$$\frac{\partial \log(p)}{\partial n} = \log(x_1) + \sum_{k=0}^{M-1} \frac{1}{n-k} \,. \tag{7}$$

Let then $\psi(\cdot)$ be the digamma function, and exploit the equivalence

$$\begin{aligned} \sum_{k=0}^{M-1} \frac{1}{n-k} &= \sum_{k=0}^{M-1} \frac{1}{(n-M+1)+k} \\ &= \psi(n+1) - \psi(n-M+1) \end{aligned} \tag{8}$$

to express the ML estimator (6) as

$$\arg \max_{\widetilde{n} \in \{M, M+1, \ldots\}} \left|\psi(\widetilde{n}+1) - \psi(\widetilde{n}-M+1) + \log(x_1)\right| .$$

Recalling that

$$\psi(w) = H_{(w-1)} - \gamma \approx \log(w-1) \,, \tag{9}$$

where $H_{(w-1)}$ is the $w-1$-th Harmonic number and γ is the Euler-Mascheroni constant, it follows that the score (7) can be approximated as

$$\frac{\partial \log(p)}{\partial n} \approx \log(x_1) + \log(n) - \log(n - M) \; . \tag{10}$$

In particular, the unique root of the right-hand side of (10) yields an approximation for the unique root of the score, and this legitimates

$$\widehat{n} = g\left(x_1\right) := \frac{M}{1 - x_1} \tag{11}$$

as an approximated form of n_{ML} for the case $n \geq M$.

To extend $\widehat{n}$ so to comprise also the case $n < M$, we consider that in the latter case only n of the M order statistics $x_1, \ldots, x_M$ contain meaningful information, while the remaining ones are arbitrarily set to zero by Algorithm 1. Therefore, a precise estimate is obtained by counting the non-zero entries x_m, $1 \leq m \leq M$. The complete estimator is thus

$$\widehat{n} = \begin{cases} \dfrac{M}{1 - x_1} & \text{if } x_1 \neq 0 \\ |\{x_m \neq 0\}| & \text{otherwise,} \end{cases} \tag{12}$$

where $|\cdot|$ indicates the cardinality of a set.

D. Statistical characterization of the approximated estimator

In the following we let $\widehat{N}$ denote the random variable associated to (12). To characterize its performance in function of the design parameter M we notice that if $n < M$ then $\mathbb{P}\left[\widehat{N} = n\right] = 1$, i.e.,

$$\mathbb{E}\left[\frac{\widehat{N}}{n} \; ; \; n < M\right] = 1, \tag{13}$$

$$\text{var}\left(\frac{\widehat{N} - n}{n} \; ; \; n < M\right) = 0. \tag{14}$$

For the moments $\mathbb{E}\left[\frac{\widehat{N}}{n} \; ; \; n \geq M\right]$ and $\text{var}\left(\frac{\widehat{N}-n}{n} \; ; \; n \geq M\right)$ we exploit the closed form of (11) and consider that the pdf of $\widehat{N}$ can be computed from

$$p_{\widehat{N}}\left(\widehat{n} \; ; \; n \geq M\right) = p_{X_1}\left(g^{-1}\left(\widehat{n}\right) \; ; \; n \geq M\right) \left| \frac{dg^{-1}\left(\widehat{n}\right)}{d\widehat{n}} \right| \; .$$

Since (under $n \geq M$) $X_1 \sim B(n - M + 1, M)$, with B indicating the Beta distribution, it follows that

$$p_{\widehat{N}}(\widehat{n} \; ; \; n \geq M) = \frac{M^M}{B_f(n - M + 1, M)} \frac{(\widehat{n} - M)^{n-M}}{\widehat{n}^{n+1}} \tag{15}$$

where $B_f(\cdot, \cdot)$ is now the Beta function. Then recall that for $X \sim B(\alpha, \beta)$ and $\beta > 2$ there holds

$$\mathbb{E}\left[\frac{1}{1 - X}\right] = \frac{\alpha + \beta - 1}{\beta - 1} \; , \tag{16}$$

$$\text{var}\left(\frac{1}{1 - X}\right) = \frac{\alpha(\alpha + \beta - 1)}{(\beta - 2)(\beta - 1)^2} \; . \tag{17}$$

Given the structure of $g(\cdot)$ in (12), for $2 < M \leq n$ it thus holds that

$$\mathbb{E}\left[\frac{\widehat{N}}{n}\ ;\ n \geq M\right] = \frac{M}{M-1}, \tag{18}$$

$$\begin{aligned}\mathrm{var}\left(\frac{\widehat{N}-n}{n}\ ;\ n \geq M\right) = &\frac{M^2}{(M-2)(M-1)^2} \\ &- \frac{M^2}{n(M-2)(M-1)}\ .\end{aligned} \tag{19}$$

Since we are considering the case $n \geq M$, we conclude that the variance (19) enjoys a $o(1/M)$ behavior asymptotically in M. Moreover

$$\mathrm{var}\left(\frac{\widehat{N}-n}{n}\ ;\ n = M\right) = \frac{M}{(M-2)(M-1)^2}\ , \tag{20}$$

$$\lim_{n\to+\infty} \mathrm{var}\left(\frac{\widehat{N}-n}{n}\right) = \frac{M^2}{(M-2)(M-1)^2}\ , \tag{21}$$

In other words, estimator (12) exhibits the following feature: if $n < M$, then the estimate is perfect; if instead $n \geq M$ but $n \approx M$, then the variance of the error behaves approximately as in (20); eventually, for $n \gg M$ the variance of the error becomes about M times bigger (cf. Figure 1). We then notice the following:

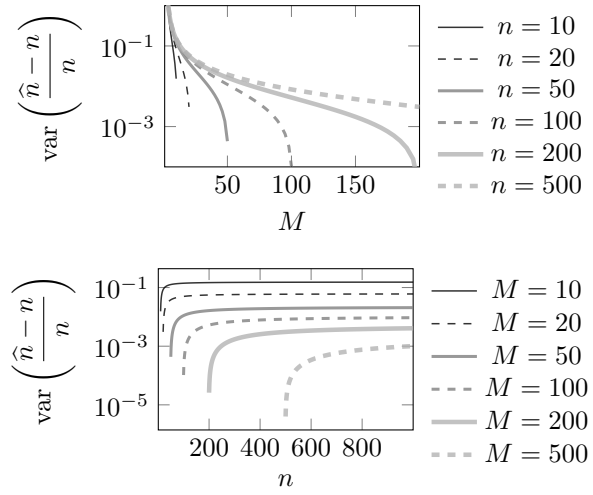

Fig. 1. Variance of the relative estimation error (19) as a function of n and M. As n approaches M with $n \geq M$, the estimator performance sees an M-fold improvement compared to (23). For $n < M$, instead, the relative error variance vanishes.

Remark 2 *The statistical performance indexes of the max-consensus based size estimator used as a building block in [9], [18], [19], [27], [20] are (calling this estimator N_{max}):*

$$\mathbb{E}\left[\frac{N_{max}}{n}\right] = \frac{M}{M-1}, \tag{22}$$

$$\mathrm{var}\left(\frac{N_{max}-n}{n}\right)=\frac{M^2}{(M-2)(M-1)^2} \tag{23}$$

i.e., asymptotically the same of the proposed estimator $\widehat{N}$ when $n \gg M$. *Therefore, the strategy proposed in this manuscript has overall better statistical properties, and in addition, has the (meaningful) property of letting nodes estimate perfectly networks sizes when the total number of nodes is smaller than the amount of shareable information. I.e., the novel estimator* $\widehat{N}$ *not only has a better variance, but it can also work as a perfect counter, while* N_{max} *cannot.*

V. Testing hypotheses on the cardinality of the network

We now consider how nodes can decide if the network cardinality n is above (or below) a given threshold $\overline{n}$ starting from the knowledge of the order statistic $x_{(n-M+1)}$ computed by Algorithm 1. We formalize this problem in the classical statistical hypothesis testing framework; to maintain the paper self-contained we then summarize the needed theoretical background in Section V-A. In Section V-B, instead, we characterize the optimal decision rule and study its statistical power as a function of the design parameter M.

A. Preliminaries in statistical hypothesis testing

(See [28] and the dedicated literature for more details.) A *hypothesis* is a statement about a parameter of a probability distribution. A (deterministic) *hypothesis test* is a deterministic rule that decides, based on observed samples, whether a given hypothesis should be accepted (i.e. considered true) or rejected at a certain level of significance.

Let $\{p_\theta\}_{\theta\in\Theta}$ be a family of parametric probability densities, X a r.v. with density $p_{\overline{\theta}}$ for some unknown $\overline{\theta}\in\Theta$, and Θ the domain of the potential parameters. We assume Θ to be divided into the two complementary sets (or hypotheses)

$$\mathcal{H}_i := \{x\sim p_\theta \ \text{ with } \ \theta\in\Theta_i\}\,, \ \ i=0,1 \tag{24}$$

with $\Theta_0\cap\Theta_1=\emptyset$ and $\Theta_0\cup\Theta_1=\Theta$.

A (deterministic) *test* to decide between the two hypotheses in (24) is thus a deterministic function $\phi(x)$: range$(X)\mapsto\{0,1\}$ that maps a generic realization of X into an integer that indicates the acceptance or rejection of $\mathcal{H}_0$. When $\phi(x)$ selects $\mathcal{H}_1$ while $\mathcal{H}_0$ is true the test is said to commit an *error of type I* (false positive). Accepting $\mathcal{H}_0$ when $\mathcal{H}_1$ is true is instead said to be an *error of type II* (false negative).

To statistically characterize the effectiveness of test $\phi(x)$ it is common to refer to the function

$$\beta_\phi(\theta) := \mathbb{E}_\theta\left[\phi(x)\right] = \int \phi(x)p_\theta(x)dx\ , \quad \forall\,\theta\in\Theta\ , \tag{25}$$

called the *power function* of ϕ, that characterizes the statistical performance of ϕ through

$$\alpha_0(\phi) := \sup_{\theta\in\Theta_0}\beta_\phi(\theta)\ , \tag{26}$$

also called the *size* or *level of significance* of ϕ. The size $\alpha_0(\phi)$ thus represents the worst probability of errors of type I given all the possible situations for which $\theta \in \Theta_0$. If $\theta \in \Theta_1$, instead, $\beta_\phi(\theta)$ represents the probability of *not* committing errors of type II for that particular θ.

The concept of optimality for a test is then usually expressed in terms of its power function: a test $\phi(\cdot)$ is indeed said to be Uniformly Most Powerful (UMP) if there exists no other test $\phi'(\cdot)$ that operates on the same hypotheses, has the same size (i.e., is s.t. $\alpha_0(\phi') = \alpha_0(\phi)$), and has a better power in the Θ_1 region. In other words, $\phi(\cdot)$ is UMP if every other $\phi'(\cdot)$ with the same size of $\phi(\cdot)$ satisfies

$$\beta_\phi(\theta) \geq \beta_{\phi'}(\theta) \quad \forall\, \theta \in \Theta_1 \ .$$

B. A UMP test for one-sided hypotheses on the network cardinality

Let now x be as in (1), and let its density be as in (4). Let the hypotheses on the size of the network n be (notice that here n plays the role of θ above):

$$\begin{cases} \mathcal{H}_0: \quad n \in \Theta_0 := \{\nu : \nu \leq \overline{n}\}, \\ \mathcal{H}_1: \quad n \in \Theta_1 := \{\nu : \nu > \overline{n}\}. \end{cases} \tag{27}$$

Notice that (27) is parameterized in the deterministic value $\overline{n} > 0$, representing a threshold size whose meaning depends on the specific application (e.g., the minimum size of the population of nodes that guarantee a certain quality of service). In this context, the decision rule takes as its input the whole vector x computed through Algorithm 1, and outputs either zero (i.e., $\mathcal{H}_0$) or one (i.e., $\mathcal{H}_1$).

Consider then that if $n < M$, then the test can perfectly discriminate between $\mathcal{H}_0$ and $\mathcal{H}_1$, since in this case it is possible to estimate the size n perfectly. For the case $n \geq M$ we instead need to design the optimal (in terms of power functions) test given x.

To this aim, observe that the one-parameter family of exponential densities (4) induces likelihood ratios of the form

$$\begin{aligned} \Lambda(x_1, \ldots, x_M, n_1, n_2, M) &= \frac{p(x_1, \ldots, x_M \ ; \ n_1, M)}{p(x_1, \ldots, x_M \ ; \ n_2, M)} \\ &= x_1^{n_1 - n_2} \prod_{\nu = n_2 + 1}^{n_1} \frac{\nu}{\nu - M}. \end{aligned} \tag{28}$$

As soon as $n_1 > n_2 > 0$, Λ is strictly increasing in the sufficient statistic x_1. This monotonicity property constitutes a sufficient condition that guarantees (see [28, Thm 3.4.1]) that for every desirable size α_0 there exists a corresponding UMP test $\phi(\cdot)$ structurally defined by

$$\phi(x_1) = \begin{cases} 0 & \text{if } x_1 \leq \lambda \\ 1 & \text{otherwise} \end{cases} \tag{29}$$

with $\lambda > 0$ an opportune threshold, and s.t. its size coincides with the test power evaluated at the frontier point $\overline{n}$, i.e.,

$$\alpha_0(\phi) = \beta_\phi(\overline{n}) \ . \tag{30}$$

In our specific case, to construct the UMP test it is thus sufficient to compute the corresponding threshold λ as a function of the desired size α_0.

Let then the desired α_0 be given. It follows from (29) that λ is a function of the quantile function of a Beta r.v. More specifically, if we denote with $Q(u\ ;\ a, b)$ the quantile of a generic Beta distribution $B(a, b)$ then the optimal λ is

$$\lambda = Q\left(1 - \alpha_0\ ;\ \overline{n} - M + 1, M\right). \tag{31}$$

Although $Q(\cdot\ ;\ a, b)$ cannot be expressed in closed form, it admits a power series expansion that can be exploited to compute (31) efficiently, see, e.g., [29]. Moreover, the value of λ in (29) is fixed once α_0 and $\overline{n}$ have been chosen and, therefore, it can be computed off-line and stored in the nodes before deployment.

Combining the above considerations, and removing the restriction $n \geq M$, we finally obtain the following UMP rule:

$$\phi(x_1) = \begin{cases} 0 & \text{if either } \left(\widehat{n} < M \text{ and } \widehat{n} \leq \overline{n}\right) \\ & \quad \text{or } \left(\widehat{n} \geq M \text{ and } x_1 \leq \lambda\right), \\ 1 & \text{otherwise.} \end{cases} \tag{32}$$

A graphical description of the performance of (32) is shown in Figure 2, where we consider $\overline{n} = 50$ and significance levels 0.05 and 0.01. Confirming the intuitions, once a certain choice of the size α_0 is made, different choices of the remaining design parameter M lead to either poor power functions when M is very small or very good power functions when M is big. Indeed, increasing M leads to more information available for inference purposes, that translates into an improved test power. In general, values of M near $\overline{n}$ yield good performance in terms of both the variance (19) of the relative error of the point estimator (12) (cf. Figure 1) and of the power of the here discussed UMP rule.

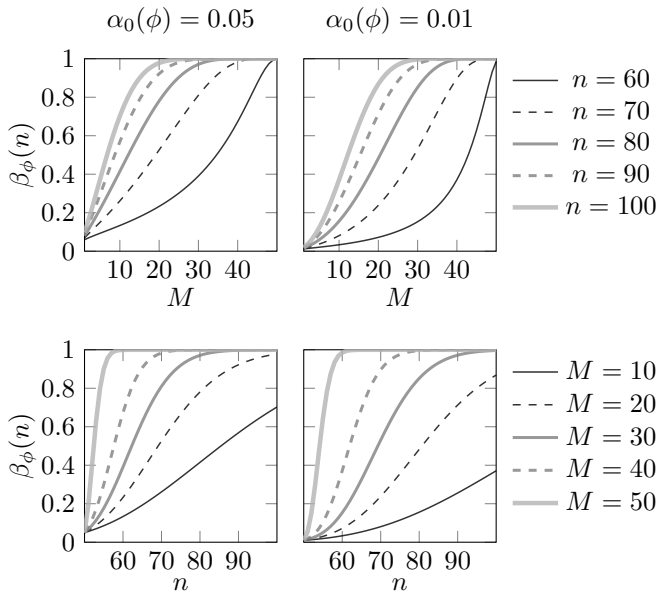

Fig. 2. The power of the UMP decision rule (32) as a function of n and M for $\overline{n} = 50$ and two different values for the size, $\alpha_0(\phi) = 0.05$ for the plots in the first column and $\alpha_0(\phi) = 0.01$ for the plots in the second column. Notice that if $n < M$, then the power is one since (32) discriminates perfectly between hypotheses (27).

VI. Numerical experiments

Before continuing we recall the concept of k-steps neighborhood: given a generic node $i \in V$ and $k \in \mathbb{N}$, the k-steps neighborhood of i is the set of nodes connected to i by at least one path of at most k links, and that is formally defined for $k = 0$ as $V_i^0 := \{i\}$ and for $k \geq 1$ through the recursion

$$V_i^k := \bigcup_{(i,j)\in E} V_j^{k-1} \,. \tag{33}$$

To show the effectiveness of the point estimator (12) and of the hypothesis test (32) we consider the following application: in the tree network of Figure 3 each node aims at estimating how many k-steps neighbors they have for $k = 1, \ldots, 6$. Moreover, each node aims also at deciding whether it has at least $\overline{n} = 100$ 6-steps neighbors or not.

To solve this problem we propose to opportunely parallelize D instances of Algorithm 1 so that nodes can generate statistical information on the size of the various k-steps neighborhoods up to distance D (a design parameter). More specifically, we let the state of the generic agent i be a matrix ${}^i x \in \mathbb{R}^{D\times M}$ and aim at letting the k-th column of ${}^i x$ aggregate information from i's k-steps neighbors.

Consider then the following scheme: *i)* during each epoch $t = 0, 1, 2, \ldots$, node i performs the same operations described in Algorithm 1 but separately on each column of the new augmented state; *ii)* to accommodate the recursive step in (33), at the beginning of each epoch, node i shifts the columns of ${}^i x$ by one to the right (so that

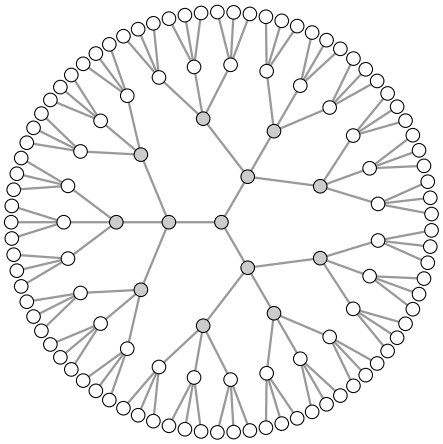

Fig. 3. A balanced tree with 5 levels and a total of 121 nodes. If a node has a darker interior then this means that it believes (through the UMP test defined in this section) that its 6-steps neighborhood contains at least $\overline{n} = 100$ nodes.

the old D-th column is effectively discarded), while the leftmost column is reinitialized with a new random ID as in step 2 of Algorithm 1.

This mechanism allows the generic node i to produce at each epoch t: *1)* a local estimate $\widehat{n}_i^k(t)$ of the cardinality of the generic neighborhood V_i^k by exploiting (12); *2)* a decision between alternative hypotheses on the cardinality of the generic neighborhood V_i^k by exploiting (32).

To address the application described above in the network of Figure 3, we thus let every node independently perform a statistical test to decide if its 6-steps neighborhood contains more than $\overline{n} = 100$ nodes or not. We considered $M = 80$ and set the level of significance to 0.01, thus bounding the rate of type I errors. In the figure, the nodes drawn in a darker color are those that set an alarm after evaluating the hypothesis test. Notice that all the nodes of the network correctly inferred whether their 6-steps neighborhood contains more then 100 peers.

In Figure 4, instead, we depict two typical realizations of the evolutions of the estimates $\widehat{n}_i^k(t)$,$k = 1, \ldots, 4$, for the node i in the center of the network. Confronting the panels it is possible to notice the main feature of the cardinality estimator proposed in this paper, i.e., the fact that if $\left|V_i^k\right| < M$, then the estimator acts as a counting mechanism. If instead $\left|V_i^k\right| \geq M$, then the estimator is a proper estimator, in the sense that it is not perfect and its statistical performance depend on M as described in (19).

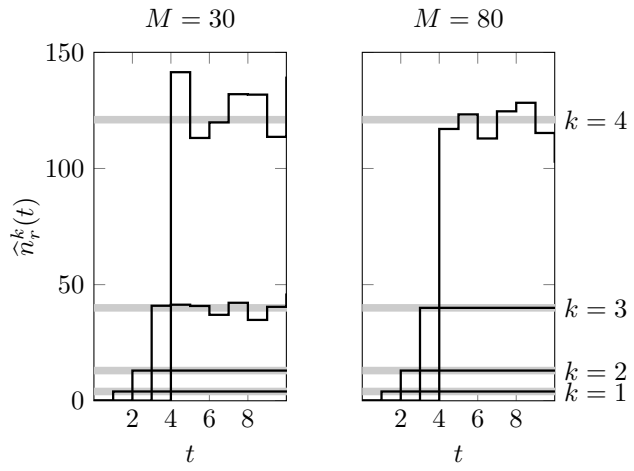

Fig. 4. Typical realizations of the evolutions of the estimators $\widehat{n}_i^k(t)$ of the sizes of the k-steps neighborhoods for $k = 1, \dots, 4$. The left panel, corresponding to $M = 30$, shows average estimation errors that are noticeably bigger than the ones committed in the right panel, for which $M = 80$. Moreover, since $30 < \left|V_i^3\right| < 80$, for the right panel the estimator of the 3-steps neighborhood works as a counting mechanism, whereas for the left panel it does not. The horizontal gray lines indicate the exact size of the considered k-steps neighborhoods after the initial transitory.

VII. Conclusions

This paper presents a novel counting strategy that is tailored for distributed networked applications and that enjoys several desirable practical properties, such as uniformly bounded per-transmission communication requirements, trivial computational requirements, perfect precisions for small networks, and fast convergence times.

The structure of the computational procedure, that is based on the distributed computation of order statistics, enables the derivation of both point and interval estimators and their complete statistical characterization. More precisely, point estimators are derived from approximated Maximum Likelihood (ML) concepts, while interval estimators are derived in the forms of opportune hypothesis tests on the size of the underlying network.

For small networks the point estimator works as a counting mechanism, and this represents an improvement with respect to the existing literature. The statistical performance of the estimator compare favorably to that of other size estimation technique that are based on max consensus operations and have the same convergence properties.

The novel technique proposed here opens up several research directions. We devise specially the following ones: *i)* studying the properties of the estimators when considering discretization effects; *ii)* finding the potential extensions that can be implemented when removing the requirement that every node should have the same final estimate; *iii)* understanding which other topological quantity can be computed by nodes while running this estimation strategy.

Acknowledgments: We thank Federica Garin for the insightful discussions and her comments on a previous version of the manuscript.

REFERENCES

[1] D. Varagnolo, G. Pillonetto, and L. Schenato, "Distributed parametric and nonparametric regression with on-line performance bounds computation," *Automatica*, vol. 48, no. 10, pp. 2468–2481, 2012.

[2] B. Codenotti, P. Gemmell, P. Pudlák, and J. Simon, "On the Amount of Randomness Needed in Distributed Computations," CNR Italy, Tech. Rep., Sept. 1997.

[3] C. Gkantsidis, M. Mihail, and A. Saberi, "Random walks in peer-to-peer networks: algorithms and evaluation," *Performance Evaluation*, vol. 63, no. 3, pp. 241–263, Mar. 2006.

[4] L. Massouliè, E. L. Merrer, A.-M. Kermarrec, and A. Ganesh, "Peer counting and sampling in overlay networks: random walk methods," in *Proceedings of the twenty-fifth annual ACM symposium on Principles of distributed computing*, 2006, pp. 123–132.

[5] B. Ribeiro and D. Towsley, "Estimating and sampling graphs with multidimensional random walks," in *Proceedings of the 10th annual conference on Internet measurement*, 2010.

[6] D. Kostoulas, D. Psaltoulis, I. Gupta, K. P. Birman, and A. J. Demers, "Active and passive techniques for group size estimation in large-scale and dynamic distributed systems," *The Journal of Systems and Software*, vol. 80, no. 10, pp. 1639–1658, Oct. 2007.

[7] M. Kurant, C. T. Butts, and A. Markopoulou, "Graph Size Estimation," *arXiv*, Oct. 2012.

[8] D. Varagnolo, L. Schenato, and G. Pillonetto, "A variation of the Newton-Pepys problem and its connections to size-estimation problems," *Statistics & Probability Letters*, vol. 83, no. 5, pp. 1472–1478, 2013.

[9] D. Varagnolo, G. Pillonetto, and L. Schenato, "Distributed cardinality estimation in anonymous networks," *IEEE Transactions on Automatic Control*, vol. 59, no. 3, pp. 645–659, 2014.

[10] I. Shames, T. Charalambous, C. N. Hadjicostis, and M. Johansson, "Distributed Network Size Estimation and Average Degree Estimation and Control in Networks Isomorphic to Directed Graphs," in *Allerton Conference on Communication Control and Computing*, 2012.

[11] F. Morbidi and A. Y. Kibangou, "A Distributed Solution to the Network Reconstruction Problem," *Systems and Control Letters*, vol. 70, pp. 85–91, 2014.

[12] C. Budianu, S. Ben-David, and L. Tong, "Estimation of the number of operating sensors in large-scale sensor networks with mobile access," *IEEE Transactions on Signal Processing*, vol. 54, no. 5, pp. 1703–1715, May 2006.

[13] P. Flajolet, E. Fusy, O. Gandouet, and F. Meunier, "HyperLogLog: the analysis of a near-optimal cardinality estimation algorithm," in *Analysis of Algorithms*, 2007.

[14] P. Flajolet and G. N. Martin, "Probabilistic counting algorithms for data base applications," *Journal of Computer and System Sciences*, vol. 31, no. 2, pp. 182–209, Oct. 1985.

[15] F. Garin and Y. Yuan, "Distributed privacy-preserving network size computation: A system-identification based method," in *IEEE Conference on Decision and Control*, Dec. 2013, pp. 5438–5443.

[16] O. Sluciak and M. Rupp, "Network Size Estimation Using Distributed Orthogonalization," *IEEE Signal Processing Letters*, vol. 20, no. 4, pp. 347–350, Apr. 2013.

[17] A. Coluccia and G. Notarstefano, "Distributed estimation of binary event probabilities via hierarchical Bayes and dual decomposition," in *52nd IEEE Conference on Decision and Control*, Dec. 2013, pp. 6753–6758.

[18] C. Baquero, P. S. S. Almeida, R. Menezes, and P. Jesus, "Extrema Propagation: Fast Distributed Estimation of Sums and Network Sizes," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 4, pp. 668 – 675, Apr. 2012.

[19] H. k. Terelius, D. Varagnolo, and K. H. Johansson, "Distributed size estimation of dynamic anonymous networks," in *IEEE Conference on Decision and Control*, 2012.

[20] J. Cichon, J. Lemiesz, W. Szpankowski, and M. Zawada, "Two-Phase Cardinality Estimation Protocols for Sensor Networks with Provable Precision," in *IEEE Wireless Communications and Networking Conference*, Paris, France, Apr. 2012.

[21] Z. Bar-Yossef, R. Kumar, and D. Sivakumar, "Reductions in streaming algorithms, with an application to counting triangles in graphs," in *ACM-SIAM symposium on Discrete algorithms on Discrete algorithms*, 2002, pp. 623–632.

[22] P. Chassaing and L. Gerin, "Efficient estimation of the cardinality of large data sets," *arXiv*, Jan. 2007.

[23] F. Giroire, "Order statistics and estimating cardinalities of massive data sets," *Discrete Applied Mathematics*, vol. 157, pp. 406–427, 2009.

[24] J. Lumbroso, "An optimal cardinality estimation algorithm based on order statistics and its full analysis," in *International Meeting on Probabilistic, Combinatorial, and Asymptotic Methods in the Analysis of Algorithms*, 2010.

[25] H. A. A. David and H. N. N. Nagaraja, *Order Statistics*. Wiley series in Probability and Statistics, 2003.

[26] N. L. Johnson, "Estimation of Sample Size," *Technometrics*, vol. 4, no. 1, pp. 59–67, 1962.

[27] M. Albano, N. Pereira, and E. Tovar, "How many are you (an approach for the smart dust world)?" in *IEEE Conference on Cyber-Physical Systems, Networks, and Applications*. IEEE, Aug. 2013, pp. 101–105.

[28] E. Lehmann and J. P. Romano, *Testing Statistical Hypotheses*. Springer, 2005.

[29] G. Steinbrecher and W. T. Shaw, "Quantile mechanics," *European Journal of Applied Mathematics*, vol. 19, no. 02, pp. 87–112, 2008.