Size estimation and topology change-dection in anonymous networks

Riccardo Lucchese and Michele Savietto

July 25, 2012

Abstract

In this report we consider two problems. The first one is the distributed estimation of the number of nodes in a network under the constraint of anonymity. We restrict our discussion to estimation strategies in which statistical information on the network size is generated by exploiting consensus. In particular, we recall from [14] two different size estimators, one based on average-consensus and the other on max-consensus. In one case we are able to provide a refined bound for the estimator error probability that sligtly improves the bound in [14] (see also [15]).

The second problem we analyze is that of distributedly detecting topology changes by employing an hypothesis test. To this purpose we formulate a Generalized Likelihood Ratio (GLR) test based on the max-consensus estimation strategy and characterize it in terms of type I and type II error probabilities.

Aided by a set of simulations we then comment on the role played by the test tunable parameters on the performances of the change detection algorithm (e.g., the rate of false positives).

1 Introduction

In the past years the problem of estimating the size of a network with limited knowledge on its topology and dynamics has been increasingly gaining momentum. Practical interest comes from the fact that on one hand it is often not possible, or not practical, to track precisely changes in the time varying structure of the network. On the other hand there is a plethora of services which are enabled or rely on such estimates for achieving better performances. Example applications range from cost-effective maintenance in

(relatively) large networks to intelligent GPS routing that accounts for traffic data.

Our reference setting is that of the so called anonymous networks. By definition in an anonymous network it is impossible to distinguish between the composing nodes and, as a consequence, it is impossible for any node to uniquely identify its peers (see [7] for the definition and a discussion on what kind of functions can be computed by anonymous networks).

The condition of anonymity characterizes the estimation problem in a critical way. It can be shown that in this context, once the rather sensible assumptions of arbitrary network topology and finite computational resources are introduced, there is no algorithm with average limited (computational) complexity that can provide correct estimates with probability one [8]. It follows that in any sufficently general estimation scheme the resulting estimate is a non-deterministic function of the true size.

The scientific literature offers many partial solutions to this estimation problem. All these approaches are mainly characterized by how nodes generate statistical information on the network size and can be grouped into three main classes: random walks, sampling methods and consensus methods.

In random walk schemes a token is exchanged in some random way between active nodes. The time-of-return, i.e. the time interval between two consecutive passages of the token at the same node, can be shown to be statistically correlated with the network size and used for statical inference. In a variant, the token is endowed with a counter which gets decreased at each hop; then the statistical information is provided by the token's time-to-vanish (see e.g. [9]).

In sampling methods a master marks a (random) subset of the active nodes with certain properties, e.g. by assigning them IDs [10]. In some schemes these markings are eventually propagated throughout the network. The master can then draw an estimate of the network size by querying a subset of the network asking those nodes whether they own a certain mark or not.

Our focus is put instead on consensus based methods. In brief, each node in the network starts by generating a local vector of data by sampling from a common probability density. Then the network computes a function of these initial values by iterating consensus. The key idea is that the outcome of consensus is statistically correlated with the network size and can be used in a statistical inference procedure [13, 14, 15].

In the second part of this report we concentrate on the problem of distributedly detecting topology changes. Although this is not a recent field of reserch, it is difficult to find material in the scientific literature dealing with the constraint of anonymity. There is apparently much more material

dedicated to the analysis of how the network can react and or re-configure itself once a change in topology has been detected (see e.g. [4]).

We will consider an hypothesis testing approach to the change detection problem. Our contribution is a test, in the form of a GLR, which exploits the max-consensus estimator from [14] and which we are able to statistically characterize in terms of type I and type II error probabilities. As we shall see, our approach GLR exhibits some optimality properties of practical relevance.

The structure of the report is the following. In Section 2 we consider the case of network with static topology and recall two size estimator from [14]: the first based on average consensus and the second using max consensus. For the average consensus (paired with a common Bernoulli distribution) case we provide a result that slghtly refines refines the bound on the estimator error probability in [14] and [15]. In Section 3 we introduce the concept of k-steps neighborhood and provide an estimation algorithm that generalizes the max-consensus estimator from static to dynamic networks. In Section 4 we outline our GLR approach to change detection and in Section 5 we comment on a set of simulations in which the algorithm is run distributedly on the nodes a prototypical grid network.

A word on notations. In the following we let bold fonts indicate vectors, plain italic fonts indicate scalars, capitalized plain italic fonts indicate matrices. For notational simplicity we will often identify agents with the fictitious indexes i and j.

2 Estimating the size of a static network

In this section we recall some results on the problem of estimating the size of an anonymous network by exploiting consensus. The analysis is restricted to connected networks with static topology. To each such network we associate a graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ where $\mathcal{N} \subset \mathbb{N}^+$ is the set of nodes and $\mathcal{E} \subset \mathcal{N}^2$ is the set of undirected edges such that $(i, j) \in \mathcal{E}$ if node i and node j can communicate (bi-directionally). The true size of the network, $S = |\mathcal{N}|$, is a deterministic but unknown parameter that we wish to estimate.

The general estimation strategy in consensus based methods comprises three main steps [13, 14]:

i) each node $i \in \mathcal{N}$ starts by generating a local vector of initial random values, $\boldsymbol{y}^{(i)} \in \mathbb{R}^M$, by sampling M i.i.d. random variables with common probability density $p(\cdot)$.

ii) the network distributedly computes a consensus function of these initial values by local communication alone and without breaking anonymity. Asymptotically each node reaches consensus on the quantity

$$\boldsymbol{f} = F\left(\left\{\boldsymbol{y}^{(i)} \ : \ i \in \mathcal{N}\right\}\right) \ ,$$

where $F(\cdot)$ is the asymptotic consensus function.

iii) each node statistically infers an estimate of S by exploiting the correlation between the consensus outcome and the network size.

This procedure can be specialized by fixing the means by which the information is generated in step i and those by which is mixed in step ii, i.e. the common density $p(\cdot)$ and the consensus function $F(\cdot)$ respectively. Since one tipically assumes no a-priori knowledge on the network topology, the guess drawn at step iii usually reduces to computing the Maximum Likelihood (ML) estimate

$$\widehat{S} := \arg\max_{\mathbb{R}^+} \ell\left(S \ ; \ \boldsymbol{f}\right) \ ,$$

where we denoted by $\ell(a \ ; \ b)$ the likelihood of a given that b is observed.

The authors of [14] devised three such estimators: two of which are based on average consensus paired with common Gaussian and Bernoulli distributions respectively, and a third exploiting max consensus with a common uniform distribution.

Remarkably, the estimator derived for the Bernoulli distribution case is, to our present knowledge, the only in the literature with an error probability decreasing exponentially fast with the number of local initial values M. In

Section 2.1 we complement the reasoning and numerical results given in [14] by providing a slightly refined bound for this error probability: the intuition behind our approach is that the estimator error-probability can be computed in closed form when S is a power of 2. However, due to the discrete nature and high non-linearity of the estimator map, it is difficult to provide a closed form probabilistic characterization for it. Moreover its accuracy depends in a critical way on the hypothesis that the network topology does not change during the consensus phase[1], rendering it impractical.

On the other hand the size estimator based on max-consensus is a continuous function of observations with a closed form probabilistic characterization and exhibits both fast convergence rate and high scalability. Moreover, this method is naturally extended to the size estimation of k-steps neighborhoods (see Section 3). After Section 2.1 the effort of this report is mostly focused on this max-consensus based estimator.

Following [14] we will henceforth assume the further simplified context in which the effects of packet loss and quantization errors are negligible.

2.1 Average consensus and common Bernoulli distribution

In this case the common density has a Bernoulli distribution with success probability p and the consensus function is the average

$$\boldsymbol{f} := \frac{1}{S} \sum_{i \in \mathcal{N}} \boldsymbol{y}^{(i)} \ . \tag{1}$$

Since Sf_m has a Binomial distribution, i.e. $Sf_m \sim \mathrm{Bin}\,(S, p)$, it follows immediately that

$$\mathbb{P}\left[f_m \ ; \ S\right] = \begin{cases} \binom{S}{f_m S} p^{f_m S} (1-p)^{S - f_m S} & \text{if } f_m S \in \{0, \dots, S\} \\ 0 & \text{otherwise ,} \end{cases} \tag{2}$$

which, by independency, fully determines the distribution of $\boldsymbol{f}$. Now suppose that f_m is observed. It follows from (2) that the likelihood of f_m as a function of S has (unbounded) support on the discrete set

$$\text{supp}\ \ell\left(S \ ; \ f_m\right) = \left\{ \nu \bar{S} \ : \ \nu \in \mathbb{N}^+, f = \frac{\bar{k}}{\bar{S}} \text{ with } (\bar{k}, \bar{S}) \text{ coprime} \right\} \ .$$

[1]It must be ensured that after a sufficient number of consensus steps the network reaches practical consensus on a point where the estimator map is defined.

By first showing that $\mathbb{P}[f_m\ ;\ S] \geq \mathbb{P}[f_m\ ;\ \nu S]$ for all $\nu \in \mathbb{N}^+$ one can prove that the ML estimator of S is the statistic [14, 15]

$$\widehat{S}(\boldsymbol{f}) = \min\left(\bigcap_{m=1}^{M} \text{supp}\,\ell\,(S\ ;\ f_m)\right)\ . \tag{3}$$

$\widehat{S}(\boldsymbol{f})$ is thus the smallest size which can explain the M independent observations, $f_1, \ldots f_M$, in terms of (1). Due to the discrete nature of the initial distribution and the discontinuousness of (3) it is difficult to provide a closed form for the estimator distribution. A characterization of the estimator's error probability, i.e. the probability of incorrectly estimating the size, can be given by means of the following argumentation. Define

$$\alpha(p) = \mathbb{P}\left[\min\left(\text{supp}\,\ell\,(S\ ;\ f_m)\right) \neq S\ ;\ S\right]\ . \tag{4}$$

Then $\alpha(p)$ is the exact error probability of (3) when $M = 1$. In [15] it is claimed that this probability can be bounded from above by $\overline{\alpha} = 1 - \phi(S)/S$, where $\phi(x)$ the Euler phi-function counting the number of totatives of the natural x. The reasoning exploits the following facts:

i) the distribution of the totatives of S (for S large enough) is approximately uniform;

ii) in the case of $M = 1$, whenever the outcome of $\sum_{i=1}^{S} y^{(i)}$ is a totative of S then the ML estimator provides the correct size.

Since $\phi(S)/S < 0.15$ at least up to $S = 10^{10}$ (see [15]) one can argue that as long as p is far from zero or one than $\alpha(p) < 0.85$. Thus an upper bound for the general case, $M \geq 1$, is found by exploiting independency

$$\mathbb{P}\left[\widehat{S}(\bar{\boldsymbol{f}}) \neq S\ ;\ S\right] \leq \overline{\alpha}^M\ . \tag{5}$$

The intent of the remaining part of this chapter is to provide a refinement to this error probability. Our bound vanishes asymptotically faster in M than (5) and provides some (minimal) insight on the role of the design parameter p.

Define $\psi : \mathbb{Q} \to \mathbb{N}^+$ as the application mapping each rational into the value taken by the denominator of its coprime representation with the convention $\psi(0) \mapsto 1$. Define the r.v.s. $d_m = \psi(f_m)$, $m = 1, \ldots, M$ and consider the (unique) prime-number factorizations

$$S = 2^{\gamma_2} 3^{\gamma_3} 5^{\gamma_5} ..., \quad d_m = 2^{\gamma_2^m} 3^{\gamma_3^m} 5^{\gamma_5^m} ... \quad m = 1, \ldots, M\ .$$

We will denote by primes(n) the set of distinct prime numbers in the prime-number factorization of $n \in \mathbb{N}^+$. Then the ML estimator in (3) provides the correct size if and only if the lowest common multiple of the M observations $d_1, \ldots, d_M$ is exactly S. In turn this happens if and only if for each prime number $\nu \in \text{primes}(S)$ there exist at least one $m \in \{1, \ldots, M\}$ such that ν^{γ_ν} divides d_m, in symbols $\nu^{\gamma_\nu} \mid d_m$. It follows immediately that

$$\mathbb{P}\left[\widehat{S} \neq S \;;\; S\right] \leq \sum_{\nu \in \text{primes}(S)} \prod_{m=1}^{M} \mathbb{P}\left[\nu^{\gamma_\nu} \nmid d_m \;;\; S\right] \;. \tag{6}$$

In order to provide a further bound on the rightmost term we shall first introduce two small lemmas.

Lemma 1. *Consider the following inequality*

$$\alpha p^{k-1}(1-p)^{n-k+1} - \beta p^k (1-p)^{n-k} + \gamma p^{k+1}(1-p)^{n-k-1} \geq 0 \;,$$

where $p \in [0,1]$, $\alpha, \beta, \gamma, n \in \mathbb{R}^+$, *and* $k \in [1, n-1]$ *is real. A sufficient condition for the inequality to hold is given by* $\beta^2 \leq 4\alpha\gamma$.

Proof. For $p \in \{0,1\}$ it is trivial. When $p \in (0,1)$ the inequality reduces to the simpler

$$\alpha(1-p)^2 - \beta p(1-p) + \gamma p^2 \geq 0 \;, \tag{7}$$

and the polynomial on the left side has roots in

$$p_{1,2} = \frac{2\alpha + \beta \pm \sqrt{\beta^2 - 4\alpha\gamma}}{2(\alpha + \beta + \gamma)} \;.$$

Whenever the determinant vanishes, i.e. $\beta^2 = 4\alpha\gamma$, the quadratic form in (7) evaluates to

$$\left(\sqrt{\alpha}(1-p) - \sqrt{\gamma}p\right)^2 \;,$$

which is clearly non-negative. On the other hand when $\beta^2 < 4\alpha\gamma$ the two roots $p_{1,2}$ are complex and by continuity reasons the inequality must still hold true. □

Lemma 2. *Consider a r.v.* b *distributed as* $\text{Bin}(n,p)$. *For all* $p \in [0,1]$ *the following inequality holds*

$$\mathbb{P}[b = k-1] - \mathbb{P}[b = k] + \mathbb{P}[b = k+1] \geq 0 \quad , \quad k = 1, \ldots, n-1 \;.$$

Proof. By Lemma 1 a sufficient condition is given by

$$\binom{n}{k}^2 \leq 4\binom{n}{k-1}\binom{n}{k+1} ,$$

and by explicitely computing the binomial coefficients one obtaines the equivalent condition

$$k^2 - kn + \frac{n+1}{3} \leq 0 .$$

The polynomial on the left, seen as a function of the real variable k, has just one critical point in $n/2$, corresponding to a global minimum. By evaluating the polynomial in $k = 1$ and $k = n - 1$ one finds that the inequality is true as long as $n \geq 2$, thus concluding the proof. □

We are now in a position to prove the main result of this section.

Proposition 1. *Let $b \sim \mathrm{Bin}(S, p)$, $d = \psi(b/S)$ and consider the following prime-numbers factorization*

$$S = \prod_{\nu \in primes S} \nu^{\gamma_\nu} . \tag{8}$$

Then for any $p \in \left[0, S/(S+1)\right]$ the probability that a factor ν^{γ_ν} does not divide d, i.e. $\mathbb{P}\left[\nu^{\gamma_\nu} \nmid d\right]$, is less or equal then the probability that b is even.

Proof. For any ν^{γ_ν} in the factorization (8) it holds

$$\mathbb{P}\left[\nu^{\gamma_\nu} \nmid d\right] = \mathbb{P}\left[b \in \left\{k\nu \ : \ k = 0, 1, \ldots, \frac{S}{\nu}\right\}\right] . \tag{9}$$

Consider the case of S even. For $\nu = 2$ the probability in (9) is exactly the probability $\mathbb{P}\left[b \text{ is even}\right]$. For any other prime factor $\nu'^{\gamma_{\nu'}}$ of S, $\nu' > 2$, we have

$$\mathbb{P}\left[\nu'^{\gamma_{\nu'}} \nmid d\right] = \mathbb{P}\left[b = 0\right] + \sum_{k \,:\, \nu \leq k\nu < S} \mathbb{P}\left[b = k\right] + \mathbb{P}\left[b = S\right] , \tag{10}$$

and this quantity can be bounded by exploiting Lemma 2 for all the probabilities appearing in the sum with $k\nu$ odd. Since this can be done without including any term $\mathbb{P}\left[b = 2k\right]$ more than once one obtains

$$\mathbb{P}\left[\nu'^{\gamma_{\nu'}} \nmid d\right] \leq \mathbb{P}\left[b \text{ is even}\right] , \qquad \nu' \in \mathrm{primes}(S) \text{ and } \nu' > 2 .$$

Assume now S odd. The probability of a factor ν^{γ_ν} not dividing d is again given by (10) and the middle term can be bounded by the same procedure of the S-even case. The remaining term $\mathbb{P}\left[b = S\right]$ can be bounded with

$\mathbb{P}[b = S-1]$ ($S-1$ is a multiple of 2 at this point) by using the fact that for $p < S/(S+1)$ it holds

$$\mathbb{P}[b = S-1] = \binom{S}{S-1} p^{S-1}(1-p) \geq \binom{S}{S} p^S = \mathbb{P}[b = S] \ ,$$

concluding the proof. □

Since the M variables $f_1, \ldots, f_M$ are i.i.d. and by recalling (6) we have the following straightforward corollary

Corollary 1. *If $p \in [0, S/(S+1)]$ then*

$$\mathbb{P}\left[\widehat{S}(\bar{\boldsymbol{f}}) \neq S \ ; \ S\right] \leq |primes(S)|\, \beta(p)^M \ , \tag{11}$$

where $\beta(p)$ is the probability that the outcome of a binomial random variable with susses probability p and a number of experiments S is even, i.e.

$$\beta(p) := \frac{1 + (1-2p)^S}{2} \ . \tag{12}$$

We remark that (11) is the exact error probability when S is a power of two. This means that if we consider as optimization strategy for p that of minimizing the worst case error probability over a certain range of possible sizes $[1, S_{max}]$ then the optimum is given by $p = 0.5$ corresponding to a global minimum of $\beta(p)$ for any S power of two.

We conclude this section by the graphical comparison in Figure 1 where we plot the bounds (5) and (11) as functions of M .

2.2 Max consensus and common uniform distribution

In this case the common distribution is uniform over $[0,1]$, i.e. $y_m^{(i)} \sim \mathcal{U}[0,1]$, and the consensus function is the component-wise maximum. Asymptotically the network reaches consensus on a point $\boldsymbol{f} \in [0,1]^M$ such that

$$f_m := \max_{i \in \mathcal{N}} y_m^{(i)} \quad , \quad m = 1, \ldots, M \ . \tag{13}$$

The distribution of each outcome f_m is given by the S-th order statistic

$$\mathbb{P}[f_m \leq a] = \prod_{i \in \mathcal{N}} \mathbb{P}\left[y_m^{(i)} \leq a\right] = \begin{cases} a^S & 0 \leq a \leq 1 \\ 0 & \text{otherwise.} \end{cases}$$

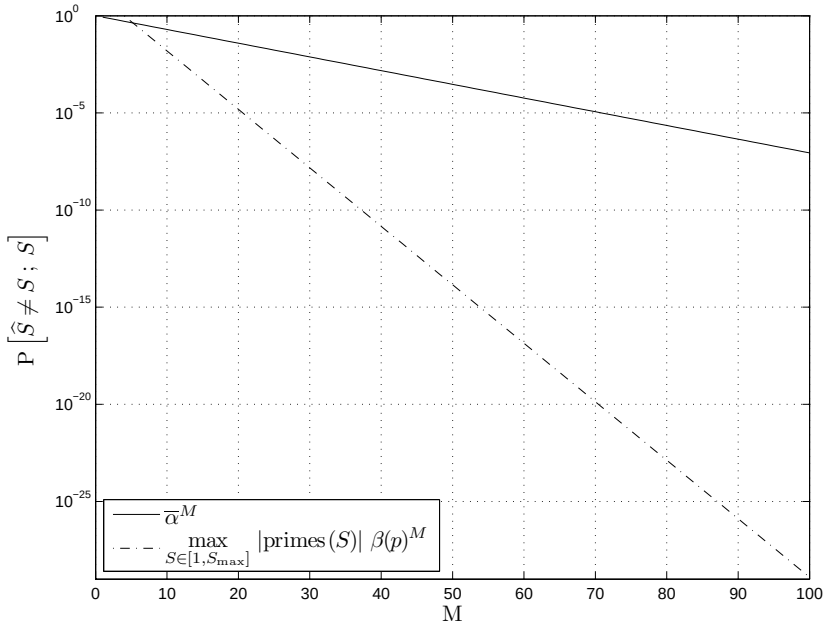


Figure 1: Comparison of the two error probability bounds given in (5) and (11). We have set $\overline{\alpha} = 0.85$ in (5) while for the bound in (11) we have considered the range of sizes $[1, 10^6]$ for which we have found numerically $\max_{S\in[1,10^6]} |\mathrm{primes}(S)| = 16$ and set $p = 0.5$.

By exploiting independency, this provides in turn the joint probability density of $\boldsymbol{f}$

$$p\left(\boldsymbol{f}\ ;\ S\right) = \begin{cases} S\displaystyle\prod_{m=1}^{M} f_m^{S-1} & 0 \leq f_m \leq 1\ ,\ m = 1, \ldots, M \\ 0 & \text{otherwise} \end{cases} \tag{14}$$

Given $\boldsymbol{f}$ and by interpreting the (unimodal) density (14) as a likelihood, one finds

$$\widehat{S}(\boldsymbol{f}) = \arg\max_{S \in \mathbb{R}^+} p\left(\boldsymbol{f}\ ;\ S\right) = \frac{M}{\sum_{m=1}^{M} -\log\left(f_m\right)}\ . \tag{15}$$

It can be seen that except from a scaling factor, the r.v. $\widehat{S}(\boldsymbol{f})$ is distributed as an inverse gamma variable and for $M > 2$ its mean and variance can be written in explicit form

$$\mathbb{E}\left[\widehat{S}; S, M\right] = S\frac{M}{M-1}\ , \qquad \text{var}\left[\widehat{S}\ ;\ S, M\right] = S^2\frac{M^2}{(M-1)^2(M-2)}.$$

Using these relations one can show that the estimator relative mean square error depends only on M and is strictly decreasing (in $\mathbb{N}^+$)

$$\mathbb{E}\left[\left(\frac{S-\widehat{S}}{S}\right)^2\ ;\ S(t), M\right] = 2\frac{M-1}{M^2-3M+2} \quad , \quad M > 2\ .$$

In Section 5 we will verify on a qualitative level that this property implies that, once M is fixed, the estimator in (15) exhibits approximately the same relative accuracy with varying S.

As a last remark, we notice that this is the same ML estimator (with the same statistical properties) of the one presented in [5] where the consensus function is the component-wise minimum and the common probability distribution is exponential.

3 Estimating the size of neighborhoods

Being able to estimate the size of a network only as a whole is limiting in practice. It is often the case for networks to consist of multiple clusters, i.e. smaller, more tightly connected sub-networks, in which information on the size of a node's neighborhood can be as important as information on the size of the whole network.

In the following we briefly review a more general estimation procedure which exploits the same basic ideas behind the max-consensus estimator in Section 2.2 extending it to the case of networks with dynamic topology.

This background material is preparatory to the hypothesis testing approach for detecting topology changes that we present in Section 4.

3.1 The max-consensus case

We start by introducing an assumption on the communication protocol used by nodes in the network. This jointly provides a natural way to partition time that allows us to deal with the effects of size changes in an aggregate fashion.

Protocol 1. *The time is divided in epochs, indexed by $t = 0, 1, 2, \ldots$. Every agent broadcasts its information exactly once per epoch. The order of the broadcasting operations is irrelevant, and can change in time. When an agent broadcasts its information, it broadcasts the information that it had at the beginning of the epoch.*

In the following we will use the terms 'epoch' and 'time' interchangeably, we remark however, that t is just the epoch index and does not represent a physical quantity (e.g. seconds).

Given protocol 1 it is meaningful in this context to consider the following network model: at each epoch t the we describe the network by means of a graph, $\mathcal{G}(t) := (\mathcal{V}(t), \mathcal{E}(t))$, where $\mathcal{V}(t)$ is the set of agents active at time t and $\mathcal{E}(t) \subseteq \mathcal{V}(t) \times \mathcal{V}(t)$ is the set of communications among active agents at time t: $(i, j) \in \mathcal{E}(t)$ indicates that agent i has successfully broadcast its information to j.

Assume now that information is generated as in Alg. 1; intuitively, this scheme is a parallelization of the procedure described in Section 2.2. Now, however, each agent i is endowed with matrices $F^{(i)}(t) \in \mathbb{R}^{M \times D}$, where M and D are fixed a-priori and t is the epoch index. For notational brevity, we will let $\boldsymbol{f}_k^{(i)}(t)$ denote the k-th column of $F^{(i)}(t)$, and $f_{m,k}^{(i)}(t)$ the element in the m-th row and k-th column of $F^{(i)}(t)$ ($k = 1, \ldots, D$, $m = 1, \ldots, M$).

Algorithm 1 Information generation scheme

1: **for** $t = 1, 2, \ldots$ **do**

2: *(Information Update)* each agent i computes $F^{(i)}(t)$ by shifting the columns of $F^{(i)}(t-1)$, in the sense that $\boldsymbol{f}_k^{(i)}(t) = \boldsymbol{f}_{k-1}^{(i)}(t-1)$ for $k = 2, \ldots, D$. $\boldsymbol{f}_1^{(i)}(t)$ is instead filled with M new i.i.d. random values $f_{m,1}^{(i)}(t) \sim \mathcal{U}[0,1]$, $m = 1, \ldots, M$;

3: *(Communication)* every agent broadcasts $F^{(i)}(t)$ to its neighbors following the communication protocol 1;

4: *(Information Mixing)* each agent i updates its $F^{(i)}(t)$ by means of the $F^{(j)}(t)$'s received from its neighbors. More specifically

$$f_{m,k}^{(i)}(t) \leftarrow \max_{(j,i)\in\mathcal{E}(t)} \left(f_{m,k}^{(i)}(t),\ \left\{ f_{m,k}^{(j)}(t) \right\} \right) \tag{16}$$

for $m = 1, \ldots, M$, $k = 1, \ldots, D$.

5: **end for**

With this it is possible to precisely define which agents contributed, from a statistical point of view, to the generation of the k-th column of $F^{(i)}(t)$:

Definition 1. *Given the communication protocol 1 and Alg. 1, the set of k-steps neighbors of agent i at time t is defined for $k = 0$ as $\mathcal{V}_0^{(i)}(t) = \{i\}$ and, for $k \geq 1$, through the recursion*

$$\mathcal{V}_k^{(i)}(t) := \bigcup_{(j,i)\in\mathcal{E}(t)} \mathcal{V}_{k-1}^{(j)}(t-1) \ .$$

Let $S_k^{(i)}(t)$ denote the size of $\mathcal{V}_k^{(i)}(t)$, i.e. $S_k^{(i)}(t) := \left|\mathcal{V}_k^{(i)}(t)\right|$. Since $\boldsymbol{f}_k^{(i)}(t)$ aggregates information just from the agents in $\mathcal{V}_k^{(i)}(t)$, it follows from Section 2.2 that the statistic

$$\widehat{S_k^{(i)}}(t) := \frac{M}{\sum_{m=1}^{M} -\log\left(f_{m,k}^{(i)}(t)\right)} \ , \tag{17}$$

is the ML estimator of $S_k^{(i)}(t)$ given $\boldsymbol{f}_k^{(i)}(t)$. Thus, at each time t, any node $i \in \mathcal{N}(t)$ possesses locally D size estimates, one for each k-steps neighborhood, $k = 1, \ldots, D$.

4 Change detection

Intuitively, identification of changes in the size of the generic k-steps neighborhood of the generic agent i can be performed by inspecting the temporal evolution of $F^{(i)}(t)$.

In this chapter we exploit the estimation scheme of Section 3 in order to formulate and characterize an hypothesis test in the form of a GLR that nodes can run locally at each epoch to test topology changes in their k-steps neighborhoods.

Our working context is framed by the assumption that the process trough which the network changes its topology and size in time is either non-random or random with unknown distribution. Moreover, as customary in the change detection literature (see e.g. [6]), we assume at most one sensible change in the true size over the time window the test is performed on.

We brifly recall some definitions relevant to hypothesis testing in Section 4.1, then, in Section 4.2 we outline the change detection algorithm and in Section 4.3 we provide its statistical characterization.

4.1 Background material on hypotheses testing

Consider a r.v. $\boldsymbol{f}$ with absolutely continuous probability density in a parametrized family $\{p_\theta\}_{\theta\in\Theta}$, where θ is in general a vector and Θ is the set of all a-priori possible values taken by the parameter. Without loss of generality, let $\mathcal{H}_0$ and $\mathcal{H}_1$ be two complementary hypotheses on θ

$$\mathcal{H}_i = \{\boldsymbol{f} \sim p_\theta \text{ with } \theta \in \Theta_i\}, \quad i \in \{0,1\}, \tag{18}$$

To the problem of deciding between $\mathcal{H}_0$ and $\mathcal{H}_1$ is naturally associated, by the Neymann-Person Lemma (see e.g. [12, 11, 6]), a decision function $g(\boldsymbol{f})$ with range $\{0,1\}$. g partitions the space of outcomes for $\boldsymbol{f}$ in two regions: the *acceptance region* $R := \{\boldsymbol{f} \text{ s.t. } g(\boldsymbol{f}) = 0\}$ and the *critical region* R^c. When testing between $\mathcal{H}_0$ and $\mathcal{H}_1$ there are two kinds of possible error: selecting $\mathcal{H}_1$ when $\mathcal{H}_0$ is true, this is said to be an *error of type I* or accepting $\mathcal{H}_0$ when $\mathcal{H}_1$ is true, i.e. an *error of type II*. For each test, with associated critical function g, one can define the test *power function*

$$\beta_g(\theta) := \mathbb{E}\left[g\ ;\ \theta\right]\ . \tag{19}$$

$\beta_g(\theta)$ characterizes the statistical performance of g in terms of the probability of deciding $\mathcal{H}_1$ for a fixed $\theta \in \Theta$. In particular, the worst case type I error probability, a.k.a. the *size* of the test, is defined as a function of β_g by means of

$$\alpha_0(g) := \sup_{\theta\in\Theta_0} \beta_g(\theta)\ . \tag{20}$$

Moreover, when $\theta \in \Theta_1$, $\beta_g(\theta)$ is the probability of *not* committing errors of type II for that particular θ.

A decision function g is said to be Uniformly Most Powerful (UMP) in the class $K_\varepsilon := \{\widetilde{g} \text{ s.t. } \alpha_0(\widetilde{g}) = \varepsilon\}$ (namely, the set of tests of given size ε) if $\beta_g(\theta) \geq \beta_{\widetilde{g}}(\theta)$ for all $\widetilde{g} \in K_\varepsilon$ and for each $\theta \in \Theta_1$.

4.2 The GLR approach

The following computations involve only local quantities. We assume that the agent index i and the neighborhood k have been already chosen by the user. Thus we drop all the superscripts (i) and subscripts k for notational brevity. We shall then indicate $\boldsymbol{f}_k^{(i)}(t)$ with $\boldsymbol{f}(t)$, $\chi_k^{(i)}(t)$ with $\chi(t)$, $\mathcal{V}_k^{(i)}(t)$ with $\mathcal{V}(t)$ and $S_k^{(i)}(t)$ with $S(t)$.

Consider the time window $t-N, \ldots, t$, of length $N+1$ with N given, and the following set of hypotheses

$$\begin{cases} \mathcal{H}_0: & S(t-N) = \ldots = S(t-T) = \overline{S}, \\ & S(i) \geq \sigma \overline{S} \text{ for all } i \in \{t-T+1, \ldots, t\} \\ \mathcal{H}_1: & S(t-N) = \ldots = S(t-T) = \overline{S}, \\ & \text{exists } i \in \{t-T+1, \ldots, t\} \text{ s.t. } S(i) < \sigma \overline{S} \end{cases} \tag{21}$$

parametrized both in $T \in \{1, \ldots, N\}$ and in $\sigma, \overline{S} \in \mathbb{R}^+$. (21) involves thus 3 unknown parameters: the pre-change value $\overline{S}$, the unknown time of change T, and the post-change value.

Deciding between the composite hypothesis $\mathcal{H}_0$ and $\mathcal{H}_1$ can be interpreted as testing a trend of decrease in size, with the size amplitude described by parameter $\sigma \in (0,1]$[2].

We implicitly assume that no prior information is available on the various estimands and thus estimate all the unknown quantities from the data through ML approaches, see [6, Chap. 2.6.2].

Assume that σ and N have already been chosen (these are both design parameters whose role is discussed in Section 5). Testing between $\mathcal{H}_0$ and $\mathcal{H}_1$ in (21) can thus be performed running Alg. 2 for each t in Alg. 1.

The algorithm performs as follows: it starts computing, for all the plausible change times, the ML estimates of the pre-change and of the post change values under no constraints and under hypothesis $\mathcal{H}_0$ (equations (22), (23) and (24) respectively). Then it computes all the possible GLRs in (25), from which it is possible to estimate the most likely change time in (26) and thus decide between the hypotheses in (27).

[2]The specular set of hypotheses, in which the inequalities are reversed, leads a test for testing against a trend of size increase.

Algorithm 2 Neighborhood change detection

(cycle on all the plausible change times)

for $\mathcal{T} = 1, \ldots, N-1$ **do**

(estimation of the pre-change value)

$$\overline{S}(\mathcal{T}) = \frac{M(N-\mathcal{T}+1)}{\sum_{\tau=t-N}^{t-\mathcal{T}} \sum_{m=1}^{M} -\log\left(f_m(t)\right)} \tag{22}$$

(estimation of the post-change values under no hypotheses and under $\mathcal{H}_0$)

for $\tau \in \{t-\mathcal{T}+1, \ldots, t\}$ **do**

$$\widehat{S}(\tau) = \frac{M}{\sum_{m=1}^{M} -\log\left(f_m(t)\right)} \tag{23}$$

$$\widehat{S}_0(\tau) = \begin{cases} \widehat{S}(\tau) \text{ if } \widehat{S}(\tau) \geq \sigma\overline{S}(\mathcal{T}) \\ \sigma\overline{S}(\mathcal{T}) \text{ otherwise} \end{cases} \tag{24}$$

end for

(computation of the GLR)

$$\Lambda(\mathcal{T}) = \frac{\prod_{\tau=t-\mathcal{T}}^{t} \ell\left(\widehat{S}_0(\tau)\,;\, \boldsymbol{f}(\tau)\right)}{\prod_{\tau=t-\mathcal{T}}^{t} \ell\left(\widehat{S}(\tau)\,;\, \boldsymbol{f}(\tau)\right)} \tag{25}$$

end for

(computation of the optimal change time)

$$T = \arg \min_{\mathcal{T} \in \{1,\ldots,N-1\}} \Lambda(\mathcal{T}) \tag{26}$$

(decision between $\mathcal{H}_0$ and $\mathcal{H}_1$)

$$g\left(\boldsymbol{f}(t-N:t)\right) = \begin{cases} 0 & \text{if } \Lambda(T) \geq \lambda \\ 1 & \text{otherwise.} \end{cases} \tag{27}$$

4.3 Statistical analysis of Algorithm 2

We now characterize the decision function (27) in terms of its power, provide some optimality results and discuss the role of the tunable parameter M in

connection with the test performance. Throughout this section we assume that T and $\sigma\overline{S}$ are fixed, i.e. they have been computed according to Alg. 2.

The following proposition and corollary establish a partial order in the parameters space:

Proposition 2. *Consider g of the form (27) as a function of the T-dimensional vector $[S(t-T+1),\dots,S(t)]$, with each scalar component in $\mathbb{R}^+$. Then for any $\bar{t}\in\{t-T+1,\dots,t\}$ and fixed $S(t-T+1),\dots,S(\tau-1)$, $S(\tau+1),\dots,S(t)$ the power of*

$$g\left(S(t-T+1),\dots,S(\tau-1),S(\bar{t}),S(\tau+1),\dots,S(t)\right)$$

is strictly monotone decreasing with $S(\bar{t})$.

Proof. The probability of rejecting $\mathcal{H}_0$ as a function of the parameters $S(t-T+1),\dots,S(t)$ is by definition

$$\mathbb{P}\left[\sum_{\tau=t-T+1}^{t}\log\frac{p_0\left(\boldsymbol{f}(\tau)\;;\;S(\tau)\right)}{p\left(\boldsymbol{f}(\tau)\;;\;S(\tau)\right)}<\log\lambda\right]\;.$$

We will show that the thesis holds in the particular case $\bar{t}=t$. By the problem's symmetry, this will prove that the proposition remains true for any $\bar{t}\in\{t-T+1,\dots,t\}$.

Define

$$\boldsymbol{\lambda}(\tau)=\log\frac{p_0\left(\boldsymbol{f}(\tau)\;;\;S(\tau)\right)}{p\left(\boldsymbol{f}(\tau)\;;\;S(\tau)\right)}\;,\quad\tau=t-T+1,\dots,t \tag{28}$$

and

$$\overline{\boldsymbol{\lambda}}=\log\lambda-\sum_{\tau=t-T+1}^{t-1}\boldsymbol{\lambda}(\tau)\;.$$

By the law of total probability the test power function can be computed by evaluating the multi-dimensional integral

$$\int\cdots\int\mathbb{P}\left[\boldsymbol{\lambda}(t)<\overline{\boldsymbol{\lambda}}\,\middle|\overline{\boldsymbol{\lambda}}\;;\;S(t)\right]\prod_{\tau=t-T+1}^{t-1}p\left(\boldsymbol{f}(\tau);\;S(\tau)\right)d\boldsymbol{f}(\tau)\;.$$

We note now that only the first factor of the integrand depends on $S(t)$ and that the same factor can be rewritten as the linear convex combination

$$\begin{aligned}\mathbb{P}\left[\boldsymbol{\lambda}(t)<\overline{\boldsymbol{\lambda}}\,\middle|\overline{\boldsymbol{\lambda}}\;;\;S(t)\right]=&\\ &\mathbb{P}\left[\boldsymbol{\lambda}(t)<\overline{\boldsymbol{\lambda}}\,\middle|\overline{\boldsymbol{\lambda}},\widehat{S}(t)<\sigma\overline{S}\;;\;S(t)\right]\mathbb{P}\left[\widehat{S}(t)<\sigma\overline{S}\;;\;S(t)\right]\\ &+\mathbb{P}\left[0<\overline{\boldsymbol{\lambda}}\;\middle|\;\overline{\boldsymbol{\lambda}}\right]\mathbb{P}\left[\widehat{S}(t)\geq\sigma\overline{S}\;;\;S(t)\right]\;,\end{aligned}$$

where $\mathbb{P}\left[0 < \overline{\boldsymbol{\lambda}} \;\middle|\; \overline{\boldsymbol{\lambda}}\right] \in \{0,1\}$ does not depend on $S(t)$. It suffices then to show that

$$\mathbb{P}\left[\boldsymbol{\lambda}(t) < \overline{\boldsymbol{\lambda}} \middle| \overline{\boldsymbol{\lambda}}, \widehat{S}(t) < \sigma\overline{S} \;;\; S(t)\right] \mathbb{P}\left[\widehat{S}(t) < \sigma\overline{S} \;;\; S(t)\right] = \mathbb{P}\left[\boldsymbol{\lambda}(t) < \overline{\boldsymbol{\lambda}}, \widehat{S}(t) < \sigma\overline{S} \,\middle|\, \overline{\boldsymbol{\lambda}} \;;\; S(t)\right] ,$$

is a strictly decreasing function of $S(t)$ (whenever it is not one). Indeed, by explicitly computing $\boldsymbol{\lambda}(t)$, one finds

$$\boldsymbol{\lambda}(t) = \left(\widehat{S}(t) - \widehat{S}_0(t)\right) x + M \log \frac{\widehat{S}_0(t)}{\widehat{S}(t)} ,$$

where

$$x := \sum_{m=1}^{M} - \log f_m(t)$$

has apriori distribution Gamma $(M, S^{-1}(t))$ (see [13]). Under the condition $\widehat{S}(t) < \sigma\overline{S}$, $\boldsymbol{\lambda}(t)$ is a strictly decreasing function of x, e.g. $\boldsymbol{\lambda}(t) = h(x)$ and such $h(\cdot)$ is easily seen to be a bijection. It follows immediately

$$\begin{aligned}
&\mathbb{P}\left[x > h^{-1}(\overline{\boldsymbol{\lambda}}) \;,\; \widehat{S}(t) < \sigma\overline{S} \,\middle|\, \overline{\boldsymbol{\lambda}} \;;\; S(t)\right] \\
&\quad = \mathbb{P}\left[x > h^{-1}(\overline{\boldsymbol{\lambda}}) \;,\; x > \frac{M}{\sigma\overline{S}} \,\middle|\, \overline{\boldsymbol{\lambda}} \;;\; S(t)\right] \\
&\quad = \begin{cases} \dfrac{\overline{\Gamma}(M, S(t)h^{-1}(\overline{\boldsymbol{\lambda}}))}{\Gamma(M)} & h^{-1}(\overline{\boldsymbol{\lambda}}) > \dfrac{M}{\sigma\overline{S}} \\ 0 & \text{otherwise} \end{cases}
\end{aligned}$$

Where $\overline{\Gamma}(\cdot,\cdot)$ and $\Gamma(\cdot)$ are the upper incomplete and the usual gamma functions respectively. $\overline{\Gamma}$ is strictly decreasing in its second argument and thus the power function is strictly decreasing as a function of $S(t)$. □

The following corollary follows immediately

Corollary 2. *If*

$$[S(t-T+1), \ldots, S(t)] \;,$$
$$[S'(t-T+1), \ldots, S'(t)] \;,$$

are s.t. $S(\tau) \leq S'(\tau)$ *for all* $\tau = t-T+1, \ldots, t$, *then*

$$\beta_g\left(S(t-T+1), \ldots, S(t)\right) \geq \beta_g\left(S'(t-T+1), \ldots, S'(t)\right) \;.$$

Moreover the inequality is strict if there exists at least one $\tau \in \{t-T+1, \ldots, t\}$ *such that* $S(\tau) < S'(\tau)$.

We are now in a position to prove the following:

Proposition 3. *The size of a most powerful test deciding between $\mathcal{H}_0$ and $\mathcal{H}_1$ can be computed by evaluating the test power function at the point on the boundary $S(\tau) = \sigma\bar{S},\ \tau = t - T + 1, \ldots, t$, i.e.:*

$$\alpha_0 = \beta_g \left(S(\tau) = \sigma\bar{S},\ \tau = t - T + 1, \ldots, t \right) . \tag{29}$$

Proof. This is trivial by recalling the definition of size of g in (20) and by exploiting Corollary 2 □

It turns out that the GLR test formulated in Sec. 4 exhibits two properties of remarkable practical importance. The first one being optimality of the decision rule (27) when testing against any simple alternative

Proposition 4. *For any given size α_0 the test of $\mathcal{H}_0$ against $\mathcal{H}_1$ induced by the decision function* (27) *is UMP.*

Proof. This is a direct consequence of the independence of α_0 from any simple alternative in $\mathcal{H}_1$. □

For the sake of clarity, let us now emphasize the dependence of the decision rule from the boundary constraint in (21) by writing $g := g(\sigma\overline{S})$. The following proposition states a sort of independency of the power function from $\sigma\overline{S}$.

Proposition 5. *Consider the two decision functions $g(\sigma\overline{S})$ and $g(\sigma'\overline{S}')$ corresponding to two different boundary constraints in the hypotheses* (21) *and define*

$$\beta := \beta_{g(\sigma\overline{S})} \left(S(t - T + 1), \ldots, S(t) \right) ,$$

and

$$\beta' := \beta_{g(\sigma'\overline{S}')} \left(S'(t - T + 1), \ldots, S'(t) \right) .$$

If for all $\tau = t - T + 1, \ldots, t$ there exists reals $\rho(\tau)$ such that

$$\rho(\tau) = \frac{S(\tau)}{\sigma\overline{S}} = \frac{S'(\tau)}{\sigma'\overline{S}'} , \tag{30}$$

then $\beta = \beta'$.

Proof. For the sake of a simpler notation, in the following we omit the parameters when writing probabilities and densities. Let us define $\mathcal{T} = \{t-T+1, \dots, t\}$ and denote by $\mathcal{P}$ the set of all possible subset of $\mathcal{T}$. Futhermore for each $\tau \in \mathcal{T}$ let

$$P_1(\tau) := \mathbb{P}\left[\widehat{S}(\tau) < \sigma\overline{S}\right] \ , \ P_0(\tau) := \mathbb{P}\left[\widehat{S}(\tau) \geq \sigma\overline{S}\right] \ .$$

By the law of total probability and Bayes rule, the type I error probability can be computed by means of

$$\sum_{p\in\mathcal{P}} \mathbb{P}\left[\sum_{\tau\in p} \boldsymbol{\lambda}(\tau) < \lambda \ , \ \widehat{S}(k) < \sigma\overline{S} \text{ if } k \in p\right] \prod_{\tau\in\mathcal{T}\setminus p} P_0(\tau) \ ,$$

where $\boldsymbol{\lambda}(\tau)$ has been defined in (28). By explicit computation of the first factor in each term of the sum one finds

$$\mathbb{P}\left[\sum_{\tau\in p} M\left(1+\log\frac{\sigma\overline{S}x(k)}{M}\right) - \sigma\overline{S}x(\tau) < \log\lambda \ , \ \frac{M}{x(k)} < \sigma\overline{S} \text{ if } k \in p\right] \ , \tag{31}$$

where $x(\tau)$ is again the r.v.

$$\sum_{m=1}^{M} -\log f_m(\tau) \ ,$$

with distribution Gamma $(M, (S(\tau))^{-1})$. By introducing the new set of r.v.s. $\bar{x}(\tau) = \sigma\overline{S}x(\tau)$ and substituting in (31) one finds

$$\mathbb{P}\left[\sum_{\tau\in p} M\left(1+\log\frac{\bar{x}(\tau)}{M}\right) - \bar{x}(\tau) < \log\lambda \ , \ \bar{x}(\tau) > M \text{ if } k \in p\right]$$

and since by the scaling property of gamma distributions, each $\bar{x}(\tau)$ is distributed as Gamma $\left(M, \sigma\overline{S}/S(\tau)\right)$. This probability depends only on the ratios in (30). Following along these lines, the same is easily proved also for the terms $\mathbb{P}\left[\widehat{S}(\tau) \geq \sigma\overline{S}\right]$. □

The following result follows immediately

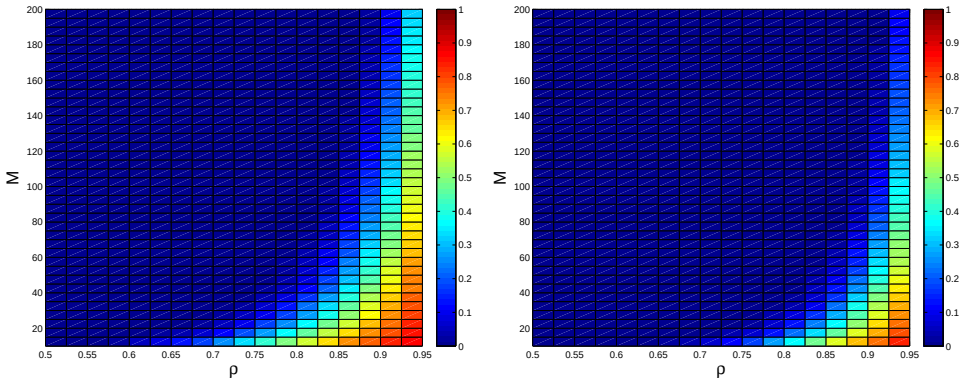
Proposition 6. (computing the test threshold λ) *Consider a most-powerful test g with size α_0. The test threshold λ does not depend on the value of $\sigma\overline{S}$.*

Proof. By proposition 2, when considering the worst case for (20), we can assume $S(\tau) = \sigma\overline{S}$ for all $\tau = t - T + 1, \ldots, t$. Futhermore, by the previous proposition, the power will depend only on the ratios $\rho(\tau) := S(\tau)/(\sigma\overline{S})$, $\tau = t - T + 1, \ldots, 1$. The proof is concluded by noting that for any outcome of $\overline{S}$ and fixed σ these ratios are all equal to one. □

As a consequence, λ depends only on M, α_0 and T, where M and α_0 are parameters fixed a-priori and the optimal change time T is estimated at each epoch t by means of (26). Thus, a string of thresholds, one for each $T \in \{1, \ldots, N\}$, can then be computed offline, e.g., by taking $\sigma\overline{S} = 1$ and $S(\tau) = 1$, $\tau = t - T + 1, \ldots, t$. At run-time, each node needs only to test the GLR outcome against the saved threshold, corresponding to the estimated value of T.

In Figures 2 and 3 we plot the test power as a function of T, M and $\rho := S(\tau)/\sigma\overline{S}$, $\tau = t - T + 1, \ldots, t$ for $\alpha_0 = 0.05$ and $\alpha_0 = 0.01$ respectively. To produce these plots the distribution of the GLR has been evaluated numerically using the Monte Carlo method (see the scripts in the folder `/scripts/matlab` and in particular `testpower.m`). As we expect from the consistency property of ML estimators the the type II error decreases with increasing M (when the values of the remaining parameters are fixed). Then M realizes a thread-off between a test with higher power and the usage of (possibly shared or limited) communication resources, eg. shared buses or limited data throughput.

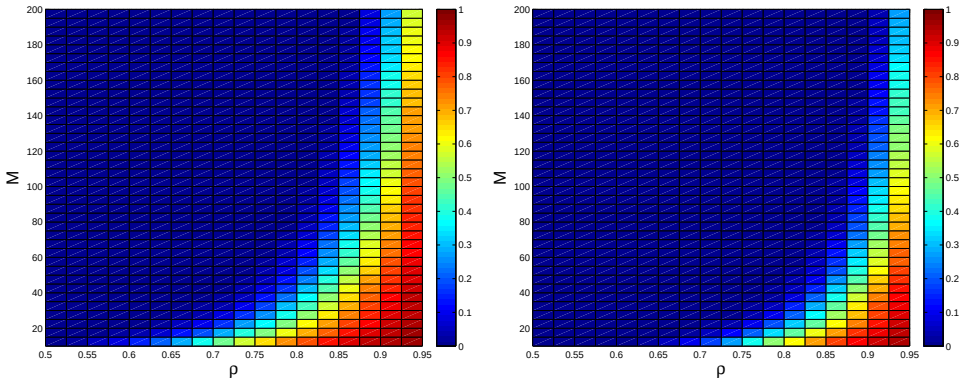
We conclude this section by noticing that, as one migth intuitively expect, the higher values assumes T the more probable is to decide for $\mathcal{H}_1$ when $\mathcal{H}_1$ is true.



(a) $T = 5$ (b) $T = 10$

(c) $T = 15$ (d) $T = 20$

Figure 2: Type II error probability as a function of M, T and $\rho := S(\tau)/\sigma\overline{S}$, $\tau = t - T + 1, \ldots, t$; the test size is $\alpha_0 = 0.05$.



(a) $T = 5$ (b) $T = 10$

(c) $T = 15$ (d) $T = 20$

Figure 3: Type II error probability as a function of M, T and $\rho := S(\tau)/\sigma\overline{S}$, $\tau = t - T + 1, \ldots, t$; the test size is $\alpha_0 = 0.01$.

5 Simulations

Here we comment some simulations of the change detection algorithm proposed in Section 4. In particular, we analyze the effects of changing the tunable parameters and discuss the algorithm's main shortcomings.

For the purpose of this section a custom simulation library, `python-netsim`, has been written in the python language [3]. The library allows the simulation of our GLR approach on arbitrary networks even with time-varying topology and is built on industry standard packages for numerical computation and analysis of complex networks [2, 1]. The library can be found under the folder `/scripts`.

In the following we will concentrate on a prototypical network with a grid topology of side 10. Each node in the network is identified by a tuple (x, y), $x, y \in \mathbb{N}$. Due to the naming convention used by the network-analysis library we employ, the lower-left node is given the id $(0, 0)$ and for a 10-sided grid the upper-right node is identified by the id $(9, 9)$. We have developed two different ways of visualizing information on the evolution of the change detection algorithm. In the first one we concentrate on a single node and plot the evolution of some relevant quantities over a window of $N+1$ epochs. An example frame from one of the attached videos is in Figure 4:

- in the upper border we report the values of the parameters chosen for the given simulation. The value of k, defining which k-steps neighborhood the quantities refer to, and the node id are reported here too.

- the trajectory of the selected k-steps neighborhood size estimate is plotted with a solid line. This line is black unless the tracked node has not yet acquired at least a window of $N+1$ observations, in which case is drawn in yellow (this can happen, e.g., right after the simulation start).

- the exact size of the k-steps neighborhood (recall Definition 1) is plotted as a dashed black line.

- the boundary constraints defining the hypotheses $\mathcal{H}_0$ and $\mathcal{H}_1$ in (21) are plotted as a dashed red line. Cyan and gray shaded areas enclose the points used in the computation of $\overline{S}$, i.e. the points in the range $[t-N, t-T]$ where T is the optimal change time in (26). On the right of the gray shaded area are the points over which the change-detection test is performed at the given epoch. Therefore the dashed red line assumes the value $\overline{S}$ in the cyan and gray areas and the value of $\sigma\overline{S}$ on their right. The exact meaning of the cyan area is explained in Section 5.1.

- whenever a change is detected at time t, this epoch index is recorded and at every successive frame a red vertical panel is overlaid around the list of past and current alarm times.

- the text at the lower left corner of each frame reports the number of active nodes at the current epoch over the total number of nodes in the network (active and inactive) and aggregate information on the alarm status of the tracked node as a string of the form

$$\text{alarms } x^{(i)}(t) \text{ in } y^{(i)}(t) \leftrightarrow z^{(i)}(t) \ .$$

  Here $x^{(i)}(t) \ \in \{0, 1, \ldots, D\}$ is the number of alarms that the tracked node i fired at the current iteration. Let us define for each node $i \in \mathcal{N}(t)$

$$\mathcal{K}^{(i)}(t) := \Big\{ k \ : \ \begin{array}{l} i \text{ detect a change in its } k\text{-steps} \\ \qquad \text{neighborhood at time } t, \ k = 1, \ldots, D \end{array} \Big\} \ .$$

  Then $x^{(i)}(t) := \left|\mathcal{K}^{(i)}(t)\right|$. $y^{(i)}(t)$ and $z^{(i)}(t)$ are the minimum and the maximum $k \in \{1, \ldots, D\}$ for which the tracked node fired an alarm at time t, i.e. $y^{(i)}(t) := \min\left(\mathcal{K}^{(i)}(t)\right)$ and $z^{(i)}(t) := \max\left(\mathcal{K}^{(i)}(t)\right)$. In Figure 4 eight alarms were fired at the current epoch and the changes were detected for eight different values of k between 1 and 11 (in particular the 1-step and 11-steps neighborhood change detectors fired an alarm).

In the following we will also use a second, more global, way to visualize information on the workings of the change-detection algorithm. An example frame of this view is in Figure 5

- nodes are drawn as circles with different colors according to their status. Gray nodes are inactive nodes which do not take part in consensus during the current epoch (although they are draw as part of the network topology this nodes are not contained in $\mathcal{N}(t)$); all other nodes are active. Nodes that fired alarms in the current epoch, i.e. those nodes $i \in \mathcal{N}(t)$ for which $\mathcal{K}^{(i)}(t) \neq \emptyset$, are drawn with different shades of red. The shade of red used is bright red when $\min\left(\mathcal{K}^{(i)}(t)\right) = 1$ and dark red when $\min\left(\mathcal{K}^{(i)}(t)\right) = D$. Active nodes that did not detect any change in their k-steps neighborhoods for any $k = 1, \ldots, D$ are drawn white. Yellow nodes instead are active nodes which haven't yet acquired a window of $N + 1$ observations: they take part in consensus but cannot yet run the change-detection algorithm.

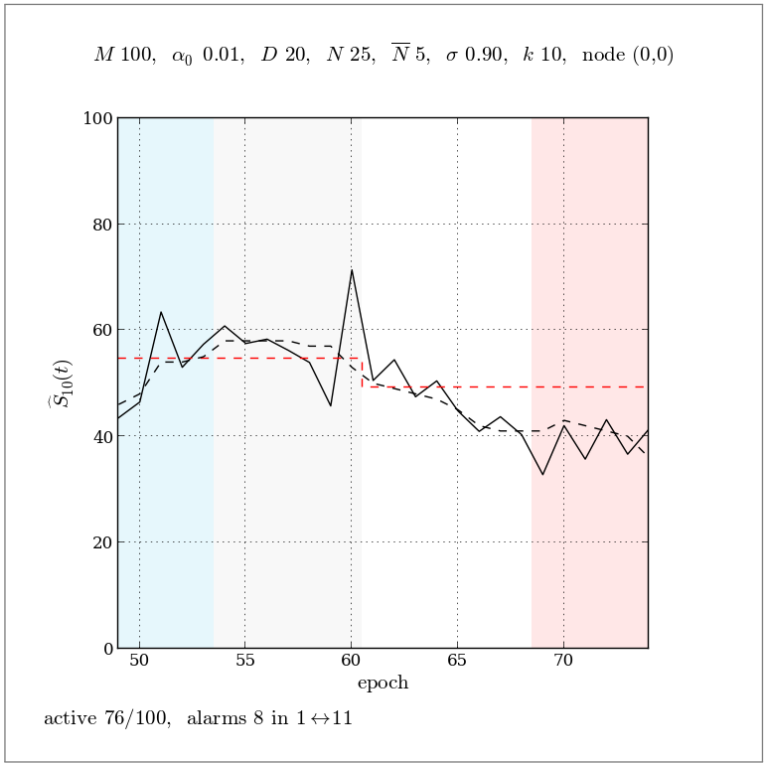


Figure 4: Example trajectories for the change detection algorithm over a window spanning $N + 1$ epochs. See the text for an explanation of the conventions used.

- in the lower left corner we report the current epoch index, the number of active nodes over the total number of nodes (active and inactive) in the network and a string carrying aggregate information on the alarms that fired in the current epoch of the form

$$\text{alarms } a(t)/b(t),\ \lfloor k \rfloor\, c(t) \leftrightarrow d(t)\ \ \lceil k \rceil\, e(t) \leftrightarrow f(t)\ .$$

  $a(t)$ is the number of (active) nodes that fired at least one alarm in the current epoch while $b(t)$ is the total number of alarms that fired in the network at time t

$$b(t) := \sum_{i \in \mathcal{N}(t)} \left| \mathcal{K}^{(i)}(t) \right| \ . \tag{32}$$

  The remaining statistics provide further information on what change-detectors are firing in the current epoch and, in a sense, how they are distributed:

$$c(t) := \min_{i \in \mathcal{N}(t)} \left\{ \mathcal{K}^{(i)}(t) \right\}, \quad d(t) := \max_{i \in \mathcal{N}(t)} \left(\min \mathcal{K}^{(i)}(t) \right) \ ,$$

$$e(t) := \min_{i \in \mathcal{N}(t)} \left(\max \mathcal{K}^{(i)}(t) \right), \quad f(t) := \max_{i \in \mathcal{N}(t)} \left\{ \max \mathcal{K}^{(i)}(t) \right\} \ .$$

It is simpler to start by analyzing first the case of a static network; we do so in Section 5.1. In Section 5.2 instead we consider network with dynamic topology; this is the case motivating the introduction of the parameter σ in the definition of the hypotheses $\mathcal{H}_0$ and $\mathcal{H}_1$.

5.1 Static network

As we shall see the pre-change size estimate, $\overline{S}$, has a dramatic effect on the test performance. Let us consider the following choice of parameters

$$M = 50,\ D = 20,\ N = 25,\ \alpha_0 = 0.01,\ \sigma = 0.01\ .$$

The attached video `00-static-net.avi` pictures the evolution of the change detection algorithm applied to the 20-steps neighborhood, for the node with id $(0, 0)$ on the lower-left corner of a grid network with static topology. The pre-change size estimates tend to be very noise and biased towards over estimating the actual value, which in this simulation is constant and equal to 100 (the number of nodes in the network) after an initial transitory. In particular out-lier outcomes of $\widehat{S}_k(t)$ at times near the lower part of the range $[t-N, \ldots, t]$ tend to trick the change time estimator into selecting high values for T leading to over-estimating the true pre-change size; see for example the extreme case at time $t = 130$ in the video.

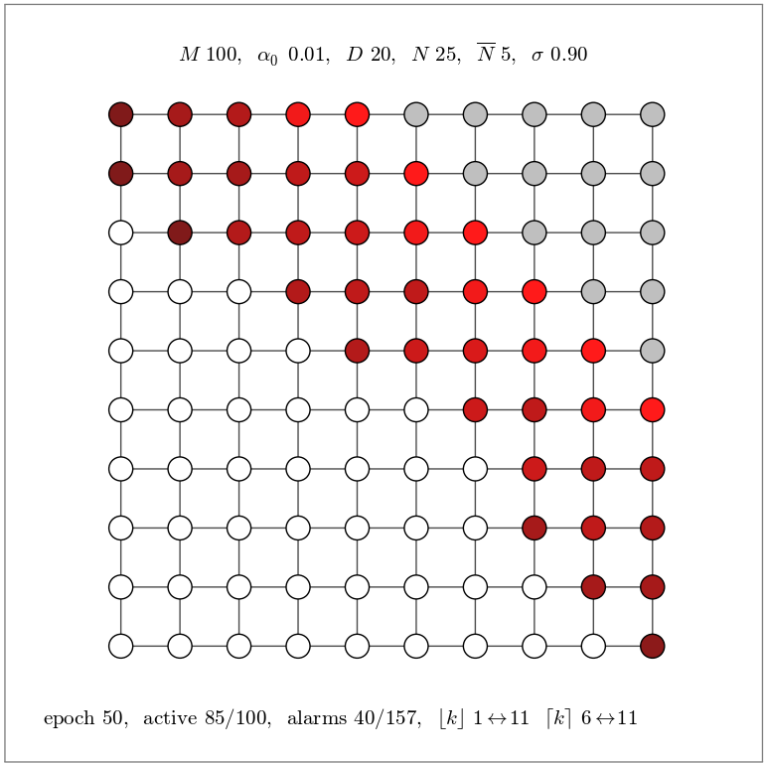


Figure 5: Global view example of the compressed status of each node in the network at a given epoch index. See the text for an explanation of the conventions used.

A partial solution to this problem is to reserve a number $\overline{N}$ of past observations at the lower end of the observation window (i.e. for times near $t-N$) for the sole computation of $\overline{S}$. This can be done by restricting the set of possible values of T over which the search for the optimal change time is performed in Algorithm 2 to $\{1, N-\overline{N}\}$. In `01-static-net.avi` we picture a realization of the trajectories for the same parameters of the previous case but with $\overline{N}$ set to 5. Now the number of false alarms is greatly reduced although the rate at which they are fired is still sensibly higher than $\alpha_0 = 0.01$. It is evident how, as in the previous case, (false) alarms are fired only when the estimate of $\overline{S}$ is inaccurate, i.e. whenever $\overline{S}$ is sensibly higher than the number of active nodes.

We have found that choosing $\overline{N} \approx 5$ tends to provide good results without introducing to much lag due to its filtering effect (even when the network has time-varying topology). We notice that the behavior of the change detection test remains qualitatively the same even when increasing the number of nodes: see for example the realization in `02-net-static.avi` in which we consider a grid network with a side increased to 20 nodes.

5.2 Dynamic network with unreliable nodes

Here we simulate unreliable networks, which we model by allowing each node to be in any of two states: active or inactive. Only active nodes take part in consensus and only nodes which are active from at least $N+1$ epochs also run the change-detection algorithm.

We first analyze, in Section 5.2.1, the case in which transitions from the active to the inactive state and vice versa are regulated by a birth-death like process. Then in Section 5.2.2 we comment briefly on the problem of distributedly detecting the 'direction' in which a disconnection happens.

5.2.1 Random state transitions

In this case the transition between the active and the inactive state is regulated by the following simple model: at each epoch each active node has a probability $P_d = 0.01$ of becoming inactive while each currently inactive node has a probability $P_b = 0.04$ of transitioning back to the active state.

In `03-dynamic-net.avi` and `04-dynamic-net.avi` we picture two realization of such process for $k = 20$ and $k = 10$ respectively. The added noise due to the time-varying nature of the network topology deteriorates sensibly the estimation of the pre-change size canceling out the improvement achieved by increasing $\overline{N}$; this effect is even more extreme in the noisier $k = 10$ case.

This is again an effect of the over-estimation of $\overline{S}$ which in turn leads to a set of hypotheses defining a 'ill-posed' change-detection test.

This is test case motivated the introduction of the parameter σ in our formulation of the hypotheses in (21). The rationale is that one might not always be interested in all kind of changes, but just in those disconnections involving a sensible amount of nodes. In practice using $\sigma < 1$ has also the side effect of greatly improving the rate of false alarms (even for static networks).

In `05-dyn-net.avi` and `06-dyn-net.avi` we picture two realizations for $k = 20$ and $k = 10$ respectively and the sensibility threshold σ set to 0.9. Now the test is able to both contrast the bias in the pre-change size estimate and still effectively detect size changes which have sensible effect on the k-steps neighborhoods.

Smaller values of σ do improve further the false alarm rate reducing it. However, the smaller σ the less sensitive the test becomes at detecting small variations in the given neighborhood size.

As a last resort, when one wants to more reliably detect even the smallest disconnections,it becomes necessary to increase M; this however comes at the cost of increased usage of computational and communication resources. In `07-dyn-net.avi` and `08-dyn-net.avi` we picture two realizations for $k = 10$ and M equal to 100 and 500 respectively (see also the 'global view' in video `09-dyn-net.avi` in which we have set $M = 100$).

5.2.2 Building a gradient towards the disconnection

In principle information from the change detectors can be exploited in order to distributedly build a gradient towards the disconnection, i.e. somehow 'identifying' the direction in which the disconnection happened. An intuitive way to do so is to build gradients using the estimates of each nodes distance from the disconnected subgraph. One way of measuring this distance in hops/epochs is to employ the estimator

$$\widehat{\delta}^{(i)}(t) := \min_{j \in \mathcal{N}(t)} \left\{ \mathcal{K}^{(j)}(t) \right\} \ .$$

Thus $\delta^{(i)}(t)$ is the smallest k for which node i detected a change in its k-steps neighborhoods.

In the attached video `10-dyn-net.avi` we deactivate and then activate a group of nodes on the upper-right corner of the grid every 30 epochs. Recall that nodes with local alarms are drawn with shades of red: bright red if $\delta^{(i)}(t) = 1$ and darker reds as $\delta^{(i)}(t)$ increases towards D. The animation shows how a gradient is naturally build towards the disconnection.

The same qualitative results are obtained also for other kinds of network failures. See for example `11-dyn-net.avi` in which a hole like disconnection happens in the center of the grid.

In these examples all the nodes still active in the network after the failure event where able to detect the change of topology. More subtle behavior emerge when the network fails in a way that effects only the communication channels such that all the component nodes are still active and part of a single connected component. One such example is given in `12-dyn-net.avi` where we disable some of the communication links leaving the graph with a comb like topology. Unintuitively (although correctly) the nodes in the two lower rows of the grid do not detect any change in their neighborhoods. This is in the nature of our approach: by design it can only detect changes in the *size* of a node's neighborhood.

We believe however that our approach could in principle be extended to detect changes in the connectivity of a given k-steps neighborhood. Let us consider again the initial part of `02-static-net.avi`. The video shows how the evolution of $\widehat{S}_{40}(t)$ starts with a slow increase, followed by higher rate of increase as the tracked node aggregates information from the (more tightly connected) center part of the grid, followed again by a slower increase rate when the tracked node finally receives information that had been generated by nodes on the opposite corner of the grid. Thus, intuitively detecting changes on the level of connectivity of a given neighborhood could be performed by inspection of the evolution of the differences $\widehat{S}_k(t) - \widehat{S}_{k-1}(t)$ as time passes.

6 Conclusion

In this report we have considered two problems: the first was estimating the size of a network under the constraint of anonymity by exploiting consensus and the second was formulating a test for detecting topology changes based on such estimates.

We started by recalling some results for two size estimators from [14]: one estimator for the average-consensus and common Bernoulli distribution case and a second estimator in which max-consensus is paired with a common uniform distribution. We have found a refined a bound on the error probability of the estimator in the Bernoulli case which performs slightly better than the previous state of art. Unfortunately this result has little practical interest due to the slow convergence rate and poor scalability of average consensus. Moreover given the discrete nature of the estimator statistic and its high non-linearity this strategy is higly sensitive to numerical errors (see also the comments in [15]).

In the second part of the report we have first introduced the concept of k-steps neighborhood and provided an estimation algorithm which generalizes the max-consensus estimation strategy for static network. We then formulated a GLR test for detecting topology changes exploiting such neighborhood size estimates and precoded to its characterization in terms of the type I and II error probabilities proving also some optimality results.

A set of simulations has highlighted the biggest shortcoming in our approach in the noisiness and biased nature of the pre-change estimate. Two tunable parameters were introduced, $\overline{N}$ and σ, in order to contrast the deleterious effects of inaccurate pre-change size estimates and have provided directions for tuning these parameters in order to optimize the rate of false-positives.

At last we have discussed how the network could distributedly build a gradient towards the disconnected subgraph starting from the outcomes of our change-detection test.

Future directions could be enhancing the pre-change size estimator and the change time estimator by introducing a model of how the network changes topology over time. Perhaps, even without introducing such stochastic model, a less naïve way of estimating the change time could be devised then the current ML procedure.

References

[1] Networkx, complex network analysis for python. `http://networkx.lanl.gov`.

[2] Scipy, scientific tools for python. `www.scipy.org`.

[3] The Python language. `www.python.org`.

[4] Nadeem Ahmed, Salil S. Kanhere, and Sanjay Jha. The holes problem in wireless sensor networks. *ACM SIGMOBILE Mobile Computing and Communications Review*, 9(2):4, April 2005.

[5] Carlos Baquero, Paulo Sergio Sérgio Almeida, Raquel Menezes, and Paulo Jesus. Extrema Propagation: Fast Distributed Estimation of Sums and Network Sizes. *IEEE Transactions on Parallel and Distributed Systems*, 23(4):668 –675, April 2012.

[6] Michele Basseville and I. V. Igor V Nikiforov. *Detection of Abrupt Changes: theory and application*. Prentice-Hall, April 1993.

[7] Paolo Boldi and Sebastiano Vigna. Computing vector functions on anonymous networks. In *Proceedings of the 16th annual ACM symposium on Principles of distributed computing*, 1997.

[8] Israel Cidon and Yuval Shavitt. Message terminating algorithms for anonymous rings of unknown size. *Information Processing Letters*, 54(2):111 – 119, April 1995.

[9] Christos Gkantsidis, Milena Mihail, and Amin Saberi. Random walks in peer-to-peer networks: algorithms and evaluation. *Performance Evaluation*, 63(3):241 – 263, March 2006.

[10] Keren Horowitz and Dahlia Malkhi. Estimating network size from local information. *Information Processing Letters*, 88(5):237 – 243, December 2003.

[11] E.L. Lehmann and Joseph P. Romano. *Testing Statistical Hypotheses*. Springer, 2005.

[12] C. Radhakrishna Rao. *Linear statistical inference and its applications*. John Wiley and Sons, 2002.

[13] Damiano Varagnolo, Gianluigi Pillonetto, and Luca Schenato. Distributed statistical estimation of the number of nodes in Sensor Networks. In *IEEE Conference on Decision and Control*, pages 1498–1503, Atlanta, USA, December 2010. Ieee.

[14] Damiano Varagnolo, Gianluigi Pillonetto, and Luca Schenato. Distribuited size estimation in anonymous networks. *na*, 2011.

[15] Damiano Varagnolo, Gianluigi Pillonetto, and Luca Schenato. Consensus based estimation of anonymous networks size using Bernoulli trials. In *American Control Conference*, 2012.