

Gossip algorithms for simultaneous distributed estimation and classification in sensor networks

Alessandro Chiuso, *Senior Member, IEEE*, Fabio Fagnani, Luca Schenato *Member, IEEE*, Sandro Zampieri

Abstract—In this work we consider the problem of simultaneously classifying sensor types and estimating hidden parameters in a network of sensors subject to gossip-like communication limitations. In particular, we consider a network of noisy sensors which measure a common scalar unknown parameter. We assume that a fraction of the nodes is subject to the same (but possibly unknown) offset. The goal for each node is to simultaneously identify the class to which the node belongs and to estimate the common unknown parameter, only through local communication and computation. We propose a distributed estimator based on the maximum likelihood (ML) approach and we show that, in case the offset is known, this estimator converges to the centralized ML as the number N of sensor nodes goes to infinity. We also compare this strategy with a distributed implementation of the estimation-maximization (EM) algorithm; we show trade-offs via numerical simulations in terms of robustness, speed of convergence and implementation simplicity.

I. INTRODUCTION

In recent years, we have witnessed an increasing interest in the design of control and estimation algorithms which can operate in a distributed manner over a network of locally communicating units. A prototype of such problems is the average consensus algorithm [1], [2], which can be used as a distributed procedure providing the average of real numbers, each of them belonging to a unit. Since the average is the building block for many estimation methods, the average consensus has been proposed as a possible way to obtain distributed estimation algorithms and, in particular, to obtain distributed Kalman filtering [3], [4]. However, while averaging is suitable for the estimation of real valued parameters, it is typically of no help when the quantities to be estimated belong to a finite alphabet. Moreover, the average is by definition an operation which fuses information losing in this way the possible information that is specific of each unit. The model we consider in the present paper has two characteristics: the information of each unit contains both a common scalar parameter and a unit specific parameter and this second parameter belongs to a finite alphabet.

This research has been partially supported by EU FP7-ICT-223866-FeedNetBack project, by CaRiPaRo Foundation “WISE-WAI” project and by the Progetto di Ateneo CPDA090135/09 funded by the University of Padova.

Alessandro Chiuso is with the Department of Management and Engineering, Università di Padova, Stradella S. Nicola, 3 - 36100 Vicenza, Italy, ph. +39-049-8277709, fax. +39-049-8277699, e-mail: chiuso@dei.unipd.it

Fabio Fagnani is with the Department of Mathematics, Polytechnic of Turin fabio.fagnani@polito.it

Luca Schenato and Sandro Zampieri are with the Department of Information Engineering, Università di Padova, Via Gradenigo, 6/b, 35100 Padova, Italy. e-mail: {schenato, zampieri}@dei.unipd.it

More precisely, we assume that we have N units and that each unit i measures y_i that can be expressed as

$$y_i = \theta + T_i + v_i. \quad (1)$$

where $\theta \in \mathbb{R}$ is a continuous parameter influencing all the units, $T_i \in \mathcal{A}$, with \mathcal{A} being a finite set, is a discrete parameter influencing each unit independently and v_i is a noise term. The goal of each unit is to estimate the common parameter θ and its specific one T_i . Notice that the presence of the common parameter θ impose that any efficient estimation technique will require cooperation between units and therefore will require communication. We will assume that communication between the units can occur only according to a graph as discussed in Section III, which is devoted to the distributed algorithm description.

There are various examples of applications in which the previous estimation problem could be of interest. One application is related to fault detection. In this case the units represent some sensors that, when working properly, measure a noisy version of the parameter θ and that, when faulty, add a bias to the measurement. A similar situation is when there are heterogeneous sensors belonging to classes which differ by the bias they add. In both cases the parameter of primary interest is θ . Another example is when there are different units belonging to different classes, the objective being to classify them based on the y_i 's while also estimating the common parameter θ . As a possible application of this last scenario, we can imagine a network for environmental monitoring; the different values of T_i could model for instance a constant external field only active in certain areas where the sensor is located, such as for instance being in the sunshine or in the shade or being inside or outside of a fire.

More in general, these problems fit in the general class of the unsupervised clustering problems, which are quite standard in statistics [5], [6]. Algorithms for clustering have been widely proposed in the computer science literature both for the standard centralized case [7] and for the distributed case [8], [9], [10], [11]. Indeed, the technique proposed in this paper can be seen as a distributed algorithm for a specific clustering problem. Preliminary work along this direction can be found in our conference paper [12]. Note also that, with the purpose of providing a comparison of the algorithm we propose with more “standard” techniques such as the EM algorithm, we have faced the problem of implementing alternating-type algorithms in distributed scenarios. This is, in our view, a topic which has its own interest but which is outside the scope of this work; as such we plan to address this problem in future work.

The structure of the paper is as follows: Section II introduces the model we consider; the decentralized estimator is studied in Section III while its limit behavior is characterized in Section IV. In Section V an alternative approach based on a Bayesian model is presented and some generalizations are discussed in Section VI. Section VII describes a distributed implementation of alternating-type algorithms such as those found Sections V and VI. Some simulations are presented in Section VIII while conclusions are drawn in Section IX. In order to streamline exposition all proofs are postponed to Appendix A.

II. THE MODEL

In this section we give a more precise description of the model we consider and of the estimation cost we aim at minimizing by the proposed estimation algorithm. Assume that the measurements y_i are as in (1), where $\theta \in \mathbb{R}$, v_i are zero mean, independent Gaussian random variables with variance σ^2 ; for simplicity, with respect to what mentioned in the introduction, we will restrict to the case in which T_i can take only two values, that are supposed to be known and which, with no loss of generality¹, can be supposed to be 0 and 1, i.e. $T_i \in \{0, 1\}$. The goal of each unit i is to estimate θ and T_i .

Extension to the case in which the difference between the two symbols is unknown are discussed in Section VI. The algorithm we propose does not need to know the variance σ^2 which therefore can be assumed to be unknown.

A. The maximum likelihood estimator

When the bias term T_i is not present, the centralized maximum likelihood estimator of θ (assuming that all measurements y_i are available) is given by

$$\hat{\theta} = N^{-1} \sum_i y_i. \quad (2)$$

This arithmetic average can be asymptotically evaluated by the agents in the graph through standard consensus algorithms as long as the graph is strongly connected.

The presence of the bias terms makes the problem quite harder. In this paper we propose a decentralized version of the centralized maximum likelihood estimator for this problem. We set some useful notation. We consider the vectors $y := (y_1, \dots, y_N)$ and $T := (T_1, T_2, \dots, T_N)$ and the following weights $w(T) := \sum T_i$, $w(y) := \sum y_i$. The maximum likelihood estimator is defined as

$$(\hat{\theta}^{ML}, \hat{T}^{ML}) := \underset{(\theta, T)}{\operatorname{argmax}} P(y|\theta, T) = \underset{(\theta, T)}{\operatorname{argmin}} \left[\sum_i \frac{(y_i - \theta - T_i)^2}{2\sigma^2} \right] \quad (3)$$

Remark 1: The choice of the maximum likelihood estimator is motivated by the simplicity of the solution we obtain from it. Of course, it would be natural to seek for “optimal” estimators which minimize, e.g., the variance of $\hat{\theta}$, $\mathbb{E}[(\hat{\theta} - \theta)^2]$ and/or the average classification error $\mathbb{E}[\sum_{i=1}^N |\hat{T}_i - T_i|]$. Unfortunately these optimal estimators are in general computationally intractable even in the centralized case. We

¹The solution we propose can be extended immediately to the case in which $T_i \in \{a, b\}$ where a, b are assumed to be known real parameters.

will show instead that the maximum likelihood estimator is not only computationally simple, but also prone to a decentralized implementation.

It is easy to solve the minimization in (3) for a fixed T :

$$\begin{aligned} \hat{\theta}(T) &:= \underset{\theta}{\operatorname{argmin}} \left[\sum_i \frac{(y_i - \theta - T_i)^2}{2\sigma^2} \right] \\ &= \frac{1}{N} \sum_i (y_i - T_i) = \frac{w(y) - w(T)}{N} \end{aligned} \quad (4)$$

The estimator $\hat{\theta}(T)$ is then a function of the average $N^{-1}w(y)$, which can be obtained by a standard consensus algorithm, and of the average bias $N^{-1}w(T)$. This second term however is not directly available, so that (4) is not an implementable solution. Rather, substituting (4) in (3) we obtain

$$\hat{T}^{ML} = \underset{T}{\operatorname{argmin}} \left[\sum_i \frac{\left(y_i - \frac{w(y)}{N} + \frac{w(T)}{N} - T_i \right)^2}{2\sigma^2} \right] \quad (5)$$

This minimization can be solved in a two-step way by considering

$$\min_{w=0, \dots, N} \left[\min_{T: w(T)=w} \sum_i \frac{\left(y_i - \frac{w(y)}{N} + \frac{w}{N} - T_i \right)^2}{2\sigma^2} \right] \quad (6)$$

For every $w = 0, \dots, N$, put

$$\hat{T}_w = \underset{T: w(T)=w}{\operatorname{argmin}} \sum_i \frac{\left(y_i - \frac{w(y)}{N} + \frac{w}{N} - T_i \right)^2}{2\sigma^2} \quad (7)$$

Let us define

$$\eta_i := y_i - \frac{w(y)}{N}$$

and consider a permutation² $[\cdot] : \{1, \dots, N\} \rightarrow \{1, \dots, N\}$ such that $\eta_{[1]} \leq \eta_{[2]} \leq \dots \leq \eta_{[N]}$. Clearly, the above minimization is solved by the vector \hat{T}_w such that

$$(\hat{T}_w)_{[j]} = \begin{cases} 0 & \text{if } j \leq N - w \\ 1 & \text{otherwise} \end{cases} \quad (8)$$

Substituting in (6) and performing simple algebraic transformations, we obtain that the solution of the outer minimization problem in (6) becomes $\hat{w} = \operatorname{argmin} F(w)$, where

$$F(w) := -\frac{w^2}{N} + w - 2 \sum_{j=N-w+1}^N \eta_{[j]} \quad (9)$$

Clearly, from (8),

$$\hat{T}_{[j]}^{ML} = (\hat{T}_{\hat{w}})_{[j]} = \begin{cases} 0 & \text{if } j \leq N - \hat{w} \\ 1 & \text{otherwise} \end{cases} \quad (10)$$

and from (4) we get:

$$\hat{\theta}^{ML} = \frac{w(y) - \hat{w}}{N} = \frac{w(y) - w(\hat{T}^{ML})}{N} \quad (11)$$

²The subscript $[\cdot]$ is quite standard in statistics to denote ordered samples.

III. A DECENTRALIZED ESTIMATOR

Notice that each agent i can compute η_i by a consensus algorithm. Moreover, as will be discussed later, there exists an efficient decentralized algorithm capable of ordering the η_i , so that each agent i knows its ordered position o_i .

E.g. if the observation η_i of agent i is the smallest³ among all observations (i.e. $\eta_i < \eta_j, \forall j \neq i$) then $o_i = 1$; if y_k is the second smallest $o_k = 2$ and so on; more precisely, $o_i \in \{1, 2, \dots, N\}$ and

$$o_i < o_j \Rightarrow \eta_i \leq \eta_j.$$

The map $o : \mathcal{N} \rightarrow \mathcal{N}$ is a permutation of the set $\mathcal{N} = \{1, \dots, N\}$. Using a notation which is rather common in the statistics literature, we can define $[\cdot] : \mathcal{N} \rightarrow \mathcal{N}$ as the inverse permutation w.r.t o , i.e. $[o_i] = i$, which implies that

$$\eta_i = \eta_{[o_i]}.$$

For each value w , the agent i is thus capable of computing $(\hat{T}_w)_i$ through (8). In order to compute \hat{T}_i^{ML} using (10) we need to know the ordered position o_i of agent i with respect to $N - \hat{w}$. This would follow if we could compute \hat{w} in a decentralized fashion, but this is not at all evident, because of the presence of the aggregation term $\sum_{j=N-w+1}^N \eta_{[j]}$ in (9).

Consider the discrete derivative of F :

$$\Delta(w) := F(w+1) - F(w) = -\frac{2w+1}{N} + 1 - 2\eta_{[N-w]} \quad (12)$$

for $w = 1, \dots, N-1$. Notice that $\Delta(w)$ can be computed by the agent in ordered position $N-w$.

Define the set of local minima:

$$\mathcal{S} := \{w \in [1, N] \mid \Delta(w-1) < 0, \Delta(w) > 0\}$$

(interpreting, conventionally, $\Delta(0) < 0$ and $\Delta(N+1) > 0$ as always true assertions). If we knew that $|\mathcal{S}| = 1$ then our computational problem could be solved in the following way. Notice that in this case we would have that $F(w)$ decreases until its minimum point \hat{w} and then starts to increase. A generic agent i in position o_i can compute $\Delta(N-o_i)$. If $\Delta(N-o_i) < 0$ then $N-o_i < \hat{w}$, namely $o_i > N-\hat{w}$, which implies by (10) that $(\hat{T}_{\hat{w}})_{[o_i]} = 1$. If instead $\Delta(N-o_i) > 0$, then $(\hat{T}_{\hat{w}})_{[o_i]} = 0$. So, in this way, each agent could compute its ML estimator \hat{T}_i^{ML} . Again, using consensus all agents can then compute $N^{-1}\hat{w} = N^{-1}w(\hat{T}^{ML})$ and can therefore also compute θ using formula (4).

Of course, the decentralized algorithm proposed above can always be implemented by the agents. In the following part of the paper we will show that, typically, for N large, F possesses just one local minimum in $[0, 1/2]$ which happens to be the global minimum on $[0, 1]$, while possibly exhibiting other local minima on $]1/2, 1]$. It follows that, with high probability, the maximum likelihood estimator can be obtained by applying the previous algorithm to all agents i whose ordered position o_i is above $N/2$ while forcing all agents whose position o_i

is below $N/2$ to estimate $\hat{T}_{[o_i]} = 0$. We can summarize the previous reasoning in the following definitions:

$$\hat{T}_i^{AML} := \begin{cases} 1 & \text{if } 2\left(y_i - \frac{w(y)}{N}\right) > 1 - \frac{2(N-o_i)+1}{N} \wedge o_i > \frac{N}{2} \\ 0 & \text{otherwise} \end{cases}$$

$$\hat{\theta}^{AML} := \frac{w(y) - w(\hat{T}^{AML})}{N} \quad (13)$$

where the superscript *AML* stands for *approximate maximum likelihood*. This approximate maximum likelihood estimator converges (as $N \rightarrow \infty$) to the maximum likelihood estimator in (3) as formally stated in Corollary 12.

Before describing the algorithm to compute $(\hat{\theta}^{AML}, \hat{T}^{AML})$ in a distributed fashion, we need to introduce some useful general distributed algorithms that will be used in our algorithm.

A. Decentralized average and ranking computation

We model the network of distributed agents with a graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ where $\mathcal{N} = \{1, 2, \dots, N\}$ is the set of nodes and \mathcal{E} is the set of edges corresponding to the communication links. We indicate with $V(i)$, the set of neighbors of node i , i.e. $V(i) = \{j \mid (i, j) \in \mathcal{E}\}$. We assume that the graph is connected, i.e. there is a path between any two nodes, and it is undirected, i.e. nodes are capable of bidirectional communications. We also assume that each sensor node i knows its label i , i.e. nodes are numbered from 1 to N . As shown in the previous section, in order to compute the AML estimators $\hat{\theta}_i^{AML}$ and \hat{T}_i^{AML} , as given in (13), it is necessary to compute the averages of some quantities, namely $w(y)$ and $w(\hat{T}^{AML})$, and the ranking of each node o_i .

Distributed algorithms for computing averages are well studied and are also known as average consensus algorithms (see surveys [13] [14] and reference therein).

We shall denote with \mathcal{P}_{ave} an ‘‘average’’ operator which takes as arguments $(x_i^{(k)}, x_j^{(k)})$, where $x_i^{(k)}$ is the ‘‘local’’ state stored in node i at the k -th iteration and $(i, j) \in \mathcal{E}$, and returns the ‘‘updated’’ state as

$$(x_i^{(k+1)}, x_j^{(k+1)}) = \mathcal{P}_{ave}(x_i^{(k)}, x_j^{(k)}) \quad (14)$$

The following proposition provides the convergence properties of the Randomized Gossip Average Consensus, which have been well studied in [15] and [16].

Algorithm 1 Randomized Gossip Average Consensus [15]

Require: graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, probability distribution p_{ij} over \mathcal{E} , measurements y_i

- 1: **for all** node i **do**
 - 2: $x_i^{(0)} = y_i, k = 0$.
 - 3: **end for**
 - 4: **repeat**
 - 5: randomly select edge $(i, j) \in \mathcal{E}$ with $\mathbb{P}[(i, j)] = p_{ij}$
 - 6: $(x_i^{(k+1)}, x_j^{(k+1)}) = \mathcal{P}_{ave}(x_i^{(k)}, x_j^{(k)})$
 - 7: $x_h^{(k+1)} = x_h^{(k)}, \forall h \neq i, h \neq j$
 - 8: $k = k + 1$
 - 9: **until** $k > M$
-

³For simplicity of exposition we shall assume that it is not possible that any two agents have the same observation, i.e. $\nexists(i, j), i \neq j : \eta_i = \eta_j$.

Algorithm 2 Randomized Gossip Ranking

Require: graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, probability distribution p_{ij} over \mathcal{E} , measurements y_i , node “ i ” has ID i .

- 1: **for all** node i **do**
- 2: Initialize $\xi_i^{(0)} = \text{init}(y_i, i)$,
- 3: **end for**
- 4: **repeat**
- 5: randomly select edge $(i, j) \in \mathcal{E}$ with $\mathbb{P}[(i, j)] = p_{ij}$
- 6: $(\xi_i^{(k+1)}, \xi_j^{(k+1)}) = \mathcal{P}_{rk}(\xi_i^{(k)}, \xi_j^{(k)})$
- 7: $\xi_h^{(k+1)} = \xi_h^{(k)}, \forall h \neq i, h \neq j$
- 8: $k = k + 1$
- 9: **until** $k > M$
- 10: $rk_i^{(k)} = \text{extract}(\xi_i^{(k)}) \quad \forall i$

Proposition 2 (Gossip average consensus [15], [16]):
 Consider Algorithm 1. If the graph \mathcal{G} is connected and $p_{ij} > 0$ for all edges $(i, j) \in \mathcal{E}$ then we have

$$\lim_{k \rightarrow \infty} x_i^{(k)} = \frac{1}{N} \sum_{i=1}^N y_i \quad \text{almost surely}$$

The randomized implementation of the average consensus has the advantage to be asynchronous, i.e. nodes do not require time-synchronization or fine coordination, and to be parallelizable, i.e. several nodes can perform the updates at the same time as long as the updating node pairs are disjoint [15].

In the sequel of the paper we shall also need a distributed algorithm which is able to rank the nodes of a network based on the ordered list of the magnitude of their measurements. The algorithm is randomized in the same spirit of the previous randomized gossip average consensus, i.e. at each time an edge of the network is selected at random and corresponding nodes exchange information and update their local variable. The pseudo-code of this algorithm is given in Algorithm 2.

Without entering into the detail which can be found in [17], it suffices here to assume that this algorithm requires, for each node i , a local “state” $\xi_i^{(k)}$ at iteration k . This local state needs to be initialized as a function of the “local” measurement value y_i and node ID i ; we shall denote the initialization procedure as

$$\xi_i^{(0)} = \text{init}(y_i, i)$$

Similarly to what has been done above, we shall denote with \mathcal{P}_{rk} a “ranking” operator, defined in [17], which takes as arguments $(\xi_i^{(k)}, \xi_j^{(k)})$, where $(i, j) \in \mathcal{E}$, and returns the “updated” state as

$$(\xi_i^{(k+1)}, \xi_j^{(k+1)}) = \mathcal{P}_{rk}(\xi_i^{(k)}, \xi_j^{(k)}) \quad (15)$$

The state $\xi_i^{(k)}$ contains, in particular, a variable $rk_i^{(k)}$ which is the current estimate at node i of its rank. We shall call extract a procedure which extract this variable from $\xi_i^{(k)}$, i.e.

$$rk_i^{(k)} = \text{extract}(\xi_i^{(k)}) \quad (16)$$

The Randomized Gossip Ranking algorithm is sketched above. and its asymptotic behavior of this algorithm is formally stated in the following theorem [17]:

Theorem 3 (Randomized Gossip Ranking): Consider Algorithm 2. If the graph \mathcal{G} is connected and $p_{ij} > 0$ for all edges $(i, j) \in \mathcal{E}$ then there exists $T > 0$ such that $rk_i^{(k)}$ in (16) satisfies

$$rk_i^{(k)} = o_i \quad \forall k \geq T, i = 1, \dots, N \quad \text{almost surely}$$

B. Decentralized estimation and classification algorithm

We are now ready to present the algorithm that allows each sensor i to compute the approximate maximum likelihood (AML) estimate for the unknown parameter θ and for its unknown class T_i . The algorithm is based on the randomized gossip average consensus and ranking presented in the previous section and it is summarized in Algorithm 3.

Algorithm 3 Gossip Estimation and Classification

Require: graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, probability distribution p_{ij} over \mathcal{E} , measurements y_i , node “ i ” has ID i .

- 1: **for all** node i **do**
- 2: $\eta_i^{(0)} = y_i, w_i^{(0)} = 0, \hat{\theta}_i^{(0)} = y_i, \hat{T}_i^{(0)} = 0$
- 3: lines 1-3 of Algorithm 1
- 4: line 1-3 of Algorithm 2
- 5: **end for**
- 6: **repeat**
- 7: randomly select edge $(i, j) \in \mathcal{E}$ with $\mathbb{P}[(i, j)] = p_{ij}$
- 8: lines 6-7 of Algorithm 1
- 9: lines 6-7 of Algorithm 2
- 10: $rk_i^{(k+1)} = \text{extract}(\xi_i^{(k+1)}), rk_j^{(k+1)} = \text{extract}(\xi_j^{(k+1)})$
- 11: $\eta_i^{(k+1)} = y_i - x_i^{(k+1)}, \eta_j^{(k+1)} = y_j - x_j^{(k+1)}$
- 12: **if** $rk_i^{(k+1)} \geq \frac{N}{2} \wedge 2\eta_i^{(k+1)} > 1 - \frac{2(N - rk_i^{(k+1)}) + 1}{N}$ **then**
- 13: $\hat{T}_i^{(k+1)} = 1$
- 14: **else**
- 15: $\hat{T}_i^{(k+1)} = 0$
- 16: **end if**
- 17: **if** $rk_{loc,j}^{(k+1)} \geq \frac{N}{2} \wedge 2\eta_j^{(k+1)} > 1 - \frac{2(N - rk_{loc,j}^{(k+1)}) + 1}{N}$ **then**
- 18: $\hat{T}_j^{(k+1)} = 1$
- 19: **else**
- 20: $\hat{T}_j^{(k+1)} = 0$
- 21: **end if**
- 22: $w_i^{(k+1)} = \frac{w_i^{(k)} + w_j^{(k)}}{2} + (\hat{T}_i^{(k+1)} - \hat{T}_i^{(k)})$
- 23: $w_j^{(k+1)} = \frac{w_i^{(k)} + w_j^{(k)}}{2} + (\hat{T}_j^{(k+1)} - \hat{T}_j^{(k)})$
- 24: $\hat{\theta}_i^{(k+1)} = x_i^{(k+1)} - w_i^{(k+1)}$
- 25: $\hat{\theta}_j^{(k+1)} = x_j^{(k+1)} - w_j^{(k+1)}$
- 26: **for all** $\ell = 1, \dots, N, \ell \neq i, \ell \neq j$ **do**
- 27: line 9 of Algorithm 1
- 28: lines 26-29 of Algorithm 2
- 29: $\eta_i^{(k+1)} = \eta_i^{(k)}, w_i^{(k+1)} = w_i^{(k)}$
- 30: $\hat{T}_i^{(k+1)} = \eta_i^{(k)}, \hat{\theta}_i^{(k+1)} = w_i^{(k)}$
- 31: **end for**
- 32: $k = k + 1$
- 33: **until** $k > M$

In practice, it is necessary to compute the mean of the measurements $w(y)$ (Algorithm 1) and the ranking of the nodes o_i (Algorithm 2) to evaluate the condition given by (13) that

correspond to lines 11-20 of Algorithm 3. The variables $\eta_i^{(k)}$, $w_i^{(k)}$ represent, respectively, the estimates of node i at time k of the displacement of its measurement from the average, i.e. $y_i - \frac{1}{N} \sum_{i=1}^N y_i$, and of the fraction of nodes that belongs to the class “1”, i.e. $\frac{1}{N} \sum_{i=1}^N \hat{T}_i^{AML}$. The variables $\hat{T}_i^{(k)}$, $\hat{\theta}_i^{(k)}$ instead represent the estimate of node i at time k of the unknown node class T_i and the unknown parameter θ . From Theorems 2 and 3 it follows that the estimates $w_i^{(k)}$ and $x_i^{(k)}$ will converge to o_i and $w(y)$, therefore the conditions stated in lines 11 and 16 will coincide with the condition of (13). As a consequence \hat{T}_i will converge (as the number of gossip iterations goes to infinity) to the asymptotic maximum likelihood \hat{T}_i^{AML} . In order to compute the centralized approximate ML estimate of the parameter θ it is necessary to compute the fraction of the “1”-class nodes $w(\hat{T}^{AML}) = \frac{1}{N} \sum_{i=1}^N \hat{T}_i^{AML}$. This is achieved by lines 21-22 in the algorithm which correspond to an average consensus applied to the time increments of the input signal $\hat{T}_i^{(k)}$. This guarantees that $\sum_{i=1}^N w_i^{(k)} = \sum_{i=1}^N \hat{T}_i^{(k)}$ at every time instant k . Since all $\hat{T}_i^{(k)}$ converge to \hat{T}_i^{AML} , then the input signal to equations of lines 21-22 tends to zero and asymptotically each $w_i^{(k)}$ will converge to $w(\hat{T}^{AML})$. This claim is formally stated in the next theorem:

Proposition 4: Consider Algorithm 3. If the graph \mathcal{G} is connected and $p_{ij} > 0$ for all edges $(i, j) \in \mathcal{E}$ then we have:

$$\lim_{k \rightarrow \infty} \hat{T}_i^{(k)} = \hat{T}_i^{AML} \quad \text{almost surely} \quad (17)$$

$$\lim_{k \rightarrow \infty} \hat{\theta}_i^{(k)} = \hat{\theta}^{AML} \quad \text{almost surely} \quad (18)$$

IV. THE LIMIT BEHAVIOR

In what follows we study the behavior (in particular the monotonicity) of the objective random function F when $N \rightarrow +\infty$. To emphasize dependence on N , from now on we will use the notation F_N .

We recall that, in our approach, the bias values T_i are fixed, even if unknown to the agents. We put

$$I^1 = \{i = 1, \dots, N \mid T_i = 1\}, \quad I^0 = \{i = 1, \dots, N \mid T_i = 0\}$$

and we assume that

$$\lim_{N \rightarrow +\infty} \frac{|I^1|}{N} = \lim_{N \rightarrow +\infty} \frac{w(T)}{N} = p \in [0, 1/2[\quad (19)$$

We start with some preliminary considerations on the ordered variables $\eta_{[w]}$. We can write $\eta_i = \xi_i + \Omega$ where

$$\xi_i = T_i + v_i, \quad \text{and} \quad \Omega = \frac{w(v)}{N} - \frac{w(T)}{N}. \quad (20)$$

The variables ξ_i are thus independent and have two possible distribution functions:

$$\begin{aligned} \mathbb{P}(\xi_i \leq t) &= \Phi_\sigma(t-1) & \text{if } i \in I^1 \\ \mathbb{P}(\xi_i \leq t) &= \Phi_\sigma(t) & \text{if } i \in I^0 \end{aligned} \quad (21)$$

where

$$\Phi_\sigma(a) := \frac{1}{\sqrt{2\pi}\sigma} \int_{-\infty}^a e^{-\frac{x^2}{2\sigma^2}} dx$$

Notice now that

$$\xi_{[w]} < t \Leftrightarrow \Lambda_t := |\{i \mid \xi_i < t\}| \geq w \quad (22)$$

Put $\Lambda_t^q := |\{i \in I^q \mid \xi_i < t\}|$ for $q = 0, 1$. Λ_t^1 and Λ_t^0 are two Binomial r.v. of type, respectively, $B(|I^1|, \Phi_\sigma(t-1))$ and $B(|I^0|, \Phi_\sigma(t))$. Since, $\Lambda_t = \Lambda_t^1 + \Lambda_t^0$, we have that $\mathbb{E}(\Lambda_t) = |I^1| \Phi_\sigma(t-1) + |I^0| \Phi_\sigma(t)$ and

$$\lim_{N \rightarrow +\infty} \frac{\mathbb{E}(\Lambda_t)}{N} = F_\xi(t) := p\Phi_\sigma(t-1) + (1-p)\Phi_\sigma(t) \quad (23)$$

Let us now consider the following normalized and scaled version of $F_N(w)$:

$$\begin{aligned} \bar{F}_N(w) &:= \frac{1}{N} F_N(Nw) = -w^2 + w - \frac{2}{N} \sum_{k=\lfloor N(1-w)+1 \rfloor}^N \eta_{[k]} \\ &= -w^2 + w - 2w\Omega - \frac{2}{N} \sum_{k=\lfloor N(1-w)+1 \rfloor}^N \xi_{[k]}, \quad w \in [N^{-1}, 1] \end{aligned}$$

Equations (22) and (23) suggest that $\xi_{[w]}$ and $F_\xi^{-1}(w/N)$ should be close to each other for large N . We can thus guess that:

$$\begin{aligned} \lim_{N \rightarrow \infty} \bar{F}_N(w) &\stackrel{a.s.}{=} \mathcal{F}(w) \\ \mathcal{F}(w) &:= -w^2 + w + 2p\omega - 2 \int_{1-\omega}^1 F_\xi^{-1}(t) dt, \quad w \in (0, 1] \end{aligned} \quad (24)$$

Likely enough, local extrema of F_N will converge, almost surely, to the local extrema of \mathcal{F} so that if \mathcal{F} possess just one local minimum on $[0, 1/2]$ which is the global minimum, then this will also happen for F_N almost surely when $N \rightarrow +\infty$. This would mean that our decentralized algorithm will almost surely coincide with the centralized ML algorithm. The next section will make precise all these considerations.

A. The analysis of the function $\mathcal{F}(w)$

In spite of its apparent simplicity, an analytical study of the function \mathcal{F} is not easy to obtain. It is immediate to verify that \mathcal{F} is continuous. Regarding its monotonicity, numerical investigations seem to show that \mathcal{F} can have one or two local minima depending on the particular values for σ and p , i.e. the derivative of \mathcal{F} is equal to zero once or three times. However, the derivative of \mathcal{F} seems to be equal to zero in only one point in $w \in (0, 1/2)$ which corresponds to the global minimum.

One case which can be studied in detail is the “small noise” case, i.e. the limit $\sigma \rightarrow 0$; this is done in the following proposition:

Proposition 5: Under the assumption of model given by (3) we have that

$$\lim_{\sigma \rightarrow 0} \mathcal{F}(w) = -w^2 + w + 2p\omega - 2p - 2(\omega - p)\delta_{-1}(p - \omega)$$

uniformly for $w \in [0, 1]$, where $\delta_{-1}(x)$ is equal to one for positive x and zero otherwise. Moreover, if $\hat{w}(\sigma) := \operatorname{argmin}_w \mathcal{F}(w)$, then

$$\lim_{\sigma \rightarrow 0} \hat{w}(\sigma) = p.$$

The previous proposition states that, if the two distributions degenerate to a single point, then the proposed algorithm exactly compute the proportions measurements generated by each of the two Gaussian distributions. However, when there is substantial overlap, the estimation has a bias toward the midpoint $1/2$, and in the limit of very large variance estimate

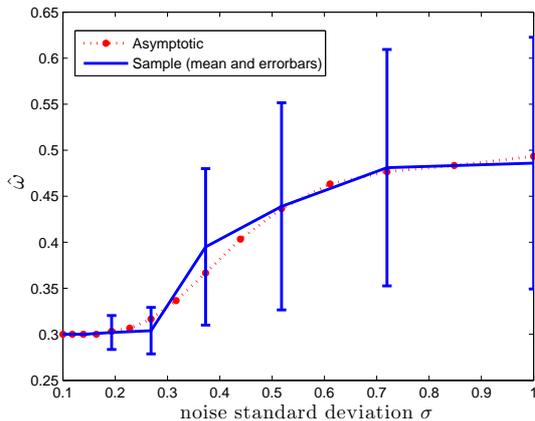


Fig. 1. Minima of $\mathcal{F}(\omega)$ (asymptotic) and of $\bar{F}_N(\omega)$ (sample, 10 Monte Carlo runs) vs. noise standard deviation σ . Data are always generated with $p = \frac{\sum \tilde{I}_i}{N} = 0.3$.

$\hat{\omega}$ is completely uninformative. In fact it can be verified that (see also Fig. 1)

$$\lim_{\sigma \rightarrow +\infty} \hat{\omega} := \operatorname{argmin}_{\omega} \mathcal{F}(\omega) = \frac{1}{2}$$

The value of the minimum $\hat{\omega}$ of the asymptotic function $\mathcal{F}(\omega)$ as a function of the noise variance σ for $p = 0.3$ is reported in Figure 1 (dotted line). As stated in the previous proposition, $\hat{\omega} = p$ for small σ and $\hat{\omega} = 1/2$ for large σ . As mentioned above, the graph shows that this minimum monotonically increases from p to $1/2$, thus confirming the hypothesis that the global minimum is always in the interval $(0, 1/2)$ for all values of p and σ . Figure 1 also shows the mean and standard deviation of the minimum of $\bar{F}_N(\omega)$ over 10 Monte Carlo runs for $N = 100$ sensor nodes.

B. The concentration results

In the sequel we present some concentration results which make rigorous the considerations done above. We recall a standard result on the concentration of binomial r.v. which will be our main technical tool.

Theorem 6: Let Z be a binomial r.v of type $B(N, p)$. Put, for $x > 0$, $\gamma(x) = x \log x - x + 1$. Then, for any $x < 1 < y$, it holds

$$\mathbb{P}(Z \leq Npx) \leq e^{-Np\gamma(x)}, \quad \mathbb{P}(Z \geq Npy) \leq e^{-Np\gamma(y)}$$

Remark: Notice that, for any $y_0 > 1$, there exists a constant $C > 0$, such that $\gamma(y) \geq Cy \log y$

The following result is standard but we will give an elementary proof in the Appendix for the sake of making the paper self-contained.

Lemma 7: For any $0 < a < b < 1$ and for every $\delta > 0$, there exists $l_\delta > 0$ such that, for N sufficiently large,

$$\mathbb{P}(|\xi_{[j]} - F_\xi^{-1}(j/N)| \geq \delta) \leq e^{-Nl_\delta}, \quad \forall j \in [aN, bN]$$

With the following bound we take care of the behavior of $\xi_{[j]}$ for j close to 0 and to N .

Lemma 8: There exist $0 < a < 1$ and $l > 0$ such that, for N sufficiently large and for $j \in [1, aN]$, it holds

$$\begin{aligned} \mathbb{P}(\xi_{[j]} \leq -(N/j)^{1/2}) &\leq e^{-lN} \\ \mathbb{P}(\xi_{[N-j]} \geq (N/j)^{1/2}) &\leq e^{-lN} \\ \mathbb{P}(\xi_{[N]} \geq N^{1/2}) &\leq e^{-lN} \end{aligned} \quad (25)$$

Theorem 9: For every $\delta > 0$ there exists $L_\delta > 0$ such that, for N sufficiently large,

$$\mathbb{P}(\exists w : \left| \frac{F_N(w)}{N} - \mathcal{F}\left(\frac{w}{N}\right) \right| \geq \delta) \leq e^{-NL_\delta}$$

Since our decentralized algorithm is influenced by the position of the local minima of F_N in $[0, 1/2]$, the result above is not sufficient to study the performance. Indeed, we need to study the asymptotic behavior of the variation function $\Delta(w)$.

Theorem 10: For every $\delta > 0$, there exists $\tilde{L}_\delta > 0$ such that

$$\mathbb{P}(\exists w : \left| \Delta(w) - \mathcal{F}'\left(\frac{w}{N}\right) \right| \geq \delta) \leq e^{-N\tilde{L}_\delta}$$

for N sufficiently large.

Proposition 11: Consider an interval $[a, b] \subseteq [0, 1]$ and $\varepsilon > 0$. Then,

$$\begin{aligned} \mathcal{F}'(x) \geq \varepsilon \quad \forall x \in [a, b] &\Rightarrow \mathbb{P}(\Delta(w) \geq 0 \quad \forall w \in [Na, Nb]) \geq p_\varepsilon(N) \\ \mathcal{F}'(x) \leq -\varepsilon \quad \forall x \in [a, b] &\Rightarrow \mathbb{P}(\Delta(w) \leq 0 \quad \forall w \in [Na, Nb]) \geq p_\varepsilon(N) \end{aligned} \quad (26)$$

where $p_\varepsilon(N) := 1 - Ce^{-\tilde{L}_\varepsilon N}$.

Proof Immediate consequence of Theorem 10 applied with $\delta = \varepsilon$. \blacksquare

We are now ready to state and prove the main theoretical result of our work. Denote by S_N the set of local minima of F_N in $[0, 1/2]$ and by S_N^{glob} the subset of S_N consisting of the global minima of F_N living in $[0, 1/2]$ (of course a priori this set could as well be empty).

Corollary 12: Assume that

- $\min_{\omega \in [0, 1/2]} \mathcal{F}(\omega) < \min_{\omega \in [1/2, 1]} \mathcal{F}(\omega)$.
- \mathcal{F} admits just one local minimum point $\bar{\omega}$ in $[0, 1/2]$ (which is thus the only global minimum for (a)).

Then, for every $\delta > 0$, there exists $J_\delta > 0$ such that

$$\mathbb{P}(S_N/N \subseteq [\bar{\omega} - \delta, \bar{\omega} + \delta]) \geq 1 - Ce^{-J_\delta N} \quad (27)$$

$$\mathbb{P}(S_N^{\text{glob}} \neq \emptyset) \geq 1 - Ce^{-J_\delta N}$$

By the way the approximate ML algorithm has been defined, the condition expressed above in (27) yields

$$\mathbb{P}(w(|\hat{T}^{\text{AML}} - \hat{T}^{\text{ML}}|)/N \leq 2\delta) \geq 1 - Ce^{-J_\delta N} \quad (28)$$

In other terms, the approximate ML algorithm is close to the ML solution with high probability for large N . We would like to remark that conditions (a) and (b) of Corollary 12 can easily be checked numerically and turn out to be satisfied in all examples considered. An analytical proof of these conditions is at the moment not available except for the limit case $\sigma \rightarrow 0$ treated in Proposition 5.

V. BAYESIAN MODELING AND EM

An alternative approach to this estimation and detection problem is possible if one postulates that T_i , $i = 1, \dots, N$ are independent and identically distributed (i.i.d.) Bernoulli random variables with parameter p

$$T_i \sim \mathcal{B}(p) \quad p := \mathbb{P}(T_i = 1). \quad (29)$$

so that

$$P(T|p) = \prod_{i=1}^N p^{T_i} (1-p)^{1-T_i} \quad (30)$$

Hence, one can formulate the problem of estimating p , θ and σ from measurements y_1, \dots, y_N . The maximum likelihood estimator is defined by

$$(\hat{p}, \hat{\theta}, \hat{\sigma}) := \arg \max_{p, \theta, \sigma} \sum_{T \in \{0,1\}^N} P(y|\theta, \sigma, T) P(T|p) \quad (31)$$

where $P(T|p)$ has been defined in (30) and

$$P(y|\theta, \sigma, T) \propto e^{-\frac{1}{2\sigma^2} \sum_i (y_i - \theta - T_i)^2}.$$

Note that in the estimation problem (3) the number of unknowns grows with the number of data; instead the i.i.d. assumption on the T_i 's allows to keep the parameter space in (31) of fixed dimension. As a result, the asymptotic properties of the estimators in (31), such as consistency and asymptotic efficiency, follow straightforwardly from standard asymptotic theory of maximum likelihood estimators, see [18].

An estimator of the variables T_1, \dots, T_N can then be obtained by maximizing the maximum likelihood estimator $\hat{P}(T|y)$ of the posterior probability

$$P(T|y) \propto P(y|T, \theta, \sigma) P(T|p)$$

i.e.

$$(\hat{T}_1, \dots, \hat{T}_N) := \arg \max_{T \in \{0,1\}^N} \hat{P}(T|y).$$

The maximum likelihood estimator $\hat{P}(T|y)$ of the posterior probability $P(T|y)$ is given, from the invariance principle (see e.g. [18]), by

$$\begin{aligned} \hat{P}(T|y) &= c P(y|T, \hat{\theta}, \hat{\sigma}) P(T|\hat{p}) \\ &= c e^{-\frac{1}{2} \sum_{i=1}^N \left(\frac{y_i - \hat{\theta} - T_i}{\hat{\sigma}} \right)^2} + \ln \left(\frac{\hat{p}}{1-\hat{p}} \right)^{\sum_{i=1}^N T_i} \end{aligned} \quad (32)$$

where c is a suitable normalization constant.

The maximum likelihood problem (31) is a typical estimation problem for a finite mixture distribution (see [5]) and does not have a closed form solution. One possible approach is to resort to the well known Expectation-Maximization (EM) algorithm in [19]. This is an iterative algorithm which is known to converge to a local maxima of the likelihood. For reasons of space we shall only report the final equations for EM iterations. We refer the reader to the book by [5] for a derivation of the EM algorithm which can be easily adapted to this specific problem.

Let $\hat{\theta}^{(k)}$, $\hat{\sigma}^{(k)}$ and $\hat{p}^{(k)}$ the estimators at the k -th iteration of the EM algorithm; the estimators for the $(k+1)$ -th iteration are given by:

1) **Expectation Step:** compute the posterior probabilities

$$\begin{aligned} \hat{\mu}_j^{(k+1)} &:= \mathbb{P}(T_j = 1 | y, \hat{\theta}^{(k)}, \hat{p}^{(k)}, \hat{\sigma}^{(k)}) \\ &= \frac{\hat{p}^{(k)} e^{-\frac{1}{2} \left(\frac{y_j - \hat{\theta}^{(k)} - 1}{\hat{\sigma}^{(k)}} \right)^2}}{\hat{p}^{(k)} e^{-\frac{1}{2} \left(\frac{y_j - \hat{\theta}^{(k)} - 1}{\hat{\sigma}^{(k)}} \right)^2} + (1 - \hat{p}^{(k)}) e^{-\frac{1}{2} \left(\frac{y_j - \hat{\theta}^{(k)}}{\hat{\sigma}^{(k)}} \right)^2}} \end{aligned} \quad (33)$$

2) **Maximization Step:**

$$\begin{aligned} \hat{p}^{(k+1)} &= \frac{1}{N} \sum_{j=1}^N \hat{\mu}_j^{(k+1)} \\ \hat{\theta}^{(k+1)} &= \frac{1}{N} \sum_{j=1}^N y_j - \hat{p}^{(k+1)} \\ \hat{\sigma}^{(k+1)} &= \sqrt{\frac{1}{N} \sum_{j=1}^N \left(\underbrace{(y_j - \theta)^2 + \mu_j - 2\mu_j(y_j - \theta)}_{\theta = \hat{\theta}^{(k+1)} \quad \mu_j = \hat{\mu}_j^{(k+1)}} \right)} \end{aligned} \quad (34)$$

The EM algorithm (33),(34) has a ‘‘centralized’’ nature. However it can be easily decentralized (i.e. computed by each node only using local information) since it is essentially based upon computing averages. It is well known that this can be done resorting to consensus algorithms; for instance an algorithm based on gossip has been proposed in [20]. The averages in (34) can be computed using standard consensus algorithms.

As expected, if the number of iterations used to compute the averages in (34) is sufficient to reach consensus, this distributed-EM algorithm converges to the centralized-EM solutions. However, as soon as the number of iterations is not sufficient to reach consensus, the distributed-EM algorithm either oscillates or even diverge, failing to provide sensible estimates. This simple simulation experiments suggest that distributed-EM is not robust against errors in computing the averages in (34) which may result from an insufficient number of consensus iterations. As such, deciding how many iterations are ‘‘enough’’ is a delicate matter. We have instead followed a different route, which is based on the algorithm discussed in Section VII.

VI. GENERALIZATION

One drawback of the model in (1) is that the T_i 's are assumed to belong to a known alphabet \mathcal{A} . In particular in this paper we have considered the case $T_i \in \{0, 1\}$. A simple yet important generalization is to allow that the alphabet is partially unknown. For instance one can assume that only the cardinality of \mathcal{A} is known. In the binary case considered in this paper this is equivalent to assume that

$$y_i = \theta + \alpha T_i + v_i \quad (35)$$

with $T_i \in \{0, 1\}$ and $\alpha \in \mathbb{R}^+$ unknown⁴.

In this more general scenario the maximum likelihood estimator (3) becomes:

⁴It is immediate to show that, for identifiability reasons, only the difference between the two symbols have to be parameterized; in addition this difference can be assumed to be positive modulo permutations.

$$\begin{aligned}
(\hat{\theta}^{ML}, \hat{T}^{ML}, \hat{\alpha}^{ML}) &= \underset{(\theta, T, \alpha)}{\operatorname{argmax}} \mathbb{P}(y|\theta, T, \alpha) \\
&= \underset{(\theta, T, \alpha)}{\operatorname{argmin}} \left[\sum_i \frac{(y_i - \theta - \alpha T_i)^2}{2\sigma^2} \right] \quad (36)
\end{aligned}$$

Solving (36) is considerably more difficult than (3); one possible approach is to utilize an alternating minimization algorithm as follows:

(i) Fix $\alpha := \hat{\alpha}^{(k-1)}$ and solve

$$\hat{T}^{(k)}(\alpha) := \underset{T}{\operatorname{argmin}} \min_{\theta} \left[\sum_i \frac{(y_i - \theta - \alpha T_i)^2}{2\sigma^2} \right] \quad (37)$$

(ii) Fix $T := \hat{T}^{(k)}$ and solve

$$(\hat{\theta}^{(k)}(T), \hat{\alpha}^{(k)}(T)) := \underset{(\theta, \alpha)}{\operatorname{argmin}} \left[\sum_i \frac{(y_i - \theta - \alpha T_i)^2}{2\sigma^2} \right] \quad (38)$$

Problem (37) is analogous to (3) with the only difference that in (3) we assumed $\alpha = 1$. Hence this can be solved as described in Section II-A.

Instead, problem (38) admits a closed form solution as:

$$\hat{\theta}^{(k)} = \frac{\sum_i y_i}{N} - \hat{\alpha}^{(k)} \frac{\sum_i \hat{T}_i^{(k)}}{N} \quad \hat{\alpha}^{(k)} = \frac{\frac{\sum_i \hat{T}_i^{(k)} y_i}{N} - \frac{\sum_i \hat{T}_i^{(k)}}{N} \frac{\sum_i y_i}{N}}{\frac{\sum_i \hat{T}_i^{(k)}}{N} \left(1 - \frac{\sum_i \hat{T}_i^{(k)}}{N} \right)} \quad (39)$$

In Section VIII we shall also report simulation experiments, in which α is not assumed to be known, using the alternating minimization approach above; experimental evidence shows that this alternating minimization algorithm converges in few steps (2 or 3) in all the examples considered. Of course, in the distributed scenario the averages in (39) will have to be computed resorting to consensus algorithms.

As an alternative one could also consider the Bayesian formulation in Section V for the measurement model (35). This is standard estimation problem for a mixture of two Gaussian distributions with unknown means and unknown (but common) variance. An EM algorithm similar to (33), (34) in Section V can be derived (see [5]). Of course, in the distributed setting, when averages are computed using consensus algorithms one encounters the same drawbacks as discussed in Section V. In the simulation experiments we have followed the strategy discussed in Section VII.

VII. DISTRIBUTED SOLUTION OF “ALTERNATING-TYPE” ALGORITHMS.

In this Section we address the problem of implementing alternating algorithms (such as those described in Sections V and VI) in a distributed scenario. For the sake of exposition we abstract from the specific algorithms in Sections V and VI and consider the following algorithm which alternates between the two steps. Assume we have N agents each having assigned a given function $H_i: \mathbb{R}^r \rightarrow \mathbb{R}^r$, $i = 1, \dots, N$ and a given function G . For $k = 1, 2, \dots$ do the following steps:

(i) Given $X^{(k)} \in \mathbb{R}^s$ compute

$$Y_i^{(k)} := H_i(X^{(k)})$$

(ii) Given $Y_i^{(k)} \in \mathbb{R}^r$ compute

$$X^{(k+1)} := G \left(\frac{1}{N} \sum_i Y_i^{(k)} \right)$$

Assume now that, as $k \rightarrow \infty$, this alternating algorithm converges to a fixed point $X^{(\infty)}$, $Y_i^{(\infty)}$. Our purpose is to compute the fixed point of this algorithm (or at least a “good” approximation) by means of distributed computations. Let $X_i^{(k)}$ be the “local copy” at the agent i of the “state” $X^{(k)}$ at iteration k . Ideally one would like that $X_i^{(k)} = X_j^{(k)}$ for all $i, j \in [1, N]$. Let us define

$$Y^{(k)} := \begin{bmatrix} Y_1^{(k)} \\ \vdots \\ Y_N^{(k)} \end{bmatrix}$$

and let $P_k[\cdot]: \mathbb{R}^{rN} \rightarrow \mathbb{R}^{rN}$ denote an “average consensus” operator, i.e. an operator which preserves the “average” of its argument and such that

$$\lim_{k \rightarrow \infty} P_k[P_{k-1}[\dots P_0[Y]\dots]] = \mathbb{1}_N \otimes \left[\frac{1}{N} \sum_i Y_i \right] \quad \forall Y$$

We propose the following algorithm in which we need the auxiliary variables $M_i^{(k)}$. These have the same dimension of $Y_i^{(k)}$ and should represent an approximation of the average values of $Y_i^{(k)}$. The vector $M^{(k)}$ is defined from $M_i^{(k)}$ in a similar way in which the vector $Y^{(k)}$ is defined from $Y_i^{(k)}$.

- 1) initialization $X_i^{(0)} := X^{(0)}$, $Y_i^{(-1)} = Y_i^{(0)} = M_i^{(0)} := H_i(X_i^{(0)}) \quad \forall i \in [1, N]$.
- 2) $\forall k = 0, 1, 2, \dots$ do:
 - (i) $M^{(k+1)} = P_k[M^{(k)} - (Y^{(k-1)} - Y^{(k)})]$
 - (ii) $X_i^{(k+1)} := G \left(M_i^{(k+1)} \right)$
 - (iii) $Y_i^{(k+1)} = \lambda Y_i^{(k)} + (1 - \lambda) H_i(X_i^{(k+1)})$

where λ is a tuning parameter which regulates how fast the update of $Y_i^{(k)}$ is allowed to be. If $\lambda = 1$ no update is performed in (iii) so that also (iv) is constant and therefore, from (ii), $M^{(k)}$ converges to its “average” i.e. to $\mathbb{1}_N \otimes \left[\frac{1}{N} \sum_i Y_i^{(0)} \right]$.

This distributed gossip algorithm, which is inspired by the distributed EM algorithms which are found in [9], has been utilized in the simulation Section VIII to implement in our distributed scenario both the EM-algorithm in Section V as well as the EM and the alternating minimization algorithms found in Section VI.

VIII. SIMULATIONS

In order to compare the algorithm introduced in this paper with more standard EM algorithms (based on gossip iterations, as proposed in Section V, VI), we consider the following setup. In Example 1 (see Fig. 3) we assume $N = 50$ sensors are deployed and connected via a random geometric graph. A typical realization of this graph (utilized for the simulations in Figure 3) is depicted in Figure 2. The sensors measure data according to the model (1) or equivalently according to the model (35) with $\alpha = 1$. We generate data with $\theta = 0$, $\sigma = 0.3$ and assume that T_i are i.i.d. Bernoulli random variables with

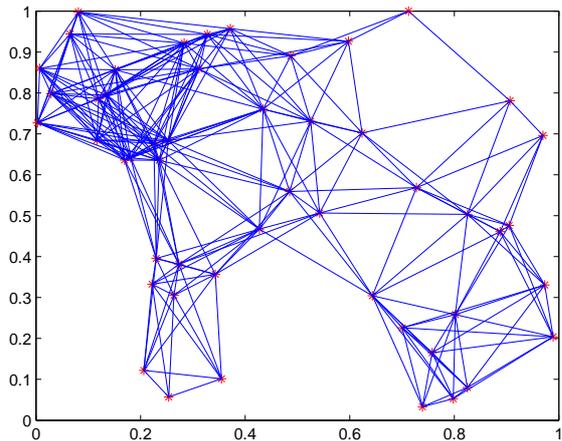


Fig. 2. Connection graph used in Example 1.

mean $p = 0.3$. In order to test the robustness of the algorithms against outliers, in Example 2 we consider a second setup in which data are generated as in Example 1, except for an outlier $y_0 = -2$ which is artificially added.

We compare the following algorithms:

- 1) **Distributed AML** ($\alpha = 1$): this is the distributed approximate Maximum Likelihood described in Section III which is based on the model (1) with $T_i \in \{0, 1\}$ as in Section II.
- 2) **Distributed AML**: the distributed approximate Maximum Likelihood based on model (35), which also estimates α using the alternating maximization approach described in Section VI, with the distributed implementation described in Section VII.
- 3) **EM** ($\alpha = 1$): this is the EM algorithm introduced in Section V with the distributed implementation described in Section VII, based on the measurement model (1) with $T_i \in \{0, 1\}$ as in Section II.
- 4) **EM**: this is the EM algorithm for the estimation of a mixture of two Gaussian distributions with different and unknown means discussed at the end of Section VI, with the distributed implementation described in Section VII.

The simulation results show that there is not a clear-cut distinction between different algorithms; from the limited number of examples we considered, it seems that our ranking-based algorithm is slightly faster than EM. Also the EM algorithm which does not assume α to be known does not seem to converge (nor at least during the first 3000 iterations (see Figure 4). The algorithm introduced in this paper exhibits, overall, a comparable rate of convergence, if not a bit faster than its competitors while being more robust to outliers in the examples considered. The simulation results suggest also that the performance just mildly degrade when α is not assumed to be known. One typical realization of the estimators $\hat{\theta}_i$ and $\hat{\alpha}_i$ (estimators for θ and α at the i -th node) obtained by the distributed AML algorithm are reported in Figure 5.

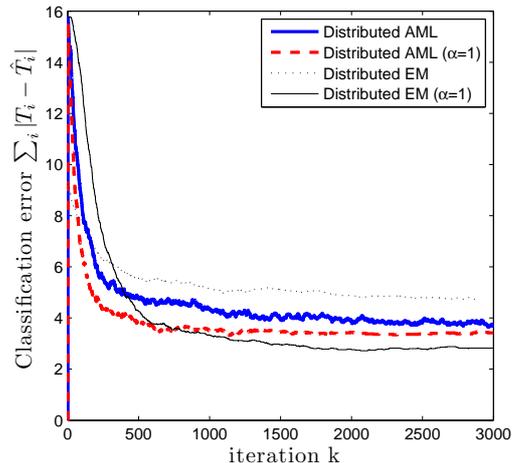


Fig. 3. Example 1: Average (over 50 Monte Carlo runs) of the classification error $\sum_{i=1}^N |T_i - \hat{T}_i|$ as a function of the number of gossip iterations. Data are generated as follows: $\theta = 0$, $T_i \sim \mathcal{B}(0.3)$, $\sigma = 0.3$.

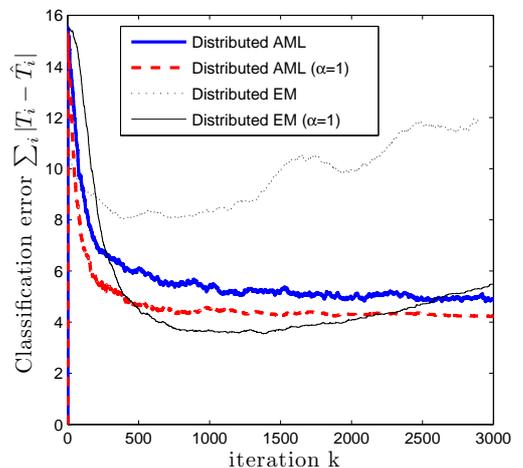


Fig. 4. Example 2 (with outlier): Average (over 50 Monte Carlo runs) of the classification error $\sum_{i=1}^N |T_i - \hat{T}_i|$ as a function of the number of gossip iterations. Data are generated as follows: $\theta = 0$, $T_i \sim \mathcal{B}(0.3)$, $\sigma = 0.3$. An outlier is added to each Monte Carlo realization by setting $y_1 = -2$.

IX. CONCLUSIONS

In this work we studied the problem of distributively computing simultaneous binary classification and noisy parameter estimation in a network of distributed sensors subject to topological communication constraints. We have proposed a fully decentralized approximate ML solution and proved that, when the number of agents N goes to ∞ , such a solution converges, with probability one, to the centralized ML solution. Compared to more classical approaches like EM methods, our algorithm presents similar convergence rates but stronger robustness in various situations, for instance when the offset of the “misbehaving” sensors is not known, or in the presence of outliers.

Different research avenues are possible, such as the generalization to multiple class, the development of more robust

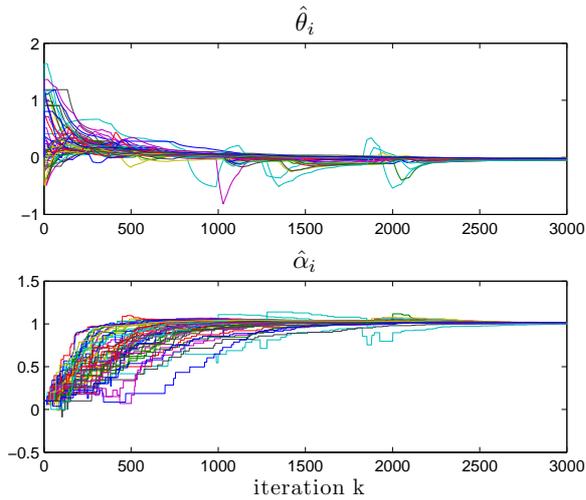


Fig. 5. Distributed AML: estimates $\hat{\theta}_i$ and $\hat{\alpha}_i$ for each node i as a function of the number of gossip iterations.

strategies when the offset is unknown as well as the problem of distributed implementation of alternating-type algorithms.

APPENDIX

Proof of Proposition 4. We start by noting that from Proposition 2 follows that almost surely $\lim_{k \rightarrow \infty} x_i^{(k)} = \frac{1}{N} \sum_{i=1}^N y_i = w(y)$ and $\lim_{k \rightarrow \infty} \eta_i^{(k)} = y_i - w(y)$, while from Theorem 3 we have that almost surely there exists T_1 such that $r k_i^{(k)} = o_i$ for $k \geq T_1$. Without loss of generality, let us now assume that all measurements are distinct, i.e. $y_{[1]} < y_{[2]} < \dots < y_{[N]}$ ⁵ and define

$$\delta = \min_{k=1, \dots, N} \left| 2(y_i - w(y)) - 1 + \frac{2(N - o_i) + 1}{N} \right|$$

From Proposition 2 it also follows that there exists T_2 such that $|\eta_i^{(k)} - (y_i - \bar{y})| < \delta$ for all $k \geq T_2$ and for all i almost surely. This fact and Theorem 3 imply that there exists T such that

$$2\eta_i^{(k)} - 1 + \frac{2(N - r k_i^{(k)}) + 1}{N} > 0 \wedge r k_i^{(k)} > \frac{N}{2}, \quad k \geq T \xLeftrightarrow{a.s.}$$

$$\xLeftrightarrow{a.s.} 2(y_i - w(y)) - 1 + \frac{2(N - o_i) + 1}{N} > 0 \wedge o_i > \frac{N}{2}$$

Therefore, according to (13), this implies that $\hat{T}_i^{(k)} = \hat{T}_i^{AML}$ holds almost surely for all $k \geq T$.

Note now that

$$\begin{aligned} \sum_{i=1}^N w_i^{(k)} &= \sum_{i=1}^N w_i^{(k-1)} + \sum_{i=1}^N (\hat{T}_i^{(k)} - \hat{T}_i^{(k-1)}) \\ &= \sum_{i=1}^N w_i^{(0)} + \sum_{i=1}^N (\hat{T}_i^{(k)} - \hat{T}_i^{(0)}) \\ &= \sum_{i=1}^N \hat{T}_i^{(k)} = \sum_{i=1}^N \hat{T}_i^{AML} = Nw(\hat{T}^{AML}), \quad k \geq T \end{aligned}$$

⁵The theorem holds also for $y_{[1]} \leq y_{[2]} \leq \dots \leq y_{[N]}$ but the proof is slightly more tedious since the ranking might not be unique

where we used the fact that $w_i(0) = \hat{T}_i(0) = 0, \forall i$ and the last equality follows from Equation (17) almost surely for some T . Since for $k \geq T$ the difference $\hat{T}_i^{(k)} - \hat{T}_i^{(k-1)} = 0$, then Proposition 2 implies that $\lim_{k \rightarrow \infty} w_i^{(k)} = w(\hat{T}^{AML})$ almost surely, and consequently also (18). \blacksquare

Proof of Proposition 5. From equation (24) it suffices to show that

$$\lim_{\sigma \rightarrow 0} \int_{1-\omega}^1 F_{\xi}^{-1}(t) dt = \mu(\omega)$$

uniformly for $\omega \in [0, 1]$, where

$$\mu(\omega) := p + (\omega - p)\delta_{-1}(p - \omega) = \begin{cases} \omega & \omega \leq p \\ p & \omega > p \end{cases}$$

Through a change of variable it is easy to verify that

$$\int_{1-\omega}^1 F_{\xi}^{-1}(t) dt = \int_{F_{\xi}^{-1}(1-\omega)}^{\infty} t f_{\xi}(t) dt$$

Denote now

$$z_{\sigma}(\omega) := \frac{F_{\xi}^{-1}(1-\omega) - 1}{\sigma}.$$

Let us define $f_{\xi}(t) := \frac{dF_{\xi}(t)}{dt}$ and $\phi_{\sigma}(t) := \frac{d\Phi_{\sigma}(t)}{dt}$ so that

$$f_{\xi}(t) = (1-p)\phi_{\sigma}(t) + p\phi_{\sigma}(t-1). \quad (\text{A.40})$$

For simplicity we shall also use $\phi(\cdot) := \phi_1(\cdot)$ and $\Phi(\cdot) := \Phi_1(\cdot)$. Using (A.40) and suitable change of variables it follows that

$$\begin{aligned} \int_{1-\omega}^1 F_{\xi}^{-1}(t) dt &= (1-p) \int_{F_{\xi}^{-1}(1-\omega)}^{\infty} t \phi\left(\frac{t}{\sigma}\right) dt + p \int_{F_{\xi}^{-1}(1-\omega)}^{\infty} t \phi\left(\frac{t-1}{\sigma}\right) dt = \\ &= \sigma(1-p) \int_{z_{\sigma}(\omega)+1/\sigma}^{\infty} x \phi(x) dx + \sigma p \int_{z_{\sigma}(\omega)}^{\infty} x \phi(x) dx + \\ &\quad + p \int_{z_{\sigma}(\omega)}^{\infty} \phi(x) dx \end{aligned}$$

and so

$$\begin{aligned} \left| \mu(\omega) - \int_{1-\omega}^1 F_{\xi}^{-1}(t) dt \right| &\leq \\ &\leq \sigma(1-p) \left| \int_{z_{\sigma}(\omega)+1/\sigma}^{\infty} x \phi(x) dx \right| + \sigma p \left| \int_{z_{\sigma}(\omega)}^{\infty} x \phi(x) dx \right| + \\ &\quad + \left| \mu(\omega) - p \int_{z_{\sigma}(\omega)}^{\infty} \phi(x) dx \right| \leq \\ &\leq \sigma(1-p) \int_{-\infty}^{\infty} |x| \phi(x) dx + \sigma p \int_{-\infty}^{\infty} |x| \phi(x) dx + \\ &\quad + |\mu(\omega) - p[1 - \Phi(z_{\sigma}(\omega))]| \end{aligned}$$

Since the first two elements of the sum do not depend on ω and they converge to zero as σ tends to zero. It remains to be proved that the third element of the sum converges uniformly to zero in ω as σ tends to zero.

Let

$$G_{\sigma}(\omega) := \mu(\omega) - p[1 - \Phi(z_{\sigma}(\omega))]$$

Notice that $z_\sigma(\omega)$ is decreasing in ω and that $\omega = 1 - F_\xi(\sigma z_\sigma(\omega) + 1)$. Now, if $\omega \leq p$, then

$$\begin{aligned} G_\sigma(\omega) &= \omega - p[1 - \Phi(z_\sigma(\omega))] = \\ &= 1 - F_\xi(\sigma z_\sigma(\omega) + 1) - p[1 - \Phi(z_\sigma(\omega))] = \\ &= 1 - (1-p)\Phi(z_\sigma(\omega) + 1/\sigma) - p\Phi(z_\sigma(\omega)) \\ &\quad - p + p\Phi(z_\sigma(\omega)) = \\ &= (1-p)[1 - \Phi(z_\sigma(\omega) + 1/\sigma)] \end{aligned}$$

which is a positive and increasing function of ω . If instead $\omega \geq p$, then

$$G_\sigma(\omega) = p - p[1 - \Phi(z_\sigma(\omega))] = p\Phi(z_\sigma(\omega))$$

which is a positive and decreasing function of ω . We can argue that

$$|\mu(\omega) - p[1 - \Phi(z_\sigma(\omega))]| = G_\sigma(\omega) \leq G_\sigma(p)$$

and so, in order to prove the uniform convergence to zero of the left hand side, it is enough to prove that $G_\sigma(p)$ converges to zero as σ converges to zero.

Notice finally that, from the previous arguments we have that $G_\sigma(p) = p\Phi(z_\sigma(p))$. To prove that $\lim_{\sigma \rightarrow 0} p\Phi(z_\sigma(p)) = 0$ it is equivalent to prove that $\lim_{\sigma \rightarrow 0} z_\sigma(p) = -\infty$. Assume by contradiction that this is not true. Then there would exist a real constant M and a sequence σ_n converging to zero such that $z_{\sigma_n}(p) \geq M$ for all n . This would imply that $F_\xi^{-1}(1-p) \geq 1 + \sigma_n M$ and so

$$1 - p \geq F_\xi(1 + \sigma_n M) = (1-p)\Phi(M + 1/\sigma_n) + p\Phi(M)$$

Notice that the right hand side converges to $1 - p + p\Phi(M)$ as n tends to infinity. This would imply that $1 - p \geq 1 - p + p\Phi(M)$ which yields a contradiction.

The fact that $\lim_{\sigma \rightarrow 0} \hat{\omega}(\sigma) = p$ follows from the uniform convergence and from the fact that p is the unique minimum of the limit function. ■

Proof of Lemma 7. Fix j and put $t = F_\xi^{-1}(j/N) - \delta$ so that

$$j = NF_\xi(t + \delta) = Np\Phi_\sigma(t + \delta - 1) + (1-p)\Phi_\sigma(t + \delta)$$

Using (22) we obtain that

$$\begin{aligned} \mathbb{P}(\xi_{[j]} \leq F_\xi^{-1}(j/N) - \delta) &= \mathbb{P}(\Lambda_t \geq j) \\ &= \mathbb{P}(\Lambda_t^1 + \Lambda_t^0 \geq Np\Phi_\sigma(t + \delta - 1) + N(1-p)\Phi_\sigma(t + \delta)) \\ &\leq \mathbb{P}(\Lambda_t^1 \geq Np\Phi_\sigma(t + \delta - 1)) + \mathbb{P}(\Lambda_t^0 \geq (1-p)\Phi_\sigma(t + \delta)) \end{aligned} \quad (\text{A.41})$$

Theorem 6 yields

$$\begin{aligned} \mathbb{P}(\Lambda_t^1 \geq Np\Phi_\sigma(t + \delta - 1)) \\ \leq \exp\left[-|I^1|\Phi_\sigma(t - 1)\gamma\left(\frac{Np\Phi_\sigma(t + \delta - 1)}{|I^1|\Phi_\sigma(t - 1)}\right)\right] \end{aligned} \quad (\text{A.42})$$

Notice that, for $j \in [aN, bN]$, t remains bounded, as well $F_\sigma(t - 1)$. In particular, this implies that, if N is sufficiently large, the argument of γ is above a constant $y_0 > 1$ for all $j \in [aN, aN]$. Therefore, we can find a positive constant C such that

$$\mathbb{P}(\Lambda_t^1 \geq Np\Phi_\sigma(t + \delta - 1)) \leq e^{-CN}$$

Arguing similarly for the other addend in (A.41) and for the analogous term $\mathbb{P}(\xi_{[j]} \geq F_\xi^{-1}(j/N) + \delta)$, we obtain the thesis. ■

Proof of Lemma 8. Arguing like in (A.41), we can estimate

$$\mathbb{P}\left(\xi_{[j]} \leq -(N/j)^{1/2}\right) \leq \mathbb{P}(\Lambda_t^1 > pj) + \mathbb{P}(\Lambda_t^0 > (1-p)j) \quad (\text{A.43})$$

where $t = -(N/j)^{1/2}$. To estimate the first addend, consider

$$y = \frac{pj}{\mathbb{E}(\Lambda_t^1)} = \frac{pN}{|I^1|t^2\Phi_\sigma(t-1)}$$

If $j \in [1, aN]$, we have that $t \leq -(1/a)^{1/2}$. Hence, if a is chosen sufficiently small, we can ensure that $y \geq e^{C_1 t^2}$ for some $C_1 > 0$. Hence, using the Remark after Theorem 6, $\gamma(y) \geq C_2 y \log y \geq C_3 y t^2$ for suitable $C_2, C_3 > 0$. Using Theorem 6, we then obtain

$$\mathbb{P}(\Lambda_t^1 > pj) \leq e^{-C_2 p j t^2} = e^{-C_3 p N}$$

Arguing in a similar way for the second addend in (A.43), we obtain the first estimation in (25).

The second estimation in (25) follows from

$$\mathbb{P}\left(\xi_{[N-j]} \geq (N/j)^{1/2}\right) = \mathbb{P}(\Lambda_t < N - j) = \mathbb{P}(N - \Lambda_t > j)$$

repeating the same arguments above with the binomial r.v $N - \Lambda_t$. This yields the second estimation. Finally, the third one can also be obtained along the same lines of reasoning. ■

Proof of Theorem 9. We can estimate, for $w \in [1, N]$,

$$\begin{aligned} \left| \frac{F_N(w)}{N} - \mathcal{F}\left(\frac{w}{N}\right) \right| \\ \leq 2|\Omega + p| + 2\frac{1}{N}\xi_{[N]} + 2\frac{1}{N}\sum_{j=N-w+1}^{N-1} \left| \xi_{[j]} - F_\xi^{-1}\left(\frac{j}{N}\right) \right| \\ + 2\left| \frac{1}{N}\sum_{j=N-w+1}^{N-1} F_\xi^{-1}\left(\frac{j}{N}\right) - \int_{1-w/N}^1 F_\xi^{-1}(t) dt \right| \end{aligned} \quad (\text{A.44})$$

Let us start with the last deterministic addend. Standard calculus shows that there exists a sequence A_N converging to 0, such that, for every $w \in \{1, \dots, N\}$,

$$\left| \frac{1}{N}\sum_{j=N-w+1}^N F_\xi^{-1}\left(\frac{j}{N}\right) - \int_{1-w/N}^1 F_\xi^{-1}(t) dt \right| \leq A_N \quad (\text{A.45})$$

The second term can be decomposed as follows

$$\begin{aligned} \frac{1}{N}\sum_{j=N-w+1}^{N-1} \left| \xi_{[j]} - F_\xi^{-1}\left(\frac{j}{N}\right) \right| \\ \leq \frac{1}{N}\sum_{j=aN}^{bN} \left| \xi_{[j]} - F_\xi^{-1}\left(\frac{j}{N}\right) \right| + \frac{1}{N}\sum_{j=1}^{aN} |\xi_{[j]}| + \frac{1}{N}\sum_{j=bN}^{N-1} |\xi_{[j]}| \\ + \frac{1}{N}\sum_{j=\beta N}^{N-1} |F_\xi^{-1}(j/N)| + \frac{1}{N}\sum_{j=1}^{aN} |F_\xi^{-1}(j/N)| \end{aligned} \quad (\text{A.46})$$

Fix $\delta > 0$. It immediately follows from (A.45) that

$$\frac{1}{N}\sum_{j=bN}^{N-1} |F_\xi^{-1}(j/N)| + \frac{1}{N}\sum_{j=1}^{aN} |F_\xi^{-1}(j/N)| < \delta/7 \quad (\text{A.47})$$

if a and b are sufficiently close to 0 and 1 respectively.

The remaining terms in (A.46) and in (A.44) are now random variables. In the sequel we will use the notation

C_1, C_2, \dots to denote positive constants and we will implicitly assume that all statements are for N sufficiently large without explicitly specifying it. It follows from Lemma 8 that

$$\mathbb{P}\left(\frac{1}{N} \sum_{j=1}^{aN} \xi_{[j]} \geq -\frac{1}{N} \sum_{j=1}^{aN} (N/j)^{1/2}\right) \geq 1 - e^{-NC_1}$$

Moreover, using Lemma 7 it also follows that

$$\mathbb{P}(\xi_{[j]} < 0, \forall j \in [1, aN]) \geq 1 - e^{-NC_2}$$

Hence, with probability at least $1 - e^{-NC_3}$ it holds that

$$\frac{1}{N} \sum_{j=1}^{aN} |\xi_{[j]}| \leq \frac{1}{N} \sum_{j=1}^{aN} (N/j)^{1/2} \leq \int_0^a x^{-1/2} dx$$

This last integral converges to 0 for $a \rightarrow 0+$, hence, we can choose $a > 0$ sufficiently small in such a way that

$$\mathbb{P}\left(\frac{1}{N} \sum_{j=1}^{aN} |\xi_{[j]}| \leq \delta/7\right) \geq 1 - e^{-NC_3} \quad (\text{A.48})$$

Similarly, we can choose $b < 1$ in such a way that

$$\mathbb{P}\left(\frac{1}{N} \sum_{j=bN}^{N-1} |\xi_{[j]}| \leq \delta/7\right) \geq 1 - e^{-NC_4} \quad (\text{A.49})$$

We now assume that a and b have been fixed in such a way that (A.47), (A.48), and (A.49) hold true. In correspondence of such a and b , it follows from Lemma 7 that

$$\mathbb{P}\left(\frac{1}{N} \sum_{j=aN}^{bN} \left| \xi_{[j]} - F_{\xi}^{-1}\left(\frac{j}{N}\right) \right| \leq \delta/7\right) \geq 1 - e^{-NC_5} \quad (\text{A.50})$$

Finally, we take care of the first two terms in (A.44). Lemmas 7 and 8 show that

$$\mathbb{P}\left(0 < \frac{1}{N} \xi_{[M]} < \frac{1}{\sqrt{N}}\right) \geq 1 - e^{-NC_6}$$

Hence,

$$\mathbb{P}\left(\frac{1}{N} |\xi_{[M]}| \leq \delta/7\right) \geq 1 - e^{-NC_6} \quad (\text{A.51})$$

By the definition of Ω (see (20)), it follows immediately that

$$\mathbb{P}(|\Omega + p| \leq \delta/7) \geq 1 - e^{-NC_7} \quad (\text{A.52})$$

Using estimations (A.45), (A.47), (A.48), (A.49), (A.50), (A.51), and (A.52) inside (A.46) and (A.44) we obtain the thesis. ■

Proof of Theorem 10. We can estimate

$$\left| \Delta(w) - \mathcal{F}'\left(\frac{w}{N}\right) \right| \leq \frac{1}{N} + 2|\Omega + p_0| + 2 \left| \xi_{[N-w]} - F_{\xi}^{-1}\left(1 - \frac{w}{N}\right) \right|$$

We then conclude with arguments similar to those used in the proof of Theorem 9. ■

Proof of Corollary 12. The first statement is an immediate consequence of Proposition 11 applied to the subintervals, respectively, $[0, \bar{\omega} - \delta]$ and $[\bar{\omega} + \delta, 1]$.

Concerning the second point, notice that by (a) we have that

$$\min_{\omega \in [1/2, 1]} \mathcal{F}(\omega) - \min_{\omega \in [0, 1/2]} \mathcal{F}(\omega) = \bar{\delta} > 0$$

Applying now Theorem 9 with $\delta = \bar{\delta}/3$, we obtain the thesis. ■

REFERENCES

- [1] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *Automatic Control, IEEE Transactions on*, vol. 49, no. 9, pp. 1520–1533, Sept. 2004.
- [2] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of IEEE*, vol. 95, no. 1, pp. 215–233, 2007.
- [3] R. Olfati-Saber, "Distributed Kalman filter with embedded consensus filters," *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC '05. 44th IEEE Conference on*, pp. 8179–8184, Dec. 2005.
- [4] R. Carli, A. Chiuso, L. Schenato, and S. Zampieri, "Distributed Kalman filtering based on consensus strategies," *IEEE Journal on Selected Areas in Communications*, vol. 26, pp. 622–633, 2008.
- [5] D. Titterton, A. Smith, and U. Makov, *Statistical Analysis of Finite Mixture Distributions*. John Wiley and Sons, 1985.
- [6] R. Duda, P. Hart, and D. Stork, *Pattern classification*. Wiley Interscience, 2001.
- [7] P. Berkhin, *Grouping Multidimensional Data: Recent Advances in Clustering*, ser. Lecture Notes in Computer Science. Springer, 2006, vol. 2466/2002, ch. A Survey of Clustering Data Mining Techniques, pp. 25–71.
- [8] M. Rabbat and R. Nowak, "Distributed optimization in sensor networks," in *IPSN '04: Proceedings of the 3rd international symposium on Information processing in sensor networks*. ACM, 2004, pp. 20–27.
- [9] R. Nowak, "Distributed EM algorithms for density estimation and clustering in sensor networks," *IEEE Trans. on Signal Processing*, vol. 51, no. 8, pp. 2245–1208, 2003.
- [10] B. Safarnejadian, M. B. Menhaj, and M. Karrari, "Distributed variational Bayesian algorithms for Gaussian mixtures in sensor networks," *Signal Process.*, vol. 90, no. 4, pp. 1197–1208, 2010.
- [11] S. Bandyopadhyay, C. Giannella, U. Maulik, H. Kargupta, K. Liu, and S. Datta, "Clustering distributed data streams in peer-to-peer environments," *Information Sciences*, vol. 176, no. 14, pp. 1952 – 1985, 2006.
- [12] A. Chiuso, F. Fagnani, L. Schenato, and S. Zampieri, "Simultaneous distributed estimation and classification in sensor networks," in *Proceedings of IFAC NECSYS*, 2010.
- [13] R. O. Saber, J. Fax, and R. Murray, "Consensus and cooperation in multi-agent networked systems," *Proceedings of IEEE*, vol. 95, no. 1, pp. 215–233, January 2007.
- [14] F. Garin and L. Schenato, *Networked Control Systems*, ser. Springer Lecture Notes in Control and Information Sciences. Springer, ch. Distributed estimation and control applications using linear consensus algorithms(to appear). [Online] Available at http://automatica.dei.unipd.it/tl_files/utenti/lucaschenato/Papers/Others/WIDEbook_GarinSchenato.pdf.
- [15] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms," *IEEE Transactions on Information Theory/ACM Transactions on Networking*, vol. 52, no. 6, pp. 2508–2530, June 2006.
- [16] F. Fagnani and S. Zampieri, "Randomized consensus algorithms over large scale networks," *Selected Areas in Communications, IEEE Journal on*, vol. 26, no. 4, pp. 634–649, May 2008.
- [17] A. Chiuso, F. Fagnani, L. Schenato, and S. Zampieri, "Gossip algorithms for distributed ranking," Univ. of Padova, Tech. Rep., 2010, available at <http://automatica.dei.unipd.it/publications.html>.
- [18] S. Zacks, *The Theory of Statistical Inference*, ser. Wiley Series in Probability and Mathematical Statistics. Wiley, 1971.
- [19] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 39, no. 1, pp. 1–38, 1977.
- [20] W. Kowalczyk and N. Vlassis, "Newcast EM," in *Advances in Neural Information Processing Systems*. MIT Press, 2005, pp. 713–720.