Gossip algorithms for distributed ranking

Alessandro Chiuso, Fabio Fagnani, Luca Schenato, Sandro Zampieri

***Abstract*— Ranking a set of data plays a key role in many application areas such as signal processing, statistics, computer science and so on. Distributed algorithms for ranking have been proposed in the computer science literature first for tree networks. Extension to general networks has been performed by constructing a spanning tree, which can be done in a distributed manner. In this paper we propose and analyze a gossip algorithm for distributed ranking. The advantage of the proposed algorithm is, on the one hand, its inherent robustness to changes and/or failures in the network and on the other its simplicity of implementation. The algorithm is proved to converge, in the sense of giving the correct ranking, with probability one in finite time.**

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 a set of real numbers. The average is the building block for many estimation methods, so that the average consensus has been proposed as a possible way to obtain distributed estimation algorithms and, in particular, to perform distributed Kalman filtering [3], [4].

It is widely recognized that the plain average is not a robust statistic [5](e.g. for the estimation of a location parameter) in the presence of possible outliers in the data [6]; robust estimators (such as *winsorized means*, see [6] page 75 for a definition) require order statistics [7], since they are based on weighting the data depending of their ranking. We recall here that given a set of N symbols $\{y_1,..,y_N\}$ belonging to a totally ordered set, ranking means attaching to each y_i an integer o_i which is the position of y_i in the ordered list (e.g. $o_k = 1$ if y_k is the smallest of the y_i's).

Besides the applications in robust estimation [8], [6], [5], order statistics are fundamental tools in a variety of estimation and classification problems in rather diverse areas, such as signal processing ([9] and references therein), in modeling extremal events e.g. in financial risk management and actuarial sciences [10] and so on.

Our main motivation for studying this problem comes indeed from a very specific distributed estimation and classification problem in sensor networks [11], [12]; in these papers it is shown that maximum likelihood estimators can be easily found as a function of ranked observations. Hence, solving the problem of distributed ranking is instrumental for the (distributed) solution of the problem in [11], [12].

Besides the few applications mentioned above, ranking is a fundamental task in computer science applications, and it has been widely studied in the past years. Even the problem of distributed ranking as a long history in the computer science community which goes back to the eighties, see for instance [13]. Most of the literature addresses the problem of distributed ranking in tree-networks; solving the problem in general (connected) networks requires finding a spanning tree, which can be done in a distributed manner [14]. Robust version of these algorithms (so-called "self-stabilizing" algorithms) have been also studied, see e.g. [15].

In this paper we take instead a different approach which has, essentially, the same motivations as gossip algorithms for (average) consensus. The algorithm has to be simple, robust to node failures and changes in the network topology. Hence, we propose a gossip based algorithm for distributed ranking in a general (connected) network. It is shown that this algorithm solves (almost surely) in finite time the ranking problem. The structure of the paper is as follows: Section II states the problem and sets up notation. Section III describes the ranking algorithm while in Section V some simulation results are presented. Conclusions end the paper. The proof of the main result in given in the Appendix.

II. Problem formulation and notation

The problem we consider in this paper can be formalized as follows: Consider a set $\mathscr{N}$ of labeled *agents* $\mathscr{N} := \{1,..,N\}$ which can be thought of as the vertices (or nodes) of a graph $\mathscr{G} := (\mathscr{N},\mathscr{E})$, where $\mathscr{E}$ is the set of *edges* which encode the communication links, i.e. nodes i and j can communicate iff $(i,j) \in \mathscr{E}$. We assume that each agent has an observation $y_i \in \mathscr{Y}$ where $\mathscr{Y}$ is a totally ordered set. Without loss of generality we shall assume $\mathscr{Y} = \mathbb{R}$.

Our purpose is to rank the y_i's, i.e. we would like each agent i is able to compute its ordered position o_i. E.g. if the observation y_i of agent i is the smallest[1] among all observations (i.e. $y_i < y_j$, $\forall j \neq i$) then $o_i = 1$; if y_k is the

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,zampi}@dei.unipd.it

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

second smallest $o_k = 2$ and so on. The map $o_\cdot : \mathcal{N} \to \mathcal{N}$ is of course a permutation of the set $\mathcal{N} = \{1,..,N\}$. Using a notation which is rather common in the statistics literature, we can define $[\cdot] : \mathcal{N} \to \mathcal{N}$ as the inverse permutation w.r.t $o_\cdot$, i.e. $[o_i] = i$, which implies that

$$y_i = y_{[o_i]}.$$

III. Decentralized ranking computation

In this section we introduce a novel distributed algorithm that can rank the nodes of a network based on the ordered list of the magnitudes of their measurements. The algorithm is randomized in the same spirit of randomized gossip average consensus [16], [17], 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 1 and illustrated in Figures 1 and 2. The algorithm works as follows.

- **Initialization** (lines 1-3 of the algorithm): In the initialization phase each node sets a number of variables as shown in Figure 1. The variable rk^{loc} is the *local* estimate that the nodes i has about its own ranking, which is initialized with the node ID i. Each node also creates a copy of its information, i.e. node ID id^{wnd}, measurement y^{wnd} and ranking rk^{wnd}, to be stored on a virtual node that will be exchanged with neighboring nodes thus creating *wandering* virtual nodes. The variables relative to the wandering nodes are distinguished by the subscript *wnd*.
- At each time step k, an edge (i, j) is selected and the corresponding nodes perform the operations illustrated in Figure 2:
  - **wnd ordering** (lines 6-10 of the algorithm): The nodes check if the ordering of the ranking rk^{wnd} in the wandering nodes is consistent with the corresponding measurements y^{wnd}. If ordering is not consistent, then the nodes exchange all their ranking rk^{wnd}, while if ordering is consistent, then the nodes keep their rk^{wnd}.
  - **loc ordering** (lines 11-15 of the algorithm): The nodes check if the ordering of the ranking rk^{loc} is consistent with the corresponding local measurements y. If ordering is not consistent, then the nodes exchange the values of rk^{loc}, while if ordering is consistent, then the nodes keep their rk^{loc}.
  - **swapping** (lines 16-19 of the algorithm): The nodes swap all their wandering variables y^{wnd}, rk^{wnd} and id^{wnd}.
  - **Overwriting** (lines 20-25 of the algorithm): Finally (iii) both the nodes check if the wandering nodes has the same *id* of the node, and in this case it overwrites the ranking rk^{wnd} from the wandering node to the local estimate rk^{loc}
  - **Remaining nodes actions** (lines 26-31 of the algorithm): All the other nodes do not perform any update.

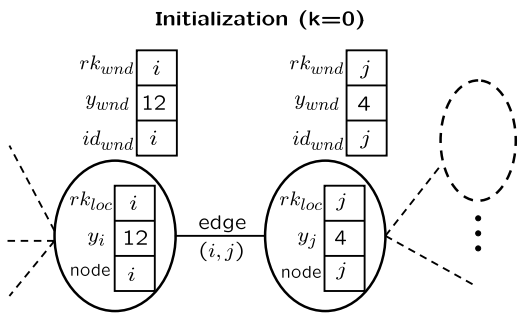

Fig. 1. Initialization of variables of Algorithm 2.

Algorithm 1 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: $rk_i^{loc}(0) = i$, $id_i^{wnd}(0) = i$, $rk_i^{wnd}(0) = i$, $y_i^{wnd}(0) = y_i$ $k = 0$.
3: **end for**
4: randomly select edge $(i, j) \in \mathcal{E}$ with $\mathbb{P}[(i, j)] = p_{ij}$
5: **repeat**
6: **if** $(y_i^{wnd}(k) - y_j^{wnd}(k))(rk_i^{wnd}(k) - rk_j^{wnd}(k)) < 0$ **then**
7: $rk_i^{wnd}(k+1) = rk_j^{wnd}(k)$, $rk_j^{wnd}(k+1) = rk_i^{wnd}(k)$
8: **else**
9: $rk_i^{wnd}(k+1) = rk_i^{wnd}(k)$, $rk_j^{wnd}(k+1) = rk_j^{wnd}(k)$
10: **end if**
11: **if** $(y_i - y_j)(rk_i^{loc}(k) - rk_j^{loc}(k)) < 0$ **then**
12: $rk_i^{loc}(k+1) = rk_j^{loc}(k)$, $rk_j^{loc}(k+1) = rk_i^{loc}(k)$
13: **else**
14: $rk_i^{loc}(k+1) = rk_i^{loc}(k)$, $rk_j^{loc}(k+1) = rk_j^{loc}(k)$
15: **end if**
16: $id_i^{wnd}(k+1) = id_j^{wnd}(k)$, $id_j^{wnd}(k+1) = id_i^{wnd}(k)$
17: $y_i^{wnd}(k+1) = y_j^{wnd}(k)$, $y_j^{wnd}(k+1) = y_i^{wnd}(k)$
18: $temp = rk_i^{wnd}(k+1)$
19: $rk_i^{wnd}(k+1) = rk_j^{wnd}(k+1)$, $rk_j^{wnd}(k+1) = temp$
20: **if** $id_i^{wnd}(k+1) = i$ **then**
21: $rk_i^{loc}(k+1) = rk_i^{wnd}(k+1)$
22: **end if**
23: **if** $id_j^{wnd}(k+1) = j$ **then**
24: $rk_j^{loc}(k+1) = rk_j^{wnd}(k+1)$
25: **end if**
26: **for all** $\ell = 1, \ldots, N, \ell \neq i, \ell \neq j$ **do**
27: $rk_\ell^{loc}(k+1) = rk_\ell^{loc}(k)$
28: $rk_\ell^{wnd}(k+1) = rk_\ell^{wnd}(k)$
29: $id_\ell^{wnd}(k+1) = id_\ell^{wnd}(k)$
30: $y_\ell^{wnd}(k+1) = y_\ell^{wnd}(k)$
31: **end for**
32: $k = k+1$
33: **until** $k > M$

The swapping step has the property that each wandering virtual node will eventually reach (almost surely) any node in the network (which of course we assume to be connected) since its dynamics is similar to a random walk on the graph.

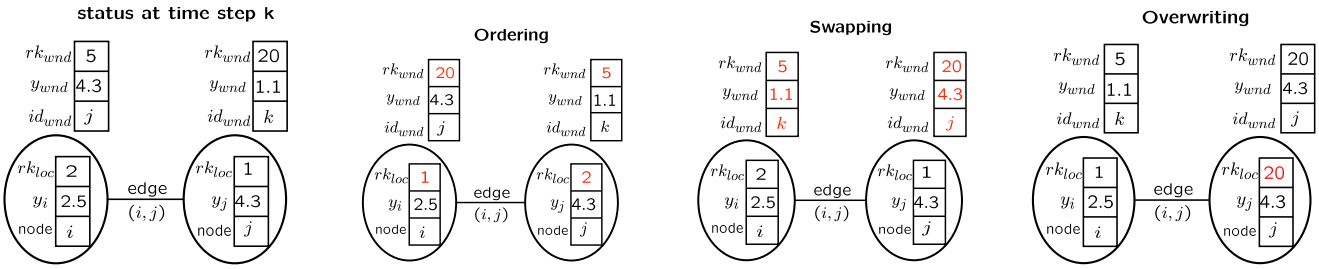

Fig. 2. Graphical representation of Algorithm 2.

Moreover, it also guarantees that eventually (almost surely in finite time) any virtual node pair i, j will meet and will perform ordering. In practice, it is as if the wandering nodes lived on a complete graph where the communication probability is state dependent. As a consequence, any distributed algorithm that works only on complete graphs can be made to work by means of the wandering virtual nodes. Since any pair of virtual nodes eventually meets, this imply that also the node with the largest measurement and the node which stores the largest ranking N will meet, and therefore after the ordering procedure the node with the largest measurement will have the correct ranking and will remain fixed from then on. A similar argument can be repeated for the node with the second largest measurement and the node with the second largest estimated ranking, so that also the node with the second largest measurement will have the correct ranking, and so on until all notes are ordered. As a consequence, eventually all wandering virtual nodes will have the exact ranking and no ordering will occur. As explained above, each wandering virtual node will eventually return to fixed node with the same ID, and therefore eventually also all local fixed nodes will have the correct ranking. This reasoning is formally stated in the following theorem, whose proof is given in the appendix:

Theorem 1 (Randomized Gossip Ranking): Consider Algorithm 1. If the graph $\mathscr{G}$ is connected and $p_{ij} > 0$ for all edges $(i, j) \in \mathscr{E}$ then there exists $T > 0$ such that

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

Before moving to the next section, a few remarks are in order. One might wonder if the wandering virtual nodes are really necessary, which is equivalent to ask whether the swapping procedure is necessary. If this was not performed, i.e. if lines 16-19 were removed from Algorithm 1 and the graph is not complete, it would be possible that the algorithm ended in a deadlock for which no additional update of the ranking is performed but the local ranking has not reached the correct value. The other question is why to perform the ordering also of the local ranking (lines 11-15), since eventually the wandering nodes will overwrite the ordering of the local ranking. In fact, Theorem 1 is valid even if local ordering of ranking (lines 11-15), is removed from Algorithm 1. The reason is that each fixed node has to wait for the corresponding wandering node to come back before performing any ranking update, thus slowing down convergence "rate". In practice, the proposed algorithm tries to take the benefits of local ranking ordering, i.e. fast convergence, and the wandering virtual nodes, i.e. guaranteed convergence to exact ordering. This is illustrated in Figure 3 and discussed in Section V.

IV. Performance: some analytical results

In this section we propose an analytical study of the performance of the previous algorithm in the case of a complete communication graph. The analysis is not easy especially for the presence of the two dynamics, the local one coded inside the variables $rk_i^{loc}(k)$, and the wandering one coded in $rk_i^{wnd}(k)$, and their intertwining. We will not be able to say much about this phenomenon, rather we will concentrate on the variables $rk_i^{wnd}(k)$ and we will estimate their convergence time. Finally, adding the time needed for the wandering variables to go back home where they originated, it will be possible to give an estimation of the total time needed to complete the ranking.

We start from a complete network with N agents and we assume that $p_{ij} = 1/e$ for all (i, j) with $i \neq j$, where $e = N(N-1)$ is the number of directed edges in the graph. We will use the same notation as in the proof of Theorem 1. In particular we put $V_k = \Psi(rk^{wnd}(k), y^{wnd}(k))$ the number of (directed) edges not correctly ordered at the time step k. Clearly, given V_k, the probability to select a correctly ordered edge at time k is exactly $(e - V_k)/e$. If this happens, then $V_{k+1} = V_k$. Otherwise (see the proof of Theorem 1), $V_{k+1} \leq V_k - 1$. Let X_t be a sequence of independent geometric random variables, of parameters, respectively, $(e-t)/e$. Previous considerations show that the average time to bring V_k to 0 is upper bounded by

$$\mathbb{E}[X_0 + X_1 + \cdots + X_{e-1}] = \sum_{t=0}^{e-1} \frac{e}{e-t} \asymp e \ln e \asymp N^2 \ln N \quad \text{for}$$

where the asymptotic equivalence holds for $N \to +\infty$. Regarding the extra time needed for all the wandering variables to go back to their generating nodes we can estimate it, in average, as follows. Starting with the max variable, its wandering is simply governed by a another geometric r.v. Y of parameter $1/N(N-1)$. The average time needed for the wandering variable to go back to its generating node is thus $\mathbb{E}(Y) = N(N-1)$. Similarly all the subsequent nodes. Therefore, a further $\asymp N^3$ time is needed for this final redistribution. The total average time is thus $\asymp N^3$.

When the graph is not complete no such simple estimation can be carried on. A possibility would be to use the mean-field techniques employed in studying spin glass models in statistical mechanics. This will be studied in a future paper.

V. Simulations

We have tested the proposed algorithm on a connected random geometric graph with 50 nodes and, on average, 5 neighbors per node; we have performed 100 Monte Carlo experiments (randomizing the values y_i's). For each realization we have computed the ranking error[2]

$$J^k_{loc} := \Psi(rk^{loc}_i(k), y) \tag{1}$$

and averaged the results over the Monte Carlo runs. The logarithm of the average ranking error is displayed in figure 3. The key role played by the "wandering" structures is clear from the results. In fact without these "wandering" virtual nodes the ranking error has a constant (and different from zero) asymptote (red dashed), meaning that the algorithm gets locked into a local minimum. Differently, the algorithm without the local ordering of the ranking eventually converges to the exact ranking, but the transitory is much worse than in the proposed algorithm. The figure also shows the ranking error of the wandering virtual nodes $J^k_{wnd} := \Psi(rk^{wnd}_i(k), y^{wnd})$ i.e. the ranking error that we would have if the wandering nodes could instantaneously communicate with their corresponding fixed node. The difference between J_{wnd} and J_{loc} highlights the effect of the delay due to the return time of each node to its corresponding fixed node in the proposed algorithm.

In addition the simulation results suggest that, after a transient dominated by the "local" ordering (blue solid/red dash-dotted curves for "small" number of gossip steps), convergence "rate" is governed by how fast the "wandering" virtual nodes travel across the network (blue solid/green dashed).

VI. Conclusions

We have presented a gossip algorithm for distributed ranking in sensor networks. It has been proved that, almost surely, the algorithm correctly ranks the nodes in finite time. Some simulation results show the performance of the algorithm on a connected network with 50 nodes.

Future work will focus on a more detailed analysis of the algorithm in particular w.r.t. to the relation between the network properties and its convergence "rate".

Appendix

Let $z \in \mathbb{R}^N$ and $x = (x_1, \ldots, x_N)$ be a permutation of $\mathcal{N}$ ($x_i \in \mathcal{N}$, $x_i \neq x_j$ if $i \neq j$). Define the following function

$$\Psi(x,z) := |\{(i,j) \in \mathcal{N} \times \mathcal{N} \mid (x_i - x_j)(z_i - z_j) < 0\}| \tag{A.2}$$

Notice that

$$\Psi(x,z) = \sum_{i=1}^{N} \sum_{j=1}^{N} \chi[-(x_i - x_j)(z_i - z_j)]$$

[2]The function Ψ is defined in (A.2).

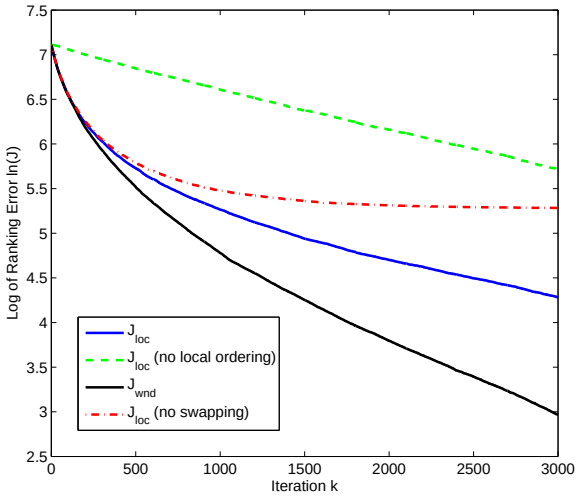

Fig. 3. Average (over 100 Monte Carlo runs) of the ranking error of Algorithm 2 in a random geometric graph with 50 nodes and 250 edges.

where $\chi : \mathbb{R} \to \mathbb{R}$ is defined as follows

$$\chi(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{if } x \leq 0 \end{cases}$$

The function Ψ will be useful to prove Theorem 1. In particular two properties of this function will be crucial.

Assume that $\Pi \in \{0,1\}^{N \times N}$ is a permutation matrix. Then it is easy to see that

$$\Psi(\Pi x, \Pi z) = \Psi(x,z) \tag{A.3}$$

The second property is more difficult to prove and it will be given through the following lemma.

Lemma 2: Let $\Pi_{ij} \in \{0,1\}^{N \times N}$ is the permutation matrix which exchange the entry i with the entry j and assume that $z \in \mathbb{R}^N$ and x is a permutation such that $(x_i - x_j)(z_i - z_j) < 0$. Then

$$\Psi(x, \Pi_{ij} z) < \Psi(x,z) \tag{A.4}$$

Proof We need compute $\Psi(x,z) - \Psi(x, \Pi_{ij} z)$. Notice indeed that

$$\Psi(x,z) - \Psi(x, \Pi_{ij} z) = \sum_{r,s} \chi[(x_r - x_s)(z_s - z_r)] - \chi[(x_r - x_s)((\Pi_{ij} z)_s - (\Pi_{ij} z)_r)]$$

We split the sum in various parts separating the cases in

which one or both the indices r,s are equal to i,j

$$\Psi(x,z)-\Psi(x,\Pi_{ij}z)=$$

$$\begin{aligned}
&\sum_{r,s\neq i,j}\chi[(x_r-x_s)(z_s-z_r)]-\chi[(x_r-x_s)((\Pi_{ij}z)_s-(\Pi_{ij}z)_r)]\\
&+\sum_{r\neq i,j}\chi[(x_r-x_i)(z_i-z_r)]-\chi[(x_r-x_i)((\Pi_{ij}z)_i-(\Pi_{ij}z)_r)]\\
&+\sum_{r\neq i,j}\chi[(x_r-x_j)(z_j-z_r)]-\chi[(x_r-x_j)((\Pi_{ij}z)_j-(\Pi_{ij}z)_r)]\\
&+\sum_{s\neq i,j}\chi[(x_i-x_s)(z_s-z_i)]-\chi[(x_i-x_s)((\Pi_{ij}z)_s-(\Pi_{ij}z)_i)]\\
&+\sum_{s\neq i,j}\chi[(x_j-x_s)(z_s-z_j)]-\chi[(x_j-x_s)((\Pi_{ij}z)_s-(\Pi_{ij}z)_j)]\\
&+\chi[(x_i-x_i)(z_i-z_i)]-\chi[(x_i-x_i)((\Pi_{ij}z)_i-(\Pi_{ij}z)_i)]\\
&+\chi[(x_j-x_j)(z_j-z_j)]-\chi[(x_j-x_j)((\Pi_{ij}z)_j-(\Pi_{ij}z)_j)]\\
&+\chi[(x_i-x_j)(z_j-z_i)]-\chi[(x_i-x_j)((\Pi_{ij}z)_j-(\Pi_{ij}z)_i)]\\
&+\chi[(x_j-x_i)(z_i-z_j)]-\chi[(x_j-x_i)((\Pi_{ij}z)_i-(\Pi_{ij}z)_j)]
\end{aligned}$$

Now we use the fact that $(\Pi_{ij}z)_r=z_r$ if $r\neq i,j$ and that $(\Pi_{ij}z)_i=z_j$ and $(\Pi_{ij}z)_j=z_i$.

$$\Psi(x,z)-\Psi(x,\Pi_{ij}z)=$$

$$\begin{aligned}
&\sum_{r\neq i,j}\chi[(x_r-x_i)(z_i-z_r)]-\chi[(x_r-x_i)(z_j-z_r)]\\
&+\sum_{r\neq i,j}\chi[(x_r-x_j)(z_j-z_r)]-\chi[(x_r-x_j)(z_i-z_r)]\\
&+\sum_{s\neq i,j}\chi[(x_i-x_s)(z_s-z_i)]-\chi[(x_i-x_s)(z_s-z_j)]\\
&+\sum_{s\neq i,j}\chi[(x_j-x_s)(z_s-z_j)]-\chi[(x_j-x_s)(z_s-z_i)]\\
&+\chi[(x_i-x_j)(z_j-z_i)]-\chi[(x_i-x_j)(z_i-z_j)]\\
&+\chi[(x_j-x_i)(z_i-z_j)]-\chi[(x_j-x_i)(z_j-z_i)]=\\
&2\sum_{r\neq i,j}\chi[(x_r-x_i)(z_i-z_r)]-\chi[(x_r-x_i)(z_j-z_r)]\\
&+2\sum_{r\neq i,j}\chi[(x_r-x_j)(z_j-z_r)]-\chi[(x_r-x_j)(z_i-z_r)]+2
\end{aligned}$$

Notice finally that, since $\chi[xy]=\chi[x]\chi[y]+\chi[-x]\chi[-y]$, we can argue that

$$\begin{aligned}
&\chi[(x_r-x_i)(z_i-z_r)]-\chi[(x_r-x_i)(z_j-z_r)]+\\
&\chi[(x_r-x_j)(z_j-z_r)]-\chi[(x_r-x_j)(z_i-z_r)]=\\
&(\chi[x_r-x_i]-\chi[x_r-x_j])(\chi[z_r-z_j]-\chi[z_r-z_i])+\\
&(\chi[x_i-x_r]-\chi[x_j-x_r])(\chi[z_j-z_r]-\chi[z_r-z_i])
\end{aligned}\tag{A.5}$$

Assume now that $x_i<x_j$ which implies that $z_i>z_j$. In this case, since $\chi[\cdot]$ is a monotonic non decreasing function, we have that $\chi[x_r-x_i]\geq\chi[x_r-x_j]$, $\chi[z_r-z_j]\geq\chi[z_r-z_i]$, $\chi[x_i-x_r]\leq\chi[x_j-x_r]$ and $\chi[z_j-z_r]\leq\chi[z_r-z_i]$ which implies that the last term of (A.5) is non-negative. We can conclude that

$$\Psi(x,z)-\Psi(x,\Pi_{ij}z)\geq 2$$

The case in which $x_i>x_j$ and $z_i<z_j$ is analogous. ■

Notice that the same proof works if we choose $\chi[x]=x$ or

$$\chi(x)=\begin{cases}x & \text{if } x>0\\ 0 & \text{if } x\leq 0\end{cases}$$

In case we take $\chi[x]=x$ we get

$$\begin{aligned}\Psi(x,z)=\textstyle\sum_{i=1}^N\sum_{j=1}^N-(x_i-x_j)(z_i-z_j)=\\ 2\left(\textstyle\sum_{i=1}^N x_i\right)\left(\textstyle\sum_{i=1}^N z_i\right)-2N\left(\textstyle\sum_{i=1}^N x_iz_i\right)\end{aligned}$$

Therefore we could use in the proof also the cost function

$$\Psi(x,z)=-\sum_{i=1}^N x_iz_i=-x^Tz$$

Notice finally that minimizing $\Psi(x,z)$ is the same as minimizing $\sum_i(z_i-x_i)^2$.

Proof of Theorem 1.

We first show that, with probability 1, there exists a time $K_1>0$ such that

$$rk_i^{wnd}(k)=o_{id_i^{wnd}(k)}\quad\forall k\geq K_1,\ \forall i\in\mathscr{N}\tag{A.6}$$

Suppose now that the edge (i,j) is selected at time k. Then if,

$$(y_i^{wnd}(k)-y_j^{wnd}(k))(rk_i^{wnd,i}(k)-rk_j^{wnd}(k))\geq 0$$

we say that (i,j) is correctly ordered at time k and we clearly have that

$$\Psi(rk^{wnd}(k+1),y^{wnd}(k+1))=\Psi(rk^{wnd}(k),y^{wnd}(k))$$

We call it a gossip step of type I. If instead

$$(y_i^{wnd}(k)-y_j^{wnd}(k))(rk_i^{wnd}(k)-rk_j^{wnd}(k))<0\tag{A.7}$$

we say that (i,j) is not correctly ordered at time k and, by the previous lemma, we have that

$$\Psi(rk^{wnd}(k+1),y^{wnd}(k+1))<\Psi(rk^{wnd}(k),y^{wnd}(k))\tag{A.8}$$

This is instead called a gossip step of type II.

Consider now the Markov chain on the state space

$$H=\{(i,j)\in\mathscr{N}\times\mathscr{N}\,|\,i\neq j\}$$

defined by

$$Q_{(i,j),(h,k)}=\begin{cases}P_{ih}\ \text{if } i\neq h,\ j=k\\ P_{jk}\ \text{if } i=h,\ j\neq k\\ P_{ij}\ \text{if } i=k,\ j=h\\ 1-\sum_{\ell\neq j}P_{i\ell}-\sum_{\ell\neq i}P_{j\ell}-P_{ij}\ \text{if } i=h,\ j=k\\ 0\ \text{otherwise}\end{cases}$$

Notice that Q is irreducible. Suppose now that at a certain time k, we have that $\Psi(rk^{wnd}(k),y^{wnd}(k))\geq 1$. Then, there exists a pair (i,j) which is not correctly ordered at time k. If $(i,j)\in\mathscr{E}$, then, consider any path of some length s, of positive probability according to Q, that starts in (i,j) and ends with the edge $(i,j)\rightarrow(j,i)$. Each step in the the path corresponds to a gossip step in the algorithm. Now there are two possibilities: either there is at least one gossip step of type II in the first $s-1$ steps, or all first $s-1$ steps correspond to gossip steps of type I. In the first case we clearly obtain that $\Psi(rk^{wnd}(k+s-1),y^{wnd}(k+s-1))<\Psi(rk^{wnd}(k),y^{wnd}(k))$. In the second case,

$$\begin{aligned}y_i^{wnd}(k+s-1)=y_i^{wnd}(k),\quad rk_i^{wnd}(k+s-1)=rk_i^{wnd}(k)\\ y_j^{wnd}(k+s-1)=y_j^{wnd}(k),\quad rk_j^{wnd}(k+s-1)=rk_j^{wnd}(k)\end{aligned}$$

Therefore, the gossip step at time $k+s$ is of type II and therefore $\Psi(rk^{wnd}(k+s),y^{wnd}(k+s)) < \Psi(rk^{wnd}(k),y^{wnd}(k))$. Since Q is irreducible, the time to hit edge $(i,j) \to (j,i)$ starting from vertex (i,j) is finite with probability one. Hence, our argument above shows that if $\Psi(rk^{wnd}(k),y^{wnd}(k)) \geq 1$ and there exists $(i,j) \in \mathscr{E}$ not correctly ordered at time k, then, with probability one, there exists s such that

$$\Psi(rk_{wnd}(k+s),y^{wnd}(k+s)) < \Psi(rk^{wnd}(k),y^{wnd}(k)) \quad \text{(A.9)}$$

If instead all edges are correctly ordered at time k, fix any $(i,j) \in \mathscr{N} \times \mathscr{N}$ not correctly ordered at time k. Consider now any path of some length s, of positive probability according to Q, connecting (i,j) to $\mathscr{E}$ and argue like above. If any of the gossip steps corresponding to this walk is of type II, then (A.9) holds true. If instead all gossip steps are of type I, at the end of it we will have an edge $(h,k) \in \mathscr{E}$ not correctly ordered at time $k+s$ and we are back in the case studied above. Therefore, we have proven that if $\Psi(rk^{wnd}(k),y^{wnd}(k)) \geq 1$, then, with probability one, there exists s such that (A.9) holds true. This easily implies that, with probability 1, there exists K_1, such that $\Psi(rk_{wnd}(K_1),y^{wnd}(K_1)) = 0$. This yields (A.6).

Let i be such that $rk_i^{wnd}(K_1) = N$ and put $j = id_i^{wnd}(K_1)$. Using now the irreducible Markov chain on $\mathscr{N}$ defined by

$$R_{i,j} = \begin{cases} P_{ij} \text{ if } i \neq j \\ 1 - \sum_{\ell \neq j} P_{i\ell} \end{cases}$$

and arguing as above, it is straightforward to prove that, with probability 1 there exists $s_1 \in \mathbb{N}$ such that $id_j^{wnd}(K_1 + s_1) = id_i^{wnd}(K_1) = j$. Moreover, since all gossip steps, since time K_1, are of type I, we also have $rk_j^{wnd}(K_1+s_1) = rk_i^{wnd}(K_1) = N$, and (see lines $23-25$ in the algorithm) $rk_j^{loc}(K_1+s_1) = N$. In other words, y_j is the maximum and at time K_1+s_1 agent j has acquired in its local register the right ordered information. Noticing that condition at line 11 of the algorithm will never be true for node j since time K_1+s_1 (because j has got the right highest ordered position), we will have that $rk_j^{loc}(k) = N$ for all $k \geq K_1+s_1$. Now we can repeat the argument with the second highest value: with probability one at a certain time $K_1+s_1+s_2$, we will have that $id_{j'}^{wnd}(K_1+s_1+s_2) = j'$ with $rk_{j'}^{wnd}(K_1+s_1+s_2) = N-1$ and, consequently, also $rk_{j'}^{loc}(K_1+s_1+s_2) = N-1$. Now notice that also in this case condition at line 11 of the algorithm will never be true for node j' since time $K_1+s_1+s_2$ (because agent j' has got the right second highest ordered position, and the agent with the highest value j has previously obtained the right highest ordered position). Hence, $rk_{j'}^{loc}(k) = N$ for all $k \geq K_1+s_1+s_2$. A formal inductive argument along these lines, leads now to the proof of the result. ■

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," *Proocedings 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] P. Huber, *Robust Statistics*. Wiley, New York, 1981,.

[6] ——, "Robust estimation of a location parameter," *The Annals of Mathematical Statistics*, vol. 35, no. 1, pp. 73–101, March 1964.

[7] H. David and H. Nagaraja, *Order Statistics*. Wiley, New Jersey, 2003.

[8] J. Hodges and E. Lehmann, "Estimates of location based on rank tests," *Annals of Mathematical Statistics*, vol. 19, pp. 598–611, 1963.

[9] R. Öten and R. Figuereido, "Adaptive alpha-trimmed meand filters under deviations from assumed noise models," *IEEE Transactions on Signal Processing*, vol. 13, no. 5, 2004.

[10] P. Embrechts, C. Klüppelberg, and T. Mikosch, *Modeling Extremal Events*. Springer, New York, 1997.

[11] A. Chiuso, F. Fagnani, L. Schenato, and S. Zampieri, "Simultaneous distributed estimation and classification in sensor networks," in *Proceedings of IFAC NECSYS*, 2010.

[12] ——, "Gossip algorithms for simultaneous distributed estimation and classification in sensor networks," Univ. of Padova, Tech. Rep., 2010, to appear in http://paduaresearch.cab.unipd.it/.

[13] S. Zaks, "Optimal distributed algorithms for sorting and ranking," *IEEE Transactions on computers*, vol. 5, no. 1, 1985.

[14] R. Gallager, P. Humbler, and P. Spira, "A distributed algorithm for minimum-weight spanning trees," *ACM Transactions on Programming Languages and Systems*, vol. 16, pp. 1–15, April 1983.

[15] A. Datta and S. Tixeuil, "Self-stabilizing distributed sorting in tree networks," *International Journal of Parallel, Emergent and Distributed Systems*, vol. 16, pp. 1–15, April 2001.

[16] 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.

[17] 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.