A Scheme for Robust Distributed Sensor Fusion Based on Average Consensus

Lin Xiao
Center for the Mathematics of Information
California Institute of Technology
Pasadena, CA 91125-9300
Email: lxiao@caltech.edu

Stephen Boyd
Department of Electrical Engineering
Stanford University
Stanford, CA 94305-9510
Email: boyd@stanford.edu

Sanjay Lall
Department of Aeronautics and Astronautics
Stanford University
Stanford, CA 94305-4035
Email: lall@stanford.edu

***Abstract*— We consider a network of distributed sensors, where each sensor takes a linear measurement of some unknown parameters, corrupted by independent Gaussian noises. We propose a simple distributed iterative scheme, based on distributed average consensus in the network, to compute the maximum-likelihood estimate of the parameters. This scheme doesn't involve explicit point-to-point message passing or routing; instead, it diffuses information across the network by updating each node's data with a weighted average of its neighbors' data (they maintain the same data structure). At each step, every node can compute a local weighted least-squares estimate, which converges to the global maximum-likelihood solution. This scheme is robust to unreliable communication links. We show that it works in a network with dynamically changing topology, provided that the infinitely occurring communication graphs are jointly connected.**

I. Introduction

Sensor networks have received significant attention in recent years because of their huge potential in applications, and the considerable technical challenges they present in communication, signal processing, routing and sensor management, and many other areas (see, *e.g.*, [1], [2]). In this paper we focus on a specific and simple model of a distributed sensor fusion problem, where the common goal is linear parameter estimation, and propose an algorithm for robust distributed sensor fusion based on average consensus in the network.

A. Maximum-likelihood parameter estimation

We consider the estimation of a vector of unknown (but constant) parameters $\theta \in \mathbf{R}^m$ using a network of n distributed sensors. Each sensor makes a noisy vector measurement

$$y_i = A_i\theta + v_i, \qquad i = 1, \ldots, n,$$

where $y_i \in \mathbf{R}^{m_i}$, A_i is a known matrix that relates the unknown parameter to the ith sensor measurement, and v_i is noise modeled as a random variable. We assume v_i has zero mean and covariance matrix Σ_i, and that the noises v_i are independent. We will also assume that the sensor noises are (jointly) Gaussian.

The aggregate measurement of all sensors is

$$y = A\theta + v$$

where

$$y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}, \quad A = \begin{bmatrix} A_1 \\ A_2 \\ \vdots \\ A_n \end{bmatrix}, \quad v = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix},$$

and the covariance matrix of v is $\Sigma = \mathbf{diag}(\Sigma_1, \ldots, \Sigma_n)$.

We assume that $m \leq \sum_{i=1}^n m_i$ and the matrix A is full rank. The maximum-likelihood (ML) estimate of θ, given the measurements $y_1, \ldots, y_n$, is the weighted least-squares (WLS) approximate solution

$$\begin{aligned} \hat{\theta}_{\mathrm{ML}} &= \left(A^T\Sigma^{-1}A\right)^{-1} A^T\Sigma^{-1}y \\ &= \left(\sum_{i=1}^n A_i^T\Sigma_i^{-1}A_i\right)^{-1} \sum_{i=1}^n A_i^T\Sigma_i^{-1}y_i. \end{aligned} \tag{1}$$

This estimate is unbiased (*i.e.*, $\mathbf{E}\ \hat{\theta}_{\mathrm{ML}} = \theta$) and has error covariance matrix

$$Q = \mathbf{E}\left((\hat{\theta}_{\mathrm{ML}} - \theta)(\hat{\theta}_{\mathrm{ML}} - \theta)^T\right) = \left(A^T\Sigma^{-1}A\right)^{-1}. \tag{2}$$

More generally, if the noises are not Gaussian, but are independent, and have zero mean and covariances Σ_i, the formula (1) gives the linear minimum-variance unbiased estimate of θ, given the measurements.

B. Sensor fusion schemes

In a centralized sensor fusion scheme, each sensor sends its data (y_i, A_i and Σ_i) either directly, or by multi-hop relay, to a data fusion center, typically via wireless communication. The fusion center then solves the WLS problem to find $\hat{\theta}_{\mathrm{ML}}$ as in (1). In the multi-hop relay case, each node must establish and maintain a routing table for the data packets to reach the fusion center (for example, by finding a minimum spanning tree rooted at the fusion center). This is extremely challenging if the communication graph (network topology) changes with time due to mobility and/or power constraints of the sensors.

In a distributed sensor fusion scheme, there is no central fusion center and the sensor nodes do not have any global knowledge of the network topology. Each sensor only exchanges data with its neighbors and carries out local computation. The goal

0-7803-9202-7/05/$20.00 (C) 2005 IEEE

is for each sensor to eventually have a good estimate of the unknown parameters, *e.g.*, to obtain $\hat{\theta}_{\rm ML}$. This is particularly important if the sensor nodes are carrying out multiple tasks and need $\hat{\theta}_{\rm ML}$ to make local on-site decisions.

There are many ways to do distributed sensor fusion. One straightforward method is flooding. Each sensor node broadcasts all its stored data (*i.e.*, y_i, A_i and Σ_i for some subset of the sensor nodes) to its neighbors, and stores all new data received. Eventually each node has all the data in the network, and thus can act as a fusion center to obtain $\hat{\theta}_{\rm ML}$. This method can require a large amount of data communication, storage memory, and book-keeping overhead. Much more sophisticated algorithms for distributed detection, estimation and inference in sensor networks have been studied in, *e.g.*, [3], [4], [5], [6], [7].

In this paper we propose a simple distributed, iterative scheme to compute $\hat{\theta}_{\rm ML}$ at each sensor node based on average consensus over the network (see, *e.g.*, [8], [9], [10], [11]). In our scheme, each sensor node maintains a fixed and small storage memory (same data structure for all nodes), and exchanges information with its neighbors in a very simple and isotropic manner. This scheme doesn't involve routing messages in the network; instead, it diffuses information across the network by updating each node's data with a weighted average of its neighbors'. At each step, every node can compute a local weighted least-square estimate, which eventually converges to the global maximum-likelihood solution.

In a network with unreliable communication links, each link may work or fail at random (*e.g.*, due to mobility, fading or power constraints). The instantaneous communication graph is time varying, and may be disconnected most of the time. Nevertheless, we show that our scheme works under a mild condition, roughly speaking, that the network should be connected in the long run.

C. Notation and outline

We model the topology of a sensor network by an undirected graph — the communication graph. Let $\mathcal{G} = (\mathcal{E}, \mathcal{V})$ denote an undirected graph with vertex set $\mathcal{V} = \{1, 2, \ldots, n\}$ and edge set $\mathcal{E} \subset \{\{i, j\} \mid i, j \in \mathcal{V}\}$, where each edge $\{i, j\}$ is an unordered pair of distinct nodes. A graph is *connected* if for any two vertices i and j there exists a sequence of edges (a path) $\{i, k_1\}, \{k_1, k_2\}, \ldots, \{k_{s-1}, k_s\}, \{k_s, j\}$ in $\mathcal{E}$.

We represent the time-varying communication graph of a sensor network by $\mathcal{G}(t) = (\mathcal{E}(t), \mathcal{V})$, where $\mathcal{E}(t)$ is the set of active edges at time t. Let $\mathcal{N}_i(t) = \{j \in \mathcal{V} \mid \{i, j\} \in \mathcal{E}(t)\}$ denote the set of neighbors of node i at time t, and $d_i(t) = |\mathcal{N}_i(t)|$ denote the degree (number of neighbors) of node i at time t. In this paper, the sequence of communication graphs $\{\mathcal{G}(t)\}_{t=0}^{\infty}$ can be either deterministic or stochastic.

Let $\mathcal{G}_i = (\mathcal{E}_i, \mathcal{V})$, $i = 1, \ldots, r$, denote a finite collection of graphs with common vertex set $\mathcal{V}$. Their *union* is a graph $\mathcal{G}$ with the same vertex set and a edge set that is the union of the $\mathcal{E}_i$'s; *i.e.*, $\mathcal{G} = \cup_{i=1}^{r} \mathcal{G}_i = (\cup_{i=1}^{r} \mathcal{E}_i, \mathcal{V})$. The set of graphs $\{\mathcal{G}_1, \ldots, \mathcal{G}_r\}$ is called *jointly connected* if their union is a connected graph [12].

This paper is organized as follows. In §II, we describe our scheme for a scalar case of the ML parameter estimation problem. This case corresponds to distributed average consensus of the scalar measurements over the network, and we give two simple rules for selecting edge weights for distributed averaging: the maximum-degree weights and the Metropolis weights. In §III, we show that both of these two weight selection methods guarantee convergence of the average consensus provided that the infinitely occurring communication graphs (there are only finitely many of them) are jointly connected. In §IV, we explain how distributed average consensus can be used in computing the ML estimate in the general setup described in §I-A. Finally, we demonstrate the proposed scheme with numerical examples in §V, and discuss some interesting extensions in §VI.

II. Distributed average consensus

We first explain our method for the following special case:

$$y_i = \theta + v_i, \qquad i = 1, \ldots, n,$$

where θ is a scalar to be estimated, and the noises are i.i.d. Gaussian: $v_i \sim \mathcal{N}(0, \sigma^2)$. In this case we have

$$\hat{\theta}_{\rm ML} = \frac{1}{n} \mathbf{1}^T y,$$

where $\mathbf{1}$ denotes the vector with all components one. In other words, the ML estimate is the average of the measurements y_i at all the sensors. The associated mean-square error is σ^2/n.

We use a distributed linear iterative method to compute the average. At $t = 0$ (after all sensors have taken the measurement), each node initializes its state as $x_i(0) = y_i$. At each following step, each node updates its state with a linear combination of its own state and the states at its instantaneous neighbors (nodes that belong to $\mathcal{N}_i(t)$)

$$x_i(t+1) = W_{ii}(t) x_i(t) + \sum_{j \in \mathcal{N}_i(t)} W_{ij}(t) x_j(t), \quad i = 1, \ldots, n. \tag{3}$$

Here $W_{ij}(t)$ is the linear *weight* on $x_j(t)$ at node i. Setting $W_{ij}(t) = 0$ for $j \notin \mathcal{N}_i(t)$, the above distributed iterative process can be written in vector form as

$$x(t+1) = W(t) x(t), \tag{4}$$

with the initial condition $x(0) = y$. Note that the weight matrix $W(t) \in \mathbf{R}^{n \times n}$ has the sparsity pattern specified by the communication graph $\mathcal{G}(t)$.

With the definition of a t-step transition matrix

$$\Phi(t) = W(t-1) \cdots W(1) W(0),$$

we have

$$x(t) = \Phi(t) x(0).$$

We would like to choose the weight matrices $W(t)$ such that the states at all the nodes converge to $\hat{\theta}_{\rm ML} = (1/n)\mathbf{1}^T y$, *i.e.*,

$$\lim_{t \to \infty} x(t) = \hat{\theta}_{\rm ML} \mathbf{1} = \left(\frac{1}{n} \mathbf{1}^T x(0) \right) \mathbf{1}.$$

Since this should hold for any $x(0) \in \mathbf{R}^n$, it is equivalent to

$$\lim_{t\to\infty} \Phi(t) = \frac{1}{n}\mathbf{1}\mathbf{1}^T. \tag{5}$$

A. Choice of weights

Although we are interested here in time-varying graphs, it is interesting to first consider the case when the communication graph is fixed, and we use a time-invariant weight matrix W. In this case, the following conditions are necessary and sufficient for (5) (in this case $\Phi(t) = W^t$)

$$\mathbf{1}^T W = \mathbf{1}^T, \quad W\mathbf{1} = \mathbf{1}, \quad \rho(W - \mathbf{1}\mathbf{1}^T/n) < 1, \tag{6}$$

where $\rho(\cdot)$ denote the spectral radius of a matrix; see [8], [9]. The asymptotic convergence rate of the average consensus, defined as

$$r_{\text{asym}}(W) = \sup_{x(0)\neq\theta_{\text{ML}}\mathbf{1}} \lim_{t\to\infty} \left(\frac{\|x(t) - \theta_{\text{ML}}\mathbf{1}\|}{\|x(0) - \theta_{\text{ML}}\mathbf{1}\|}\right)^{1/t}, \tag{7}$$

is precisely $\rho(W - \mathbf{1}\mathbf{1}^T/n)$. In other words, it is the second-largest eigenvalue modulus of W. Moreover, as shown in [8], for a fixed graph, we can find the symmetric weight matrix that has the smallest asymptotic convergence rate using semidefinite programming.

In the time-varying case, we determine the weight matrix at each step as a function of the instantaneous communication graph. In particular, we focus on the following two simple rules for choosing the weights:

- **Maximum-degree weights.** Here we use the constant weight $1/n$ on all the edges, and choose the self-weights so that the sum of weights at each node is 1:

$$W_{ij}(t) = \begin{cases} \dfrac{1}{n} & \text{if } \{i,j\} \in \mathcal{E}(t), \\ 1 - \dfrac{d_i(t)}{n} & \text{if } i = j, \\ 0 & \text{otherwise.} \end{cases} \tag{8}$$

- **Metropolis weights.** The Metropolis weight matrix is defined as

$$W_{ij}(t) = \begin{cases} \dfrac{1}{1+\max\{d_i(t), d_j(t)\}} & \text{if } \{i,j\} \in \mathcal{E}(t), \\ 1 - \displaystyle\sum_{\{i,k\}\in\mathcal{E}(t)} W_{ik}(t) & \text{if } i = j, \\ 0 & \text{otherwise.} \end{cases} \tag{9}$$

  With Metropolis weights, the weight on each edge is one over one plus the larger degree at its two incident vertices, and the self-weights are chosen so the sum of weights at each node is 1.

These two rules are adapted from methods for constructing fast mixing Markov chains on a graph (see, *e.g.*, [13]). In particular, these weight matrices are symmetric and (doubly) stochastic, so their eigenvalues are real and no more than 1 in absolute value. For a fixed communication graph, both the maximum-degree and Metropolis weights satisfy the conditions in (6) if the graph is connected. In §III we will show that in the time-varying case, these two rules for choosing weights lead to the convergence (5), provided that the infinitely occurring communication graphs are jointed connected.

We note that the maximum-degree and Metropolis weights defined above are not quite the same as the weights defined in the traditional Markov chain literature: Here we have an extra 1 added to the denominators for the edge weights. (In the maximum-degree case, the maximum-degree is $n - 1$.) This modification guarantees that the diagonal entries W_{ii} are always strictly positive, which will be needed in §III to prove convergence.

The Metropolis weights are well suited for distributed implementation, since each node only needs to know the degrees of its neighbors to determine the weights. The nodes do not need any global knowledge of the communication graph, or even the number of nodes n. For the maximum-degree weights, each node has to know n, or at least an upper bound of n (replacing n with an upper bound will also give weights that guarantee convergence to the average).

III. Convergence of average consensus

In this section, we show that both the maximum-degree and Metropolis weights guarantee convergence of the average consensus provided that the time-varying communication graphs are "connected in a long run"; more precisely, if the infinitely occurring communication graphs are jointly connected (see definition in §I-C).

For a network of n nodes, there are only finitely many, say a total of r, possible communication graphs. We denote the set of all possible graphs by $\{\mathcal{G}_1, \ldots, \mathcal{G}_r\}$, and the set of corresponding weight matrices by $\{W_1, \ldots, W_r\}$ (they are determined by either the maximum-degree rule or the Metropolis rule). Then equation (4) can be written as

$$x(t+1) = W_{i(t)}x(t) \tag{10}$$

where the indices $i(t)$ are integers and satisfy $1 \le i(t) \le r$ for all t. The sequence $\{i(t)\}_{t=0}^{\infty}$ can be either deterministic or stochastic. In any case, there is a subset of the graphs (equivalently a subset of the indices $\{1, 2, \ldots, r\}$) that occur infinitely often in the sequence. We have the following convergence theorem:

Theorem 1: If the collection of communication graphs that occur infinitely often are jointly connected, then the iteration process (10) converges with either the maximum-degree weights or the Metropolis weights, and

$$\lim_{t\to\infty} x(t) = \left(\frac{1}{n}\mathbf{1}^T x(0)\right)\mathbf{1}$$

for all $x(0) \in \mathbf{R}^n$. Equivalently, $\lim_{t\to\infty}\Phi(t) = (1/n)\mathbf{1}\mathbf{1}^T$.

This theorem requires a very weak condition on the long-term connectivity of the graph (*cf.* [12], [14], [15]). In particular, it does not require each link to be active for an infinite amount of time. In fact, many links can fail permanently. The only requirement is that the surviving links (those active infinitely often) make a connected graph. For example, it suffices for the surviving links to make a spanning tree of the

original arbitrary graph. In this sense, the distributed average consensus is quite robust to temporary and permanent link failures.

The proof of this theorem is based on a convergence result for nonhomogeneous infinite products of *paracontracting matrices* [16], which we will explain next. (Theorem 1 will be proved in §III-B.)

A. Infinite products of paracontracting matrices

The concept of paracontracting matrices was introduced in [17]. A matrix $M \in \mathbf{R}^{n\times n}$ is called *paracontracting* with respect to a vector norm $\|\cdot\|$ if

$$Mx \neq x \quad \Longleftrightarrow \quad \|Mx\| < \|x\|. \tag{11}$$

It is clear that a symmetric matrix is paracontracting with respect to the Euclidean norm if and only if all its eigenvalues lie in the interval $(-1, 1]$.

For a paracontracting matrix M, let $\mathcal{H}(M)$ denote its fixed-point subspace, *i.e.*, its eigenspace associated with the eigenvalue 1,

$$\mathcal{H}(M) = \{x \mid x \in \mathbf{R}^n,\ Mx = x\}.$$

The following is a key result in [16]. (See also the book [18], which puts together much of the basic work on the convergence of nonhomogeneous matrix products.)

Theorem 2 ([16]): Suppose that a finite set of square matrices $\{W_1, \ldots, W_r\}$ are paracontracting. Let $\{i(t)\}_{t=0}^{\infty}$, with $1 \leq i(t) \leq r$, be a sequence of integers, and denote by $\mathcal{J}$ the set of all integers that appear infinitely often in the sequence. Then for all $x(0) \in \mathbf{R}^n$ the sequence of vectors $x(t+1) = W_{i(t)}x(t)$, $t \geq 0$, has a limit $x^* \in \bigcap_{i\in\mathcal{J}} \mathcal{H}(W_i)$.

Intuitively, each paracontracting matrix preserves vectors in its fixed-point subspace, and is contractive for all other vectors. If some paracontracting matrices occur infinitely often in the iterative process, then the limit can only be in the intersection of their fixed-point subspace.

B. Proof of Theorem 1

To use the result of Theorem 2, we need to prove the following two lemmas.

Lemma 1: For any graph, both the maximum-degree weight matrix and the Metropolis weight matrix are paracontracting with respect to the Euclidean norm.

Proof: By the definitions (8) and (9), both the maximum-degree and the Metropolis weight matrices are symmetric stochastic matrices, so they are valid transition probability matrices for a Markov chain on the graph; see, *e.g.*, [13]. This implies that their eigenvalues are all real and lie in the interval $[-1, 1]$.

The two definitions also guarantee that the diagonal entries W_{ii} are all strictly positive. (see the comments in §II-A). This means that the corresponding Markov chains are aperiodic, which is equivalent to say that -1 cannot be an eigenvalue. Thus all the eigenvalues lie in the interval $(-1, 1]$, and the weight matrices are paracontracting with respect to the Euclidean norm. ■

Lemma 2: If a collection of graphs $\{\mathcal{G}_1, \ldots, \mathcal{G}_p\}$ are jointly connected, then their corresponding Metropolis weight matrices (or the maximum-degree weight matrices) satisfy

$$\bigcap_{i=1}^{p} \mathcal{H}(W_i) = \mathrm{span}\{\mathbf{1}\}. \tag{12}$$

Proof: By definition (9), the Metropolis weight matrices are stochastic, so we have $\mathbf{1} \in \mathcal{H}(W_i)$ for $i = 1, \ldots, p$. Therefore

$$\mathrm{span}\{\mathbf{1}\} \subset \bigcap_{i=1}^{p} \mathcal{H}(W_i). \tag{13}$$

Notice that if $W_i x = x$ for $i = 1, \ldots, p$, then we have $\left(\frac{1}{p}\sum_{i=1}^{p} W_i\right) x = x$. Therefore

$$\bigcap_{i=1}^{p} \mathcal{H}(W_i) \subset \mathcal{H}\left(\frac{1}{p}\sum_{i=1}^{p} W_i\right). \tag{14}$$

But the matrix $\frac{1}{p}\sum_{i=1}^{p} W_i$ is symmetric, stochastic and irreducible (since the underlying graphs are jointly connected), which implies that $\mathbf{1}$ is its only eigenvector (up to scaling) associated with the eigenvalue 1 (see, *e.g.*, [19]). In other words,

$$\mathcal{H}\left(\frac{1}{p}\sum_{i=1}^{p} W_i\right) = \mathrm{span}\{\mathbf{1}\}. \tag{15}$$

Putting the equations (13), (14) and (15) together, we get the desired result (12). It is clear that the same result holds for the maximum-degree weights defined in (8). ■

Now we are ready to prove Theorem 1. Without loss of generality, let $\{\mathcal{G}_1, \ldots, \mathcal{G}_p\}$ ($p \leq r$) be the set of infinitely occurring communication graphs, and assume that they are jointly connected. It is clear that the set of all communication graphs $\{\mathcal{G}_1, \ldots, \mathcal{G}_r\}$ are also jointly connected. By Lemma 2, we have

$$\bigcap_{i=1}^{r} \mathcal{H}(W_i) = \bigcap_{i=1}^{p} \mathcal{H}(W_i) = \mathrm{span}\{\mathbf{1}\}.$$

We write the initial vector $x(0) = u(0) + w(0)$, where $u(0)$ is the orthogonal projection of $x(0)$ onto $\mathrm{span}\{\mathbf{1}\}$, *i.e.*,

$$u(0) = \left(\frac{1}{n}\mathbf{1}^T x(0)\right)\mathbf{1},$$

and $w(0) \in \mathbf{1}^{\perp} = \{z \in \mathbf{R}^n \mid \mathbf{1}^T z = 0\}$. Because $u(0) \in \cap_{i=1}^{p}\mathcal{H}(W_i)$, the sequence given by $u(t+1) = W_{i(t)}u(t)$ is constant, and we have the limit $u^* = u(0)$. On the other hand, because $\mathbf{1}^{\perp}$ is invariant under all $W_{i(t)}$, the sequence of vectors given by $w(t+1) = W_{i(t)}w(t)$ all belong to the subspace $\mathbf{1}^{\perp}$, and so is the limit w^*. But by Theorem 2 we have $w^* \in \mathrm{span}\{\mathbf{1}\}$. So the only possibility is $w^* = 0$. We conclude that the iterative process (10) has the limit

$$x^* = u^* + w^* = \left(\frac{1}{n}\mathbf{1}^T x(0)\right)\mathbf{1}.$$

This finishes the proof for Theorem 1.

C. Remarks

Because of the very weak condition we require on the long-term connectivity, in general there may not exist a linear convergence rate, as defined in (7) for the case of fixed graph. To have linear convergence, we have to impose more conditions on how often joint connectedness occurs. For example, if there exists an integer $T > 0$ such that the set of graphs $\{\mathcal{G}(\tau) \mid t \leq \tau < t+T\}$ are jointly connected for all t, then it can be shown that the sequence $x(t)$ converges geometrically (linear convergence). For detailed analysis of this kind, see, *e.g.*, [18] and references therein.

The paper [12] studied the convergence property of the following *nearest neighbor rule* for coordination over time-varying communication graphs

$$x_i(t+1) = \frac{1}{1+|\mathcal{N}_i(t)|}\left(x_i(t) + \sum_{j\in\mathcal{N}_i(t)} x_j(t)\right)$$

for $i = 1, \ldots, n$. (See also [14], [15].) Note that the associated weight matrices, though (row) stochastic, are not symmetric. It was shown that the sequence $\{x(t)\}_{t=0}^{\infty}$ converges to a vector in $\mathrm{span}\{\mathbf{1}\}$ if there exists an infinite sequence of contiguous, non-empty, bounded time intervals such that across each such interval the graphs are jointly connected (a stronger assumption than ours). However, the limit vector usually is not the orthogonal projection $(\mathbf{1}^T x(0)/n)\mathbf{1}$. It depends on $x(0)$ as well as the switching sequence $\{i(t)\}_{t=0}^{\infty}$. For this reason, the nearest neighbor rule cannot be used for average consensus.

IV. Distributed sensor fusion based on average consensus

Now we explain how distributed average consensus can be used to compute $\hat{\theta}_{\mathrm{ML}}$ in the general setup of §I-A. Throughout this section, we assume that the collection of infinitely occurring communication graphs are jointly connected.

A. The distributed sensor fusion scheme

In this scheme, each sensor node maintains a local *composite information matrix* $P_i(t) \in \mathbf{R}^{m\times m}$ and a local *composite information state* $q_i(t) \in \mathbf{R}^m$, initialized at $t = 0$ as

$$\begin{aligned} P_i(0) &= A_i^T \Sigma_i^{-1} A_i, \\ q_i(0) &= A_i^T \Sigma_i^{-1} y_i, \end{aligned}$$

for $i = 1, \ldots, n$. Then the nodes conduct average consensus *entry-wise* for the matrices $P_i(0)$ and vectors $q_i(0)$, *i.e.*,

$$\begin{aligned} P_i(t+1) &= W_{ii}(t)P_i(t) + \sum_{j\in\mathcal{N}_i(t)} W_{ij}(t)P_j(t), \\ q_i(t+1) &= W_{ii}(t)q_i(t) + \sum_{j\in\mathcal{N}_i(t)} W_{ij}(t)q_j(t). \end{aligned}$$

Here we can use either the maximum-degree weights or the Metropolis weights.

By assumption, the communication graphs that occur infinitely often are jointly connected, so we have the following convergence result (by Theorem 1)

$$\begin{aligned} \lim_{t\to\infty} P_i(t) &= (1/n)\sum_{i=1}^{n} A_i^T \Sigma_i^{-1} A_i, \\ \lim_{t\to\infty} q_i(t) &= (1/n)\sum_{i=1}^{n} A_i^T \Sigma_i^{-1} y_i. \end{aligned}$$

Therefore, each node asymptotically computes the ML estimate via

$$\hat{\theta}_{\mathrm{ML}} = \lim_{t\to\infty} P_i(t)^{-1} q_i(t), \qquad i = 1, \ldots, n.$$

This scheme doesn't involve explicit point-to-point message passing or routing. Instead, it diffuses information in the network by updating each node's data with a weighted average of its neighbors'. It has the following properties.

- *Universal data structure.* All the nodes maintain the same, fixed data structure: $P_i(t) \in \mathbf{R}^{m\times m}$ and $q_i(t) \in \mathbf{R}^m$, which are independent of local dimensions m_i.
- *Isotropic protocol.* All the nodes communicate with their instantaneous neighbors, and use the same rule to choose averaging weights in computation.
- *robust to link failures and topology changes.* This scheme works in a network with unreliable communication links, provided the infinitely occurring communication graphs are jointly connected.

Other distributed algorithms have been proposed to solve the normal equation

$$\left(\sum_{i=1}^{n} A_i^T \Sigma_i^{-1} A_i\right)\hat{\theta}_{\mathrm{ML}} = \sum_{i=1}^{n} A_i^T \Sigma_i^{-1} y_i.$$

directly by either iterative methods such as block Gauss-Seidel iteration [4], or by Gauss elimination [20]. However, they may not enjoy all the properties above. More importantly, our scheme offers reasonable intermediate estimates without waiting for the asymptotic convergence, as analyzed next.

B. Properties of intermediate estimates

The true ML estimate is found at each node only in the limit as $t \to \infty$. In this section we study the properties of the intermediate estimates,

$$\hat{\theta}_i(t) = P_i(t)^{-1} q_i(t), \qquad i = 1, \ldots, n. \tag{16}$$

(This is available at node i as soon as $P_i(t)$ is invertible.) We show that they have some satisfying properties to serve as intermediate estimates.

First, let's examine the expressions for the composite information matrices and composite information states

$$\begin{aligned} P_i(t) &= \sum_{j=1}^{n} \Phi_{ij}(t) A_j^T \Sigma_j^{-1} A_j, \\ q_i(t) &= \sum_{j=1}^{n} \Phi_{ij}(t) A_j^T \Sigma_j^{-1} y_j. \end{aligned}$$

Thus at each step, each node has a *block-scaled* WLS solution, where the scaling factors are the entries of $\Phi(t)$. As $t \to \infty$, the matrix $\Phi(t)$ converges to $(1/n)\mathbf{1}\mathbf{1}^T$, *i.e.*, all the scaling factors $\Phi_{ij}(t)$ converge to $1/n$. Therefore the estimate $\hat{\theta}_i(t)$ converges to $\hat{\theta}_{\rm ML}$ for every $i = 1, \ldots, n$. Furthermore, we have the following theorem.

Theorem 3: Assume that the communication graph $\mathcal{G}(t)$ is a random variable, and it is independent of the measurement noise v. Then all the intermediate estimates $\hat{\theta}_i(t)$ are *unbiased*, *i.e.*,

$$\mathbf{E}\,\hat{\theta}_i(t) = \theta, \qquad i = 1, \ldots, n \tag{17}$$

whenever $\hat{\theta}_i(t)$ is defined (*i.e.*, when $P_i(t)$ is invertible); and the error covariance matrix at each node converges to that of the global ML solution, *i.e.*,

$$\lim_{t\to\infty} \mathbf{E}\left[(\hat{\theta}_i(t) - \theta)(\hat{\theta}_i(t) - \theta)^T\right] = \left(A^T\Sigma^{-1}A\right)^{-1}. \tag{18}$$

Here the expectation is taken with respect to both the measurement noise v and the sequence of random graphs $\{\mathcal{G}(t)\}_{t=0}^{\infty}$. (The above results also hold if the sequence of time-varying communication graphs is deterministic.)

Proof: Whenever $P_i(t)$ is invertible, we have

$$\begin{aligned}
\mathbf{E}\,\hat{\theta}_i(t) &= \mathbf{E}\,P_i(t)^{-1}q_i(t) \\
&= \mathbf{E}\,P_i(t)^{-1}\left(\sum_{j=1}^{n}\Phi_{ij}(t)A_j^T\Sigma_j^{-1}(A_j\theta + v_j)\right) \\
&= \theta + \mathbf{E}\,P_i(t)^{-1}\left(\sum_{j=1}^{n}\Phi_{ij}(t)A_j^T\Sigma_j^{-1}v_j\right) \\
&= \theta + \sum_{j=1}^{n}\mathbf{E}\left(\Phi_{ij}(t)P_i(t)^{-1}A_j^T\Sigma_j^{-1}\right)\mathbf{E}v_j \\
&= \theta.
\end{aligned}$$

In the second-to-last equality, we used the fact that $P_i(t)$ and $\Phi(t)$ only depend on $\{\mathcal{G}(s)\}_{s=0}^{t-1}$, which is independent of v. To obtain the last equality, we used the assumption that $\mathbf{E}v_j = 0$ for all j.

Unlike the expectation of $\hat{\theta}_i(t)$, the error covariance matrices do depend on the distribution of the random communication graphs. We consider the covariance matrix conditioned on a particular sequence $\{\mathcal{G}(s)\}_{s=0}^{t-1}$

$$\begin{aligned}
Q_i(t) &= \mathbf{E}\left[(\hat{\theta}_i(t) - \theta)(\hat{\theta}_i(t) - \theta)^T \;\middle|\; \{\mathcal{G}(s)\}_{s=0}^{t-1}\right] \\
&= P_i(t)^{-1}\left(\sum_{j=1}^{n}\Phi_{ij}^2(t)A_j^T\Sigma_j^{-1}A_j\right)P_i(t)^{-1}.
\end{aligned}$$

As t increases, the coefficients $\Phi_{ij}(t)$ all converge to $1/n$, and the error covariance matrices $Q_i(t)$ depend less and less on the particular sequence of graphs. In the limit, we have

$$\lim_{t\to\infty} Q_i(t) = \left(\sum_{j=1}^{n}A_j^T\Sigma_j^{-1}A_j\right)^{-1} = \left(A^T\Sigma^{-1}A\right)^{-1},$$

which is precisely the error covariance for the ML estimate, *i.e.*, the matrix Q in (2). Since this limit is independent of the sequence $\{\mathcal{G}(s)\}_{s=0}^{\infty}$, we have proved (18). ■

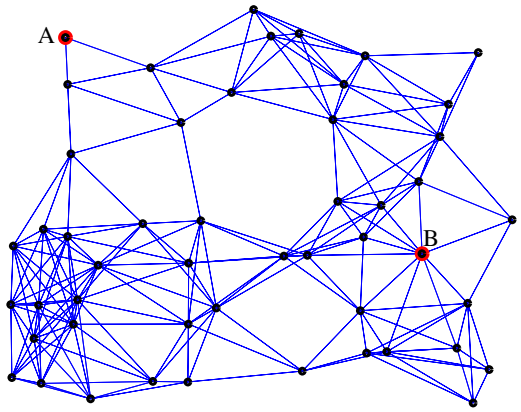

Fig. 1. A randomly generated sensor network with 50 nodes and 200 links.

In summary, at each step, every node can compute a local, unbiased, weighted least-square estimate $\hat{\theta}_i(t)$ whenever $P_i(t)$ is invertible. All such intermediate estimates converge to the global maximum-likelihood solution, and so do the corresponding error covariance matrices.

One key feature of this scheme is that the sensors can always use the intermediate estimates, whenever they are queried, before waiting for the asymptotic convergence of the estimation covariance. This is in contrast to those iterative estimation schemes that requires final convergence to compute the best estimate. There is always a trade-off between variance of the estimate and the number of iterations. Intermediate estimates after a few iterations tend to weight heavily on measurement data of nearby sensors; better and better global picture is revealed as the iterative process goes on.

V. Numerical Example

We demonstrate our method with a randomly generated numerical example. Consider a sensor network whose communication graph is shown in Figure 1. This graph is generated as follows: we first randomly generate $n = 50$ sensor nodes, uniformly distributed on the unit square $[0, 1] \times [0, 1]$; then connect two nodes by an edge if their distance is less than $1/4$. In the particular setup of Figure 1, there are 200 edges. The vector of unknown parameters, θ, has dimension $m = 5$. Each sensor takes a scalar measurement $y_i = a_i^T\theta + v_i$, where the vectors a_i were chosen from a uniform distribution on the unit sphere in $\mathbf{R}^5$, and the noises are i.i.d. Gaussian with unit variance: $v_i \sim \mathcal{N}(0, 1)$.

In our first experiment, we fixed the sensor network and simulated the proposed sensor fusion scheme, with both the max-degree weights and the Metropolis weights. We picked two nodes, A and B (with degrees 2 and 10 respectively, labeled in Figure 1), to plot their mean-square errors (MSE) $\mathbf{E}\|\hat{\theta}_i(t) - \theta\|^2 = \text{trace}(Q_i(t))$; see Figure 2. All the mean-square errors converge to the optimal ML estimation error. It can be seen that using the Metropolis weights leads to much faster convergence than the max-degree weights. (The

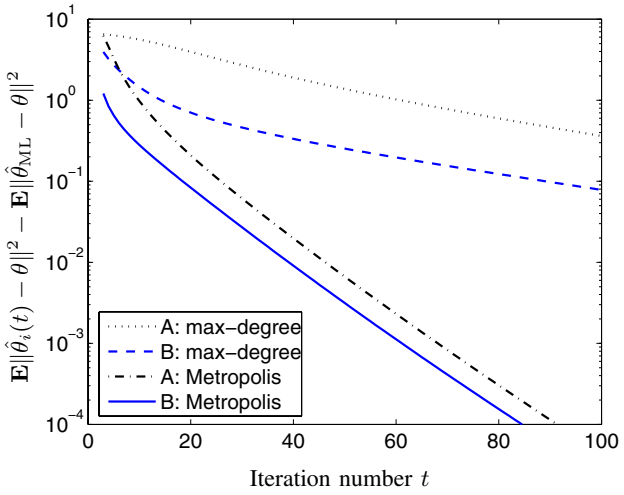

Fig. 2. MSE at nodes A and B: fixed communication graph.

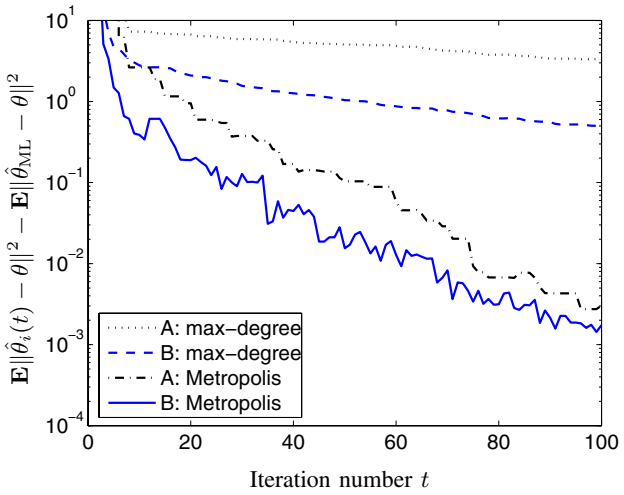

Fig. 3. MSE at nodes A and B: dynamically changing communication graph.

advantage of Metropolis weights becomes more pronounced as the number of nodes increases.) In either case, node B has slightly faster convergence than node A, because node B has a larger degree and thus has "better connection" in the network.

We also simulated the proposed sensor fusion scheme with dynamically changing communication graphs. We generated the sequence of communication graphs as follows: at each time step, each edge in the graph is only available with probability $1/4$, independent of other edges and all previous steps. The MSE at node A and B are plotted in Figure 3. In this case, the curves shown are conditioned on the particular sequence of graphs used in simulation. However, as we have shown in Theorem 3, they always converge to the same optimal ML estimation error.

VI. Discussions and Extensions

In this paper, we proposed a scheme for distributed sensor fusion based on average consensus. In this scheme, all the sensor nodes maintain a universal data structure: a composite information matrix and a composite information state whose sizes are independent of the local measurement dimensions. All the nodes use an isotropic protocol for communication and computation: they only communicate with their instantaneous neighbors, and use the same rule to determine the edge weights. At each step (whenever the composite information matrix is invertible), every node can compute a local, unbiased estimate, and all such intermediate estimates converge to the global ML solution.

This scheme doesn't involve explicit point-to-point message passing or routing; Instead, it diffuses information by updating each node's data with a weighted average of its neighbors'. Moreover, it guarantees convergence so long as the infinitely occurring communication graphs are jointly connected. Similar diffusive schemes have also been studied in distributed load balancing problems, *e.g.*, [21], [22]. The convergence result with unreliable communication links established in this paper applies directly to such problems.

A very interesting extension is to consider distributed data fusion with multiple measurements at each node, where the measurements are taken at different time steps and happen asynchronously in the network. In this scenario, we hope to develop a *space-time diffusion* scheme to do recursive least-squares estimation (the scheme in this paper can be considered *space-diffusion*). A further step would be trying to apply such diffusion schemes in distributed Kalman filter (or information filter); see [23] and the recent work [7].

The ML estimation problem considered in this paper can also be considered as a special case of distributed optimization in sensor networks. For example, incremental subgradient methods are used in [24] to solve such problems. In our case, the solution can be obtained by solving a set of linear equations (see (1)), and we used an iterative scheme based on average consensus to approximate the solution.

We have focused on a simple model which does not consider many important practical issues such as power efficiency, quantization of measurements, finite-bit communication, transmission errors and delays, and reliable communication protocols. As an effort in these directions, we will discuss the issue of finite-bit communication in some detail.

A. Average consensus with finite-bit communication

We assume that the storage and float-point computation at the sensor nodes use a much higher precision than the finite-bit communication between nodes.

One of the key properties of the iterative algorithm (4) is that the sum (and hence the average) is always preserved, *i.e.*, $\mathbf{1}^T x(t+1) = \mathbf{1}^T x(t)$, because $\mathbf{1}^T W(t) = \mathbf{1}^T$ holds for all the weight matrices we use. Since the weight matrices are stochastic, we can eliminate $W_{ii}(t)$ using

$$W_{ii}(t) = 1 - \sum_{j \in \mathcal{N}_i(t)} W_{ij}(t).$$

So we can rewrite (3) as

$$x_i(t+1) = x_i(t) + \sum_{j \in \mathcal{N}_i(t)} W_{ij}(t)(x_j(t) - x_i(t)). \quad (19)$$

Now it is clear that preserving the sum relies on the fact that the weights are symmetric, so the interchanges between two neighboring nodes always have the same modulus with different signs, *i.e.*,

$$W_{ij}(t)(x_j(t) - x_i(t)) = -W_{ji}(t)(x_i(t) - x_j(t)).$$

However, this may no longer be the case with finite-bit communication where the $x_i(t)$'s are always rounded first before transmission. This can be seen with the following example (we skip the time index to simplify notation). Let $x_i = 4.53$ and $x_j = 5.39$. Suppose we can only transmit two digits and use the usual rounding scheme. Then node i receives $\tilde{x}_j = 5.4$ and the corresponding modification is

$$W_{ij}(\tilde{x}_j - x_i) = 0.87W_{ij},$$

and node j receives $\tilde{x}_i = 4.5$ and the corresponding modification is

$$W_{ji}(\tilde{x}_i - x_j) = -0.89W_{ij}.$$

So the interchanges between the two nodes have different modulus, and the sum $\mathbf{1}^T x(t)$ across the network will not be preserved. This phenomenon will cause a drift in computing the average.

There is a simple protocol to fix this drifting problem. The idea is to let two neighboring nodes always agree on the same modulus of their interchange (with different signs). Suppose node i sends a rounded version $\tilde{x}_i(t)$ to its neighbor j. Node j computes the difference $x_j(t) - \tilde{x}_i(t)$, but uses a rounded version of this difference, $\Delta_{ji}(t)$, in computing its local value $x_j(t+1)$. At the same time, it sends $\Delta_{ij}(t) = -\Delta_{ji}(t)$ back to node i. This makes sure that two neighboring nodes always increase/decrease their data by the same modulus with opposite signs. This modified algorithm can be written as

$$x_i(t+1) = x_i(t) + \sum_{j \in \mathcal{N}_i(t)} W_{ij}(t)\Delta_{ij}(t). \qquad (20)$$

Since we have $\Delta_{ij}(t) = -\Delta_{ji}(t)$, it is clear that this algorithm always preserves the sum, and hence the average, of the x_i's in the network.

Of course, the algorithm (20) is a nonlinear iteration. It will only converge to a neighborhood of the average vector $(\mathbf{1}^T x(0)/n)\mathbf{1}$, and may have some complex limiting behavior. We hope to address this and other practical issues in future research.

ACKNOWLEDGMENT

This work was partially supported by the Stanford URI *Architectures for Secure and Robust Distributed Infrastructures*, AFOSR DoD award number 49620-01-1-0365.

REFERENCES

[1] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar, "Next century challenges: Scalable coordination in sensor networks," in *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking*, Seattle, Washington, USA, 1999, pp. 263–270.

[2] H. Gharavi and S. P. Kumar, Eds., *Proceedings of the IEEE: Special Issue on Sensor Networks and Applications*, vol. 91, IEEE, August 2003.

[3] J. N. Tsitsiklis, "Decentralized detection," in *Advances in Signal Processing*, H. V. Poor and J. B. Thomas, Eds., vol. 2, pp. 297–344. JAI Press, 1993.

[4] V. Delouille, R. Neelamani, and R. Baraniuk, "Robust distributed estimation in sensor networks using the embedded polygons algorithm," in *Proceedings of The Third International Conference on Information Processing in Sensor Networks*, Berkeley, California, USA, April 2004, pp. 405–413.

[5] M. Alanyali, S. Venkatesh, O. Savas, and S. Aeron, "Distributed Bayesian hypothesis testing in sensor networks," in *Proceedings of American Control Conference*, Boston, Massachusetts, June 2004, pp. 5369–5374.

[6] Z.-Q. Luo, "An isotropic universal decentralized estimation scheme for a bandwidth constrained ad hoc sensor network," *IEEE Journal on Selected Areas in Communications*, 2005, Special issue on Self-Organizing Distributed Collaborative Sensor Networks, to appear.

[7] D. P. Spanos, R. Olfati-Saber, and R. M. Murray, "Distributed Kalman filtering in sensor networks with quantifiable performance," in *Proceedings of The Fourth International Conference on Information Processing in Sensor Networks*, Los Angeles, California, USA, April 2005.

[8] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Systems and Control Letters*, vol. 53, pp. 65–78, 2004.

[9] D. S. Scherber and H. C. Papadopoulos, "Locally constructed algorithms for distributed computations in ad-hoc networks," in *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks*, Berkeley, CA, April 2004, pp. 11–19.

[10] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1520–1533, September 2004.

[11] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Gossip algorithms: design, analysis and applications," to appear in Proceedings of IEEE INFOCOM, 2005.

[12] A. Jadbabaie, J. Lin, and A. S. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Transactions Automatic Control*, vol. 48, no. 6, pp. 988–1001, June 2003.

[13] S. Boyd, P. Diaconis, and L. Xiao, "Fastest mixing Markov chain on a graph," *SIAM Review, problems and techniques section*, vol. 46, no. 4, pp. 667–689, 2004.

[14] L. Moreau, "Stability of multi-agent systems with time-dependent communication links," Accepted for publication in *IEEE Transactions on Automatic Control*, 2004.

[15] W. Ren and R. W. Beard, "Consensus seeking in multi-agent systems under dynamically changing interaction topologies," *IEEE Transactions on Automatic Control*, 2005, to appear.

[16] L. Elsner, I. Koltracht, and M. Neumann, "On the convergence of asynchronous paracontractions with applications to tomographic reconstruction from incomplete data," *Linear Algebra Appl.*, vol. 130, pp. 65–82, 1990.

[17] S. Nelson and M. Neumann, "Generalization of the projection method with applications to SOR theory for Hermitian positive semidefinite linear systems," *Numerische Mathematik*, vol. 51, pp. 123–141, 1987.

[18] D. J. Hartfiel, *Nonhomogeneous Matrix Products*, World Scientific Publishing Co. Pte. Ltd., Singapore, 2002.

[19] R. A. Horn and C. A. Johnson, *Matrix Analysis*, Cambridge University Press, 1985.

[20] K. Plarre and P. R. Kumar, "Extended message passing algorithm for inference in loopy gaussian graphical models," *Ad Hoc Networks*, vol. 2, pp. 153–169, 2004.

[21] G. Cybenko, "Load balancing for distributed memory multiprocessors," *Journal of Parallel and Distributed Computing*, vol. 7, pp. 279–301, 1989.

[22] R. Diekmann, S. Muthukrishnan, and M. V. Nayakkankuppam, "Engineering diffusive load balancing algorithms using experiments," in *Solving Irregularly Structured Problems in Parallel*, vol. 1253 of *Lecture Notes in Computer Science*, pp. 111–122. Springer-Verlag, Berlin, 1997.

[23] S. Grime and H. F. Durrant-Whyte, "Data fusion in decentralized sensor networks," *Control Engineering Practice*, vol. 2, no. 5, pp. 849–863, 1994.

[24] M. Rabbat and R. D. Nowak, "Distributed optimization in sensor networks," in *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks*, Berkeley, CA, April 2004, pp. 20–27.